Contact State Estimation using Machine Learning

Nawid Jamali, Petar Kormushev and Darwin G. Caldwell

***Abstract*— In this paper we present an approach that uses machine learning to determine the location of a contact between a gripper and a T-bar valve based on force/torque sensor data. The robot performs an exploratory behaviour that produces distinct force/torque data for each contact location of interest: no contact, a contact aligned with the central axis of the valve, and an off-center contact. Probabilistic clustering is utilised to transform the multidimensional data into a one-dimensional sequence of symbols, which is then used to train a hidden Markov model classifier. We present the results of an experiment where the learned classifier can predict a contact location with an accuracy of 97% on an unseen dataset.**

I. Introduction

There is an increased interest in developing robots that can dexterously manipulate objects. Events such as the 2011 Fukushima Daiichi nuclear disaster have sparked an increased interest in the research community to develop disaster response robots that can autonomously inspect and manipulate objects in hazardous environments. In such environments often sensing modalities such as vision are limited or unreliable, hence, direct contact information is important to successfully inspect and/or manipulate an object. Applications of such a system include search and rescue operations and underwater manipulation of objects.

In this paper we propose a method that can be used to inspect an object. As part of the EU FP7 PANDORA project [1], [2] we investigate an object inspection task that consists of determining the location of a contact between a robotic gripper and a T-bar valve. We use an exploratory behavior that at different contact locations induces the force/torque sensor differently. This produces a multidimensional time-series data on the contact location. A contact location, namely, an edge-contact, a center-contact and no-contact can be detected by studying the temporal patterns in the data. Our contribution is a robust autonomous contact determination based on machine learning that relies on non-vision sensing, namely, a force/torque sensor.

To learn a classifier for the gripper-valve contact, we use 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 location classification, we also introduce the use of active exploratory actions to enable the robot to perceive the contact state.

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 Technologia, via Morego 30, 16163 Genoa, Italy email: {nawid.jamali, petar.kormushev, darwin.caldwell}@iit.it

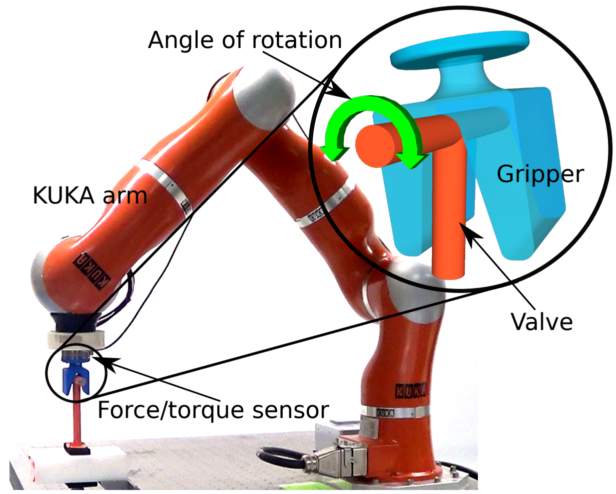


Fig. 1. Experimental setup.

The learning algorithm is divided into two stages: dimensionality reduction using clustering; and temporal pattern extraction. In the first stage, probabilistic clustering [4] 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 analyses the sequence and builds a probabilistic model of temporal patterns that correspond to different abstract contact conditions.

Fig. 1 shows the experimental setup, which consists of a gripper attached to a KUKA 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 an exploratory behavior, which, as illustrated by the green arrow in Fig. 1, is a periodic rotary movement around the handle of the valve with a given angle-of-rotation. This action induces the force/torque sensor differently at different contact locations. For example, a contact at the center limits the angle of rotation, resulting in the force/torque sensor registering higher values, which is different to an edge-contact. We show that using this method the robot can successfully predict a contact location.

II. Background

Earlier research in valve detection and manipulation assumes a structured environment, where the robot stores detailed information about the environment [5], [6]. 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 [5], and detect orientation of a valve handle and manipulate the valve without over-tightening/loosening [6]. 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 [7], [8].

Anisi et al. [6] 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 inductive proximity sensor limits the use of the system to metallic objects.

Marani et al. [9] 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. [10] 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. [11] 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 sensing information for the learning algorithm is produced by a force/torque sensor. We also include the control signal, that is, the commanded angle-of-rotation, resulting in a seven-dimensional time-series data. To learn concepts from a multi-dimensional time-series we divide the problem into two stages: dimensionality reduction using clustering and temporal pattern extraction.

