

A Visuo-Tactile Control Framework for Manipulation and Exploration of Unknown Objects

Qiang Li, Robert Haschke and Helge Ritter

Abstract—We present a novel hierarchical control framework that unifies our previous work on tactile-servoing with visual-servoing approaches to allow for robust manipulation and exploration of *unknown* objects, including – but not limited to – robust grasping, online grasp optimization, in-hand manipulation, and exploration of object surfaces.

The framework is divided into three layers: a joint-level layer, a tactile servoing layer, and a visual servoing layer. While the middle layer provides “blind” surface exploration skills, maintaining desired contact patterns, the visual layer monitors and controls the actual object pose providing high-level fingertip motion commands that are merged with the tactile-servoing control commands.

We illustrate the efficiency of the proposed framework using a series of manipulation actions performed with two KUKA LWR arms equipped with a tactile sensor array as a “sensitive fingertip”. The two considered objects are unknown to the robot, i.e. neither shape nor friction properties are available.

I. INTRODUCTION

Flexible interaction with objects, including grasping, exploration and manipulation, as we observe it in humans, is still a major challenge for today’s robots. A key factor for these skills is the highly developed integration of the visual, tactile, and force sensing channels when carrying out an action: this integration plays a major role to compensate for the numerous uncertainties involved in the mechanical interaction with an object when its properties (such as pose, shape, mass and friction) are not or only approximately known [3], [20], [8]. While the restricted availability of good tactile sensors has been for a long time a major impediment for a broader development of similar capabilities for robots, the accelerating progress in tactile sensing and its availability for robot manipulators is now increasingly opening up exciting possibilities for integrating touch and vision to enhance dexterous manipulation skills of robots.

The present contribution is focused on the augmentation of a purely tactile-driven servoing control loop in a previously developed system [13] by a second, visually driven control loop such that both modalities complement each other in a context-specific and suitably prioritized way. We exploit the capabilities of a previously developed sensor [18] and focus on two interaction phases which are at the core of many daily actions:

- 1) align, approach and establish contact with an object using touch and vision

- 2) adjustment and maintenance of the created contact for controlling the object (e.g. grasping, manipulation, and exploration)

Our approach follows the idea to shape the contributions of the participating control loops through suitably specified projection matrices, which are switched between phases. We investigate this as a main integration mechanism, along with suitably prioritized controllers, to unify the control of specialized interaction phases that are sequenced when manipulating an object: contact creation, contact maintenance, contact optimization w.r.t. a specific objective, grasping, manipulation, and finally object release. The present work is part of a larger endeavor towards obtaining a general framework for multimodal control of object manipulation.

Integrating vision and force feedback to improve robot skills has been in the focus of a number of previous authors. Two particularly important frameworks are the stack-of-tasks framework [14], and the iTaSC framework [4], which allow to specify tasks as constraints in different spaces which facilitates the combination of different sensor-based control modalities. Our work follows ideas of the control basis framework [7], which allowed the specification of basic control tasks and their subsequent prioritized combination.

Although there is a large body of work on force/torque-based robot control for manipulation and exploration of objects, these results cannot directly be applied to tactile-feedback based control because tactile sensors typically do not provide 6D wrench measurements, but instead an array of 1D normal force measurements. Depending on their spatial resolution, these tactile sensor arrays could be considered to provide “tactile images” of spatially distributed force interaction patterns [18], [6]. Tactile sensor arrays have been previously employed e.g. for object identification [22] and exploration [2], [19].

Prats et. al [15] point out that force control should be augmented with the tactile modality providing evidence from experiments that tactile information provides valuable information about local contact geometry (in addition to force alone) and that this may help considerably to improve the contact quality. They, however, only improve contact quality by sliding/twisting the fingertip on the object, thereby only exploiting three of the full six DOFs of the interaction. The tactile-servoing control framework [10] and a similar work [9] propose exploration of unknown object surfaces by fusing tactile proprioceptive feedback to estimate the contact position and contact force. Both works develop their control framework in a rather problem-specific way, either focusing on grasping and manipulation or on unknown environment

