



HAL
open science

Stability Analysis of Plane-to-Plane Positioning by Proximity-based Control

John Thomas, François Chaumette

► **To cite this version:**

John Thomas, François Chaumette. Stability Analysis of Plane-to-Plane Positioning by Proximity-based Control. IEEE Robotics and Automation Letters, 2023, 8 (8), pp.4473-4480. 10.1109/LRA.2023.3284353 . hal-04106631

HAL Id: hal-04106631

<https://inria.hal.science/hal-04106631>

Submitted on 25 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Stability Analysis of Plane-to-Plane Positioning by Proximity-based Control

John Thomas, François Chaumette

Abstract—In this paper, we discuss the stability analysis of Plane-to-Plane positioning task when the task is designed in proximity sensor space. We utilize a multi-sensor arrangement of proximity sensors that forms a proximity array to obtain the necessary information in sensor space. For the task considered, we provide closed-form equations for the closed-loop system by obtaining the analytical form of pseudo-inverse for the interaction matrix involved. This further enables us to suggest a new control law producing a decoupled exponential decrease of the sensor errors in perfect conditions, while being more robust to estimation errors in the surface normal. By applying Gershgorin’s theorem to the closed-form matrices, we are able to provide conditions for stability with respect to errors in extrinsic parameters and surface normal. Simulation results are provided to discuss the robustness of the task with respect to these modeling parameters.

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

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 (VS) [1], which refers to the use of data acquired by a camera. Proximity sensors have also been intensively used for achieving specific tasks in robotic systems. In most of the past works such as [2–7], tasks associated with proximity sensors are designed primarily either in Cartesian space or operational space. A detailed overview regarding the use of proximity sensors in the context of human-centered robotics is provided in [8]. Typical use of proximity sensors concerns obstacle avoidance while the plane-to-plane positioning task considered in this paper may be involved for the inspection of large industrial texture-less parts.

In practice, closed loop nature of the control scheme makes the system stable and robust wrt. modeling parameters. Even though the domain of convergence is often found experimentally to be quite large, it is usually extremely difficult to formally define this region. From a theoretical point of view, stability can be analysed using Lyapunov theory [9]. Applied to sensor-based control, two different situations occur [1]. One in which the number of features chosen as inputs of the control scheme enables the user to control an equal number of degree of freedom (DoF) of the

system. In this case, it is possible to obtain global asymptotic stability (GAS) in ideal case with the use of perfect model. In the second case, redundant number of features are used to control the system and hence only local asymptotic stability (LAS) can be demonstrated in ideal situation due to the occurrence of non-trivial null space [1]. The task considered in this paper falls into the first category.

More precisely, in Pose-Based Visual Servoing (PBVS) where 6 DoF of the camera are controlled using 6 pose parameters as features, GAS is ensured under perfect pose estimation [1]. However, for imperfect estimations, the stability analysis becomes impossible due to the non-availability of an analytical form relating the image measurements to the estimated pose parameters [10]. In hybrid approaches like 2.5 D VS, where the system is partitioned to control orientation separately, by combining features expressed in the image space and in 3D, GAS is again obtained in ideal scenario of perfect model [11]. Moreover, sufficient conditions of GAS and both necessary and sufficient conditions for LAS wrt. camera calibration errors were also obtained in [11]. The study of stability has been further extended to few other controllers in [12], where necessary and sufficient conditions for global stability of a hybrid control scheme similar to 2.5 D is provided wrt. camera calibration errors. What enables such an in-depth analysis is partly due to the choice of sensor features that results in decoupling the control of camera rotation. This leads to an interaction matrix that is upper triangular, which allows obtaining the analytical form of the inverse of the interaction matrix, while it is a crucial step in the stability analysis. When features are defined purely in image-space (IBVS), theoretical analysis of stability is hindered due to the lack of technique in approaching the highly non-linear and coupled character of the closed loop equation of the system [13]. In [14], plane-to-plane positioning task is performed using IBVS by combining it with structured light. A careful choice of features results in a decoupled interaction matrix in which GAS is ensured in ideal case.

In this paper, we study the stability of plane-to-plane positioning task where, as in our previous work [15], 3 sensor features in proximity space are used to control 3 DoF of the end-effector. These sensor features are defined either using a minimal or redundant number of proximity sensors. The interaction matrix in each case is coupled and non-linear. As a first step towards understanding about the system’s stability wrt. modeling parameters, we obtain in this paper the analytical form of the pseudo-inverse of interaction matrix for both cases. This analytical form enables us to

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

obtain a closed-form equation for the closed loop system, which is otherwise rather difficult for a task with coupled and non-linear interaction matrix. We then simplify the standard matrix positivity condition by providing regional bounds to the eigenvalues with the application of Gershgorin's circle theorem. Additionally, the analytical form also enables us to suggest a new controller based on a generalized inverse that is less subject to surface normal errors (unknown parameter of the environment), still guaranteeing an exponential decoupled decrease of the sensor errors in ideal conditions. Based on the stability conditions obtained, we analyse the influence of both extrinsic parameters and surface normal errors towards the execution of the task with both classical (pseudo-inverse based) and new controllers for both minimal and redundant number of proximity sensors.

This paper is divided into five sections including the current one. In Section II, we recall from [15] basic aspects of plane-to-plane positioning task. Section III is devoted to the stability analysis. In this section we provide the analytical form of pseudo-inverse, suggestion for a new control law, and closed-form matrices that need to be evaluated. Section III ends with the stability conditions on these matrices for few scenarios with the application of Gershgorin's theorem. In Sections IV, we provide validation to the theory through simulation results. Finally we end the paper with conclusions and future work in Section V.

II. PLANE-TO-PLANE POSITIONING TASK

In this section we recall the modeling of proximity sensor mentioned in [15]. 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 wrt. the target. \mathbf{n}_T is the unit vector direction of the surface normal at point T. F_S represents the sensor coordinate frame located at origin S and F_T represents the target frame. Let

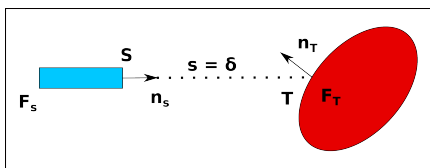


Fig. 1: Thin field proximity sensor reproduced from [16]. The blue rectangle indicates the sensor and red ellipse the surface detected.

us now consider the sensor feature as the distance measured by the proximity sensor, $s = \delta$. As usual in SBC [1,16], 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

$$\dot{s} = \mathbf{L}_S \mathbf{v}_S \quad (1)$$

The model for $\dot{\delta}$ for a motionless target is obtained from [16] as

$$\dot{\delta} = \frac{-1}{\mathbf{n}_T \cdot \mathbf{n}_S} (\mathbf{n}_T \cdot \mathbf{v}_S + (\delta \mathbf{n}_S \times \mathbf{n}_T) \cdot \boldsymbol{\omega}_S) \quad (2)$$

where \mathbf{v}_S represents the translational velocity of sensor point S in F_S and $\boldsymbol{\omega}_S$ represents the angular velocity of F_S . By considering $\mathbf{v}_S = (v_S, \boldsymbol{\omega}_S)$ as the spatial velocity of the proximity sensor, we obtain the classical interaction matrix representation as

$$\mathbf{L}_S = \begin{bmatrix} \mathbf{u}^T & \mathbf{H}(S)^T \end{bmatrix} \quad (3)$$

with

$$\mathbf{u} = -\frac{\mathbf{n}_T}{\mathbf{n}_T \cdot \mathbf{n}_S}, \quad \mathbf{H}(S) = \delta \mathbf{n}_S \times \mathbf{u} \quad (4)$$

Here $\mathbf{H} = (\mathbf{H}(\cdot), \mathbf{u})$ denotes the interaction screw with \mathbf{u} as its vector and $\mathbf{H}(\cdot)$ as the value of the corresponding vector field. By applying the shifting law, we can evaluate the screw value at the target point T , from which it is clear that \mathbf{H} is a slider through T with direction along the local normal at point T [16] since

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

The above property is 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 surface point T to the desired point where we wish to control the spatial velocity.

