Collision Avoidance with Proximity Servoing for Redundant Serial Robot Manipulators

Conference Paper · May 2020

DOI: 10.1109/ICRA40945.2020.9196759

2 authors:

Yitao Ding
Technische Universität Chemnitz
15 PUBLICATIONS **139** CITATIONS
SEE PROFILE

Ulrike Thomas
Technische Universität Chemnitz
51 PUBLICATIONS **406** CITATIONS
SEE PROFILE

Collision voidance with Proximity Servoing for Redundant Serial Robot Manipulators

Yitao Ding and Ulrike Thomas

***bstract*— Collision avoidance is a key technology towards safe human-robot interaction, especially on-line and fast-reacting motions are required. Skins with proximity sensors mounted on the robot's outer shell provide an interesting approach to occlusion-free and low-latency perception. However, collision avoidance algorithms which make extensive use of these properties for fast-reacting motions have not yet been fully investigated. We present an improved collision avoidance algorithm for proximity sensing skins by formulating a quadratic optimization problem with inequality constraints to compute instantaneous optimal joint velocities. Compared to common repulsive force methods, our algorithm confines the approach velocity to obstacles and keeps motions pointing away from obstacles unrestricted. Since with repulsive motions the robot only moves in one direction, opposite to obstacles, our approach has better exploitation of the redundancy space to maintain the task motion and gets stuck less likely in local minima. Furthermore, our method incorporates an active behaviour for avoiding obstacles and evaluates all potentially colliding obstacles for the whole arm, rather than just the single nearest obstacle. We demonstrate the effectiveness of our method with simulations and on real robot manipulators in comparison with commonly used repulsive force methods and our prior proposed approach.**

I. Introduction

Recently, sensing technology for proximity skins has reached a technically mature level usable for safety applications in robotics. The technology complements existing vision and force-sensing safety layers by bridging the perception gap in the vicinity of the robot and providing low-latency perception. These capabilities are particularly important in workspaces shared by humans and robots where fast-reacting collision avoidance is required for increased safety. External vision systems suffer from occlusion and force sensing methods only react after an impact has already occurred. nother advantage of proximity sensing skins lies in the nature of proximity measurements. Close obstacles are represented with a minimal amount of data (Fig. 1). In comparison, 3D cameras transmit a large amount of point cloud data of which only a small percentage are necessary. The high information density of proximity measurements reduces data processing efforts and enables low-latency perception of environments. In other words, proximity sensing skins lose spatial resolution in favour of less but more meaningful data and higher responsiveness. Unlike fixed camera systems, the spatial resolution increases with approaching obstacles and the reduction of latency is relevant for fast-reacting collision avoidance motions.

The authors are with the Lab of Robotics and Human-Machine-Interaction at Chemnitz University of Technology, 09126 SN Chemnitz, Germany. Emails: yit o.ding, ulrike.thom s}@etit.tu-chemnitz.de

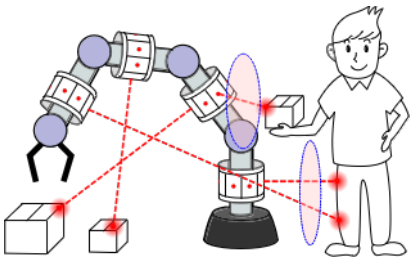


Fig. 1: Collision avoidance with proximity servoing in a shared workspace scenario. Only minimal amount of data is required for occlusion-free perception of obstacles. Our method limits the approach velocity illustrated as circles.

s discussed in the early work of Cheung et. al [1], three major components have to be addressed for such systems, which are perception hardware, the sensor's signal processing, and robot control algorithms. The authors also point out that proximity servoing is more suitable for reactive motion generation (local path planning) that complements global path planning, since only incomplete information of the changing environment is available. While research focuses on the development of sensing and signal processing technology, collision avoidance algorithms which make extensive use of aforementioned properties for fast-reacting motions have not yet been fully investigated. Many reactive collision avoidance methods with proximity servoing rely on repulsive motions [2], [3], which have a passive avoidance behaviour by moving in the opposite direction to obstacles. These methods are robust with respect to the avoidance task, but are not able to actively move around an obstacle. Furthermore, they do not exploit all feasible motions in the redundancy space to maintain the main task motion.

In this paper, we extend our previous work [4] in which we investigated tangential avoidance strategies for proximity servoing. In the prior two-step approach, sampling based optimization determines the tangential avoidance vector with respect to certain cost criteria for an active avoidance motion. Quadratic optimization keeps the motion towards these vectors in the redundancy space if feasible. We propose a general approach in this work through the definition of the avoidance task as quadratic program with linear constraints (Fig. 2). Common solvers for quadratic objective functions are available, such as in M TL B, to solve this problem. The space of feasible motions increases by confining the approach velocity to the obstacle with inequality constraints, instead of only limiting them within a tangential plane or

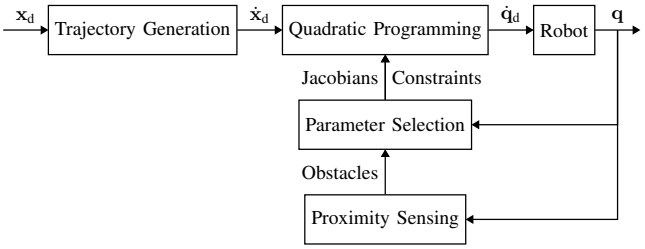


Fig. 2: Inverse differential kinematics with collision avoidance based on constrained quadratic programming.

