Proactive slip control by learned slip model and trajectory adaptation

Anonymous Author(s)
Affiliation
Address
email

Abstract:
This paper presents a novel control approach to dealing with a slip during robotic manipulative movements. Slip is a major cause of failure in many robotic grasping and manipulation tasks. Existing works use increased gripping forces to avoid/control slip. However, this may not be feasible, e.g., because (i) the robot cannot increase the gripping force– the max gripping force has already applied or (ii) an increased force yields a damaged grasped object, such as soft fruit. Moreover, the robot fixes the gripping force when it forms a stable grasp on the surface of an object, and changing the gripping force during manipulative movements in real-time may not be feasible, e.g., with the Franka robot. Hence, controlling the slip by changing gripping forces is not an effective control policy in many settings. We propose a novel control approach to slip avoidance including a learned action-conditioned slip predictor and a constrained optimizer avoiding an expected slip given the desired robot actions. We show the effectiveness of this receding horizon controller in a series of test cases in real robot experimentation. Our experimental results show our proposed data-driven predictive controller can control slip for objects unseen in training.

Keywords: Slip avoidance, Robot Control, manipulation

1 Introduction

Sense of touch in humans and primates is key to building sophisticated dexterous manipulation skills [1]. It provides localized contact information on human skin in particular as an effective perception in our daily life [2]. A slip event occurs when the ratio of applied shear to the normal force on the contact area exceeds the static friction coefficient. Recent studies show that human reflex to the feeling of slip (we use the term slip to refer to slipping action of object between fingers) during manipulative movements not only by grip adjustment but also specific movements of (i.e. reflex in) their shoulder and elbow to control slip [3]. Existing slip avoidance control methods in robotic manipulation using artificial tactile sensing are limited to: (i) only grip force increase (adaptation) in case of slip [4, 5], (ii) reactive algorithms that activate after slip onset and detection [6, 7], and (iii) gentle manipulation tasks, *e.g.*, slow lifting movements for 10-20 [cm] [5, 6]. Increased gripping force may not be an effective action when (i) the gripper has already reached its force limits, (ii) the grasped object is delicate/fragile such as soft fruits (*i.e.* strawberry where the average grip force to damage the fruit is 1.5 N [8]). Moreover, many grippers, such as Robotiq's Hand-E Gripper or Franka Emika Gripper, do not have enough force resolution, or (iii) the gripper cannot be controlled in real-time due to connection limitations (*i.e.* blocking TCP/IP connection in Franka Emika gripper.

We proposed a novel means of controlling slip by adapting trajectory of manipulative movements: (i) Reactive Slip Control (**RSC**) uses detected slip– slip classifier at time t– to reduce the robot movements velocity to avoid slip; (ii) Proactive Slip Control (**PSC**) policy for avoiding slip in robotic manipulative movements (see schematic of the approach is shown in Fig. 1). PSC includes an

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

action-conditioned slip predictor– slip classifier at $t + h$ where h is prediction horizon – (based on LSTM (Long-Short Term Memory) NN (Neural Network)) and a Predictive Controller. The controller minimizes the likelihood of predicted slip in a prediction horizon by adjusting the robot planned movements and deviating from the reference trajectories. We use a quadratic term penalizing deviating from a reference trajectory; hence, PSC avoid slip while following closely a given reference trajectory. Our constrained optimization enables the robot meet the its limits in generating optimal future robot actions. We study the effectiveness of our on-the-fly adaptation of a real robot's pre-planned trajectory (which may be minimum time or minimum jerk) in a series of real robot experiments. We compare *RSC* and *PSC* in terms of their performance in slip avoidance, computation complexity, and optimality. Our experimental results indicates PSC outperforms RSC and can change the game in robotic manipulation success where increased gripping force is not possible.

2 Related works

Slip detection algorithms are thoroughly reviewed in [9, 10]. Veiga et al. [11] use Support Vector Machine (SVM) and Random Forest (RF) on SynTouch BioTac's raw tactile readings and James et al. [12] use SVM on TacTip signals for slip classification. SVM and RF have computational complexity problems in training for large data sets with high feature dimensionality. Specifying a threshold on the rate of shear forces for slip detection is a common approach for slip classification [13, 4]. However, it needs initial friction estimation and finding the threshold for large object sets can be expensive. Other techniques rely on analyzing micro-vibration in the incipient slip phase such as spectral analysis [14, 15]. These approaches can fail when robot vibrations interfere with sensor-object vibrations. Multi-modal approaches are also proposed to use proximity [16] or visual sensing data [17] to improve slip detection. Recurrent Neural Networks and Graph Neural Networks are also used for slip classification [18, 19]. They only use tactile readings for slip detection. We use an LSTM with combined tactile readings and robot actions to predict slip in a prediction horizon.

