A control framework for tactile servoing

Qiang Li, Carsten Schürmann, Robert Haschke, Helge Ritter
CITEC, Bielefeld University, Germany

***Abstract*—The advent of sensor arrays providing tactile feedback with high spatial and temporal resolution asks for new control strategies to exploit this important and valuable sensory channel for grasping and manipulation tasks.**

In this paper, we introduce a control framework to realize a whole set of tactile servoing tasks, i.e. control tasks that intend to realize a specific tactile interaction pattern. This includes such simple tasks like tracking a touched object, maintaining both contact location and contact force, as well as more elaborate tasks like tracking an object's pose or tactile object exploration.

Exploiting methods known from image processing, we introduce robust feature extraction methods to estimate the 2D contact position, the contact force, and the orientation of an object edge being in contact to the sensor. The flexible control framework allows us to adapt the PID-type controller to a large range of different tasks by specification of a projection matrix toggling certain control components on and off. We demonstrate and evaluate the capabilities of the proposed control framework in a series of experiments employing a 16×16 tactile sensor array attached to a Kuka LWR as a large fingertip.

I. Introduction

The sense of touch allows humans and other animals to perform coordinated and efficient interactions within their environment. An early experiment demonstrated the importance of tactile feedback for manual interactions: without the sense of touch, subjects had severe difficulties maintaining a stable grasp or performing a complex action [11, 10]. In recent years, the resolution and sensitivity of tactile sensors only sufficed for basic force feedback during blind grasping [4]. However, tactile sensor arrays providing high spatial and temporal resolution as well as high sensitivity [8, 21] emerged recently that will allow for more advanced control methods involving tactile feedback.

Such control approaches – which we denote as tactile servoing in accordance to corresponding control approaches involving direct visual feedback – require advanced tactile perception methods and their integration into control programs for direct robot control. Tactile servoing includes important tasks like sliding a finger tip across an object's surface, following specific surface structures like ridges, searching for distinctive tactile patterns, or exploring the object shape by groping. Most of these tasks are essential for both in-hand object manipulation [13], and haptic object identification [16].

Drawing on ideas for visual servoing and applying image processing algorithms to the tactile force image provided by modern tactile sensor arrays, it is possible to extract basic tactile features in real time and employ them for robot control. The challenging mission is to find generic features, which not only work in specific hard-coded control scenarios on a specific type of tactile sensor, but that generalize to a rich set of control tasks and sensor types. Raibert's [17] hybrid position and force control approach provides the basis to explore unknown environments and we will draw on his ideas.

In this paper, we propose a unified and extensible tactile servoing framework that can cover many tactile exploration tasks. The control approach can be easily adapted to various tasks by defining a projection matrix that determines which of the proposed control primitives should be utilized. Addition of further control primitives is easily possible, when new tactile features are developed. The proposed framework easily integrates with classical robot velocity and position control, thus facilitating integration of further feedback modalities like vision or proprioceptive sensing. In summary, the proposed tactile servoing framework is a practical toolkit to realize complex manipulation and exploration tasks as discussed in [22, 5, 1].

The remaining paper is organized as follows: In sec. II we present the state of the art of tactile-feedback-based control. In sec. III we propose some basic tactile features extracted from the tactile sensor image (contact position, object orientation and contact force) that are utilized in the tactile servoing framework introduced in sec. IV. Finally, in sec. V we evaluate the control results in a set of five different servoing tasks, which are also shown in the accompanying video [12].

II. State of the art

3D object exploration based on tactile feedback is an important research issue in the robotics community because this sensor modality provides more accurate position information than visual sensors and it is the most important sensor modality to reliably assess object properties like roughness, softness, friction, or mass [25].

The research question most related to our work is how to design the exploration process to drive a tactile-sensitive fingertip across an object's surface. In the context of unknown object recognition based on haptic (i.e. tactile and proprioceptive) feedback, Dario et. al [5] proposed a multi-stage recognition approach. The authors divided the whole exploration and recognition process into five stages: approach, shape, texture, hardness and thermal. They claimed that every stage can be realized by a specific subroutine. Allen et. al [1] recognize the object by manipulating it. They propose a hierachical recognition process: first, they grasp the object with a multi-fingered robot hand in order to roughly estimate the shape and pose of the object. Subsequently, the fingers can be employed to explore the object's surface. Although their work did not focus on tactile servoing, they derive some task requirements for a tactile servoing controller: the ability to track linear features, to explore an unknown object's surface and to track a given contact location.

Considering the sensor feedback, a tactile sensor image can provide rapid and accurate information to the robot control system. For example, [1] and [8] claimed that they can extract the contact point location, object orientation and contact force information by analyzing the image moments in real time, i.e. within the control cycle of the robot. Another approach to extract the orientation of an object edge w.r.t. the sensor surface is to apply a Hough transform to fit a line to the sensor image [15, 2] and subsequently compute the orientation of this line w.r.t. the sensor's coordinate frame.

Chen [3] proposed to employ elasticity theory to derive contact models between several geometric object primitives and the planar tactile sensor surface. Their approach can also model the relation of the contact geometry and expected tactile image moments. Because their method is based on a physical model, there are many constraints to obey for their approach. We follow the approach proposed by Ho et. al [8], using image moments to derive the contact geometry, in order to extract tactile image features.

