Stabilizing Novel Objects by Learning to Predict Tactile Slip

Filipe Veiga¹, Herke van Hoof¹, Jan Peters^{1,2}, Tucker Hermans¹

Abstract—During grasping and other in-hand manipulation tasks maintaining a stable grip on the object is crucial for the task's outcome. Inherently connected to grip stability is the concept of slip. Slip occurs when the contact between the fingertip and the object is partially lost, resulting in sudden undesired changes to the objects state. While several approaches for slip detection have been proposed in the literature, they frequently rely on previous knowledge of the manipulated object. This previous knowledge may be unavailable, seeing that robots operating in real-world scenarios often must interact with previously unseen objects.

In our work we explore the generalization capabilities of well known supervised learning methods, using random forest classifiers to create generalizable slip predictors. We utilize these classifiers in the feedback loop of an object stabilization controller. We show that the controller can successfully stabilize previously unknown objects by predicting and counteracting slip events.

I. INTRODUCTION

Robust grasping and dexterous in-hand manipulation of objects remain challenging tasks in robotics. A host of issues contribute to the difficulty of in-hand manipulation including finger positioning, estimation of object and finger dynamics, finger coordination, and dynamic grip stability [1–3]. Furthermore, in realistic open-ended environments, robots will encounter many novel objects. Therefore, we claim it is crucial that strategies for grasping or manipulation do not rely on object models. Instead, strategies should generalize previous experiences for use with novel objects in order to be useful in these scenarios.

Slip, the partial loss of contact between a robot finger and object, is a fundamental concept in manipulation tasks [4, 5]. Accurately detecting slip provides rich feedback for a robot to maintain grip stability during manipulation [5, 6]. Such feedback could additionally be used by a robot to reposition objects in its hand through controlled sliding [2]. We believe endowing a robot with the ability to detect slip will enable more reliable and more sophisticated manipulation of objects. We also consider slip prediction as a means to improve manipulation capabilities. Predicting slip allows the robot to react prior to slip occurring, compensating for controller latencies and negating all undesired changes to the object's state.



Fig. 1. Example data collection trial: the robot holds the object against a vertical surface; it then slowly moves the finger away inducing a slip event.

While several tactile sensing technologies have been used in robotic manipulation applications [4], some show a number of attractive benefits compared to alternative sensing modalities for detecting and predicting slip. Tactile sensors that function at high frequencies allow for the detection of incipient slip, letting the robot react before gross slip occurs [6]. Such quick feedback is crucial to combat undesired object dynamics caused by slip. Beyond this, tactile sensors are directly in contact with the object, not suffering from visual occlusion. Furthermore tactile sensors that have high spatial resolution provide richer and more direct feedback of the object of interest's behaviour when compared to joint encoders or force-torque sensors. Slip detection based on force and joint sensing modalities also frequently relies on strong modelling assumptions.

As a step towards robust in-hand manipulation, this work focuses on the detection and prediction of slip via tactile sensing. We incorporate our learned slip classifiers into a feedback controller to perform grip stabilization. We take a data-driven approach, where the robot collects tactile data of objects being held stably as well as slipping, as illustrated in Fig. 1. Based on this data, the robot learns a classifier to detect or predict slip events. Compared to approaches based on modelling and analysis of slip physics, our approach has the advantage that the friction coefficient and shape of the object need not be known. We compare different learning methods for classification trained on data collected on common household objects. We examine the ability of the slip classifiers to generalize to previously unseen objects. We ultimately validate our slip classification approach through use in a grip stabilizing feedback controller. We analyze the differences in control performance when using detection and

¹Computer Science Department, TU Darmstadt, {veiga, hoof, peters, hermans}@ias.tu-darmstadt.de

²Jan Peters is also with the Max-Planck-Institut für Intelligente Systeme, Tübingen, Germany

The research leading to these results has received funding from the European Communitys Seventh Framework Programme (FP7/20072013) under grant agreement 610967 (TACMAN). The authors would like to thank Janine Hoelscher for the implementation of features used in our experiments.

prediction classifiers.

We present the remainder of this work as follows. Section II gives an overview of related research efforts on slip detection and tactile sensing. We formalize slip detection and slip prediction as similar learning problems and discuss possible approaches in Sec. III. We introduce our grip stabilization controller in Sec. IV. We follow this with a description of our experimental setup and results of our evaluation in Sec. V. We conclude with a discussion of future applications and extensions of our work in Sec. VI.

II. RELATED WORK

In this section, we review related work on slip detection and tactile sensing. We first give an overview of various modalities used to predict slip, and then focus on approaches using tactile sensing specifically. Finally we give a short overview on approaches for detecting grasp stability.

Other than tactile sensing, modalities such as vision, forcetorque sensing, and range sensing have been used for slip detection. As an example of the first, Ikeda et al. [7] use a camera to detect deformations of a fingertip. Problems due to occlusion were avoided by gripping a transparent object, limiting the possible applications. The work of Viña et al. [8] is an example of using force-torque sensing for slip detection. Using learned models in conjunction with Coulomb's friction model, they determine continuous bounds on the forces and torques that can be withstood by a grasp before slipping occurs. A laser-based range sensor is proposed by Maldonado et al. [9] for detecting slip between a fingertip and object.

