Use of Screw Theory in Proximity-based Control

John Thomas, François Chaumette

Use of Screw Theory in Proximity-based Control

John Thomas, François Chaumette

***Abstract*— In this short paper, we discuss the utility of screw theory in proximity-based control. In this scheme, a task function is defined based on the proximity signals, whose regulation obtains the appropriate robotic behavior. Screw theory can be used in modelling a multi-sensor arrangement of proximity sensors. Additionally, it can also be used in identifying the set of control directions that result in identical path in sensor-space while the task is being executed.**

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

I. INTRODUCTION

Proximity sensors can be considered as devices that enable the detection of local objects through non-contact interaction [1]. Main technologies such as capaciflective, infrared optical, and ultrasonic sensors [2] provide relative geometric data consisting of distance and orientation. In this work, we consider proximity sensors with thin field assumption where the distance measurements can be approximated along proximity axis. In recent years, proximity perception has been envisioned as a complementary bridge between vision and touch [3]. Robotic tasks involved with detection within 50 cm range of the target object could be considered ideal for use of proximity sensors.

Proximity-based Control (PBC) is considered here as a part of Sensor-based Control (SBC) framework that enables the user to define specific sensor-based task functions [4]. 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 [5], which refers to the use of camera data in closed loop control.

II. MODELING

Let us consider the modeling of a proximity sensor. We look at a thin field 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 wrt. the target. $\mathbf{n}_T$ is the unit vector direction of the target surface normal at point T. Let F_S denote the frame attached to the proximity sensor at point S.

A sensor feature s is a function of sensor signals that are C^2 functions with range in $\mathbb{R}$ and domain in $\mathbf{SE}(3)$ [4]. Its derivative wrt. time can be represented as a screw product,

$$\dot{s} = H \cdot V \tag{1}$$

This work was supported by BPI France Lichie project

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

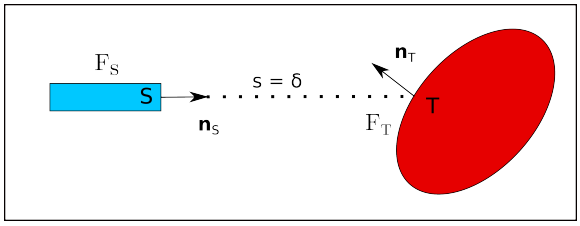

Fig. 1: Thin field proximity sensor reproduced from [4]. The blue rectangle indicates the sensor and red ellipse the target detected.

where V represents the velocity screw of proximity sensor (frame F_S) wrt. target object (frame F_T) and $H = (H(.), \mathbf{u})$ denote the interaction screw that characterizes the variation of sensor feature. It has $\mathbf{u}$ as its vector and $H(.)$ as the value of the corresponding vector field and is defined as an element of $\mathbf{se}^*(3)$, the dual space of lie algebra of $\mathbf{SE}(3)$. For a choice of sensor feature $s = \delta$, we obtain H [4] as,

$$\begin{aligned} \mathbf{u} &= -\frac{\mathbf{n}_T}{\mathbf{n}_T \cdot \mathbf{n}_S} \\ H(S) &= \delta \mathbf{n}_S \times \mathbf{u} \end{aligned} \tag{2}$$

In the above formulation, a further simplification is considered for the context of current paper by assuming the target to be static. V becomes the velocity screw of proximity sensor wrt. a fixed frame, whereas in the general case target tracking requires to be considered [5].

The corresponding matrix representation is,

$$\dot{\delta} = \mathbf{L}\mathbf{v} \tag{3}$$

where $\mathbf{L}$ is called interaction matrix and $\mathbf{v}$ is the vector representation of the velocity screw. $\mathbf{L}$ is related to interaction screw as follows

$$\mathbf{L} = \begin{bmatrix} \mathbf{u}^T & H(.)^T \end{bmatrix} \tag{4}$$

By applying the shifting law in screw theory, we can evaluate the screw value at the target point T. It becomes clear that H is a slider through point T with direction along the local normal at point of detection [4] since

$$H(T) = H(S) + \vec{TS} \times \mathbf{u} = \mathbf{0} \tag{5}$$

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.

