



HAL
open science

Assembly Task in Congested Area using Sensor-based Control

John Thomas

► **To cite this version:**

John Thomas. Assembly Task in Congested Area using Sensor-based Control. Robotics [cs.RO]. Université de Rennes, 2024. English. NNT : . tel-04665917

HAL Id: tel-04665917

<https://theses.hal.science/tel-04665917v1>

Submitted on 31 Jul 2024

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

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES

ÉCOLE DOCTORALE N° 601

*Mathématiques, Télécommunications, Informatique, Signal, Systèmes,
Électronique*

Spécialité : Automatique, Productique et Robotique

Par

John THOMAS

Assembly Task in Congested Area using Sensor-based Control

Thèse soutenue à Rennes, le 17 avril 2024

Unité de recherche : Centre Inria de l'Université de Rennes

Rapporteurs avant soutenance :

Philippe FRAISSE Professeur, Université de Montpellier, LIRMM
Sébastien BRIOT Directeur de Recherche CNRS, LS2N, Nantes

Composition du Jury :

Présidente :	Marie BABEL	Professeure, INSA de Rennes, IRISA/Inria
Examineurs :	Philippe FRAISSE	Professeur, Université de Montpellier, LIRMM
	Sébastien BRIOT	Directeur de Recherche CNRS, LS2N, Nantes
	Olivier KERMORGANT	Maître de conférences, Ecole Centrale de Nantes, LS2N
Dir. de thèse :	François CHAUMETTE	Directeur de recherche Inria, Centre Inria de l'Université de Rennes

SYNTHÈSE EN FRANÇAIS

Motivation

Les chaînes d'assemblage de satellites sont devenues une réalité ces dernières années. Contrairement aux industries automobile et aéronautique, le volume de production n'incitait pas à ce type d'activité auparavant. Satellites OneWeb d'Airbus¹, une entreprise commune entre Airbus² et OneWeb³ tente d'apporter un changement pour transformer l'industrie des satellites en production de masse. En janvier 2024, elle aura mis en orbite 634 satellites fabriqués en série qui feront partie d'une constellation de satellites capables de fournir un accès à l'internet à un prix abordable dans le monde entier. La Figure 1 montre un manipulateur robotique envisagé pour effectuer des opérations d'assemblage à l'avenir. Thales Alenia Space⁴ a automatisé l'installation d'inserts dans les panneaux structurels des satellites de communication en 2017 (voir Figure 2). L'utilisation de robots pour l'assemblage des composants d'un satellite est encore minime⁵. Toutefois, cette situation est susceptible de changer au fur et à mesure que le volume des commandes augmente. En 2007, le système de démonstration Orbital Express a présenté des capacités de maintenance autonome des satellites en orbite [Ogilvie et al., 2008]. Un bras manipulateur robotisé a capturé de manière autonome un satellite à l'aide d'un système d'asservissement visuel. Il a également effectué le transfert d'une batterie et d'un ordinateur fonctionnels à l'aide d'une unité remplaçable en orbite. On-Orbit Space Assembly (OSA) est un autre domaine au potentiel croissant complémentaire à On-Orbit Servicing (OOS) [Roa Garzon et al., 2017]. Cela permettrait de déployer des satellites de grande taille, ce qui ne serait pas possible autrement en raison de la limitation de la capacité de chargement des vaisseaux spatiaux opérationnels. De telles opérations nécessitent des robots manipulateurs dédiés. CAESAR est l'un de ces manipulateurs robotiques polyvalents qualifiés pour l'espace, capable d'effectuer des opérations telles que OSA et OOS [Beyer et al., 2018].

1. <https://airbusonewebsatellites.com/>

2. <https://www.airbus.com/en>

3. <https://oneweb.net/>

4. <https://www.thalesaleniaspace.com/en>

5. <https://spacenews.com/fully-automated-satellite-assembly-lines-not-quite-yet/>

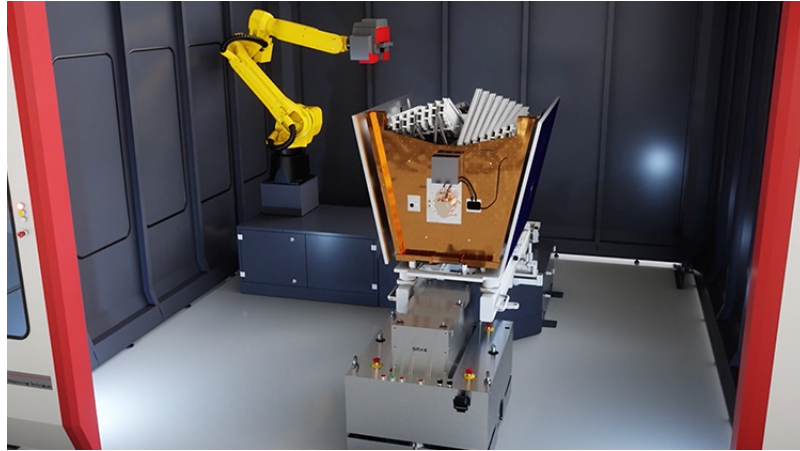


Figure 1 – Robot manipulateur envisagé pour l'assemblage d'un satellite (Photo/vidéo avec l'aimable autorisation d'Airbus OneWeb Satellites. Copyright 2022.)

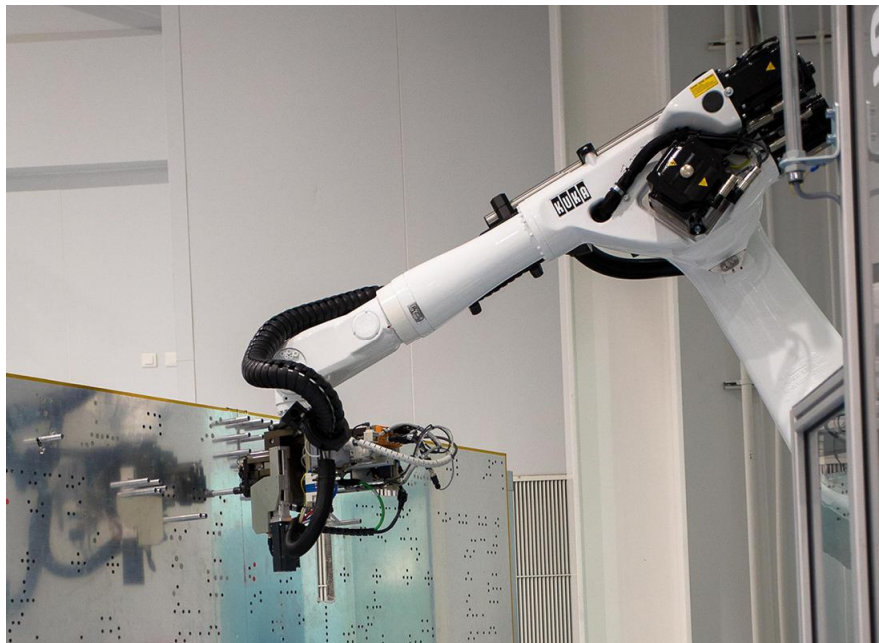


Figure 2 – Le système robotique SAPHIR insère des panneaux structuraux pour les satellites de télécommunications (Photo extraite de : <https://www.thalesgroup.com/en/worldwide/space/news/factory-future-robotic-saphir-system-operation>)

Cet intérêt croissant pour l'assemblage de satellites pose des questions de recherche intéressantes. Lors de l'assemblage des composants, il convient de ne pas endommager les pièces se trouvant à proximité dans l'espace de travail. Il est donc important de disposer de méthodes permettant d'assembler les pièces dans un espace encombré. En outre, ces méthodes doivent être intrinsèquement suffisamment robustes pour gérer des scénarios inattendus qui pourraient survenir en OSA ou OOS. Il est nécessaire de rendre l'exécution des tâches d'assemblage flexible, adaptative et précise. Des ajustements fins peuvent être nécessaires en raison des incertitudes et des déformations qui peuvent apparaître sur la pièce [Espiau, 1990]. Lorsqu'il s'agit de grandes structures légères comme celles des satellites, il peut être nécessaire de compenser les erreurs terminales [Samson et al., 1991]. Cela nous amène à l'utilité des capteurs extéroceptifs sans contact dans les opérations d'assemblage. Une caméra peut fournir suffisamment d'informations pour permettre au manipulateur de se positionner d'un endroit à l'autre dans l'espace de travail. Il est également nécessaire de voir comment la détection de l'environnement local au moyen de capteurs de proximité peut être utilisée efficacement pour assurer la sécurité.

La commande référencée capteurs SBC est un cadre très utile qui permet à l'utilisateur d'incorporer de tels signaux provenant de capteurs extéroceptifs pour définir des tâches basées sur des capteurs. La régulation de ces tâches permet à un robot d'atteindre le comportement souhaité. Les tâches exécutées à l'aide de ce cadre présentent une grande robustesse, une nature réactive, une grande précision et une dépendance minimale à l'égard des connaissances préalables de l'environnement. Cette approche élimine la nécessité d'une reconstruction globale de l'environnement, de la localisation du robot et de la planification de la trajectoire. Le schéma le plus populaire en SBC est l'asservissement visuel [Chaumette & Hutchinson, 2006], qui se réfère à l'utilisation de données acquises par une caméra. La capacité d'une caméra imitant l'œil humain a été utilisée efficacement dans l'asservissement visuel pour effectuer des tâches telles que le positionnement, la navigation, le suivi de cibles, l'assemblage, etc. La proximité est une information complémentaire de la vision. Les capteurs proximétriques permettent la détection d'objets locaux par une forme physique d'interaction sans contact [France et al., 1999]. La commande basée sur la proximité fait référence à l'utilisation d'informations locales fournies par des capteurs de proximité dans une commande en boucle fermée.

L'asservissement visuel et la commande basée sur la proximité ont été considérés séparément par le passé. Pour réaliser l'assemblage des composants d'un satellite en toute sécurité, il est important de combiner des informations provenant de capteurs de vision

et de proximité dans une architecture commune pour réaliser une tâche. Il y a un manque de littérature à ce sujet et c'est ce qui est étudié dans cette thèse. Pour développer des architectures de commande qui utilisent efficacement la nature complémentaire des informations fournies par la vision et la proximité, nous considérons un manipulateur redondant. Une caméra fixée à l'extrémité de l'effecteur permet au robot de se positionner avec précision à l'endroit désiré. Pour permettre la détection locale d'obstacles potentiels, un système multisensoriel avec des capteurs de proximité pourrait être fixé autour du manipulateur pour agir comme une peau. Les capteurs situés autour des bras du manipulateur empêcheraient les collisions avec des obstacles ou des parties de l'espace de travail. Les capteurs fixés sur l'effecteur terminal à proximité de la caméra garantissent que les composants situés à proximité de la position finale désirée sont également protégés. Ces capteurs assurent également la protection du manipulateur et de la caméra contre les événements inattendus qui peuvent survenir lors de l'assemblage dans une zone encombrée. Dans les situations où une collision est probable, les manœuvres d'évitement des obstacles utilisant la détection locale par le biais des capteurs de proximité doivent être prioritaires pour garantir la sécurité.

Cette interaction en boucle fermée avec l'environnement, réalisée lors de l'exécution de SBC, garantit la robustesse aux erreurs et aux approximations de modélisation, aux imprécisions d'étalonnage et au bruit des capteurs. Il est également capable de gérer les incertitudes liées à l'environnement. L'exécution d'une tâche peut être perçue comme une régulation jusqu'à un certain point ou comme la satisfaction d'inégalités. Il devient important de disposer d'une analyse de la stabilité de la pose finale obtenue. Cela donne plus de confiance à l'utilisateur et à l'industrie dans l'application pratique des méthodes sur la chaîne de montage. Ces aspects sont également discutés avec la méthodologie proposée dans la thèse.

Dans la section suivante, nous décrivons la structure générale de la thèse.

Structure de la thèse

Cette thèse est structurée en trois parties. Chaque partie est d'abord composée d'une section d'introduction décrivant ses grandes lignes. Dans la première partie, un chapitre préliminaire est fourni au début afin d'apporter le contexte nécessaire aux chapitres suivants. Dans les parties I et II, un chapitre est consacré à l'état de l'art pour le sujet considéré. Il est suivi par des chapitres pertinents qui développent le thème de chaque

partie.

Dans la première partie, nous examinons la commande basée sur la proximité. Nous présentons un système multisensoriel composé de capteurs de proximité. Nous considérons deux tâches qui peuvent être effectuées en attachant le système de capteurs à l'effecteur. La première tâche consiste à se positionner par rapport à un objet plan. La méthodologie de modélisation envisagée est vérifiée par des expérimentations. Une analyse détaillée de la stabilité est également effectuée. La seconde tâche consiste à se positionner par rapport à l'intérieur ou à l'extérieur d'un cylindre. La vérification de la méthodologie de modélisation est effectuée par le biais de simulations et de résultats expérimentaux.

Dans la deuxième partie, nous combinons les informations des capteurs de vision et de proximité pour effectuer une tâche de positionnement dans un espace encombré. Nous commençons cette partie en considérant la tâche d'évitement d'obstacles. Dans ce cas, le système multisensoriel est considéré comme étant enroulé autour du bras manipulateur en plus d'être attaché à l'effecteur final. Une architecture de commande est ensuite proposée pour prendre en compte la tâche d'évitement d'obstacles en même temps que la tâche de positionnement, celle-ci étant réalisée à l'aide du capteur de vision. Nous fournissons des résultats expérimentaux et de simulation pour valider la théorie développée.

Dans la troisième partie, nous discutons de l'utilisation de la théorie des torseurs pour identifier certaines directions de commande présentant des propriétés intéressantes dans le cadre la commande référencée capteurs. Ces directions de commande sont fournies pour les tâches considérées dans la première partie tout en utilisant un nombre minimal de capteurs. Elles sont également obtenues pour une tâche de positionnement à l'aide d'une caméra observant 3 points. Ces directions de commande conduisent à des équations explicites.

Enfin, nous fournissons une conclusion générale pour la thèse, qui est suivie par des perspectives de recherche futures pour chaque partie de la thèse.

Liste des publications

Cette thèse a donné lieu aux publications suivantes,

1. Thomas, J., Pasteau, F., & Chaumette, F., [2022], Plane-to-plane positioning by proximity-based control, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 12795–12802
2. Thomas, J., & Chaumette, F., [2023a], Stability analysis of plane-to-plane posi-

tioning by proximity-based control, *IEEE Robotics and Automation Letters*, 8[8], 4473–4480

3. Thomas, J., & Chaumette, F., [2023b], Use of Screw Theory in Proximity-based Control, *ICRA workshop - "Geometric Representations: The Roles of Modern Screw Theory, Lie algebra, and Geometric Algebra in Robotics"*, 1–5
4. Thomas, J., & Chaumette, F., Positioning in Congested Space by Combining Vision-based and Proximity-based Control, soumis à *IEEE Robotics and Automation Letters*.
5. Thomas, J., & Chaumette, F., Positioning with respect to a cylinder using proximity-based control, soumis à *IEEE Robotics and Automation Letters*.

ACKNOWLEDGEMENT

First and foremost, I would like to acknowledge my supervisor François Chaumette for providing me with the opportunity to do a PhD. I am grateful for the insights I have gained by interacting with him in the past few years. This enabled me to gain a deeper understanding of the subject.

Second, I would like to thank François Pasteau, who fabricated the proximity array used to obtain good experimental results. I would also like to acknowledge the help of Fabien Spindler in the validation work.

Third, I would like to also thank Marie Babel and Olivier Kermorgant for being part of my "Comité de suivi individuel doctoral" and for providing me with recommendations for obtaining a good PhD.

Fourth, I would like to thank my team, Rainbow. I would like to acknowledge all the permanent members and team mates with whom I have interacted in the past few years. Special mention to my roommates, Nicola de Carli, Maxime Bernard and Thibault Noël. Another special mention to Alexander Oliva, who helped me during my first PhD year.

Fifth, I would like to thank the jury members Philippe Fraise, Sébastien Briot, Olivier Kermorgant, Marie Babel and François. Thanks to Philippe and Sébastien for providing a detailed evaluation of the thesis. The constructive comments from these reports helped me obtain a good perspective on the work. I also value fruitful discussions during defense.

Finally, I would like to thank my family. I am deeply in debt to the enormous support I have received from my father, Thomas Antony, and my mother, Asha Lonappan, over the years, especially during my PhD. I would also like to thank my brother, Antony Thomas, for providing an example by pursuing a path similar to mine.

TABLE OF CONTENTS

Introduction	14
I Proximity-based Control	19
1 Preliminaries	22
1.1 Sensor Modeling	22
1.2 Interaction Screw and Matrix	23
1.3 Virtual Linkage	24
1.4 Sensor-based Task	24
2 State of the Art	25
3 Proximity Array	31
3.1 Modeling	31
3.1.1 Approximation on unknown parameter	33
3.1.2 Choice of Sensor Frame	34
3.1.3 Proximity Ring	35
3.1.4 Proximity Array	35
4 Plane-to-Plane Positioning Task	39
4.1 Introduction	39
4.2 Plane-to-Plane Positioning	39
4.2.1 Minimal sensors	42
4.2.2 Redundant Sensors	44
4.2.3 Stability Analysis	46
4.3 Proximity Array as an End-Effector	47
4.4 Experimental Results	48
4.4.1 Case 1	51
4.4.2 Case 2	52

TABLE OF CONTENTS

4.4.3	Case 3	53
4.5	Conclusion	54
5	Stability Analysis of Plane-to-Plane Positioning Task	57
5.1	Plane-to-Plane Positioning Task	58
5.2	Stability Analysis	60
5.2.1	Gershgorin's theorems	60
5.2.2	Analytical Form of the pseudo-inverse	61
5.2.3	Stability Conditions for Classical Control Law	63
5.2.4	Stability Conditions for New Control Law	65
5.2.5	Stability Conditions in special case of perfect surface normal estimation	66
5.3	Simulation Results	66
5.3.1	Case I: Perfect Model	69
5.3.2	Case II: Model with errors on all parameters	69
5.3.3	Case III: Model with only extrinsic parameter errors	69
5.3.4	Case IV: Model with only surface normal errors	70
5.4	Conclusions	70
6	Positioning Task with respect to a cylinder	73
6.1	Introduction	73
6.2	Modelling	73
6.2.1	Minimal sensors	74
6.2.1.1	Simplified Model	75
6.2.1.2	Desired Pose A^*	78
6.2.1.3	Desired Pose B^*	79
6.2.1.4	Trivial Singularity	80
6.2.2	Redundant Sensors	81
6.2.2.1	Special Case	83
6.2.3	Stability Analysis	84
6.3	Simulation Results	85
6.3.1	Case 1 : Positioning outside the cylinder	85
6.3.2	Case 2 : Positioning inside the cylinder to align with its axis.	87
6.4	Experimental Results	92
6.5	Conclusion	92

II	Combining Proximity and Vision in Sensor-based Control	95
7	State of the Art	97
7.1	Past works on Obstacle Avoidance	97
7.2	Past works on using proximity sensors for Obstacle Avoidance	100
7.3	Positioning current work	102
8	Obstacle Avoidance Task	103
8.1	Introduction	103
8.2	Task interpretation	103
8.3	Model for interaction matrix	106
8.3.1	Actual Model : Estimation of Surface Normal	106
8.3.2	Approximate Model: Approximation in intensity of interaction screw	109
8.3.3	Approximate Model: Approximation in direction of interaction screw	110
8.4	Anti-collision using Proximity-based Control	110
8.4.1	Safety-Critical Control	110
8.4.2	Safety-Critical Control in Sensor-based Control	112
8.5	Conclusion	113
9	Positioning in Congested Space	115
9.1	Introduction	115
9.2	Visual Servo Control	115
9.2.1	4-point Visual Servoing	117
9.3	Control Architecture	119
9.3.1	Preliminaries	121
9.3.2	Optimality	122
9.3.2.1	Stopping Condition	123
9.3.2.2	Case 1 : Empty Active Set	124
9.3.2.3	Case 2 : Two parallel planes	124
9.3.2.4	Case 3 : Two blocking planes	125
9.3.2.5	Case 4 : Global convergence with active proximity sensor .	127
9.4	Stability Analysis	127
9.4.1	Specific Scenario	127
9.4.2	General Scenario	129
9.5	Experimentation	130

TABLE OF CONTENTS

9.5.1	Case 1 : Two parallel planes	132
9.5.2	Case 2 : Two blocking planes	132
9.5.3	Case 3 : Obstacles affecting ring 1 in the initial phase of the task . . .	135
9.5.4	Case 4 : Obstacle affecting ring 1 in the final phase of the task . . .	137
9.5.5	Case 5 : Obstacles affecting both rings 1 and 2 in the final phase of the task	138
9.5.6	Case 6: Behavior with Dynamic Obstacle	141
9.6	Simulation	141
9.6.1	Case 1 : Positioning Inside a Box	145
9.6.2	Case 2 : Obstacle affecting ring 1 in the final phase of the task . . .	145
9.6.3	Case 3 : Three Blocking Planes	149
9.7	Conclusion & Future Work	149
 III Use of Screw Theory in Sensor-based Control		155
 10 Application In Sensor-based Tasks		156
10.1	Introduction	156
10.2	Dual Basis	157
10.3	Plane-to-Plane Positioning	159
10.4	Positioning wrt. a cylinder	161
10.5	3-point Visual Servoing	163
10.6	Conclusion	165
 IV Conclusion and Future Work		167
 Conclusion		169
 Future Work		171
 A Evaluation of Kernel Space of Combination matrix for Redundant num- ber of sensors in Plane-to-plane Positioning task		173
 B Decomposition of closed loop matrix for pseudo-inverse based controller for plane-to-plane positioning task		176

C Verification of Dual Basis for plane-to-plane following task	178
D Verification of Dual Basis for Positioning task wrt. a Cylinder	179
E Verification of Dual Basis for 3 point Visual Servoing	181
Bibliography	187

INTRODUCTION

Motivation

Satellite Assembly lines are becoming a reality in recent years. Unlike automobile and aircraft industries the amount of volume produced disincentivized this previously. Airbus OneWeb Satellites⁶, a joint venture between Airbus⁷ and OneWeb⁸ tries to bring a change to transform satellite industry to mass manufacturing. As of January 2024, it has launched 634 mass produced satellites in orbit that would be part of a constellation of satellites that can provide affordable internet access globally. Figure 3 shows a robotic manipulator envisioned to perform assembly operations in future. Thales Alenia Space⁹ had automated installation of inserts in structural panel of communication satellites in 2017 (see Figure 4). The use of robot in assembling the components of satellite is still minimal¹⁰. However this is likely to change as the volume of order goes up. In 2007, Orbital Express Demonstration System exhibited capabilities to autonomously service satellites on-orbit [Ogilvie et al., 2008]. A robotic arm manipulator performed autonomous capture of a satellite using a visual servo system. Additionally it also performed On-Orbit Replaceable Unit transfer of a functional battery and computer. On-Orbit Space Assembly (OSA) is another field with growing potential complementary to On-Orbit Servicing (OOS) [Roa Garzon et al., 2017]. This would enable in deployment of large satellites that would otherwise be not possible due to limitation of cargo capacity of operational spacecrafts. Such operations require dedicated robot manipulators. CAESAR is one such space qualified multi-purpose robotic manipulator system that can perform operations including OSA and OOS [Beyer et al., 2018].

This growing interest for satellite assembly poses interesting research questions. While assembling the components, it must be ensured that the parts in vicinity in the workspace must not be damaged. It is also important to have methods that would enable to assemble

6. <https://airbusoneweb satellites.com/>

7. <https://www.airbus.com/en>

8. <https://oneweb.net/>

9. <https://www.thalesaleniaspace.com/en>

10. <https://spacenews.com/fully-automated-satellite-assembly-lines-not-quite-yet/>

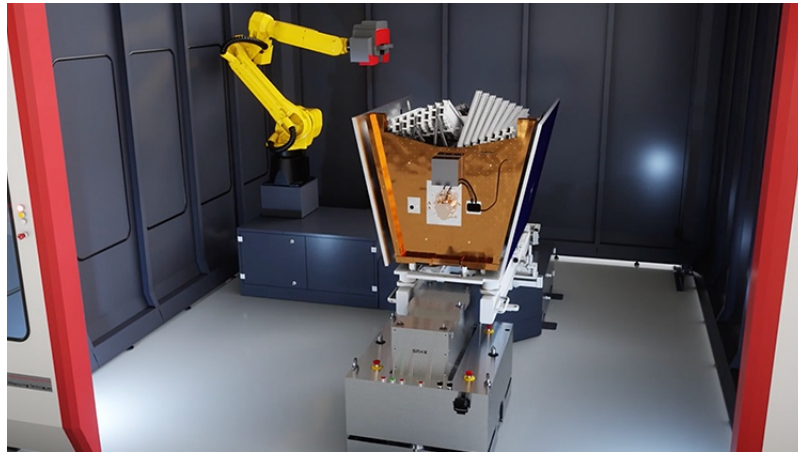


Figure 3 – Robot manipulator envisioned in satellite assembly (Photo/video courtesy of Airbus OneWeb Satellites. Copyright 2022.)

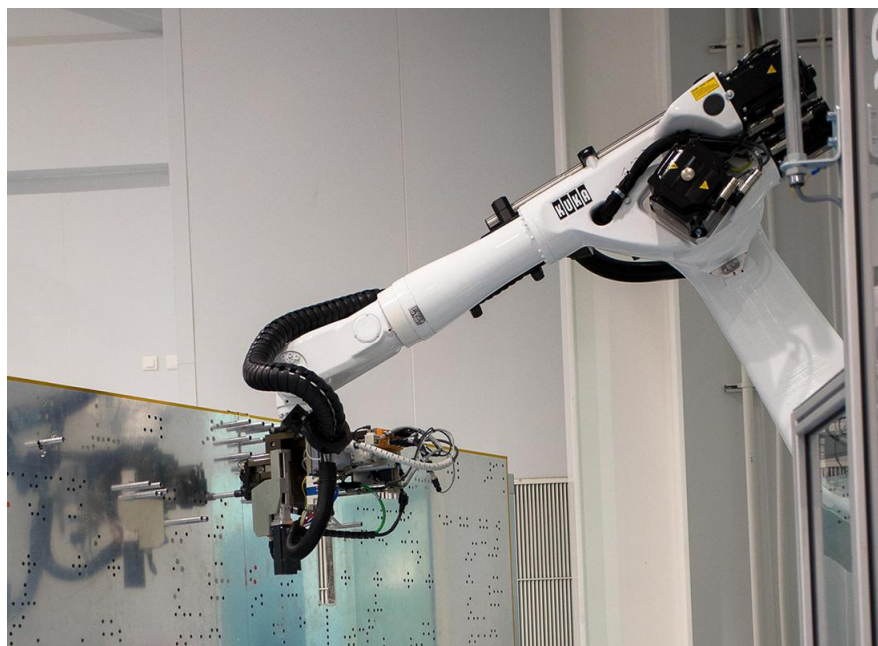


Figure 4 – SAPHIR robotic system inserting structural panels for telecommunications satellites (Photo taken from : <https://www.thalesgroup.com/en/worldwide/space/news/factory-future-robotic-saphir-system-operation>)

the parts in a congested space. Additionally, these methods must be inherently robust enough to handle unexpected scenarios that could arise in OSA or OOS. It is required to make assembly task execution flexible, adaptive and accurate. While dealing with large light structures such as in satellites there can be a need to compensate for terminal errors [Samson et al., 1991]. This leads us to the utility of exteroceptive non-contact sensors in assembly operation. Camera can provide enough information to have a global outlook to enable the manipulator to position the workspace from one location to the other. It is also necessary to see how local environment sensing through proximity sensors can be effectively used in achieving safety.

SBC is a very useful framework that enables the user to incorporate such exteroceptive sensor signals 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. This approach eliminates the need for global reconstruction of the environment, robot localization and trajectory planning. The most popular scheme in SBC is Visual Servoing [Chaumette & Hutchinson, 2006], 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. A complementary sensor to vision is proximity. These sensors enable the detection of local objects through a physical form of interaction without contact [France et al., 1999]. Proximity-based Control refers to the use of local information by proximity sensors in closed-loop control.

Visual Servoing and Proximity-based Control have been considered separately in the past. To perform assembly of satellite components in complete safety it is important to combine both vision and proximity sensor information in a common architecture to perform the task. There is a lack of literature in doing so and this is explored in this thesis. For developing control architectures that effectively utilize the complementary nature of information provided by vision and proximity we consider a redundant manipulator. A camera attached to the end-effector enables the robot to achieve precise positioning to a desired location. To enable local sensing a multi-sensory system with proximity sensors could be attached around the manipulator to act like a skin. Sensors around the arms of the manipulator would prevent collision with obstacles or parts of workspace. Sensors attached to the end-effector close to the camera ensures that the components in the neighbourhood of the final positioning task is also protected. Such sensors also ensure protection of

the manipulator and camera to unexpected events that may occur while assembling in congested area. In situations where a collision is probable, obstacle avoidance maneuvers using local sensing through proximity sensors must take precedence to ensure safety.

Such closed-loop interaction with the environment achieved in executing SBC ensures robustness to modeling errors/ approximations, calibration inaccuracies and sensor noise. It is also able to handle uncertainties that arises from the environment. Task executed can be perceived as regulation to a point or as the satisfaction of inequalities. It becomes important to have an analysis of the stability of the final pose achieved. This gives more confidence to the user and to the industry in practical application of the methods in assembly line. Such discussions are also considered along with the suggested methodology in the thesis.

In the next section we describe the general structure of the thesis.

Thesis Structure

This thesis is structured into three parts. Each part is first composed by an introduction section to describe its general outline. In Part I, a preliminary chapter is provided in the beginning to provide necessary background for the chapters that follow. Both in Part I and II we provide a chapter to discuss the state of the art for the topic considered. It is then followed by relevant chapters expanding on the theme of each part.

In the first part, we consider Proximity-based Control. Here we introduce a multi-sensory system consisting of proximity sensors. We consider two tasks that can be performed while attaching the sensor system to the end-effector. The first task involves positioning wrt. a planar target. Modeling methodology considered is verified through experimentations. A detailed stability analysis is also considered. The second task considered involves positioning wrt. a cylinder either inside or outside. Verification of modeling methodology is considered through simulation and experimental results.

In Part II we combine vision and proximity sensor information to perform the task of positioning in congested space. We start the part by considering the task of obstacle avoidance. Here the multi-sensory system is considered to be wrapped around the arm of a manipulator in addition to being attached to the end-effector. A control architecture is then provided to consider obstacle avoidance task along with positioning task performed using vision sensor. We provide experimental and simulation results for validating the theory developed.

In the third part, we discuss about the use of Screw Theory in identifying certain control directions with interesting properties within the framework of Sensor-based Control. These control directions are provided for the tasks considered in Part I while using minimal number of sensors. It is also obtained for task of positioning using a camera through the observation of 3 points. These control directions leads to closed-form equations in Sensor-based Control.

We finally provide general conclusion for the thesis, which is followed by future research prospects for the each part of the thesis.

List of publications

This thesis has lead to the following publications,

1. Thomas, J., Pasteau, F., & Chaumette, F., [2022], Plane-to-plane positioning by proximity-based control, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 12795–12802
2. Thomas, J., & Chaumette, F., [2023a], Stability analysis of plane-to-plane positioning by proximity-based control, *IEEE Robotics and Automation Letters*, 8[8], 4473–4480
3. Thomas, J., & Chaumette, F., [2023b], Use of Screw Theory in Proximity-based Control, *ICRA workshop - "Geometric Representations: The Roles of Modern Screw Theory, Lie algebra, and Geometric Algebra in Robotics"*, 1–5
4. Thomas, J., & Chaumette, F., Positioning in Congested Space by Combining Vision-based and Proximity-based Control, submitted to *IEEE Robotics and Automation Letters*
5. Thomas, J., & Chaumette, F., Positioning with respect to a cylinder using proximity-based control, submitted for *IEEE Robotics and Automation Letters*

PART I

Proximity-based Control

INTRODUCTION

In the first part of the thesis, we consider the use of proximity sensor information in closed-loop control to perform positioning tasks. This part is divided into six chapters. Chapter 1 introduces preliminary concepts and definitions necessary for understanding the remaining chapters. Discussion on state of the art for Part I is provided in Chapter 2. In Chapter 3, a multi-sensory system is proposed consisting of proximity sensors arranged as an array in a particular configuration. Modeling methodology involved in utilising such a system in SBC framework is considered here. In Chapter 4, we attach this proximity array as an end-effector to perform the task of plane-to-plane positioning. Stability analysis of the task is considered in detail in Chapter 5. In Chapter 6, the sensor system is used in performing the task of positioning wrt. a cylinder.

PRELIMINARIES

In this chapter we provide preliminary information that aids in understanding Part I of the thesis. Definitions are obtained from [Samson et al., 1991].

1.1 Sensor Modeling

In Sensor-based Control (SBC) framework, sensor signals that have high data rate are considered for enabling them to interact in closed-loop with the environment. The sensor signals should also depend on robot joint coordinates (\mathbf{q}) and time (t). Let us consider a sensor S with frame \mathcal{F}_S rigidly attached to link i of the robot that observes a target object T with frame \mathcal{F}_T . Consider a one dimensional sensor signal s from this sensor that is a function of relative position and orientation of \mathcal{F}_S wrt. \mathcal{F}_T . Let \mathbf{p} indicate the relative pose between \mathcal{F}_S and \mathcal{F}_T . Since \mathbf{p} is an element of SE_3 , this sensor signal is indirectly a function of \mathbf{q} and t . This can be mathematically represented as:

$$s = s(\mathbf{p}(\mathbf{q}, t)) \quad (1.1)$$

s must also be a C^2 function that maps between SE_3 and \mathbb{R} . The derivative of such a function can be represented as a screw product,

$$\dot{s} = \frac{\partial s}{\partial \mathbf{p}} \circ \frac{d\mathbf{p}}{dt} = \mathcal{H} \circ \mathcal{K}_{ST} \quad (1.2)$$

where $\mathcal{H} = (\mathbf{H}(\cdot), \mathbf{u})$ is called the interaction screw and $\mathcal{K}_{ST} = (\mathbf{K}_{ST}(\cdot), \mathbf{w}_{ST})$ is the velocity screw of \mathcal{F}_S with respect to \mathcal{F}_T (this notation is used in Part III of the thesis). This velocity screw can also be represented using the more common notation as $\mathcal{K}_{ST} = (\mathbf{v}_{S/\mathcal{F}_T}, \omega_{\mathcal{F}_S/\mathcal{F}_T})$ where $\mathbf{v}_{S/\mathcal{F}_T}$ represents the translational velocity of sensor point S in \mathcal{F}_T and $\omega_{\mathcal{F}_S/\mathcal{F}_T}$ represents the angular velocity of \mathcal{F}_S wrt. \mathcal{F}_T . Further discussion on interaction screw is to follow in the next section. Symbol \circ indicates screw product and

$\mathcal{H} \circ \mathcal{K}_{ST}$ can be evaluated as,

$$\mathcal{H} \circ \mathcal{K}_{ST} = (\mathbf{H}(\cdot), \mathbf{u}) \circ (\mathbf{K}_{ST}(\cdot), \mathbf{w}_{ST}) = \mathbf{H}(\cdot) \cdot \mathbf{w}_{ST} + \mathbf{K}_{ST}(\cdot) \cdot \mathbf{u} \quad (1.3)$$

1.2 Interaction Screw and Matrix

Interaction screw encodes the fundamental aspect of interaction involved in using sensors in SBC framework,

$$\mathcal{H} = (\mathbf{H}(S), \mathbf{u}) \quad (1.4)$$

where \mathbf{u} is the vector of the screw and its vector field is indicated by $\mathbf{H}(\cdot)$. Understanding \mathcal{H} is an essential part of design and analysis of sensor-based tasks (explained below). Also, \mathcal{H} remains independent of the robot involved and depends purely on the characteristics of sensor and target.

A screw is well defined by its value at a point $\mathbf{H}(S)$ and its vector \mathbf{u} . A screw evaluated at certain point such as S can also be evaluated at a different point in space using shifting law involving,

$$\mathbf{H}(T) = \mathbf{H}(S) + \vec{TS} \times \mathbf{u} \quad (1.5)$$

Such an evaluation is motivated by the fact that the screw product considered in evaluating the derivative of sensor signal is independent of reference point S considered in (1.2). It is also part of derivation of interaction model considered in Section (3.1). Evaluating interaction screw at a different point in space can enable to understand its characteristic if it leads to simplification. A slider is a screw whose value at a particular point is zero, i.e., $\mathbf{H}(T) = \mathbf{0}$. This leads to simpler characterization of the interaction screw to enable in better understanding of the interaction when several sensor signals are used. More discussion on this is provided in Chapter 10.

When p such sensor signals are used in SBC, we obtain interaction screws associated with each of them. It is often represented in matrix form denoted as interaction matrix,

$$\mathbf{L}_s = \begin{pmatrix} \mathbf{u}_1^T & \mathbf{H}_1^T(S) \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \mathbf{u}_p^T & \mathbf{H}_p^T(S) \end{pmatrix} \quad (1.6)$$

such that

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v}_{\mathcal{F}_S/\mathcal{F}_T} \quad (1.7)$$

where $\dot{\mathbf{s}}$ is the time variation of vector \mathbf{s} consisting of sensor signals and $\mathbf{v}_{\mathcal{F}_S/\mathcal{F}_T} = \begin{pmatrix} \mathbf{v}_{S/\mathcal{F}_T} \\ \omega_{\mathcal{F}_S/\mathcal{F}_T} \end{pmatrix}$ is the vector representation of \mathcal{K}_{ST} .

1.3 Virtual Linkage

Let us consider a set of p sensors attached to robotic link i which again corresponds to p interaction screws $\mathcal{H}_1, \dots, \mathcal{H}_p$. In that case it is possible to identify velocity screws \mathcal{K}^* that leaves the sensor signals unchanged,

$$\dot{\mathbf{s}} = \mathcal{H}_i \circ \mathcal{K}^* = 0 \quad (1.8)$$

Here \mathcal{K}^* is called as the reciprocal to linear space involving the p interaction screws. When the desired value of sensor signals represented as \mathbf{s}^* is reached a constraint $\mathbf{s} - \mathbf{s}^* = 0$ is achieved in sensor space. This constraint can be imagined to as imposing a virtual linkage between the robotic link and the target object. \mathcal{K}^* would then indicate the possible motions allowed for the linkage and the dimension of this reciprocal subspace denotes the class of virtual linkage.

1.4 Sensor-based Task

A sensor-based task of dimension m can be built using a set of p sensor signals \mathbf{s} and a combination matrix \mathbf{C} as,

$$\mathbf{e} = \mathbf{C} (\mathbf{s} - \mathbf{s}^*) \quad (1.9)$$

where \mathbf{s}^* is the desired sensor signals to be achieved through task execution and $\mathbf{C}\mathbf{s}^*$ is the reference task vector. Regulation of the task \mathbf{e} helps the user to achieve the desired robotic behavior in SBC framework. If $p = m$ then we have a minimal set of sensors used for the task and \mathbf{C} can be chosen as the identity matrix. If $p > m$, then we use a redundant number of sensors for executing the task and matrix \mathbf{C} has to be selected adequately. An example will be given in Section 4.2.2 and 6.2.2.

STATE OF THE ART

In SBC, camera is widely used through visual servoing schemes for positioning tasks. However, vision 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. 7 cm¹. Additionally, 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 [Navarro et al., 2022]. Main technologies here such as capacitive, infrared optical, and ultrasonic sensors [Cutkosky et al., 2008] 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 [Navarro et al., 2022]. 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. Use of such information in closed-loop control is defined as Proximity-based Control.

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 [Espiau, 1988] the need for multi-sensory systems to provide local environment sensing is emphasized to realize non-trivial tasks and to provide flexibility and transparency from the point of view of control. A multi-proximity sensor system from [Andre, 1985] which can be arranged on a robot arm or as end-effector is considered as an ideal example. Later in [Espiau, 1990], various proximity sensor combinations were proposed by the author to be attached as end-effector to perform robotic tasks. For guidance of an end-effector inside a cylindrical pipe a sensor system consisting of two rings is proposed with 4 proximity sensors each

1. <https://www.intelrealsense.com/depth-camera-d405/>

arranged in a symmetric manner. For plane-to-plane positioning two systems are considered, one involving 3 sensors and another with 4 sensors each arranged with sensor axes being parallel to each other. Additionally, a two sensor system is proposed for the problem of target tracking in a plane. In [Nunes et al., 1994], contour following task is considered in Cartesian space, by emphasizing on the distance and orientation of the surface detected. Task is implemented in a 6 DoF serial manipulator by attaching an ultrasound ranging system consisting of three sensors to the end-effector that can detect surfaces of low curvature. The application domain is considered as industrial operations such as painting, welding, glues administration and surface polishing. In [Navarro et al., 2016], for performing 3D contour following a cylindrical end-effector consisting of capacitive sensors are attached to a serial manipulator. The task achieves in aligning the sensor parallel to the surface and at a desired distance. In [Devigne et al., 2019], proximity sensors were mounted around a wheelchair to provide assistive solutions while avoiding negative obstacles (curbs, steps, descending slopes, etc.).

When proximity sensors are arranged around the arms of a robot it acts like a sensitive skin that is described as a paradigm for sensing and control [Lumelsky et al., 2001]. It is considered in general as a large array of sensors that can cover the entire surface of a machine or part of human body and provide local sensing of surrounding via modalities such as proximity, touch, pressure, temperature, etc. This is considered to enable robots to operate in unstructured environment and perform dexterous motions, as every point on its surface would have enough information about its surrounding through local sensing. The sensitive skin is envisioned to provide robots with a character trait of being *cautious* to its surroundings. One early implementation of such a skin can be seen in Figure 2.1. A recent work in [Cheng et al., 2019] considers the realization of artificial skin for robots providing multiple human-like sensing modalities such as temperature, pressure, acceleration and proximity information. This skin is attached to dual arm manipulator and humanoid robot to exploit applications involving tactile sensing. The main application of such a sensitive skin using the modality of proximity sensing is to perform the task of obstacle avoidance. This is described in details in Part II of the thesis.

By building task functions directly from proximity signals, we can benefit from the advantages of SBC framework. Our work can be considered a continuation of the seminal work in [Espiau, 1990] 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

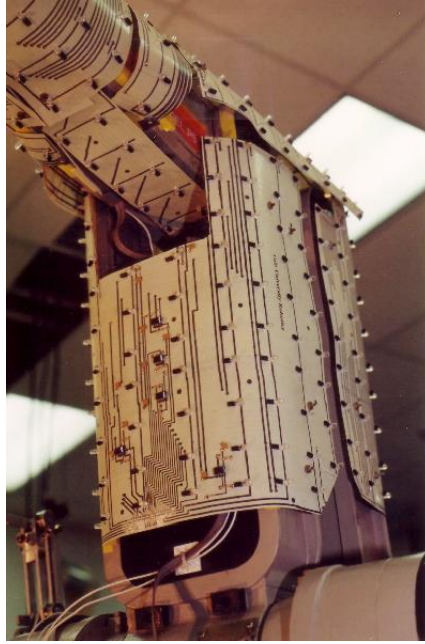


Figure 2.1 – One of the first major work on proximity skin covering whole manipulator arm [Cheung & Lumelsky, 1992] (Photo taken from : <https://www.edcheung.com/job/sbr/sbr.html>)

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. This task was performed with a specific parallel arrangement of proximity sensors in [Espiau, 1990] while, as just described above, we use another arrangement. In [Pages et al., 2004, 2006], a camera is augmented with a structured light system consisting of four laser pointers to achieve this task. Vision-based task functions are built from the camera coordinates of the projected pattern for applications including docking, welding, or painting. Another task that can be performed using the sensor system is positioning wrt. a cylinder. This task has been considered in [Espiau, 1990] as sensor-aided guidance inside a cylindrical pipe using a mobile body. In [Samson et al., 1991], positioning task wrt. to the outside of a cylinder is considered using a parallel arrangement of proximity sensors. This task can also be considered using vision sensor as described in the seminal work in [Chaumette et al., 1993]. In [Chaumette, 1994], positioning of a camera with respect to a motionless cylinder is considered using visual features associated with its projection in the image plane. In non-degenerate cases, this projection is composed of two straight lines and their polar

representation can be used as visual features. In [Rives & Borrelly, 1997], the same set of image features is used to perform underwater pipe inspection. Bi-normalized Plucker coordinates of cylindrical border is considered as image features in [Allibert et al., 2019] to perform pipe following task using a nonlinear visual servo control approach. In [J. Thomas et al., 2014], points on the cylinder that cast tangential rays and remain in the sagittal plane of a quadrotor system were used to perform high-speed aerial grasping task. However, detection of the cylinder using vision sensors requires a large distance relative to the radius of the cylinder due to the limited field of view in detecting the edges involved. While using proximity sensors, we perceive points on the surface of the cylinder at much closer distance. This enables achieving much closer positioning of the system wrt. cylindrical target. We use proximity signals to execute both the above tasks 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/cylindrical object compared to vision, and a more reactive nature due to its higher frequency.

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 [Khalil, 2015]. Applied to SBC, two different situations occur [Chaumette & Hutchinson, 2006]. 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 [Chaumette & Hutchinson, 2006]. The tasks considered in this part falls into the first category. Even while using redundant number of sensors, these measurements are combined together using combination matrix to give task functions of good dimension.

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 [Chaumette & Hutchinson, 2006]. 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 [Chaumette, 1998]. In hybrid ap-

proaches 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 [Malis et al., 1999]. Moreover, sufficient conditions of GAS and both necessary and sufficient conditions for LAS wrt. camera calibration errors were also obtained in [Malis et al., 1999]. The study of stability has been further extended to few other controllers in [Malis & Chaumette, 2002], 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 [Espiau, 1993]. In [Pages et al., 2006], 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 the next chapter we discuss about the multi-sensory proximity system considered for performing various robotic tasks.

PROXIMITY ARRAY

3.1 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 sensor. As depicted in Figure 3.1, \mathbf{n}_S denotes the unit vector indicating sensor axis and δ the distance measured by the sensor wrt. the target. Detection point T indicates the point on the object surface where sensor axis intersects. \mathbf{n}_T indicated the local surface normal at detection point. \mathcal{F}_S represents the sensor coordinate frame located at origin S and \mathcal{F}_T represents the target frame. Let us

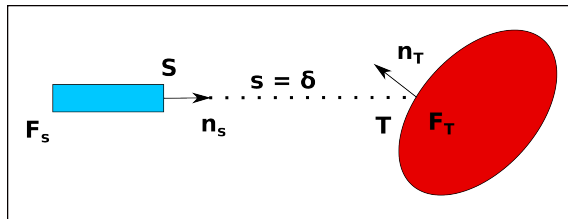


Figure 3.1 – Thin field proximity sensor reproduced from [Samson et al., 1991]. The blue rectangle indicates the sensor and red ellipse the target detected.

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

$$s = \delta \quad (3.1)$$

The model for the first order variation of the sensor feature is given in [Samson et al., 1991] as,

$$\begin{aligned} \dot{\delta} &= \frac{-1}{\mathbf{n}_T \cdot \mathbf{n}_S} (\mathbf{n}_T \cdot \mathbf{v}_{S/\mathcal{F}_T} + \delta (\mathbf{n}_S \times \mathbf{n}_T) \cdot \omega_{\mathcal{F}_S/\mathcal{F}_T}) \\ &= \begin{pmatrix} \mathbf{u}^T & \mathbf{H}(S)^T \end{pmatrix} \begin{pmatrix} \mathbf{v}_{S/\mathcal{F}_T} \\ \omega_{\mathcal{F}_S/\mathcal{F}_T} \end{pmatrix} \end{aligned} \quad (3.2)$$

where $\mathbf{v}_{S/\mathcal{F}_T}$ represents the translational velocity of sensor point S in \mathcal{F}_T and $\omega_{\mathcal{F}_S/\mathcal{F}_T}$ represents the angular velocity of \mathcal{F}_S wrt. \mathcal{F}_T . Point T slides on the surface perpendicular to the local target normal i.e, $\mathbf{n}_T \cdot \mathbf{v}_{T/\mathcal{F}_T} = 0$. Interaction screw $\mathcal{H} = (\mathbf{H}(S), \mathbf{u})$ evaluated at point S is given by,

$$\begin{aligned} \mathbf{u} &= -\frac{\mathbf{n}_T}{\mathbf{n}_T \cdot \mathbf{n}_S} \\ \mathbf{H}(S) &= \delta \mathbf{n}_S \times \mathbf{u} \end{aligned} \quad (3.3)$$

As pointed out it in [Espiau, 1990],

$$\begin{aligned} \mathbf{v}_{S/\mathcal{F}_T} &= \mathbf{v}_{S/\mathcal{F}_o} - \mathbf{v}_{T/\mathcal{F}_o} - \omega_{\mathcal{F}_T/\mathcal{F}_o} \times \vec{TS} \\ \omega_{\mathcal{F}_S/\mathcal{F}_T} &= \omega_{\mathcal{F}_S/\mathcal{F}_o} - \omega_{\mathcal{F}_T/\mathcal{F}_o} \end{aligned} \quad (3.4)$$

where \mathcal{F}_o is the fixed frame. Substituting (3.4) in (3.2) we get,

$$\begin{aligned} \dot{\delta} &= \begin{pmatrix} \mathbf{u}^T & \mathbf{H}(S)^T \end{pmatrix} \begin{pmatrix} \mathbf{v}_{S/\mathcal{F}_o} - \mathbf{v}_{T/\mathcal{F}_o} - \omega_{\mathcal{F}_T/\mathcal{F}_o} \times \vec{TS} \\ \omega_{\mathcal{F}_S/\mathcal{F}_o} - \omega_{\mathcal{F}_T/\mathcal{F}_o} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{u}^T & \mathbf{H}(S)^T \end{pmatrix} \begin{pmatrix} \mathbf{v}_{S/\mathcal{F}_o} \\ \omega_{\mathcal{F}_S/\mathcal{F}_o} \end{pmatrix} - \begin{pmatrix} \mathbf{u}^T & \mathbf{H}(S)^T \end{pmatrix} \begin{pmatrix} \mathbf{v}_{T/\mathcal{F}_o} + \omega_{\mathcal{F}_T/\mathcal{F}_o} \times \vec{TS} \\ \omega_{\mathcal{F}_T/\mathcal{F}_o} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{u}^T & \mathbf{H}(S)^T \end{pmatrix} \begin{pmatrix} \mathbf{v}_{S/\mathcal{F}_o} \\ \omega_{\mathcal{F}_S/\mathcal{F}_o} \end{pmatrix} - \begin{pmatrix} \mathbf{u}^T & (\mathbf{H}(S) + \vec{TS} \times \mathbf{u})^T \end{pmatrix} \begin{pmatrix} \mathbf{v}_{T/\mathcal{F}_o} \\ \omega_{\mathcal{F}_T/\mathcal{F}_o} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{u}^T & \mathbf{H}(S)^T \end{pmatrix} \begin{pmatrix} \mathbf{v}_S \\ \omega_{\mathcal{F}_S} \end{pmatrix} \Big|_{\mathcal{F}_o} - \begin{pmatrix} \mathbf{u}^T & \mathbf{H}(T)^T \end{pmatrix} \begin{pmatrix} \mathbf{v}_T \\ \omega_{\mathcal{F}_T} \end{pmatrix} \Big|_{\mathcal{F}_o} \end{aligned} \quad (3.5)$$

In the above equation we have evaluated the contribution of target motion at T . Shifting law was involved in this calculation. We can compute the value of the screw at T as,

$$\mathbf{H}(T) = \mathbf{H}(S) + \vec{TS} \times \mathbf{u} \quad (3.6)$$

$$= -\frac{\delta}{\mathbf{n}_T \cdot \mathbf{n}_S} (\mathbf{n}_S \times \mathbf{n}_T) + (-\delta \mathbf{n}_S) \times -\frac{\mathbf{n}_T}{\mathbf{n}_T \cdot \mathbf{n}_S} \quad (3.7)$$

$$= -\frac{\delta \mathbf{n}_S \times \mathbf{n}_T}{\mathbf{n}_T \cdot \mathbf{n}_S} + \frac{\delta \mathbf{n}_S \times \mathbf{n}_T}{\mathbf{n}_T \cdot \mathbf{n}_S} = 0 \quad (3.8)$$

This allows us to have a qualitative characterization of the screw that is not associated to the frame of reference but rather to the point of evaluation. It becomes clear that \mathcal{H} is a slider through T with direction along the local normal at point of detection [Samson et al.,

1991]. Additionally it can be noticed that a rotation of the target around T would not change the sensor signal value as expected. For a motionless target the term associated with the velocity of the target nullifies.

