2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Macau, China, November 4-8, 2019

With Proximity Servoing towards Safe Human-Robot-Interaction

Yitao Ding , Felix Wilhelm , Leonhard Faulhammer and Ulrike Thomas

bstract— **In this paper, we present a serial kinematic robot manipulator equipped with multimodal proximity sensing modules not only on the T P but distributed on the robot's surface. The combination of close distance proximity information from capacitive and time-of-flight (ToF) measurements allows the robot to perform safe reflex-like and collision-free motions in a changing environment, e.g. where humans and robots share the same workspace. Our methods rely on proximity data and combine different strategies to calculate orthogonal avoidance motions. These motions are instantaneous optimal and are fed directly into the motion controller (proximity servoing). The strategies are prioritized, firstly to avoid collision and then secondly to maintain the task motion if kinematic redundancy is available. The motion is then optimized for avoidance, best manipulability, and smallest end-effector velocity deviation. We compare our methods with common force field based methods.**

I. INTRODUCTION

The proceeding introduction of human-robot-interaction applications in industrial and service sectors requires increasingly high safety and versatility standards. Today's image, tactile and force sensing technologies cover a wide field of perception which is needed to guarantee safety. Even though, a perception gap still exists in the vicinity of the robot, where the camera's sight of view might be occluded and the tactile and force sensing are not able to react due to the lack of contact. Current solutions avoid paths through occluded areas [1] and reduce the velocity of the robot to ensure safety when the robot detects contact [2].

Proximity sensors fill the perception gap by providing information within the pre-touch area. The research has been focusing on the development of these sensors but only simple virtual force field [3] based solutions for collision avoidance at the TCP (see sec. II) were shown. These force fields push the robot away along the normal direction of the obstacle's surface, which is not always a satisfactory solution for collision avoidance. For example, it may be strategically more reasonable to dodge orthogonally to the normal direction, which allows a faster avoidance and reduces the probability to get stuck in a local minima. The sensors provide information for an intermediate fast reacting control layer. Referring to biological reflexes, this control layer only uses near-field information and the robot's current state for the collision avoidance strategies, without further prior planning. Only the knowledge the robot has in the current state where the stimulus occurs is taken into account by the decision process. This can be the direction the manipulator moves best or the deviation from the task trajectory. On the other hand, our methods can be combined with other higher or lower level control schemes, e.g. path planning and impedance control, or used solely where the robot finds its path on its own while avoiding collisions. The algorithm considers motions in the null-space as well as motions when the null-space is lost in order to reduce the deviation from the task motion.

Fig. 1: Proximity sensing cuffs allow the robot to perform avoidance motions.

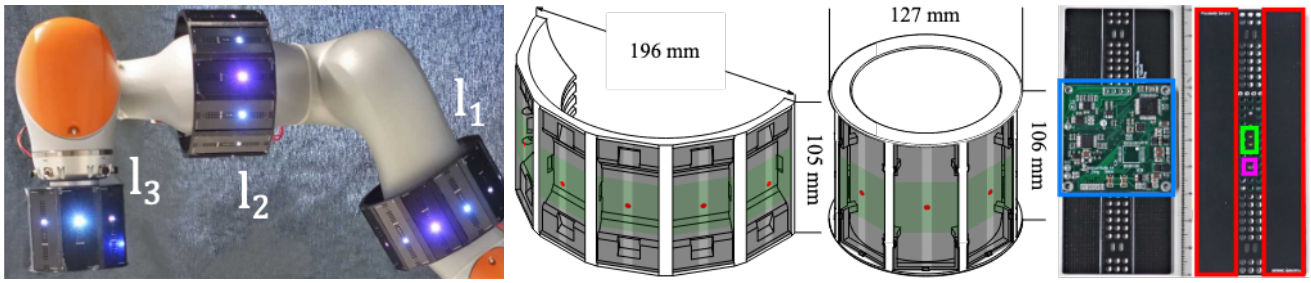
Consequently, our work contributes to the following fields: Firstly, we developed cuffs attached with proximity sensing elements from our prior research based on time-of-flight (ToF) and capacitive measurements. We equipped a redundant 7-DoF robot manipulator with three cuffs, not only on the TCP, but also on the robot's surface. Secondly, the mounted sensors' data allow us to compute instantaneous optimal collision avoidance motions within an orthogonal plane to the obstacle for the complete kinematic chain. Thirdly, we fill the gap between offline path planning and impedance control by providing an intermediate reflex control level. Compared to path planning methods with camera data, the low level proximity servoing loop ensures quick reflex-like reaction in changing environments, especially in a shared human-robot workspace, which results in contactless compliance capabilities. We increase the robot's versatility to react to obstacles and consequently improve safety.

This paper is organized as follows: Section II gives an overview of related work. Section III explains the hardware setup. This is followed by the concept of our collision avoidance strategies using proximity servoing in Section IV. We demonstrate our approach with avoidance experiments in section V and give a conclusion in VI.

