

Robot-Object Contact Perception using Symbolic Temporal Pattern Learning

Nawid Jamali, Petar Kormushev and Darwin G. Caldwell

Abstract—This paper investigates application of machine learning to the problem of contact perception between a robot's gripper and an object. The input data comprises a multidimensional time-series produced by a force/torque sensor at the robot's wrist, the robot's proprioceptive information, namely, the position of the end-effector, as well as the robot's control command. These data are used to train a hidden Markov model (HMM) classifier. The output of the classifier is a prediction of the contact state, which includes no contact, a contact aligned with the central axis of the valve, and an edge contact. To distinguish between contact states, the robot performs exploratory behaviors that produce distinct patterns in the time-series data. The patterns are discovered by first analyzing the data using a probabilistic clustering algorithm that transforms the multidimensional data into a one-dimensional sequence of symbols. The symbols produced by the clustering algorithm are used to train the HMM classifier. We examined two exploratory behaviors: a rotation around the x-axis, and a rotation around the y-axis of the gripper. We show that using these two exploratory behaviors we can successfully predict a contact state with an accuracy of $88 \pm 5\%$ and $81 \pm 10\%$, respectively.

I. INTRODUCTION

Robots can benefit from the ability to dexterously manipulate objects in hazardous environments. Often, such environments adversely affect sensing modalities such as vision making them unreliable, hence, making direct contact an important sensing modality that can help in successfully inspecting and/or manipulating an object. Applications of such a system include search and rescue operations and underwater manipulation of objects. In underwater environments, increased turbidity in the water makes visual detection of objects unreliable. In disaster zones a burst pipe might reduce the quality of vision.

In this paper we present a method that can be used to inspect an object that only relies on direct contact. Within the framework of the PANDORA project [1], [2], the object inspection task, considered for testing our method, is to determine the location of a contact between a robotic gripper and a T-bar valve. Such a system is necessary to allow a robot to manipulate the valve. We propose two exploratory behaviors that can be used to determine the location of a contact between the robot and the valve. The exploratory behaviors are selected such that at different contact locations a unique pattern is produced by the sensors. A contact state such as no

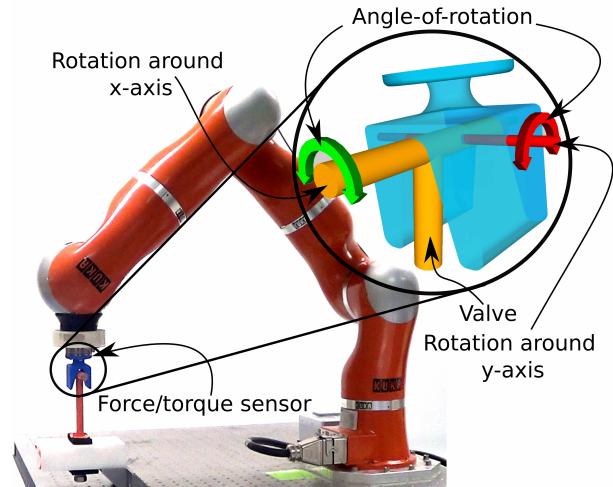


Fig. 1. Experimental setup.

contact, a contact aligned with the central axis of the valve, and an edge contact can be detected by mining the data for temporal patterns. Our contribution is a robust autonomous contact location determination based on machine learning that relies on non vision data, namely, the contact forces and torques, and the proprioceptive information, that is, the Cartesian position of the end-effector of the robot.

We learn a classifier for predicting a contact state using our previously developed method to analyze temporal patterns in a multidimensional time-series data [3]. In this paper we apply the learning method to a new problem, i.e., gripper-valve contact state classification. We modified the method by adding the control signal in the feature space of the algorithm to correlate patterns that arise due to control commands. We also propose exploratory behaviors that help in perceiving the contact state between a T-bar valve and a robot.

Fig. 1 shows the experimental setup used to test the proposed method. It consists of a gripper attached to a KUKA-LWR robotic arm. An ATI Mini45 force/torque sensor is sandwiched between the gripper and the robot's end-effector. A T-bar valve is placed in the robot's workspace. The robot performs the proposed exploratory behaviors, which, as illustrated by the green arrow and the red arrow in Fig. 1, are periodic rotary movements with a given angle-of-rotation. This action induces the force/torque sensors and affects the position of the end-effector. For example, by performing the rotation indicated by the green arrow, the gripper will be obstructed by the valve's axis, resulting in the force/torque sensor registering higher values. However, a contact at the edge is free to rotate. We show that using the proposed

This research was sponsored by the PANDORA EU FP7-Project under the Grant agreement No. ICT-288273.

The authors are with the Department of Advanced Robotics, Istituto Italiano di Tecnologia, via Morego, 30, 16163 Genova, Italy email: {nawid.jamali, petar.kormushev, darwin.caldwell}@iit.it

exploratory behaviors the robot is capable of learning a classifier to successfully predict a contact location.