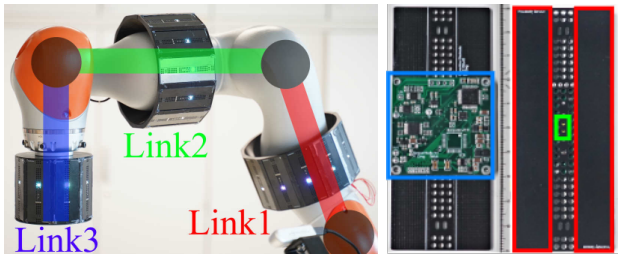
defining a repulsive motion in the opposite direction. The maximum approach velocity changes dynamically. It is firstly limited for far distances and reduces to zero until the obstacle is too close. Through the early correction motions and the smaller motion limitations, the avoidance motion creates less disturbance on the robot's main task, as it makes full use of the redundancy space and the solution is optimized for all potential obstacles. further constraint realizes an active avoidance behaviour by increasing the obstacle distances, which allows the robot to move around an obstacle. In fact, our collision avoidance approach is applicable on other sensing systems, such as point clouds from 3D cameras. However, better solutions, such as global path planning, already exist for these systems, where more and complete spatial information is available. Our method specifically handles data where such information is unavailable and where fast instantaneous reaction capabilities without planning are required.

The contributions of this paper are: a general optimization formulation for fast-reacting collision avoidance with proximity servoing, the introduction of an active avoidance behaviour, and the consideration of all obstacles for the complete kinematic chain in the optimization step.

II. Rel ted Work

Common collision avoidance approaches with proximity sensing skins rely on simple repulsive motions. Besides of our previous work [4], only a limited number of approaches for collision avoidance with proximity servoing are not based on repulsive motions or projections in the null-space [5]. In [6] the authors take advantage of the higher spatial resolution of close obstacles to estimate the contour with proximity sensors attached on a cylindrical end-effector. The end-effector follows the contour for collision avoidance. Though, only end-effector motions controlled in Cartesian space are shown and the contour estimation requires higher spatial information, thus small distances or large obstacles. Fogale Robotics [7] offers a commercial available robotic skin with collision avoidance capabilities. The underlying principles are not published but demonstrations show repulsive motion on the lower links and active avoidance on the end-effector. In [8] a predefined reflex behaviour generates retraction motions when an obstacle is sensed by capacitive sensors. The retraction motion overrides the main task motion and redundancies of the manipulator are not exploited. In [9] and [10] the authors present a complex proximity sensing system with a large amount of optical sensing elements covering the complete manipulator. They define danger-fields [11] and move the robot away from them. The authors in [12] demonstrate a constrained based optimization algorithm for collision avoidance similar to our approach. Their approach limits only the approach velocity, which by itself, behaves passively similar to a virtual wall on which the robot slides. In a similar work [13] from another group, these methods are applied on proximity skins. Their sensing system comprises of a large number of sensing elements with high spatial resolution, but only with near field sensing capabilities of up to $30\,\mathrm{mm}$. They propose several additional constraints, such as for trajectory and velocity following. With the higher spatial resolution in close proximity, an estimation of the obstacle's surface can be performed to calculate an avoidance vector and to minimize the number of constraints. However, the avoidance vectors act as repulsive motions to the opposite direction of the obstacle. The repulsive motion makes sense for their small detection range to move away as fast as possible.

III. Proximity Sensing



(a) (b)

Fig. 3: a) Proximity servoing system with three proximity sensing cuffs. b) ToF (green) and electric field (red) for narrow-far and wide-near field proximity measurements.

We provide a short system overview, for more information we refer to our previous work [4]. In short, our system comprises of three proximity sensing cuffs attached on a 7 DoF Kuka LBR iiwa 7 R800 (Fig. 3). Two cuffs on the lower link house twelve proximity sensing modules each and the cuff on the TCP has eight sensing modules. The arrangement of the sensing elements within the cuffs provide 360° coverage. The proximity sensors [14] use time-of-flight (ToF) based laser range sensors for far range measurements up to $1\,\mathrm{m}$ and capacitive sensors to cover the wide close range area. Note, the coverage is not complete but sufficient to provide data for our algorithms. The capacitive measurements do not provide quantitative information and are highly affected by material properties. Therefore, obstacles only trigger binary signals when certain thresholds are exceeded. The signals form virtual objects with a constant range in close distance. The update rate for ToF and capacitive measurements are $100\,\mathrm{Hz}$ and $1000\,\mathrm{Hz}$ respectively. In the end, the proximity sensing system provides a point cloud of which the minimum

distances to the links of the robots are calculated. The algorithms self-measurement removal are also provided in our previous work.

