Positioning in Congested Space by Combining Vision-based and Proximity-based Control

John Thomas[1], François Chaumette[1]

***Abstract*—In this paper, we consider positioning in congested space within the framework of sensor-based control using vision and proximity sensors. Vision acts as primary sensing modality for performing the positioning task, while proximity sensors complement it by ensuring that the robotic platform does not collide with objects in the workspace. Sensor information is combined in a shared manner using the QP formalism where ideas from safety-critical control are used to express inequality constraints. The proposed method is validated through various real experiments.**

***Index Terms*—Sensor-based Control, Visual Servoing, Proximity-based Control, Positioning Task, Collision Avoidance**

I. INTRODUCTION

ROBOTIC manipulators operating in congested spaces can enhance their performance with the incorporation of multi-modal sensor signals that complement each other. One way to incorporate such exteroceptive sensor signals is to define sensor-based tasks using Sensor-based Control (SBC) framework. Tasks executed using this framework have high robustness, reactive nature and accuracy. Additionally, this approach eliminates the need for global reconstruction of the environment, robot localization and trajectory planning. Visual Servoing (VS) is effective for accurate positioning tasks, especially when the camera is attached to the end-effector of the robot in *eye-in-hand* configuration [1]. However, in this case, the camera fails to provide information around the links of the robot, which is problematic when positioning occurs in a tight space for ensuring collision avoidance while reaching the target pose. Proximity perception enables the robot to close this perception gap and provides complementary information to vision [2]. In SBC, there is lack of literature in using both vision and proximity data in a common control architecture as previous works in the area considered vision and proximity information separately [3].

In robotics, the idea of *Safety* is often linked to the task of obstacle avoidance. Local sensing can enable online detection of obstacles for autonomous robots operating in unknown environments. In [4] and [5] several infrared sensors are wrapped around the links of a manipulator to address motion planning and collision avoidance respectively. In [6] and [7], capacitance-based Whole Arm Proximity sensors are considered as a collision avoidance system to be used for the application of remote handling of hazardous waste. Safety enhancement of human-robot interaction in industrial environments is considered in [8] using distributed infrared proximity sensors. In [9] proximity sensing modality of artificial skin developed by [10], [11] was used for human robot collaboration for an industrial use case involving assembly operation. In [12] and [13], proximity sensing cuff with Time-of-Flight (ToF) and capacitive measurements is presented for enabling safe human-robot interaction with high priority given to collision avoidance. Sensitive skin consisting of capacitive sensors was used in [14] for obstacle avoidance in applications involving human-robot interactions.

A classical method for obstacle avoidance in robotics is artificial potential field where a potential barrier is created within the neighbourhood of obstacles to create repulsive force [15]. Potential field type methods often suffer from few drawbacks including oscillation for opposite obstacles, high repulsion from adjacent obstacles compared to just one, and difficulty to get closer to the obstacle. One alternative is to consider obstacle avoidance as inequality constraint using the concept of velocity damper [16]. In [17], such a constraint was used using ultrasonic distance sensors in shared control for wheelchair navigation for assisting users with motor or visual impairments. The idea of velocity damper has been generalized in safety-critical controllers through Control Barrier Function (CBF) [18].

For kinematically redundant robots, local optimization based techniques can provide real-time implementation with possibility to consider sensory information [19]. If one of the task or a constraint is modeled as an inequality then the Quadratic Programming (QP) framework can be used to realize the primary task at best subject to the satisfaction of constraint [16]. The constraint achieves higher priority in QP framework. For instance, visual servoing task is integrated into a multi-objective model-based QP controller in [20], while also considering visual constraints (occlusion avoidance, field of view maintenance, etc.) expressed as inequalities. However, obstacle avoidance is not considered. Finally, in [21], a multi-sensor multi-constraint task is considered to perform positioning with VS with no-concept of priority : an automatic weighting scheme finds an appropriate balance to fulfill the task. However, for positioning in tight space, preventing collision with the environment has to take higher priority.

In this work, we consider both vision and proximity information in a single control architecture to perform positioning

Manuscript received: April, 06, 2024; Revised July, 03, 2024; Accepted August, 02, 2024.

This paper was recommended for publication by Editor Pascal Vasseur upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by BPI France Lichie project. Experiments presented in this paper were carried out thanks to a platform of the Robotex 2.0 French research infrastructure.

[1]The authors are with Inria, Univ Rennes, CNRS, IRISA - Rennes, France. john.thomas@inria.fr

Digital Object Identifier (DOI): 10.1109/LRA.2024.3444713.

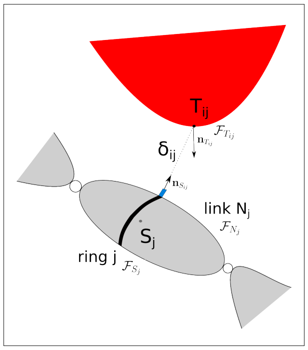


Fig. 1: Proximity ring attached with proximity sensor detecting an obstacle. Notations are defined in text.

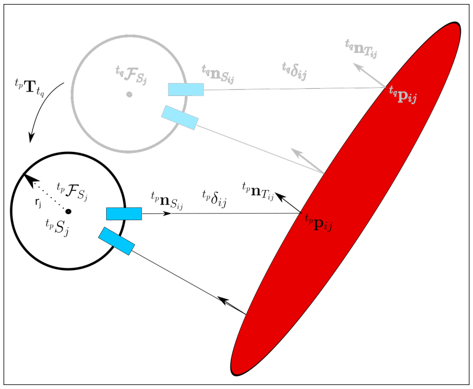

${}^{t_p}\mathbf{T}_{t_q}$
${}^{t_q}\mathcal{F}_{S_j}$
${}^{t_q}\mathbf{n}_{S_{ij}}$
${}^{t_q}\delta_{ij}$
${}^{t_q}\mathbf{n}_{T_{ij}}$
${}^{t_q}\mathbf{p}_{ij}$
r_j
${}^{t_p}\mathcal{F}_{S_j}$
${}^{t_p}S_j$
${}^{t_p}\mathbf{n}_{S_{ij}}$
${}^{t_p}\delta_{ij}$
${}^{t_p}\mathbf{n}_{T_{ij}}$
${}^{t_p}\mathbf{p}_{ij}$

Fig. 2: Multiple adjacent points for estimating surface normal

task while ensuring safety. We attach a camera to the end-effector of a serial manipulator and control its 6-DoF by using classical VS from four image points. We also attach proximity rings (each consisting of several proximity sensors) around the arm of the manipulator to provide local sensing required for obstacle avoidance. We consider modelling of these rings by providing the associated interaction matrix and an algorithm to estimate its unknown parameters. Positioning in congested space is achieved by using the QP formalism where the cost function is defined from vision-based task function and constraints are from proximity-based task functions through the application of CBF theory. We provide an analysis that ensures the stability of the QP solution at convergence in the presence of model uncertainties and sensor noise. The control framework developed to enable safe positioning is verified using three experimental cases.