We prefer to follow the convention followed in visual servoing in terms of evaluating both interaction and velocity screws in the sensor frame \mathcal{F}_S [Chaumette & Hutchinson, 2006]. For brevity, $\omega_{\mathcal{F}_S}$ will be denoted in short as ω_S and it is assumed that all vectors are evaluated in the same frame if not explicitly mentioned. The model considered in (3.2) simplifies to,

$$\dot{\delta} = \begin{pmatrix} \mathbf{u}^T & \mathbf{H}(S)^T \end{pmatrix} \begin{pmatrix} \mathbf{v}_S \\ \omega_S \end{pmatrix} \quad (3.9)$$

By considering $\mathbf{v}_S = (\mathbf{v}_S, \omega_S)$ as the spatial velocity of the proximity sensor expressed in \mathcal{F}_S , we obtain the classical interaction matrix representation as,

$$\dot{\delta} = \mathbf{L}_\delta \mathbf{v}_S \quad (3.10)$$

The link between the interaction screw and its matrix representation is given by,

$$\mathbf{L}_\delta = \begin{bmatrix} \mathbf{u}^T & \mathbf{H}(S)^T \end{bmatrix} \quad (3.11)$$

The above interaction matrix could be represented in a more simpler form as explained in the subsections below

3.1.1 Approximation on unknown parameter

A common approximation of the interaction matrix is to approximate target normal to be in the opposite direction of sensor axis $\mathbf{n}_T = -\mathbf{n}_S$. We obtain a particular form for the interaction screw,

$$\begin{aligned} \mathbf{u} &= -\mathbf{n}_S \\ \mathbf{H}(S) &= \mathbf{0} \end{aligned} \quad (3.12)$$

The interaction screw which was a slider at target point is also now a slider at sensor point which can be represented concisely as,

$$\mathbf{L}_\delta = \begin{bmatrix} -\mathbf{n}_S^T & \mathbf{0} \end{bmatrix} \quad (3.13)$$

3.1.2 Choice of Sensor Frame

If we choose the sensor frame such that the sensor axis \mathbf{n}_s is along the z-axis of \mathcal{F}_S then from (3.11) and (3.3), the interaction matrix \mathbf{L}_δ is given as,

$$\mathbf{L}_\delta = \begin{bmatrix} -\frac{n_{T_x}}{n_{T_z}} & -\frac{n_{T_y}}{n_{T_z}} & -1 & \delta\frac{n_{T_y}}{n_{T_z}} & -\delta\frac{n_{T_x}}{n_{T_z}} & 0 \end{bmatrix} \quad (3.14)$$

where $\mathbf{n}_T = \begin{bmatrix} n_{T_x} & n_{T_y} & n_{T_z} \end{bmatrix}^T$.

If we consider the approximation in target normal and choose the sensor frame as above we obtain the simplest possible model of interaction matrix,

$$\mathbf{L}_\delta = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \quad (3.15)$$

From (3.14) and (3.3), it seems to be important to evaluate the normal direction \mathbf{n}_T of target at detection point T for using a proximity sensor in a robotic task. In general, it can be stated that it is better to estimate the target normal term \mathbf{n}_T which is unknown. From the point of view of screws, it indicates the direction of the screw and can become important in specific cases as discussed in Section 9.3.2.4. However, thanks to the robustness of closed loop control scheme it is not always necessary. In Section 4.2.1, for the task of plane-to-plane positioning it is shown that target normal can be approximated as the value evaluated at final equilibrium. In Section 6.2.2 for the task of positioning wrt. to the inside of a cylinder approximation discussed in Section 3.1.1 is used.

The property of interaction screw being a slider can be effectively used at modelling stage for a system consisting of multiple proximity sensors. In order to evaluate the interaction matrix at any point in space, we just need 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 $\delta\mathbf{n}_S$.

In the following subsections we look at how individual proximity sensors are arranged to characteristic units that can be attached to a robot. To be able to sense neighbouring

locality with a high spatial resolution from a given number of sensors we consider the initial multi-sensory unit to be arranged in a circular fashion. This enables us to locally sense all regions surrounding the associated link of the robot to which this unit would be rigidly attached.

3.1.3 Proximity Ring

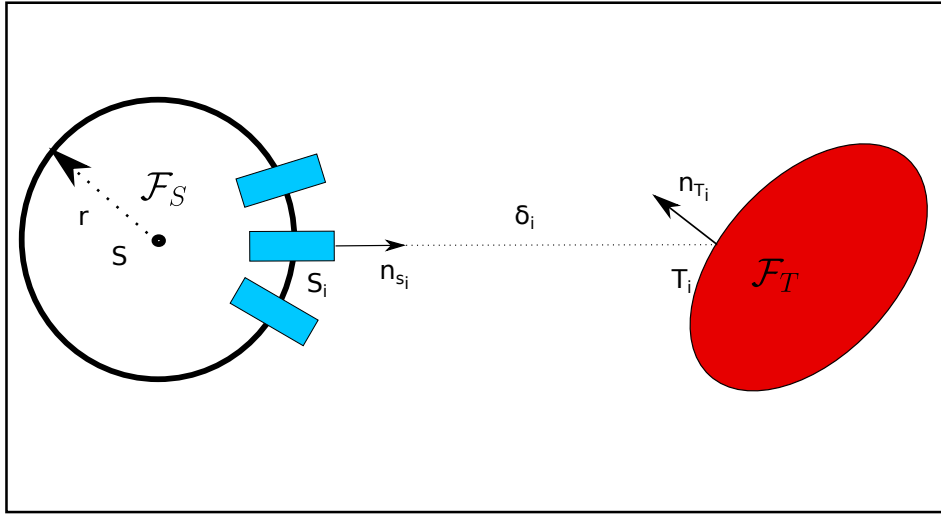


Figure 3.2 – Ring attached with proximity sensors

The first characteristic unit is a proximity ring as shown in Figure 3.2. Each ring consists of m_j proximity sensors arranged in such a way that the axis remains radial and passes through center S_j with radius r_j . Interaction screw of the i th sensor in this ring where $i = \{1, 2, \dots, m_j\}$ is \mathcal{H}_{ij} and is evaluated based on the model in (3.3) at respective sensor points S_{ij} . In order to obtain a common spatial velocity representation, we evaluate the model at ring center S_j . In terms of interaction, it means that we need to shift the screw by a displacement of $r_j \mathbf{n}_{S_{ij}}$. We thus obtain,

$$\begin{aligned} \mathbf{u}_{ij} &= -\frac{\mathbf{n}_{T_{ij}}}{\mathbf{n}_{T_{ij}} \cdot \mathbf{n}_{S_{ij}}} \\ \mathbf{H}_{ij}(S_j) &= (\delta_{ij} + r_j) \mathbf{n}_{S_{ij}} \times \mathbf{u}_{ij} \end{aligned} \quad (3.16)$$

3.1.4 Proximity Array

We now introduce the proximity sensor system that is considered for performing the plane-to-plane positioning and positioning wrt. a cylinder. As depicted in Figure 3.3, let

us consider an array consisting of n rings with radii r_j and centers at S_j where $j = \{1, 2, \dots, n\}$.

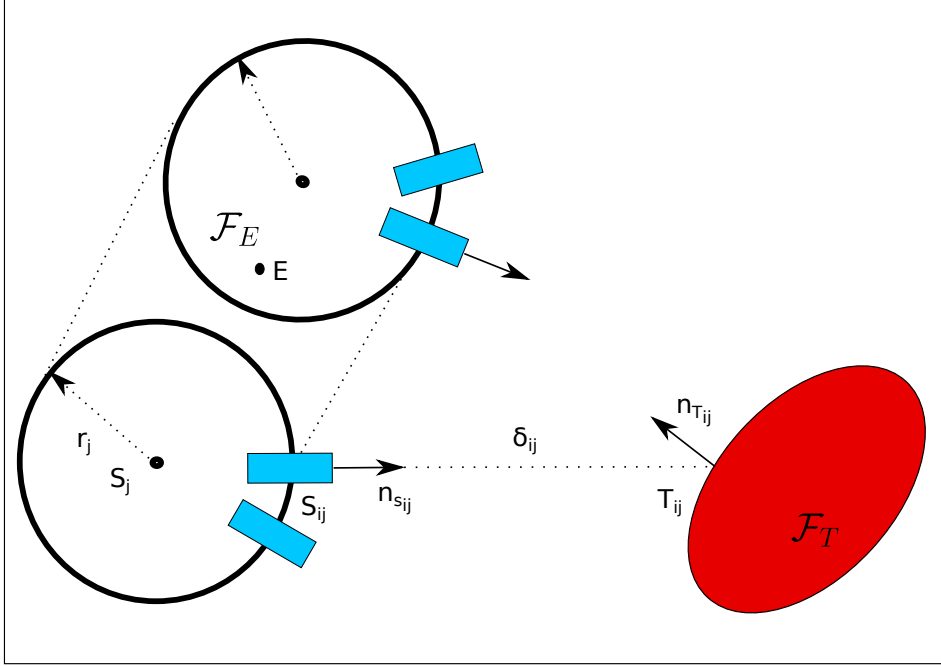


Figure 3.3 – Proximity sensor array consisting of n rings

In order to derive the model for 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 i th proximity sensor attached to the j th ring is obtained as,

$$\begin{aligned} \mathbf{u}_{ij} &= -\frac{\mathbf{n}_{T_{ij}}}{\mathbf{n}_{T_{ij}} \cdot \mathbf{n}_{S_{ij}}} \\ \mathbf{H}_{ij}(E) &= \mathbf{m}_{T_{ij}} \times \mathbf{u}_{ij} \end{aligned} \quad (3.17)$$

where

$$\mathbf{m}_{T_{ij}} = (\delta_{ij} + r_j)\mathbf{n}_{S_{ij}} + \vec{\mathbf{E}}\mathbf{S}_j \quad (3.18)$$

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

$$\dot{\delta}_{ij} = \mathbf{L}_{\delta_{ij}} \mathbf{v}_E \quad (3.19)$$

with $\mathbf{L}_{\delta_{ij}} = [\mathbf{u}_{ij}^T \quad \mathbf{H}_{ij}(E)^T]$.

In the above modelling stage, we have tried to emphasize on the use of interaction screw. A common practice in Sensor-based Control for modelling a multi-sensor system is to obtain the interaction matrix at individual sensor points and then transform it to a common point using twist transformation matrix [Chaumette & Hutchinson, 2006], [Kermorgant & Chaumette, 2011]. Even though the final result would exactly be the same, by modelling it using screw displacements, we obtain more concise set of equations for the case of proximity array.

PLANE-TO-PLANE POSITIONING TASK

4.1 Introduction

In this chapter we use the proximity array considered in Chapter 3 to perform 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.

The following part of the chapter is structured in four sections. In Section 4.2, we introduce plane-to-plane positioning task and describe various aspects in achieving the task. The experimental setup used is discussed in Section 4.3 and the validation of the task considered is discussed in Section 4.4 through experimental results. The final Section 4.5 provides a conclusion for the chapter.

4.2 Plane-to-Plane Positioning

Plane-to-plane positioning task refers to converging to a relative position wrt. the plane that enables the robotic system to scan the plane. To obtain such a convergence, we define a task function (\mathbf{e}) directly using the proximity signals (\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 signal at equilibrium. The first order kinematics of the task can be represented using interaction matrix \mathbf{L}_e corresponding to the task and end-effector velocity \mathbf{v}_E ,

$$\dot{\mathbf{e}} = \mathbf{L}_e \mathbf{v}_E \quad (4.1)$$

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.*, $\mathbf{n}_{T_{ij}} = \mathbf{n}_T$.

$$\begin{aligned}\mathbf{u}_{ij} &= -\frac{\mathbf{n}_T}{\mathbf{n}_T \cdot \mathbf{n}_{S_{ij}}} \\ \mathbf{H}_{ij}(E) &= \mathbf{m}_{T_{ij}} \times \mathbf{u}_{ij}\end{aligned}\tag{4.2}$$

The subspace spanned by interaction screws $H = \text{span}\{\mathbf{H}_{11}, \mathbf{H}_{21}, \mathbf{H}_{12}, \dots\}$ essentially tells us the DoF that are controlled based on the defined sensor-based tasks. Here we have three sliders at target points which are independent as long as the points remain non-collinear. 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 $\{\mathcal{K}^*\}$ of subspace H [Samson et al., 1991]. Elements of this subspace satisfy this condition, leaving the proximity signals invariant to such motions,

$$\dot{\mathbf{s}} = \mathcal{H}_{ij} \circ \mathcal{K}^* = 0\tag{4.3}$$

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 $\{\mathcal{K}^*\}$ [Espiau, 1990], [Chaumette et al., 1993]. 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 the DoF to 2. 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 [Chaumette & Hutchinson, 2006] is used 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.

$$\mathbf{v}_{E,k} = -\lambda \widehat{\mathbf{L}}_{\mathbf{e}_k}^+ \mathbf{e}_k, \quad k = 1, 2 \quad (4.4)$$

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

Figure 4.1 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} .

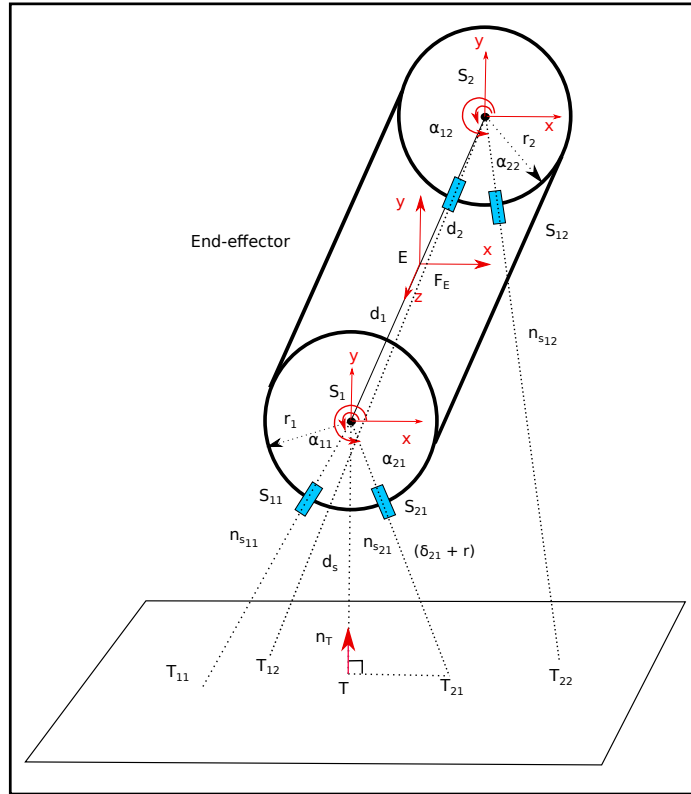


Figure 4.1 – Proximity array for plane-to-plane positioning task

4.2.1 Minimal sensors

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

$$\mathbf{e}_1 = \begin{pmatrix} \delta_{11} - \delta_{11}^* \\ \delta_{21} - \delta_{21}^* \\ \delta_{12} - \delta_{12}^* \end{pmatrix} \quad (4.5)$$

The interaction screw of sensor feature \mathbf{e}_1 is obtained from the special form (4.2) in case of a planar target by substituting the chosen configuration for sensor axis $\mathbf{n}_{S_{ij}}$.

$$\begin{aligned} \mathbf{n}_{S_{ij}} &= \cos \alpha_{ij} \hat{\mathbf{i}} + \sin \alpha_{ij} \hat{\mathbf{j}} \\ \mathbf{u}_{ij} &= -\frac{\mathbf{n}_T}{n_{Tx} \cos \alpha_{ij} + n_{Ty} \sin \alpha_{ij}} \\ \mathbf{m}_{T_{ij}} &= (\delta_{ij} + r_j) \cos \alpha_{ij} \hat{\mathbf{i}} + (\delta_{ij} + r_j) \sin \alpha_{ij} \hat{\mathbf{j}} + d_j \hat{\mathbf{k}} \\ \mathbf{H}_{ij}(E) &= \mathbf{m}_{T_{ij}} \times \mathbf{u}_{ij} \end{aligned} \quad (4.6)$$

where we recall that $\mathbf{n}_T = [n_{Tx} \ n_{Ty} \ n_{Tz}]^T$

The interaction matrix $\mathbf{L}_{\mathbf{e}_1}$ for this task is directly obtained from (3.19) and (4.6):

$$\mathbf{L}_{\mathbf{e}_1} = \begin{bmatrix} \mathbf{u}_{11}^T & \mathbf{H}_{11}(E)^T \\ \mathbf{u}_{21}^T & \mathbf{H}_{21}(E)^T \\ \mathbf{u}_{12}^T & \mathbf{H}_{12}(E)^T \end{bmatrix} \quad (4.7)$$

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 $\mathbf{m}_{T_{ij}}$, we obtain three non-collinear vectors on the plane.

$$\begin{aligned} \mathbf{p}_{1121} &= \mathbf{m}_{S_{11}} - \mathbf{m}_{S_{21}} \\ \mathbf{p}_{2112} &= \mathbf{m}_{S_{21}} - \mathbf{m}_{S_{12}} \\ \mathbf{p}_{1211} &= \mathbf{m}_{S_{12}} - \mathbf{m}_{S_{11}} \end{aligned} \quad (4.8)$$

Since the normal \mathbf{n}_T is perpendicular to each of these vectors, we can build a homogeneous

system of the form,

$$\mathbf{A} \mathbf{n}_T = \mathbf{0} \quad (4.9)$$

where $\mathbf{A} = \begin{bmatrix} \mathbf{p}_{1121}^T \\ \mathbf{p}_{2112}^T \\ \mathbf{p}_{1211}^T \end{bmatrix}$. Nontrivial solution of (4.9) 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 (4.6), *i.e.*, $\delta_{ij} = \delta_{ij}^*$ and $\mathbf{n}_T = \mathbf{n}_T^*$.

$$\begin{aligned} \mathbf{u}_{ij} &= -\frac{\mathbf{n}_T^*}{\mathbf{n}_T^* \cdot \mathbf{n}_{S_{ij}}} \\ \mathbf{m}_{T_{ij}}^* &= (\delta_{ij}^* + r_j) \mathbf{n}_{S_{ij}} + d_j \hat{\mathbf{k}} \\ \mathbf{H}_{ij}(E) &= \mathbf{m}_{T_{ij}}^* \times \mathbf{u}_{ij} \end{aligned} \quad (4.10)$$

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 Figure 4.1). 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 Figure 4.1. 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

$$\delta_{ij}^* = -\frac{d_s}{\sin \alpha_{ij}} - r_j \quad (4.11)$$

The corresponding interaction matrix has the following simple form:

$$\mathbf{L}_{e_1}^* = \begin{bmatrix} 0 & -\frac{1}{\sin \alpha_{11}} & 0 & \frac{d_1}{\sin \alpha_{11}} & 0 & \frac{d_s \cos \alpha_{11}}{\sin^2 \alpha_{11}} \\ 0 & -\frac{1}{\sin \alpha_{21}} & 0 & \frac{d_1}{\sin \alpha_{21}} & 0 & \frac{d_s \cos \alpha_{21}}{\sin^2 \alpha_{21}} \\ 0 & -\frac{1}{\sin \alpha_{12}} & 0 & \frac{d_2}{\sin \alpha_{12}} & 0 & \frac{d_s \cos \alpha_{12}}{\sin^2 \alpha_{12}} \end{bmatrix} \quad (4.12)$$

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} . (4.12) does not depend on parameters r_j explicitly as we have expressed them in terms of desired distance d_s through (4.11).

4.2.2 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 [Espiau, 1990]. 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 Figure 4.1, let us consider a symmetrical sensor arrangement in proximity array,

$$\begin{aligned} \alpha_{11} &= 270^\circ - \alpha, & \alpha_{21} &= 270^\circ + \alpha, & d_1 &= d, & r_1 &= r \\ \alpha_{12} &= 270^\circ - \alpha, & \alpha_{22} &= 270^\circ + \alpha, & d_2 &= -d, & r_2 &= r \end{aligned} \quad (4.13)$$

The symmetry of the above array can be exploited to select sensor features that are less dependent on the extrinsic parameters. One possible choice of task function is

$$\mathbf{e}_2 = \mathbf{C} (\mathbf{s} - \mathbf{s}^*) \quad (4.14)$$

where $\mathbf{C} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$, $\mathbf{s} = \begin{pmatrix} \delta_{11} \\ \delta_{21} \\ \delta_{12} \\ \delta_{22} \end{pmatrix}$ and $\mathbf{s}^* = \begin{pmatrix} \delta_{11}^* \\ \delta_{21}^* \\ \delta_{12}^* \\ \delta_{22}^* \end{pmatrix}$. \mathbf{C} is the combination

matrix that is chosen to obtain a task of dimension 3 and $\mathbf{C} \mathbf{s}^*$ becomes its associated

reference vector.

The interaction matrix $\mathbf{L}_{\mathbf{e}_2}$ for this task can be derived directly from the generic equations (3.19) and (4.6). It is given by

$$\mathbf{L}_{\mathbf{e}_2} = \mathbf{C} \begin{bmatrix} \mathbf{u}_{11}^T & \mathbf{H}_{11}^T \\ \mathbf{u}_{21}^T & \mathbf{H}_{21}^T \\ \mathbf{u}_{12}^T & \mathbf{H}_{12}^T \\ \mathbf{u}_{22}^T & \mathbf{H}_{22}^T \end{bmatrix} \quad (4.15)$$

Simplified Model

To obtain the interaction model at equilibrium, we used the same substitution used in the case of minimal sensors (4.2.1). Interaction screw values obtained using (4.10) can be further substituted into (4.15) 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 reference vector of the task simplifies to,

$$\mathbf{C}\mathbf{s}^* = \begin{pmatrix} 0 \\ 0 \\ \frac{4 d_s}{\cos \alpha} - 4 r \end{pmatrix} \quad (4.16)$$

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 the individual proximity sensors simplifies as,

$$\begin{aligned} \mathbf{u}_{11} &= \frac{1}{\cos \alpha} \hat{\mathbf{j}} \quad ; \quad \mathbf{H}_{11}(E) = -\frac{d}{\cos \alpha} \hat{\mathbf{i}} - \frac{d_s \sin \alpha}{\cos^2 \alpha} \hat{\mathbf{k}} \\ \mathbf{u}_{21} &= \frac{1}{\cos \alpha} \hat{\mathbf{j}} \quad ; \quad \mathbf{H}_{21}(E) = -\frac{d}{\cos \alpha} \hat{\mathbf{i}} + \frac{d_s \sin \alpha}{\cos^2 \alpha} \hat{\mathbf{k}} \\ \mathbf{u}_{12} &= \frac{1}{\cos \alpha} \hat{\mathbf{j}} \quad ; \quad \mathbf{H}_{12}(E) = \frac{d}{\cos \alpha} \hat{\mathbf{i}} - \frac{d_s \sin \alpha}{\cos^2 \alpha} \hat{\mathbf{k}} \\ \mathbf{u}_{22} &= \frac{1}{\cos \alpha} \hat{\mathbf{j}} \quad ; \quad \mathbf{H}_{22}(E) = \frac{d}{\cos \alpha} \hat{\mathbf{i}} + \frac{d_s \sin \alpha}{\cos^2 \alpha} \hat{\mathbf{k}} \end{aligned} \quad (4.17)$$

from which we deduce the very simple form

$$\mathbf{L}_{\mathbf{e}_2}^* = \frac{4}{\cos \alpha} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -d_s \tan \alpha \\ 0 & 0 & 0 & -d & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.18)$$

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 $\mathbf{L}_{\mathbf{e}_2}^*$.

4.2.3 Stability Analysis

Before providing the stability conditions it is required to show the uniqueness of equilibrium point of the task,

$$\mathbf{e}_k = 0 \quad \iff \quad \mathbf{s} = \mathbf{s}^* \quad (4.19)$$

Additionally, it is important also for practical execution of the task. While using minimal sensors it is evident from (4.5). In case of using redundant number of sensors (4.14), there could be a situation for violating condition (4.19) when,

$$\mathbf{s} - \mathbf{s}^* \in \text{Ker } \mathbf{C} \quad (4.20)$$

In Appendix A we show that such a situation is not possible for a practical task as there is no physically consistent configuration satisfying (4.20) with the proximity array model considered.

As is well known in SBC, the system is globally asymptotically stable (GAS) if the following property is ensured [Chaumette & Hutchinson, 2006],

$$\mathbf{L}_{\mathbf{e}_k} \widehat{\mathbf{L}}_{\mathbf{e}_k}^+ > 0 \quad (4.21)$$

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 (4.7) for \mathbf{e}_1 and (4.15) 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 (4.12) and (4.18) 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 through experimentation results that the domain of convergence is very large even when using these simplified models. A more detailed analysis of stability is provided in Chapter 5.

4.3 Proximity Array as an End-Effector

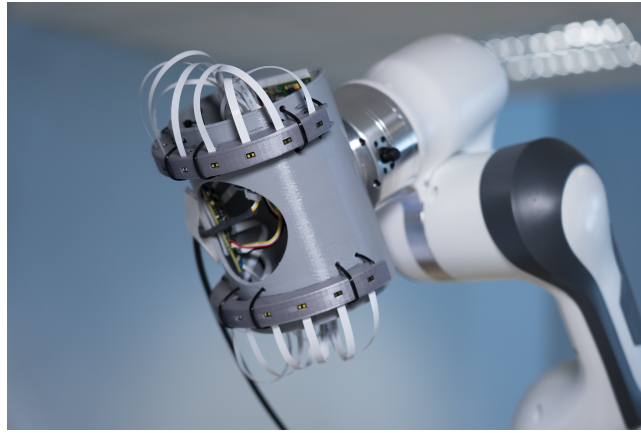


Figure 4.2 – Proximity Array as an end-effector

The proximity array used for experimentation is displayed in Figure 4.2. It consists of two half rings that are rigidly attached to a semi-cylindrical part which is grey in color. Each of these half-rings contains high performance proximity sensors from STMicroelectronics. The model used is VL53L1x¹ which uses *FlightSense*TM Time-of-Flight (ToF) technology to measure distances accurately, based on the time required for emitted photons to be reflected rather than the signal strength received. Hence, these measurements are not affected by the surface characteristics of the obstacle. Maximum field of view for a sensor is 20° *deg* which in turn motivated in installing 18 such sensors in one ring to maximize detection.

For the current experiment these sensors are tuned to short range of 1 *m* to output measurements at 66 *Hz* frequency with 1 *mm* precision and 5° field of view (FoV). Lower FoV is chosen to be more closer to the modeling approximation of thin field of view. The minimum ranging distance of the sensors is 4 *cm*.

1. <https://www.st.com/en/imaging-and-photonics-solutions/vl53l1x.html>

4.4 Experimental Results



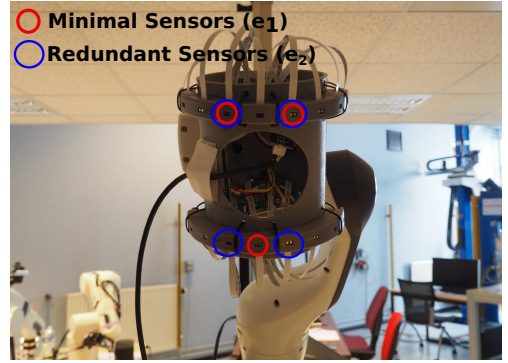
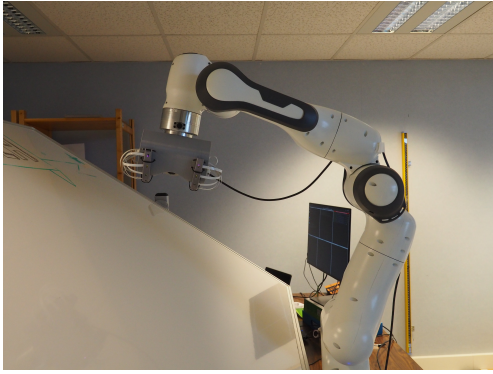
Figure 4.3 – Experimental setup when proximity array is used as an end-effector

The experimental setup consists of a Panda robot, which is a 7 DoF serial manipulator developed by Franka Emika to which the proximity array is attached as end-effector as shown in Figure 4.3. Figure 4.4b 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 4.1. 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 \times 8. Laptop runs on Ubuntu 20.04.2 LTS with RTLinux kernel in Fully Preemptible Mode.

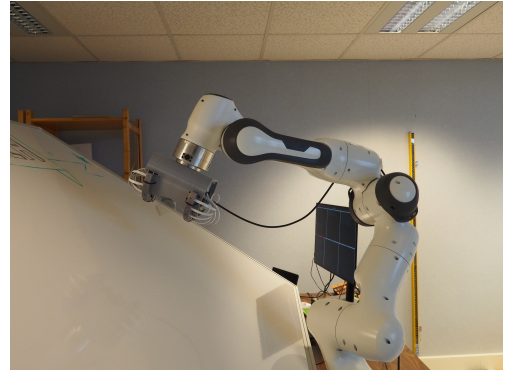
In this section, we show the validation of our model for performing the described task. We consider three scenarios for the experiments starting from the same initial pose shown in Figure 4.4a. The initial pose is chosen such that the proximity signal readings are just within the range of 50 *cm*, which is considered a suitable operational range for proximity sensors [Navarro et al., 2022]. In the first case, we show convergence to an arbitrary final pose, where the proximity measurements are less than 30 *cm*, as shown in Figure 4.4c. 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



(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

Figure 4.4 – Experimental setup

converge to a final pose shown in Figure 4.4d, 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 (4.11) with value $d_s = 12 \text{ cm}$, 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 (4.4) is set to $\lambda = 0.8$, whereas for the final case we use $\lambda = 1$. The video of the experimentation results can be seen here.

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

Table 4.1 – Extrinsic parameters of proximity array.

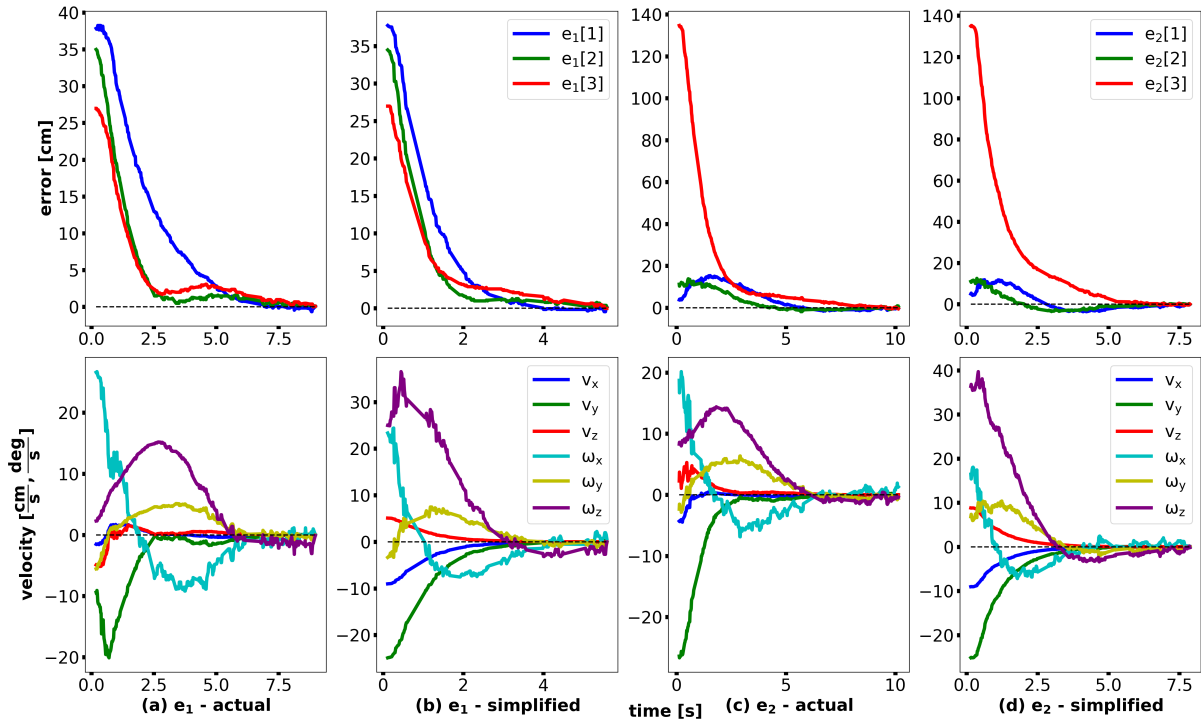


Figure 4.5 – Experimental results for Case 1 using both task functions \mathbf{e}_1 (on the left) and \mathbf{e}_2 (on the right) with actual ((a) and (c)) and simplified ((b) and (d)) interaction models. Task function components (cm) versus time (s) on the top, end-effector velocities (cm/s and deg/s) versus time (s) on the bottom).

4.4.1 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 Figure 4.4c). The desired sensor signal δ_{ij}^* , obtained from offline teaching, is shown in Table 4.2. Additionally, the table also contains the desired target normal used for the simplified models and the initial proximity signal values (δ_{ij}^0). Figure 4.5

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

Table 4.2 – Parameters for Case 1.

displays the plots 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 (4.4), $\widehat{\mathbf{L}}_{\mathbf{e}_k} = \mathbf{L}_{\mathbf{e}_k}$ 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) 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.

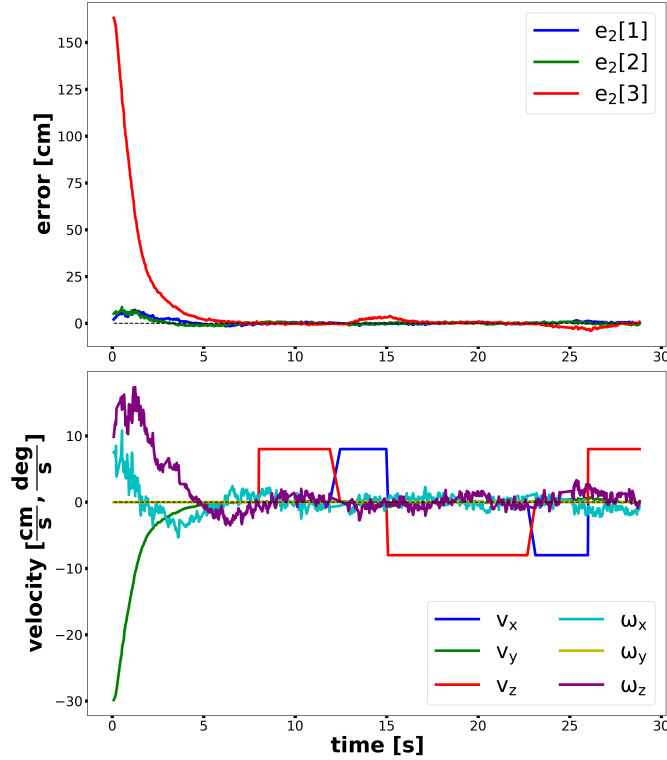


Figure 4.6 – Experimental results for Case 2 using task function \mathbf{e}_2 with simplified interaction model. Task function components (cm) versus time (s) on the top, end-effector velocities (cm/s and deg/s) versus time (s) on the bottom.

4.4.2 Case 2

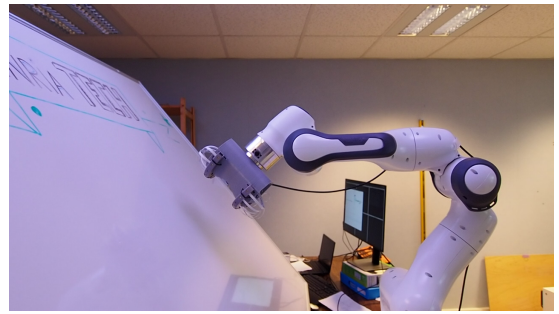
In this scenario, we use task \mathbf{e}_2 with the simplified model to show the special case where the end-effector reaches a pose parallel to the target plane at a distance $d_s = 12\text{ cm}$, as shown in Figure 4.4d. Once the system has converged, we implement a basic surface inspection task by 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 Figure 4.6, 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 of $\mathbf{L}_{\mathbf{e}_2}^*$ in (4.18)). Even though the sensor arrangement for the experimental case is not exactly symmetric, we still obtain a performance close to the decoupled behaviour in (4.18). This can be observed in Figure 4.6, 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.

4.4.3 Case 3



(a) Initial convergence sequence with static target plane



(b) At convergence after rotation of target plane



(c) Final convergence after rotation plus translation sequence

Figure 4.7 – Different configurations of the robot and the experimental setup for Case 3

In this final scenario, task \mathbf{e}_2 is executed using the same simplified model as that of Case 2. At equilibrium, we perform sequences of a rotation and a translation to the

plane target. The position of the target plane after each of these sequence can be seen in Figure 4.7. It can be understood by looking at the relative position of the target plane in Figure 4.7 (b) and (c) in comparison to Figure 4.7 (a). Figure 4.7 (a) is the same as final pose of Case 2 which is shown in Figure 4.4d. As shown in Figure 4.8, the control scheme ensures rapid reactions to such unexpected motions of the target object and thereby ensuring parallel positioning. The convergence is achieved with satisfactory effect even though target motions were not accounted for in modelling. The gain of $\lambda = 1$ is chosen for the experimentation empirically to obtain faster reaction than previous cases.

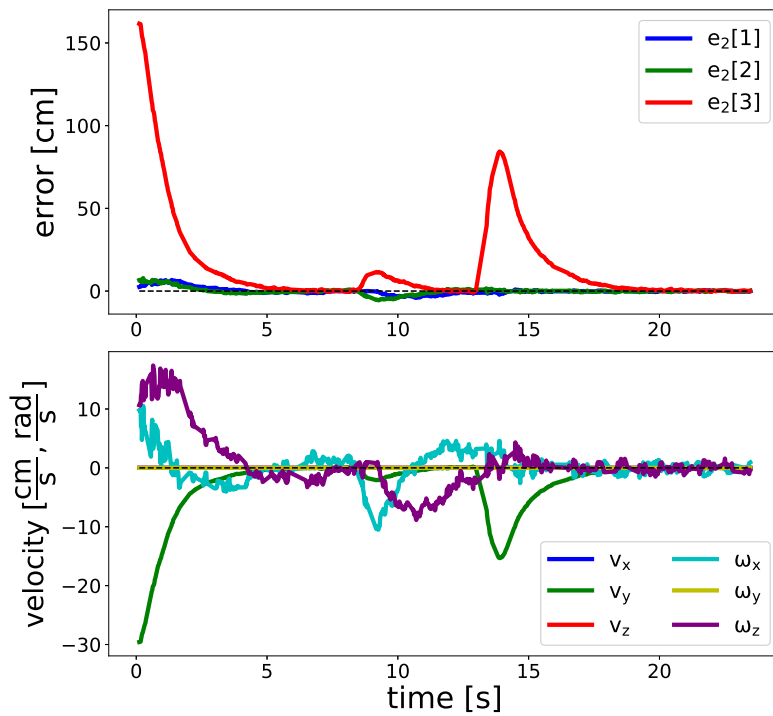


Figure 4.8 – Experimental results for Case 3 using task function \mathbf{e}_2 with simplified interaction model. Task function components (cm) versus time (s) on the top, end-effector velocities (cm/s and deg/s) versus time (s) on the bottom.

4.5 Conclusion

In this chapter, the modeling strategy developed for a generic proximity sensor arrangement was applied to the task of plane-to-plane positioning. This was further validated through experimentation. The task considered may be involved for the inspection of large industrial texture-less parts.

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 seems to be large even though it was demonstrated to be only locally asymptotically stable. Further studies must be done to see how large this domain actually is.

STABILITY ANALYSIS OF PLANE-TO-PLANE POSITIONING TASK

In this chapter, we study the stability of plane-to-plane positioning tasks considered in Chapter 4. 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 the analytical form of the pseudo-inverse of interaction matrix for both cases. This analytical form enables us to 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 plane 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 controller for both minimal and redundant number of proximity sensors.

This chapter is divided into five sections including the current one. In Section 5.1, we reformulate the basic aspects of plane-to-plane positioning task to a form suitable for the chapter. Section 5.2 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 5.2 ends with the stability conditions on these matrices for few scenarios with the application of Gershgorin's theorem. In Sections 5.3, we provide validation to the theory through simulation results. Finally we end the paper with conclusions in Section 5.4.

5.1 Plane-to-Plane Positioning Task

In this section we briefly summarize the modeling developed in Chapter 4. The modeling in the previous chapter was adapted to the nomenclature used in identifying a single proximity sensor in proximity array as described in Chapter 3. However, in this chapter we try to adapt it to have concise equations while analyzing stability.

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^* :

$$\mathbf{e}_m = \begin{pmatrix} \cdot \\ \delta_i - \delta_i^* \\ \cdot \end{pmatrix}_{3 \times 1} \quad (5.1)$$

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

$$\mathbf{L}_{\mathbf{e}_m} = \begin{bmatrix} \cdot & \cdot \\ \mathbf{u}_i^T \mathbf{H}_i(E)^T & \\ \cdot & \cdot \end{bmatrix}_{3 \times 6} = \begin{bmatrix} \cdot & \cdot \\ {}^m\beta_i \mathbf{n}_T^T ({}^m\mathbf{m}_{\beta_i} \times \mathbf{n}_T)^T & \\ \cdot & \cdot \end{bmatrix}_{3 \times 6} \quad (5.2)$$

with

$$\begin{aligned} \mathbf{u}_i &= -\frac{\mathbf{n}_T}{\mathbf{n}_T \cdot \mathbf{n}_{S_i}} = {}^m\beta_i \mathbf{n}_T \\ \mathbf{H}_i(E) &= \mathbf{m}_{T_i} \times \mathbf{u}_i = {}^m\mathbf{m}_{\beta_i} \times \mathbf{n}_T \\ \mathbf{n}_{S_i} &= \cos \alpha_i \hat{\mathbf{i}} + \sin \alpha_i \hat{\mathbf{j}} \\ \mathbf{m}_{T_i} &= (\delta_i + r_i) \cos \alpha_i \hat{\mathbf{i}} + (\delta_i + r_i) \sin \alpha_i \hat{\mathbf{j}} + d_i \hat{\mathbf{k}} \\ {}^m\beta_i &= -\frac{1}{\mathbf{n}_T \cdot \mathbf{n}_{S_i}}, \quad {}^m\mathbf{m}_{\beta_i} = {}^m\beta_i \mathbf{m}_{T_i} \end{aligned} \quad (5.3)$$

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

$$\mathbf{e}_r = \mathbf{C} \begin{pmatrix} \cdot \\ \delta_i - \delta_i^* \\ \cdot \end{pmatrix}_{4 \times 1} \quad (5.4)$$

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

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

$$\begin{aligned} \mathbf{L}_{\mathbf{e}_r} &= \mathbf{C} \begin{bmatrix} \cdot & \cdot \\ \mathbf{u}_i^T & \mathbf{H}_i(E)^T \\ \cdot & \cdot \end{bmatrix}_{4 \times 6} \\ &= \begin{bmatrix} \cdot & \cdot \\ {}^r\beta_i \mathbf{n}_T^T & ({}^r\mathbf{m}_{\beta_i} \times \mathbf{n}_T)^T \\ \cdot & \cdot \end{bmatrix}_{3 \times 6} \end{aligned} \quad (5.5)$$

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

$${}^r\beta_i = \sum_{j=1}^4 \frac{-C(i,j)}{\mathbf{n}_T \cdot \mathbf{n}_{S_j}}, \quad {}^r\mathbf{m}_{\beta_i} = \sum_{j=1}^4 \frac{-C(i,j)}{\mathbf{n}_T \cdot \mathbf{n}_{S_j}} \mathbf{m}_{T_j} \quad (5.6)$$

Finally, the standard control law considered in Chapter 4 with $k = m$ for the minimal case or $k = r$ for the redundant case, is used for achieving the task,

$$\mathbf{v}_{E,k} = -\lambda \widehat{\mathbf{L}}_{\mathbf{e}_k}^+ \mathbf{e}_k, \quad k = m, r \quad (5.7)$$

with

$$\mathbf{L}_{\mathbf{e}_k} = \begin{bmatrix} \cdot & \cdot \\ {}^k\beta_i \mathbf{n}_T^T & ({}^k\mathbf{m}_{\beta_i} \times \mathbf{n}_T)^T \\ \cdot & \cdot \end{bmatrix}_{3 \times 6} \quad (5.8)$$

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 $\widehat{\mathbf{L}}_{\mathbf{e}_k}$. In the following sections a hat symbol indicates whether the term is estimated or approximated.

5.2 Stability Analysis

The closed loop dynamic equation for the system is obtained by substituting (5.7) in the kinematic equation (4.1) of the task

$$\dot{\mathbf{e}}_k = -\lambda \mathbf{L}_{\mathbf{e}_k} \widehat{\mathbf{L}}_{\mathbf{e}_k}^+ \mathbf{e}_k = -\lambda \mathbf{M}_p(\mathbf{e}_k, \Xi_k, \hat{\Xi}_k) \mathbf{e}_k \quad (5.9)$$

where $\Xi_m = \{d_1, d_2, r_1, r_2, \alpha_{11}, \alpha_{21}, \alpha_{12}\}$ indicate extrinsic parameters for minimal case and $\Xi_r = \{d_1, d_2, r_1, r_2, \alpha_{11}, \alpha_{21}, \alpha_{12}, \alpha_{22}\}$ is for the case of redundant number of sensors. From now on we drop Ξ_1 and Ξ_2 for brevity. As well known, the system is globally asymptotically stable (GAS) if the following property is ensured [Khalil, 2015],

$$\mathbf{M}_p(\mathbf{e}_k) > 0 \quad (5.10)$$

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 $\hat{\Xi}_k$ 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 [Strang, 2005]:

$$\lambda_m > 0 \quad (5.11)$$

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.

5.2.1 Gershgorin's theorems

Gershgorin's theorems help us to provide circular disc regions in which characteristic roots lie. Two important theorems are stated below [Wilkinson, 1988]:

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 the circular discs by providing corresponding center and radius. The centers of these discs are its corresponding diagonal term and the radii is composed of the off-diagonal terms. 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 (5.11) 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

$$O_i = m_{ii} \quad , \quad \rho_i = \sum_{j \neq i} |m_{ij}| \quad (5.12)$$

To satisfy the test (5.11), we need to ensure,

$$\min \{O_i - \rho_i : i = 1, 2, 3\} > 0 \quad (5.13)$$

5.2.2 Analytical Form of the pseudo-inverse

The pseudo-inverse of \mathbf{L}_{e_k} given in (5.8) 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:

$$\mathbf{L}_{e_k}^+ = \mathbf{P} \mathbf{L}_{e_k}^- \quad (5.14)$$

where

$$\mathbf{P} = \begin{bmatrix} \mathbf{n}_T \mathbf{n}_T^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 - \mathbf{n}_T \mathbf{n}_T^T \end{bmatrix} \quad (5.15)$$

$$\mathbf{L}_{e_k}^- = \frac{1}{l} \begin{bmatrix} \cdot & {}^k \mathbf{m}_{\beta_{j+}} \times {}^k \mathbf{m}_{\beta_{j-}} & \cdot \\ \cdot & {}^k \beta_{j+} {}^k \mathbf{m}_{\beta_{j-}} - {}^k \beta_{j-} {}^k \mathbf{m}_{\beta_{j+}} & \cdot \end{bmatrix}$$

$$l = \sum_{i=1}^3 \mathbf{n}_T \cdot ({}^k \beta_{i-} {}^k \mathbf{m}_{\beta_i} \times {}^k \mathbf{m}_{\beta_{i+}}) \quad (5.16)$$

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 [Strang, 2005]. 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 $\mathbf{L}_{\mathbf{e}_k}$ with \mathbf{P} , we get back $\mathbf{L}_{\mathbf{e}_k}$. Indeed, from (5.8), we have

$${}^k\beta_i \mathbf{n}_T^T (\mathbf{n}_T \mathbf{n}_T^T) = {}^k\beta_i (\mathbf{n}_T^T \mathbf{n}_T) \mathbf{n}_T^T = {}^k\beta_i \mathbf{n}_T^T \quad (5.17)$$

$$\left({}^k\mathbf{m}_{\beta_i} \times \mathbf{n}_T \right)^T \left(\mathbf{n}_T \mathbf{n}_T^T \right) = \left(\left({}^k\mathbf{m}_{\beta_i} \times \mathbf{n}_T \right)^T \mathbf{n}_T \right) \mathbf{n}_T^T = \mathbf{0}^T \quad (5.18)$$

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

From the property $\mathbf{L}_{\mathbf{e}_k} \mathbf{P} = \mathbf{L}_{\mathbf{e}_k}$, it is clear that matrix $\mathbf{L}_{\mathbf{e}_k}^-$ is a reflexive generalized (left-)inverse [Rao, 1972] of $\mathbf{L}_{\mathbf{e}_k}$ since it satisfies

$$\mathbf{L}_{\mathbf{e}_k} \mathbf{L}_{\mathbf{e}_k}^- \mathbf{L}_{\mathbf{e}_k} = \mathbf{L}_{\mathbf{e}_k} \quad (5.19)$$

$$\mathbf{L}_{\mathbf{e}_k}^- \mathbf{L}_{\mathbf{e}_k} \mathbf{L}_{\mathbf{e}_k}^- = \mathbf{L}_{\mathbf{e}_k}^- \quad (5.20)$$

$$\left(\mathbf{L}_{\mathbf{e}_k} \mathbf{L}_{\mathbf{e}_k}^- \right)^T = \mathbf{L}_{\mathbf{e}_k}^- \mathbf{L}_{\mathbf{e}_k} \quad (5.21)$$

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

$$\left(\mathbf{L}_{\mathbf{e}_k}^- \mathbf{L}_{\mathbf{e}_k} \right)^T \neq \mathbf{L}_{\mathbf{e}_k}^- \mathbf{L}_{\mathbf{e}_k} \quad (5.22)$$

As $\mathbf{L}_{\mathbf{e}_k}^-$ is a generalized inverse of $\mathbf{L}_{\mathbf{e}_k}$, it can also be used to execute the task as an alternative to $\mathbf{v}_{E,k}$ given in (5.7) using as control scheme $\mathbf{v}_{E,k}^n = -\lambda \mathbf{L}_{\mathbf{e}_k}^- \mathbf{e}_k$. The potential advantage is that $\mathbf{L}_{\mathbf{e}_k}^+$ highly depends on \mathbf{n}_T through \mathbf{P} while it is not the case for $\mathbf{L}_{\mathbf{e}_k}^-$. For a practical implementation the control scheme will be either estimated or approximated like in (5.7). This results in the analysis of two closed loop matrices $\mathbf{M}_p(\mathbf{e})$ and $\mathbf{M}_n(\mathbf{e})$. In case of $\mathbf{M}_p(\mathbf{e})$ it is an expansion of closed loop matrix considered in (5.9),

$$\mathbf{M}_p(\mathbf{e}) = \mathbf{L}_{\mathbf{e}_k} \widehat{\mathbf{P}} \widehat{\mathbf{L}}_{\mathbf{e}_k}^- \quad (5.23)$$

$$\mathbf{M}_n(\mathbf{e}) = \mathbf{L}_{\mathbf{e}_k} \widehat{\mathbf{L}}_{\mathbf{e}_k}^- \quad (5.24)$$

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 ($\epsilon = \|\mathbf{n}_T \times \widehat{\mathbf{n}}_T\| \neq 0$), and the second when it is perfect ($\epsilon = 0$).

5.2.3 Stability Conditions for Classical Control Law

As detailed in Appendix B, matrix \mathbf{M}_p can be written as the sum of two square matrices of dimension 3.

$$\mathbf{M}_p = \mathbf{A}_p + \mathbf{B}_p \quad (5.25)$$

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

$$\begin{aligned} {}^p a_{ij} &= \frac{{}^p \Delta_{ij}}{pl} \\ {}^p \Delta_{ij} &= \begin{cases} pl_i & i = j \\ (-1)^\gamma \left(\widehat{\mathbf{m}}_{\beta_k} \cdot \epsilon'_i + \widehat{\beta}_k \mathbf{n}_T \cdot \epsilon_i^\times \right) & i \neq j \neq k \end{cases} \\ {}^p l_i &= (\mathbf{n}_T \cdot \widehat{\mathbf{n}}_T) \widehat{\mathbf{n}}_T \cdot \left(\beta_i \widehat{\mathbf{m}}_{\beta_{i+}} \times \widehat{\mathbf{m}}_{\beta_{i-}} \right) \\ &\quad + \mathbf{n}_T \cdot \left(\widehat{\beta}_{i+} \widehat{\mathbf{m}}_{\beta_{i-}} \times \mathbf{m}_{\beta_i} + \widehat{\beta}_{i-} \mathbf{m}_{\beta_i} \times \widehat{\mathbf{m}}_{\beta_{i+}} \right) \\ \gamma &= \begin{cases} 1 & \text{if } (i, j, k) \text{ in clockwise order} \\ 0 & \text{otherwise} \end{cases} \\ pl &= \sum_{i=1}^3 \widehat{\mathbf{n}}_T \cdot \left(\widehat{\beta}_{i-} \widehat{\mathbf{m}}_{\beta_i} \times \widehat{\mathbf{m}}_{\beta_{i+}} \right) \\ \epsilon_i^\times &= \mathbf{m}_{\beta_i} \times \widehat{\mathbf{m}}_{\beta_i} \\ \epsilon'_i &= (\mathbf{n}_T \cdot \widehat{\mathbf{n}}_T) \widehat{\mathbf{n}}_T \times \beta_i \widehat{\mathbf{m}}_{\beta_i} - \mathbf{n}_T \times \widehat{\beta}_i \mathbf{m}_{\beta_i} \end{aligned} \quad (5.26)$$

Diagonal terms ${}^p a_{ii}$ of \mathbf{A}_p depend mainly on terms ${}^p l_i$, that are closely related to l given in (5.16). Off-diagonal terms of \mathbf{A}_p are functions of two main error vectors, ϵ_i^\times and ϵ'_i . The term ϵ_i^\times mainly depends on the errors on the extrinsic parameters while ϵ'_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.