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

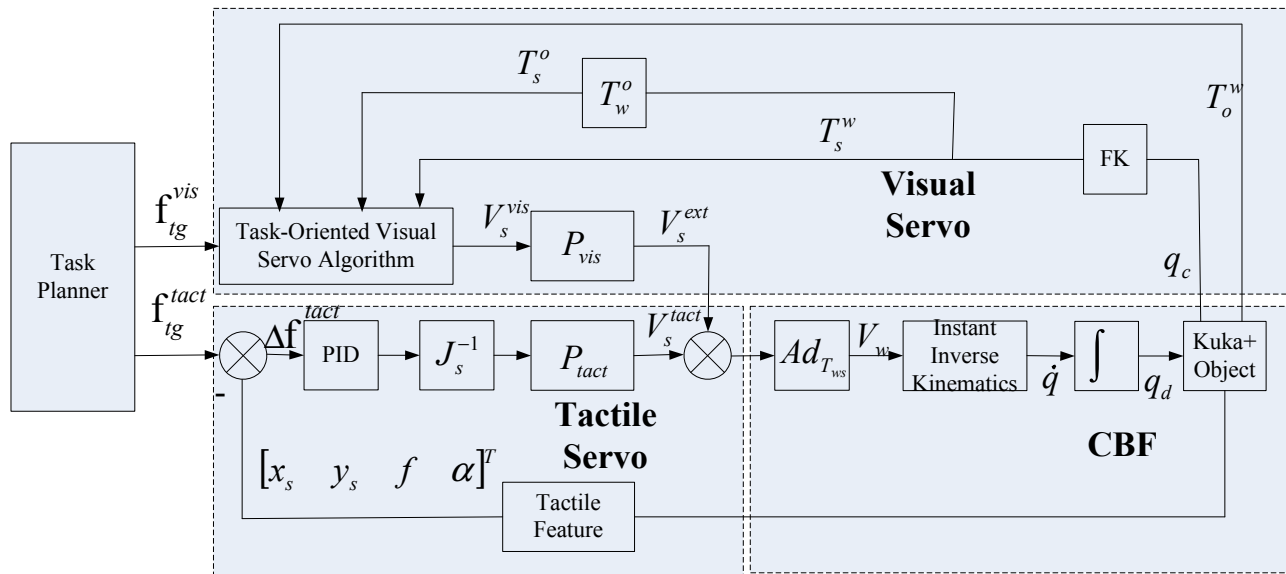


Fig. 1. Hierarchical controllers structure

exploration. The present paper attempts to present framework that allows to represent all these task in a unified fashion.

The paper is organized as follows: In the next section we introduce the three-layered control architecture composed from a joint-based controller, the tactile-servoing layer, and a visual-servoing layer. Next, we evaluate the framework on various bimanual manipulation tasks performed with two KUKA LWR robots, each equipped with a 16×16 tactile sensor array serving as “large fingertips” [18]. Finally, section V provides a conclusion.

II. HIERARCHICAL CONTROLLER STRUCTURE

The proposed approach augments our previous tactile-servoing control framework [13] that aimed at realizing sliding and rolling motions about the contact point while maintaining a specified normal contact force during manipulation. This was achieved by directly mapping errors in the tactile feature vector (contact position and force) to a suitable, error-reducing Cartesian velocity twist \mathbf{V}_s of the sensor frame. Subsequently the inverse velocity kinematics (using our implementation of the control basis framework CBF [17]) yields the appropriate joint velocities for the desired sensor frame motion. The present work extends this framework by complementing the tactile feedback loop with a visual-servoing layer that provides high-level fingertip motion commands that then become merged into the lower-level control flow. The overall controller structure is illustrated in Fig. 1 and will be explained in the following subsections, first summarizing our previous work on tactile servoing in Sec. II-A and subsequently focusing on the new visual-servoing branch in Sec. II-B. Fig. 2 depicts all involved coordinate frames (subscripts denoting the global world frame (w), the end-effector frame (e), the tactile sensor frame (s), and the object frame (o), along with l and r to distinguish the left and right hand when referencing the end-effector or sensor frames).

A. Tactile-servoing Controller

The tactile servoing control cycle (in the bottom part of Fig. 1) computes a vector $\Delta \mathbf{f}(t) = [\Delta x_s, \Delta y_s, \Delta f, \Delta \alpha]$ of tactile-feature errors (position errors of the contact blob centroid, force error, and angular error $\Delta \alpha$ of the desired contact blob orientation) from the raw sensor “tactile image”. They are fed into a PID-type controller to obtain a control vector \mathbf{u} . The key idea to utilize all 6 control DOF is achieved through computing a 6D velocity twist $\mathbf{V}_s^{\text{tact}}$ specifying an error-reducing motion twist expressed in the sensor frame. In its barest form, this can be computed by applying a fixed, task-independent, inverted sensor Jacobian J_s^{-1} :

$$\mathbf{V}_s^{\text{tact}} = J_s^{-1} \cdot \Delta \mathbf{f} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \Delta x_s \\ \Delta y_s \\ \Delta f \\ \Delta \alpha \end{pmatrix} \quad (1)$$

The particular form of J_s^{-1} arises from the contact geometry: positional deviations are compensated by sliding