IV. Collision voidance

. Quadratic Programming

1) Main Task Definition: Let $\dot{\mathbf{q}} \in \ ^{n}$ define the joint velocity of a redundant robot with n joints. The corresponding Cartesian motion of the robot's main task $\dot{\mathbf{x}} \in \ ^{m}$ with m degrees of freedom (DoF) is $\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}$, where $\mathbf{J} \in \ ^{m\times n}$ is the Jacobian of the robot's main task. With redundancy $m < n$ an inverse solution with a minimizer for $\dot{\mathbf{q}}$ is $\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\dot{\mathbf{x}}$, where $\mathbf{J}^{\dagger} \in \ ^{n\times m}$ is the Moore-Penrose pseudo inverse of $\mathbf{J}$. In the presence of hard constraints, such as obstacles or joint limits, it is desired to maintain $\dot{\mathbf{x}}$ as long as the redundancy of the robot allows and therefore the quadratic expression for minimization is:

$$g\left(\dot{\mathbf{q}}\right) = \frac{1}{2}\left(\dot{\mathbf{x}} \quad \mathbf{J}\dot{\mathbf{q}}\right)\ \left(\dot{\mathbf{x}} \quad \mathbf{J}\dot{\mathbf{q}}\right) + \frac{}{2}\dot{\mathbf{q}}\ \dot{\mathbf{q}} \tag{1}$$

$$= \frac{1}{2}\dot{\mathbf{q}} \quad \mathbf{J}\ \mathbf{J} + \ \mathbf{I}\right)\dot{\mathbf{q}} \quad \dot{\mathbf{x}}\ \mathbf{J}\dot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{x}}\ \dot{\mathbf{x}}\ . \tag{2}$$

The first term in (1) expresses the quadratic error of the main task motion and the second term represents the joint velocities which are damped by when the manipulator is close to singularity [15]. Since the robot is not able to move within singular joint configurations, their prevention is required for collision avoidance.

2) Joint Constraints: Joint limitations are restrictions either in joint velocities $\dot{\mathbf{q}}$ or joint positions $\mathbf{q}$. These limitations represent bounded constraints

$$\left.\begin{matrix} 0, & \mathbf{q} \le \mathbf{q}_{\text{lb}} \\ \dot{\mathbf{q}}_{\text{lb}}, & \text{otherwise} \end{matrix}\right\} \le \dot{\mathbf{q}} \le \begin{cases} 0, & \mathbf{q} \ge \mathbf{q}_{\text{ub}} \\ \dot{\mathbf{q}}_{\text{ub}}, & \text{otherwise} \end{cases} \tag{3}$$

where $\mathbf{q}_{\text{xb}}$, $\dot{\mathbf{q}}_{\text{xb}}$ are lower and upper bounds of joint position and joint velocity. If a limitation of joint acceleration or jerk is required, $\dot{\mathbf{q}}_{\text{xb}}$ can be dynamically calculated according to the dynamic state of the robot. However, further details in how to choose joint boundaries are out of scope of this work and a method for this problem is proposed in [16].

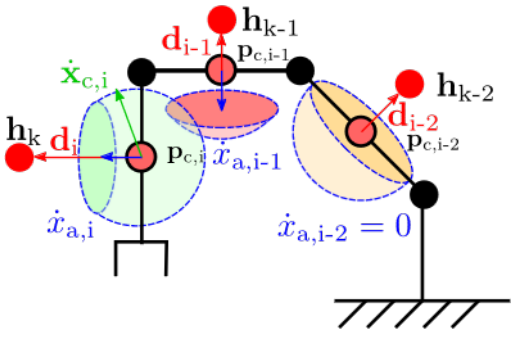


Fig. 4: Visualization of avoidance space. Different limits for approach velocities are selected according to the obstacle distance $\|\mathbf{d}\|$. $\dot{x}_{\text{a}}$: max. approach speed in direction $\mathbf{d}$; $\mathbf{d}$: obstacle distance vector; $\dot{\mathbf{x}}_{\text{c}}$: collision avoidance velocity vector; $\mathbf{h}$: obstacles; $\mathbf{p}_{\text{c}}$: closest point on manipulator.

The kinematic structure of our robot does not allow self-collision as long as the joints are within their constraints. However, an end-effector can cause self-collision. To avoid self-collisions generally and to comply with our avoidance scheme, the forwared kinematics is used to move additional obstacle points which are attached to the robots links and end-effector. These points are only considered by links other than those to which they are attached.

