Learning to Switch between Sensorimotor Primitives using Multimodal Haptic Signals

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

[1] Department of Biomedical Engineering, University of Southern California,
[2] Department of Computer Science, University of Southern California,
Los Angeles, CA, USA

Abstract. Most manipulation tasks can be decomposed into sequences of sensorimotor primitives. These primitives often end with characteristic sensory events, e.g., making or breaking contact, which indicate when the sensorimotor goal has been reached. In this manner, the robot can monitor the tactile signals to determine when to switch between primitives. In this paper, we present a framework for automatically segmenting contact-based manipulation tasks into sequences of sensorimotor primitives using multimodal haptic signals. These signals include both the robot's end-effector position as well as the low- and high-frequency components of its tactile sensors. The resulting segmentation is used to learn to detect when the robot has reached a sensorimotor goal and it should therefore switch to the next primitive. The proposed framework was evaluated on guided peg-in-hole tasks. The experiments show that the framework can extract the subtasks of the manipulations and the sensorimotor goals can be accurately detected.

Keywords: multimodal tactile sensing, sensorimotor primitives, primitive segmentation, learning from demonstration

1 Introduction

Manipulation tasks typically involve executing a series of discrete sensorimotor primitives. For example, humans pick and place objects by grasping, lifting, transporting, placing, and releasing the objects. These primitives are usually bound by mechanical events that represent sensorimotor subgoals of the task [1], e.g., making or breaking contact between either the hand and an object or a grasped object and another object.

These changes in the contact state result in discrete and distinct sensory events that are characterized by specific neural signatures in human tactile afferents [2]. For example, when fingers make contact with an object during grasping, signals from the slow- and fast-adapting type one afferents (SA-I, FA-I) provide information about the outcome of the grasp. Similarly, the FA-II afferents detect the contact vibrations during tool use when contact between the grasped object and another object is made or broken, or when slip occurs. An example of a sensory event for a robot is shown in Fig. 1. The tactile signals indicate that the fingers made contact, and thus reached the goal, earlier than expected. If this sensory event was completely absent, then it would indicate that the goal was not achieved. These sensory events thus provide information about if and when a primitive's goal has been reached. Given this information, the robot can determine when to terminate the current primitive and start the next one.

In this paper, we present a framework for segmenting manipulation tasks into sensorimotor primitives and subsequently learning to switch between these primitives based on tactile events. The segmentation is performed using Bayesian on-line changepoint detection [3] with multimodal haptic signals. Each changepoint indicates a sensorimotor subgoal of the task. The haptic time series signals include the Cartesian position of the robot's hand and the low- and high-frequency signals of the tactile sensors [4].

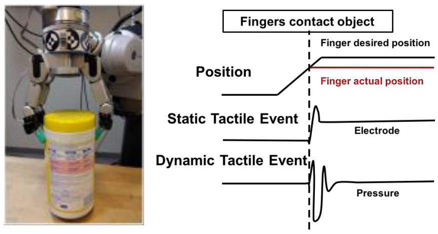

Fig. 1: An illustration of a sensorimotor event resulting from finger-object contact during grasping.

The sensory signals before and after each changepoint are used to learn a classifier for detecting the sensory event when the primitive is executed. In this manner, the robot can monitor whether the subgoal has been reached and switch to the next sensorimotor primitive accordingly. Rather than manually designing features for representing the haptic signals, the robot uses Spatio-Temporal Hierarchical Matching Pursuit (ST-HMP) [5] to learn suitable features. The detection of the sensory events is then achieved using linear support vector machines.

The proposed framework was evaluated using guided peg-in-hole tasks. The experiments evaluated the segmentation using different sets of sensor modalities, and the accuracy of the classifiers for switching between sensorimotor primitives. In a validation experiment, the robot used the learned primitives and switching behaviours to autonomously perform guided peg-in-hole tasks.

2 Related Work

Learning from demonstration (LfD) methods have emerged as an effective approach to transfer human manipulation skills to robots. Many of these methods learn libraries of movement primitives that adapt to the context of the task [6–8]. These primitives are often trained on presegmented data, and they are usually run for a fixed duration or until they reach a predefined threshold from the goal state. Kappler et al [9] also proposed a framework for switching between primitives based on multimodal signals. However, their approach is based on modeling the stereotypical sensor values at every time step of the primitives rather than detecting the characteristic sensory event of the primitive's sensorimotor goal.