Zhang et. al [26] extended the work of Chen [3] and proposed a first tactile servoing framework. They draw on elasticity theory to formulate a tactile sensor model and the inverse model. With this theory, it is possible to predict the sensor image from the known contact geometry. Their closed-form solution to compute tactile features is highly efficient. We extend their work in the following two aspects: First, we formulate a flexible set of controller primitives, which can be combined to realize a multitude of various tactile servoing tasks. Second, our framework integrates classical end-effector servoing and tactile servoing, thus providing an easy-to-use interface to compose complex control strategies to realize grasping, manipulation and object exploration tasks.

III. Feature extraction from tactile images

Many tactile sensor designs propose an array of tactile sensing elements (tactels) providing only *normal force* information [21, 18, 24] for each element. However, sometimes it is also possible to compute contact *force directions* from this information [24]. Most array structures also have a reasonable spatial resolution to allow for an explicit control of the tactile force pattern sensed in a contact region. As a consequence, in our control framework, we assume the availability of a tactile sensor array providing a tactile image of force values measured by individual tactels.

The sensor employed in our experiments provides an array of 16×16 tactels with a spacing of 5mm in each direction. It is tuned towards high frame rates (up to 1.9 kHz), rendering a use for real-time robot control feasible [21]. This sensor exploits the piezo-resistive sensing principle, measuring changes in resistance of a conductive foam due to an applied force. The analog measurement of each individual tactel is converted to a 12bit digital value covering a pressure range of 0.1–10 kPa.[1] Due to varying local conductive properties of the foam, every tactel has a distinguished, squashed and noisy sensor characteristics as shown in Fig. 1. To obtain a coarse force calibration, we inverted the characteristic measurement function of each individual tactel in its linear range.[1]

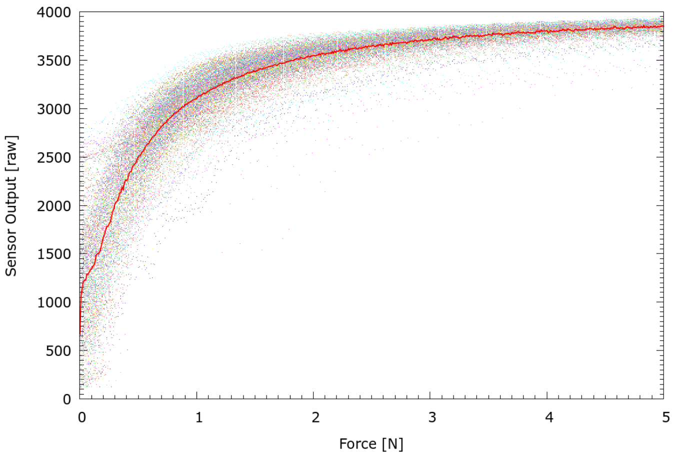


Fig. 1: Sensor characteristics of all 256 tactels (and an individual one – red solid line) as acquired using a 3-axis calibration bench and an industrial strain gauge providing ground-truth force measurements with an accuracy of ±2.5mN.

The intended tactile servoing tasks aim for controlling a) the contact position on the fingertip, b) the contact force, and c) the orientation of an object edge relative to the sensor array. To this end, we need feature extraction methods providing the current value of these control variables.

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 [23], 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. The binarization threshold was chosen rather small, just above the noise level to consider as much tactile information as possible.

Subsequently, we compute the overall contact (normal) force f as the sum of forces f_{ij} within the contact region and the contact position $\mathbf{c}$ as the force-weighted center of pressure (COP) of R:

$$f = \sum_{ij \in R} f_{ij} \qquad \mathbf{c} = f^{-1} \sum_{ij \in R} f_{ij} \mathbf{c}_{ij} \,, \tag{1}$$

where 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. 2.

[1] The sensor's sensitivity and force range can be adjusted to the task. Here, we have chosen the characteristics to provide a linear range from 0.1–1 kPa.

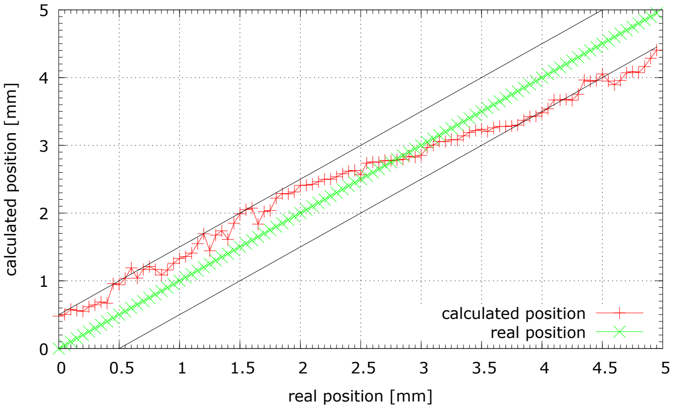


Fig. 2: Computed contact position (COP) when moving a 2mm-diameter probe tip from one to the other end of a single tactel. Due to the weighted averaging between different tactel locations (Eq. 1), we yield a spatial accuracy of ca. 0.5mm despite the much larger tactel size of 5mm. Note that, due to the specific shape of the sensor electrode, there is a small bias towards the center of the tactel (located at 2.5mm).