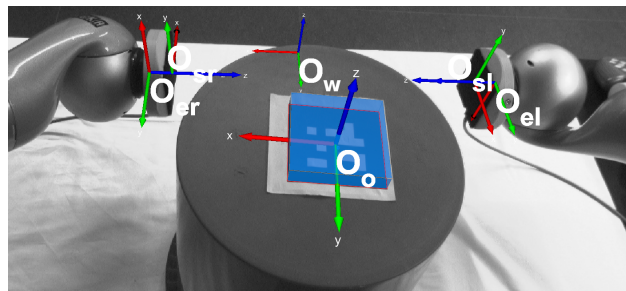


Fig. 2. Coordinate frames: $O_w, O_{el}, O_{er}, O_{sl}, O_{sr}, O_o$. The sensor frames are slightly displaced and turned by 90° relative to the end-effector frames. The world frame is located *on* the table, but rendered above the image.

motions along the same axes in the tangential (x - y) plane of the sensor (1st and 2nd row); however, a deviation along the x -axis can also be compensated by a rotation about the y -axis through the contact point, and vice versa (4th and 5th row). The orientation of a contact edge on the sensor can be adjusted by rotation about the z axis of the sensor frame (6th row). Normal force errors are corrected by a translational motion along the z -axis (3rd row), which is normal to the sensor plane, pointing towards the object.

This basic scheme is augmented with a task-dependent projector matrix P_{tact} that selects task-relevant motion components. Usually, P_{tact} is a simple 6×6 diagonal matrix, where ones and zeros are used to toggle individual twist components on and off. For example, if contact position control is desired, one will choose $P_{tact} = \text{diag}(1, 1, 0, 0, 0, 0)$. When additionally force control is required, the third diagonal entry should be set to one too. The 4th and 5th entries will enable rolling, and the 6th entry will enable twisting.

Preprocessing the vector of tactile feature deviations with a PID-type controller (with diagonal gain matrices $K_{P,D,I}$), Eq.1 finally becomes:

$$\mathbf{V}_s^{\text{tact}} = P_{tac} \cdot J_s^{-1} \cdot \left(K_P \cdot \Delta \mathbf{f}(t) + K_I \cdot \int \Delta \mathbf{f}(t) dt + K_D \cdot (\Delta \mathbf{f}(t) - \Delta \mathbf{f}(t-1)) \right). \quad (2)$$

Finally, the twists from the tactile feedback loop are fed into the low-level inverse kinematics module of the control basis framework. To this end, the twist \mathbf{V}_s expressed in terms of the sensor frame O_s will be transformed into the world frame O_w employing the adjoint matrix derived from the current forward kinematics $T_s^w = T_e^w \cdot T_s^e = (R_s^w, \mathbf{p}_s^w)$:

$$Ad_{T_{ws}} = \begin{pmatrix} R_s^w & \hat{\mathbf{p}}_s^w R_s^w \\ 0 & R_s^w \end{pmatrix} \quad (3)$$

For a more detailed account of the tactile-servoing control loop, we refer to [13].

B. Visual-servoing controller

The tactile-servoing control framework [13] already allowed for an external motion component $\mathbf{V}_s^{\text{ext}}$ to be fed in and merged with the tactile-based motion component $\mathbf{V}_s^{\text{tact}}$. Previously we used this input to generate a fixed forward motion to slide along an edge or to explore a two-dimensional surface. Here we use it to feed in a high-level motion component computed from visual object-pose feedback to realize object manipulation and grasping.

To this end we adapted and integrated our previous work on vision-based manipulation [10]. In that work we used a fiducial marker to track the object pose w.r.t. the world frame (assuming a calibrated, monocular, BW-camera). From the difference between the current and the desired object pose, we compute a desired object motion twist $\mathbf{V}_w^{\text{obj}}$ of the object relative to the world. The world frame representation $\mathbf{V}_w^{\text{obj}}$ can easily be transformed into the sensor frame to yield $\mathbf{V}_s^{\text{obj}}$ assuming non-slipping contacts.

