Learning a Tool's Homogeneous Transformation by Tactile-based Interaction

Qiang Li, Robert Haschke and Helge Ritter

Abstract— We propose a tactile-based manipulation strategy to learn the homogeneous transformation of a grasped rigid tool, using tactile sensing delivered through a tactile matrix sensor covering the tool surface. Exploiting the self-learning tactile servoing controller, a robot safely use the tactile tool to implement different tactile-based exploration primitives (EPs). Considering EPs as input and observing the tactile contacts as output, the robot can robustly estimate the tool's homogeneous transformation. The learned transformation are combined with the known robot's kinematics model to form a new tool manipulation kinematics chain, thereby realizing a step towards a "plastic body schema" for flexible tool use by a robot.

We numerically evaluate the method's feasibility and robustness assuming that measurements are only polluted by Gaussian white noise, then evaluate the proposed method with a real robot setup – a KUKA LWR and a SCHUNK SDH-2 hand grasping a rigid tactile tool. With the new manipulation chain, we demonstrate two tactile tool's servoing experiments: reactively sliding and rolling tool and tracking an unknown object edge.

I. Introduction

Recently lots of progress has been made at tactile sensors' development and applications on robots. e.g. new developed tactile sensors [5] have been covered on the robots body to provide them with body self-awareness, on the robots hand in order to improve the hands dexterous manipulation capability [8] , [13] and on wearable devices [4] to facilitate the human robot interaction. In this work, we will apply tactile sensing to tools' usage domain and let robots learn to use an in-house developed tactile tool (Fig. 1 left). The tool is composed of two parts: a plastic handle and a planar tactile matrix sensor. There are two reasons why we want robots to use this modular "intelligent" tool. (1) It can help robots "touch and feel" an object outside of their workspace. (2) It can be used by different robot platforms which have a hand. The tactile tool will serve as a "tactile camera" and help "blind" robots percept and manipulate objects in an unknown environment.

[22], [2] developed a similar hand-held tactile tool and used it to acquire multi-modal tactile sensing data and classify touched objects. The authors' work was following the research line of perception and focusing on recognizing unknown objects. Our study is going towards the autonomous robotic manipulation line and focusing on learning the tool's homogeneous transformation T_s^e in order that robots can control the tool based on the tactile feedback.

The paper is organized as follows: In Sec. II, we review the state of the art of learning to use a tool. In Sec. III,

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

the self-learning tactile servoing controller and estimation method are introduced. In Sec. IV, we present the result from the simulations and real robot experiments. Finally, a conclusion is given in Sec. V.

II. State of the Art

Sensors frequently used to estimate an tool's transformation include cameras, force torque sensors. Vision is probably most studied modality to study the tool's translation. In [11], [12], authors used vision to estimate a tool tip's offset with respect to the robot end-effector (EEF). To this end, the authors used optical flow to extract the unmodeled tool's tip, rotated the tool in free space and estimated 3D position of the tip which can best explained the noisy 2D detections. In [20] a more general tool's representation – also considering the tool's orientation was learned. Authors modeled a robot body and tool with the Bayesian network. Every network node was described by joint pose, which was measured by a visual marker. By observing the robot's babbling movement, authors learned the kinematics model of the robot and the tool's homogeneous transformation. One constraint of this method is that all markers have to be observed simultaneously, which makes the method feasible only when an external camera is used.

Haptic perception is an important complementary modality which works very well in the dark or occlusion case. Depending on the location of sensors, it can be divided into: intrinsic and extrinsic haptic[7]. Bicchi [3] proposed a contact position estimation method in which a known fingertip interacts with an unknown object based on the soft finger contact model using measurement from an intrinsic force and torque sensor. Yiannis [10] extended this approach to not only estimate the translation but the contacted plane's normal direction. Intrinsic haptic based method has been widely used in learning a tip-shaped tool's offset.