Plane-to-Plane positioning task refers to the convergence of a robot end-effector to a relative pose wrt. a planar surface. To achieve this task we considered in [15] an arrangement of proximity sensors as shown in Fig. 2 for the particular case of a minimal number of sensors. The proximity array consists of two rings that are arranged in such a way that the planes containing the rings are parallel to the x-y plane of the end-effector frame F_E . Each of the proximity sensors is located at S_i , aligned by an angle α_i wrt. x-axis of F_E . Extrinsic parameters of the array consist of distance values r_i, d_i and angular values α_i . Here r_i corresponds to the radius of the ring on which the proximity sensor is attached and d_i corresponds to the distance of the ring from the origin along z-axis of F_E .

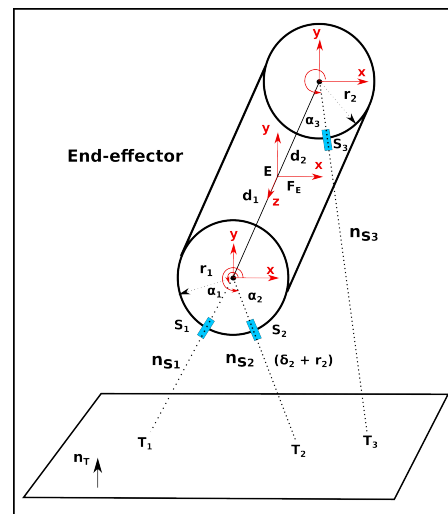


Fig. 2: Proximity array considered for plane-to-plane positioning in the case of a minimal number of 3 sensors.

The task function \mathbf{e}_m for the case of minimal sensors is built based on the current proximity signal values δ_i and their desired value δ_i^* :

$$\mathbf{e}_m = \begin{pmatrix} \delta_i - \delta_i^* \\ \vdots \\ \delta_i - \delta_i^* \end{pmatrix}_{3 \times 1} \quad (6)$$

The interaction matrix $\mathbf{L}_{\mathbf{e}_m}$ for the task considered corresponds to

$$\mathbf{L}_{\mathbf{e}_m} = \begin{bmatrix} \dot{\mathbf{u}}_i^T & \mathbf{H}_i(E)^T \\ \vdots & \vdots \end{bmatrix}_{3 \times 6} = \begin{bmatrix} {}^m\boldsymbol{\beta}_i \mathbf{n}_T^T & ({}^m\mathbf{m}_{\beta_i} \times \mathbf{n}_T)^T \\ \vdots & \vdots \end{bmatrix}_{3 \times 6} \quad (7)$$

with

$$\begin{aligned} \mathbf{u}_i &= -\frac{\mathbf{n}_T}{\mathbf{n}_T \cdot \mathbf{n}_{S_i}} = {}^m\boldsymbol{\beta}_i \mathbf{n}_T \\ \mathbf{H}_i(E) &= \mathbf{m}_{S_i} \times \mathbf{u}_i = {}^m\mathbf{m}_{\beta_i} \times \mathbf{n}_T \\ \mathbf{n}_{S_i} &= \cos \alpha_i \hat{\mathbf{i}} + \sin \alpha_i \hat{\mathbf{j}} \\ \mathbf{m}_{S_i} &= (\delta_i + r_i) \cos \alpha_i \hat{\mathbf{i}} + (\delta_i + r_i) \sin \alpha_i \hat{\mathbf{j}} + d_i \hat{\mathbf{k}} \\ {}^m\boldsymbol{\beta}_i &= -\frac{1}{\mathbf{n}_T \cdot \mathbf{n}_{S_i}}, \quad {}^m\mathbf{m}_{\beta_i} = {}^1\boldsymbol{\beta}_i \mathbf{m}_{S_i} \end{aligned} \quad (8)$$

When a redundant number of sensors are used, the task function \mathbf{e}_r can be expressed as a linear combination of the sensors. In [15], the task function considered using four proximity sensors (two on each ring) is:

$$\mathbf{e}_r = \mathbf{C} \begin{pmatrix} \delta_i - \delta_i^* \\ \vdots \\ \delta_i - \delta_i^* \end{pmatrix}_{4 \times 1} \quad (9)$$

where $\mathbf{C} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$.

The interaction matrix $\mathbf{L}_{\mathbf{e}_r}$ can be written as,

$$\begin{aligned} \mathbf{L}_{\mathbf{e}_r} &= \mathbf{C} \begin{bmatrix} \dot{\mathbf{u}}_i^T & \mathbf{H}_i(E)^T \\ \vdots & \vdots \end{bmatrix}_{4 \times 6} \\ &= \begin{bmatrix} {}^r\boldsymbol{\beta}_i \mathbf{n}_T^T & ({}^r\mathbf{m}_{\beta_i} \times \mathbf{n}_T)^T \\ \vdots & \vdots \end{bmatrix}_{3 \times 6} \end{aligned} \quad (10)$$

Here, the parameters ${}^r\boldsymbol{\beta}_i$ and ${}^r\mathbf{m}_{\beta_i}$ are defined differently compared to the minimal case:

$${}^r\boldsymbol{\beta}_i = \sum_{j=1}^4 \frac{-C(i,j)}{\mathbf{n}_T \cdot \mathbf{n}_{S_j}}, \quad {}^r\mathbf{m}_{\beta_i} = \sum_{j=1}^4 \frac{-C(i,j)}{\mathbf{n}_T \cdot \mathbf{n}_{S_j}} \mathbf{m}_{S_j} \quad (11)$$

Finally, the standard control law used in visual servoing [1] is used here also for achieving the task. It tries to ensure at best a decoupled exponential decrease of the task function: $\dot{\mathbf{e}}_k = -\lambda \mathbf{e}_k$ with λ a positive gain, and $k = m$ for the minimal case or $k = r$ for the redundant case, leading to

$$\mathbf{v}_{E,k} = -\lambda \widehat{\mathbf{L}}_{\mathbf{e}_k}^+ \mathbf{e}_k, \quad k = m, r \quad (12)$$

with

$$\widehat{\mathbf{L}}_{\mathbf{e}_k} = \begin{bmatrix} \dot{\mathbf{u}}_i^T & \mathbf{H}_i(E)^T \\ \vdots & \vdots \end{bmatrix}_{3 \times 6} = \begin{bmatrix} {}^k\boldsymbol{\beta}_i \mathbf{n}_T^T & ({}^k\mathbf{m}_{\beta_i} \times \mathbf{n}_T)^T \\ \vdots & \vdots \end{bmatrix}_{3 \times 6} \quad (13)$$

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 $\widehat{\mathbf{L}}_{\mathbf{e}_k}$.