Grip force control based on tactile readings is usually designed to avoid slip [11]. Dong et al. [20], Kaboli et al. [4], Khamis et al. [7] increase the normal force until the ratio of normal to shear force becomes smaller than the known/estimated static friction coefficient. This method can lead to deformation or damage of delicate objects, e.g., soft fruits. Kaboli et al. [4] propose a heuristic to limit deformations of a deformable object but it needs to measure the initial grasp width and measure deformation. Kobayashi et al. [21] suggest increased number of gripping fingers to avoid slip with multi-finger hand. In bi-manual manipulation, [6] performs hand pose correction to bring the resultant force inside the friction cone as a slip avoidance controller. The above approaches have certain limitations where increased force is not an option (the articulated kinematic control only allows gripping force adjustment before manipulative movements, e.g., in Franka Emika, or in delicate object manipulation).

Online trajectory re-planning is effectively used in areas such as agile quadrotor control [22, 23], but to the authors' knowledge, it is the first time it is being used for slip avoidance in dynamic manipulation tasks. The main difficulty relates to the lack of analytical models for tactile/slip dynamics. By the proposed constrained optimization setting, we seek to benefit from the data-driven models for slip prediction and model-based approaches for online trajectory optimization.

3 Data set

Tactile sensor and manipulation task: Spatial resolution, reading frequency, and mechanical compliance are the main features of a tactile sensor reviewed in [24]. We use uSkin, a magnetic-based tactile sensor from XELA ROBOTICS which is one of the challenging sensors for our purpose due to its calibration complexities. It works with 100 Hz frequency, it publishes 3-axis point-wise force measurements for 16 taxels arrayed in a 4x4 matrix (Fig. 1a). A pair of sensors are mounted on 3-D printed fingers for a jaw gripper. The directions of shear and normal readings are shown beside the sensor in Fig. 1a.

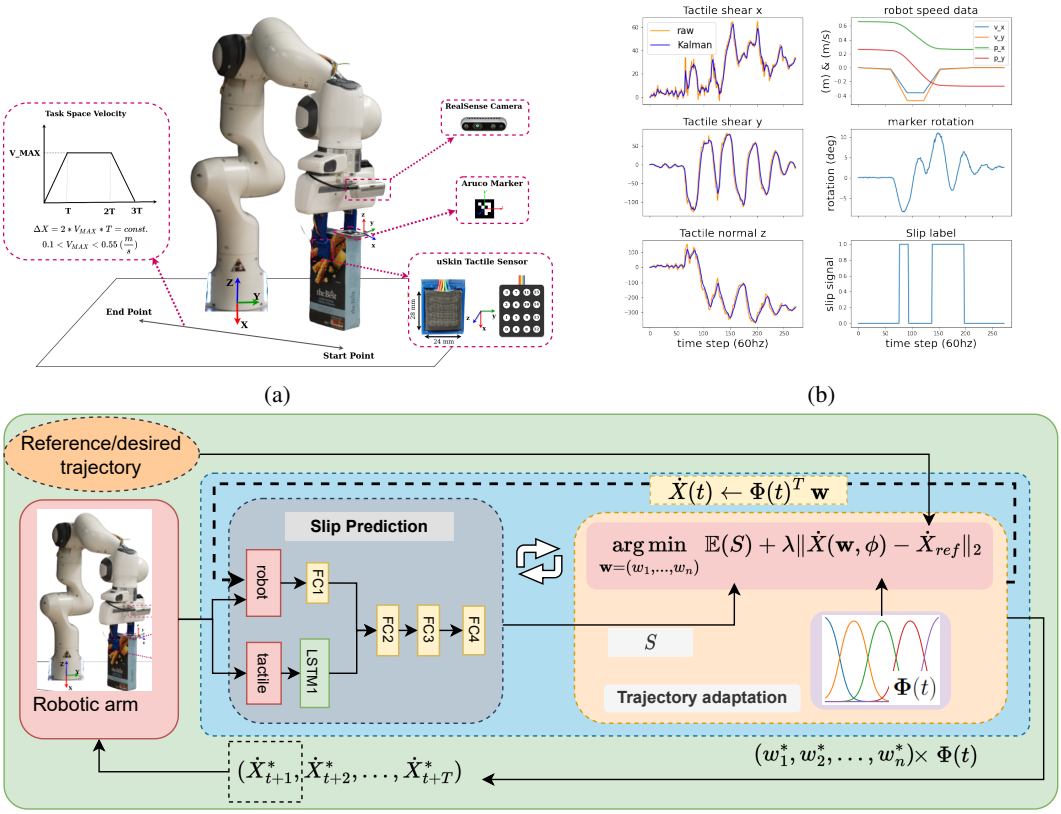

Figure 1: (a) Experimental setup including Franka Robot, 4x4 uSkin tactile sensor on its fingers, box-shaped object, ArUco marker on the object, and RealSense camera. A sample of trapezoidal task space velocity trajectory is shown on the left. (b) Data set features: tactile readings, robot trajectories, marker trajectory, and slip signal. (c) A schematic of the proposed *proactive* slip control (blue block) that is running at 30 Hz and includes a slip prediction module (grey block), and predictive controller for trajectory adaptation (orange block).