II. RELATED WORK

Earlier research in valve detection and manipulation assumes a structured environment, where the robot stores detailed information about the environment [4], [5]. Non-vision sensors such as force/torque sensors, binary touch sensors and inductive proximity sensors have been used to confirm contact and monitor applied forces [4], and detect orientation of a valve handle and manipulate the valve without over-tightening/loosening [5]. However, these approaches have been limited to in-air applications. To facilitate underwater manipulation, grippers instrumented with tactile sensors that can operate underwater have been developed [6], [7].

Anisi et al. [5] propose use of an inductive proximity sensor and a torque sensor to detect the orientation of a metallic T-bar valve handle and subsequently manipulate the valve without over-tightening/over-loosening the valve. However, use of an inductive proximity sensor limits the application of the system to metallic objects.

Marani et al. [8] used vision to locate an underwater object and hook a cable to it so that it can be lifted to surface. Recently, Prats et al. [9] used a laser scanner to build a 3D point cloud of an underwater object, which is then used to plan and execute a grasp. However, in both cases an operator has to indicate the region of interest to the robot. Moreover, vision and laser are adversely affected by turbidity in water.

Recently, Ahmedzadeh et al. [10] proposed a hierarchical learning approach that allows a robot to safely approach and manipulate a valve. The authors developed a reactive controller that commands the robot to retract its gripper when the relative movement between the robot's gripper and the valve is oscillating with a large variance. When it is safe, the robot approaches the valve and turns it. However, the valve turning is hardcoded and doesn't consider contact forces.

III. METHODOLOGY

The input to the algorithm consists of the force/torque sensor data and the position of the end-effector. We also include the control command in our analysis, resulting in an eight-dimensional time-series data. To learn concepts from a multidimensional time-series we divide the problem into two stages: dimensionality reduction using clustering and temporal pattern extraction. In the first stage, probabilistic clustering [11] is applied to discover the intrinsic structure of the data. Clustering transforms the multidimensional time-series into a sequence of symbols, each of which is an identifier for a cluster. Thus, an abstract contact condition such as the location of a contact can be represented by a sequence of cluster identifiers. In the second stage, using a supervised hidden Markov model (HMM), the algorithm analyzes the sequence and builds a probabilistic model of temporal patterns that correspond to different abstract contact conditions.

An overview of the learning algorithm is shown in Fig. 2. The first step of the analysis involves preprocessing the

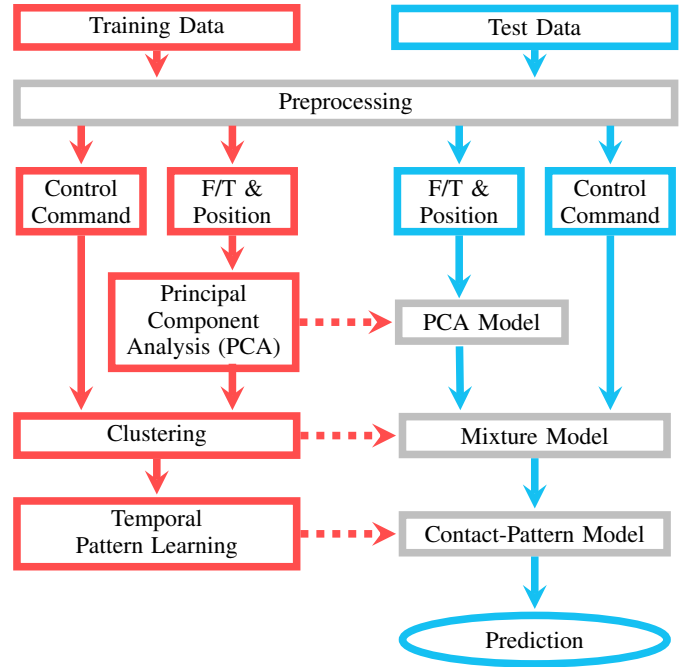


Fig. 2. Overview of the learning method.

data. The clustering method assumes that all of the variables are independent [11]. Hence, in the first stage of training, the sensor data are projected onto a new basis using principal component analysis (PCA). The principal component coefficients from the training data are saved. Later, these coefficients are used to transform the test data to the new coordinate system. The control command along with the output of PCA are then used as an input to the clustering algorithm. Clustering plays two roles. Firstly, clustering is used to discover the intrinsic structure within the data. Secondly, it reduces the high dimensional time-series data into a one-dimensional sequence of clusters. The mixture model output by the clustering algorithm is also saved, which is subsequently used to transform the test data into a sequence of clusters. Each cluster is denoted by a letter. $S = \{BEDBCCACDDADECBCCAEBDA...\}$ is an example of a sequence of cluster memberships.

