

3D Contour Following for a Cylindrical End-Effector Using Capacitive Proximity Sensors

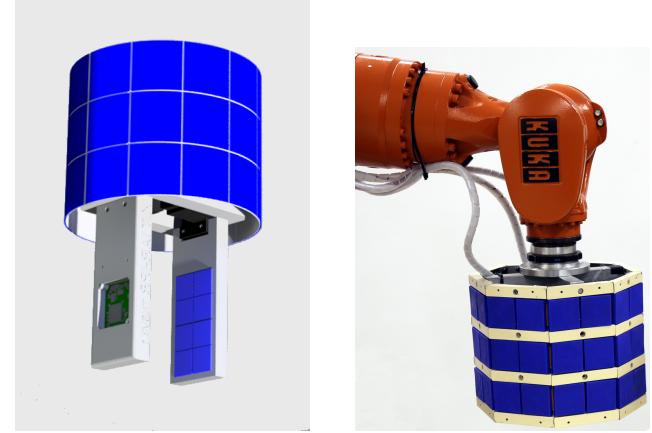
Stefan Escaida Navarro⁺, Stefan Koch⁺ and Björn Hein⁺

Abstract—In this paper we've equipped a cylindrical end-effector with an array of capacitive sensors in order to implement 3D contour following. During the task, using proximity servoing, the sensors are aligned parallel to the surface and kept at a target distance. In addition, due to the spatial resolution, it's possible to estimate the surface's curvature in two dimensions along the rows and columns of the array. We show how a compound movement can be derived from both curvatures that pre-aligns the end-effector in each step, yielding a predictive component for the control scheme. We evaluate our approach with different geometries and show that the curvature information produces smooth contour following paths. We show that the system can handle speeds up to 150mm s^{-1} .

I. INTRODUCTION

One of the persisting goals in robotics is to allow humans and robots to share a common workspace in order for the robot to assist or cooperate with the human in a variety of interaction intense contexts, which can be widely found in industrial, domestic or medical applications. An adequate perception is necessary to provide the required safety – for the human and the equipment – and to be able to implement the desired autonomy. Much effort in research has gone towards robot workspace surveillance by means of optical systems in order to determine static and dynamic features. Path planning schemes can work under these conditions, but the information of the robot surroundings will still be only partially known because of occlusions. Safety concerns can also not be set aside by tactile perception, because in the worst case contact is prohibited. In the better case, allowing contact means to slow down the movement velocity of the robot. The research on proximity sensing aims at closing this perception gap.

In this paper we investigate contour following/obstacle avoidance based on modular capacitive proximity sensors, which were previously presented in [1], [2]¹. The particularity of our sensors is that they work in a self-capacitive mode providing quasi point-like measurements, which is a necessary property for implementing spatial resolution. In the long term, our work is aimed at implementing a complete sensing system for a gripper covering the inside of its fingers



(a) Our goal is to have a sensing system covering the exterior of a gripper, as well as its fingers
(b) The end-effector used in this work, featuring a 3×16 arrangement of proximity sensors

Fig. 1. Our concepts for equipping an end-effector with capacitive proximity sensors

as well as its exterior, as shown in Fig. 1. The current work concentrates on the second aspect.

Our approach builds on the research of others that have provided the foundations for contour following/collision avoidance schemes. Furthermore, we include curvature measurements, which have shown to improve the smoothness of contour following trajectories and execution speed. Also, our own line of research blends into the current approach, because we have shown similar applications using the same sensors, i.e. proximity servoing in preshaping and in haptic exploration with a gripper [4], [5] so far. Here, these results are extended to the end-effector having proximity sensing with a 2D spatial resolution and therefore the ability to perform contour following in 3D. In particular, curvatures along the two main sensor directions can be estimated simultaneously, which is used to derive a compound movement using both curvatures.

The remainder of the paper is structured as follows: in Sec. II we review related work from the field, in Sec. III we discuss the working principle of the sensor, the design of the modules and how to obtain the measurements for the contour following task. Next, in Sec. IV we describe our concept and implementation of contour following and in Sec. V we discuss the results from our experiments. Finally, conclusions are given in Sec. VI.

*This research was funded by the German Research Foundation (DFG) under the grants HE 7000/1-1 and WO 720/43-1.

⁺Authors are with Institute for Anthropomatics and Robotics - Intelligent Process Control and Robotics Lab (IAR-IPR), Karlsruhe Institute of Technology stefan.navarro@kit.edu, stefan.koch@student.kit.edu, bjoern.hein@kit.edu

¹This will likely be the last work that uses this generation of sensors. Recently, we have shown an improved design [3], which want to use for the development of applications from now on.