A box-shaped object with a flat contact geometry is chosen as the training set object. To encourage more slip-cases, we reduce the friction coefficient by covering the uSkin sensor with a *latex* cover and increase object weight by adding metal pieces inside the box ($4 \times 15 \times 25$ [cm] and 250 [gr]). A wrist camera reads the pose of the ArUco marker attached on the top side of the object measuring the relative motion of the object w.r.t the hand. This is used to create labels for ground truth slip values. Maximum noise amplitude for marker readings is 0.38 degrees for rotation and 0.5 [mm] for the position in this range of distance.

We use a task space velocity controller during linear robot movements in the XY plane. We tested several reference trajectories with trapezoidal velocity profiles and various maximum velocity (V_{MAX}) values. The velocity commands are converted to joint torques by the robot's low-level joint impedance controller. V_{MAX} in x direction increases from 0.1 [m/s] to 0.5 [m/s] with a step of 1 [cm/s] for data collection. Fig. 1a shows a sample of the reference velocity profile; As V_{MAX} changes, the acceleration and deceleration values change accordingly such that the area under the curve which is equal to the translation remains constant.

Data collection and pre-processing: We collected a data set of 660 linear motion tasks containing (i) tactile readings from two fingers (at 100 [Hz]), (ii) robot state data (at 1,000 [Hz]) including joints

position, joints velocity, task space pose, and task space velocity, (iii) Marker pose data (at 60 [Hz]), and (iv) Slip label values based on marker's rotation (see Fig. 1b). We use ROS *ApproximateTime* policy to synchronize the data.

We applied Kalman Filter on the raw tactile readings to delete the measurement noise on the data (See Fig. 1b). To be able to use the same preprocessing method in both offline and online test cases, Kalman Filter is chosen over Kalman Smoothing. A threshold on the rotation of the ArUco marker is used to create the slip ground truth labels. Based on marker rotations for slower ranges of motions with no slip, rotations $> 6°$ are labeled as slip-case. Considering the linear robot motion, slip can happen either as pure rotational slip, or rotational slip followed by translational slip (which usually leads to object dropping and task failure). As such, object rotation is used to create slip ground truth values. The data set along with preprocessing and models will be *publicly available* for future investigations of action-conditioned tactile or slip prediction or slip detection models in a dynamic manipulation task.

4 Methodology

4.1 Slip detection and slip prediction models

Slip classification is explored by three main approaches in the literature: (i) incipient slip detection [10], (ii) full slip detection [13, 4], and (iii) slip prediction [5]. Incipient slip detection requires precise slip calibration and sensor mechanical compliance [7, 25]. Without having the required mechanical compliance among the uSkin taxels, we use data-driven approaches for full slip detection and slip prediction. Tactile shear forces have proven to be a useful feature for slip classification [10]. This grounds the higher rate of changes for shear force when the taxels are losing contact with the objects, which is classified as an anomaly versus a stuck phase.

Zapata-Impata et al. [26], Veiga et al. [27] show the benefits of using a history of tactile features and temporal processing by modules such as LSTM cells. As such we use the shear forces of the uSkin sensor with two LSTM-based models for slip *detection* and *prediction* problems. In *slip detection*, a time window of length C of shear tactile forces $\mathbf{x}_{t-C:t} \in \mathbb{R}^{C\times 48}$ are used as input features to classify the binary slip class $c_t = f(\mathbf{x}_{t-C:t})$ where $c_t \in \{c_{slip}, c_{non-slip}\}$ at time step t. For *slip prediction*, we use a multi-modal LSTM-based model (inspired by Veiga et al. [11, 27]). This uses time windows of tactile features $\mathbf{x}_{t-C:t} \in \mathbb{R}^{C\times 48}$ and robot future actions $\mathbf{a}_{t+1:t+T} \in \mathbb{R}^{T\times 6}$ to classify the slip state c_{t+T} at time step $t+T$ (where T is the prediction horizon). The architecture of the *slip prediction* model is shown in Fig. 1c. The *slip detection* model has a similar architecture without the robot data input and the corresponding dense layer. Fusing the latent vectors of two input modalities seeks to exploit the future robot motion for slip prediction.

4.2 Close-loop slip control by trajectory adaptation

The studies in closed-loop control design for slip avoidance are limited to gentle short-range lifting motions [4, 20, 28]. Our proposed approach extends slip control beyond simple slow lifting motions. First, we collected a data set of dynamic moving tasks by pre-defined task space trajectory, i.e. a linear motion with various velocity profiles and ranges of acceleration/deceleration values. We learned slips are initiated by the dynamic motions of the robot as opposed to being caused by object weight or a non-stable grasp for slow lifting tasks in literature [4, 20, 28]. Therefore, we propose two trajectory optimization/adaptation methods incorporating tactile and robot proprioceptive data for slip control. We tested/compared PSC and RSC approaches in real-time scenarios on real robot experimentation. PSC has the benefit over RSC of activating the controller before the slip's actual onset.