This paper is divided into seven sections. In Section II we present a particular proximity sensor arrangement forming a ring and model its interaction matrix. We also provide an estimation strategy to evaluate the unknown normal of a potential obstacle. After that we propose in Section III the control framework for achieving positioning in congested space. Stability analysis of this framework is considered in Section IV. In Section V we provide experimental results for validation purpose and discuss these results in Section VI. We end the paper with conclusions in Section VII.

II. Proximity Ring

The basic characteristic arrangement of proximity sensors considered is a proximity ring. Fig. 1 indicates such a proximity ring with a sensor that detects an obstacle in the workspace. Several such rings can be attached to the manipulator to detect obstacles. Let us consider the jth ring, consisting of m_j proximity sensors arranged in such a way that the axis remains radial and passes through center S_j with radius r_j. We consider a thin-field range finder in which detection occurs along the axis of sensor where $\mathbf{n}_{S_{ij}}$ denotes the unit vector indicating sensor axis and δ_{ij} the distance measured by the sensor with respect to the obstacle for the ith sensor. $\mathbf{n}_{T_{ij}}$ is the unit vector direction of obstacle surface normal at point of detection T_{ij}. In order to obtain a common spatial velocity representation, we evaluate the model at ring center S_j. Let us now consider the sensor feature as the distance δ_{ij} measured by proximity sensor. As usual in SBC [22], [1], the design of the control scheme is based on the interaction matrix $\mathbf{L}_{\delta_{ij}}$ that relates the time variation of the sensor feature δ_{ij} to the sensor spatial velocity $\mathbf{v}_{S_j}$ under the form

$$\dot{\delta}_{ij} = \mathbf{L}_{\delta_{ij}} \mathbf{v}_{S_j} \tag{1}$$

where $\mathbf{L}_{\delta_{ij}}$ is given by [23]

$$\mathbf{L}_{\delta_{ij}} = \beta_{ij} \left[\mathbf{n}_{T_{ij}}^T \quad (\mathbf{m}_{T_{ij}} \times \mathbf{n}_{T_{ij}})^T \right] \tag{2}$$

where $\beta_{ij} = -1/(\mathbf{n}_{T_{ij}} \cdot \mathbf{n}_{S_{ij}})$ and $\mathbf{m}_{T_{ij}} = (\delta_{ij} + r_j)\mathbf{n}_{S_{ij}}$ is the displacement vector from T_{ij} to S_j. Note that the only unknown term in $\mathbf{L}_{\delta_{ij}}$ is the obstacle normal $\mathbf{n}_{T_{ij}}$ at point T_{ij}.

A. *Model for interaction matrix*

In our previous work [23], we considered a single planar target whose normal is easily estimated from proximity sensors located on multiple rings. We cannot use the same strategy when dealing with obstacle avoidance as each ring may detect different obstacles. That is why we present in this section a new estimation scheme of the target normal for each proximity sensor that is based on the successive measurements on adjacent sensors located on the same ring. We also present simplified models that have been used in our experimental results.

1) Estimation of Surface Normal: Let us assume that the obstacles are large enough to have multiple detection from adjacent sensors on the same ring (see Fig. 2). The point detected by a sensor at current instance t_p in sensor frame would be ${}^{t_p}\mathbf{p}_{ij} = ({}^{t_p}\delta_{ij} + r_j)\ {}^{t_p}\mathbf{n}_{S_{ij}}$. Let $\mathcal{N}_{ij}$ denote the set containing indices of the sensors that are detecting the same object as currently detected by sensor i on j-th ring in previous instances $\{t_q : q = 1 \dots p\}$. Let us consider a sensor $l \in \mathcal{N}_{ij}$ for a past instance t_q, the point detected by this sensor is transformed to the current sensor frame of the ring using homogeneous transformation ${}^{t_p}\mathbf{T}_{t_q}$, i.e. ${}^{t_p}\mathbf{T}_{t_q}\ {}^{t_q}\mathbf{p}_{lj}$. Points from previous instances of a sensor are ignored if its relative distance with the point of detection of current instance is lower than a threshold. This leads us to consider a set of n_{ij} eligible points $\{., {}^{ij}\mathbf{p}_s, .\}_{s=1\dots n_{ij}}$ assumed to lie on a locally planar surface defined by $\mathbf{n}^T\mathbf{p} + d = 0$, where $\mathbf{p}$ represents an arbitrary point and $\left(\mathbf{n}^T,\ d\right)$ such that $||\mathbf{n}|| = 1$ indicates the plane parameters. This leads to the least squares problem

$$\hat{\mathbf{n}}_{T_{ij}}, d^*_{ij} = \underset{\mathbf{n},d}{\operatorname{argmin}} \sum_{s=1}^{n_{ij}} \left({}^{ij}\mathbf{p}_s^T\mathbf{n} - d\right)^2 \tag{3}$$

where a computationally fast approximation of the solution is given by [24]

$$\hat{\mathbf{n}}_{T_{ij}} = \frac{1}{||\widetilde{\mathbf{B}}_n^{-1}\ \tilde{\mathbf{b}}||}\ \widetilde{\mathbf{B}}_n^{-1}\ \tilde{\mathbf{b}} \tag{4}$$

where $\widetilde{\mathbf{B}}_n = \sum_{s=1}^{n_{ij}} {}^{ij}\mathbf{p}_s {}^{ij}\mathbf{p}_s{}^T$ and $\tilde{\boldsymbol{b}} = \sum_{s=1}^{n_{ij}} {}^{ij}\mathbf{p}_s$. The result obtained could then be injected in (2).

2) Approximate Model - Approximation in intensity: A practical approximation of the model (2) is to consider $\beta_{ij} = 1$ in (2), leading to

$$\mathbf{L}_{\delta_{ij}} = \left[\hat{\mathbf{n}}_{T_{ij}} \quad (\mathbf{m}_{T_{ij}} \times \hat{\mathbf{n}}_{T_{ij}})^T\right] \tag{5}$$

Indeed, for close encounters with obstacles, β_{ij} could result in undesirable non-smooth behavior in the evolution of task function.

3) Approximate Model - Approximation in direction of target normal : Another simple way to approximate the interaction matrix is to assume that surface normal is aligned along the opposite direction of proximity axis $\mathbf{n}_{T_{ij}} = -\mathbf{n}_{S_{ij}}$. In that case, the model reduces to

$$\mathbf{L}_{\delta_{ij}} = \left[\begin{array}{cc} -\mathbf{n}^T_{S_{ij}} & \mathbf{0^T} \end{array}\right] \tag{6}$$

This is considered as obstacle avoidance primitive in [25], where the surface is considered to be locally orthogonal to the sensor axis. For situations where obstacles are far away, it is enough to consider this obstacle avoidance primitive. However, we show a scenario in Section V-B where this coarse model is inadequate.