Comparing with the intrinsic haptic, the extrinsic perception has not been well studied yet. Recent years, lots of advanced tactile sensors [17], [8], [13] became available and were used to measure accurate contact position, shape, pressure distribution. They have been used to "blindly" recognize and manipulate unknown objects. Lots of tactile based methods have been employed to identify contacted objects' parameters [15], classify contacted objects [21] [18] [23] [22] and study in-hand manipulation [24]. We assembled a planar tactile matrix sensor at a robot EEF and proposed a tactile servoing framework [14]. Within the framework, the robot could be used to explore unknown objects' surface by combining different tactile-based exploration primitives. In
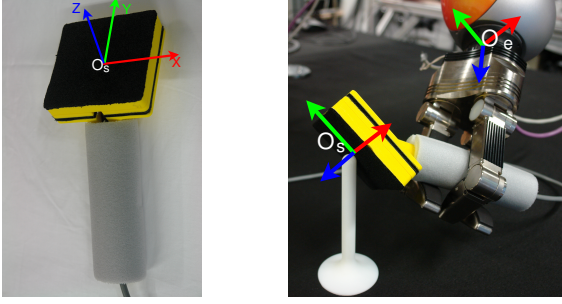
Fig. 1: Tactile tool with tool frame (left) and tool's homogeneous transformation T_e^s (right)

[14], the homogeneous transformation of the tactile matrix sensor with respect to the robot EEF was obtained from the CAD model. In the present work, we are releasing this assumption and employing tactile feedback to learn the homogeneous transformation of the tool in an iterative fashion.

III. METHODS

A. Tactile sensor

The proposed estimation approach depends heavily on a sensitive tactile matrix sensor providing spatially localized contact measurements. In our experiments we employ an array of 16×16 taxels 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 [19]. The sensor exploits the piezo-resistive sensing principle, measuring changes in resistance of a conductive foam due to an applied force.

In the present scenario, the tactile feature vector comprises the overall contact pressure p and the contact position $\vec{c}$ which are determined from the sum of all taxel pressures p_{ij} within a contact region R and the pressure-weighted center of R:

$$p = \sum_{ij \in R} p_{ij} \qquad \vec{c} = p^{-1} \sum_{ij \in R} p_{ij} \vec{c}_{ij}, \qquad (1)$$

where c_{ij} are the discrete coordinates of the taxels on the sensor array. Due to this averaging we can achieve a spatial resolution of 0.5mm [14].

B. Tactile servoing controller and homogeneous transformation estimation of a tool

Because the tool's transformation is unknown, we adapt the previous proposed tactile servoing framework and deliberately design a self-learning tactile servoing controller shown in Fig. 2. It is composed of a controller and three estimators. The tool's EPs (sliding/rolling tool on an object) are implemented using the controller with the initialized transformation of the tool. Using the obtained contact data, the tool's transformation is updated by the estimators. The basic idea of the controller is to use tactile feedback control loop to maintain the desired contact force f_{des} on the normal direction of the tool and slide the tool on the orthogonal plane of the normal direction. Combining with the feedback loop,
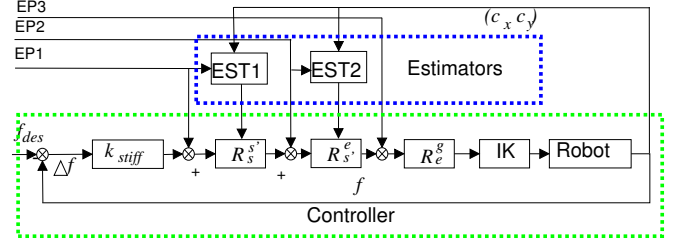

Fig. 2: In the self-learning tactile servoing controller $R_s^{s'}$ and $R_{s'}^e$ are unknown. The transformation matrix is estimated by EST1–rotational angle, EST2–normal direction and EST3–translation (not shown) by implementing the EPs

feedforward control commands are used to implement the EPs simultaneously. Several EPs are defined as following

1) EP1: the tactile tool is moved along a line on the orthogonal plane of the tool's normal direction.
2) EP2: the tactile tool follows a square-shaped trajectory on the orthogonal plane of the tool's normal direction.
3) EP3: the robot EEF is rotated around its local frame's x, y, z axis.

The controller can maintain the contact in presence of disturbance of the tool's transformation when the EPs are implemented. In order to maintain the desired contact force, the desired linear velocity of the tactile tool and the robot arm EEF can be computed from the deviation of the contact force Δf, manipulation stiffness k_{stiff} and the estimated normal direction $\vec{n}_s$.

$$\vec{v}_e = \vec{v}_s = k_{stiff} \cdot \Delta f \cdot \vec{n}_s \qquad (2)$$

The desired Cartesian motion of the robot arm EEF is implemented by an inverse kinematics controller [16].

Fig. 3 shows the whole manipulation scenario and the relevant coordinate frames. We assume that

1) the object is static.
2) the point contact model is used.