Reactive slip control by trajectory adaptation Inspired by episodic policy search in [29], we represent the robot actions (*e.g.* task space velocity) by linear Gaussian parametric models. As such, robot's action at time t can be represented as $\dot{X}_t := \sum_{j=1}^{N} w_j \phi_j(t)$, where $w_j \in \mathbb{R}$ are the weight

parameters and ϕ_js are the basis functions with Gaussian forms $\phi_j(t) = \exp(-\frac{(t-\mu_j)^2}{2\sigma})$ where $j = 1, 2, .., N$ is the number of basis functions.

A sequence of future robot actions with length T in a matrix form will become $\dot{X}_{t+1:t+T} := \Phi^T \times W$ where $\Phi(t) = [\phi_1(t), \phi_2(t), ..., \phi_n(t)] \in \mathbb{R}^{N\times T}$ and $W = \{w_1, w_2, ..., w_n\} \in \mathbb{R}^N$ are the matrices of basis functions and weight values respectively. The mean values of the Gaussians are spread with equal distance in the horizon length $1 : T$ and the σ parameters are chosen such that Gaussians have reasonable widths and overlap.

The main policy in grip force adaptation for slip control is to incrementally increase the normal force ($N(t)$) until the output of the slip detection model (S) is zero ([27]). This is usually formalized as $N(t+1) = N(t) + S \times \delta N$, where δN is the force increasing step. Likewise, we implement RSC by adapting the robot movements trajectory by decreasing task space velocity by fine steps (we learned slip happens during the initial parts of movements with high acceleration) until the output of the slip detection model is zero. If no slip is detected, the robot will follow the desired reference trajectory. We empirically come up with a formulation in Eq. 1 after testing different ones. This includes an affine combination of slip cost and reference trajectory tracking cost. Although in RSC we do not need optimizing over a receding horizon as per Eq. 1, this constrained optimization will be later extended in PSC (in Eq. 2) allowing easy comparison between *PSC* and *RSC*.

$$\begin{aligned} \arg\min_{\mathbf{w}} \quad & \|\dot{X}(w,\phi) - \dot{X}_{ref}\|_2 + a^{\frac{1}{S+\epsilon}} \|\dot{X}(w,\phi)\|_2 \\ subject\ to: \quad & lb < \dot{X}_{t+1} - \dot{X}^r_t < ub \end{aligned} \tag{1}$$

Where $\dot{X}_{ref}$ is the reference trajectory, $S \in \{0, 1\}$ is the output of the slip detection model introduced in 4.1, a and $\epsilon \in \mathbb{R}^+$ are hyper-parameters, and $\dot{X}^r_t$ is robot's observed velocity at time t. The first term in the objective function aims to minimize the Euclidean distance between the generated and the reference trajectory. The second term in Eq. 1 results in generated smaller velocities (where the object is moved with a large acceleration). As such, by choosing a to be in $0 < a < 1$ and ϵ as a small positive value *i.e.* 0.01, for non-slip mode when $S = 0$, the effect of the second term vanishes (a^∞ , $a \in (0, 1)$) and if $S = 1$, it constantly reduces task space robot's velocity until there is no slip. The imposed constraint keeps the robot action at t bounded to avoid abrupt deceleration. These constraints ensure the admissibility condition of the robot's low-level controller is met. We selected the exponential term for the slip signal that results in stable optimization and slip avoidance behavior more than other terms, e.g., linear cost of slip. The hyper-parameters are empirically chosen as $a = 0.5$ and $\epsilon = 0.01$.

Proactive slip control by trajectory adaptation Predictive controllers, e.g. [30], showed success when dealing with nonlinear systems with the risk of being stuck in a local minimum of the corresponding utility function. They can achieve a solution close to the global optimum by finite-horizon optimization. We use the same approach by solving the constrained optimization in a prediction horizon and applying the first component of the generated action sequence at each time step.

