Context-Based Adaptation of In-Hand Slip Detection for Service Robots

Jose Sanchez * **Sven Schneider** * **Nico Hochgeschwender** *
Gerhard K. Kraetzschmar * **Paul G. Plöger** *

* *Department of Computer Science, Bonn-Rhein-Sieg University, Germany (e-mail: {jose.sanchez, sven.schneider, nico.hochgeschwender, gerhard.kraetzschmar, paul.ploeger}@h-brs.de).*

Abstract: Mobile manipulators are intended to be deployed in domestic and industrial environments where they will carry out tasks that require physical interaction with the surrounding world, for example, picking or handing over fragile objects. In-hand slippage, i.e. a grasped object moving within the robot's grasp, is inherent to many of these tasks and thus, a robot's ability to detect a slippage is vital for executing a manipulation task successfully. In this paper, we develop a slip detection approach which is based on the robot's tactile sensors, a force/torque sensor and a combination thereof. The evaluation of our approach, carried out on the Care-O-bot 3 platform, highly suggests that the actions and motions performed by the robot during grasping should be taken into account during slip detection for improved performance. Based on this insight, we propose an in-hand slip detection architecture that is able to adapt to the current robot's actions at run time.

Keywords: Manipulation tasks, slip detection, sensor fusion, tactile sensing, force sensing, run-time adaptation

1. INTRODUCTION

Mobile manipulators such as ARMAR (see **?**) or the Care-O-bot 3 (see **?**) are intended to be deployed in domestic and industrial environments to support humans in their work. Even though, these environments are highly dynamic, the robots are required to successfully and robustly perform a wide range of tasks. The first step to deal with such environments is their perception. Therefore, modern robots, as the ones introduced before, are equipped with a multitude of sensors, not only exteroceptive, but also proprioceptive ones to create a coherent representation of their environment and detect external disturbances. The major problems here are *1*) the limitations inherent to the sensors; *2*) the different characteristics and modalities of the data measured by the sensors; and *3*) the fusion of the measurements considering the previous two problems.

Especially, highly delicate manipulation and grasping tasks that require physical interaction, such as handling of food and fragile goods or handing over objects to humans, demand the precise reaction to disturbances at run time. One such disturbance, which we investigate in this paper, is in-hand slippage, i.e. a grasped object moves within the robot's grasp. To detect in-hand slippage, our robots are equipped with tactile sensors in their hands, but also force/torque sensors in their arms. However, the involved sensors are affected by the actions which the robot performs, for instance, a force/torque sensor is sensitive to motion whereas a tactile sensor might not be affected at all. Also the reactions to accommodate for detected slippage are context-dependent. For instance, when the robot carries an object and feels slippage, it should grasp tighter to avoid loosing the object. Contrary, when a robot hands over an object to a human, a more suitable reaction is to release the grasped object so that the human can take the object.

The contributions of this paper are three-fold:

- We develop three types of slip detectors based on the tactile and force measurements of the robot as well as a fusion of these.
- Then, we experimentally show how the performance of each slip detector varies with the task currently executed by the robot. In this context, the performance of the slip detectors is evaluated on a Care-O-bot 3 platform, where we measure the robustness by comparing the number of successful and unsuccessful slip detection results.
- Finally, we propose an adaptive slip detection approach which enables the run-time selection of slip detectors suitable for the current task.

Following this introduction, Section 2 describes the slip detectors used in this work. The context-adaptive approach and the associated architecture is proposed in Section 3. Section 4 presents the related work and discusses the approach. Finally, our conclusions are summarized in Section 5.

2. SLIP DETECTION

In this section we describe three slip detectors using tactile sensing, force sensing and a combination of them.

2.1 Approach

Based on the study of human tactile sensing, **?** proposed to equip robot manipulators with sensors able to perceive different signals (e.g. vibration, contacts). For instance, a tactile sensor estimates a pressure distribution while a force/torque sensor is able to measure external forces and torques. Furthermore, **?** showed that slip detection can be performed with a combination of force/torque and tactile sensors using a Coulomb friction model. However, it requires knowledge of the friction coefficients which might not be available when handling unknown objects.