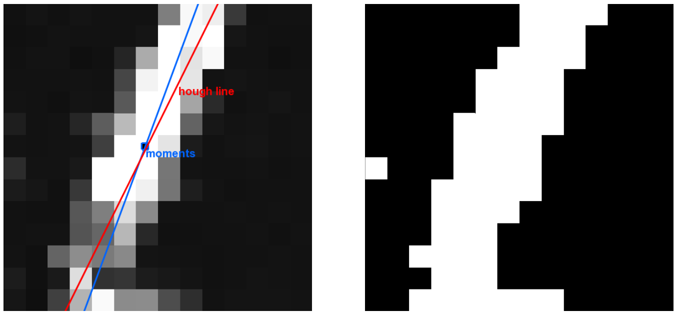


Fig. 3: Raw 16×16 tactile image (left) and extracted contact region (right) when pressing a pen onto the sensor area. The extracted line feature is visualized as well (green: Hough transform, red: image moment analysis).

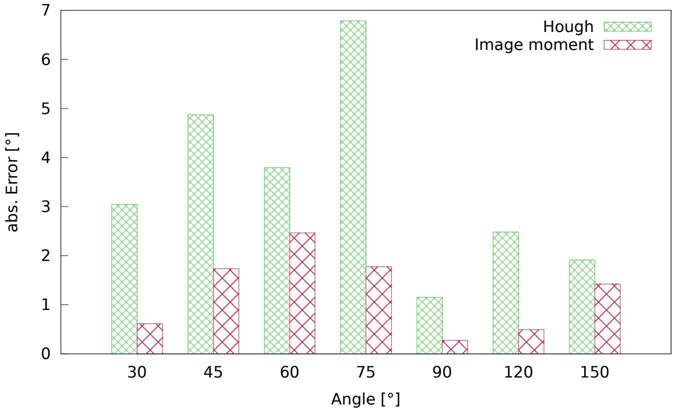


Fig. 4: Comparison of the test results for the Hough and image moment method. It can be seen that the image moment produced a lower error for the estimation of line angle than the Hough method.

Usually, we want to control the contact pressure instead of the overall contact force. Considering manipulation of fragile objects, like an egg, it is the local pressure that should be limited to not damage the object. To obtain a pressure value, we normalize the overall measured force by the size of the contact region (measured as the number of pixels in R):

$$p = \frac{f}{|R|} \tag{2}$$

In our experiments we target a pressure value of p=1, which roughly corresponds to 2 kPa or a force of 2 N spread over an area of 40 pixels.

For extraction of the orientation angle of a line-shaped contact region originating from an object edge (cf. Fig. 3) we compared two methods: a) line detection with Hough transform [6] and b) image moment analysis [8]. While the former method fits a line segment into a gray-scale image, the latter one extracts the major principal component of the contact region. In order to quantitatively compare the accuracy of both methods for our application, we pressed a pen-like object onto the sensor surface, thus stimulating a line-shaped contact region. A comparison of the orientation angle computed with the two mentioned methods to the value actually measured with a protractor at various object poses is shown in Fig. 4 revealing a smaller error for the PCA approach.

All image processing was realized employing the Image Component Library [7], providing real-time performance for all required methods.

IV. Tactile Servoing Framework

Our control framework for tactile servoing builds on top of the control basis framework (CBF) originally developed in the lab of Prof. Grupen [9]. This framework realizes resolved motion rate control, mapping updates of task control variables to joint angle updates of the robot. The main idea of the tactile servoing framework proposed in this paper is to define an inverted task Jacobian J_s^{-1} mapping errors in the tactile feature vector to a suitable Cartesian velocity twist $\mathbf{V}_s$ of the sensor frame. Subsequently we employ the powerful motion control framework of CBF [20] to realize the computed sensor frame motion with appropriate joint motions. A schematic scenario setup with all involved coordinate frames is depicted in Fig. 5. O_g denotes the global coordinate frame that is fixed in the robot base, O_e denotes the end-effector frame, and O_s denotes the sensor frame that is fixed relative to the tactile sensor array shown in blue color.

The proposed tactile servoing control framework aims for realizing sliding and rolling motion about the contact point while maintaining a specified normal contact force during manipulation. However, dependent on the actual task at hand, specific motion components can be explicitly switched off. Additionally to this tactile-driven robot motion, an external task planner can provide a motion component $\mathbf{V}_s^{\text{ext}}$, which is again a twist expressed in terms of the sensor frame O_s. This motion component allows to realize externally controlled

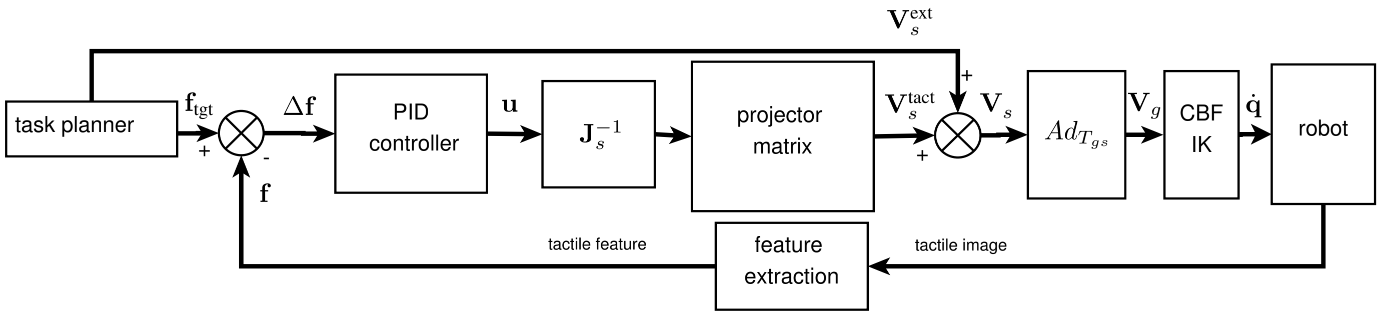


Fig. 6: Control scheme for tactile servoing: The core feedback part computes a sensor motion $\mathbf{V}_s^{\text{tact}}$ from tactile-feature deviations $\Delta\mathbf{f}$, which is superimposed with an external motion signal $\mathbf{V}_s^{\text{ext}}$ and subsequently fed into the inverse kinematics module.