1. Note that we have dropped the superscript k from this point to have simpler notations.

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

$$\begin{aligned}
 {}^p b_{ij} &= -\frac{(\epsilon_{n_T}^\times \cdot \mathbf{m}_{\beta_i}) \epsilon_j^*}{pl} \\
 \epsilon_j^* &= \widehat{\mathbf{n}}_T \cdot (\widehat{\beta_{j+}} \widehat{\mathbf{m}}_{\beta_{j-}} - \widehat{\beta_{j-}} \widehat{\mathbf{m}}_{\beta_{j+}}) \\
 \epsilon_{n_T}^\times &= \mathbf{n}_T \times \widehat{\mathbf{n}}_T
 \end{aligned} \tag{5.27}$$

Terms b_{ij} of matrix \mathbf{B}_p are influenced by the error in surface normal estimation through $\epsilon_{n_T}^\times$. Additionally, $(\widehat{\beta_{j+}} \widehat{\mathbf{m}}_{\beta_{j-}} - \widehat{\beta_{j-}} \widehat{\mathbf{m}}_{\beta_{j+}})$ 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 ($\epsilon = 0$), 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:

$$\begin{aligned}
 O_i &= {}^p a_{ii} + {}^p b_{ii} \\
 \rho_i &= \sum_{j \neq i} \left| \frac{{}^p a_{ij} + {}^p a_{ji}}{2} + \frac{{}^p b_{ij} + {}^p b_{ji}}{2} \right|
 \end{aligned} \tag{5.28}$$

where

$$\begin{aligned}
 {}^p a_{ij} + {}^p a_{ji} &= \frac{(-1)^\gamma}{pl} \left(\widehat{\beta_k} \mathbf{n}_T \cdot (\epsilon_i^\times - \epsilon_j^\times) + \widehat{\mathbf{m}}_{\beta_k} \cdot (\epsilon'_i - \epsilon'_j) \right) \\
 {}^p b_{ij} + {}^p b_{ji} &= \frac{-\epsilon_{n_T}^\times \cdot (\epsilon_j^* \mathbf{m}_{\beta_i} + \epsilon_i^* \mathbf{m}_{\beta_j})}{pl}
 \end{aligned} \tag{5.29}$$

As observed in (5.29), 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 $\epsilon_{n_T}^\times$. It can be observed from (5.29) that it is influenced mainly by $\epsilon_{n_T}^\times$. When surface normal estimation errors are small, this term can be neglected. In Section 5.3.2, 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 $\epsilon_{n_T}^\times$. Section 5.3.3 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 5.3.4, we consider a situation where the errors values are high in $\epsilon_{n_T}^\times$, resulting in violation of stability conditions and eventual failure in executing the task.

5.2.4 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:

$$\mathbf{M}_n = \mathbf{A}_n \quad (5.30)$$

where

$$\begin{aligned} {}^n a_{ij} &= \frac{{}^n \Delta_{ij}}{n_l} \quad (5.31) \\ {}^n \Delta_{ij} &= \begin{cases} {}^n l_i & i = j \\ (-1)^\gamma \mathbf{n}_T \cdot (\widehat{\mathbf{m}}_{\beta_k} \times \epsilon_i^- + \widehat{\beta}_k \epsilon_i^\times) & i \neq j \neq k \end{cases} \\ {}^n l_i &= \mathbf{n}_T \cdot (\beta_i \widehat{\mathbf{m}}_{\beta_{i+}} \times \widehat{\mathbf{m}}_{\beta_{i-}} + \widehat{\beta}_{i+} \widehat{\mathbf{m}}_{\beta_{i-}} \times \mathbf{m}_{\beta_i} \\ &\quad + \widehat{\beta}_{i-} \mathbf{m}_{\beta_i} \times \widehat{\mathbf{m}}_{\beta_{i+}}) \\ {}^n l &= \sum_{i=1}^3 \widehat{\mathbf{n}}_T \cdot (\widehat{\beta}_{i-} \widehat{\mathbf{m}}_{\beta_i} \times \widehat{\mathbf{m}}_{\beta_{i+}}) \\ \epsilon_i^- &= \widehat{\beta}_i \mathbf{m}_{\beta_i} - \beta_i \widehat{\mathbf{m}}_{\beta_i} \end{aligned}$$

from which the corresponding stability conditions become

$$\min \left\{ {}^n a_{ii} - \sum_{j \neq i} \frac{1}{2} |{}^n a_{ij} + {}^n a_{ji}| : i = 1, 2, 3 \right\} > 0 \quad (5.32)$$

where

$${}^n a_{ij} + {}^n a_{ji} = \frac{(-1)^\gamma}{n_l} \left(\widehat{\beta}_k \mathbf{n}_T \cdot \left((\epsilon_i^\times - \epsilon_j^\times) + \widehat{\mathbf{m}}_{\beta_k} \times (\epsilon_i^- - \epsilon_j^-) \right) \right) \quad (5.33)$$

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 (5.29). 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 $\epsilon_{n_T}^\times$ are not present here. This results in higher values for the stability criterium $\min\{O_i - \rho_i\}$, as seen in Sections 5.3.2 and 5.3.3.

5.2.5 Stability Conditions in special case of perfect surface normal estimation

In the special case of perfect estimation of surface normal ($\epsilon = 0$), the closed loop matrices of both control laws become identical:

$$\mathbf{M}_{p|\epsilon=0} = \mathbf{L}_{e_k} \mathbf{P} \widehat{\mathbf{L}}_{e_k}^- = \mathbf{L}_{e_k} \widehat{\mathbf{L}}_{e_k}^- = \mathbf{M}_{n|\epsilon=0} = \mathbf{A}_{n|\epsilon=0} \quad (5.34)$$

This leads to significant simplification in stability conditions for classical controller, with term $({}^p b_{ij} + {}^p b_{ji})$ disappearing from (5.28). Stability conditions are the same as in (5.32), with terms in (5.31) 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 5.3.3 where the stability conditions can be noticed to be the same for both controllers.

5.3 Simulation Results

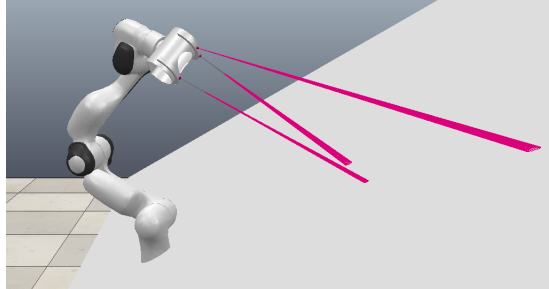


Figure 5.1 – Simulation setup in Coppeliassim.

Experimental results obtained in normal conditions, *i.e.*, with calibrated parameters, are described in Chapter 4. In this section we provide relevant simulation results considering calibration errors as well as errors on the surface normal estimation. This sort of analysis when done in a pre-operative step enables the user to determine the range of acceptable errors. Furthermore, we recall that the true parameters have to be known to evaluate the stability conditions while performing experimentation. This limitation makes simulation results most appropriate for such analysis. Simulation was done using FrankaSim [Oliva et al., 2022], a simulator based on ROS and Coppeliassim for Panda robot from Franka Emika. The simulator is integrated with ViSP [Marchand et al., 2005], which is used as the main software library for programming. Figure 5.1 displays the simulation

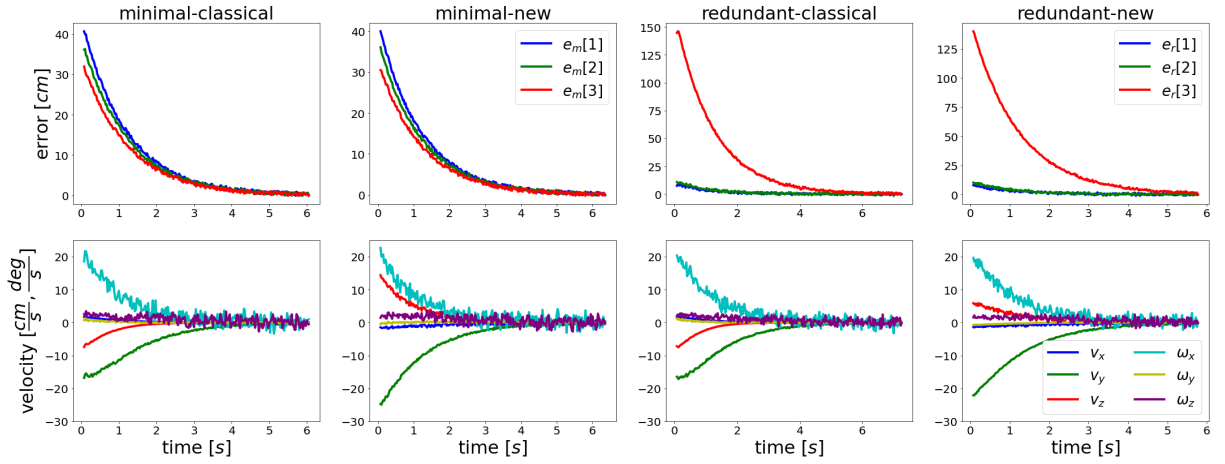


Figure 5.2 – Simulation results for Case I (Section 5.3.1) with minimal (left) and redundant (right) number of sensors using both classical and new control laws. Task function components (cm) versus time (s) on the top, end-effector velocities (cm/s and deg/s) versus time (s) in the bottom.

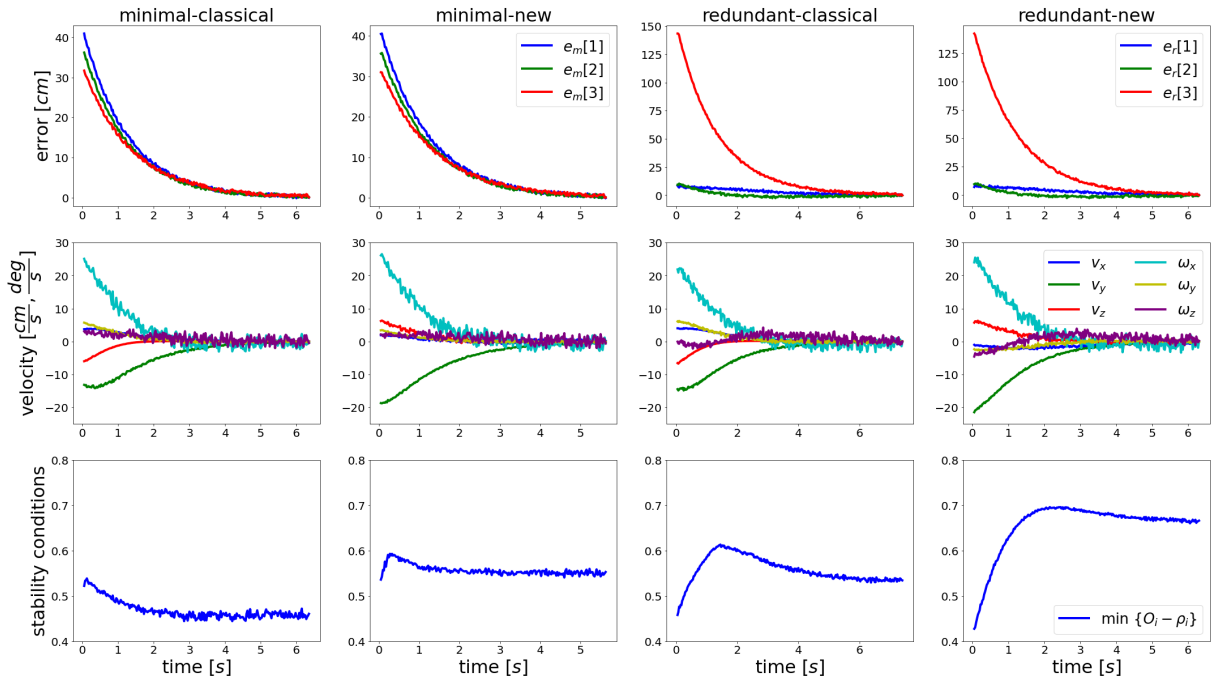


Figure 5.3 – Simulation results for Case II (Section 5.3.2). The bottom figures represents stability criterium $\min\{O_i - \rho_i\}$ versus time (s).

setup in Coppeliassim. 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. As for Chapter 4, 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 Figures 4.1 and 5.1). 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 5.1 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 frame F_E . As for the experimental results presented in Chapter 4, the initial pose is chosen such that the proximity signal readings are near to 50 cm . 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 for simulation results can be seen here.

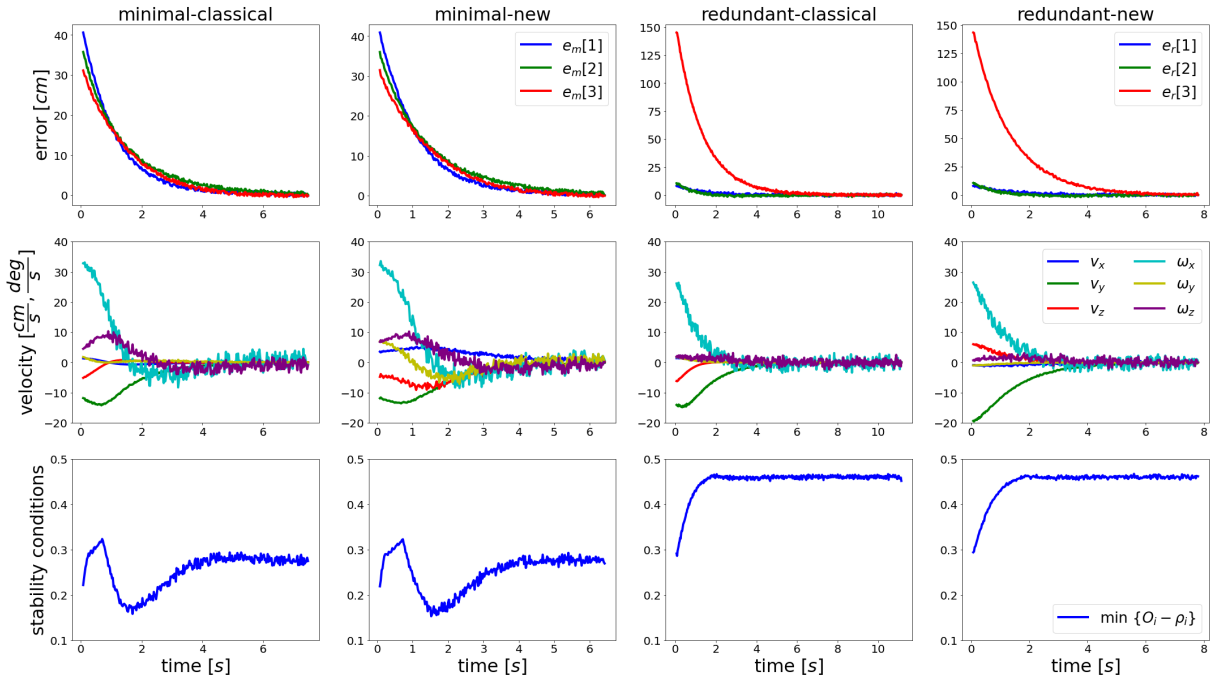


Figure 5.4 – Simulation results for Case III (Section 5.3.3)

\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 5.1 – Extrinsic parameters of proximity array.

5.3.1 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 Figure 5.2, 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.

5.3.2 Case II: Model with errors on all parameters

\mathbf{e}_m		\mathbf{e}_r	
$\widehat{\alpha}_1 = \alpha_1 + 10^\circ$	$\widehat{\alpha}_3 = \alpha_3 - 10^\circ$	$\widehat{\alpha}_1 = \alpha_1 + 10^\circ$	$\widehat{\alpha}_3 = \alpha_3 - 10^\circ$
$\alpha_2 = \alpha_2 + 10^\circ$		$\alpha_2 = \alpha_2 + 10^\circ$	$\alpha_4 = \alpha_4 - 10^\circ$
$\widehat{d}_i = 1.2 d_i, \widehat{r}_i = 1.2 r_i, (i = 1, 2), \theta = 10^\circ$			

Table 5.2 – Model parameters for Case II (Section 5.3.2)

In the second case, we consider a practical situation with errors in both extrinsic parameters and surface normal as shown in Table 5.2 where surface normal error is expressed as a rotational angle θ about the z -axis of F_E (see Figure 4.1). As seen in Figure 5.3, stability conditions are ensured for all control schemes, which shows their robustness. Once again, similar performances are obtained in all cases.

5.3.3 Case III: Model with only extrinsic parameter errors

\mathbf{e}_m		\mathbf{e}_r	
$\widehat{\alpha}_1 = \alpha_1 + 18^\circ$	$\widehat{\alpha}_3 = \alpha_3 - 18^\circ$	$\widehat{\alpha}_1 = \alpha_1 + 18^\circ$	$\widehat{\alpha}_3 = \alpha_3 - 18^\circ$
$\alpha_2 = \alpha_2 + 18^\circ$		$\alpha_2 = \alpha_2 + 18^\circ$	$\alpha_4 = \alpha_4 - 18^\circ$
$\widehat{d}_i = 1.2 d_i, \widehat{r}_i = 1.2 r_i, (i = 1, 2), \theta = 0^\circ$			

Table 5.3 – Model parameters for Case III (Section 5.3.3)

In the third case, we consider larger angular errors (18°) on each of the rings about its $\widehat{\mathbf{k}}$ 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 5.3. As surface normal estimation is perfect, we have identical closed loop matrices (5.34) for

both controllers and thereby resulting in the same task evolution as seen in Figure 5.4. Stability conditions also remain identical using both control laws and they are ensured all along the execution of the task. However, the end-effector trajectory (shown in video) 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.

5.3.4 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 Figure 5.5, 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 (5.14)). 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.

5.4 Conclusions

This chapter 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.

As future work, 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?

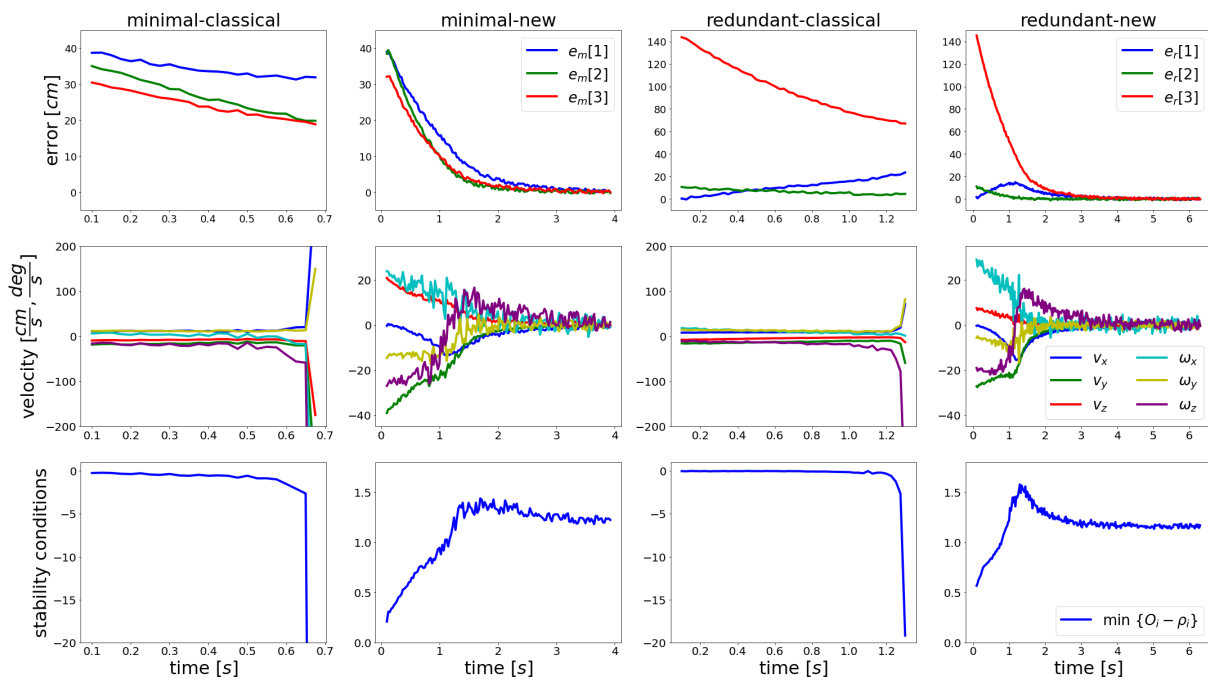


Figure 5.5 – Simulation results for Case IV (Section 5.3.4)

POSITIONING TASK WITH RESPECT TO A CYLINDER

6.1 Introduction

In this chapter, we discuss about using the proximity array attached as an end-effector to perform positioning task wrt. to a cylinder. At convergence the sensor system achieves a class C2 virtual linkage with the cylinder, thereby enabling it to perform translation and rotation about the axis of the cylinder without changing the proximity sensor value. Positioning task wrt. to the outside of the cylinder can be used for non-contact inspection of the exterior of pipes in chemical plants or in oil/gas industries. While positioning wrt. to the inside of a hollow cylinder could enable in performing guidance task inside a pipe for non-contact inspection. This chapter is divided into five sections. In Section 6.2 modelling aspects of the task are considered. In Section 6.3 we provide simulation results and select two interaction matrix models for executing the task. In Section 6.4 we provide experimental results while utilizing these two models to converge to the desired pose wrt. the cylinder and then perform a contactless surface following task. The chapter is concluded in Section 6.5.

6.2 Modelling

The modelling approach followed is the same as in Chapter 4. We consider both minimal and redundant number of sensors for performing the task and also provide practical approximations for the interaction matrix model. While positioning with respect to a cylinder using task functions composed of sensor features δ_{ij} , only four DoF of the proximity array can be controlled. This is because at any given relative pose of the proximity array with respect to the cylinder, when detection occurs, there are two independent motions that do not change the value of the corresponding sensor features. These motions

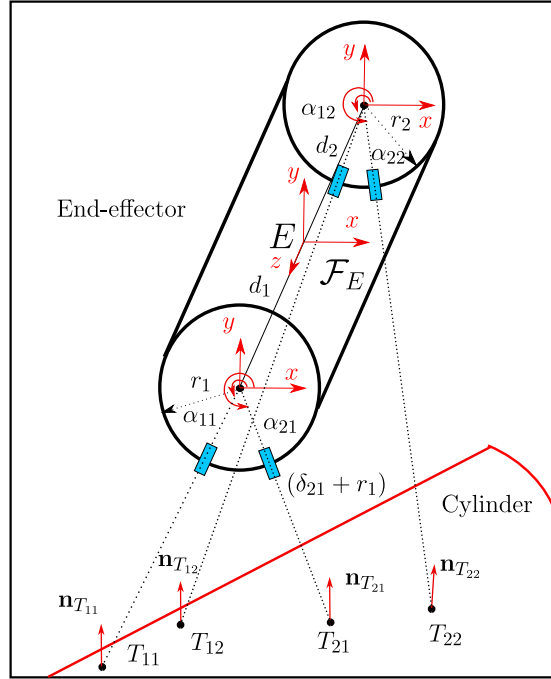


Figure 6.1 – Proximity Array with minimal sensors used for the task of positioning task wrt. cylinder

include a translation and a rotation about the axis of the cylinder. At final relative pose, a virtual sliding pivot linkage is formed between the proximity array and the cylinder. This virtual linkage is categorized as class 2 or C2 [Samson et al., 1991], where 2 refers to the dimension of the motions that belong to the null space of the corresponding interaction matrix.

6.2.1 Minimal sensors

Minimal set of sensors required to perform the task is four as considered in Figure 6.1 with two sensors each on two rings. The task function, denoted \mathbf{e}_{1cy} , for the case of minimal sensors is built based on the current proximity signal values δ_{ij} and their desired value δ_{ij}^* ,

$$\mathbf{e}_{1cy} = \mathbf{s} - \mathbf{s}^* \quad (6.1)$$

where $\mathbf{s} = \begin{pmatrix} \delta_{11} \\ \delta_{21} \\ \delta_{12} \\ \delta_{22} \end{pmatrix}$ and $\mathbf{s}^* = \begin{pmatrix} \delta_{11}^* \\ \delta_{21}^* \\ \delta_{12}^* \\ \delta_{22}^* \end{pmatrix}$. Interaction matrix corresponding to the above task function is built based on the interaction screw. Let us remind the interaction screw value for a generic sensor,

$$\begin{aligned} \mathbf{u}_{ij} &= -\frac{\mathbf{n}_{T_{ij}}}{\mathbf{n}_{T_{ij}} \cdot \mathbf{n}_{S_{ij}}} \\ \mathbf{H}_{ij}(E) &= \mathbf{m}_{T_{ij}} \times \mathbf{u}_{ij} \end{aligned} \quad (6.2)$$

with parameters

$$\begin{aligned} \mathbf{n}_{S_{ij}} &= \cos \alpha_{ij} \hat{\mathbf{i}} + \sin \alpha_{ij} \hat{\mathbf{j}} \\ \mathbf{m}_{T_{ij}} &= (\delta_{ij} + r_j) \mathbf{n}_{S_{ij}} + d_j \hat{\mathbf{k}} \end{aligned} \quad (6.3)$$

The interaction matrix $\mathbf{L}_{\mathbf{e}_{1cy}}$ for this task is directly obtained from (3.19) and (6.2),

$$\mathbf{L}_{\mathbf{e}_{1cy}} = \begin{bmatrix} \mathbf{u}_{11}^T & \mathbf{H}_{11}(E)^T \\ \mathbf{u}_{21}^T & \mathbf{H}_{21}(E)^T \\ \mathbf{u}_{12}^T & \mathbf{H}_{12}(E)^T \\ \mathbf{u}_{22}^T & \mathbf{H}_{22}(E)^T \end{bmatrix} \quad (6.4)$$

In the next subsection we consider a simplified model of the above interaction matrix.

6.2.1.1 Simplified Model

One way to consider a simplified model is by making an approximation as described in Section 3.1.1,

$$\widehat{\mathbf{n}}_{T_{ij}} = -\mathbf{n}_{S_{ij}} \quad (6.5)$$

This assumption is more valid for positioning inside a cylinder if the task is designed to align with the axis of the cylinder. Indeed, at the equilibrium position, the target normal align with the negative direction of the sensor axes due to the geometry of the cylinder.

The above assumption simplifies the interaction screw of proximity sensor to,

$$\begin{aligned}\mathbf{u}_{ij} &= -\cos \alpha_{ij} \hat{\mathbf{i}} - \sin \alpha_{ij} \hat{\mathbf{j}} \\ \mathbf{H}_{ij}(E) &= d_j \sin \alpha_{ij} \hat{\mathbf{i}} - d_j \cos \alpha_{ij} \hat{\mathbf{j}}\end{aligned}\quad (6.6)$$

where d_j is the offset distance of the ring from the end-effector point along \hat{k} direction as seen in Figure 6.1 leading to,

$$\hat{\mathbf{L}}_{\mathbf{e}_{1cy}} = \begin{bmatrix} -\cos \alpha_{11} & -\sin \alpha_{11} & 0 & d_1 \sin \alpha_{11} & -d_1 \cos \alpha_{11} & 0 \\ -\cos \alpha_{21} & -\sin \alpha_{21} & 0 & d_1 \sin \alpha_{21} & -d_1 \cos \alpha_{21} & 0 \\ -\cos \alpha_{12} & -\sin \alpha_{12} & 0 & d_2 \sin \alpha_{12} & -d_2 \cos \alpha_{12} & 0 \\ -\cos \alpha_{22} & -\sin \alpha_{22} & 0 & d_2 \sin \alpha_{22} & -d_2 \cos \alpha_{12} & 0 \end{bmatrix}\quad (6.7)$$

In practice, we consider a symmetrical sensor arrangement in proximity array with the following extrinsic parameters,

$$\begin{aligned}\alpha_{11} &= 270^\circ - \beta, & \alpha_{21} &= 270^\circ + \beta, & d_1 &= d, & r_1 &= r \\ \alpha_{12} &= 270^\circ - \beta, & \alpha_{22} &= 270^\circ + \beta, & d_2 &= -d, & r_2 &= r\end{aligned}\quad (6.8)$$

Here parameter d represents the location of the sensor ring along the z-axis of proximity array system. In later analysis we perform the task using two different values of d . The model parameters of the proximity array system for minimal case of executing the task therefore reduces to (β, d) and the interaction matrix simplifies to

$$\hat{\mathbf{L}}_{\mathbf{e}_{1cy}} = \begin{bmatrix} \sin \beta & \cos \beta & 0 & -d \cos \beta & d \sin \beta & 0 \\ -\sin \beta & \cos \beta & 0 & -d \cos \beta & -d \sin \beta & 0 \\ \sin \beta & \cos \beta & 0 & d \cos \beta & -d \sin \beta & 0 \\ -\sin \beta & \cos \beta & 0 & d \cos \beta & d \sin \beta & 0 \end{bmatrix}\quad (6.9)$$

From the above interaction matrix it is clear that its nullspace corresponds to pure translation and rotation along the z-axis of \mathcal{F}_E , which, due to the approximation done, corresponds only to the expected one when the array is aligned on the cylinder axis. This is why we will use this model only for positioning inside a cylinder.

Another approximated model is obtained by evaluating the interaction matrix for the desired configuration. This approach is classical in visual servoing [Chaumette et al., 2016] and was already followed for plane-to-plane positioning as described in Section 4.2.1.

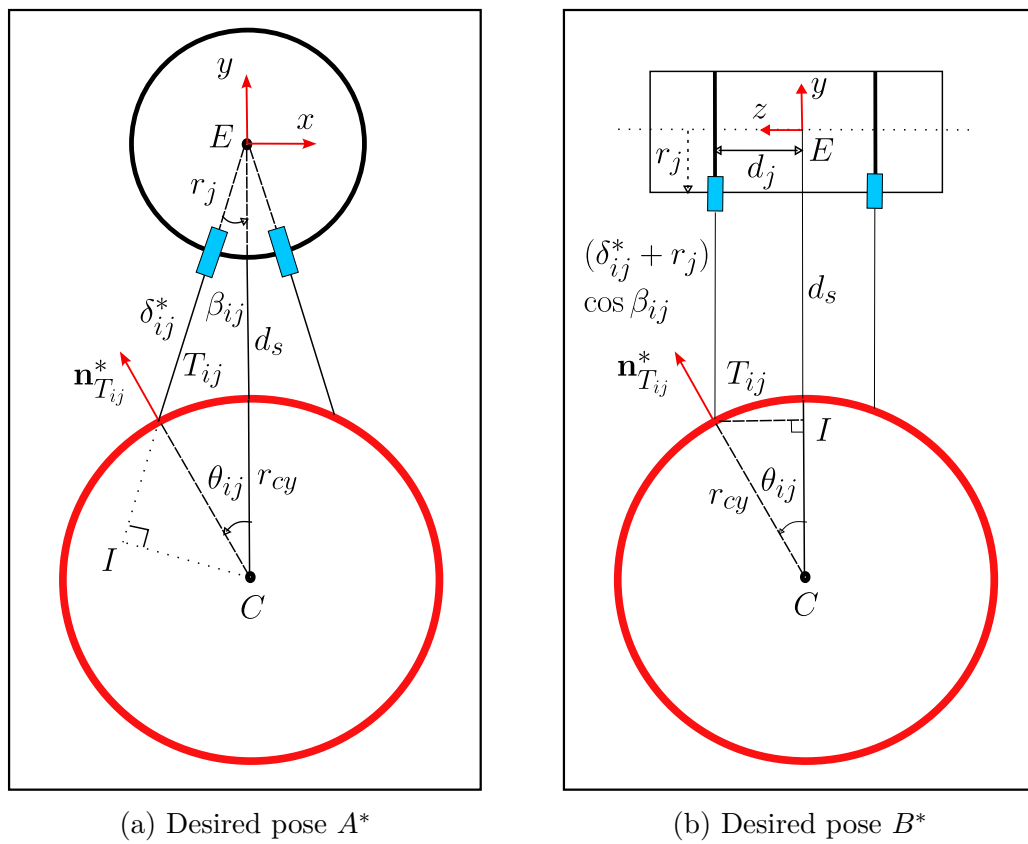


Figure 6.2 – Special Cases while positioning outside the cylinder

Here we consider the desired configuration in such a way that either z -axis or x -axis of \mathcal{F}_E aligns with the axis of the cylinder. These two configurations are respectively considered as desired poses A^* and B^* as indicated in Figure 6.2.

6.2.1.2 Desired Pose A^*

Let us consider the case when z -axis of frame \mathcal{F}_E in Figure 6.1 is parallel to the axis of cylinder. Front view of the configuration is considered in Figure 6.2a. Let us consider d_s as the distance measured along the negative y -axis in \mathcal{F}_E till the surface of cylinder with r_{cy} as its radius. It is to be noted that r_{cy} is an unknown parameter here and the interaction matrix is evaluated based on the measurement or suitable approximation \hat{r}_{cy} of this value. At desired pose A^* , the axis of the cylinder and the z -axis of frame \mathcal{F}_E are separated by a distance of $(d_s + \hat{r}_{cy})$. Desired value of proximity sensors δ_{ij}^* can then be evaluated from the two right angled triangles ΔEIC and $\Delta T_{ij}IC$,

$$\begin{aligned} \delta_{ij}^* + r_j &= |\vec{EI}| - |T_{ij}\vec{I}| \\ &= (d_s + \hat{r}_{cy}) \cos \beta_{ij} - \sqrt{r_{cy}^2 - (d_s + \hat{r}_{cy})^2 \sin^2 \beta_{ij}} \\ &= \tilde{\delta}_{ij}^* \end{aligned} \quad (6.10)$$

The desired value of target normal is (see Figure 6.2a)

$$\mathbf{n}_{T_{ij}}^* = -\sin \theta_{ij} \hat{\mathbf{i}} + \cos \theta_{ij} \hat{\mathbf{j}} \quad (6.11)$$

where $\sin \theta_{ij} = \frac{\tilde{\delta}_{ij}^* \sin \beta_{ij}}{\hat{r}_{cy}}$, $\cos \theta_{ij} = \sqrt{1 - \sin^2 \theta_{ij}}$ and $\beta_{ij} = 270^\circ - \alpha_{ij}$. The desired value of the interaction screw is thus

$$\begin{aligned} \mathbf{u}_{ij}^* &= \frac{1}{\cos \theta_{ij} + \beta_{ij}} \left(-\sin \theta_{ij} \hat{\mathbf{i}} + \cos \theta_{ij} \hat{\mathbf{j}} \right) \\ \mathbf{H}_{ij}(E)^* &= \mathbf{m}_{T_{ij}}^* \times \mathbf{u}_{ij}^* \end{aligned} \quad (6.12)$$

where $\mathbf{m}_{T_{ij}}^* = \tilde{\delta}_{ij}^* \mathbf{n}_{S_{ij}} + d_j \hat{\mathbf{k}}$. The above model is used in the experiments presented in Section 6.4 with extrinsic parameters mentioned in (6.23). For the simulation results presented in Section 6.3, we consider a symmetrical configuration with extrinsic parameters given in Table 4.1 for which $\beta_{1j} = -\beta_{2j} = \beta$ and $\theta_{1j} = -\theta_{2j} = \theta$. By denoting $\gamma = \theta + \beta$,

the interaction matrix in this case corresponds to

$$\hat{\mathbf{L}}_{\mathbf{e}_{1cy}}^{A*} = \frac{1}{\cos \gamma} \begin{bmatrix} -\sin \theta & \cos \theta & 0 & -d \cos \theta & -d \sin \theta & -\tilde{\delta}^* \sin \gamma \\ \sin \theta & \cos \theta & 0 & -d \cos \theta & d \sin \theta & \tilde{\delta}^* \sin \gamma \\ -\sin \theta & \cos \theta & 0 & d \cos \theta & d \sin \theta & -\tilde{\delta}^* \sin \gamma \\ \sin \theta & \cos \theta & 0 & d \cos \theta & -d \sin \theta & \tilde{\delta}^* \sin \gamma \end{bmatrix} \quad (6.13)$$

When proximity array is at equilibrium we can apply an angular velocity $\omega^* \hat{\mathbf{k}}$ about the axis of cylinder passing through point C and a linear velocity $v^* \hat{\mathbf{k}}$ along the axis of cylinder. These twists when transferred to end-effector point E can be represented as $\mathbf{v}_1 = \left(-(d_s + \hat{r}_{cy})\omega^* \quad 0 \quad 0 \quad 0 \quad 0 \quad \omega^* \right)^T$ and $\mathbf{v}_2 = \left(0 \quad 0 \quad v^* \quad 0 \quad 0 \quad 0 \right)^T$. These twists also represent the null space of interaction matrix, i.e. $\mathbf{v}_1, \mathbf{v}_2 \in \text{Ker } \mathbf{L}_{\mathbf{e}_{1cy}}^{A*}$ therefore do not cause any change to task \mathbf{e}_{1cy} at equilibrium. \mathbf{v}_1 can be verified to be the nullspace by applying law of sines to $\triangle ET_{ij}C$, which gives us the condition $\frac{\sin \pi - (\theta + \beta)}{(d_s + \hat{r}_{cy})} = \frac{\sin \theta}{\tilde{\delta}^*}$. This condition simplifies into $(d_s + \hat{r}_{cy}) \sin \gamma - \tilde{\delta}^* \sin \delta = 0$.

6.2.1.3 Desired Pose B^*

While using the proximity array it is also interesting to consider the desired final configuration where x -axis of frame \mathcal{F}_E is aligned with the axis of cylinder. This configuration is depicted in Fig. 6.2b. The desired value of proximity sensors δ_{ij}^* can be evaluated by considering the length of line segment $|\vec{EC}|$.

$$\tilde{\delta}_{ij}^* = \frac{1}{\cos \beta_{ij}} (d_s + \hat{r}_{cy}(1 - \cos \theta_{ij})) \quad (6.14)$$

where $\theta_{ij} = \arcsin(d/\hat{r}_{cy})$ is obtained by considering triangle $\triangle CT_{ij}I$. The desired value of target normal is

$$\mathbf{n}_{T_{ij}}^* = \cos \theta_{ij} \hat{\mathbf{j}} + \sin \theta_{ij} \hat{\mathbf{k}} \quad (6.15)$$

The desired value of interaction screw is now

$$\begin{aligned} \mathbf{u}_{ij}^* &= \frac{1}{\cos \theta_{ij} \cos \beta_{ij}} (\cos \theta_{ij} \hat{\mathbf{j}} + \sin \theta_{ij} \hat{\mathbf{k}}) \\ \mathbf{H}_{ij}(E)^* &= \mathbf{m}_{T_{ij}}^* \times \mathbf{u}_{ij}^* \end{aligned} \quad (6.16)$$

This model is used in experiment considered in Section 6.4 with extrinsic parameters given in (6.23). For the simulation results presented in Section 6.3 we utilize the same

symmetrical configuration as before for which $\beta_{1j} = -\beta_{2j} = \beta$ and $\theta_{1j} = -\theta_{2j} = \theta$. The interaction matrix in this case corresponds to

$$\mathbf{L}_{\mathbf{e}_{1cy}^*}^B = \frac{1}{\cos \theta \cos \beta} \begin{bmatrix} 0 & \cos \theta & \sin \theta & -d \cos \theta - \tilde{\delta}^* \sin \theta \cos \beta & \tilde{\delta}^* \sin \theta \sin \beta & -\tilde{\delta}^* \sin \beta \cos \theta \\ 0 & \cos \theta & \sin \theta & -d \cos \theta - \tilde{\delta}^* \sin \theta \cos \beta & -\tilde{\delta}^* \sin \theta \sin \beta & \tilde{\delta}^* \sin \beta \cos \theta \\ 0 & \cos \theta & -\sin \theta & d \cos \theta + \tilde{\delta}^* \sin \theta \cos \beta & -\tilde{\delta}^* \sin \theta \sin \beta & -\tilde{\delta}^* \sin \beta \cos \theta \\ 0 & \cos \theta & -\sin \theta & d \cos \theta + \tilde{\delta}^* \sin \theta \cos \beta & \tilde{\delta}^* \sin \theta \sin \beta & \tilde{\delta}^* \sin \beta \cos \theta \end{bmatrix} \quad (6.17)$$

At equilibrium, we can apply an angular velocity $\omega^* \hat{\mathbf{i}}$ about the axis of cylinder passing through point C and a linear velocity $v^* \hat{\mathbf{i}}$ along the axis of cylinder. The spatial velocity vector applied at \mathcal{F}_E can be represented as $\mathbf{v}_1 = (0 \ 0 \ (d_s + r_{cy})\omega^* \ \omega^* \ 0 \ 0)^T$ and $\mathbf{v}_2 = (v^* \ 0 \ 0 \ 0 \ 0 \ 0)^T$. Like in Section 6.2.1.2, it can be noticed that $\mathbf{v}_1, \mathbf{v}_2 \in \text{Ker } \hat{\mathbf{L}}_{\mathbf{e}_{1cy}}^{B*}$. To verify \mathbf{v}_2 as a component of nullspace of the interaction matrix, we can evaluate from $\Delta CT_{ij}I$, $\tan \theta = \frac{d}{(d_s + \hat{r}_{cy}) - \tilde{\delta}^* \cos \beta}$. This can be rearranged to obtain the following condition $(d_s + \hat{r}_{cy}) \sin \theta - d \cos \theta - \tilde{\delta}^* \cos \beta \sin \theta = 0$.

6.2.1.4 Trivial Singularity

For the case of minimal sensors, there is a situation where the interaction matrix would be close to singularity. Imagine all four detection points are within a small neighbourhood at a given point in execution for the task. In that case, the target normal could be approximated as $\widehat{\mathbf{n}}_{T_{ij}} \approx \mathbf{n}_T$ where locally there would be an attempt to perform plane-to-plane positioning task using four sensor features. The interaction matrix $\mathbf{L}_{\mathbf{e}_{1cy}}$ can be locally approximated as,

$$\mathbf{L}_{\mathbf{e}_{1cy}}|_{\mathbf{n}_{T_{ij}}=\mathbf{n}_T} = \begin{bmatrix} \cdot & \cdot \\ \beta_i \mathbf{n}_T^T \beta_i (\mathbf{m}_{T_{ij}} \times \mathbf{n}_T)^T \\ \cdot & \cdot \end{bmatrix}_{4 \times 6} \quad (6.18)$$

where $\beta_i = -\frac{1}{\mathbf{n}_T \cdot \mathbf{n}_{S_{ij}}}$. The interaction screws would become linearly dependent and the interaction matrix would experience a loss of rank. In Section 10.4, we evaluate the dual basis for the interaction screws involved in the task considered. It is evident from term l_i which approaches 0 value when $\widehat{\mathbf{n}}_{T_{ij}} \approx \mathbf{n}_T$. Since the term is present in the denominator this would lead to infinite value in control input and hence is nearing a singularity. For a practical task it is possible to be close to such a situation as explained later in Section 6.3.1.

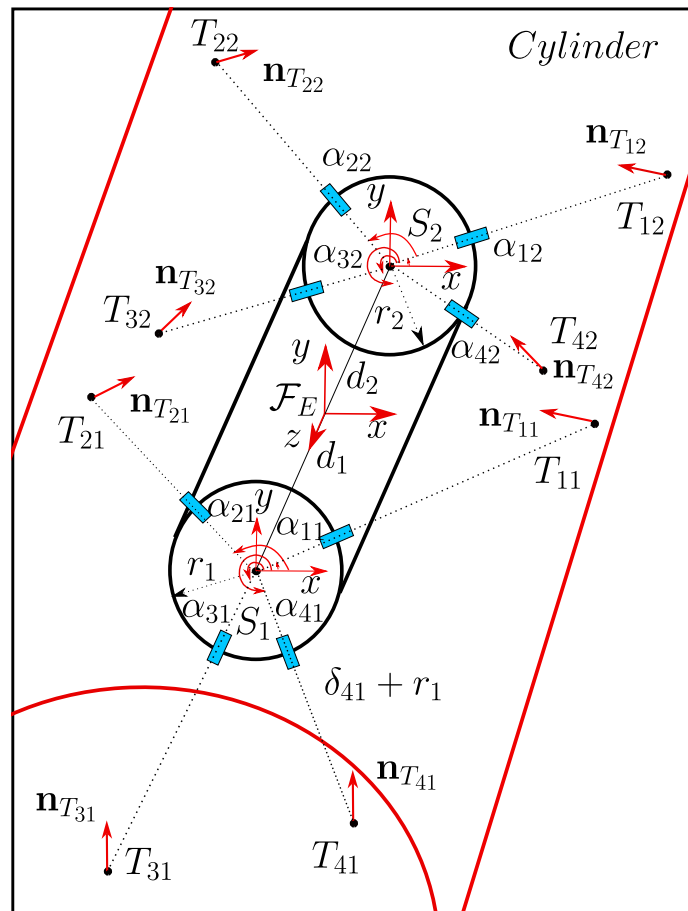


Figure 6.3 – Proximity Array with redundant sensors used for the task of positioning task wrt. cylinder

6.2.2 Redundant Sensors

Let us consider a sensor configuration consisting of two rings with each containing four proximity sensors from each other as shown in Figure 6.3. One possible choice of task function is,

$$\mathbf{e}_{2cy} = \mathbf{C}(\mathbf{s} - \mathbf{s}^*) \quad (6.19)$$

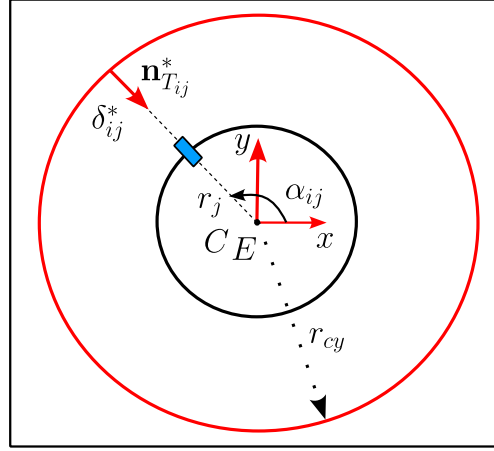


Figure 6.4 – Special Case while positioning inside a cylinder

$$\text{where } \mathbf{C} = \begin{bmatrix} 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 \\ 0 & -1 & 0 & 1 & 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 \end{bmatrix}, \mathbf{s} = \begin{pmatrix} \delta_{11} \\ \delta_{21} \\ \delta_{31} \\ \delta_{41} \\ \delta_{12} \\ \delta_{22} \\ \delta_{32} \\ \delta_{42} \end{pmatrix} \text{ and } \mathbf{s}^* = \begin{pmatrix} \delta_{11}^* \\ \delta_{21}^* \\ \delta_{31}^* \\ \delta_{41}^* \\ \delta_{12}^* \\ \delta_{22}^* \\ \delta_{32}^* \\ \delta_{42}^* \end{pmatrix}$$

\mathbf{C} is the combination matrix that is chosen to obtain a task of dimension 4 and $\mathbf{C} \mathbf{s}^*$ becomes its associated reference vector. The interaction matrix $\mathbf{L}_{\mathbf{e}_{2cy}}$ for this task is represented using the combination matrix from (3.19) and (6.2),

$$\mathbf{L}_{\mathbf{e}_{2cy}} = \mathbf{C} \begin{bmatrix} \mathbf{u}_{11}^T & \mathbf{H}_{11}^T \\ \mathbf{u}_{21}^T & \mathbf{H}_{21}^T \\ \mathbf{u}_{31}^T & \mathbf{H}_{31}^T \\ \mathbf{u}_{41}^T & \mathbf{H}_{41}^T \\ \mathbf{u}_{12}^T & \mathbf{H}_{12}^T \\ \mathbf{u}_{22}^T & \mathbf{H}_{22}^T \\ \mathbf{u}_{32}^T & \mathbf{H}_{32}^T \\ \mathbf{u}_{42}^T & \mathbf{H}_{42}^T \end{bmatrix} \quad (6.20)$$

From a practical point of view such a task can be used to align with the axis of the cylinder. In the next subsection we provide a simplified model of interaction matrix and

a symmetrical arrangement of sensors that leads to decoupling in interaction matrix.

6.2.2.1 Special Case

In this case we consider a final configuration when proximity array is positioned inside the cylinder such that its z -axis coincides with the axis of the cylinder. Figure 6.4 depicts this scenario where the target normal aligns with the negative direction of the sensor axis at final pose, which is also the assumption for simplified model considered in Section 6.2.1.1. The desired sensor feature in this case can be evaluated as

$$\delta_{ij}^* = r_{cy} - r_j \quad (6.21)$$

This simplifies the reference vector as $\mathbf{C} \mathbf{s}^* = \mathbf{0}$ resulting in simpler task function considered in [Espiau, 1990],

$$\mathbf{e}_{2cy} = \begin{pmatrix} (\delta_{11} + \delta_{12}) - (\delta_{31} + \delta_{32}) \\ (\delta_{21} + \delta_{22}) - (\delta_{41} + \delta_{42}) \\ (\delta_{41} + \delta_{22}) - (\delta_{21} + \delta_{42}) \\ (\delta_{11} + \delta_{32}) - (\delta_{31} + \delta_{12}) \end{pmatrix} \quad (6.22)$$

For the sensor arrangement considered in [Espiau, 1990], we have extrinsic values in angles as,

$$\begin{aligned} \alpha_{11} = 0^\circ, \quad \alpha_{21} = 90^\circ, \quad \alpha_{31} = 180^\circ, \quad \alpha_{41} = 270^\circ, \quad d_1 = d, \quad r_1 = r \\ \alpha_{12} = 0^\circ, \quad \alpha_{22} = 90^\circ, \quad \alpha_{32} = 180^\circ, \quad \alpha_{42} = 270^\circ, \quad d_2 = -d, \quad r_2 = r \end{aligned} \quad (6.23)$$

Using the above values we can obtain a simplified version of the interaction matrix that is decoupled,

$$\mathbf{L}_{\mathbf{e}_{2cy}}^* = \begin{bmatrix} -4 & 0 & 0 & 0 & 0 & 0 \\ 0 & -4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -4d & 0 & 0 \\ 0 & 0 & 0 & 0 & -4d & 0 \end{bmatrix} \quad (6.24)$$

The standard control law for achieving both tasks is,

$$\mathbf{v}_{E,kcy} = -\lambda \hat{\mathbf{L}}_{\mathbf{e}_{kcy}}^+ \mathbf{e}_{kcy}, \quad k = 1, 2 \quad (6.25)$$

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

6.2.3 Stability Analysis

As discussed in Section 4.2.3, uniqueness of equilibrium point of the task exists in case of minimal sensors (6.1),

$$\mathbf{e}_{1cy} = 0 \quad \iff \quad \mathbf{s} = \mathbf{s}^* \quad (6.26)$$

However for the case of redundant number of sensors there could be a situation for violating condition (6.26) when,

$$\mathbf{s} - \mathbf{s}^* \in \text{Ker } \mathbf{C} \quad (6.27)$$

An analysis similar to Appendix A needs to be done as part of future work to understand if any physically consistent configuration satisfying (6.27) exist.

The system is globally asymptotically stable (GAS) in non-singular locations for task \mathbf{e}_{1cy} if the following property is ensured [Chaumette & Hutchinson, 2006],

$$\mathbf{L}_{\mathbf{e}_{1cy}} \widehat{\mathbf{L}}_{\mathbf{e}_{1cy}}^+ > 0 \quad (6.28)$$

Here, $\mathbf{L}_{\mathbf{e}_{1cy}} \widehat{\mathbf{L}}_{\mathbf{e}_{1cy}}^+$ is a 4x4 matrix and we have $\mathbf{L}_{\mathbf{e}_{1cy}} \widehat{\mathbf{L}}_{\mathbf{e}_{1cy}}^+ = \mathbf{I}_4 > 0$ when $\widehat{\mathbf{L}}_{\mathbf{e}_{1cy}} = \mathbf{L}_{\mathbf{e}_{1cy}}$, 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 (6.4) if the target normal is correctly evaluated and if no calibration errors occur. On the other hand, when using either simplified or approximate models (6.7, 6.13, 6.17) for task \mathbf{e}_{1cy} or while using both actual (6.20) and simplified models (6.24) for task \mathbf{e}_{2cy} in the control scheme, we can only state that the system is locally asymptotically stable since $\mathbf{L}_{\mathbf{e}_{kcy}} \mathbf{L}_{\mathbf{e}_{kcy}}^{*+} = \mathbf{I}_4$ only at the desired configuration $\delta_{ij} = \delta_{ij}^*$. It can be seen from the simulation results that the simplified model can become unstable for particular initial poses of the proximity array. However, it can also be observed from the simulation results for task function \mathbf{e}_{1cy} that the domain of convergence is large while using interaction matrix models of desired pose A^* and B^* . For task \mathbf{e}_{2cy} we show simulation result suggesting stable behavior for practical task.