We propose an approach to detect slippage of a grasped object that does not require a priori information about the object being manipulated. In this context, a slip is defined as the object being translated within the grasp (e.g. if the object is pulled down this will result in a downwards slip, see Fig. 1 and Fig. 2). Torque and tactile sensors are used to compute signals that indicate a possible slippage.

Based on the torque sensor in each joint, the KUKA Lightweight Robot 4 (LWR4), see **?**, can estimate the wrench (force and torque) applied to the arm's end-effector. The wrench is measured at a rate of 50 Hz. For grasping and manipulating objects, the robot is equipped with the three-fingered Schunk dexterous hand SDH-2. Each finger has two tactile sensors built by **?** to measure pressure caused by contacts. The tactile sensors operate at an average rate of 30 Hz.

Force slip detection We assume a slip occurs whenever a force is exerted in the right direction (e.g. downwards with respect of the grasp frame). A $f_{direction}$ signal is computed as follows:

$$f_{grasp} = R_{grasp}^{sensor} \cdot f_{sensor} \tag{1}$$

$$f_{direction} = f_{grasp} \cdot (f_x, f_y, f_z)^T \tag{2}$$

Where f_{sensor} is the force measured w.r.t. the sensor frame, and R_{grasp}^{sensor} represents the orientation of the grasp w.r.t. the sensor frame. The orientation depends on the hand and grasp type. (f_x, f_y and f_z) selects the direction in which an object can slip up or down within the hand. For example, as shown Fig. 2 this vector is then set to (1, 0, 0). Note that the torque components, for this particular setup, are ignored since their measurements were in the range of noise level (see Fig. 1a). However, the force components were sufficient to detect a slip.

Tactile slip detection To compute the $slip_{tactile}$ signal we apply the algorithm proposed by **?** to each tactile sensor, which estimates the tangential force on the sensor caused by a sliding pressure (e.g. a grasped object slipping). Specifically, a two-dimensional convolution is computed between a tactile sensor's pressure matrix $\mathbf{P}[k]$ of size $(m \times n)$, and its previous pressure matrix, $\mathbf{P}[k-1]$; the output is the convolved matrix, $\mathbf{C}[k]$, of size $(r \times s)$, with $r = (2m-1)$ and $s = (2n-1)$.

We then compute the tactile flow in each axis using the following equations, adapted from **?**, as:

$$flow_x = \frac{\mathbf{a}\mathbf{p}^T}{\sum_{i=1}^{s} p_i}, \quad flow_y = \frac{\mathbf{q}^T\mathbf{b}}{\sum_{i=1}^{r} q_i} \tag{3}$$

Where $\mathbf{a}$ and $\mathbf{b}$ defined as: $\mathbf{a} = [-(n-1), \ldots, -(n-s)]$ and $\mathbf{b} = [-(m-1), \ldots, -(m-r)]^T$, and represent the cell positions in the X and Y direction of the pressure matrix, respectively. Furthermore, $\mathbf{p}$ and $\mathbf{q}$ are vectors representing the mean value of columns and rows of the convolution matrix $\mathbf{C}[k]$, respectively. Defined formally as,

$$\mathbf{p} = \left\{ \frac{1}{r} \sum_{i=1}^{r} c_{ij} \right\} \qquad for \quad j = 1, \ldots, s \tag{4}$$

$$\mathbf{q} = \left\{ \frac{1}{s} \sum_{j=1}^{s} c_{ij} \right\} \qquad for \quad i = 1, \ldots, r \tag{5}$$

Finally, the tactile flow is found using the results of equation 3,

$$flow_{tactile} = \|flow[k] - flow[k-1]\|^2 \tag{6}$$

with $flow = [flow_x, flow_y]$. This computation can be applied to tactile sensors of different shapes, provided their output is a two dimensional array. Having N tactile sensors, we define $slip_{tactile}$ as:

$$slip_{tactile} = \sum_{n=1}^{N} \mathbb{E}\left[\mathbf{P}[n]\right] \cdot flow_{tactile}[n] \tag{7}$$

Where $\mathbb{E}[\mathbf{P}[n]]$ is the pressure average and $flow_{tactile}[n]$ is the tactile flow of the n-th tactile sensor. This linear combination allows tactile sensors with higher pressure values to contribute more information regarding how an object is slipping from the grasp, since sensors with lower intensity values, arising from spurious contacts, might not provide an accurate measure of slippage. Note that the $flow_{tactile}$, as defined in equation 6, is an absolute value and thus the direction of the slippage is not considered to produce the $slip_{tactile}$ (see Fig. 1b).