We now introduce a proximity sensor system that could be attached to the arm of a manipulator or as an end-effector

to perform various robotic tasks including plane-to-plane positioning, guidance and obstacle avoidance. As depicted in Fig. 2, let us consider an array consisting of n rings with radii r_j and centers at R_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 point R_j.

To obtain a common spatial velocity representation, we evaluate the model at an arbitrary point E on the proximity array i.e., either attached as an end-effector or to an intermediate link. In terms of interaction, it means that we should shift the screw from target point to point E. The interaction

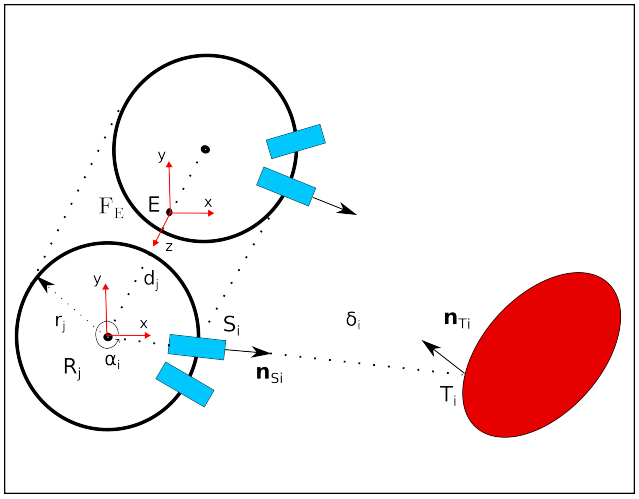

Fig. 2: Proximity sensor array consisting of n rings

screw value for the ith proximity sensor is obtained as [6],

$$\begin{aligned} \mathbf{u}_i &= -\frac{\mathbf{n}_{T_i}}{\mathbf{n}_{T_i} \cdot \mathbf{n}_{S_i}} \\ H_i(E) &= \mathbf{m}_{T_i} \times \mathbf{u}_i \end{aligned} \tag{6}$$

$\mathbf{m}_{T_i}$ is essentially the displacement vector in shifting the slider from the target location to point E. For the proximity array frame considered in Fig. 2, which is also used in Sec. IV, we have,

$$\begin{aligned} \mathbf{m}_{T_i} &= (\delta_i + r_j)\mathbf{n}_{S_i} + \vec{ER}_j \\ \mathbf{n}_{S_i} &= \cos\alpha_i\,\hat{\mathbf{i}} + \sin\alpha_i\,\hat{\mathbf{j}} \\ \vec{ER}_j &= d_j\hat{j} \end{aligned} \tag{7}$$

III. CONTROL SYNTHESIS

In PBC, we define a task function ($\mathbf{e}$) using several sensor features represented as $\mathbf{s}$. A simple definition of the function can be considered as $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$, where $\mathbf{s}^*$ is the desired sensor features at equilibrium. Control synthesis in PBC can be interpreted as to finding a suitable velocity that would obtain a desired behavior ($\dot{\mathbf{e}}^*$) in task space. This would correspond to the solutions of the linear equation shown below,

$$\dot{\mathbf{e}}^* = \mathbf{L}\mathbf{v} \tag{8}$$

The general solution to the above equation can be expressed in terms of a generalized inverse $\mathbf{G}$, which satisfies the condition $\mathbf{LGL} = \mathbf{L}$ [7] as,

$$\mathbf{v} = \mathbf{G}\dot{\mathbf{e}}^* \tag{9}$$

For minimal tasks, where the dimension of the task function correspond to the number of controlled DoF, one set of generalized inverse solutions can be represented as,

$$\mathbf{G} = \mathbf{P_i}\mathbf{L}^- \tag{10}$$

where $\mathbf{P}_i$ is a matrix that satisfies the property $\mathbf{LP}_i = \mathbf{L}$ and $\mathbf{L}^-$ is matrix that satisfies the property $\mathbf{LL}^- = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. $\mathbf{L}^-$ can be obtained from the screw system formed by the proximity sensors used in the task function using the reciprocity property. Here $\mathbf{G}$ is a reflexive generalized (left-)inverse [7] of $\mathbf{L}$ since it satisfies

$$\begin{aligned} \mathbf{LGL} &= \mathbf{L} \\ \mathbf{GLG} &= \mathbf{G} \\ (\mathbf{LG})^{\mathbf{T}} &= \mathbf{LG} \end{aligned} \tag{11}$$