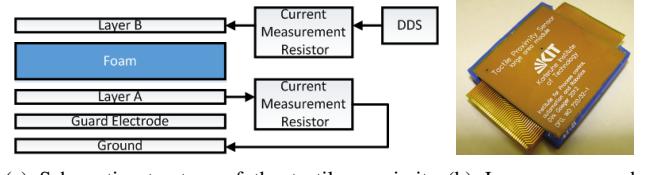
II. RELATED WORK

Regarding proximity sensing principles in the domain of robot-skins, two measuring principles are widely used: infrared ([6], [7], [8] and others) and capacitive sensing ([9], [10], [11] and others). Optical proximity sensors have the advantages of good precision and low circuit complexity; as shown for instance in [6], [8], a considerable spatial resolution can be achieved. Similar to cameras, optical sensing principles are prone to changing illumination, visibility conditions, mirroring effects and translucence. Thus, their use is not always adequate. Capacitive proximity sensors show strong non-linear signal behavior, with dependency on object size, its material and its grounding state. Therefore, the identification of object-sensor-distance is challenging. However, using multiple sensors can help to improve distance estimations. Also, capacitive sensors are insensitive to optical conditions. In a scenario where proximity sensing is used in parallel to vision, capacitive sensing seems to be more appropriate because of the complementary nature of the measuring principles.

In this work we use a self-capacitive system, but mutual-capacitive systems are also suitable for proximity measurements and are even preferable for insulating materials. However, locating the proximity event is more challenging. The recent work of Schlegl et al. [11] concerning obstacle avoidance with capacitive sensors in a mutual-capacitance configuration is focused on showing the sensor's capability of detecting events with low latency and different materials, but does not (yet) constitute a system for 3D contour following or collision avoidance.

A fundamental aspect of achieving dynamic contour following/obstacle avoidance is to estimate the obstacle position and orientation through a sensing system. Cheung and Lumelsky [6], [12] as well as Nunes et al. [13] have both formulated schemes where a tangential plane detection is used to derive a movement that drives parallel to the obstacle surface. Cheung und Lumelsky address the problem in C -space, because in their scenario the surfaces of the robot's links are covered with an IR skin. Nunes et al. address this problem in Cartesian space using an end-effector equipped with an ultrasound sensor. Nunes et al. argue that for contour following a target direction on the obstacle surface must be specified, for instance provided by the commands of a teleoperator. In the case of collision avoidance, this is the to-goal vector (see Fig. 7) projected on the tangent plane, as suggested by Cheung and Lumelsky. In the absence of further planning, the collision avoidance problem is thus reduced to a contour following problem. These early approaches are not concerned with the inclusion of curvature information into contour following schemes.

Curvature estimation has been proposed to improve contour following by providing feed-forward information. Curvatures can be extracted using an external sensing system, e.g. a camera in force-based contour following in the work by Baeten and De Schutter [14], by locally fitting a regression to the past trajectory like Mi and Jia [15] or directly from



(a) Schematic structure of the tactile proximity sensor. (b) Large area module.

Fig. 2. Overview of the tactile proximity sensor module used in this work.

sensor measurements like Walker et al. in [16]. In these works it is shown that the curvature estimation increases the performance of the systems. Still, in all of these cases, curvature information is only 1D. Jia et al. [17] show how to reconstruct and extract the two principal curvatures of a surface by sampling several explored curves lying in different tangential planes. Yet, the acquisition of both curvatures is not simultaneous. Interestingly, direct acquisition of the curvatures has been proposed for tactile perception [18], but not for proximity sensing to our knowledge.

III. SYSTEM DESCRIPTION

A. Sensor Platform

The basic working principle for the tactile proximity sensor illustrated in Fig. 2(a) was first presented in [1]. In proximity mode layer B and the environment are capacitively coupled. The guard electrode is switched to the same potential as layer B by unit amplification to shield it from the ground layer. Objects in proximity will increase the coupling. In order to measure capacitances a direct digital synthesizer (DDS) generates a signal of approx. 100kHz which is routed to the capacitor plates. A simple measuring resistor is used to measure the current flow through the capacitor. The sensor modules, shown in Fig. 2(b) are used in this work were described in detail in the previous work [2]. Each module has a footprint of 40mm×40mm and can detect conducting objects, such as human hands, in the proximity of about 10cm (for insulating materials it's significantly less). Up to 16 modules can be connected together to form a stripe that is connected to one channel on the signal processing board. Therefore, the signal processing board with 10 channels is able to drive up to 10×16 sensor modules. Signal processing in the digital domain, i.e. filtering and DFT-demodulation, is further implemented on an FPGA-board [1], [19]. As seen in Fig. 1 we've used 3 stripes of 16 sensors each, i.e. 48 in total, to cover our cylindrical end-effector. The stripes are the *rows* of our proximity array. Respectively, we talk about *columns* in the perpendicular direction. The read-out rate is approx. 25 fps.

B. Calibration

After turning the sensors on, they must be calibrated to equal output, i.e. the values have to be normalized. The most significant difference is between stripes (channels), so for adequate performance it is sufficient to calibrate representative sensors of each stripe. After this step, the output value for each sensor will be, within some tolerance,

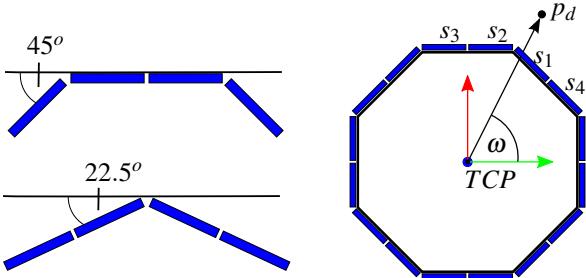


Fig. 3. Calibration for sensor values at different angles w.r.t. the calibration sheet. The angle ω at which the obstacle is detected is used to determine a blending factor between the calibrations.

