

# Integrating vision, haptics and proprioception into a feedback controller for in-hand manipulation of unknown objects

Qiang Li, Christof Elbrechter, Robert Haschke and Helge Ritter

**Abstract**—We propose a feedback-based solution for the accurate manipulation of an unknown object in hand. This method does not explicitly model friction and surface geometry details, but employs a fast feedback loop based on visual and tactile feedback to perform robust manipulation even in the presence of unexpected slippage or rolling.

At every control step, fingertip motions are computed to realize the intended object relocation, employing a composite position/force controller. Subsequently inverse hand kinematics is employed to retrieve joint-level motions, which are implemented on the robot with a position servo loop.

We evaluate our method on a setup of two KUKA robot arms, each equipped with a tactile sensor array as end-effectors to perform the object manipulation task. The experimental results show the feasibility of our proposed method, even in presence of slippage or external disturbances.

## I. INTRODUCTION

Humans use a large number of in-hand object manipulation behaviors in their everyday lives. For example, we grab our mobile phone and change its pose to make a call or send a message. We pinch a pen and change its pose in hand to a comfortable posture for writing. Such basic and simple behaviors can be performed easily even by children. However, they cannot yet be realized by the most advanced robot hands. In this paper, we focus on this challenging task of dexterously manipulating an object within a multi-fingered robot hand, i.e. moving the object with respect to the hand.

A large body of work addressing this problem follows an analytical approach based on robot manipulation theory [11]. While sound, this theory makes strong assumptions in order to perform dexterous manipulation tasks: the object properties (friction, geometry, mass, etc.), the relative motion between the fingertips and the object, and the robot hand model must be known. Using this knowledge, the robot hand can deliberately plan its motion offline and implement accurate manipulation tasks in the real world [7].

In contrast to this open-loop planning approach, also many feedback-based approaches have been proposed. For example, [17], [18] proposed vision-free grasping and manipulation for a triple-fingered robot hand. Solely relying on tactile feedback, they define the current object pose by a virtual frame calculated from estimated contact points and relocate the object w.r.t. this frame following an empirical feed-forward control model. However, because visual feedback about the actual object pose is disregarded, accurate control in the presence of unexpected slippage or rolling is impossible.

The authors are with the Neuroinformatics Group / CITEC, Bielefeld University, Germany. {qli, celbrech, rhaschke, helge}@techfak.uni-bielefeld.de

Wimboeck et al [19] also use the virtual frame idea to realize object-level impedance control. Their model-based control method requires an accurate robot dynamics model to compute desired joint torques. Gruben et al [10] proposed a control basis framework to perform complex manipulation tasks. They use gradient descent to search for maximum grasp quality configurations employing torque sensor feedback and multiple probing of the object surface. Finally, they consider dexterous manipulation by sequencing a series of stable grasps. Both methods can solve the manipulation task by torque-controlled robots, but their methods are not suitable for position-controlled ones.

In this paper, we propose the integration of vision, tactile sensing and proprioception to accurately perform object manipulation tasks. Concretely, we use vision to extract the object pose, joint-angle feedback and forward kinematics to estimate contact positions w.r.t. the robot frame, and tactile sensing to estimate contact forces. All sensory feedback signals are incorporated into a closed-loop manipulation controller.

Conceptually, we divide the object manipulation process into two stages: a local object relocation step and a global regrasping step. The local controller reactively moves the object by a small amount limited by the motion range of the hand. A higher level planner will rearrange the finger configuration to allow continuing the local motion. To this end, a finger is selected for relocation, while the other fingers maintain a stable grasp. The selected finger actively explores the object's surface to find a new optimal grasp configuration. In our previous work, we have employed physics-based simulation to show the feasibility of this approach [5]. The present paper will focus on an evaluation of the local manipulation controller, i.e. the first stage, on a real robot.

The paper is arranged as follows. In the next section, we discuss our methods to estimate the object pose and contact positions/forces from visual and tactile sensors. In section III we discuss the manipulation scenario and the local manipulation controller, before in section IV the experimental setup is detailed and the relocation capabilities are evaluated. Section V discusses the results and parameter choices. Finally, we conclude with a summary and an outlook.

