

DISSERTAÇÃO DE MESTRADO Nº 975

**WHOLE-BODY KINEMATIC CONTROL OF NONHOLONOMIC MOBILE
MANIPULATORS USING LINEAR PROGRAMMING**

Juan José Quiroz Omaña

DATA DA DEFESA: 03/03/2017

Universidade Federal de Minas Gerais

Escola de Engenharia

Programa de Pós-Graduação em Engenharia Elétrica

**WHOLE-BODY KINEMATIC CONTROL OF NONHOLONOMIC
MOBILE MANIPULATORS USING LINEAR PROGRAMMING**

Juan José Quiroz Omaña

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do Título de Mestre em Engenharia Elétrica.

Orientador: Prof. Bruno Vilhena Adorno

Belo Horizonte - MG

Março de 2017

Q8w	<p>Quiroz Omanã, Juan José. Whole-body kinematic control of nonholonomic mobile manipulators using linear programming [manuscrito] / Juan José Quiroz Omanã. - 2017. x, 72 f., enc.: il.</p> <p>Orientador: Bruno Vilhena Adorno.</p> <p>Dissertação (mestrado) - Universidade Federal de Minas Gerais, Escola de Engenharia.</p> <p>Anexos e apêndices: f. 70-72.</p> <p>Bibliografia: f. 63-69.</p> <p>1. Engenharia elétrica - Teses. 2. Cinemática - Teses. 3. Robôs - Sistemas de controle - Teses. 4. Programação linear - Teses. 5. Sistemas dinâmicos não holonômicos - Teses. I. Adorno, Bruno Vilhena. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.</p> <p style="text-align: right;">CDU: 621.3(043)</p>
-----	--

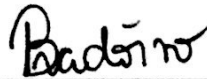
"Whole-body Kinematic Control of Nonholonomic Mobile Manipulators Using Linear Programming"

Juan José Quiroz Omaña

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica.

Aprovada em 03 de março de 2017.

Por:



Prof. Dr. Bruno Vilhena Adorno
DEE (UFMG) - Orientador



Prof. Dr. Guilherme Augusto Silva Pereira
DEE (UFMG)



Prof. Dr. Vinicius Mariano Gonçalves
DEE (UFMG)

macro@ufmg

Universidade Federal de Minas Gerais

Programa de Pós-Graduação em Engenharia Elétrica

MACRO Research Group - Mechatronics, Control and Robotics

**WHOLE-BODY KINEMATIC CONTROL OF
NONHOLONOMIC MOBILE MANIPULATORS
USING LINEAR PROGRAMMING**

Juan José Quiroz Omaña

Belo Horizonte, Brazil

2017

Juan José Quiroz Omaña

**WHOLE-BODY KINEMATIC CONTROL OF
NONHOLONOMIC MOBILE MANIPULATORS
USING LINEAR PROGRAMMING**

Dissertation submitted to the Electrical Engineering Graduate Program at the Federal University of Minas Gerais, in partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

Advisor: Bruno Vilhena Adorno

Belo Horizonte, Brazil

2017

*To Magali, José, Paula and Camila.
Thanks for always believing in me.*

Acknowledgements

I once heard the saying “God does not give us heavier loads than we can bear” and now I understand what it really means. Usually, behind every personal achievement, there are always dozens of people who helped in some way to get it. This part, which I consider one of the most important of the work, is dedicated to all those people who made possible not only the conclusion of this master thesis, but also made my life easier and more bearable at this incredible stage and in this wonderful country, Brazil.

First of all, I would like to say thanks to Professor Bruno Adorno for accepting me as his student. From the first moment, he was patient and understanding with my *Portuñol* and provided me with excellent infrastructure conditions to study and work. I always wondered how he could understand me when I spoke to him quickly and without fluency, but then I learned that he speaks at least three Romance languages, Spanish being one of them. His excellent and precise technical observations guided me and helped me to consolidate my academic training in the best way. Likewise, his notes, corrections and English explanations transformed my archaic and primitive text into a decent and understandable one. One of the best qualities of Professor Bruno is to encourage a student who is going through an existential crisis (which is very common in postgraduate studies), using his unbeatable argumentative ability and humorous sentences, making the student, ironically now with twice tasks to be done, feels better. Finally, I would like to say, that I lost the friendly debate that I had for two years with him, regarding Windows and GNU/Linux. Now I can not imagine my personal computer without a free distribution.

I would like to thank the faculty of the engineering school of UFMG for their teachings and explanations on the various topics that helped my academic education. Especially the professors Bruno Adorno, Guilherme Raffo, Leonardo Tôrres and Luis Aguirre, who, with their pedagogy and engineering jokes, kept the class interactive, making them not just masterful, but also agreeable, arousing more and more interest in research in the students. Thanks also to the Brazilian agencies CAPES, CNPQ and FAPEMIG for their financial support during these two years.

Thanks to my friends and Laboratory colleagues Daniel, Marcelo, Brenner, Rafael, Ana, Ernesto, Gabriella, Stella, Antonio, Diana, Freddy, Frederico, Mariana, PetruX, Wendy, Marcus, who accompanied me at this stage and confirmed what everyone suspects about

Brazilians, that they are extremely friendly, jokers and great people. I could not forget those Colombians who were always with me from the first moment and those who were arriving, with “arepas”, “vallenatos” and “aguardiente”, living the football matches of the Colombian team from the distance and remembering that Colombia is carried in the heart. Thank you Diego, Vicko, Brayan, Camilo, Arturo, Edna, Marcela.

My heart will always have a special place for Liliana, that sweet, magical and charismatic little girl with a giant heart, who, despite living in the same Colombian city, like me for many years, San José de Cúcuta, I only had the opportunity to meet in Belo Horizonte city. Thank you not only for making my days more beautiful, or for forcing me to know the city, or getting up me early to make her the cappuccino, but also for supporting me when I needed it most. Thank you for making me feel at home, thank you for making me feel a bit less the nostalgia of being away.

I could not fail to mention my deep thanks to all those people in Colombia who supported me, cheered me and prayed for me. Thanks to my aunts Luz Mila, Miriam, Feliza, Vidalia, Alix and Omaira Jaimes for their unconditional support, always forming a great pillar in the Quiroz family. In the same way, thanks to the “Omaña”, those aunts who keep several families together and see me as another son, thanks Aunt Rosa, Empera and Ofelia. Thanks to my uncles, Alfredo, Juan Carlos and especially Uncle Luis and Aunt Jacqueline, who always supported me in everything, thank you for always showing me another point of view, another way of seeing life and of course, thank you for making my “Chevetico” work always. Thanks to all those Omañas soccer players who made me feel like a star player, for speaking Portuguese despite being a disaster in football, thanks Diego, Manuel, Alfredito, Nicolás, Simón, Juan Carlos and Brayan.

Last but most importantly, I would like to thank the unconditional support of my parents, José Alberto and Magali, who have always taught me and formed me to take on the challenges and overcome them with humility. All this achievement, all the effort, is dedicated to them, who are, together with my sisters Paula and Camila, the most important thing for me. Thank you for being by my side and making my life more beautiful and happy.

Agradecimientos

Alguna vez escuché la frase “Dios no nos da cargas más pesadas de las que podemos soportar” y ahora entiendo lo que realmente significa. Por lo general, detrás de cada logro personal, siempre hay decenas de personas que ayudaron de alguna manera a conseguirlo. Esta parte, la cual considero una de las más importantes del trabajo, es dedicada para todas aquellas personas que hicieron posible no sólo la conclusión de esta disertación de maestría, sino que hicieron mi vida más fácil y llevadera en esta increíble etapa y en este maravilloso país, Brasil.

Antes de todo, me gustaría agradecer al profesor Bruno Adorno por aceptarme como su estudiante. Desde el primer momento se mostró paciente y comprensivo con mi *portuñol* y me proporcionó excelentes condiciones de estudio, infraestructura y trabajo. Siempre me pregunté cómo lograba entenderme cuando le hablaba rápido y sin fluidez, aunque, después me enteré que él hablaba por lo menos tres lenguas romances, el español entre ellas. Sus excelentes y precisas observaciones técnicas me orientaron y me ayudaron a consolidar mi formación de la mejor manera. Así mismo, sus apuntes, correcciones y explicaciones del inglés transformaron mi arcaico y primitivo texto en uno decente e inteligible. Una de las mejores cualidades del profesor Bruno es animar a un estudiante que está pasando por alguna crisis existencial (lo cual, es muy común en el postgrado), usando su imbatible capacidad de argumentación y frases jocosas, haciendo que irónicamente el estudiante, ahora con el doble de tareas por hacer, se sienta mejor. Finalmente, me gustaría decir, que perdí el amistoso debate que durante dos años mantuve con él con respecto a Windows y GNU/Linux. Ahora no puedo imaginar mi computador personal sin una distribución libre.

Me gustaría agradecerle al cuerpo docente de la escuela de ingeniería de la UFMG por sus enseñanzas y explicaciones en los diversos temas que ayudaron a mi formación académica. Especialmente a los profesores Bruno Adorno, Guilherme Raffo, Leonardo Tôrres y Luis Aguirre, los cuales, con su pedagogía y chistes de ingeniería, mantenían interactiva la clase, haciéndolas además de magistrales, muy agradables, despertando cada vez más el interés por la investigación en los estudiantes. Gracias también a las agencias brasileñas CAPES, CNPQ y FAPEMIG por su apoyo financiero durante estos dos años.

Gracias a mis amigos y colegas del Laboratorio Daniel, Marcelo, Brenner, Rafael, Ana, Ernesto, Gabriella, Stella, Antonio, Diana, Freddy, Frederico, Mariana, Petru, Wendy,

Marcus, quienes me acompañaron en esta etapa de la maestría y confirmaron lo que todos sospechan de los brasileños, que son extremadamente amigables, bromistas y grandes personas. No podría olvidar a esos colombianos que siempre estuvieron conmigo desde el primer momento y aquellos que fueron llegando, con arepas, vallenatos y aguardiente, viviendo los partidos de la selección desde la distancia y recordándome, que Colombia se lleva en el corazón. Gracias Diego, Vicko, Brayan, Camilo, Arturo, Edna y Marcela.

Mi corazón siempre tendrá un lugar muy especial para Liliana, aquella dulce, mágica y carismática pequeña niña de corazón gigante, que a pesar de vivir en la misma ciudad colombiana que yo por muchos años, San José de Cúcuta, sólo tuve la oportunidad de conocer en Belo Horizonte. Gracias no sólo por hacer mis días más bonitos, o por forzarme a conocer la ciudad, o a levantarme temprano para hacerle el capuccino, sino también por apoyarme cuando más lo necesitaba. Gracias por hacerme sentir en casa, gracias por hacerme sentir un poco menos la nostalgia de estar lejos.

No podría dejar de mencionar mis profundos agradecimientos a todas esas personas en Colombia que me apoyaron, hicieron fuerza y oraron por mí. Gracias a mis tías Luz Mila, Miriam, Feliza, Vidalia, Alix y Omaira Jaimes por su incondicional apoyo, siempre formando un gran pilar en la familia Quiroz. De igual manera, gracias a las “Omaña”, aquellas tías que mantienen unida varias familias y me ven como un hijo más, gracias tía Rosa, Empera y Ofelia. Gracias a mis tíos, Alfredo, Juan Carlos y especialmente al tío Luis y la tía Jacqueline, quienes me apoyaron en todo desde siempre, gracias por siempre mostrarme otro punto de vista, otra forma de ver la vida y claro, gracias por hacer que mi “chevetico” siempre funcionara. Gracias a todos esos Omañas futbolistas que me hicieron sentir como un jugador estrella por hablar portugués a pesar de ser un desastre en el fútbol, gracias Diego, Manuel, Alfredito, Nicolás, Simón, Juan Carlos y Brayan.

De último, pero más importante, me gustaría agradecer el incondicional apoyo de mis padres, José Alberto y Magali, quienes desde siempre me enseñaron y formaron para asumir los retos y superarlos con humildad. Todo este logro, todo el esfuerzo, va dedicado para ellos, quienes son, junto a mis hermanas Paula y Camila, lo más importante para mí. Gracias por estar a mi lado y hacer mi vida más bonita y feliz.

Resumo

Este trabalho apresenta o estudo e implementação do controle cinemático de robôs baseado em programação linear proposto recentemente por Gonçalves et al. (2016). Este método tem como vantagem a possibilidade de se incluir restrições de igualdade e desigualdade nas entradas de controle do sistema, além de ser computacionalmente eficiente e ter garantia formal de estabilidade. O método foi aplicado a um robô manipulador móvel de base não holonômica e algumas melhorias foram propostas à formulação original, como a proposta de uma nova função positiva definida dependente da variação do erro para evitar movimentos nas juntas quando o efetuador do robô estabiliza em um ponto diferente do ponto desejado. Além disso, as restrições de não holonomia da base móvel são impostas como restrições de igualdade no programa linear e, portanto, não há necessidade de se utilizar uma estrutura de controle em cascata para realizar o controle de corpo completo. Definiu-se ainda restrições de desigualdade para evitar tanto violações dos limites das juntas quanto colisões com obstáculos. Para garantir um bom desempenho do controlador, a sua implementação foi feita no *Robot Operating System* (ROS) utilizando-se C++. Além disso, também foi integrado um sistema de visão baseado em sensores RGB-D para reconhecimento de marcadores, cujo objetivo foi de melhorar a odometria da base móvel e detectar obstáculos no espaço de trabalho. Para avaliar o desempenho do controlador, foi apresentada uma comparação com uma estrutura de controle em cascata na qual uma malha interna é usada para lidar com as restrições de não holonomia da base móvel, mediante uma linearização entrada-saída, e uma malha mais externa é usada para lidar com os movimentos do corpo completo usando a pseudo-inversa da matriz Jacobiana de corpo completo. Resultados de simulação e experimentais mostram que o controle baseado na programação linear possui baixo custo computacional, o robô consegue executar tarefas de controle de pose e posição do efetuador sem que haja colisão com obstáculos no plano e sem que haja violação dos limites das juntas do robô manipulador. Porém, o método baseado em programação linear, utilizando-se o algoritmo Simplex, gera sinais de controle mais abruptos do que aqueles gerados pelo controle em cascata baseado na pseudo-inversa da matriz Jacobiana do robô.

Palavras-chave: Programação linear, controle cinemático, otimização, não holonômico.

Abstract

This work presents the study and implementation of the robot kinematic control strategy based on linear programming recently proposed by Gonçalves et al. (2016). In addition to being computationally efficient, this approach enables the inclusion of inequality and equality constraints in the system control inputs and has formal guarantee of stability. This method was applied to a nonholonomic mobile manipulator and some improvements were proposed to the original formulation, such as the addition of a new positive definite function of the error variation to avoid joint movements when the robot end effector stabilizes at a point different from the desired one. In addition, nonholonomic constraints of the mobile base are imposed as equality constraints in the linear program and therefore there is no need to use a cascade control structure to perform whole body control. Inequality constraints were also defined to avoid both violation of joint limits and collisions with obstacles. To guarantee a good performance, the controller was implemented on the Robot Operating System (ROS) using C++. In addition, a computer vision system based on RGB-D sensors for marker recognition was also integrated into the experimental testbed with the goal of improving robot localization and detecting obstacles in the workspace. In order to evaluate the controller performance, a comparison was made with a cascade control structure in which an inner loop is used to deal with the nonholonomic constraints of the mobile base by using an input-output linearization and an outer loop is used to tackle the whole-body motion by using the pseudoinverse of the whole-body Jacobian matrix. Simulation and experimental results show that the controller based on linear programming has low computational cost, and the robot is able to control its end effector without colliding with obstacles in the plane and without violating its joints limits. However, when using the Simplex algorithm the method based on linear programming generates more abrupt control signals than those generated by the cascade controller based on the pseudoinverse of the robot Jacobian matrix.

Keywords: Linear programming, kinematic control, optimization, nonholonomic.

Contents

List of Figures	xii
Acronyms	xv
Notation	xvi
1 Introduction	1
1.1 Objectives and Contributions	3
1.2 Structure of the Text	4
2 State of the Art	5
2.1 Kinematic Control of Redundant Robots: Classic Techniques	5
2.1.1 Task-Space Augmentation	6
2.1.2 Pseudoinverse of the Jacobian Matrix	7
2.1.3 Task-Priority Framework	7
2.2 Kinematic Control as a Linear Programming	8
2.3 Kinematic Control of Nonholonomic Mobile Manipulators	10
2.4 Chapter Conclusions	12
3 Mathematical Background	15
3.1 Fundamentals of Dual Quaternion Algebra	15
3.1.1 Quaternions	16
3.1.2 Dual Quaternions	17
3.2 Rigid Motions using Dual Quaternions	18
3.3 Fundamentals in Linear Programming	21
3.3.1 Basic Operations	22
3.3.2 Example	22
3.4 Conclusions	24
4 Forward and Differential Kinematic Model	25
4.1 Kinematic Model of the Holonomic Mobile Base	25

4.2	Kinematic Model of the Manipulator Arm	27
4.3	Whole-body Kinematic Model	27
4.4	Position Jacobian Matrix	28
4.5	Chapter Conclusions	29
5	Whole-Body Kinematic Control	30
5.1	Kinematic Control using Linear Programming	30
5.2	Improvements to the whole-body control based on linear programming . .	32
5.3	Nonholonomic constraint of the mobile base	34
5.4	Avoidance of joint limits	34
5.5	Collision avoidance of the mobile base	35
5.6	Final formulation considering all constraints	38
5.7	Parsimonious behavior	39
5.8	Stability Consideration	39
5.9	Chapter Conclusions	39
6	Experiments and Results	41
6.1	Implementation Details	41
6.1.1	Vision System	42
6.2	Experiments	45
6.2.1	Control of the end-effector pose	45
6.2.2	Control of the end-effector position	51
6.2.3	Obstacle avoidance	53
6.3	Computational Efficiency Test	55
6.3.1	Test 1: Holonomic mobile manipulator case.	55
6.3.2	Test 2: Nonholonomic mobile manipulator case	56
6.4	Chapter Conclusions	58
7	Conclusion and Future Works	59
7.1	Conclusions	59
7.2	Future Works	61
	Bibliography	63
	Appendix A Dimensions of the Mobile Manipulator	70
	Appendix B Whole-body kinematic control based on the Pseudoinverse	72

List of Figures

1.1	Recently emerged mobile manipulators: (a) Kuka youBot, (b) PR2 Robot , (c) Justin robot.	2
3.1	Representation of a rotation of an angle ϕ around the unit-norm axis \mathbf{n} . . .	19
3.2	Sequence of rigid transformations using dual quaternions.	19
3.3	Given a reference frame \mathcal{F}_0 , the rigid motion from the reference frame to the frame \mathcal{F}_1 is represented by the unit dual quaternion $\underline{\mathbf{x}}$. The unit dual quaternion $\underline{\mathbf{x}}^{\{\frac{1}{2}\}}$ represents the half of the translation and the half of the rotation angle of $\underline{\mathbf{x}}$. This resultant frame is shown inside the red dashed square.	21
4.1	Pose of the nonholonomic mobile base with respect to \mathcal{F}_0	26
4.2	Nonholonomic mobile manipulator.	28
5.1	Two definite positive functions. The <i>top</i> figure shows a function that depends on the error. The <i>bottom</i> figure shows a function that depends on the error time-derivative.	33
5.2	The margin considered for the joint limits $\mathbf{q}^- \leq \mathbf{q} \leq \mathbf{q}^+$ is determined by the security factor β_l	35
5.3	Behavior of the function $f(\mathbf{q}_{xy}, \mathbf{s})$. In the <i>left</i> , the robot is outside the collision region and $f(\mathbf{q}_{xy}, \mathbf{s}) < 0$; in this case there is no risk of collision between the robot and the obstacle in the plane. In the <i>center</i> , the robot is on the border of the collision region and $f(\mathbf{q}_{xy}, \mathbf{s}) = 0$; also in this case, there is no also risk of collision. Finally, in the <i>right</i> , the robot is inside the collision region and $f(\mathbf{q}_{xy}, \mathbf{s}) > 0$; in this case, there is a high risk of collision.	37
5.4	Start and final configurations of the mobile base are denoted by \mathbf{q}_i and \mathbf{q}_d , respectively. The configuration \mathbf{q}_d is the projection of the end-effector position onto the xy -plane of the inertial frame \mathcal{F}_0 , which provides a rough approximation of where the center of the mobile base should be, as the manipulator is relatively small.	38