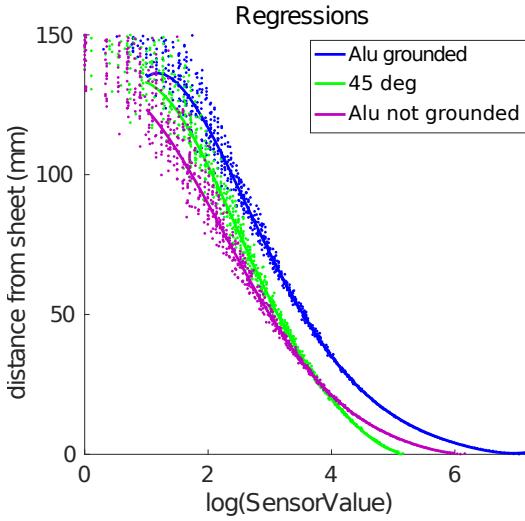


Fig. 4. Exemplary regressions for an aluminum sheet.

the same under equal conditions. The sensors are calibrated to a certain material, which achieves the desired effect of normalization, but also provides means of calculating actual distances for that material. Three materials were used in this work: aluminum (grounded), aluminum (floating) and wood. For the first, a value range of about 1300 units is recorded, for the second 500 units and for the last 50 units.

In addition, due to the octagonal construction of the end-effector, two calibration procedures are executed in order to compensate for the sensitivity of the measurement to orientation between the sensor and the obstacle surface, as shown by Fig. 3. One is with the sensors of one side of the octagon in parallel to the calibration surface (0°) and one is with a relative orientation of 22.5° . From the first collection of values also a calibration for the 45° case can be extracted from the outer sensors.

In Fig. 4 exemplary polynomial regressions are shown for a sheet of grounded aluminum (size: $50\text{cm} \times 60\text{cm}$) for the sensors parallel to the calibration sheet and at 45° . Also, a calibration for the non-grounded aluminum at 0° is shown. The corresponding deviations from the regressions are plotted in Fig. 5. In the case of the grounded aluminum for distances up to 45 mm, the maximum deviations are below 3 mm and the mean deviations are below 1 mm, which

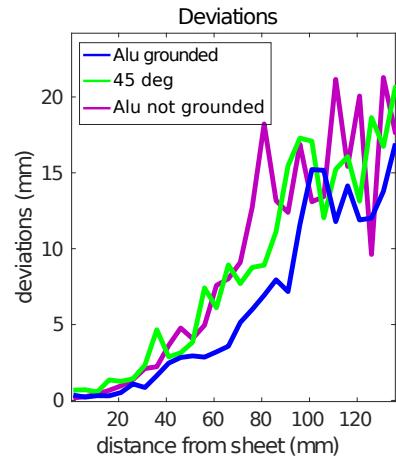


Fig. 5. Deviations of the data from the regressions. Data was collected in bins of size 5 mm, the maximum value of each bin is plotted.

enables the detection of the obstacle with good accuracy.

From the regression plots it is clear that the orientation and the grounding state of the sheet significantly affect the measurements. Also, the size of the sheet itself can affect the measurements. Therefore, a value measured by the sensor can only be matched robustly to a distance value when the orientation of the surface, its size and material are known. This fact precludes a universal and simple scheme of calibration and in our eyes this situation calls for an online calibration/exploration procedure, which estimates said parameters, including distance, using the movement of the sensors (the robot). This is an open problem for capacitive proximity sensing. For now, we use the pre-calculated calibrations and weighting to account for the different orientations.

C. Position Measurements

As a first step, noise reduction techniques are applied to the proximity image. In addition to applying threshold values, spurious pixels are removed using structuring elements. Then, the foreground is converted to cylindrical coordinates w.r.t. the *TCP*, which is defined in the center of the cylinder. The angle ω and height h at which the obstacle is detected (point p_d) is calculated from the center of mass in the foreground. From ω a blending factor between the calibrations at 0° , 22.5° and 45° is established. ω also defines what the “main sensors” are, namely the ones of columns s_1 and s_2 , as shown by Fig. 6. s_1 is the column cut by the vector towards p_d and s_2 is the column closest to this vector without being cut by it. From these, the gradient used for the orientation servoing (see IV-C) is calculated. The distance of p_d is taken as the value of the sensor with the highest reading/lowest distance, i.e. one of $s_{1,i}$, where i is the index of one of the rows of the end-effector. s_3 and s_4 are defined to be the sensor columns left and right of s_1 and s_2 and are used to measure the curvature, as explained in the next subsection.