## II. OBTAINING OBJECT POSE AND CONTACT LOCATIONS

Focusing on the manipulation task, we employ a fiducial marker attached to the object in order to easily estimate the object's 6D pose. The monocular camera was calibrated applying standard calibration methods using a known 3D calibration object [4]. The BCH-code-based marker provides

TABLE I: static object pose estimation variance

position [cm]			orientation [deg]		
	raw	filtered		raw	filtered
x	$\pm 0.052$	$\pm 0.012$	roll	$\pm 0.017$	$\pm 0.004$
y	$\pm 0.008$	$\pm 0.002$	pitch	$\pm 0.013$	$\pm 0.003$
z	$\pm 0.052$	$\pm 0.012$	yaw	$\pm 0.004$	$\pm 0.001$

four highly reliable and efficiently detectable key-points (at the corners) with known coordinates with respect to the object frame. From these we can calculate the object's pose employing standard pose detection from planar targets [20].

To evaluate the accuracy of this pose estimation method, we compare its results to ground truth data obtained from a stereo-camera system (stereo basis: 20cm). The results are shown in Tab. I given a static, known object pose. The position error is naturally dominated by the depth estimation error. The raw pose estimation results considerably vary over time due to visual sensor noise. To reduce this variance, we employ a sliding-window averaging filter (across 10 samples), which reduces variance by a factor of four to five, thus providing suitable feedback signals for robot manipulation.

As can be seen from Fig. 1, showing the pose estimation result and the recorded pose error during a human manipulation sequence, the angular error gets larger if the marker's normal and the camera's view-vector become more (anti)parallel. However, both position and angular error stay in an acceptable range for the envisioned manipulation task.

The marker-detection and pose estimation system runs at the full camera frame rate of 30Hz, which is sufficient for the robot control cycle. In more natural manipulation scenarios, the marker-based pose detection module could be replaced by marker-less object tracking frameworks such as [3] or even Microsoft Kinect-based methods such as [9].

Another, even more important feedback channel for manipulation is tactile sensing. As absolute accuracy of human tactile sensing is also limited, we propose to estimate contact positions and force from a modern tactile sensor providing an array of  $16 \times 16$  tactels with a spacing of 5mm in each direction [14]. The sensor is tuned towards high frame rates (up to 1.9 kHz), rendering a use for real-time robot control feasible. It exploits the piezo-resistive sensing principle, measuring changes in resistance of a conductive foam due to an applied normal force.

As a first processing step we need to identify the contact region on the sensor, which typically extends over a larger image region due to the softness of the sensor foam. To this end, we employ connected component analysis [15], well known from image processing, to extract all connected regions in the binarized tactile image and choose the largest one as the considered contact region  $R$  – neglecting all smaller regions as originating from noise or spurious contacts. An example contact region is shown in Fig. 2. Subsequently, we compute the overall contact (normal) force  $f$  as the sum of forces  $f_{ij}$  within the contact region and the

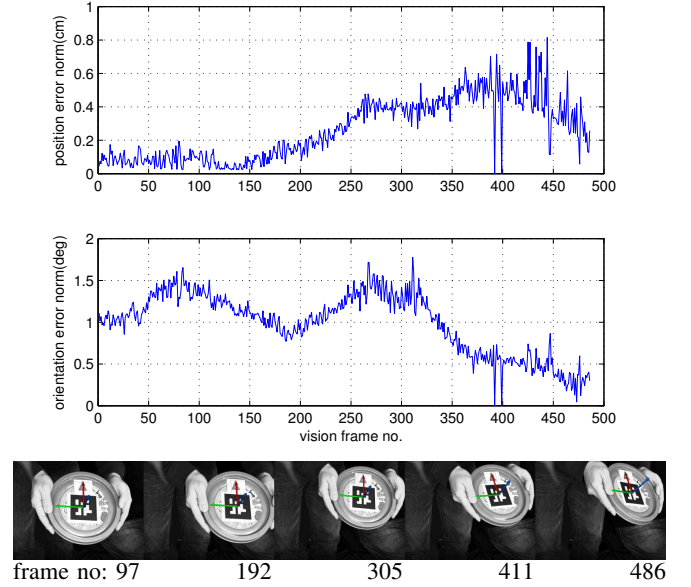


Fig. 1: Human manipulation sequence and obtained pose estimation error.