In the second stage, the output of the clustering algorithm, that is, the sequence of cluster memberships is analyzed to discover and learn different patterns that represent different contact locations. We want to discover unique patterns that emerge during each contact. For example, in the sequence S , BCCA is a recurring pattern that can be learned as a pattern that represents, say, a center-contact. The model for these patterns is also saved.

The algorithm is tested using an unseen test set. In the testing phase, the models saved during training are used to transform the data into a temporal sequence of cluster memberships. The models for the patterns discovered during training are used to predict the state of the contact. For example, encountering the pattern BCCA will signify a center-contact. We use hidden Markov models to discover the patterns for each contact condition. The following sections provide a detailed description of each step.

A. Preprocessing

All signals are preprocessed, in MATLAB, before any further analysis is performed.

1) *Zero-mean normalization*: The force/torque data are zero-mean normalized. We are interested in the range of forces, the absolute value of the applied force is not important. This process helps in making the forces and the torques comparable between trials.

2) *End-effector position*: The end-effector position is calculated using the distance between the end-effector and the valve, that is, their relative position. The position of the valve is estimated by monitoring the force/torque data for the impact force. We take the position at impact to be the position of the valve. The relative distance helps in making data between trials comparable.

3) *Filtering*: The force/torque data are sampled at 250 Hz, the control signal changes at 0.125 Hz. Hence, the force/torque data is filtered using a digital filter with a 3 dB point of 1.25 Hz, which is ten times the frequency of the control signal. The sensor signals are oversampled to avoid aliasing. After the application of the digital filters, the data is downsampled by a factor of 50.

B. Principal Component Analysis

Principal component analysis (PCA) is a mathematical transformation that converts a set of variables to a new basis, called the principal components. Each principal component is a linear combination of the original variables. In this space, all variables are uncorrelated. Moreover, in the principal component space, the first component explains the greatest variance in the data, every subsequent component captures lower variance as compared to the preceding component. A consequence of this property is that PCA can also be used as a dimensionality reduction tool.

Dimensionality reduction is achieved by only keeping the lower components. A rule of thumb is to drop components once the ratio of cumulative-variance¹ to total-variance has exceeded a predetermined threshold, usually 0.8. Hence, in the subsequent analysis we only consider the components that have not exceeded the cumulative-variance threshold. In our dataset the threshold was exceeded after the second component. Hence, we consider only two components in the subsequent analysis.

C. Control Command

The control command plays an important role in inducing the sensors. The control signal for our experiments is the commanded angle-of-rotation. We feed the control command directly to the clustering algorithm to allow the algorithm to correlate the control with the sensor data.

D. Clustering

We use probabilistic clustering [11], which uses the minimum message length (MML) principle as the optimization

¹Cumulative-variance is calculated by summing the variance of all components up to the component of interest.

criterion, to build a Gaussian mixture model of the data. In this section we explain the theory behind MML clustering.

The MML principle is based on the information-theoretic Bayesian principle of inductive inference[12], [11]. Let D be the data and H be an hypothesis explaining the data. The posterior probability of the hypothesis is given in Equation 1, which is derived by repeated application of Bayes' theorem.

$$Pr(H|D) = \frac{Pr(H, D)}{Pr(D)} = \frac{Pr(H)Pr(D|H)}{Pr(D)} \quad (1)$$

$$\begin{aligned} MessageLength &= -\log_2(Pr(H)Pr(D|H)) \\ &= -\log_2(Pr(H)) - \log_2(Pr(D|H)) \end{aligned} \quad (2)$$

Since the goal is to infer a hypothesis (H) that best explains the given data (D), the problem can be viewed as maximizing the posterior probability, $Pr(H|D)$. From information theory, we know that an event of probability p can be coded by a message of length $l = -\log_2 p$ bits. Equation 2 is derived by applying the coding theory to equation 1. Hence, maximizing the posterior probability can be achieved by minimizing the message length of a two-part message conveying the theory, H , and the data, D , in the light of the theory, H . Using the two-part message method, MML prefers a simple hypothesis over more complex hypotheses. We use Vanilla-Snob[13] by Chris Wallace to build a mixture model of our data.

We use MML as optimization criterion to stop the clustering algorithm. Any other method such as the Minimum Description Length[14] will also work.

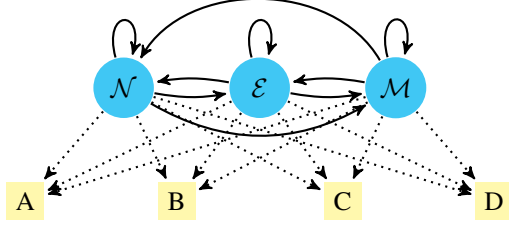
E. Learning

Once the multidimensional signals from the robot are transformed into a temporal sequence of clusters, we use HMMs to discover the patterns for each contact condition. The training examples are generated by allowing the robot to perform an action. In this case the action is to perform an exploratory behavior at a contact point. The training sequence is labeled with the contact location, which is recorded during data collection. This allows the algorithm to learn a mapping from the temporal sequence of clusters to a classification of the contact state. The accuracy of the classifier is tested by applying it to a novel sequence, where the contact state is unknown to the robot.