For unification of both control frameworks we assume that a visual-servoing controller can always transform a

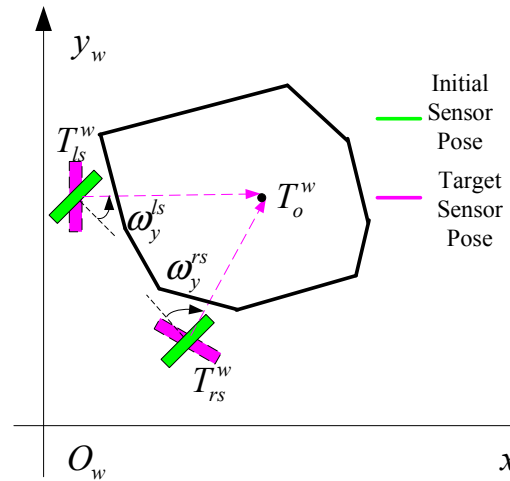


Fig. 3. Align and approach phase: The sensor normal is aligned to the approach vector towards the center of the object.

desired object motion twist into corresponding contact twists expressed in the sensor frame. As we will see in the following, for certain tasks, we can also compute the sensor frame motion directly in the sensor frame. The obtained contact twist, now expressed in the same coordinates as $\mathbf{V}_s^{\text{tact}}$, can be masked (again using a task-specific projector matrix P_{vis}) and finally added to the tactile motion component to yield an overall twist \mathbf{V}_s . The projector matrix P_{vis} typically selects motion components orthogonal to the ones of $\mathbf{V}_s^{\text{tact}}$ in order to allow for proper hybrid control. However, we will also consider parallel tactile and visual control, where contributions from both controllers are simultaneously employed along the same motion directions.

C. Visuo-tactile-servoing control scheme

In the subsequent subsections we will outline four exemplary visual-servoing control schemes that are representative examples of the two interaction phases described in the introduction section:

- 1) **Align and Approach** the fingertips to the object center and approach the object to establish proper contact,
- 2) **Maintain and Adjust** sliding fingers across the object surface in order to reach desired contact points or to improve grasp stability as demonstrated before [11],
- 3) **In-Hand Manipulation** of the object
- 4) **Exploration** of the object's surface and acquiring a tactile point cloud.

These actions are also shown in the accompanying video.

In all cases we assume, that no knowledge about the object is available: neither detailed shape models nor friction coefficients are known, such that we can deal with arbitrary unknown objects. We only assume, that the pose of the object can be tracked (in our case using a fiducial marker) and that we might extract coarse shape information, e.g. using a depth camera.

1) *Align and approach*: This phase aims at coarsely aligning the two tactile sensors towards the object center (which can be easily estimated from vision) and subsequently

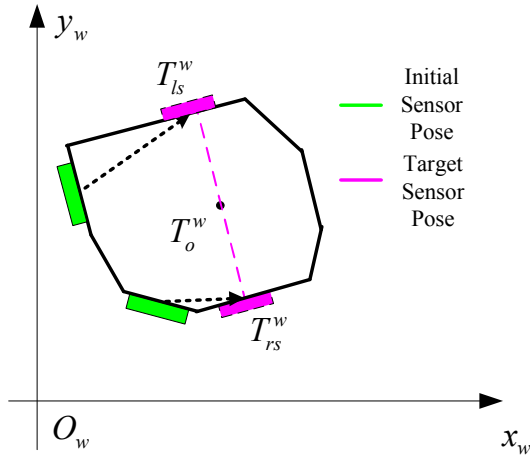


Fig. 4. Contact maintenance/adjustment phase: Slide fingers across the object’s surface to reach optimal contact points.

approaching the object as illustrated in Fig. 3. Obviously, the required motion twist is most easily represented in world frame. The linear velocity component is given by the direction vector “sensor \rightarrow object origin” multiplied by a gain k_p :

$$\mathbf{v}_w = k_p \cdot (\mathbf{p}_o^w - \mathbf{p}_s^w) \quad (4)$$

The angular velocity $\boldsymbol{\omega}_w$ is computed using quaternion slerp to move the sensor normal $R_s^w \cdot (0, 0, 1)^t$ towards the approach direction.

Projection matrices $P_{tact} = \mathbf{0}$ and $P_{vis} = \mathbf{1}$ ensure that only the visual-servoing controller will be active.

2) *Maintain and Adjust*: As soon as contacts are detected, the approach phase is finished and we switch from visual servoing to tactile servoing by setting the projection matrices $P_{vis} = \mathbf{0}$ and $P_{tact} = \text{diag}(0, 0, 1, 1, 1, 0)$. This choice allows to maintain a specified, small contact force and will actively roll the tactile sensor to improve the alignment with the object surface. Then we aim to stabilize the grasp by sliding the fingers across the object’s surface reaching better contact points. We have shown previously, that we can autonomously find and follow the gradient direction of an arbitrarily chosen cost function, e.g. maximizing grasp stability and manipulability [11]. Here, we assume that the desired contact points are available on the object which we should servo to. Hence, in this phase, the tactile servoing controller will be in charge of maintaining the contact, reactively rolling on the surface of the object and maintaining the contact force. Hence the corresponding task projector equals $P_{tact} = \text{diag}(0, 0, 1, 1, 1, 0)$. On the other hand, the visual-servoing controller will generate the sliding motion on the object surface employing the complementary projector $P_{vis} = \text{diag}(1, 1, 0, 0, 0, 0)$. The linear sensor velocity is computed as follows:

$$\mathbf{v}_s = R_w^s \cdot (R_o^w \cdot \mathbf{p}_c^o - R_s^w \cdot \mathbf{p}_c^s), \quad (5)$$

where \mathbf{p}_c^o is the desired contact position described in the object frame, \mathbf{p}_c^s is the current contact position in the sensor