II. REL TED WORK

Research in proximity servoing technology has been mainly focused on the sensor technology with rather simple demonstrations of use cases. In [4] and [5] the authors use proximity information to calculate joint torque/velocity

ll authors are with the Lab of Robotics and Human-Machine-Interaction at Chemnitz University of Technology, 09126 SN Chemnitz, Germany. Emails: {yitao.ding, felix.wilhelm, ulrike.thomas}@etit.tu-chemnitz.de, leonhard.faulhammer@s2013.tu-chemnitz.de



(a) (b) (c)

Fig. 2: a) Proximity sensor cuffs (PSCs) positioned on link l_1, l_2 and l_3; b) Design of the PSCs; c) Back- and frontside of the proximity sensor: blue: modular sensor mainboard attached to (black) capacitive electrode pcb, red: capacitive electrodes, green: ToF-sensor, pink: RGB-LED.

control signals which push the robot away along the normal direction of the obstacle's surface. The work in [6] uses a similar approach, but the sensor elements are attached to a single joint within the kinematic chain. The virtual forces created from the sensor signals push the 9-DoF redundant robot within its null-space. However, this approach fails in situations when the sensor is attached to the TCP. The authors in [7] create a compliant end-effector by approaching objects with proximity servoing. In terms of reflex-like motions, the work in [8] generates reflex-like trajectories in real-time with capacitive sensors which let the robot retract from obstacles.

n in-depth examination of proximity servoing based obstacle avoidance is given in [9]. The authors use virtual joint torques to push the robot links away from safety-critical areas in the workspace. To maintain task consistency, these joint space motions are applied to the null-space if a lower link of the robot is affected. Otherwise the task motion is overridden. Our method relies on the generation of an orthogonal vector and uses Cartesian avoidance motions for the null-space method. This allows us to better exploit the null-space, since a upper link motion are affected by the lower links. Furthermore, we introduce a combination of quadratic optimization and sampling based methods with different quality criteria to calculate the avoidance vector.

In another context, the effectiveness of reflex motions in proximity servoing is shown in [10], where a robotic gripper with proximity sensors catches delicate objects with high responsiveness.

The commercially available Bosch P S assistant [11] uses a capacitive sensing skin which detects objects in vicinity, but the system has only deceleration and stopping capabilities. Fogale Robotics' Smart Skin [12] offers similar technology with the addition of collision avoidance and contour following features. However, the underlying methods are not made public and demonstrations only show avoidance motions limited to the robot's TCP. Motions on the lower links are not addressed.

III. H RDW RE DESCRIPTION

. *Hardware rchitecture*

The hardware architecture is shown in Fig. 3. The sensor nodes are connected via C N-Bus to a gateway that manages the sensors on each cuff and serves as an interface to the Ethernet L N. The sensor system communicates with a

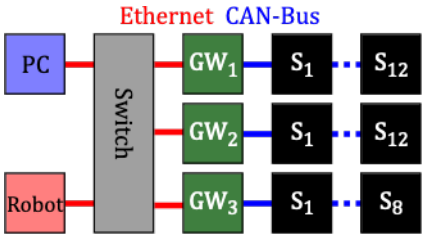


Fig. 3: Hardware architecture diagram.

proprietary protocol that provides data in streaming mode, in triggered mode, or on request. The event triggered data mode (ToF or capacitive measurements reach certain thresholds) reduces data loads on the bus and thus reduces latency, which is ideal for this application. The proximity servoing motion is sent to a 7-DoF robot (Kuka LBR iiwa 7 R800) via a low latency UDP connection with a $50\,\mathrm{Hz}$ update rate. The lower level interpolated robot motion with impedance control is generated within the robot control cabinet with a much higher control rate for smooth motions.

B. *Proximity Sensor Cuff (PSC)*

Three PSCs are mounted on the links 2, 4 and 7 (Fig. 2a) since the robot's kinematic structure can be simplified to three position controllable links l_1, l_2 and l_3. The outer diameter of the PSCs is selected so that the links are completey surrounded by sensor modules to have $360°$ of view. The two lower PSCs' outer diameter is $196\,\mathrm{mm}$ (Fig. 2b) which allows the integration of twelve sensing elements (Fig. 2c). They are made of two half circles that clamp onto the robot. The smaller PSC on the end-effector has eight sensing elements and is made out of a single unit with an outer diameter of $127\,\mathrm{mm}$. The sensing elements are a modification from our prior research [13], which use ToF for far range and capacitance based methods for close range measurements. The modified version uses a single ToF module and has a usable range of $30\,\mathrm{mm}$ to $1000\,\mathrm{mm}$. The now larger capacitive electrodes ($100\,\mathrm{mm} \times 40\,\mathrm{mm}$) cover a wider near field area.

