Learning Manipulation Graphs from Demonstrations Using Multimodal Sensory Signals

Zhe Su[1,2], Oliver Kroemer[3,5], Gerald E. Loeb[4], Gaurav S. Sukhatme[3], Stefan Schaal[1,2]

***Abstract*— Complex contact manipulation tasks can be decomposed into sequences of motor primitives. Individual primitives often end with a distinct contact state, such as inserting a screwdriver tip into a screw head or loosening it through twisting. To achieve robust execution, the robot should be able to verify that the primitive's goal has been reached as well as disambiguate it from erroneous contact states. In this paper, we introduce and evaluate a framework to autonomously construct manipulation graphs from manipulation demonstrations. Our manipulation graphs include sequences of motor primitives for performing a manipulation task as well as corresponding contact state information. The sensory models for the contact states allow the robot to verify the goal of each motor primitive as well as detect erroneous contact changes. The proposed framework was experimentally evaluated on grasping, unscrewing, and insertion tasks on a Barrett arm and hand equipped with two BioTacs. The results of our experiments indicate that the learned manipulation graphs achieve more robust manipulation executions by confirming sensory goals as well as discovering and detecting novel failure modes.**

I. Introduction

Object manipulation tasks can be decomposed into sequences of discrete motor primitives. For instance, an assembly task like unscrewing a screw involves inserting the screwdriver tip into the head of the screw and twisting it. Each of these motor primitives terminates in a sensory event that corresponds to a sensorimotor subgoal of the task [1], e.g., making contact between the screwdriver tip and the head of the screw, and loosening the screw. In humans, these distinct sensory events have been characterized by specific neural responses in cutaneous sensory afferents on the fingertips [2]. For example, fast- and slow-adapting type one afferents (FA-I, SA-I) respond strongly to making or breaking contact as well as normal and tangential forces between fingertips and hand-held tools. When a tool makes contacts or slides against an object, the fast-adapting type two afferents (FA-II) sense the vibrations indicating the contact changes.

Klingbeil et.al [3] showed humans can perform manipulation tasks requiring complicated contact changes indirectly through tools and they achieve robust manipulation by spending significant amount of time at a few distinct types of contact states which often require exploratory strategies to disambiguate. It is desirable to equip robots with this capability. For example, a robot could learn to disambiguate between a successful insertion into a screw head versus making contacts with a flat surface, as shown in Fig. 1.

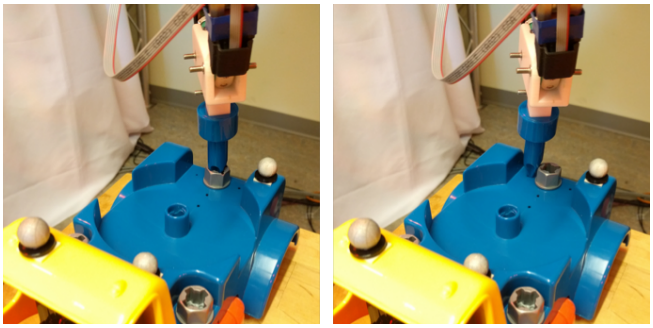

Fig. 1: Robot needs to learn to disambiguate successful insertion from failed insertion into a screw.

In our previous work [4], we proposed a method to segment demonstrated manipulation tasks into a sequence of sensorimotor primitives using unsupervised Bayesian online changepoint detection (BOCPD) [5] with multimodal haptic signals. In this paper, we expand our previous work into a manipulation skill acquisition framework by making the following improvements. First, correspondences of the segmented motor primitives from multiple demonstrations are found by clustering the final poses of all the segments extracted from BOCPD. After clustering these segments, skill clusters and frequency of transitions between clusters within demonstrations are used as nodes and edges to build a skill graph. The robot then performs the task by replaying a sequence of motor primitives by traversing the skill graph. A sequence of exploratory movements is performed at the end of each motor primitive execution. The resulting sensory signals, i.e. from tactile sensors in this work, are clustered to identify sensory events corresponding to distinct contact states, which we refer to as modes. These modes are formed from successful and failed skill executions. Finally, a unified manipulation graph is built with both the motor primitives and the modes, and the learned graph is used by the robot to verify successful skill executions by detecting contact state changes and discovering novel failures. The overall framework is shown in Fig. 2.

The proposed framework was evaluated on three manipulation tasks: a grasping task, an unscrewing task, and a peg insertion task (0.5mm tolerance). The experiments evaluated the robot on segmenting demonstrations, clustering segments from multiple demonstrations, and building manipulation

[1]Autonomous Motion Department, MPI-IS, Tübingen, Germany.
[2]CLMC-Lab, University of Southern California, Los Angeles, USA.
[3]RESL-Lab, University of Southern California, Los Angeles, USA.
[4]MDDF-Lab, University of Southern California, Los Angeles, USA.
[5]The Robotics Institute, Carnegie Mellon University, Pittsburgh, USA.

This research was supported in part by National Science Foundation grants IIS-1205249, IIS-1017134, EECS-0926052, the Office of Naval Research, the Okawa Foundation, and the Max-Planck-Society.