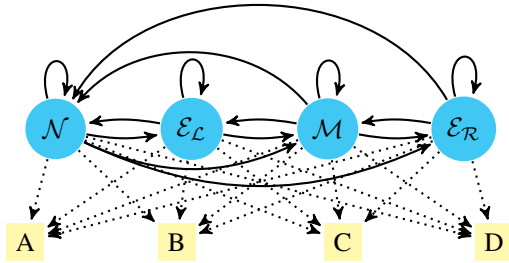
Fig. 3(a) shows an example of a sequence generated after the application of the clustering algorithm. The corresponding feature vector is also shown and consists of temporal sequence of couples of the form (*cluster-membership*, *contact-state*). Fig. 3(b) and Fig. 3(c) show the topology of the HMMs used to learn a representation for the emerging temporal patterns. The HMM illustrated in Fig. 3(b) is used as a model for the rotation around y-axis. It has three states, one for each contact condition, which will be explained in Section IV-B. Similarly, the HMM in Fig. 3(c) is used for the rotation around x-axis. It has four states, more details will

$$\begin{aligned}
S_{original} &= \{BCCDDCAAAABAABB\dots\} \\
S_{labelled} &= \left\{ (B, \mathcal{N})(C, \mathcal{N})(C, \mathcal{N})(D, \mathcal{N})(D, \mathcal{N})(C, \mathcal{E}) \right. \\
&\quad (A, \mathcal{E})(A, \mathcal{E})(A, \mathcal{E})(A, \mathcal{E})(B, \mathcal{M}) \\
&\quad (A, \mathcal{M})(A, \mathcal{M})(B, \mathcal{M})(B, \mathcal{M}) \dots \left. \right\}
\end{aligned}$$

(a) Example of a feature vector.



(b) A three state hidden Markov model.



(c) A four state hidden Markov model.

Fig. 3. A feature vector and two hidden Markov models. The letters A, B, C and D indicate membership to the corresponding cluster. The letters \mathcal{N} , \mathcal{E} , \mathcal{E}_L , \mathcal{E}_R and \mathcal{M} represent contact states: no-contact, edge-contact, left-edge-contact, right-edge-contact, and center-contact, respectively.

be provided in Section IV-B. The HMMs are trained using the sequence of clusters as the observation. When the robot is presented with a novel pattern, the robot uses the model to make a prediction.

IV. EXPERIMENTAL SETUP

The experimental setup illustrated in Fig. 1, described in Section I, was used to test the method. In this section we will define the contact locations, which is followed by a description of the exploratory behaviors and a detailed description of the experiments.

A. Contact-Location Categorization

A contact between the gripper and the valve is categorized as follows:

1) **Edge-contact**: In an edge-contact, the area of contact between the valve and the gripper is at least $\frac{1}{3}$ of the width of the gripper, and the central axis of the valve is not covered. An edge-contact is further divided into **left-edge-contact** and **right-edge-contact**. In a left-edge-contact, the contact point is to the left of the central axis of the valve relative to the robot. Similarly, in a right-edge-contact the edge is to the right of the central axis of the valve.

2) **Middle-contact**: In a middle-contact, the full length of the gripper is in contact with the valve, i.e., it is somewhere in between the two edges. A middle-contact is categorized as

center-contact when the central axis of the valve and center of the gripper are aligned within a tolerance, ϵ , which is the error margin that will allow the robot to rotate the valve.

B. The Exploratory Behaviors

Two exploratory behaviors, namely, a rotation around x-axis of the gripper, illustrated in Fig. 4(b), and a rotation around y-axis of the gripper, depicted in Fig. 4(c), were chosen to help distinguish between contact locations.

An exploratory behavior is achieved by rotating the gripper around one of its principal axes at a pivot point, as illustrated in the video attachment. Fig. 4(a) shows the principal axes of the gripper. The pivot point is at the gripper-valve boundary. The exploratory behavior must follow the contour of the valve, this ensures that there is always a contact with the valve and the sensors are properly stimulated. To achieve this, the robot uses the force/torque sensor to maintain a 3 N force along the z-axis of the gripper. It also maintains a neutral force along the x-axis and a neutral torque around the z-axis. This allows the robot to follow the contour of the valve. One of the advantages of the contour following strategy is that the gripper will adapt to the position and the orientation of the valve.