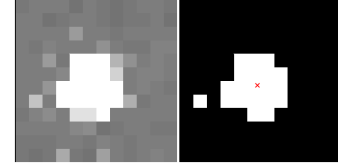


Fig. 2: contact blob and center of gravity

contact position  $\mathbf{c}$  as the center of gravity (cog) of  $R$ :

$$f = \sum_{ij \in R} f_{ij} \quad \mathbf{c} = f^{-1} \sum_{ij \in R} f_{ij} \mathbf{c}_{ij}, \quad (1)$$

where  $\mathbf{c}_{ij}$  are the discrete coordinates of the tactels on the sensor surface. Due to the averaging effect from multiple tactels composing a contact region, we obtain a sub-tactel resolution for the contact position as can be seen from Fig. 3. Exploiting proprioceptive feedback and calculating the forward kinematics, the 2D contact location on the sensor is mapped onto a 3D Cartesian position.

### III. LOCAL MANIPULATION CONTROLLER

Conventional grasp and manipulation planning methods [2], [8] uncouple the planning from the control stage. The planning stage strongly depends on global knowledge about the geometry of the object and the fingertips. Certain works also explicitly consider spherical finger tips to facilitate the geometry-based planning process [16]. Furthermore, the friction coefficients for all contacts are required to evaluate grasp stability.

In real world scenarios, especially when handling unknown objects, this information is not available. Nevertheless humans can easily manipulate objects without this knowledge. We assume that the incredible dexterity of human manipulation originates from tight control loops employing

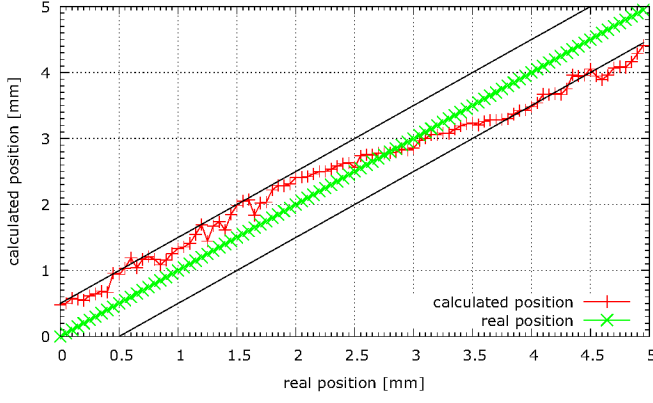


Fig. 3: Computed contact position (cog) when moving a 2mm-diameter probe tip between two adjacent tactels. Due to the weighted averaging between different tactel locations, we yield a spatial accuracy of ca. 0.5mm despite the much larger tactel size of 5mm.

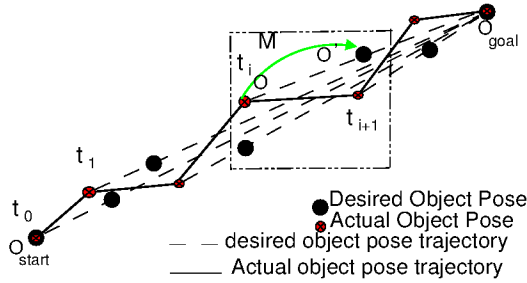


Fig. 4: Incremental manipulation of object pose  $O$ .

tactile sensor feedback. Consequently we propose to employ tactile feedback to estimate contact positions and forces and introduce a manipulation strategy mainly based on this feedback. If friction properties and joint torques are not available anymore, we cannot actively control rolling and slipping, because internal forces cannot be designed. However, as we will show, local object manipulation is possible without explicitly designing all details of physical hand-object interaction.