III. Control Architecture

In this section we consider the control architecture used.

A. Positioning with 4-image point Visual Servoing

Let us consider $\mathbf{e}_{4p}(\mathbf{q})$ as the task function associated to normalized coordinates $\mathbf{x} = (x, y)$ of four image points [1] where $\mathbf{q} \in \mathbb{R}^n$ indicates the joint state of the robot, n being the number of robot joints. Regulation of $\mathbf{e}_{4p}(\mathbf{q})$ through control input $\mathbf{u}$ to the low-level controller of the robot could be defined in linear least-squares representation as

$$\dot{\mathbf{q}} = \underset{\mathbf{u}\in\mathbb{R}^n}{\operatorname{argmin}}\ \frac{1}{2}\ ||\dot{\mathbf{e}}^*_{4p} - \mathbf{J}_{4p}\mathbf{u}||^2 \tag{7}$$

where $\mathbf{J}_{4p}$ is the $8 \times n$ feature Jacobian matrix, $\dot{\mathbf{e}}^*_{4p} = -\lambda_{4p}\mathbf{e}_{4p}$ and λ_{4p} is a positive gain.

B. Anti-collision using Proximity-based Control

While the robot moves to achieve the vision-based task, the robot also has to prevent collision with obstacles. We consider an inequality condition on the task function $\mathbf{e}_{\delta_{ij}} = \delta_{ij} - \delta^*_{ij} \geq 0$, where δ^*_{ij} is the threshold distance below which we assume virtual penetration with the obstacle. $\mathbf{e}_{\delta_{ij}} = 0$ is considered as virtual contact. We formulate safety achieved through obstacle avoidance as the invariance of the superlevel set defined on $\mathbf{e}_{\delta_{ij}}$ for the system with joint state $\mathbf{q}$ as $C = \left\{\mathbf{q} \in \mathbb{R}^n : \mathbf{e}_{\delta_{ij}}(\mathbf{q}) \geq 0\right\}$ [18]. The standard task function defined in proximity space $\mathbf{e}_{\delta_{ij}}$ becomes a CBF if we find an extended class $\mathcal{K}$ function $\alpha(.)$ that satisfies condition

$$\dot{\mathbf{e}}_{\delta_{ij}}(\mathbf{q}) \geq -\alpha(\mathbf{e}_{\delta_{ij}}) \tag{8}$$

We now define the set $K_k(\mathbf{q})$ as

$$K_k(\mathbf{q}) = \left\{\mathbf{u} \in \mathbb{R}^n : \mathbf{a}_k^T\ \mathbf{u} \geq b_k\right\} \tag{9}$$

with $\mathbf{a}_k^T = \ \mathbf{J}_{e_{\delta_{ij}}} = \mathbf{L}_{\delta_{ij}}{}^j\mathbf{V}_{N_j}\ \mathbf{J}_{N_j},\ b_k = -\alpha(\mathbf{e}_{\delta_{ij}}),\ \mathbf{u} = \dot{\mathbf{q}}$, and $k = (j-1) * m_j + i$ where ${}^j\mathbf{V}_{N_j}$ is the twist transformation between frame $\mathcal{F}_{S_j}$ and frame $\mathcal{F}_{N_j}$ and $\mathbf{J}_{N_j}$ is the geometric Jacobian of link N_j. The candidate $\alpha(\mathbf{e}_{\delta_{ij}})$ is chosen here as $\lambda_j\mathbf{e}_{\delta_{ij}}$ where λ_j is a positive gain. CBF is both a necessary and sufficient condition for safety [18]. As proven in [26], any controller $\mathbf{u} \in K_k(\mathbf{q})$ at a given $\mathbf{q}$ renders the set C forward invariant. This ensures that the system state does not reach the complement of the set, which in this case would correspond to virtual penetration and eventual real collision. Additionally, it also makes the set asymptotically stable. In practical implementation, when the system leaves the safe set to its complement due to modelling errors and sensor uncertainties, it is brought back to safety. This makes it useful in SBC where inherent robustness of the controller is valued for execution of the task without exact calibration.

The formulation of positioning as minimization of a quadratic form and obstacle avoidance as linear constraints leads us to QP as a natural framework that can be expressed as

$$\begin{aligned} \dot{\mathbf{q}} = \underset{\mathbf{u}\in\mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2}\mathbf{u}^T\mathbf{G}\mathbf{u} + \mathbf{u}^T\mathbf{c} \\ : \mathbf{a}_k^T\mathbf{u} \geq b_k, k \in \mathcal{I}_p \\ \mathbf{u}_{upper} \geq \mathbf{u} \geq \mathbf{u}_{lower} \end{aligned} \tag{10}$$

where $\mathbf{G} = \mathbf{J}_{4p}{}^T\mathbf{J}_{4p} + \mu\mathbf{I},\ \mathbf{c} = -\mathbf{J}^T_{4p}\dot{\mathbf{e}}^*_{4p} = \lambda_{4p}\mathbf{J}^T_{4p}\mathbf{e}_{4p}$. Here $\mathbf{J}_{4p}{}^T\mathbf{J}_{4p}$ is a symmetric $n \times n$ matrix. It is composed of feature Jacobian matrix $\mathbf{J}_{4p}$ that is of rank 6 in case $n \geq 6$. If a redundant robot with $n > 6$ joints is used, which is our case, matrix $\mathbf{J}_{4p}{}^T\mathbf{J}_{4p}$ is a semi-definite matrix, which makes it a convex QP problem. To make it strictly convex and for getting a well-conditioned solution, a regularization term $\mu\mathbf{I}$ is added so that $\mathbf{G}$ is positive definite. An explanation for this is provided in Section IV. The linear part of the quadratic term $\mathbf{c} \in \mathbb{R}^n$ consists of task function in image space and becomes minimal at convergence. $\mathcal{I}_p$ indicates the set of inequality constraints from proximity sensors. We consider several proximity rings as described in Section II, consisting of j rings with each carrying m_j sensors. As already defined after (9), the i-th sensor on ring j is denoted by index k. Therefore we have in total $j * m_j$ sensors ensuring anti-collision. Usually, at a given instant, only few constraints where the sensors are in close encounter with obstacles become relevant. The last inequalities come from the joint velocity limits, where $\mathbf{u}_{upper} \in \mathbb{R}^n$ indicates its upper limit and $\mathbf{u}_{lower} \in \mathbb{R}^n$ the lower limits.