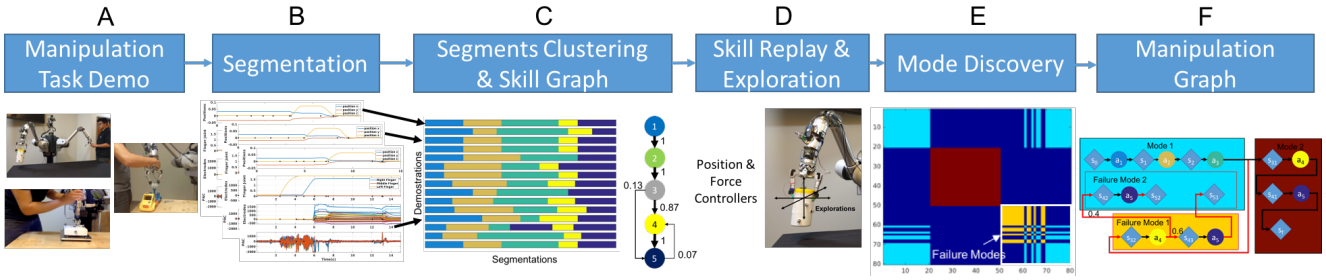

Fig. 2: Overview of the framework used in this experiment.

graphs as well as discovering novel failure cases.

II. Related Work

Imitation learning methods are an effective approach for transferring human manipulation skills to robots. These methods often learn motor primitive libraries that generalize between different contexts of the task [6, 7, 8]. The motor primitives are usually trained on presegmented trajectories, and they tend to terminate after a fixed duration or once they have reached a predefined pose threshold from their goal. Kappler et al [9] proposed a framework for using multimodal signals to switch between primitives. Their approach models the stereotypical sensor signals as functions of time rather than monitoring for specific sensory event of the primitive's goal and failures. Niekum et. al [10] learn a finite-state automaton or skill graph that select the next primitive based on the current state.

Methods for segmenting manipulations into sequences of primitives [10, 11, 12] usually use proprioceptive signals of the robot and the locations of the objects to segment the demonstrations. Konidaris et.al [13] used the returns from a Reinforcement Learning framework to segment demonstrated trajectories. Niekum et. al [14] have proposed an approximate online Bayesian changepoint detection method to segment demonstrations by detecting changes in articulated motion of objects. The authors also proposed verification tests to verify skills have been successfully executed before switching onto subsequent skills. We use low and high frequency tactile signals, inspired by human sensorimotor primitives [1], to detect contact events for segmentation as well as discovering modes corresponding to success and failure executions. It has been shown that high frequency tactile signals are particularly important for manipulation tasks [15, 16]. Recently, Chu et. al [17] have shown the importance of using multiple sensory modalities, such as force/torque sensing and vision, to improve skill segmentation. Other techniques have been proposed for decomposing tasks into modes based on changes in the state transition model [18, 19]. Motor primitives are subsequently optimized for switching between the modes.

In the planning domain, a contact manipulation task can be treated as a contact motion planning problem by dividing it into a sequence of contact state transitions which can be represented as connections in a graph [20]. However, the graph size grows combinatorially with the number of contact states. Lee et. al [21] proposed a hierarchical approach to decrease the search space by planning for three subproblems: finding sequences of object contact states, finding sequences of object's poses, and finding sequences of contact points for manipulators on the object. Jain et. al [22] proposed to solve contact manipulation tasks with a hierarchical POMDP motion planner that develops high-level discrete state plans to find sequences of local models to visit and a low-level cost-optimized continuous state belief-space plans. Previous work in motion planning [23, 24] has also created manipulation graphs with modes, which represent finite and discrete submanifolds of the full configuration space. In our paper, the modes are discovered through clustering the sensory signals after executing a sequence of skills and they correspond to different types of contact constraints formed at the end of these skill executions.

In [3], Klingbeil et.al developed a framework to analyze human control strategies while humans demonstrated complex contact manipulation tasks in a virtual environments with visual and haptic feedback. Their experiments showed that humans tend to explicitly control and explore only a few contact states along the manipulation trajectories due to physiological delay limits. This work seems to agree with our assumption that these few states correspond to the subgoals of the manipulation tasks and a robot should develop sensory models at these key contact states.

Guarded motions are primitives that terminate when a sensory condition is fulfilled. These primitives are widely used in industrial application and prosthetics to avoid excessive force [25, 26]. The termination conditions are usually manually predefined.

III. Approach

We present a framework for autonomously segmenting manipulations, clustering segments into skill primitives, and discovering corresponding modes to create a manipulation graph. The manipulation graph is learned from successful and failed executions during skill replays, and therefore also includes failure modes. The success and failure modes are subsequently learned for each skill primitive to determine when to switch to the next primitive and to detect when an error has occurred. We explain the segmentation of the demonstrations into primitives in Sec. III-B, finding corresponding segments among demonstrations to build skill graphs and removing oversegmentations in Sec. III-C,

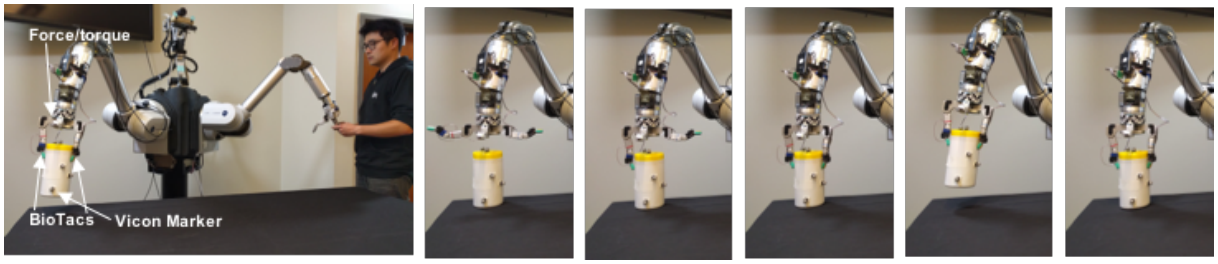

Fig. 3: Experimental setup of demonstrating the grasping task.