A behavior is achieved by rotating the gripper around one of its axes by a given angle-of-rotation. The angle-of-rotation can take an arbitrary value, but it should be sufficiently large to allow the tip of the gripper to make a contact with the central-axis of the valve when the angle-of-rotation is at maximum. In our experiments, the angle-of-rotation was set to 0.5 radians on either side of the origin.

1) **Rotation around x-axis**: As depicted in Fig. 4(b) the gripper rotates around the x-axis as indicated by the red arrows. This behavior produces signals that can distinguish a left-edge-contact, a right-edge-contact and a middle-contact. At the edge, the gripper is not obstructed by the valve's handle. Thereby, it is able to move freely counterclockwise when the gripper is at the left edge and clockwise when the gripper is at the right edge. However, in the opposite direction the gripper makes a contact with the valve's handle, exerting larger forces as well as increasing the relative distance between the valve and the gripper.

At the middle-contact location, the valve is obstructed on both sides. As a result, both in the clockwise and counterclockwise directions the gripper exerts a large force.

2) **Rotation around y-axis**: This behavior is particularly useful to discern a center-contact. A demonstration of this behavior is given in Fig. 4(c). The robot rotates the gripper as indicated by the green arrows. During the center-contact, the gripper is obstructed by the central axis of the valve, thereby, inducing a large force on the gripper that also pushes the gripper away from the valve. However, during the edge-contact, no forces are exerted on the gripper and it follows the contour of the valve's handle.

C. The learning task

The experimental setup shown in Fig. 1 is used collect data to train and test the learning algorithm. In this setup, the

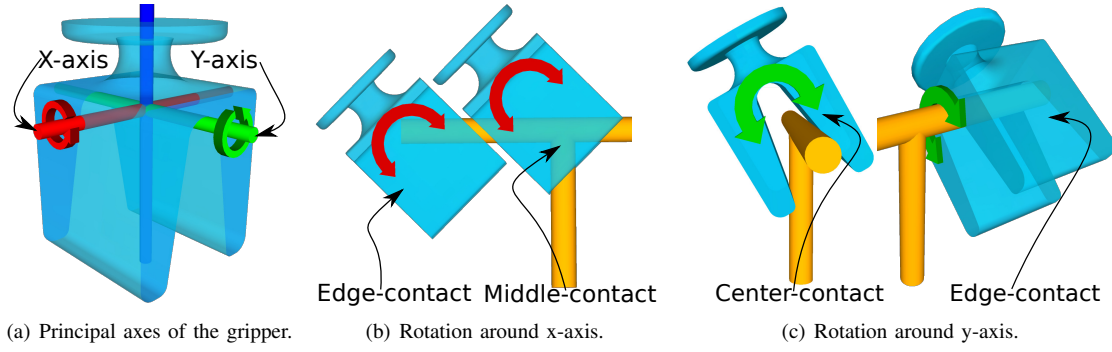


Fig. 4. Illustration of the exploratory behaviors, also illustrated in the video attachment. The arrows show the direction of the rotation. a) Shows the principal axes of the gripper. b) Depicts the rotation around x-axis behavior. c) Demonstrates the rotation around y-axis behavior.

gripper is attached to a KUKA-LWR arm. An ATI Mini-45 force/torque sensor is attached between the gripper and the robot's end-effector. A T-bar valve is placed in the robot's workspace. The robot performs an exploratory behavior in the workspace, which we will refer to as a trial henceforth.

Each trial is started by positioning the gripper a small distance from the valve. The robot moves the gripper until a contact is made, that is, it pushes forward along the z-axis of the gripper until a desired force is registered in the force/torque sensor. In our experiments, the desired force was set to 3 N. Then the robot performs the selected exploratory behavior. Two learning tasks were conducted, one for rotation around y-axis and one for rotation around x-axis.

1) **Rotation around y-axis:** Three positions are sampled for an edge-contact, where the contact area, l , between the valve and the gripper is varied. In the first position $l = \frac{1}{3}$ of the length of the gripper, in the second position $l = \frac{2}{3}$ of the length of the gripper, and in the third position there is a full contact. Similarly, for the center-contact three positions are sampled. The first position is selected such that the central-axis of the valve is aligned with the first $\frac{1}{3}$ of the length of the gripper, in the second position the central-axis of the gripper and the central-axis of the valve are aligned. The third position is a mirror image of the first position. These positions are chosen to expose the learning algorithm to positions that are valid, and within the tolerance of the control of the robot for safe manipulation. We also collect control samples, where the robot performs the behavior in the air, i.e., the robot is not making a contact with any object.

Four separate datasets were collected. Each set consists of six samples for no-contact, six samples for center-contact, and six samples for edge-contact. The samples for the edge-contact were collected on either side of the central axis of the valve, which resulted in twelve samples for the edge-contact. Fig. 5 shows the end-effector position and the force along the z-axis of the gripper.