C. Optimality

In this section we discuss the optimality conditions for the QP problem. These conditions enable us to verify if a feasible point is a local minimizer. The feasible set Ω of the problem is defined as

$$\Omega = \left\{\mathbf{u} : \mathbf{a}_k^T\mathbf{u} \geq b_k, k \in \mathcal{I}_p, \mathbf{u}_{upper} \geq \mathbf{u} \geq \mathbf{u}_{lower}\right\} \tag{11}$$

The active set $\mathcal{A}(\mathbf{u})$ at any feasible point $\mathbf{u}$ is defined as

$$\mathcal{A}(\mathbf{u}) = \left\{k \in \mathcal{I}_p : \mathbf{a}_k^T\mathbf{u} = b_k\right\} \tag{12}$$

We consider linear independence constraint qualification (LICQ) to hold, resulting in linear independence in the set of active constraint gradients $\{\mathbf{a}_k, k \in \mathcal{A}(\mathbf{u})\}$. The Lagrangian of QP problem is

$$\mathcal{L}(\mathbf{u}, \kappa) = \frac{1}{2}\mathbf{u}^T\mathbf{G}\mathbf{u} + \mathbf{u}^T c - \sum_{k\in\mathcal{I}_p} \kappa_k \left(\mathbf{a}_k^T\mathbf{u} - b_k\right). \tag{13}$$

Here κ_k are the Lagrangian multipliers and the gradient of Lagrangian with respect to $\mathbf{u}$ is

$$\nabla_{\mathbf{u}}\mathcal{L}\left(\mathbf{u}, \kappa\right) = \mathbf{G}\mathbf{u} + \mathbf{c} - \sum_{k\in\mathcal{A}(\mathbf{u})} \kappa_k \mathbf{a}_k \tag{14}$$

The first order necessary conditions (FONC) when LICQ holds results in *Karush-Kahn-Tucker* (KKT) conditions [27]

$$\begin{aligned} \nabla_{\mathbf{u}}\mathcal{L}\left(\mathbf{u}^*, \kappa^*\right) &= 0, \\ \mathbf{a}_k^T\mathbf{u}^* &= b_k, \forall k \in \mathcal{A}(\mathbf{u}^*), \\ \mathbf{a}_k^T\mathbf{u}^* &\geq b_k, \forall k \in \mathcal{I}_p \backslash \mathcal{A}(\mathbf{u}^*), \\ \kappa_k^* &\geq 0, \forall k \in \mathcal{A}(\mathbf{u}^*). \end{aligned} \tag{15}$$

where $\mathbf{u}^*$ is a local solution of (10) and κ^* indicates corresponding Lagrangian multipliers. In case of convex QP (such as the problem considered), if LICQ holds then FONC are sufficient to be a global minimizer [27]. These conditions lead naturally to stopping conditions that indicate the robot configuration when a minimum is reached in case of static obstacles.

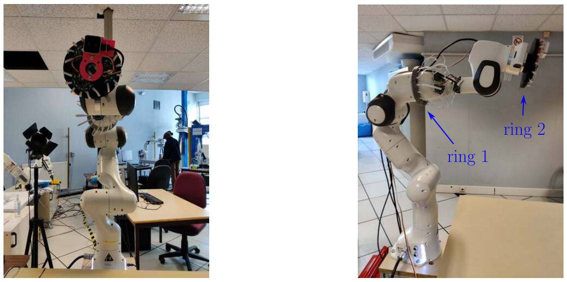
Stopping Conditions: Here we consider two stopping criteria that are effective in two distinct situations. In the first case, the global minimum of the vision-based task function is reached. For this we evaluate if the norm of $\mathbf{e}_{4p}$ is below a low threshold ϵ_{4p}. The second case is where the obstacles prevent the positioning task from converging to its global minimum. In those situations the system stops when the norm of gradient of Lagrangian with the solution at $\mathbf{u}^* = 0$ is below a threshold ϵ_{qd},

if $\left(\left(||\mathbf{e}_{4p}|| \leq \epsilon_{4p}\right) \vee \left(||\nabla_{\mathbf{u}}\mathcal{L}\left(\mathbf{0}, \kappa^*\right)|| \leq \epsilon_{qd}\right)\right)$ Stop Task
else Solve QP.

IV. Stability Analysis

In this section we consider general conditions that proves stability of solutions to perturbations of the linear part of QP. Let us consider a perturbation of $(\mathbf{A}_p^*,\ \mathbf{b}^*,\ \mathbf{c}^*)$ of appropriate dimensions with non-negative parameter δ_g,

$$\begin{aligned} \dot{\mathbf{q}} = \underset{\mathbf{u}\in\mathbb{R}^n}{\text{argmin}} &\frac{1}{2}\mathbf{u}^T\mathbf{G}\mathbf{u} + \mathbf{u}^T\left(\mathbf{c} + \delta_g\mathbf{c}^*\right) \\ &: (\mathbf{A}_p + \delta_g\mathbf{A}_p^*)\mathbf{u} \geq \mathbf{b} + \delta_g\mathbf{b}^* \end{aligned} \tag{16}$$



(a) Front View (b) Side View

Fig. 3: Robot arm with camera and two proximity rings.

where $\mathbf{A}_p = \begin{bmatrix} \cdot \\ \mathbf{a}_k^T \\ \cdot \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} \cdot \\ b_k \\ \cdot \end{bmatrix}$. These perturbations can be considered from model uncertainties and measurement noise from both vision and proximity sensors. The model uncertainty for proximity sensors arises from the estimation of surface normal in the interaction matrix. Stability of the solution is considered in situation where the perturbation term δ_g is bounded with $0 \leq \delta_g \leq \epsilon$ and the perturbed problem has an optimal solution. As stated in [28], QP is stable to linear perturbations if and only if it is regular. QP is considered to be regular if it satisfies the following conditions:

Condition 1: There does not exist a nonzero vector $\mathbf{z}$ satisfying

$$\mathbf{G}\mathbf{z} = 0, \qquad \mathbf{A}_p\mathbf{z} \geq 0, \qquad \mathbf{c}^T\mathbf{z} \leq 0 \tag{17}$$

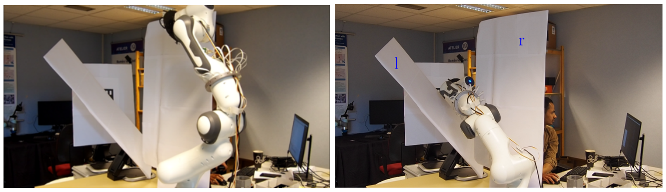
Condition 2: There do not exist vectors $\mathbf{z}$ and $\mathbf{w}$, such that $(\mathbf{z},\ \mathbf{w}) \neq \mathbf{0}$, satisfying

$$\mathbf{G}\mathbf{z} = 0, \qquad \mathbf{A}_p^T\mathbf{w} = 0, \qquad \mathbf{b}^T\mathbf{w} = 0, \qquad \mathbf{w} \geq 0 \tag{18}$$

If $\mathbf{G}$ is positive definite then Condition 1 is satisfied since $\mathbf{z} = 0$ is the only solution to $\mathbf{G}\mathbf{z} = 0$. It is here ensured through the regularization term. For Condition 2 we only need to concentrate on the inequalities belonging to the active set. If LICQ holds for those then we have $\mathbf{A}_p^T\mathbf{w} \neq 0$ if $\mathbf{w} \neq 0$, satisfying Condition 2. This means that all positioning tasks are stable to linear perturbations.