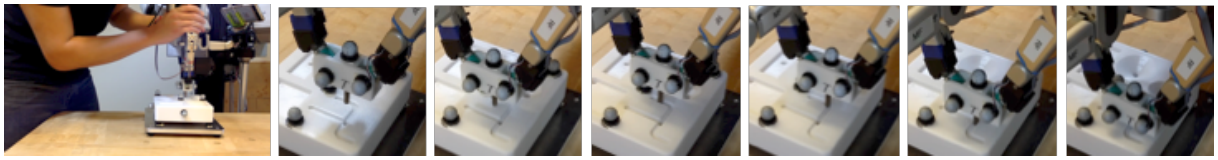

Fig. 4: Experimental setup of demonstrating the peg-in-hole task.

discovering unique sensory events associated with contact state changes at the end of each skill replay in Sec. III-D, and building manipulation graphs from skill graphs and the corresponding modes in Sec. III-E. An overview of our framework is shown in Fig. 2.

A. Demonstration and Multimodal Sensory Signals

The graph generation process is initialized from demonstrations, as shown in Fig. 2A. Our experimental setup consists of a 7-DOF Barrett WAM arm and a Barrett hand, which is equipped with two biomimetic tactile sensors (BioTacs) [27]. We demonstrate manipulation tasks through two types of demonstrations: kinesthetic demonstrations and teleoperated demonstrations. In the kinesthetic demonstrations, the human expert demonstrates tasks by directly moving the robot arm. In the teleoperated demonstration, the human operates the bi-manual robot by manually moving the robot's master arm where the slave arm mimics the movements of the master arm to manipulate the objects. More details can be found in Figs. 3–5.

Multimodal haptic signals, including proprioceptive signals and both low and high frequency tactile signals, are captured throughout human demonstration. The proprioceptive signals are the 6D Cartesian position and orientation of the robot's end-effector $\mathbf{y}_{\text{pos}} \in \mathbb{R}^6$ derived from the robot's forward kinematics. We also recorded the 6D Cartesian pose of the object in the robot's surroundings with a Vicon motion capture system $\mathbf{y}_{\text{obj}} \in \mathbb{R}^6$.

The low frequency tactile signals ($\leq$ 100Hz) are measured from an array of 19 impedance sensing electrodes that detect sensitive skin deformations. The electrode impedances $\mathbf{y}_{\text{E}} \in \mathbb{R}^{19}$ are sampled at 100Hz. For dynamic tactile sensing, high frequency vibration signals (10 - 1040Hz) are available from the hydro-acoustic pressure sensor. These vibration signals $\mathbf{y}_{\text{PAC}} \in \mathbb{R}^1$ are sampled at 2200Hz and often correspond to transient mechanical events, such as micro-vibrations between the sensor's skin and external environment. Detailed descriptions of the tactile signals can be found in [4].

B. Sensorimotor Primitive Segmentation

To discover primitives that terminate in distinct sensory events, the robot must segment demonstrations and skill executions according to the sensory signals. The tactile signals are particularly important for segmenting sensory trajectories into primitives with sensory goals [4, 15].

Unlike the relatively smooth proprioceptive signals, dynamic tactile sensor signals are highly sensitive to transient mechanical events. Some of these detected events correspond to the end of a contact state, at which point a new primitive should start, but others may be caused by noise. In a peg-in-hole task, the vibrations from scratching the peg over the rough surface are irrelevant for the segmentation, but the vibrations from entering the hole are relevant.

BOCPD [5] is used to segment demonstrated trajectories into a sequence of primitives, as shown in Fig. 2B. Each of the primitives ends with a sensory event. Because our previous work showed the superior segmentation results with multimodal sensory signals [4], we apply the BOCPD jointly to the proprioceptive and tactile signals. The number of segments for each demonstration is automatically determined by the algorithm.

BOCPD passes through the sensory trajectories and calculates the posterior distribution $p(r_t|y_{1:t})$ over the current run length $r_t \in \mathbb{Z}$ at time t given the previously observed data $y_{1:t}$. r_t represents the number of time steps since the last changepoint was detected. The posterior distribution is computed by normalizing the joint likelihood $P(r_t|y_{1:t}) = \frac{P(r_t, y_{1:t})}{P(y_{1:t})}$, where the joint likelihood $P(r_t, y_{1:t})$ over the run length r_t and the observed data $y_{1:t}$ is computed online using a recursive message passing scheme [5]

$$P(r_t, y_{1:t}) = \sum_{r_{t-1}} P(r_t|r_{t-1})P(y_t|r_{t-1}, y_t^{(r)}; \theta_m)P(r_{t-1}, y_{1:t-1}) \ ,$$

where $P(r_t|r_{t-1})$ is the conditional changepoint prior over r_t given r_{t-1}. The multivariate time series sensory

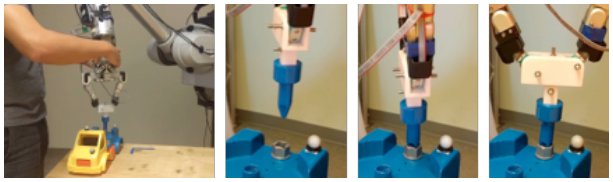

Fig. 5: Experimental setup of demonstrating the unscrewing task.

signals are modelled as a joint Student's t-distribution $P(\mathbf{y_t}|r_{t-1}, \mathbf{Y_{1:t}}; \theta_\mathbf{m})$, where $\mathbf{y_t}$ are multimodal sensory signals described in Sec. III-A and $\theta_\mathbf{m}$ are hyperparameters.