6.1	Frames of the vision system. The unit dual quaternions $\underline{\mathbf{x}}_0^c$, $\underline{\mathbf{x}}_1^c$, $\underline{\mathbf{x}}_2^c$ and $\underline{\mathbf{x}}_3^c$ represent the pose of the frames \mathcal{F}_0 , \mathcal{F}_1 , \mathcal{F}_2 and \mathcal{F}_3 respectively with respect to the fixed frame \mathcal{F}_c	43
6.2	Global reference frame of the vision system. All markers are referenced with respect to the global reference frame \mathcal{F}_0 . This and the other markers, can be placed anywhere within the the visual range of the camera.	44
6.3	Structure of the experimental environment.	45
6.4	Description of the experiment: The box must be placed inside the a trash bin. The relation between the frame \mathcal{F}_1 and the \mathcal{F}_2 with respect to the reference frame \mathcal{F}_0 are given by the unit dual quaternions $\underline{\mathbf{x}}_1^0$ and $\underline{\mathbf{x}}_2^0$, respectively. The desired pose $\underline{\mathbf{x}}_d^0$ is computed by using a known rigid transformation between the frames \mathcal{F}_2 and \mathcal{F}_d and is given by $\underline{\mathbf{x}}_d^0 = \underline{\mathbf{x}}_2^0 \underline{\mathbf{x}}_d^2$	46
6.5	Control of the end-effector pose : snapshots from the experiment. A Microsoft Kinect sensor located at the ceiling is used to recognize fiducial markers placed on the robot, the obstacles, and the desired goal.	47
6.6	Control of the end-effector pose : control inputs using the classic kinematic control (PINV) and the parsimonious control using linear programming with Simplex (LP).	48
6.7	Control of the end-effector pose : number of nonzero entries generated using the classic kinematic control (PINV) and the parsimonious control using linear programming with Simplex (LP).	49
6.8	Control of the end-effector pose : number of nonzero entries used in the experimental test using the classic kinematic control (PINV) and the parsimonious control using linear programming with Simplex (LP).	49
6.9	Control of the end-effector pose : time response of each coefficient of the dual quaternion error.	50
6.10	Control of the end-effector pose : coefficients of the robot configuration \mathbf{q} (see (4.9)) using the classic kinematic control (PINV) and the parsimonious control using linear programming with Simplex (LP). The joints limits are represented by <i>dashed green</i> lines.	50
6.11	Control of the end-effector position : control inputs using the classic kinematic control (LEFT) and the parsimonious control using linear programming with Simplex (RIGHT).	52
6.12	Control of the end-effector position : number of nonzero entries generated using the classic kinematic control (PINV) and the parsimonious control using linear programming with Simplex (LP).	52
6.13	Control of the end-effector position : number of nonzero entries used in the experimental test using the classic kinematic control (PINV) and the parsimonious control using linear programming with Simplex (LP).	52

6.14	Control of the end-effector position : End effector translation using the classic kinematic control (PINV) and the parsimonious control using linear programming with Simplex (LP). The reference is represented by the straight green line.	53
6.15	Description of the task: The robot must to grasp a box in \mathcal{F}_e and put it inside a trash bin located in \mathcal{F}_d . There is an obstacle in the plane located in \mathcal{F}_{obs} and the controller must to prevent a collision between the mobile platform and the obstacle. The pose of all frames are computed with respect to \mathcal{F}_0 by using the vision system.	54
6.16	Control of the end-effector pose while avoiding an obstacle: snapshots from the experiment.	54
6.17	Control of the end-effector pose while avoiding an obstacle: number of nonzero entries using linear programming with Simplex (LP).	55
6.18	Convergence time in a pose task that required 46 iterations for both controllers. In the <i>left</i> , it is presented the convergence time in each trial performed. In the <i>right</i> , it is presented the average convergence time with the errors bar, which denote the respective standard deviation.	56
6.19	Convergence time in a pose task that required 77 iterations for both controllers. In the <i>left</i> , it is presented the convergence time in each trial performed. In the <i>right</i> , it is presented the average convergence time with the errors bar, which denote the respective standard deviation.	57
6.20	Average time per iteration for both controllers.	57
A.1	AX18 arm manipulator.	70
A.2	Differential mobile base.	71
B.1	Whole-body kinematic control scheme.	73
B.2	Point offset used in the control of the nonholonomic mobile base.	73

Acronyms

DFKM	Differential Forward Kinematic Model
D-H	Denavit-Hartenberg
DOF	Degrees of Freedom
DQ	Dual Quaternion
FKM	Forward Kinematic Model
LP	Linear Programming
MACRO	Mechatronics, Control and Robotics
PINV	Pseudoinverse
QP	Quadratic Programming
ROS	Robot Operating System
UFMG	Universidade Federal de Minas Gerais
3D	Three-Dimensional
2D	Two-Dimensional

Notation

\mathbf{c}	Center coordinate of a circular obstacle.
\mathcal{F}	Coordinate system or frame.
\mathbb{H}	Set of quaternions.
\mathcal{H}	Set of dual quaternions.
$\mathbf{h}, \mathbf{x}, \mathbf{y}$	Quaternions.
$\underline{\mathbf{h}}, \underline{\mathbf{x}}, \underline{\mathbf{y}}$	Dual quaternions.
\mathbf{h}^*	Dual quaternion conjugate.
$\overset{+}{\mathbf{H}}_4, \bar{\mathbf{H}}_4$	Hamilton operators.
$\overset{+}{\mathbf{H}}, \bar{\mathbf{H}}$	Hamilton operators extende for dual quaternions.
$\hat{i}, \hat{j}, \hat{k}$	Quaternion units.
\mathbf{I}	Identity matrix.
\mathbf{J}	Jacobian matrix.
\mathbf{J}^T	Transpose of the Jacobian matrix.
\mathbf{J}^\dagger	Pseudoinverse of the Jacobian matrix.
\mathbf{J}_p	Position Jacobian matrix.
$\mathbf{J}_{\mathcal{P}}, \mathbf{J}_{\mathcal{D}}$	Jacobian matrices related with the primary and dual parts of $\underline{\mathbf{x}}$ that satisfies $\text{vec}_8 \dot{\underline{\mathbf{x}}} = \mathbf{J}\dot{\mathbf{q}}$.
k	Scalar gain used to scale the feasible region in the joint-limit constraints.
\mathbf{n}	Pure quaternion representing the rotation axis.

\mathbf{p}_{ab}^a	Pure quaternion representing the translation from the frame \mathcal{F}_a to \mathcal{F}_b with respect to \mathcal{F}_a .
\mathbf{q}	Joints vector.
$\dot{\mathbf{q}}_b$	Velocities of the mobile base.
\mathbf{q}_d	Projection of the end-effector position onto the xy-plane of the inertial frame \mathcal{F}_0 .
\mathbf{q}_i	Start configuration of the mobile base.
\mathbf{q}_{xy}	Mobile base position.
R	Radius of a circular obstacle.
\mathbf{r}_b^a	Unit quaternion representing the rotation from the \mathcal{F}_a to \mathcal{F}_b .
\mathbf{s}	Configuration of a obstacle with radius R and center \mathbf{c} .
T_{DOF}	Number of DOF needed to execute a particular taks.
U_{NZ}	Non-zero entries.
$\text{vec}_4, \text{vec}_8$	Vec operators representing a one-to-one mapping from \mathbb{H} to \mathbb{R}^4 and \mathcal{H} to \mathbb{R}^8 respectively.
$\ \underline{\mathbf{x}}\ $	Dual quaternion norm.
$\alpha(t)$	Time dependent slack variable .
β	Scalar gain of the definite positive function.
β_l	Scalar gain used to define a security magin of the joint limits.
ε	Dual unit.
η	Scalar gain of the control law.
$\dot{\boldsymbol{\theta}}$	Joint velocities vector of the manipulator arm.
$\mathbf{q}^-, \mathbf{q}^+$	Lower and upper limits of the joints manipulator.
$\dot{\mathbf{q}}^-, \dot{\mathbf{q}}^+$	Lower and upper velocity limits of the joints manipulator.
λ	Scalar gain used in the error dynamics of the constraint that prevents collision.
ϕ	Rotation angle.
$\mathbf{0}$	Zero matrix.

1

Introduction

Nature has been the main source of inspiration and creation of human beings (Snell-Rood, 2016). Millions of years of evolution have created extremely complex organisms adapted to the different environments and analogously most of today's robots have undergone an artificial evolution to adapt to new and challenging problems as well as human needs. Some of these robots are the humanoid robots and the mobile manipulators, which are equipped with many (DOF) and have emerged for applications such as human-robot interaction (De Santis et al., 2008), medical care (Taylor, 2006), education (Chin et al., 2014), military (Carlson and Murphy, 2005), and also to assist humans in daily activities (Cha et al., 2015).

In order to perform manipulation tasks or generate desired movements, motion control techniques are required. When the physical forces acting on the system is the main concern, the robot dynamic model must be computed and therefore the inertial parameters that describe the robot dynamics should be known, but such parameters are not always available. However, when the main concern is the position and orientation of the end effector and the dynamics of the internal loop (i.e., the inner velocity control loop) is sufficiently fast—which in practice happens when the robot is sufficiently rigid and operates under relatively low velocities and accelerations—the system dynamics can be regarded as a single integrator and thus the robot can be actuated in velocity (Siciliano and Slotine, 1991). This allows the use of kinematic control, an approach more appropriate under aforementioned considerations. Moreover, kinematic control is not affected by uncertainties in the inertial parameters (e.g., mass and moment of inertia) and requires simpler models,

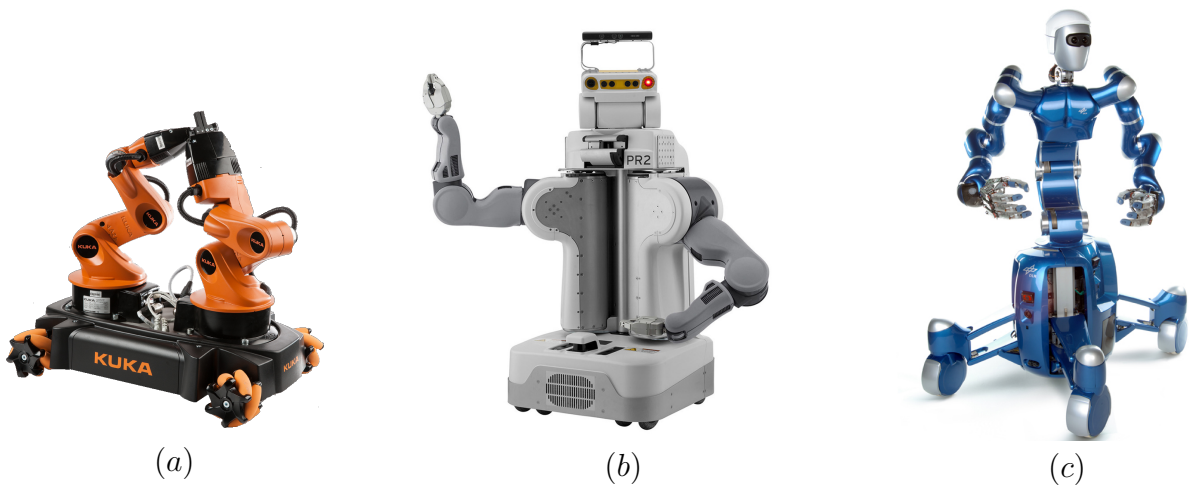


Figure 1.1: Recently emerged mobile manipulators: (a) Kuka youBot, (b) PR2 Robot , (c) Justin robot.

such as the forward kinematics model (FKM) and the differential forward kinematics model (DFKM). The former provides the relation between the joints configuration and the end-effector pose (i.e., position and orientation), whereas the latter provides the relation between the joints velocities and the end-effector generalized velocities by means of the Jacobian matrix (Spong et al., 2006).

Traditionally, the kinematic control problem has been approached by using the generalized pseudoinverse of the Jacobian matrix (a solution that corresponds to the least squares minimization problem). When a robot is redundant (i.e., it has more actuated (DOF) than the amount required for executing a particular task) infinite solutions exist. Therefore, it is desirable that the control system select the most appropriate control action to achieve the desired task while simultaneously avoiding singularities, external obstacles, or optimizing any other criterion of interest (Liégeois, 1977). It is common in redundant systems to impose additional restrictions or tasks in order to enhance the versatility of the robot, and usually a task-priority strategy is used to manage those multiple tasks based on the null space of the Jacobian Matrix. When several incompatible goals are involved, hierarchical inverse kinematics is often used (Siciliano and Slotine, 1991; Chiaverini, 1997; Marani, 2004; Escande et al., 2010; Flacco and De Luca, 2013; Escande et al., 2014). Often, the robot motion generation is expressed as an optimization problem (Laumond et al., 2015) and when incompatible objectives or constraints are involved, and there is no an analytical solution, numerical solvers can be used (Escande et al., 2010; Kanoun et al., 2011; Escande et al., 2014; Rauscher et al., 2016). This approach has been addressed mostly using quadratic programming, but when a large amount of constraints are imposed the computational cost increases considerably (Nakamura et al., 2005) . This disadvantage can be mitigated using linear programming, where recent works (Ho et al., 2005; Goncalves et al., 2016; Quiroz-Omana and Adorno, 2016) showed low computational cost and good

performance when this approach was used in redundant systems with additional imposed constraints.

This work focuses on the kinematic control approach to control a nonholonomic mobile manipulator based on the recently parsimonious control proposed by Goncalves et al. (2016), in which the inverse kinematic formulation is based on linear programming and the Lyapunov stability was formally proved. This approach has some important features such as low computational cost and the possibility of including equality and inequality constraints. The remainder of this chapter is organized as follows: Section 1.1 explains the objectives and Contributions of this dissertation; and Section 1.2 details the structure of this work.

1.1 Objectives and Contributions

The main objective of this work is to apply the kinematic control using linear programming proposed by Goncalves et al. (2016) in a nonholonomic mobile manipulator. Furthermore to exploit its features in order to enhance the performance with respect to the traditional approach, as well verify the control strategy in the real robot. The specific objectives are:

- Implement the whole-body kinematic control using linear programming in a nonholonomic mobile manipulator.
- Impose additional constraints in the linear programming formulation in order to prevent violations of joint limits and prevent collisions with obstacles in the plane.

The traditional kinematic control approach used for comparison in the nonholonomic mobile manipulator is based on a cascade scheme composed of an outer loop that takes into account all DOF by using the pseudoinverse of the robot Jacobian matrix and an inner loop that explicitly deals with the nonholonomic constraints of the mobile base by means of an input-output linearizing controller (Salazar-Sangucho and Adorno, 2014). Although that approach works well in practice the formal proof of the system stability is not trivial and was not shown in (Salazar-Sangucho and Adorno, 2014).

A performance comparison between the traditional kinematic control and the linear programming approach was presented in the 2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR) (Quiroz-Omana and Adorno, 2016). Both strategies were based on a cascade control scheme in order to deal with the nonholonomic constraint of the mobile base.

The contributions of this work can be summarized as follows:

- The nonholonomic constraint of the mobile base was taken into account as an equality constraint in the linear programming formulation and the cascade structure used in (Quiroz-Omana and Adorno, 2016) was removed. This avoids the necessity of

prove the stability using such structure. Because the Lyapunov stability in the linear programming formulation was formally proved by Goncalves et al. (2016), when the cascade structure is not used the Lyapunov stability is guaranteed.

- A new definite positive function that depends on the error variation is proposed in order to avoid joint movement when the robot end-effector stabilizes in a point different from the desired one. The original formulation of Goncalves et al. (2016), guarantees that the robot joints will stop only if the stabilization is asymptotic.
- Additional inequality constraints are imposed in the linear program to avoid violation of the manipulator's joints limits and to prevent collision between the mobile base and obstacles in the plane. The application of the kinematic control based on linear programming without cascade scheme to a nonholonomic system, such as the nonholonomic mobile manipulator used in this work, is novel.

1.2 Structure of the Text

This thesis is organized as follows:

Chapter 2 presents some of the most important works in nonholonomic and redundant robotic systems, including kinematic control strategies as optimization problems. Chapter 3 reviews the mathematical foundation required and recommended in order to understand the presented methods. In addition it is established the notation used in this work. Chapter 4 describes the modeling methodology based on dual quaternion algebra used in order to obtain the forward and differential kinematic model of the nonholonomic mobile manipulator. Chapter 5 describes the control strategy based on linear programming and some improvements to the original formulation. In addition, this chapter presents the additional imposed constraints to prevent violations of the joint limits and avoid collision with obstacles in the plane. Chapter 6 describes the experimental platform and the experiments performed, as well as the obtained results. Finally, Appendix A contain the D-H parameters of the manipulator arm and the dimensions of the mobile base.

2

State of the Art

This chapter reviews some recent works related to the research on nonholonomic mobile manipulators and is organized as follows: Section 2.1 presents a review of some kinematic control strategies for redundant robots based on the pseudoinverse of the robot Jacobian matrix. Section 2.2 reviews some kinematic control strategies based on linear programming. Section 2.3 presents some works related to the kinematic control of nonholonomic mobile manipulators.

2.1 Kinematic Control of Redundant Robots: Classic Techniques

It is well known that the general optimization problem for motion control is difficult to solve. In most cases, solving the general problem require very long computation time and there is a possibility of not even finding a solution. However, for a few classes of optimization problems there are effective algorithms that require low computational time, as the least-squares or linear programming, both particular cases of convex optimization that have been widely used in robotics as principle of motion in last decades (Laumond et al., 2015).

Finding the configuration of each actuator of the robot in order to achieve a particular position and orientation of the end-effector is called inverse kinematic problem. The nonlinear equations that govern the robot kinematics are very expensive to solve. Global

optimization techniques have been used to solve such equations but are impractical for on-line feedback control due to its high computational cost (Nakamura and Hanafusa, 1987; Laumond et al., 2015). This has been addressed by using a local optimization that deals with the differential kinematics, which uses the relation between the joints velocities and the end-effector velocities by means of the Jacobian matrix (Siciliano, 1990) as follows

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}, \quad (2.1)$$

where $\dot{\mathbf{x}} \in \mathbb{R}^m$ and $\dot{\mathbf{q}} \in \mathbb{R}^n$ denote the end-effector and the joint-space velocities, respectively, and $\mathbf{J} \in \mathbb{R}^{m \times n}$ denotes the Jacobian matrix.