6.3 Simulation Results

In this section we consider simulation results for positioning outside and inside of a cylinder. We consider a simulation setup composed of a Franka Emika Panda robot (7 Dof serial manipulator) in CoppeliaSim with proximity array involving minimal and redundant sensors as depicted in Figures 6.5 and 6.7 involving both solid and hollow cylinder. The simulation results are described in two separate cases. In Case 1 we consider minimal set of sensors for performing positioning on the outside of the cylinder with extrinsic parameters from (6.8). For Case 1, positioning task is performed from two initial poses with two different extrinsic parameters denoted as Array I and II as shown in Table 6.1. In Case 2, while positioning on the inside of the cylinder we consider redundant number of sensors with extrinsic parameters mentioned in (6.23). 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 of the control scheme (6.25) is set to $\lambda = 0.8$. The video for simulation results can be seen here.

Array I	Array II
$d = 5.5\text{ cm}$	$d = 16.5\text{ cm}$
$\alpha = 20^\circ, r = 7\text{ cm}$	

Table 6.1 – Extrinsic parameters of proximity array used for positioning on the outside of the cylinder.

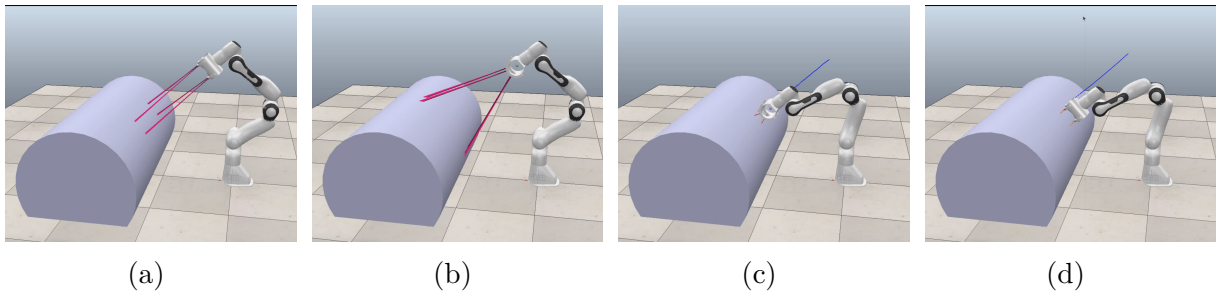


Figure 6.5 – Positioning task with Array I (smaller d): (a) initial pose A, (b) initial pose B, (c) final pose when using desired pose A^* , (d) final pose when using desired pose B^* .

6.3.1 Case 1 : Positioning outside the cylinder

In Figure 6.5, we consider the extrinsic parameter indicating the distance of a half ring from end-effector point E along z -axis as $d = 5.5\text{ cm}$ and in Figure 6.6 we consider

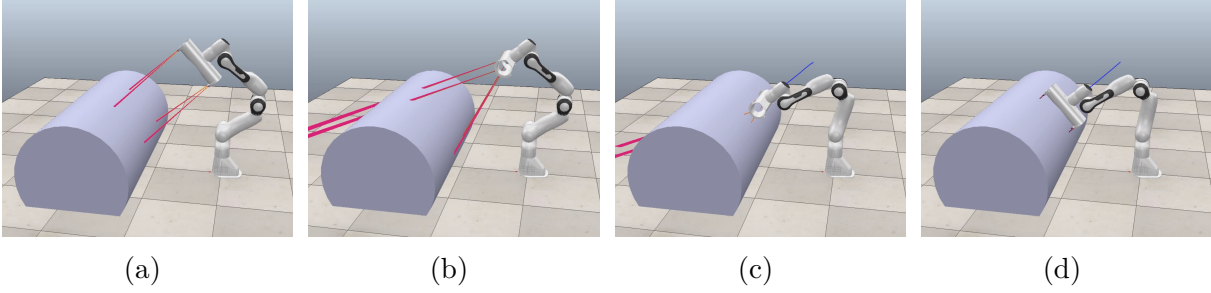
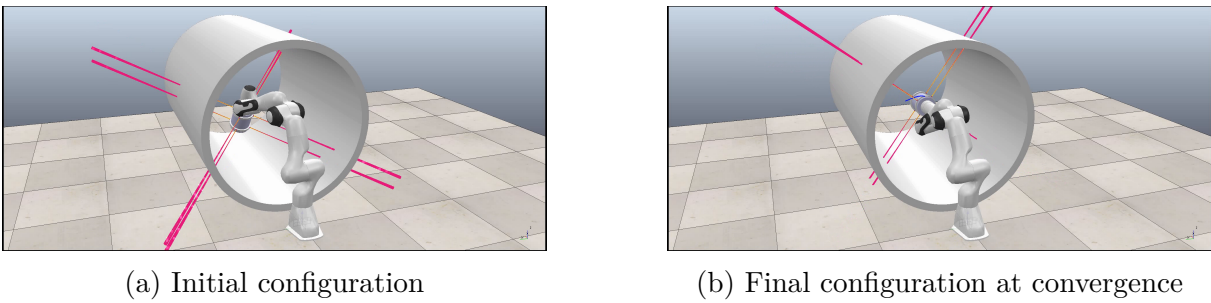

 Figure 6.6 – Positioning task with Array II (larger d): same configuration as for Fig. 6.5


Figure 6.7 – Positioning task inside the cylinder

it as $d = 16.5 \text{ cm}$. For each extrinsic parameter d we consider two initial poses A and B. Initial pose A is characterized by x-axis of \mathcal{F}_E being parallel to the cylinder axis as shown in Figure 6.5a and Figure 6.6a. Initial pose B is characterized by the z-axis of \mathcal{F}_E being parallel to the axis of cylinder as shown in Figure 6.5b and Figure 6.6b. Each of these initial poses are also chosen such that the proximity signal readings are approximately around 50 cm from target. We execute the task for each value of d using four interaction matrix models consisting of actual, simplified, desired pose A^* , and desired pose B^* . We show convergence to a configuration where the desired proximity measurements are $\delta_{ij}^* = 4 \text{ cm}$. While using desired pose A^* for interaction matrix a final configuration as shown in Figure 6.5c and Figure 6.6c is reached. For tasks executed using desired pose B^* the final configuration achieved is shown in Figure 6.5d and Figure 6.6d.

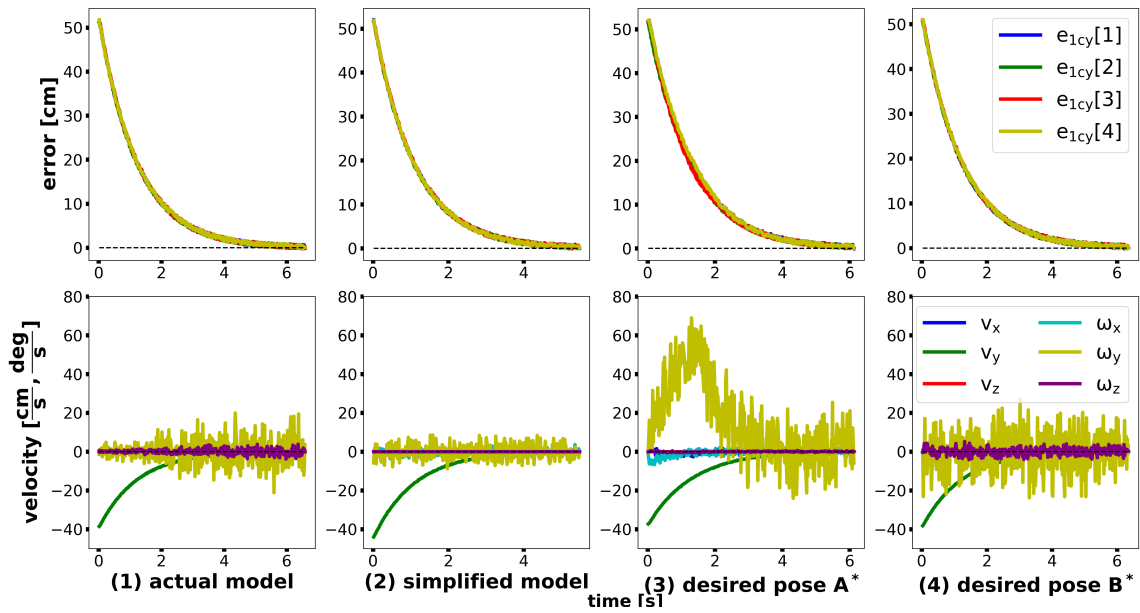
Figures 6.8 and 6.9 displays the plots of task function \mathbf{e}_{1cy} and end-effector velocity for both Array I and II from initial poses A and B respectively. Task considered converges using the control scheme in all four cases for both values of d when executed from initial pose A. For results involving initial pose B, we observe that the tasks do not converge while using the simplified model (6.9) with both values of d . This makes the simplified model a less reliable choice for the task considered. In plots involving smaller value of

d , we observe the influence of noise in proximity sensor measurements, especially for angular velocity term ω_y while using models such as actual, desired pose A^* and desired pose B^* . As observed in Figures 6.8a(1) and 6.9a(1) where the actual model is used, the influence of noise on ω_y increases close to convergence. Whereas for model evaluated from desired pose A^* (Figures 6.8a(3), 6.9a(3)) or B^* (Figures 6.8a(4), 6.9a(4)), it is more prevalent throughout the entire execution of task as the model is evaluated at equilibrium. Interaction screws are more closer to linear dependence in these situations as their target normal values are quite close. This is related to the trivial singularity mentioned in Section 6.2.1.4.

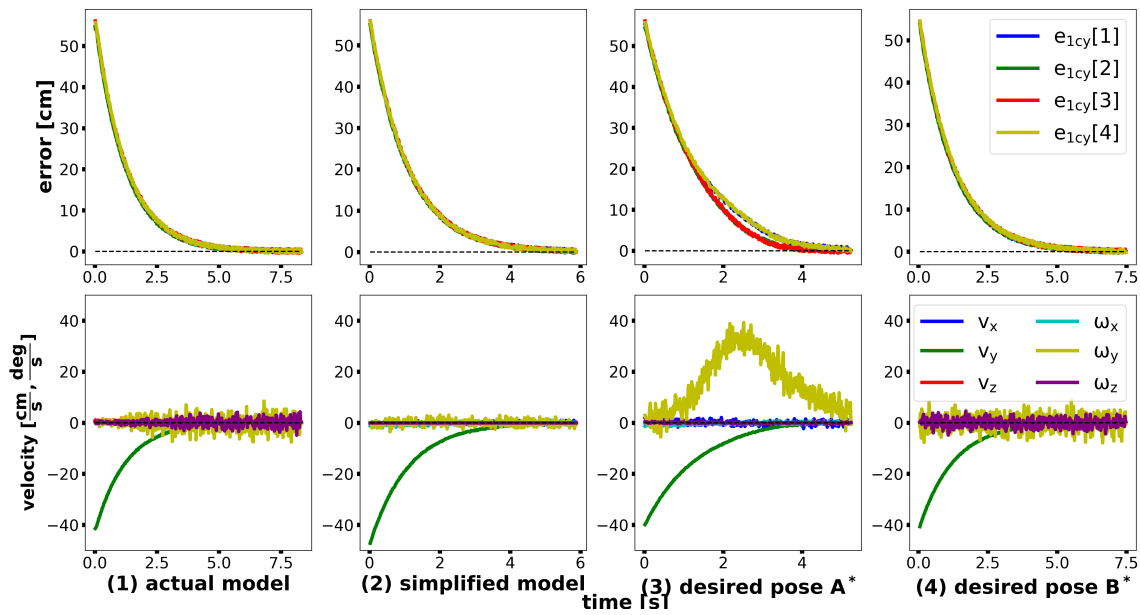
While positioning using Array II with larger value of d the influence of noise on end-effector velocity significantly reduces. This can be observed in Figures (6.8b(1), 6.9b(1)) where actual model is used, in Figures (6.8b(3), 6.9b(3)) where model evaluated at desired pose A^* and Figures (6.8b(4), 6.9b(4)) where model evaluated at desired pose B^* is used. While using actual model for the tasks considered, the final relative pose achieved by the end-effector depends on the initial pose. However while using models evaluated from desired pose A^* and B^* the final relative pose achieved is the same as equilibrium pose corresponding to the desired proximity signal values on which interaction matrix is evaluated. It is therefore more effective to use model evaluated at desired pose A^* and B^* using Array II with larger value of d .

6.3.2 Case 2 : Positioning inside the cylinder to align with its axis.

In this final case, we consider the task \mathbf{e}_{2cy} for positioning in the inside of the cylinder from an arbitrary initial pose. Task converges effectively as seen in Figure 6.10. The choice of task function and simplification in special case (6.24) leads to a decoupled interaction matrix. This leads to exponentially decoupled closed-loop behavior close to equilibrium, with strong correlation between components of task function and spatial velocity vector by comparing the behaviours in the following pairs, $(\mathbf{e}_{2cy}[1], v_x)$, $(\mathbf{e}_{2cy}[2], v_y)$, $(\mathbf{e}_{2cy}[3], \omega_x)$ and $(\mathbf{e}_{2cy}[4], \omega_y)$ as observed in Figure 6.10.

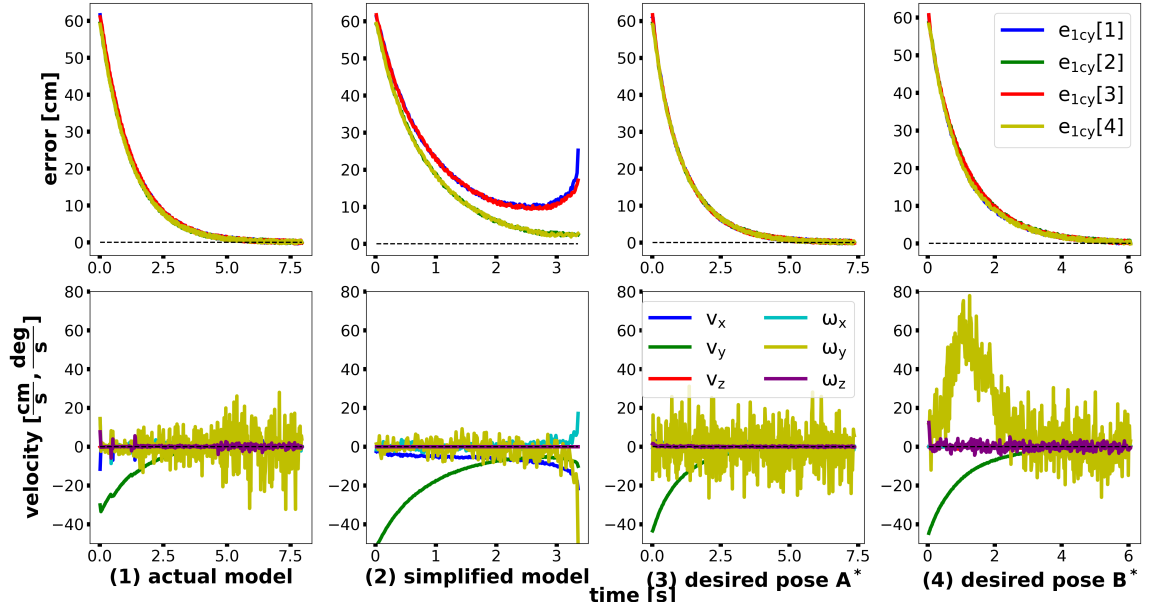


(a) Smaller d - Array I

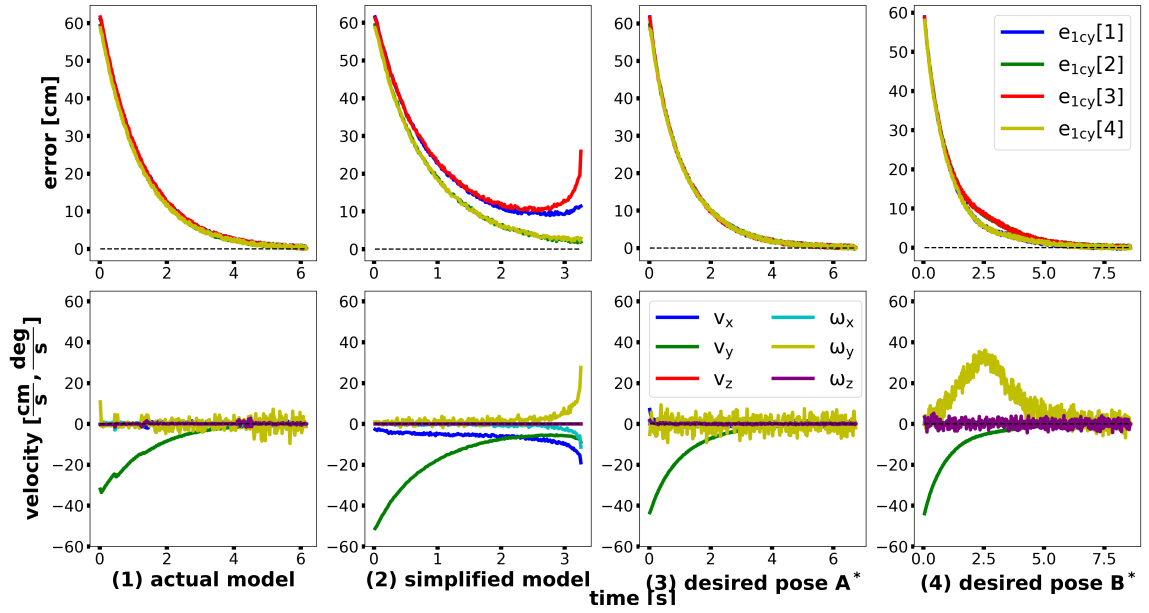


(b) Larger D - Array II

Figure 6.8 – Simulation results of Case 1 with task function e_{1cy} for initial pose A (a) with Array I and (b) Array II. For each value of d we execute task with actual model (1), simplified model (2) desired pose A^* (3) and desired pose B^* (4) of interaction matrix. Task function components (cm) versus time (s) on the top, end-effector velocities (cm/s and deg/s) versus time (s) on the bottom).



(a) Smaller d - Array I



(b) Larger D - Array II

Figure 6.9 – Simulation results of Case 1 with task function \mathbf{e}_{1cy} for initial pose B (a) with Array I and (b) Array II. For each value of d we execute task with actual model (1), simplified model (2) desired pose A^* (3) and desired pose B^* (4) of interaction matrix. Task function components (cm) versus time (s) on the top, end-effector velocities (cm/s and deg/s) versus time (s) on the bottom).

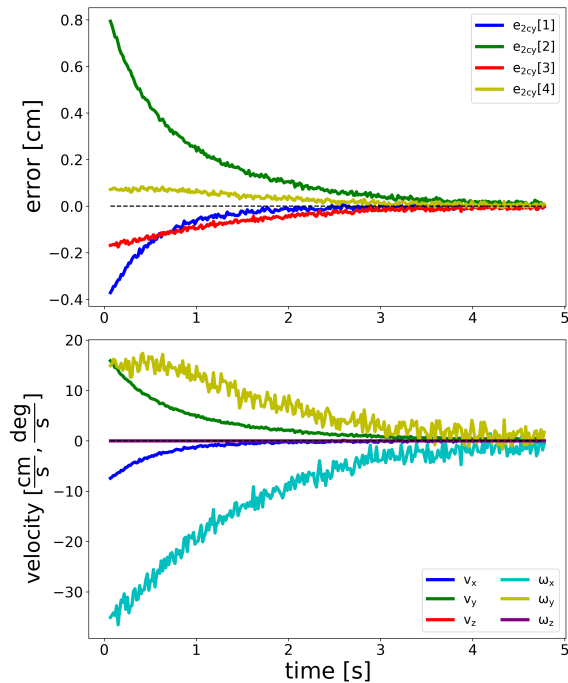


Figure 6.10 – Simulation results for Case 2 using task function \mathbf{e}_{2cy} with interaction model from special case. Task function components (cm) versus time (s) on the top, end-effector velocities (cm/s and deg/s) versus time (s) on the bottom.

\mathbf{e}_1	
$\alpha_{11} = 254^\circ$	$\alpha_{12} = 246^\circ$
$\alpha_{21} = 294^\circ$	$\alpha_{22} = 286^\circ$
$d_1 = -d_2 = 5.5 \text{ cm}, r_1 = r_2 = 7 \text{ cm}$	

Table 6.2 – Extrinsic parameters of proximity array used in experimentation.

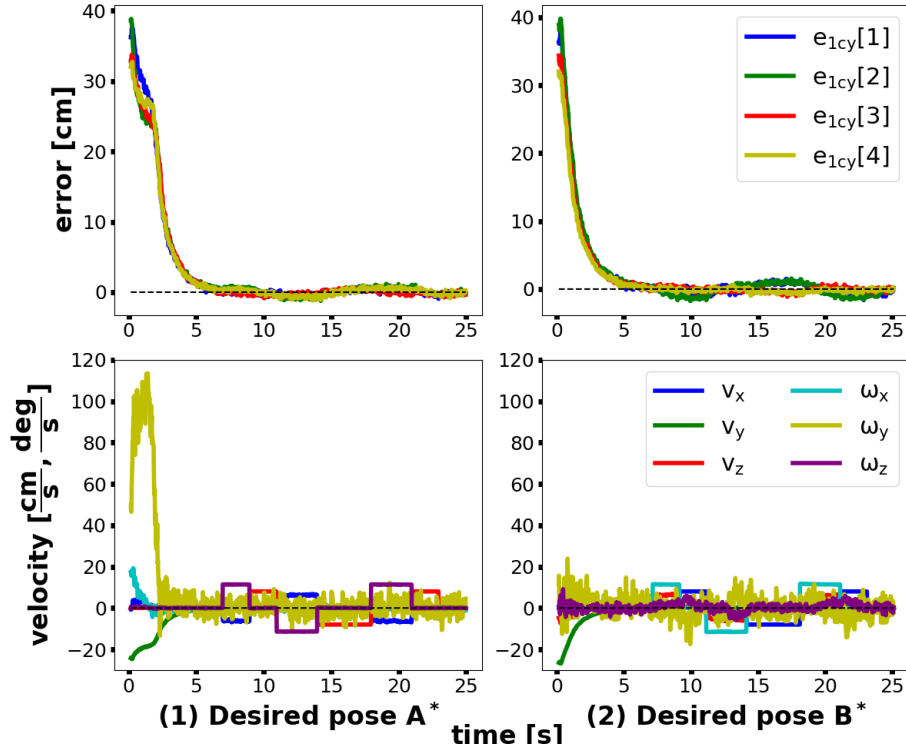


Figure 6.11 – Experimental results for surface following using task function e_{1cy} with models evaluated at desired pose A^* and B^* respectively from initial pose A. Task function components (cm) versus time (s) on the top, end-effector velocities (cm/s and deg/s) versus time (s) on the bottom.

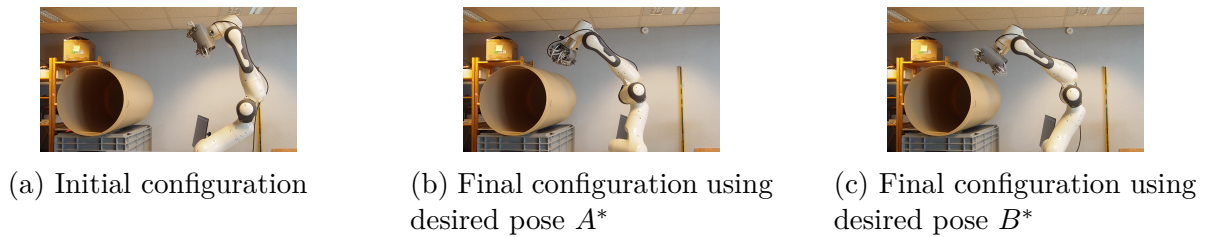


Figure 6.12 – Experimental setup for surface following

6.4 Experimental Results

Experimental results have been obtained using a Franka Emika Panda robot with a proximity array attached with minimal sensors as depicted in Figure 6.1 to detect the outside of a cylinder as seen in Figure 6.12. The extrinsic parameters of the proximity array are shown in Table 6.2. The video for experimental results can be seen here. Task \mathbf{e}_{1cy} starts from the initial configuration shown in Figure 6.12a within 50 *cm* range, which is similar to initial pose A. When the model (6.13) corresponding to desired pose A^* is used in the control scheme, the system converges to the configurations shown in Figure 6.12b, while the system converges to the configurations shown in Figure 6.12c with the model (6.17) corresponding to desired pose B^* . At convergence desired proximity measurements are $\delta_{ij}^* = 4$ *cm* and specific sequence of control inputs \mathbf{v}_1 and \mathbf{v}_2 for both cases mentioned in Sections 6.2.1.2 and 6.2.1.3 are applied to perform surface following using $v^* = 8$ *cm/s* and $\omega^* = -11.5$ *deg/s*. As seen in Figure 6.11 both control schemes ensure convergence to the final pose within approximately 8 sec of execution. In Figure 6.11(1), we can notice an initial rotation around the y -axis of \mathcal{F}_E with significant value in ω_z to reach the final desired pose in desired pose A^* . Once convergence occurs, surface following starts and as observed in the plots, the control scheme is able to still regulate \mathbf{e}_{1cy} to zero, thereby enabling the inspection of the cylindrical surface without any perturbation.

6.5 Conclusion

In this chapter we have considered an important application of proximity array system when mounted as an end-effector. We provided various interaction matrix models that enable this array to position with respect to the outside of the cylinder as close as 4 *cm* in proximity signal reading. The best results were obtained while evaluating the interaction matrix at the desired equilibrium pose. The control law was shown to be locally asymptotically stable. We verified the results for two desired poses where either z -axis or x -axis of end-effector frame became parallel to the axis of the cylinder at final configuration. This was verified through both simulation and experimentation. In experimentation at convergence a surface following task was considered by providing sequence of velocities that lie in the null space of the interaction matrix, which ensured that the corresponding task function remained achieved at convergence. Additionally, it was also shown that the interaction matrix model evaluated with a common approximation that

considers the target normal as aligned in the negative direction of sensor axis can lead to non-convergence.

As part of future work it could be interesting to obtain the optimal value of distance between rings and angular orientation of the proximity sensors attached to the ring for ensuring the best performance in terms of robustness. A detailed stability analysis could be interesting to provide more clarity on the error tolerance for each parameter of the model towards convergence of task function. While using redundant number of sensors, it is unclear now if GAS could be achieved. Thus it would be interesting to evaluate if physically consistent poses of the proximity array exist for which sensor feature error lies in the null space of combination matrix.

PART II

Combining Proximity and Vision in Sensor-based Control

INTRODUCTION

In Part II of this thesis, proximity and vision sensor information are combined together in closed-loop control within Sensor-based Control formalism. It is divided into three chapters. The part begins by introducing the state of the art in Chapter 7. In Chapter 8 we consider obstacle avoidance task using proximity sensors. This part ends with Chapter 9 where positioning in congested space is considered by utilizing both proximity and vision information. Numerous experimental and simulation results are provided to validate the theory considered.

STATE OF THE ART

In this chapter we provide the State of the Art for Part II. The chapter is divided into three sections. In Section 7.1 we give a general review on the past works in robotics to address the task of obstacle avoidance. The section ends with discussion on methodologies to incorporate and execute multiple tasks in robotics. In Section 7.2 we give more specific discussion on previous works where proximity sensors were used for the task of obstacle avoidance. The chapter ends with Section 7.3 where the current work is positioned wrt. past works and contributions are stated.

7.1 Past works on Obstacle Avoidance

In robotics the idea of *Safety* is often linked to the task of obstacle avoidance. A classical reference for obstacle avoidance in robotics is artificial potential field method [Khatib, 1985]. In this work collision avoidance was considered at low level control to obtain real-time behavior. Prior to this work, it was mainly considered at higher level planning. At each point on the obstacle a potential barrier is considered within its neighbourhood to create a repulsive force that prevents possible collisions with the environment. In [Maciejewski & Klein, 1985], obstacle avoidance point is referred to as the point on the manipulator that is closest to the obstacle. A velocity vector is assigned at the point to move the manipulator directly away from the obstacle surface. For a practical implementation this direction is considered to be obtained from sensory range finding devices. The main algorithm considers only a single point for avoidance, however it also recommends to incorporate multiple points to avoid oscillatory effects. Importance of local sensing for online collision detection and obstacle avoidance for autonomous robots operating in unknown environments or teleoperation in hostile environments is emphasized in [Espiau, 1988]. In [Espiau, 1988] among the various application of using proximity sensing in SBC framework, task of obstacle avoidance is considered. Here an obstacle avoidance primitive is considered to simplify the interaction matrix involved, which is designed to make the

robotic link attached to the sensor to move away from the obstacle along the opposite direction of the sensor axis. Although such a motion would allow to increase the distance between a single sensor and an obstacle as described in the work, it could be interesting to consider the actual model which makes the robotic link move along the local target normal direction when multiple obstacles are involved in close encounters. Potential field type method was used in [Marchand & Hager, 1998] for collision avoidance by providing a collision free path for the problem of dynamic sensor planning in Visual Servoing. In [Mansard & Chaumette, 2005], [Mansard & Chaumette, 2007] obstacle avoidance is considered using potential field method for SBC by considering obstacles in Cartesian space with simulation results in visual servoing. In [Flacco et al., 2012], a reactive collision avoidance approach is proposed using a depth camera sensor, using which repulsive vectors are evaluated for the end-effector to achieve safe human-robot coexistence.

One of the earliest work that recognized the need to include obstacle avoidance (or anti-collision as they called it) as inequality constraint is in [Faverjon & Tournassoud, 1987]. It was considered as a substitute to potential field method to overcome its drawback including oscillation for opposite obstacles, high repulsion from adjacent obstacles compared to just one and difficulty to get closer to the obstacle. Here such an inequality is obtained from the concept of velocity damper which ensures that the distance between the robotic link and the obstacle must not decrease too fast below an influential distance. It is pointed out that such a description enables the robotic link to go close to a security distance considered and that it remains closer to the original geometry of the problem. This was also considered as an Ordinary Differential Inequality (ODI) in [Kanoun et al., 2011], where an exponential convergence is achieved to a desired inequality constraint which is perceived as a task. This is recommended for obstacle avoidance. An alternative framework of dealing with constraints in SBC is shown in [Kermorgant & Chaumette, 2014], where it is treated as additional features. Inequality constraints on a certain sensor feature is taken into account at the feature level using a weighting function. In case of inequality constraints the weights cause a similar behaviour to that of repulsive potentials by increasing to infinity towards the boundary of inequality and being null in the safe region. Such a design would prevent the robotic link from being very close to an obstacle and can experience oscillations when several obstacles are present. It is suggested to use an adaptive gain that depends on the weights to slow the system down in such situations and is verified in a simulation scenario with 2 DoF Cartesian robot while passing a corridor. In [Devigne et al., 2016], such a constraint was used for obstacle avoidance using

ultrasonic distance sensors in shared control for wheelchair navigation for assisting users with motor or visual impairments. The idea of velocity damper or ODI has been generalized in safety-critical controllers through control barrier function (CBF) [Ames et al., 2019]. A safe set is ensured through an inequality constraint in the control input of the corresponding dynamical system.

One way to consider such practical constraints encountered during the operation of the robot is to define it as a cost function. The minimization of this function leads to safety and can be incorporated in control with gradient projection method [Rosen, 1960]. For kinematically redundant robots, local optimization based techniques can provide real-time implementation with possibility to consider sensory information [Nenchev, 1989]. One way to fulfill a secondary task while executing a primary task is to use gradient projection method in which the secondary task is realized without affecting the primary task as long as the primary task permits it [Liégeois, 1977]. In SBC, this is considered in the realization of a hybrid task [Samson & Espiau, 1990], [Espiau et al., 1992]. In situations where there are multiple tasks one effective way is to use Task priority framework [Nakamura et al., 1987], [Slotine & Siciliano, 1991] and [Chiaverini, 1997] where a predetermined hierarchy is provided to fulfill tasks. Execution of a task at lower priority would not interfere with or disturb the execution of higher priority tasks. One direct disadvantage of such a static priority-order is that the task at lowest priority need not be always executed. In some applications it might be more desirable to change the priority order dynamically. In [Mansard & Chaumette, 2007] a generic solution is provided for sequencing of tasks with application in SBC. If one of the task or a constraint is modeled as an inequality then a Quadratic Programming (QP) framework can be used to realize the primary task at best subject to the satisfaction of constraint [Faverjon & Tournassoud, 1987]. The inequality task or constraint achieves higher priority in QP framework. In this work, QP formalism with a hierarchical description of a cluttered simulation environment of a nuclear plant enabled a 10 DoF manipulator with 20 *cm* maximum velocity at end-effector point to generate local collision free trajectories in real-time. In [Decre et al., 2009] inequality constraints are considered in constrained based programming approach iTASC, where sensor-based tasks specified as constraint sets are executed through a least-squares formalism. This is applied to a laser tracing task in simulation, where laser distance to be maintained is specified as inequality. In [Kanoun et al., 2011], the priority based framework is extended to consider inequality task or constraints. The computation cost of executing the above priority based framework is improved significantly in [Escande et al.,

2014] using matrix decomposition that respects priorities. The above algorithm is tested on a 36 DoF humanoid robot with control loop operating at 200 Hz . In [Vaillant et al., 2016], a multi-objective model based QP controller is used to perform tasks generated offline from a multi-contact planner for climbing a vertical industrial-norm ladder. The QP controller runs in real-time with maximum of 3 or 4 contacts while climbing the ladder. Visual servoing task is integrated into the above framework in [Agravante et al., 2017], while also considering visual constraints (occlusion avoidance, field of view maintenance, etc.) expressed as inequalities. Experimental results are provided for vision tasks including gaze control and grasping of a box with vision operating at 30 Hz and velocity controller for 36 DoF humanoid robot at 10 Hz . In [Ames et al., 2019], it is recommended to use optimization based controllers to combine inequalities obtained from the application of CBF theory to guarantee safety in a minimally invasive fashion to an existing controller that would define a robotic task.

In the next section we discuss past works where proximity sensors was used to perform obstacle avoidance.

7.2 Past works on using proximity sensors for Obstacle Avoidance

In [Cheung & Lumelsky, 1988], a local strategy is envisioned to move along the local tangent in configuration space whenever an obstacle is detected for a planar scenario, thereby ensuring that the global planning method does not lead to collision or loss of detection with obstacle. Calculation of local tangent is based on what point of the robot body is in close proximity with the obstacle. The work is extended to motion planning of 3D arm manipulator in uncertain environment in [Cheung & Lumelsky, 1989] with 500 infrared sensors and later for the task of collision avoidance in case of a hybrid teleoperation system in [Lumelsky & Cheung, 1993]. In [Novak & Feddema, 1992] and [Feddema & Novak, 1994], a capacitance based Whole Arm Proximity (WHAP) sensors is considered as a collision avoidance system. Such a system is designed to be used for the application of remote handling of hazardous waste using a robotic manipulator. The authors argue about the need for spatially resolved proximity data containing distance to obstacle and to design a collision avoidance motion along the surface normal vector at location that is likely to collide. The control system restricts motion along this normal direction to prevent collision and thereby also allowing other unconstrained motions for continuing fulfillment

of the original task. In [Novak & Feddema, 1992] experimental results are provided for planar arm while following a trajectory, while in [Feddema & Novak, 1994] the results are extended to a 6 DoF serial manipulator for the task of teleoperation. In [D. G. Wegerif & Rosinski, 1992] and [D. Wegerif et al., 1993], collision avoidance capability is provided to a manipulator using whole-arm sensing mechanism involving proximity sensors with mainly infrared technology. The use case application was to perform hazardous tasks in nuclear and space applications. Task was performed using the approach in [Maciejewski & Klein, 1985] but by giving higher priority to obstacle avoidance compared to that of maintaining an end-effector velocity. Experimentation were performed on 3 DoF planar mechanism and 6 DoF serial manipulator. It was pointed out that oscillations were visible and can be reduced by considering multiple obstacle avoidance points. Safety enhancement of human-robot interaction in industrial environments is considered in [Ceriani et al., 2013] using distributed infrared proximity sensors. Collision avoidance is considered using a potential like function called danger field where robot is the field of source rather than the obstacle [Buizza Avanzini et al., 2014]. Danger field is then combined with a virtual impedance approach and the gradient of cumulative sum of various danger fields from possible collisions is used in control strategy as a virtual force.

In [Vergara Perico et al., 2019] proximity sensing modality of artificial skin developed by [Mittendorfer & Cheng, 2012], [Cheng et al., 2019] was used for human robot collaboration for an industrial use case involving assembly operation. Around 373 interconnected cells were arranged around a 6 DoF serial manipulator. Reactive collision avoidance was achieved by defining a reference velocity where the robot is made to move backwards along surface normal or along the normal plane at detection cell. In [Ding et al., 2019], proximity sensing cuff with ToF and capacitive measurements is presented for enabling safe human-robot interaction. Higher priority is given to collision avoidance over task motion. Collision avoidance velocity is evaluated in a direction orthogonal to the detection vector at obstacle avoidance point to the closest point on obstacle. The work is then extended in [Ding & Thomas, 2020], to increase the set of possible feasible motions through inequality constraints. These constraints are evaluated on the collision avoidance velocity based on the relative distance to collision. Experimental results are shown in 7 DoF serial manipulator. Sensitive skin consisting of capacitive sensors with proximity range within 300 mm developed by FOGALE robotics¹ was used in [M'Colo et al., 2019] for obstacle avoidance in applications involving human-robot interactions. Collision avoidance velocity vector is

1. <https://www.fogale-robotics.com/>

evaluated from the distance to obstacle and normal vector associated to the set of active electrodes that constitute the skin. Set of all collision vectors associated with each link in proximity to obstacle is combined together along with the main task in a least-squares formalism using a weighting strategy. Experimentation was performed on a 6 DoF serial manipulator to show the capability of the controller developed to avoid dynamic obstacles.

In the next section we conclude this chapter by positioning our work wrt. state of the art.

7.3 Positioning current work

In [Cherubini & Navarro-Alarcon, 2021], a comprehensive review is provided to emphasize on the importance of SBC as a paradigm of task execution towards applications requiring human-robot collaboration. It is pointed out here that distance information from proximity sensors would address the aspect of *Safety*. The work also emphasizes on the lack of literature in combining both vision and proximity sensors in SBC. It also points out that task constraints has not been fully implemented while using multiple sensing modalities. In [Cherubini & Chaumette, 2013] and [Cherubini et al., 2014], distance measurements from laser range sensors were integrated into a visual navigation task for mobile robots to perform obstacle avoidance for static as well as dynamic obstacles. However, laser range sensors cannot be considered as proximity sensors as it is not meant for local sensing.

Previous works in the area considered vision information and proximity information separately. In this work, we would like to combine vision and proximity to perform positioning tasks. Sensor information is combined in a shared manner using a QP formalism. Previous works in proximity-based control approximated the unknown target normal parameter in the interaction matrix. Here we show that there exist practical situations where such an approximation leads to bad results. We provide estimation strategies whose results ensure achieving the task and stable behavior while interacting with obstacles that prevent the robot from reaching its final goal. We adapt ideas from safety-critical control to define task function in proximity space as inequalities. This ensures anti-collision while positioning in congested spaces. The simplicity achieved in modeling is reflected in the various experimental and simulation results obtained. It also provides good platform for future work.

In the next chapter we consider the problem of obstacle avoidance.

OBSTACLE AVOIDANCE TASK

8.1 Introduction

In this chapter we consider the task of obstacle avoidance using proximity sensors as an inequality. We model the task in SBC framework by considering the interaction matrix involved with distance feature from a thin field proximity sensor. The chapter is divided into five sections including this. In Section 8.2, interpretation of the task is provided. After that different interaction matrix models are discussed in Section 8.3. Section 8.4 details on how the obstacle task can be modelled as a linear inequality constraint on the control input involving the interaction matrix model. Finally the chapter ends with a conclusion in Section 8.5.

8.2 Task interpretation

As in Chapter 3, let us consider the detection of an object in the workspace using a proximity sensor i located on j th ring as shown in Figure 8.1. Let us recall that δ_{ij} indicates the distance measured by the sensor wrt. the target along the axis $\mathbf{n}_{S_{ij}}$ with $\mathbf{n}_{T_{ij}}$ indicating the target normal at detection point T_{ij} . The modeling for this has already been considered in Section 3.1.4. It was pointed out in that section that the interaction screw is a slider at the target point T_{ij} . Let us therefore look at proximity signal kinematics from (3.9), (3.16) and express it at target point T_{ij} using shifting law (3.6)

$$\dot{\delta}_{ij} = \begin{pmatrix} \mathbf{u}_{ij}^T & \mathbf{H}_{ij}(S_j)^T \end{pmatrix} \begin{pmatrix} \mathbf{v}_S \\ \omega_S \end{pmatrix} = \begin{pmatrix} \mathbf{u}_{ij}^T & \mathbf{H}_{ij}(T_{ij})^T \end{pmatrix} \begin{pmatrix} \mathbf{v}_S(T_{ij}) \\ \omega_S \end{pmatrix} \quad (8.1)$$

$$\begin{aligned} &= \frac{-1}{\mathbf{n}_{T_{ij}} \cdot \mathbf{n}_{S_{ij}}} \begin{pmatrix} \mathbf{n}_{T_{ij}}^T & \mathbf{0}^T \end{pmatrix} \begin{pmatrix} \mathbf{v}_S(T_{ij}) \\ \omega_S \end{pmatrix} \\ &= \beta_{ij} \zeta_{ij} \end{aligned} \quad (8.2)$$

where $\beta_{ij} = \frac{-1}{\mathbf{n}_{T_{ij}} \cdot \mathbf{n}_{S_{ij}}}$ and $\zeta_{ij} = (\mathbf{n}_{T_{ij}}^T \quad \mathbf{0}^T) \begin{pmatrix} \mathbf{v}_S(T_{ij}) \\ \omega_S \end{pmatrix}$ is called separation velocity at contact in rigid body contact mechanics [Featherstone, 2008]. Since the first order derivative of proximity sensor signal involves the contact separation term, we could model obstacle avoidance task with proximity signals using contact constraints.

Let us first consider the interaction of proximity sensor with a target point T_{ij} on obstacle O as shown in Figure 8.1. Let us also assume the obstacles to have smooth surfaces, which results in smooth variation of the detected sensor signal. If the sensor

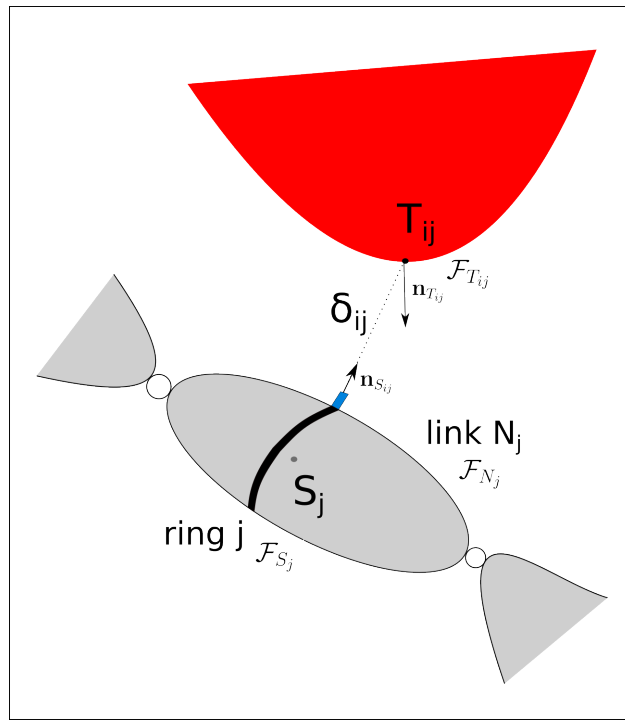


Figure 8.1 – Detection of a single point on Obstacle

signal falls below a threshold δ_{ij}^* , we assume a virtual penetration with the obstacle. Therefore, we can define a simple function $\mathbf{e}_{\delta_{ij}}$ that would act as our contact constraint, as it indicates the signed distance between two bodies [Featherstone, 2008]

$$\mathbf{e}_{\delta_{ij}} = \delta_{ij} - \delta_{ij}^* \quad (8.3)$$

To achieve obstacle avoidance we have to satisfy the following inequality/unilateral constraint,

$$\mathbf{e}_{\delta_{ij}} \geq 0 \quad (8.4)$$

The moment we have $\mathbf{e}_{\delta_{ij}} = 0$, we can consider a virtual contact. Kinematically, we have three scenarios from this moment,

$$\begin{aligned} \text{Case 1} & : \dot{\mathbf{e}}_{\delta_{ij}} < 0 \\ \text{Case 2} & : \dot{\mathbf{e}}_{\delta_{ij}} = 0 \\ \text{Case 3} & : \dot{\mathbf{e}}_{\delta_{ij}} > 0 \end{aligned} \tag{8.5}$$

With Case 1, we would have virtual penetration and eventual real contact with the obstacle. If the situation is as per Case 2, then we would have virtual contour following. Finally with Case 3, we enable the two bodies to move away and avoid the risk of real contact. This third scenario is what we would like to achieve through control whenever virtual contact occurs or is in its vicinity.

Let us recall the interaction matrix involved from Chapter 3,

$$\mathbf{L}_{\delta_{ij}} = -\frac{1}{\mathbf{n}_{T_{ij}} \cdot \mathbf{n}_{S_{ij}}} \begin{bmatrix} \mathbf{n}_{T_{ij}}^T & (\mathbf{m}_{T_{ij}} \times \mathbf{n}_{T_{ij}})^T \end{bmatrix} \tag{8.6}$$

Therefore, the condition for avoidance of collision with obstacle using proximity sensors at virtual contact is,

$$\dot{\mathbf{e}}_{\delta_{ij}} = \dot{\delta}_{ij} = \begin{pmatrix} \mathbf{u}_{ij}^T & \mathbf{H}_{ij}(T_{ij})^T \end{pmatrix} \begin{pmatrix} \mathbf{v}_S(T_{ij}) \\ \omega_S \end{pmatrix} = \mathbf{L}_{\delta_{ij}} \mathbf{v}_{S_j} \geq \alpha \quad \text{when } \mathbf{e}_{\delta_{ij}} = 0 \tag{8.7}$$

where $\mathbf{L}_{\delta_{ij}}$ and \mathbf{v}_{S_j} are evaluated at ring center S_j associated with frame \mathcal{F}_{S_j} . Here $0 \leq \alpha \leq \beta_{ij}\zeta_{ij}$. This condition has to be activated only in the vicinity of virtual penetration ($\delta_{ij} = \delta_{ij}^*$). In situations that demand faster reactions the behavior can be achieved by increasing the value of α . When the value of ζ_{ij} is maximum we essentially confine the velocity along target normal direction. This would result in an idea similar to obstacle avoidance point in [Maciejewski & Klein, 1985], which is a more restrictive way of achieving avoidance.

The above defined condition for avoidance defined using spatial velocity of the sensor can be considered in the joint space of the robot. Let us consider as shown in Figure 8.1 that the sensor is attached to link N_j of the robot. In that case the first order derivative

of the task function can be written as,

$$\dot{\mathbf{e}}_{\delta_{ij}} = \mathbf{L}_{\delta_{ij}} \mathbf{v}_{S_j} = \mathbf{L}_{\delta_{ij}} {}^j\mathbf{V}_{N_j} \mathbf{J}_{N_j} \dot{\mathbf{q}} = \mathbf{J}_{e_{\delta_{ij}}} \dot{\mathbf{q}} \quad (8.8)$$

where ${}^j\mathbf{V}_{N_j}$ is the twist transformation between frame \mathcal{F}_{S_j} and frame \mathcal{F}_{N_j} . \mathbf{J}_{N_j} is the geometric Jacobian of the link in its own frame. In the next section we consider the interaction matrix model $\mathbf{L}_{\delta_{ij}}$ that can be considered for the task.

8.3 Model for interaction matrix

In this section we discuss about the two kinds of interaction matrix models that are suitable for achieving the task. We also provide recommendations on when to use each of these models.

8.3.1 Actual Model : Estimation of Surface Normal

If for an application, we require to maximally utilize the space available then the actual model would be required. In those cases, the unknown term of local target normal $\mathbf{n}_{T_{ij}}$ requires to be estimated. This can be considered in two ways, either with a single sensor detection or with multiple sensors.

For one sensor

Here we consider the estimation $\hat{\mathbf{n}}_{T_{ij}}$ of $\mathbf{n}_{T_{ij}}$ assuming that only one sensor is detecting the obstacle. Let us consider the case of proximity sensor and detected surface as shown in Figure 8.1. To group the target normal term $\hat{\mathbf{n}}_{T_{ij}}$, we start with (8.1) to get

$$\dot{\delta}_{ij} = -\frac{\hat{\mathbf{n}}_{T_{ij}} \cdot \mathbf{v}_{S_j}|_{T_{ij}}}{\hat{\mathbf{n}}_{T_{ij}} \cdot \mathbf{n}_{S_{ij}}} \quad (8.9)$$

The above equation can be further rearranged to obtain a linear equation in $\hat{\mathbf{n}}_{T_{ij}}$

$$\hat{\mathbf{n}}_{T_{ij}} \cdot (\dot{\delta}_{ij} \mathbf{n}_{S_{ij}} + \mathbf{v}_{S_j}|_{T_{ij}}) = 0 \quad (8.10)$$

By applying the shifting law, the linear velocity at point T_{ij} is

$$\mathbf{v}_{S_j}|_{T_{ij}} = \mathbf{v}_{S_j} - (\delta_{ij} \mathbf{n}_{S_{ij}} + r_j) \times \boldsymbol{\omega}_{S_j} \quad (8.11)$$

To obtain target normal estimate expressed in the current ring frame \mathcal{F}_{S_j} at current time instant t_p , let us consider a set of past measurements at previous instances $\{t_q : q = 1 \dots p\}$.

$${}^{t_q}\hat{\mathbf{n}}_{T_{ij}} \cdot {}^{t_q}(\dot{\delta}_{ij}\mathbf{n}_{S_{ij}} + \mathbf{v}_{S_j}|_{T_{ij}}) = 0 \quad (8.12)$$

Each of these measurements can be expressed in the current ring frame using rotation matrix ${}^{t_p}R_{t_q}$ between ring frames at instance t_q and t_p .

$${}^{t_p}R_{t_q} {}^{t_q}\hat{\mathbf{n}}_{T_{ij}} \cdot {}^{t_p}R_{t_q} {}^{t_q}(\dot{\delta}_{ij}\mathbf{n}_{S_{ij}} + \mathbf{v}_{S_j}|_{T_{ij}}) = 0 \quad (8.13)$$

From a set of past measurements we obtain a homogeneous system of equations of the form $\mathbf{A}_n {}^{t_p}\hat{\mathbf{n}}_{T_{ij}} = \mathbf{0}$, where

$$\mathbf{A}_n = \begin{bmatrix} \cdot \\ {}^{t_p}R_{t_q} {}^{t_q}(\dot{\delta}_{ij}\mathbf{n}_{S_{ij}} + \mathbf{v}_{S_j}|_{T_{ij}})^T \\ \cdot \end{bmatrix}_{p \times 3} \quad (8.14)$$

This can be expressed as a homogeneous least-squares problem [Inkila, 2005],

$$\begin{aligned} {}^{t_p}\hat{\mathbf{n}}_{T_{ij}} &= \underset{\mathbf{x} \in \mathbb{R}^3}{\operatorname{argmin}} \|\mathbf{A}_n \mathbf{x}\|^2 \\ \text{s.t. } &\|\mathbf{x}\| = 1 \end{aligned} \quad (8.15)$$

One way to find a solution for the above problem is by using singular value decomposition, $\mathbf{A}_n = \mathbf{U}\Sigma\mathbf{V}^T$. The last column of \mathbf{V} is the solution to estimate ${}^{t_p}\hat{\mathbf{n}}_{T_{ij}}$ [Hartley & Zisserman, 2004].