#### A. Contact Position Planning

Fig. 4 illustrates our overall control concept showing the evolution of the object pose  $O$  from an initial one ( $O_{start}$ ) to the goal ( $O_{goal}$ ). In every control cycle  $t_i$  (see the box) we compute an intermediately targeted object pose  $O'$  towards the final target pose. While more complex motion planning methods could be applied here, we adopt a simple PI controller to do so:

$$O' = O + (k_p * (O_{goal} - O) + k_i * \int (O_{goal} - O)) \quad (2)$$

Here,  $O_{goal} - O$  denotes the pose error between the current and the target pose. In order to calculate corresponding finger tip motions realizing this new object pose  $O'$  without knowledge about the exact object geometry, we make the assumption that contact positions  $\mathbf{p}_i^o$  do not move relative

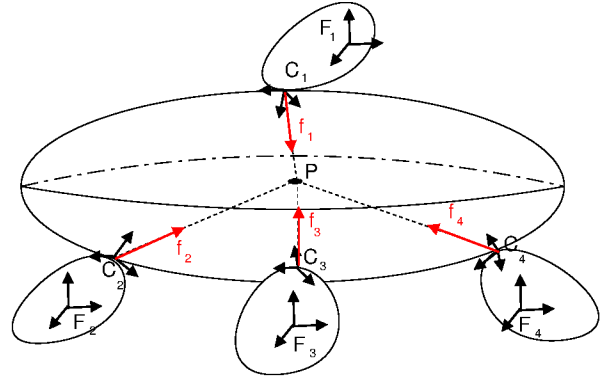


Fig. 5: Force planner employs centroid  $\bar{\mathbf{p}}$  of contact locations to obtain contact force directions.

to the object within a control cycle. Under this essential assumption, we calculate the new contact positions  $\mathbf{p}_i'$  with respect to the palm as follows:

$$\mathbf{p}_i' = O' \cdot \mathbf{p}_i^o. \quad (3)$$

From this we can compute the required positional changes  $\Delta \mathbf{p}_i = \mathbf{p}_i' - \mathbf{p}_i$  for all contact points.

Of course, the assumption that there is no relative motion between the fingertips and the object is only an approximation. Because the exact contact geometry as well as grasp stability measures are not explicitly taken into account, some slipping and rolling will occur. However, the sensor feedback available in the next control cycle will allow us to recognize and correct this undesired contact motion.

#### B. Contact Force Planning

A mere kinematic consideration of the problem as discussed so far is not sufficient. In order to maintain a stable grasp and to not break the object, we have to control contact forces as well. Conventional contact force planners strive for a globally optimal contact force distribution ensuring grasp stability, for example, all contact forces stay within corresponding friction cones, the totally applied force exactly resists the external forces (e.g. gravity), and the local contact forces are limited. This general solution is meaningful only if the contact force is controllable. However, we assume that there is no 3D contact force feedback (obtained directly or indirectly), but only the force magnitude is available from the tactile sensors.

Following the approach from [16], we plan the force direction such that the resultant moment will be zero by ensuring that the contact force directions of all fingers intersect in one point, which is chosen to be the centroid  $\bar{\mathbf{p}}$  of contact locations (see Fig. 5). Subsequently we can prescribe force magnitudes along these directions such that the resultant force becomes zero as well. Hence, the force planner calculates desired contact force magnitudes along the *contact directions*, from which we can obtain force errors.

Both the force ( $\Delta \mathbf{f}_i$ ) and positional errors ( $\Delta \mathbf{p}_i$ ) are fed into a composite position/force controller, which calculates the effective contact position error  $u$ , which in turn is fed to

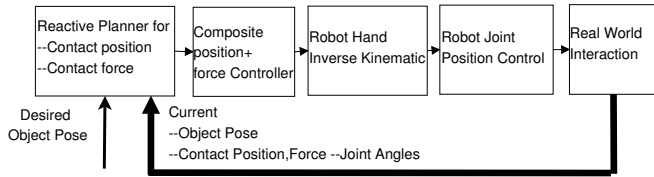


Fig. 6: Low level, local manipulation control scheme.

an inverse hand kinematics module to compute actual joint velocities. Fig. 6 summarizes this control scheme. For a more detailed introduction of the local manipulation controller we refer to our previous work [6].