An alternative body of literature addresses the identification of slip explicitly using tactile sensing. An early example is the work of Tremblay et al. [6], who employ a hand-tuned slip detector based on the amount of vibrations, inspired by the FAII receptors in humans. Human-inspired features are also used by Romano et al. [10] to stabilize grasps during the lifting and transport of objects. Their human inspired features are used as inputs to a high level control policy composed of phase specific lower level controllers. Van Anh Ho et al. [11] treat a tactile array as an image and compute sparse optical flow to estimate "slipped points". A point feature is classified as a slipped point if the change in feature position over two subsequent frames is greater than a predefined slip threshold. The method classifies the object as slipping based on the percentage of tracked features identified as slipped points. Heyneman and Cutkosky [12] present a method for differentiating between hand-object and object-world slip events. Spectral decomposition is used to separate vibration signals caused by the slip between the object and finger from external vibrations. Reinecke et al. [13] compare three different approaches for slip detection of objects grasped between two fingers equipped with BioTac sensors. The first approach performs model-based slip detection by estimating the contact location of forces applied by the fingers using the tactile sensors. Slip is predicted if the forces applied lie outside the estimated friction cones. The second method, following a method proposed by Fishel [14], uses spectral analysis of vibration data to detect slip. If the energy content

between 30 and 200 Hz is above a nominal value slip is believed to be occurring. The third method uses a random forest classifier to detect slip events. While the three methods are compared to one another they are only evaluated on a single object. Schopfer et al. [15] present a method for slip detection by learning a neural network trained on spectral features computed on a large tactile array. The method uses the robot kinematics to determine the end effector velocity when sliding across a known material and estimates this velocity as the ground truth slip. Li et al. [16] use tactile sensing to adapt grasps in order to maintain grasp stability. They predict if a grasp is stable on a combination of kinematic data from the robot hand and the raw tactile information from BioTac sensors embedded in the robot's fingers.

Although we are interested in explicit detection of slip across different objects, the related task of detecting grasp stability is also relevant. For example, Bekiroglu et al. [17] use the moments of the activation patterns of tactile arrays to assess the current grasp as stable or unstable. Dang and Allen [18] present an alternative method for detecting stable grasps from tactile sensing using a learned dictionary of contact locations. In contrast, Madry et al. [19] used deep learning to find tactile features for assessing grasp stability.

III. LEARNING TO PREDICT SLIP

In this section we describe our approach to detecting and predicting slip using tactile feedback and supervised learning methods. We formalize the prediction problem as a classification problem where we wish to learn a function, $f(\cdot)$, to classify the current state as slip or non slip: $c_t = f(\phi(\mathbf{x}_{1:t}))$ where $c_t \in \{c_{\text{slip}}, c_{\text{non_slip}}\}$ is the state class at time t and $\phi(\cdot)$ is a feature function over the raw sensor data $\mathbf{x}_{1:t}$ covering sensor samples from initial time 1 to the most recent time step t.

Using this formalization we detect slip at the current time step. While detecting slip is important, being able to predict slip allows the robot to react before any undesired changes to the object state occur. In our approach, slip prediction is performed using the previous formalization while training the classification methods with future labels $c_{t+\tau_f} = f(\phi(\mathbf{x}_{1:t}))$ where τ_f is a positive step size, indicating how many steps in the future the predictor is trained for. Detection is then the special case of prediction when $\tau_f = 0$.

We aim at having stable classification throughout the slip event without compromising on the generalization capabilities of the learned classifier. We also wish to achieve prediction of t_{slip} as early as possible. Early prediction of the class transitions from $c_{\text{non_slip}}$ to c_{slip} will provide for the possibility of more robust control during object manipulation. With these three goals in mind, we explore how the feature function, $\phi(\cdot)$, and the use of different classification methods, $f(\cdot)$, influence the outcome of the prediction.

A. Feature Comparison

Our raw tactile data is extracted from the BioTac [20], a multi-channel tactile sensor whose design was inspired by the



Fig. 2. The Mitsubishi PA-10 robot arm equipped with a BioTac tactile sensor. An RGB-D camera records the experiments to facilitate the data labeling process.

human finger. The sensor provides several channels including an array of impedance-sensing electrodes measuring the local pressure on the fingertip, E, a pressure transducer which measures low frequency, P_{dc} , and high frequency, P_{ac} , pressure variations, and a set of heaters coupled with a thermistor that measure temperature, T_{dc} , and temperature flow, T_{ac} . A single timestep generates a vector of 44 values which make up x_t . The electrodes account for 19 of these elements. The high frequency P_{ac} values are sample 22 times per single data frame accounting for half of vector. The remaining P_{dc} , T_{dc} and T_{ac} channels are all single values. These 44 values are sampled at a rate of 100 Hz. Thus a single timestep in terms of classification is actually accumulating data over a small time window.