Previous work has also proposed methods for automatically segmenting manipulation tasks into sequences of skills [10, 11]. These approaches focus on using proprioceptive signals to segment the tasks. By including tactile data, our segmentation approach results in primitives that terminate in sensory events that can be monitored to determine if contact goals have been reached. Methods have also been proposed for segmenting tasks into phases based on changes in the dynamics [12, 13]. Primitives can then be learned for transitioning between the segmented phases. Our approach learns primitives directly and does not require learning explicit models of the task.

A primitive that terminates early depending on sensory conditions is also known as a guarded motion. Guarded motions have been widely adopted for industrial robotic manipulators and prosthetic hands to avoid applying excessive force to the external objects [14, 15]. The sensory conditions for switching between the primitives are usually hand-designed by human experts.

Tactile servoing has also been successfully integrated into direct robot control to continuously follow distinctive surface features of objects, such as edges [16] [17]. Our work focuses on switching between primitives based on discrete sensory events and is thus a complimentary approach to including tactile feedback.

Approximate online Bayesian changepoint detection has been used in combination with articulation models to segment demonstrated manipulation tasks by detecting changes in the motions of objects [18]. In this work, authors relied only on the relative pose of two objects/parts to segment manipulation tasks, and not the force-torque or tactile signals. Given the importance of high frequency tactile signals in manipulation tasks [19, 20], our approach incorporates these signals into the online Bayesian changepoint detection.

3 Approach

The goal of our work is to autonomously segment manipulations into sensorimotor primitives and to subsequently learn classifiers for determining when to switch between the primitives. We introduce the multimodal signals and the sensorimotor primitives used in this work in Sec. 3.1 and 3.2 respectively. We then explain the segmenting of the demonstrations into primitives in Sec. 3.3, and learning to detect sensory events for switching between primitives in Sec. 3.4.

3.1 Multimodal Haptic Signals

In our experiments, we use a robot consisting of a 7-DOF Barrett WAM arm and Barrett hand, whose three fingers are equipped with biomimetic tactile sensors (BioTacs). This system provides rich multimodal haptic signals, including proprioceptive signals, and both static and dynamic tactile signals. On our robot, the proprioceptive signals include the Cartesian position of the robot's end-effector $\mathbf{y}_{\text{pos}} \in \mathbb{R}^3$ derived from the forward kinematics of the robot manipulator, as well as the force-torque signals $\mathbf{y}_{\text{FT}} \in \mathbb{R}^6$ measured on the robot's wrist force-torque sensor.

Static tactile signals are mainly sensitive to constant contacts, such as static forces applied to an object being grasped. BioTacs [17] consist of a rigid core housing an array of 19 electrodes surrounded by an elastic skin. The skin is inflated with an incompressible and conductive liquid. When the skin is in contact with an object, the liquid is displaced, and the conductance of the electrodes changes. The electrode conductance changes $\mathbf{y}_{\mathrm{E}} \in \mathbb{R}^{19}$ are used to measure the static contact forces at 100Hz.

Dynamic tactile signals are sensitive to transient mechanical events, e.g., making and breaking contact between hand-held tools and other objects. Micro-vibrations in the skin can propagate through the fluid in the BioTac and are detected as high-frequency signals by the hydro-acoustic pressure sensor embedded in the sensor's core. These high-frequency 2200Hz vibration signals, $\mathbf{y}_{\mathrm{PAC}} \in \mathbb{R}^{22}$ at 100Hz, are used to detect transient mechanical events.

3.2 Sensorimotor Primitives

A sensorimotor primitive is a parametrized synergy of motion and sensing that can be used to build task strategies. For example, the motion for inserting a peg into a hole and the sensory feedback from the peg hitting the hole bottom form a sensorimotor primitive. This sensorimotor primitive can be sequenced together with other sensorimotor primitives to perform insertion tasks.