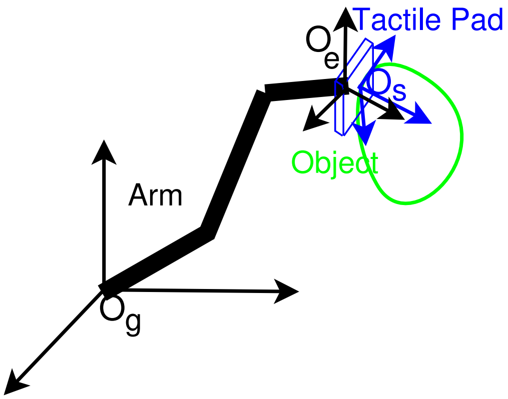


Fig. 5: Definition of coordinate frames

tactile object exploration, e.g. to follow an object edge or to run the sensor over the whole object surface as detailed in the experimental section V.

Please note, that – as a side-effect – the robot's forward kinematics $T_{gs} = T_{ge} \cdot T_{es}$ mapping points from the sensor to the global coordinate frame, can be used to determine the spatial location of a contact region w.r.t. the global frame. In this sense, the robot can act like a *ruler* to measure the path length while sliding over an object's surface.

The overall control scheme of our proposed controller is depicted in Fig. 6. The control cycle starts by computing the deviation of the current tactile feature vector $\mathbf{f}$ from the targeted one. This error is fed into PID-type controllers, acting independently on all feature-error components. The resulting control variable $\mathbf{u}$ is a (non-uniformly) scaled version of $\Delta\mathbf{f}$. Please notice, that for effective force control a non-zero integral component is required to compensate for static errors caused by a pure P-controller. Additionally, the derivative component is necessary to suppress undesired oscillations.

Subsequently, we compute a sensor motion $\mathbf{V}_s^{\text{tact}}$ aiming to reduce the feature error. This is realized with a fixed, task-*independent*, inverted Jacobian matrix J_s^{-1} and subsequent application of a task-*dependent* projector matrix P selecting certain twist-components for control and neglecting others. P is a simple 6×6 diagonal matrix, where ones and zeros indicate, that the corresponding component is or is not used for control. Summarizing the feedback-part of the tactile servoing controller is determined by the following equation:

$$\mathbf{V}_s^{\text{tact}} = P \cdot J_s^{-1} \cdot \Big(K_P \cdot \Delta\mathbf{f}(t) + K_I \cdot \int \Delta\mathbf{f}(t)dt \\ + K_D \cdot (\Delta\mathbf{f}(t) - \Delta\mathbf{f}(t-1)) \Big) \tag{3}$$

Here $\mathbf{V}_s = [\mathbf{v}_s, \boldsymbol{\omega}_s]$ denotes the 6-dimensional twist vector composed of linear and angular velocity components $\mathbf{v}_s$, $\boldsymbol{\omega}_s$. $K_{P,I,D}$ denote diagonal matrices of PID-controller gains and $\Delta\mathbf{f}(t) = [\Delta x_s, \Delta y_s, \Delta f, \Delta\alpha]$ denotes the deviation of the feature vector composed of the positional error $\Delta x_s, \Delta y_s$, the normal force error Δf, and the angular error $\Delta\alpha$ of the line orientation. Note, that the latter one is measured modulo π in order to obtain angular errors in the range $(-\frac{\pi}{2}, \frac{\pi}{2}]$ and thus circumventing singularities due to their circular nature. The rotational symmetry allows to restrict the errors to this range instead of $(-\pi, \pi]$.

Finally, the twists originating from the tactile feedback-loop and the external task planner are superimposed and fed to the 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 needs to be transformed to the global frame O_g, which is realized by the adjoint matrix derived from the current forward kinematics $T_{gs} = T_{ge} \cdot T_{es} = (R_{gs}, \mathbf{p}_{gs})$:

$$Ad_{T_{gs}} = \begin{pmatrix} R_{gs} & \hat{\mathbf{p}}_{gs} R_{gs} \\ 0 & R_{gs} \end{pmatrix} \tag{4}$$

At the core of the tactile-feedback controller is the inverse Jacobian that maps feature deviations into a motion twist of the tactile sensor array:

$$\mathbf{V}_s^{\text{tact}} = J_s^{-1} \cdot \Delta\mathbf{f} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ 0 & 1 & & \\ 1 & 0 & & \\ & & & 1 \end{pmatrix} \cdot \begin{pmatrix} \Delta x_s \\ \Delta y_s \\ \Delta f \\ \Delta\alpha \end{pmatrix} \tag{5}$$

This matrix can be easily determined in the sensor coordinate frame O_s: Positional deviations are simply mapped into corrective tangential motions in the x-y-plane of the sensor.