A generic control law that leads to identical path in task space would be,

$$\mathbf{v} = \mathbf{G}\dot{\mathbf{e}}^* = \mathbf{P_i}\mathbf{L}^-\dot{\mathbf{e}}^* \tag{12}$$

Knowledge from screw-system enables us to obtain closed-form equations of the closed loop system formed based on the above feedback law. In a practical task in SBC, it is common practise to choose desired task behaviour as exponential reduction ($\dot{\mathbf{e}}^* = -\lambda\mathbf{e}$), to obtain straight line path in task space in ideal case. This leads to the following closed loop system,

$$\dot{\mathbf{e}} = -\lambda\mathbf{L}\widehat{\mathbf{G}}\mathbf{e} = -\lambda\mathbf{L}\widehat{\mathbf{P}_i}\widehat{\mathbf{L}^-}\mathbf{e} \tag{13}$$

where symbol $\widehat{\ }$ indicates estimation or approximation of the corresponding matrices.

A. PLANE-TO-PLANE POSITIONING TASK

In this subsection we apply the above approach to Plane-to-Plane positioning task. It refers to the convergence of a robot end-effector to a relative pose 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.*, $\mathbf{n_{T_i}} = \mathbf{n_T}$ in (6). Since the interaction screw is a slider at target point T_i, we have a special 3-system of screws with a maximum dimension of three [8]. This allows us to control a maximum of 3-DoF of the system which consists of pure translation along the target normal and rotations perpendicular to it. Additionally, at equilibrium we are able to perform planar motions. This allows us to implement a complementary task such as surface inspection.

To achieve this task we consider detection from three proximity sensors attached to two rings from the proximity array shown in Fig. 2. The task function for the case of minimal sensors is built based on the current proximity signal values δ_i and their desired value δ_i^*,

$$\mathbf{e} = \begin{pmatrix} \cdot \\ \delta_i - \delta_i^* \\ \cdot \end{pmatrix}_{3\times 1} \tag{14}$$

The first order kinematics of the task can be represented using interaction matrix $\mathbf{L}$ corresponding to the task and proximity array velocity $\mathbf{v_E}$ at point E,

$$\dot{\mathbf{e}} = \mathbf{L}\mathbf{v_E} \tag{15}$$

where

$$\mathbf{L} = \begin{bmatrix} \cdot & \cdot \\ \mathbf{u}_i^T & H_i(E)^T \\ \cdot & \cdot \end{bmatrix}_{3\times 6} \tag{16}$$

From the screw system we can consider the following,

$$\begin{aligned} \mathbf{u}_i^T(\mathbf{n}_T\mathbf{n}_T{}^T) &= \mathbf{u}_i^T = \mathbf{u}_i^T\mathbf{I_3} \\ H_i(E)^T(\mathbf{n}_T\mathbf{n}_T{}^T) &= \mathbf{0^T} \end{aligned} \tag{17}$$

Therefore, we can identify the set of matrices $\mathbf{P}_i \in \mathbf{Q}$ as

$$\begin{aligned} \mathbf{Q} = \{&\mathbf{I_6}, \begin{bmatrix} \mathbf{n}_T\mathbf{n}_T{}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I_3} - \mathbf{n_T}\mathbf{n_T^T} \end{bmatrix}, \begin{bmatrix} \mathbf{n}_T\mathbf{n}_T{}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I_3} \end{bmatrix} \\ &, \begin{bmatrix} \mathbf{I_3} & \mathbf{0} \\ \mathbf{0} & \mathbf{I_3} - \mathbf{n_T}\mathbf{n_T^T} \end{bmatrix}, \ldots\} \end{aligned} \tag{18}$$

$\mathbf{L}^-$ can be obtained by observing the interaction screws at the target. As depicted in Fig. 3, the red lines indicate the screws aligned along plane normal.