3) Collision voidance Constraints: For collision avoidance behaviour, a linear inequality constraint limits the allowed avoidance motion $\dot{\mathbf{x}}_{\text{c}} \in \ ^{3}$ towards direction $\hat{\mathbf{d}} \in \ ^{3}, \|\hat{\mathbf{d}}\| = 1$ of obstacle $\mathbf{h} \in \ ^{3}$ within a maximum approach velocity $\dot{x}_{\text{a}} \in \ $. To emphasize, the constraint only limits the approach velocity. Motions in tangential directions are still permitted.

$$\hat{\mathbf{d}}\ \ \dot{\mathbf{x}}_{\text{c}} = \hat{\mathbf{d}}\ \ \mathbf{J}_{\text{p}_{\text{c}}}\dot{\mathbf{q}} \le \dot{x}_{\text{a}} \tag{4}$$

where $\mathbf{J}_{\text{p}_{\text{c}}} \in \ ^{3\times n}$ is the position Jacobian at the closest point on the manipulator. The above constraint only applies to a single obstacle point on the manipulator $\mathbf{p}_{\text{c}}$ for clarity. Further constraints are appended to the vectors.

$$\begin{bmatrix} \hat{\mathbf{d}}_1\,\mathbf{J}_{\text{p}_{\text{c,1}}} \\ \vdots \\ \hat{\mathbf{d}}_{\text{i}}\ \mathbf{J}_{\text{p}_{\text{c,i}}} \end{bmatrix} \dot{\mathbf{q}} \le \begin{bmatrix} \dot{x}_{\text{a,1}} \\ \vdots \\ \dot{x}_{\text{a,i}} \end{bmatrix} = \dot{\mathbf{x}}_{\text{a}} \tag{5}$$

The resulting approach velocity constraints are illustrated in Fig. 4, where spheres represent the constrained velocity spaces. The maximum approach velocity changes dynamically depending on the distance to the obstacle. Thus, an almost full sphere (green) reduces the approach velocities at large distances and the red sphere creates a repulsive motion for small distances.

In passive behaviours, the robot may get stuck in an equilibrium state of attraction to the target position and repulsion from the obstacle. n additional linear inequality constraint contributes to an active avoidance motion, which increases the sum of all distances to obstacles within a certain distance threshold. The resulting behaviour allows the robot to move around obstacles. We compute the Jacobian $\mathbf{J}_{\text{c}}$ of the sum of all weighted Euclidean distances to obstacles numerically

$$\mathbf{J}_{\text{c}} = \mathbf{w}\ \ \mathbf{C} \qquad \mathbf{C} = \nabla\ \ \|\mathbf{d}\| \tag{6}$$

Close obstacles receive greater weightings than obstacles in the distance. We set the derivative of the weighted distance to a positive value

$$\mathbf{J}_{\text{c}}\dot{\mathbf{q}} \le 0\ . \tag{7}$$

4) Final Equation: To bring all equations into quadratic programming notation, the final expressions are:

$$\min \frac{1}{2}\dot{\mathbf{q}}\ \ \mathbf{H}\dot{\mathbf{q}} + f\ \ \dot{\mathbf{q}} \quad \text{s.t.} \begin{cases} \dot{\mathbf{q}} \le \mathbf{b}, \\ \mathbf{b}_{\text{l}} \le \dot{\mathbf{q}} \le \mathbf{b}_{\text{u}} \end{cases} \tag{8}$$

$$\mathbf{H} = \mathbf{J}\ \mathbf{J}+\ \mathbf{I} \qquad f = \ \dot{\mathbf{x}}\ \mathbf{J} \tag{9}$$

$$= \begin{bmatrix} \hat{\mathbf{d}}\ \mathbf{J}_{\mathrm{p_c}} \\ \mathbf{J}_{\mathrm{c}} \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} \dot{\mathbf{x}}_{\mathrm{a}} \\ 0 \end{bmatrix} \tag{10}$$

$$\mathbf{b}_{\mathrm{l}} = \begin{cases} 0, & \mathbf{q} \leq \mathbf{q}_{\mathrm{lb}} \\ \dot{\mathbf{q}}_{\mathrm{lb}}, & \text{otherwise} \end{cases} \qquad \mathbf{b}_{\mathrm{u}} = \begin{cases} 0, & \mathbf{q} \geq \mathbf{q}_{\mathrm{ub}} \\ \dot{\mathbf{q}}_{\mathrm{ub}}, & \text{otherwise} \end{cases} \tag{11}$$

where $\mathbf{H}$ and f are the quadratic and linear objective term, and $\mathbf{b}$ are the linear inequality constraints, and $\mathbf{b}_{\mathrm{l}}$ and $\mathbf{b}_{\mathrm{u}}$ are the lower and upper bounds of the parameter set.

B. Parameter Selection

The quadratic program requires input parameters for joint limitations, approach velocity $\dot{x}_{\mathrm{a}}$, and damping factor . While joint limitations are technical and physical constraints of the robot, latter parameters can by dynamically chosen.

1) pproach Velocity: The maximum approach velocity $\dot{x}_{\mathrm{a}}$ is variable and is chosen according to different sections of obstacle distances $\|\mathbf{d}\|$ (Fig. 5) with separations at d_{a}, $d_{0,\mathrm{u}}$, $d_{0,\mathrm{l}}$, $d_{\mathrm{r}} \in \ _{>}$ where $d_{\mathrm{a}} > d_{0,\mathrm{u}} > d_{0,\mathrm{l}} > d_{\mathrm{r}}$.