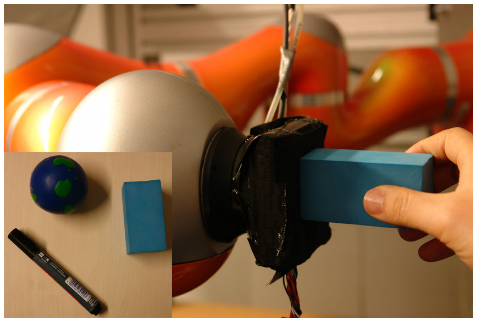


Fig. 7: Experimental setup: tactile sensor mounted on LWR.

Normal force errors are mapped into a corrective translational motion along the z-axis of the sensor frame, which is normal to the sensor plane, pointing towards the object. These linear motion components are determined by the first three rows of J_s^{-1}. The rotational error $\Delta\alpha$ is mapped onto a rotational velocity around the z-axis (last row). The motion components corresponding to the fourth and fifth row of the inverted Jacobian realize a rolling motion of the sensor. These are triggered by positional deviations again. Thus, an error Δx_s is not only reduced by an appropriate tangential linear motion of the sensor, but also by a rolling motion around the y-axis of the sensor, thus also moving the COP of the contact region closer towards its target location.

The task-dependent projector matrices P can be used to toggle these individual twist components on and off. For example, if contact position control is desired, one will choose $P = \text{diag}(1, 1, 0,\ 0, 0, 0)$. When additionally force control is required, the third diagonal entry should be set to 1 too. In order to enable or disable the orientation tracking of an object edge, you will set the last diagonal entry to 1 or 0 resp. Finally, the fourth and fifth entries in the diagonal projector matrix determine, whether rolling is enabled or not. In the following section, we will discuss several application scenarios of the proposed tactile-servoing framework.

V. Experimental Evaluation

As shown in Fig. 7, we mounted the tactile sensor pad as a large fingertip to a 7-dof Kuka lightweight robot arm operated in joint-space compliance mode using a stiffness parameter of 200 Nm/rad and a damping parameter of 0.2 Nm·s/rad. The control basis framework maps Cartesian-space twists into joint-angle velocities, thus changing the equilibrium posture of the robot controller. The tactile sensor pad provides an array of 16×16 tactels measuring contact forces with 12bits resolution [21]. The sampling frequency of the tactile sensor as well as the control cycle frequency of the robot arm are set to 250Hz. We use manually tuned PID parameters for the tactile servoing controller.

All the experiments discussed in the following are also shown in the accompanying video [12] and follow the same course: Initially the robot is moved to its working area, holding this posture until object contact is established. As soon as a pressure threshold is exceeded, the robot switches to a specific, previously determined tactile servoing task.

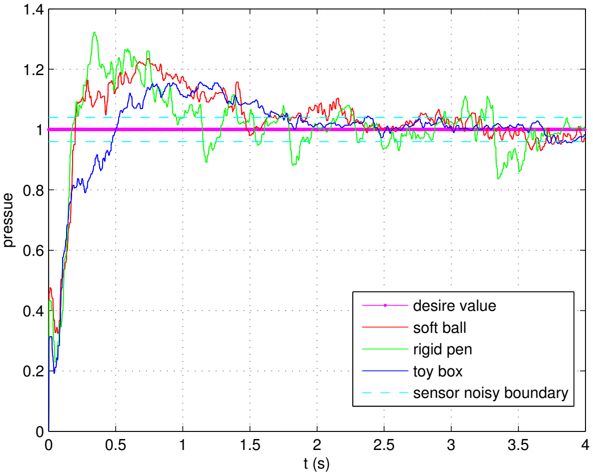


Fig. 8: Normal force control on objects of different stiffnesses. Tracking results get close to the sensor's noise level of 0.04.

In order to reduce the noisiness of the feature signals, we apply a smoothing filter to both the force/pressure feature and the line orientation feature α. To this end, we average the ten most recent measurements, i.e. in a time window of 40ms. The position feature is smooth enough due to the averaging of Eq. 1.

A. *Tracking point contacts*

Contact point tracking has an important application for multi-finger grasping and manipulation. In both cases, fingers need to maintain object contact with a given contact force and they should ensure, that the contact location remains on the fingertip area – optimally in its center – to avoid slipping off.

1) Normal force control: In the first experiment we only control the normal force applied to the object. Accordingly, the task-dependent projector matrix has the form $P = \text{diag}(0, 0, 1,\ 0, 0, 0)$. As the quality of force control heavily depends on the stiffness of objects (softer objects allow for a larger motion range given a fixed force range) we evaluated the control performance on various objects of different stiffnesses: a rigid pen, a toy box from rigid foam, and a soft ball. The results for maintaining a desired pressure level of p=1 are shown in Fig. 8. As expected, stiffer objects take longer to converge to a stable tracking result (response time) and exhibit stronger force oscillations given similar deflections. However, in all cases the desired force level will eventually be well maintained as shown in Tab. I. The steady state error and standard deviation are computed from the time series starting after the response time and lasting until 20s. All values are obtained by averaging over 20 trials.