C. Segmentation Clustering and Skill Graph Generation

The BOCPD algorithm decomposes the trajectories from multiple demonstrations into multiple sequences of segments, but it does not provide correspondences among the segments of multiple demonstrations. These correspondences are required to construct a unified skill graph for each task. Rather than manually define the correspondences, our framework finds correspondences between segments by clustering the 3D goal positions and 3D orientations represented in Euler angles of the segments, as shown in Fig. 2C. We assume that the same skills from multiple demonstrations of the same task will have similar goal poses, which correspond to the articular configuration of the robot learned by demonstrations. The goal poses are extracted from the final poses of these segments.

The segments are clustered using spectral clustering [28] [29]. We first compute the similarity of pairs of segments' final poses (x_i and x_j) using a squared exponential kernel:

$$[K]_{ij} = k(x_i, x_j) = e^{\frac{-(x_i - x_j)^2}{2\sigma^2}}$$

Then, a normalized Laplacian is computed as: $L = I - D^{-1}K$, where D is a diagonal matrix with its jth diagonal element is given by $[D]_{jj} = \sum_i^n [K]_{ij}$, where n is the total number of segments. Subsequently, k-means clustering is performed on a lower-dimensional space of the eigenvectors of the normalized Laplacian. As shown in Fig. 2C, each demonstration has a sequence of segments which are labeled with unique colors based on the assigned clusters.

After the segments' final poses were clustered, a skill graph is constructed as shown on the right side of Fig. 2C. The segments from the same cluster are used to learn a skill primitive, which corresponds to a node in the skill graph. After the nodes are formed, we added directed edges if pairs of skills are demonstrated consecutively. The strength of each edge indicates the probability of the connected skills being performed in sequence.

D. Skill Replay with Exploration and Mode Discovery

Given the skill graph of a task, the robot can execute the task by traversing the skill graph through a sequence of skill primitives, which are represented as force-position controllers. The position and force signals at the end of the segments are used to compute the final desired state for the controller. The feedback gains for the controllers are predefined. A detailed overview of the control architecture can be found in [30].

Due to the nonlinear cable stretch and motor-side encoders on our robot, our robot has poor accuracy (1.5cm) as well as significantly different accuracies for different regions inside the robots workspace. Simply executing a sequence of skills will tend to result in successful replays if they are executed at the same pose as the demonstrations but failed replays if they are executed at different robot poses as the demonstrations. A sensory model for monitoring the progress of each motor primitive is therefore essential to achieve robust execution performance. The sensory model is used to confirm whether a sensory subgoal of the task has been reached by the end of each skill.

At the end of each motor primitive, the robot performs a sequence of exploratory movements, which are 5mm position deviations, along the three orthogonal directions of the current pose of the robot's end-effector, see Fig. 2D. The goal of these exploratory movements is to collect sensory signals for observing the distinct types of contact states.

The tactile sensory signals from these exploratory movements are clustered using spectral clustering to identify distinct modes, as shown in Fig. 2E. The clusters from the spectral clustering represent distinct types of contact states. These clusters are used to learn sensory models to confirm whether the goal of each primitive has been achieved or if an error has occurred.

E. Manipulation Graph Generation

Given a skill graph and the modes for the manipulation task discovered from both successful and failed skill executions, a unified manipulation graph can be created for the robot, as shown in Fig. 2F. The large rectangles in a manipulation graph correspond to the unique modes discovered by skill replays and exploration. The directed edges indicate the transition probabilities between the vertices in the graph. Some skills result in the robot remaining in the same mode while others result in switching into different modes, as indicated by the connections within the same mode or between different modes.

After generating the manipulation graph, the robot can perform the task through graph traversal by executing a sequence of skills in the graph. It can also confirm successful or failed skill executions by clustering the tactile sensory signals at the end the skill execution against the corresponding discovered success and failure modes.

IV. Experimental Evaluations

In this section we describe the experiments and results obtained for evaluating the proposed framework for building manipulation graphs by segmenting demonstrations into skill primitives and discovering corresponding modes to verify successful and failed skill executions.

A. Segmentation

We evaluated the segmentation method in our skill learning framework on three tasks: a grasping task, an unscrewing