With multiple sensors

In our case, proximity sensors are attached to the robot in the form of rings. If we assume that the obstacles are large enough to be detected at an instant by multiple proximity sensors on the ring then the target normal can be evaluated using the values from adjacent sensors. Let us consider sensor i on j th ring. The point detected by the

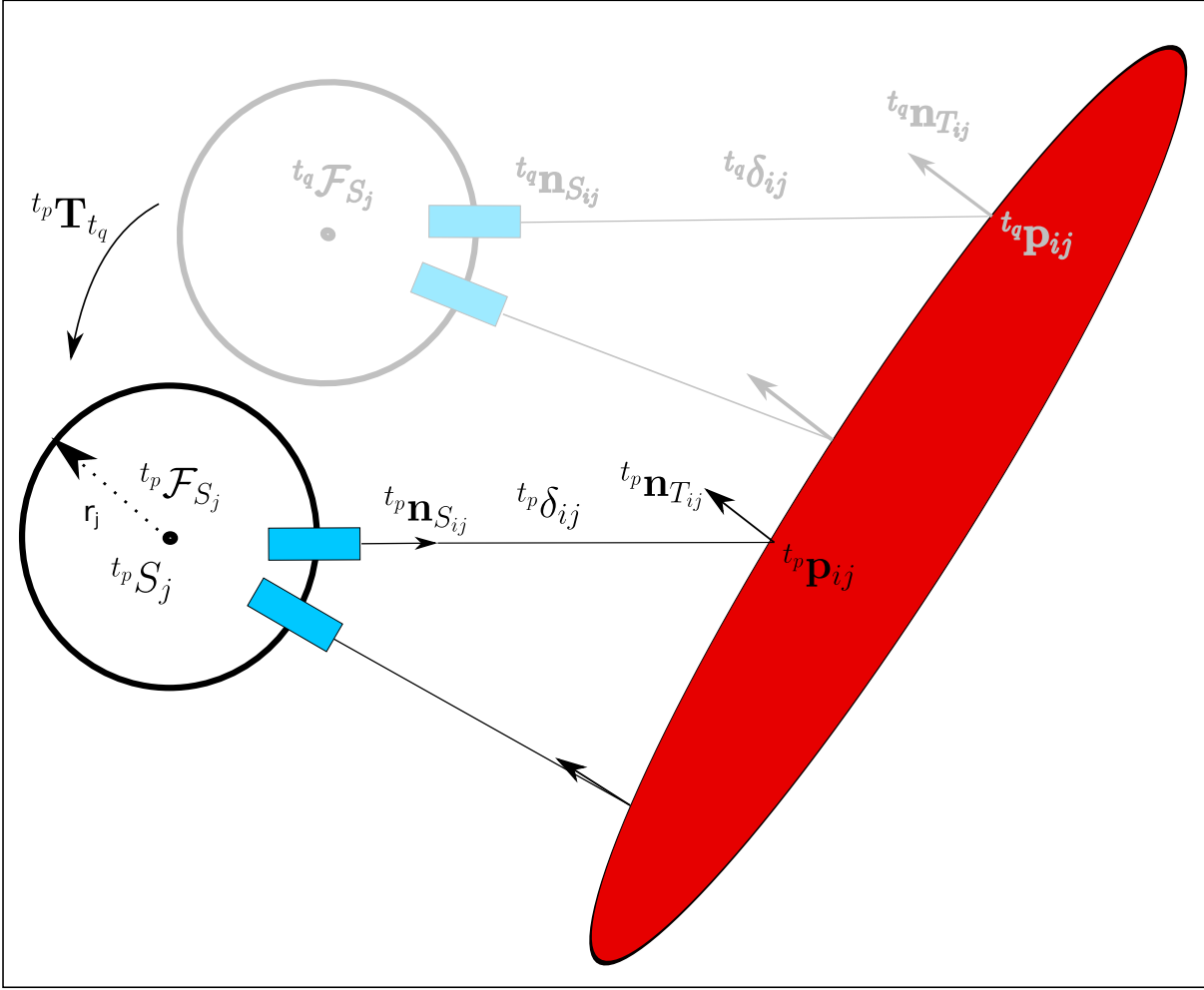


Figure 8.2 – Detection of multiple points for estimating target normal

sensor at current instance t_p in sensor frame would be

$${}^{t_p} \mathbf{p}_{ij} = ({}^{t_p} \delta_{ij} + r_j) {}^{t_p} \mathbf{n}_{S_{ij}} \quad (8.16)$$

Let us consider \mathcal{N}_{ij} to contain the indices of the sensors that are detecting the same object as currently detected by sensor i on j th ring in previous instances $\{t_q : q = 1 \dots p\}$. Let us consider a sensor $k \in \mathcal{N}_{ij}$ for a past instance t_q , we transform the point detected by this sensor to the current sensor frame of the ring using homogeneous transformation ${}^{t_p} \mathbf{T}_{t_q}$, i.e. ${}^{t_p} \mathbf{T}_{t_q} {}^{t_q} \mathbf{p}_{k_j}$. Points from previous instances of a sensor are ignored if its relative distance with the point of detection of current instance is lower than a threshold. This leads us to consider a set of n_{ij} eligible points $\{., {}^{ij} \mathbf{p}_s, .\}_{s=1 \dots n_{ij}}$. All these measurements

can be assumed to lie on a planar surface defined by $\mathbf{n}^T \mathbf{p} + d = 0$, where \mathbf{p} represents an arbitrary point and (\mathbf{n}^T, d) indicates plane parameters. The target normal can be framed as a total linear least squares problem

$$\begin{aligned} \hat{\mathbf{n}}_{T_{ij}}, d_{ij}^* &= \underset{\mathbf{n} \in \mathbb{R}^3, d \in \mathbb{R}}{\operatorname{argmin}} \sum_{s=1}^{n_{ij}} \left({}^{ij} \mathbf{p}_s^T \mathbf{n} - d \right)^2 \\ \text{s.t. } & \|\mathbf{n}\| = 1 \end{aligned} \quad (8.17)$$

The solution can be obtained as the smallest eigen vector corresponding to the covariance matrix \mathbf{B}_n given by [Hoppe et al., 1992], [Mitra & Nguyen, 2003]

$$\mathbf{B}_n = \frac{1}{n_{ij}} \sum_{s=1}^{n_{ij}} \left({}^{ij} \mathbf{p}_s - \mathbf{p}_c \right) \left({}^{ij} \mathbf{p}_s - \mathbf{p}_c \right)^T \quad (8.18)$$

where $\mathbf{p}_c = \frac{1}{s} \sum_{s=1}^{n_{ij}} {}^{ij} \mathbf{p}_s$. For the practical task involving evaluation of target normal for multiple sensors on the ring, a computationally fast approximation of this solution given in [Badino et al., 2011] is used

$$\hat{\mathbf{n}}_{T_{ij}} = \frac{1}{\|\tilde{\mathbf{B}}_n^{-1} \tilde{\mathbf{b}}\|} \tilde{\mathbf{B}}_n^{-1} \tilde{\mathbf{b}} \quad (8.19)$$

where $\tilde{\mathbf{B}}_n = \sum_{s=1}^{n_{ij}} {}^{ij} \mathbf{p}_s {}^{ij} \mathbf{p}_s^T$ and $\tilde{\mathbf{b}} = \sum_{s=1}^{n_{ij}} {}^{ij} \mathbf{p}_s$.

8.3.2 Approximate Model: Approximation in intensity of interaction screw

A practical approximation of the actual model would be to consider $\beta_{ij} = 1$ in (8.1). From the perspective of interaction screw this approximation can be considered as considering unit intensity. This term would involve the reciprocal of the dot product of $\mathbf{n}_{T_{ij}}$ and $-\mathbf{n}_{S_{ij}}$. For close encounters with obstacles, β_{ij} could result in undesirable non-smooth behavior in the evolution of task function.

$$\mathbf{L}_{\delta_{ij}} = \left[\mathbf{n}_{T_{ij}}^T \quad (\mathbf{m}_{T_{ij}} \times \mathbf{n}_{T_{ij}})^T \right] \quad (8.20)$$

8.3.3 Approximate Model: Approximation in direction of interaction screw

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

$$\mathbf{L}_{\delta_{ij}} = \begin{bmatrix} -\mathbf{n}_{S_{ij}}^T & \mathbf{0}^T \end{bmatrix} \quad (8.21)$$

This is considered as obstacle avoidance primitive in [Espiau, 1990], where the surface is considered to be locally orthogonal to the sensor axis. For situations where obstacles are far away or if the number of obstacle are less in number, then it is enough to consider this obstacle avoidance primitive. In Section 9.3.2.4, we show a scenario with two obstacles where the approximate mode is inadequate. This could also be the case while operating in tight spaces. In the next section we consider how obstacle avoidance task is implemented through control design.

8.4 Anti-collision using Proximity-based Control

In this section we consider the control design for implementing avoidance of obstacle or achieving anti-collision. Initially a brief introduction is given regarding achieving safety in control. Then these ideas are adapted to the case of proximity sensors.

8.4.1 Safety-Critical Control

In this subsection, we summarize the notion of safety-critical control that is later used to achieve the task of obstacle avoidance. The basic intuitive notion is that safety essentially corresponds to the prevention of undesirable situations. This notion of safety is formalised as ensuring the invariance of a set whose complement essentially corresponds to the undesirable situations. This set denoted by \mathcal{C} is defined as a superlevel set based on a continuously differentiable function $h(\mathbf{x})$ for a system with state \mathbf{x} [Ames et al., 2019]

$$\begin{aligned} \mathcal{C} &= \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}) \geq 0\} \\ \partial\mathcal{C} &= \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}) = 0\} \\ \text{Int}(\mathcal{C}) &= \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}) > 0\} \end{aligned} \quad (8.22)$$

Here $\partial\mathcal{C}$ denote its boundary and $\text{Int}(\mathcal{C})$ its interior. Let us define the idea of forward invariant set for an autonomous dynamical system $\dot{\mathbf{x}} = f(\mathbf{x})$ [Ames et al., 2017],

Definition 1 *A set \mathcal{C} is forward invariant if for every $\mathbf{x}(t_0) \in \mathcal{C}$ we also have $\mathbf{x}(t) \in \mathcal{C}$ with $t > t_0$*

If we ensure the set \mathcal{C} as forward invariant the system can be considered as safe.

A fundamental result called Nagumo's theorem [Blanchini, 1999] provides a necessary and sufficient condition for set invariance of \mathcal{C} where $\nabla h(x) \neq \mathbf{0}$ for all x in $\partial\mathcal{C}$,

$$\dot{h}(\mathbf{x}) \geq 0 \quad \forall \mathbf{x} \in \partial\mathcal{C} \quad (8.23)$$

Here ∇ operator indicates the gradient of a scalar valued function wrt. \mathbf{x} . The above condition which is essentially defined on the boundary is then extended to the entire set and also to accommodate non-autonomous dynamical systems to make it invariant based on the design of control input.

Before proceeding ahead, let us define an extended class \mathcal{K} function [Tan & Dimarogonas, 2024],

Definition 2 *A continuous function $\alpha : (-b, a) \rightarrow \mathbb{R}$ that is strictly increasing such that $\alpha(0) = 0$ with $a, b \in \mathbb{R}_{>0}$ is called extended class \mathcal{K} function.*

An example of such function is $\alpha(\mathbf{x}) = \mathbf{x}^c$ where c is odd positive integer. Here c is a positive integer which results in functions like x, x^3 etc. and a and b are some positive numbers which decides the domain of \mathbf{x} . A very promising way to guarantee safety is utilizing the idea of Control Barrier Functions (CBFs). It provides strong conditions on function $h(\mathbf{x})$ which was used in (8.22) to describe safe set. Here we restate the definition of a CBF in [Ames et al., 2019]

Definition 3 *For a nonlinear affine control system $\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}$, function $h(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ is a CBF if there exists an extended class \mathcal{K} function $\alpha(\cdot)$ such that,*

$$\sup_{\mathbf{u} \in U} [L_f h(\mathbf{x}) + L_g h(\mathbf{x})\mathbf{u}] \geq -\alpha(h(\mathbf{x})) \quad \forall \mathbf{x} \in D \quad (8.24)$$

where $L_f h(\mathbf{x}) = \nabla h(\mathbf{x})^T f(\mathbf{x})$ and $L_g h(\mathbf{x}) = \nabla h(\mathbf{x})^T g(\mathbf{x})$ are the Lie derivatives for the function $h(\mathbf{x})$ for the dynamical system considered [Sastry, 1999]. It is also assumed that there is no singularity at the boundary which corresponds to $L_g h(\mathbf{x}) \neq 0$ at $\partial\mathcal{C}$ [Xiao

et al., 2023]. CBF is both a necessary and sufficient condition for safety [Ames et al., 2019].

Let us define a set $K_{\text{cbf}}(\mathbf{x})$ consisting of control inputs that satisfies (8.24),

$$K_{\text{cbf}}(\mathbf{x}) = \{\mathbf{u} \in U : L_f h(\mathbf{x}) + L_g h(\mathbf{x})\mathbf{u} + \alpha(h(\mathbf{x})) \geq 0\} \quad (8.25)$$

As proven in [Ames et al., 2017], any controller $\mathbf{u} \in K_{\text{cbf}}(\mathbf{x})$ at a given \mathbf{x} renders the set \mathcal{C} forward invariant. This ensures that the system state does not reach the complement of the set, which is described as unsafe. Additionally, it also makes the set asymptotically stable. In situations where the system leaves the safe set to its complement, it is brought back to safety. Such a situation can occur in a practical implementation of a task using robots due to modeling errors and sensor uncertainties. This is another characteristics of CBF which makes it useful for practical implementations.

8.4.2 Safety-Critical Control in Sensor-based Control

In context of obstacle avoidance this refers to prevention of collision with obstacles. From the task interpretation considered in subsection 8.2, CBF becomes a suitable way of ensuring safety from collision using proximity sensors. In this subsection, we derive the conditions of task realization based on the application of CBFs.

Let us consider a velocity controller evaluated in SBC to control the joint state $\mathbf{q} \in \mathbb{R}^n$ of the robot. The control input is evaluated from regulation of a task function in sensor space. Let us consider task function $\mathbf{e}_c(\mathbf{q})$ in vision space, which leads to a control input \mathbf{u} . The corresponding dynamical system in joint space is

$$\dot{\mathbf{q}} = \mathbf{u} \quad (8.26)$$

As an example, let us consider the standard control input obtained from an exponentially decoupled reduction of $\mathbf{e}_c(\mathbf{q})$ expressed in joint space [Chaumette & Hutchinson, 2007]. This is further explained in Chapter 9 (mentioned in (9.7)).

$$\mathbf{u} = -\lambda \mathbf{J}_{e_c}^+ \mathbf{e}_c(\mathbf{q}) \quad (8.27)$$

This can be considered as a dynamical system in joint space as

$$\dot{\mathbf{q}} = -\lambda \mathbf{J}_{e_c}^+ \mathbf{e}_c(\mathbf{q}) \quad (8.28)$$

Now let us define the safe set based on Section 8.2.

$$\begin{aligned}\mathcal{C} &= \left\{ \mathbf{q} \in \mathbb{R}^n : \mathbf{e}_{\delta_{ij}}(\mathbf{q}) \geq 0 \right\} \\ \partial\mathcal{C} &= \left\{ \mathbf{q} \in \mathbb{R}^n : \mathbf{e}_{\delta_{ij}}(\mathbf{q}) = 0 \right\} \\ \text{Int}(\mathcal{C}) &= \left\{ \mathbf{q} \in \mathbb{R}^n : \mathbf{e}_{\delta_{ij}}(\mathbf{q}) > 0 \right\}\end{aligned}\quad (8.29)$$

It becomes clear that our intuitive notion of obstacle avoidance task as considered in Section 8.2 is infact Nagumo's classical theorem, which for the i th proximity sensor in j th ring would be

$$\mathbf{e}_{\delta_{ij}}(\mathbf{q}) \geq 0 \quad \forall \mathbf{q} \in \partial\mathcal{C} \quad (8.30)$$

In case of the dynamical system (8.26), the Lie derivatives would be $L_f \mathbf{e}_{\delta_{ij}}(\mathbf{q}) = 0$ and $L_g \mathbf{e}_{\delta_{ij}}(\mathbf{q}) = \left(\frac{\partial \mathbf{e}_{\delta_{ij}}}{\partial \mathbf{q}} \right)^T = \mathbf{J}_{e_{\delta_{ij}}}$.

The standard task function defined in proximity space becomes a CBF if we find an extended class \mathcal{K} function $\alpha(\cdot)$ that satisfies the following condition for a given control input,

$$\sup_{\mathbf{u} \in U} \left[\mathbf{J}_{e_{\delta_{ij}}} \mathbf{u} \right] \geq -\alpha(\mathbf{e}_{\delta_{ij}}) \quad \forall \mathbf{e}_{\delta_{ij}} \in D \quad (8.31)$$

We now define the set $K_{\text{cbf}}(\mathbf{e}_c)$ as,

$$K_{\text{cbf}}(\mathbf{e}_c) = \left\{ \mathbf{u} \in U : \mathbf{J}_{e_{\delta_{ij}}} \mathbf{u} + \alpha(\mathbf{e}_{\delta_{ij}}) \geq 0 \right\}. \quad (8.32)$$

The above inequality condition obtained from the application of theory on CBF can be utilized for providing safety from collision with obstacle while utilizing proximity sensors.

8.5 Conclusion

In this chapter we have considered the modelling of obstacle avoidance task using proximity sensor. Classical idea in proximity-based control was using obstacle avoidance primitive or through definition of potential-field like methods. In the above modelling approach we have emphasized on the estimation of target normal that would enable in operating in congested space among multiple obstacles. Additionally, the choice of CBF enables the robot to be in virtual contact and thereby at predefined threshold distance. In the next chapter we combine the obstacle avoidance task considered through proximity sensor with a positioning task implemented through vision sensor to achieve positioning

in congested space.

POSITIONING IN CONGESTED SPACE

9.1 Introduction

In this chapter, we consider positioning in congested space within the framework of SBC using vision and proximity sensors. Vision acts as primary sensing modality for performing the task considered, while proximity sensors try to complement it by ensuring that the robotic platform does not collide with objects in the workspace. Proximity sensors can be imagined to share the responsibility of enabling the completion of primary task by ensuring safety.

In the following sections, we give a brief introduction for the use of vision in closed-loop control from the context of Visual Servoing. After that we provide control strategy used for positioning in tight spaces. The methodology developed is validated through experimentation and simulation. Finally, we end the chapter with brief conclusion and discussion on future work.

9.2 Visual Servo Control

The objective of visual servo control is to utilize information from camera in closed-loop feedback control for performing a task using robot [Chaumette & Hutchinson, 2006]. Classical visual servoing utilizes information of geometric primitives such as points, lines etc. which are broadly called as image features. In this section, we assume that the camera is attached rigidly to the end-effector of a robot, which corresponds to the *eye-in-hand* case. Change in sensor feature is either due to the motion of camera or due to the motion of the target from which the image feature is extracted. Additionally, we also assume motionless target or its motion to be negligible enough to be ignored.

Time derivative of image feature denoted by symbol \mathbf{s}_c of dimension m is mapped to

the spatial velocity of the camera (\mathbf{v}_c of dimension 6) using an Interaction matrix \mathbf{L}_{s_c} ,

$$\dot{\mathbf{s}}_c = \mathbf{L}_{s_c} \mathbf{v}_c \quad (9.1)$$

The above quantities are all represented in the camera frame \mathcal{F}_c . When $m \leq 6$, we can control upto m DoF of camera motion, depending on the rank of interaction matrix \mathbf{L}_{s_c} . There are also situation where we use redundant number of features $m > 6$, which enables us to control a maximum of 6 DoF of camera motions.

In order to perform an action with a robot using camera, a task function [Samson et al., 1991] is defined based on image features. Its regulation to origin in m-dimensional space would lead to the fulfillment of the task. Simplest possible definition is a linear function of image features,

$$\mathbf{e}_c = \mathbf{s}_c - \mathbf{s}_c^* \quad (9.2)$$

whose first order derivative can be written as,

$$\dot{\mathbf{e}}_c = \mathbf{L}_{e_c} \mathbf{v}_c \quad (9.3)$$

The regulation of task function can be achieved in the simplest possible way by an exponential decay with rate constant λ ($\dot{\mathbf{e}}_c^* = -\lambda \mathbf{e}_c$), using the following velocity controller,

$$\mathbf{v}_c = \mathbf{L}_{e_c}^+ \dot{\mathbf{e}}_c^* = -\lambda \mathbf{L}_{e_c}^+ \mathbf{e}_c \quad (9.4)$$

where Moore-Penrose pseudo inverse ($\mathbf{L}_{e_c}^+$) based controller, tries to obtain a least squares and minimum norm solution for the linear equation, $-\lambda \mathbf{e}_c = \mathbf{L}_{e_c} \mathbf{v}_c$. It is to be noted that we have treated spatial velocity vector as Euclidean, due to its practical advantage in easy of implementation and computation.

Another possible way is to consider a joint state velocity controller [Chaumette & Hutchinson, 2007]. For that, let us also include the Jacobian mapping of the spatial velocity of the camera to joint state \mathbf{q} of dimension n ,

$$\dot{\mathbf{s}}_c = \mathbf{L}_{s_c} {}^c\mathbf{V}_e \mathbf{J}_e \dot{\mathbf{q}} = \mathbf{J}_{s_c} \dot{\mathbf{q}} \quad (9.5)$$

where ${}^c\mathbf{V}_e$ represents the twist transformation from camera frame \mathcal{F}_c to end-effector frame \mathcal{F}_e .

The equivalent equation for task function is,

$$\dot{\mathbf{e}}_c = \mathbf{J}_{e_c} \dot{\mathbf{q}} \quad (9.6)$$

Corresponding velocity controller can be obtained in joint space as,

$$\dot{\mathbf{q}} = -\lambda \mathbf{J}_{e_c}^+ \mathbf{e}_c \quad (9.7)$$

where Moore-Penrose pseudo inverse $\mathbf{J}_{e_c}^+$ based controller, tries to obtain least square and minimum norm solution for linear equation $-\lambda \mathbf{e}_c = \mathbf{J}_{e_c} \dot{\mathbf{q}}$. The least square solution corresponds to the accuracy of regulation task and the minimum norm ensures a sense of least effort. The above controller can also be expressed as a linear least-squares problem [Nocedal & Wright, 2006] [Cherubini & Navarro-Alarcon, 2021] as,

$$\dot{\mathbf{q}} = \underset{\mathbf{u} \in \mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2} \|\dot{\mathbf{e}}_c^* - \mathbf{J}_{e_c} \mathbf{u}\|^2 \quad (9.8)$$

In the next subsection we consider the task of positioning using image points.

9.2.1 4-point Visual Servoing

The most commonly used features in Visual Servoing for positioning task are image points. Observation of a point in the environment through a standard perspective camera

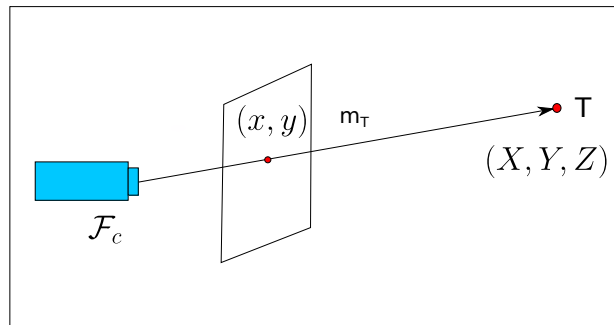


Figure 9.1 – Observation of point through camera

can be modelled in normalized coordinates. As show in Figure 9.1, point T represented in camera coordinate frame \mathcal{F}_c as $\mathbf{X} = (X, Y, Z)$ gets projected in image plane as $\mathbf{x} = (x, y)$.

This relation is depicted in the equation below,

$$\begin{cases} x = \frac{X}{Z} \\ y = \frac{Y}{Z} \end{cases} \quad (9.9)$$

Differentiating the above equation wrt. time allows us obtain the interaction matrix corresponding to an image points [Chaumette & Hutchinson, 2006],

$$\mathbf{L}_x = \begin{bmatrix} \frac{-1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & \frac{-1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{bmatrix} \quad (9.10)$$

Minimal set of features that enables positioning of a camera wrt. a target can be obtained from the observation of 3 points on the target. It is appropriately termed as 3-point Visual Servoing. However, it is not used in a practical scenario due to the existence of four global minima [Fischler & Bolles, 1987] that makes it difficult to differentiate final position as perceived from observation of points. Additionally, the cylinder of singularity [Michel & Rives, 1993], [Briot et al., 2017], reduces the region of operation for a basic velocity controller. The most widely used method is 4-point visual servoing; referring to the observation of four points. Usually these points are chosen to lie in the same plane and as vertices of a rectangle.

The sensor features in this particular case is,

$$\mathbf{s}_{4p} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{bmatrix} \quad (9.11)$$

The interaction matrix for the corresponding task function $\dot{\mathbf{e}}_{4p} = \mathbf{L}_{4p}\mathbf{v}_{4p}$ is,

$$\mathbf{L}_{4p} = \begin{bmatrix} \mathbf{L}_{x_1} \\ \mathbf{L}_{x_2} \\ \mathbf{L}_{x_3} \\ \mathbf{L}_{x_4} \end{bmatrix} \quad (9.12)$$

The linear least-squares representation of the task similar to (9.8) is,

$$\dot{\mathbf{q}} = \underset{\mathbf{u} \in \mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2} \|\dot{\mathbf{e}}_{4p}^* - \mathbf{J}_{4p} \mathbf{u}\|^2 \quad (9.13)$$

where $\mathbf{J}_{4p} = \mathbf{L}_{4p} {}^c \mathbf{V}_e \mathbf{J}_e$

It can also be represented in Quadratic form as,

$$\dot{\mathbf{q}} = \underset{\mathbf{u} \in \mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2} \mathbf{u}^T \mathbf{G} \mathbf{u} + \mathbf{u}^T \mathbf{c} \quad (9.14)$$

where $\mathbf{G} = \mathbf{J}_{4p}^T \mathbf{J}_{4p}$ and $\mathbf{c} = -\mathbf{J}_{4p}^T \dot{\mathbf{e}}_{4p}^*$.

In the next section we combine the positioning task achieved using 4 point visual servoing as discussed here with obstacle avoidance task using proximity sensors described in Section 8.4.2 of Chapter 8.

9.3 Control Architecture

In this section we consider the control architecture to achieve positioning in congested space using vision and proximity sensors. The formulation of positioning as minimization of a quadratic form in previous Section 9.2.1 and obstacle avoidance as linear constraints in Section 8.4.2, naturally leads us to \mathcal{QP} as an ideal framework [Faverjon & Tournassoud, 1987]. The velocity controller needs to ensure minimization of the quadratic cost function that arises from the least squares description of the primary task in image space (9.8), while satisfying linear constraints of safety imposed through CBFs in proximity space (8.32),

$$\begin{aligned} \dot{\mathbf{q}} &= \underset{\mathbf{u} \in \mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2} \mathbf{u}^T \mathbf{G} \mathbf{u} + \mathbf{u}^T \mathbf{c} \\ \text{s.t.} \quad & \mathbf{a}_k^T \mathbf{u} \geq b_k \quad k \in \mathcal{I}_p \\ & \mathbf{u}_{upper} \geq \mathbf{u} \geq \mathbf{u}_{lower} \quad \mathcal{I}_{limits} \end{aligned} \quad (9.15)$$

where,

$$\begin{aligned}
 \mathbf{G} &= \mathbf{J}_{4p}^T \mathbf{J}_{4p} + \mu \mathbf{I} \\
 \mathbf{c} &= -\mathbf{J}_{4p}^T \dot{\mathbf{e}}_{4p}^* \\
 \dot{\mathbf{e}}_{4p}^* &= -\lambda_{4p} \mathbf{e}_{4p} \\
 \mathbf{a}_k &= \mathbf{J}_{e_{\delta_{ij}}}^T \\
 b_k &= -\alpha(\mathbf{e}_{\delta_{ij}}) \\
 \alpha(\mathbf{e}_{\delta_{ij}}) &= \lambda_j (\mathbf{e}_{\delta_{ij}})^l \\
 k &= (j-1) * m_j + i, \quad k \in \mathcal{I}_p
 \end{aligned} \tag{9.16}$$

Here $\mathbf{J}_{4p}^T \mathbf{J}_{4p}$ is a symmetric $n \times n$ matrix. It is composed of feature Jacobian matrix \mathbf{J}_{4p} , that is of rank 6 in case of redundant robots with $n > 6$ joints. In this situation, matrix $\mathbf{J}_{4p}^T \mathbf{J}_{4p}$ is a semi-definite matrix throughout execution which makes it a convex \mathcal{QP} problem [Sharma & Hutchinson, 1997]. For getting a well-conditioned solution, a regularization term μ is added to cost function, which makes $\mathbf{G} = \mathbf{J}_{4p}^T \mathbf{J}_{4p} + \mu \mathbf{I}$, positive definite. A detailed explanation for this is provided in Section 9.4. The linear part of the quadratic term $\mathbf{c} \in \mathbb{R}^n$ consists of task function in image space and becomes minimal at convergence. \mathcal{I}_p indicates the inequality constraints from proximity sensors. We consider a proximity array as described in Chapter 3, consisting of j rings with each carrying m_j sensors. A generic proximity sensor denoted as i th sensor on the ring j is denoted here by index k . Therefore we have in total $j * m_j$ inequalities ensuring anti-collision. At a given instant, only few of these constraints where the sensors are in close encounter with obstacle are active and the remaining constraints stay inactive. The candidate $\alpha(\cdot)$ is chosen here of the form $\lambda_j \mathbf{e}_{\delta_{ij}}^l$, where l is a positive odd integer. The remaining inequalities \mathcal{I}_{limits} are from the joint velocity limits, where $\mathbf{u}_{upper} \in \mathbb{R}^n$ indicated its upper limit and $\mathbf{u}_{lower} \in \mathbb{R}^n$ indicates the lower limits. In the following analysis of the \mathcal{QP} problem, we ignore the constraints from upper and lower bounds on joint velocity. It can also be noted that such bounds are not reached during Experimentation and Simulation results. This enables us to focus on the influence of proximity and vision sensors towards the execution of the task. Total inequality constraints is denoted as $\mathcal{I} = \mathcal{I}_p + \mathcal{I}_{limits}$.

9.3.1 Preliminaries

Now, we define certain terminology and sets that are necessary to understand optimality conditions and discuss the \mathcal{QP} problem for certain specific scenarios. These definitions are all taken from [Nocedal & Wright, 2006] and adapted for the problem considered.

We first define the feasible set Ω ,

Definition 4 *Feasible set Ω is the set of points that satisfy the constraints*

$$\Omega = \left\{ \mathbf{u} \mid \mathbf{a}_k^T \mathbf{u} \geq b_k, \quad k \in \mathcal{I}_p, \quad \mathbf{u}_{upper} \geq \mathbf{u} \geq \mathbf{u}_{lower} \right\} \quad (9.17)$$

For the problem considered, feasible set consists of $K_{\text{cbf}}(\mathbf{e}_c)$ along with constraints on joint limits.

Next we define active set. Here we only consider inequalities from proximity sensors

Definition 5 *The active set $\mathcal{A}(\mathbf{u})$ at any feasible point \mathbf{u} consists of the indices of inequality constraints that have become equalities, i.e.,*

$$\mathcal{A}(\mathbf{u}) = \left\{ k \in \mathcal{I}_p \mid \mathbf{a}_k^T \mathbf{u} = b_k \right\}. \quad (9.18)$$

Now we introduce the idea of constrained qualification that is required for framing the optimality conditions. These are sufficient conditions which ensures that the key geometrical aspects of the feasible set Ω is captured by its linearized algebraic characterization in the neighbourhood of the solution [Nocedal & Wright, 2006]. Linearity of constraints is an adequate constraint qualification. Therefore optimality conditions can be directly applied for the \mathcal{QP} problem considered. In addition, it is also desirable to have these constraints to be linearly independent. This constraint qualification is denoted as linear independence constraint qualification (\mathcal{LICQ}) and can be defined as

Definition 6 *Given a point \mathbf{u} and active set $\mathcal{A}(\mathbf{u})$, \mathcal{LICQ} holds if the set of active constraint gradients $\{\mathbf{a}_k, k \in \mathcal{A}(\mathbf{u})\}$ is linearly independent*

\mathcal{LICQ} is neither weak nor strong compared to the constraints being linear. However, it enables us to have a good analysis on the stability of solution as described in Section 9.4. Also, we observe \mathcal{LICQ} being held in practical experimentation and simulation.

Let us defines the Lagrangian of \mathcal{QP} problem that is required for deriving optimality conditions,

$$\mathcal{L}(\mathbf{u}, \kappa) = \frac{1}{2} \mathbf{u}^T \mathbf{G} \mathbf{u} + \mathbf{u}^T \mathbf{c} - \sum_{k \in \mathcal{I}_p} \kappa_k \left(\mathbf{a}_k^T \mathbf{u} - b_k \right). \quad (9.19)$$

Here κ_k are called the Lagrangian multipliers of the corresponding inequality constraint.

9.3.2 Optimality

In this section we discuss the optimality conditions for the \mathcal{QP} problem. First we consider the first order necessary conditions (\mathcal{FONC}) [Nocedal & Wright, 2006],

Theorem 3 *If \mathbf{u}^* is a local solution of (9.15), and constrained qualification holds at \mathbf{u}^* , then there is a Lagrangian multiplier vector $\boldsymbol{\kappa}^*$, with components κ_k^* , $k \in \mathcal{A}(\mathbf{u}^*)$, such that the following conditions are satisfied at $(\mathbf{u}^*, \boldsymbol{\kappa}^*)$,*

$$\begin{aligned} \nabla_{\mathbf{u}} \mathcal{L}(\mathbf{u}^*, \boldsymbol{\kappa}^*) &= \mathbf{G}\mathbf{u}^* + \mathbf{c} - \sum_{k \in \mathcal{A}(\mathbf{u}^*)} \kappa_k^* \mathbf{a}_k = 0, \\ \mathbf{a}_k^T \mathbf{u}^* &= b_k, \quad \forall k \in \mathcal{A}(\mathbf{u}^*), \\ \mathbf{a}_k^T \mathbf{u}^* &\geq b_k \quad \forall k \in \mathcal{I}_p \setminus \mathcal{A}(\mathbf{u}^*), \\ \kappa_k^* &\geq 0, \quad \forall k \in \mathcal{A}(\mathbf{u}^*). \end{aligned} \tag{9.20}$$

The above equations are known popularly as *Karush-Kahn-Tucker (KKT)* conditions. In case of convex \mathcal{QP} (such as the problem considered) if \mathcal{LICQ} holds \mathcal{FONC} are sufficient to be a global minimizer [Gros & Diehl, 2022]. Let us define strict complementarity that is going to be used while considering second order optimality conditions

Definition 7 *Given a KKT point $(\mathbf{u}^*, \boldsymbol{\kappa}^*)$ we say that the strict complementarity condition holds if constraints in active set have positive Lagrangian multipliers i.e. $(\kappa_k^* > 0, \forall k \in \mathcal{A}(\mathbf{u}^*))$.*

Next we consider the second order conditions that provides sufficiency conditions for a generic non-linear optimization problem. It is discussed here as it becomes important in considering the stability of the solutions in a particular case as considered in Section 9.4. We consider the second order conditions in case of strict complementarity for a solution satisfying \mathcal{KKT} conditions. We adapt this from [Gros & Diehl, 2022] for the problem considered. Before proceeding ahead, let us consider an active constraint matrix ($\mathbf{A}_{\mathcal{A}}$) defined as,

$$\mathbf{A}_{\mathcal{A}} = \begin{bmatrix} \cdot \\ \mathbf{a}_k^T \\ \cdot \end{bmatrix} \quad \text{if } k \in \mathcal{A}(\mathbf{u}^*) \tag{9.21}$$

\mathbf{Z} is defined as the basis matrix for the null space of \mathbf{A}_A . In other words, \mathbf{Z} has full column rank and $\mathbf{A}_A \mathbf{Z} = 0$.

Theorem 4 *Let us consider that FONC are satisfied at a KKT point (\mathbf{u}^*, κ^*) and let strict complementarity holds. Let us consider a basis matrix \mathbf{Z} of the null space of \mathbf{A}_A . Then the following two statements hold:*

1. *If \mathbf{u}^* is a local minimizer, then $\mathbf{Z}^T \mathbf{G} \mathbf{Z} \geq 0$.*

(SONC : Second Order Necessary Condition)

2. *If $\mathbf{Z}^T \mathbf{G} \mathbf{Z} > 0$, then \mathbf{u}^* is a local minimizer. This minimizer is unique in its neighborhood, i.e a strict local minimizer.*

(SOSC : Second Order Sufficiency Condition)

Without the regularity term, Hessian matrix \mathbf{G} is semi-definite. Satisfaction of first order conditions is sufficient for solution of \mathcal{QP} to be global. Additionally, with regularization \mathbf{G} becomes positive definite and then the solution is a unique global solution.

In the next subsection we consider the stopping conditions used during task execution.

9.3.2.1 Stopping Condition

Here we consider two criteria that are effective in two distinct situations. In the first case, the global minimum of task function in vision space is reached. For this we evaluate if the norm of task function \mathbf{e}_{4p} is below a low threshold (ϵ_{4p}) and in that case we consider the positioning task to have converged and the robot can be stopped. The second case is where the obstacles prevent positioning task from converging to global minimum. In those situations we stop by using the norm of gradient of Lagrangian with the solution at $\mathbf{u}^* = 0$,

$$\|\nabla_{\mathbf{u}} \mathcal{L}(\mathbf{0}, \kappa^*)\| = \left\| -\mathbf{J}_{4p}^T \dot{\mathbf{e}}_{4p}^* - \sum_{i=\mathcal{A}(\mathbf{u}^*)} \kappa_i^* \mathbf{a}_i \right\| \leq \epsilon_{qd} \quad (9.22)$$

The stopping condition essentially indicates that the task function for positioning cannot be further minimized as the constraints are blocking it. The stopping conditions considered can be summarized as,

if $(\|\mathbf{e}_{4p}\| \leq \epsilon_{4p}) \quad \vee \quad (\|\nabla_{\mathbf{u}} \mathcal{L}(\mathbf{0}, \kappa^*)\| \leq \epsilon_{qd})$ **then**

Stop Task

else

Solve \mathcal{QP}

end if

In the following subsections we analyze few cases to consider the optimal solution of the \mathcal{QP} at the end of task execution.

9.3.2.2 Case 1 : Empty Active Set

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

$$\dot{\mathbf{q}} = -\lambda_{4p}(\mathbf{J}_{4p}^T \mathbf{J}_{4p} + \mu \mathbf{I})^{-1} \mathbf{J}_{4p}^T \mathbf{e}_{4p} \quad (9.23)$$

In this situation it is not necessary to have the regularization term. The solution would then become the same as (9.7).

9.3.2.3 Case 2 : Two parallel planes

Let us consider a scenario where a proximity ring j interacts with two parallel planes on either side. The goal position from visual servoing lies in between the parallel planes. While performing this forward motion let us assume the minimal case of two detections from two proximity sensors on this ring, one per plane. Additionally, we consider that each of these proximity sensors on j th ring are in virtual contact i.e. $\mathbf{e}_{\delta_{1j}} = \mathbf{e}_{\delta_{2j}} = 0$. We perform analysis relative to sensor frame attached to the j th ring which is rigidly attached to link N_j . This scenario can be considered as a situation similar to the case of ball rolling in a frictionless passage as in Figure 9.2. In this case the active set would consist of the

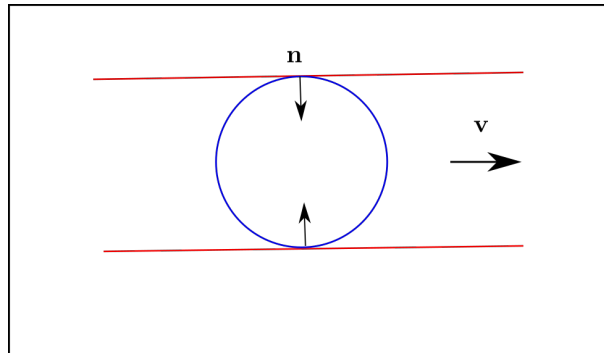


Figure 9.2 – Ball rolling inside a passage. Equivalent scenario from contact mechanics

two detections,

$$\mathcal{A}(\mathbf{u}^*) = \begin{cases} \mathbf{J}_{e_{\delta_{1j}}}^T \mathbf{u}^* = 0 \\ \mathbf{J}_{e_{\delta_{ij}}}^T \mathbf{u}^* = 0 \end{cases} \quad (9.24)$$

The Lagrangian multipliers corresponding to the active set is zero and the term $\sum_{k \in \mathcal{A}(\mathbf{u}^*)} \kappa_k^* \mathbf{a}_k$ in \mathcal{KKT} condition vanishes. Therefore the optimal solution can be obtained by solving the gradient of Lagrangian,

$$\begin{aligned} (\mathbf{J}_{4p}^T \mathbf{J}_{4p} + \mu \mathbf{I}) \mathbf{u}^* - \mathbf{J}_{4p}^T \dot{\mathbf{e}}_{4p}^* - \sum_{k \in \mathcal{A}(\mathbf{u}^*)} \kappa_k^* \mathbf{a}_k &= (\mathbf{J}_{4p}^T \mathbf{J}_{4p} + \mu \mathbf{I}) \mathbf{u}^* - \mathbf{J}_{4p}^T \dot{\mathbf{e}}_{4p}^* \\ &= \mathbf{0} \end{aligned} \quad (9.25)$$

which gives us the optimal solution in (9.23) which was the case of empty active set. This essentially tells us that it is possible to be at the desired threshold distance from each of the planes without experiencing oscillations.

9.3.2.4 Case 3 : Two blocking planes

Let us now consider a case with two blocking planes that are arranged at an angle with the intention of preventing forward motion. Let us again assume that visual servoing is making the ring move forward. Let us also consider the minimal case of two detections from two proximity sensors attached to ring j one on each plane. Let us consider the moment in which the blocking planes result in virtual contact. This scenario can be considered as a situation similar to the case of ball rolling in a frictionless passage with its width reducing as in Figure 9.3.

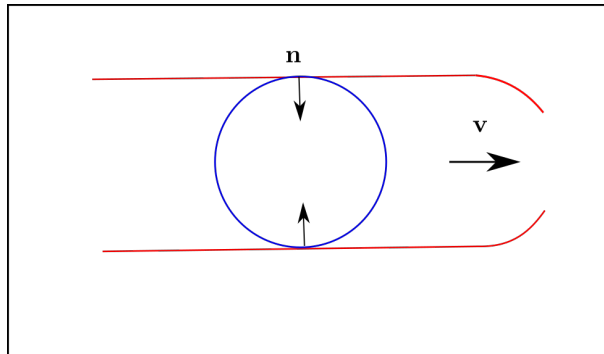


Figure 9.3 – Ball rolling inside a passage with width decreasing. Equivalent scenario from contact mechanics

In this analysis we show that a feasible solution exists that causes the forward motion to halt. So, let us imagine that this proximity ring halts when having virtual contact with the blocking planes, i.e. $\mathbf{u}^* = 0$. The active set again consists of the equalities associated with the two proximity sensors,

$$\mathcal{A}(\mathbf{u}^*) = \begin{cases} \mathbf{J}_{e\delta_{1j}}^T \mathbf{u}^* = 0 \\ \mathbf{J}_{e\delta_{ij}}^T \mathbf{u}^* = 0 \end{cases} \quad (9.26)$$

From the \mathcal{KKT} conditions (9.20), we can see that the Lagrangian multipliers would choose a suitable value to resist the term associated with the vision task function,

$$\mathbf{J}_{4p}^T \dot{\mathbf{e}}_{4p}^* = \sum_{i \in \mathcal{A}(\mathbf{u}^*)} \kappa_i^* \mathbf{a}_i \quad (9.27)$$

Let us consider the case when both camera and proximity ring are both rigidly attached to the end-effector of the robot. We have $\mathbf{J}_{4p} = \mathbf{L}_{4p} {}^c\mathbf{V}_e {}^e\mathbf{J}_e$ then we can simplify the equation as,

$$(\mathbf{L}_{4p} {}^c\mathbf{V}_e \mathbf{J}_e)^T \dot{\mathbf{e}}_{4p}^* = \kappa_1 (\mathbf{L}_{\delta_{1j}} {}^j\mathbf{V}_e \mathbf{J}_e)^T + \kappa_2 (\mathbf{L}_{\delta_{2j}} {}^j\mathbf{V}_e \mathbf{J}_e)^T \quad (9.28)$$

Re-arranging the equations we get,

$$\mathbf{J}_e^T \left((\mathbf{L}_{4p} {}^c\mathbf{V}_e)^T \dot{\mathbf{e}}_{4p}^* - \kappa_1 (\mathbf{L}_{\delta_{1j}} {}^j\mathbf{V}_e)^T - \kappa_2 (\mathbf{L}_{\delta_{2j}} {}^j\mathbf{V}_e)^T \right) = 0 \quad (9.29)$$

The above equation can be interpreted as a linear combination of interaction screws associated with four-point visual servoing being balanced by the linear combination of the interaction screws associated with proximity sensors when expressed in end-effector frame. In case of vision sensor, the linear combination is achieved by the desired evolution of task function in image space, whereas for the case of proximity sensors it is through the Lagrangian multipliers. In Section 9.4.1 we show that such a solution is stable wrt. uncertainties in interaction matrices and measurement noise involved with vision sensors and with measurement noise involved with proximity sensors.

Let us consider the case when a bad approximation or estimation of target normal results in interaction matrices associated with proximity sensors $\mathbf{L}_{\delta_{1j}}$ and $\mathbf{L}_{\delta_{2j}}$ to be linearly dependent. In such a situation, Lagrangian multipliers would not be able to balance the term associated to image task function i.e. $\mathbf{J}_{4p}^T \dot{\mathbf{e}}_{4p}^*$. This would result in an impulsive

motion or large oscillation while performing experimentation.

9.3.2.5 Case 4 : Global convergence with active proximity sensor

In the final case let us consider the generic situation where at-least one proximity sensor is active while achieving convergence in vision space. At convergence we can substitute $\mathbf{u}^* = 0$ and $\dot{\mathbf{e}}_{4p}^* = 0$ to the gradient of Lagrangian to obtain weak complementarity condition, i.e. $\kappa_k^* = 0$ when \mathcal{LICQ} holds,

$$\begin{aligned} (\mathbf{J}_{4p}^T \mathbf{J}_{4p} + \mu \mathbf{I}) \mathbf{u}^* - \mathbf{J}_{4p}^T \dot{\mathbf{e}}_{4p}^* - \sum_{k \in \mathcal{A}(\mathbf{u}^*)} \kappa_k^* \mathbf{a}_k &= - \sum_{k \in \mathcal{A}(\mathbf{u}^*)} \kappa_k^* \mathbf{a}_k \\ &= \mathbf{0} \end{aligned} \quad (9.30)$$

In such a situation we can obtain stability of solutions, with more discussion provided in Section 9.4.2. This situation is considered in experimentation Section 9.5.4 and in simulation Sections 9.6.1 and 9.6.2.

9.4 Stability Analysis

9.4.1 Specific Scenario

In the specific scenario we consider a situation discussed in Section 9.3.2.4, where strict complementarity holds. In those situations, optimization can be reduced to an equality constrained problem involving the active set locally. In this situation we prove stability of optimal solution to model uncertainties and measurement noise in vision space and measurement noise in proximity space which in turn corresponds to perturbations in terms \mathbf{c} and b_k respectively in (9.15). From this analysis we would like to confirm about the solution obtained from the control architecture in certain situations such as the case discussed in Section. 9.3.2.4 involving the use of both vision and proximity while at final configuration,

$$\begin{aligned} \dot{\mathbf{q}} &= \underset{\mathbf{u} \in \mathbb{R}^m}{\operatorname{argmin}} \frac{1}{2} \mathbf{u}^T \mathbf{G} \mathbf{u} + \mathbf{u}^T (\mathbf{c} + \delta_c) \\ \text{s.t.} \quad & \mathbf{a}_k^T \mathbf{u} \geq b_k + \delta_{p_k} \quad k \in \mathcal{I}_p \end{aligned} \quad (9.31)$$

Let us denote these uncertainties in combined vector $\delta_{qp} = (\delta_c, \delta_p)$ with appropriate dimensions. We use results that proves stability to such perturbation from $\mathcal{SO}\mathcal{SC}$. This theorem is adapted from [Golub & Van Loan, 2013] to the problem at hand

Theorem 5 *Consider the family of perturbed optimization problems (9.31) and assume that for $\delta_{qp} = 0$ there exists a local solution $(\mathbf{u}^*(0), \kappa^*(0))$ that satisfies \mathcal{LICQ} , \mathcal{FONC} , strict complementarity, and $\mathcal{SO}\mathcal{SC}$. Then there exists an $\epsilon > 0$ so that for all $\|\delta_{qp}\| \leq \epsilon$ it exists a unique local solution $(\mathbf{u}^*(\delta_{qp}), \kappa^*(\delta_{qp}))$ that depends differentiably on δ_{qp} . This local solution has the same active set as the nominal one. If $\mathbf{A}_{\mathcal{A}}$ is the active constrained matrix and $\tilde{\delta}_p$ the active constraint perturbations, then the derivative of the solution $(\mathbf{u}^*(\delta_{qp}), \kappa^*(\delta_{qp}))$ with respect to $(\delta_c, \tilde{\delta}_p)$ is given by,*

$$\frac{d}{d(\delta_c, \tilde{\delta}_p)} \begin{bmatrix} \mathbf{u}^*(\delta_{qp}) \\ \tilde{\kappa}^*(\delta_{qp}) \end{bmatrix} \Big|_{\delta_{qp}=0} = - \begin{bmatrix} \mathbf{G} & \mathbf{A}_{\mathcal{A}}^T \\ \mathbf{A}_{\mathcal{A}} & \mathbf{0} \end{bmatrix}^{-1} \quad (9.32)$$

The above results are guaranteed for the problem considered by including a regularization term μ to ensure the satisfaction of $\mathcal{SO}\mathcal{SC}$. Also the \mathcal{KKT} matrix is a class of symmetric indefinite system which is invertible if matrix \mathbf{G} is symmetric positive definite and $\mathbf{A}_{\mathcal{A}}^T$ has full column rank [Golub & Van Loan, 2013]. Since \mathcal{LICQ} condition is assumed to be satisfied we have full column rank for $\mathbf{A}_{\mathcal{A}}^T$. Additionally, a further analysis on the gradient of cost function can also be considered,

$$\frac{d}{d\delta_{qp}} \left(\frac{1}{2} \mathbf{u}^T \mathbf{G} \mathbf{u} + \mathbf{u}^T \mathbf{c} \right) \Big|_{\delta_{qp}=0}^{\top} = - \begin{bmatrix} \mathbf{G} & \mathbf{A}_{\mathcal{A}}^T \\ \mathbf{A}_{\mathcal{A}} & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{G} \mathbf{u}^* + \mathbf{c} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{\kappa}^* \end{bmatrix} \quad (9.33)$$

where $\tilde{\kappa}^*$ corresponds to Lagrangian multipliers of active set. The above result enables us to draw two conclusions that are important for practical experimentation. The first inference is that the objective function is insensitive to the perturbation δ_c considered. The second inference is that perturbations $\tilde{\delta}_p$ from proximity sensors in active set causes changes in cost function. However this change is proportional to the corresponding Lagrangian multipliers $\tilde{\kappa}^*$ which is restricted to be positive. Positive perturbations in proximity space would result in tightening of inequality and the increase of objective function. From the above results we can conclude that if the conditions are satisfied then in situations where obstacles prevent execution of the primary task such as the case discussed in Section 9.3.2.4 the robot would stop and also not cause explosive behavior for subsequent control loops.

9.4.2 General Scenario

In Section 9.3.2.5 and 9.3.2.3, it was pointed out that weak complementarity occurs in situations when proximity sensor is in virtual contact and \mathcal{LICQ} holds. In this section we consider more general conditions that proves stability of solutions to perturbations to the linear part of \mathcal{QP} . Let us consider the \mathcal{QP} problem in (9.15) written in a more concise way,

$$\begin{aligned} \dot{\mathbf{q}} &= \underset{\mathbf{u} \in \mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2} \mathbf{u}^T \mathbf{G} \mathbf{u} + \mathbf{u}^T \mathbf{c} \\ \text{s.t.} \quad & \mathbf{A}_p \mathbf{u} \geq \mathbf{b} \end{aligned} \tag{9.34}$$

where $\mathbf{A}_p = \begin{bmatrix} \cdot \\ \mathbf{a}_k^T \\ \cdot \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} \cdot \\ b_k \\ \cdot \end{bmatrix}$. Let us consider a perturbation of $(\mathbf{A}_p^*, \mathbf{b}^*, \mathbf{c}^*)$ of appropriate dimensions. For a given non-negative parameter δ_g we have the following perturbed problem,

$$\begin{aligned} \dot{\mathbf{q}} &= \underset{\mathbf{u} \in \mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2} \mathbf{u}^T \mathbf{G} \mathbf{u} + \mathbf{u}^T (\mathbf{c} + \delta_g \mathbf{c}^*) \\ \text{s.t.} \quad & (\mathbf{A}_p + \delta_g \mathbf{A}_g^*) \mathbf{u} \geq \mathbf{b} + \delta_g \mathbf{b}^* \end{aligned} \tag{9.35}$$

This is a useful analysis as these perturbations can be considered from model uncertainties and measurement noise from both vision and proximity sensors. The model uncertainty for proximity sensor arises from the estimation target normal term in the interaction matrix. Stability of the solution in this situation corresponds to,

Definition 8 *The quadratic program is defined to be stable if for any perturbation $(\mathbf{A}_p^*, \mathbf{b}^*, \mathbf{c}^*)$, there exists an $\epsilon > 0$ such that the perturbed problem has an optimal solution for all $\delta_g, 0 \leq \delta_g \leq \epsilon$*

The stability of solution for such general practical perturbations is proved by ensuring that the \mathcal{QP} problem satisfies two regularity conditions. These conditions given in [Best & Chakravarti, 1990] are presented below by adapting to the problem considered.

