Plane-to-Plane Positioning by Proximity-based Control

John Thomas, François Pasteau, François Chaumette

Plane-to-Plane Positioning by Proximity-based Control

John Thomas[1], François Pasteau[2], François Chaumette[1]

***Abstract*— In this paper, we consider a multi-sensor arrangement of proximity sensors that forms a proximity array. A general modeling methodology is considered within the framework of Sensor-based Control. It incorporates multiple sensor signals from the proximity array by giving primary emphasis on the interaction screw. To prove its effectiveness, modeling approach is applied to the task of plane-to-plane positioning. We discuss the development of two sensor-based task functions for the specific task considered. The validity of the methodology is provided using relevant experimental results.**

***Index Terms* - Sensor-based Control, Proximity-based Control, Plane-to-Plane Positioning, Interaction Screw.**

I. INTRODUCTION

Sensor-based Control (SBC) is a very useful framework that enables the user to define sensor-based tasks. The regulation of such tasks enables a robot to achieve its desired behavior. Tasks executed using this framework have high robustness, reactive nature, accuracy and minimal dependence on prior knowledge of the environment. The most popular scheme in SBC is Visual Servoing [1], which refers to the use of data acquired by a camera. The ability of a camera to generate global information mimicking the human eye has been effectively used in visual servoing for performing tasks such as positioning, navigation, target tracking, assembly etc. However, the camera can become an unreliable sensing modality in few scenarios. For a passive camera that provides RGB data, feature identification from the observation of untextured objects is very challenging. Active ranging cameras that additionally provide depth images using either structured light or Time-of-Flight (ToF) technology can perform better in the case of untextured target objects. However, they become unreliable below the closest working depth of approx. 30 cm [2]. This limitation in minimum sensing distance, has been highlighted in [3] as the reason to use RGB images rather than RGB-D data for maintaining closed-loop positioning behaviour through visual servoing before grasping an object. Additionally, the camera may also get affected by occlusion, blind spots and has a relative large size to be used for close-range sensing. Proximity sensors are often suggested as the sensing modality with the potential to close this perception gap [4].

Devices that enable the detection of local objects through a physical form of interaction without contact is called Proximity Sensors [5]. Main technologies here such as capaciflective, infrared optical, and ultrasonic sensors [6] provide relative geometric data consisting of distance and orientation. A detailed overview regarding the use of proximity sensors in the context of human-centered robotics is provided in [4]. Proximity perception is envisioned here as the complementary bridge between vision and touch. Detection within 50 cm range of the target object can be considered ideal for use of proximity sensors.

Various types of proximity sensors have been used in the past for achieving specific tasks in robotic systems, such as mobile robots and manipulators. In [7], the authors proposed to use 500 infrared proximity sensors to cover the entire arm of a manipulator as a sensitive skin, to detect obstacles that approach the arm. This work was then extended in [8] with a similar sensitive skin, to address the problem of collision-avoidance during robotic teleoperation in space applications. In [9], various proximity sensor combinations were proposed to perform tasks including obstacle avoidance, guidance inside a pipe, plane-to-plane positioning, tracking a pin-point object, etc. In [10], WHole Arm Proximity (WHAP) sensor of capacitance type was mounted on a two-link planar robotic arm to avoid obstacles. This work was later extended in [11], where the WHAP sensor was used in a 6-DoF manipulator meant for teleoperation of hazardous waste. A proximity sensor system consisting of three ultrasound sensors, attached to the end-effector of a 6-DoF manipulator was used in [12], for contour-following task with applications to painting, welding, glue administration and surface polishing operations. A distributed sensor prototype consisting of around 20 infrared LED sensors was used to improve human safety for industrial activity involving human-robot interaction in [13]. Ultrasonic sensors were used in [14] to detect obstacles while performing navigation using a power wheel chair for patients with motor or visual impairments. In [15], proximity sensors were mounted around a wheelchair to provide assistive solutions while avoiding negative obstacles (curbs, steps, descending slopes, etc.). In [16] and [17], proximity sensing cuffs with both capacitive and ToF measurements were attached to a 7-DoF serial manipulator to achieve collision avoidance in applications where humans and robots share a common workspace. Sensitive skin consisting of capacitive sensors developed by FOGALE robotics [18] was used in [19] for obstacle avoidance in applications involving human-robot interactions.

In most of the previously cited studies, tasks associated with proximity sensors are designed primarily either in Cartesian space or operational space. There is a lack of literature on defining task functions in the proximity sensor space for multi-sensory systems. By building task functions

This work was supported by BPI France Lichie project

[1] J. Thomas and F. Chaumette are with Inria, Univ Rennes, CNRS, IRISA - Rennes, France. john.thomas@inria.fr

[2] F. Pasteau is with INSA, Univ Rennes, Inria, CNRS, IRISA - Rennes, France. Francois.Pasteau@irisa.fr

directly from proximity signals, we want to benefit from the advantages of SBC framework mentioned in the initial part of this section.

Our work can be considered a continuation of the seminal work in [9] where various sensor arrangements were considered performing sensor-based tasks using proximity signals. We have chosen a specific configuration with cylindrical arrangement of sensors that was used to perform guidance inside a pipe. We generalize this arrangement as a proximity array consisting of various proximity rings that could either be attached to the end-effector or around the robotic arm. We use this sensor arrangement to perform plane-to-plane positioning task, which essentially refers to the convergence of a robot end-effector to a relative pose wrt. a planar surface. Potential applications include surface inspection, docking, welding and painting. This task was performed with a specific parallel arrangement of proximity sensors in [9]. In [20], [21], a camera is augmented with a structured light system consisting of four laser pointers to achieve the task. We use proximity signals to execute the task in a more effective manner. Indeed, tasks executed with proximity signals have many additional benefits, including simpler task functions, ability to position much closer to the planar object compared to vision, and a more reactive nature due to its higher frequency.

The following part of the paper is structured in four sections. In Section II, we introduce the proximity sensor array and elaborates the modeling approach followed. Section III is dedicated to the various aspects in achieving plane-to-plane positioning task. Validation of the task considered is discussed in Section IV through experimental results. We end the paper by providing conclusions and future works that can be undertaken.