Combined slip detection The $slip_{combined}$ is computed by combining both tactile and force slip signals as

$$slip = \begin{cases} \texttt{slip up} & \text{if} \quad (slip_{tactile} \geq threshold_{tactile}) \wedge \\ & \quad\quad (slip_{force} \geq threshold_{force}) \\ \texttt{slip} & \text{if} \quad (slip_{tactile} \geq threshold_{tactile}) \wedge \\ \texttt{downwards} & \quad\quad (slip_{force} \leq -threshold_{force}) \\ \texttt{n/a} & \text{otherwise} \end{cases}$$

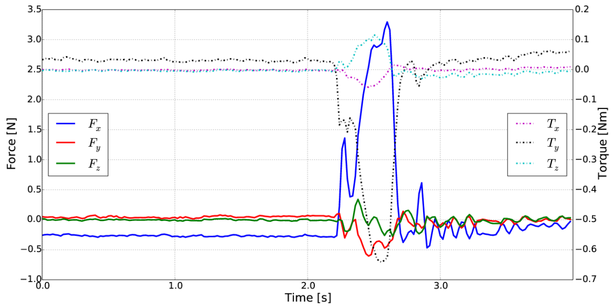
Where the $threshold_{tactile}$ and $threshold_{force}$, represent numerical values[1] for detecting a slip based on the $slip_{tactile}$ and $slip_{force}$, respectively. Both of these thresholds are chosen experimentally.

2.2 Experiments & Evaluation

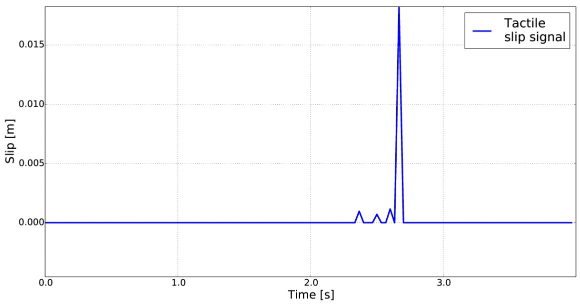
The three slip detectors were evaluated using two different grasp shapes, namely a grasp that uses all three fingers and one that only uses two fingers. The $threshold_{force}$ and $threshold_{tactile}$ were set to 1.5 and $5e^{-3}$, respectively. Three different objects were used in the experiments. A coffee paper cup, an empty Pringles can and a Sprite bottle. Seven actions were performed on each object ten times, producing 60 tests per action. The seven actions are described as follows:

grasp: the fingers of the gripper close to hold the object

[1] These values can be chosen to increase the sensitivity of the slip detectors, e.g. lower values result in higher false positives.



(a) Force/torque sensor raw output.



(b) Tactile slip signal computed using Equation 7.

Fig. 1. A slip downwards occurring at approximately 2.5 seconds. (a) shows the raw output of the force/torque sensor and (b) presents a computed signal based on the tactile array sensors. Slip down perceived by the force (a) and tactile (b) sensors.

move base: the robot's base moves back-and-forth approximately 20 cm while holding the object
release: the fingers of the gripper open to release the object
rotate counter-clockwise: the grasped object was rotated counterclockwise within the grasp
rotate clockwise: grasped object was rotated clockwise within the grasp
slip downwards: the grasped object was pulled down to simulate the object slipping down (shown in Figure 2)
slip up: the grasped object was pulled up from the gripper.

Only **slip downwards** and **slip up** were considered as a slip. The results for the three detectors are summarized in Table 1 and Table 2.