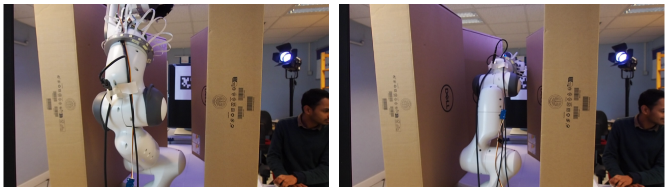
V. Experiments

Let us consider the experimental scenario as shown in Fig. 3. It consists of a 7 DoF Panda robot arm to which are attached two rings, each carrying 18 proximity sensors that use ToF technology. Ring 1 which is grey in color is mounted to link 5 and ring 2 which is black in color is attached to the flange of robot and acts as the end-effector. On top of ring 2, a D405 Intel Realsense camera is mounted. The gain of the visual task is chosen as $\lambda_{4p} = 0.4$. The threshold distance δ_{ij}^* is either set to a value of 4 cm, 6 cm or 8 cm depending on the case considered. If there is no detection, the value of δ_{ij} is set to maximum detection of 1 m. Thereby we have setup the workspace in such a manner that it leads to close encounters with obstacles. Regularization term in the cost function of QP is selected as $\mu = 0.01$. In first and third experiments, we use the simple model given by (6). Such a model becomes unsuitable in situation when there is a large angular deviation between the surface normal $\mathbf{n}_{T_{ij}}$ at the point of detection and the negative direction $-\mathbf{n}_{S_{ij}}$ of the axis of corresponding proximity sensors. That is why in the second experiment where such a situation occurs for ring 2, we use the surface normal estimation method described in Section II-A1 and use the model given by (5). ViSP [29] functionalities are used in the implementation of control architecture. Numerical solver was chosen as ProxQP which provided fast and efficient C++ implementation [30]. The control loop for updating joint velocity $\dot{\mathbf{q}}$ was run at 30 Hz. At each control loop, QP solver is warm started with previous results. It was executed on a laptop with Intel® Core™ i7 CPU @ 1.90GHz × 8 and Ubuntu 20.04.2 LTS with RTLinux kernel in Fully Preemptible Mode. The accompanying video illustrates the three experiments described below.



(a) Initial configuration (b) Final configuration

Fig. 4: Experimental setup for Case 1 consisting of Panda robot, two obstacles, and 4-point target.



(a) Initial configuration (b) Final configuration

Fig. 5: Experimental setup for Case 2 where obstacles prevent global convergence of visual task function.

A. Case 1 : Global convergence of visual task function

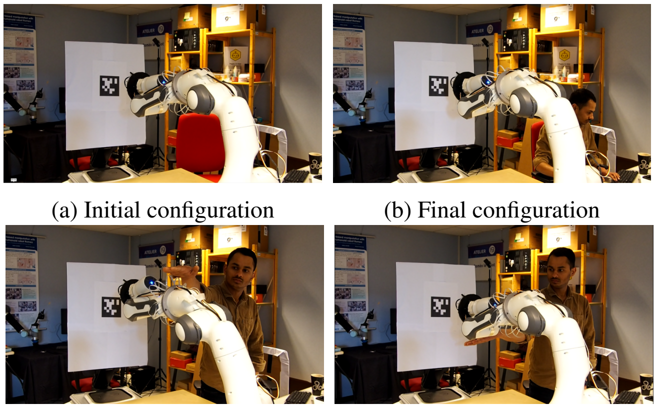
For this first experiment, we place two planar obstacles in the workspace to create a congested space around the robot arm while it has to converge to the goal position. Note that, in case safety inequalities from proximity sensors are not taken into account, the robot arm collides with the obstacles. The initial configuration of the robot with joint position {0, −60, 0, −90, 0, 100, 45} expressed in degrees is shown in Fig. 4a. The target points are placed in such a way that the regulation of visual task function leads to a final configuration of the manipulator in between the obstacles as shown in Fig. 4b. The gain of sensors on ring 1 and 2 are chosen as $\lambda_1 = 1$ and $\lambda_2 = 3$ and threshold distance is selected as $\delta^*_{i1} = 4$ cm and $\delta^*_{i2} = 8$ cm. As can be seen in Fig. 7a, visual task converges. First, the manipulator moves towards the target and encounters the obstacle on the right, which results in a change in the evolution of visual task function due to the change in control input seen around 2.5 s as observed in Fig. 7b for bringing back proximity sensors {2, 3, 4, 5} to the safe set, as seen in Fig. 7d. As expected, the robot moves to the left for avoiding collision with obstacle on the right. While moving close to the target around 4 s, sensors {16, 17} on ring 1 come close to virtual contact as can be seen in Fig. 7c and the control architecture ensures to satisfy safety. Immediately after, sensors {12, 13} come close to virtual contact on ring 2 from around 5 s. It can be noticed that the controller eventually converges to the final goal position while being close to virtual contact in both rings till the end of task execution and thereby maintaining safety. Joint velocities {5, 7} are effectively used by the control architecture to have this back and forth motion in between the obstacles to reach the target.

B. Case 2 : Local convergence of visual task function

In this scenario, as shown in Fig. 5, we create a workspace composed of four planar obstacles. Two planar obstacles are aligned approximately parallel to each other close to the same initial configuration as in Case 1. Two other planar obstacles are aligned with each other at an angle closer to the final goal to stop convergence of $\mathbf{e}_{4p}$ before its global minimum. The gain of sensors on both rings 1 and 2 are chosen as $\lambda_j = 1$ and threshold distance is selected as $\delta^*_{ij} = 4$ cm. As already said, we use the estimated value of target normal for sensors on ring 2 while evaluating the interaction matrix with (5). An explanation for this is provided in Section VI. As observed in Fig. 8a, the task $\mathbf{e}_{4p}$ converges to a local minimum with control input converging to very low value as seen in Fig. 8b. For the QP problem, this local solution is in fact the global one. The CBFs in Fig. 8c indicate that $\mathbf{e}_{\delta_{91}}$ corresponding to sensor 9 on ring 1 is close to the situation of virtual contact. The abrupt detection of sensor 9 on ring 1 results in jerky motion around 1 s as seen in Fig. 8b. Due to the choice of control design, the robot is able to navigate through this passage while being very close to the threshold distance without encountering oscillations after the jerky motion. At around 3 s, with closer detection of the blocking planes, change in control input can be noticed. Eventually from approximately 8 s, the inequality constraints prevent the robot from further advancing as can be seen in Fig. 8d with sensors {4, 12} on ring 2 being in virtual contact. The experiment is stopped when the norm of gradient of Lagrangian $||\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{0}, \kappa^*)||$ falls below the stopping condition value of $\epsilon_{qd} = 0.01$. It can also be observed that the robot does not undergo impulsive motions due to the good value of estimated surface normals as explained in Section VI.