2.1.1 Task-Space Augmentation

When the robot is redundant respect to a given task, that is $n > m$, the Jacobian matrix is not invertible. One technique of redundancy solution is to extend the dimension of the task space by imposing additional equality constraints on the joint variables.

$$\underbrace{\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}}_c \end{bmatrix}}_{\dot{\mathbf{x}}_a} = \underbrace{\begin{bmatrix} \mathbf{J} \\ \mathbf{J}_c \end{bmatrix}}_{\mathbf{J}_a} \dot{\mathbf{q}}, \quad (2.2)$$

where $\mathbf{J}_c \in \mathbb{R}^{p \times n}$, $p \leq n - m$ is the constraint-task Jacobian matrix that relates the constraint-task velocity vector $\dot{\mathbf{x}}_c$ and the joint velocities. This approach is called the task-space augmentation (Sciavicco and Siciliano, 1988) and the solution is given as

$$\dot{\mathbf{q}} = \mathbf{G}\dot{\mathbf{x}}_a, \quad (2.3)$$

where \mathbf{G} is a suitable matrix based on the Jacobian matrix, and when $p=n - m$, the problem (2.2) is called Extended Jacobian (Baillieul, 1985) and the solution is given by

$$\dot{\mathbf{q}} = \mathbf{J}_a^{-1}\dot{\mathbf{x}}_a. \quad (2.4)$$

The solution (2.4) is cyclic (i.e., cyclic joint trajectories are generated when the end-effector is required to trace a closed path) a desired behavior, which is advantageous over pseudoinverse techniques (De Luca et al., 1992). One disadvantage is the occurrence of algorithmic singularities that appear when the augmented Jacobian is singular whereas the original Jacobian matrix is not. Moreover, additional singularities happens when the constraint task conflicts with the end-effector task (Chiaverini, 1997).

2.1.2 Pseudoinverse of the Jacobian Matrix

The inverse kinematic problem is often solved via optimization aiming to find a set of velocities that minimizes the error (i.e., the difference between the current pose of the end-effector and the desired pose). A solution for this problem is given by the Moore-Penrose pseudoinverse (Siciliano and Khatib, 2016) that generates the least-norm velocities and is given as

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger \dot{\mathbf{x}}, \quad (2.5)$$

where

$$\mathbf{J}^\dagger = \begin{cases} (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T & m > n, \text{ rank}(\mathbf{J}) = n \\ \mathbf{J}^{-1} & m = n, \text{ rank}(\mathbf{J}) = n = m \\ \mathbf{J}^T (\mathbf{J} \mathbf{J}^T)^{-1} & m < n, \text{ rank}(\mathbf{J}) = m. \end{cases}$$

In this method, however, singularity avoidance cannot be guaranteed. On the other hand, when the robot is redundant with respect to the task infinite solutions exists and optimization is used to select admissible velocities based on a specified criterion. In that case some velocities produce no effect in the end-effector. This subset of velocities is called the null space of the task and can be used to perform secondary tasks. The general solution of equation (2.1) is given as follows

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger \dot{\mathbf{x}} + (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J}) \dot{\mathbf{q}}_0, \quad (2.6)$$

where $(\mathbf{I} - \mathbf{J}^\dagger \mathbf{J})$ represents the orthogonal projection matrix into the null space of \mathbf{J} and $\dot{\mathbf{q}}_0$ is an arbitrary joint-space velocity. This concept was used in robotic applications by Liégeois (1977) to perform kinematic control of a redundant manipulator arm while avoiding violations of joint limits.

2.1.3 Task-Priority Framework

When there are conflicts between the end-effector task and the constraint task, the task-space augmentation approach leads to singularities. This problem can be tackled by the task-priority strategy. In this approach the task is divided into subtasks and is assigned a priority level for each one. The lower priority tasks are satisfied only in the null space of the higher-priority task. The pseudoinverse solution is used to generate least-norm velocities. In this way, the robot will try to perform multiple tasks and when an exact solution does not exist, the higher priority task is fulfilled at expenses of ignoring the lower-priority ones. The generalization to any number of tasks was proposed by Nakamura et al. (1987) and its recursive expression was proposed by Siciliano and Slotine (1991).

This strategy has been widely applied to redundant manipulators (Chiaverini et al.,

2016) and in general to redundant systems, but it considers equality constraints only. In order to take into account inequality constraints some works were addressed by adding repulsive fields (Khatib, 1986) in the task priority strategy to impose for instance joint-limit constraints (Sentis and Khatib, 2005). However, this is a restrictive method (in the sense that the inequality constraints are transformed into equality constraints) that produces suboptimal solutions and is likely to produce a discontinuity (Kanoun et al., 2011).

Mansard et al. (2009) proposed a smooth control law to deal with equality and inequality constraints in the task-priority strategy. The authors used an inverse operator to smooth the irregularity of the unilateral constraint. The control law is based on the hierarchical set of tasks framework with inequality constraints. However, this method has a high computational cost.

Recent works have been approached by using numerical solvers in quadratic programming (QP) in order to take into account equalities and inequalities constraints, requiring a lower computational cost (Escande et al., 2010; Kanoun et al., 2011; Escande et al., 2014).

2.2 Kinematic Control as a Linear Programming

The analytical solution of the inverse kinematic problem is not just elegant but also very efficient. However, when inequality constraints are imposed, such solution does not exist. In that case, the inverse kinematic problem can be expressed as mathematical optimization (also called mathematical programming) and numerical solvers, mostly quadratic programming, have been widely used (Escande et al., 2012; Kim and Oh, 2013; Escande et al., 2014; Dai and Tedrake, 2016). The main advantage of the mathematical optimization is the possibility of imposing easily inequality constraints directly in the mathematical formulation, a feature that can be exploited in robotic systems in order to bound the accelerations and joint velocities or avoid, for instance, violation of joint-limits and collision with obstacles. However, when the number of (DOF) or constraints is very large, the computational cost increases considerably. This issue can be tackled by using linear programming, where recent works showed that it is a powerful and efficient tool to solve highly constrained system. Furthermore, in the linear programming formulation, a parsimonious behavior, when feasible, can be obtained when the Simplex method is used (Goncalves et al., 2016).

Nakamura et al. (2005) proposed a method to compute the forward and inverse dynamics of a musculoskeletal human model, a hyper-redundant model, using linear programming and the Simplex algorithm. The proposed method was compared to convex quadratic programming using an algorithm based on interior points. Simulation results shown that the method based on linear programming has lower computational cost than the method based on quadratic programming; in the particular hyper-redundant model used, the former was about 4.77 faster than the latter.

Ho et al. (2005) proposed an inverse kinematic formulation based on linear programming applied to highly redundant structures. The optimization criterion is based on the minimum sum of the absolute values of the joint velocities instead of calculating the least-square solution (as convex quadratic programming). In this approach the control inputs suffer from jittering. In order to mitigate it the authors proposed a new objective function that takes into account a solution calculated in a previous step. Simulation results shown a low computational cost of the linear programming approach and even more efficiency (when the number of constraints is large) than the nonlinear programming approach based on Lagrange multipliers.

Kingston et al. (2015) presented a kinematic control method for robot manipulators using linear optimization subject to position, acceleration and velocity constraints and robust to kinematic singularities. In this approach the acceleration of the joints are minimized in the 1-norm sense. Experimental results shown that the proposed method is fast enough to perform online feedback control despite being 20 times slower than the analytical solution based on the damped pseudoinverse of the Jacobian matrix. However, it is important highlight that the latter does not allow to define inequality constraints.

Goncalves et al. (2016) proposed the parsimonious kinematic control where the inverse kinematic formulation was approached by using linear programming and the Simplex algorithm. In this paradigm, it is desired to use a minimum amount of actuators to perform a particular task. This formulation is more general than the one proposed by Ho et al. (2005) and is more computationally efficient and easier to implement. The Lyapunov stability was formally proved and the experimental results shown than with the parsimonious approach the robot uses less actuators than when it uses the traditional approach that uses the the pseudoinverse of the Jacobian matrix. However, the control inputs generated by the former are more abrupt than those generated by the latter.

It is important highlight that in comparison with the other methods based on linear programming, the method proposed by Goncalves et al. (2016) is the only one that has a formal proof of stability. In addition, the linear program proposed by the authors to deal with the inverse kinematic problem is computationally efficient. In the method proposed by Nakamura et al. (2005) the main concern is to compute forces and tensions instead of joint velocities. Finally, the method proposed by Kingston et al. (2015) is concerned with computing the joint accelerations and the second-order differential kinematic model is required.

2.3 Kinematic Control of Nonholonomic Mobile Manipulators

Mobile manipulators are mobile platforms equipped with a conventional manipulator arm. Often, those robot systems are redundant and allow to increase not just the dexterity of the robot, but also the workspace of the manipulator. In general, the mobile platform is composed of wheeled bases or walking and flying robots, although the former are by far the most common locomotion systems in the mobile platforms. A mobile platform can be holonomic (i.e., it is able to perform movements instantaneously in any direction) or nonholonomic (i.e., it is not able to perform movements instantaneously in any direction but first the orientation of the platform must be aligned before the movement) if it has holonomic or nonholonomic constraints, respectively. Although the nonholonomic constraints limit the instantaneous mobility of the platform it can attain any position and orientation in the plane.

In order to perform motion control of a nonholonomic mobile manipulator, its nonholonomic constraints must be taken into account. Often, this can be addressed including the nonholonomic constraints into the differential kinematic description (i.e., the nonholonomic constraint is modeled as an equality constraint and it is included in the extended differential kinematic description) (Seraji, 1998; Jia et al., 2014). However, a more efficient formulation is express explicitly in the differential kinematic description the admissible platform velocities Gardner and Velinsky (2000); Bayle et al. (2003).

De Luca et al. (2006) presented an extension of redundancy resolution schemes originally developed to standard manipulators and applied to a nonholonomic mobile manipulator. The authors obtained the kinematic model of the robot considering the nonholonomic constraints of the mobile platform. Two control strategies were implemented, namely the Projected and Reduced Gradient optimization methods. The former is based on the pseudoinverse of the robot Jacobian matrix and its null space projection operator. A suitable null space vector (as secondary task) that allows to obtain admissible velocities for the nonholonomic mobile platform is used. The latter is based on the Reduced Gradient method (De Luca and Oriolo, 1991) where only the extra DOF are used for optimization. A numerical comparison was performed and results shown superior performance of the Reduced Gradient method over the Projected Gradient method.

Liu and Li (2006) proposed an extended gradient projection as redundancy resolution method for multiple secondary tasks and applied the technique to redundant mobile manipulators. A real-time fuzzy logic planner was implemented in order to generate self-motions and an adaptive neural-network controller was also implemented to perform multiple secondary tasks without affecting the primary one. The nonholonomic constraints of the mobile platform were taken into account by a general dynamic modeling.

Zhang et al. (2012) proposed a redundancy resolution method based on online sensor

information on the environment task and the pseudoinverse of the Jacobian matrix as well as its null space projection operator. The differential kinematic description included the admissible platform velocities and the nonholonomic constraint is taken into account by generating feasible velocities. Additional constraints and particular requirements are modeled as a multi-objective optimization problem as done by De Luca et al. (2006).

Others works (Giordano et al., 2009; De Luca et al., 2010) addressed the control of nonholonomic mobile manipulators with steering wheels. In this case, it is common that the steering velocity inputs do not appear in the forward differential kinematic model and, consequently, the number of effective control inputs are not used in the control laws. The authors proposed control solutions based on the framework of input-output feedback linearization. However, the modeling and control of such robots is often more complex. In general the mobile platform is over-actuated (i.e., several wheels to control the pose of the mobile platform with two actuators per wheel to control its angular velocity and its orientation). In such system it is required to orient precisely the wheels and a method commonly used is based on the Instantaneous Center of Rotation (ICA) (Tin Lun Lam et al., 2009). The idea is to define a point in the world frame, which instantaneously does not change with respect to the robot and before to perform a movement, each wheel axis of the robot must be aligned with respect to such point. However, this strategy leads to singularities (Dietrich et al., 2011). Furthermore, it is common that the steerable wheels located in the robot center lead to kinematic singularities Reister and Unseren (1993); Thuilot et al. (1996); Giordano et al. (2009). Those issues are tackled by Stoger et al. (2015) by using a regular parametrization of the robot's motions and by using a control strategy based on input-output feedback linearization in terms of a path parameter.

Jia et al. (2014) presented a planning and control method to efficiently handle uncertainties of the system. This paradigm assumes that the nonholonomic manipulator has uncertainties and could suffer unexpected events because both the arm manipulator and the mobile platform are different structures resulting in different motion dynamics and errors. The authors approached this issue by using a cascade control structure. An inner loop dealt with the kinematic control and is based on the work proposed by De Luca et al. (2006) and the outer loops dealt with a planning control process based on the system's output measurements to handle these events and achieve the best possible performance.

Salazar-Sangucho and Adorno (2014) proposed a kinematic control strategy applied to a nonholonomic differential mobile manipulator based on the damped pseudoinverse of the robot Jacobian matrix. The robot was modeled based on a systematic procedure using dual quaternion algebra proposed by Adorno (2011) and the nonholonomic constraint was not added in the differential kinematic description. Instead, this strategy was solved by using a cascade control scheme where the inner loop dealt with the nonholonomic constraint by using an input-output feedback linearization and the outer loop was responsible by the whole-body motions. Experiments on a real platform shown that the control strategy

proposed works well in practice, although the stability of the system was not formally proved using the cascade scheme.

Silva and Adorno (2016) proposed a whole-body pose control of a nonholonomic mobile manipulator based on a nonlinear controller using the cascade scheme proposed by Salazar-Sangucho and Adorno (2014). In the outer loop a nonlinear controller based on dual-quaternion feedback linearization was proposed. Experimental results shown lower abrupt initial movement in comparison with the classic approach proposed by Salazar-Sangucho and Adorno (2014). However, the strategy proposed by Silva and Adorno, allows to perform pose task only. Other tasks as position of the end-effector using this strategy are not trivial and were not shown. In addition, this kinematic strategy works well in practice but its stability was not formally proved.

Quiroz-Omana and Adorno (2016) presented a kinematic control strategy of a nonholonomic mobile manipulator using the cascade control scheme proposed by Salazar-Sangucho and Adorno (2014) in order to deal with the nonholonomic constraint of the mobile platform. However, the outer loop was approached in a different way by using the recently parsimonious kinematic control strategy proposed by Goncalves et al. (2016), where the whole-body motions are based on linear programming and the Simplex algorithm. In this paradigm, only a minimum number of actuators required to execute a particular task is used. Experiments on a real platform shown that the parsimonious controller has a low computational cost and it used less joints per iteration than the classic controller based on the pseudoinverse of the Jacobian matrix. However, control inputs generated by the former were more abrupt than the latter and the stability of the system was not formally proved using the cascade scheme.

The work presented in this dissertation is based on the inverse kinematic control using linear programming proposed by Goncalves et al. (2016) taking into account the nonholonomic constraint of the mobile platform as an equality constraint, avoiding in this way the need to use a cascade control scheme. Additional constraints are imposed in order to prevent violations of joint limits and prevent collisions between the mobile base and obstacles in the plane.

2.4 Chapter Conclusions

This chapter presented some important works related to the kinematic control of redundant systems. Section 2.1 reviewed the classic methods widely used in redundancy solution based mainly on the pseudoinverse of the Jacobian matrix. When it is desired to perform additional tasks using the extra DOF available the task-space augmentation can be used. In this strategy the end-effector task and the constraints task are stacked resulting in an augmented kinematic problem. Nevertheless, when conflicts between the task arise, this method leads to algorithmic singularities. In order to handle this problem, the task-priority

strategy has been widely used. In this technique, the task is divided into subtasks according of the order of priority and lower priority tasks are satisfied only in the null space of task with higher priority. Section 2.2 presented some important works in kinematic control of redundant system using linear programming. The main advantage of this approach with respect to the classic ones is the possibility to easily include inequality constraints in the formulation, in addition off being computationally efficient. Nakamura et al. (2005) used the linear programming and the Simplex algorithm to compute the forward and inverse dynamics of a hyper-redundant human-like model. In this approach was used the Jacobian matrix to relate the joint torques and the tension of the muscles. Ho et al. (2005) used this approach to approach the inverse kinematic problem on hyper-redundant system. But the general formulation was proposed by Goncalves et al. (2016) where the authors shown not just experimental results with good performance but also the formal proof of stability. Kingston et al. (2015) proposed a kinematic control formulation based on linear programming, where it is performed a minimization of the joint accelerations. The experimental results shown that the method proposed by the authors works well in practice but the stability was not proved. Finally, Section 2.3 presented some works related to the kinematic control of nonholonomic mobile manipulators. The classic techniques to perform kinematic control in redundant robots, which are mostly based on the pseudoinverse of the Jacobian matrix were extended to nonholonomic mobile manipulators by De Luca et al. (2006). This approach was used by Liu and Li (2006) and Zhang et al. (2012), which different approaches to solve secondary task in the null space of the Jacobian matrix were used. Jia et al. (2014) proposed a method to take into account the different motion dynamics between the mobile base and the manipulator arm using planning control. In absence of uncertain or unexpected events this method is equivalent to the traditional task-level control with redundancy resolution. Salazar-Sangucho and Adorno (2014) used the classic method based on the pseudoinverse of the Jacobian matrix and a cascade control scheme were proposed in order to deal with the nonholonomic constraint of the mobile base. Silva and Adorno (2016) used a nonlinear controller to perform kinematic control and the nonholonomic constraint was tackled as done by Salazar-Sangucho and Adorno (2014).

The control strategies based on the pseudoinverse of the Jacobian matrix allow to impose equality constraints only. Often, the inequality constraints are addressed by transforming those inequalities into equalities constraints. On the other hand, when the kinematic control is formulated as mathematical programming, inequality constraints as well as equality constraints can be imposed easily. The method proposed by Goncalves et al. (2016) is used in this work because is computationally efficient and the Lyapunov stability is guaranteed. Furthermore, when the Simplex method is used, the controller tends to use a minimum amount of actuators. This parsimonious behavior could be induced less dynamics disturbances, a useful feature in the sense that, in the kinematic control

approach the dynamics components are not considered. Such parsimonious behavior is not guaranteed (at least easily) using solvers based on quadratic programming.

3

Mathematical Background

This chapter reviews some concepts, foundations and operations related to quaternions and dual quaternions. Furthermore it presents a brief exposition on linear programming.

3.1 Fundamentals of Dual Quaternion Algebra

Unit dual quaternions have proven to be a powerful mathematical tool in robotics, not only in the representation of rigid motions, but also in robot modeling (Adorno 2011; Selig 2005), robot design (Perez and McCarthy, 2004), and control (Pham et al., 2010; Xiangke Wang et al., 2012; Figueredo et al., 2013; Wang and Yu, 2013; Marinho et al., 2015; Kussaba et al., 2017). They are more compact and computationally efficient than homogeneous transformation matrices and also do not present representational singularities (Adorno, 2011; Adorno and Fraitse, 2016). Thanks to their strong algebraic properties, different robots can be modeled using the same systematic procedure (e.g., single or cooperative manipulators (Adorno et al., 2010), mobile manipulators (Adorno, 2011) and humanoids (Oliveira and Adorno, 2015; Fonseca and Adorno, 2016), and the resultant models can be directly used with standard kinematic controllers without the need of any intermediate parameterization (Pham et al., 2010; Figueredo et al., 2013). Unit dual quaternions represents rigid motions in a very compact way, by combining a unit quaternion¹ representing rotation and a pure quaternion² representing translation.