III. STABILITY ANALYSIS

The closed loop dynamic equation for the system is obtained by substituting (12) in the kinematic equation (1) of the task

$$\dot{\mathbf{e}}_k = -\lambda \mathbf{L}_{\mathbf{e}_k} \widehat{\mathbf{L}}_{\mathbf{e}_k}^+ \mathbf{e}_k = -\lambda \mathbf{M}_p(\mathbf{e}_k) \mathbf{e}_k \quad (14)$$

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

$$\mathbf{M}_p(\mathbf{e}_k) > 0 \quad (15)$$

In ideal situation, when perfect model of interaction matrix is known and used in the control scheme ($\widehat{\mathbf{L}}_{\mathbf{e}_k}^+ = \mathbf{L}_{\mathbf{e}_k}^+$), the system is GAS as $\mathbf{M}_p(\mathbf{e}_k) = \mathbf{I}_3$. In a real scenario, model approximations and errors on extrinsic parameters (r_i, d_i, α_i) and surface normal \mathbf{n}_T cause the closed loop matrix $\mathbf{M}_p(\mathbf{e}_k)$ to deviate from its ideal state. In these situations, the above sufficiency condition corresponds to the positive definiteness of the symmetric part $\mathbf{M}_{sym}^p = \frac{1}{2} (\mathbf{M}_p(\mathbf{e}_k) + \mathbf{M}_p^T(\mathbf{e}_k))$. If λ_m denotes the minimal eigenvalue of matrix \mathbf{M}_{sym}^p , one test that is necessary and sufficient for the positive definiteness of a real symmetric matrix is [17]:

$$\lambda_m > 0 \quad (16)$$

We try in the following to obtain simple conditions for the above test that enhance the understanding of the behavior of the system wrt. extrinsic parameters and surface normal errors.

A. Gershgorin's theorems

Gershgorin's theorems help us to provide circular disc regions in which characteristic roots lie. Two important theorems are stated below [18]:

Theorem 1: Every eigenvalue of a matrix \mathbf{Q} with components q_{ij} lies in at least one of the circular discs with centers q_{ii} and radii $\sum_{j \neq i} |q_{ij}|$.

Theorem 2: If c of the circular discs of Theorem 1 form a connected domain that is isolated from the other discs, then there are precisely c eigenvalues of \mathbf{Q} within this connected domain.

Theorem 1 defines these circular discs by providing corresponding center and radius. Theorem 2 ensures that even if the circular discs overlap, the corresponding eigenvalues must be within this union. Since the eigenvalues of a symmetric matrix are real, circular discs reduce to intervals in the real axis. The above theorems enable us to directly obtain simple sufficiency condition for positive definiteness of \mathbf{M}_{sym}^p . If the matrix has positive diagonals that are dominant, *i.e.*, greater than the radii evaluated from off-diagonal terms, then the test in (16) is satisfied.

Let us consider m_{ij} as the components of \mathbf{M}_{sym}^p . The three corresponding intervals are defined as $[O_i - \rho_i, O_i + \rho_i]$ where

$$O_i = m_{ii} \quad , \quad \rho_i = \sum_{j \neq i} |m_{ij}| \quad (17)$$

To satisfy the test (16), we need to ensure,

$$\min \{O_i - \rho_i : i = 1, 2, 3\} > 0 \quad (18)$$

B. Analytical Form of the pseudo-inverse

The pseudo-inverse of \mathbf{L}_{e_k} given in (13) has been obtained from a symbolic software, utilizing the knowledge of the screw-system involved in the mapping from proximity space to Cartesian space for the task considered. It has the following form:

$$\mathbf{L}_{e_k}^+ = \mathbf{P} \mathbf{L}_{e_k}^- \quad (19)$$

where

$$\begin{aligned} \mathbf{P} &= \begin{bmatrix} \mathbf{n}_T \mathbf{n}_T^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 - \mathbf{n}_T \mathbf{n}_T^T \end{bmatrix} \\ \mathbf{L}_{e_k}^- &= \frac{1}{l} \begin{bmatrix} \cdot & {}^k \mathbf{m}_{\beta_{j+}} \times {}^k \mathbf{m}_{\beta_{j-}} & \cdot \\ \cdot & {}^k \beta_{j+} {}^k \mathbf{m}_{\beta_{j-}} - {}^k \beta_{j-} {}^k \mathbf{m}_{\beta_{j+}} & \cdot \end{bmatrix} \\ l &= \sum_{i=1}^3 \mathbf{n}_T \cdot \left({}^k \beta_{i-} {}^k \mathbf{m}_{\beta_i} \times {}^k \mathbf{m}_{\beta_{i+}} \right) \end{aligned} \quad (20)$$

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

Matrix \mathbf{P} is an orthogonal projection matrix as it is idempotent ($\mathbf{P}^2 = \mathbf{P}$) and symmetric [17]. The first submatrix of \mathbf{P} is a projection along surface normal \mathbf{n}_T . The second submatrix is a vector rejection operation from surface normal \mathbf{n}_T .

If we multiply the interaction matrix \mathbf{L}_{e_k} with \mathbf{P} , we get back \mathbf{L}_{e_k} . Indeed, from (13), we have

$${}^k \beta_i \mathbf{n}_T^T (\mathbf{n}_T \mathbf{n}_T^T) = {}^k \beta_i (\mathbf{n}_T^T \mathbf{n}_T) \mathbf{n}_T^T = {}^k \beta_i \mathbf{n}_T^T \quad (22)$$

$$\left({}^k \mathbf{m}_{\beta_i} \times \mathbf{n}_T \right)^T (\mathbf{n}_T \mathbf{n}_T^T) = \left(\left({}^k \mathbf{m}_{\beta_i} \times \mathbf{n}_T \right)^T \mathbf{n}_T \right) \mathbf{n}_T^T = \mathbf{0}^T \quad (23)$$

from which we obtain $\mathbf{L}_{e_k} \mathbf{P} = \mathbf{L}_{e_k}$.

From the property $\mathbf{L}_{e_k} \mathbf{P} = \mathbf{L}_{e_k}$, it is clear that matrix $\mathbf{L}_{e_k}^-$ is a reflexive generalized (left-)inverse [19] of \mathbf{L}_{e_k} since it satisfies

$$\mathbf{L}_{e_k} \mathbf{L}_{e_k}^- \mathbf{L}_{e_k} = \mathbf{L}_{e_k} \quad (24)$$

$$\mathbf{L}_{e_k}^- \mathbf{L}_{e_k} \mathbf{L}_{e_k}^- = \mathbf{L}_{e_k}^- \quad (25)$$

$$(\mathbf{L}_{e_k} \mathbf{L}_{e_k}^-)^T = \mathbf{L}_{e_k} \mathbf{L}_{e_k}^- \quad (26)$$

The only difference between $\mathbf{L}_{e_k}^-$ and the pseudo-inverse $\mathbf{L}_{e_k}^+$ is that $\mathbf{L}_{e_k}^-$ is such that

$$(\mathbf{L}_{e_k}^- \mathbf{L}_{e_k})^T \neq \mathbf{L}_{e_k}^- \mathbf{L}_{e_k} \quad (27)$$