Parametric action representation has benefits in exploration complexity versus direct search in action space [29]. The benefit is more effective when the action is optimized in a future time horizon rather than a single time step. In this regard, in the trajectory adaptation problem, we use linear Gaussian parametric representation [31] for the robot action sequence. This leads to fewer search parameters in the optimization problem relative to searching in the action space. In this scenario, by the *slip prediction* model introduced in section 4.1, we close the loop for PSC by solving the constraint optimization in Eq. 2. The predicted slip value at time t is a function of robot's future trajectory, $S(\dot{X}_{t+1:t+T}(\mathbf{w}, \phi))$.

$$\begin{aligned} & \arg\min_{\mathbf{w}} \|\dot{X}(\mathbf{w},\phi) - \dot{X}_{ref}\|_2 \\ subject\ to: \quad & S(\dot{X}_{t+1:t+T}(\mathbf{w},\phi)) = 0 \\ & lb < \dot{X}_{t+1} - \dot{X}^r_t < ub \end{aligned} \tag{2}$$

Our PSC aims at minimizing the expected slip over the prediction horizon while keeping close to the given reference trajectory as shown in Fig. 1. In contrast to RSC in Eq. 1, we consider the slip term

as a non-linear constraint in Eq.2. This forms a more effective optimization for slip minimization since the optimization library we use forms a Lagrangian function with optimizing the Lagrange multipliers as opposed to the heuristic form of the multiplier in the second term of Eq.1.

5 Experimental results and discussion

We tested the proposed trajectory adaptation methods to control slip in real-time in several test cases of manipulative trajectories with Franka Emika robot connected to a PC with AMD® Ryzen threadripper 2950x 16-core processor and 64 GB RAM running Ubuntu 20.04 and ROS Noetic. Torch library is used for the neural networks of the slip classification models. To accelerate the inference time of the torch models, we use ONNX library which reduces the inference time of the slip detection/prediction models from 1 [ms] to 0.2 [ms]. We use scipy optimization library in python 3 (i.e. *minimize()* function with *'trust-constr'*) to solve constrained optimization in Eq. 1 and 2. We also test our PSC with unseen objects in the training of the slip prediction indicating generalizability of our PSC.

Each test trial is executed 10 times on the robot to achieve statistically meaningful results. The metrics to compare the performance of the proposed slip controllers are (i) Maximum amount of object rotation (MOR) in the trial, (ii) Resulted optimality value (ROV) from the optimization library (*i.e.* gradient of the Lagrangian function, and indicative of the convergence of the optimization), and (iii) Execution time (ET) which shows the computation complexity. To see the combined result of the distance to the reference trajectory and slip constraint violation, we also report a customized optimality metrics calculated as $M_{opt} = \|\dot{X} - \dot{X}_{ref}\|_2 + \alpha\, t_{slips}$ where t_{slip} is the number of time steps that the slip constraint is violated (*i.e.* marker rotation is $> 6°$). The first term measures the Euclidean distance to the reference trajectory and the second term adds a penalty for each instance of slip. The smaller the customized metrics the better the performance of the controller. We report the mean and variance for each metric in Table 1. The variance can be related to (i) the variance in the initial contact state, (ii) changing the sensor cover across the experiments, and (iii) taxels' deformation and measurement noise. In total, 250 testing trials are executed with the train and test objects. Slip detection and prediction networks are trained offline before the tests on the robot. Slip classification accuracy and f-score on the test set (20% of the data) are 98% and 0.91 for the detection and 94% and 0.86 for the prediction models, respectively.

Reactive versus proactive slip control with trajectory adaptation: For the first testing case, we execute PSC and RSC with the same design parameters including the number of basis functions and the reference trajectory. Fig. 2a shows the object rotation between robot fingers during executing the reference trajectory in data collection mode without any controller. The start of the horizontal lines in Fig. 2a shows the time instance the object is dropped off the robot's hand due to slip. Figs. 2b and 2c show object's rotation for the *reactive* and *proactive* controllers, respectively. RSC activates after slip is observed in reality (*i.e.* rotation $> 6°$ around time step 70-80 in Fig. 2b), but it keeps the rotation bounded in the rest of the trial. PSC (shown in Fig. 2c) outperforms RSC in the first phase of object rotation as the slip control activates by predicting slip signals before slip actually initiates. Both controllers show 100% success in avoiding task failure while in the mode without slip controller (Fig. 2a) 19 trials out of 20 led to object dropping and failure.

The velocity profiles generated by RSC and PSC are shown in Figs. 2d and 2e. Although these figures show RSC keeps closer than PSC to the reference trajectory, it yields much more rotation of the object and none-smooth (with a higher jerk that is undesirable) movements. PSC yields trajectories farther from the reference trajectory but has smoother acceleration and deceleration trends which can lead to smaller slip instances. This is mostly because the weight of the slip avoidance relative to Euclidean distance term in Eq. 2 is set by the Lagrange multipliers of the optimizer in PSC. This makes the trajectories get more distant from the reference trajectory to satisfy the slip constraint.