2.3 Preliminary Results & Discussion

The performance of each slip detector varies considerably depending on the action, e.g. the *tactile slip detector* outputs a slip whenever grasping an object, as shown in Table 1. Contrary, the *force slip detector* achieved a perfect accuracy for detecting actual slips as displayed in Table 2, however its performance was poor when no slippage occurred, particularly in the **move base** action (see Table 1). The *combined slip detector* had the lowest accuracy for actual slips, but its performance was by far the best when no slippage occurred. The experiments also showed that the slip detectors achieve a similar performance (±5%) regardless of the object and grasp. Already in the few experiments we have performed with

Fig. 2. Robot platform used for the evaluation.

Table 1. Performance rate of slip detectors in detecting a true slippage.

	False positives		
	Tactile	**Force**	**Combined**
`grasp`	60/60	10/60	0/60
`move base`	1/60	51/60	0/60
`release`	50/60	1/60	1/60
`rotate counter-clockwise`	39/60	6/60	1/60
`rotate clockwise`	50/60	9/60	4/60

Table 2. Performance rate of slip detectors in detecting a true slippage.

	True positives		
	Tactile	**Force**	**Combined**
`slip downwards`	53/60	60/60	49/60
`slip up`	50/60	60/60	47/60

the inclusion of actions (i.e. grasp, release and move base), we can see that the action has a major influence on the performance of the slip detection.

3. ROBUST SLIP DETECTION VIA CONTEXT-BASED ADAPTATION

The evaluation of the stand-alone slip detectors in the previous section indicates that the robustness of slip detection clearly benefits from the run-time adaptation of the manipulation architecture based on the action that the robot performs. At the core, the proposed architecture (see Fig. 3) consists of a control loop where a grasp controller commands the robot's hand to grasp an object. As input, the grasp controller receives, on the one hand, the grasp to be executed (e.g. desired joint values) and, on the other hand, a signal if the object slips within the hand. As described in the previous section, the slip signal is derived from the data provided by the robot's tactile sensors and the force/torque applied to the robot's arm. To

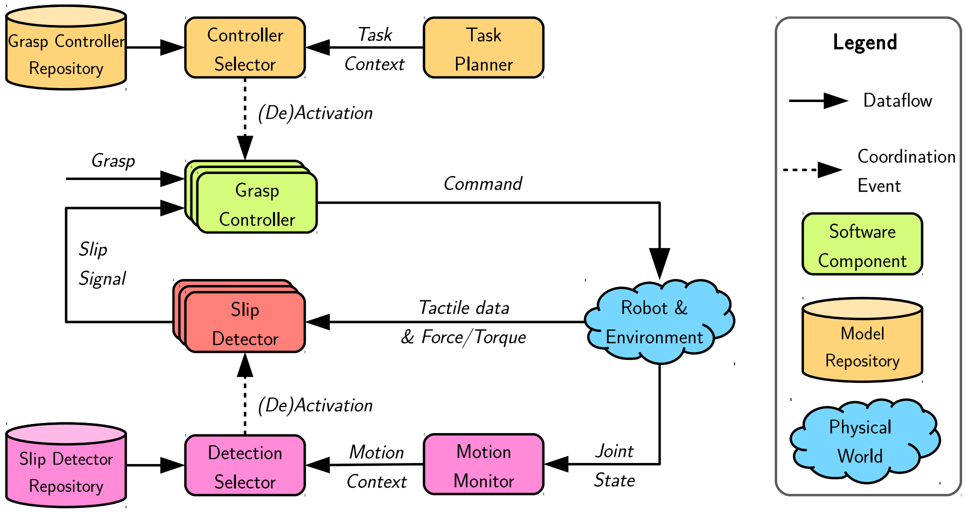


Fig. 3. The core of the architecture consists of the slip detector (red) to analyze the robot's state and the grasp controller (green) which commands the robot. At run time, a specific slip detector is selected by the detector selection pipeline (violet). Similarly, a grasp controller can be selected online, given the robot's task context (yellow).

accommodate for the different robot's actions, we propose the design of multiple slip detectors and grasp controllers which are in turn selected at run time with the knowledge of the robot's action and task context.