IV. Reflex-Like Collision void nce

The avoidance vector is chosen by sampling based optimization methods (Section IV-B). These methods consider different quality criteria for motions in different Cartesian directions. Each criterion can be directly calculated, such as manipulability and closest obstacles. We feed the avoidance vectors into a quadratic optimization (Section IV-) to calculate the joint motions for the motion controller (Section IV-C). The motion controller decides whether the robot should perform an avoidance motion based on the distance d to the nearest object. It also combines the avoidance motion with a virtual wall depending on d.

. *Quadratic Optimization*

First of all, let $\dot{\mathbf{x}}_1$ be the Cartesian velocity of a given task at the end-effector. The resolved motion $\dot{\mathbf{q}}$ of this task can be computed with the end-effector Jacobian $\mathbf{J}_1$ for a redundant manipulator:

$$\dot{\mathbf{q}} = \mathbf{J}_1^{\dagger}\dot{\mathbf{x}}_1 \tag{1}$$

where † denotes the Moore-Penrose pseudo-inverse and $\mathbf{J}_1^{\dagger}$ solves the optimization constraint $\dot{\mathbf{x}}_1 \quad \mathbf{J}_1\dot{\mathbf{q}} =$ while minimizing $\dot{\mathbf{q}}$. Since (1) directly affects the task motion, it is used for transforming avoidance motions at the TCP in the joint space. With further secondary motions $\dot{\mathbf{q}}_0$, there are additional optimization constraints which can be solved with the common null-space approach [14]

$$\dot{\mathbf{q}} = \mathbf{J}_1^{\dagger}\dot{\mathbf{x}}_1 + \left(\mathbf{I} \quad \mathbf{J}_1^{\dagger}\mathbf{J}_1\right)\dot{\mathbf{q}}_0. \tag{2}$$

However, this solution requires motions in joint space $\dot{\mathbf{q}}_0$ for the lower priority task and in our case avoidance motions $\dot{\mathbf{x}}_2$ are calculated in Cartesian space. $\dot{\mathbf{x}}_2$ can be converted into joint space by using the Moore-Penrose pseudo-inverse $\mathbf{J}_2^{\dagger}$ of the lower priority task Jacobian, but this confines the null-space.

In the following, a quadratic optimization architecture is presented according to Fig. 4. Desirably, the robot executes a Cartesian avoidance motion $\dot{\mathbf{x}}_2$ while maintaining $\dot{\mathbf{x}}_1$ with minimal $\dot{\mathbf{q}}$. The quadratic cost function can then be expressed as follows:

$$\begin{aligned} g_1(\dot{\mathbf{q}}, \ ) = \frac{\mu}{2}\dot{\mathbf{q}}^T\dot{\mathbf{q}} + \frac{1}{2}\left(\dot{\mathbf{x}}_2 \quad \mathbf{J}_2\dot{\mathbf{q}}\right)^T\left(\dot{\mathbf{x}}_2 \quad \mathbf{J}_2\dot{\mathbf{q}}\right) \\ + \ ^T\left(\dot{\mathbf{x}}_1 \quad \mathbf{J}_1\dot{\mathbf{q}}\right) \end{aligned} \tag{3}$$

where μ is a weighting factor.
The common equation form for quadratic programming is as follows and its solution is giving by solving a linear system (5):

$$\min \ \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} + d \tag{4}$$

$$\text{s.t. } \mathbf{G}\mathbf{x} = \mathbf{h}$$

$$\begin{bmatrix}\mathbf{Q} & \mathbf{G}^T\\ \mathbf{G} & \end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \end{bmatrix} = \begin{bmatrix}\mathbf{c}\\ \mathbf{h}\end{bmatrix}. \tag{5}$$

(note: different variable notation)

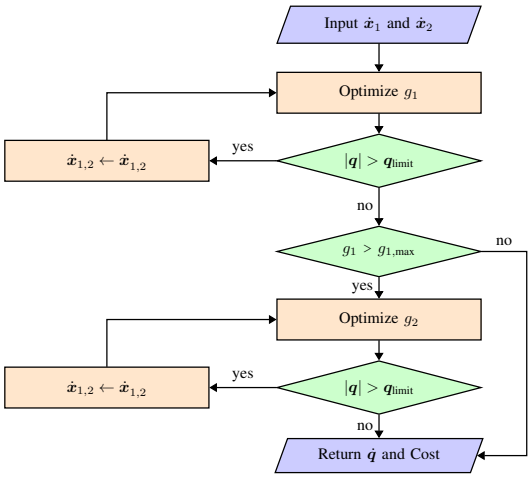


Fig. 4: Optimization structure.