2) Contact position control: In the second experiment, we focus on contact position tracking and neglect the applied normal force. The goal is to maintain the COP of the contact region at the center of the tactile sensor frame. Accordingly, the projection matrix equals $P = \text{diag}(1, 1, 0,\ 0, 0, 0)$. The

TABLE I: Normal force control: tracking results

object	steady state error	standard deviation	response time
rigid pen	0.0032	0.039	2.5s
toy box	0.0026	0.039	2s
soft ball	0.0010	0.043	2s

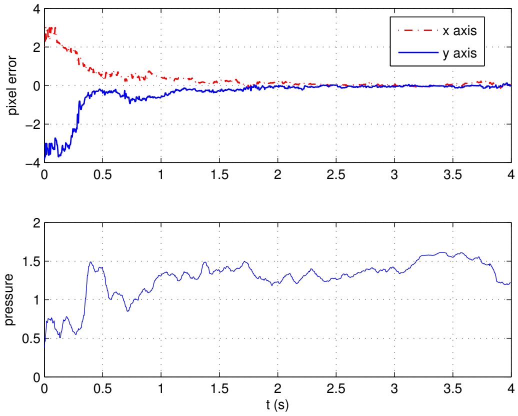


Fig. 9: Contact position control; above: position error; below: normal force evolution

experimental results are shown in Fig. 9. As can be seen from the top sub-figure an initial position offset is corrected within less than 1s. The steady state error and response time are summarized in Tab. II. As can be seen from bottom sub-figure the normal force applied in this experiment evolves randomly as it is not controlled. Note, that a large normal force – due to friction – will also cause large tangential forces, rendering the sliding motion more difficult. Hence, normal force control should be generally enabled.

3) Simultaneous position and force control: Finally, we combine both controllers to evaluate the more realistic scenario of contact location tracking while maintaining a given small contact force. The (superimposed) projector matrix becomes $P = \text{diag}(1, 1, 1,\ \ 0, 0, 0)$. The evolution of the errors in contact position and force are shown in Fig. 10. While we obtain similar position tracking results as before (upper sub-figure), we observe some high-frequency oscillations along the normal axis initially (lower sub-figure). This is due to tangential friction occurring with the sliding motion. Quantitative results are summarized in Tab. III.

B. Track contact point and increase contact area by rolling

The fourth and fifth row of the task Jacobian (Eq. 3) provide another mode of operation to compensate for positional errors of the COP: Instead of realizing a translational sliding motion, this control behavior realizes a rolling motion, thus changing the contact point both on the tactile sensor and the object's surface. While previous approaches to realize rolling employed complex algorithms to determine the point of revolution and a corresponding joint-space robot motion [19], the tactile servoing approach proposed here is conceptually much easier: a deviation in contact position is simply mapped to a rotational twist within the tangential plane of the sensor. Because we do not explicitly compute the point of revolution and do not know the shape of the object, the normal force will probably be disturbed due to this motion. However, the normal force controller, running in parallel, will counteract and maintain a predefined force level. The employed projector matrix equals $P = \text{diag}(1, 1, 1,\ \ 1, 1, 0)$, i.e. simultaneously realizing sliding and rolling as well as force control.

TABLE II: Contact position control: tracking results

Axis	steady state error	standard deviation	response time
X	-0.0027 pixel	0.0440 pixel	2s
Y	-0.0406 pixel	0.0509 pixel	2s

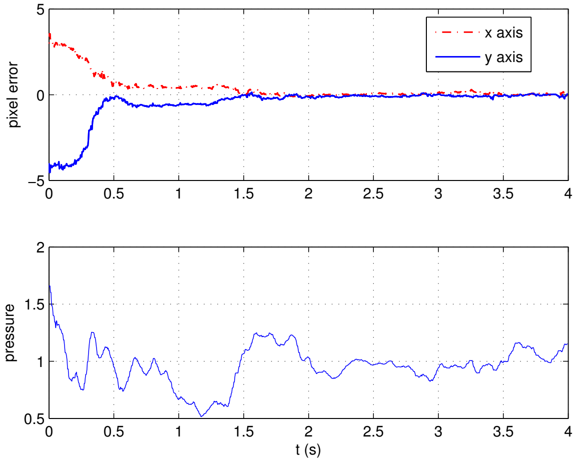


Fig. 10: Hybrid position and force control.

TABLE III: Hybrid position/force control: tracking results

Axis	steady state error	standard deviation	response time
X	0.0041 pixel	0.1146 pixel	1.8s
Y	0.0082 pixel	0.1158 pixel	1.8s
force	-0.0014	0.1335	2s

The resulting rolling motion is visualized in Fig. 11. An initial positional offset along the y-axis is compensated by a rolling motion about the sensor's x-axis (stage S1). When there is no contact point offset, the rolling motion stops (stage S2). After 4s the object was displaced yielding a negative position offset that was compensated by a rolling motion into the opposite direction (stage S3). This behavior can nicely be seen in the video [12] as well.

The rolling behavior has the beneficial side-effect of increasing the area of contact between the finger tip and the object. This is an important capability for grasp stabilization. Although classical grasp planning considers point contacts only, a large contact area naturally increases the grasp wrench space and thus increases the ability to resist to external distur-

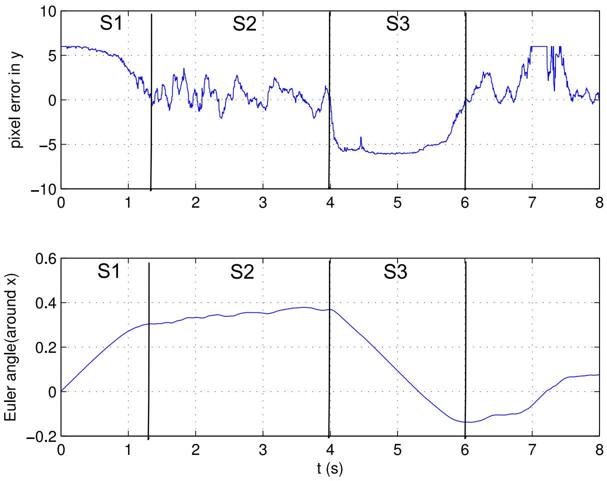


Fig. 11: Orientation control of surface normals by rolling

bances. Furthermore, a prerequisite for successful tactile object exploration will be to maintain a large contact area during exploration in order to collect as much shape information about the object as possible.