As $\mathbf{L}_{e_k}^-$ is a generalized inverse of \mathbf{L}_{e_k} , it can also be used to execute the task as an alternative to $\mathbf{v}_{E,k}$ given in (12) using as control scheme $\mathbf{v}_{E,k}^n = -\lambda \mathbf{L}_{e_k}^- \mathbf{e}_k$. The potential advantage is that $\mathbf{L}_{e_k}^+$ highly depends on \mathbf{n}_T through \mathbf{P} while it is not

the case for $\mathbf{L}_{e_k}^-$. This results in the analysis of two closed loop matrices $\mathbf{M}_p(\mathbf{e})$ and $\mathbf{M}_n(\mathbf{e})$ with:

$$\mathbf{M}_p(\mathbf{e}) = \mathbf{L}_{e_k} \widehat{\mathbf{P}} \widehat{\mathbf{L}}_{e_k} \quad (28)$$

$$\mathbf{M}_n(\mathbf{e}) = \mathbf{L}_{e_k} \widehat{\mathbf{L}}_{e_k} \quad (29)$$

In the following we study the stability of the closed loop system in two situations from the analysis of the matrices $\mathbf{M}_p(\mathbf{e})$ and $\mathbf{M}_n(\mathbf{e})$. The first situation corresponds to the generic case when we have imperfect surface normal estimation ($\varepsilon = \|\mathbf{n}_T \times \widehat{\mathbf{n}}_T\| \neq 0$), and the second when it is perfect ($\varepsilon = 0$).

C. Stability Conditions for Classical Control Law

Matrix \mathbf{M}_p can be written as the sum of two square matrices of dimension 3:

$$\mathbf{M}_p = \mathbf{A}_p + \mathbf{B}_p \quad (30)$$

where the terms involved in \mathbf{A}_p are given by¹

$$\begin{aligned} {}^p a_{ij} &= \frac{{}^p \Delta_{ij}}{{}^p l} \\ {}^p \Delta_{ij} &= \begin{cases} {}^p l_i & i = j \\ (-1)^\gamma \left(\widehat{\mathbf{m}}_{\beta_k} \cdot \boldsymbol{\varepsilon}'_i + \widehat{\beta}_k \mathbf{n}_T \cdot \boldsymbol{\varepsilon}_i^\times \right) & i \neq j \neq k \end{cases} \\ {}^p l_i &= (\mathbf{n}_T \cdot \widehat{\mathbf{n}}_T) \widehat{\mathbf{n}}_T \cdot \left(\beta_i \widehat{\mathbf{m}}_{\beta_{i+}} \times \widehat{\mathbf{m}}_{\beta_{i-}} \right) \\ &\quad + \mathbf{n}_T \cdot \left(\widehat{\beta}_{i+} \widehat{\mathbf{m}}_{\beta_{i-}} \times \mathbf{m}_{\beta_i} + \widehat{\beta}_{i-} \mathbf{m}_{\beta_i} \times \widehat{\mathbf{m}}_{\beta_{i+}} \right) \\ \gamma &= \begin{cases} 1 & \text{if } (i, j, k) \text{ in clockwise order} \\ 0 & \text{otherwise} \end{cases} \\ {}^p l &= \sum_{i=1}^3 \widehat{\mathbf{n}}_T \cdot \left(\widehat{\beta}_{i-} \widehat{\mathbf{m}}_{\beta_i} \times \widehat{\mathbf{m}}_{\beta_{i+}} \right) \\ \boldsymbol{\varepsilon}_i^\times &= \mathbf{m}_{\beta_i} \times \widehat{\mathbf{m}}_{\beta_i} \\ \boldsymbol{\varepsilon}'_i &= (\mathbf{n}_T \cdot \widehat{\mathbf{n}}_T) \widehat{\mathbf{n}}_T \times \beta_i \widehat{\mathbf{m}}_{\beta_i} - \mathbf{n}_T \times \widehat{\beta}_i \mathbf{m}_{\beta_i} \end{aligned} \quad (31)$$

Diagonal terms ${}^p a_{ii}$ of \mathbf{A}_p depend mainly on terms ${}^p l_i$, that are closely related to l given in (21). Off-diagonal terms of \mathbf{A}_p are functions of two main error vectors, $\boldsymbol{\varepsilon}_i^\times$ and $\boldsymbol{\varepsilon}'_i$. The term $\boldsymbol{\varepsilon}_i^\times$ mainly depends on the errors on the extrinsic parameters while $\boldsymbol{\varepsilon}'_i$ depends substantially on both the extrinsic parameter errors and the surface normal error. Note that when there is perfect estimation in surface normal vector and extrinsic parameters, these off-diagonal terms vanish and the diagonal terms become 1.

As for terms involved in \mathbf{B}_p , they are given by

$$\begin{aligned} {}^p b_{ij} &= -\frac{(\boldsymbol{\varepsilon}_{\mathbf{n}_T}^\times \cdot \mathbf{m}_{\beta_i}) \boldsymbol{\varepsilon}_i^*}{{}^p l} \\ \boldsymbol{\varepsilon}_j^* &= \widehat{\mathbf{n}}_T \cdot \left(\widehat{\beta}_{j+} \widehat{\mathbf{m}}_{\beta_{j-}} - \widehat{\beta}_{j-} \widehat{\mathbf{m}}_{\beta_{j+}} \right) \\ \boldsymbol{\varepsilon}_{\mathbf{n}_T}^\times &= \mathbf{n}_T \times \widehat{\mathbf{n}}_T \end{aligned} \quad (32)$$

Terms b_{ij} of matrix \mathbf{B}_p are influenced by the error in surface normal estimation through $\boldsymbol{\varepsilon}_{\mathbf{n}_T}^\times$. Additionally, $(\widehat{\beta}_{j+} \widehat{\mathbf{m}}_{\beta_{j-}} -$

¹Note that we have dropped the superscript k from this point to have simpler notations. The developments leading to this decomposition are given in the appendix.

$\widehat{\beta}_j \widehat{\mathbf{m}}_{\beta_{j+}}$ is a vector lying on the target plane in ideal case and hence the terms ε_j^* can also be viewed as an error on the structure of the target as perceived by the sensor system. In case of perfect surface normal estimation ($\varepsilon = 0$), matrix \mathbf{B}_p becomes $\mathbf{0}$.

Applying Gershgorin's theorem to \mathbf{M}_{sym}^p obtained as the sum of symmetric part of \mathbf{A}_p and \mathbf{B}_p , we get the following intervals:

$$\begin{aligned} O_i &= {}^p a_{ii} + {}^p b_{ii} \\ \rho_i &= \sum_{j \neq i} \left| \frac{{}^p a_{ij} + {}^p a_{ji}}{2} + \frac{{}^p b_{ij} + {}^p b_{ji}}{2} \right| \end{aligned} \quad (33)$$

where

$$\begin{aligned} {}^p a_{ij} + {}^p a_{ji} &= \frac{(-1)^\gamma}{p_l} \left(\widehat{\beta}_k \mathbf{n}_T \cdot (\varepsilon_i^\times - \varepsilon_j^\times) + \widehat{\mathbf{m}}_{\beta_k} \cdot (\varepsilon_i' - \varepsilon_j') \right) \\ {}^p b_{ij} + {}^p b_{ji} &= \frac{-\varepsilon_{\mathbf{n}_T}^\times \cdot (\varepsilon_j^* \mathbf{m}_{\beta_i} + \varepsilon_i^* \mathbf{m}_{\beta_j})}{p_l} \end{aligned} \quad (34)$$

As observed in (34), terms $({}^p a_{ij} + {}^p a_{ji})$ are functions of error terms ε_i^\times , ε_j^\times and ε_i' , ε_j' . The other terms $({}^p b_{ij} + {}^p b_{ji})$ are functions of error terms ε_i^* , ε_j^* and $\varepsilon_{\mathbf{n}_T}^\times$. It can be observed from (34) that it is influenced mainly by $\varepsilon_{\mathbf{n}_T}^\times$. When surface normal estimation errors are small, this term can be neglected. In Section IV-B, we provide simulation results for a case which involves practical errors in all model parameters, leading to the involvement of errors including ε_i^\times , ε_i' , ε_i^* and $\varepsilon_{\mathbf{n}_T}^\times$. Section IV-C provides results for scenario when the stability conditions are satisfied even though the error values were significant in extrinsic parameters of the model. Finally in Section IV-D, we consider a situation where the errors values are high in $\varepsilon_{\mathbf{n}_T}^\times$, resulting in violation of stability conditions and eventual failure in executing the task.