pplied on (3) we get the following solution for $\dot{\mathbf{q}}$:

$$\begin{aligned} g_1(\dot{\mathbf{q}}, \ ) = \frac{1}{2}\dot{\mathbf{q}}^T\left(\mathbf{J}_2^T\mathbf{J}_2 + \mu\mathbf{I}\right)\dot{\mathbf{q}} \quad \dot{\mathbf{x}}_2^T\mathbf{J}_2\dot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{x}}_2^T\dot{\mathbf{x}}_2 \\ + \ ^T\left(\dot{\mathbf{x}}_1 \quad \mathbf{J}_1\dot{\mathbf{q}}\right) \end{aligned} \tag{6}$$

$$\Rightarrow \begin{bmatrix}\mathbf{J}_2^T\mathbf{J}_2 + \mu\mathbf{I} & \mathbf{J}_1^T\\ \mathbf{J}_1 & \end{bmatrix}\begin{bmatrix}\dot{\mathbf{q}}\\ \end{bmatrix} = \begin{bmatrix}\dot{\mathbf{x}}_2^T\mathbf{J}_2\\ \dot{\mathbf{x}}_1\end{bmatrix} \tag{7}$$

$$\begin{bmatrix}\dot{\mathbf{q}}\\ \end{bmatrix} = \begin{bmatrix}\mathbf{J}_2^T\mathbf{J}_2 + \mu\mathbf{I} & \mathbf{J}_1^T\\ \mathbf{J}_1 & \end{bmatrix}^{1}\begin{bmatrix}\dot{\mathbf{x}}_2^T\mathbf{J}_2\\ \dot{\mathbf{x}}_1\end{bmatrix}. \tag{8}$$

(3) is used to maintain task consistency $\dot{\mathbf{x}}_1$ by allowing a certain deviation of the avoidance motion $\dot{\mathbf{x}}_2$. If the deviation $\dot{\mathbf{x}}_2 \quad \mathbf{J}_2\dot{\mathbf{q}}$ is too large, the task motion is overridden and becomes a minimization problem of:

$$\begin{aligned} g_2(\dot{\mathbf{q}}, \ ) = \frac{\mu}{2}\dot{\mathbf{q}}^T\dot{\mathbf{q}} + \frac{1}{2}\left(\dot{\mathbf{x}}_1 \quad \mathbf{J}_1\dot{\mathbf{q}}\right)^T\left(\dot{\mathbf{x}}_1 \quad \mathbf{J}_1\dot{\mathbf{q}}\right) \\ + \ ^T\left(\dot{\mathbf{x}}_2 \quad \mathbf{J}_2\dot{\mathbf{q}}\right) \end{aligned} \tag{9}$$

$$\begin{bmatrix}\dot{\mathbf{q}}\\ \end{bmatrix} = \begin{bmatrix}\mathbf{J}_1^T\mathbf{J}_1 + \mu\mathbf{I} & \mathbf{J}_2^T\\ \mathbf{J}_2 & \end{bmatrix}^{1}\begin{bmatrix}\dot{\mathbf{x}}_1^T\mathbf{J}_1\\ \dot{\mathbf{x}}_2\end{bmatrix}. \tag{10}$$

dditionally, joint limits represent inequality constraints which disturb the task or avoidance trajectory $\dot{\mathbf{x}}_{1,2}$. If the set is active then motions from joints which move away from their range limits $\dot{\mathbf{q}}^{\mathrm{c}}$ must be compensated by the remaining joints $\dot{\mathbf{q}}'$:

$$\dot{\mathbf{x}}_{1,2} \quad \mathbf{J}'_{1,2}\dot{\mathbf{q}}' \quad \mathbf{J}^{\mathrm{c}}_{1,2}\dot{\mathbf{q}}^{\mathrm{c}} = 0. \tag{11}$$

new task/avoidance motion $\dot{\mathbf{x}}$ replaces $\dot{\mathbf{x}}_{1,2}$ in the equality constraint of (3) and (9):

$$\dot{\mathbf{x}}_{1,2} = \dot{\mathbf{x}}_{1,2} \quad \mathbf{J}^{\mathrm{c}}_{1,2}\dot{\mathbf{q}}^{\mathrm{c}} = \mathbf{J}'_{1,2}\dot{\mathbf{q}}'. \tag{12}$$

B. Sampling based optimization

Let $\mathcal{F}_{\mathrm{c}}$ define the collision frame with the rotation matrix $\mathcal{R}_{\mathrm{c}} = [\mathbf{x}_{\mathrm{c}}\,\mathbf{y}_{\mathrm{c}}\,\mathbf{z}_{\mathrm{c}}]$ and the position vector $\mathbf{p}_{\mathrm{c}}$, which is the point on the robot closest to the obstacle. The collision direction vector $\mathbf{z}_{\mathrm{c}}$ points from $\mathbf{p}_{\mathrm{c}}$ to the closest obstacle with distance d. The sampling direction vector $\mathbf{x}_{\mathrm{c}}$ is within an orthogonal plane of $\mathbf{z}_{\mathrm{c}}$ and rotates (with $\mathbf{y}_{\mathrm{c}}$) around $\mathbf{z}_{\mathrm{c}}$. This allows the generation of orthogonal motion directions. We sample the quality criteria around the plane with respect

to Euclidean distances Q_{dist}, manipulability Q_{man}, deviation from the current motion vector Q_{dir}, and aforementioned costs from the quadratic optimization Q_{opt}. gainst common convention, the quality criterion Q ranges from 0 to 1 with 1 being the best value, forming a maximization problem. Because in addition to a more intuitive visualization of the formed direction map, constraints can be applied by simple multiplication.