The sensorimotor primitives used in this paper consist of a force-position controller and a sensory goal detector. The closed-loop controller defines the behaviour for reaching a desired state while the goal detector continuously monitors if the sensory goal has been reached. The primitives are segmented such that they each terminate with a sensory event, as shown in Fig. 2. These sensory events have a short duration, which is chosen to be 160ms long. This duration is chosen by comparing the goal detector's success rates under different durations of sensory events. The signals observed during the sensory event are used to train the goal detector, as detailed in Sec. 3.4. The position and force signals 100ms after the sensory event are used to compute the final desired state for the controller. The feedback gains for the controllers are predefined. The desired force is incrementally increased by 1N, if the primitive failed to reach the desired sensory event. The desired position is defined relative to the starting position of the skill. Thus, if a skill terminates early, the following primitives' desired positions are offset accordingly.

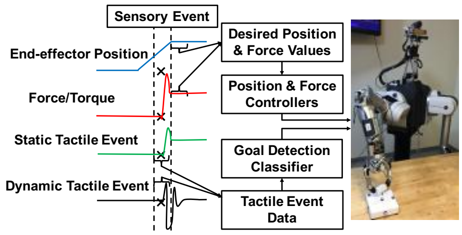

Fig. 2: Illustration of our framework of segmentation of sensorimotor primitives from demonstrated trajectories.

3.3 Sensorimotor Primitives Segmentation

Proprioceptive signals are often used to segment action primitives [10, 11]. However, these signals do not capture task-specific tactile events during motions involving contact with the environment. As a result, it is often difficult to verify if the contact goal of a primitive was achieved in these cases.

In contrast to the relatively smooth proprioceptive signals, the dynamic tactile signals are sensitive to contact events. Some of these events will be relevant to the task and result in switching between primitives, but others may be irrelevant. For example, in a peg-in-hole task, the vibrations from the peg entering the hole and making contact with the bottom of the hole both relate to task-relevant contact events. However, the vibrations resulting from scratching the peg over a rough surface are not considered to be relevant to this task and are effectively noise.

We use unsupervised Bayesian online changepoint detection (BOCPD) [3] to segment trajectories into unknown numbers of primitives with discrete sensory events. We apply this method jointly on both the proprioceptive and the tactile signals. BOCPD sequentially calculates the posterior distribution over the current run length $r_t \in \mathbb{Z}$ at time t, i.e., r_t is the number of time steps since the last changepoint. The posterior distribution $p(r_t|y_{1:t})$, given the previously observed data $y_{1:t}$, is computed by normalizing the joint likelihood $P(r_t|y_{1:t}) = \frac{P(r_t, y_{1:t})}{P(y_{1:t})}$.

The joint likelihood over the run length and the observed data is computed online using a recursive message passing scheme [3]

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

where $P(r_t|r_{t-1})$ is the conditional changepoint prior over r_t given r_{t-1}, which is nonzero in only two scenarios: $H(r_{t-1}+1|\theta_h)$ when a changepoint occurs $r_t = 0$ or $1 - H(r_{t-1}+1|\theta_h)$ when the run length continues to grow $r_t = r_{t-1} + 1$. The function $H(\tau)$ is the hazard function $H(\tau) = \frac{P_(g=\tau)}{\sum_{t=\tau}^{\infty} P_(g=\tau)}$, where $P_(g)$ is a geometric distribution with timescale θ_h. The hazard function is constant at $H(\tau) = 1/\theta_h$. The predictive distribution $P(y_t|r_{t-1}, y_{1:t}; \theta_m)$ only depends on the recent data $y_t^{(r)}$ and the model parameters θ_m. The parameters $\theta = \{\theta_h, \theta_m\}$ form the set of hyperparameters for the model.

Similar to Turner et al. [21], we use a joint BOCPD algorithm with multivariate time series sensory signals by modelling the signals as a joint Student's t-distribution $P(\mathbf{y_t}|r_{t-1}, \mathbf{Y_{1:t}}; \theta_\mathbf{m})$, where $\mathbf{y_t}$ could be any unimodal or multimodal sensory signals mentioned in Sec. 3.1. The joint model, with multimodal sensory signals, can extract more information from the data as simultaneous changes in multiple time series is a stronger indication of a changepoint.

3.4 Learning to Detect Sensory Events

After segmenting a demonstrated skill into a sequence of sensorimotor primitives, the robot should learn to autonomously determine when to switch from one primitive to the next. We treat this detection process as a classification problem. We train a classifier using the segmented sensorimotor primitives.