¹ $\mathbf{h} \in \mathbb{H}$ is a unit quaternion if $\|\mathbf{h}\| = 1$.

² $\mathbf{h} \in \mathbb{H}$ is a pure quaternion if $\text{Re}(\mathbf{h}) = 0$.

3.1.1 Quaternions

Quaternions are algebraic structures that can be regarded as an extension of complex numbers introduced first by Sir William Rowan Hamilton in 1843 (1844, apud Adorno). They are composed of a real part and three imaginary components \hat{i} , \hat{j} , \hat{k} , also called imaginary or quaternionic units. The imaginary units have the following properties

$$\hat{i}^2 = \hat{j}^2 = \hat{k}^2 = \hat{i}\hat{j}\hat{k} = -1. \quad (3.1)$$

Definition 3.1. Given $h_1, h_2, h_3, h_4 \in \mathbb{R}$, the quaternion $\mathbf{h} \in \mathbb{H}$ is defined as

$$\mathbf{h} \triangleq h_1 + \hat{i}h_2 + \hat{j}h_3 + \hat{k}h_4, \quad (3.2)$$

where the real part is denoted by $\text{Re}(\mathbf{h}) \triangleq h_1$, and the imaginary part is denoted by $\text{Im}(\mathbf{h}) \triangleq \hat{i}h_2 + \hat{j}h_3 + \hat{k}h_4$, such that $\mathbf{h} = \text{Re}(\mathbf{h}) + \text{Im}(\mathbf{h})$.

Definition 3.2. Given $\mathbf{h} \in \mathbb{H}$, its conjugate is defined as

$$\mathbf{h}^* \triangleq \text{Re}(\mathbf{h}) - \text{Im}(\mathbf{h}). \quad (3.3)$$

Definition 3.3. Given $\mathbf{h} \in \mathbb{H}$, its norm is defined as

$$\|\mathbf{h}\| \triangleq \sqrt{\mathbf{h}^* \mathbf{h}} = \sqrt{\mathbf{h} \mathbf{h}^*}. \quad (3.4)$$

Definition 3.4. The $\text{vec}_4 : \mathbb{H} \rightarrow \mathbb{R}^4$ operator performs a one-to-one mapping. Given the quaternion $\mathbf{h} \triangleq h_1 + \hat{i}h_2 + \hat{j}h_3 + \hat{k}h_4$, this operator is defined as (Adorno, 2011)

$$\text{vec}_4(\mathbf{h}) \triangleq \begin{bmatrix} h_1 & h_2 & h_3 & h_4 \end{bmatrix}^T. \quad (3.5)$$

Definition 3.5. Given $\mathbf{h} \in \mathbb{H}$, the Hamilton operators (Adorno, 2011) are defined as

$$\overset{+}{\mathbf{H}}_4(\mathbf{h}) \triangleq \begin{bmatrix} h_1 & -h_2 & -h_3 & -h_4 \\ h_2 & h_1 & -h_4 & h_3 \\ h_3 & h_4 & h_1 & -h_2 \\ h_4 & -h_3 & h_2 & h_1 \end{bmatrix}, \quad \bar{\mathbf{H}}_4(\mathbf{h}) \triangleq \begin{bmatrix} h_1 & -h_2 & -h_3 & -h_4 \\ h_2 & h_1 & h_4 & -h_3 \\ h_3 & -h_4 & h_1 & h_2 \\ h_4 & h_3 & -h_2 & h_1 \end{bmatrix} \quad (3.6)$$

Definition 3.6. Given $\mathbf{a}, \mathbf{b} \in \mathbb{H}$, the Hamilton operators satisfy the following equalities (Adorno, 2011)

$$\text{vec}_4(\mathbf{ab}) = \overset{+}{\mathbf{H}}_4(\mathbf{a}) \text{vec}_4(\mathbf{b}) = \bar{\mathbf{H}}_4(\mathbf{b}) \text{vec}_4(\mathbf{a}). \quad (3.7)$$

Definition 3.7. The conjugating matrix \mathbf{C}_4 is defined as (Adorno, 2011)

$$\mathbf{C}_4 \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}. \quad (3.8)$$

Given $\mathbf{h} \in \mathbb{H}$, this matrix satisfies the following condition

$$\text{vec}_4(\mathbf{h}^*) = \mathbf{C}_4 \text{vec}_4(\mathbf{h}).$$

3.1.2 Dual Quaternions

Dual quaternions are dual numbers where the primary and dual part are quaternions. The dual numbers were introduced by Clifford (Adorno, 2011) and can be regarded as an extension of quaternions.

Definition 3.8. Given two numbers $d_{\mathcal{P}}$ and $d_{\mathcal{D}}$, the dual number $\underline{\mathbf{d}}$ is defined as (Adorno, 2011)

$$\underline{\mathbf{d}} = d_{\mathcal{P}} + \varepsilon d_{\mathcal{D}}, \quad (3.9)$$

where ε is the dual unit proposed by Clifford (1873), which is nilpotent and follows the following properties

$$\varepsilon^2 = 0 \text{ with } \varepsilon \neq 0. \quad (3.10)$$

The primary part and the dual part can be extracted using the operators $\mathcal{P}(\underline{\mathbf{d}})$ and $\mathcal{D}(\underline{\mathbf{d}})$, respectively. For instance, in (3.9) $\mathcal{P}(\underline{\mathbf{d}}) = d_{\mathcal{P}}$ and $\mathcal{D}(\underline{\mathbf{d}}) = d_{\mathcal{D}}$.

Definition 3.9. The dual quaternion $\underline{\mathbf{h}} \in \mathcal{H}$ is defined as

$$\underline{\mathbf{h}} \triangleq \mathbf{h}_{\mathcal{P}} + \varepsilon \mathbf{h}_{\mathcal{D}},$$

where $\mathbf{h}_{\mathcal{P}}, \mathbf{h}_{\mathcal{D}} \in \mathbb{H}$.

Definition 3.10. Given the dual quaternion $\underline{\mathbf{h}} = \mathbf{h}_{\mathcal{P}} + \varepsilon \mathbf{h}_{\mathcal{D}}$, its conjugate is defined as

$$\underline{\mathbf{h}}^* \triangleq \mathbf{h}_{\mathcal{P}}^* + \varepsilon \mathbf{h}_{\mathcal{D}}^*. \quad (3.11)$$

Definition 3.11. Given $\underline{\mathbf{h}} \in \mathcal{H}$, its norm is defined as

$$\|\underline{\mathbf{h}}\| \triangleq \sqrt{\underline{\mathbf{h}}^* \underline{\mathbf{h}}} = \sqrt{\underline{\mathbf{h}} \underline{\mathbf{h}}^*}.$$

Definition 3.12. The $\text{vec}_8 : \mathbb{H} \rightarrow \mathbb{R}^8$ operator performs a one-to-one mapping. Given the quaternion $\underline{\mathbf{h}} = h_1 + \hat{i}h_2 + \hat{j}h_3 + \hat{k}h_4 + \varepsilon (h_5 + \hat{i}h_6 + \hat{j}h_7 + \hat{k}h_8)$, this operator is defined as

$$\text{vec}_8(\underline{\mathbf{h}}) \triangleq \begin{bmatrix} h_1 & \dots & h_8 \end{bmatrix}^T.$$

The multiplication and addition operations between dual quaternions follow the same rules of their counterparts between real numbers, but respecting the additional rules determined by (3.1) and (3.10). It can be verified that, in general, multiplication of dual quaternions is not commutative (i.e., given $\underline{\mathbf{x}}, \underline{\mathbf{y}} \in \mathcal{H}$, in general $\underline{\mathbf{x}}\underline{\mathbf{y}} \neq \underline{\mathbf{y}}\underline{\mathbf{x}}$). However, one can use Hamilton operators (Adorno, 2011) for manipulating algebraic expressions containing dual quaternions such that

$$\text{vec}(\underline{\mathbf{x}}\underline{\mathbf{y}}) = \overset{+}{\mathbf{H}}(\underline{\mathbf{x}}) \text{vec} \underline{\mathbf{y}} = \overset{-}{\mathbf{H}}(\underline{\mathbf{y}}) \text{vec}(\underline{\mathbf{x}}),$$

where

$$\overset{+}{\mathbf{H}}(\underline{\mathbf{h}}) \triangleq \begin{bmatrix} \overset{+}{\mathbf{H}}_4(\mathbf{h}_{\mathcal{P}}) & \mathbf{0}_{4 \times 4} \\ \overset{+}{\mathbf{H}}_4(\mathbf{h}_{\mathcal{D}}) & \overset{+}{\mathbf{H}}_4(\mathbf{h}_{\mathcal{P}}) \end{bmatrix}, \quad \overset{-}{\mathbf{H}}(\underline{\mathbf{h}}) \triangleq \begin{bmatrix} \overset{-}{\mathbf{H}}_4(\mathbf{h}_{\mathcal{P}}) & \mathbf{0}_{4 \times 4} \\ \overset{-}{\mathbf{H}}_4(\mathbf{h}_{\mathcal{D}}) & \overset{-}{\mathbf{H}}_4(\mathbf{h}_{\mathcal{P}}) \end{bmatrix}. \quad (3.12)$$

3.2 Rigid Motions using Dual Quaternions

Definition 3.13. consider a rotation of an angle ϕ around a unit-norm axis $\mathbf{n} = n_x \hat{i} + n_y \hat{j} + n_z \hat{k}$, with respect to a fixed frame \mathcal{F}_a and let \mathcal{F}_b be the resultant frame after this rotation. This rotation is represented as

$$\mathbf{r}_b^a = \cos\left(\frac{\phi}{2}\right) + \mathbf{n} \sin\left(\frac{\phi}{2}\right). \quad (3.13)$$

The rotation represented by (3.13) is shown in Fig. 3.1.

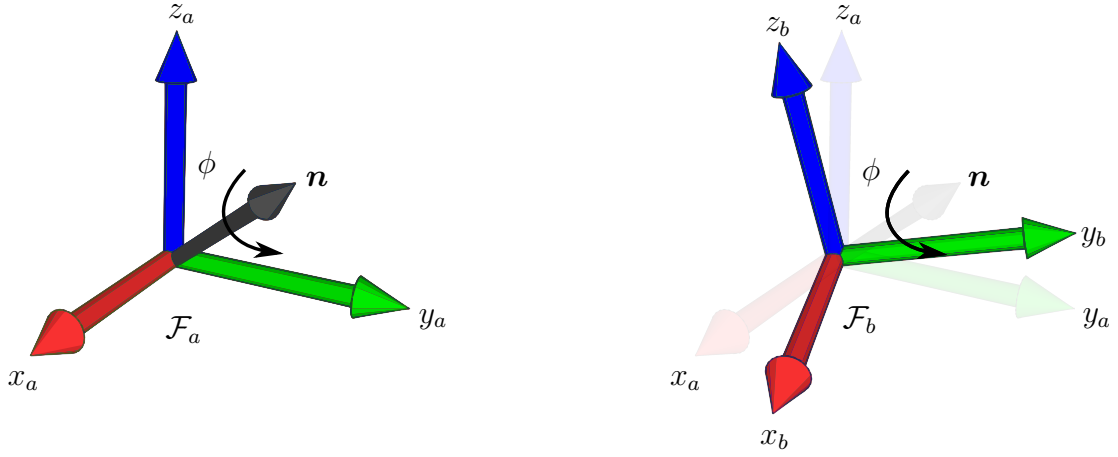


Figure 3.1: Representation of a rotation of an angle ϕ around the unit-norm axis \mathbf{n} .

Definition 3.14. The translation from \mathcal{F}_a to frame \mathcal{F}_b , expressed in \mathcal{F}_a is represented by a pure quaternion \mathbf{p}_{ab}^a given by

$$\mathbf{p}_{ab}^a = p_x \hat{i} + p_y \hat{j} + p_z \hat{k}. \quad (3.14)$$

Definition 3.15. Given $\mathbf{r}_b^a, \mathbf{p}_{ab}^a \in \mathbb{H}$, defined by (3.13) and (3.14), representing the rotation and translation, respectively. The rigid transformation composed of a translation and then a orientation is represented by the unit dual quaternion and is defined as (Adorno, 2011)

$$\mathbf{x}_1^0 = \mathbf{r}_1^0 + \varepsilon \frac{1}{2} \mathbf{p}_{01}^0 \mathbf{r}_1^0. \quad (3.15)$$

Definition 3.16. Given the frames $\mathcal{F}_0, \mathcal{F}_1$, and \mathcal{F}_2 , the unit dual quaternions \mathbf{x}_1^0 and \mathbf{x}_2^1 represent the rigid motions from \mathcal{F}_0 to \mathcal{F}_1 and \mathcal{F}_1 to \mathcal{F}_2 , respectively. The transformation from \mathcal{F}_0 to \mathcal{F}_2 is given by $\mathbf{x}_2^0 = \mathbf{x}_1^0 \mathbf{x}_2^1$ (Adorno, 2011), as shown in Fig. 3.2.

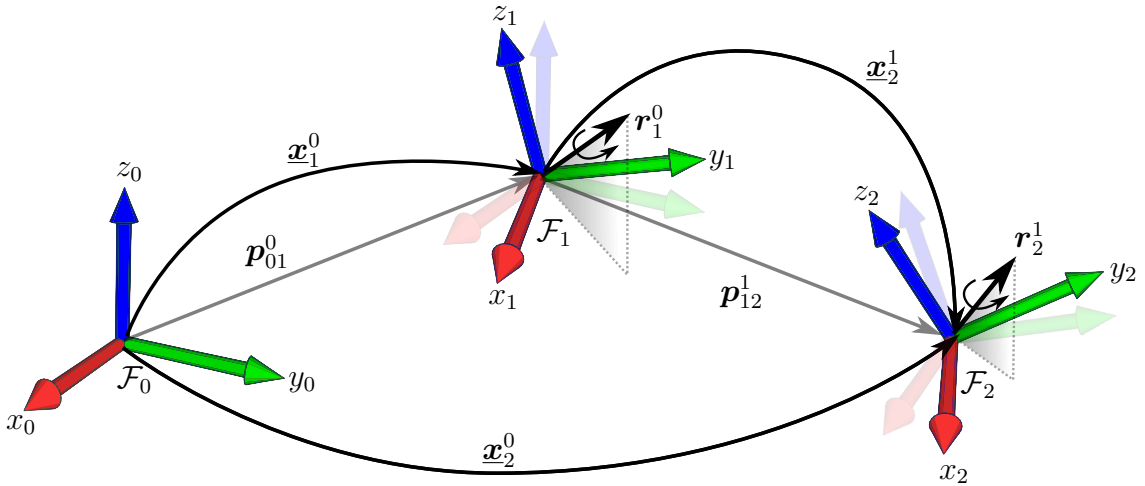


Figure 3.2: Sequence of rigid transformations using dual quaternions.

Definition 3.17. Given the unit dual quaternion $\underline{\mathbf{x}} = \mathbf{r} + \frac{1}{2}\varepsilon\mathbf{p}\mathbf{r}$, with $r = \cos(\phi/2) + \mathbf{n} \sin(\phi/2)$, $\mathbf{n} = n_x\hat{i} + n_y\hat{j} + n_z\hat{k}$ and $\mathbf{p} = p_x\hat{i} + p_y\hat{j} + p_z\hat{k}$, the logarithm of $\underline{\mathbf{x}}$ is defined as (Adorno, 2012)

$$\log \underline{\mathbf{x}} \triangleq \frac{\mathbf{n}\phi}{2} + \varepsilon\frac{\mathbf{p}}{2}. \quad (3.16)$$

The logarithm of a unit dual quaternion is a dual quaternion with real part equal to zero, however and not necessarily has unit norm. Note that when $\log \underline{\mathbf{x}}$ is multiplied by a scalar, the rotation axis and the direction of the translation are not affected. Only the magnitude of the rotation angle and the translation are affected.

Definition 3.18. Given the pure dual quaternion $\underline{\mathbf{a}}$ (i.e, $\text{Re}(\underline{\mathbf{a}}) = 0$), its exponential is expressed by Adorno (2012)

$$\exp \underline{\mathbf{a}} \triangleq \mathcal{P}(\exp \underline{\mathbf{a}}) + \varepsilon \mathcal{D}(\underline{\mathbf{a}}) \mathcal{P}(\exp \underline{\mathbf{a}}), \quad (3.17)$$

where

$$\mathcal{P}(\exp \underline{\mathbf{a}}) = \begin{cases} \cos \|\mathcal{P}(\underline{\mathbf{a}})\| + \frac{\sin \|\mathcal{P}(\underline{\mathbf{a}})\|}{\|\mathcal{P}(\underline{\mathbf{a}})\|} \mathcal{P}(\underline{\mathbf{a}}) & \text{if } \|\mathcal{P}(\underline{\mathbf{a}})\| \neq 0 \\ 1 & \text{otherwise.} \end{cases}$$

Definition 3.19. Given $\lambda \in \mathbb{R}$, the unit dual quaternion $\underline{\mathbf{x}}$ raised by λ is given by Adorno (2012)

$$\underline{\mathbf{x}}^{\{\lambda\}} = \exp(\lambda \log \underline{\mathbf{x}}). \quad (3.18)$$

Example 3.1. Given a unit dual quaternion $\underline{\mathbf{x}} = \mathbf{r} + \varepsilon(1/2)\mathbf{p}\mathbf{r}$, where $\mathbf{r} = \cos(\phi/2) + \mathbf{n} \sin(\phi/2)$ and $\mathbf{p} = p_x\hat{i} + p_y\hat{j} + p_z\hat{k}$ that represents the transformation from a reference frame \mathcal{F}_0 to the frame \mathcal{F}_1 , the operation of exponentiation $\underline{\mathbf{x}}^{\{1/2\}}$ is given as follows

$$\begin{aligned} \underline{\mathbf{x}}^{\{1/2\}} &= \exp\left(\frac{1}{2} \log \underline{\mathbf{x}}\right) \\ &= \exp\left(\frac{\mathbf{n}\phi}{4} + \varepsilon\frac{\mathbf{p}}{4}\right) \\ &= \left(\cos \frac{\phi}{4} + \mathbf{n} \sin \frac{\phi}{4}\right) + \frac{1}{2}\varepsilon \left[\frac{\mathbf{p}}{2} \left(\cos \frac{\phi}{4} + \mathbf{n} \sin \frac{\phi}{4}\right)\right] \\ &= (\mathbf{r})^{\{1/2\}} + \frac{1}{2}\varepsilon \left(\frac{\mathbf{p}}{2}\right) (\mathbf{r})^{\{1/2\}}, \end{aligned}$$

and provides the rigid movement that corresponds to intermediate frame between the frames \mathcal{F}_0 and \mathcal{F}_1 . The unit dual quaternion $\underline{\mathbf{x}}^{\{1/2\}}$ represents a rigid motion given by half rotation and half translation of $\underline{\mathbf{x}}$, as shown in Fig. 3.3, denoted by the red dashed square.