To obtain Q_{dist}, a bounding cylinder in the direction of $\mathbf{x}_{\text{c}}$ is formed with a set radius and length allowing only obstacles within this cylinder to be processed. The distance of the closest obstacles to the cylinder's center rates Q_{dist} for each sampling direction. fter sampling, the quality function is normalized to the largest value.

The Eigenvalues of the manipulability ellipsoid [15] describe the amount of manipulability in the direction of the Eigenvectors $\mathbf{v}$. It is desirable to move the robot to high manipulability areas to avoid singularities. Therefore Q_{man} represents the deviation of $\mathbf{x}_{\text{c}}$ to Eigenvectors $\mathbf{v}$ with largest $_{\text{max}}$ and second largest Eigenvalues $_{\text{mid}}$. With a transformation of $\mathbf{v}$ to frame $\mathcal{F}_{\text{c}}$

$$\mathbf{v}_{\text{c}} = \mathcal{R}_{\text{c}}^{T}\mathbf{v} \tag{13}$$

the deviation of $\mathbf{x}_{\text{c}}$ with respect to $\mathbf{v}$ is its x component $v_{\text{c,x}}$.

$$Q_{\text{man}} = \frac{_{\text{max}}\left|v_{\text{c,max,x}}\right| + _{\text{mid}}\left|v_{\text{c,mid,x}}\right|}{_{\text{max}} + _{\text{mid}}} \tag{14}$$

In the same way Q_{dir} is calculated with $\mathbf{m}$ being the normalized current motion direction to avoid large direction changes during obstacle avoidance.

$$\mathbf{m}_{\text{c}} = \mathcal{R}_{\text{c}}^{T}\mathbf{m} \tag{15}$$

$$Q_{\text{dir}} = \frac{m_{\text{c,x}} + 1}{2} \tag{16}$$

Because Q_{dist} and Q_{man} are constraints, a multiplication to the overall quality criteria act as boolean intersection. n offset b is required to pass through the constraints in case the quality criteria is zero.

$$Q = Q_{\text{dist}}Q_{\text{man}}\left(b + Q_{\text{dir}} + Q_{\text{opt}}\right) \tag{17}$$

The sampling vector $\mathbf{x}_{\text{c}}$ and its particular joint motions from the quadratic optimization with largest Q is chosen as avoidance motion.

C. *voidance Motion Control*

The avoidance motion control is an intermediate control level and is located above the robot control cabinet. It uses velocity control mode and sends corrected joint velocity values to the control cabinet in which the lower high frequency control takes place with acceleration and jerk controls. The architecture of the avoidance motion control is shown in Fig. 5. First, the ToF and capacitive measurements are transformed into a pointcloud in the base frame (Fig. 6). Capacitive data represent points with fixed ranges that are very close to the robot due to the fact that capacitive sensors don't provide absolute distances. Then, self measurements are removed by converting the pointcloud to the cuff's

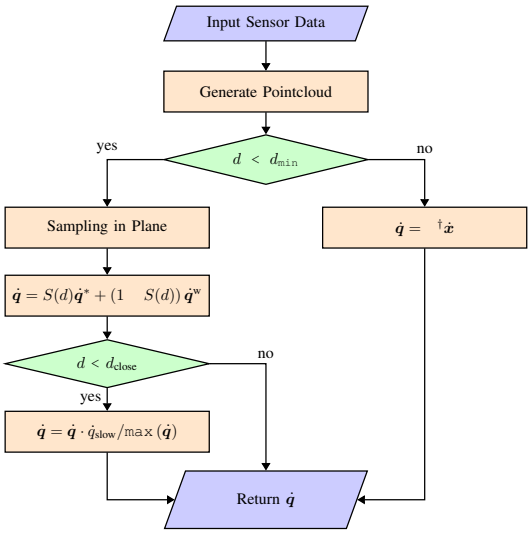


Fig. 5: voidance motion architecture.

function POINTCLOUD(s) s: sensor data
  for $i = 1 : n_{\texttt{Sensor}}$ **do**
    P $:, i) = T_{\texttt{uff}} \cdot T_{\texttt{Sensor}}$ $:, :, i) \cdot s$ $i)$
  end for
  for $i = 1 : n_{\text{Cuff}}$ **do**
    P $_{\texttt{uff,i}} = \texttt{inv}\ T_{\texttt{uff,i}}) \cdot$ P
    id $= \texttt{vecnorm}$ P $_{\texttt{uff,i}}$ $1 : 2, :)) >$ R
    $\texttt{id}_2 =$ P $_{\texttt{uff,i}}$ $3, :) >$ L $/2$
    $\texttt{id}_3 =$ P $_{\texttt{uff,i}}$ $3, :) <$ L $/2$
    P $= T_{\texttt{uff,i}} \cdot$ P $_{\texttt{uff,i}}$ $:, \texttt{id}$ &$\texttt{id}_2$&$\texttt{id}_3$)
  end for
  return P
end function

Fig. 6: Pointcloud creation out of sensor data from a single PSC and removal of self measurements. R is the radius and L the length of the cylinder.

coordinate system in which all points inside a bounding cylinder are removed.