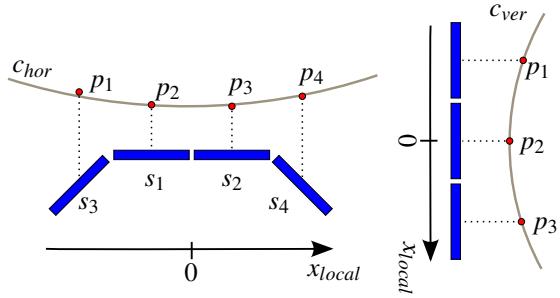


Fig. 6. Fitting of circles to the measurements in order to determine a horizontal and vertical curvature for the obstacle surface.

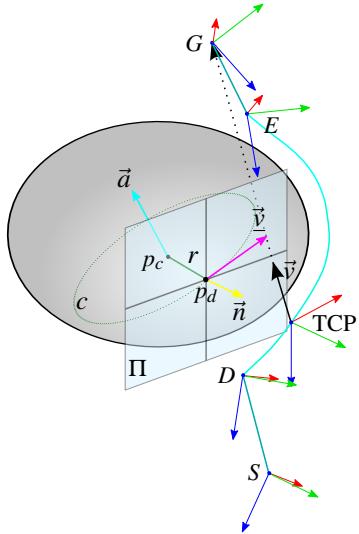


Fig. 7. The four characteristic configurations S, D, E and G for an end-effector in an obstacle avoidance scenario. The to-goal vector is \vec{v} and its projection on the surface tangent plane Π is $\bar{\vec{v}}$. The tangent is detected at a point p_d . The curvature radius of the surface in the direction of $\bar{\vec{v}}$ is r and the center p_c of the circle c . The vector \vec{a} is the normal of the plane, where the circle is defined.

D. Curvature Estimation

We estimate the curvature of the obstacle directly from the measurements, as proposed by [16], using a local coordinate system in the horizontal and vertical direction of the proximity sensor array, as shown in Fig. 6. The blending factor for distance regression is also considered (see Sec. III-C). In these coordinate systems the distance of the points are fitted to circles, c_{hor} and c_{ver} respectively, using the Gauss-Newton optimization method. As a result, we obtain the horizontal and vertical radii r_{hor} and r_{ver} , respectively the curvatures $\kappa_{ver} = \frac{1}{r_{ver}}$ and $\kappa_{hor} = \frac{1}{r_{hor}}$. For the horizontal curvature, only the center stripe of the end-effector is used, for the vertical curvature the mean values of the two main columns, i.e. s_1 and s_2 , are used. It should be noted that the curvature value of small edges cannot be estimated accurately, because of the coarse spatial resolution available.

IV. CONTOUR FOLLOWING

Our concept of proximity sensing-based contour following gets inspiration from the work of Lumelsky and Cheung [12] and Nunes et al. [13] and can be considered to be a hybrid of both. From the first, we use the derivation of the key robot (end-effector) configurations relevant for the task, as shown in Fig. 7. From the first and second, we use the derivation of the to-goal vector and the associated planning of a parallel movement, also shown in Fig. 7. In addition, our concept is similar to the second approach in that it is performed in Cartesian space and uses closed-loop control (proximity servoing) to maintain the desired relative orientation and distance of the end-effector to the obstacle surface. This last aspect matches with our previous work where we used proximity servoing in preshaping [4], [5].

The end-effector configurations shown in Fig. 7 are explained as follows: the start configuration is S , where the end-effector is clear of obstacles, and G is the desired goal configuration. During the task, the end-effector moves on a direct path to G until the sensors detect the presence of an obstacle that blocks this path. The configuration at this moment is called D . From here, the end-effector moves along the object until it reaches the exit point E , where the direct path to the target is no longer blocked.

As an addition to the concept, we propose, like other authors, to measure the curvature of the obstacle in every step with the aim to include this information in the control-loop. Then, the robot can “look ahead” by changing the end-effector pose according to the curvature, which has shown to be more robust and faster than only using the tangents. To perform curvature-based traversal in 3D, we use a simultaneous estimation of horizontal and vertical curvatures, which are blended together, to result in a joint movement along both curvatures, as illustrated by Fig. 8.

Our concept is implemented by two parallel, overlaid controllers that implement the complementary and sometimes redundant strategies to drive the end-effector to the target, while maintaining it free of collision. The strategies are the following and will be explained in the subsections below:

- Navigate along the tangent using the curvature of the obstacle. This strategy is only active, when the end-effector is close enough and sufficiently well aligned to the obstacle.
- Maintain a safety-distance to the obstacle and an orientation parallel to the obstacle tangent plane. This strategy is always active.

A. Curvature-based Movement

As mentioned before, the collision avoidance problem is reduced to a contour-following problem, where the direction of traversal is determined by the vector that points from the current configuration, i.e. the TCP, towards the goal G (see Fig. 7). In other words, in order to bypass the object, we have to determine the normal \vec{n} of the obstacle tangent plane Π and the projection $\bar{\vec{v}}$ of the to-goal vector \vec{v} onto Π to plan increments that will allow the end-effector to skirt the obstacle.