II. MODELING

Let us consider the modeling of a proximity sensor. We look at a thin film range finder in which detection occurs along the axis of the sensor. As depicted in Fig. 1, $\mathbf{n_S}$ denotes the unit vector indicating sensor axis and δ the distance measured by the sensor w.r.t. the target. $\mathbf{n_T}$ is the unit vector direction of the target surface normal at point T. F_S represents the sensor coordinate frame located at origin S and F_T represents the target frame. Let us now consider

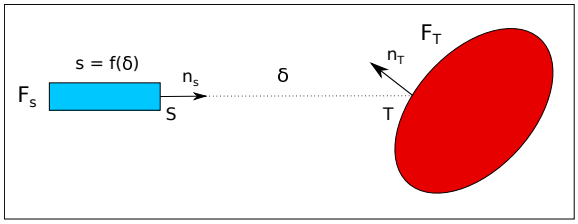


Fig. 1: Thin field proximity sensor reproduced from [22]. The blue rectangle indicates the sensor and red ellipse the target detected.

the sensor feature as the distance measured by the proximity sensor,

$$s = \delta \tag{1}$$

As usual in SBC [22], [1], the design of the control scheme is based on the interaction matrix $\mathbf{L}_s$, that relates the time variation of the sensor feature s to the sensor spatial velocity $\mathbf{v}_S$ under the form

$$\dot{s} = \mathbf{L}_s \mathbf{v}_S \tag{2}$$

The model for $\dot{\delta}$ for a motionless target is obtained from [22] as

$$\dot{\delta} = \frac{-1}{\mathbf{n_T} \cdot \mathbf{n_S}} (\mathbf{n_T} \cdot v_S + (\delta \mathbf{n_S} \times \mathbf{n_T}) \cdot \omega_S) \tag{3}$$

where v_S represents the translational velocity of sensor point S in F_S and ω_S represents the angular velocity of F_S. Let us note that the model for $\dot{\delta}$ in [22] was derived in target frame F_T, while we prefer to follow the convention set in visual servoing in terms of evaluating both interaction and velocity screws in the sensor frame F_S [1]. By considering $\mathbf{v_S} = (v_S, \omega_S)$ as the spatial velocity of the proximity sensor, we obtain the classical interaction matrix representation as

$$\dot{\delta} = \mathbf{L}_\delta \mathbf{v}_S \tag{4}$$

Let us assume that sensor axis $\mathbf{n_s}$ is along the z-axis of F_S. From (3), the interaction matrix $\mathbf{L}_\delta$ is given as

$$\mathbf{L}_\delta = \begin{bmatrix} -\frac{n_{T_x}}{n_{T_z}} & -\frac{n_{T_y}}{n_{T_z}} & -1 & \delta\frac{n_{T_y}}{n_{T_z}} & -\delta\frac{n_{T_x}}{n_{T_z}} & 0 \end{bmatrix} \tag{5}$$

where $\mathbf{n_T} = \begin{bmatrix} n_{T_x} & n_{T_y} & n_{T_z} \end{bmatrix}^T$. From (5), it seems to be important to evaluate the normal direction $\mathbf{n_T}$ of the target at detection point T for using a proximity sensor in a robotic task. However, thanks to the robustness of closed loop control scheme, we will see in Section III that it is not always necessary to estimate $\mathbf{n_T}$.

An alternative way of representing the variation of the proximity signal δ is to use an interaction screw. Let $\mathbf{H} = (\mathbf{H}(.), \mathbf{u})$ denote the interaction screw with $\mathbf{u}$ as its vector and $\mathbf{H}(.)$ as the value of the corresponding vector field. The link between the interaction screw and its matrix representation is obvious and is given by

$$\mathbf{L}_\delta = \begin{bmatrix} \mathbf{u}^T & \mathbf{H}(S)^T \end{bmatrix} \tag{6}$$

from which we deduce by identification with (3) and (5)

$$\begin{aligned} \mathbf{u} &= -\frac{\mathbf{n_T}}{\mathbf{n_T} \cdot \mathbf{n_S}} \\ \mathbf{H}(S) &= \delta \mathbf{n_S} \times \mathbf{u} \end{aligned} \tag{7}$$

By applying the shifting law, we can evaluate the screw value at the target point T. It becomes clear that $\mathbf{H}$ is a slider through T with direction along the local normal at point of detection [22] since

$$\mathbf{H}(T) = \mathbf{H}(S) + \vec{\mathbf{TS}} \times \mathbf{u} = \mathbf{0} \tag{8}$$

The above property can be effectively used at modeling stage for a system consisting of multiple proximity sensors. To evaluate the interaction matrix at any point in space, all we must do is to translate this slider from target point T to the desired point where we wish to control the spatial velocity.

In fact, in the above model, we have indeed translated this slider from target point T to the sensor point S along the position vector $\delta\mathbf{n_S}$.

We now introduce the proximity sensor system that is considered for performing the plane-to-plane positioning task. As depicted in Fig. 2, let us consider an array consisting of n rings with radii r_j and centers at S_j where $j = \{1, 2, ..., n\}$. Each ring consists of m_j proximity sensors arranged in such a way that the axis remains radial and passes through center S_j.

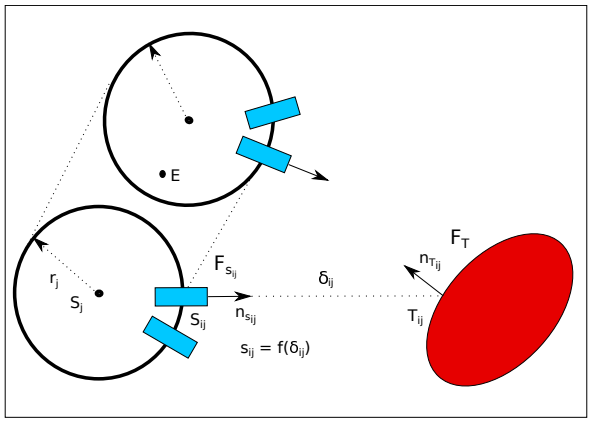


Fig. 2: Proximity sensor array consisting of n rings

A. Proximity Ring

Let us now consider the jth ring in the proximity array system. The interaction screw of the ith sensor in this ring where $i = \{1, 2, ..., m_j\}$ is $\mathbf{H}_{ij}$ and is evaluated based on the model in (7) at the respective sensor points S_{ij}. To obtain a common spatial velocity representation, we evaluate the model at ring center S_j. In terms of interaction, it means that we should shift the screw by displacement of $r_j\mathbf{n_{S_{ij}}}$. We thus obtain

$$\begin{aligned} \mathbf{u_{ij}} &= -\frac{\mathbf{n_{T_{ij}}}}{\mathbf{n_{T_{ij}}} \cdot \mathbf{n_{S_{ij}}}} \\ \mathbf{H}_{ij}(S_j) &= (\delta_{ij} + r_j)\mathbf{n_{S_{ij}}} \times \mathbf{u_{ij}} \end{aligned} \tag{9}$$