The first step of the analysis involves preprocessing the data. The clustering method assumes that all of the variables are independent [4]. Hence, in the first stage of training, the 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 output of the PCA is 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. $S = \{\text{BED}\underline{\text{BCCA}}\text{CDDADEC}\underline{\text{BCCA}}\text{EBDA}\ldots\}$ 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 analysed 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 is used to predict the state of 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 before any further analysis is performed. The preprocessing procedures are implemented in MATLAB.

1) Zero-mean normalization: The force/torque data are zero-mean normalized from the point of first contact. This process is necessary to make the forces and the torques comparable between different trials.

2) Filtering: The force/torque data are sampled at 500 Hz, the control signals change at 0.5 Hz. Hence, the force torque data is filtered using a digital filter with a 3 dB point of 5 Hz, which is ten times the frequency of the control signals.

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 orthogonal to each other, i.e, they 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-variance[1] to total-variance has exceeded a predetermined threshold, usually 0.8. In our training data set, the first component exceeded this threshold by reaching a ratio of 0.96. Hence, in the subsequent analysis we only consider the first principal component.

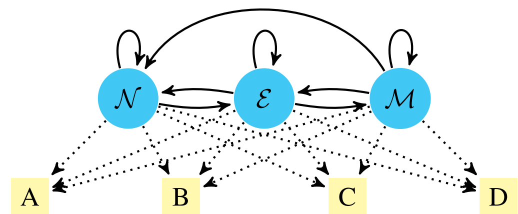
C. Clustering

We use probabilistic clustering [4], which uses the minimum message length (MML) principle as the optimization criterion, to build a Gaussian mixture model of the data. In this section we explain the theory behind MML clustering and show how we use it to extract features.

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

$$
\begin{aligned}
S_{original} &= \{\text{BCCDDCAAAABAABB}\ldots\} \\
S_{labelled} &= \Big\{(\text{B},\mathcal{N})(\text{C},\mathcal{N})(\text{C},\mathcal{N})(\text{D},\mathcal{N})(\text{D},\mathcal{N})(\text{C},\mathcal{E}) \\
&\quad (\text{A},\mathcal{E})(\text{A},\mathcal{E})(\text{A},\mathcal{E})(\text{A},\mathcal{E})(\text{B},\mathcal{M}) \\
&\quad (\text{A},\mathcal{M})(\text{A},\mathcal{M})(\text{B},\mathcal{M})(\text{B},\mathcal{M})\ldots\Big\}
\end{aligned}
$$

(a) Example of a feature vector.



(b) A three state hidden Markov model.

Fig. 2. A feature vector and a hidden Markov model. The letters A, B, C and D indicate membership to the corresponding cluster. The letters $\mathcal{N}$, $\mathcal{E}$ and $\mathcal{C}$ represent contact states: no-contact, edge-contact, center-contact, respectively

The MML principle is based on the information-theoretic Bayesian principle of inductive inference[12], [4]. 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.

$$Pr(H|D) = \frac{Pr(H\&D)}{Pr(D)} = \frac{Pr(H)Pr(D|H)}{Pr(D)} \tag{1}$$

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

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.

D. 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 the 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. 2(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. 2(b) is an HMM used to learn a representation for the the emerging temporal patterns. The HMM has three states, one for each contact condition. It is 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 method was tested using the setup (Fig. 1) described in Section I. In this section we will define the contact locations, which is followed by a description of the exploratory behavior and a detailed description of the experiment.

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.

2) ***Centre-contact:*** In center-contact, the central axis of the valve is covered the gripper.

B. The exploratory behavior

The exploratory behavior is chosen to produce distinctive signals depending on the location of a contact. A roll that is pivoting on the valve will produce a different signal when it is on either one of the edges compared to a contact that is at the center of the valve, where the movement of the gripper is restrained by the valve's axis. Thereby, inducing the force/torque sensor. The angle-of-rotation can take an arbitrary value, but it should be sufficiently large to guarantee the tip of the valve makes a contact with the central-axis of the valve when the gripper is at the center. In our experiments the angle of rotation was varied by 0.5 radians on either side of the starting position.

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 a gripper is attached to a KUKA 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 the exploratory behavior in the workspace, which we will refer to as a trial henceforth.

Fig. 3 shows the configuration of the valve in the beginning of each trial. 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

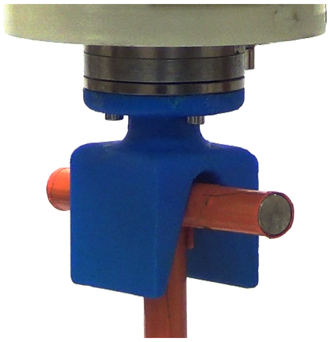


Fig. 3. Gripper placement.