O_g is the global reference frame. O_e is the robot arm EEF frame. O_s is the tool frame which has its origin in the center of tactile matrix sensor, x, y axes parallel to the taxel grid and z axis along the normal direction of the tactile matrix sensor (Fig. 1). $O_{s'}$ is a virtual frame which has the same normal direction like O_s, but is randomly rotated about that axis. The relative orientation is denoted by $R_{s'}^s(\alpha)$. (c_x, c_y) is the contact position on the tactile matrix sensor. We use $R_{s'}^e$ to represent the relative transform matrix from the robot arm EEF frame to the virtual frame.

The homogeneous transformation T_e^s is composed of two parts: a rotation matrix – R_s^e and a translation – $\vec{r}_{es}$. The rotation matrix can be estimated by Eq. 3.

$$R_s^e = R_{s'}^e R_s^{s'} = R_g^e R_{s'}^g R_s^{s'} = R_g^e R_{s'}^g R_z(\alpha) \qquad (3)$$

Where R_g^e is the rotation matrix of the robot arm EEF. $R_{s'}^g$ is the rotation matrix of the virtual frame and can be computed from the normal direction of the tool. and $R_s^{s'}$ is decided by the rotational angle based on the current estimated normal direction. After the rotation matrix R_s^e is computed,
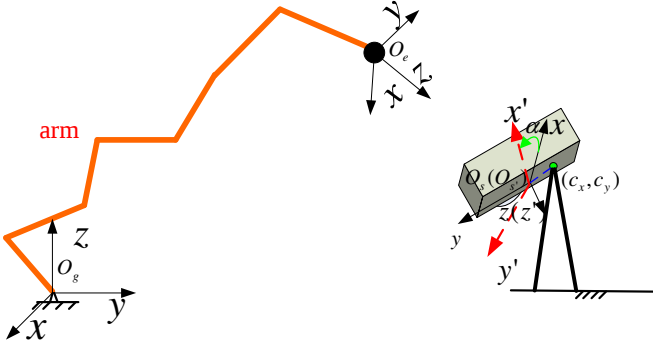
Fig. 3: In the exploration manipulation scenario the homogeneous matrix T_s^e is estimated

Fig. 4: Rotation angle estimation from contact points trajectory

EP2 is implemented to update the estimation of the normal direction. Via iteratively estimating normal direction and rotation matrix $R_s^{s'}$, the accuracy of the rotation matrix R_s^e is improved.

The translation of the tool is estimated based on the Eq. 4.

$$\vec{v}_s^e = \vec{v}_e^e + \omega_e^e \times \vec{r}_{es} \tag{4}$$

where $\vec{r}_{es}$ is the translation between robot arm EEF and tool EEF with respect to the robot arm EEF frame. A robot arm EEF is rotated around its local frame with angular velocity ω_e^e. If the linear velocity of the tool in the robot EEF frame $\vec{v}_s^e$, the linear velocity of the robot EEF in the robot EEF frame $\vec{v}_e^e$ and angular velocity ω_e^e are known, the translation $\vec{r}_{es}$ can be iteratively estimated.

1) Normal direction estimation of the tool–EST2: Kinesthetic teaching is used to provide an initialized estimation of the tool's normal direction. To this end, the human guides the tool to approach an object along the tool's normal direction. Because the tool is rigid, the tool's approach direction is parallel to the moving direction of robot arm's EEF. We use the robot arm's EEF moving direction to serve as the initialized tool's normal direction. After implementing EP1, the normal direction is updated following the integral adaptive law [9]:

$$\dot{\vec{n}}_s = -\Gamma_n * \bar{P}(\vec{n}_s)L_n(t)\vec{n}_s \tag{5}$$

$$\dot{L}_n = -\beta_n L_n + \frac{1}{1+\| v_s^g \|^2}v_s^g v_s^{gT} \tag{6}$$

Where $L_n(0) = O_3$. $\vec{n}_s(0)$ is initialized normal direction. Γ_n is a positive constant for tuning the speed of convergence and β_n is a positive forgetting factor. $\bar{P}(\vec{n}_s)$ is defined as

$$\bar{P}(\vec{n}_s) \triangleq \mathbb{1} - \vec{n}_s \vec{n}_s^T \tag{7}$$

The virtual frame $R_{s'}^g$ can be constructed by the normal direction and its two orthogonal vectors in the null space.