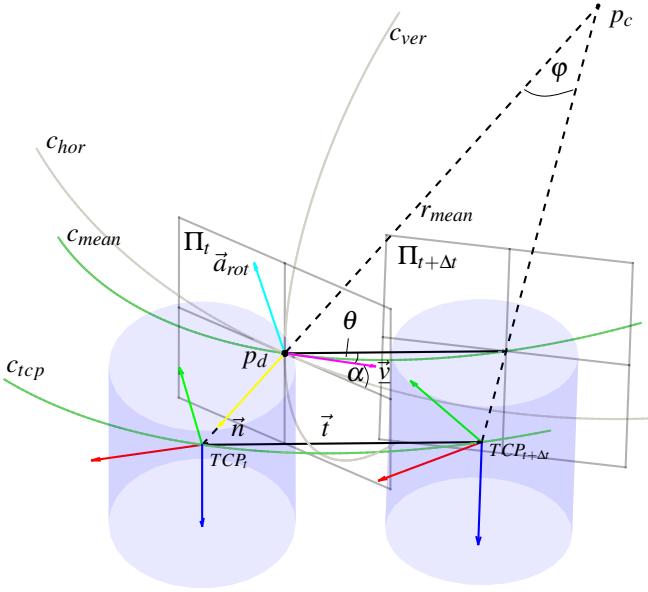


Fig. 8. The end-effector at TCP_t executes a rotational movement according to the two estimated curvatures in order to keep alignment to the obstacle surface when reaching the next pose $TCP_{t+\Delta t}$. In short, it means to rotate the TCP_t around the point p_c and axis \vec{a}_{rot} by the angle φ . The translational part \vec{t} is a chord on the circle c_{tcp} that has a deviation θ from the tangent plane Π_t . The angle α is the elevation of \vec{v} with respect to the xy -plane of the TCP_t -frame and serves as a blending factor to determine the parameters of the mean circle c_{mean} .

Additionally, the curvature on the surface along the direction of \vec{v} determines a circle c_{tcp} in the plane spanned by \vec{n} and \vec{v} (see Fig. 7) can be used to calculate a *shortcut* that is related to the maximum desired speed v_{max} for the end-effector. The shortcut consists of a translational movement, i.e. a chord \vec{t} on c_{tcp} , whose length is related to the desired maximum speed and a change in orientation that aligns the end-effector to its new pose such that it will be parallel to the obstacle again, i.e. a rotation about an axis \vec{a}_{rot} . Fig. 8 illustrates the movement².

In the following, the calculation of the vectors and angles shown in Figs. 7 and 8 is explained. We use the superscript T and R to indicate the translational part or rotational part of a frame, e.g. TCP^T means the translational part of the TCP frame. The tangent plane normal (the surface normal) can be calculated by taking the vector pointing from the detection point p_d towards the end-effector

$$\vec{n} = \frac{TCP_t^T - p_d}{\|TCP_t^T - p_d\|}. \quad (1)$$

The axis \vec{a}_{rot} and \vec{v} respectively are

$$\vec{a}_{rot} = \vec{n} \times \vec{v} \text{ and } \vec{v} = \vec{a}_{rot} \times \vec{n}, \quad (2)$$

the radius of the circle c_{mean} is the inverse of weighted average of the curvatures corresponding to circles c_{hor} and c_{ver}

$$r_{mean} = (\cos^2 \alpha \kappa_{hor} + \sin^2 \alpha \kappa_{ver})^{-1}. \quad (3)$$

²Please note that the end-effector is scaled down significantly in the figures for clarity.

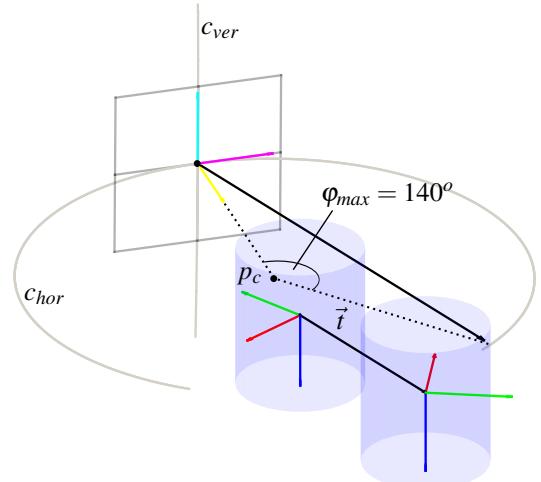


Fig. 9. A special movement has to be executed when the end-effector is farther away from the obstacle than the calculated center p_c of c_{mean} .

The blending factor, i.e. α , is the elevation of \vec{v} with respect to the xy -plane of the TCP_t frame. With r_{mean} it is then possible to calculate r_{tcp} that defines the circle on which the end-effector moves

$$r_{tcp} = r_{mean} + d, \quad (4)$$

where d is the distance to the obstacle. The length l of a chord given a radius r and angle φ can be calculated as

$$l = 2r \sin(\frac{1}{2}\varphi). \quad (5)$$

Since in our case the length of the chord is $l = \|\vec{t}\| = v_{max}\Delta t$, we can solve for the first desired angle φ

$$\varphi = 2 \arcsin(\frac{v_{max}\Delta t}{2r_{tcp}}). \quad (6)$$

Finally, the second desired angle is calculated to be

$$\theta = \frac{1}{2}\varphi \quad (7)$$

by inspection of the triangle constellation. In order to translate the TCP_t frame to $TCP_{t+\Delta t}$, the vector $\vec{u} = \vec{v}l$ is rotated around \vec{a}_{rot} by θ and added to the old frame

$$TCP_{t+\Delta t}^T = TCP_t^T + \vec{t}, \text{ where } \vec{t} = R_{\vec{a}, \theta} \vec{u}. \quad (8)$$