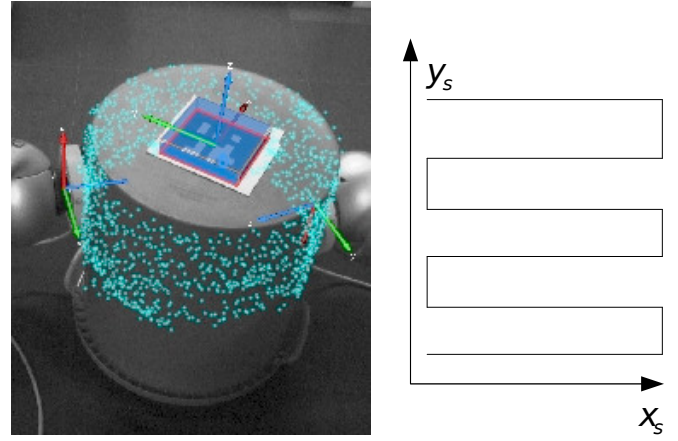


Fig. 5. Point cloud acquisition by superimposing an S-shaped motion trajectory with tactile-servoing commands to maintain optimal contact.

frame, and R_s^w and R_o^w are the sensor and object frames respectively, both given w.r.t. the world frame.

3) *In-Hand Manipulation*: If friction properties and joint torques are not available, we cannot actively control rolling and slipping, because internal forces cannot be designed. However, as we have shown, short-range object manipulation is possible without explicitly designing all details of the physical hand-object interaction [10]. In order to calculate finger tip motions to realize a desired object motion T_o^o without knowledge about the exact object-finger geometry, we made the assumption that contact positions \mathbf{p}_c^o do not move relative to the object during a control cycle. Under this essential assumption, we could calculate the new contact positions $\mathbf{p}_c^{w'}$ in the world frame as follows:

$$\mathbf{p}_c^{w'} = T_o^w \cdot T_o^o \cdot \mathbf{p}_c^o. \quad (6)$$

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 probably occur. However, the sensor feedback available in the next control cycle will allow us to recognize and correct this undesired contact motion.

Having the tactile-servoing framework available, we can replace the complex contact-force planning approach of [10] with simple tactile control primitives to maintain contact position and force as described in the previous subsection, i.e. employing the task projector $P_{tact} = \text{diag}(0, 0, 1, 1, 1, 0)$. To allow full 6-DoF object control, the visual projector equals the identity, $P_{vis} = \mathbf{1}$. With this choice, the projectors are not complementary anymore but overlap. Hence, in this case we use parallel position and force control, superimposing velocity control commands from both tactile and visual servoing. This is necessary to allow object motion along the contact normal direction as well as to compensate drift due to poorly calibrated force sensors.

4) *Exploration for Tactile Point Cloud Acquisition*: In our previous work [13] we used the tactile servoing controller to explore an unknown object surface along a one-dimensional

manifold by superimposing the tactile-servoing commands with a fixed forward motion command. Here we extend this work to palpate the complete two-dimensional object surface and collect acquired contact points within a single registered point cloud. To this end, we generate an S-shaped motion profile as shown in Fig. 5 to be superimposed on the tactile-based motion command. This motion profile generates a tangential sliding motion of the sensor, while the tactile-servoing controller is again maintaining contact using the well-known projector matrix $P_{tact} = \text{diag}(0, 0, 1, 1, 1, 0)$. Visual-servoing is replaced by the hard-coded motion profile. In order to register all acquired contact points within a single point cloud, we transform all contact points \mathbf{p}_c^s initially acquired with respect to the sensor frame to the object frame utilizing the known forward kinematics of the robot and the visually observed object pose T_o^w :

$$\mathbf{p}_c^o = T_o^w \cdot T_w^s \cdot \mathbf{p}_c^s. \quad (7)$$

III. EXPERIMENTAL EVALUATION

The experimental setup comprises two KUKA LWR arms, each equipped with a 16×16 tactile sensor array providing up to 1.9 kHz frame rate and a nominal spatial resolution of 5mm. Averaging the contact position over several taxels, the spatial accuracy can be improved by a magnitude [13]. The tactile sensor is only coarsely force-calibrated using the method proposed in [13], because the sensor characteristics is highly taxel-specific and changes over time due to wear as well as temperature fluctuations. However, as we show in the following, the parallel position and force controller employed during manipulation can successfully compensate poor force measurements that typically would induce drift, i.e. pushing the object into a certain direction.