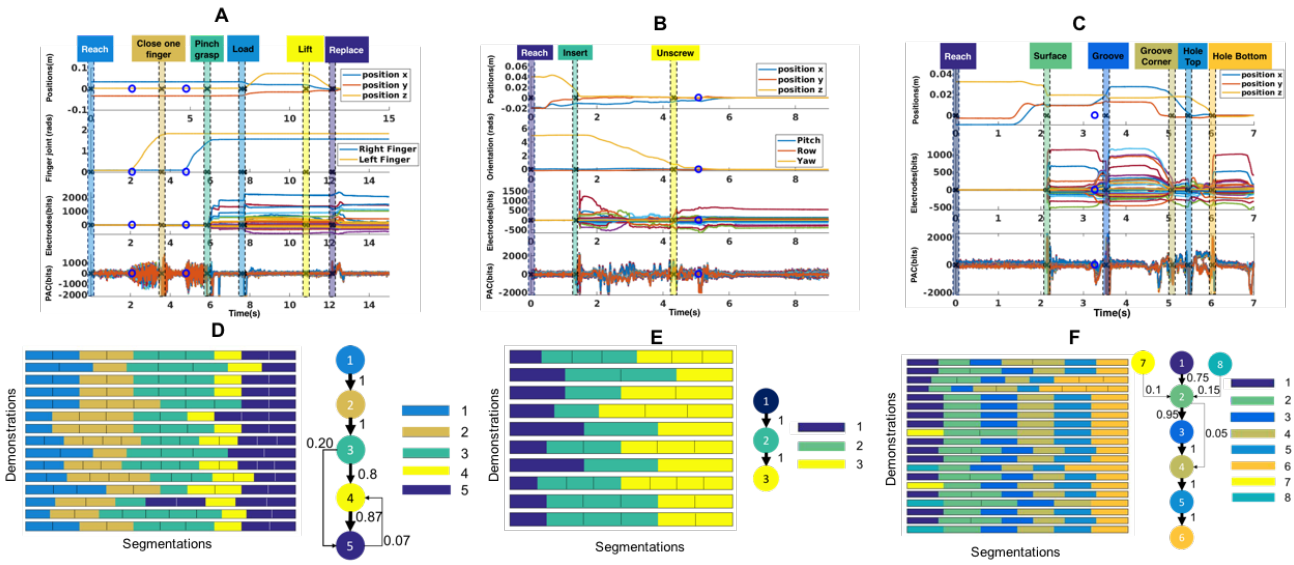

Fig. 6: A: An example of joint BOCPD to segment sensorimotor primitives in the grasping task; B: An example of joint BOCPD to segment sensorimotor primitives in the unscrewing task; C: An example of joint BOCPD to segment sensorimotor primitives in the peg insertion task; D: Segments clustering and skill graph for the grasping task; E: Segments clustering and skill graph for the unscrewing task; F: Segments clustering and skill graph for the peg-in-hole task

task, and a peg insertion task. Because kinesthetic demonstration for the grasping task will require direct contact with the fingertip which will corrupt the tactile sensory signals, we use the master-slave dual setup to move the fingers to grasp the object, as shown in Fig. 3. During each grasp demonstration, the human expert moves the master arm a sequence of movements while visually observing the slave arm and a cylinder so that the slave arm reaches on the top of the object, closes its fingers on it, lifts the object off the supporting table by about 15 cm and places it back on the supporting table. Kinesthetic demonstrations were used for the peg-in-hole and unscrewing tasks as we can hold on to the wrist above the force torque sensor, as shown in Fig. 4 and Fig. 5. The initial pose of each object is recorded by Vicon motion capture system.

The results of using BOCPD with the proprioceptive and tactile data for the grasping task is shown in Fig. 6A. The ground truth primitive switches, as indicated by the double vertical dashed lines, were manually labeled only for the purpose of showing this exemplary segmentation result. In this example case, six significant sensorimotor events were labeled, including reaching the object, closing one finger on the object, forming a pinch grasp on the object, loading the object off a supporting table, lifting the object above the targeted pose, and placing the object back on the supporting table, as shown in Fig. 3. The changepoints detected by the BOCPD algorithm are indicated by black crosses. If these changepoints are between the double vertical dashed lines, we consider the BOCPD algorithm as having successfully segmented the primitive. If changepoints fall between two consecutive sensorimotor events, we consider these changepoints as false positives, such as the changepoint at $2sec$ and $4.4sec$, indicated by the open blue circle. These two false positives are caused by the vibrations resulting from the motors and gears of the robot fingers during finger closing. They are not considered to be relevant to this task and are effectively oversegmentations caused by noisy sensory signals. The false positive changepoints are manually labelled only for the purpose of showing exemplary oversegmentation from BOCPD.

For the unscrewing task, there are three significant sensorimotor events: reaching the object, inserting the screwdriver into the head of the screw, and unscrewing the screw, as shown in Fig. 6B.

For the peg insertion task, we have a peg reaching the board, making contact with the surface of the board, sliding into the groove, reaching the corner of the groove, reaching the top of the hole, and making contact with the bottom of the hole, as shown in Fig. 6C.

B. Segmentation Clustering and Skill Graph Generation

The segmentation clustering and skill graph generation are evaluated on the segmented primitives from all three tasks: grasping, unscrewing and peg insertion. The segmented primitives from 15 trials of grasping demonstrations are clustered by applying spectral clustering on the goal poses of these segments, which are highlighted by the colored columns in Fig. 6A. On the left of Fig. 6D, each row represents one of the 15 demonstrations, and its segments are colored based on the assigned cluster. We use the same color coding to visualize the correspondences between segments from BOCPD and segments used in segment clustering, as shown in Fig. 6A and Fig. 6D, respectively. Due to oversegmentation caused by noisy sensory signals, sometimes multiple segments are assigned into the same cluster. We can keep the