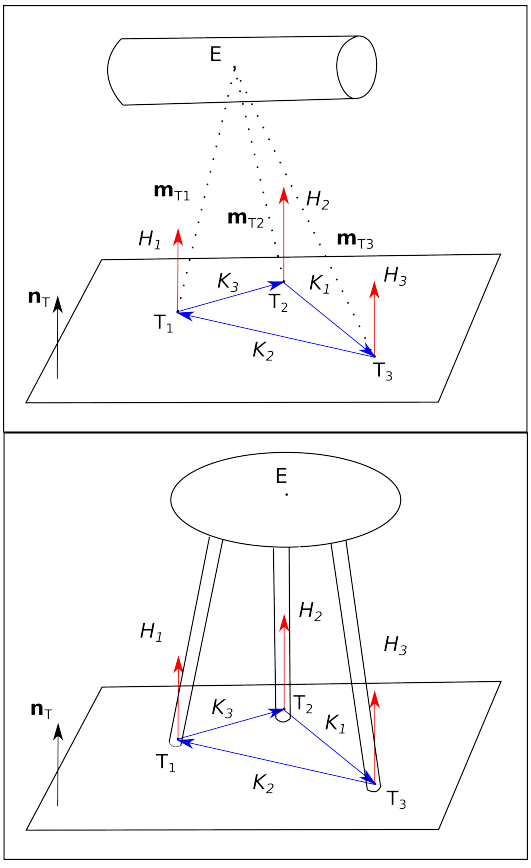

Fig. 3: Interaction screws and control directions at target plane (top), and equivalent mechanical structure (bottom).

The column vectors of $\mathbf{L}^-$ are in fact pure rotations passing through pairs of target points. We denote them as $K_i = (K_i(.), \mathbf{v}_i)$, which indicate pure rotation non-reciprocal to interaction screw H_i at target point T_i and reciprocal to the other interaction screws, *i.e.*, such that

$$K_i = (\mathbf{0}\ ,\ \mathbf{m_{T_{i-}}} - \mathbf{m_{T_{i+}}})\ , \quad \text{along } \overleftrightarrow{T_{i-}T_{i+}} \tag{19}$$

where $i+$ refers to the number after i and $i-$ refers to the number before i, when $1,2,3$ are represented as a circular stack.

These control directions can be called as actuation twist from its similarity in logic to actuation wrenches in parallel robotics. At the proximity array point E, these screws can be evaluated from shifting law (5) as

$$K_i = (\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}}\ ,\ \mathbf{m}_{T_{i-}} - \mathbf{m}_{T_{i+}}) \tag{20}$$

By scaling the above screws using the norms of interaction screw directions $\mathbf{u}_i$ and quantity l to obtain $\mathbf{L}\mathbf{L}^- = \mathbf{I_3}$ we get,

$$\mathbf{L}^- = \frac{1}{l}\begin{bmatrix} \cdot & \frac{\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}}}{\|\mathbf{u}_i\|} & \cdot \\ \cdot & \frac{\mathbf{m}_{T_{i-}} - \mathbf{m}_{T_{i+}}}{\|\mathbf{u}_i\|} & \cdot \end{bmatrix} \tag{21}$$

where $l = \mathbf{n}_T \cdot (\mathbf{m}_{T_1} \times \mathbf{m}_{T_2} + \mathbf{m}_{T_2} \times \mathbf{m}_{T_3} + \mathbf{m}_{T_3} \times \mathbf{m}_{T_1})$. The term $(\mathbf{m}_{T_1} \times \mathbf{m}_{T_2} + \mathbf{m}_{T_2} \times \mathbf{m}_{T_3} + \mathbf{m}_{T_3} \times \mathbf{m}_{T_1})$ in l has a geometric meaning. It is twice the area $(\mathbf{a}_i)$ sum of three sides of a tetrahedron containing point E. We can recollect here that the sum of vector area of a closed surface is zero and hence as shown in Fig. 4 we get,

$$\begin{aligned} l &= \mathbf{n}_T \cdot (2\mathbf{a}_1 + 2\mathbf{a}_2 + 2\mathbf{a}_3) = \mathbf{n}_T \cdot (-2\mathbf{a}_4) \\ &= \mathbf{n}_T \cdot \left(-\left(\mathbf{m}_{T_2} - \mathbf{m}_{T_3}\right) \times \left(\mathbf{m}_{T_1} - \mathbf{m}_{T_3}\right)\right) \\ &> 0 \end{aligned} \tag{22}$$

Singularity for this task would occur when the points are collinear and this would correspond to $l = 0$. Since l appears in the denominator, this would result in infinite velocity input while using $\mathbf{L}^-$ in control law. As pointed out in [4], the