#### IV. EXPERIMENTAL EVALUATION

In contrast to our previous works, where we proved the feasibility of the manipulation approach in physics-based simulation only (due to the lack of appropriate tactile sensors), in the present work we evaluate the method using two KUKA LWR arms, each equipped with a tactile sensor module. Thus, both arms act as two large fingers with tactile-sensitive fingertips.

The information about the shape, size, and friction of the manipulated object is not available to the robot. The experimental setup is shown in Fig. 7. The camera observes the object from the top. We point out that we use only the joint encoder feedback and not the torque feedback provided by the KUKA arms. Vision feedback frequency is 30Hz, and we limit the tactile feedback to the same frame rate. They are both processed by a smoothing filter averaging within a window 20 frames. Proprioceptive feedback and joint angle control rate are both fixed at 125Hz. We extract joint angle measurements and send joint control commands via the KUKA FRI interface [13], [12]. All controller gains are manually tuned to guarantee the stability of controllers in all manipulation experiments.

The whole manipulation process comprises three phases:

- Vision-guided grasping of the object.
- Moving along world's  $Z$ ,  $X$ , and  $Y$  axis in sequence.
- Rotating around  $Z$  and  $X$  axis.

Each phase is described in detail in the following.

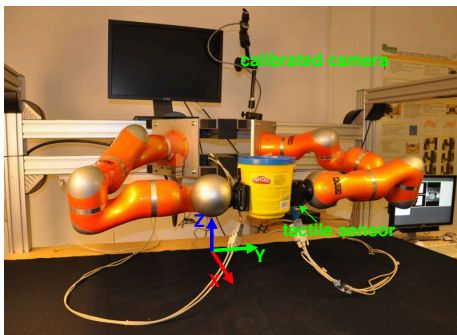


Fig. 7: Experimental setup using two KUKA arms with attached tactile sensor arrays as large fingertips.

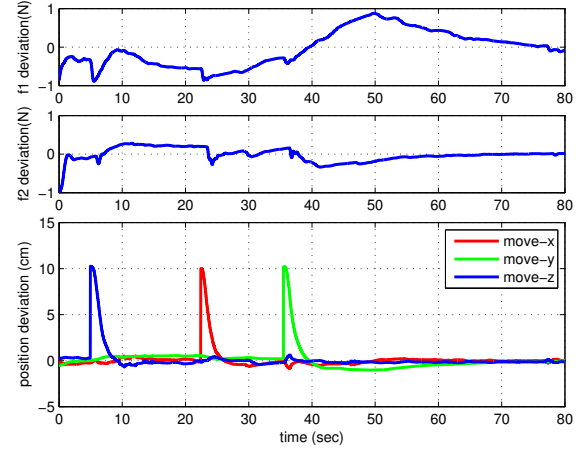


Fig. 8: Evolution of force ( $f_1$ ,  $f_2$  of both tactile sensors / fingertips) and positional errors during translating motion.  $x, y, z$  errors are mapped to red, green, blue colors resp.

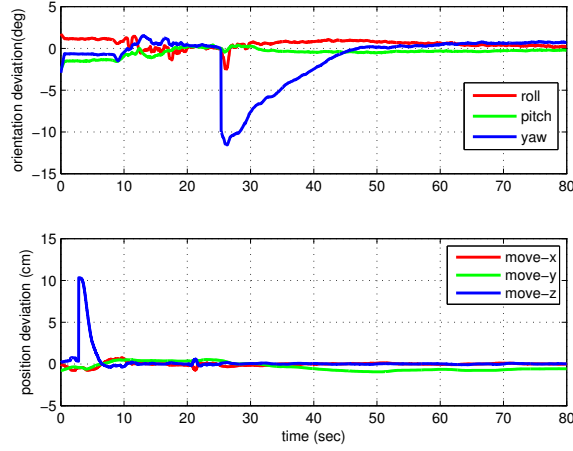
*a) Vision-guided grasping:* The first stage is to guide the arms to contact the object, exert the planned grasp force (see III-B) and hold the object. The force planner – originally designed for multi-fingered hands – has been simplified for the two-arm scenario: While the contact force vectors still point towards the centroid of contact locations (see Fig. 5), their magnitudes are set equally to a predefined value.