The effect of the number of basis functions on trajectory adaptation: We tested *RSC* and *PSC* (*Reactive* and *proactive*, respectively, in Table 1) with basis functions ranging from 2 to 8 for the prediction horizon of length 10. The higher the number of the basis functions, the more com-

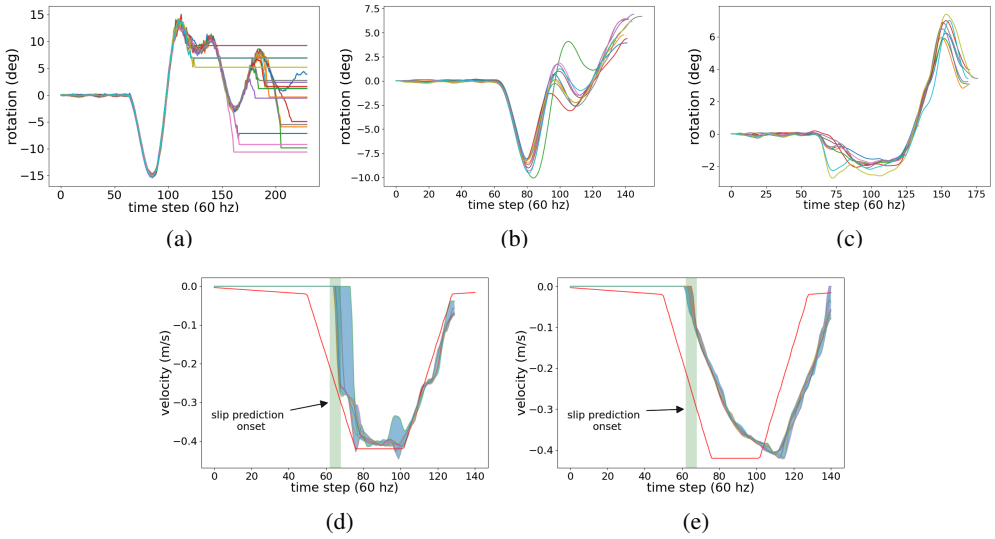

Figure 2: Object rotation around grasp axis during following a reference trajectory (a) without slip control and with (b) RSC, and (c) PSC where $> 6°$ are slip events. Trajectories generated by RSC (d) and PSC (e): the blue areas show variations across 10 trails; the green bands (at 60) show approximate time slip detected (predicted) for the first time.

Model	# Basis Functions	Optimality	Max rotation (deg)	Computation time (ms)	Customized optimality
Reactive	2	0.80 ± 0.72	12.80 ± 7.58	9.79 ± 1.33	2.41 ± 0.37
	3	0.71 ± 0.56	9.06 ± 0.56	10.03 ± 1.57	2.41 ± 0.18
	4	0.67 ± 0.45	9.65 ± 1.50	10.49 ± 1.62	2.35 ± 0.28
	5	0.63 ± 0.31	$\mathbf{8.82} \pm 0.61$	$\mathbf{9.96} \pm 1.49$	$\mathbf{2.27} \pm 0.30$
	6	0.47 ± 0.23	9.09 ± 1.19	10.15 ± 1.70	3.29 ± 0.48
	7	0.37 ± 0.22	9.66 ± 1.17	9.70 ± 2.66	3.22 ± 0.40
	8	$\mathbf{0.28} \pm 0.19$	9.80 ± 0.75	9.99 ± 1.50	3.75 ± 0.28
Proactive	2	0.75 ± 0.60	$\mathbf{3.99} \pm 0.58$	$\mathbf{11.73} \pm 5.35$	$\mathbf{1.67} \pm 0.05$
	3	0.72 ± 0.67	6.14 ± 1.52	22.57 ± 1.79	2.02 ± 0.37
	4	0.51 ± 0.41	7.16 ± 1.07	25.73 ± 2.14	2.38 ± 0.43
	5	0.40 ± 0.34	9.06 ± 0.55	29.61 ± 3.26	2.84 ± 0.09
	6	0.33 ± 0.22	9.75 ± 0.65	32.09 ± 5.05	3.17 ± 0.31
	7	0.30 ± 0.22	9.41 ± 0.55	35.27 ± 3.65	3.12 ± 0.26
	8	$\mathbf{0.25} \pm 0.20$	9.14 ± 0.37	37.55 ± 3.84	3.50 ± 0.41

Table 1: Trajectory optimization performance by different number of basis functions. For all of the metrics, smaller values show better performance.

plex/nonlinear trajectory they can generate. However, there are two pitfalls in increased numbers of basis functions (i) the resulted trajectory may become very noisy, and (ii) increased computational cost exponential proportional to the increased number of basis functions.

While optimality of slip control improves by an increased number of basis functions in both cases, maximum object rotation achieved by RSC does not show a clear improvement after 5 basis functions. In contrast, it leads to larger maximum rotations for PSC. The computation time increases proportionally in PSC, while it remains roughly constant in the RSC. We observed the Jacobians of the slip constraint should be numerically estimated by the optimization library which needs 30-50 calls of the slip detection/prediction networks in each optimization iteration. For choosing the best values for basis functions the customized optimality can be used. While five basis functions are best for RSC, PSC performs best with two basis functions. It can be observed that PSC shows better performance for all the metrics but the execution time.

Figure 3: Train and test objects.