The feature function $\phi(\cdot)$ takes several forms each of them representing distinct assumptions about the detectability of c_t from the raw sensor data $\mathbf{x}_{1:t}$. If we assume the class label to be directly observable from the current sensor reading, c_t than a memoryless feature $\phi(\cdot)$ takes the form $\phi(\mathbf{x}_{1:t}) =$ \mathbf{x}_t and is denoted as the single element feature function. Another interesting case is when we assume that c_t depends not only on the current sensor data but also on the change of the data with respect to the previous time step. In this case $\phi(.)$ takes the form $\phi(\mathbf{x}_{1:t}) = [\mathbf{x}_t, \Delta \mathbf{x}_t]$ and is denoted as the delta feature function. Finally, the last assumption considers c_t heavily dependent on past data. This can be interpreted as an attempt to teach a classifier the tactile patterns that lead to slip, and can be represented through the feature function $\phi(\mathbf{x}_{1:t}) = \mathbf{x}_{t-\tau:t}$ where τ is the size of the time window of past data to be considered. This feature function is denoted as time window feature function. In addition to these feature functions we also show results when using the features introduced by Chu et al. [21]. Originally designed with the goal of object property learning, these features are divided into four distinct groups that try to depict not only the compliance, roughness, and thermal properties of the object, but also the correlation between the electrode data retrieved from the sensor. For a more detailed description of these features please refer to [21].

B. Classification Methods

For classification, the methods used in this work are support vector machines and random forest classifiers. We have chosen these methods as they are well understood in the machine learning community and have been successfully applied to a number of problems in computer vision, which also deal with classification of high-level concepts from complex sensor data [22–25].

Support vector machines (SVM) are discriminate classifiers separating the training samples by partitioning the feature space with a single decision boundary [26]. Each partition of the feature space defined by the decision boundary represents a single class. The decision boundary is chosen with respect to the closest samples of each class referred to as the support vectors. During training the decision function which maximizes the classification margin, defined as the sum over the distances to each support vector, is found. The resulting linear classifier evaluating feature vector \mathbf{z} takes the form:

$$f(\mathbf{z}) = \sum_{i=1}^{k} {}^{i} \alpha(\mathbf{z} \cdot {}^{i} \mathbf{z}) + b$$

where ${}^{i}\alpha$ is the weight associated with the i^{th} support vector, ${}^{i}z$, and b is a constant offset term. The support vectors and weights can be found efficiently by solving a quadratic program.

A random forest classifier is an ensemble of randomly trained binary decision tree classifiers [27]. Each decision tree classifies a given test example independently. The result of the entire forest is obtained by averaging over the distributions of the leaves reached in each of the trees. The class with the highest probability is then selected as the corresponding class for the current sample. Each decision tree is a binary tree where all non-terminal nodes have an associated splitting function, which decide if the currently evaluated example should traverse down the tree following the left or right branch. Leaf nodes contain a probability distribution over the class labels of training examples which reach this node. Tree training consists of selecting the feature and threshold to split at each node. These values are selected through the optimization of a specific performance criterion.

In this work we minimize the Gini impurity score to acquire the node splitting function. Randomness in each tree is introduced during training by providing only a random subset $s \in \mathscr{S}$ of the complete feature set \mathscr{S} to the optimization of the node splitting criterion. We perform hyper-parameter optimization on the number of trees per forest and the size of the feature subset *s* by maximizing the *F*_{score} using grid search. The *F*_{score} is introduced in Section V-C. Trees have no maximum depth, and nodes are split until they are either pure (all samples have the same label) or contain only two samples. All classifier implementations in our work come from the scikit-learn library for python [28].

IV. STABILITY CONTROL USING SLIP PREDICTION

We propose a simple feedback controller which makes use of the slip classifier's discrete output in the feedback loop. In brief, the controller increases the force applied to the object when slip is predicted to occur until the robot no longer predicts slip. By increasing the force in the direction of the contact normal, tangential forces should not increase and the force should stay within the friction cone of the contact location. We give a detailed explanation of our implementation below. We show evaluations of our controller in Section V-E.

We assume the sensor is in contact with the object when control begins since we can easily detect contact using thresholds on the sensor pressure values. At each timestep the classifier evaluates if slip is occurring. If the robot predicts slip, then the controller increases the force, $F_n[t]$, applied normal to the point of contact, P_c at time t. If the robot predicts no slip, then the current force is maintained. The robot increases the force applied by a fixed amount δ .

$$F_N[t+1] = \begin{cases} F_N[t] + \delta \hat{F}_N[t+1] & \text{if } c_t = c_{\text{slip}} \\ F_N[t] & \text{otherwise} \end{cases}$$

where $\delta \hat{F}_N$ denotes the unit contact normal.

We estimate both the contact normal \hat{F}_n and point of contact P_c using the electrode sensors in the BioTac using a method detailed to us by the manufacturers. We estimate P_c as the center of applied pressure on the BioTac skin using a simple interpolation method. We average the spatial locations of all electrodes weighting this average by the responses at each electrode to determine the point of highest pressure. The contact normal is estimated in an analogous manner. The surface normals of all sensing electrodes are averaged, weighted again by the electrode responses. We normalize by the magnitude of the resulting vector to give us a unit vector in the direction of applied contact force.