Definition 9 *Condition R1: There does not exist a nonzero vector \mathbf{z} satisfying the following condition*

$$\begin{aligned}
\mathbf{Gz} &= 0 \\
\mathbf{A}_p \mathbf{z} &\geq 0 \\
\mathbf{c}^T \mathbf{z} &\leq 0
\end{aligned} \tag{9.36}$$

Definition 10 *Condition R2: There do not exist vectors \mathbf{z} and \mathbf{w} , such that $(\mathbf{z}, \mathbf{w}) \neq \mathbf{0}$, satisfying the following condition:*

$$\begin{aligned}
\mathbf{Gz} &= 0 \\
\mathbf{A}_p^T \mathbf{w} &= 0 \\
\mathbf{b}^T \mathbf{w} &= 0 \\
\mathbf{w} &\geq 0
\end{aligned} \tag{9.37}$$

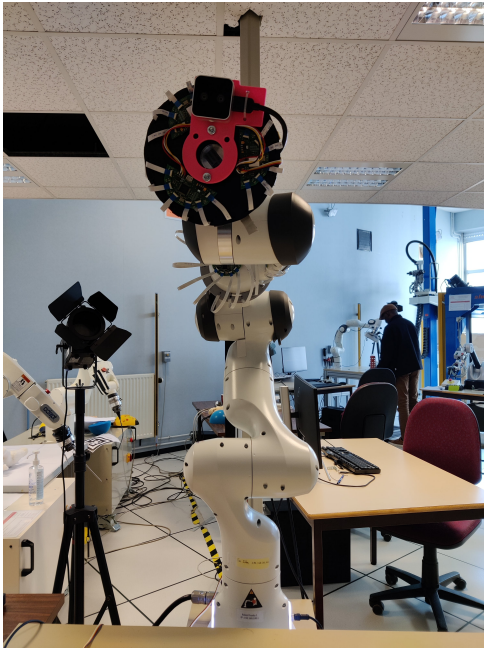
\mathcal{QP} is considered to be regular if it satisfies both R1 and R2 conditions. Let us restate the theorem present in [Best & Chakravarti, 1990]

Theorem 6 *\mathcal{QP} is stable to linear perturbations if and only if it is regular.*

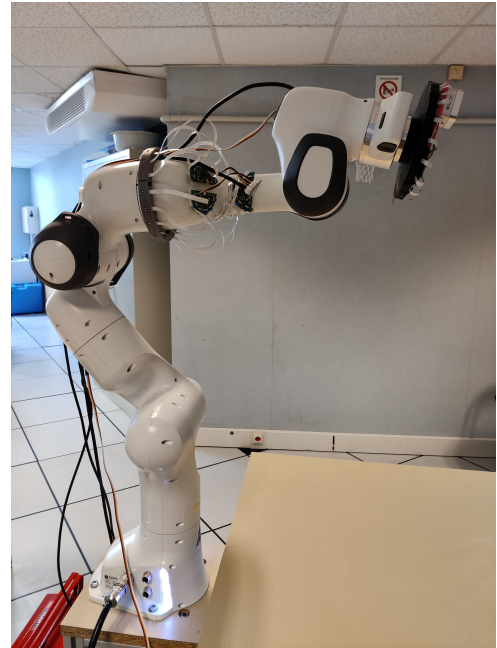
Here if \mathbf{G} is positive definite then we satisfy R1. For condition R2 we only need to concentrate on the inequalities belonging to active set. If \mathcal{LICQ} holds for those then we have $\mathbf{A}_p^T \mathbf{w} \neq 0$ if $\mathbf{w} \neq 0$ and so we satisfy condition R2. This would mean that positioning task in situation in Case 1 (Section 9.5.1), Case 2 (Section 9.5.4) in experimentation and Case 1 (Section 9.6.1), Case 2 (Section 9.6.2) in simulation are both stable to linear perturbations. However it is to be noted that in Case 1 (Section 9.5.1) large perturbations could lead to infeasible solution.

9.5 Experimentation

Let us consider the experimental scenario as shown in Figure 9.4. It consists of the same Panda robot used in Part I of this manuscript. To the robot, we attach two rings each carrying 18 proximity sensors. Ring 1 which is grey in color is mounted to link 5 and ring 2 which is black in color is attached to the flange of robot and act as the end-effector. On top of ring 2, a D405 Intel Realsense camera is mounted. The gain of positioning task is chosen as $\lambda_{4p} = 0.4$ and gains for all proximity sensors are set to $\lambda_j = 1$ for each ring ($j = 1, 2$) except in Cases 5 and 6. The threshold distance δ_{ij}^* is set to a value of 4 cm



(a) Front View



(b) Side View

Figure 9.4 – Serial manipulator with camera and proximity array

in all cases, which is also the minimum detectable distance of the sensor. If there is no detection, the value of the proximity ring is set to maximum detection of 1 m . Thereby we have setup the workspace in such a manner that it leads to very close encounters with obstacles. Regularization term in the cost function of QP is selected as $\mu = 0.01$. In all the cases we use the approximation of target normal as mentioned in Section 8.3.3. As already explained, such an approximation becomes unsuitable in situation when there is a large angular deviation between the surface normal at the point of detection and the negative direction of the axis of corresponding proximity sensors. In Case 2 (Section 9.5.2), where such a situation occurs for ring 2, we use target normal estimation using the method described in Section 8.3.1 with multiple sensors.

ViSP [Marchand et al., 2005] functionalities are used in the implementation of control architecture. Numerical solver was chosen as ProxQP which provided fast and efficient C++ based implementation designed for modern robotic applications [Bambade et al., 2022]. It uses Eigen library¹ for linear algebra part and this enables easy conversion to ViSP data structure. The inner control loop for updating joint velocity $\dot{\mathbf{q}}$ was run at 30 Hz with an outer loop sending these velocities to Panda at 1 kHz satisfying the

1. Eigen v3, <https://eigen.tuxfamily.org>

communication constraints. At each control loop, \mathcal{QP} solver is warm started with previous results. It was executed on a laptop with Intel® Core™ i7 CPU @ 1.90GHz \times 8 which on Ubuntu 20.04.2 LTS with RTLinux kernel in Fully Preemptible Mode. The video of experimentation for each of the cases below can be seen here.

9.5.1 Case 1 : Two parallel planes

Let us consider the scenario where two parallel planes are placed on either side of the final goal position as shown in Figure 9.5. On each side we have two cardboard boxes that together indicate the plane. The cardboard boxes are placed close to the manipulator, so that few sensors on each ring has proximity signal measure near the threshold distance of 4 *cm*. The final goal position is placed farther away so that these detections close to the threshold distance remain so for a long duration. Any motion towards the planes would result in violation of at least one safety constraint associated with the proximity measurements. This experimental scenario can be considered as a situation similar to the case of ball rolling in a frictionless passage as depicted in Figure 9.2.

In Figure 9.6, we plot the results of the experimentation. The top left subplot indicates visual task function \mathbf{e}_{4p} , top right indicates control input $\dot{\mathbf{q}}$, bottom plots indicates CBFs for proximity sensors on both rings $\mathbf{e}_{\delta_{ij}}$. The structure of the plot also remains similar for the other cases considered below. The task is considered to be converged when $\|\mathbf{e}_{4p}\| \leq 0.0005$. As observed in Figure 9.6, the visual task function converges below the threshold and this also results in the convergence of control input to zero norm. The proximity task function in Figure 9.6 (c) and (d) indicates that few sensors are very close to achieving virtual contact while passing between the parallel planes. For ring 1, it includes proximity sensors {17, 18, 1} and {9, 8} on diametrically opposite side. For ring 2, it includes sensors {2, 3, 4} and {11, 12} on its either side. Due to the choice of control design, the robot is able to achieve the positioning task while being very close to the threshold distance without encountering oscillations. It can be observed from the smoothness of control input graph in Figure 9.6 (b). It can be seen from the remaining cases that while encountering obstacles there is abrupt change in control velocity.

9.5.2 Case 2 : Two blocking planes

In this scenario, we rotate the two cardboard boxes closer to the final goal from the previous scenario to stop convergence of \mathbf{e}_{4p} before its global minimum. This can be seen in

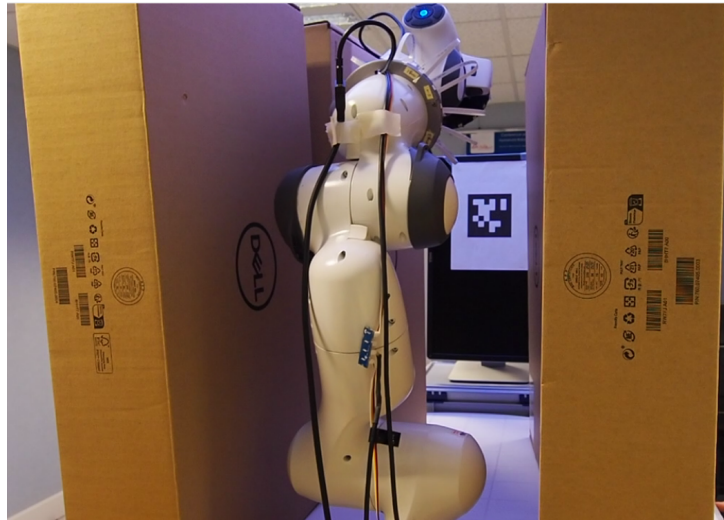


Figure 9.5 – Parallel planes placed on either side of the robot

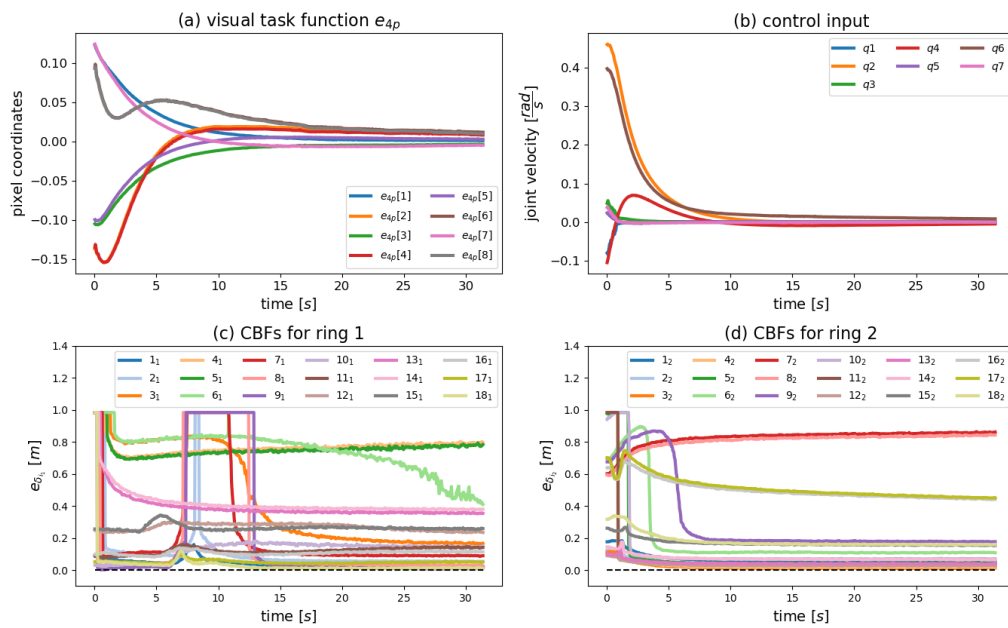


Figure 9.6 – Plots for Case 1 : Parallel Planes



Figure 9.7 – Two blocking planes placed on either side of the robot

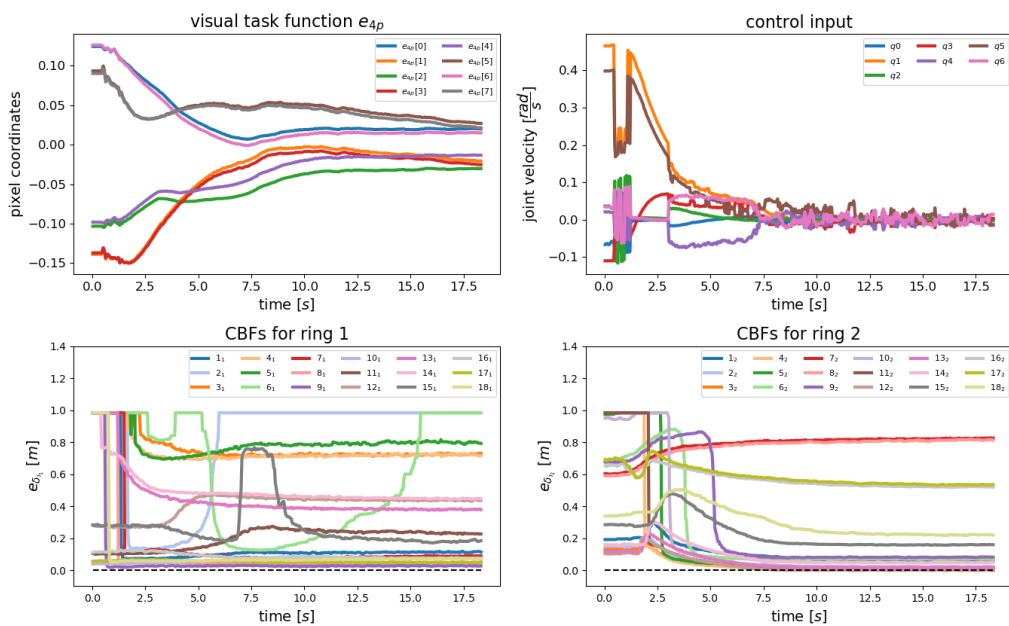


Figure 9.8 – Plots for Case 2 : Planes preventing positioning

Figure 9.7. The additional planes prevent positioning task from converging and the robot is required to come to a halt for providing safety. A contrary behavior could lead to collision with the obstacles before obtaining convergence in positioning. This setup is chosen to validate the analysis in Section 9.3.2.4 and Section 9.4.1. We use the estimated value of target normal for sensors on ring 2 while evaluating the interaction matrix involved.

As observed in Figure 9.8, the task \mathbf{e}_{4p} converges to a local minimum from the perspective of visual servoing. For the QP problem this local solution is in fact also the global one. The proximity task function in Figure 9.8 (c) indicates that task function $\mathbf{e}_{\delta_{s_1}}$ corresponding to sensor 9 is close to the situation of virtual contact for ring 1. From Figure 9.8 d, we can see that the contact constraints are satisfied for sensors on ring 2 including 4 and 12. The abrupt detection of sensor 9 on ring 1 results in jerky motion as seen in Figure 9.8 around 1 *sec*. At around 3 *sec* round, with closer detection of the blocking planes, change in control input can be noticed. Eventually from approximately 8 *sec*, the inequality constraints prevent the robot from further advancing as observed in Figure 9.8 c as the control input converges to very low value. The experiment is stopped when the norm of gradient of Lagrangian ($\|\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{0}, \kappa^*)\|$) falls below the stopping condition value of $\epsilon_{qd} = 0.01$. It can also be observed that the task does not undergo impulsive motions as anticipated in Section 9.3.2.4, due to the good value of estimation of target normals.

9.5.3 Case 3 : Obstacles affecting ring 1 in the initial phase of the task

In this scenario, we show the interest of robot redundancy by placing obstacles on either side of the robot closer to ring 1 as shown in Fig. 9.9. The workspace is chosen to observe the reaction of link 5 of Panda robot on which ring 1 is attached while the task is executed.

As seen in Figure 9.10 (c), task execution begins with ring 1 being closer to the obstacle on the right while violating the contact constraint for sensors $\{16, 17\}$ and coming close to the threshold distance for sensor 18. The control law as seen in Figure 9.10 (b) in initial phase of approximately 1.5 *secs* ensures that the robot is pushed back into safety. This results in a motion towards the obstacle on the left with sensors $\{8, 9, 10\}$ on the diametrically opposite side to come close to its threshold value. The control law again ensures safety with change in its value around 2 *sec* and after executes the task with no

influence from proximity sensors. Overall it ensures that the positioning task is achieved with minimal influence from these obstacles and in safety. From plots (a) and (b) it is clear that the task converges.

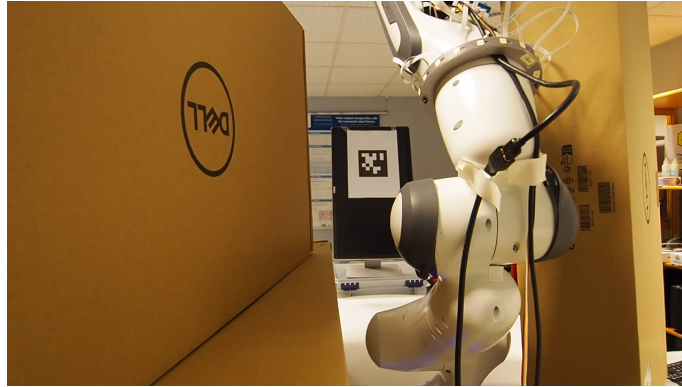


Figure 9.9 – Obstacle on either side

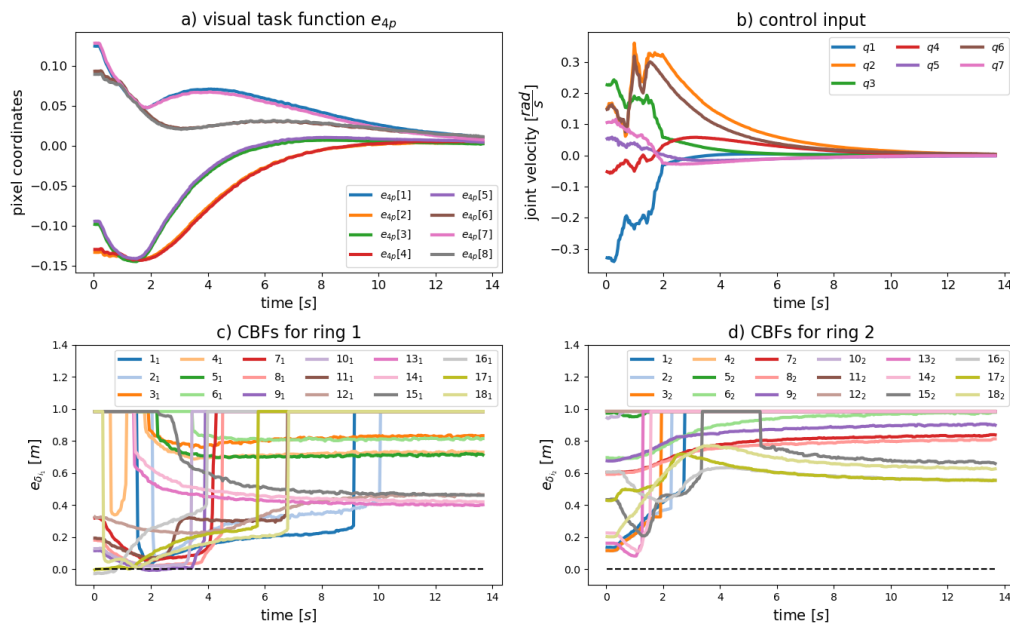


Figure 9.10 – Plots for Case 3 : Obstacle on either side

Additionally we provide results of execution without considering the obstacle avoidance task. We execute the same task as above and it leads to collision of the obstacle on the right. This is shown in Figure 9.11. The plots in Figure 9.12, shows that the contact constraints that were violated for sensors $\{16, 17\}$ of ring 1 still remains below threshold

eventually leading to collision.



Figure 9.11 – Robot colliding when obstacle avoidance is not considered

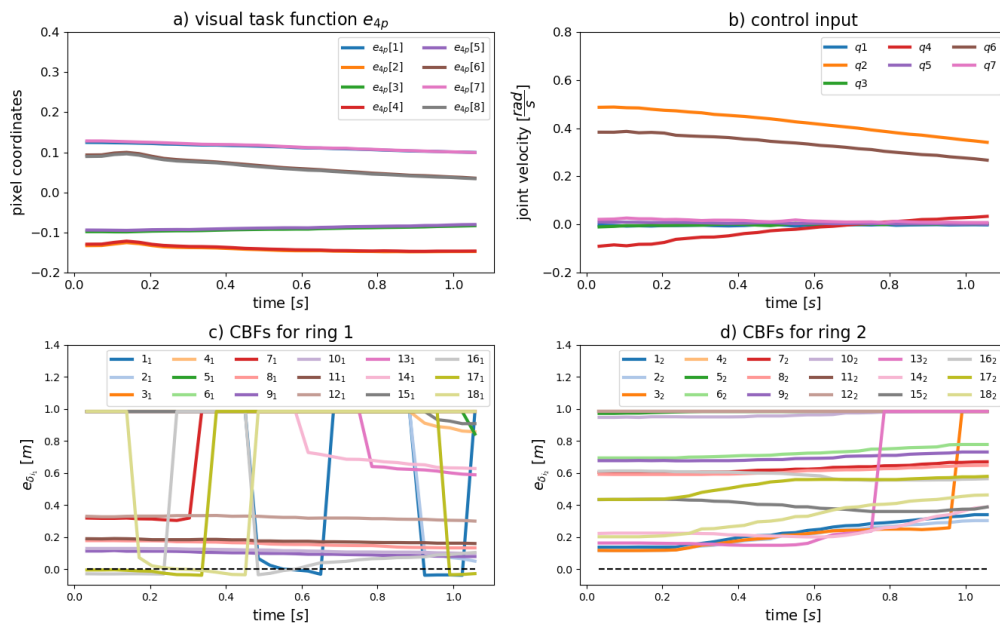


Figure 9.12 – Plots for Case 3 : when obstacle avoidance is not considered.

9.5.4 Case 4 : Obstacle affecting ring 1 in the final phase of the task

In this scenario, we follow up on the previous case by placing an obstacle that remains close to ring 1 during the final configurations of the robot while the task is being executed.

Figure 9.13 show the initial and final configurations of the task during positioning. As observed in Figure 9.14 c around 5 *sec*, proximity sensor 13 on ring 1 comes into virtual contact with the obstacle. An appropriate control input is evaluated by the \mathcal{QP} to ensure convergence of vision task to its global minimum as seen in Figure 9.14 b around 5 *sec*. Concurrently, the proximity sensor 13 on ring 1 remains in virtual contact or at its designed threshold distance until convergence in positioning task. It can also be concluded from this that the control design was able to use the redundancy of the robot for reaching the desired position with minimal influence from obstacle.

In addition, we also show the execution of the task in the same experimental workspace without activation of obstacle avoidance task. The final configuration of the positioning task is shown in Figure 9.15. When compared with Figure 9.13b it can be noticed that the configuration of robot for the link on which ring 1 is attached is closer to the top of the obstacle. From Figure 9.16 (b) it is visible that the obstacle avoidance task is not activated due to the smooth nature of control input from executing only the vision task. Meanwhile sensor {10, 11, 12, 13} go below the threshold value during the interval between 5 *sec* and 10 *sec*, thus not ensuring anymore the safety of the system.

9.5.5 Case 5 : Obstacles affecting both rings 1 and 2 in the final phase of the task

In this case, we place two planar obstacles in the workspace to create a congested space around the robot arm while it has to converge to the goal position. Note that, in case safety inequalities from proximity sensors are not taken into account, the robot arm collides with the obstacles. The initial configuration of the robot is shown in Figure 9.17a. The target points are placed in such a way that the regulation of visual task function leads to a final configuration of the manipulator in between the obstacles as shown in Figure 9.17b. The gain of sensors on rings 1 and 2 are chosen as $\lambda_1 = 1$ and $\lambda_2 = 3$ and threshold distance is selected as $\delta_{i1}^* = 4 \text{ cm}$ and $\delta_{i2}^* = 8 \text{ cm}$. As can be seen in Figure 9.18a, visual task converges. First, the manipulator moves towards the target and encounters the obstacle on the right, which results in a change in the evolution of visual task function due to the change in control input seen around 2.5 *s* as observed in Figure 9.18b for bringing back proximity sensors {2, 3, 4, 5} to the safe set, as seen in Figure 9.18d. As expected, the robot moves to the left for avoiding collision with obstacle on the right. While moving close to the target around 4 *s*, sensors {16, 17} on ring 1 come close to virtual contact

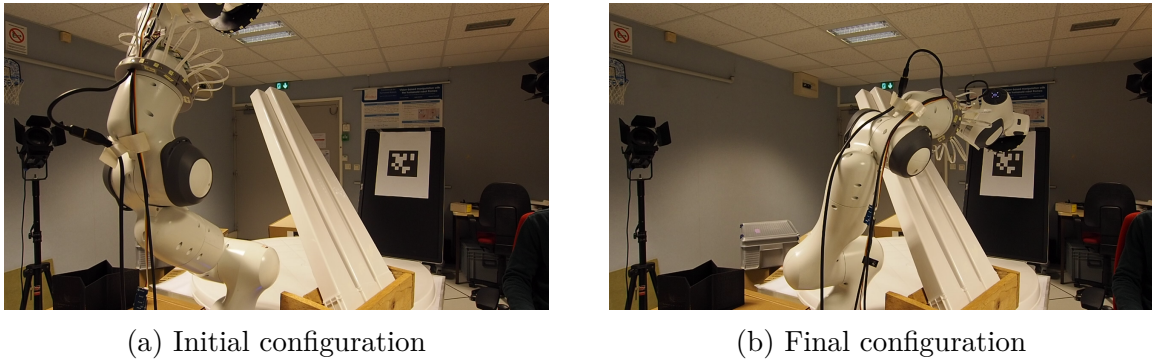


Figure 9.13 – Experimental setup for obstacle affecting ring 1 in the final phase of the task

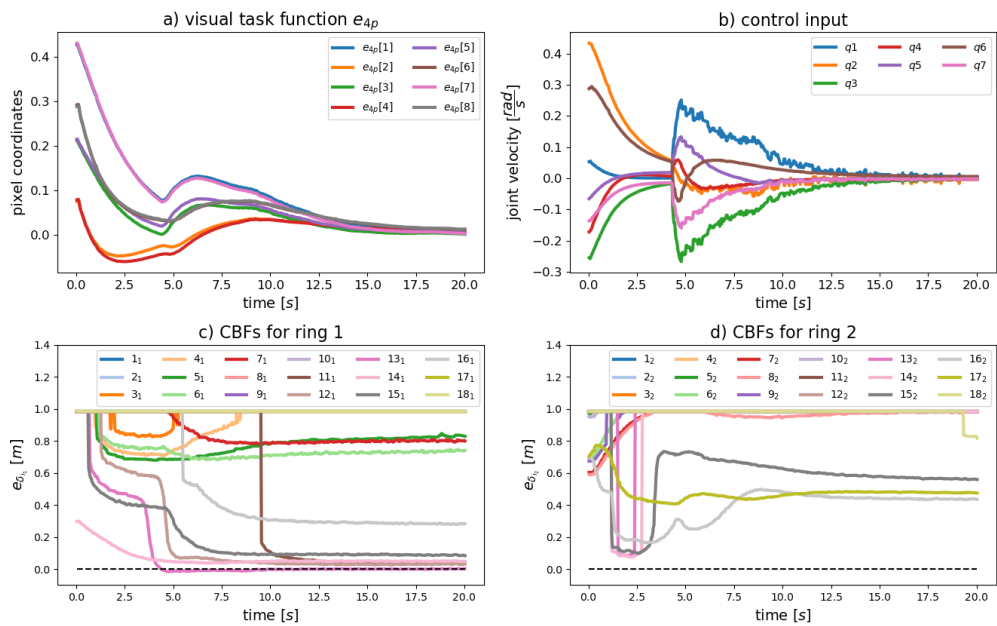


Figure 9.14 – Plot for Case 4 with obstacle affecting ring 1 in the final phase of the task

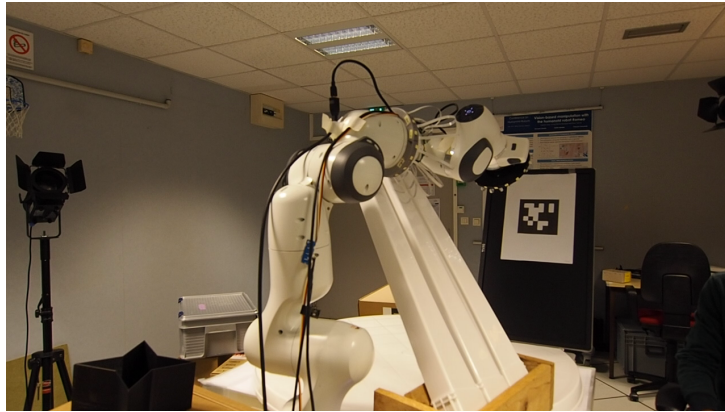


Figure 9.15 – Robot configuration at the end of experimental Case 4 without activation of obstacle avoidance task

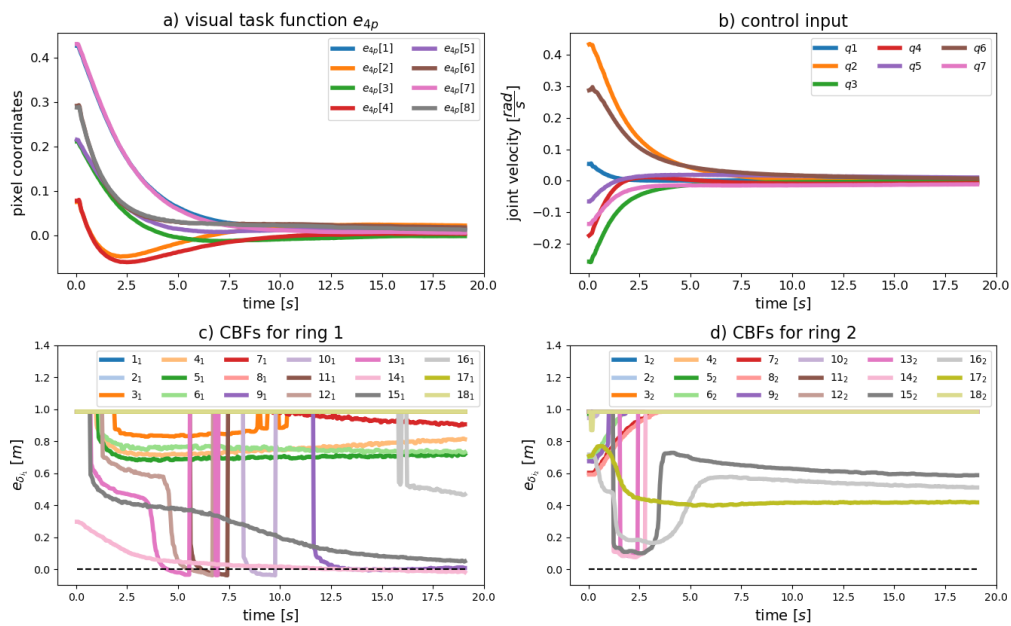


Figure 9.16 – Plots for Case 4 with obstacle affecting ring 1 in the final phase of the task without activation of obstacle avoidance task

as can be seen in Figure 9.18c, and the control architecture ensures to satisfy safety. Immediately after, sensors $\{12, 13\}$ come close to virtual contact on ring 2 from around 5 s. It can be noticed that the controller eventually converges to the final goal position while being close to virtual contact in both rings till the end of task execution and thereby maintaining safety. Joint velocities $\{5, 7\}$ are effectively used by the control architecture to have this back and forth motion in between the obstacles to reach the target.

9.5.6 Case 6: Behavior with Dynamic Obstacle

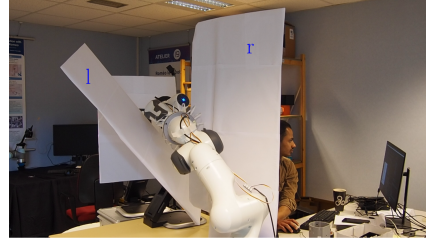
In the final case, we consider the situation when positioning task is achieved and then a dynamic obstacle interacts with it. The gain for sensors on rings 1 and 2 are chosen as $\lambda_1 = 1$ and $\lambda_2 = 3$ respectively and threshold distance is $\delta_{i1}^* = 4$ cm and $\delta_{i2}^* = 6$ cm respectively. We start the experiment from the initial configuration shown in Figure 9.19a. A human operator then interacts with different parts of the ring by bringing the hands closer as shown in Figure 9.19c and Figure 9.19d. As seen in Figure 9.20 the components of visual task function is close to zero at the beginning and becomes non-zero in situations when the hand of the human operator is brought close to the robot. Towards the end of the experiment, the components approach again the value of zero. At each occasion when visual task function \mathbf{e}_{4p} components are non-zero, it can be observed that control input components are also non-zero. It can be seen in Figure 9.20d the four instance in the interval between 5 s to 18 s when proximity sensors on ring 2 reach close to virtual contact. In the later phase from approximately 20 s to 30 s, the hand interacts with ring 1 as seen in Figure 9.20c. There are four instances when few sensors on ring 2 would reach their threshold value. The results suggest that the control architecture works well even though obstacles were assumed to be static in the modeling.

9.6 Simulation

In this section we consider simulation results for the theory considered. As shown in Figure 9.21, we utilize the same Panda robot with two proximity rings attached as explained in the beginning of Section 9.5. We use FrankaSim simulator [Oliva et al., 2022], which is based on CoppeliSim and ROS. It is also fully compatible with ViSP library. The dynamic model used is from model identification of the real Panda robot [Gaz et al., 2019] and the simulation results can be considered as realistic. In the following subsections



(a) Initial configuration where camera looks at target point that is placed in between two obstacles



(b) Final configuration achieved while being close to virtual contact through proximity sensors on both rings 1 and 2

Figure 9.17 – Experimental setup for Case 5 which involves two obstacles affecting two rings at convergence

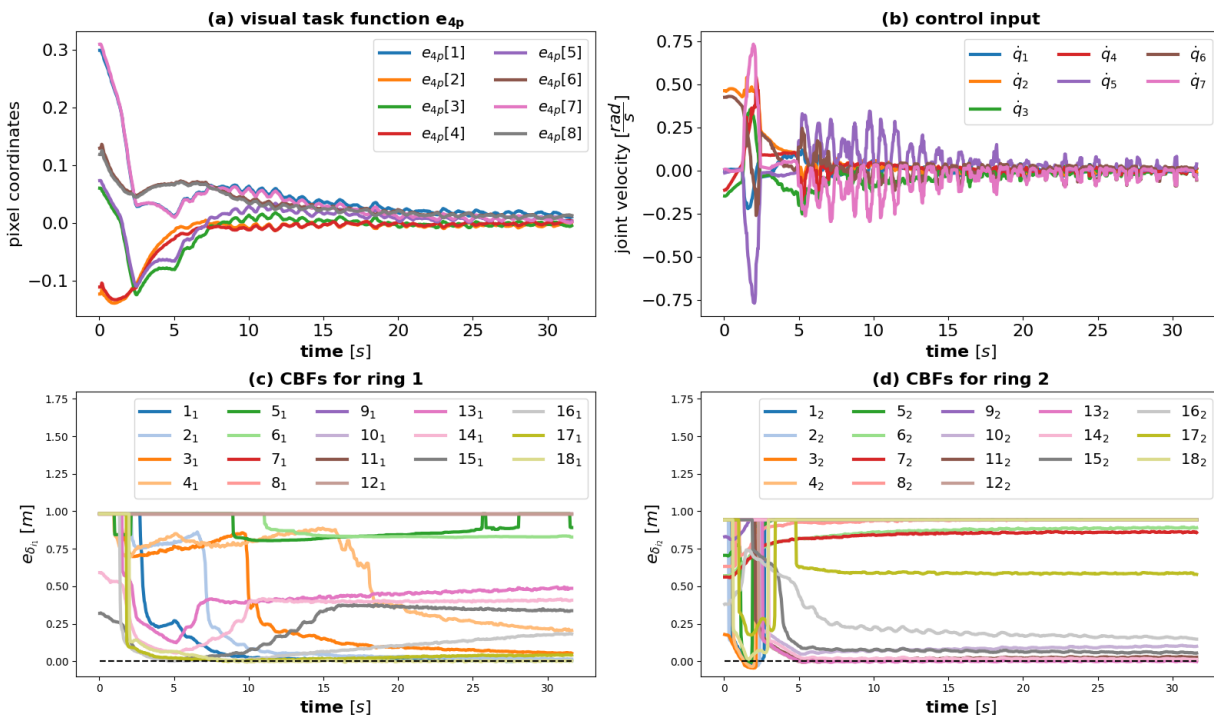


Figure 9.18 – Plots for Case 5 with obstacle affecting rings 1 and 2 in the final phase of the task

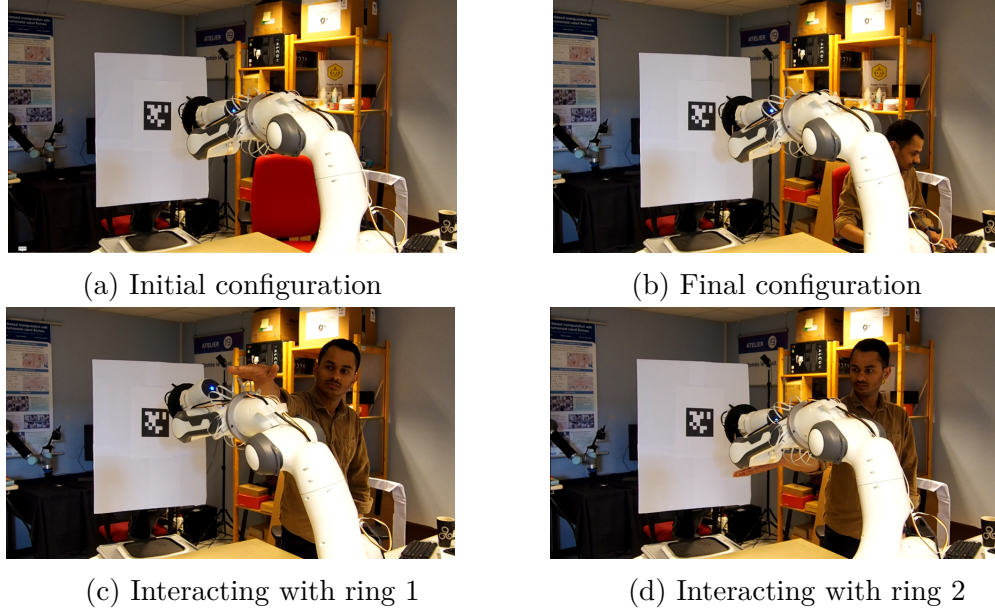


Figure 9.19 – Experimental setup for Case 6 where a human operator interacts with each of the rings

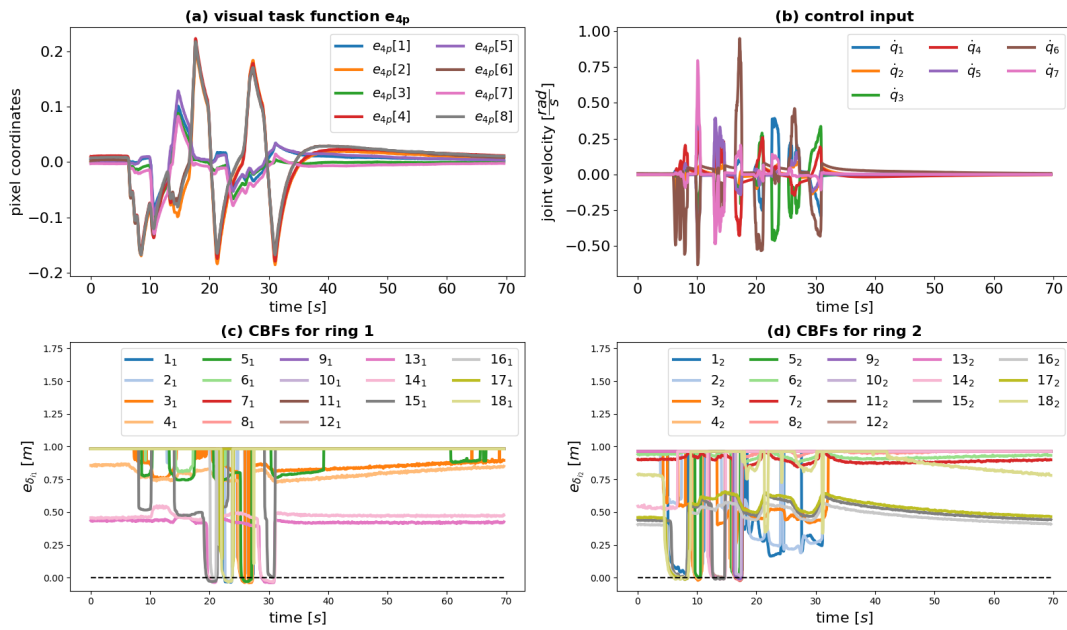


Figure 9.20 – Experimental results for Case 6 showing a scenario with dynamic obstacle

we provide results for three scenarios that along with the experimental results from Section 9.5 provide completeness to the validation process. In two such scenarios we consider positioning with involvement of vision and sensors from each of the proximity rings. In the third scenario we consider a third blocking plane augmenting the case considered for experimentation in Section 9.5.2 involving two blocking planes with a third one. A uniform noise in the interval $[-0.005\text{ m}, 0.005\text{ m}]$ is considered for the proximity signals from each ring. The avoidance distance δ_{ij} is considered as 1 cm in both rings. For both scenarios, we use the stopping condition value of $\epsilon_{qd} = 0.01$, which is empirically determined. All other values are the same as for the experiments: the gain of positioning task is chosen as $\lambda_{4p} = 0.4$ and gains for all proximity sensors are set to $\lambda_j = 1$. Regularization term is selected as $\mu = 0.01$, to get well conditioned results when constraints are active. For each proximity sensor considered, we evaluate the actual model of the interaction matrix from the value of target normal obtained from the simulator. The video of simulation for each of the cases below can be seen here.

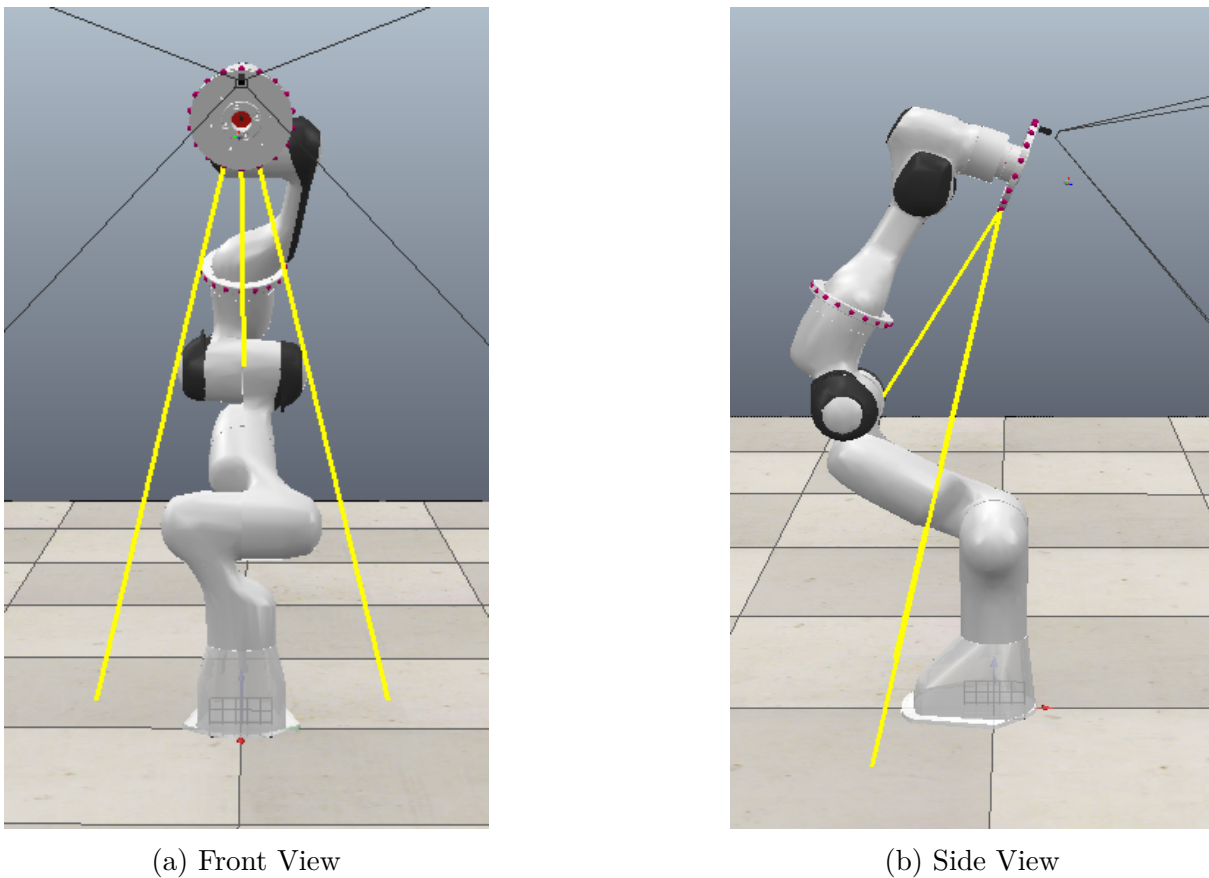


Figure 9.21 – Serial manipulator with camera and proximity array in Coppeliassim

9.6.1 Case 1 : Positioning Inside a Box

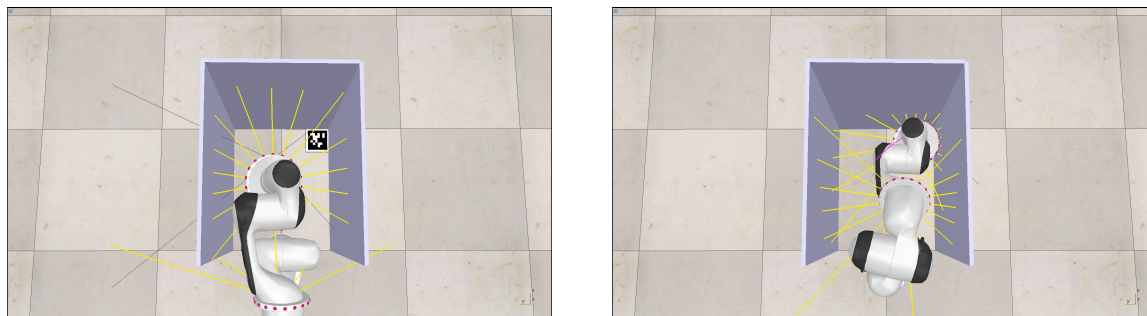
In the first scenario we consider a situation when the positioning task is achieved with shared contribution from both proximity and vision sensors. Figure 9.22 displays the workspace involving three planes indicating a box with a QR code placed in one corner. The initial configuration is close to the center of the box as seen in Figure 9.22a. Figure 9.23 shows the results from the simulation. At around 2 *sec* into the simulation, proximity sensor 18 on ring 2 achieves virtual contact with sensor 17 being very close to it. The control input displays slight changes in joint velocity $\{q_3, q_5, q_7\}$ to maintain safety. The task evolution in vision space is smooth until convergence. During the execution of the task, ring 1 does not involve itself while positioning. The task is achieved while proximity sensors on ring 2 is active. The final configuration of the robot can be seen in Figure 9.22b.

In addition, we show the results in simulation for the same scenario without involving the obstacle avoidance task. As seen in Figure 9.25, the visual task function e_{4p} does not converge. Proximity sensors $\{17, 1, 2\}$ on ring 2 goes below the value of safety as the ring collides with one side of the box. It is to be noted that in situation of no detection, the proximity signal is recorded as the maximum value. This is why the value of proximity task function associated to these sensors goes close to 1 during collision. Figure 9.24 shows this collision configuration from Coppeliassim.

9.6.2 Case 2 : Obstacle affecting ring 1 in the final phase of the task

In this scenario we consider a spherical obstacle in the workspace as seen in Figure 9.26. The obstacle is placed in such a manner that proximity sensors from ring 1 must contribute towards positioning task to ensure safety. The initial configuration of the robot is shown in Figure 9.26a. During the execution of task, as observed in Figure 9.27, proximity sensors $\{11, 12, 13\}$ from ring 1 comes close to virtual contact with sensor 12 maintaining contact. The control input gets perturbed from approximately 2 *sec* with significant changes in $\{q_2, q_4, q_6\}$. The final configuration can be seen in Figure 9.26b, where the vision task converges below the threshold while maintaining contact through proximity sensors from ring 1.

The simulation scenario is also executed without involving obstacle avoidance task. As can be observed from approximately 3 *sec* in Figure 9.30 when the proximity signals



(a) Initial configuration

(b) Final configuration

Figure 9.22 – Simulation setup for positioning inside a box

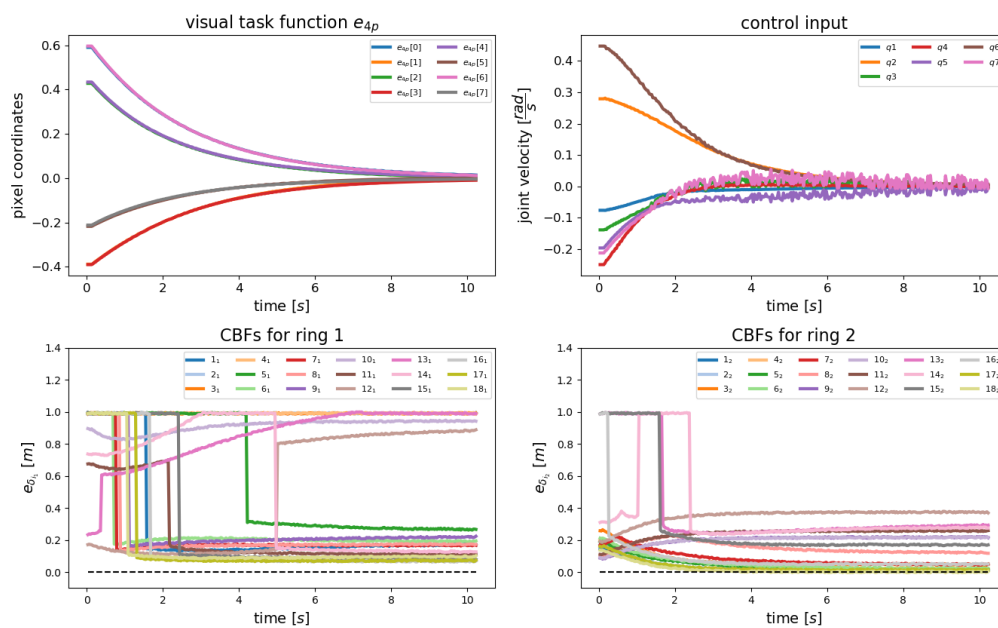


Figure 9.23 – Plot for simulation involving positioning inside a box

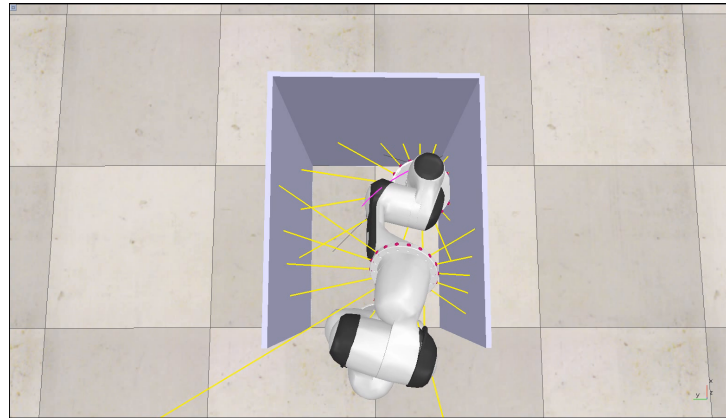


Figure 9.24 – End configuration for simulation involving positioning inside a box without considering obstacle avoidance task

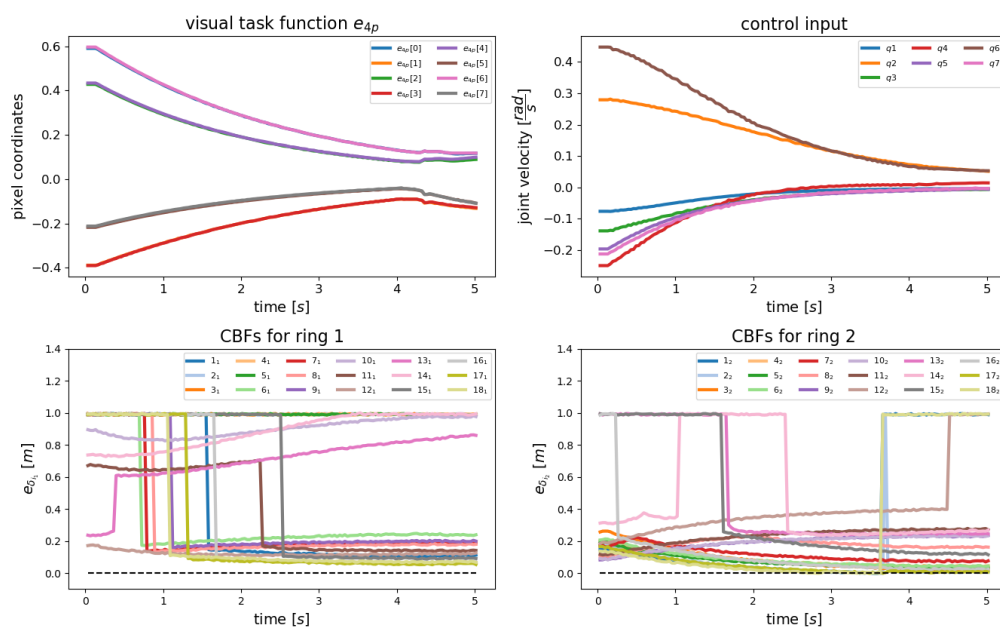
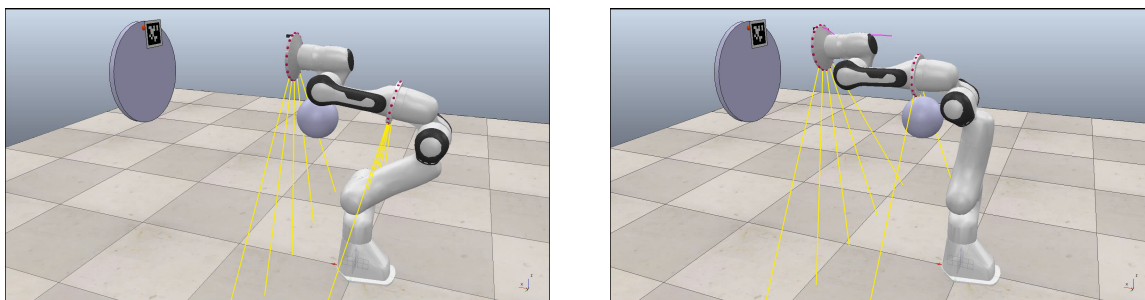


Figure 9.25 – Plot for simulation involving positioning inside a box without considering obstacle avoidance task



(a) Initial configuration

(b) Final configuration

Figure 9.26 – Simulation setup for positioning with spherical obstacle

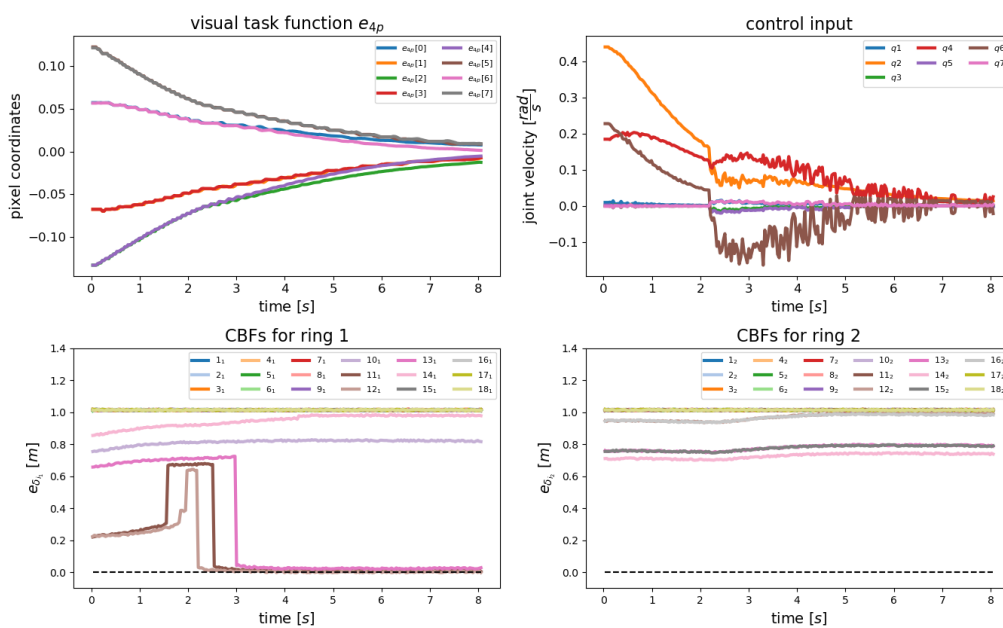


Figure 9.27 – Plot for simulation involving positioning with spherical obstacle

come close to its threshold value the vision task function diverges. The final configuration of the robot when the simulation is stopped is shown in Figure 9.29.