B. Proximity Array

To derive the model for the proximity array, we can further shift these interaction screws from ring centers to the end effector point E of the robot on which it is mounted. In that case, the interaction screw value for the ith proximity sensor attached to the jth ring is obtained as

$$\begin{aligned} \mathbf{u_{ij}} &= -\frac{\mathbf{n_{T_{ij}}}}{\mathbf{n_{T_{ij}}} \cdot \mathbf{n_{S_{ij}}}} \\ \mathbf{H_{ij}(E)} &= \mathbf{m_{S_{ij}}} \times \mathbf{u_{ij}} \end{aligned} \tag{10}$$

where

$$\mathbf{m_{S_{ij}}} = (\delta_{ij} + r_j)\mathbf{n_{S_{ij}}} + \vec{\mathbf{ES}}_\mathbf{j} \tag{11}$$

$\mathbf{m_{S_{ij}}}$ is essentially the overall displacement vector in shifting the slider from the target location to point E. Finally, we obtain the overall model for the proximity array as

$$\dot{\delta}_{ij} = \mathbf{L}_{\delta_{ij}}\mathbf{v_E} \tag{12}$$

with $\mathbf{L}_{\delta_{ij}} = \begin{bmatrix}\mathbf{u_{ij}}^T & \mathbf{H_{ij}}(E)^T\end{bmatrix}$.

In the above modeling stage, we have tried to emphasis on the use of an interaction screw. A common practise in SBC for modeling a multi-sensor system is to obtain the interaction matrix at individual sensor points and then transform it to a common point using the twist transformation matrix [1], [23]. Even though the final result would exactly be the same, by modeling it using screw displacements, we obtain a more concise set of equations for the case of proximity array.

III. PLANE-TO-PLANE POSITIONING

The proximity array model described above can be used for various robotic tasks, including guidance in a pipe, obstacle avoidance, etc. Here, we concentrate on the task of positioning wrt. a planar surface. The interaction screw attains a special form as the target normal at the point of detection is the same as the normal of the plane, *i.e.*, $\mathbf{n_{T_{ij}}} = \mathbf{n_T}$.

$$\begin{aligned} \mathbf{u_{ij}} &= -\frac{\mathbf{n_T}}{\mathbf{n_T} \cdot \mathbf{n_{S_{ij}}}} \\ \mathbf{H_{ij}(E)} &= \mathbf{m_{S_{ij}}} \times \mathbf{u_{ij}} \end{aligned} \tag{13}$$

The subspace spanned by interaction screws $S = \text{span}\{\mathbf{H_{11}}, \mathbf{H_{21}}, \mathbf{H_{12}}, ..\}$ essentially tells us the DoF that are controlled based on the defined sensor-based tasks. Since the interaction screw is a slider or 0-pitch screw at target point T_{ij}, we have a special 3-system of screws with a maximum dimension of three [24]. Hence, it is possible to control a maximum of 3-DoF of the system which consists of pure translation along the target normal and rotations perpendicular to it. At the final configuration when the sensor-based task is regulated to zero, we obtain the so-called virtual linkage, which is defined by the reciprocal subspace $\{\mathbf{V}^*\}$ of subspace S [22]. Elements of this subspace satisfy this condition, leaving the proximity signal invariant to such motions,

$$\dot{\mathbf{s}} = \mathbf{H} \cdot \mathbf{V}^* = 0 \tag{14}$$

In the case of plane-to-plane positioning, the reciprocal system can be easily identified as pure translations along the plane and pure rotation around $\mathbf{n_T}$. At an equilibrium pose, a virtual plane/plane contact of class C3 is achieved, where the class is defined as the dimension of subspace $\{\mathbf{V}^*\}$ [9], [25]. As plane-to-plane positioning leaves 3 DoF of the end-effector free, they can be used to implement a complementary task such as surface inspection.

In the following, we propose two fundamental sensor-based task functions that allow the robot to achieve the task. The first choice is to build a function $\mathbf{e}_1$ from the minimal number of sensors. We can select such a minimal set by selecting three sensors that result in independent screws. This requires a proximity array system of at least two rings for the model described above. If all three sensors are selected from the same ring, the target points would lie on a straight line on the plane and this collinearity would reduce our screw system to a 2-DoF system. This would therefore not allow us to

obtain a plane-to-plane positioning. The second task function $\mathbf{e}_2$ is based on a redundant number of sensors for obtaining better efficiency in the system. The standard control law used in visual servoing [1] is used for achieving the task,

$$\mathbf{v}_{E,k} = -\lambda \widehat{\mathbf{L}_{\mathbf{e}_k}}^{+} \mathbf{e}_k \, , k = 1,2 \tag{15}$$

where $\mathbf{v}_{E,k}$ is the velocity of the end-effector sent to the low-level robot controller, $\widehat{\mathbf{L}_{\mathbf{e}_k}}$ is an estimation or an approximation of the interaction matrix $\mathbf{L}_{\mathbf{e}_k}$, and $\widehat{\mathbf{L}_{\mathbf{e}_k}}^{+}$ is the Moore-Penrose pseudoinverse of $\widehat{\mathbf{L}_{\mathbf{e}_k}}$.

Fig. 3 describes the proximity array system along with the coordinate frames considered building functions $\mathbf{e}_1$ and $\mathbf{e}_2$. It consists of two rings with radii r_j, located at S_j with a distance d_j from the origin along z-axis of the end-effector frame where $j = \{1,2\}$. The coordinate frame of the rings are aligned in such a way that the x-y plane of the rings and end-effector are parallel. Each of these rings consists of sensors at S_{ij}, aligned by an angle α_{ij} wrt. x-axis where $i = \{1,2\}$. The extrinsic parameters of the array include the distance values r_j, d_j and angular values α_{ij}.