The change in orientation is achieved by rotating the old frame around \vec{a}_{rot} by φ :

$$TCP_{t+\Delta t}^R = R_{\vec{a}, \varphi} TCP_t^R. \quad (9)$$

In order to pass the orientation increments to the robot, we convert them to (roll, pitch, yaw) angles

$$(\text{roll}, \text{pitch}, \text{yaw}) = \text{axangle2rpy}(\vec{a}_{rot}, \varphi). \quad (10)$$

B. Special TCP positions

Fig. 9 shows the case where the current TCP is farther away from the obstacle than the calculated center p_c of c_{mean} . Here, the definition of c_{tcp} doesn't make sense and in these cases $\varphi = \varphi_{max} = 140^\circ$. The vector \vec{t} is parallel to the resulting chord, as shown by Fig. 9.

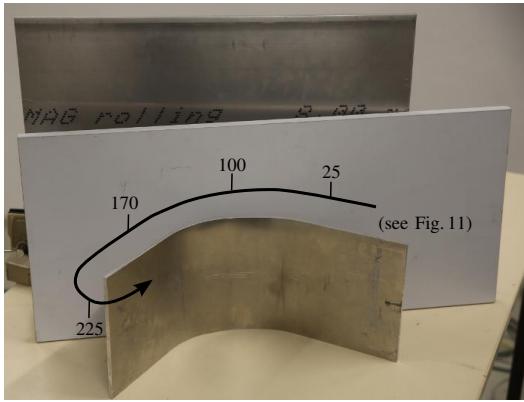


Fig. 10. The obstacles with which we performed experiments: an aluminum sheet, a bent aluminum sheet (“concave” on the inside, “convex” on the outside) and a press wood sheet

C. Proximity Servoing

In our previous work [4], [5], we showed that using a proximity gradient-based closed-loop control (proximity servoing) it was possible to align a gripper endowed with our sensors to preshape to objects. Here, we calculate a *vertical gradient* using the 3×2 arrangement of sensors facing towards the obstacle, i.e. columns s_1 and s_2 . A line is fitted through the mean values, similar to the calculation of the curvature in Sec. III-D (see Fig. 6). The angle between the line and the reference plane of the sensors is the deviation, which is passed to the robot control proportionally in order to adjust that deviation, i.e. a P-controller. A horizontal gradient can also be calculated, but is not used, because a rotation about the TCP’s z-axis is deemed undesirable.

The distance controller only supervises that the end-effector is within some distance range to the obstacle, also applying a proportional factor, i.e. a P-controller, if deviations are present. If collision is imminent or the obstacle has been lost unexpectedly, the other controllers will be halted, until a stable condition is reestablished.

V. EVALUATION

For our evaluation we use the obstacles shown in Fig. 10.

A. Performance of Curvature-based Traversal

In order to check the added value due to curvature detection, we ran experiments in three scenarios, shown in Fig. 11, where the curvature-based motion is switched on (blue path) and off (red path) respectively. When the curvature-based motion is switched off, the system plans movements parallel to the estimated tangent, i.e. using \vec{v} (magenta), and the servoing has to take charge of keeping the desired distance and orientation. The data was recorded for a end-effector speed of 80mm s^{-1} . The target distance was set between 45 and 60 mm.

In Fig. 11(a) it is shown that by the use of curvature information, a movement is performed that avoids getting too close to the obstacle, as seen in the blue path. Without curvature information, the end-effector overshoots, violating the distance constraint, as seen by the dent in the red path.

It is also visible due to the thin line that contour following has been interrupted in order to reestablish a safe distance. For the “convex” shape there is no big difference between using curvature information and not doing so in terms of the continuity of the procedure. Due to the curvature estimation, the blue trajectory has a more consistent distance to the obstacle. In general, the system has difficulties in achieving smooth trajectories around sharp edges, since the measurements are unreliable due to the surface being small compared to that of the sensor (and compared to the calibration sheet). Similar observations as in the cases before can be made for the diagonal paths, where the use of curvature information delivers a more consistent path in term of distance to the obstacle.

In Fig. V-A, radii measurements are shown for the “convex” shape. Fig. 10 relates the values of Fig. V-A to the geometry of the shape. Since the surface is flat in vertical direction, the measurements are high und jump between positive and negative values, as the estimated radius can be a large positive or negative value. In the horizontal case, at the beginning, measurements correspond to the flatter region. The region in the middle shows measurements, with values around 20 cm. The real curvature radii are between 26 cm and 13 cm, meaning the measured values show some agreement with reality, but are not highly accurate. In the last part, corresponding to the edge, the radii measurements indicate around 10 cm, which is the minimum output the system will give. Despite the fact that the sheet has some flat regions, the sensor will detect a smaller curvature, because the sensor arrangement is big compared to the flat part, i.e. the measurement of the outer sensors will already be influenced by the edges.