In order to detect different sensory events, we use Spatio-Temporal Hierarchical Matching Pursuit (ST-HMP) [5] to learn rich feature representations from the time series data of both static and dynamic tactile signals. The ST-HMP method was built upon the Hierarchical Matching Pursuit (HMP) [22] algorithm, which is a multilayer sparse coding network that creates feature hierarchies from raw data. It extends HMP by also extracting features across time series data.The ST-HMP method has achieved high accuracy in grasp stability assessment and object recognition using only low-frequency tactile sensory data on several synthetic and real tactile datasets [5]. In this paper, we incorporate signals from other sensor modalities, including high-frequency tactile data.

Including both spatial and temporal patterns of tactile information is important for achieving high classification accuracy. The ST-HMP extracts rich spatial structures from raw multimodal data without pre-defining discriminative data characteristics. Given a set of high-dimensional observations, it uses K-SVD [23] to learn a dictionary and the associated sparse code matrix in an unsupervised fashion over a large collection of spatial patches sampled from multimodal data. With the learned dictionary, the ST-HMP computes sparse code features for each high-dimensional observation in a small neighborhood using orthogonal matching pursuit. Then those sparse code features are max pooled over the spatial and temporal dimensions at several scales with an increasing size of a receptive field (cell) to generate robust feature vectors for both spatial and temporal variations. The final feature describing the whole sensor data sequence is the concatenation of aggregated sparse codes in each spatio-temporal cell. Algorithm details can be found in the paper of Madry et al. [5].

In order to represent the robot's haptic data using HMP features, we need to first arrange the tactile signals into 2D tactile data arrays. The layout of the BioTac sensor's electrodes is shown in Fig. 3. The Xs on the finger indicate the reference electrodes, and the 19 BioTac electrodes $E1...E19$ are measured relative to these 4 reference electrodes. $V1$ and $V2$ are two virtual electrodes computed by taking an average response of the neighboring electrodes $V1 = \mathbb{E}[E17, E18, E12, E2, E13, E3]$ and $V2 = \mathbb{E}[E17, E18, E15, E5, E13, E3]$. The high-frequency vibration signals (PAC) from one pressure sensor on each finger are separated into 22 virtual channels over time $P1...P22$, and the vibration signals from the three fingers ($F1, F2, F3$) are concatenated side by side. Thus, HMP is essentially extracting temporal features from these 22 virtual vibration channels within one finger as well as learning features to reflect the dependencies of sensors on multiple fingers.

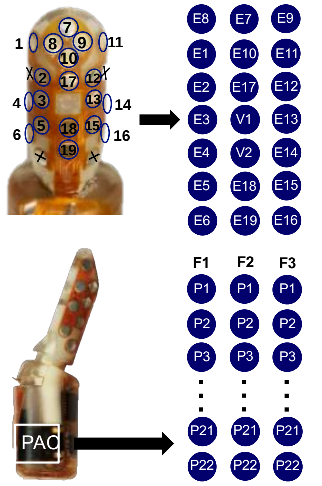

Fig. 3: Schematic of the electrode and pressure sensor arrangement on the BioTac (left). Tactile data array used for the ST-HMP features (right).

In order to structure the data, the 19 electrodes and two virtual electrodes ($V1$ and $V2$) on each finger are laid out as a 7×3 2D data array. The vibration signals (PAC) on the three fingers are laid out as 22×3 2D tactile data array, as shown in Fig. 3. In this manner, three BioTacs create total four 2D tactile data arrays: three 7×3 tactile arrays for electrodes and one 22×3 tactile data array for vibration signals. We then apply the HMP to each tactile data array separately and then concatenate feature vectors. HMP learns a dictionary of size $M = 100$ with the sparsity level set to $K = 4$. The spatial pooling is performed with a 3 level pyramid: the data array is divided into 1×1, 2×2 and 3×3 cell grids, which results in $S = (1 + 2^2 + 3^2) = 14$ spatial cells. The temporal pyramid consists of 4 max-pooling levels: the sequence is divided into 1, 2, 4, and 8 parts, which results in $T = (1+2+4+8) = 15$ temporal cells. To prevent losing the signs of HMP features due to max-pooling on absolute values, we save the feature vector with both positive and negative signs. Therefore, the size of the feature descriptors is doubled. The total number of ST-HMP features is $4 \times S \times T \times M \times 2 = 4 \times 14 \times 15 \times 100 \times 2 = 168000$.