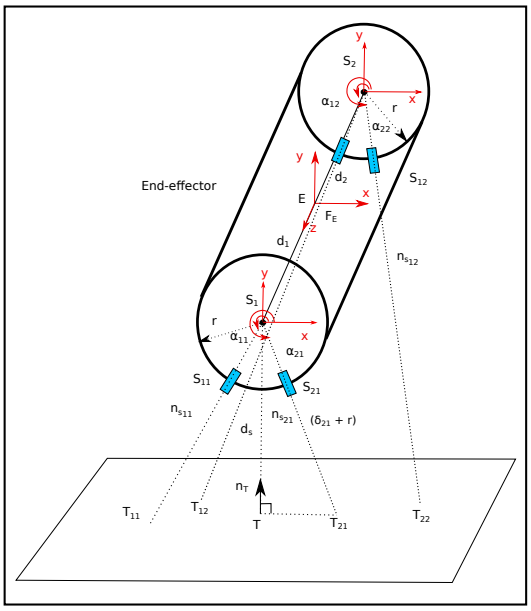


Fig. 3: Proximity array for plane-to-plane positioning task

A. Minimal sensors

The task function for the case of minimal sensors is built based on the current proximity signal values δ_{ij} and their desired value δ^*_{ij},

$$\mathbf{e}_1 = \begin{pmatrix} \delta_{11} - \delta^*_{11} \\ \delta_{21} - \delta^*_{21} \\ \delta_{12} - \delta^*_{12} \end{pmatrix} \tag{16}$$

The interaction screw of sensor feature $\mathbf{e}_1$ is obtained from the special form (13) in case of a planar target by substituting the chosen configuration for sensor axis $n_{S_{ij}}$.

$$\begin{aligned} \mathbf{n_{S_{ij}}} &= \cos\alpha_{ij}\,\hat{\mathbf{i}} + \sin\alpha_{ij}\,\hat{\mathbf{j}} \\ \mathbf{u_{ij}} &= -\frac{\mathbf{n_T}}{n_{Tx}\cos\alpha_{ij} + n_{Ty}\sin\alpha_{ij}} \\ \mathbf{m_{S_{ij}}} &= (\delta_{ij} + r_j)\cos\alpha_{ij}\,\hat{\mathbf{i}} \\ &\quad +(\delta_{ij} + r_j)\sin\alpha_{ij}\,\hat{\mathbf{j}} + d_j\hat{\mathbf{k}} \\ \mathbf{H}_{ij}(E) &= \mathbf{m_{S_{ij}}} \times \mathbf{u_{ij}} \end{aligned} \tag{17}$$

where we recall that $\mathbf{n_T} = \begin{bmatrix} n_{T_x} & n_{T_y} & n_{T_z} \end{bmatrix}^T$

The interaction matrix $\mathbf{L}_{\mathbf{e}_1}$ for this task is directly obtained from (12) and (17):

$$\mathbf{L}_{\mathbf{e}_1} = \begin{bmatrix} \mathbf{u}_{11}^T & \mathbf{H}_{11}(E)^T \\ \mathbf{u}_{21}^T & \mathbf{H}_{21}(E)^T \\ \mathbf{u}_{12}^T & \mathbf{H}_{12}(E)^T \end{bmatrix} \tag{18}$$

Since the target object is planar, we can evaluate its normal $\mathbf{n_T}$ directly from the proximity sensor readings. Indeed, from relative position vector $\mathbf{m_{S_{ij}}}$, we obtain three non-collinear vectors on the plane.

$$\begin{aligned} \mathbf{p_{1121}} &= \mathbf{m_{S_{11}}} - \mathbf{m_{S_{21}}} \\ \mathbf{p_{2112}} &= \mathbf{m_{S_{21}}} - \mathbf{m_{S_{12}}} \\ \mathbf{p_{1211}} &= \mathbf{m_{S_{12}}} - \mathbf{m_{S_{11}}} \end{aligned} \tag{19}$$

Since the normal $\mathbf{n_T}$ is perpendicular to each of these vectors, we can build a homogeneous system of the form,

$$\mathbf{A}\,\mathbf{n_T} = \mathbf{0} \tag{20}$$

where $\mathbf{A} = \begin{bmatrix} \mathbf{p_{1121}}^T \\ \mathbf{p_{2112}}^T \\ \mathbf{p_{1211}}^T \end{bmatrix}$. Nontrivial solution of (20) is the plane normal $\mathbf{n_T}$.

Simplified Model: In this subsection, we derive a constant interaction matrix that can be used as an approximation $\widehat{\mathbf{L}_{\mathbf{e}_1}}$ in the control scheme. This model is obtained primarily by evaluating the interaction screw vector and sensor signal values at equilibrium using the following substitution in (17), *i.e.*, $\delta_{ij} = \delta^*_{ij}$ and $\mathbf{n_T} = \mathbf{n^*_T}$.

$$\begin{aligned} \mathbf{u_{ij}} &= -\frac{\mathbf{n_T}^*}{\mathbf{n_T}^* \cdot \mathbf{n_{S_{ij}}}} \\ \mathbf{m^*_{S_{ij}}} &= (\delta^*_{ij} + r_j)\mathbf{n_{S_{ij}}} + d_j\hat{\mathbf{k}} \\ \mathbf{H_{ij}}(E) &= \mathbf{m^*_{S_{ij}}} \times \mathbf{u_{ij}} \end{aligned} \tag{21}$$

In that case, it is no longer necessary to estimate $\mathbf{n_T}$ at each iteration of the control scheme.

Special Case: Here we discuss the special case when the desired configuration is such that the target plane and end-effector x and z axes are parallel (see Fig. 3). This particular configuration is well adapted for the task of surface inspection. In that case, the target plane normal is aligned with the y-axis of the proximity array, *i.e.*, $\mathbf{n_T} = \hat{\mathbf{j}}$ and we can directly evaluate the desired value of the sensor signals, avoiding any off-line teaching step. Indeed, the desired values δ^*_{ij} can be evaluated from the right angled triangle that is formed with sensor axis $\mathbf{n_{S_{ij}}}$, y axis of end-effector, and

the line joining target point T_{ij} and point T, as shown in Fig. 3. Let d_S be the distance along the plane normal that the end-effector has to achieve wrt. the plane. From the right angled triangle ($\triangle S_j T T_{ij}$), the desired sensor signal value is obtained as

$$\delta_{ij}^* = -\frac{d_s}{\sin\alpha_{ij}} - r \tag{22}$$

The corresponding interaction matrix has the following simple form:

$$\mathbf{L}_{\mathbf{e_1}}^* = \begin{bmatrix} 0 & -\frac{1}{\sin\alpha_{11}} & 0 & \frac{d_1}{\sin\alpha_{11}} & 0 & \frac{d_s\cos\alpha_{11}}{\sin^2\alpha_{11}} \\ 0 & -\frac{1}{\sin\alpha_{21}} & 0 & \frac{d_1}{\sin\alpha_{21}} & 0 & \frac{d_s\cos\alpha_{21}}{\sin^2\alpha_{21}} \\ 0 & -\frac{1}{\sin\alpha_{12}} & 0 & \frac{d_2}{\sin\alpha_{12}} & 0 & \frac{d_s\cos\alpha_{12}}{\sin^2\alpha_{12}} \end{bmatrix} \tag{23}$$