The monocular camera was calibrated applying standard calibration methods using a known 3D calibration object [5]. The BCH-code-based marker provides 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 [23]. The fiducial marker estimation accuracy can be found in [10]. A monocular camera observes the scene from the top, estimating the object pose from the fiducial marker at a maximal frame rate of 30 Hz. The robot control cycle is fixed to 250 Hz (using the KUKA FRI interface) and in sync with the reduced tactile feedback rate. Individually, tactile and visual feedback are smoothed over time using a windowed averaging filter of 20 frames width. We point out that we use only the joint encoder feedback, but not the torque feedback provided by the KUKA arms. PID controller gains are manually tuned to guarantee the stability of controllers in all experiments.

The accompanying video shows the same action sequence of approaching, grasping and manipulating, performed on two different objects, a straight cylinder and a tapered prism. Both objects have different, un-modeled friction properties. While the cylinder yields a smooth surface, the prism exhibits flat surfaces with sharp edges.

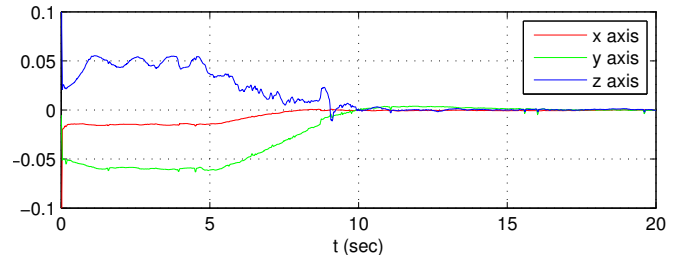


Fig. 6. Contact error evolution (in object frame) while establishing contact and sliding towards the final contact point of the right hand.

1) *Vision-guided contact creation*: In the first phase we establish contact with the object according to the strategy outlined in Sec. II-C.1: The robot aligns and moves its finger tips / tactile sensors towards the object center, which we assume to be located 20cm below the recognized marker position. This approaching motion is stopped as soon as contact to the object is detected by the tactile sensor. Then the visual-servoing controller will be switched to the tactile-servoing controller that attempts to increase contact area by a rolling motion while maintaining a small contact force to not kick off the object.

2) *Grasping and in-hand manipulation*: Because the object shape is unknown a-priori, the initial contact can be anywhere on the object surface. Before we attempt to increase the contact force to perform a stable grasp, the contact location will be optimized using the sliding strategy outlined in Sec. II-C.2. Error convergence towards the final grasp points on the cylinder is shown for the two initial phases in Fig. 6. The graphs start as soon as contact is established with the object. During about the first 5secs, the robot aligns the sensor surface with the object surface. Afterwards the sliding motion towards the final grasp point begins – showing smooth convergence.

In order to finally grasp and lift the object, we increase the contact forces. Subsequently the object is moved about 10cm along the world's z , x , and y axes in sequence. The resulting trajectories for force and positional errors are shown in Fig. 7. As can be seen from the deflections in the bottom subfigure (positional errors), a new target pose was set after 8, 12, and 17 seconds. In all cases the positional error quickly decays to the noise level. The two top subfigures show the fingers' contact force evolution and the desired contact force. As can be seen from the force graphs, the left-hand sensor underestimates the contact force while the right-hand sensor overestimates. As there is no object drift (enforced by the visual-servoing controller) both contact forces actually balance out. Further, we notice that contact forces suddenly change as soon as a new object pose is commanded.

Fig. 8 shows similar force and error tracking trajectories when rotating the object about the world's z and y -axes. Again, 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. Combining the demonstrated in-hand manipulation skills with repeated regrasping, we can also

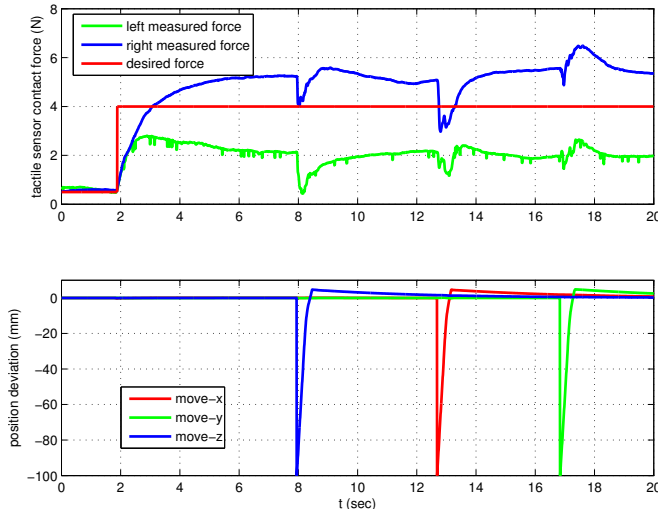


Fig. 7. Evolution of force (f_l , f_r of both tactile sensors / fingertips) and positional errors during translating motion. x, y, z errors are mapped to red, green, blue colors resp.