Given the ST-HMP tactile features, a Support Vector Machine (SVM) is then used to classify these features. For rich features provided by sparse coding, a linear kernel obtains satisfactory results and there is no need to apply more complex distance measures.

4 Evaluation and Discussion

In this section we describe the experiments and results obtained for evaluating the proposed sensorimotor primitive segmentation and goal detection framework.

4.1 Sensorimotor Primitives Segmentation for Peg-in-hole tasks

Experimental Setup We evaluated our method on our robot platform. For the guided peg-in-hole tasks, we use a 3D printed peg-in-hole set consisting of holes with 1mm clearance and various geometric features, including a curved groove leading into a hole, a straight groove leading into a hole, and a squared groove with a hole at one of its corners. These geometric features of the board are shown in the inset of Fig. 4. These features are designed to create constraints that guide the robot while performing the peg-in-hole tasks. Interacting with these geometric features results in tactile events. The robot should therefore learn sequences of sensorimotor primitives that reach the individual geometric features, and switch between the primitives accordingly to perform the task. An

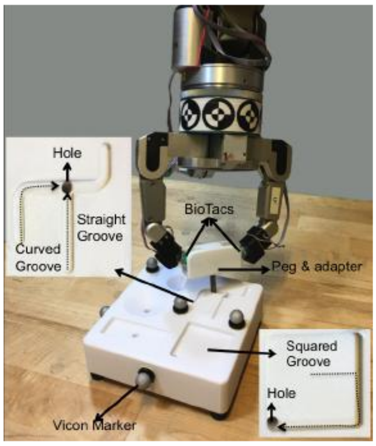

Fig. 4: Experiment setup of the peg-in-hole manipulation task.

adapter was 3D printed to hold the 5.7mm diameter peg, such that it can be firmly grasped by two BioTacs using a pinch grasp, as shown in Fig. 4.

In the experiment, the robot was taught by a demonstrator to perform the guided peg-in-hole tasks using kinesthetic teaching. For example, to use the curved groove, the demonstrator moved the robot's hand down until the peg made contact with the surface of the board, slid the peg into the curved groove, traced the groove with the peg until reaching the opening of the hole, and finally inserted the peg into the hole. We collected 50 demonstrations with each geometric feature on the peg board.

We recorded the 3D Cartesian position of the robot's end-effector from the robot's motor encoders using its forward kinematics. We also tracked the 3D Cartesian position of the board with a Vicon motion capture system. Thus, we can calculate the relative position of the end-effector and the board (pos). In order to compare the segmentation performance with different sensor modalities, we also recorded the signals from the signals from the force/torque sensor at the wrist (FT), the BioTacs' electrodes (E), and the BioTacs' high-frequency pressure sensor (vib).

The joint predictive distributions over the sensor values were modelled using Student's t-distributions with hyper-parameters θ_m: $\mu_{\text{pos}} = 0.02, \sigma_{\text{pos}} = 10^{2.5}$; $\mu_{\text{FT}} = 0, \sigma_{\text{FT}} = 1$; $\mu_{\text{E}} = 0, \sigma_{\text{E}} = 1$; and $\mu_{\text{vib}} = 1000$, $\sigma_{\text{vib}} = 10^{-2}$, respectively. The hazard function's hyper-parameter was set to $\theta_h = 250$.

Results The results of using joint BOCPD with the proprioceptive and tactile data for the curved-groove task is shown in Fig. 5. The ground truth primitive switches were manually labeled, as indicated in Fig. 5 by double vertical dashed lines. In this example case, five significant sensorimotor events were labeled, including the peg impacting the surface of the board, entering 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. 4. 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 there is no changepoint between these double vertical dotted lines (red), then BOCPD missed the event, e.g., the corner of the curved groove. If changepoints fall between two consecutive sensorimotor events, we consider these changepoints as false positives, such as the changepoint at $0.93s$ and $2.76s$ shown by circles (blue). The first of these false positive is caused by the bumpy surface of the peg board. The second false positive is caused by the peg jamming against the inner surface of the hole.