It can be easily computed from the desired distance d_S and the extrinsic parameters $d_1, d_2, \alpha_{11}, \alpha_{12}$ and α_{21}

B. Redundant Sensors

Here, we look at a more effective way of defining the sensor feature using a redundant set of sensors. In a practical scenario, such a choice leads to several advantages [9]. First, the additional sensors increase the region of proximity sensing. Also, it allows introducing symmetry to the signal measurements at final pose. This can be exploited during modeling stage by designing the task function as a linear combination of few signals. We can thereby avoid using the final desired value of sensor signals to a large extent, which usually requires a separate teaching step. It also reduces noise effect in control and increases the efficiency of task execution.

Based on the coordinate frame description shown in Fig. 3, let us consider a symmetrical sensor arrangement in proximity array,

$$\begin{aligned} &\alpha_{11} = 270^o - \alpha,\ \ \alpha_{21} = 270^o + \alpha,\ \ d_1 = \ \ d,\ r_1 = r \\ &\alpha_{12} = 270^o - \alpha,\ \ \alpha_{22} = 270^o + \alpha,\ \ d_2 = -d,\ r_2 = r \end{aligned} \tag{24}$$

The symmetry of the above array can be exploited to select sensor features that are less dependent on the extrinsic parameters.

$$\mathbf{e}_2 = \mathbf{s} - \mathbf{s}^* \tag{25}$$

where

- $s = \begin{pmatrix} \delta_{11}+\delta_{12}-\delta_{21}-\delta_{22} \\ \delta_{11}+\delta_{21}-\delta_{12}-\delta_{22} \\ \delta_{11}+\delta_{21}+\delta_{12}+\delta_{22} \end{pmatrix}$
- $s^* = \begin{pmatrix} \delta_{11}^*+\delta_{12}^*-\delta_{21}^*-\delta_{22}^* \\ \delta_{11}^*+\delta_{21}^*-\delta_{12}^*-\delta_{22}^* \\ \delta_{11}^*+\delta_{21}^*+\delta_{12}^*+\delta_{22}^* \end{pmatrix}$

The interaction matrix $\mathbf{L}_{\mathbf{e}_2}$ for this task can be derived directly from the generic equations (12) and (17). It is given by

$$\mathbf{L}_{\mathbf{e}_2} = \begin{bmatrix} \mathbf{u}_{11}^T+\mathbf{u}_{12}^T-\mathbf{u}_{21}^T-\mathbf{u}_{22}^T & \mathbf{H}_{11}^T+\mathbf{H}_{12}^T-\mathbf{H}_{21}^T-\mathbf{H}_{22}^T \\ \mathbf{u}_{11}^T+\mathbf{u}_{21}^T-\mathbf{u}_{12}^T-\mathbf{u}_{22}^T & \mathbf{H}_{11}^T+\mathbf{H}_{21}^T-\mathbf{H}_{12}^T-\mathbf{H}_{22}^T \\ \mathbf{u}_{11}^T+\mathbf{u}_{21}^T+\mathbf{u}_{12}^T+\mathbf{u}_{22}^T & \mathbf{H}_{11}^T+\mathbf{H}_{21}^T+\mathbf{H}_{12}^T+\mathbf{H}_{22}^T \end{bmatrix} \tag{26}$$

Simplified Model: To obtain the interaction model at equilibrium, we used the same substitution used in the case of minimal sensors (III-A). Interaction screw values obtained using (21) can be further substituted into (26) to obtain the constant interaction matrix $\widehat{\mathbf{L}_{\mathbf{e}_2}}$.

Special Case: In the special case, we again look at the situation when the target plane is parallel to the x and z end-effector axes. The desired sensor features in case of symmetrical arrangement simplifies to

$$\mathbf{s}^* = \begin{pmatrix} 0 \\ 0 \\ \frac{4\,d_s}{\cos\alpha} - 4\,r \end{pmatrix} \tag{27}$$

which does not necessitate computing or learning the desired values δ_{ij}^*. Let us now evaluate the interaction at equilibrium. In that case, the interaction screw corresponding to an individual proximity sensor simplifies as

$$\begin{aligned} \mathbf{u_{11}} &= \frac{1}{\cos\alpha}\hat{\mathbf{j}} \ \ ; \ \ \mathbf{H}_{11}(E) = -\frac{d}{\cos\alpha}\hat{\mathbf{i}} - \frac{d_s\sin\alpha}{\cos^2\alpha}\hat{\mathbf{k}} \\ \mathbf{u_{21}} &= \frac{1}{\cos\alpha}\hat{\mathbf{j}} \ \ ; \ \ \mathbf{H}_{21}(E) = -\frac{d}{\cos\alpha}\hat{\mathbf{i}} + \frac{d_s\sin\alpha}{\cos^2\alpha}\hat{\mathbf{k}} \\ \mathbf{u_{12}} &= \frac{1}{\cos\alpha}\hat{\mathbf{j}} \ \ ; \ \ \mathbf{H}_{12}(E) = \frac{d}{\cos\alpha}\hat{\mathbf{i}} - \frac{d_s\sin\alpha}{\cos^2\alpha}\hat{\mathbf{k}} \\ \mathbf{u_{22}} &= \frac{1}{\cos\alpha}\hat{\mathbf{j}} \ \ ; \ \ \mathbf{H}_{22}(E) = \frac{d}{\cos\alpha}\hat{\mathbf{i}} + \frac{d_s\sin\alpha}{\cos^2\alpha}\hat{\mathbf{k}} \end{aligned} \tag{28}$$

from which we deduce the very simple form

$$\mathbf{L}_{\mathbf{e}_2}^* = \frac{4}{\cos\alpha}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -d_s\tan\alpha \\ 0 & 0 & 0 & -d & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{29}$$

Hence, we can obtain a perfectly decoupled control scheme with our choice of sensor arrangement and the use of desired values in the interaction matrix $\mathbf{L}_{\mathbf{e}_2}^*$.