3.1 Representation and Storage of Slip Detectors

To make different slip detectors available during run time, we need to model and store them in an explicit manner. We employ the Robot Perception Specification Language (RPSL) introduced in **?** to model the various slip detectors and their corresponding properties. The RPSL is a domain-specific language (DSL) which provides suitable abstractions to model robot perception systems such as those introduced in this paper (see Fig. 4). In particular, with RPSL we can model multi-stage slip detectors in the form of directed acyclic graphs where sensor components (e.g. tactile and force sensors) are root nodes and processing components are leaf nodes. Further, RPSL provides means to model the data in- and output types employed in the slip detectors such as pressure matrices of the tactile sensors. Beyond structural information the RPSL enables the attachment of contextual information about the appropriateness of a slip detector for a certain context. In this work, the appropriateness is expressed as a ranking of symbols where each symbol represents a motion context such as `grasp`, `move_base` and `release`. Similarly to our previous work (see **?** and **?**), the attachment of contextual information is performed during design time and after some preliminary experimentation when more *domain knowledge* is available.

3.2 Monitoring Motion

To retrieve the current motion context we perform symbol grounding by employing the Conceptual Space (CS) knowledge representation framework, as proposed by **?**. The CS framework is supported by RPSL and allows to ground different motion contexts such as `grasp`, `move_base` and `release` through the notion of concepts, domains and dimensions. In the CS framework, concepts are convex regions in a set of domains which are composed of measurable dimensions. In the context of this work the measurable dimensions are the joint values (velocities) for each joint. This allows us to model for each concept a set of prototypes encoding typical values. For example, the prototype for the `move_base` concept has only zero-values for each arm joint whereas the base joints are non-zero. During run time, the motion monitor then computes for each joint state sample the closest matching prototype by employing the Euclidean distance as a metric.

3.3 Run-time Selection of Slip Detectors

To select a slip detector which is appropriate for the current motion context we apply a simple, yet powerful rule-based approach. During design time we devise a set of decision rules where the motion context is part of the condition and the selection of a slip detector is part of the rule body. The slip selector also performs the activation of slip detectors on an implementation level. This happens only, if the selected slip detector is not already activated. One might argue that activating and deactivating components is unfeasible from a timing perspective. However, as we have shown in **?** activating and deactivating system components in a time-critical robotic application context is feasible.

3.4 Task-based controller adaptation

In parallel to the adaptation of the slip detection pipeline it is also beneficial to select specific grasp controllers during the robot's run time. Here, we build upon our previous work in **?**, where we proposed a grasp controller that reduces the exerted force on the grasp object. Obviously, the exerted force is not the only criterion to be optimized in grasping, but also the stability of the object within the hand should be taken into account. This is achieved by feeding the signal produced by the slip detector back into the grasp controller. In a pick and place task the feedback signal should be represented numerically so that the grasp controller can adapt the grasp accordingly.

```
rpsl.sensor_component do
   name "force_sensor"
   add_port :out, "out_port", "wrench"
end

rpsl.processing_component do
   name "slip_detection"
   add_port :in, "in_port", "wrench"
   add_port :out, "out_port", "force_slip"
end

rpsl.perception_graph do
   name "force_slip_detector"
   connect "force_sensor", "out_port",
           "slip_detection", "in_port"
   attach_prototype "task_context", "move_base"
end
```

Fig. 4. An excerpt of the domain model of the force-based slip detector represented in RPSL (see **?**). Two atomic components are modeled, namely a `force_sensor` providing wrench data (`out_port`) and a `slip_detection` component demanding wrench data (`in_port`) and providing a slip signal (see Sec. 2). Both components are connected in the `force_slip_detector` yielding a structurally complete specification of the force-based slip detector. Note, the attached `task_context` expressing the suitability of the `force_slip_detector` for scenarios when the base is moving. Due to space reasons the specification of the data types (e.g. `wrench`, `force_slip` and `task_context`) is not shown.

However, for a hand-over task, it is already sufficient to provide the grasp controller with a symbolic signal, where upwards slip indicates that the robot should release the object, whereas downwards slip means that the robot should grasp the object more tightly. Further task-specific controllers are, for example, required in tactile exploration as proposed by **?**, measuring object's in-hand motion (see **?**) or reactively placing a grasped object as shown by **?**.