2) **Rotation around x-axis:** This experiment follows the same pattern as for the rotation around y-axis. The only difference is that for an edge-contact, the contact area, l , between the valve and the gripper is less than or equal to $\frac{1}{3}$ of the length of the gripper. A contact area larger than $\frac{1}{3}$ of the length of the gripper is classified as a middle-

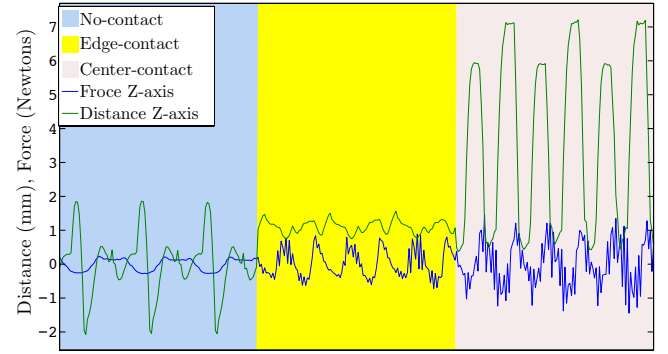


Fig. 5. Data generated during the rotation around y-axis behavior. Each colored region corresponds to a contact location.

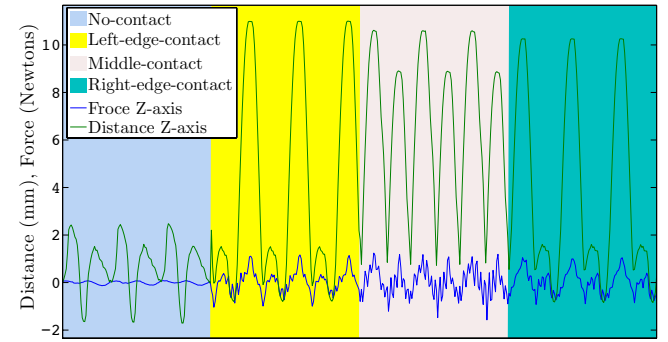


Fig. 6. Data generated during the rotation around x-axis behavior. Each colored region corresponds to a contact location.

contact. Hence, $l = \frac{1}{3}$, $l = \frac{1}{2}$ and $l = \frac{2}{3}$ are the three locations sampled for an edge. Four separate datasets were also collected for the rotation around x-axis behavior. Each set consists of six samples for each contact: no-contact, left-edge-contact, middle-contact, and right-edge-contact. Fig. 6 shows the end-effector position and the force along the z-axis of the gripper.

V. RESULTS

The feature vector for the learning algorithm is an eight-dimensional time-series that consists of the six-dimensional force/torque sensor data, the distance between the gripper and the valve, and the commanded angle-of-rotation. As

described in Section III-E, one HMM was trained for each behavior. A three state HMM (Fig. 3(b)) for the rotation around y-axis, and a four state HMM (Fig. 3(c)) for the rotation around x-axis.

We classify a contact state at the conclusion of the performed exploratory behavior, which is when the commanded angle-of-rotation reaches its maximum. At this point the HMM should have enough evidence to converge.

The performance of the classifiers were evaluated using fourfold cross-validation, i.e., one set out of the four independently collected sets is held as a test set, and the other three sets are used to train the classifier. This process is repeated four times to obtain an average performance for the classifiers. The results reported are based on this method of evaluation.

A. Effect of Increasing the Number of Clusters

We examined the effect of the number of clusters on the performance of the classifiers. Fig. 7 and Fig. 8 show the result of increasing the number of clusters. As the number of clusters is increased, the performance of the classifier also increases. This is expected as a larger number of clusters can form more expressive patterns that better capture the underlying phenomena.

The results for the rotation around x-axis (Fig. 8) show a strong support that increasing the number of clusters increases the performance of the classifier. We notice that as we increase the number of clusters from five to six, we get an increase in the accuracy from 52 ± 34 % to 88 ± 5 %, respectively. However, such an increase is not observed for the rotation around y-axis (Fig. 7). This is explained by the complexity of the data and the learning task. In the former, for example, the robot needs to distinguish between a contact with the left edge and a contact with the right edge. Fig. 6 shows the data for the rotation around x-axis. We notice that the difference in the time-series patterns are subtle, hence, requiring more clusters to be able to capture the differences.

However, in the case of the rotation around y-axis in Fig. 5, we notice that the data for different contact states are significantly different. As a result, even a small number of clusters is capable of successfully distinguishing the contact locations.

B. Confusion Matrices

Table II shows the confusion matrices for the rotation around x-axis. Note that, for this behavior, the instances of the left-edge and right-edge are combined together. As a result, in Table I there are 48 instances for the edge-contact. Hence, the misclassification of an edge-contact look large in numbers compared to a center-contact.