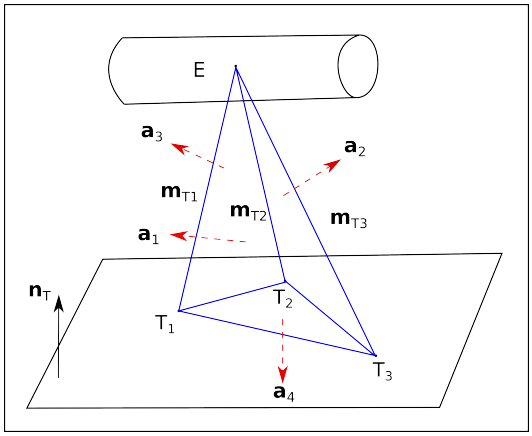

Fig. 4: Tetrahedron formed during plane-to-plane following task.

formalism of screws can also be used to describe rigid body contact. When proximity sensors are used, the interaction screw is along the surface normal at the point of detection, resembling frictionless single point contact [9]. This has been identified in [10], [4] while describing sensor-based

primitives. For our specific task, since there are three points of detection through three different proximity sensors, we can imagine an equivalent mechanical structure such as three legged stool resting on a frictionless floor, as shown in Fig. 3. Such a visualization of mechanical structure has become insightful in SBC in recent years through hidden robot concept [11] [12], where an equivalent parallel robot is used for analysing visual servoing tasks.

Among the several control directions the unique inverse that satisfies all four of Moore-Penrose conditions for the task considered is,

$$\mathbf{L}^{+} = \mathbf{P}\mathbf{L}^{-} \tag{23}$$

where

$$\mathbf{P} = \begin{bmatrix} \mathbf{n}_T \mathbf{n}_T{}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 - \mathbf{n}_T \mathbf{n}_T{}^T \end{bmatrix} \tag{24}$$

Matrix $\mathbf{P}$ is an orthogonal projection matrix as it is idempotent $\left(\mathbf{P^2} = \mathbf{P}\right)$ and symmetric [13]. The first submatrix of $\mathbf{P}$ is a projection along target normal $\mathbf{n}_T$. The second submatrix is a vector rejection operation from target normal $\mathbf{n}_T$.

Matrix $\mathbf{L}^{+}$ in addition to (11) also satisfies,

$$(\mathbf{L}^{+}\mathbf{L})^T = \mathbf{L}^{+}\mathbf{L} \tag{25}$$

While using $\mathbf{L}^{-}$ in (12), extrinsic parameters of the proximity array contribute significantly to control directions through displacement vector $\mathbf{m}_{T_i}$. In case of $\mathbf{L}^{+}$, due to the involvement of $\mathbf{n}_T$ in the projection matrix $\mathbf{P}$, target normal would contribute more.

IV. SIMULATION RESULTS

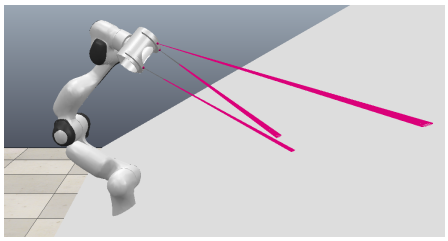

Fig. 5: Simulation setup in Coppeliasim. It consists of a proximity array system consisting of two half rings and three sensors attached to the end-effector of a 7 DoF serial manipulator.

In this section we provide simulation results while assuming that we have perfect knowledge of the model. Simulation was done using **FrankaSim** [14], a simulator based on **ROS** and **Coppeliasim** for **Panda** robot from **Franka Emika**. The simulator is integrated with **ViSP** [15], which is used as the main software library for programming. Fig. 5 displays the simulation setup in Coppeliasim. It consists of Panda robot with proximity array as the end-effector, detecting a planar target at an orientation of 30^o wrt. lateral axis of the base frame of the robot. The array consists of two half rings with two sensors on the first and remaining one on the second. Extrinsic parameters of the proximity array are shown in Table I for the task considered. The desired configuration is chosen as a parallel configuration with a distance of 20 *cm* along the y axis of the frame F_E. The initial pose is chosen such that the proximity signal readings are near to 50 *cm*, which is considered a suitable operational range for proximity sensors [3]. The gain λ for both control scheme is set to $\lambda = 0.8$. We execute the task using two controllers, one with $\mathbf{L}^{+}$ and the other with $\mathbf{L}^{-}$. As shown in Fig. 6, we obtain identical path in task space while the output of the control scheme, and thus the robot trajectory, are different.