D. Stability Conditions for New Control Law

When $\widehat{\mathbf{L}}_{\mathbf{e}_k}^-$ is used in the control scheme, closed loop matrix \mathbf{M}_n can be written in a highly simpler form:

$$\mathbf{M}_n = \mathbf{A}_n \quad (35)$$

where

$$\begin{aligned} {}^n a_{ij} &= \frac{{}^n \Delta_{ij}}{n_l} \\ {}^n \Delta_{ij} &= \begin{cases} {}^n l_i & i = j \\ (-1)^\gamma \mathbf{n}_T \cdot \left(\widehat{\mathbf{m}}_{\beta_k} \times \varepsilon_i^- + \widehat{\beta}_k \varepsilon_i^\times \right) & i \neq j \neq k \end{cases} \\ {}^n l_i &= \mathbf{n}_T \cdot \left(\widehat{\beta}_i \widehat{\mathbf{m}}_{\beta_{i+}} \times \widehat{\mathbf{m}}_{\beta_{i-}} + \widehat{\beta}_{i+} \widehat{\mathbf{m}}_{\beta_{i-}} \times \mathbf{m}_{\beta_i} \right. \\ &\quad \left. + \widehat{\beta}_{i-} \mathbf{m}_{\beta_i} \times \widehat{\mathbf{m}}_{\beta_{i+}} \right) \\ {}^n l &= \sum_{i=1}^3 \widehat{\mathbf{n}}_T \cdot \left(\widehat{\beta}_i \widehat{\mathbf{m}}_{\beta_i} \times \widehat{\mathbf{m}}_{\beta_{i+}} \right) \\ \varepsilon_i^- &= \widehat{\beta}_i \mathbf{m}_{\beta_i} - \beta_i \widehat{\mathbf{m}}_{\beta_i} \end{aligned} \quad (36)$$

from which the corresponding stability conditions become

$$\min \left\{ {}^n a_{ii} - \sum_{j \neq i} \frac{1}{2} |{}^n a_{ij} + {}^n a_{ji}| : i = 1, 2, 3 \right\} > 0 \quad (37)$$

where

$${}^n a_{ij} + {}^n a_{ji} = \frac{(-1)^\gamma}{n_l} \left(\widehat{\beta}_k \mathbf{n}_T \cdot \left((\varepsilon_i^\times - \varepsilon_j^\times) + \widehat{\mathbf{m}}_{\beta_k} \times (\varepsilon_i^- - \varepsilon_j^-) \right) \right) \quad (38)$$

Term ${}^n a_{ij} + {}^n a_{ji}$ are functions of error terms ε_i^\times , ε_j^\times and ε_i^- , ε_j^- that are simpler than ε_i' , ε_j' in (34). Stability conditions for this new controller are simpler than in the case of classical controller due to the fact that errors such as ε_i^* and $\varepsilon_{\mathbf{n}_T}^\times$ are not present here. This results in higher values for the stability criterion $\min\{O_i - \rho_i\}$, as seen in Section IV-B and IV-C.

E. Stability Conditions in special case of $\varepsilon = 0$

In the special case of perfect estimation of surface normal, the closed loop matrices of both control laws become identical:

$$\mathbf{M}_{p|\varepsilon=0} = \mathbf{L}_{\mathbf{e}_k} \widehat{\mathbf{P}} \widehat{\mathbf{L}}_{\mathbf{e}_k}^- = \mathbf{L}_{\mathbf{e}_k} \widehat{\mathbf{L}}_{\mathbf{e}_k}^- = \mathbf{M}_{n|\varepsilon=0} = \mathbf{A}_{n|\varepsilon=0} \quad (39)$$

This leads to significant simplification in stability conditions for classical controller, with term $({}^p b_{ij} + {}^p b_{ji})$ disappearing from (33). Stability conditions are the same as in (37), with terms in (36) such as ${}^n l$, $\widehat{\mathbf{m}}_{\beta_i}$ and $\widehat{\beta}_i$ evaluated using \mathbf{n}_T instead of $\widehat{\mathbf{n}}_T$. We provide simulation results for this special case in Section IV-C where the stability conditions can be noticed to be the same for both controllers.

IV. SIMULATION RESULTS

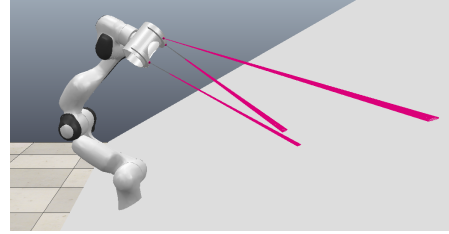


Fig. 3: Simulation setup in CoppeliaSim.

Experimental results obtained in normal conditions, *i.e.*, with correctly calibrated parameters, are described in [15]. In this section we provide relevant simulation results considering calibration errors as well as errors on the surface normal estimation. Simulation was done using FrankaSim [20], a simulator based on ROS and CoppeliaSim for Panda robot from Franka Emika. The simulator is integrated with ViSP [21], which is used as the main software library for programming. Fig. 3 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° wrt. lateral axis of the base frame of the robot. The array consists of two half rings with two sensors on the first and one on the second for the case of minimal sensors (see Figs. 2 and 3). In case of redundant sensors, one additional sensor is used on the second half ring. Extrinsic parameters of the proximity array are shown in Table I for each task considered. The desired configuration is chosen as a parallel configuration with a distance of 20 cm along the y axis of the

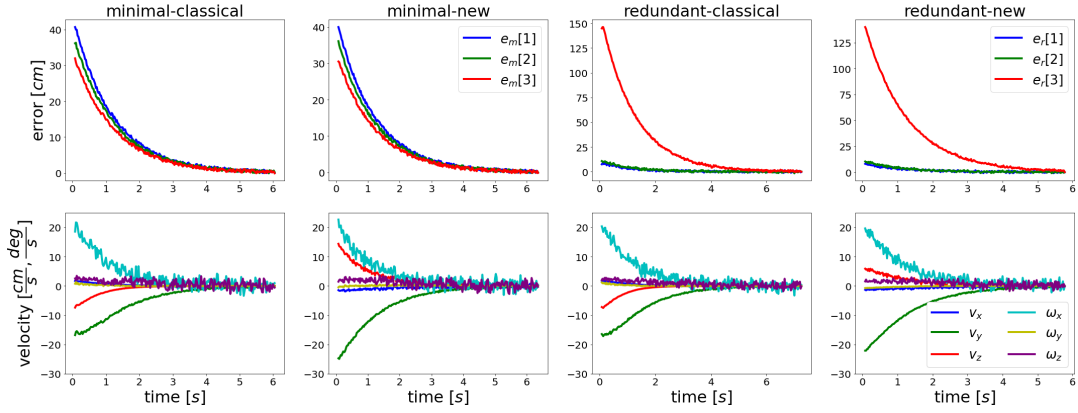


Fig. 4: Simulation results for Case I (Section IV-A) with minimal (left) and redundant (right) number of sensors using both classical and new control law. Task function components (cm) versus time (s) on the top, end-effector velocities (cm/s and deg/s) versus time (s) in the bottom.