2) Orientation estimation of the tool–EST1 : The relative orientation $R_s^{s'}$ is computed from a rotational angle α around current estimated normal direction. In order to estimate this angle, we implement EP1 – sliding the tool on an object along the positive direction of the x' axis with a constant linear velocity. The exploring behavior is stopped when the
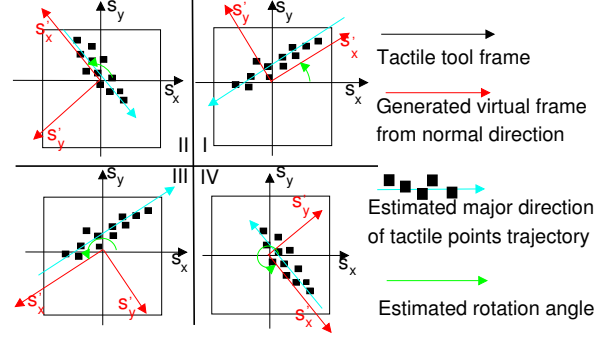
contact point approaches to the boundary of the taxel grid. Because the object is static, the tactile contact trajectory is along the x' axis negative direction (Fig. 4). The slope of the tactile contact trajectory can be computed from the major direction of the 2D tactile contact trajectory using Principle Component Analysis (PCA). The rotational angle can be computed from this slope considering four possible relative pose.

$$I, IV : \alpha = arctan(k) + \pi \tag{8}$$

$$II, III : \alpha = arctan(k) \tag{9}$$

3) Translation estimation of the tool–EST3: When the EP3 is implemented, the contact points on the tactile tool will stay nearby the tool frame's origin. Hence, we can assume that the linear velocity of the tool frame's origin and the contact point's linear velocity are equal and opposite in direction. $\vec{v}_s^e$ is computed from

$$\vec{v}_s^e = (-1) \times R_s^e \cdot \vec{v}_s^s \tag{10}$$

where $\vec{v}_s^s$ is the derivative of the contact position. R_s^e is the computed tool frame orientation from III-B.2. The required $\vec{v}_e^e$ and ω_e^e in Eq. 4 are extracted from the robot arm EEF body velocity $\hat{V}_e^e$ with respect to the global reference frame.

$$\hat{V}_e^e = g_e^{-1}\dot{g}_e = \begin{bmatrix} R_e^{gT}\dot{R}_e & R_e^{gT}\dot{p}_e \\ 0 & 0 \end{bmatrix} \tag{11}$$

where g_e is the homogeneous matrix of the robot arm EEF. $\omega_e^e = R_e^T \dot{R}_e$ is the skew symmetric matrix generated by the angular velocity ω_e^e. $\vec{v}_e^e = R_e^{gT}\dot{p}_e$ is the linear velocity of the robot arm EEF with respect the robot arm EEF frame.

$\vec{r}_{es}$ can be estimated following the integral adaptive law [9]

$$\dot{\vec{r}}_{es} = -\Gamma_r[L_{r_{es}}(t)\vec{r}_{es} - c_{r_{es}}(t)] \tag{12}$$

$$\dot{L}_{r_{es}} = -\beta_r L_{r_{es}} - \hat{\omega}_e^e \hat{\omega}_e^e \tag{13}$$

$$\dot{c}_{r_{es}} = -\beta_r c_{r_{es}} + \hat{\omega}_e^e \vec{v}_s^e \tag{14}$$

In the equations, the initialized $L_{r_{es}}(0) = O_3$ and $c_{r_{es}}(0) = O_3$. Γ_r is a positive constant for tuning the speed of convergence and β_r is a positive forgetting factor.
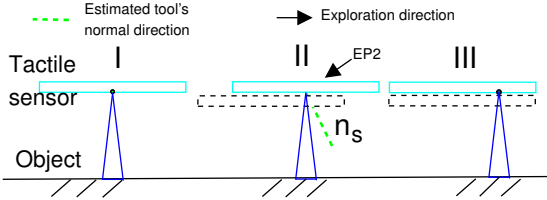
Fig. 5: Simulated tactile servoing scenario – the dashed square is the tool's sliding result if an inaccurate normal direction is used in the stage II. The tool (solid square) moves along the real normal direction to compensate the penetration in the stage III.