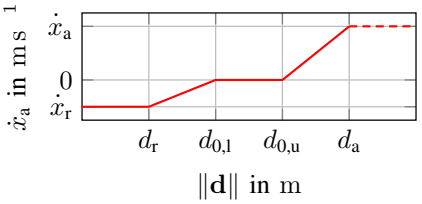


Fig. 5: pproach velocity $\dot{x}_{\mathrm{a}}$ vs. distance to obstacle $\|\mathbf{d}\|$.

The points between the segments are linear interpolated

$$\mathcal{S}\left(x, x_{\mathrm{u}}, x_{\mathrm{l}}, y_{\mathrm{u}}, y_{\mathrm{l}}\right) = \frac{y_{\mathrm{u}} \quad y_{\mathrm{l}}}{x_{\mathrm{u}} \quad x_{\mathrm{l}}}\left(x \quad x_{\mathrm{l}}\right) + y_{\mathrm{l}} \tag{12}$$

and the behaviour is defined as follows:

$$\dot{x}_{\mathrm{a}} = \begin{cases} \mathcal{S}\left(\|\mathbf{d}\|, d_{\mathrm{a}}, d_{0,\mathrm{u}}, \dot{x}_{\mathrm{a}}, 0\right), & d_{0,\mathrm{u}} < \|\mathbf{d}\| \leq d_{\mathrm{a}} \\ 0, & d_{0,\mathrm{l}} < \|\mathbf{d}\| \leq d_{0,\mathrm{u}} \\ \mathcal{S}\left(\|\mathbf{d}\|, d_{0,\mathrm{l}}, d_{\mathrm{r}}, 0, \dot{x}_{\mathrm{r}}\right), & d_{\mathrm{r}} < \|\mathbf{d}\| \leq d_{0,\mathrm{l}} \\ \dot{x}_{\mathrm{r}}, & \|\mathbf{d}\| \leq d_{\mathrm{r}} \end{cases} \tag{13}$$

t d_{a} the manipulator is allowed to approach the obstacle but limited to the maximum approach velocity $\dot{x}_{\mathrm{a}} \in \ _{>}$. $\dot{x}_{\mathrm{a}}$ is undefined for obstacles with distances above d_{a} since the algorithm does not include them in the calculation. Further movements to d_0 reduce the approach velocity $\dot{x}_{\mathrm{a}}$ to zero. In the last section the location of the manipulator is too close to the obstacle so that repulsive motions $\dot{x}_{\mathrm{r}} \leq \dot{x}_{\mathrm{a}} < 0$ are applied with the repulse velocity down to $\dot{x}_{\mathrm{r}} \in \ _{<}$ in the opposite direction.

2) Damping Factor: We select the damping factor [15] according to the measure of manipulability w proposed in [17] in order to avoid singular configurations of the joints.

$$w = \sqrt{\det\left(\mathbf{J}\mathbf{J}^{\ }\right)} \tag{14}$$

$$= \begin{cases} \left(1 \quad \frac{w}{w}\right)^2, & w < w \\ 0, & w \geq w \end{cases} \tag{15}$$

w is a threshold at which the damping activates and is a proportional gain.

C. Constraint Satisfaction

The satisfaction of all constraints are not always feasible due to conflicting constraints, especially with larger numbers of active obstacles. solution involving high spatial resolution is the segmentation and combination of obstacles and thus a reduction of constraints [13]. However, those methods are not applicable with low spatial resolution . We reduce constraints depending on the distance $\|\mathbf{d}\|$ in two steps until only obstacles with minimum allowed distances remain. First, constraints are removed for obstacles in uncritical distances, such as above $\|\mathbf{d}\| > d_{\ \mathrm{u}}$. If unsatisfied constraints still remain, the maximum approach velocity for repulsive motions are reduced to zero, which confines the avoidance space less than a repulsive motion.

V. Experiment l Results

First, we verify each function of our algorithm, particularly collision constraints, joint constraints, and damping close to singularities, in a 3D simulation as a controlled environment. Then, we evaluate the avoidance motions and compare the results with a repulsive motions and our prior approach. t the end, experiments demonstrate the effectiveness of our method on a real robotic system with dynamic obstacles.

. Experimental Parameters

The collision avoidance algorithm is implemented in M T-L B and controls only the position (3 DoF) of the 7 DoF robot to have enough redundancy left (4 DoF) for obstacle avoidance. The robot is controlled in inverse differential kinematics mode, where we provide joint velocities $\dot{\mathbf{q}}$ to an ethernet interface with an update rate of 50 Hz. The 3D simulation and the robot control cabinet access the interface to control the virtual or real robot. The robot control cabinet interpolates the data for low-level control with higher update rate and handles acceleration and jerk limitations. We use V-Rep [18] for simulations in which an exact model of the robot including the proximity sensing cuffs is added. The parameters for our algorithm are set as follows:

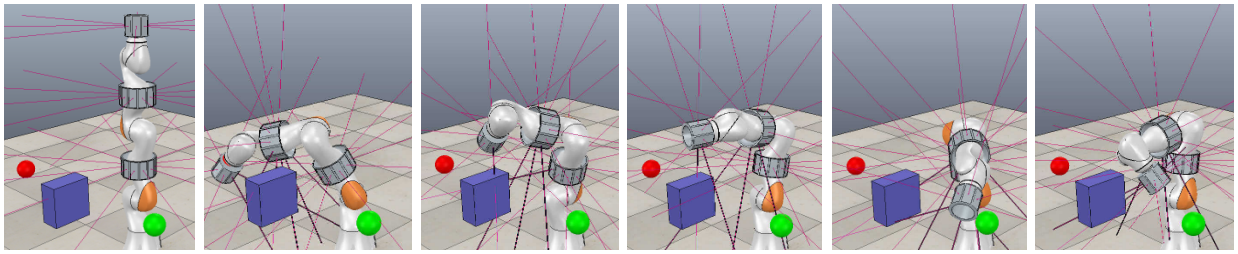
$$\begin{bmatrix} \dot{x}_{\mathrm{a}} & \dot{x}_{\mathrm{r}} \end{bmatrix} = \begin{bmatrix} 0.2 & 0.2 \end{bmatrix}\ \mathrm{m\,s}^{-1} \tag{16}$$

$$\begin{bmatrix} d_{\mathrm{a}} & d_{0,\mathrm{u}} & d_{0,\mathrm{l}} & d_{\mathrm{r}} \end{bmatrix} = \begin{bmatrix} 0.3 & 0.25 & 0.22 & 0.17 \end{bmatrix}\ \mathrm{m} \tag{17}$$

$$\begin{bmatrix} & w & \end{bmatrix} = \begin{bmatrix} 0.1 & 0.001 \end{bmatrix} \tag{18}$$

B. Function Verification

To verify the constraint satisfaction, the robot moves between the points $\mathbf{a}$ and $\mathbf{b}$ with a cuboid obstacle with the size $\mathbf{s}$ at position $\mathbf{c}$ (Fig 6). The robot starts in a singular joint configuration $\mathbf{q}$ with joint limitations $\mathbf{q}_{\mathrm{ub}}$ and $\mathbf{q}_{\mathrm{lb}}$.



(a) (b) (c) (d) (e) (f)

Fig. 6: Robot motion during functional verification. Red/green: target position **a**/**b**; blue: obstacle at position **c**.

$$\mathbf{a} = [0.4 \quad 0.5 \quad 0.5]\,\mathrm{m} \tag{19}$$
$$\mathbf{b} = [0.4 \quad 0.5 \quad 0.5]\,\mathrm{m} \tag{20}$$
$$\mathbf{c} = [0.5 \quad 0 \quad 0.5]\,\mathrm{m} \tag{21}$$
$$\mathbf{s} = [0.2 \quad 0.1 \quad 0.2]\,\mathrm{m} \tag{22}$$
$$\mathbf{q} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^\circ \tag{23}$$
$$\mathbf{q}_{\mathrm{ub}} = [170 \quad 115 \quad 165 \quad 105 \quad 165 \quad 100 \quad 170]^\circ \tag{24}$$
$$\mathbf{q}_{\mathrm{lb}} = [\ 10 \quad 115 \quad 165 \quad 105 \quad 165 \quad 100 \quad 170]^\circ \tag{25}$$

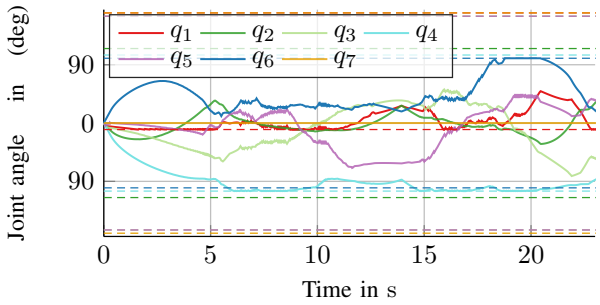
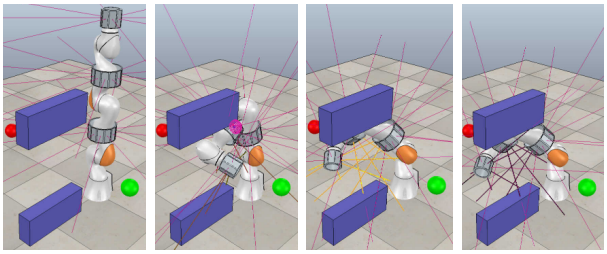


Fig. 7: Joint motion and joint constraints.

The mean calculation time of our method in M TL B (Intel 6300HQ) is 7.8 ms, compared to 4.2 ms for the repulsive method. Fig. 7 shows the joint positions during motion. The joints do not exceed their limits but experience noise during the hard switching procedure of constraints. In correlation with Fig. 9 the noise is particularly prominent when obstacles are detected. This behaviour can be minimized by taking the robot's dynamics and motion state into account during the calculation of joint boundaries and approach velocities. The robot successfully avoids obstacles, but in some cases the minimum distances fall below the predefined values. This is caused by the small coverage of the sensors, where obstacles often slide into the sensing cone from the side causing sudden negative spikes. The robot moves around the obstacle and reaches the targets, which is shown in Fig. 10. Since the initial joint configurations are singular, the damping effect moves the robot out of singularity.