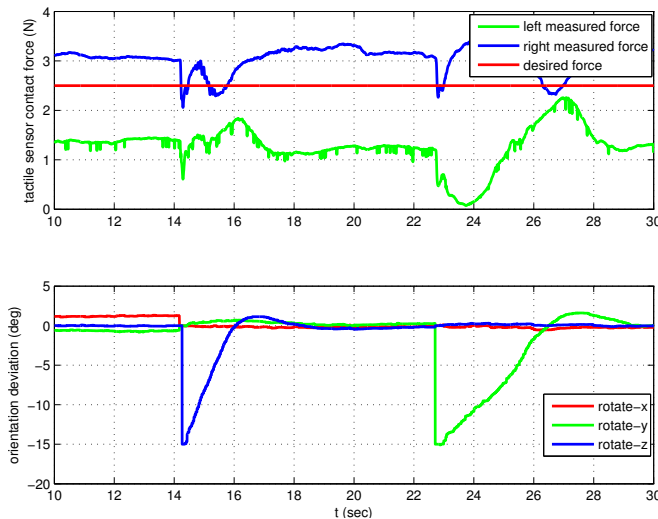


Fig. 8. Evolution of force and rotational errors during object rotations. The force tracking performance reflects the framework’s lower prioritization of force as compared to position [cf. text]

realize large-scale object manipulation as shown in the accompanying video.

3) Object Exploration Acquiring a Tactile Point Cloud:

For the acquisition of a tactile point cloud, the robot fingers palpate the object with the S-shaped motion trajectory shown in Fig. 5. To record a complete surface model of the object, the object needs to be rotated (due to the limited workspace of the robot). This action is performed manually in the video (rotating the cylindrical object around its axis) to speedup the whole process. Alternatively we could – of course – have applied the large-scale manipulation action alternating to the palpation process to fully automate the process.

The palpation of the tapered prism requires some manual intervention to stabilize the rather light object on the table. This is necessary because the transition between object faces across the sharp edge creates rather large interaction

forces. Nevertheless the control algorithm can handle the discontinuous transition between the discrete faces.

IV. DISCUSSION

As we pointed out in the discussion of the grasping and manipulation experiments, the force calibration of the piezo-resistive sensor is too coarse to allow force-only feedback control for grasping. Deviations in measured 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 the drifting issue, employing a parallel position and force controller. However, even if visual object pose feedback is not available (e.g. due to occlusion), we can compensate for drifts using proprioceptive feedback: In this case the object position would be estimated as the center of gravity of estimated contact or end-effector locations that should not drift either when we aim to stably hold the object.

As detailed in [12], the composite controller computes the final control signal by superimposing the control signals from both sub controllers, position and force. 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 scenario, controlling the pose of the object is more important, which therefore uses a PI-type controller. In contrast, force control is using a P-controller. This, on the other hand contributes to the poor force tracking results visible in Figs 8. Although we cannot completely eliminate the static force error, it is possible to reduce and control this deviation by tuning the P-controller parameters. In this paper, we demonstrated and quantitatively verified that the proposed controller framework can realize accurate manipulation of the object pose while stably holding the object using the manually tuned parameters.

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 steady state errors. We used the same parameter sets for both arms.

V. CONCLUSION

In this paper, we proposed a visuo-tactile servoing control framework to realize a comprehensive set of visuo-tactile interaction primitives on unknown objects, ranging from aligned approach to grasping, optimal object contact and in-hand manipulation to finally surface exploration. Common to all actions is a tight feedback loop maintaining optimal object

contact using tactile-servoing controllers. On top of that, higher-level motion commands are fed in to realize visual-servoing or surface exploration. Utilizing two task-dependent projection matrices we can flexibly adapt the control system to individual task requirements. The accompanying video illustrates typical control capabilities that can be realized by this simple yet powerful approach.

ACKNOWLEDGMENT

This work was funded by SPP priority program “Autonomous Learning” (SPP 1527-1) funded by DFG. Authors would like to thank Christof Elbrechter for rendering the tactile point clouds using the ICL vision library.