IV. SIMULATIONS AND ROBOT EXPERIMENTS EVALUATION

The proposed method is evaluated in simulation and real robot experiments. Because a robot hand can grasp a tactile tool's handle at any unknown configure, it is very difficult to get the ground truth of the tactile tool's homogeneous transformation to evaluate the accuracy of the estimation method. On the other hand, it is a very time consuming process to evaluate the robustness of the estimation because many times experiments have to be repeated to evaluate the algorithm's convergence in different initialized conditions. Hence, we use simulations to evaluate the algorithm's feasibility and robustness, use real robot experiments to show the estimation's results and finally use tool's tactile servoing experiments to demonstrate the results' practical application.

A. Simulation

The Robotics Toolkit for Matlab [6] is employed to facilitate robot kinematics simulation. We are using KUKA LWR DH parameters from [1] and modeling the SCHUNK hand and the tool as a simplified external rigid link specified by a homogeneous transformation matrix. The tactile matrix sensor is modeled as a $8cm \times 8cm$ square which is the same size like the real tactile matrix sensor and located on the tool EEF. In the simulations, we do not explicitly model the contact force but use a simulated tactile servoing scenario to compute the contact position on the square. The simulated tactile servoing scenario is shown in Fig. 5. At the stage I, the tool's contact point is in the center of the tactile matrix sensor. At the stage II, the EP2 is implemented using the inaccurate tool normal direction $\vec{n}_s$ (green dashed line) and the tactile tool will penetrate the object. At the stage III, the tactile matrix sensor moves along the inverse direction of the real normal direction in order to keep the contact point on the object's and tactile tool's surface.

1) Normal direction estimation of the tool: The core method is based on Eq. 5, Eq. 6. We initialize the tool's normal direction with the heuristic information (e.g. kinesthetic teaching). All other parameters β_n, Γ_n are manually tuned to guarantee the learned parameters converge. The EP2 is very important factor which can decide whether the

TABLE I: 20 times noisy simulation results for the estimation of the normal direction and the translation

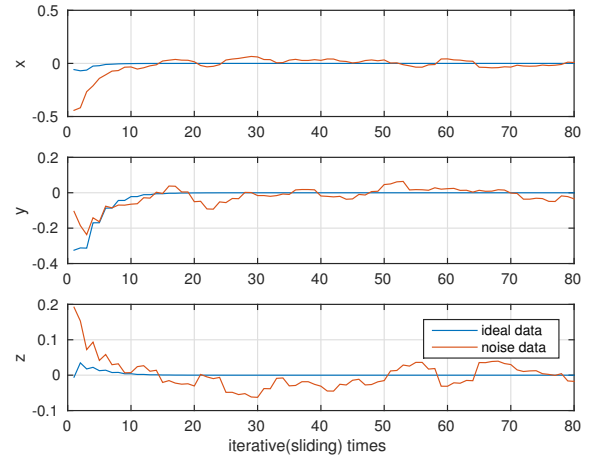|	average of deviation	average of standard deviation
$\vec{n}_s$	(0.0002,0.0027,0.004)	(0.0256,0.0228,0.0111)
$\vec{r}_{es}$	(-0.0009,0.0006,0.0002)m	(0.0015,0.0012,0.0014)m

Fig. 6: Tool frame's normal direction estimation in simulation. the deviation of the estimation is shown in y axes

normal direction will converge to the real value. The system identification theory [9] has shown that estimated parameters will theoretically and robustly converge to the real values with a full persistent exciting (PE) exploration. In this paper, we are using EP2 – the tactile tool following a square-shaped trajectory and moving with the linear velocity – 0.02m/s. The result shows that this exploration action is enough to estimate the normal direction very well.

The estimation result is shown in Fig. 6. With around 15 steps, the estimated normal direction converge to its real value without bias. The ideal simulation is repeated 20 times with different initialized normal directions. The initialized normal directions are obtained from randomly initialized orientation with disturbed Euler angles drawn from a uniform distribution $\mathcal{U}(-\pi/4, \pi/4)$. In all simulations, the estimated normal directions converge to real value without bias.