V. EXPERIMENTAL EVALUATION

We now explain in detail our experimental evaluation. Section V-A describes our robot platform and the sensors used. The experiments performed for data collection purposes as well as the description of the data set used for the training and testing of the learning methods are described in section V-B. We present results of our detection and prediction methods as well as comparisons to other methods in Sections V-C and V-D. Finally, we show results for our grip stabilization controller in Section V-E.

A. Experimental Setup

Our experimental setup consists of a Mitsubishi PA-10 robot with seven degrees of freedom. A BioTac tactile sensor is rigidly mounted to the force-torque sensor as a single finger for manipulation. Additionally to the on-board sensing, an external RGB-D camera (Asus Xtion Pro) was placed in order to capture the robot's work space to record each trial. The complete setup can be seen in Figure 2.

B. Data Collection

A single data collection trial is initialized by placing an object between the robot finger and a vertical plane. The robot arm then slowly moves away from the plane



Fig. 3. The objects comprising our data set. We selected objects covering a range of shapes and stiffness in order to adequately test classifier generalization. From left to right the objects names are ball, box, cup, marker, measuring stick, tape, and watering can.

at a constant task-space velocity (1 cm/s) until slip occurs between the object and finger. The robot continues moving until the object eventually falls to the ground. Data was collected for each object from the set of seven common house old objects depicted in Figure 3. Ten trials were performed on each object using different initial contact locations and object poses, producing a total of seventy trials. The labeling of the data was performed with the aid of the camera. Slip was labeled at all times at which the object was visibly sliding between the finger and the vertical plane. Figure 4 show an example of the labels and the corresponding visual feedback at each label transition.

Two observations can be made regarding the data. First we notice that slip occurs for very short durations. This results in a low ratio between the number of slip labels and non-slip labels. Secondly, the labeling accuracy of the human expert may be biased by the camera data which is captured at a much slower rate then the tactile data. This potentially results in a lag in the labeling causing an inaccurate measurement of t_{slip} by the human expert. It is worth mentioning that the vision labelling could have been done autonomously by the robot but we wished to simplify the experimental procedure.

Both of the previous observations are taken into account when analyzing the results in Sections V-D and V-E. Appropriate measurements for classifier performance are used, taking into account the low ratio of positive examples. We also expect the transition step t_{slip} to be estimated earlier by the learning algorithms when compared to the human expert. We examine this in evaluating our slip controller in Section V-E.

C. Slip Detection

We now analyze the described methods focusing on detection rates and generalization capabilities. We start by introducing the experimental procedure and the criteria used to compare the methods. We split the collected experimental data into training sets containing seven trials for each object. The remaining three trials per object were amassed into the test set. In all experiments we set the time window feature of



Fig. 4. An example slip trial. The blue plots represent the ground truth labels. Images below correspond to the box being held stably, slipping, and finally loosing contact. Ground truth labels were given by a human observer after the trial was completed. The solid vertical lines correspond to the start and end of the human labeled slip event.

 $\tau = 10$. We examined a range of values for τ and found this to perform the best, although not by a significant margin.

For the first set of experiments we compare the different feature functions and classifiers described above in Section III. We examine two different training scenarios. The first approach trains a single classifier for each object. In the second scenario a single classifier was trained across all available training data. We report the results of these experiments in Table I. This table summarizes the F_{score} for each classifier. The F_{score} is a weighted average of the precision and recall measures and has the form

$$F_{score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Precision depicts the ratio between accurate positive classifications and total positive classifications

$$precision = \frac{true \ positives}{true \ positives + false \ positives}$$

while recall is the ratio between accurate positive classifications and positive examples

$$recall = \frac{true \ positives}{true \ positives + false \ negatives}$$

We choose to report the F_{score} instead of classification accuracy, as we care much more about detecting when slip occurs then when it doesn't. Since the majority of labels are negative (no slip) classification accuracy could be quite high while not detecting any of the slip events.

We see that random forests with the single time element feature performed best in detecting slip on average. The single time element features perform best on four of the seven object classes. For two of the remaining three objects–the box and the watering can– the linear SVM with delta features performed best. The final object, the cup, was best classified by the linear SVM with the time window. On average the delta feature random forest classifier performed at 95% of the single time step feature. The SVM performed similarly well with these features. We see no dramatic difference in training per object specific versus object agnostic classifiers.

The features we adapted from Chu et al. perform worse than the simple feature functions we defined. This is not surprising since they were designed with other tasks in mind. In fact the features from Chu et al. pool information over temporal windows which are quite large in comparison to the time in which slip events occur. These results show that there is a need for preserving the distinct information available at the immediate time of slip.

We see that the ball is the most difficult object across all classifiers for predicting slip. We attribute this to the fact that the spherical shape causes contact to be over a very small area. This causes the object to quickly lose contact with the finger once slip occurs giving few slip examples for training as well as few available testing examples for a classifier to correctly label. Slip is more easily detected by all classifiers on the roll of tape, the measuring stick, and the box. We attribute this to the flat contact surfaces of the objects which generate a long and consistent sensory signal. This consistency simplifies the learning problem, while also giving more chances for testing examples to be correctly classified.