We store the remaining pointcloud in a small buffer with a certain depth which allows the controller to include past obstacle points that are no longer seen but still important during the avoidance motion. If the closest distance d is smaller than the threshold d_{min} the avoidance motion control takes over. Sampling for the lowest cost criteria in the avoidance plane provides the joint motions $\dot{\mathbf{q}}^{*}$ from the quadratic optimization steps. To generate the reactive motion which the manipulator finally executes, the $\dot{\mathbf{q}}^{*}$ is applied for far distances whereas a motion along a virtual wall is used for short distances. The Sigmoid function S combines both methods, where a changes the shape of the S-curve and c sets its center position.

$$\dot{\mathbf{q}}^{\text{w}} = \quad \mathbf{J}^{\dagger}\mathbf{z}_{\text{c}}(d_{\text{min}} \quad d) \quad t \tag{18}$$

$$S(x) = \frac{1}{1 + e^{\ a(x \quad c)}} \tag{19}$$

$$\dot{\mathbf{q}} = S(d)\dot{\mathbf{q}}^{*} + (1 \quad S(d))\,\dot{\mathbf{q}}^{\text{w}} \tag{20}$$

t last, too close obstacles, such as from capacitive measurements, decelerate the robot by limiting the maximum joint velocity to $\dot{q}_{\text{slow}}$.

V. EXPERIMENTS ND RESULTS

In this section, the results from different approaching scenarios on the real robot are shown. n obstacle is placed next to different links and the corresponding avoidance direction and joint motion is calculated according to the quality criteria. In the end, we compare our method with virtual force fields.

. *Experimental Parameters*

In the experiments, the distance d_{min} is set to $250\,\text{mm}$ and we sample twelve times around the $\mathbf{z}_{\text{c}}$. The length and diameter of the cylinder around $\mathbf{x}_{\text{c}}$ are $250\,\text{mm}$ and $200\,\text{mm}$.

B. voidance Motion Results

We start with the scenario in Fig. 1, where a human hand approaches the end-effector. The robot's proximity perception is shown in Fig. 7. The hand and the human is

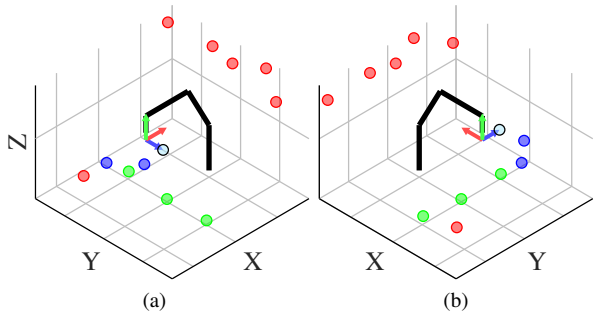


Fig. 7: Pointcloud from Fig. 1 in two different viewing angles. Light blue dot: nearest obstacle with the blue collision vector $\boldsymbol{z}_{\text{c}}$ pointing to it. Red dots (cuff1): mostly measurements from a curtain behind the robot; green dots (cuff2): table; blue dots (cuff3): human body and hand.

mostly detected by the cuff at the end-effector. Because the hand is the closest obstacle within d_{min}, the collision vector $\mathbf{z}_{\text{c}}$ (blue arrow) points directly to it. In Fig. 8a the sampling results on the orthogonal plane (red and green arrow) of $\mathbf{z}_{\text{c}}$ are shown for Q_{dir}, Q_{man} and Q_{dist}. The human body located in front of the manipulator reduces the values in the quality map around angle π. Fig. 8b shows the combination with (17). The avoidance direction corresponds to the maximum quality value and points in this case in negative x-direction in the world frame. Since an avoidance motion of the end-effector does not contain a null-space, the joint velocity is calculated with (1) while taking (12) into account.

In (Fig. 9), the obstacle is located at link 1. In addition to the previous scenario, the polar plot of the quality map factors in Q_{opt}. In this case, g_1 is optimized since the end-effector has enough joints left, and therefore enough redundancy to execute the avoidance motion while maintaining the task. In Fig. 10 link 2 is affected by an obstacle. Being closer to the end-effector, the task motion and avoidance motion are closely coupled. This leads to a larger error in g_1 and the optimization algorithm switches to the solution of g_2 for Q_{opt}. In the last experiment, we compare our algorithm with a force field based virtual wall around the obstacle. The robot

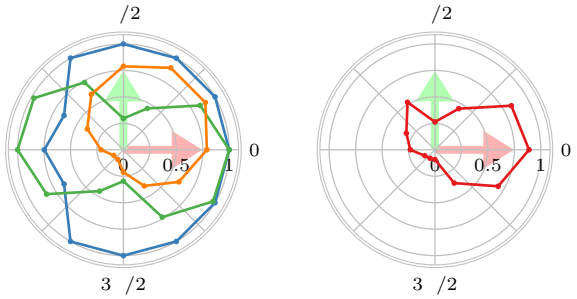


Fig. 8: a) Individual quality criteria - orange: Q_{dir}, green: Q_{man}, blue: Q_{dist}. b) Overall quality map Q of Fig. 7.