From Eq. 5, Eq. 6, it can been seen that the estimation of the normal direction only depends on the linear velocity of the tactile tool. In oder to simulate the measure noise, we superimpose Gaussian noise $n_{v_s^g} \sim \mathcal{N}(0, 0.001)$ on the linear velocity v_s^g (0.02m/s) to evaluate robustness of the method. The simulation result is also shown in Fig. 6. The noisy simulation is also repeated 20 times with different initialized normal directions. Numerical results at the stable stage (40-80 steps) are shown in Tab. I

2) Translation estimation of the tool: The core method is based on Eq. 12, Eq. 13 and Eq. 14. In simulation, the linear velocity $\vec{v}_s^e$ and $\vec{v}_e^e$ and angular velocity $\hat{\omega}_e^e$ can be computed from the robot arm's forward kinematics and the known tool's homogeneous transformation. Without loss of generality, we assume that $\vec{v}_e^e = 0$. The EP3 is implemented
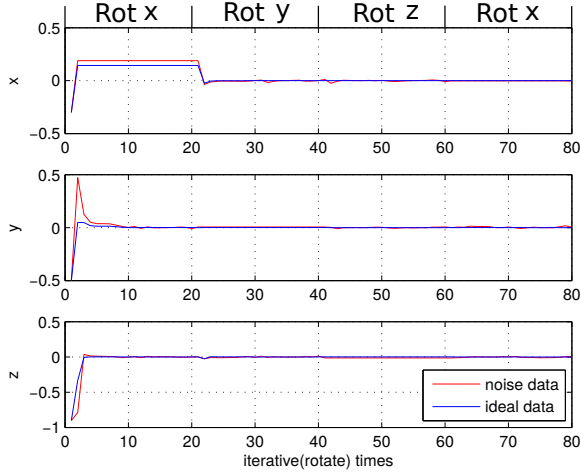
Fig. 7: Tool frame's translation parameters estimation in simulation, Rotate around the robot EEF local x, y, z axis every 20 steps

Fig. 8: Tool's frame translation parameters estimation in the real robot platform

– rolling robot arm EEF around its local x axis (0-20 steps), y axis (21-40 steps) and z axis (41-60 steps) in sequence. The rotation is a sinusoidal velocity function with amplitude 0.15. The initial values of translation are randomly sampled from a normal distribution $\vec{r}_{es} \sim \mathcal{N}(0, 1)$. All parameters $\beta_{r_{es}}, \Gamma_r$ are manually tuned to guarantee the learned translation converge. The simulation result is shown in Fig. 7. It can be seen that the estimation of translation converges to the real translation value without bias in less than 30 steps. The simulation is repeated 20 times with different initial conditions, which are randomly sampled from the normal distribution $\vec{r}_{es} \sim \mathcal{N}(0, 1)$. The estimation of translation in all experiments converge to the real translation value without bias.

We superimpose Gaussian noise $n_{v_s^e} \sim \mathcal{N}(0, 0.001)$ on the linear velocity of the tool. Numerical results at the stable stage (50-80 steps) are shown in Tab. I

B. Robot experiment

In the real robot experiment, a KUKA LWR is assembled on a fixed base and a SCHUNK hand is assembled on the KUKA LWR's effector grasping a tactile tool at a random position on the handle. The KUKA LWR is operated in joint-space compliance mode using a stiffness parameter of 500 Nm/rad and a damping parameter of 0.4 Nm·s/rad. The sampling frequency of the tactile matrix sensor as well as the sampling and control cycle frequency of the robot arm are set to 250Hz. We use manually tuned manipulation stiffness k_{stiff} in order that the contact and manipulation are stable. The contacted object has a sharp tip in order to satisfy the assumption of point contact model.