In Table I we additionally compare our learning approach to the spectral slip classifier introduced in [14]. This spectral slip method computes the total energy in the $P_{\rm ac}$ channel over 8 frames, after bandpass filtering the output from 30 to 200 Hz. We choose the threshold for total energy present by optimizing the F_{score} on the training set. Unsurprisingly this method performs quite poorer than the learning methods. The method has a very strong assumption about how slip events manifests in the sensor ignoring all information but that found in the AC pressure component.

For the last set of experiments, we were interested in examining how well learned classifiers could generalize to previously unseen objects. As such, we removed one object class at a time from the training set and trained a classifier on data from the six remaining classes. We repeated this procedure for all seven objects and we report results for different features and learning methods in Table II.

We note that the random forest again performs best in this generalization scenario with the delta feature function outperforming the single time step features on average. The overall F_{score} for the best performing method is only slightly worse than for results when all objects have been seen. In fact the F_{score} reduces by only 10%. Spectral slip again performs poorly, with a very minor decrease in score from being trained on all objects. The performance using the features of

TABLE I

 F_{score} for various combinations of classifier and features. "Per object" denotes classifiers trained independently for each object. "All objects" refers to training a single, general classifier across all objects. The best performing method in each column is highlighted in bold.

Features	Classifier	Training	F _{score}								
Teatures	Classifier	ITanning	Mean	Ball	Box	Cup	Marker	Measuring Stick	Tape	Watering Can	
	Linear SVM	per object	0.7451	0.6437	0.9765	0.7825	0.3258	0.8041	0.9697	0.7137	
		all objects	0.7341	0.5065	0.911	0.7114	0.5849	0.9163	0.9749	0.5339	
A t	Pandom Forest	per object	0.7224	0.2266	0.9775	0.7741	0.6611	0.9647	0.9956	0.4571	
	Kandoni Porest	all objects	0.7502	0.2886	0.9777	0.7437	0.7049	0.9126	0.999	0.6247	
	Linear SVM	per object	0.7174	0.6121	0.9795	0.7827	0.1495	0.7824	0.9879	0.7278	
$[\mathbf{x}_t, \Delta \mathbf{x}_t]$		all objects	0.7336	0.5289	0.908	0.7198	0.5532	0.9328	0.9774	0.5148	
	Random Forest	per object	0.7123	0.2462	0.978	0.6922	0.6591	0.9287	0.9963	0.4854	
		all objects	0.7097	0.2163	0.9754	0.705	0.6621	0.9192	0.9927	0.4969	
	Linear SVM	per object	0.7174	0.4132	0.9671	0.7849	0.4524	0.7719	0.9069	0.7255	
$\mathbf{x}_{t-\tau:t}$		all objects	0.6571	0.461	0.8141	0.7016	0.3813	0.97	0.8181	0.4535	
		per object	0.7212	0.2428	0.9759	0.7297	0.7035	0.9446	0.9941	0.4579	
	Random Forest	all objects	0.7151	0.2402	0.9701	0.7067	0.686	0.9227	0.996	0.4841	
Chu et al.	Pandom Forest	per object	0.6956	0.6053	0.9754	0.6862	0.6364	0.8666	0.6778	0.4219	
	Kanuonii Forest	all objects	0.5374	0.4742	0.9417	0.7026	0.2966	0.7803	0.5181	0.0486	
D	Spectral Slip	per object	0.2751	0.1237	0.3586	0.2122	0.0796	0.3908	0.4039	0.357	
rac	spectral Shp	all objects	0.2565	0.0917	0.3207	0.2071	0.0791	0.3883	0.3714	0.3368	

TABLE II

Fscore FOR VARIOUS CLASSIFIERS IN GENERALIZING TO PREVIOUSLY UNSEEN OBJECTS. THE BEST PERFORMING METHOD IN EACH COLUMN IS

HIGHLIGHTED	IN BOLD.	
-------------	----------	--

Features	Classifier	F _{score}									
1 catures	Classifier	Mean	Ball	Box	Cup	Marker	Measuring Stick	Tape	Watering Can		
x _t	Linear SVM	0.5141	0.5432	0.7866	0.4467	0.6384	0.6629	0.0193	0.5019		
	Random Forest	0.5936	0.0816	0.8596	0.6966	0.5414	0.6442	0.756	0.5757		
$[\mathbf{x}_t, \Delta \mathbf{x}_t]$	Linear SVM	0.4788	0.5639	0.7718	0.3756	0.5813	0.6453	0.0183	0.3953		
	Random Forest	0.6739	0.1626	0.8922	0.7076	0.7052	0.7762	0.9021	0.5716		
v	Linear SVM	0.4406	0.3997	0.0121	0.6089	0.4554	0.2432	0.843	0.5216		
$\mathbf{A}_{t-\tau:t}$	Random Forest	0.6149	0.0686	0.8235	0.697	0.6975	0.7906	0.709	0.518		
Chu et al.	Random Forest	0.2926	0.1925	0.8609	0.625	0.0898	0.229	0.0496	0.0017		
Pac	Spectral Slip	0.2485	0.0917	0.2957	0.2059	0.0791	0.3758	0.3714	0.3202		

Chu et al. perform quite a bit worse than in the earlier experiments. As noted before, these features were designed with a very different purpose in mind. As such, the results show that for the case of tactile slip detection, there is a need for designing tactile features that explore the information gained at each time step. Furthermore, the strong performance of the delta features suggests that dynamic information over short periods is important while the performance of the time window features suggest that excessive dynamic information might lead to overfitting.