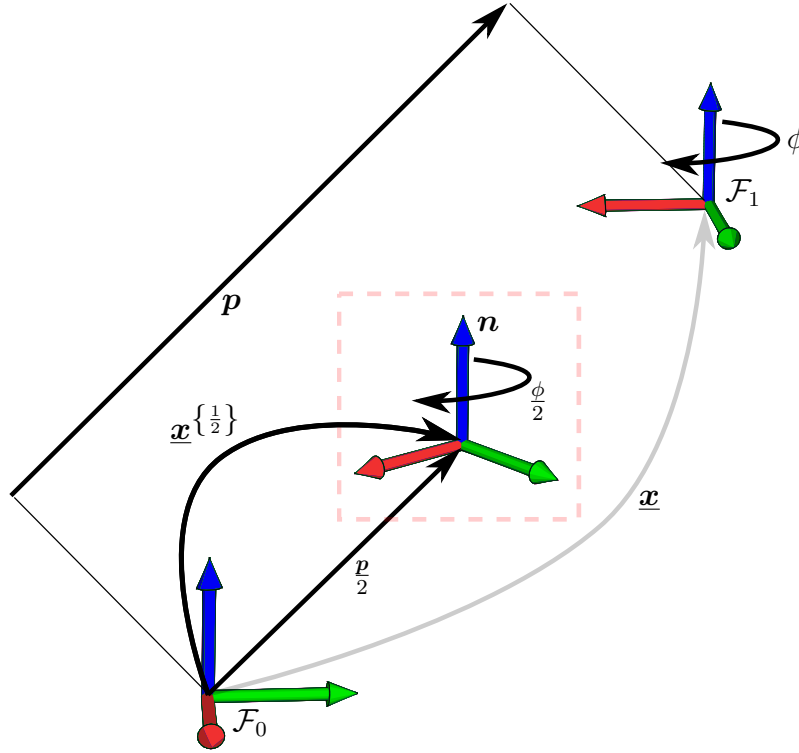


Figure 3.3: Given a reference frame \mathcal{F}_0 , the rigid motion from the reference frame to the frame \mathcal{F}_1 is represented by the unit dual quaternion \underline{x} . The unit dual quaternion $\underline{x}^{\{\frac{1}{2}\}}$ represents the half of the translation and the half of the rotation angle of \underline{x} . This resultant frame is shown inside the red dashed square.

3.3 Fundamentals in Linear Programming

The main purpose of the optimization problem is to find the best solution with respect to some criterion from some set of available possibilities. This is obtained maximizing or minimizing a function called objective function, respecting a set of equality or inequality constraints. When the objective function and the equality and inequality constraints are linear functions, the optimization problem is called linear programming. An optimization problem can be represented in the canonical form as follows

$$\begin{aligned}
 \min_{\mathbf{g}} \quad & \mathbf{c}^T \mathbf{g} + c \\
 \text{subject to} \quad & \mathbf{A} \mathbf{g} = \mathbf{b} \\
 & \mathbf{g} \geq 0,
 \end{aligned} \tag{3.19}$$

where $\mathbf{c}^T \mathbf{g} + c$ is the objective function, \mathbf{g} is the vector of decision variables, and $\mathbf{A} \mathbf{g} = \mathbf{b}$ are the equality constraints.

3.3.1 Basic Operations

Any model of linear program can be rewritten in the canonical form without loss of generality by using basic operations (Vanderbei, 1998).

Remark 3.1. Given a linear program with a objective function $f(\mathbf{g})$, the optimization criterion can be modified. Minimizing an objective function $f(\mathbf{g})$ is equivalent to maximizing $-f(\mathbf{g})$ and vice versa. This is expressed as follows

$$\min_{\mathbf{g}} f(\mathbf{g}) = -\max_{\mathbf{g}} -f(\mathbf{g}),$$

$$\max_{\mathbf{g}} f(\mathbf{g}) = -\min_{\mathbf{g}} -f(\mathbf{g}). \quad (3.20)$$

Remark 3.2. A free variable $g_i \in \mathbb{R}$, (i.e., an unconstrained variable that can assume positive, negative, or zero values) can be expressed as the sum of two nonnegative variables as follows

$$g_i = g_{1i} - g_{2i}, \quad (3.21)$$

where $g_{1i}, g_{2i} \geq 0$.

Remark 3.3. An linear inequality can be transformed into an linear equality and vice versa by adding a nonnegative variable called slack variable. Thus, given an inequality

$$x_1 + x_2 + \dots + x_n \leq b, \quad (3.22)$$

it can be rewritten as follows

$$x_1 + x_2 + \dots + x_n + x_{n+1} = b, \quad (3.23)$$

where $x_{n+1} \geq 0$ is called slack variable.

3.3.2 Example

Given the linear program,

$$\begin{aligned} \max_{x_1, x_2} \quad & 3x_1 + 2x_2 \\ \text{subject to} \quad & 2x_1 + x_2 \leq 18 \\ & 2x_1 + 3x_2 \leq 42 \\ & x_1 \geq 0, x_2 \in \mathbb{R}, \end{aligned} \quad (3.24)$$

we desired solve it by using the simplex method, which can be applied when the linear program is expressed in the canonical form. However, this is not the case because the problem 3.24 is subjected to inequalities constraints and the variable x_2 is a free variable.

In order to rewrite the linear program 3.24 in the canonical form, two basic operations are required. First, it is added a slack variable z in each inequality constraint as follows

$$2x_1 + x_2 + z_A = 18, \quad (3.25)$$

$$2x_1 + 3x_2 + z_B = 42, \quad (3.26)$$

where z_A and z_B are nonnegative variables; that is, $z_A, z_B \geq 0$.

The second operation is to express the free variable x_2 as the sum of two nonnegative variables as follows

$$x_2 \triangleq x_{2P} - x_{2N}, \quad (3.27)$$

where $x_{2P}, x_{2N} \geq 0$.

Then, the problem 3.24 can be rewritten in the canonical form and is given as follows

$$\begin{aligned} & \max_{x_1, x_{2P}, x_{2N}, z_A, z_B} && 3x_1 + 2x_{2P} - 2x_{2N} \\ & \text{subject to} && 2x_1 + x_{2P} - x_{2N} + z_A = 18 \\ & && 2x_1 + 3x_{2P} - 3x_{2N} + z_B = 42 \\ & && x_1 \geq 0, x_{2P} \geq 0, x_{2N} \geq 0, z_A \geq 0, z_B \geq 0. \end{aligned} \quad (3.28)$$

Some numerical solvers, such as the Matlab linprog function, requires the linear program as a minimization problem and expressed in matrix form, whose transformation is trivial in this case. A maximization of the linear function f is equivalent to the minimization of $-f$. The problem 3.28 can be rewritten as follows,

$$\begin{aligned} & \min_{\mathbf{g}} && \mathbf{c}^T \mathbf{g} \\ & \text{subject to} && \mathbf{A} \mathbf{g} = \mathbf{b} \\ & && \mathbf{g} \geq 0, \end{aligned} \quad (3.29)$$

where $\mathbf{c} \triangleq [-3 \ -2 \ 2 \ 0 \ 0]^T$, $\mathbf{g} \triangleq [x_1 \ x_{2P} \ x_{2N} \ z_A \ z_B]^T$, $\mathbf{A} \triangleq \begin{bmatrix} 2 & 1 & -1 & 1 & 0 \\ 2 & 3 & -3 & 0 & 1 \end{bmatrix}$

and $\mathbf{b} \triangleq [18 \ 42]^T$.

The optimization problem 3.29 is expressed in the canonical form and using the matrix form and can be solved by using Matlab as follows

$$g = \text{linprog}(\mathbf{c}, \mathbf{A}, \mathbf{b}),$$

where the function linprog returns the optimal values that minimizes the objective function while respecting the imposed constraints .

3.4 Conclusions

This chapter reviewed some fundamental concepts, definitions and operations about quaternions, dual quaternions and linear programming.

Section 3.1 and 3.2 presented definitions, operations and properties with quaternions and dual quaternions, as well as the representation of rotations and translation using quaternions, and rigid motions using dual quaternions. In addition, some concepts and useful examples were presented in order to perform algebraic manipulations. These concepts are used to obtain the kinematic model of the mobile manipulator in the chapter 4. Furthermore, in the chapter 6 are used to implement the vision system and to perform manipulation task on the real robot. Section 3.3 presented a brief introduction about linear programming and reviewed some basic operations. These concepts are used in 5 to express the linear program in the canonical form.

4

Forward and Differential Kinematic Model

This chapter presents a quick review of kinematic modeling using dual quaternion algebra. The robot used in this work is composed of two subsystems: a five-DOF manipulator and a nonholonomic mobile base. Both systems can be modeled separately and then serially composed into a single model according to the methodology presented in Adorno (2011).

4.1 Kinematic Model of the Holonomic Mobile Base

First, the mobile base is modeled without considering the nonholonomic constraint (which arises only in the differential kinematics) because this constraint will be imposed further as an equality constraint in the linear program (see Section 5.3). The pose of frame \mathcal{F}_1 rigidly attached to the mobile base with respect to the fixed frame \mathcal{F}_0 is given by the dual quaternion

$$\underline{\mathbf{x}}_1^0 = \mathbf{r}_1^0 + \varepsilon \frac{1}{2} \mathbf{p}_{01}^0 \mathbf{r}_1^0, \quad (4.1)$$

where $\mathbf{p}_{01}^0 = x\hat{i} + y\hat{j}$ is the translation from the origin of \mathcal{F}_0 to the origin of \mathcal{F}_1 , expressed in \mathcal{F}_0 , and $\mathbf{r}_1^0 = \cos(\phi/2) + \hat{k} \sin(\phi/2)$ is the rotation angle ϕ around the rotation axis \hat{k} (which corresponds to the z -axis). In order to take into account the base height, frame \mathcal{F}_2 is rigidly attached in the same orientation of \mathcal{F}_1 on top of the base, with a displacement $\mathbf{p}_{12}^1 = z_c \hat{k}$; that is, a constant translation z_c along \hat{k} , which corresponds to the z -axis, expressed in \mathcal{F}_1 . Hence, the pose of frame \mathcal{F}_2 with respect to frame \mathcal{F}_1 is given by the

dual quaternion

$$\underline{\mathbf{x}}_2^1 = 1 + \varepsilon \frac{1}{2} \mathbf{p}_{12}^1, \quad (4.2)$$

This transformation is shown in the side view of Fig. 4.1.

The pose of frame \mathcal{F}_2 respect to the fixed frame \mathcal{F}_0 is given by the dual quaternion

$$\underline{\mathbf{x}}_2^0 = \underline{\mathbf{x}}_1^0 \underline{\mathbf{x}}_2^1.$$

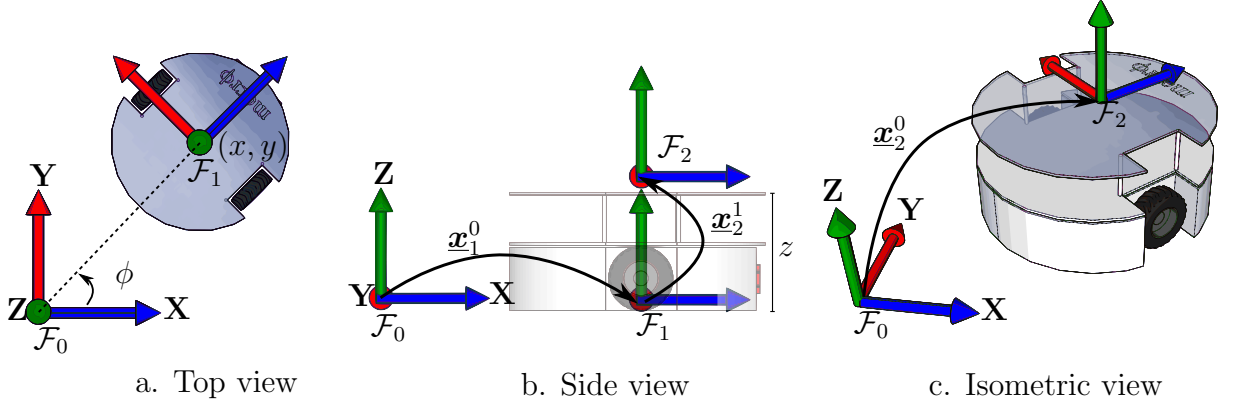


Figure 4.1: Pose of the nonholonomic mobile base with respect to \mathcal{F}_0 .

The first time-derivative of (4.1) provides the differential kinematics of the mobile base without considering the nonholonomic constraints. Using dual quaternion algebra as presented in Adorno (2011), $\dot{\underline{\mathbf{x}}}_1^0$ is expressed as

$$\text{vec } \dot{\underline{\mathbf{x}}}_1^0 = \underbrace{\begin{bmatrix} 0 & 0 & j_{13} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & j_{43} \\ 0 & 0 & 0 \\ j_{61} & j_{62} & j_{63} \\ j_{71} & j_{72} & j_{73} \\ 0 & 0 & 0 \end{bmatrix}}_{\mathbf{J}_b} \underbrace{\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix}}_{\dot{\mathbf{q}}_b} \quad (4.3)$$

where $j_{13} = -j_{62} = j_{71} = -\frac{1}{2} \sin\left(\frac{\phi}{2}\right)$, $j_{43} = j_{61} = j_{72} = \frac{1}{2} \cos\left(\frac{\phi}{2}\right)$, $j_{63} = \frac{1}{4} \left[-x \sin\left(\frac{\phi}{2}\right) + y \cos\left(\frac{\phi}{2}\right)\right]$, $j_{73} = \frac{1}{4} \left[-x \cos\left(\frac{\phi}{2}\right) - y \sin\left(\frac{\phi}{2}\right)\right]$ and \mathbf{J}_b is the analytical Jacobian matrix of the mobile base without considering the nonholonomic constraints.

4.2 Kinematic Model of the Manipulator Arm

The second subsystem is the five-DOF arm, which is serially coupled to the mobile base, as shown in Fig 4.2. The dual quaternion $\underline{\mathbf{x}}_3^2$ represents the pose of the end-effector with respect to frame \mathcal{F}_2 and is a function of the manipulator joints. It is given by (Adorno, 2011)

$$\underline{\mathbf{x}}_3^2 = \underline{\mathbf{f}}(\mathbf{q}_a) = \prod_{i=1}^n \left(\mathbf{r}_{\theta_i} \underline{\mathbf{p}}_{d_i} \underline{\mathbf{p}}_{a_i} \mathbf{r}_{\alpha_i} \right), \quad (4.4)$$

where \mathbf{q}_a is the joints vector, $\theta_i, d_i, a_i, \alpha_i$ are the Denavit-Hartenberg parameters of the i -link, $\mathbf{r}_{\theta_i} = \cos(\theta_i/2) + \hat{k} \sin(\theta_i/2)$ and $\mathbf{r}_{\alpha_i} = \cos(\alpha_i/2) + \hat{i} \sin(\alpha_i/2)$ are pure rotations around the z -axis and x -axis, respectively, and $\underline{\mathbf{p}}_{d_i} = 1 + \varepsilon(1/2)d_i\hat{k}$ and $\underline{\mathbf{p}}_{a_i} = 1 + \varepsilon(1/2)a_i\hat{i}$ are pure translations along the z -axis and x -axis, respectively.

The time derivative of (4.4) provides the differential kinematics of the robot manipulator, given by

$$\text{vec } \dot{\underline{\mathbf{x}}}_3^2 = \mathbf{J}_a \dot{\mathbf{q}}_a, \quad (4.5)$$

where $\mathbf{q}_a = [\theta_1 \ \dots \ \theta_5]^T$, with θ_i being the i -th joint's angle, and \mathbf{J}_a is the analytical Jacobian matrix that is obtained by using dual quaternion algebra (Adorno, 2011).

4.3 Whole-body Kinematic Model

The forward kinematics of the whole-body composed of the mobile base and the manipulator is obtained by computing the dual quaternion $\underline{\mathbf{x}}_3^0$ that represents the pose of the end-effector with respect to the fixed frame \mathcal{F}_0 and is given by

$$\underline{\mathbf{x}}_3^0 = \underline{\mathbf{x}}_1^0 \underline{\mathbf{x}}_2^1 \underline{\mathbf{x}}_3^2. \quad (4.6)$$

The time derivative of (4.6) provides the differential kinematics of the whole system and is given by

$$\dot{\underline{\mathbf{x}}}_3^0 = \dot{\underline{\mathbf{x}}}_1^0 \underline{\mathbf{x}}_2^1 \underline{\mathbf{x}}_3^2 + \underline{\mathbf{x}}_1^0 \dot{\underline{\mathbf{x}}}_2^1 \underline{\mathbf{x}}_3^2, \quad (4.7)$$

where $\dot{\underline{\mathbf{x}}}_2^1 = 0$ as $\underline{\mathbf{x}}_2^1$ represents a constant displacement between frames \mathcal{F}_1 and \mathcal{F}_2 . Using the Hamilton operators, we rewrite (4.7) as

$$\text{vec } \dot{\underline{\mathbf{x}}}_3^0 = \bar{\mathbf{H}}(\underline{\mathbf{x}}_3^1) \text{vec } \dot{\underline{\mathbf{x}}}_1^0 + \mathbf{H}^+(\underline{\mathbf{x}}_2^0) \text{vec } \dot{\underline{\mathbf{x}}}_3^2. \quad (4.8)$$

Substituting (4.3) and (4.5) in (4.8), the whole-body differential kinematics is given by (Adorno, 2011)

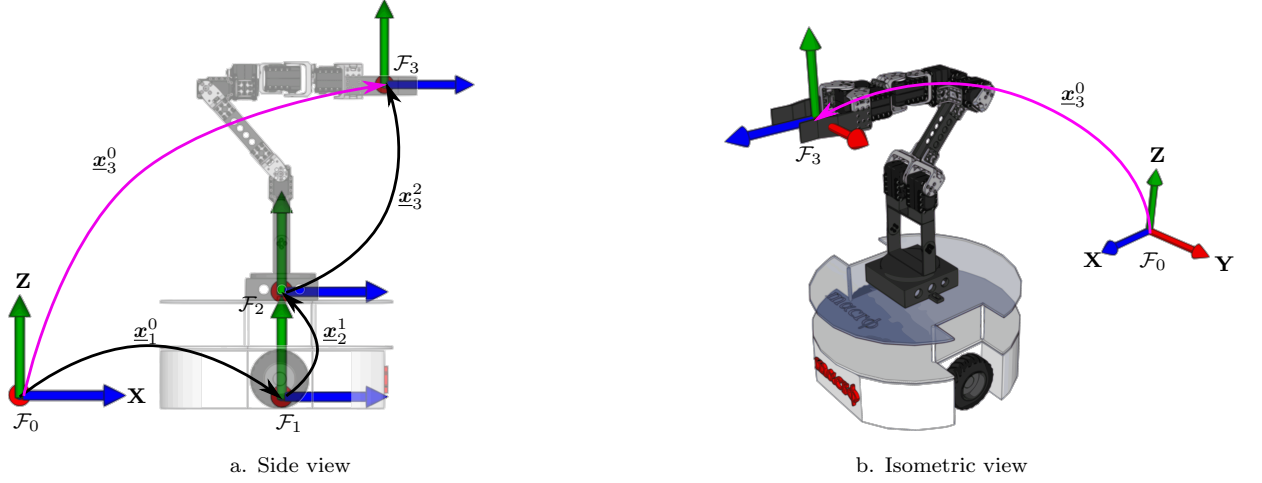


Figure 4.2: Nonholonomic mobile manipulator.

$$\text{vec } \dot{\mathbf{x}}_3^0 = \underbrace{\begin{bmatrix} \bar{\mathbf{H}}(\mathbf{x}_3^1) \mathbf{J}_b & \bar{\mathbf{H}}(\mathbf{x}_2^0) \mathbf{J}_a \end{bmatrix}}_{\mathbf{J}} \underbrace{\begin{bmatrix} \dot{\mathbf{q}}_b \\ \dot{\mathbf{q}}_a \end{bmatrix}}_{\dot{\mathbf{q}}}, \quad (4.9)$$

where the stacked vector $\mathbf{q} = [\mathbf{q}_b^T \ \mathbf{q}_a^T]^T$ corresponds to the whole-body configuration and \mathbf{J} is the whole-body Jacobian matrix and provides the relation between the joints velocities and the generalized velocities of the end-effector (Adorno, 2011).