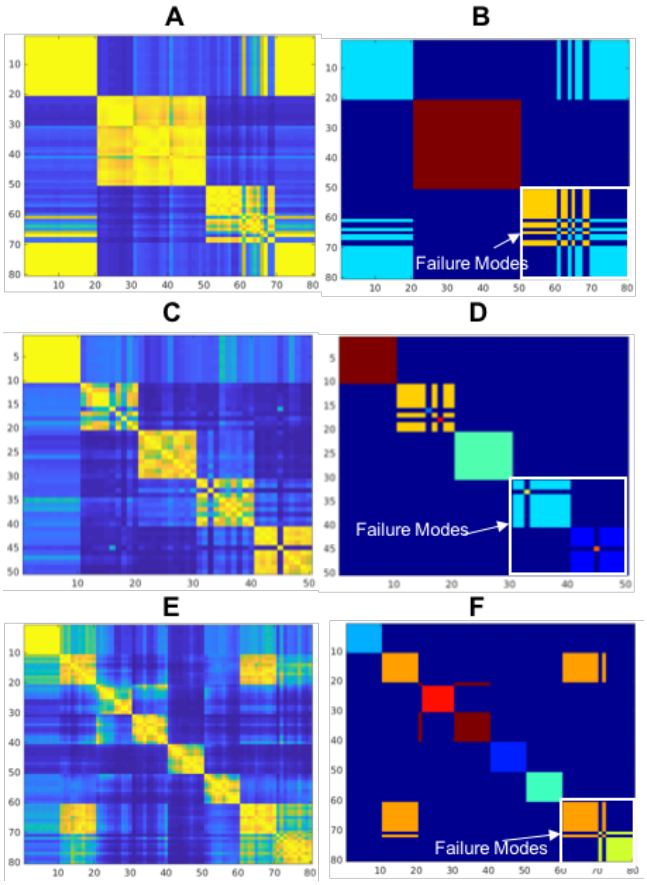

Fig. 7: Similarity matrix heat-map (A, C, E) and spectral clustering (B, D, F) of tactile signals of the exploratory movements at the goal of each phase of grasping, unscrewing and peg-in-hole tasks.

first segment assigned into a cluster and reject the segments that are subsequently assigned into the same cluster. Thus, clustering the goal poses of segments allows the robot to not only find the correspondences among segments from multiple demonstrations, but also eliminate oversegmentations. After eliminating these oversegmentation, segments among all 15 demonstrations with the same cluster label are assumed to represent the same skill primitive, therefore the mean of those segments' pose are used to form a node in the skill graph, shown as colored circles on the right of Fig. 6D. The five clusters are used to form five nodes in this grasping skill graph.

As shown on the left of Fig. 6E, the segments from 10 unscrewing demonstrations are clustered into three clusters, shown in dark blue, green and yellow. Because the first segments only neighbor with the second segment and the second segments only neighbor with the third segments, only two sequential connections are created among these three nodes, as shown on the right of Fig. 6E.

The segments from 20 trials of the peg insertion demonstrations are clustered into eight unique clusters, shown as eight nodes in the Fig. 6F.

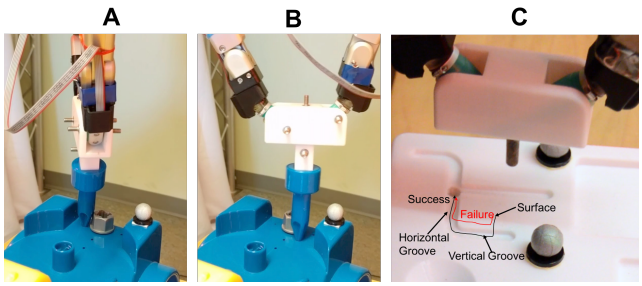

Fig. 8: A: failed to insert tool-tip into the screw head; B: after failed insertion, continue twisting the screwdriver failed to unscrew the screw; C: failed to slide into the vertical groove therefore missed the corner

C. Mode Discovery

The grasping task is executed by traversing through those five skill nodes in the grasping skill graph. Each graph traversal is sampled based on the connections' strength among those nodes. For example, traversals sampled from the grasping skill graph are $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$, and $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 4 \rightarrow 5$ with 80% and 7% probabilities, respectively. The robot performs exploratory movements, which are 5mm position deviations, along the three-orthogonal directions of the current pose at the end of each skill. It samples 10 sequences of the five nodes resulting in a total of 50 trials of explorations. During executions of the skill 3, 4, and 5, which correspond to the robot forming a pinch grasp on the object, lifting it above the table and placing it back on the table, we sample an additional 10 times when failures are introduced by the experimenter resulting in a total of 30 trials of explorations from failed executions of skill 3-5. In Fig. 7A, a heatmap shows the similarity matrix of the tactile sensory signals corresponding to those 80 explorations. The brighter color in the heatmap represents high similarity. For example, we can see that the 21-30 diagonal elements and the $31-40$ diagonal elements in the similarity matrix have high similarities, each of which corresponds to the robot forming a pinch grasp on the object and the object being lifted above the table, respectively. This is due to a stable grasp has been formed regardless of if the object is still supported by the table or if the object is lifted off of the table.

After applying spectral clustering, 80 trial explorations are clustered into three distinct clusters, shown in cyan, dark red and orange in the middle of Fig. 7. The dark red cluster (trials 21-50) in the middle of Fig. 7B represents that successful executions of skills 3, 4 and 5 are clustered into the same cluster. This is due to, when the stable grasps have been formed, the tactile sensory signals are very similar among these three skills. Trials 51-60 corresponding to failed executions of skill 3, forming a pinch grasp, are clustered into a different cluster, as shown in orange, from successful execution of skill 3, as shown in dark red. Trials 61-70 corresponds to failed executions of skill 4 when the robot attempts to lift the object. Some of them are clustered into the same cluster as successful executions of skill 1 when the robot does not make contact with the object. This is due to

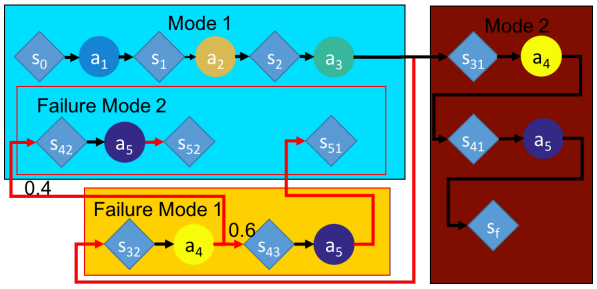

Fig. 9: Grasping Manipulation Graph

the object slipping out of the fingers when the robot tries to lift the object.