9.6.3 Case 3 : Three Blocking Planes

In the final scenario we augment the Case 2 from experimentation with an additional plane to get more sensors close to their threshold value. As seen in Figure 9.31, we have placed three planes that would prevent positioning from converging to the global minimum. From approximately 2 *sec*, we have several sensors from ring 2 reaching close to the threshold value. Control input indicates change around this time which eventually converges to zero value along with vision task function converging to a local minima.

We execute the above scenario without involving obstacle avoidance task. As can be seen in Figure 9.34, from approximately 2.5 *sec*, the task diverges in vision space. The final configuration of the robot can be seen in Figure 9.33 with ring 2 colliding with the plane on the left and bottom.

9.7 Conclusion & Future Work

In this chapter we have considered the task of positioning in congested space using Sensor-based Control framework. For this we combine signals from camera and proximity sensors. Control barrier functions provided the perfect theory to incorporate our intuitive understanding of using proximity sensors for ensuring anti-collision. Safety was achieved in all the experimental and simulation scenarios considered. For the task considered QP framework provided an ideal control architecture to combine sensor information in vision space and proximity space. It also ensured that the complementarity nature of the sensors were effectively utilized. We have carefully chosen scenarios that is of significant practical importance and performed real experimentations. In these experimentations, positioning task is achieved at a distance of 4 *cm* of obstacles in proximity space, which is the limit of distance measurement of our sensors. We have also considered other scenarios in simulation to complement the experimentation scenarios to have a comprehensive validation of the theory. In simulation we have performed positioning at a close distance of 1 *cm*. For proximity sensors the accuracy of data reduces around minimum distance. However, this would not cause problems in positioning as the solution would be still stable. Additionally, such stability also holds for uncertainties involved in the modelling parameters.

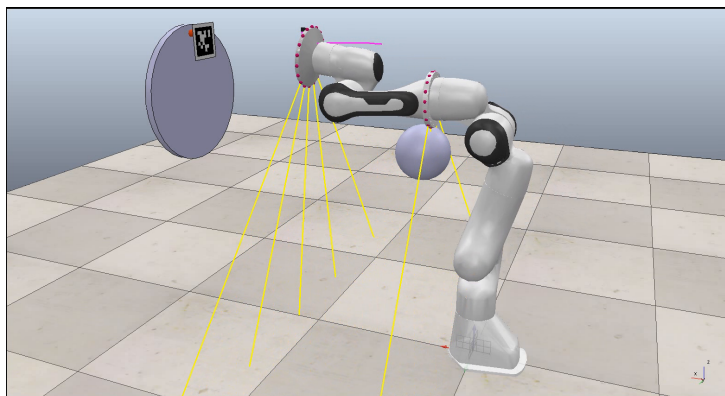


Figure 9.28 – End

Figure 9.29 – Simulation setup for involving Sphere without Obstacle Avoidance task

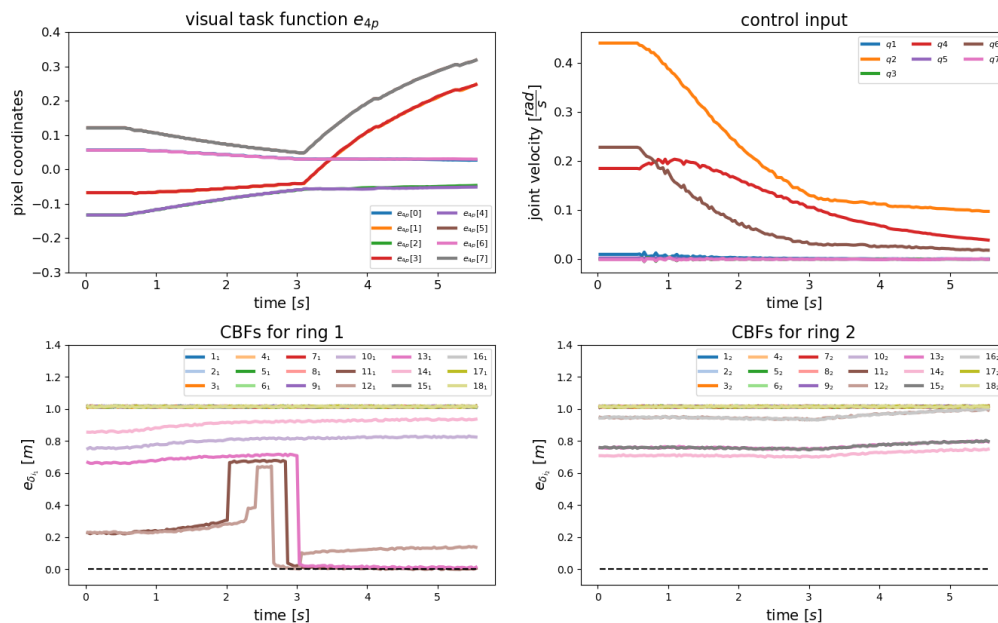


Figure 9.30 – Plot for simulation involving Sphere without obstacle Avoidance task

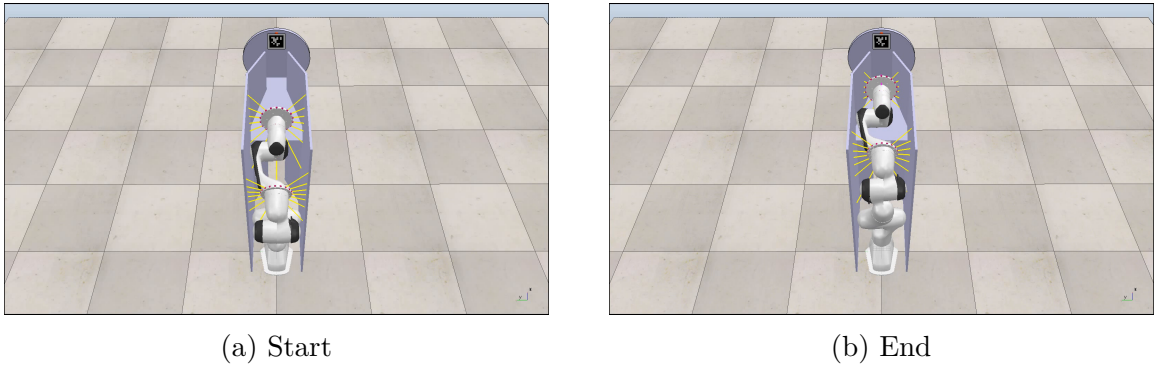


Figure 9.31 – Simulation setup for involving three blocking planes

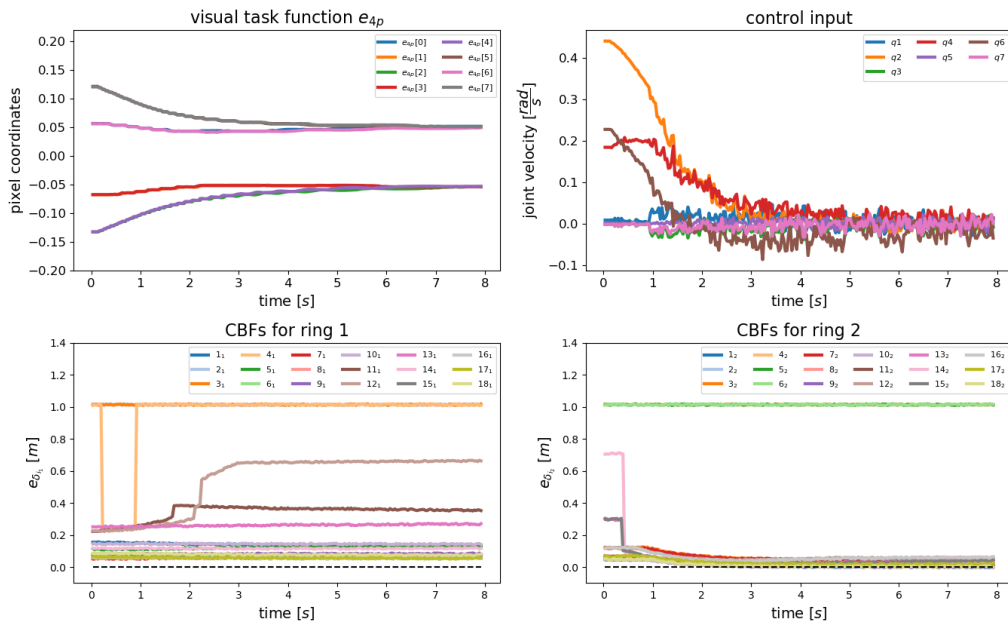


Figure 9.32 – Plot for simulation involving three blocking planes

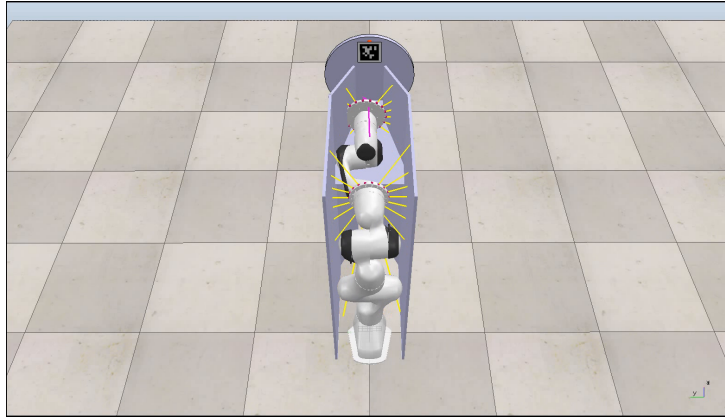


Figure 9.33 – End Configuration for the case of three blocking planes without obstacle avoidance task

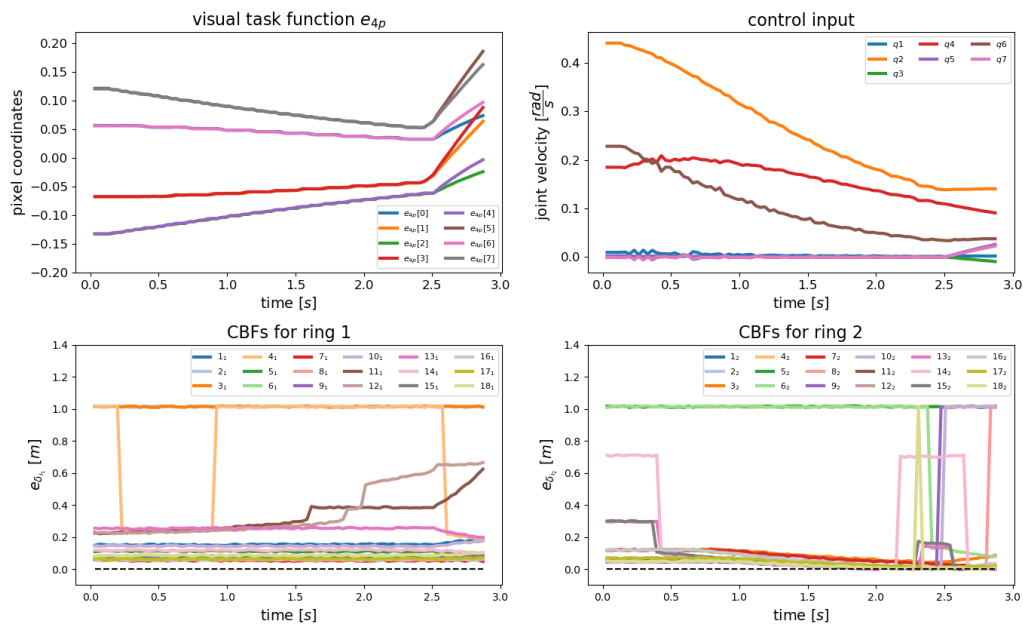


Figure 9.34 – Plot for simulation involving three blocking planes without obstacle avoidance task

The analysis was mainly considered for static obstacles. In future the framework can be adapted to consider fast dynamic motions in the workspace. Application of the above developed controller for human-robot collaboration can also be another prospective research direction. It would be required to have more proximity sensors covering the whole body of the manipulator like a skin. This can also lead to more smoother reactions reducing abrupt changes. It would also be interesting to integrate force sensors into the above framework, to consider a fully autonomous assembly operation in congested space such as the components of a satellite.

PART III

Use of Screw Theory in Sensor-based Control

APPLICATION IN SENSOR-BASED TASKS

10.1 Introduction

In Chapter 5 where we considered the plane-to-plane positioning task using proximity sensors, which is a rather simple task function, the explicit form of the interaction matrix pseudo-inverse was obtained. Inside the pseudo-inverse was another matrix denoted as \mathbf{L}^- in (5.14), that satisfied the first three conditions of the Moore-Penrose which made it a reflexive generalized (left-)inverse. In this part we explain more about such inverses that can also be used as suitable control directions for regulation of the task function. It is in fact the dual basis of the linear space of interaction screws formed from the chosen sensor features. In this part, we restrict our study to the two tasks considered in Part I, that is, plane-to-plane positioning and positioning wrt. a cylinder. Additionally we also extend it to a task using vision sensors. Although not practically used, we obtain the dual basis for 3-point Visual Servoing to finish the study.

Screw Theory was used in SBC classically in the modelling of interactions with the environment using sensor-based tasks embedded in the task function approach formalism [Samson et al., 1991]. In this approach, sensor features are functions of the relative pose between the sensor and the detected target. The variation of such sensor features involves a screw product involving interaction screw which when represented in matrix form is popularly called as interaction matrix. Set of screws associated with the sensor features characterizes the interaction with the environment with an analogy made to virtual contact. This virtual contact is defined based on constraints in sensor space involving sensor features and their desired value. This can be considered as a virtual linkage categorized by the dimension of reciprocal subspace of the subspace of interaction screws involved. The idea of virtual contact was used in defining sensor-based primitives that aided in the synthesis of sensor-based tasks [Espiau, 1990], [Chaumette et al., 1993]. The reciprocal subspace was also used in achieving secondary tasks while maintaining virtual contact.

In this part we use another idea in screw theory that could be interesting in SBC.

Given a sensor-based task we can identify a basis for the linear space involving interaction screws. Associated with it we identify a unique dual basis that can be used in designing the desired dynamical system in sensor space for the task considered. In Chapter 5, use of dual basis lead to simpler stability analysis compared to pseudo-inverse based closed-loop system.

In the following of this chapter, we discuss the dual basis of the space of interaction screws involving two tasks using proximity sensors and one task using vision sensor. The chapter is divided into 5 sections. In Section 10.2, we introduce the idea of dual basis for SBC. After that we discuss the dual basis in plane-to-plane positioning task in Section 10.3. Then we provide it for the task of positioning wrt. a cylinder in Section 10.4. In Section 10.5, we describe the dual basis for 3-point Visual Servoing. We end the chapter by providing conclusions in Section 10.6. We also perform a verification step in appendix for all tasks considered to show the validity of the dual basis.

10.2 Dual Basis

Let us consider n independent sensor features identified to perform a robotic task that control n DoF of the robot. These features would lead to a set of independent interaction screws of the form $\mathcal{H}_i = (\mathbf{H}_i(\cdot), \mathbf{u}_i)$. Let us consider a set $H = \{., \mathcal{H}_i, .\}$ consisting of all these interactions. For a given basis H there always exist a unique dual basis $K = \{., \mathcal{K}_j, .\}$ [Bullo & Lewis, 2005] such that,

$$\mathcal{K}_j(\mathcal{H}_i) = \mathcal{H}_i \circ \mathcal{K}_j = \Delta_{ij} \quad (10.1)$$

where Δ_{ij} indicates kronecker delta function. The elements of dual basis can be represented as $\mathcal{K}_i = (\mathbf{K}_i(\cdot), \mathbf{w}_i)$. To recall, term $\mathcal{H}_i \circ \mathcal{K}_j$ represents the screw product which can be evaluated as,

$$\mathcal{H}_i \circ \mathcal{K}_j = (\mathbf{H}_i(\cdot), \mathbf{u}_i) \circ (\mathbf{K}_j(\cdot), \mathbf{w}_j) = \mathbf{H}_i(\cdot) \cdot \mathbf{w}_j + \mathbf{K}_j(\cdot) \cdot \mathbf{u}_i \quad (10.2)$$

Let us consider a task function \mathbf{e} constructed from these n -independent sensor features with components defined as $e_i = s_i - s_i^*$. Its first order derivative can be written from (1.2) as

$$\dot{e}_i = \dot{s}_i = \mathcal{H}_i \circ \mathcal{K}_{ST} \quad (10.3)$$

The velocity screw \mathcal{K}_{ST} can be considered as linear combination of dual basis and can be expressed in the following manner,

$$\mathcal{K}_{ST} = \sum_k \sum_j \kappa_{kj} e_k \mathcal{K}_j \quad (10.4)$$

where κ_{kj} is a scalar. Substituting (10.4) in (10.3) we get,

$$\begin{aligned} \dot{e}_i &= \sum_k \sum_j \kappa_{jk} e_k \mathcal{H}_i \circ \mathcal{K}_j \\ &= \sum_k \kappa_{ik} e_k \end{aligned} \quad (10.5)$$

The closed-loop equation of the task can be represented as

$$\dot{\mathbf{e}} = \mathbf{M}\mathbf{e} \quad (10.6)$$

where $\mathbf{M} = [\kappa_{ik}]$ is an $n \times n$ closed-loop matrix.

When κ_{ik} is chosen as a function of task function components, non-linear behavior can be obtained in sensor space. As part of future work it would be interesting to explore such choices that could lead to stable attractors involved with non-linear dynamical systems such as limit cycles.

To obtain classical regulation of task function \mathbf{e} to origin all we need to do is select γ_{ik} as

$$\gamma_{ik} = \begin{cases} -\lambda & i = k \\ 0 & i \neq k \end{cases} \quad (10.7)$$

The velocity screw for the above choice of γ_{ik} can be expressed in matrix form as,

$$\mathbf{v} = -\lambda \mathbf{G}\mathbf{e} \quad (10.8)$$

where \mathbf{G} is the reflexive generalized (left-)inverse for the task considered and is given by

$$\mathbf{G} = \begin{bmatrix} \cdot & \mathbf{K}_i(\cdot) & \cdot \\ \cdot & \mathbf{w}_i & \cdot \end{bmatrix} \quad (10.9)$$

In SBC, we identify an interaction matrix from the set H to get an interaction matrix

defined as,

$$\mathbf{L} = \begin{bmatrix} \cdot & \cdot \\ \mathbf{u}_i^T & \mathbf{H}_i(\cdot)^T \\ \cdot & \cdot \end{bmatrix} \quad (10.10)$$

Matrix \mathbf{G} built from dual basis provide a least-squares solution to the linear equation $-\lambda \mathbf{e} = \mathbf{L}\mathbf{v}$. It would be different from pseudo-inverse \mathbf{L}^+ since it would not satisfy the final condition of Moore-Penrose, $(\mathbf{GL})^T \neq \mathbf{GL}$, as described in (5.22).

In a practical implementation we would have to estimate matrix associated to dual basis $\hat{\mathbf{G}}$, so that the control scheme becomes $\mathbf{v} = -\lambda \hat{\mathbf{G}}\mathbf{e}$, leading to the closed-loop system

$$\dot{\mathbf{e}} = \mathbf{L}\mathbf{v} = -\lambda \mathbf{L}\hat{\mathbf{G}}\mathbf{e} \quad (10.11)$$

whose stability is ensured if $\mathbf{L}\hat{\mathbf{G}} > 0$. This leads to closed-form equations in SBC when dual basis is identified through inspection. In the next section we consider the task of plane-to-plane positioning considered in Chapter 4.

10.3 Plane-to-Plane Positioning

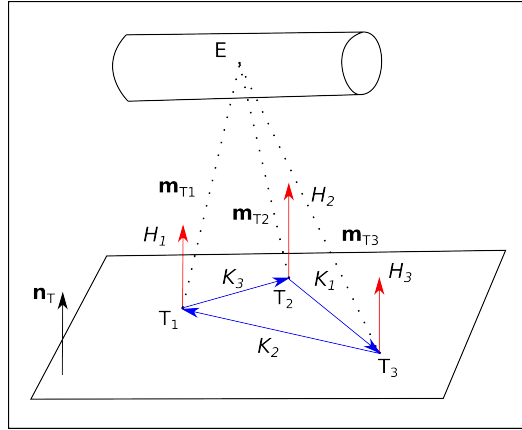


Figure 10.1 – Interaction screws and control directions at target plane

In this subsection we apply the above approach to plane-to-plane positioning task in case of minimal sensors discussed in Section 4.2.1 with notation of a interaction screw from Section 5.1. The set H for the considered task is of dimension 3 with an element

represented as,

$$\mathcal{H}_i = \frac{-1}{\mathbf{n}_T \cdot \mathbf{n}_{S_i}} (\mathbf{m}_{T_i} \times \mathbf{n}_T, \mathbf{n}_T) \quad (10.12)$$

where we recall from (5.3) that \mathbf{n}_{S_i} sensor axis and \mathbf{m}_{T_i} is the translational vector from target point T_i to end-effector point E as seen in Figure 10.1. As pointed out in Section 4.2 the interaction screw is a slider at target point T_i and aligned along the target plane normal \mathbf{n}_T . As depicted in Figure 10.1, the red lines indicate the screws aligned along plane normal. The dual basis can be obtained through inspection by observing the interaction screws at the target. So now, we have a situation where there are three equivalent pure forces aligned along planar normal direction at three different points. To find the dual basis we can choose an arbitrary interaction screw and find a twist that would be reciprocal to the other two and but produces unit power with it. Since we have two pure forces on a plane, we can find a pure rotation lying on the plane and passing through T_{i+} and T_{i-} to be reciprocal to two interaction screws. The intensity of the screw can then be chosen to produce unit power with the interaction screw at T_i .

We denote them as $\mathcal{K}_i = (\mathbf{K}_i(\cdot), \mathbf{w}_i)$ and at the end-effector point E , these screws are expressed as,

$$\mathcal{K}_i = \gamma_i (\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}}, \mathbf{m}_{T_{i-}} - \mathbf{m}_{T_{i+}}) \quad (10.13)$$

where

$$\begin{aligned} \gamma_i &= \frac{-\mathbf{n}_T \cdot \mathbf{n}_{S_i}}{l} \\ l &= \mathbf{n}_T \cdot (\mathbf{m}_{T_1} \times \mathbf{m}_{T_2} + \mathbf{m}_{T_2} \times \mathbf{m}_{T_3} + \mathbf{m}_{T_3} \times \mathbf{m}_{T_1}) \end{aligned} \quad (10.14)$$

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 are the components of matrix \mathbf{L}^- matrix in (5.14) where a common notation was used for both minimal and redundant number of sensors through parameters ${}^k\beta_{j+}$, ${}^k\beta_{j-}$, ${}^k\mathbf{m}_{\beta_{j+}}$, ${}^k\mathbf{m}_{\beta_{j-}}$. By substituting the parameter values for minimal sensors from (5.3) the above form of dual basis is obtained.

In Appendix C, verification step is provided for the dual basis considered. Term l has a geometric meaning. It is twice the area 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

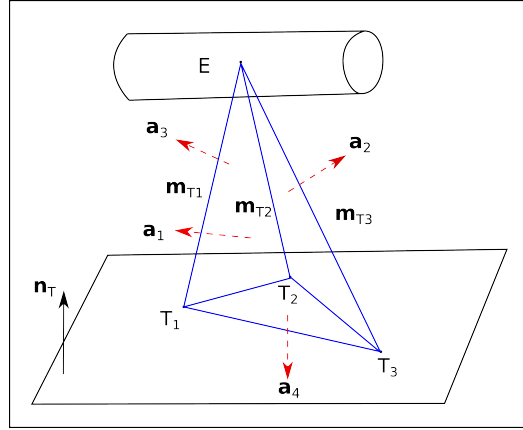


Figure 10.2 – Tetrahedron formed during plane-to-plane following task.

hence as shown in Figure 10.2 we get,

$$\begin{aligned}
 l &= \mathbf{n}_T \cdot (2\mathbf{A}_1 + 2\mathbf{A}_2 + 2\mathbf{A}_3) = \mathbf{n}_T \cdot (-2\mathbf{A}_4) \\
 &= \mathbf{n}_T \cdot (-(\mathbf{m}_{T_2} - \mathbf{m}_{T_3}) \times (\mathbf{m}_{T_1} - \mathbf{m}_{T_3})) \\
 &> 0
 \end{aligned} \tag{10.15}$$

Singularity for this task would occur when the points are collinear and this would correspond to $l = 0$.

As pointed out in [Samson et al., 1991], the 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 [Featherstone, 2008]. This has been identified in [Espiau, 1990], [Samson et al., 1991] while describing sensor-based primitives.

10.4 Positioning wrt. a cylinder

In this section we obtain the dual basis for the positioning task wrt. a cylinder with minimal sensors described in Section 6.2.1. The modeling in that section used notations to identify a single proximity sensor in proximity array. For our current analysis we indicate each interaction screw with index i . As pointed out there, we can consider the set H of

dimension 4 composed of screws when represented at point E as,

$$\mathcal{H}_i = \frac{-1}{\mathbf{n}_{T_i} \cdot \mathbf{n}_{S_i}} (\mathbf{m}_{T_i} \times \mathbf{n}_{T_i}, \mathbf{n}_{T_i}) \quad (10.16)$$

To find the dual basis through inspection, let us first represent it at the target as shown in

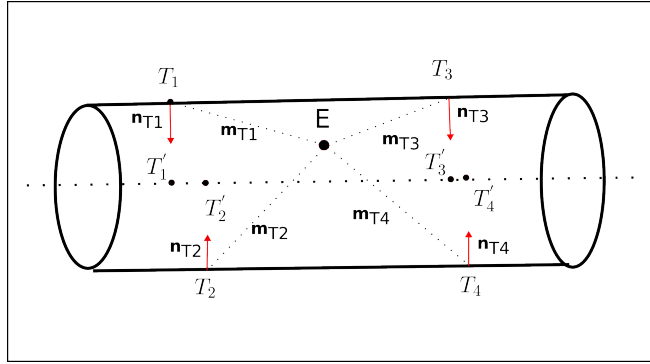


Figure 10.3 – Interaction for guidance task with minimal sensors

Figure 10.3. These are again sliders aligned along the target normal of a cylinder, which are along the radial direction of the cylinder. Therefore we can further travel along the radius to reach the axis, where these screws would remain as sliders. So now, we have a situation where there are equivalent pure forces on the axis of the cylinder. To find the dual basis we can choose an arbitrary interaction screw and find a twist that would be reciprocal to the other three and also produce unit power with the chosen interaction screw. Since we have pure forces on a line, we can find two pure rotations each located at T'_{i+} and T'_{i-} which are aligned along $\mathbf{n}_{T_{i-}}$ and $\mathbf{n}_{T_{i+}}$ and are reciprocal to both interactions screws. A suitable linear combination of these twists can be evaluated that becomes reciprocal to the third interaction screw located at T'_{i++} and produce unit power with the interaction screw at T'_i .

So the twist K_i when represented now in proximity array point is

$$\frac{-\mathbf{n}_{T_i} \cdot \mathbf{n}_{S_i}}{l_i} \left(a_{1i} \mathbf{m}_{T_{i+}} \times \mathbf{n}_{T_{i-}} - a_{2i} \mathbf{m}_{T_{i-}} \times \mathbf{n}_{T_{i+}}, a_{1i} \mathbf{n}_{T_{i-}} - a_{2i} \mathbf{n}_{T_{i+}} \right) \quad (10.17)$$

where

$$a_{1i} = (\mathbf{m}_{T_{i++}} - \mathbf{m}_{T_{i-}}) \cdot (\mathbf{n}_{T_{i++}} \times \mathbf{n}_{T_{i+}}) \quad (10.18)$$

$$a_{2i} = (\mathbf{m}_{T_{i++}} - \mathbf{m}_{T_{i+}}) \cdot (\mathbf{n}_{T_{i++}} \times \mathbf{n}_{T_{i-}}) \quad (10.19)$$

$$l_i = \begin{vmatrix} (\mathbf{m}_{T_i} - \mathbf{m}_{T_{i+}}) \cdot (\mathbf{n}_{T_i} \times \mathbf{n}_{T_{i-}}) & (\mathbf{m}_{T_{i++}} - \mathbf{m}_{T_{i+}}) \cdot (\mathbf{n}_{T_{i++}} \times \mathbf{n}_{T_{i-}}) \\ (\mathbf{m}_{T_i} - \mathbf{m}_{T_{i-}}) \cdot (\mathbf{n}_{T_i} \times \mathbf{n}_{T_{i+}}) & (\mathbf{m}_{T_{i++}} - \mathbf{m}_{T_{i-}}) \cdot (\mathbf{n}_{T_{i++}} \times \mathbf{n}_{T_{i+}}) \end{vmatrix} \quad (10.20)$$

A trivial singularity involved would be when $\mathbf{n}_{T_i} = \mathbf{n}_T$, which results in $l_i = 0$. This is considered in Section 6.2.1.4 for positioning task outside cylinder. The validity of the basis is verified in Appendix D.

10.5 3-point Visual Servoing

In this section we consider the dual basis involved in 3-point Visual Servoing. In Section 9.2.1, we presented the form of the interaction matrix from an observation of an image point represented in normalized coordinates as

$$\mathbf{L}_x = \begin{bmatrix} \frac{-1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & \frac{-1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{bmatrix} \quad (10.21)$$

Let us consider the observation of three points through the camera. Let these points be denoted as T_i . The corresponding interaction screw at camera point C can be represented as

$$\begin{aligned} \mathcal{H}_{ix} &= (\mathbf{m}_{T_i} \times \mathbf{u}_{ix}, \mathbf{u}_{ix}) \\ \mathcal{H}_{iy} &= (\mathbf{m}_{T_i} \times \mathbf{u}_{iy}, \mathbf{u}_{iy}) \end{aligned} \quad (10.22)$$

with

$$\begin{aligned} \mathbf{u}_{ix} &= \frac{\mathbf{m}_{T_i} \times \hat{\mathbf{j}}}{(\mathbf{m}_{T_i} \cdot \hat{\mathbf{k}})^2} \\ \mathbf{u}_{iy} &= -\frac{\mathbf{m}_{T_i} \times \hat{\mathbf{i}}}{(\mathbf{m}_{T_i} \cdot \hat{\mathbf{k}})^2} \end{aligned} \quad (10.23)$$

As shown in Figure 10.4 we consider \mathbf{m}_{T_i} as the displacement vector from target point T_i to camera point C . The set consisting of interaction screws can be considered as $H =$

$\{.., \mathbf{H}_{ix}, \mathbf{H}_{iy}, ..\}$ of dimension 6. Interaction screw corresponding to sensor features x and y can be represented as sliders at target points T_i . As shown in Figure 10.4, when image plane coordinates of a point is used as a sensor feature it would span a plane with normal along the ray of light at target point. In our case we therefore obtain 3 such planes spanned by each corresponding screw directions.

In order to obtain the dual basis we look at the tetrahedron that is formed. One immediate candidate is pure rotation passing through two points. This would definitely be reciprocal to four interactions each time. Another candidate is a pure rotation that would lie on two planes at the same time, then it can be used to cancel 4 interactions like the first type of candidates. A linear combination of these two candidates becomes reciprocal to five interaction screws and produce unit power with the associated interaction screw.

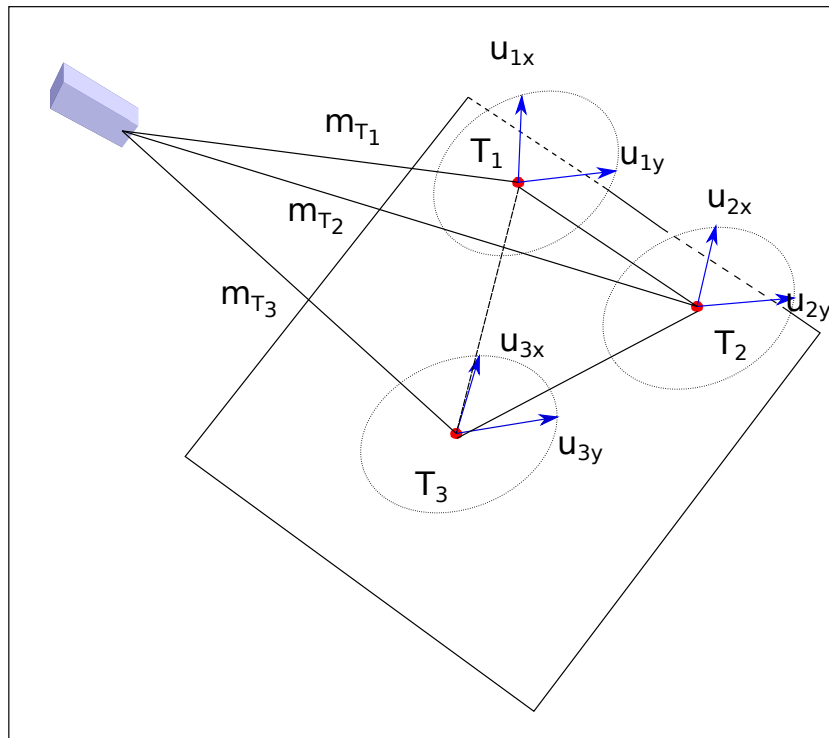


Figure 10.4 – Interaction screws for Visual Servoing of 3 points

The dual basis consists of $K = \{ \cdot \mathbf{K}_{ix}, \mathbf{K}_{iy} \cdot \}$. It can be concisely represented as,

$$\begin{aligned} \mathbf{K}_{ix} &= \gamma_i((a_i \cdot \hat{\mathbf{i}})(\mathbf{m}_{T_{i-}} \times \mathbf{m}_{T_{i+}}) - (\mathbf{b}_i \cdot \hat{\mathbf{i}}) \mathbf{q}_i, (a_i \cdot \hat{\mathbf{i}})(\mathbf{m}_{T_{i+}} - \mathbf{m}_{T_{i-}}) - (\mathbf{b}_i \cdot \hat{\mathbf{i}})(\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}})) \\ \mathbf{K}_{iy} &= \gamma_i((a_i \cdot \hat{\mathbf{j}})(\mathbf{m}_{T_{i-}} \times \mathbf{m}_{T_{i+}}) - (\mathbf{b}_i \cdot \hat{\mathbf{j}}) \mathbf{q}_i, (a_i \cdot \hat{\mathbf{j}})(\mathbf{m}_{T_{i+}} - \mathbf{m}_{T_{i-}}) - (\mathbf{b}_i \cdot \hat{\mathbf{j}})(\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}})) \end{aligned} \quad (10.24)$$

where

$$\begin{aligned} \gamma_i &= \frac{(\mathbf{m}_{T_i} \cdot \hat{\mathbf{k}})^2}{\begin{vmatrix} a_i \cdot \hat{\mathbf{i}} & a_i \cdot \hat{\mathbf{j}} \\ b_i \cdot \hat{\mathbf{i}} & b_i \cdot \hat{\mathbf{j}} \end{vmatrix}} \\ a_i &= (\mathbf{p}_i \times \mathbf{m}_{T_i}) \\ b_i &= (l \mathbf{n}_T \times \mathbf{m}_{T_i}) \\ \mathbf{p}_i &= (\mathbf{m}_{T_{i-}} \cdot (\mathbf{m}_{T_{i-}} - \mathbf{m}_{T_i})) \mathbf{m}_{T_{i+}} - (\mathbf{m}_{T_{i+}} \cdot (\mathbf{m}_{T_{i+}} - \mathbf{m}_{T_i})) \mathbf{m}_{T_{i-}} \\ \mathbf{q}_i &= (\mathbf{m}_{T_{i-}} \cdot \mathbf{m}_{T_{i-}}) \mathbf{m}_{T_{i+}} - (\mathbf{m}_{T_{i+}} \cdot \mathbf{m}_{T_{i+}}) \mathbf{m}_{T_{i-}} \\ l \mathbf{n}_T &= \mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_i} + \mathbf{m}_{T_{i-}} \times \mathbf{m}_{T_{i+}} + \mathbf{m}_{T_i} \times \mathbf{m}_{T_{i-}} \end{aligned} \quad (10.25)$$

l is the same as in the case of plane-to-plane positioning task with \mathbf{n}_T as target normal of the plane on which points lie. The validity of the basis is verified in Appendix E.

10.6 Conclusion

In this part we have provided a set of control directions that can be used in regulation of a certain set of task functions. These control directions lead to closed-form equations in SBC. As shown in Chapter 5, it also leads to simpler stability conditions. In two such task functions the dual basis was able to identify the singularity location. In the task of 3-point Visual Servoing however, the singularity of dual basis is not associated to the classical cylinder of singularity [Michel & Rives, 1993]. This will be part of future work.

It would be interesting to obtain dual basis for other tasks used in Visual Servoing especially image moments. It is important to explore more control properties that are associated with such control directions involved in regulation. Alternatively, these control directions could be utilized to obtain alternative behaviors in sensor space including stable focus and limit cycles.

PART IV

Conclusion and Future Work

CONCLUSION

The thesis started by studying a multi-sensory system of proximity sensors that enables local environment sensing. This modality could be used as the primary means to execute a task. The task we considered concerned both positioning wrt. a planar target and a cylinder. Proximity enables in achieving closer positioning when compared to vision and is independent of the texture of the object. This has practical applications in satellite or aircraft industry where objects with uniform color are prevalent. It was also shown through experimentation and simulation that approximate model of the interaction matrix was good enough to achieve execution of the task. This has practical utility as it avoids estimation of local target normal at detection point which remains unknown. A study of stability analysis was performed in case of plane-to-plane positioning where an explicit form of pseudo-inverse was obtained. This leads to a new controller that was more robust to the errors in unknown target normal. This new controller also motivated further exploration of such directions in other tasks in SBC.

The multi-sensory system can also enhance the capability of a robotic manipulator to ensure safety while performing generic tasks such as assembly. Obstacle avoidance task is considered using such multiple proximity sensors by ensuring the satisfaction of an inequality. Each proximity sensor leads to a corresponding inequality and ensures reactive behavior in the robot only when it becomes close to the obstacle. In tight spaces it becomes important to estimate the target normal parameter present in the interaction matrix unlike the previous tasks considered such as plane-to-plane positioning or positioning wrt. a cylinder. Estimation algorithm involving a single sensor or multiple sensors is provided. Obstacle avoidance task is then combined with 4 point Visual Servoing to enable positioning in congested space. The final pose achieved while involving vision and proximity sensor information in control is shown to be stable to both uncertainties in interaction matrix model and to sensor noise. Various experimental and simulation results were provided to validate control architecture modeling. It was shown that it is possible to move along a narrow passage without encountering oscillations and that the robot stops without having explosive motion in case obstacles block the execution of task. Additionally obstacles encountered in the initial and final phase of the task were also appropriately

handled. This has practical benefits in assembly of satellite where such close encounters with neighbouring work-pieces or delicate structures can be prevalent while moving the workspace to the desired position.

While obtaining the explicit form for the pseudo-inverse for plane-to-plane positioning task, a set of control directions were observed. These had meaning while considering the space of interaction screws formed from the task. They form the dual basis of the basis considered for the space of interaction screws. These directions were initially identified in plane-to-plane positioning task which could be argued as a rather simple task. It was then obtained for the task of positioning wrt. a cylinder which could be considered as a bit more complex than the previous task. It was also further extended to the case of 3 point visual servoing which is a little more complex than the tasks with proximity sensors. The dual basis can be used as control directions for regulation tasks and for obtaining closed-form equations in sensor space.

This thesis has tried to emphasis on the use of Sensor-based Control methodology for performing tasks in robotics. Complementarity nature of both vision and proximity sensors has been used in addressing the research problem involved in the application area of satellite assembly using non-contact sensors.

FUTURE WORK

To end this thesis we would like to give future research perspectives based on the work done.

In Part I, we looked at positioning wrt. either a planar target or a cylinder using proximity sensors. Positioning task could be generalized to an object of generic shape. At equilibrium, it can also be interesting to consider a contour following task for such arbitrary shape. Another task that can be performed using proximity sensors is reactivereshaping of a robotic hand for aiding grasping. This task has applications in the area of human-centered robotics.

In Part II, the task of positioning in congested space only involved two proximity rings. It would be interesting to cover the arm of manipulator with multiple rings or sensors to extend the cover of local environment sensing and obtain more smoother behavior in control input. This would enable testing the developed algorithm to more complex scenarios such as the inside of satellite as considered in Figure 3 at the beginning of this document. While performing the task in such scenarios, to improve the autonomy of operation it would also be necessary to include other constraints that could arise such as joint limit avoidance, singularity avoidance, occlusion avoidance, etc. It would also be interesting to combine non-contact sensors (vision and proximity) with force sensors to the existing experimental setup. This would enable in implementing the full cycle of assembly: grasping the workpiece from a specific location, positioning it to the final goal location while traversing through workspace obstacles, inserting the part to appropriate place from the final goal (as shown in Figure 4).

The proximity sensor model considered in the thesis involved the assumption of thin-field of view. This obtained simpler interaction matrix. It could be interesting to consider a more generic proximity sensor with a detection cone. This might be more helpful in dealing with dynamic obstacles. Also it could reduce the need of number of sensors necessary to ensure safety in assembly task. There are also new sensors such as VL53L5CX¹, that provide multi-zone range map that could be effectively utilized in proximity-based control

1. <https://www.st.com/en/imaging-and-photonics-solutions/time-of-flight-sensors.html#recommended-for-you>

framework.

In Part III, we have obtained the dual basis of three tasks in Sensor-based Control. These dual basis enables us to design the desired controller in sensor space. This could be interesting in scenarios where an alternative behavior is desired than regulation of the task to an equilibrium point. One such alternate behavior that could be interesting is to converge to a stable orbit around the equilibrium point using the same task designed. Also, it could be interesting to explore on how to switch between regulation to the equilibrium point and moving around it in stable orbits.

EVALUATION OF KERNEL SPACE OF COMBINATION MATRIX FOR REDUNDANT NUMBER OF SENSORS IN PLANE-TO-PLANE POSITIONING TASK

In this Appendix we prove the uniqueness of equilibrium point for plane-to-plane positioning task while using a redundant number of sensors as discussed in Section 4.2.3. Let us reconsider the target points (5.3) using extrinsic parameters given in (4.13),

$$\begin{aligned}
 \mathbf{m}_{T_{11}} &= -(\delta_{11} + r) \sin \alpha \mathbf{i} - (\delta_{11} + r) \cos \alpha \mathbf{j} + d \mathbf{k} \\
 \mathbf{m}_{T_{21}} &= (\delta_{21} + r) \sin \alpha \mathbf{i} - (\delta_{21} + r) \cos \alpha \mathbf{j} + d \mathbf{k} \\
 \mathbf{m}_{T_{12}} &= -(\delta_{12} + r) \sin \alpha \mathbf{i} - (\delta_{12} + r) \cos \alpha \mathbf{j} - d \mathbf{k} \\
 \mathbf{m}_{T_{22}} &= (\delta_{22} + r) \sin \alpha \mathbf{i} - (\delta_{22} + r) \cos \alpha \mathbf{j} - d \mathbf{k}
 \end{aligned} \tag{A.1}$$

Let us now evaluate three vectors lying on the target plane,

$$\begin{aligned}
 \mathbf{m}_{T_{11}} - \mathbf{m}_{T_{21}} &= -(\delta_{11} + \delta_{21} + 2r) \sin \alpha \mathbf{i} - (\delta_{11} - \delta_{21}) \cos \alpha \mathbf{j} \\
 \mathbf{m}_{T_{11}} - \mathbf{m}_{T_{12}} &= -(\delta_{11} - \delta_{12}) \sin \alpha \mathbf{i} - (\delta_{11} - \delta_{12}) \cos \alpha \mathbf{j} + 2d \mathbf{k} \\
 \mathbf{m}_{T_{11}} - \mathbf{m}_{T_{22}} &= -(\delta_{11} + \delta_{22} + 2r) \sin \alpha \mathbf{i} - (\delta_{11} - \delta_{22}) \cos \alpha \mathbf{j} + 2d \mathbf{k}
 \end{aligned} \tag{A.2}$$

Since the above three vectors are coplanar, the scalar triple product is 0, which is represented as

$$\det \begin{bmatrix} (\mathbf{m}_{T_{11}} - \mathbf{m}_{T_{21}})^T \\ (\mathbf{m}_{T_{11}} - \mathbf{m}_{T_{12}})^T \\ (\mathbf{m}_{T_{11}} - \mathbf{m}_{T_{22}})^T \end{bmatrix} = 0 \quad (\text{A.3})$$

Let us evaluate the determinant,

$$\begin{aligned} & \begin{vmatrix} -(\delta_{11} + \delta_{21} + 2r) \sin \alpha & -(\delta_{11} - \delta_{21}) \cos \alpha & 0 \\ -(\delta_{11} - \delta_{12}) \sin \alpha & -(\delta_{11} - \delta_{12}) \cos \alpha & 2d \\ -(\delta_{11} + \delta_{22} + 2r) \sin \alpha & -(\delta_{11} - \delta_{22}) \cos \alpha & 2d \end{vmatrix} = 0 \\ & d \sin 2\alpha \begin{vmatrix} -(\delta_{11} + \delta_{21} + 2r) & -(\delta_{11} - \delta_{21}) & 0 \\ -(\delta_{11} - \delta_{12}) & -(\delta_{11} - \delta_{12}) & 1 \\ -(\delta_{11} + \delta_{22} + 2r) & -(\delta_{11} - \delta_{22}) & 1 \end{vmatrix} = 0 \\ & d \sin 2\alpha \left(-(\delta_{11} + \delta_{21} + 2r)(\delta_{11} - \delta_{22}) + \right. \\ & \quad \left. (\delta_{11} - \delta_{21})(\delta_{11} + \delta_{22} + 2r) + \right. \\ & \quad \left. (\delta_{11} + \delta_{21} + 2r)(\delta_{11} - \delta_{12}) - (\delta_{11} - \delta_{21})(\delta_{11} - \delta_{12}) \right) = 0 \\ & d \sin 2\alpha \left((\delta_{11} + \delta_{21} + 2r)(\delta_{22} - \delta_{12}) + \right. \\ & \quad \left. (\delta_{11} - \delta_{21})(\delta_{22} + \delta_{12} + 2r) \right) = 0 \end{aligned} \quad (\text{A.4})$$

To be a physically consistent configuration of the proximity array the above condition must be satisfied. Now let us consider the case when $\mathbf{s} - \mathbf{s}^* \in \text{Ker } \mathbf{C}$ from (4.20). The proximity signal values of each sensor can be represented in terms of the kernel space as,

$$\begin{pmatrix} \delta_{11} \\ \delta_{21} \\ \delta_{12} \\ \delta_{22} \end{pmatrix} = c \begin{pmatrix} -1 \\ 1 \\ 1 \\ -1 \end{pmatrix} + \begin{pmatrix} \delta_{11}^* \\ \delta_{21}^* \\ \delta_{12}^* \\ \delta_{22}^* \end{pmatrix} \quad (\text{A.5})$$

where $c \neq 0$ is an arbitrary scalar. Substituting (A.5) in (A.4) we get,

$$\begin{aligned}
& d \sin 2\alpha ((\delta_{11}^* + \delta_{21}^* + 2r)(\delta_{22}^* - \delta_{12}^* - 2c) + \\
& \quad (\delta_{11}^* - \delta_{21}^* - 2c)(\delta_{22}^* + \delta_{12}^* + 2r)) = 0 \\
& \quad \frac{d \sin 2\alpha ((\delta_{11}^* + \delta_{21}^* + 2r)(\delta_{22}^* - \delta_{12}^*) + \\
& \quad \quad (\delta_{11}^* - \delta_{21}^*)(\delta_{22}^* + \delta_{12}^* + 2r)) +}{2 c d \sin 2\alpha (\delta_{11}^* + \delta_{21}^* + \delta_{22}^* + \delta_{12}^* + 2r)} = 0
\end{aligned} \tag{A.6}$$

The first part of the above equation nullifies as at the desired configuration we have to satisfy (A.4). Since $\delta_{ij}^* > 0$, $\sin 2\alpha \neq 0$, $d \neq 0$ (refer to Section 4.2.2), to satisfy the above constraint we need to have $c = 0$. This essentially proves that there are no physically coherent configuration for which $\mathbf{s} - \mathbf{s}^* \in \text{Ker } \mathbf{C}$.