1) Orientation estimation of the tool: The KUKA LWR works in the gravity compensation mode for kinesthetic teaching. A linear regression method e.g. PCA. is used to estimate the normal direction of the tool. The KUKA implements EP1 – sliding along x' axis of the virtual frame $O_{s'}$ and the obtained tactile trajectory is used to estimate the rotational angle α using Eq. 8 and Eq. 9.
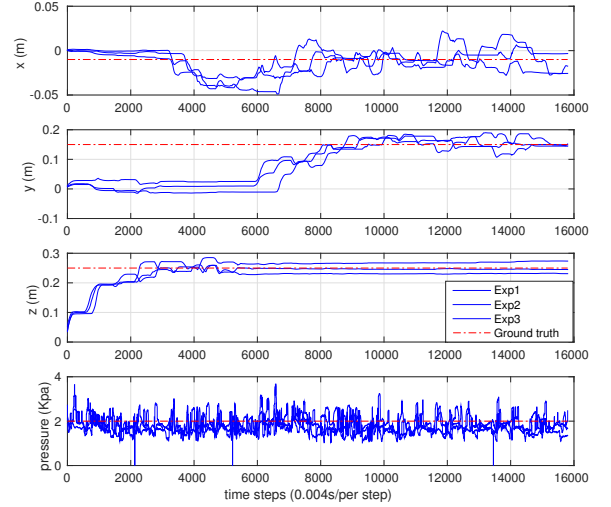
2) Translation estimation of the tool: The KUKA's EEF is rotated around its local x, y, z axis in sequence. We manually limit the rolling scale in order to maintain the contact. The contact point's linear velocity with respect to the tool frame $\vec{v}_s^s$ is computed by the derivative of the contact position. The contact point's linear velocity with respect to the KUKA EEF $\vec{v}_s^e$ is computed by Eq. 10. The contact position measurement is quite noisy, so a median filter whose "window" size is 15 is used in order to get rid of outliers. The derivative of the contact position is passed by an average filter whose "window" size is 100 in order to get rid of the high frequency noise. The orientation and position of the KUKA EEF are obtained from the forward kinematics model, then are used for estimating the angular velocity and linear velocity. These values are also filtered by average filters whose "window" size are 10. The translation estimation result is shown in Fig. 8. It can be seen that the estimation finally converges to the stable value. We can not judge the estimation accuracy because the ground truth is not available. From the experience of simulations, the estimation accuracy depends on the covariance of the measurement noise. The whole estimation process can be found in the accompanying video.

3) update of the tool's normal direction: If the transform matrix R_s^e is estimated, the tactile tool can be slided safely and reached to the specified contact taxels. The normal direction can be updated using Eq. 5 and Eq. 6. We are using a square-shaped EP2 in which the goal taxels in sequence are $(7.5, 7.5) \rightarrow (3, 3) \rightarrow (3, 12.5) \rightarrow (12.5, 12.5) \rightarrow (12.5, 3) \rightarrow (3, 3)$. In order to qualitatively evaluate the updating result of the normal direction, we guide the tactile tool horizontally contact the table. In this configure, the normal direction of tool is $(0, 0, -1)^T$. The comparison result is shown Fig. 9. As we can see that the estimation of the normal direction (red line) from the updated one is more accurate than the estimation from kinesthetic teaching. The
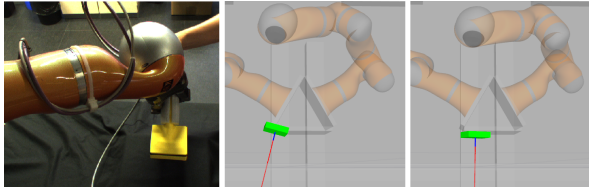
Fig. 9: Normal direction estimation – red line, estimated pose of tactile matrix sensor – green cube. left: ground truth normal direction $(0, 0, -1)^T$, middle: kinesthetic teaching estimation, right: updated by sliding experiment

experiment can be found in the accompanying video.

With the learned tool homogeneous transformation, we can apply the previous tactile servoing controller [14] to the use case of the tactile tool. Although there is the estimation error of translation, the tactile tool is implemented sliding, rolling primitives and also the task of tracking edge of a box successfully. The robust tactile feedback controller compensate the estimation deviation of the translation. The tactile tool servoing experiments can be found in the accompanying video.

V. CONCLUSION AND FUTURE WORK

In this paper, we combine EPs and tactile feedback to endow a robot capability to use a rigid tactile tool. A self-learning tactile servoing controller is proposed to guarantee that EPs are implemented safely. The tactile feedback during the course of implementing EPs is used to estimate the unknown homogeneous transformation of the tool. Our proposed method is proved feasible and robust by using the simulations and robot experiments. In ideal simulations, all estimations of homogeneous transformation robustly converge to the real values. Superimposing artificial noise on the measurement, the mean of transformation estimation also converge to the neighboring area of the real value. The real robot experiments validate the feasibility of the proposed method in the real world. The learned transformation are combined with the known robot kinematics to form a new manipulation chain. Applying the new manipulation chain in the previous proposed tactile servoing work, we demonstrate the tactile tool servoing experiment. Next step the robot will learn more complex tactile tool e.g. with a rotational degree of freedom.