By following the skill graph of the unscrewing task, a sequence of three skills are executed on the robot including the robot moving the screwdriver towards the screw, inserting the tip into the head of the screw, and twisting it. Eight exploratory movements are applied at the end of each primitive. They are six 5mm translational movements along the three-orthogonal directions of the robots end-effector and two rotations ($\pm 5°$) along the normal of the robot's palm. This process was repeated 10 times. A similarity matrix of these 30 trials of tactile sensory signals corresponding to these eight exploratory movements is shown in the first 30 diagonal elements in the similarity matrix on the left of Fig. 7C. Three clusters are formed for these 30 trials of explorations, shown in dark red, orange, and green at the diagonal elements of spectral clustering matrix in Fig. 7D.

The robot subsequently executes these motor primitives under pose uncertainties introduced by the experimenter. Although the robot tracks its trajectory depending on the object pose measured by the Vicon system, the poor accuracy as well as significantly different accuracies inside the robot workspace causes the robot to fail its execution under pose uncertainties induced by the experimenter. Failures are detected if the tactile sensory signals at the end of each primitive failed to be clustered into the same clusters as the modes formed from successful executions. A common failure mode is discovered: such as the robot failed to insert the tool-tip into the screw hole, as shown in Fig. 8A, corresponding to the cyan cluster in Fig. 7D. This failure mode results in an additional failure mode, corresponding to the blue cluster in Fig. 7D, if the robot still twists the screwdriver without a successful insertion, as shown in Fig. 8B.

As shown in Fig. 7F, six unique modes are discovered from the exploration movements, trials 1-60, sampled from eight motor primitives in the peg-in-hole skill graph Fig. 6F. Two failure modes are also discovered due to failing to slide the peg into the vertical groove. It remains in the same mode as making contact with the surface and continues executing the next motor primitive resulting in a new mode due to sliding the peg from the flat surface directly into the horizontal groove instead of sliding into the corner along the vertical groove as shown on the right of Fig. 8.

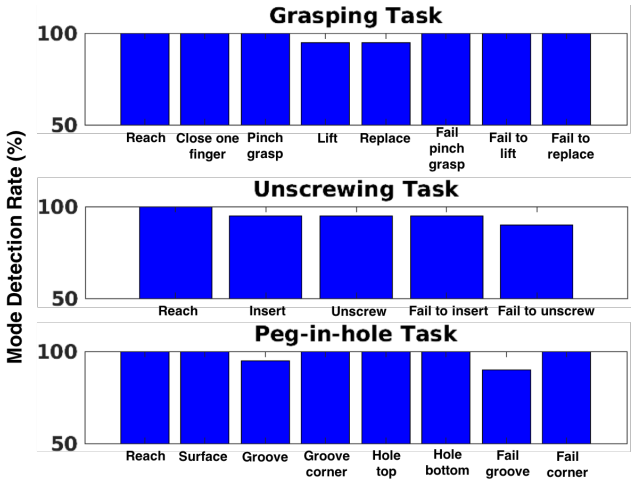

Fig. 10: Success and Failure Mode Detection

D. Manipulation Graph Generation with Failure Modes

Once we have a skill graph and discovered the corresponding modes for the grasping task, a manipulation graph can be formed by combining them. As shown in Fig. 9, starting state (s_0) is in mode 1, and a sequence of actions a_1 reaching to the object and a_2 forming a pinch graph on the object result in states s_1 and s_2 respectively, which are clustered into the same mode as s_0. Then action a_3, closing both fingers on the object, causes a mode switch as state s_{31} is in mode 2. Executing action a_4 lifts the object off of the table and action a_5 places the object back onto the table but does not cause any detected mode switches because both states s_{41} and s_f are still in mode 2, where s_f is the final state. Executing action a_3 could also result in failure mode 1 which is represented as failure state s_{32} in red in Fig. 9. After discovering this failure mode, continuing to execute the next action a_4 either stays in the same failure mode or results in an additional failure mode 2, which is clustered together with mode 1. This corresponds to the object slipping out of robot's fingers when it attempts to lift the object off the table.

We evaluated the built manipulation graphs on the Barrett robot to perform all three tasks. It performs these three tasks by doing graph traversal as well as clustering the tactile sensory signals from these executions against the success modes and failure modes. The robot executed each task 40 times which included 20 successful executions and 20 failed executions. The failed executions are caused by pose variations on the objects introduced by experimenter. The ground truth successes and failures are manually labelled by the experimenter. By comparing the manual labels against the predicted clusters, we reported success rates for successful and failed executions of each of the three tasks in Fig. 10. The detection success rates are above 90% for all the discovered success modes as well as failure modes. For these modes with only 90% detection success rates, such as failing to unscrew in the unscrewing task and failing to slide the peg into the groove in the peg-in-hole task, are due to other novel failure modes that were not present in the training data.

V. Conclusions and Future Work

We presented a framework for segmenting contact-based manipulation tasks into sequences of motor primitives. The correspondences among segments from multiple demonstrations are found by clustering the final poses of these segments. During skill replays, a sequence of exploratory movements are performed at the end of each skill to discover distinct modes that correspond to distinct contact states. Failure modes could be discovered under environment variations and uncertainties by clustering sensory events against successful skill executions. A manipulation graph is built by using skill graphs as well as discovered modes corresponding to both successful and failed skill executions.The proposed framework was successfully evaluated on grasping, unscrewing and peg insertion tasks.

Learning from demonstration allows the robot to initialize a skill, but the demonstrated skills tend to fail if they are deviated from the demonstrated trajectory due to environment uncertainties. By building a manipulation graph which incorporate both successful modes as well as distinct failure modes, it enables the robot to discover novel failure modes, which open the possibility for the robot to acquire recovery behaviors for each failure from human teachers or reinforcement learning.