Object	Optimality	Max rotation	Customized Optimality
theBest	0.51 ± 0.41	7.16 ± 1.07	2.38 ± 0.43
THINS	0.54 ± 0.43	7.88 ± 1.38	2.48 ± 0.52
Carrs	0.52 ± 0.39	8.02 ± 2.12	2.73 ± 0.23
RICE	0.61 ± 0.44	10.58 ± 3.56	3.02 ± 0.47
Monster	0.66 ± 0.46	36.20 ± 13.93	5.31 ± 1.82
Slug killer	0.52 ± 0.41	8.23 ± 1.86	2.78 ± 0.39

Table 2: Tests on novel objects.

Generalization tests with novel objects: Four basis functions is empirically chosen for PSC tests on unseen objects as it yields a balance between small optimality values and admissible computation time and slip control performance. To test the performance of the proposed approaches on unseen objects, we test PSC with four basis functions on 5 novel objects (see Fig. 3) each repeated 10 times. Three of the objects have a box shape similar to the training object, and two are cylindrical for a more difficult generalization cases. Table 2 shows the resulted metrics for the train (first row) and the novel objects. The first two objects have similar geometry and weight to the training object but with different friction behavior where the PSC performance is close to the ones with train object. Although the RICE box has similar geometry, the PSC performance is relatively poor. This is partially related to the lighter weight and different friction behavior of the box cover. We observed our PSC can successfully avoid slip in manipulating objects unseen in the training.

The Monster can with dynamic weight (liquid inside), deformable surface and different geometry (cylindrical) and friction is the most challenging test object for the controller and resulted in the worst performance metrics values. The PSC can effectively control the Slug Killer can slip which has a plastic and rigid material very different from the train object. The results in Table 2 demonstrates generalization capability of our PSC to unseen objects.

In the design of the RSC, since the slip signal is not a function of future robot trajectories, it was possible to form the cost over one time step of trajectory values; Nonetheless, we use optimization over receding horizons to make a valid comparison to PSC cases in terms of the effect of the hyper-parameters *e.g.* number of basis functions.

6 Conclusion

We presented a novel approach to slip control by online trajectory optimization. This is very important as the increased gripping force control policy is not an effective solution in many settings. For instance, grip force adaptation can fail in resolving slip avoidance problems. We presented two novel approaches for slip control by robot trajectory adaptation (i) Reactive Slip Control and (ii) Proactive Slip Control. We showed the success of the slip controllers in real-time tests on a dynamic real-world manipulation task. We show Proactive Slip Control outperforms the reactive one and it can generalize to objects unseen in training. Our future works include extending our dataset to include different tactile readings, e.g. camera-based tactile readings, and more real-world manipulation experiments, i.e. different objects and manipulative movements.

Limitations: Our approach is heavily relying on the slip data set to predict the slip. We have only one class of trajectories (i.e. a path with different velocity profiles) in our dataset which limits the prediction performance. We will extend the dataset in our future works and collect data with different classes of manipulative movement trajectories. The training dataset only includes one object with varying weights. Nonetheless, the slip controller could successfully generalize to unseen objects in our test cases. We will improve our dataset by adding more objects (from multiple classes) to our dataset in our future works. The controller is using one predicted slip signal to adjust its next action. Using all the time steps in a prediction horizon may improve the performance. In our future works, we will use a classifier for different predicted tactile readings provided by an action-conditioned tactile predictive model to improve the performance of slip control.

References

[1] R. Johansson and J. Flanagan. 6.05-tactile sensory control of object manipulation in humans. *Senses Compr. Ref. Acad. Press NY NY*, 67:86, 2008.

[2] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini. Tactile sensing—from humans to humanoids. *IEEE transactions on robotics*, 26(1):1–20, 2009.

[3] C. R. Hernandez-Castillo, R. S. Maeda, J. A. Pruszynski, and J. Diedrichsen. Sensory information from a slipping object elicits a rapid and automatic shoulder response. *Journal of neurophysiology*, 123(3):1103–1112, 2020.

[4] M. Kaboli, K. Yao, and G. Cheng. Tactile-based manipulation of deformable objects with dynamic center of mass. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 752–757. IEEE, 2016.

[5] F. Veiga, B. Edin, and J. Peters. Grip stabilization through independent finger tactile feedback control. *Sensors*, 20(6):1748, 2020.

[6] F. R. Hogan, J. Ballester, S. Dong, and A. Rodriguez. Tactile dexterity: Manipulation primitives with tactile feedback. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 8863–8869. IEEE, 2020.

[7] H. Khamis, B. Xia, and S. J. Redmond. Real-time friction estimation for grip force control. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1608–1614. IEEE, 2021.

[8] T. Hietaranta and M.-M. Linna. Penetrometric measurement of strawberry fruit firmness: device testing. *HortTechnology*, 9(1):103–105, 1999.

[9] R. A. Romeo and L. Zollo. Methods and sensors for slip detection in robotics: A survey. *IEEE Access*, 8:73027–73050, 2020.