The joint BOCPD on the multimodal signals performed better than the independent BOCPD on the unimodal signals. Fig. 6-8 show the segmentation success rates and false positive rates for each sensorimotor event in the three guided peg-in-hole tasks, i.e., curved groove, straight groove, and square groove respectively. The proprioceptive and multimodal tactile signals, including the electrodes and pressure sensors, usually achieved the highest success rates and the lowest false positive rates. This result is due to the changepoints of the joint

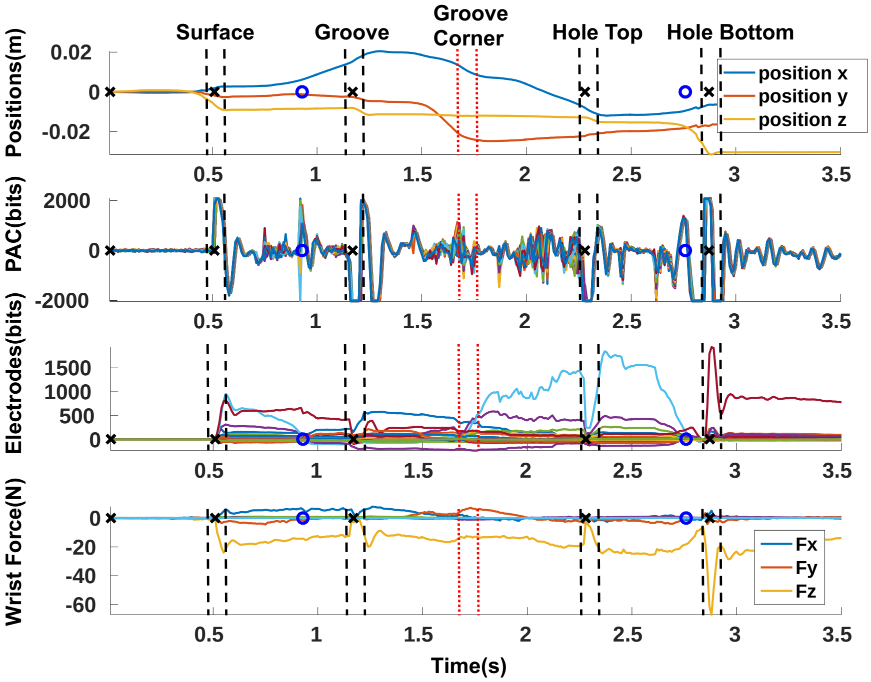

Fig. 5: An example of joint BOCPD to segment sensorimotor primitives in a peg-in-hole task with curved groove.

BOCPD using the effects of both the low- and high-frequency sensory information. Thus, the joint model can extract more information from the data as simultaneous changes in multiple time series is a stronger indication of a sensorimotor changepoint.

4.2 Sensorimotor Primitives Goal Detection

Experimental Setup We evaluated the sensorimotor primitive goal detection method using the changepoints detected by the joint proprioceptive and tactile BOCPD. The goal is to have the robot autonomously detect whether it has reached the goal of the current sensorimotor primitive. For every changepoint detected by the segmentation method, except the first one, we extracted 16 sensory data samples (160ms) directly before and after the changepoint. These samples represent the tactile signals from the goal's sensory event. We also extracted 16 samples randomly selected between the last changepoint and the current changepoint. These samples correspond to the signal before the goal has been reached. In this manner, we collected 560 positive (goal detected) and 560 negative (goal not detected) samples from 35 trials for the evaluation.

In this experiment, we compared the goal detection accuracies using either HMP or ST-HMP features. The difference between ST-HMP and HMP is that ST-HMP combines the tactile information from multiple time steps t to create the features. In contrast, HMP creates features for each time step separately and then concatenates them. To evaluate the HMP and ST-HMP features for goal

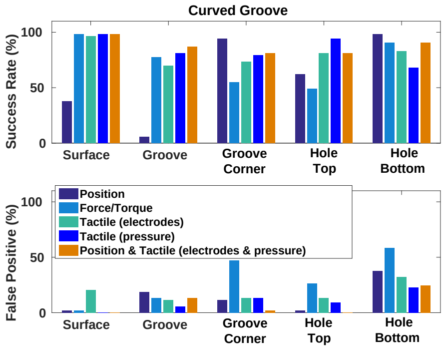

Fig. 6: Curved groove's segmentation success rate and false positive rate.