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 [8]. A uniform noise in the interval $[-0.005\text{ m}, 0.005\text{ m}]$ is considered for the proximity signals from the array. The gain λ for both control scheme is set to $\lambda = 0.8$. The video accompanying this paper illustrates the behavior of the system during the execution of the tasks.

\mathbf{e}_m		\mathbf{e}_r	
$\alpha_1 = 250^\circ$	$\alpha_3 = 270^\circ$	$\alpha_1 = 250^\circ$	$\alpha_3 = 250^\circ$
$\alpha_2 = 290^\circ$		$\alpha_2 = 290^\circ$	$\alpha_4 = 290^\circ$
$d_1 = 5.5\text{ cm}, r_1 = 7\text{ cm}, d_2 = -5.5\text{ cm}, r_2 = 7\text{ cm}$			

TABLE I: Extrinsic parameters of proximity array.

A. Case I: Perfect Model

In the first case we consider the execution of the task with a perfect model for minimal and redundant number of sensors using both classical and new controllers. Closed loop matrices in all scenarios are equal to identity, which means the stability conditions are of course ensured. As seen in Fig. 4, this results in a perfect exponential reduction of the task function. A slight difference occurs for each scenario in the value of end-effector velocity, and thus on the robot trajectory to reach the desired parallel configuration.

B. Case II: Model with errors on all parameters

In the second case, we consider a practical situation with errors in both extrinsic parameters and surface normal as shown in Table II where surface normal error is expressed as a rotational angle θ about the z-axis of F_E (see Fig. 2). As

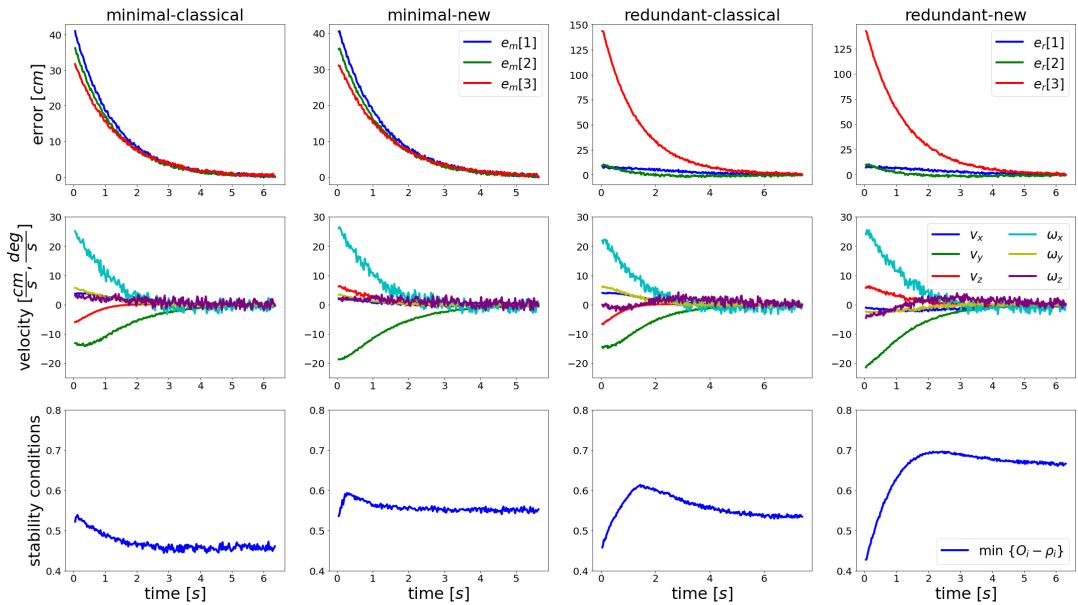


Fig. 5: Simulation results for Case II (Section IV-B). The bottom figures represents stability criterium $\min\{O_i - \rho_i\}$ versus time (s).

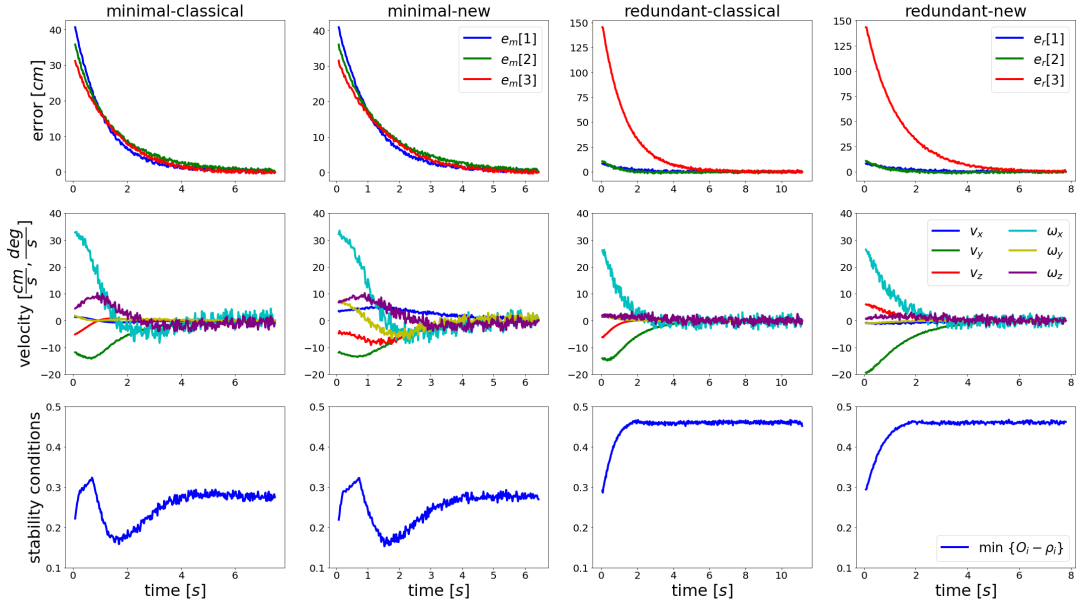


Fig. 6: Simulation results for Case III (Section IV-C)

\mathbf{e}_m		\mathbf{e}_r	
$\alpha_1 = \alpha_1 + 10^\circ$	$\alpha_3 = \alpha_3 - 10^\circ$	$\alpha_1 = \alpha_1 + 10^\circ$	$\alpha_3 = \alpha_3 - 10^\circ$
$\alpha_2 = \alpha_2 + 10^\circ$		$\alpha_2 = \alpha_2 + 10^\circ$	$\alpha_4 = \alpha_4 - 10^\circ$
$\hat{d}_i = 1.2 d_i, \hat{r}_i = 1.2 r_i, (i = 1, 2), \theta = 10^\circ$			

TABLE II: Model parameters for Case II (Section IV-B)

\mathbf{e}_m		\mathbf{e}_r	
$\alpha_1 = \alpha_1 + 18^\circ$	$\alpha_3 = \alpha_3 - 18^\circ$	$\alpha_1 = \alpha_1 + 18^\circ$	$\alpha_3 = \alpha_3 - 18^\circ$
$\alpha_2 = \alpha_2 + 18^\circ$		$\alpha_2 = \alpha_2 + 18^\circ$	$\alpha_4 = \alpha_4 - 18^\circ$
$\hat{d}_i = 1.2 d_i, \hat{r}_i = 1.2 r_i, (i = 1, 2), \theta = 0^\circ$			

TABLE III: Model parameters for Case III (Section IV-C)

seen in Fig. 5, stability conditions are ensured for all control schemes, which shows their robustness. Once again, similar performances are obtained in all cases.