[10] W. Chen, H. Khamis, I. Birznieks, N. F. Lepora, and S. J. Redmond. Tactile sensors for friction estimation and incipient slip detection—toward dexterous robotic manipulation: A review. *IEEE Sensors Journal*, 18(22):9049–9064, 2018.

[11] F. Veiga, H. Van Hoof, J. Peters, and T. Hermans. Stabilizing novel objects by learning to predict tactile slip. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5065–5072. IEEE, 2015.

[12] J. W. James, N. Pestell, and N. F. Lepora. Slip detection with a biomimetic tactile sensor. *IEEE Robotics and Automation Letters*, 3(4):3340–3346, 2018.

[13] Z. Su, K. Hausman, Y. Chebotar, A. Molchanov, G. E. Loeb, G. S. Sukhatme, and S. Schaal. Force estimation and slip detection/classification for grip control using a biomimetic tactile sensor. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 297–303. IEEE, 2015.

[14] K. Van Wyk and J. Falco. Calibration and analysis of tactile sensors as slip detectors. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2744–2751. IEEE, 2018.

[15] J. M. Romano, K. Hsiao, G. Niemeyer, S. Chitta, and K. J. Kuchenbecker. Human-inspired robotic grasp control with tactile sensing. *IEEE Transactions on Robotics*, 27(6):1067–1079, 2011.

[16] H. Hasegawa, Y. Mizoguchi, K. Tadakuma, A. Ming, M. Ishikawa, and M. Shimojo. Development of intelligent robot hand using proximity, contact and slip sensing. In *2010 IEEE International Conference on Robotics and Automation*, pages 777–784. IEEE, 2010.

[17] J. Li, S. Dong, and E. Adelson. Slip detection with combined tactile and visual information. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7772–7777. IEEE, 2018.

[18] Z. Deng, Y. Jonetzko, L. Zhang, and J. Zhang. Grasping force control of multi-fingered robotic hands through tactile sensing for object stabilization. *Sensors*, 20(4):1050, 2020.

[19] A. Garcia-Garcia, B. S. Zapata-Impata, S. Orts-Escolano, P. Gil, and J. Garcia-Rodriguez. Tactilegcn: A graph convolutional network for predicting grasp stability with tactile sensors. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[20] S. Dong, D. Ma, E. Donlon, and A. Rodriguez. Maintaining grasps within slipping bounds by monitoring incipient slip. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3818–3824. IEEE, 2019.

[21] F. Kobayashi, H. Kanno, H. Nakamoto, and F. Kojima. Slip based pick-and-place by universal robot hand with force/torque sensors. In *2014 IEEE Symposium on Robotic Intelligence in Informationally Structured Space (RiiSS)*, pages 1–4. IEEE, 2014.

[22] A. Romero, R. Penicka, and D. Scaramuzza. Time-optimal online replanning for agile quadrotor flight. *arXiv preprint arXiv:2203.09839*, 2022.

[23] B. Zhou, J. Pan, F. Gao, and S. Shen. Raptor: Robust and perception-aware trajectory replanning for quadrotor fast flight. *IEEE Transactions on Robotics*, 37(6):1992–2009, 2021.

[24] A. Yamaguchi and C. G. Atkeson. Recent progress in tactile sensing and sensors for robotic manipulation: can we turn tactile sensing into vision? *Advanced Robotics*, 33(14):661–673, 2019.

[25] W. Yuan, R. Li, M. A. Srinivasan, and E. H. Adelson. Measurement of shear and slip with a gelsight tactile sensor. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 304–311. IEEE, 2015.

[26] B. S. Zapata-Impata, P. Gil, and F. Torres. Learning spatio temporal tactile features with a convlstm for the direction of slip detection. *Sensors*, 19(3):523, 2019.

[27] F. Veiga, J. Peters, and T. Hermans. Grip stabilization of novel objects using slip prediction. *IEEE transactions on haptics*, 11(4):531–542, 2018.

[28] Y. Zhang, W. Yuan, Z. Kan, and M. Y. Wang. Towards learning to detect and predict contact events on vision-based tactile sensors. In *Conference on Robot Learning*, pages 1395–1404. PMLR, 2020.

[29] M. Plappert, R. Houthooft, P. Dhariwal, S. Sidor, R. Y. Chen, X. Chen, T. Asfour, P. Abbeel, and M. Andrychowicz. Parameter space noise for exploration. *arXiv preprint arXiv:1706.01905*, 2017.

[30] J. B. Rawlings, D. Q. Mayne, and M. Diehl. *Model predictive control: theory, computation, and design*, volume 2. Nob Hill Publishing Madison, 2017.

[31] M. P. Deisenroth, G. Neumann, J. Peters, et al. A survey on policy search for robotics. *Foundations and trends in Robotics*, 2(1-2):388–403, 2013.