REFERENCES

- [1] Johan Baeten, Herman Bruyninckx, and Joris De Schutter. Shared control in hybrid vision/force robotic servoing using the task frame. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2128–2133. IEEE, 2002.
- [2] A. Bierbaum, M. Rambow, T. Asfour, and R. Dillmann. A potential field approach to dexterous tactile exploration of unknown objects. In *Humanoid Robots, 8th IEEE-RAS International Conference on*, 2008.
- [3] J. Bimbo, L. Seneviratne, K. Althoefer, and H. Liu. Combining touch and vision for the estimation of an object’s pose during manipulation. In *Proc. IROS*, pages 4021–4026, 2013.
- [4] Joris De Schutter, Tinne De Laet, Johan Rutgeerts, Wilm Decré, Ruben Smits, Erwin Aertbeliën, Kasper Claes, and Herman Bruyninckx. Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. *The International Journal of Robotics Research*, 26(5):433–455, 2007.
- [5] Olivier Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, Cambridge, MA, USA, 1993.
- [6] Jeremy A. Fishel. *Design and use of a biomimetic tactile microvibration sensor with human-like sensitivity and its application in texture discrimination using Bayesian exploration*. PhD thesis, University of Southern California, 2012.
- [7] S. Hart, S. Sen, S. Ou, and R. Grupen. The control basis API – a layered software architecture for autonomous robot learning. In *2009 Workshop on Software Development and Integration in Robotics at ICRA*, 2009.
- [8] Paul Hebert, Nicolas Hudson, Jeremy Ma, and Joel Burdick. Fusion of stereo vision, force-torque, and joint sensors for estimation of in-hand object location. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5935–5941. IEEE, 2011.
- [9] C. Jara, J. Pomares, F. Candelas, and F. Torres. Control framework for dexterous manipulation using dynamic visual servoing and tactile sensors feedback. *Sensors*, 14(1):1787–1804, 2014.
- [10] Q. Li, Ch. Elbrechter, R. Haschke, and H. Ritter. Integrating vision, haptics and proprioception into a feedback controller for in-hand manipulation of unknown objects. In *Proc. IROS*, Tokyo, 2013.
- [11] Q. Li, R. Haschke, B. Bolder, and H. Ritter. Grasp point optimization by online exploration of unknown object surface. In *Proc. Humanoids*, Osaka, 2012.
- [12] Q. Li, R. Haschke, H. Ritter, and B. Bolder. Simulation results for manipulation of unknown objects in hand. In *Proc. ROBIO*, 2011.
- [13] Q. Li, C. Schürmann, R. Haschke, and H. Ritter. A control framework for tactile servoing. In *Proc. RSS*, 2013.
- [14] Nicolas Mansard, Olivier Stasse, Paul Evrard, and Abderrahmane Kheddar. A versatile generalized inverted kinematics implementation for collaborative humanoid robots: The stack of tasks. In *Advanced Robotics, 2009. ICAR 2009. International Conference on*, 2009.
- [15] M. Prats, P. Sanz, and A. del Pobil. Vision-tactile-force integration and robot physical interaction. In *Proc. ICRA*, 2009.
- [16] Alexander Schmitz, Marco Maggiali, Lorenzo Natale, Bruno Bonino, and Giorgio Metta. A tactile sensor for the fingertips of the humanoid robot icub. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2212–2217. IEEE, 2010.
- [17] M Schöpfer, F. Schmidt, M. Pardowitz, and H. Ritter. Open source real-time control software for the kuka light weight robot. In *8th World Congress on Intelligent Control and Automation (WCICA)*, 2010.
- [18] C. Schürmann, R. Kõiva, and R. Haschke. A modular high-speed tactile sensor for human manipulation research. In *World Haptics Conference*, 2011.
- [19] N. Sommer, Miao Li, and A. Billard. Bimanual compliant tactile exploration for grasping unknown objects. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 6400–6407, May 2014.
- [20] Xiaojing Song, Hongbin Liu, K. Althoefer, T. Nanayakkara, and L.D. Seneviratne. Efficient break-away friction ratio and slip prediction based on haptic surface exploration. *Robotics, IEEE Transactions on*, 30(1):203–219, Feb 2014.
- [21] C. Strub, F. Worgotter, H. Ritter, and Y. Sandamirskaya. Correcting pose estimates during tactile exploration of object shape: a neuro-robotic study. In *Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2014 Joint IEEE International Conferences on*, 2014.
- [22] Danfei Xu, G.E. Loeb, and J.A. Fishel. Tactile identification of objects using bayesian exploration. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013.
- [23] 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.