How this side effect is achieved? Assuming large object and sensor surfaces, a small contact area typically results from a badly tilted sensor w.r.t. the object surface. In this situation the sensor only touches an object edge instead of the whole surface. This contact is often located off-center on the sensor array. The corrective rolling motion to move the COP into the sensor's center will also reduce the tilting and eventually result in the desired surface contact. This state also constitutes a fixed point of the controller dynamics, because the COP will be in the center of tactile array in this case.

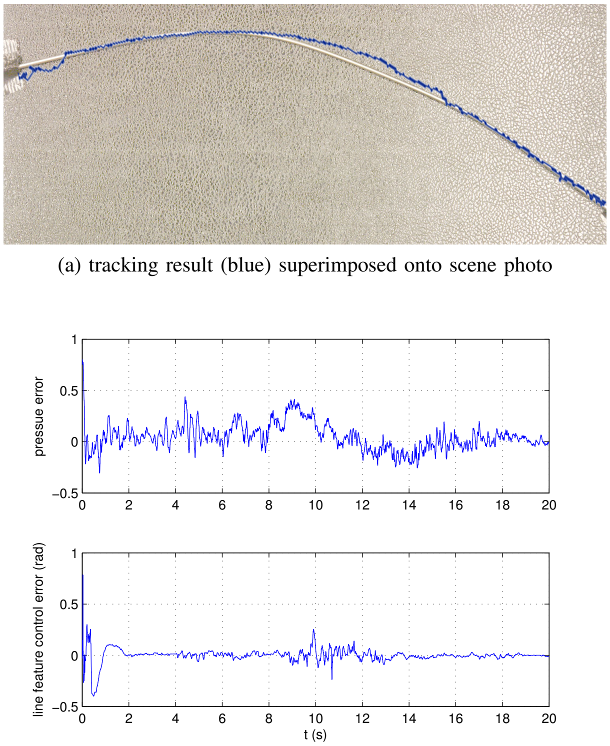
C. *Tracking an object edge on the sensor surface*

The orientation around the normal axis is controlled using the orientation angle α of a line in the tactile image emerging from an object edge on the sensor. For this control task the last row of the Jacobian matrix is important, resulting in a projector matrix $P = \text{diag}(0, 0, 1,\ 0, 0, 1)$. The tracking result for this experiment is qualitatively shown in the video [12] only. However, the next experiment also employs this subtask and provides an evaluation in Fig. 12b.

D. *Tracking of an unknown cable*

The previous experiments illustrated the performance of the proposed tactile servoing controllers in various scenarios, neglecting external motion commands $\mathbf{V}_s^{\text{ext}}$. However, the aim of the following two tasks is to illustrate, that we can realize complex emergent exploration behavior combining the tactile servoing and some externally provided guidance motion.

In the first experiment, we consider the task of tracking the unknown shape of a cable lying flat on the table. To this end, the sensor should be aligned to the local orientation of the cable, it should maintain the tactile imprint within its sensor boundaries (optimally in the center), and it should actively control the contact force. Accordingly we choose a projector matrix $P = \text{diag}(1, 1, 1,\ 0, 0, 1)$. In order to follow the cable in space, we additionally impose an external tangential motion onto the sensor along its y-axis, which coincides with the current orientation of the cable (if orientation tracking is successful). Thus $\mathbf{V}_s^{\text{ext}} = [0, 1, 0,\ 0, 0, 0]^t$.



(a) tracking result (blue) superimposed onto scene photo

(b) Evolution of contact pressure and orientation angle α

Fig. 12: Tracking of an unknown cable on the table

Fig. 12a shows a photo of the tracked cable superimposed with the object shape (blue line) estimated from the forward kinematics of the robot arm when following the cable with tactile servoing. The lower subfigures show the evolution of the normal force and the line orientation error over time. After some initial oscillations, the robot manages to align the cable imprint on the sensor with its y-axis. The force error – targeting a feature value of p=1 – is relatively high initially but the force eventually converges towards the desired value.

E. *Exploring the shape of an unknown object*

The second experiment illustrating the power of the proposed tactile servoing framework, aims at tactile object exploration: The sensor should slide over the unknown surface of the object in order to accumulate a dense shape model. Lacking an appropriate control framework, previous work acquired the corresponding tactile information by repeated establishment and breaking of object contact [14].

To realize this complex task, we decompose it into several phases: after establishing contact to the object, the robot maximizes the sensed contact area and aligns its y-axis with the major axis of the contact region applying the control schemes of sec. V-B and sec. V-C simultaneously.

Subsequently, by imposing a tangential motion along the sensor's x-axis (orthogonal to the major axis of contact region), we induce the exploration motion. The simultaneously running tactile servoing controller maintains the optimal orientation and position of the tactile sensor on the object's surface by generating appropriate sliding and rolling motions. This task exploits all tactile servoing behaviors employing the projector matrix $P = \mathbf{1}$. As a result the object exploration behavior emerges automatically.

Similarly we can explore the object along the other direction, if we follow the contact's major axis instead (cf. previous task in sec. V-D). In the accompanying video [12] we change the direction of the external guidance motion $\mathbf{V}_s^{\text{ext}}$ in order to realize a scanning of the object into both directions.