To establish object contact, we apply a simple, hard-coded opposition strategy: Starting from the estimated pose  $(x, y, z)$  of the marker attached to the object we attempt to drive both fingertips, i.e. arm end-effectors, to the virtual grasp point  $(x, y, z - z_0)$  slightly below the marker frame, where the offset  $z_0$  is a constant determined by the size of the tactile sensor module. The approaching motion of both arms is stopped as soon as contact to the object is detected by the tactile sensor.

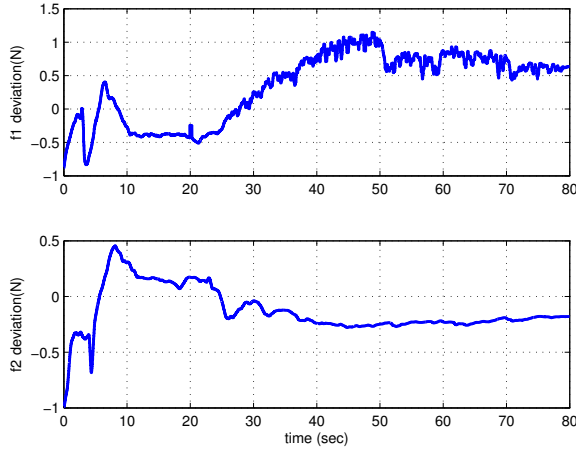
The coarse calibration of the tactile sensors does not provide accurate enough force feedback to stably hold the object with a pure force-feedback controller. Rather, the object will slowly drift away. However, exploiting visual feedback about the object position too, the composite position/force controller successfully accomplishes the grasping task.

*b) Translating motion:* In the first experiment, the object is moved 10cm along the world's  $z$ ,  $x$ , and  $y$  axes in sequence. The resulting trajectories for force and positional errors are shown in Fig. 8. As can be seen from the deflections in the bottom subfigure, a new target pose was set after 5, 22, and 35 seconds. In all cases the positional error quickly decays to the noise level.

Considering the force error trajectories, we see that the motion along the  $y$ -axis generates most deviations. This is because, this motion direction is parallel to the contact normal, thus heavily demanding the composite force/position controller. As soon as the positional error along the  $y$ -axis stabilizes around zero, also the force error starts to decay.



(a) rotational and positional errors

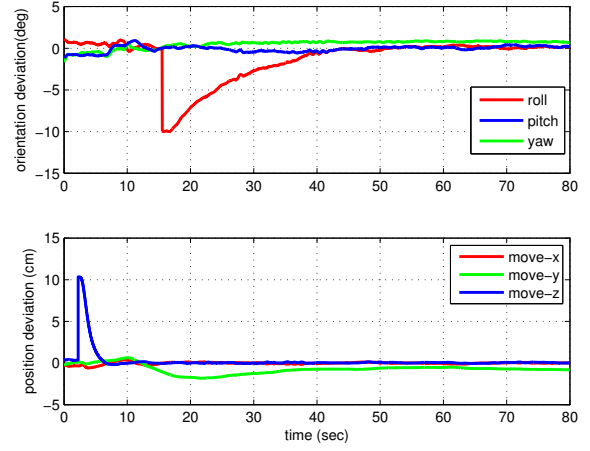


(b) contact force errors

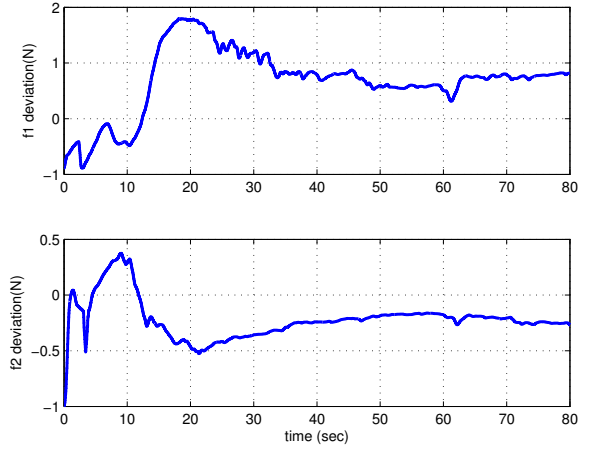
Fig. 9: Error evolution rotating around world's  $z$ -axis