4.4 Position Jacobian Matrix

In some cases it is desired to control only the end-effector position and, in those cases it is necessary to compute the position Jacobian matrix \mathbf{J}_p that provides the relation between the joints velocities and the linear velocities of the end-effector as follows

$$\text{vec}_4 \dot{\mathbf{p}}_{03}^0 = \mathbf{J}_p \dot{\mathbf{q}}. \quad (4.10)$$

Now, Let \mathbf{x}_3^0 be expressed without loss of generality as

$$\mathbf{x}_3^0 = \mathbf{r}_3^0 + \varepsilon \frac{1}{2} \mathbf{p}_{03}^0 \mathbf{r}_3^0. \quad (4.11)$$

where \mathbf{r}_3^0 and \mathbf{p}_{03}^0 are the rotation and the translation quaternion related to \mathbf{x}_3^0 . The translation \mathbf{p}_{03}^0 is computed from (4.11) as (Adorno et al., 2010)

$$\mathbf{p}_{03}^0 = 2 \mathcal{D}(\mathbf{x}_3^0) \mathcal{P}(\mathbf{x}_3^{0*}). \quad (4.12)$$

The time derivative of equation (4.12) is given by

$$\dot{\mathbf{p}}_{03}^0 = 2 \mathcal{D}(\dot{\mathbf{x}}_3^0) \mathcal{P}(\mathbf{x}_3^{0*}) + 2 \mathcal{D}(\mathbf{x}_3^0) \mathcal{P}(\dot{\mathbf{x}}_3^{0*}), \quad (4.13)$$

Using the Hamilton operators (4.13) is rewritten as

$$\text{vec}_4 \dot{\mathbf{p}}_{03}^0 = 2\bar{\mathbf{H}}_4 \left(\mathcal{P} \left(\mathbf{x}_3^{0*} \right) \right) \text{vec}_4 \mathcal{D} \left(\dot{\mathbf{x}}_3^0 \right) + 2\mathbf{H}_4^+ \left(\mathcal{D} \left(\mathbf{x}_3^0 \right) \right) \text{vec}_4 \mathcal{P} \left(\dot{\mathbf{x}}_3^{0*} \right). \quad (4.14)$$

The whole-body Jacobian matrix \mathbf{J} computed in (4.9) can be decomposed as (Adorno et al., 2010)

$$\mathbf{J} \triangleq \begin{bmatrix} \mathbf{J}_{\mathcal{P}(\mathbf{x}_3^0)} \\ \mathbf{J}_{\mathcal{D}(\mathbf{x}_3^0)} \end{bmatrix}, \quad (4.15)$$

such that

$$\text{vec}_4 \mathcal{P} \left(\dot{\mathbf{x}}_3^0 \right) = \mathbf{J}_{\mathcal{P}(\mathbf{x}_3^0)} \dot{\mathbf{q}}, \quad (4.16)$$

$$\text{vec}_4 \mathcal{D} \left(\dot{\mathbf{x}}_3^0 \right) = \mathbf{J}_{\mathcal{D}(\mathbf{x}_3^0)} \dot{\mathbf{q}}. \quad (4.17)$$

Substituting (4.16) and (4.17) in (4.14) the position whole-body differential kinematic model is given by

$$\text{vec}_4 \dot{\mathbf{p}}_{03}^0 = \underbrace{\left[2\bar{\mathbf{H}}_4 \left(\mathcal{P} \left(\mathbf{x}_3^{0*} \right) \right) \mathbf{J}_{\mathcal{D}(\mathbf{x}_3^0)} \quad 2\mathbf{H}_4^+ \left(\mathcal{D} \left(\mathbf{x}_3^0 \right) \right) \mathbf{C}_4 \mathbf{J}_{\mathcal{P}(\mathbf{x}_3^0)} \right]}_{\mathbf{J}_p} \dot{\mathbf{q}}. \quad (4.18)$$

4.5 Chapter Conclusions

This chapter presented the forward differential kinematic model for a nonholonomic mobile manipulator based on a systematic procedure using dual quaternions. The robot is modeled separately and then a whole-body kinematic control is computed. Section 4.1 presented the kinematic model of the differential base, which is modeled as holonomic mobile base because the nonholonomic constraint is take into account in the linear programming formulation as equality constraint. Section 4.2 presented the kinematic model of the arm manipulator. Section 4.3 presented the whole-body kinematic model of the nonholonomic mobile manipulator. Finally, Section 4.4 reviewed the position Jacobian matrix computed from the whole-body Jacobian matrix. The whole-body kinematic models are used in the chapter 5 in the inverse kinematic formulation as linear programming.

5

Whole-Body Kinematic Control

This chapter discusses the kinematic control strategy applied to the nonholonomic mobile manipulator presented in chapter 4. The chapter is organized as follows: Section 5.1 presents the kinematic control strategy as an optimization problem and its formulation by using linear programming; in Section 5.2 the improvements to the original linear programming formulation proposed by Goncalves et al. (2016) are explained; in Section 5.3 the nonholonomic constraint of the differential mobile base is imposed as an equality constraint in the formulation of the linear program; Section 5.4 presents the constraints that are imposed in order to prevent violation of the joint limits; Section 5.5 presents a method to prevent collisions in the plane by adding an additional equality constraint in the linear program; Section 5.6 presents the final formulation of the linear program in order to perform control of the end-effector pose while respecting all the aforementioned constraints. Section 5.7 discusses the parsimonious behavior presented when the solver used in the linear program is the Simplex method.

5.1 Kinematic Control using Linear Programming

In order to control the end-effector pose by using whole-body motions (i.e., motions that can potentially—and automatically—use all available DOF), the kinematic control strategy traditionally has been approached by using the pseudoinverse of the Jacobian matrix. This solution to the differential inverse kinematics problem often consists in solving an optimization problem, in most cases quadratic ones, when there are no imposed constraints.

More specifically, let $\underline{\mathbf{e}} = \underline{\mathbf{x}} - \underline{\mathbf{x}}_d$ be the error between the measured ($\underline{\mathbf{x}}$) and desired ($\underline{\mathbf{x}}_d$) dual quaternions that represent the current and desired end-effector poses, respectively. If the desired pose is constant, the error dynamics is given by

$$\text{vec}_8 \dot{\underline{\mathbf{e}}} = \text{vec}_8 \dot{\underline{\mathbf{x}}} = \mathbf{J}\dot{\underline{\mathbf{q}}}, \quad (5.1)$$

and a desirable control input $\mathbf{u} \triangleq \dot{\underline{\mathbf{q}}}$ would be one that enforces an exponential convergence; that is, a control input such that the closed loop dynamics is $\text{vec}_8 \dot{\underline{\mathbf{e}}} = -\eta \text{vec}_8 \underline{\mathbf{e}}$, with $\eta > 0$.

The generation of such control signal can be written as the optimization problem (Goncalves et al., 2016)

$$\min_{\mathbf{u}} \|\mathbf{J}\mathbf{u} + \eta \text{vec}_8 \underline{\mathbf{e}}\|_2, \quad (5.2)$$

whose analytical minimal norm solution is given by

$$\mathbf{u} = -\mathbf{J}^\dagger \eta \text{vec}_8 (\underline{\mathbf{x}} - \underline{\mathbf{x}}_d) \quad (5.3)$$

in absence of equality and inequality constraints. In general, all entries in the control input $\mathbf{u} = \dot{\underline{\mathbf{q}}}$ are different from zero and hence all actuated DOF are used. An alternative approach for generating whole-body motions is the kinematic control based on linear programming recently proposed by Goncalves et al. (2016) for redundant systems. This approach is computationally efficient and the stability was formally proved by the authors. In this formulation, all actuated DOF are available to perform the whole-body motion, and when the Simplex method (Murty, 1983) is used only the minimum amount of DOF strictly needed to perform the task is used at each instant. In order to achieve this behavior, the solution to the differential inverse kinematics problem consists in solving an optimization problem as (5.2) where a convex positive definite error metric is used, namely the 1-norm, and the problem can be written as follows

$$\min_{\mathbf{u}} \|\mathbf{J}\mathbf{u} + \eta \text{vec}_8 \underline{\mathbf{e}}\|_1. \quad (5.4)$$

This formulation can be transformed into a linear program, which can be solved with a numerical solver. A computationally efficient formulation (Li, 1998, apud Gonçalves et al., 2016) is given by

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{y}} \quad & \mathbf{1}^T (2\mathbf{y} - (\mathbf{J}\mathbf{u} + \eta \text{vec}_8 \underline{\mathbf{e}})) \\ \text{subject to} \quad & \mathbf{J}\mathbf{u} - \mathbf{y} \leq -\eta \text{vec}_8 \underline{\mathbf{e}} \\ & \mathbf{W}\mathbf{u} \leq \mathbf{w} \\ & \mathbf{y} \geq \mathbf{0}, \end{aligned} \quad (5.5)$$

where $\mathbf{1}^T$ is a vector of ones of appropriate dimension and $\mathbf{W}\mathbf{u} \leq \mathbf{w}$ represents constraints that may be imposed for \mathbf{u} . In order to use a linear programming numerical solver, the

optimization problem (5.5) must be rewritten into the canonical form of a linear program, which is given by (3.19).

In order to do this, two changes should be made (Goncalves et al., 2016). Since the elements of the vector \mathbf{g} in 3.19 must be always nonnegative, the vector \mathbf{u} must be decomposed into two non-negative variables $\mathbf{u}_P - \mathbf{u}_N$, see remark 3.2, in chapter 3. In addition, as the solution to the linear program does not ensure that $\mathbf{u} = \mathbf{0}$ when $\text{vec}_8 \underline{\mathbf{e}} = \mathbf{0}$, Goncalves et al. (2016) imposed an additional constraint

$$\|\mathbf{u}\|_1 = \|\mathbf{u}_P - \mathbf{u}_N\|_1 \leq \beta \|\text{vec}_8 \underline{\mathbf{e}}\|_1 \quad (5.6)$$

where $\beta > 0$. Therefore, when $\text{vec}_8 \underline{\mathbf{e}} = \mathbf{0}$ necessarily $\mathbf{u} = \mathbf{0}$.

From the triangle inequality,

$$\|\mathbf{u}_P - \mathbf{u}_N\|_1 \leq \|\mathbf{u}_P\|_1 + \|\mathbf{u}_N\|_1 = \mathbf{1}^T \mathbf{u}_P + \mathbf{1}^T \mathbf{u}_N,$$

where the last equality holds because both \mathbf{u}_P and \mathbf{u}_N are nonnegative; thus, the constraint

$$\mathbf{1}^T \mathbf{u}_P + \mathbf{1}^T \mathbf{u}_N \leq \beta \|\text{vec}_8 \underline{\mathbf{e}}\|_1 \quad (5.7)$$

enforces (5.6). It is important to note that (5.7) guarantees that $\mathbf{u} = \mathbf{0}$ if and only if $\text{vec}_8 \underline{\mathbf{e}} = \mathbf{0}$, hence if the end-effector stabilizes far from the desired set point the robot configuration can still change. The solution to this problem is shown in Section 5.2.

Adding the slack variables z_A, z_B, z_C and the constraint (5.7), the linear program (5.5) rewritten in the canonical form (3.19) is given by (Goncalves et al., 2016)

$$\begin{aligned} \min_{\mathbf{g}} \quad & \left[-\mathbf{1}^T \mathbf{J} \quad \mathbf{1}^T \mathbf{J} \quad 2 \cdot \mathbf{1}^T \quad \mathbf{0}^T \quad \mathbf{0}^T \quad 0 \right] \mathbf{g} - \eta \mathbf{1}^T \text{vec } \underline{\mathbf{e}} \\ \text{subject to} \quad & \begin{bmatrix} \mathbf{J} & -\mathbf{J} & -\mathbf{I} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{W} & -\mathbf{W} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{1}^T & \mathbf{1}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & 1 \end{bmatrix} \mathbf{g} = \begin{bmatrix} -\eta \text{vec } \underline{\mathbf{e}} \\ \mathbf{w} \\ \beta \|\text{vec } \underline{\mathbf{e}}\|_1 \end{bmatrix} \\ & \mathbf{g} \geq \mathbf{0}, \end{aligned} \quad (5.8)$$

where $\mathbf{g} = \left[\mathbf{u}_P^T \quad \mathbf{u}_N^T \quad \mathbf{y}^T \quad z_A^T \quad z_B^T \quad z_C^T \right]^T$.

5.2 Improvements to the whole-body control based on linear programming

This section presents improvements to the original whole-body control based on linear programming introduced by Goncalves et al. (2016). First, a new constraint is proposed in order to avoid joints movements when the robot does not stabilizes asymptotically.

In addition, more constraints are imposed in order to avoid violation of joint limits and collision with obstacles in the plane.

The constraint (5.7) ensures that $\mathbf{u} = \mathbf{0}$ when the robot end-effector stabilizes asymptotically. However, because only Lyapunov stability is guaranteed (Goncalves et al., 2016) for control inputs generated by (5.8), it is possible that the robot end-effector stabilizes at some pose different from the desired one (that is, $\text{vec}_8 \underline{\mathbf{e}} > \mathbf{0}$ but $\text{vec}_8 \underline{\dot{\mathbf{e}}} = \mathbf{0}$) and, consequently, it may be the case that $\mathbf{u} \neq \mathbf{0}$ when the robot end-effector stabilizes. In order to prevent this problem, we propose a definite positive function that depends on the error time derivative. This way, (5.6) is modified as follows:

$$\|\mathbf{u}\|_1 = \|\mathbf{u}_P - \mathbf{u}_N\|_1 \leq \beta \|\text{vec}_8 \underline{\dot{\mathbf{e}}}\|_1. \quad (5.9)$$

Note that when the robot stabilizes at some pose different from the desired pose, $\text{vec}_8 \underline{\dot{\mathbf{e}}} = \mathbf{0}$, which implies $\mathbf{u} = \mathbf{0}$. Both functions are shown in Fig. 5.1.

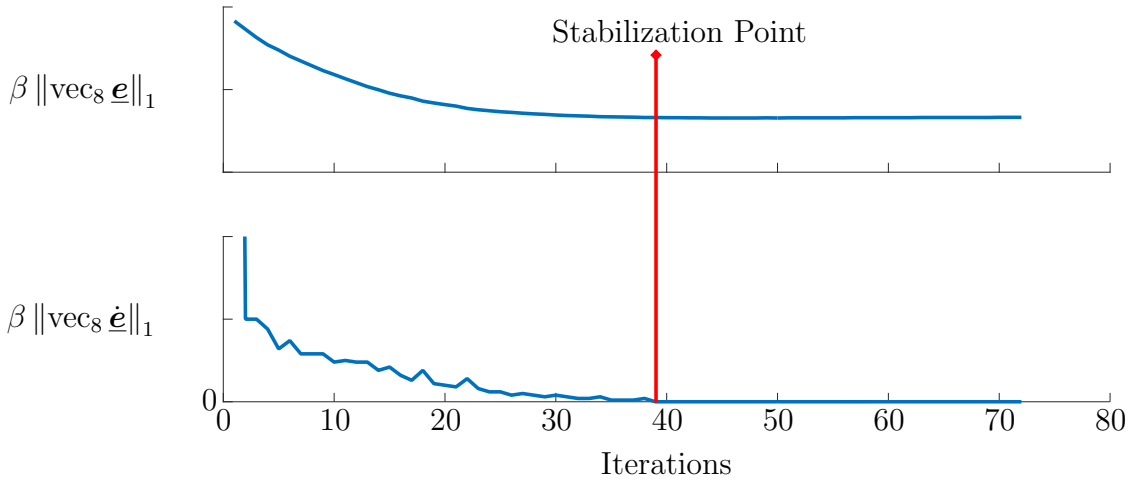


Figure 5.1: Two definite positive functions. The *top* figure shows a function that depends on the error. The *bottom* figure shows a function that depends on the error time-derivative.

From the triangle inequality,

$$\|\mathbf{u}_P - \mathbf{u}_N\|_1 \leq \|\mathbf{u}_P\|_1 + \|\mathbf{u}_N\|_1 = \mathbf{1}^T \mathbf{u}_P + \mathbf{1}^T \mathbf{u}_N,$$

where the last equality holds because both \mathbf{u}_P and \mathbf{u}_N are nonnegative; thus, the constraint

$$\mathbf{1}^T \mathbf{u}_P + \mathbf{1}^T \mathbf{u}_N \leq \beta \|\text{vec}_8 \underline{\dot{\mathbf{e}}}\|_1 \quad (5.10)$$

enforces (5.9).

5.3 Nonholonomic constraint of the mobile base

In Quiroz-Omana and Adorno (2016) we used a cascade scheme to control a nonholonomic mobile manipulator, where the outer loop was responsible to generate the whole-body motion and the inner loop dealt with the nonholonomic constraint of the mobile base by using an input-output linearizing controller. Although Goncalves et al. (2016) proved that the stability of any system described by first order differential kinematics (i.e., $\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}$) is guaranteed when the control inputs are generated by (5.8), such stability was not formally proved for the cascade scheme used in (Quiroz-Omana and Adorno, 2016). However, the nonholonomic constraint of the mobile base can be imposed into the linear program as an equality constraint, therefore a cascade control scheme is not required. The nonholonomic constraint of the differential drive mobile base is given by (Siciliano et al., 2009)

$$\dot{x} \sin \phi - \dot{y} \cos \phi = 0, \quad (5.11)$$

where ϕ is the rotation angle around the z -axis, \dot{x} and \dot{y} are the velocities of the mobile base in x and y directions of the inertial frame, respectively. Eq. (5.11) can be rewritten as

$$\mathbf{W}_{nh} \mathbf{u}_P - \mathbf{W}_{nh} \mathbf{u}_N = 0, \quad (5.12)$$

where $\mathbf{W}_{nh} = \begin{bmatrix} \sin \phi & -\cos \phi & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$.

5.4 Avoidance of joint limits

In real robotic systems, joints movements are mechanically limited and hence the controller should generate control inputs considering these physical constraints, otherwise a saturation could happen, resulting in large tracking errors or even damage to the robot. In order to prevent this, additional constraints must be imposed.