DECOMPOSITION OF CLOSED LOOP MATRIX FOR PSEUDO-INVERSE BASED CONTROLLER FOR PLANE-TO-PLANE POSITIONING TASK

In this Appendix we provide steps for decomposing matrix \mathbf{M}_p following the discussion in Section 5.2.3,

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

VERIFICATION OF DUAL BASIS FOR PLANE-TO-PLANE FOLLOWING TASK

Let us consider the screw product between an element of dual basis \mathcal{K}_i and elements of set H consisting of interactions screws based on the description in Section 10.3.

$$\begin{aligned}
& \mathbf{K}_i \circ \mathbf{H}_i \\
&= \gamma_i (\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}}, \mathbf{m}_{T_{i-}} - \mathbf{m}_{T_{i+}}) \circ -\frac{1}{\mathbf{n}_T \cdot \mathbf{n}_{S_i}} (\mathbf{m}_{T_i} \times \mathbf{n}_T, \mathbf{n}_T) \\
&= \frac{1}{l} \left((\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}}) \cdot \mathbf{n}_T + (\mathbf{m}_{T_{i-}} - \mathbf{m}_{T_{i+}})(\mathbf{m}_{T_i} \times \mathbf{n}_T) \right) \\
&= \frac{1}{l} \left((\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}} + \mathbf{m}_{T_{i-}} \times \mathbf{m}_{T_i} + \mathbf{m}_{T_i} \times \mathbf{m}_{T_{i+}}) \cdot \mathbf{n}_T \right) \\
&= 1
\end{aligned} \tag{C.1}$$

$$\begin{aligned}
& \mathbf{K}_i \circ \mathbf{H}_{i+} \\
&= \gamma_i (\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}}, \mathbf{m}_{T_{i-}} - \mathbf{m}_{T_{i+}}) \circ -\frac{1}{\mathbf{n}_T \cdot \mathbf{n}_{S_{i+}}} (\mathbf{m}_{T_{i+}} \times \mathbf{n}_T, \mathbf{n}_T) \\
&= \frac{\mathbf{n}_T \cdot \mathbf{n}_{S_i}}{l(\mathbf{n}_T \cdot \mathbf{n}_{S_{i+}})} \left((\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}} + \mathbf{m}_{T_{i-}} \times \mathbf{m}_{T_{i+}} + \mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i+}}) \cdot \mathbf{n}_T \right) \\
&= 0
\end{aligned} \tag{C.2}$$

$$\begin{aligned}
& \mathbf{K}_i \circ \mathbf{H}_{i-} \\
&= \gamma_i (\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}}, \mathbf{m}_{T_{i-}} - \mathbf{m}_{T_{i+}}) \circ -\frac{1}{\mathbf{n}_T \cdot \mathbf{n}_{S_{i-}}} (\mathbf{m}_{T_{i-}} \times \mathbf{n}_T, \mathbf{n}_T) \\
&= \frac{\mathbf{n}_T \cdot \mathbf{n}_{S_i}}{l(\mathbf{n}_T \cdot \mathbf{n}_{S_{i-}})} \left((\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}} + \mathbf{m}_{T_{i-}} \times \mathbf{m}_{T_{i-}} + \mathbf{m}_{T_{i-}} \times \mathbf{m}_{T_{i+}}) \cdot \mathbf{n}_T \right) \\
&= 0
\end{aligned} \tag{C.3}$$

VERIFICATION OF DUAL BASIS FOR POSITIONING TASK WRT. A CYLINDER

Let us consider the screw product between an element of dual basis \mathcal{K}_i and elements of set H consisting of interactions screws based on the description in Section 10.4,

$$\begin{aligned}
\mathbf{K}_i \circ \mathbf{H}_i &= \frac{-\mathbf{n}_{T_i} \cdot \mathbf{n}_{S_i}}{l_i} \left(a_{1i} \mathbf{m}_{T_{i+}} \times \mathbf{n}_{T_{i-}} - a_{2i} \mathbf{m}_{T_{i-}} \times \mathbf{n}_{T_{i+}}, a_{1i} \mathbf{n}_{T_{i-}} - a_{2i} \mathbf{n}_{T_{i+}} \right) \circ \\
&\quad - \frac{1}{\mathbf{n}_{T_i} \cdot \mathbf{n}_{S_i}} (\mathbf{m}_{T_i} \times \mathbf{n}_{T_i}, \mathbf{n}_{T_i}) \\
&= \frac{1}{l_i} \left(a_{1i} \mathbf{m}_{T_i} \cdot (\mathbf{n}_{T_i} \times \mathbf{n}_{T_{i-}}) - a_{2i} \mathbf{m}_{T_i} \cdot (\mathbf{n}_{T_i} \times \mathbf{n}_{T_{i+}}) \right. \\
&\quad \left. + a_{1i} \mathbf{m}_{T_{i+}} \cdot (\mathbf{n}_{T_{i-}} \times \mathbf{n}_{T_i}) - a_{2i} \mathbf{m}_{T_{i-}} \cdot (\mathbf{n}_{T_{i+}} \times \mathbf{n}_{T_i}) \right) \\
&= \frac{1}{l_i} \left(a_{1i} (\mathbf{m}_{T_i} - \mathbf{m}_{T_{i+}}) \cdot (\mathbf{n}_{T_i} \times \mathbf{n}_{T_{i-}}) - a_{2i} (\mathbf{m}_{T_i} - \mathbf{m}_{T_{i-}}) \cdot (\mathbf{n}_{T_i} \times \mathbf{n}_{T_{i+}}) \right) \\
&= 1 \tag{D.1}
\end{aligned}$$

$$\begin{aligned}
\mathbf{K}_i \circ \mathbf{H}_{i+} &= \frac{-\mathbf{n}_{T_i} \cdot \mathbf{n}_{S_i}}{l_i} \left(a_{1i} \mathbf{m}_{T_{i+}} \times \mathbf{n}_{T_{i-}} - a_{2i} \mathbf{m}_{T_{i-}} \times \mathbf{n}_{T_{i+}}, a_{1i} \mathbf{n}_{T_{i-}} - a_{2i} \mathbf{n}_{T_{i+}} \right) \circ \\
&\quad - \frac{1}{\mathbf{n}_{T_{i+}} \cdot \mathbf{n}_{S_{i+}}} (\mathbf{m}_{T_{i+}} \times \mathbf{n}_{T_{i+}}, \mathbf{n}_{T_{i+}}) \\
&= \frac{\mathbf{n}_{T_i} \cdot \mathbf{n}_{S_i}}{l_i (\mathbf{n}_{T_{i+}} \cdot \mathbf{n}_{S_{i+}})} \left(a_{1i} \mathbf{m}_{T_{i+}} \cdot (\mathbf{n}_{T_{i+}} \times \mathbf{n}_{T_{i-}}) - a_{2i} \mathbf{m}_{T_{i+}} \cdot (\mathbf{n}_{T_{i+}} \times \mathbf{n}_{T_{i+}}) \right. \\
&\quad \left. + a_{1i} \mathbf{m}_{T_{i+}} \cdot (\mathbf{n}_{T_{i-}} \times \mathbf{n}_{T_{i+}}) - a_{2i} \mathbf{m}_{T_{i-}} \cdot (\mathbf{n}_{T_{i+}} \times \mathbf{n}_{T_{i+}}) \right) \\
&= \frac{\mathbf{n}_{T_i} \cdot \mathbf{n}_{S_i}}{l_i (\mathbf{n}_{T_{i+}} \cdot \mathbf{n}_{S_{i+}})} \left(a_{1i} (\mathbf{m}_{T_{i+}} - \mathbf{m}_{T_{i+}}) \cdot (\mathbf{n}_{T_{i+}} \times \mathbf{n}_{T_{i-}}) \right) \\
&= 0 \tag{D.2}
\end{aligned}$$

$$\begin{aligned}
\mathbf{K}_i \circ \mathbf{H}_{i++} &= \frac{-\mathbf{n}_{T_i} \cdot \mathbf{n}_{S_i}}{l_i} \left(a_{1i} \mathbf{m}_{T_{i+}} \times \mathbf{n}_{T_{i-}} - a_{2i} \mathbf{m}_{T_{i-}} \times \mathbf{n}_{T_{i+}}, a_{1i} \mathbf{n}_{T_{i-}} - a_{2i} \mathbf{n}_{T_{i+}} \right) \circ \\
&\quad - \frac{1}{\mathbf{n}_{T_{i++}} \cdot \mathbf{n}_{S_{i++}}} \left(\mathbf{m}_{T_{i++}} \times \mathbf{n}_{T_{i++}}, \mathbf{n}_{T_{i++}} \right) \\
&= \frac{\mathbf{n}_{T_i} \cdot \mathbf{n}_{S_i}}{l_i (\mathbf{n}_{T_{i++}} \cdot \mathbf{n}_{S_{i++}})} \left(a_{1i} \mathbf{m}_{T_{i++}} \cdot (\mathbf{n}_{T_{i++}} \times \mathbf{n}_{T_{i-}}) - a_{2i} \mathbf{m}_{T_{i++}} \cdot (\mathbf{n}_{T_{i++}} \times \mathbf{n}_{T_{i+}}) \right. \\
&\quad \left. + a_{1i} \mathbf{m}_{T_{i+}} \cdot (\mathbf{n}_{T_{i-}} \times \mathbf{n}_{T_{i++}}) - a_{2i} \mathbf{m}_{T_{i-}} \cdot (\mathbf{n}_{T_{i+}} \times \mathbf{n}_{T_{i++}}) \right) \\
&= \frac{\mathbf{n}_{T_i} \cdot \mathbf{n}_{S_i}}{l_i (\mathbf{n}_{T_{i++}} \cdot \mathbf{n}_{S_{i++}})} \left(a_{1i} (\mathbf{m}_{T_{i++}} - \mathbf{m}_{T_{i+}}) \cdot (\mathbf{n}_{T_{i++}} \times \mathbf{n}_{T_{i-}}) - \right. \\
&\quad \left. a_{2i} (\mathbf{m}_{T_{i++}} - \mathbf{m}_{T_{i-}}) \cdot (\mathbf{n}_{T_{i++}} \times \mathbf{n}_{T_{i+}}) \right) \\
&= \frac{\mathbf{n}_{T_i} \cdot \mathbf{n}_{S_i}}{l_i (\mathbf{n}_{T_{i++}} \cdot \mathbf{n}_{S_{i++}})} (a_{1i} a_{2i} - a_{2i} a_{1i}) \\
&= 0
\end{aligned} \tag{D.3}$$

$$\begin{aligned}
\mathbf{K}_i \circ \mathbf{H}_{i-} &= \frac{-\mathbf{n}_{T_i} \cdot \mathbf{n}_{S_i}}{l_i} \left(a_{1i} \mathbf{m}_{T_{i+}} \times \mathbf{n}_{T_{i-}} - a_{2i} \mathbf{m}_{T_{i-}} \times \mathbf{n}_{T_{i+}}, a_{1i} \mathbf{n}_{T_{i-}} - a_{2i} \mathbf{n}_{T_{i+}} \right) \circ \\
&\quad - \frac{1}{\mathbf{n}_{T_{i-}} \cdot \mathbf{n}_{S_{i-}}} \left(\mathbf{m}_{T_{i-}} \times \mathbf{n}_{T_{i-}}, \mathbf{n}_{T_{i-}} \right) \\
&= \frac{\mathbf{n}_{T_i} \cdot \mathbf{n}_{S_i}}{l_i (\mathbf{n}_{T_{i-}} \cdot \mathbf{n}_{S_{i-}})} \left(a_{1i} \mathbf{m}_{T_{i-}} \cdot (\mathbf{n}_{T_{i-}} \times \mathbf{n}_{T_{i-}}) - a_{2i} \mathbf{m}_{T_{i-}} \cdot (\mathbf{n}_{T_{i-}} \times \mathbf{n}_{T_{i+}}) \right. \\
&\quad \left. + a_{1i} \mathbf{m}_{T_{i+}} \cdot (\mathbf{n}_{T_{i-}} \times \mathbf{n}_{T_{i-}}) - a_{2i} \mathbf{m}_{T_{i-}} \cdot (\mathbf{n}_{T_{i+}} \times \mathbf{n}_{T_{i-}}) \right) \\
&= 0
\end{aligned} \tag{D.4}$$

VERIFICATION OF DUAL BASIS FOR 3 POINT VISUAL SERVOING

Let us consider the screw product between an element of dual basis \mathbf{K}_{ix} and elements of set H consisting of interactions screws based on the description in Section 10.5. We do not have to evaluate for \mathbf{K}_{iy} as the difference is only involved in \hat{i} and \hat{j} coordinates.

$\mathbf{K}_{ix} \circ \mathbf{H}_{ix}$

$$\begin{aligned}
&= \gamma_i \left((a_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{T_{i-}} \times \mathbf{m}_{T_{i+}}) - (\mathbf{b}_i \cdot \hat{\mathbf{i}}) \mathbf{q}_i, (a_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{T_{i+}} \right. \\
&\quad \left. - \mathbf{m}_{T_{i-}}) - (b_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}}) \right) \circ \\
&\quad \left(\mathbf{m}_{T_i} \times \frac{\mathbf{m}_{T_i} \times \hat{\mathbf{j}}}{(\mathbf{m}_{T_i} \cdot \hat{\mathbf{k}})^2}, \frac{\mathbf{m}_{T_i} \times \hat{\mathbf{j}}}{(\mathbf{m}_{T_i} \cdot \hat{\mathbf{k}})^2} \right) \\
&= \frac{1}{\begin{vmatrix} a_i \cdot \hat{\mathbf{i}} & a_i \cdot \hat{\mathbf{j}} \\ b_i \cdot \hat{\mathbf{i}} & b_i \cdot \hat{\mathbf{j}} \end{vmatrix}} \left((a_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{T_{i-}} \times \mathbf{m}_{T_{i+}} + \mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_i} + \mathbf{m}_{T_i} \times \mathbf{m}_{T_{i-}}) \cdot (\mathbf{m}_{T_i} \times \hat{\mathbf{j}}) \right. \\
&\quad \left. - (b_i \cdot \hat{\mathbf{i}}) (\mathbf{q}_i \cdot (\mathbf{m}_{T_i} \times \hat{\mathbf{j}}) + (\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}}) \cdot (\mathbf{m}_{T_i} \times (\mathbf{m}_{T_i} \times \hat{\mathbf{j}}))) \right) \\
&= \frac{1}{\begin{vmatrix} a_i \cdot \hat{\mathbf{i}} & a_i \cdot \hat{\mathbf{j}} \\ b_i \cdot \hat{\mathbf{i}} & b_i \cdot \hat{\mathbf{j}} \end{vmatrix}} \left((a_i \cdot \hat{\mathbf{i}}) \mathbf{l}_{\mathbf{n}_T} \cdot (\mathbf{m}_{T_i} \times \hat{\mathbf{j}}) - (\mathbf{b}_i \cdot \hat{\mathbf{i}}) ((\mathbf{m}_{T_{i-}} \cdot \mathbf{m}_{T_{i-}}) \mathbf{m}_{T_{i+}} \right. \\
&\quad \left. - (\mathbf{m}_{T_{i+}} \cdot \mathbf{m}_{T_{i+}}) \mathbf{m}_{T_{i-}} - \mathbf{m}_{T_i} \times (\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}})) \cdot (\mathbf{m}_{T_i} \times \hat{\mathbf{j}}) \right) \\
&= \frac{1}{\begin{vmatrix} a_i \cdot \hat{\mathbf{i}} & a_i \cdot \hat{\mathbf{j}} \\ b_i \cdot \hat{\mathbf{i}} & b_i \cdot \hat{\mathbf{j}} \end{vmatrix}} \left((a_i \cdot \hat{\mathbf{i}}) (\mathbf{l}_{\mathbf{n}_T} \times \mathbf{m}_{T_i}) \cdot \hat{\mathbf{j}} - (\mathbf{b}_i \cdot \hat{\mathbf{i}}) ((\mathbf{m}_{T_{i-}} \cdot \mathbf{m}_{T_{i-}}) \mathbf{m}_{T_{i+}} \right. \\
&\quad \left. - (\mathbf{m}_{T_{i+}} \cdot \mathbf{m}_{T_{i+}}) \mathbf{m}_{T_{i-}} - (\mathbf{m}_{T_i} \cdot \mathbf{m}_{T_{i-}}) \times \mathbf{m}_{T_{i+}} + (\mathbf{m}_{T_i} \cdot \mathbf{m}_{T_{i+}}) \mathbf{m}_{T_{i-}}) \cdot (\mathbf{m}_{T_i} \times \hat{\mathbf{j}}) \right) \\
&= \frac{1}{\begin{vmatrix} a_i \cdot \hat{\mathbf{i}} & a_i \cdot \hat{\mathbf{j}} \\ b_i \cdot \hat{\mathbf{i}} & b_i \cdot \hat{\mathbf{j}} \end{vmatrix}} \left((a_i \cdot \hat{\mathbf{i}}) (\mathbf{b}_i \cdot \hat{\mathbf{j}}) \right. \\
&\quad \left. - (b_i \cdot \hat{\mathbf{i}}) ((\mathbf{m}_{T_{i-}} \cdot (\mathbf{m}_{T_{i-}} - \mathbf{m}_{T_i})) \mathbf{m}_{T_{i+}} - (\mathbf{m}_{T_{i+}} \cdot (\mathbf{m}_{T_{i+}} - \mathbf{m}_{T_i})) \mathbf{m}_{T_{i-}}) \cdot (\mathbf{m}_{T_i} \times \hat{\mathbf{j}}) \right) \\
&= \frac{1}{\begin{vmatrix} a_i \cdot \hat{\mathbf{i}} & a_i \cdot \hat{\mathbf{j}} \\ b_i \cdot \hat{\mathbf{i}} & b_i \cdot \hat{\mathbf{j}} \end{vmatrix}} \left((a_i \cdot \hat{\mathbf{i}}) (\mathbf{b}_i \cdot \hat{\mathbf{j}}) - (\mathbf{b}_i \cdot \hat{\mathbf{i}}) (\mathbf{p}_i \cdot (\mathbf{m}_{T_i} \times \hat{\mathbf{j}})) \right) \\
&= \frac{1}{\begin{vmatrix} a_i \cdot \hat{\mathbf{i}} & a_i \cdot \hat{\mathbf{j}} \\ b_i \cdot \hat{\mathbf{i}} & b_i \cdot \hat{\mathbf{j}} \end{vmatrix}} \left((a_i \cdot \hat{\mathbf{i}}) (\mathbf{b}_i \cdot \hat{\mathbf{j}}) - (\mathbf{b}_i \cdot \hat{\mathbf{i}}) (\mathbf{a}_i \cdot \hat{\mathbf{j}}) \right) \\
&= 1
\end{aligned} \tag{E.1}$$

$$\begin{aligned}
& \mathbf{K}_{ix} \circ \mathbf{H}_{iy} \\
&= -\gamma_i \left((a_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{T_{i-}} \times \mathbf{m}_{T_{i+}}) - (\mathbf{b}_i \cdot \hat{\mathbf{i}}) \mathbf{q}_i, (a_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{T_{i+}} \right. \\
&\quad \left. - \mathbf{m}_{T_{i-}}) - (b_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}}) \right) \circ \\
&\quad \left(\mathbf{m}_{T_i} \times \frac{\mathbf{m}_{T_i} \times \hat{\mathbf{i}}}{(\mathbf{m}_{T_i} \cdot \hat{\mathbf{k}})^2}, \frac{\mathbf{m}_{T_i} \times \hat{\mathbf{i}}}{(\mathbf{m}_{T_i} \cdot \hat{\mathbf{k}})^2} \right) \\
&= \frac{-1}{\begin{vmatrix} a_i \cdot \hat{\mathbf{i}} & a_i \cdot \hat{\mathbf{j}} \\ b_i \cdot \hat{\mathbf{i}} & b_i \cdot \hat{\mathbf{j}} \end{vmatrix}} \left((a_i \cdot \hat{\mathbf{i}}) (\mathbf{b}_i \cdot \hat{\mathbf{i}}) - (\mathbf{b}_i \cdot \hat{\mathbf{i}}) (a_i \cdot \hat{\mathbf{i}}) \right) \\
&= 0
\end{aligned} \tag{E.2}$$

$$\begin{aligned}
& \mathbf{K}_{ix} \circ \mathbf{H}_{i+x} \\
&= \gamma_i \left((a_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{T_{i-}} \times \mathbf{m}_{T_{i+}}) - (\mathbf{b}_i \cdot \hat{\mathbf{i}}) \mathbf{q}_i, (a_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{T_{i+}} \right. \\
&\quad \left. - \mathbf{m}_{T_{i-}}) - (b_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}}) \right) \circ \\
&\quad \left(\mathbf{m}_{T_{i+}} \times \frac{\mathbf{m}_{T_{i+}} \times \hat{\mathbf{j}}}{(\mathbf{m}_{T_{i+}} \cdot \hat{\mathbf{k}})^2}, \frac{\mathbf{m}_{T_{i+}} \times \hat{\mathbf{j}}}{(\mathbf{m}_{T_{i+}} \cdot \hat{\mathbf{k}})^2} \right) \\
&= \frac{\gamma_i}{(\mathbf{m}_{T_{i+}} \cdot \hat{\mathbf{k}})^2} \left((a_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{T_{i-}} \times \mathbf{m}_{T_{i+}} + \mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i+}} + \mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}}) \cdot (\mathbf{m}_{T_i} \times \hat{\mathbf{j}}) \right) \\
&\quad - (b_i \cdot \hat{\mathbf{i}}) (\mathbf{q}_i \cdot (\mathbf{m}_{T_{i+}} \times \hat{\mathbf{j}}) + (\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}}) \cdot (\mathbf{m}_{T_{i+}} \times (\mathbf{m}_{T_{i+}} \times \hat{\mathbf{j}}))) \\
&= \frac{\gamma_i}{(\mathbf{m}_{T_{i+}} \cdot \hat{\mathbf{k}})^2} \left((a_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{T_{i-}} \times \mathbf{m}_{T_{i+}} + \mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i+}} + \mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}}) \cdot (\mathbf{m}_{T_i} \times \hat{\mathbf{j}}) \right) \\
&\quad - (b_i \cdot \hat{\mathbf{i}}) \left((\mathbf{m}_{T_{i-}} \cdot \mathbf{m}_{T_{i-}}) \mathbf{m}_{T_{i+}} - (\mathbf{m}_{T_{i+}} \cdot \mathbf{m}_{T_{i+}}) \mathbf{m}_{T_{i-}} \right. \\
&\quad \left. - \mathbf{m}_{T_{i+}} \times (\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}}) \right) \cdot (\mathbf{m}_{T_{i+}} \times \hat{\mathbf{j}}) \\
&= \frac{\gamma_i}{(\mathbf{m}_{T_{i+}} \cdot \hat{\mathbf{k}})^2} \left(- (b_i \cdot \hat{\mathbf{i}}) \left((\mathbf{m}_{T_{i-}} \cdot \mathbf{m}_{T_{i-}}) \mathbf{m}_{T_{i+}} - \mathbf{m}_{T_{i+}} \cdot \mathbf{m}_{T_{i+}} \mathbf{m}_{T_{i-}} \right. \right. \\
&\quad \left. \left. - (\mathbf{m}_{T_{i+}} \cdot \mathbf{m}_{T_{i-}}) \mathbf{m}_{T_{i+}} + (\mathbf{m}_{T_{i+}} \cdot \mathbf{m}_{T_{i+}}) \mathbf{m}_{T_{i-}} \right) \cdot (\mathbf{m}_{T_{i+}} \times \hat{\mathbf{j}}) \right) \\
&= \frac{\gamma_i}{(\mathbf{m}_{T_{i+}} \cdot \hat{\mathbf{k}})^2} \left(- (b_i \cdot \hat{\mathbf{i}}) \left((\mathbf{m}_{T_{i-}} - \mathbf{m}_{T_{i+}}) \cdot \mathbf{m}_{T_{i-}} \right) \mathbf{m}_{T_{i+}} \cdot (\mathbf{m}_{T_{i+}} \times \hat{\mathbf{j}}) \right) \\
&= 0
\end{aligned} \tag{E.3}$$

$$\begin{aligned}
& \mathbf{K}_{ix} \circ \mathbf{H}_{i+y} \\
&= \gamma_i \left((a_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{T_{i-}} \times \mathbf{m}_{T_{i+}}) - (b_i \cdot \hat{\mathbf{i}}) \mathbf{q}_i, (a_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{T_{i+}} \right. \\
&\quad \left. - \mathbf{m}_{T_{i-}}) - (b_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}}) \right) \circ \\
&\quad \left(\mathbf{m}_{T_{i+}} \times \frac{\mathbf{m}_{T_{i+}} \times \hat{\mathbf{i}}}{(\mathbf{m}_{T_{i+}} \cdot \hat{\mathbf{k}})^2}, \frac{\mathbf{m}_{T_{i+}} \times \hat{\mathbf{i}}}{(\mathbf{m}_{T_{i+}} \cdot \hat{\mathbf{k}})^2} \right) \\
&= \frac{-\gamma_i}{(\mathbf{m}_{T_{i+}} \cdot \hat{\mathbf{k}})^2} \left(-(b_i \cdot \hat{\mathbf{i}}) ((\mathbf{m}_{T_{i-}} - \mathbf{m}_{T_{i+}}) \cdot \mathbf{m}_{T_{i-}}) \mathbf{m}_{T_{i+}} \right) \cdot (\mathbf{m}_{T_{i+}} \times \hat{\mathbf{i}}) \\
&= 0 \tag{E.4}
\end{aligned}$$

$$\begin{aligned}
& \mathbf{K}_{ix} \circ \mathbf{H}_{i-x} \\
&= \gamma_i \left((a_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{T_{i-}} \times \mathbf{m}_{T_{i+}}) - (b_i \cdot \hat{\mathbf{i}}) \mathbf{q}_i, (a_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{T_{i+}} \right. \\
&\quad \left. - \mathbf{m}_{T_{i-}}) - (b_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}}) \right) \circ \\
&\quad \left(\mathbf{m}_{T_{i-}} \times \frac{\mathbf{m}_{T_{i-}} \times \hat{\mathbf{j}}}{(\mathbf{m}_{T_{i-}} \cdot \hat{\mathbf{k}})^2}, \frac{\mathbf{m}_{T_{i-}} \times \hat{\mathbf{j}}}{(\mathbf{m}_{T_{i-}} \cdot \hat{\mathbf{k}})^2} \right) \\
&= \frac{\gamma_i}{(\mathbf{m}_{T_{i-}} \cdot \hat{\mathbf{k}})^2} \left((a_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{T_{i-}} \times \mathbf{m}_{T_{i+}}) \cdot (\mathbf{m}_{T_{i-}} \times \hat{\mathbf{j}}) - (b_i \cdot \hat{\mathbf{i}}) \mathbf{q}_i \cdot (\mathbf{m}_{T_{i-}} \times \hat{\mathbf{j}}) \right. \\
&\quad \left. (a_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{T_{i+}} - \mathbf{m}_{T_{i-}}) \cdot (\mathbf{m}_{T_{i-}} \times (\mathbf{m}_{T_{i-}} \times \hat{\mathbf{j}})) \right. \\
&\quad \left. - (b_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}}) \cdot (\mathbf{m}_{T_{i-}} \times (\mathbf{m}_{T_{i-}} \times \hat{\mathbf{j}})) \right) \\
&= \frac{\gamma_i}{(\mathbf{m}_{T_{i-}} \cdot \hat{\mathbf{k}})^2} \left((a_i \cdot \hat{\mathbf{i}}) (\cancel{\mathbf{m}_{T_{i-}} \times \mathbf{m}_{T_{i+}}} + \cancel{\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}}} + \mathbf{m}_{T_{i-}} \times \mathbf{m}_{T_{i-}}) \cdot (\mathbf{m}_{T_{i-}} \times \hat{\mathbf{j}}) \right. \\
&\quad \left. - (b_i \cdot \hat{\mathbf{i}}) (\mathbf{q}_i \cdot (\mathbf{m}_{T_{i-}} \times \hat{\mathbf{j}}) + (\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}}) \cdot (\mathbf{m}_{T_{i-}} \times (\mathbf{m}_{T_{i-}} \times \hat{\mathbf{j}}))) \right) \\
&= \frac{\gamma_i}{(\mathbf{m}_{T_{i-}} \cdot \hat{\mathbf{k}})^2} \left(-(b_i \cdot \hat{\mathbf{i}}) (((\mathbf{m}_{T_{i-}} \cdot \mathbf{m}_{T_{i-}}) \mathbf{m}_{T_{i+}} - (\mathbf{m}_{T_{i+}} \cdot \mathbf{m}_{T_{i+}}) \mathbf{m}_{T_{i-}} \right. \\
&\quad \left. - \mathbf{m}_{T_{i-}} \times (\mathbf{m}_{T_{i+}} \times \mathbf{m}_{T_{i-}})) \cdot (\mathbf{m}_{T_{i-}} \times \hat{\mathbf{j}}) \right. \\
&= \frac{\gamma_i}{(\mathbf{m}_{T_{i-}} \cdot \hat{\mathbf{k}})^2} \left(-(b_i \cdot \hat{\mathbf{i}}) (((\cancel{\mathbf{m}_{T_{i-}} \cdot \mathbf{m}_{T_{i-}}} \mathbf{m}_{T_{i+}} - (\mathbf{m}_{T_{i+}} \cdot \mathbf{m}_{T_{i+}}) \mathbf{m}_{T_{i-}} \right. \\
&\quad \left. - \cancel{(\mathbf{m}_{T_{i-}} \cdot \mathbf{m}_{T_{i-}}) \mathbf{m}_{T_{i+}}} + (\mathbf{m}_{T_{i-}} \cdot \mathbf{m}_{T_{i+}}) \mathbf{m}_{T_{i-}})) \cdot (\mathbf{m}_{T_{i-}} \times \hat{\mathbf{j}}) \right) \\
&= 0 \tag{E.5}
\end{aligned}$$

$$\begin{aligned}
\mathbf{K}_{ix} \circ \mathbf{H}_{i-y} &= \gamma_i \left((a_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{\mathbf{T}_{i-}} \times \mathbf{m}_{\mathbf{T}_{i+}}) - (b_i \cdot \hat{\mathbf{i}}) \mathbf{q}_i, (a_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{\mathbf{T}_{i+}} \right. \\
&\quad \left. - \mathbf{m}_{\mathbf{T}_{i-}}) - (b_i \cdot \hat{\mathbf{i}}) (\mathbf{m}_{\mathbf{T}_{i+}} \times \mathbf{m}_{\mathbf{T}_{i-}}) \right) \circ \\
&\quad \left(\mathbf{m}_{\mathbf{T}_{i-}} \times \frac{\mathbf{m}_{\mathbf{T}_{i-}} \times \hat{\mathbf{i}}}{(\mathbf{m}_{\mathbf{T}_{i-}} \cdot \hat{\mathbf{k}})^2}, \frac{\mathbf{m}_{\mathbf{T}_{i-}} \times \hat{\mathbf{i}}}{(\mathbf{m}_{\mathbf{T}_{i-}} \cdot \hat{\mathbf{k}})^2} \right) \\
&= \frac{-\gamma_i}{(\mathbf{m}_{\mathbf{T}_{i-}} \cdot \hat{\mathbf{k}})^2} \left(-(b_i \cdot \hat{\mathbf{i}}) \left(\cancel{(\mathbf{m}_{\mathbf{T}_{i-}} \cdot \mathbf{m}_{\mathbf{T}_{i-}}) \mathbf{m}_{\mathbf{T}_{i+}}} - (\mathbf{m}_{\mathbf{T}_{i+}} \cdot \mathbf{m}_{\mathbf{T}_{i+}}) \mathbf{m}_{\mathbf{T}_{i-}} \right. \right. \\
&\quad \left. \left. - \cancel{(\mathbf{m}_{\mathbf{T}_{i-}} \cdot \mathbf{m}_{\mathbf{T}_{i-}}) \mathbf{m}_{\mathbf{T}_{i+}}} + (\mathbf{m}_{\mathbf{T}_{i-}} \cdot \mathbf{m}_{\mathbf{T}_{i+}}) \mathbf{m}_{\mathbf{T}_{i-}} \right) \cdot (\mathbf{m}_{\mathbf{T}_{i-}} \times \hat{\mathbf{i}}) \right) \\
&= 0
\end{aligned} \tag{E.6}$$

BIBLIOGRAPHY

- Agravante, D. J., Claudio, G., Spindler, F., & Chaumette, F., (2017), Visual servoing in an optimization framework for the whole-body control of humanoid robots, *IEEE Robotics and Automation Letters*, 2(2), 608–615.
- Allibert, G., Hua, M.-D., Krupinski, S., & Hamel, T., (2019), Pipeline following by visual servoing for autonomous underwater vehicles, *Control Engineering Practice*, 82, 151–160.
- Ames, A. D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., & Tabuada, P., (2019), Control barrier functions: theory and applications, *18th European Control Conference (ECC)*, 3420–3431.
- Ames, A. D., Xu, X., Grizzle, J. W., & Tabuada, P., (2017), Control barrier function based quadratic programs for safety critical systems, *IEEE Transactions on Automatic Control*, 62(8), 3861–3876.
- Andre, G., (1985), A multiproximity sensor system for the guidance of robot end-effectors, *5th Int. Conference on Robotics, Vision and Sensory Control*, 483–493.
- Badino, H., Huber, D., Park, Y., & Kanade, T., (2011), Fast and accurate computation of surface normals from range images, *IEEE International Conference on Robotics and Automation*, 3084–3091.
- Bambade, A., El-Kazdadi, S., Taylor, A., & Carpentier, J., (2022), PROX-QP: Yet another Quadratic Programming Solver for Robotics and beyond, *RSS 2022 - Robotics: Science and Systems*.
- Best, M. J., & Chakravarti, N., (1990), Stability of linearly constrained convex quadratic programs, *Journal of Optimization Theory and Applications*, 64, 43–53.
- Beyer, A., Grunwald, G., Heumos, M., Schedl, M., Bayer, R., Bertleff, W., Brunner, B., Burger, R., Butterfaß, J., Gruber, R., et al., (2018), Caesar: space robotics technology for assembly, maintenance, and repair, *Proceedings of the International Astronautical Congress, IAC*.
- Blanchini, F., (1999), Set invariance in control, *Automatica*, 35(11), 1747–1767.

-
- Briot, S., Chaumette, F., & Martinet, P., (2017), Revisiting the determination of the singularity cases in the visual servoing of image points through the concept of hidden robot, *IEEE Transactions on Robotics*, 33(3), 536–546.
- Buizza Avanzini, G., Ceriani, N. M., Zanchettin, A. M., Rocco, P., & Bascetta, L., (2014), Safety control of industrial robots based on a distributed distance sensor, *IEEE Transactions on Control Systems Technology*, 22(6), 2127–2140.
- Bullo, F., & Lewis, A. D., (2005), Linear and multilinear algebra. In *Geometric control of mechanical systems: modeling, analysis, and design for simple mechanical control systems* (pp. 15–48), Springer New York.
- Ceriani, N. M., Buizza Avanzini, G., Zanchettin, A. M., Bascetta, L., & Rocco, P., (2013), Optimal placement of spots in distributed proximity sensors for safe human-robot interaction, *IEEE International Conference on Robotics and Automation*, 5858–5863.
- Chaumette, F., (1994), Visual servoing using image features defined upon geometrical primitives, *33rd IEEE Conf. on Decision and Control (CDC)*, 4, 3782–3787 vol.4.
- Chaumette, F., (1998), Potential problems of stability and convergence in image-based and position-based visual servoing. In *The confluence of vision and control* (pp. 66–78), Springer, London.
- Chaumette, F., & Hutchinson, S., (2006), Visual servo control. i. basic approaches, *IEEE Robotics and Automation Magazine*, 13(4), 82–90.
- Chaumette, F., & Hutchinson, S., (2007), Visual servo control. ii. advanced approaches [tutorial], *IEEE Robotics and Automation Magazine*, 14(1), 109–118.
- Chaumette, F., Hutchinson, S., & Corke, P., (2016), Visual servoing. In *Springer handbook of robotics* (pp. 841–866).
- Chaumette, F., Rives, P., & Espiau, B., (1993), Classification and realization of the different vision-based tasks. In *Visual Servoing* (pp. 199–228, Vol. 7).
- Cheng, G., Dean-Leon, E., Bergner, F., Rogelio Guadarrama Olvera, J., Leboutet, Q., & Mittendorf, P., (2019), A comprehensive realization of robot skin: sensors, sensing, control, and applications, *Proceedings of the IEEE*, 107(10), 2034–2051.
- Cherubini, A., & Chaumette, F., (2013), Visual navigation of a mobile robot with laser-based collision avoidance, *The International Journal of Robotics Research*, 32(2), 189–205.

-
- Cherubini, A., & Navarro-Alarcon, D., (2021), Sensor-based control for collaborative robots: fundamentals, challenges, and opportunities, *Frontiers in Neurorobotics*, 14.
- Cherubini, A., Spindler, F., & Chaumette, F., (2014), Autonomous visual navigation and laser-based moving obstacle avoidance, *IEEE Transactions on Intelligent Transportation Systems*, 15(5), 2101–2110.
- Cheung, E., & Lumelsky, V., (1988), Motion planning for robot arm manipulators with proximity sensing, *IEEE International Conference on Robotics and Automation*, 740–745 vol.2.
- Cheung, E., & Lumelsky, V., (1989), Development of sensitive skin for a 3d robot arm operating in an uncertain environment, *International Conference on Robotics and Automation*, 1056–1061 vol.2.
- Cheung, E., & Lumelsky, V., (1992), A sensitive skin system for motion control of robot arm manipulators, *Robotics and Autonomous Systems*, 10(1), 9–32.
- Chiaverini, S., (1997), Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators, *IEEE Transactions on Robotics and Automation*, 13(3), 398–410.
- Cutkosky, M. R., Howe, R. D., & Provancher, W. R., (2008), Force and tactile sensors, In B. Siciliano & O. Khatib (Eds.), *Springer handbook of robotics* (pp. 455–476), Springer Berlin Heidelberg.
- Decre, W., Smits, R., Bruyninckx, H., & De Schutter, J., (2009), Extending itasc to support inequality constraints and non-instantaneous task specification, *2009 IEEE International Conference on Robotics and Automation*, 964–971.
- Devigne, L., Narayanan, V. K., Pasteau, F., & Babel, M., (2016), Low complex sensor-based shared control for power wheelchair navigation, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5434–5439.
- Devigne, L., Pasteau, F., Carlson, T., & Babel, M., (2019), A shared control solution for safe assisted power wheelchair navigation in an environment consisting of negative obstacles: a proof of concept, *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 1043–1048.
- Ding, Y., & Thomas, U., (2020), Collision avoidance with proximity servoing for redundant serial robot manipulators, *IEEE International Conference on Robotics and Automation (ICRA)*, 10249–10255.

-
- Ding, Y., Wilhelm, F., Faulhammer, L., & Thomas, U., (2019), With proximity servoing towards safe human-robot-interaction, *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4907–4912.
- Escande, A., Mansard, N., & Wieber, P.-B., (2014), Hierarchical quadratic programming: fast online humanoid-robot motion generation, *The International Journal of Robotics Research*, *33*(7), 1006–1028.
- Espiau, B., (1988), An overview of local environment sensing in robotics applications, In P. Dario (Ed.), *Sensors and sensory systems for advanced robots* (pp. 125–151), Springer Berlin Heidelberg.
- Espiau, B., Chaumette, F., & Rives, P., (1992), A new approach to visual servoing in robotics, *IEEE Transactions on Robotics and Automation*, *8*(3), 313–326.
- Espiau, B., (1990), Sensory-based control robustness issues and modelling techniques application to proximity sensing, In G. E. Taylor (Ed.), *Kinematic and dynamic issues in sensor based control* (pp. 3–44).
- Espiau, B., (1993), Effect of camera calibration errors on visual servoing in robotics, *Int. Symp. on Experimental Robotics (ISER)*.
- Faverjon, B., & Tournassoud, P., (1987), A local based approach for path planning of manipulators with a high number of degrees of freedom, *IEEE International Conference on Robotics and Automation*, *4*, 1152–1159.
- Featherstone, R., (2008), Contact and impact. In *Rigid body dynamics algorithms* (pp. 213–239), Springer US.
- Feddema, J., & Novak, J., (1994), Whole arm obstacle avoidance for teleoperated robots, *IEEE International Conference on Robotics and Automation*, 3303–3309 vol.4.
- Fischler, M. A., & Bolles, R. C., (1987), Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, In M. A. Fischler & O. Firschein (Eds.), *Readings in computer vision* (pp. 726–740), Morgan Kaufmann.
- Flacco, F., Kröger, T., De Luca, A., & Khatib, O., (2012), A depth space approach to human-robot collision avoidance, *IEEE International Conference on Robotics and Automation*, 338–345.
- France, L., Girault, A., Gascuel, J.-D., & Espiau, B., (1999), Sensor modeling for a walking robot simulation, In N. Magnenat-Thalmann & D. Thalmann (Eds.), *Computer animation and simulation '99* (pp. 189–198), Springer Vienna.

-
- Gaz, C., Cognetti, M., Oliva, A., Robuffo Giordano, P., & De Luca, A., (2019), Dynamic identification of the franka emika panda robot with retrieval of feasible parameters using penalty-based optimization, *IEEE Robotics and Automation Letters*, 4(4), 4147–4154.
- Golub, G. H., & Van Loan, C. F., (2013), Special linear systems. *In Matrix computations* (Vol. 4th edition), Johns Hopkins University Press, Baltimore.
- Gros, S., & Diehl, M., (2022), Numerical optimal control (draft), <https://www.syscop.de/files/2020ss/NOC/book-NOCSE.pdf>
- Hartley, R. I., & Zisserman, A., (2004), Multiple view geometry in computer vision, Cambridge University Press, ISBN: 0521540518.
- Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., & Stuetzle, W., (1992), Surface reconstruction from unorganized points, *19th annual conference on computer graphics and interactive techniques*, 71–78.
- Inkila, K., (2005), Homogeneous least squares problem, *The Photogrammetric Journal of Finland*, 19(2), 34–42.
- Kanoun, O., Lamiraux, F., & Wieber, P.-B., (2011), Kinematic control of redundant manipulators: generalizing the task-priority framework to inequality task, *IEEE Transactions on Robotics*, 27(4), 785–792.
- Kermorgant, O., & Chaumette, F., (2011), Multi-sensor data fusion in sensor-based control: application to multi-camera visual servoing, *IEEE International Conference on Robotics and Automation*, 4518–4523.
- Kermorgant, O., & Chaumette, F., (2014), Dealing with constraints in sensor-based robot control, *IEEE Transactions on Robotics*, 30(1), 244–257.
- Khalil, H. K., (2015), *Nonlinear control* (Vol. 406), Pearson, New York.
- Khatib, O., (1985), Real-time obstacle avoidance for manipulators and mobile robots, *IEEE International Conference on Robotics and Automation*, 2, 500–505.
- Liégeois, A., (1977), Automatic supervisory control of the configuration and behavior of multibody mechanisms, *IEEE Transactions on Systems, Man, and Cybernetics*, 7(12), 868–871.
- Lumelsky, V. J., Shur, M. S., & Wagner, S., (2001), Sensitive skin, *IEEE Sensors Journal*, 1(1), 41–51.
- Lumelsky, V., & Cheung, E., (1993), Real-time collision avoidance in teleoperated whole-sensitive robot arm manipulators, *IEEE Transactions on Systems, Man, and Cybernetics*, 23(1), 194–203.

-
- Maciejewski, A. A., & Klein, C. A., (1985), Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments, *The International Journal of Robotics Research*, 4(3), 109–117.
- Malis, E., Chaumette, F., & Boudet, S., (1999), 2 1/2 d visual servoing, *IEEE Transactions on Robotics and Automation*, 15(2), 238–250.
- Malis, E., & Chaumette, F., (2002), Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods, *IEEE Trans. on Robotics and Automation*, 18(2), 176–186.
- Mansard, N., & Chaumette, F., (2005), Visual servoing sequencing able to avoid obstacles, *IEEE International Conference on Robotics and Automation*, 3143–3148.
- Mansard, N., & Chaumette, F., (2007), Task sequencing for high-level sensor-based control, *IEEE Transactions on Robotics*, 23, 60–72.
- Marchand, E., & Hager, G., (1998), Dynamic sensor planning in visual servoing, *IEEE International Conference on Robotics and Automation*, 3, 1988–1993.
- Marchand, E., Spindler, F., & Chaumette, F., (2005), ViSP for visual servoing: a generic software platform with a wide class of robot control skills, *IEEE Robotics and Automation Magazine*, 12(4), 40–52.
- M’Colo, K.-E., Luong, B., Crosnier, A., Néel, C., & Fraise, P., (2019), Obstacle avoidance using a capacitive skin for safe human-robot interaction, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6742–6747.
- Michel, H., & Rives, P., (1993), *Singularities in the determination of the situation of a robot effector from the perspective view of 3 points* (Research Report No. RR-1850), INRIA.
- Mitra, N. J., & Nguyen, A., (2003), Estimating surface normals in noisy point cloud data, *Nineteenth Annual Symposium on Computational Geometry*, 322–328.
- Mittendorf, P., & Cheng, G., (2012), 3d surface reconstruction for robotic body parts with artificial skins, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4505–4510.
- Nakamura, Y., Hanafusa, H., & Yoshikawa, T., (1987), Task-priority based redundancy control of robot manipulators, *The International Journal of Robotics Research*, 6(2), 3–15.
- Navarro, S. E., Koch, S., & Hein, B., (2016), 3d contour following for a cylindrical end-effector using capacitive proximity sensors, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 82–89.

-
- Navarro, S. E., Mühlbacher-Karrer, S., Alagi, H., Zangl, H., Koyama, K., Hein, B., Duriez, C., & Smith, J. R., (2022), Proximity perception in human-centered robotics: a survey on sensing systems and applications, *IEEE Transactions on Robotics*, 38(3), 1599–1620.
- Nenchev, D. N., (1989), Redundancy resolution through local optimization: a review, *J. Field Robotics*, 6, 769–798, <https://api.semanticscholar.org/CorpusID:33686541>
- Nocedal, J., & Wright, S. J., (2006), *Numerical optimization* (2e), Springer.
- Novak, J., & Feddema, I., (1992), A capacitance-based proximity sensor for whole arm obstacle avoidance, *IEEE International Conference on Robotics and Automation*, 1307–1314 vol.2.
- Nunes, U., Faia, P., & de Almeida, A., (1994), Sensor-based 3-d autonomous contour-following control, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, 1, 172–179 vol.1.
- Ogilvie, A., Allport, J., Hannah, M., & Lymer, J., (2008), Autonomous satellite servicing using the orbital express demonstration manipulator system, *9th International Symposium on Artificial Intelligence, Robotics and Automation in Space, iSAIRAS*.
- Oliva, A. A., Spindler, F., Giordano, P. R., & Chaumette, F., (2022), Frankasim: a dynamic simulator for the franka emika robot with visual-servoing enabled capabilities, *17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 868–874.
- Pages, J., Collewet, C., Chaumette, F., & Salvi, J., (2004), Plane-to-plane positioning from image-based visual servoing and structured light, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1, 1004–1009 vol.1.
- Pages, J., Collewet, C., Chaumette, F., & Salvi, J., (2006), Optimizing plane-to-plane positioning tasks by image-based visual servoing and structured light, *IEEE Transactions on Robotics*, 22(5), 1000–1010.
- Rao, C. R., (1972), Generalized inverse of a matrix and its applications, In L. M. L. Cam, J. Neyman, & E. L. Scott (Eds.), *Volume 1 theory of statistics* (pp. 601–620), University of California Press.
- Rives, P., & Borrelly, J.-J., (1997), Underwater pipe inspection task using visual servoing techniques, *IEEE/RSJ Int. Conf. on Intelligent Robot and Systems (IROS)*, 1, 63–68 vol.1.

-
- Roa Garzon, M. A., Nottensteiner, K., Wedler, A., & Grunwald, G., (2017), Robotic technologies for in-space assembly operations, *14th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*.
- Rosen, J. B., (1960), The gradient projection method for nonlinear programming. part i. linear constraints, *Journal of the Society for Industrial and Applied Mathematics*, 8(1), 181–217.
- Samson, C., & Espiau, B., (1990), Application of the task-function approach to sensor-based control of robot manipulators, *11th IFAC World Congress on Automatic Control, Tallinn*, 23(8, Part 5), 269–274.
- Samson, C., Espiau, B., & Borgne, M. L., (1991), *Robot control: the task function approach*, Oxford University Press, Inc.
- Sastry, S., (1999), Basics of differential geometry. In *Nonlinear systems: analysis, stability, and control* (pp. 349–383), Springer New York.
- Sharma, R., & Hutchinson, S., (1997), Motion perceptibility and its application to active vision-based servo control, *IEEE Transactions on Robotics and Automation*, 13(4), 607–617.
- Slotine, S. B., & Siciliano, B., (1991), A general framework for managing multiple tasks in highly redundant robotic systems, *5th International Conference on Advanced Robotics*, 2, 1211–1216.
- Strang, G., (2005), *Linear algebra and its applications*, 4th edition, Brooks Cole, New York.
- Tan, X., & Dimarogonas, D. V., (2024), On the undesired equilibria induced by control barrier function based quadratic programs, *Automatica*, 159, 111359.
- Thomas, J., & Chaumette, F., (2023a), Stability analysis of plane-to-plane positioning by proximity-based control, *IEEE Robotics and Automation Letters*, 8(8), 4473–4480.
- Thomas, J., & Chaumette, F., (2023b), Use of Screw Theory in Proximity-based Control, *ICRA workshop - "Geometric Representations: The Roles of Modern Screw Theory, Lie algebra, and Geometric Algebra in Robotics"*, 1–5.
- Thomas, J., Pasteau, F., & Chaumette, F., (2022), Plane-to-plane positioning by proximity-based control, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 12795–12802.
- Thomas, J., Loianno, G., Sreenath, K., & Kumar, V., (2014), Toward image based visual servoing for aerial grasping and perching, *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2113–2118.

-
- Vaillant, J., Kheddar, A., Audren, H., Keith, F., Brossette, S., Escande, A., Bouyarmane, K., Kaneko, K., Morisawa, M., Gergondet, P., Yoshida, E., Kajita, S., & Kanehiro, F., (2016), Multi-contact vertical ladder climbing with an HRP-2 humanoid, *Autonomous Robots*, 40(3), 561–580.
- Vergara Perico, C., Borghesan, G., Aertbeliën, E., & Schutter, J., (2019), Incorporating artificial skin signals in the constraint-based reactive control of human–robot collaborative manipulation tasks, *Industrial Robot: the international journal of robotics research and application*, 46.
- Wegerif, D., Rosinski, D., & Parton, W., (1993), Whole arm proximity control system for articulated robots working near space vehicles and flight hardware, *Space Programs and Technologies Conference and Exhibit*, 4158.
- Wegerif, D. G., & Rosinski, D. J., (1992), Sensor-based whole-arm obstacle avoidance for kinematically redundant robots, In P. S. Schenker (Ed.), *Sensor fusion v* (pp. 417–426, Vol. 1828), SPIE.
- Wilkinson, J. H., (1988), *The algebraic eigenvalue problem*, Oxford University Press, Inc.
- Xiao, W., Cassandras, C. G., & Belta, C., (2023), Control barrier functions. In *Safe autonomy with control barrier functions: theory and applications* (pp. 7–18), Springer International Publishing.

Titre : Commande référencée capteurs pour l'assemblage en milieu encombré

Mot clés : Commande référencée capteurs, asservissement visuel, capteurs proximétriques, assemblage, prévention des collisions

Résumé : Dans cette thèse, un système multi-sensoriel composé de capteurs proximétriques, appelé réseau proximétriques, est proposé. En l'attachant à l'effecteur, il permet à un robot d'effectuer des tâches de positionnement plan sur plan et de positionnement par rapport à un cylindre. L'analyse de la stabilité de la tâche de positionnement plan sur plan est effectuée en obtenant la forme explicite pour la pseudo-inverse de la matrice d'interaction. Les informations de proximité et de vision sont ensuite combinées pour traiter le positionnement dans un espace encombré pour une tâche d'assemblage à l'aide de capteurs sans contact dans le cadre de la commande référencée multi-capteurs. Le réseau proximé-

trique est enroulé autour du bras du manipulateur pour permettre d'éviter les collisions alors que l'asservissement visuel à partir de 4 points assure le positionnement. Divers résultats expérimentaux et de simulation sont fournis pour valider la théorie. Des formes explicites de la base duale sont obtenues pour des tâches incluant le suivi plan sur plan et le positionnement par rapport à un cylindre avec un minimum de capteurs et un asservissement visuel à partir de 3 points. La base duale est associée à l'espace linéaire des torseurs d'interaction qui forment les composantes de la matrice d'interaction. Cela conduit à des équations explicites capteurs.

Title: Assembly Task in Congested Area using Sensor-based Control

Keywords: Sensor-based Control, Visual Servoing, Proximity-based Control, Assembly, Collision Avoidance

Abstract: In this thesis, a multi-sensory system consisting of proximity sensors termed proximity array is proposed. While attaching it to the end-effector it enables a robot to perform plane-to-plane positioning task and positioning wrt. a cylinder. Stability analysis of plane-to-plane positioning task is considered by obtaining an explicit form for the pseudo-inverse of the interaction matrix. Proximity and vision information are then combined to address positioning in congested space for assembly task using non-contact sensors. The proximity arrays are wrapped around the

arm of the manipulator to enable collision-avoidance and 4-point Visual Servoing ensures positioning. Various experimental and simulation results are provided to validate the theory. Explicit forms of dual basis are obtained for tasks including plane-to-plane following and positioning wrt. a cylinder with minimal sensors along with 3-point Visual Servoing. Dual basis is associated to the linear space of interaction screws that form the components of interaction matrix. This leads to closed-form equations in Sensor-based Control.