Here, we have used direct curvature measurements and shown that it is useful as a predictive component. Nonetheless, it is a good idea to use the geometry of the path and of the obstacle points to achieve a more robust curvature estimation. Combined with this, in future approaches, the direct measurements may eventually be used mainly for detecting the trend of the curvature, in order for the system to adapt fastly to changes in the shape. It will be also interesting to see how the proposed approach performs when using other types of proximity sensor, such as optical ones.

B. Sensitivity to Material

The system was tested for the aluminum obstacles in grounded and floating state as well as with a wood sheet. For the first two cases, the performance is robust, even though the sensing range is somewhat reduced for the floating aluminum (see Sec. III-B). Even faster runs at 150mm s^{-1} with the “convex” shape in floating state were completed satisfactorily. The capability of the sensors of detecting a human is even better than that of detecting floating aluminum, because the human has a higher capacitance to ground. In fact, using a multimeter we measured a capacitance coupling to ground of $\sim 0.05\text{nF}$ for the aluminum sheets and $\sim 0.4\text{nF}$ (almost 10 times higher) for a person standing on its two feet. Therefore, the results shown here are promising for safe HRI.

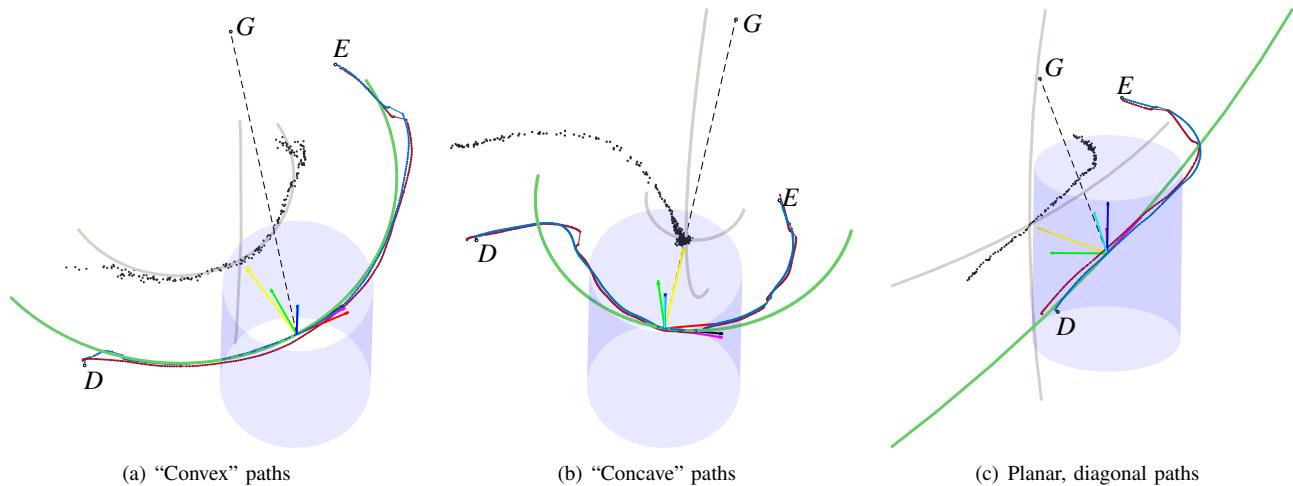


Fig. 11. Paths for the TCP in experiments, where curvature-based traversal was enabled shown in blue and disabled shown in red. (please use the zoom-function of your pdf-viewer to view the details)

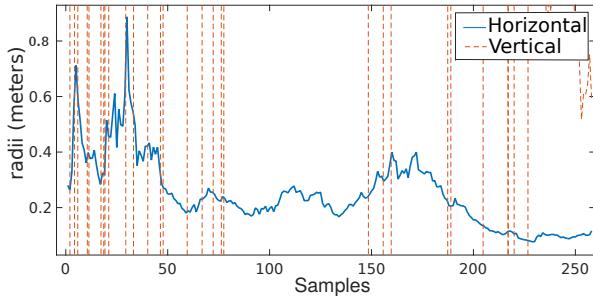


Fig. 12. Measurement data of horizontal and vertical radii for the exterior of the bent aluminum sheet. An approximate correspondence between sample number and the real obstacle feature is provided by Fig. 10.

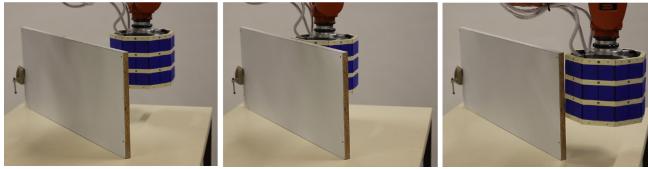


Fig. 13. Sequence of end-effector poses while trying to skirt around a wooden sheet

As stated before, self-capacitive systems struggle in detecting insulating materials and in fact our experiments show that the performance for the wooden obstacle is not satisfactory, as illustrated by Fig. 13. At 60 mm s^{-1} , the system is capable of detecting the obstacle and to follow the flat part of the surface, but is not able to turn around the edge without touching it, as seen by the last frame in Fig. 13.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we've shown a scheme for contour following for a cylindrical end-effector endowed with capacitive proximity sensors. We focused on enabling the system to detect and use 2D curvature information with the aim of obtaining more robust and smoother trajectories. We show that the system performs well under different obstacle geometries and that the curvature information in fact improves the results.