*c) Rotating motion:* Secondly, we will show how the object *orientation* can be controlled. In each experiment, firstly the object will be lifted 10cm along the  $z$ -axis before being rotated around the  $z$  resp.  $x$  axis. The results of both experiments are shown in Figures 9 and Fig. 10. Again, the positional and rotational errors quickly decay after setting a new target pose. However, rotational errors are corrected more slowly due to a more conservative choice of controller gains.

Looking at the force error trajectories, we observe that the errors do not completely decay anymore. This is due to the fact, that the physically applied force direction isn't normal to the sensor surface anymore. However, the sensor only measures normal forces. Again, incorporating visual feedback and employing the composite position/force controller we can realize stable object rotation nevertheless. All experiments are also shown in the accompanying video [1].



(a) rotational and positional errors



(b) contact force errors

Fig. 10: Error evolution rotating around world's  $x$ -axis

## V. DISCUSSION

As we pointed out in the discussion of the grasping stage, the force calibration of the piezo-resistive sensor is too coarse to allow force-only feedback control for grasping. Differing force magnitudes at opposing contacts will lead to a drift of the object. However, as we have seen from our experimental results, it is not necessary to improve on the force measurement accuracy of the hardware. Rather, we can compensate for this weakness using intelligent control strategies.

In the present work, we relied on object pose feedback from vision to solve this issue using a composite position and force controller. However, even if object pose feedback is not available, we can compensate for drifts using proprioceptive feedback: Aiming to stably hold an object, we will expect a stable end-effector pose as well (within noise level). However, if we observe a drift of the end-effector pose, we can trace it back to force deviations. In other words, instead of using external feedback about the object pose, we could also employ proprioceptive feedback to estimate the current



object's pose and use that in the composite controller to compensate for drifts.

As detailed in our previous work [6], the composite controller computes the final control signal by superimposing the control signals from both sub controllers, the position and force controllers. Naturally, linear superposition may lead to destructive interference, i.e. non-zero control inputs from sub controllers may add up to zero.

To circumvent this effect, we exploit the fact, that PI-type controllers can compensate for systematic errors, thus realizing higher priority control. That is, the more important control variable will be controlled using a PI-type controller, while the sub-ordinated one employs a P-type controller. In our case, controlling the pose of the object, the position control part is most important, thus using a PI-type controller. In contrast, force is controlled using a P controller. This also contributes to the poor tracking results visible in Figures 9b and 10b.

As a matter of fact, PID controllers are sensitive to proper parameter tuning. We obeyed general rules for PID gain tuning: Firstly  $K_p$  parameters are regulated until the system begins to oscillate. Then the derivative gain  $K_d$  is employed to reduce oscillations. Finally, the integration component  $K_i$  is added to eliminate the steady state errors. We used the same parameter sets for both arms.

## VI. SUMMARY

We proposed a reactive control strategy to realize local manipulation motions for unknown objects. In contrast to traditional manipulation strategies, which require a lot of information about the object and which plan in an offline fashion, our method plans in an online fashion and employs minimal object and world information. The position and contact force planners are designed independently and are coordinated by a composite controller. Finally, we proved the feasibility of the method to manipulate unknown objects using a real robot platform composed from two KUKA arms with attached tactile sensor arrays acting as two large finger tips grasping the object. Because the proposed method does not depend on the object geometry, it can be easily employed for differently shaped objects as was shown in the accompanying video [1].

We are also working on improving the tactile sensing capabilities of our anthropomorphic Shadow Robot Hand. Once new tactile sensors will be available, we will extend the experimental evaluation to the regrasping strategy as well, which already proved feasible in our previous work employing physics-based simulation [5].

## ACKNOWLEDGEMENT

This work is funded by SPP priority program "Autonomous Learning" funded by DFG. Authors would like to thank Mr. Slobodan Vukanović for checking the English writing.