C. Case 3: Behavior with Dynamic Obstacle

In the final case, we consider the situation when positioning task is achieved and then a dynamic obstacle interacts with it. The gain for sensors on ring 1 and 2 are chosen as $\lambda_1 = 1$ and $\lambda_2 = 3$ respectively and threshold distances are $\delta^*_{i1} = 4$ cm and $\delta^*_{i2} = 6$ cm respectively. The experiment starts from the initial configuration shown in Fig. 6a. A human operator then interacts with different parts of the ring by bringing the hands closer as shown in Fig. 6c and Fig. 6d. As seen in Fig. 9 the components of visual task function is close to zero at the beginning and becomes non-zero when the hand of the human operator is brought close to the robot. Towards the end of the experiment,



(a) Initial configuration

(b) Final configuration

(c) Interacting with ring 1

(d) Interacting with ring 2

Fig. 6: Experimental setup for Case 3 where a human operator interacts with each of the rings.

the components approach again the value of zero. Each time visual task function $\mathbf{e}_{4p}$ components are non-zero, control input components are also non-zero. It can be seen in Fig. 9d the four instances in the interval between 5 s to 18 s when proximity sensors on ring 2 reach close to virtual contact. In the later phase from approximately 20 s to 30 s, the hand interacts with ring 1 as seen in Fig. 9c. There are four instances when few sensors on ring 1 reach their threshold value. The results suggest that the control architecture works well even though obstacles were assumed to be static in the modelling.

VI. Discussion

In this section we discuss about the experimental results considered with regards to the control architecture provided. We start by providing general comments. The architecture considers all proximity detections separately throughout the experimentation. This enables it to be flexible in avoiding obstacles of any shape as long as detection occurs. From the point of view of QP solver, only the detections close to virtual contact matter in finding the optimal solution. A possible limitation may be related to the capacity of the solver in considering many constraints. However, in the experimental scenarios, the QP solver (ProxQP) was adequate in providing fast solutions. In the remaining paragraphs we describe more specific properties about the experimental results obtained.

In case when there is no obstacles in the workspace or if the obstacles are far away from the threshold distance, the active set $\mathcal{A}(\mathbf{u}^*)$ is empty. The solution of QP problem in this scenario reduces to the least-squares solution

$$\dot{\mathbf{q}} = -\lambda_{4p}(\mathbf{J}_{4p}^T\mathbf{J}_{4p} + \mu\mathbf{I})^{-1}\mathbf{J}_{4p}^T\mathbf{e}_{4p} \tag{19}$$

However there can also be another situation even when the active set is non-empty where the controller could choose such a solution. One example is when the manipulator has to pass through a narrow passage similar to the one seen in the initial part of the workspace for Case 2. Let us consider a scenario where a proximity ring j interacts with two parallel planes on either side with two virtual contacts and the goal position from visual servoing lies in between the parallel planes. In this case the active set would consist of the above two detections. The Lagrangian multipliers corresponding to the active set is zero and the term $\sum_{k\in\mathcal{A}(\mathbf{u}^*)}\kappa_k^*\mathbf{a}_k$ in KKT condition vanishes. Therefore the optimal solution can be obtained by solving the gradient of Lagrangian

$$\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}^*,\kappa^*) = \left(\mathbf{J}_{4p}{}^T\mathbf{J}_{4p} + \mu\mathbf{I}\right)\mathbf{u}^* - \mathbf{J}_{4p}^T\dot{\mathbf{e}}_{4p}^* = \mathbf{0} \tag{20}$$

which gives us the same optimal solution as in (19). It is thus possible to be at the desired threshold distance from each of the planes without experiencing oscillations.

If the obstacles are arranged so that they prevent the global convergence of the visual-task function, then we obtain strict complementarity condition in (15), i.e., $\kappa_k > 0$. From the KKT conditions (15), at convergence when $\mathbf{u}^* = 0$, we can see that the Lagrangian multipliers have a suitable value to resist the term associated with the vision-based task function

$$\mathbf{J}_{4p}^T\dot{\mathbf{e}}_{4p}^* = \sum_{k\in\mathcal{A}(\mathbf{u}^*)}\kappa_k^*\mathbf{a}_k = \sum_{k\in\mathcal{A}(\mathbf{u}^*)}\kappa_k^*\mathbf{J}_{e_{\delta_{ij}}}^T \tag{21}$$

For the final part of Case 2, when local convergence is achieved in vision-based task function, we have a particular case of having proximity sensors $\{4,\ 12\}$ at virtual contact from ring 2 while the camera is also rigidly attached to this ring. Feature Jacobian matrix associated with $\mathbf{e}_{4p}$ in this case can be expanded as $\mathbf{J}_{4p} = \mathbf{L}_{4p}\ {}^c\mathbf{V}_2\ {}^2\mathbf{V}_e\ \mathbf{J}_e$ where ${}^c\mathbf{V}_2$ is the twist transformation from camera frame $\mathcal{F}_c$ to the ring 2 frame $\mathcal{F}_{S_2}$, ${}^2\mathbf{V}_e$ is the twist transformation from $\mathcal{F}_{S_2}$ to end-effector frame $\mathcal{F}_e$, and $\mathbf{J}_e$ is the robot Jacobian expressed in $\mathcal{F}_e$. We can expand (21) as

$$\left(\mathbf{L}_{4p}\,{}^c\mathbf{V}_2\,{}^2\mathbf{V}_e\ \mathbf{J}_e\right)^T\dot{\mathbf{e}}_{4p}^* = \kappa_1(\mathbf{L}_{\delta_{42}}\ {}^2\mathbf{V}_e\ \mathbf{J}_e)^T + \kappa_2(\mathbf{L}_{\delta_{122}}\ {}^2\mathbf{V}_e\ \mathbf{J}_e)^T \tag{22}$$