VI. Conclusion

The tactile servoing control framework proposed in this paper allows to realize a large range of tactile tracking and exploration tasks. To this end, it's only necessary to choose the task-specific projector matrix P choosing which tactile servoing primitives (sliding, rolling, turning, force control) should be applied.

The integration of an externally driven guidance motion $\mathbf{V}_s^{\text{ext}}$ allows to realize complex exploratory behavior. In the shown example tasks, we only used very simple, static guidance motions. However, if those guidance motions are computed from tactile feedback as well, one can easily realize even more complex exploration behavior, e.g. to drive the tactile sensor towards interesting spots on the object's surface, like ridges, edges or corners. In future work, we will extend the feature base and develop appropriate control algorithms into this direction.

Acknowledgments

This work was supported by the German Research Foundation, DFG (SPP 1527-1 and SFB 673).

References

[1] P. Allen and P. Michelman. Acquisition and interpretation of 3-D sensor data from touch. *Trans. Robotics and Automation*, 6(4):397–404, 1990.

[2] A. Berger and P. Khosla. Using tactile data for real-time feedback. *Int. Journal of Robotics Research*, 10(2), 1991.

[3] N. Chen, R. Rink, and H. Zhang. Efficient edge detection from tactile data. In *Proc. IROS*, volume 3, 1995.

[4] Hao Dang, J. Weisz, and P.K. Allen. Blind grasping: Stable robotic grasping using tactile feedback and hand kinematics. In *Proc. ICRA*, 2011.

[5] P. Dario and D. De Rossi. Tactile sensors and the gripping challenge. *IEEE spectrum*, 22(8):46–52, 1985.

[6] R.O. Duda and P.E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Comm. ACM*, 15(1):11–15, 1972.

[7] Ch. Elbrechter, M. Götting, and R. Haschke. Image Component Library (ICL): www.iclcv.org.

[8] V. Ho, T. Nagatani, A. Noda, and Sh. Hirai. What can be inferred from a tactile arrayed sensor in autonomous in-hand manipulation? In *Proc. CASE*, page 461, 2012.

[9] M. Huber. *A hybrid architecture for adaptive robot control*. PhD thesis, Univ. of Massachusetts, 2000.

[10] P. Jenmalm and R.S. Johansson. Visual and somatosensory information about object shape control manipulative fingertip forces. *J. Neuroscience*, 17:4486–4499, 1997.

[11] R.S. Johansson and G. Westling. Roles of glabrous skin receptors and sensorimotor memory in automatic control of precision grip when lifting rougher or more slippery objects. *Exp. Brain Research*, 56:550–564, 1984.

[12] Q. Li. Video: www.youtube.com/watch?v=TcWipks3qJ0.

[13] Q. Li, R. Haschke, B. Bolder, and H. Ritter. Grasp Point Optimization by Online Exploration of Unknown Object Surface. In *Proc. Humanoids*, 2012.

[14] M. Meier, M. Schöpfer, R. Haschke, and H. Ritter. A Probabilistic Approach to Tactile Shape Reconstruction. *Trans. Robotics*, 27(3):630–635, 2011.

[15] C. Muthukrishnan, D. Smith, D. Myers, J. Rebman, and A. Koivo. Edge detection in tactile images. In *Proc. ICRA*, volume 4, pages 1500–1505, 1987.

[16] Z. Pezzementi, E. Plaku, C. Reyda, and G.D. Hager. Tactile-Object Recognition From Appearance Information. *Trans. Robotics*, 27(3):473–487, 2011.

[17] M.H. Raibert and J.J. Craig. Hybrid position/force control of manipulators. In *American Society of Mechanical Engineers, Winter Annual Meeting, Chicago, Ill*, 1980.

[18] A. Schmitz, P. Maiolino, M. Maggiali, L. Natale, G. Cannata, and G. Metta. Methods and Technologies for the Implementation of Large-Scale Robot Tactile Sensors. *Trans. Robotics*, 27(3):389–400, 2011.

[19] M. Schöpfer, H. Ritter, and G. Heidemann. Acquisition and Application of a Tactile Database. In *Proc. ICRA*, pages 1517–1522, 2007.

[20] M. Schöpfer, F. Schmidt, M. Pardowitz, and H. Ritter. Open source real-time control software for the Kuka light weight robot. In *Proc. WCICA*, pages 444–449, 2010.

[21] C. Schürmann, R. Kõiva, and R. Haschke. A modular high-speed tactile sensor for human manipulation research. In *World Haptics Conference*, 2011.

[22] S.A. Stansfield. A robotic perceptual system utilizing passive vision and active touch. *International Journal of Robotics Research*, 7(6):138–161, 1988.

[23] Kenji Suzuki, Isao Horiba, and Noboru Sugie. Linear-time connected-component labeling based on sequential local operations. *Computer Vision and Image Understanding*, 89(1):1–23, 2003.

[24] N. Wettels and G.E. Loeb. Haptic feature extraction from a biomimetic tactile sensor: Force, contact location and curvature. In *Proc. ROBIO*, pages 2471–2478, 2011.

[25] Danfei Xu, Gerald Loeb, and Jeremy Fishel. Tactile identification of objects using Baysian exploration. In *Proc. ICRA*, Karlsruhe, 2013.

[26] H. Zhang and N.N. Chen. Control of contact via tactile sensing. *Trans. Robotics and Automation*, 16(5), 2000.