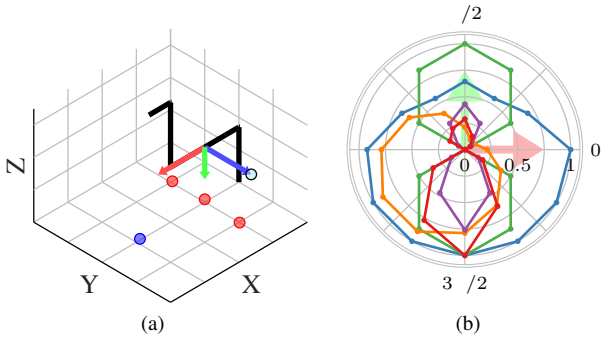


Fig. 9: Obstacle next to link l_1. Violet: Q_{opt}, otherwise same colors as an Fig. 8.

tries to move from the start position to the target position and back with an obstacle blocking the direct path at the TCP and lower link (Fig. 11). With the virtual wall, the manipulator successfully avoids collisions but gets stuck in front of the obstacle in the beginning (Fig. 12). This resembles a local minimum, where the target's pulling force is compensated by the virtual wall. In contrast, the reflex-like avoidance method calculates an orthogonal a fast reacting avoidance motion and the end-effector successfully reaches the target. With the orthogonal motion the deviation from the direct path is larger and the distance may fall below d_{min} but is limited by (20). The sensor cuff loses sight to the obstacle in Fig. 12a due to the small spatial resolution of the narrow laser beam of ToF sensors but it recovers the sight when the obstacle gets closer again. This also shows the importance of the wide area coverage capabilities of capacitive sensing. However, the sight loss for larger distances generate jerky avoidance motions (Fig. 12b). lower level jerk control is

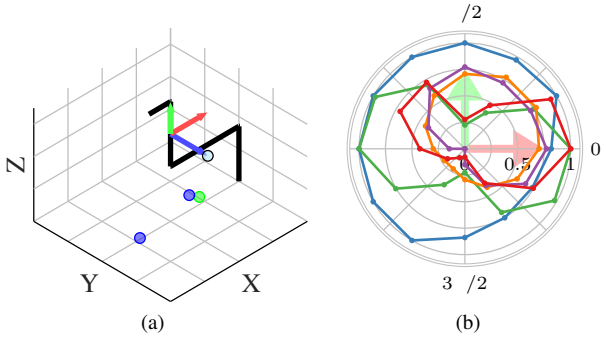
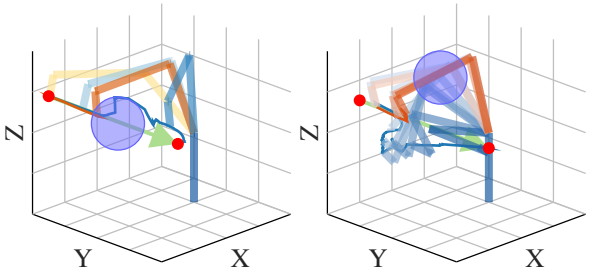


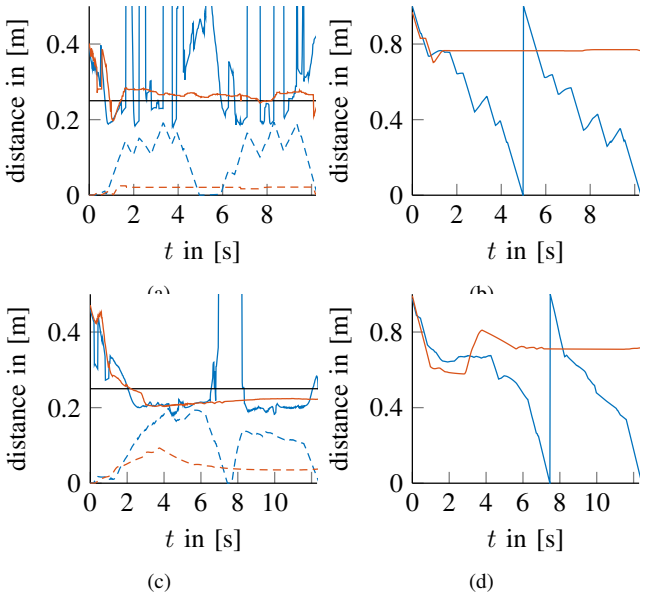
Fig. 10: Obstacle next to link l_2.

out of scope of this paper since we focus on the effectiveness of the avoidance strategy but a transition function which softens the hard transition step can reduce the jerkiness.