The confusion matrix for the rotation around y-axis is shown in Table II. It reveals that the algorithm confuses the middle-contact at a much larger rate than an edge contact. If we pay attention to the data in Fig. 6, we notice that as the gripper rotates, on either side of the rotation when it reaches the maximum angle-of-rotation, the data generated will be

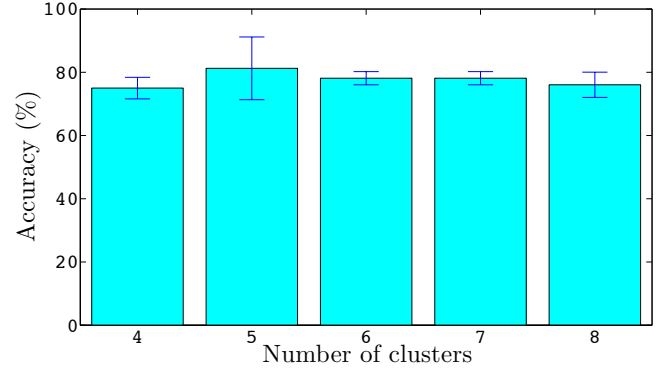


Fig. 7. Classifier accuracy for the rotation around y-axis behavior as a function of the number of clusters. The error bars are standard deviations.

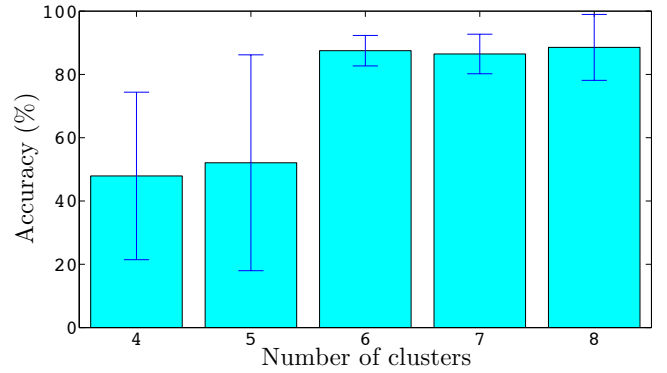


Fig. 8. Classifier accuracy for the rotation around x-axis behavior as a function of the number of clusters. The error bars are standard deviations.

TABLE I
CONFUSION MATRIX FOR THE ROTATION AROUND Y-AXIS BEHAVIOR,
ACCURACY $81 \pm 10\%$

\mathcal{N}	\mathcal{E}	\mathcal{M}	Class
24	0	0	No-Contact= \mathcal{N}
0	34	4	Edge-Contact= \mathcal{E}
0	14	20	Center-Contact= \mathcal{M}
100 %	71 %	83 %	Accuracy

TABLE II
CONFUSION MATRIX FOR THE ROTATION AROUND X-AXIS BEHAVIOR,
ACCURACY $88 \pm 5\%$

\mathcal{N}	$\mathcal{E}_{\mathcal{L}}$	\mathcal{M}	$\mathcal{E}_{\mathcal{R}}$	Class
36	0	0	0	No-Contact= \mathcal{N}
0	31	13	2	Left-Edge-Contact= $\mathcal{E}_{\mathcal{L}}$
0	4	23	3	Middle-Contact= \mathcal{M}
0	1	0	31	Right-Edge-Contact= $\mathcal{E}_{\mathcal{R}}$
100 %	86 %	64 %	86 %	Accuracy

similar to either left-edge or right edge. Making it harder to distinguish a middle-contact.

The misclassification of the middle-contact exhibits a strong bias toward left-edge-contact. An analysis of the results revealed that this is because we classify a contact state when the angle-of-rotation is maximum. It turns out that for the left-edge-contact the maximum angle-of-rotation coincides with the gripper being rotated counterclockwise as

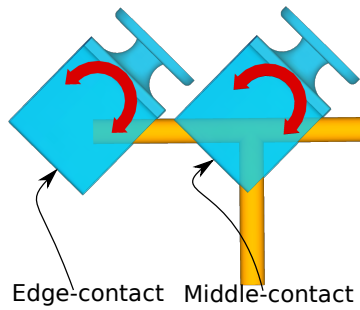


Fig. 9. Illustration of the gripper when the angle-of-rotation is maximum. Showing similarity between a left-edge-contact and a middle-contact.

illustrated in Fig. 9. This makes the two contacts: the left-edge-contact and the middle-contact look similar, increasing the chances of misclassification. Hence, giving rise to the strong asymmetry observed in the results.

VI. DISCUSSION

A. Experimental setup

The experimental setup is designed such that the position and orientation of the valve relative to the robot does not affect the exploratory behaviors. The contour following requirement was introduced to ensure that the exploratory behaviors stimulate the sensors similarly irrespective of the position and orientation. In our experiments, the gripper was positioned near the valve. In its current state, lack of large area body-tactile sensors makes it difficult to conduct a full blind search of the robot's workspace. Collision with the valve, where the contact is not with the gripper cannot be handled.