C. Stability Analysis

As is well known in SBC, the system is globally asymptotically stable (GAS) if the following property is ensured [1],

$$\mathbf{L}_{\mathbf{e}_k}\widehat{\mathbf{L}_{\mathbf{e}_k}}^+ > 0 \tag{30}$$

In this study, $\mathbf{L}_{\mathbf{e}_k}\widehat{\mathbf{L}_{\mathbf{e}_k}}^+$ is a 3x3 matrix and we have $\mathbf{L}_{\mathbf{e}_k}\widehat{\mathbf{L}_{\mathbf{e}_k}}^+ = \mathbf{I}_3 > 0$ when $\widehat{\mathbf{L}_{\mathbf{e}_k}} = \mathbf{L}_{\mathbf{e}_k}$, i.e., when the interaction matrix used in the control scheme corresponds to the real one. This is the case when using the actual model (18) for $\mathbf{e}_1$ and (26) for $\mathbf{e}_2$ if the target normal is correctly evaluated and if no calibration errors occur. Thus, the system is GAS in that case. On the other hand, when using the simplified models (23) and (29) in the control scheme, we can only state that the system is locally asymptotically stable since $\mathbf{L}_{\mathbf{e}_k}\mathbf{L}_{\mathbf{e}_k}^{*\,+} = \mathbf{I}_3$ only at the desired configuration $\mathbf{s} = \mathbf{s}^*$. However, we will see in the next section that the domain of convergence is very large even when using these simplified models.

IV. EXPERIMENTAL RESULTS

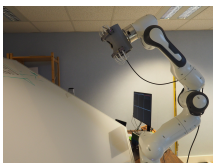
In this section, we show the validation of our model for performing the described task. We consider three scenarios for the experiments starting from the same initial pose shown in Fig. 4a. The initial pose is chosen such that the proximity signal readings are just within the range of 50 *cm*, which is considered a suitable operational range for proximity sensors [4]. In the first case, we show convergence to an arbitrary final pose, where the proximity measurements are less than 30 cm, as shown in Fig. 4c, which is approx. the minimum sensing distance for depth measurements using active ranging cameras. These values are obtained offline before the experiment and are used as the desired sensor signal values for building task functions $\mathbf{e}_1$ and $\mathbf{e}_2$. In the second case, the robot must converge to a final pose shown in Fig. 4d, where the x-z plane of end-effector is parallel to the target plane using the simplified model $\mathbf{e}_2$. For this special case, we evaluate the desired sensor features based on (22) with value $d_s = 12$ *cm*, where the proximity signal readings are approx. 6 *cm*, to build task function $\mathbf{e}_2$. Once the robot reaches this final pose, we apply a sequence of control inputs to perform a surface inspection. Finally, in Case 3, we attempt to display the reactive nature of the control law. We repeat the task performed in Case 2 without performing surface inspection, while moving the target plane abruptly close to equilibrium. In the first two cases, the gain λ of the control scheme (15) is set to $\lambda = 0.8$, whereas for the final case we use $\lambda = 1$. We attach a video displaying the robot behavior as a supplementary material to complete the results.

A. Experimental Setup

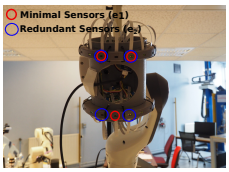
The experimental setup is displayed in Fig. 4. It consists of a Panda robot, which is a 7 DoF serial manipulator developed by Franka Emika. Attached to its end-effector is our proximity array consisting of two half rings. Each of these half-rings contains VL53L1x proximity sensors from STMicroelectronics [26]. These sensors use ToF technology, and are tuned to short range, to output measurements at 66 Hz frequency with 1 *mm* precision and 5^o FoV. The minimum ranging distance of the sensor was 4 *cm*. Fig. 4b displays the sensors chosen from these half-rings to build task functions $\mathbf{e}_1$ and $\mathbf{e}_2$. The extrinsic parameters of the proximity array model used for both $\mathbf{e}_1$ and $\mathbf{e}_2$ are shown in Table I. These values are measured using a scale and a protractor. From the results shown below, we prove the robustness of the control law in handling sensor systems that are not perfectly calibrated. Experimental software for control and communication is executed on a laptop with Intel® Core™ i7 CPU @ 1.90GHz × 8. Laptop runs on Ubuntu 20.04.2 LTS with RTLinux kernel in Fully Preemptible Mode.

B. Case 1

In the first scenario, we consider functions $\mathbf{e}_1$ and $\mathbf{e}_2$ to converge to an arbitrary final pose (shown in Fig. 4c). The desired sensor signal δ^*_{ij} obtained from offline teaching, is shown in Table II. Additionally, the table also contains the

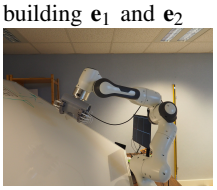


(a) Initial Pose for Case 1, Case 2 and Case 3.



(b) Sensors chosen for building $\mathbf{e}_1$ and $\mathbf{e}_2$

(c) Final pose for Case 1



(d) Final pose for Case 2

Fig. 4: Experimental setup

$\mathbf{e}_1$		$\mathbf{e}_2$	
$\alpha_{11} = 240^o$	$\alpha_{21} = 280^o$	$\alpha_{11} = 240^o$	$\alpha_{21} = 280^o$
$\alpha_{12} = 285^o$		$\alpha_{12} = 265^o$	$\alpha_{22} = 305^o$
$d_1 = 5.5$ *cm*, $r_1 = 7$ *cm*, $d_2 = -5.5$ *cm*, $r_2 = 7$ *cm*			

TABLE I: Extrinsic parameters of proximity array.

desired target normal used for the simplified models and the initial proximity signal values (δ^0_{ij}). Fig. 5 displays the plots

$\mathbf{e}_1$		$\mathbf{e}_2$	
$\delta^0_{11} = 48.5$ *cm*	$\delta^*_{11} = 10.3$ *cm*	$\delta^0_{11} = 48.5$ *cm*	$\delta^*_{11} = 10.3$ *cm*
$\delta^0_{21} = 47.8$ *cm*	$\delta^*_{21} = 12.7$ *cm*	$\delta^0_{21} = 47.8$ *cm*	$\delta^*_{21} = 12.7$ *cm*
$\delta^0_{12} = 43.5$ *cm*	$\delta^*_{12} = 16.4$ *cm*	$\delta^0_{12} = 45.9$ *cm*	$\delta^*_{12} = 14.5$ *cm*
		$\delta^0_{22} = 45.1$ *cm*	$\delta^*_{22} = 14.7$ *cm*
$\mathbf{n_T}^* = 0.33\,\hat{\mathbf{i}} + 0.92\,\hat{\mathbf{j}} - 0.19\,\hat{\mathbf{k}}$		$\mathbf{n_T}^* = 0.32\,\hat{\mathbf{i}} + 0.89\,\hat{\mathbf{j}} - 0.31\,\hat{\mathbf{k}}$	