In order to select an appropriate grasp controller, the robot requires knowledge about the individual grasp controllers which we also represent in RPSL. This is possible because both, slip detectors and grasp controllers, share structural properties which are general enough to be modeled with RSPL. In addition the selector requires knowledge of the task which the robot is about to perform. This task context is represented symbolically, e.g. as `hand_over`, `pick` or `place`. In contrast to the slip detector selection, the task context does not have to be derived from the robot's motions. Instead, for instance, a knowledge base of the robot can be queried for information about the current task or a task planner, situated above the manipulation pipeline in the robot control architecture, can provided this information directly.

4. RELATED WORK AND DISCUSSION

4.1 Use cases

The ability to detect in-hand slippage can be helpful in different use cases. One such use case is that of a grasp controller (see **?** and **?**), where the grasp force exerted on an object can be computed proportionally to a continuous slip signal. A continuous slip signal is also required for tasks such as measuring the in-hand motion of a grasped object (see **?**). On the contrary, a hand-over task might only require a discrete slip signal, for instance, to detect when a person is pulling the grasped object from the robotic gripper. Similarly, a robot placing an object can benefit from a discrete event signaling a slippage, i.e. the object is in contact with a support surface and is thus slipping from the grasp.

4.2 Limitations

As part of a critical discussion we also would like to outline the main limitations of our approach:

- Up to now, we have not evaluated in how far the run-time adaptation of the grasp controller and slip detector influences the stability of the core control loop stability. Should this turn out to be a critical issue, it seems worthwhile to identify further knowledge and annotate the model repository of slip detectors and grasp controllers accordingly.
- Additionally, the task context is only represented symbolically. Here, a more in-depth analysis of the relationship between the task context and the grasp controller selection could reveal further information that allows to improve the selection.
- Even though, the current chosen rule-based approach is simplistic in structure, we do not depend on it. In fact, the adaptation mechanism could be replaced with other adaptation methods (e.g. constraint satisfaction etc.) as the knowledge to represent the context and the slip detectors would remain the same.

4.3 Related Work

Although tactile sensing in robotics has been researched for over two decades (see **?**), it has recently been applied to very diverse manipulation tasks. For instance, **?** use tactile information to execute corrective actions on a PR2 that improved the robot's ability to grasp an object when greater positional errors are present. Although usage of tactile information improves the sensing capabilities of a robot, it is the addition of different sensing modalities that has proven to be a go-to solution to increase a robot's robustness in an uncertain and dynamic environment (see **?**). **?** present an example of multi-sensor fusion, where they integrate tactile sensing with vision and force signals to improve their robot's ability to physically interact with the environment (e.g. sliding a door open), demonstrating the increase of robustness when using the three modalities combined. Slip detection has been another application where tactile sensing plays a major role; **?** increase the grip force of the robot when a slippage is detected. Furthermore, Romano et al. propose a phase-based architecture to address the task of grasping an object and placing it in another location, where the phases are detected based on information produced by tactile and force sensors.

5. CONCLUSIONS

This paper presented an approach for run-time selection of a slip detector with the best performance for the current task being executed by the robot. More specifically, we implemented three slip detectors based on tactile, force/-torque signals and the fusion of these signals. A preliminary evaluation provided insights on how the performance of the slip detection approaches depends on the action that the robot executes. Based on these insights, we proposed a context-adaptive architecture that improves the robustness of the slip detection and is also able to select an appropriate grasp controller based on the required task. Future work will be focused on quantitatively evaluating the run-time adaptation of the grasp controller, as well as developing controllers required for different tasks as mentioned in Section 4. Furthermore, we would like to evaluate different robot actions (e.g. arm motions) to observe their impact on the developed slip detectors.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the on-going support of the Bonn-Aachen International Center for Information Technology. Nico Hochgeschwender and Sven Schneider received a PhD scholarship from the Graduate Institute of the Bonn-Rhein-Sieg University which is gratefully acknowledged.