TABLE I

CONFUSION MATRIX 3-CLUSTERS AND 4-CLUSTERS.

$\mathcal{N}$	$\mathcal{E}$	$\mathcal{M}$	$\mathcal{N}$	$\mathcal{E}$	$\mathcal{M}$	Class
Three clusters			Four clusters			
9	0	0	9	0	0	No-Contact=$\mathcal{N}$
0	17	1	0	17	1	Edge-Contact=$\mathcal{E}$
0	2	7	0	0	7	Center-Contact=$\mathcal{M}$
100	94	78	100	94	100	Accuracy (%)

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 contact positions, within the tolerance of the control of the robot for safe manipulation.

Two separate datasets were collected: a training set and a testing set. Each set consists of nine samples for no-contact, nine samples for center-contact, and nine 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 eighteen samples for the edge-contact. Fig. 4 shows the force/torque data where the colored regions represent a specific contact location.

V. RESULTS

The feature vector for the learning algorithm is a seven-dimensional vector that consists of the six-dimensional force/torque sensor data, and the control data, that is, the commanded angle-of-rotation. An HMM model was learned using the training set. We also varied the number of clusters to study the effect of this parameter on the performance. Table I shows the confusion matrix for three clusters and four clusters. We get an accuracy of 92% with three clusters, which is increased to 97% with four clusters. This is expected as an increase in the number of clusters increases the expressive power of the models.

VI. CONCLUSION & FUTURE WORK

We have presented a method that can successfully predict the location of a contact between a gripper and a valve using only force/torque sensor data. The presented approach is suitable for autonomous inspection of objects. In the future we will consider a larger set of behaviors that can result in a more detailed information such as the orientation of the valve or whether the valve is stuck. We will also study the performance of the proposed method in underwater scenarios.

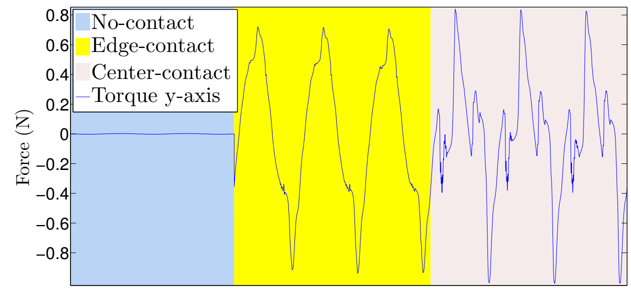


Fig. 4. Force/torque data showing the torque data around the y-axis. Each colored region corresponds to a specific contact location.

REFERENCES

[1] D. M. Lane, F. Maurelli, P. Kormushev, M. Carreras, M. Fox, and K. Kyriakopoulos, "Persistent autonomy: the challenges of the PANDORA project," *Proceedings of IFAC MCMC*, 2012.

[2] "Persistent Autonomy through learNing, aDaptation, Observation, and Re-plAnning (PANDORA). Online: http://persistentautonomy.com/."

[3] N. Jamali and C. Sammut, "Slip prediction using hidden Markov models: Multidimensional sensor data to symbolic temporal pattern learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 215–222.

[4] 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.

[5] 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.

[6] D. A. Anisi, E. Persson, and C. Heyer, "Real-world demonstration of sensor-based robotic automation in oil & gas facilities," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept. 2011, pp. 235–240.

[7] D. M. Lane, J. B. C. Davies, G. Casalino, G. Bartolini, G. Cannata, G. Veruggio, M. Canals, C. Smith, D. J. O'Brien, M. Pickett, G. Robinson, D. Jones, E. Scott, A. Ferrara, D. Angelleti, M. Coccoli, R. Bono, P. Virgili, R. Pallas, and E. Gracia, "AMADEUS: advanced manipulation for deep underwater sampling," *Robotics Automation Magazine, IEEE*, vol. 4, no. 4, pp. 34–45, 1997.

[8] J. Lemburg, P. Kampmann, and F. Kirchner, "A small-scale actuator with passive-compliance for a fine-manipulation deep-sea manipulator," in *OCEANS 2011*, 2011, pp. 1–4.

[9] G. Marani, S. K. Choi, and J. Yuh, "Underwater autonomous manipulation for intervention missions AUVs," *Ocean Engineering*, vol. 36, no. 1, pp. 15–23, Jan. 2009.

[10] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 106–112.

[11] S. Ahmadzadeh, P. Kormushev, and D. Caldwell, "Autonomous Robotic Valve Turning: A Hierarchical Learning Approach," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[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. URL: http://www.datamining.monash.edu.au/software/snob, accessed March 2013.