C. Collision voidance Comparison

We compare our approach with our prior proposed method and a method based on repulsive motions. In our prior method the robot moves tangentially to avoid obstacles, while the repulsive method moves in the opposite direction to obstacles. In the same setup, as in Fig. 6 the robot reaches the targets faster with our proposed method than with tangential motions (Fig. 10). The robot gets stuck in a local minimum with repulsive motions.



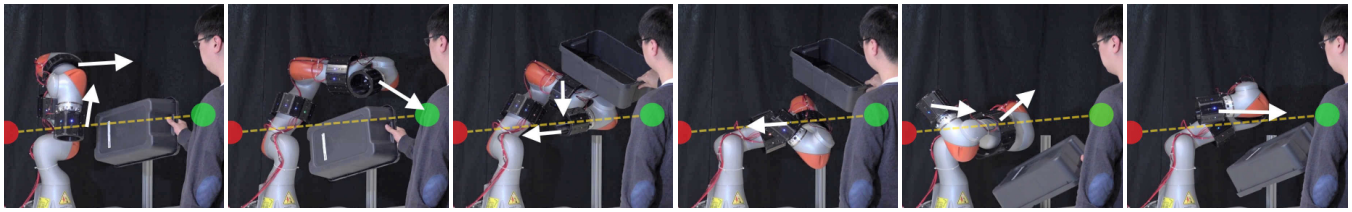
(a) (b) (c) (d)

Fig. 11: a) Singular starting position for all algorithms. b) The our proposed method goes through the obstacles, c) prior proposed tangential motion moves in front tangentially, d) repulsive motion gets stuck.

In the next setup (Fig. 11), two larger cuboid obstacles with the size of $\mathbf{s} = [0.5\ 0.1\ 0.2]\mathrm{m}$ are placed between both targets at $\mathbf{c}_1 = [0.5\,0\,0.1]\mathrm{m}$, $\mathbf{c}_2 = [0.5\ 0\ 0.8]\mathrm{m}$. The gap between both obstacles is just large enough for the manipulator to slide through while maintaining the safety distance. In Fig. 13, the robot moves between both targets with the proposed method, while with the tangential method it has difficulties reaching the target. The repulsive motion fails and the robot is stuck in a local minimum. In Fig. 12, the robustness of repulsive motions are shown. The robot maintains the minimum distance most of the time, while the other methods fall below these thresholds.

D. Experiment on Robotic Systems

In the last experiment (Fig. 8), a real robot performs collision avoidance with dynamic obstacles. First, an obstacle is held within the path of the end-effector, which gets deflected. Next, the obstacle is held to the lower links from above and below. The robot responds with avoidance motions, while trying to maintain the end-effector motion. Compared to the simulation results, the minimum distances fall below the set thresholds more often (Fig. 15), especially at link 3. This



(a) (b) (c) (d) (e) (f)

Fig. 8: Real world experiment with dynamic obstacles.

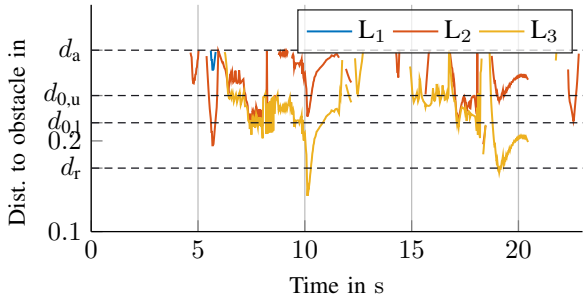


Fig. 9: Min. distance to the closest obstacle for each link.

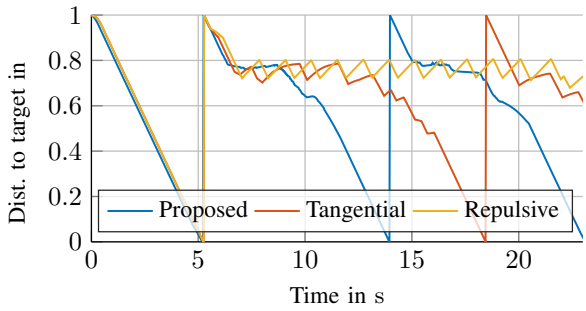


Fig. 10: Distance between end-effector and target position. The target position is switched back when it is reached which creates a sawtooth pattern.

is caused by the control cabinet, which incorporates the robot dynamics into the trajectories and does not apply the correction motions immediately. The joint angle recordings support this in Fig. 14, where disturbances are less prominent due to dynamics control in the control cabinet.

VI. Conclusion

In this paper we present a method for collision avoidance for proximity sensing skins. Our approach generates instantaneous joint motions with quadratic optimization based on the current state of the environment and the joint angles. The experiments show significant improvements towards obstacle avoidance, less path deviation, and robustness. With a full coverage robotic skin safety can be increased. It is worth noting that our method provides an additional fast reacting mechanism in the layers of safety algorithms and complements other methods, such as global path planning. fter collision avoidance, there is enough time to replan the path with a global path planner while our method ensures safe states in which no collisions occur.