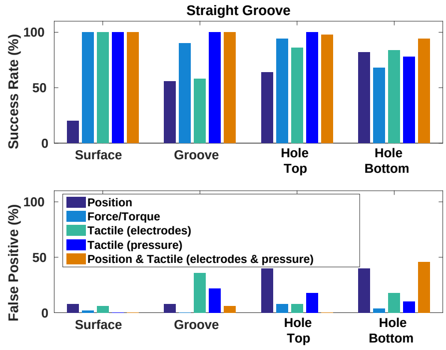

Fig. 7: Straight groove's segmentation success rate and false positive rate

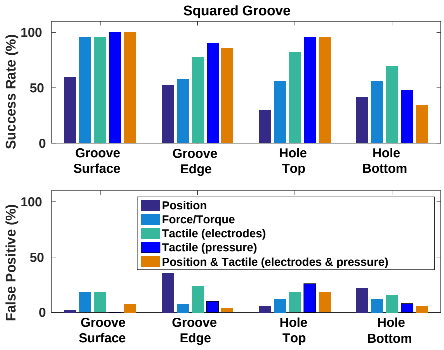

Fig. 8: Squared groove's segmentation success rate and false positive rate.

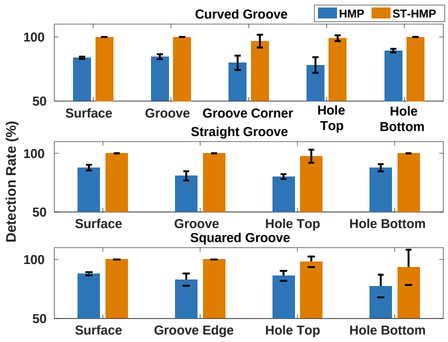

Fig. 9: Peg-in-hole sensorimotor primitive detection results.

detection, we performed a 5-fold cross-validation on the data set by using 896 samples for training the classifier and the rest for testing.

Results By using all tactile sensor modalities, as shown in Fig. 3, the average classification accuracies among the different sensorimotor primitives range from 77.5% to 100%. The classification accuracies and the standard deviations for the different sensorimotor primitives are shown in Fig. 9.

Overall, the ST-HMP achieves higher accuracies and lower standard deviations than the HMP. This is due to ST-HMP pooling over the time steps, which results in temporal invariances. The results thus show the importance of combining information from multiple time scales when detecting sensory events.

4.3 Robot Performing Peg-in-hole Task

In this experiment, the robot uses the segmented primitives and goal detectors from the previous experiments to autonomously perform the guided peg-in-hole task with the curved groove. The segmentation was performed using the proprioceptive and tactile signals, while the sensory event detection only uses the tactile data. The position and force signals 100ms after each segmentation are used to compute the final desired position and contact force for each controller. The desired positions generated by a minimum jerk trajectory generator are tracked by a velocity-based operational space controller together with an inverse dynamic law and PD feedback error compensation in joint space [6]. Tracking of desired contact forces on the arm is achieved with a PI controller on the force/torque sensor located at the wrist [6].

Fig. 10: Sensorimotor primitive sequence for the curved groove peg-in-hole task.

An example sequence of sensorimotor primitives successfully executing the peg-in-hole task with a curved groove is shown in Fig. 10. Without the sensory event detection, we observed two common failure modes: i) the robot misses the groove (failed transition from 2nd to 3rd picture), and ii) the robot jams the peg around the groove corner (failed transition from 4th to 5th picture). The sensory event detection alleviates these issues by detecting when the goal state was not reached, i.e., the sensory event was not detected, and repeating the current primitive to reach the goal. The required correction is usually rather small, and the primitive terminates once the goal has been reached.

5 Conclusions

We presented a framework for segmenting contact-based manipulation tasks using both proprioceptive and tactile signals. We used the unsupervised online Bayesian changepoint detection algorithm to automatically segment manipulations into sensorimotor primitives. Classifiers using ST-HMP features, were trained to detect sensory events for switching between primitives. The proposed method was successfully evaluated on guided peg-in-hole tasks. The robot could accurately segment the tasks and detect the sensory events using the proposed approach.

In the future, we will extend the proposed framework to learn to detect failure events through autonomous exploration.

Acknowledgments. Research supported by the MPI for Intelligent Systems. BioTac sensors were provided by SynTouch LLC. Gerald E. Loeb is an equity partner in SynTouch LLC, manufacturer of the BioTac sensors used in this research. Special thanks to Felix Grimminger for helping design the 3D printed parts.