B. Overfitting

We use three methods to reduce the risk of a classifier overfitting the training data. Firstly, we take advantage of the PCA to reduce the dimensionality of the sensor data from seven to two. Secondly, MML prefers a simple hypothesis over a more complex one. Thereby, reducing the risk of overfitting. Finally, we used fourfold cross-validation to obtain an average performance and the standard deviation for each classifier to reflect such errors.

VII. FUTURE WORK

In future we will consider a larger repertoire of behaviors that provide more detailed information such as the orientation of the valve or whether the valve is stuck. The set of behaviors can be learned using reinforcement learning in simulation. We will also look into improving the method to autonomously choose an exploratory behavior that, based on its current belief, will be more informative in eliminating and/or increasing the probability of the contact state estimate. Furthermore, we will study the effectiveness of the method for underwater applications.

Future research will also consider relaxing the approximate knowledge of the location of the valve, where a blind search of the robot's workspace will be conducted for detecting the valve. We will also look into extending the

method to handle other type of valve found in industrial environments.

The clustering method provides a probabilistic cluster membership. Further studies will look at ways to take advantage of the probabilistic, symbolic representation. The prediction power of the method can be improved by taking advantage of the cluster membership probabilities. Such a system can be implemented using a feed-forward mechanism. The learned models, based on the current belief of the system, can project the next expected symbol in the sequence. This projection can be used to put more weight on the probability of the projected symbol. This will allow a symbol that might be of lower probability, but in consensus with the projected symbol to be selected. This will result in a reduction of noise in the sequence due to cluster misclassification.

VIII. CONCLUSION

We have presented a method that uses machine learning to successfully perceive a contact state between the gripper of a robot and a T-bar valve. The main contribution of this paper is a method that can robustly perceive a contact state using non-vision information such as force/torque sensor data and the robot's proprioceptive information.

REFERENCES

- [1] D. M. Lane, F. Maurelli, P. Kormushev, M. Carreras, M. Fox, and K. Kyriakopoulos, "Persistent autonomy: the challenges of the PANDORA project," in *Proc. IFAC MCMC*, 2012.
- [2] PANDORA, "Persistent Autonomy through learNing, aDaptation, Observation, and Re-plAnning," 2012. [Online]. Available: <http://www.persistentautonomy.com>
- [3] N. Jamali and C. Sammut, "Slip prediction using hidden Markov models: multidimensional sensor data to symbolic temporal pattern learning," in *Proc. IEEE Int. Conf. Robot. Auto.*, 2012, pp. 215–222.
- [4] M. A. Abidi, R. O. Eason, and R. C. Gonzalez, "Autonomous robotic inspection and manipulation using multisensor feedback," *Computer*, vol. 24, no. 4, pp. 17–31, 1991.
- [5] D. A. Anisi, E. Persson, and C. Heyer, "Real-world demonstration of sensor-based robotic automation in oil & gas facilities," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sept. 2011, pp. 235–240.
- [6] D. M. Lane, et al., "AMADEUS: advanced manipulation for deep underwater sampling," in *IEEE Robot. Auto. Mag.*, IEEE, vol. 4, no. 4, pp. 34–45, 1997.
- [7] J. Lemburg, P. Kampmann, and F. Kirchner, "A small-scale actuator with passive-compliance for a fine-manipulation deep-sea manipulator," in *Proc. MTS/IEEE OCEANS*, 2011, pp. 1–4.
- [8] G. Marani, S. K. Choi, and J. Yuh, "Underwater autonomous manipulation for intervention missions AUVs," *IEEE J. Ocean. Eng.*, vol. 36, no. 1, pp. 15–23, Jan. 2009.
- [9] M. Prats, J. J. Fernandez, and P. J. Sanz, "Combining template tracking and laser peak detection for 3D reconstruction and grasping in underwater environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 106–112.
- [10] S. Ahmadzadeh, P. Kormushev, and D. Caldwell, "Autonomous robotic valve turning: a hierarchical learning approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013.
- [11] C. S. Wallace and D. L. Dowe, "MML clustering of multi-state, Poisson, von Mises circular and Gaussian distributions," *Statistics and Computing*, vol. 10, no. 1, pp. 73–83, 2000.
- [12] C. S. Wallace, D. M. Boulton, and B. C. S. Wallace, "An information measure for classification," *Computer Journal*, vol. 11, no. 2, pp. 185–194, 1968.
- [13] Vanilla Snob, accessed March 2013. [Online]. Available: <http://www.datamining.monash.edu.au/software/snob/%23vanilla>
- [14] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.