$\alpha_1 = 250^o$	$\alpha_3 = 270^o$	$d_1 = 5.5\ cm$	$r_1 = 7\ cm$
$\alpha_2 = 290^o$		$d_2 = -5.5\ cm$	$r_2 = 7\ cm$

TABLE I: Extrinsic parameters of proximity array.

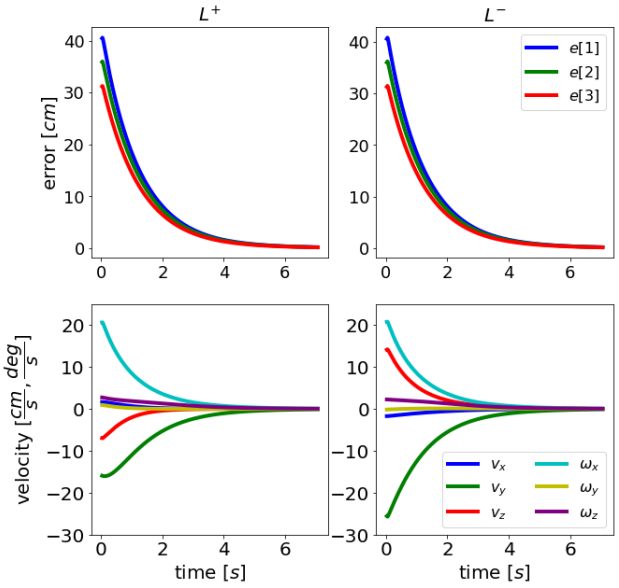

Fig. 6: Simulation results for perfect model with minimal number of sensors using both $\mathbf{L}^{+}$ and $\mathbf{L}^{-}$. Task function components (cm) versus time (s) on the top, end-effector velocities (cm/s and dg/s) versus time (s) in the bottom.

V. CONCLUSIONS

In this work, we emphasis on the utility of screw theory to design control schemes using proximity sensors, that behave as a closed loop system with the environment. It aids in the modelling stage and also in choosing the appropriate control law. In addition, the closed-form equations of the closed loop system can further enable in the study of stability of the task execution wrt. model parameters. It would be interesting to see if such solutions are also obtained in situations when the screw-systems get slightly more complex.

REFERENCES

[1] 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.

[2] M. R. Cutkosky, R. D. Howe, and W. R. Provancher, *Force and Tactile Sensors*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 455–476.

[3] 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.

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

[5] F. Chaumette, S. Hutchinson, and P. Corke, "Visual servoing," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds., 2016, pp. 841–866.

[6] J. Thomas, F. Pasteau, and F. Chaumette, "Plane-to-plane positioning by proximity-based control," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 12 795–12 802.

[7] C. R. Rao, "Generalized inverse of a matrix and its applications," in *Volume 1 Theory of Statistics*, L. M. L. Cam, J. Neyman, and E. L. Scott, Eds. University of California Press, 1972, pp. 601–620.

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

[9] R. Featherstone, "Contact and impact," in *Rigid Body Dynamics Algorithms*. Boston, MA: Springer US, 2008, pp. 213–239.

[10] 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., 1990, pp. 3–44.

[11] S. Briot, F. Chaumette, and P. Martinet, "Revisiting the determination of the singularity cases in the visual servoing of image points through the concept of hidden robot," *IEEE Trans. on Robotics*, vol. 33, no. 2, Jan. 2017.

[12] S. Briot, P. Martinet, and F. Chaumette, "Determining the Singularities for the Observation of Three Image Lines," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Singapore, May 2017.

[13] G. Strang, *Linear algebra and its applications*, 4th ed. Belmont, CA: Thomson, Brooks/Cole, 2006.

[14] A. Oliva, F. Spindler, P. Robuffo Giordano, and F. Chaumette, "Frankasim: A dynamic simulator for the franka emika robot with visual-servoing enabled capabilities," in *17th Int. Conf. on Control, Automation, Robotics and Vision (ICARCV)*, Singapore, 2022.

[15] 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 Magazine*, vol. 12, no. 4, pp. 40–52, December 2005.