We observe poor performance for all combinations of classifiers and features when generalizing for the ball. The poor generalization of the ball, results from it's distinct shape with respect to the other training objects.

D. Slip Prediction

We now turn our analysis to our learned slip predictors. We trained classifiers to predict slip with look-ahead horizons τ_f of 10, 15, and 20 steps, equating to times of 0.01, 0.015, and 0.02 seconds respectively. Performing a similar analysis as for the slip detectors we show the prediction rates for the trained slip predictors in Table III. In this instance we only report the *F*_{score} for the random forests with single element and delta features, since these proved to be the most relevant in the detection experiments. With respect to the features and the training scenarios, the results are similar to the ones

obtained for the detectors. We observe that the highest scores for each object are again condensed in the single element features and that the average F_{score} obtained by the single feature classifiers is slightly higher than the that of the delta features. With respect to the look-ahead horizons, we observe the highest average performance when $\tau_f = 10$. Considering each object separately, we see that most objects also perform best when $\tau_f = 10$. However, predicting slip for the marker works best when $\tau_f = 15$, while setting $\tau_f = 20$ gives the best performance on the ball and cup trials. This suggest that for objects with higher curvature the changes in finger deformation make it easier to predict slip events at an earlier stage.

We also performed the generalization experiment for the predictors. The results are shown in Table IV. The results are again very similar to the ones obtained for the detectors, as the delta features still show the best generalization capabilities. With respect to the values of τ_f , the best scores for each object can be seen for $\tau_f = 10$ and $\tau_f = 15$ while the average best performance is still observed for $\tau_f = 10$.

E. Grip Stabilizing Control

As a final test of the generalization capabilities of our slip detection method, we examine the ability to perform grip stabilization on novel objects. We embedded the random forest slip predictor in the grip stabilization controller presented

TABLE III

 F_{score} for various combinations of classifier and features when performing prediction. "Per object" denotes classifiers trained independently for each object. "All objects" refers to training a single, general classifier across all objects. Prediction is done for 3 different values of τ_f .

Features	τ_{c}	Training					F_{sco}	ore		
1 catales	•f		Mean	Ball	Box	Cup	Marker	Measuring Stick	Tape	Watering Can
	10	per object	0.717	0.4255	0.9715	0.7324	0.5695	0.9538	0.9874	0.3792
	10	all objects	0.6806	0.2857	0.9233	0.7104	0.4165	0.896	0.9926	0.5395
v	15	per object	0.7047	0.1966	0.9675	0.7334	0.6563	0.9414	0.9862	0.4513
A t	15	all objects	0.6427	0.2369	0.9532	0.7188	0.3081	0.8949	0.9255	0.4617
	20	per object	0.6795	0.0938	0.9661	0.7274	0.6767	0.9241	0.9471	0.4211
		all objects	0.6989	0.4626	0.9359	0.6948	0.4034	0.8991	0.9732	0.5232
	10	per object	0.6797	0.0937	0.9685	0.7361	0.602	0.9163	0.9885	0.4528
		all objects	0.6743	0.0865	0.9665	0.7149	0.6041	0.914	0.9912	0.4429
	15	per object	0.6918	0.1294	0.9615	0.7385	0.7228	0.866	0.9823	0.4422
$[\mathbf{X}_t, \Delta \mathbf{X}_t]$	15	all objects	0.6727	0.32	0.9561	0.7119	0.4001	0.9101	0.9776	0.433
	20	per object	0.6918	0.0972	0.964	0.7439	0.7145	0.9235	0.9721	0.4273
	20	all objects	0.6697	0.3275	0.9592	0.7161	0.4497	0.8983	0.9741	0.3626

TABLE IV

 F_{score} for various classifiers in generalizing to previously unseen objects when performing prediction. The best performing method in each column is highlighted in bold.

Features	τ.	F _{score}										
	^l	Mean	Ball	Box	Cup	Marker	Measuring Stick	Tape	Watering Can			
	10	0.5266	0.1741	0.4054	0.7017	0.4032	0.7101	0.6281	0.6638			
\mathbf{x}_t	15	0.564	0.2315	0.5874	0.6507	0.2373	0.7819	0.7739	0.685			
	20	0.5962	0.1643	0.7936	0.669	0.4432	0.7558	0.759	0.5885			
	10	0.6406	0.1943	0.8154	0.6965	0.5778	0.7678	0.9502	0.4819			
$[\mathbf{x}_t, \Delta \mathbf{x}_t]$	15	0.5562	0.2326	0.4383	0.6909	0.4278	0.8036	0.8765	0.4239			
	20	0.5337	0.1667	0.5861	0.6906	0.3724	0.7763	0.879	0.2649			