ACKNOWLEDGEMENT

REFERENCES

[1] Bruno Vilhena Adorno. *Two-arm manipulation: From manipulators to enhanced human-robot collaboration*. PhD thesis, Université Montpellier II-Sciences et Techniques du Languedoc, 2011.
[2] Tapomayukh Bhattacharjee, Joshua Wade, Yash Chitalia, and Charles C Kemp. Data-driven thermal recognition of contact with people and objects. In *2016 IEEE Haptics Symposium (HAPTICS)*, pages 297–304. IEEE, 2016.
[3] Antonio Bicchi, J Kenneth Salisbury, and David L Brock. Contact sensing from force measurements. *The International Journal of Robotics Research*, 12(3):249–262, 1993.
[4] Gereon H Büscher, Risto Kõiva, Carsten Schürmann, Robert Haschke, and Helge J Ritter. Flexible and stretchable fabric-based tactile sensor. *Robotics and Autonomous Systems*, 63:244–252, 2015.
[5] Giorgio Cannata, Marco Maggiali, Giorgio Metta, and Giulio Sandini. An embedded artificial skin for humanoid robots. In *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*, pages 434–438. IEEE, 2008.
[6] Peter I. Corke. *Robotics, Vision & Control: Fundamental Algorithms in Matlab*. Springer, 2011.
[7] Ravinder S Dahiya, Giorgio Metta, Maurizio Valle, and Giulio Sandini. Tactile sensingfrom humans to humanoids. *Robotics, IEEE Transactions on*, 26(1):1–20, 2010.
[8] 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.
[9] Petros A Ioannou and Jing Sun. *Robust adaptive control*. Courier Corporation, 2012.
[10] Yiannis Karayiannidis, Colin Smith, Francisco E Vina, and Danica Kragic. Online contact point estimation for uncalibrated tool use. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2488–2494. IEEE, 2014.
[11] Charles C Kemp and Aaron Edsinger. Visual tool tip detection and position estimation for robotic manipulation of unknown human tools. *Computer Science and Artificial Intelligence Laboratory Technical Report*, 83, 2005.
[12] Charles C Kemp and Aaron Edsinger. Robot manipulation of human tools: Autonomous detection and control of task relevant features. In *Proc. of the Fifth Intl. Conference on Development and Learning*, 2006.
[13] Risto Kiva, Matthias Zenker, C Schrmann, Robert Haschke, and Helge J Ritter. A highly sensitive 3d-shaped tactile sensor. In *Proc. Advanced Intelligent Mechatronics (AIM)*, pages 1084–1089, 2013.
[14] Qiang Li, Carsten Schürmann, Robert Haschke, and Helge J Ritter. A control framework for tactile servoing. In *Robotics: Science and systems*, 2013.
[15] Gerald E Loeb. Estimating point of contact, force and torque in a biomimetic tactile sensor with deformable skin.
[16] F. Schmidt and R. Haschke. Control basis framework (CBF) | control basis framework (CBF). https://github.com/fps/CBF.
[17] Alexander Schmitz, Marco Maggiali, Lorenzo Natale, Bruno Bonino, and Giorgio Metta. A tactile sensor for the fingertips of the humanoid robot icub. In *Proc. IROS*, pages 2212–2217. IEEE, 2010.
[18] Alexander Schneider, Jürgen Sturm, Cyrill Stachniss, Marco Reisert, Hans Burkhardt, and Wolfram Burgard. Object identification with tactile sensors using bag-of-features. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 243–248. IEEE, 2009.
[19] C. Schürmann, R. Kõiva, and R. Haschke. A modular high-speed tactile sensor for human manipulation research. In *World Haptics Conference*, 2011.
[20] Jürgen Sturm, Christian Plagemann, and Wolfram Burgard. Adaptive body scheme models for robust robotic manipulation. In *Robotics: Science and Systems*. Zurich, 2008.
[21] Zhe Su, Yao Li, and Gerald E Loeb. Estimation of curvature feature using a biomimetic tactile sensor. In *Proc Amer Soc Biomech*, 2011.
[22] Joshua Wade, Tapomayukh Bhattacharjee, and Charles C Kemp. A handheld device for the in situ acquisition of multimodal tactile sensing data. *arXiv preprint arXiv:1511.03152*, 2015.
[23] Danfei Xu, Gerald E Loeb, and Jeremy A Fishel. Tactile identification of objects using bayesian exploration. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3056–3061. IEEE, 2013.
[24] Hanna Yousef, Mehdi Boukallel, and Kaspar Althoefer. Tactile sensing for dexterous in-hand manipulation in roboticsa review. *Sensors and Actuators A: physical*, 167(2):171–187, 2011.