References

[1] J.R. Flanagan, M.C. Bowman, and R.S. Johansson. Control strategies in object manipulation tasks. *Current opinion in neurobiology*, 16(6):650–659, 2006.

[2] R.S. Johansson and J.R. Flanagan. Coding and use of tactile signals from the fingertips in object manipulation tasks. *Nat. Rev. Neurosci.*, 10:345–359, 2009.

[3] E. Klingbeil, S. Menon, and O. Khatib. Experimental analysis of human control strategies in contact manipulation tasks. In *International Symposium on Experimental Robotics*, pages 275–286. Springer, 2016.

[4] Z. Su, O. Kroemer, G.E. Loeb, G.S. Sukhatme, and S. Schaal. Learning to switch between sensorimotor primitives using multimodal haptic signals. In *International Conference on Simulation of Adaptive Behavior*, pages 170–182. Springer, 2016.

[5] R.P. Adams and D.JC MacKay. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*, 2007.

[6] P. Pastor, M. Kalakrishnan, L. Righetti, and S Schaal. Towards associative skill memories. In *IEEE-RAS International Conference on Humanoid Robots*, pages 309–315. IEEE, 2012.

[7] Yevgen Chebotar, Oliver Kroemer, and Jan Peters. Learning robot tactile sensing for object manipulation. In *IROS*, pages 3368–3375. IEEE, 2014.

[8] S. Manschitz, J. Kober, M. Gienger, and J. Peters. Learning movement primitive attractor goals and sequential skills from kinesthetic demonstrations. *Robotics and Autonomous Systems*, 74:97–107, 2015.

[9] D. Kappler, P. Pastor, M. Kalakrishnan, W. Manue, and S. Schaal. Data-Driven Online Decision Making for Autonomous Manipulation. *in Proceedings of RSS*, 2015.

[10] S. Niekum, S. Osentoski, G. Konidaris, S. Chitta, B. Marthi, and A.G. Barto. Learning grounded finite-state representations from unstructured demonstrations. *The International Journal of Robotics Research*, 34(2):131–157, 2015.

[11] F. Meier, E. Theodorou, F. Stulp, and S. Schaal. Movement segmentation using a primitive library. In *IROS*, pages 3407–3412. IEEE, 2011.

[12] R. Lioutikov, G. Neumann, G. Maeda, and J. Peters. Learning movement primitive libraries through probabilistic segmentation. *The International Journal of Robotics Research*, 36(8): 879–894, 2017.

[13] G. Konidaris, S. Kuindersma, R. Grupen, and A.G. Barto. Robot learning from demonstration by constructing skill trees. *The International Journal of Robotics Research*, 31(3):360–375, 2012.

[14] S. Niekum, S. Osentoski, C.G. Atkeson, and A.G. Barto. Online bayesian changepoint detection for articulated motion models. In *ICRA*, 2015.

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

[16] Z. Su, K. Hausman, Y. Chebotar, A. Molchanov, G.E. Loeb, G.S. Sukhatme, and S. Schaal. Force estimation and slip detection/classification for grip control using a biomimetic tactile sensor. In *IEEE-RAS International Conference on Humanoid Robots*, pages 297–303, 2015.

[17] V. Chu, R.A. Gutierrez, S. Chernova, and A.L. Thomaz. The role of multisensory data for automatic segmentation of manipulation skills. In *RSS 2017 Workshop on Empirically Data-Driven Manipulation*, 2017.

[18] O. Kroemer, H. van Hoof, G. Neumann, and J. Peters. Learning to predict phases of manipulation tasks as hidden states. In *ICRA*. IEEE, 2014.

[19] J. Kulick, S. Otte, and M. Toussaint. Active exploration of joint dependency structures. In *ICRA*, pages 2598–2604. IEEE, 2015.

[20] X. Ji and J. Xiao. Planning motions compliant to complex contact states. *The International Journal of Robotics Research*, 20(6):446–465, 2001.

[21] G. Lee, T. Lozano-Pérez, and L.P. Kaelbling. Hierarchical planning for multi-contact non-prehensile manipulation. In *IROS*, pages 264–271. IEEE, 2015.

[22] A. Jain and S. Niekum. Efficient hierarchical robot motion planning under uncertainty and hybrid dynamics. *arXiv preprint arXiv:1802.04205*, 2018.

[23] Rachid Alami, Jean-Paul Laumond, and Thierry Siméon. Two manipulation planning algorithms. In *WAFR Proceedings of the workshop on Algorithmic foundations of robotics*, pages 109–125. AK Peters, Ltd. Natick, MA, USA, 1994.

[24] Rachid Alami, Thierry Simeon, and Jean-Paul Laumond. A geometrical approach to planning manipulation tasks. the case of discrete placements and grasps. In *The fifth international symposium on Robotics research*, pages 453–463. MIT Press, 1990.

[25] J. Deiterding and D. Henrich. Automatic adaptation of sensor-based robots. In *IROS*, pages 1828–1833. IEEE, 2007.

[26] B. Matulevich, G.E. Loeb, and J.A. Fishel. Utility of contact detection reflexes in prosthetic hand control. In *IROS*, pages 4741–4746. IEEE, 2013.

[27] N. Wettels, V.J. Santos, R.S. Johansson, and G.E. Loeb. Biomimetic tactile sensor array. *Advanced Robotics*, 22(8): 829–849, 2008.

[28] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[29] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[30] Peter Pastor, Ludovic Righetti, Mrinal Kalakrishnan, and Stefan Schaal. Online movement adaptation based on previous sensor experiences. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 365–371. IEEE, 2011.