in Section IV. We used the random forest trained with one object held out to perform a number of grip stabilization trials. Each trial consisted of the robot initially pinning the object in a similar manner to the training method. However, the robot initially applies a much lighter contact force ($\approx 2N$) when initially pinning. We introduce additional variability into the testing by moving away the robot with a randomly selected exit velocity. The exit velocity is sampled uniformly between 0.02m/s and 0.07m/s in the reverse direction. We change the direction of motion by adding vertical and lateral velocity components sampled from Gaussian distributions with 0.0 cm/s mean and 0.05 cm/s standard deviation. We compare the performance of using the single time step features to using the delta features for a number of different look-ahead values. The results achieved using a variety of different classifiers are shown in Tables V. We conducted ten trials per object and report the percentage of successful grip stabilization trials for each object.

When using detection, and not prediction, we see that the controller performs better on average when trained with the delta features. This echoes the offline results for leaveone-out detection, where the delta features also performed best. This is intuitively appealing as information about the change in state available in the delta features should aid in compensating for slip compared to the raw features. As shown in Table V the controller is able to stabilize most objects successfully when using the learned predictor.

When using prediction we see that the overall performance of the controllers trained with single element features increases above the detection rates, for all look-ahead values. On the other hand the performance of the controllers trained with the delta features seems to be highly correlated with the look-ahead value, achieving the highest overall stabilization performance out of all controllers when $\tau_f = 20$. However, no single method performs best consistently. Looking at each object specifically, we observe that the four worst performing objects are the measuring stick, marker, ball and watering can. The measuring stick is quite heavy, shortening the duration of the slip events, making it quite hard to stabilize. Stabilizing the marker and the ball requires a good balance between the normal force and the contact location, as increasing the normal force may result in the object being ejected from the grip. Finally the watering can has an uneven weight distribution creating torsional slip which our controller does not mitigate well.

VI. CONCLUSIONS AND FUTURE WORK

We have presented a learning based approach to feedback control for stabilizing objects. Our controller relies on learning to predict slip through tactile sensing. Our approach learns from real-world robot interactions to better detect slip events better than previous proposed approaches. Our binary classification formulation of slip events also corresponds well with neuroscientific evidence suggesting that the human tactile system has a strong discrete feedback component [5]. We show that our learning method effectively generalizes learned knowledge to predict slip when interacting with novel objects. We believe these results show great promise for the use of slip detection to improve control during in-hand manipulation. We intend to extend our controllers into other

TABLE V

PERCENTAGE OF SUCCESSFUL GRIP STABILIZATION TRIALS USING OUR GRIP STABILIZATION CONTROLLER. ALL CONTROLLERS USED A RANDOM FOREST SLIP CLASSIFIER TRAINED WITHOUT DATA FOR THE TEST OBJECT. BOLD VALUES INDICATE THE BEST PERFORMANCE FOR A GIVEN OBJECT.

Test Object		Х	t		$[\mathbf{x}_t, \Delta \mathbf{x}_t]$				
Test Object	$ au_f = 0$	$\tau_{f} = 10$	$\tau_{f} = 15$	$\tau_f = 20$	$ au_f = 0$	$\tau_f = 10$	$\tau_{f} = 15$	$\tau_f = 20$	
Ball	0%	0%	0%	100%	90%	0%	90%	80%	
Box	100%	100%	100%	100%	100%	100%	90%	100%	
Cup	90%	60%	40%	70%	100%	10%	60%	60%	
Marker	80%	40%	80%	10%	30%	10%	0%	100%	
Measuring Stick	20%	90%	60%	10%	20%	10%	0%	10%	
Tape	10%	100%	80%	100%	30%	80%	100%	90%	
Watering Can	10%	60%	100%	50%	30%	60%	60%	80%	
Overall	44.28%	64.28%	65.71%	62.85%	57.14%	38.57%	57.14%	74.28%	

manipulation tasks such as performing grip stabilization during object transport or controlled sliding of grasped objects.

While we focus particularly on the detection of slip, our formulation can be extended to detecting other types of tactile events. Slip was chosen as a first step, since it is a pervasive problem in both grasping and in-hand manipulation. We plan to extend our detection framework to other tactile events such as making and breaking contact with an object or detecting when an object being lifted breaks contact with the supporting surface. At present our method is limited by the need of a human to provide labels based on visual information. We hope to circumvent this problem in the future by measuring slip externally during training. Nevertheless we still find our method to be more attractive than heuristic or model-based methods of detecting slip.