Re-arranging the equation we get

$$\mathbf{J}_e^T\ {}^2\mathbf{V}_e^T\left((\mathbf{L}_{4p}\ {}^c\mathbf{V}_2)^T\dot{\mathbf{e}}_{4p}^* - \kappa_1\mathbf{L}_{\delta_{42}}^T - \kappa_2\mathbf{L}_{\delta_{122}}^T\right) = 0 \tag{23}$$

If we use coarse model (6) that is not based on the obstacle surface normal, then the above equation expands to

$$\mathbf{J}_e^T\ {}^2\mathbf{V}_e^T\left((\mathbf{L}_{4p}\ {}^c\mathbf{V}_2)^T\dot{\mathbf{e}}_{4p}^* + \begin{pmatrix}\kappa_1\mathbf{n}_{\mathbf{S}42} + \kappa_2\mathbf{n}_{\mathbf{S}122}\\ \mathbf{0}\end{pmatrix}\right) = \mathbf{0} \tag{24}$$

Term $\kappa_1\mathbf{n}_{\mathbf{S}42} + \kappa_2\mathbf{n}_{\mathbf{S}122}$ lies in the plane containing the ring and does not have a component in the direction perpendicular to it. This would create oscillations in the system as the vision-based task function is essentially trying to move the system to its corresponding minimum with a significant component of motion in this direction. This does not occur using model (5) with a correct estimation of the surface model.

For Case 1, when global convergence is achieved in visual task function at convergence, we can substitute $\mathbf{u}^* = 0$ and $\dot{\mathbf{e}}_{4p}^* = 0$ to the gradient of Lagrangian to obtain weak complementarity condition, i.e., $\kappa_k^* = 0$ when LICQ holds:

$$\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}^*,\kappa^*) = -\sum_{k\in\mathcal{A}(\mathbf{u}^*)}\kappa_k^*\mathbf{a}_k = \mathbf{0} \tag{25}$$

In all cases, the optimal solution at convergence is stable according to the stability analysis in Section IV.

Since the architecture is embedded in the framework of SBC, we obtain reactive motion that is robust to modelling

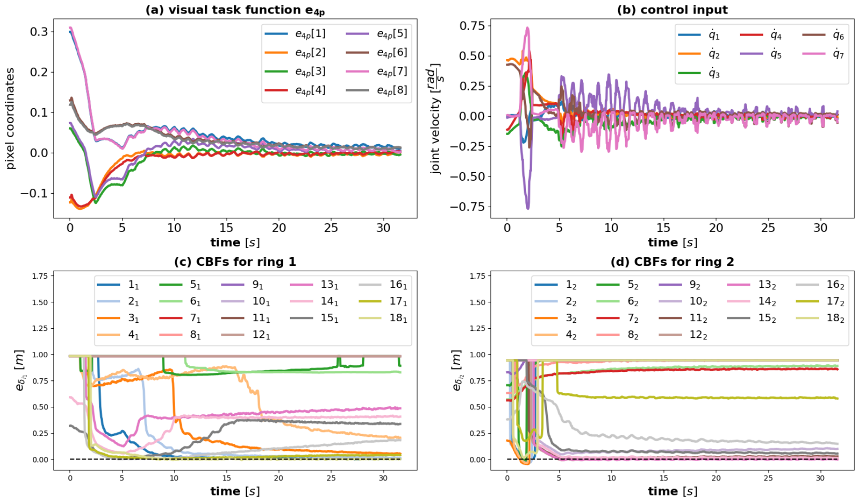


Fig. 7: Experimental results for Case 1 with two planar obstacles showing global convergence of visual task function. The plots consist of four parts. Top left plot indicates visual task function components represented by pixel coordinates versus time (s) and top right indicates joint velocity components (rad/s) versus time (s). The bottom plots indicate CBFs (m) versus time (s) for ring 1 (on bottom left) and for ring 2 (on bottom right).

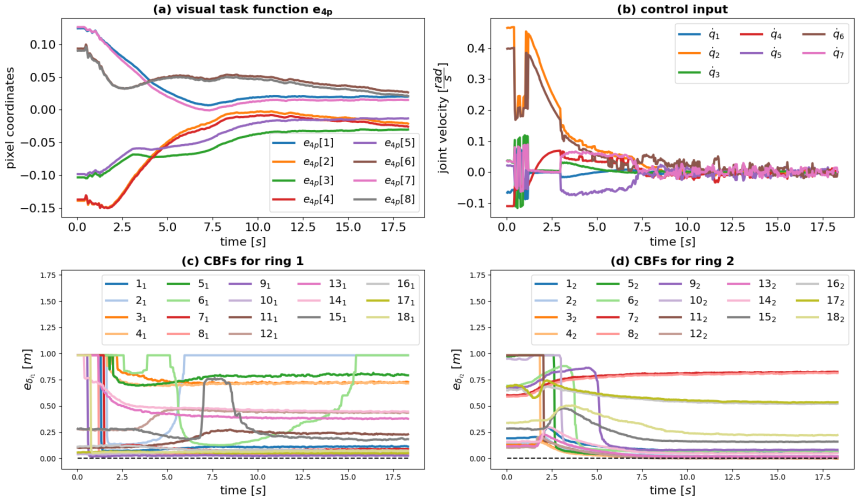


Fig. 8: Experimental results for Case 2 with two blocking planar obstacles showing a scenario of local convergence. The structure of the plot is the same as Fig. 7.

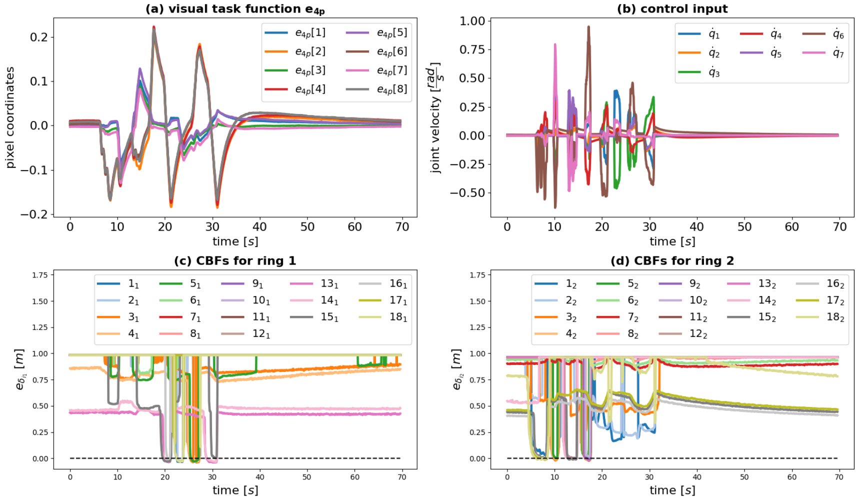


Fig. 9: Experimental results for Case 3 showing a scenario with dynamic obstacle. The structure of the plot is the same as Fig. 7.

uncertainties and stable while being in close encounter with obstacles. However, being a kinematic controller, it is not well adept to handle high speed moving obstacles.