TABLE II: Parameters for Case 1.

of task function and end-effector velocity for $\mathbf{e}_1$ and $\mathbf{e}_2$ using both actual and simplified interaction matrices in the control scheme (*i.e.*, in (15), $\widehat{\mathbf{L}_{\mathbf{e}_k}} = \mathbf{L}_{\mathbf{e}_k}$ with estimation of the target normal at each iteration for the actual model, and $\widehat{\mathbf{L}_{\mathbf{e}_k}} = \mathbf{L}^*_{\mathbf{e}_k}$, which is a constant matrix, for the simplified one). We first note that all control schemes allow the task to be achieved. In subplots (a) and (c), where actual models are used for $\mathbf{e}_1$ and $\mathbf{e}_2$, we observe a small deviation from the expected decoupled exponential decrease of the task functions due to the calibration errors existing in the experimental setup. This deviation is more pronounced for $\mathbf{e}_1$. In both (b) and (d) where simplified models are used for $\mathbf{e}_1$ and $\mathbf{e}_2$, we observe a more direct convergence of the end-effector velocities, which is more satisfactory for practical applications. This is quite evident by observing the rotational components of velocity in both (a) and (c) where the actual models are used. This rotational aspect is directly related to the nature of interaction screws, which allows to control two rotational components perpendicular to the target normal and only one translational component that is along the normal. The task execution time is also less in both (b) and (d) compared to its counterparts (a) and (c). Furthermore, the velocities computed using simplified models are less noisy in both (b)

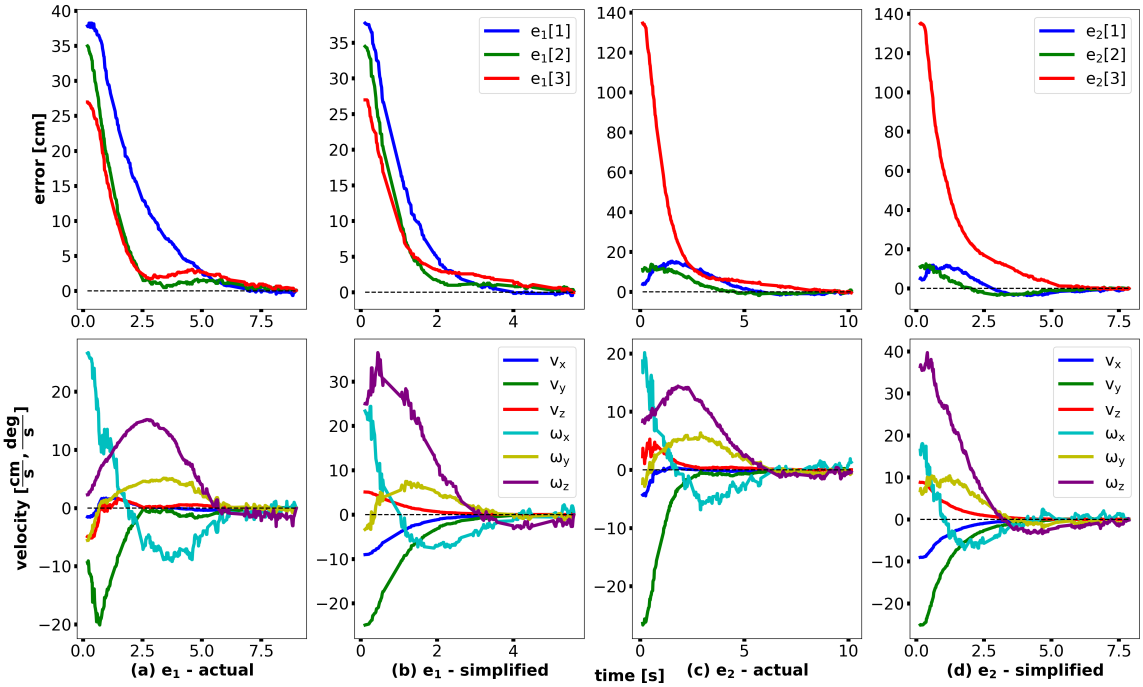


Fig. 5: Experimental results for Case 1 using both task functions $\mathbf{e}_1$ (on the left) and $\mathbf{e}_2$ (on the right) with actual ((a) and (c)) and simplified ((b) and (d)) interaction models. Task function components (cm) versus time (s) on the top, end-effector velocities (cm/s and deg/s) versus time (s) on the bottom).

and (d) due to the use of a constant interaction matrix in the control law, whereas for actual models, the current value of the interaction matrix is influenced by both sensor measurement noise and target normal estimation errors. This results in the noisy velocities observed in subplots (a) and (c). Also, compared to (b) we observe less noise in (d) since we use one additional proximity sensor compared to the minimal case. From the above results, we would recommend the use of (d), that is, with a redundant number of sensors and the simplified model. It is to be noted that the high precision of proximity sensors enables the robot end-effector to achieve high accuracy for positioning task when offline-teaching method is used to measure the desired sensor features.