REFERENCES

- J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis: A survey," *Robotics, IEEE Transactions on*, vol. 30, no. 2, pp. 289–309, 2014.
- [2] D. L. Brock, "Enhancing the dexterity of a robot hand using controlled slip," in *IEEE International Conference on Robotics and Automation* (*ICRA*), 1988, pp. 249–251.
- [3] A. Bicchi and V. Kumar, "Robotic grasping and contact a review," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2000, pp. 348–353.
- [4] H. Yousef, M. Boukallel, and K. Althoefer, "Tactile Sensing for Dexterous In-hand Manipulation in Robotics–A Review," *Sensors and Actuators A: Physical*, vol. 167, no. 2, pp. 171–187, 2011.
- [5] R. S. Johansson and J. R. Flanagan, "Coding and use of tactile signals from the fingertips in object manipulation tasks." *Nature reviews. Neuroscience*, vol. 10, no. 5, pp. 345–59, May 2009.
- [6] M. Tremblay and M. Cutkosky, "Estimating friction using incipient slip sensing during a manipulation task," in *IEEE International Conference on Robotics and Automation (ICRA)*, 1993, pp. 429–434 vol.1.
- [7] A. Ikeda, Y. Kurita, J. Ueda, Y. Matsumoto, and T. Ogasawara, "Grip force control for an elastic finger using vision-based incipient slip feedback," in *IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS)*, vol. 1, 2004, pp. 810–815.
- [8] F. Viña, Y. Bekiroglu, C. Smith, Y. Karayiannidis, and D. Kragic, "Predicting Slippage and Learning Manipulation Affordances through Gaussian Process Regression," in *IEEE/RAS International Conference* on Humanoid Robots (Humanoids), 2013.
- [9] A. Maldonado, H. Alvarez, and M. Beetz, "Improving robot manipulation through fingertip perception," in *IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS)*, Oct. 2012, pp. 2947–2954.
- [10] J. M. Romano, K. Hsiao, G. Niemeyer, S. Chitta, and K. Kuchenbecker, "Human-Inspired Robotic Grasp Control With Tactile Sensing," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1067–1079, 2011.
- [11] V. A. Ho, T. Nagatani, A. Noda, and S. Hirai, "What can be inferred from a tactile arrayed sensor in autonomous in-hand manipulation?" 2012 IEEE International Conference on Automation Science and Engineering (CASE), pp. 461–468, Aug. 2012.

- [12] B. Heyneman and M. Cutkosky, "Slip interface classification through tactile signal coherence," in *IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS)*, 2013.
- [13] J. Reinecke, A. Dietrich, and F. Schmidt, "Experimental Comparison of Slip Detection Strategies by Tactile Sensing with the BioTac on the DLR Hand Arm System," *IEEE International Conference on Robotics* and Automation (ICRA), pp. 2742–2748, 2014.
- [14] J. A. Fishel, "Design and Use of a Biomimetic Tactile Microvibration Sensor with Human-Like Sensitivity and its Application in Texture Discrimination using Bayesian Exploration," Ph.D. dissertation, University of Southern California, 2012.
- [15] M. Shöpfer, C. Schürmann, M. Pardowitz, and H. Ritter, "Using a piezo-resistive tactile sensor for detection of incipient slippage," in *ISR / ROBOTIK*, 2010.
- [16] M. Li, Y. Bekiroglu, D. Kragic, and A. Billard, "Learning of grasp adaptation through experience and tactile sensing," in *IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS)*, 2014.
- [17] Y. Bekiroglu, J. Laaksonen, J. A. Jørgensen, V. Kyrki, and D. Kragic, "Assessing grasp stability based on learning and haptic data," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 616–629, 2011.
- [18] H. Dang and P. K. Allen, "Stable grasping under pose uncertainty using tactile feedback," *Autonomous Robots*, pp. 1–22, 2013.
- [19] M. Madry, L. Bo, D. Kragic, and D. Fox, "ST-HMP: Unsupervised spatio-temporal feature learning for tactile data," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- [20] N. Wettels, J. Fishel, and G. Loeb, "Multimodal Tactile Sensor," *The Human Hand as an Inspiration for Robot Hand Development*, no. 0912260, pp. 1–20, 2014.
- [21] V. Chu, I. McMahon, L. Riano, C. G. McDonald, J. M. Perez-Tejada, M. Arrigo, N. Fitter, J. C. Nappo, T. Darrell, and K. J. Kuchenbecker, "Using robotic exploratory procedures to learn the meaning of haptic adjectives," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 3048–3055.
- [22] E. Osuna, R. Freund, and F. Girosi, "Training Support Vector Machines: An Application to Face Detection," in *IEEE Conference On Computer Vision and Pattern Recognition (CVPR)*. Washington, DC, USA: IEEE Computer Society, 1997, pp. 130–.
- [23] F. Li, J. Carreira, and C. Sminchisescu, "Object Recognition as Ranking Holistic Figure-Ground Hypotheses," in *IEEE Conference On Computer Vision and Pattern Recognition (CVPR)*, 2010.
- [24] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time Human Pose Recognition in Parts from Single Depth Images," in *IEEE Conference On Computer Vision and Pattern Recognition (CVPR)*, Washington, DC, USA, 2011, pp. 1297–1304.
- [25] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *IEEE International Conference on Computer Vision (ICCV)*, 2013.
- [26] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [27] A. Criminisi, J. Shotton, and E. Konukoglu, "Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning," *Microsoft Research Cambridge, Tech. Rep. MSRTR-2011-114*, vol. 5, no. 6, p. 12, 2011.
- [28] F. Pedregosa and G. Varoquaux, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825– 2830, 2011.