The system had no problems in detecting (metallic) obstacles at velocities up to 150 mm s^{-1} .

Regarding future research, we would like to improve the position and curvature estimation by including information of the end-effector movement and path, as well as the geometry of the obstacle that was already observed. We expect that the detection of features, especially for small ones, such as corners or edges, should be improved. Our current approach now is limited to conducting materials, but we hope to be able to use our most recent sensor design [3] to concurrently implement a mutual-capacitive mode on an end-effector, which will allow us to handle insulating materials as well. It will be also interesting to look at contour following in the scope of haptic exploration tasks.

REFERENCES

- [1] D. Göger, M. Blankertz, and H. Wörn, "A Tactile Proximity Sensor," in *IEEE Sensors 2010*, 2010, pp. 589–594.
- [2] D. Göger, H. Alagi, and H. Wörn, "Tactile Proximity Sensors for Robotic Applications," in *International Conference on Industrial Technology (ICIT)*, 2013.
- [3] H. Alagi, S. Escaida Navarro, M. Mende, and B. Hein, "A Versatile and Modular Capacitive Tactile Proximity Sensor," in *Haptics Symposium 2016*, 2016.
- [4] S. Escaida Navarro, M. Schonert, B. Hein, and H. Wörn, "6D Proximity Servoing for Preshaping and Haptic Exploration using Capacitive Tactile Proximity Sensors," in *Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on*, 2014, pp. 7–14.
- [5] S. Escaida Navarro, F. Heger, F. Putze, T. Beyl, T. Schultz, and B. Hein, "Telemansipulation with Force-based Display of Proximity Fields," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, 2015, pp. 4578–4574.
- [6] E. Cheung and V. Lumelsky, "A sensitive skin system for motion control of robot arm manipulators," *Robotics and Autonomous Systems*, vol. 10, no. 1, pp. 9 – 32, 1992. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/092188909290012N>
- [7] P. Mittendorfer and G. Cheng, "Humanoid multimodal tactile-sensing modules," *Robotics, IEEE Transactions on*, vol. 27, no. 3, pp. 401–410, June 2011.
- [8] K. Terada, Y. Suzuki, H. Hasegawa, S. Sone, A. Ming, M. Ishikawa, and M. Shimojo, "Development of omni-directional and fast-responsive net-structure proximity sensor," in *International Conference on Intelligent Robots and Systems*, 2011.

- [9] Y. Yamada, N. Tsuchida, and M. Ueda, "A proximity-tactile sensor to detect obstacles for a cylindrical arm," *Journal of the Robotics Society of Japan*, vol. 6, no. 4, pp. 292–300, 1988.
- [10] J. Feddema and J. L. Novak, "Whole arm obstacle avoidance for teleoperated robots," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, May 1994, pp. 3303–3309 vol.4.
- [11] T. Schlegl, T. Kroger, A. Gaschler, O. Khatib, and H. Zangl, "Virtual whiskers – highly responsive robot collision avoidance," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Nov 2013, pp. 5373–5379.
- [12] V. J. Lumelsky and E. Cheung, "Real-time collision avoidance in teleoperated whole-sensitive robot arm manipulators," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 23, no. 1, pp. 194–203, Jan 1993.
- [13] U. Nunes, P. Faia, and A. de Almeida, "Sensor-based 3-d autonomous contour-following control," in *Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', IROS '94. Proceedings of the IEEE/RSJ/GI International Conference on*, vol. 1, Sep 1994, pp. 172–179 vol.1.
- [14] J. Baeten and J. De Schutter, "Improving force controlled planar contour following using online eye-in-hand vision based feedforward," in *Advanced Intelligent Mechatronics, 1999. Proceedings. 1999 IEEE/ASME International Conference on*. IEEE, 1999, pp. 902–907.
- [15] L. Mi and Y.-B. Jia, "High precision contour tracking with a joystick sensor," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 1. IEEE, 2004, pp. 804–809.
- [16] S. Walker, K. Loewke, M. Fischer, C. Liu, and J. K. Salisbury, "An optical fiber proximity sensor for haptic exploration," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 473–478.
- [17] Y.-B. Jia, L. Mi, and J. Tian, "Surface Patch Reconstruction via Curve Sampling," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 1371–1377.
- [18] K. Teramoto and K. Watanabe, "Acoustical tactile sensor utilizing multiple reflections for direct curvature measurement," in *SICE 2002. Proceedings of the 41st SICE Annual Conference*, vol. 1. IEEE, 2002, pp. 121–124.
- [19] S. Escaida Navarro, M. Marufo, Y. Ding, S. Puls, D. Göger, B. Hein, and H. Wörn, "Methods for Safe Human-Robot-Interaction Using Capacitive Tactile Proximity Sensors," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 2013, pp. 1149–1154.