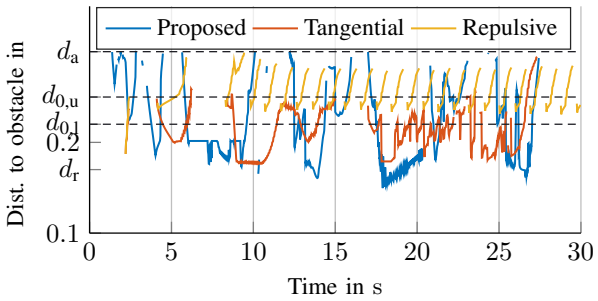


Fig. 12: Overall min. distance to the closest obstacle.

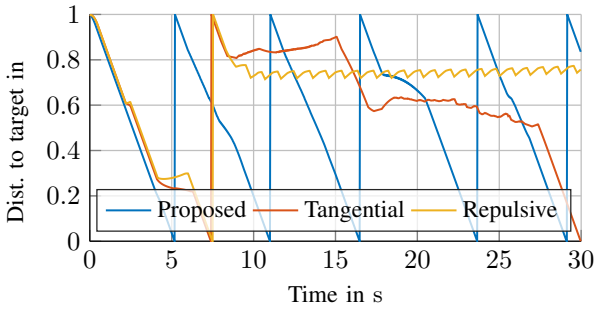


Fig. 13: Distance between end-effector and target position.

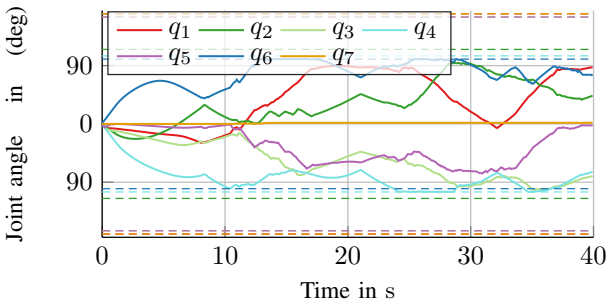


Fig. 14: Joint motion and joint constraints.

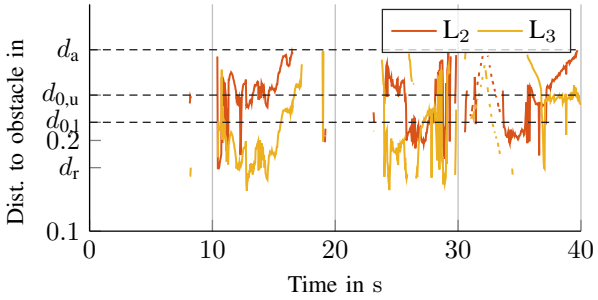


Fig. 15: Distance to the closest obstacle for each link.

REFERENCES

[1] E. Cheung and V. J. Lumelsky, "Proximity sensing in robot manipulator motion planning: system and implementation issues," *IEEE Transactions on Robotics and utomation*, vol. 5, no. 6, pp. 740–751, Dec 1989.

[2] 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.

[3] 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.

[4] Y. Ding, F. Wilhelm, L. Faulhammer, and U. Thomas, "With proximity servoing towards safe human-robot-interaction," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2019, pp. 4907–4912.

[5] 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.

[6] S. E. Navarro, S. Koch, and B. Hein, "3d contour following for a cylindrical end-effector using capacitive proximity sensors," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 82–89.

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

[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] N. M. Ceriani, . M. Zanchettin, P. Rocco, . Stolt, and . Robertsson, "Reactive task adaptation based on hierarchical constraints classification for safe industrial robots," *IEEE/ SME Transactions on Mechatronics*, vol. 20, no. 6, pp. 2935–2949, Dec 2015.

[11] B. Lacevic and P. Rocco, "Kinetostatic danger field - a novel safety assessment for human-robot interaction," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2010, pp. 2169–2174.

[12] P. Bosscher and D. Hedman, "Real-time collision avoidance algorithm for robotic manipulators," in *2009 IEEE International Conference on Technologies for Practical Robot pplications*, Nov 2009, pp. 113–122.

[13] C. Vergara, G. Borghesan, E. ertbelien, and J. De Schutter, "Incorporating artificial skin signals in the constraint-based reactive control of human-robot collaborative manipulation tasks," in *2018 3rd International Conference on dvanced Robotics and Mechatronics (IC RM)*, July 2018, pp. 280–287.

[14] 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.

[15] Y. Nakamura and H. Hanafusa, "Inverse Kinematic Solutions With Singularity Robustness for Robot Manipulator Control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 108, no. 3, pp. 163–171, 09 1986.

[16] F. Flacco, . De Luca, and O. Khatib, "Motion control of redundant robots under joint constraints: Saturation in the null space," in *2012 IEEE International Conference on Robotics and utomation*, May 2012, pp. 285–292.

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

[18] E. Rohmer, S. P. N. Singh, and M. Freese, "V-rep: versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 1321–1326.