VII. CONCLUSION

In this paper we have considered the task of positioning in congested space using SBC framework. For this we combine signals from camera and proximity sensors. Safety was achieved with the use of CBFs with proximity signals. For the task considered, QP framework was used as the control architecture to combine sensor information in vision space and proximity space. It also ensured that the complementary nature of the sensors were effectively utilized. We have considered three experiments that provide validation for the effectiveness of the control architecture considered. It is recommended to use (5) as model for the interaction matrix of proximity sensors as it was shown in Section VI that model (6) would fail in cases such as the one presented in Section V-B. In some of these experiments a positioning task is achieved while remaining at a maximal distance of 4 *cm* from obstacles, which is the limit of distance measurement of our proximity sensors. For these sensors the accuracy of data reduces around minimum distance. However, this did not cause problems in positioning as the solution is still stable. Additionally, such stability also holds for uncertainties involved in the modelling parameters.

The theoretical analysis was considered for static obstacles. In future the framework could be adapted to consider fast dynamic motions in the workspace. Miniaturization of the rings and application of the developed controller for human-robot collaboration are also other prospective research directions. It would be required to have more proximity sensors covering the whole body of the manipulator like a skin. This would also lead to more smoother reactions reducing abrupt changes since no more abrupt detection of obstacles would occur.

ACKNOWLEDGMENT

The authors want to thank François Pasteau for fabricating the proximity rings and Fabien Spindler for helping in the validation part of this work.

REFERENCES

[1] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics and Automation Mag.*, vol. 13, no. 4, pp. 82–90, 2006.

[2] S. E. Navarro, S. Mühlbacher-Karrer, H. Alagi, H. Zangl, K. Koyama, B. Hein, C. Duriez, and J. R. Smith, "Proximity perception in human-centered robotics: A survey on sensing systems and applications," *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 1599–1620, 2022.

[3] A. Cherubini and D. Navarro-Alarcon, "Sensor-based control for collaborative robots: Fundamentals, challenges, and opportunities," *Frontiers in Neurorobotics*, vol. 14, 2021.

[4] E. Cheung and V. Lumelsky, "Development of sensitive skin for a 3d robot arm operating in an uncertain environment," in *Int. Conf. on Robotics and Automation*, 1989, pp. 1056–1061 vol.2.

[5] V. Lumelsky and E. Cheung, "Real-time collision avoidance in teleoperated whole-sensitive robot arm manipulators," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 23, no. 1, pp. 194–203, 1993.

[6] J. Novak and I. Feddema, "A capacitance-based proximity sensor for whole arm obstacle avoidance," in *IEEE Int. Conf. on Robotics and Automation*, vol. 2, 1992, pp. 1307–1314.

[7] J. Feddema and J. Novak, "Whole arm obstacle avoidance for teleoperated robots," in *IEEE Int. Conf. on Robotics and Automation*, vol. 4, 1994, pp. 3303–3309.

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

[9] C. Vergara Perico, G. Borghesan, E. Aertbeliën, and J. Schutter, "Incorporating artificial skin signals in the constraint-based reactive control of human–robot collaborative manipulation tasks," *Industrial Robot*, vol. 46, 2019.

[10] P. Mittendorfer and G. Cheng, "3d surface reconstruction for robotic body parts with artificial skins," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012, pp. 4505–4510.

[11] G. Cheng, E. Dean-Leon, F. Bergner, J. Rogelio Guadarrama Olvera, Q. Leboutet, and P. Mittendorfer, "A comprehensive realization of robot skin: Sensors, sensing, control, and applications," *Proceedings of the IEEE*, vol. 107, no. 10, pp. 2034–2051, 2019.

[12] Y. Ding, F. Wilhelm, L. Faulhammer, and U. Thomas, "With proximity servoing towards safe human-robot-interaction," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2019, pp. 4907–4912.

[13] Y. Ding and U. Thomas, "Collision avoidance with proximity servoing for redundant serial robot manipulators," in *IEEE Int. Conf. on Robotics and Automation*, 2020, pp. 10 249–10 255.

[14] K.-E. M'Colo, B. Luong, A. Crosnier, C. Néel, and P. Fraisse, "Obstacle avoidance using a capacitive skin for safe human-robot interaction," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2019, pp. 6742–6747.

[15] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *IEEE Int. Conf. on Robotics and Automation*, vol. 2, 1985, pp. 500–505.

[16] B. Faverjon and P. Tournassoud, "A local based approach for path planning of manipulators with a high number of degrees of freedom," in *IEEE Int. Conf. on Robotics and Automation*, vol. 4, 1987, pp. 1152–1159.

[17] L. Devigne, V. K. Narayanan, F. Pasteau, and M. Babel, "Low complex sensor-based shared control for power wheelchair navigation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2016, pp. 5434–5439.

[18] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *European Control Conf.*, 2019, pp. 3420–3431.

[19] D. N. Nenchev, "Redundancy resolution through local optimization: A review," *J. Field Robotics*, vol. 6, pp. 769–798, 1989.

[20] D. J. Agravante, G. Claudio, F. Spindler, and F. Chaumette, "Visual servoing in an optimization framework for the whole-body control of humanoid robots," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 608–615, 2017.

[21] O. Kermorgant and F. Chaumette, "Dealing with constraints in sensor-based robot control," *IEEE Trans. on Robotics*, vol. 30, no. 1, pp. 244–257, 2014.

[22] C. Samson, B. Espiau, and M. L. Borgne, *Robot control: the task function approach*. Oxford University Press, Inc., 1991.

[23] J. Thomas, F. Pasteau, and F. Chaumette, "Plane-to-plane positioning by proximity-based control," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2022, pp. 12 795–12 802.

[24] H. Badino, D. Huber, Y. Park, and T. Kanade, "Fast and accurate computation of surface normals from range images," in *IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 3084–3091.

[25] B. Espiau, "Sensory-based control robustness issues and modelling techniques application to proximity sensing," in *Kinematic and Dynamic Issues in Sensor Based Control*, 1990, pp. 3–44.

[26] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[27] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.

[28] M. J. Best and N. Chakravarti, "Stability of linearly constrained convex quadratic programs," *Journal of Optimization Theory and Applications*, vol. 64, pp. 43–53, 1990.

[29] E. Marchand, F. Spindler, and F. Chaumette, "ViSP for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics and Automation Mag.*, vol. 12, no. 4, pp. 40–52, 2005.

[30] A. Bambade, S. El-Kazdadi, A. Taylor, and J. Carpentier, "PROX-QP: Yet another Quadratic Programming Solver for Robotics and beyond," in *Robotics: Science and Systems*, June 2022.