References

1. Flanagan, J.R., Bowman, M.C., Johansson, R.S.: Control strategies in object manipulation tasks. Current opinion in neurobiology **16**(6) (2006) 650–659
2. Johansson, R.S., Flanagan, J.R.: Coding and use of tactile signals from the fingertips in object manipulation tasks. Nat. Rev. Neurosci. **10**(5) (2009) 345–359
3. Adams, R.P., MacKay, D.J.: Bayesian online changepoint detection. arXiv preprint arXiv:0710.3742 (2007)
4. Wettels, N., Santos, V., Johansson, R., Loeb, G.: Biomimetic tactile sensor array. Advanced Robotics **22**(8) (2008) 829–849
5. Madry, M., Bo, L., Kragic, D., Fox, D.: St-hmp: Unsupervised spatio-temporal feature learning for tactile data. In: ICRA, IEEE (2014) 2262–2269
6. Pastor, P., Kalakrishnan, M., Righetti, L., Schaal, S.: Towards associative skill memories. In: Humanoids, IEEE (2012) 309–315
7. Chebotar, Y., Kroemer, O., Peters, J.: Learning robot tactile sensing for object manipulation. In: IROS, IEEE (2014) 3368–3375
8. Manschitz, S., Kober, J., Gienger, M., Peters, J.: Learning movement primitive attractor goals and sequential skills from kinesthetic demonstrations. Robotics and Autonomous Systems **74** (2015) 97–107
9. Kappler, D., Pastor, P., Kalakrishnan, M., Manue, W., Schaal, S.: Data-Driven Online Decision Making for Autonomous Manipulation. Rss (2015)
10. Meier, F., Theodorou, E., Stulp, F., Schaal, S.: Movement segmentation using a primitive library. In: IROS, IEEE (2011) 3407–3412
11. Niekum, S., Osentoski, S., Konidaris, G.D., Chitta, S., Marthi, B., Barto, A.G.: Learning grounded finite-state representations from unstructured demonstrations. IJRR **34**(2) (2015) 131–157
12. Kroemer, O., van Hoof, H., Neumann, G., Peters, J.: Learning to predict phases of manipulation tasks as hidden states. In: ICRA, IEEE (2014) 4009–4014
13. Kulick, J., Otte, S., Toussaint, M.: Active exploration of joint dependency structures. In: ICRA, IEEE (2015) 2598–2604
14. Deiterding, J., Henrich, D.: Automatic adaptation of sensor-based robots. In: IROS, IEEE (2007) 1828–1833
15. Matulevich, B., Loeb, G.E., Fishel, J.A.: Utility of contact detection reflexes in prosthetic hand control. In: IROS, IEEE (2013) 4741–4746
16. Li, Q., Schürmann, C., Haschke, R., Ritter, H.J.: A control framework for tactile servoing. In: Robotics: Science and systems. (2013)
17. Su, Z., Fishel, J., Yamamoto, T., Loeb, G.: Use of tactile feedback to control exploratory movements to characterize object compliance. Front. neurorob. **6**(7) (2012)
18. Niekum, S., Osentoski, S., Atkeson, C.G., Barto, A.G.: Online bayesian changepoint detection for articulated motion models. In: ICRA, IEEE (2015) 1468–1475
19. Romano, J.M., Hsiao, K., Niemeyer, G., Chitta, S., Kuchenbecker, K.J.: Human-inspired robotic grasp control with tactile sensing. IEEE Transactions on Robotics **27**(6) (2011) 1067–1079
20. Su, Z., Hausman, K., Chebotar, Y., Molchanov, A., Loeb, G.E., Sukhatme, G.S., Schaal, S.: Force estimation and slip detection/classification for grip control using a biomimetic tactile sensor. In: Humanoids, IEEE (2015) 297–303
21. Turner, R., Saatci, Y., Rasmussen, C.E.: Adaptive sequential bayesian change point detection. In: Temporal Segmentation Workshop at NIPS. (2009)

22. Bo, L., Ren, X., Fox, D.: Hierarchical matching pursuit for image classification: Architecture and fast algorithms. In: NIPS. (2011) 2115–2123
23. Aharon, M., Elad, M., Bruckstein, A.: k -svd: An algorithm for designing overcomplete dictionaries for sparse representation. Trans. Sig. Proc. **54**(11) (Nov 2006)