## REFERENCES

- [1] Qiang li. <http://www.youtube.com/watch?v=UgCv5ESAYfc>.
- [2] A.A.Cole, P. Hsu, and S.S.Sastry. Dynamic regrasping by coordinated control of sliding for multi-fingered hand. In *Robotics and Automation, IEEE International Conference on*, volume 2, pages 781 – 786, Scottsdale, AZ, USA, May 1989.
- [3] M. Armstrong and A. Zisserman. Robust Object Tracking. In *Proc. Asian Conf. Computer Vision*, 1995.
- [4] Olivier Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, Cambridge, MA, USA, 1993.
- [5] Qiang Li, Robert Haschke, Bram Bolder, and Helge Ritter. Grasp point optimization by online exploration of unknown object surface. In *12th IEEE-RAS Intl Conf on Humanoid Robots*, Osaka, Japan, 29/11/2012 2012.
- [6] Qiang Li, Robert Haschke, Helge Ritter, and Bram Bolder. Simulation results for manipulation of unknown objects in hand. In *Robotics and Biomimetics, IEEE International Conference on*, 2011.
- [7] Paul Michelman. Precision object manipulation with a multifingered robot hand. *Robotics and Automation, IEEE Transactions on*, 14(1):105–113, 1998.
- [8] M.Zribi, J. Chen, and M.S.Mahmoud. Control of multifingered robot hands with rolling and sliding contacts. *Journal of Intelligent and Robotic Systems*, v24(no.2), February 1999.
- [9] Youngmin Park, V. Lepetit, and Woontack Woo. Texture-less object tracking with online training using an rgb-d camera. In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, pages 121–126, Oct.
- [10] Robert J Platt Jr. *Learning and generalizing control-based grasping and manipulation skills*. PhD thesis, Citeseer, 2006.
- [11] R.M.Murray, Z.X.Li, and S.S.Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.
- [12] M Schopfer, Florian Schmidt, Michael Pardowitz, and Helge Ritter. Open source real-time control software for the kuka light weight robot. In *Intelligent Control and Automation (WCICA), 2010 8th World Congress on*, pages 444–449. IEEE, 2010.
- [13] Günter Schreiber, Andreas Stemmer, and Rainer Bischoff. The fast research interface for the kuka lightweight robot. In *IEEE Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications How to Modify and Enhance Commercial Controllers (ICRA 2010)*, 2010.
- [14] Carsten Schürmann, Risto Kõiva, and Robert Haschke. A modular high-speed tactile sensor for human manipulation research. In *IEEE World Haptics Conference (WHC 2011)*, Istanbul, Turkey, 21/06/2011 2011.
- [15] Kenji Suzuki, Isao Horiba, and Noboru Sugie. Linear-time connected-component labeling based on sequential local operations. *Comput. Vis. Image Underst.*, 89(1):1–23, January 2003.
- [16] K. Tahara, S. Arimoto, and M. Yoshida. Dynamic object manipulation using a virtual frame by a triple soft-fingered robotic hand. In *Robotics and Automation, IEEE International Conference on*, pages 4322 – 4327, 2010.
- [17] Kenji Tahara, Suguru Arimoto, and Morio Yoshida. Dynamic object manipulation using a virtual frame by a triple soft-fingered robotic hand. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4322–4327. IEEE, 2010.
- [18] Kenji Tahara, Keigo Maruta, Akihiro Kawamura, and Motoji Yamamoto. Externally sensorless dynamic regrasping and manipulation by a triple-fingered robotic hand with torsional fingertip joints. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3252–3257. IEEE, 2012.
- [19] Thomas Wimboeck, Christian Ott, and Gerhard Hirzinger. Passivity-based object-level impedance control for a multifingered hand. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 4621–4627. IEEE, 2006.
- [20] Yang Yang, Qixin Cao, Charles Lo, and Zhen Zhang. Pose estimation based on four coplanar point correspondences. In *Proc. of the 6th int. conference on Fuzzy systems and knowledge discovery - Volume 5*, pages 410–414, Tianjin, China, 2009. IEEE Press.