Let \mathbf{q}^- and \mathbf{q}^+ be the vector of lower and upper joints limits, respectively. The constraints related to both joints limits can be written as follows

$$\mathbf{q}^- \leq \mathbf{q} \leq \mathbf{q}^+. \quad (5.13)$$

However, the kinematic control formulation as linear programming is solved at the velocity level, hence the joints range $[\mathbf{q}^-, \mathbf{q}^+]$ has to be converted to bounds in the joints velocities. A simple way to do that is presented by Fan-Tien Cheng et al. (1994) and is given by

$$k(\beta_l \mathbf{q}^- - \mathbf{q}) \leq \mathbf{u} \leq k(\beta_l \mathbf{q}^+ - \mathbf{q}), \quad (5.14)$$

where $0 \ll \beta_l \leq 1$ is selected to define a security margin for the joints limits (see Fig. 5.2),

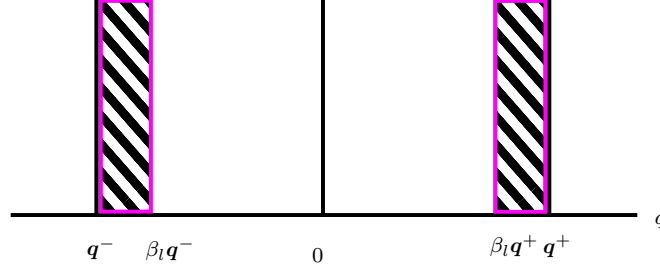


Figure 5.2: The margin considered for the joint limits $q^- \leq q \leq q^+$ is determined by the security factor β_l .

whereas $k > 0$ is used to scale the feasible region of \dot{q} .

When constraints in joints velocities are also imposed, (5.14) can be rewritten as

$$\max \{ \dot{q}^-, k(\beta_l q^- - q) \} \leq \mathbf{u} \leq \min \{ \dot{q}^+, k(\beta_l q^+ - q) \}, \quad (5.15)$$

where \dot{q}^- and \dot{q}^+ are the lower and upper bounds for the joints velocities, respectively. In order to impose the constraints (5.15), two inequality constraints must be written, one for each limit. Let $\boldsymbol{\eta}^- = \max \{ \dot{q}^-, k(\beta_l q^- - q) \}$ and $\boldsymbol{\eta}^+ = \min \{ \dot{q}^+, k(\beta_l q^+ - q) \}$, then $\boldsymbol{\eta}^- \leq \mathbf{u} \leq \boldsymbol{\eta}^+$ can be rewritten as

$$-\mathbf{u} \leq -\boldsymbol{\eta}^-, \quad (5.16)$$

$$\mathbf{u} \leq \boldsymbol{\eta}^+. \quad (5.17)$$

The inequality constraints (5.16) and (5.17) are rewritten in matrix form as

$$W_l \mathbf{u}_P - W_l \mathbf{u}_N \leq \mathbf{w}_l, \quad (5.18)$$

where $\mathbf{W}_l = [-\mathbf{I}_8 \quad \mathbf{I}_8]^T$, with \mathbf{I}_n denoting the identity matrix of size n , and $\mathbf{w}_l = [(-\boldsymbol{\eta}^-)^T \quad (\boldsymbol{\eta}^+)^T]^T$. Note that (5.18) is defined to prevent the violation of the manipulator joints limits and also to impose a limit to the velocity of the mobile base.

5.5 Collision avoidance of the mobile base

One of the main objectives in robotics is to autonomously execute tasks in the real world (Latombe, 1991) while avoiding collisions with obstacles in the workspace. This fundamental robotics task is usually solved by using motion planning algorithms, where the goal is to generate collision-free paths from a start configuration to a final configuration. However, sometimes the nominal plan can fail due to uncertainties or other unforeseen obstacles, thus an additional low-level protection may be included as a constraint in the linear program in order to prevent collisions and provide a reactive behavior for the robot.

In order to define the constraint for the collision avoidance, we model obstacles as circles in the 2D plane and we consider only collisions with the mobile base (the manipulator can also be considered in a conservative way by enlarging the obstacles' radiuses). Let $\mathbf{q}_{xy} = [x \ y]^T$ be the mobile base configuration and $\mathbf{s} = (\mathbf{c}, R_{obs})$ represent the circular obstacle with center $\mathbf{c} = [x_c \ y_c]^T$ and a security extended radius R_{obs} . In order to prevent collisions in the plane a constraint can be imposed to limit the velocities of the mobile base as it approaches to the obstacle as follows

$$-d \leq \dot{\mathbf{q}}_b \leq d, \quad (5.19)$$

where d is a defined distance between the base and the obstacle and $\dot{\mathbf{q}}_b$ denotes the velocities of the mobile base. Note that constraint (5.19) is a conservative solution because when the robot moves toward the obstacle its speed limit decreases until zero. Although this constraint prevent collision, the robot is not able to perform movements around the obstacle. On the other hand, the constraint (5.19) is simple and its implementation is easy.

In order to define a less conservative constraint to prevent collisions in the plane, the next inequality is imposed to the mobile base:

$$R \leq \|\mathbf{q}_{xy} - \mathbf{c}\|, \quad (5.20)$$

where $\|\mathbf{q}_{xy} - \mathbf{c}\|$ is the distance between the robot and the obstacle center (for convenience's sake, the robot radius is taken into consideration when defining the obstacle's radius R , that is $R = R_{obs} + R_{robot}$).

We define

$$f(\mathbf{q}_{xy}, \mathbf{s}) \triangleq -\|\mathbf{q}_{xy} - \mathbf{c}\|^2 + R^2, \quad (5.21)$$

such that $f(\mathbf{q}_{xy}, \mathbf{s}) \leq 0$ guarantees that the robot will not collide with the obstacle, as shown in Fig 5.3. Thus,

$$f(\mathbf{q}_{xy}, \mathbf{s}) + \alpha(t) = 0, \quad (5.22)$$

where $\alpha(t)$ is a slack variable.

The time derivative of (5.22) provides its relation at the velocity level and is given by

$$\underbrace{\frac{\partial f(\mathbf{q}_{xy}, \mathbf{s})}{\partial \mathbf{q}_{xy}}}_{\mathbf{J}_c} \dot{\mathbf{q}}_{xy} + \dot{\alpha}(t) = 0, \quad (5.23)$$

where we assumed that the object is stationary and

$$\mathbf{J}_c = \begin{bmatrix} -2(x - x_c) & -2(y - y_c) \end{bmatrix}.$$

Since the constraint (5.23) must be imposed only when necessary—that is, when the

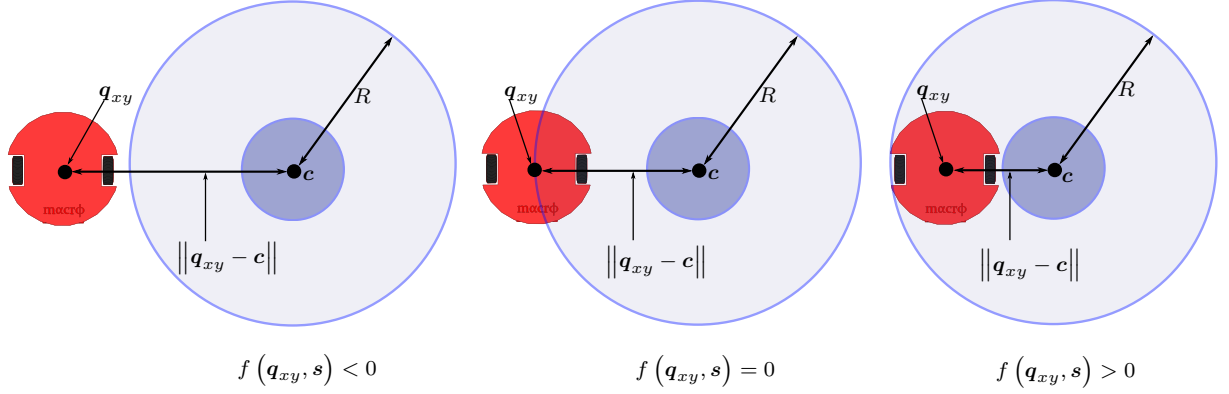


Figure 5.3: Behavior of the function $f(\mathbf{q}_{xy}, \mathbf{s})$. In the *left*, the robot is outside the collision region and $f(\mathbf{q}_{xy}, \mathbf{s}) < 0$; in this case there is no risk of collision between the robot and the obstacle in the plane. In the *center*, the robot is on the border of the collision region and $f(\mathbf{q}_{xy}, \mathbf{s}) = 0$; also in this case, there is no also risk of collision. Finally, in the *right*, the robot is inside the collision region and $f(\mathbf{q}_{xy}, \mathbf{s}) > 0$; in this case, there is a high risk of collision.

robot is inside the forbidden region—we include the variable β as follows:

$$\beta \mathbf{J}_c \dot{\mathbf{q}}_{xy} = -\beta \dot{\alpha}(t), \quad (5.24)$$

where

$$\beta = \begin{cases} 0, & f(\mathbf{q}_{xy}, \mathbf{s}) \leq 0 \\ 1, & f(\mathbf{q}_{xy}, \mathbf{s}) > 0 \text{ and } h(\mathbf{q}_{xy}, \mathbf{q}_d, \mathbf{s}) > 0, \end{cases} \quad (5.25)$$

with $h(\mathbf{q}_{xy}, \mathbf{q}_d, \mathbf{s}) = \|\mathbf{q}_d - \mathbf{q}_{xy}\| - \|\mathbf{q}_d - \mathbf{c}\|$, and \mathbf{q}_d is the projection of the end-effector position onto the xy -plane of the inertial frame \mathcal{F}_0 , which provides a rough approximation of where the center of the mobile base should be, as the manipulator is relatively small. The geometrical interpretation of this function is shown in Fig. 5.4.

When the constraint (5.24) is enabled (i.e., $\beta = 1$), we desire that the robot moves towards or along the border, moving away from the center of the obstacle. This behavior can be accomplished by defining a suitable dynamics for $\alpha(t)$. Since $\alpha(t) = 0$ implies that \mathbf{q}_{xy} moves along the border of the circle around the obstacle, we enforce the dynamics

$$\dot{\alpha}(t) = -\lambda \alpha(t). \quad (5.26)$$

Substituting (5.26) into (5.24) we obtain

$$\beta \mathbf{J}_c \dot{\mathbf{q}}_{xy} = \beta \lambda \alpha(t), \quad (5.27)$$

where $\alpha(t) = -f(\mathbf{q}_{xy}, \mathbf{s})$.

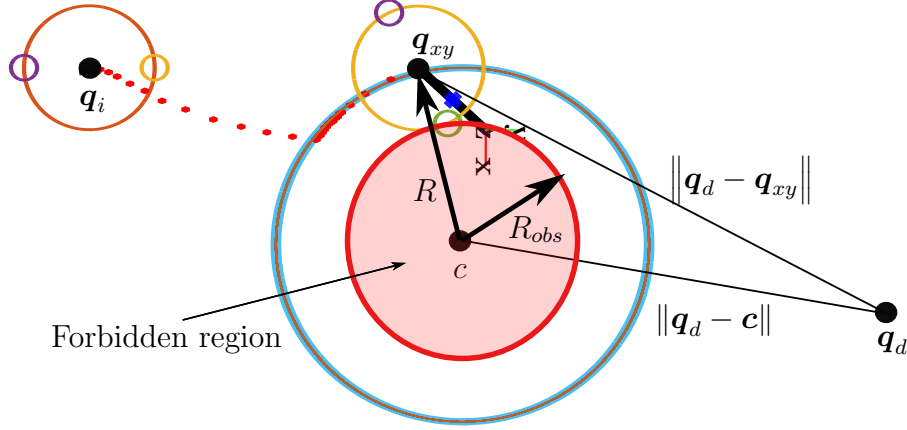


Figure 5.4: Start and final configurations of the mobile base are denoted by \mathbf{q}_i and \mathbf{q}_d , respectively. The configuration \mathbf{q}_d is the projection of the end-effector position onto the xy -plane of the inertial frame \mathcal{F}_0 , which provides a rough approximation of where the center of the mobile base should be, as the manipulator is relatively small.

The constraint (5.27) can be rewritten as

$$\mathbf{J}_{obs}\mathbf{u}_P - \mathbf{J}_{obs}\mathbf{u}_N = \beta\lambda\alpha(t), \quad (5.28)$$

where $\mathbf{J}_{obs} = \begin{bmatrix} \beta\mathbf{J}_c & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$.

5.6 Final formulation considering all constraints

Considering the new equality constraints (5.12) and (5.28), and the new inequality constraints (5.10) and (5.18), we add the corresponding slack variables \mathbf{z}_A , \mathbf{z}_B and \mathbf{z}_C to the vector \mathbf{g} to obtain the linear program in the final (canonical) form:

$$\begin{aligned} \min_{\mathbf{g}} \quad & \left[-\mathbf{1}_{(1 \times 8)} \mathbf{J} \quad \mathbf{1}_{(1 \times 8)} \mathbf{J} \quad 2 \cdot \mathbf{1}_{(1 \times 8)} \quad \mathbf{0}_{(1 \times 8)} \quad \mathbf{0}_{(1 \times 2n)} \quad 0 \right] \mathbf{g} - \eta \mathbf{1}^T \text{vec } \underline{\mathbf{e}} \\ \text{subject to} \quad & \begin{bmatrix} \mathbf{J}_{obs} & -\mathbf{J}_{obs} & \mathbf{0}_{(1 \times 8)} & \mathbf{0}_{(1 \times 8)} & \mathbf{0}_{(1 \times 2n)} & 0 \\ \mathbf{W}_{nh} & -\mathbf{W}_{nh} & \mathbf{0}_{(1 \times 8)} & \mathbf{0}_{(1 \times 8)} & \mathbf{0}_{(1 \times 2n)} & 0 \\ \mathbf{J} & -\mathbf{J} & -\mathbf{I}_8 & \mathbf{I}_8 & \mathbf{0}_{(8 \times 2n)} & \mathbf{0}_{(8 \times 1)} \\ \mathbf{W}_l & -\mathbf{W}_l & \mathbf{0}_{(2n \times 8)} & \mathbf{0}_{(2n \times 8)} & \mathbf{I}_{(2n \times 2n)} & \mathbf{0}_{(2n \times 1)} \\ \mathbf{1}_{(1 \times n)} & \mathbf{1}_{(1 \times n)} & \mathbf{0}_{(1 \times 8)} & \mathbf{0}_{(1 \times 8)} & \mathbf{0}_{(1 \times 2n)} & 1 \end{bmatrix} \mathbf{g} = \begin{bmatrix} \beta\lambda(\alpha(t)) \\ 0 \\ -\eta \text{vec } \underline{\mathbf{e}} \\ \mathbf{w}_l \\ \beta \|\text{vec } \underline{\dot{\mathbf{e}}}\|_1 \end{bmatrix} \\ & \mathbf{g} \geq \mathbf{0}, \end{aligned} \quad (5.29)$$

where $\mathbf{g} = [\mathbf{u}_P^T \quad \mathbf{u}_N^T \quad \mathbf{y}^T \quad \mathbf{z}_A^T \quad \mathbf{z}_B^T \quad \mathbf{z}_C^T]^T$. Furthermore, $\mathbf{J} \in \mathbb{R}^{8 \times n}$, $\mathbf{J}_{obs}, \mathbf{W}_{nh} \in \mathbb{R}^{1 \times n}$, and $\mathbf{W}_l \in \mathbb{R}^{2n \times n}$.

5.7 Parsimonious behavior

A special feature can be obtained when the Simplex method is used: parsimonious solutions (i.e., solutions that use the minimum number of joints). To see why, consider the canonical form (3.19) of linear programs where $\mathbf{g} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{A} \in \mathbb{R}^{m \times n}$, with $n > m$. A set of m linearly independent columns is selected from n columns of \mathbf{A} , resulting in the matrix $\mathbf{B} \in \mathbb{R}^{m \times m}$. Furthermore, the vector \mathbf{g} is split in \mathbf{g}_B and \mathbf{g}_{NB} with m basic variables and $n - m$ nonbasic variables, respectively.

Since the matrix \mathbf{B} forms a basis and is nonsingular, the solution of the equation $\mathbf{B}\mathbf{g}_B = \mathbf{b}$ exists. Then the first m components of \mathbf{g} are \mathbf{g}_B and the remaining components are zero. Thus, when feasible, the Simplex algorithm provides parsimonious solutions (Luenberger and Ye, 2016). It is important to highlight that control inputs generated by (5.29) are Lyapunov stable (Goncalves et al., 2016) and at most

$$U_{NZ} = T_{DOF} + 1 \quad (5.30)$$

entries of \mathbf{u} will be nonzero at a given instant of time, where T_{DOF} is the number of DOF needed to execute the task. It is important to note that the additional nonzero entry is due to constraint (5.10).

5.8 Stability Consideration

It is important to note that a stability condition of the linear programming formulation is that the solution $\mathbf{u} = \mathbf{0}$ belong to the set of feasible solutions. In that sense, all constraints proposed in this thesis do not violate this premise of stability, except the constraint (5.27). This is due to the fact that $\dot{\mathbf{q}}_{xy} = \mathbf{0}$ if and only if $\alpha = 0$, which never happens when the constraint is activated, thus the solution $\mathbf{u} = \mathbf{0}$ does not belong to the set of feasible solutions. Because of this, the constraint (5.27) could destabilize the system. In that sense, the constraint (5.19) is better than (5.27), in terms of stability guarantees, despite being more conservative.

5.9 Chapter Conclusions

This chapter presented the kinematic control strategy applied to the nonholonomic mobile manipulator.

Section 5.1 presented the original formulation of the kinematic control using linear programming presented by Goncalves et al. (2016). Section 5.2 discussed the original formulation and presented improvements in the kinematic control. Section 5.3 discussed the nonholonomic constraint and its formulation as equality constraint in the linear program.

Section 5.4 presented the constraints imposed in order to prevent violation of the joint limits. Section 5.5 presented two constraints used to prevent collisions between the mobile base and obstacles in the plane. The first solution proposed is easy to implement but it is conservative in the sense that it prevents collisions by stopping the robot when it approaches an obstacle. A second constraint is proposed, which could allow that the robot borders the obstacle, when necessary. Section 5.6 presented the final formulation of the linear program considering all constraints. Section 5.7 discussed a special feature when the Simplex algorithm is used to solve the linear program, the parsimonious behavior.

6

Experiments and Results

The whole-body kinematic control strategy presented in this thesis was implemented both in simulation and on a real mobile manipulator, and this chapter presents and discusses the results. The strategy based on linear programming is compared with the traditional strategy, which is based on the pseudoinverse of the Jacobian matrix and a cascade control scheme in order to deal with the nonholonomic constraint, see Appendix B. This chapter is organized as follows: Section 6.1 shows some aspects of the implementation as the robotic platform and the vision system used. Section 6.2 presents three experiments performed.

6.1 Implementation Details

The methods proposed in this work were validated in a nonholonomic mobile manipulator, using the following specifications

- ROS Indigo running on Ubuntu 14.04 64 bits
- A computer equipped with Intel Xeon 2.4GHz and 12Mb of cache memory with 12GB RAM
- A Microsoft Kinect Sensor
- A mobile manipulator composed of a 5-DOF AX18 arm manipulator serially coupled to a differential iRoomba base using the Dynamixel¹ and Turtlebot² drivers,

¹http://wiki.ros.org/dynamixel_motor

²<http://wiki.ros.org/Robots/TurtleBot>

respectively.

- The linear program was implemented by using the CPLEX studio optimization software package³.
- The open-source library DQ Robotics⁴.