C. Case 2

In this scenario, we use task $\mathbf{e}_2$ with the simplified model to show the special case where the end-effector reaches a pose parallel to the target plane at a distance $d_s = 12\,cm$, as shown in Fig. 4d. Once the system has converged, we implement a basic surface inspection task by applying translational velocities in the reciprocal subspace of the interaction screws. In the proximity array frame, we apply a constant velocity of 8 cm/s along the z-axis and x-axis in sequence to enable the end-effector to follow a rectangular trajectory. As shown in Fig. 6, the control scheme effectively regulates $\mathbf{e}_2$ to zero and thereby ensuring the parallel configuration necessary to inspect the surface. The choice of final pose (where target normal is aligned with the y-axis of end-effector) results exactly zero values for components v_x, v_z and ω_y computed in case of simplified models (remember the particular form

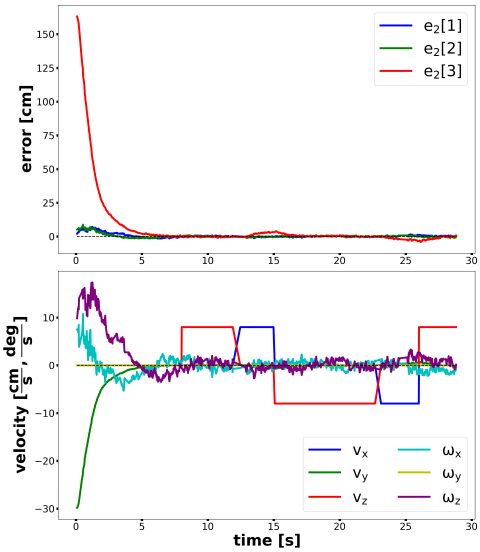


Fig. 6: Experimental results for Case 2 using task function $\mathbf{e}_2$ with simplified interaction model. Task function components (cm) versus time (s) on the top, end-effector velocities (cm/s and deg/s) versus time (s) on the bottom.

of $\mathbf{L}^*_{\mathbf{e}_2}$ in (29)). Even though the sensor arrangement for the experimental case is not exactly symmetric, we still obtain a performance close to the decoupled behaviour in (29). This can be observed in Fig. 6, by comparing the behaviours in the following pairs, $(\mathbf{e}_2[1], w_z)$, $(\mathbf{e}_2[2], w_x)$ and $(\mathbf{e}_2[3], v_y)$. It is also important to note that due to errors in extrinsic

parameters and bad accuracy of proximity sensors close to 4 *cm* from the target, the final configuration of the end-effector can be a bit inaccurate.

D. Case 3

This final scenario, task $\mathbf{e}_2$ is executed using a simplified model as that of Case 2. At equilibrium, we perform sequences of a translation and a rotation to the plane target. For lack of space, we provide results in the attached video. It displays how the control scheme rapidly reacts to such unexpected motions of the target object. The gain of $\lambda = 1$, is chosen for the experimentation empirically to obtain faster reaction than previous cases.

V. CONCLUSIONS

In this work, we have presented a modeling strategy for a generic proximity sensor arrangement. We have carefully looked at its application for performing plane-to-plane positioning and validated it through experimentation. From the model, it is clear that the knowledge about the target normal is important to accurately characterize the interaction through proximity sensors. However, from the experimental results, we conclude that using the simplified model that does not rely on such knowledge in the control law provides a satisfactory behaviour, particularly when considering redundant sensors. The convergence domain is very large in practice, even though it was demonstrated to be only locally asymptotically stable. In the future, we would try to determine this region of convergence from a theoretical point of view and obtain relevant stability results in the presence of calibration errors. Additionally, we would also like to use the proximity sensor arrangement and modeling approach for satisfying other tasks such as guidance and obstacle avoidance.

ACKNOWLEDGMENT

The authors want to thank Fabien Spindler and Alexander Oliva for their help in the validation part of this work.

REFERENCES

[1] F. Chaumette and S. Hutchinson, "Visual servo control, Part I: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006. [Online]. Available: https://hal.inria.fr/inria-00350283

[2] G. Alenyà, S. Foix, and C. Torras, "Tof cameras for active vision in robotics," *Sensors and Actuators A: Physical*, vol. 218, pp. 10–22, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924424714003458

[3] J. Haviland, F. Dayoub, and P. Corke, "Control of the final-phase of closed-loop visual grasping using image-based visual servoing," 2020. [Online]. Available: https://arxiv.org/abs/2001.05650

[4] 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.

[5] L. France, A. Girault, J.-D. Gascuel, and B. Espiau, "Sensor modeling for a walking robot simulation," in *Computer Animation and Simulation '99*, N. Magnenat-Thalmann and D. Thalmann, Eds. Vienna: Springer Vienna, 1999, pp. 189–198.

[6] M. R. Cutkosky, R. D. Howe, and W. R. Provancher, *Force and Tactile Sensors*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 455–476. [Online]. Available: https://doi.org/10.1007/978-3-540-30301-5_20

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

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

[9] B. Espiau, "Sensory-based control robustness issues and modelling techniques application to proximity sensing," in *Kinematic and Dynamic Issues in Sensor Based Control*, G. E. Taylor, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990, pp. 3–44.

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

[11] J. Feddema and J. Novak, "Whole arm obstacle avoidance for teleoperated robots," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, 1994, pp. 3303–3309 vol.4.

[12] U. Nunes, P. Faia, and A. de Almeida, "Sensor-based 3-d autonomous contour-following control," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, vol. 1, 1994, pp. 172–179 vol.1.

[13] 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 Transactions on Control Systems Technology*, vol. 22, no. 6, pp. 2127–2140, 2014.

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

[15] L. Devigne, F. Pasteau, T. Carlson, and M. Babel, "A shared control solution for safe assisted power wheelchair navigation in an environment consisting of negative obstacles: a proof of concept," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019, pp. 1043–1048.

[16] 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)*, 2019, pp. 4907–4912.

[17] Y. Ding and U. Thomas, "Collision avoidance with proximity servoing for redundant serial robot manipulators," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 10 249–10 255.

[18] "FOGALE-robotics - sensitive surfaces for human / robot interactions & cooperation," https://www.fogale-robotics.com/, accessed: 2022-7-8.

[19] 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 *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 6742–6747.

[20] J. Pages, C. Collewet, F. Chaumette, and J. Salvi, "Plane-to-plane positioning from image-based visual servoing and structured light," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 1, 2004, pp. 1004–1009 vol.1.

[21] ——, "Optimizing plane-to-plane positioning tasks by image-based visual servoing and structured light," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 1000–1010, 2006.

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

[23] O. Kermorgant and F. Chaumette, "Multi-sensor data fusion in sensor-based control: Application to multi-camera visual servoing," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 4518–4523.

[24] K. H. Hunt, *Kinematic geometry of mechanisms*. Oxford, England: Clarendon Press, 1978.

[25] F. Chaumette, P. Rives, and B. Espiau, "Classification and realization of the different vision-based tasks," in *Visual Servoing*, ser. World Scientific Series in Robotics and Intelligent Systems, 1993, vol. 7, pp. 199 – 228. [Online]. Available: https://hal.inria.fr/hal-01548352

[26] "VL53L1X," https://www.st.com/en/imaging-and-photonics-solutions/vl53l1x.html, accessed: 2022-7-8.