C. Case III: Model with only extrinsic parameter errors

In the third case, we consider larger angular errors (18°) on each of the rings about its $\hat{\mathbf{k}}$ axis, in anticlockwise

direction for the first ring and clockwise for the second but no errors on the surface normal. The parameters of the system can be seen in Table III. As surface normal estimation is perfect, we have identical closed loop matrices (39) for both controllers and thereby resulting in the same task evolution as seen in Fig. 6. Stability conditions also remain identical using both control laws and they are ensured all along the

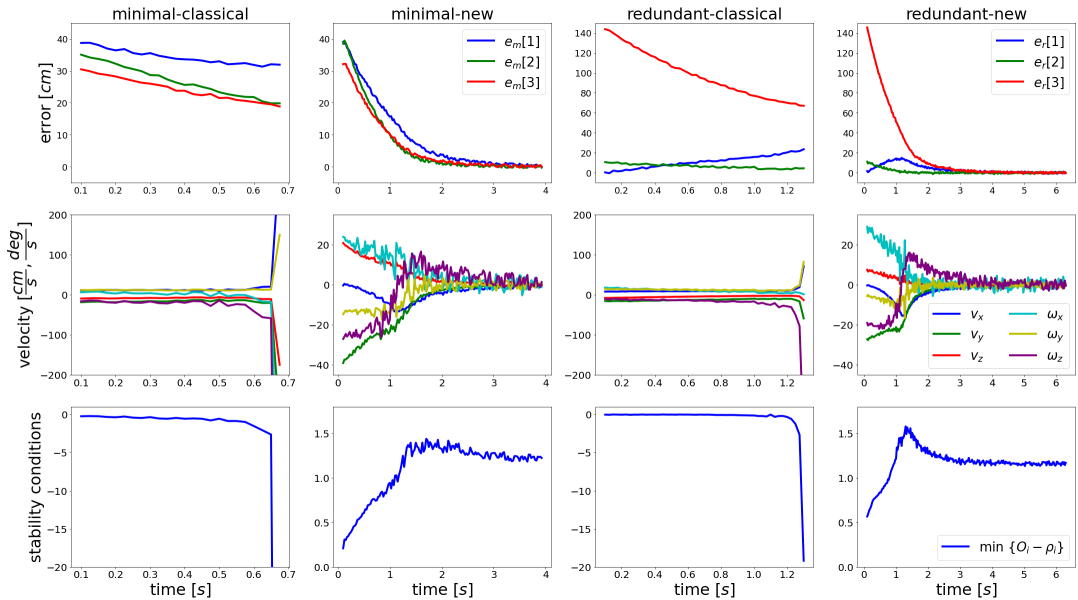


Fig. 7: Simulation results for Case IV (Section IV-D)

execution of the task. However, the end-effector trajectory (shown in video attached) while using the new controller and the minimal number of sensors is less desirable. This is because the control directions with the new controller are highly dependent on model parameter $\{\beta_i, \mathbf{m}_{\beta_i}\}$, which become corrupted with high angular parameter errors, while the pseudo-inverse used in the classical control law projects the control directions given by the new control law along the columns of matrix \mathbf{P} that remains perfect due to the exact value of surface normal being used in \mathbf{P} . Using a redundant number of sensors, an acceptable behaviour is noticed for both controllers. In this case, the symmetry of the proximity array enables to obtain good control directions even for the new controller.

D. Case IV: Model with only surface normal errors

In this last case, we introduce significant surface normal estimation error with $\theta = 27^\circ$, while maintaining perfect model for the extrinsic parameters. As shown in Fig. 7, the task execution with classical control law in both minimal and redundant cases leads to instability. The stability conditions can be observed to be clearly violated. Indeed, the influence of $\hat{\mathbf{P}}$ in the classical control law leads to significant deviation of closed-form matrix from ideal case, which is not the case for the new controller that does not use $\hat{\mathbf{P}}$ at all (remember (19)). The control directions for the classical controller are highly corrupted as linear velocity and angular velocity are respectively projected along an inaccurate $\hat{\mathbf{n}}_T$. Interestingly, the new controller is stable and successful. The control directions in this case remain largely unaffected as they are mainly influenced by the extrinsic parameters of the system, which are perfect for the scenario considered.

V. CONCLUSIONS

This paper was devoted to the stability analysis of a plane-to-plane positioning task using proximity-based control. We obtained closed-form equations for the actual model of the task, which enables us to propose a new controller without compromising the behaviour in sensor space. Additionally, it enables us to have a closer look at the closed-form matrices and characterize the various errors associated with modeling parameters. Stability conditions were formulated using Gershgorin's theorem leading to regional bounds to the eigenvalues involved. It was shown through simulation results that classical controller behaves better when errors in extrinsic parameters of the model are high while the new controller behaves better when errors in surface normal are high. For practical errors in model parameters, both controllers can be used interchangeably to obtain reasonable behaviour. Due to the complexity of the terms involved, further analysis on stability is hindered. However, we believe our results have still a practical interest by determining which parameters are the most sensitive per controller and what level of errors on these parameters is acceptable or not, which is useful in terms of "price to pay" for the calibration and estimation parts of the system. It would be interesting to determine bounds on model parameters for ensuring GAS

or to determine convergence domains for given modelling errors. Additionally, the above results also let us ask few basic questions in the area of SBC: is it possible to get closed-form equations and similar decomposition of the interaction matrix for other sensor-based tasks? Is it possible to find other control directions which are relevant, leading to similar behaviour in sensor space? Our future work would be revolving around such themes.