(a) (b)

Fig. 11: Comparison of the proposed avoidance algorithm (blue) with the potential field method based on a virtual wall (orange). The robot moves from the start position to the target position with an obstacle blocking the direct path.



(c) (d)

Fig. 12: Orange: potential field; blue: avoidance motion; left solid: distance to closest obstacle; left dashed: deviation from direct path; right: distance to target position.

VI. CONCLUSION

In this paper, we present proximity servoing methods which allow the robot not only to avoid collisions but also to move around obstacles with orthogonal motions. The motions are instantaneous optimal with respect to several criteria. The robot is able to reach the target without getting stuck in a local minimum. Furthermore, the avoidance motion attempts to maintain the task motion as long as the collision avoidance motion permits. These properties of our methods are well suited for safe human-robot-interaction applications. Compared to data from 3D depth cameras, our approach is able to cover most of the relevant workspace area with minimal points. Every single measurement contributes significantly to the environmental perception. On the other hand the the narrow laser beam of ToF sensing limits the spatial resolution in which the system loses sight but the combination with wide area capacitive sensing covers this perception gap. Furthermore a transition algorithm is required for later implementation for the reduction of jerky motions during the transition phase between normal operation and avoidance motion. Due to the implementation layer in the control architecture, our methods can be effortlessly combined with existing lower and higher level controllers. Depending on their needs, the proposed quality criteria can be adapted and extended to specific applications.

REFERENCES

[1] F. Flacco, T. Kröger, . De Luca, and O. Khatib, " depth space approach to human-robot collision avoidance," in *2012 IEEE International Conference on Robotics and utomation*, May 2012, pp. 338–345.

[2] S. Haddadin, . lbu-Schaffer, . De Luca, and G. Hirzinger, "Collision detection and reaction: contribution to safe physical human-robot interaction," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2008, pp. 3356–3363.

[3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[4] F. Bergner, E. Dean-Leon, and G. Cheng, "Efficient distributed torque computation for large scale robot skin," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018, pp. 1593–1599.

[5] L. Bascetta, G. Magnani, P. Rocco, R. Migliorini, and M. Pelagatti, " nti-collision systems for robotic applications based on laser time-of-flight sensors," in *2010 IEEE/ SME International Conference on dvanced Intelligent Mechatronics*, July 2010, pp. 278–284.

[6] M. Brandstötter, S. Mühlbacher-Karrer, D. Schett, and H. Zangl, "Virtual compliance control of a kinematically redundant serial manipulator with 9 dof," in *dvances in Robot Design and Intelligent Control*, . Rodić and T. Borangiu, Eds. Cham: Springer International Publishing, 2017, pp. 38–46.

[7] D. Nakhaeinia, P. Laferrière, P. Payeur, and R. Laganière, "Safe close-proximity and physical human-robot interaction using industrial robots," in *2015 12th Conference on Computer and Robot Vision*, June 2015, pp. 237–244.

[8] T. Schlegl, T. Kröger, . Gaschler, O. Khatib, and H. Zangl, "Virtual whiskers - highly responsive robot collision avoidance," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 5373–5379.

[9] G. Buizza vanzini, N. M. Ceriani, . M. Zanchettin, P. Rocco, and L. Bascetta, "Safety control of industrial robots based on a distributed distance sensor," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 6, pp. 2127–2140, Nov 2014.

[10] K. Sasaki, K. Koyama, . Ming, M. Shimojo, R. Plateaux, and J. Choley, "Robotic grasping using proximity sensors for detecting both target object and support surface," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018, pp. 2925–2932.

[11] Robert Bosch Manufacturing Solutions GmbH, *P S ssistant - Contact-Free Human-Robot Collaboration*, https://www.bosch-apas.com/media/en/apas/downloads/2017_12_01_apas_assistant_doppelseite_de.pdf, 2017, online: accessed 1-March-2019.

[12] FOG LE robotics, *Sensitive Surfaces for Human/Robot interaction & cooperation*, http://www.fogale-robotics.com/pdf/Sensitive_surfaces.pdf, online: accessed 1-March-2019.

[13] Y. Ding, H. Zhang, and U. Thomas, "Capacitive proximity sensor skin for contactless material detection," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018, pp. 7179–7184.

[14] B. Siciliano and J. . E. Slotine, " general framework for managing multiple tasks in highly redundant robotic systems," in *Fifth International Conference on dvanced Robotics 'Robots in Unstructured Environments*, June 1991, pp. 1211–1216 vol.2.

[15] T. Yoshikawa, "Dynamic manipulability of robot manipulators," in *Proceedings. 1985 IEEE International Conference on Robotics and utomation*, vol. 2, March 1985, pp. 1033–1038.