REFERENCES

- [1] F. Chaumette, S. Hutchinson, and P. Corke, "Visual servoing," in *Springer Handbook of Robotics*, 2016, pp. 841–866.
- [2] J. Novak and I. Feddema, "A capacitance-based proximity sensor for whole arm obstacle avoidance," in *IEEE Int. Conf. on Robotics and Automation (ICRA'92)*, 1992, pp. 1307–1314 vol.2.
- [3] V. Lumelsky and E. Cheung, "Real-time collision avoidance in tele-operated whole-sensitive robot arm manipulators," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 23, no. 1, pp. 194–203, 1993.
- [4] U. Nunes, P. Faia, and A. de Almeida, "Sensor-based 3-d autonomous contour-following control," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'94)*, vol. 1, 1994, pp. 172–179.
- [5] 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.
- [6] Y. Ding and U. Thomas, "Collision avoidance with proximity servoing for redundant serial robot manipulators," in *IEEE Int. Conf. on Robotics and Automation (ICRA'20)*, 2020, pp. 10249–10255.
- [7] 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 (IROS'19)*, 2019, pp. 6742–6747.
- [8] 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 Trans. on Robotics*, vol. 38, no. 3, pp. 1599–1620, 2022.
- [9] H. K. Khalil, *Nonlinear control*. Pearson, New York, 2015, vol. 406.
- [10] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The confluence of vision and control*. Springer, London, 1998, pp. 66–78.
- [11] E. Malis, F. Chaumette, and S. Boudet, "2d 1/2 visual servoing," *IEEE Trans. on Robotics and Automation*, 1999.
- [12] E. Malis and F. Chaumette, "Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 2, pp. 176–186, 2002.
- [13] B. Espiau, "Effect of camera calibration errors on visual servoing in robotics," in *Int. Symp. on Experimental Robotics (ISER)*, 1993.
- [14] J. Pages, C. Collewet, F. Chaumette, and J. Salvi, "Optimizing plane-to-plane positioning tasks by image-based visual servoing and structured light," *IEEE Trans. on Robotics*, vol. 22, no. 5, pp. 1000–1010, 2006.
- [15] 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. 12795–12802.
- [16] C. Samson, B. Espiau, and M. Le Borgne, *Robot control: the task function approach*. Oxford University Press, Inc., 1991.
- [17] G. Strang, *Linear algebra and its applications*, 4th ed. Belmont, CA: Thomson, Brooks/Cole, 2006.
- [18] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*. USA: Oxford University Press, Inc., 1988.
- [19] 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.
- [20] 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.
- [21] 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, 2005.

APPENDIX

The steps for decomposing matrix \mathbf{M}_p (30) are the followings:

$$\begin{aligned}
\mathbf{M}_p &= \mathbf{L}_{\mathbf{e}_k} \widehat{\mathbf{P}} \widehat{\mathbf{L}}_{\mathbf{e}_k} \\
&= \frac{1}{p_l} \begin{bmatrix} \cdot \\ \beta_i \mathbf{n}_T^T & (\mathbf{m}_{\beta_i} \times \mathbf{n}_T)^T \\ \cdot \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{n}}_T \widehat{\mathbf{n}}_T^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 - \widehat{\mathbf{n}}_T \widehat{\mathbf{n}}_T^T \end{bmatrix} \begin{bmatrix} \cdot & \widehat{\mathbf{m}}_{\beta_{j_+}} \times \widehat{\mathbf{m}}_{\beta_{j_-}} & \cdot \\ \cdot & \widehat{\beta}_{j_+} \widehat{\mathbf{m}}_{\beta_{j_-}} - \widehat{\beta}_{j_-} \widehat{\mathbf{m}}_{\beta_{j_+}} & \cdot \end{bmatrix} \\
&= \frac{1}{p_l} \begin{bmatrix} \cdot \\ \beta_i \mathbf{n}_T^T & (\mathbf{m}_{\beta_i} \times \mathbf{n}_T)^T \\ \cdot \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{n}}_T \widehat{\mathbf{n}}_T^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \cdot & \widehat{\mathbf{m}}_{\beta_{j_+}} \times \widehat{\mathbf{m}}_{\beta_{j_-}} & \cdot \\ \cdot & \widehat{\beta}_{j_+} \widehat{\mathbf{m}}_{\beta_{j_-}} - \widehat{\beta}_{j_-} \widehat{\mathbf{m}}_{\beta_{j_+}} & \cdot \end{bmatrix} \\
&\quad + \frac{1}{p_l} \begin{bmatrix} \cdot \\ \beta_i \mathbf{n}_T^T & (\mathbf{m}_{\beta_i} \times \mathbf{n}_T)^T \\ \cdot \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\widehat{\mathbf{n}}_T \widehat{\mathbf{n}}_T^T \end{bmatrix} \begin{bmatrix} \cdot & \widehat{\mathbf{m}}_{\beta_{j_+}} \times \widehat{\mathbf{m}}_{\beta_{j_-}} & \cdot \\ \cdot & \widehat{\beta}_{j_+} \widehat{\mathbf{m}}_{\beta_{j_-}} - \widehat{\beta}_{j_-} \widehat{\mathbf{m}}_{\beta_{j_+}} & \cdot \end{bmatrix} \\
&= \frac{1}{p_l} \begin{bmatrix} \cdot \\ \beta_i \mathbf{n}_T^T & (\mathbf{m}_{\beta_i} \times \mathbf{n}_T)^T \\ \cdot \end{bmatrix} \begin{bmatrix} \cdot & \widehat{\mathbf{n}}_T \left(\widehat{\mathbf{n}}_T \cdot \left(\widehat{\mathbf{m}}_{\beta_{j_+}} \times \widehat{\mathbf{m}}_{\beta_{j_-}} \right) \right) & \cdot \\ \cdot & \widehat{\beta}_{j_+} \widehat{\mathbf{m}}_{\beta_{j_-}} - \widehat{\beta}_{j_-} \widehat{\mathbf{m}}_{\beta_{j_+}} & \cdot \end{bmatrix} \\
&\quad + \frac{1}{p_l} \begin{bmatrix} \cdot \\ \beta_i \mathbf{n}_T^T & (\mathbf{m}_{\beta_i} \times \mathbf{n}_T)^T \\ \cdot \end{bmatrix} \begin{bmatrix} \cdot & 0 & \cdot \\ \cdot & -\widehat{\mathbf{n}}_T \left(\widehat{\mathbf{n}}_T \cdot \left(\widehat{\beta}_{j_+} \widehat{\mathbf{m}}_{\beta_{j_-}} - \widehat{\beta}_{j_-} \widehat{\mathbf{m}}_{\beta_{j_+}} \right) \right) & \cdot \end{bmatrix} \\
&= \frac{1}{p_l} \left[(\mathbf{n}_T \cdot \widehat{\mathbf{n}}_T) \left(\widehat{\mathbf{n}}_T \cdot \left(\widehat{\beta}_{j_+} \widehat{\mathbf{m}}_{\beta_{j_+}} \times \widehat{\mathbf{m}}_{\beta_{j_-}} \right) \right) + \mathbf{n}_T \cdot \left(\widehat{\beta}_{j_+} \widehat{\mathbf{m}}_{\beta_{j_-}} \times \mathbf{m}_{\beta_i} - \widehat{\beta}_{j_-} \widehat{\mathbf{m}}_{\beta_{j_+}} \times \mathbf{m}_{\beta_i} \right) \right]_{ij} \\
&\quad - \frac{1}{p_l} \left[\mathbf{m}_{\beta_i} \cdot (\mathbf{n}_T \times \widehat{\mathbf{n}}_T) \right] \left(\widehat{\mathbf{n}}_T \cdot \left(\widehat{\beta}_{j_+} \widehat{\mathbf{m}}_{\beta_{j_-}} - \widehat{\beta}_{j_-} \widehat{\mathbf{m}}_{\beta_{j_+}} \right) \right)_{ij} \\
&= \frac{{}^p \Delta_{ij}}{p_l} + {}^p b_{ij} \\
&= {}^p a_{ij} + {}^p b_{ij} \\
&= \mathbf{A}_p + \mathbf{B}_p
\end{aligned} \tag{40}$$