6.1.1 Vision System

A Microsoft Kinect sensor was placed at the ceiling and used in order to improve the localization of the mobile base, to detect the desired end-effector pose, and to detect obstacles in the workspace by using the `ar_track_alvar` package⁵ on a frequency of 30Hz (see Fig. 6.1). The position and orientation of each marker is published on topics. In order to minimize the measurements noise, the average pose of each markers is computed with respect to the Kinect sensor frame \mathcal{F}_c . This average is computed in the domain of unit dual quaternions as follows (Adorno, 2012)

$$\bar{\mathbf{x}}_k = \bar{\mathbf{x}}_{k-1} \left(\bar{\mathbf{x}}_{k-1}^* \mathbf{x}_k \right)^{\{1/k\}}, \quad (6.1)$$

where $\bar{\mathbf{x}}_k$ and $\bar{\mathbf{x}}_{k-1}$ denotes the current pose and the previous pose and k is the number of samples and is selected experimentally in order to obtain a smooth and fast convergence. A high value of k provides low noise level in measurements but slower response. A high value of k is used in static markers as the reference frame \mathcal{F}_0 , and obstacles markers denoted by \mathcal{F}_2 and \mathcal{F}_3 . On the other hand, a low number of samples is used to filter the pose of \mathcal{F}_1 , which represents the mobile base pose.

Note that 6.1 acts as a low pass filter providing a smooth reading of the markers.

³<http://www-01.ibm.com/software/websphere/products/optimization/cplex-studio-community-edition/>

⁴<http://dqrobotics.sourceforge.net/>

⁵http://wiki.ros.org/ar_track_alvar

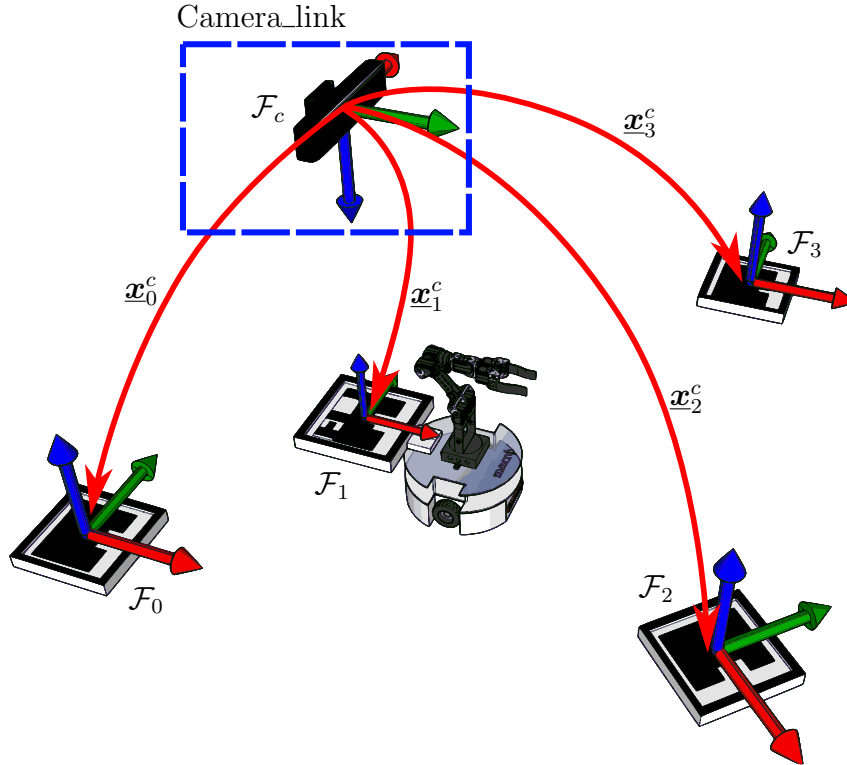


Figure 6.1: Frames of the vision system. The unit dual quaternions \mathbf{x}_0^c , \mathbf{x}_1^c , \mathbf{x}_2^c and \mathbf{x}_3^c represent the pose of the frames \mathcal{F}_0 , \mathcal{F}_1 , \mathcal{F}_2 and \mathcal{F}_3 respectively with respect to the fixed frame \mathcal{F}_c .

The frame \mathcal{F}_0 is finally used as a fixed global reference as shown in Fig. 6.2, and unit dual quaternions \mathbf{x}_1^0 , \mathbf{x}_2^0 and \mathbf{x}_3^0 that represent the pose of frames \mathcal{F}_1 , \mathcal{F}_2 , and \mathcal{F}_3 , respectively, with respect to the fixed frame \mathcal{F}_0 , are computed directly as follows (Adorno, 2012)

$$\mathbf{x}_i^0 = (\mathbf{x}_0^c)^* \mathbf{x}_i^c, \quad (6.2)$$

with $i = \{1, 2, 3\}$.

This approach has useful advantages such as the possibility of placing the markers arbitrarily within the visual range of the camera.

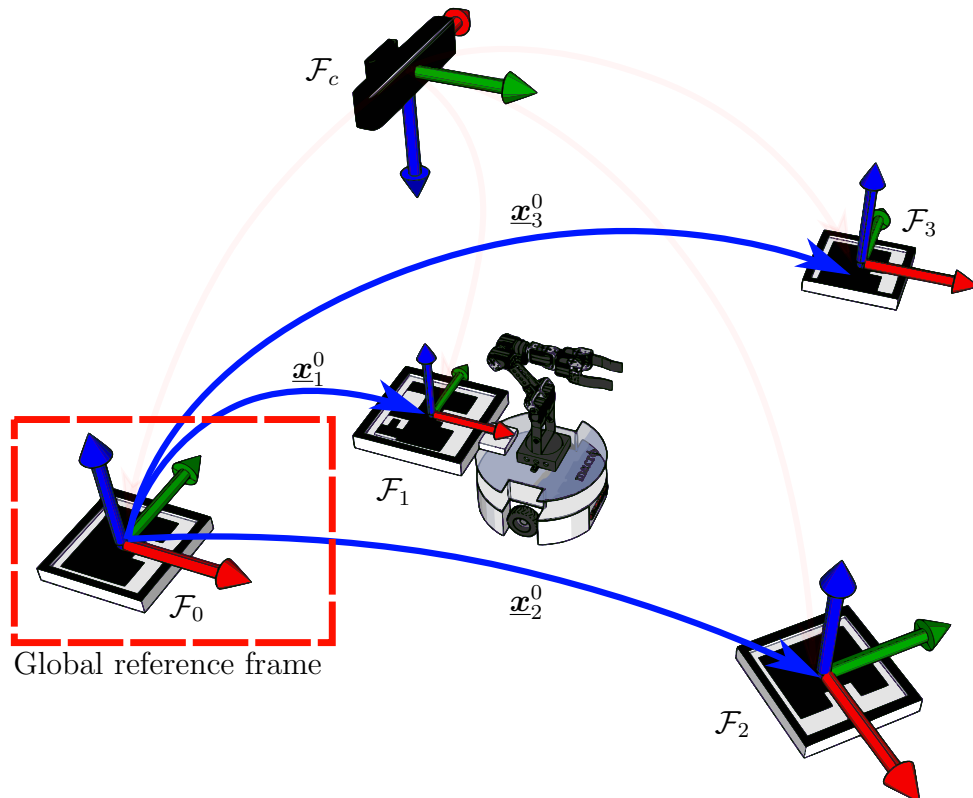


Figure 6.2: Global reference frame of the vision system. All markers are referenced with respect to the global reference frame \mathcal{F}_0 . This and the other markers, can be placed anywhere within the the visual range of the camera.

The structure of the experimental environment is based on nodes and topics. A vision system node publishes on a topic related to markers and the kinematic control node subscribes to that topic. The robot drivers, which are based on the Dynamixel and Turtlebot packages, run their respective nodes and the kinematic control node publishes and subscribes to them in order to command the robot and read the robot sensors, as shown in Fig. 6.3.

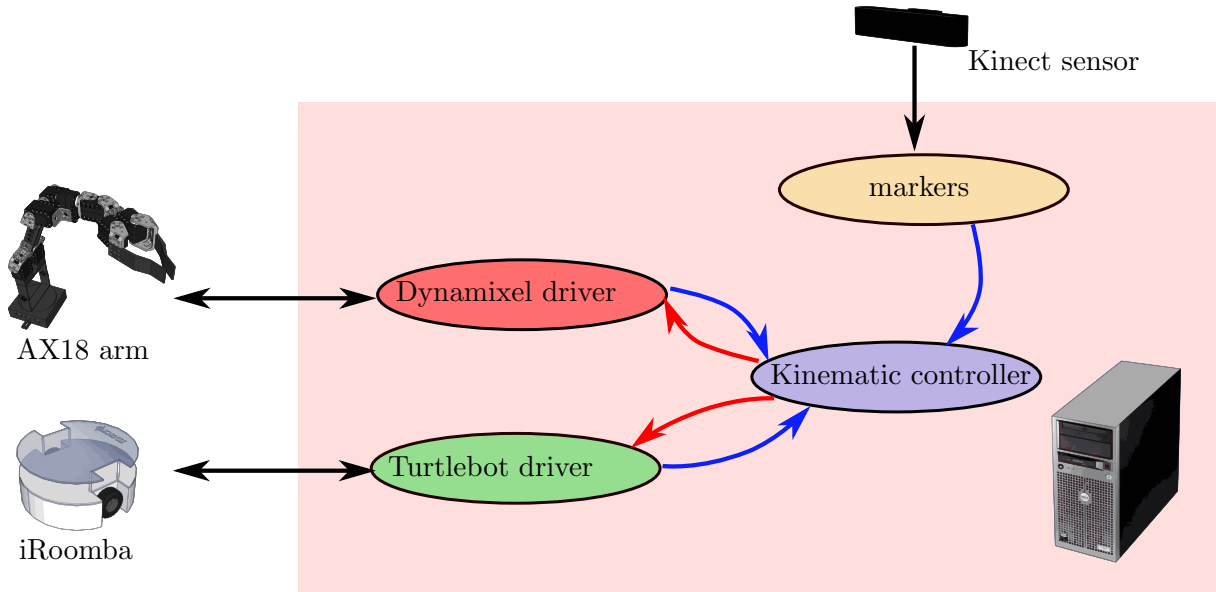


Figure 6.3: Structure of the experimental environment.

6.2 Experiments

6.2.1 Control of the end-effector pose

In the first experiment, whose setup is shown in Fig 6.5, the goal was to control the end-effector pose by using the proposed approach based on linear programming (i.e., the control inputs were generated by (5.29)). The robot had to grasp a box and put it, with certain orientation, inside a trash bin located in a certain position. The robot and the trash bin can be located arbitrary within the visual range of the camera. The relation between the frames \mathcal{F}_0 , \mathcal{F}_1 , and \mathcal{F}_2 are known by means of the vision system (see Fig. 6.4). The desired pose is computed as follow

$$\mathbf{x}_d^0 = \mathbf{x}_2^0 \mathbf{x}_d^2, \quad (6.3)$$

where \mathbf{x}_d^2 is a constant unit dual quaternion that is known and represents the relation between the frames \mathcal{F}_2 and \mathcal{F}_d , as shown in Fig. 6.4.

A performance comparison with the classic continuous controller (5.3) is presented. For both controllers $\eta = 0.6$, and for the linear program $\beta = 15$. Both parameters were adjusted experimentally in order to obtain a fast and smooth convergence. For the controller based on linear programming $k = 1$ and $\beta_l = 0.9$, which are parameters used in the constraint (5.14).

Fig. 6.6 shows the comparison between the control inputs generated by the classic approach (PINV), given by (5.3), and the linear programming control (LP) given by (5.29), which was implemented with the Simplex algorithm. LP generates more abrupt control inputs, but Fig. 6.7 shows that it uses fewer actuators than the classic controller

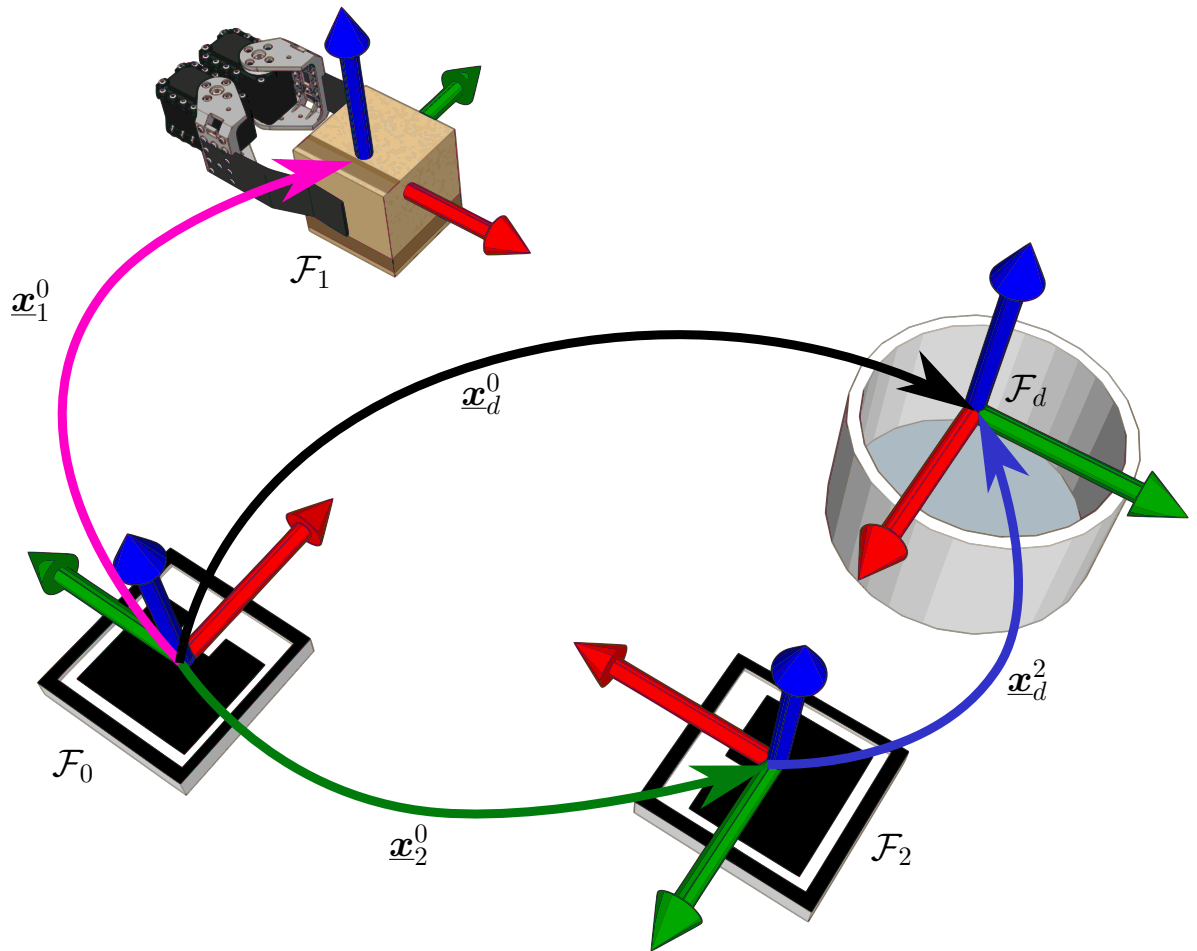


Figure 6.4: Description of the experiment: The box must be placed inside the a trash bin. The relation between the frame \mathcal{F}_1 and the \mathcal{F}_2 with respect to the reference frame \mathcal{F}_0 are given by the unit dual quaternions $\underline{\mathbf{x}}_1^0$ and $\underline{\mathbf{x}}_2^0$, respectively. The desired pose $\underline{\mathbf{x}}_d^0$ is computed by using a known rigid transformation between the frames \mathcal{F}_2 and \mathcal{F}_d and is given by $\underline{\mathbf{x}}_d^0 = \underline{\mathbf{x}}_2^0 \underline{\mathbf{x}}_d^2$.

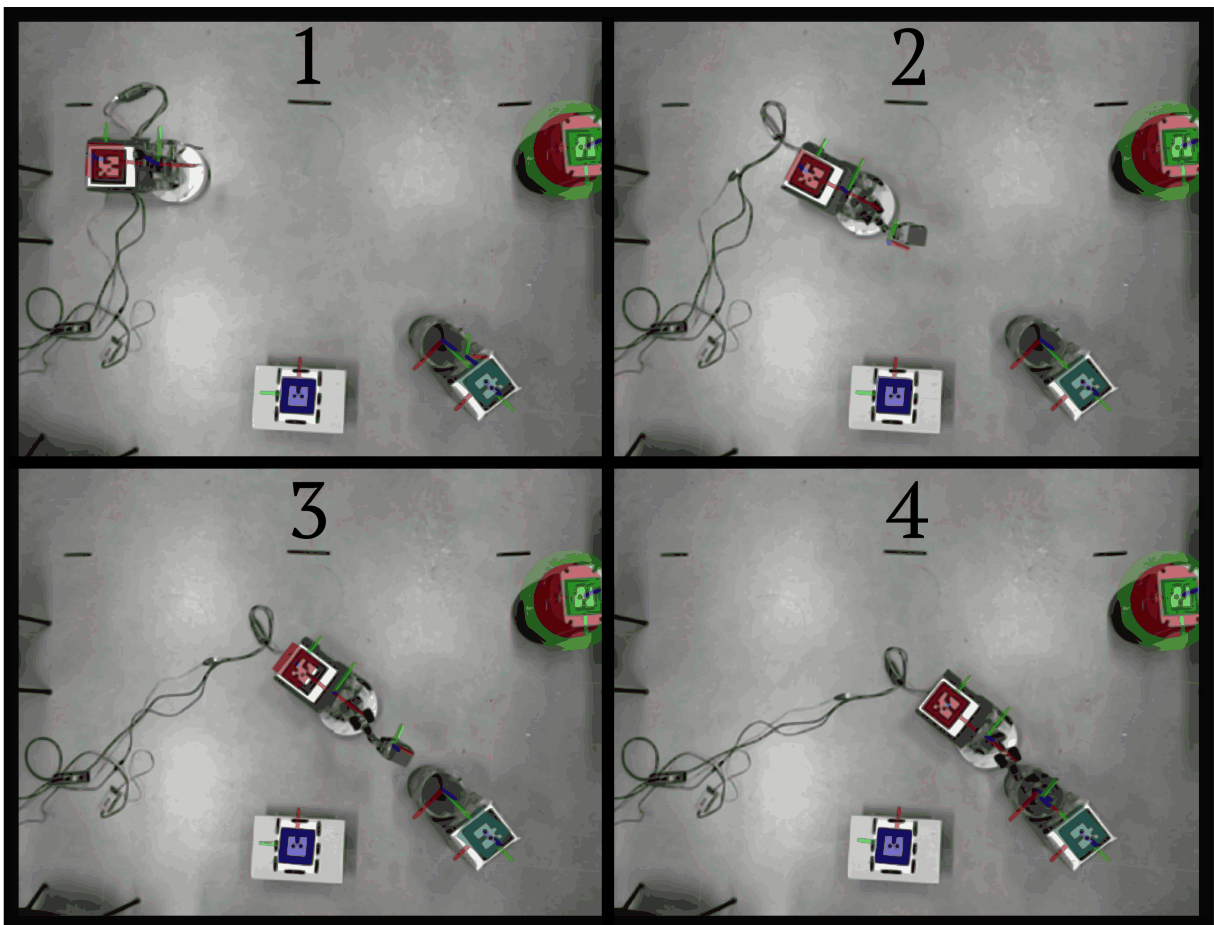


Figure 6.5: Control of the end-effector **pose**: snapshots from the experiment. A Microsoft Kinect sensor located at the ceiling is used to recognize fiducial markers placed on the robot, the obstacles, and the desired goal.

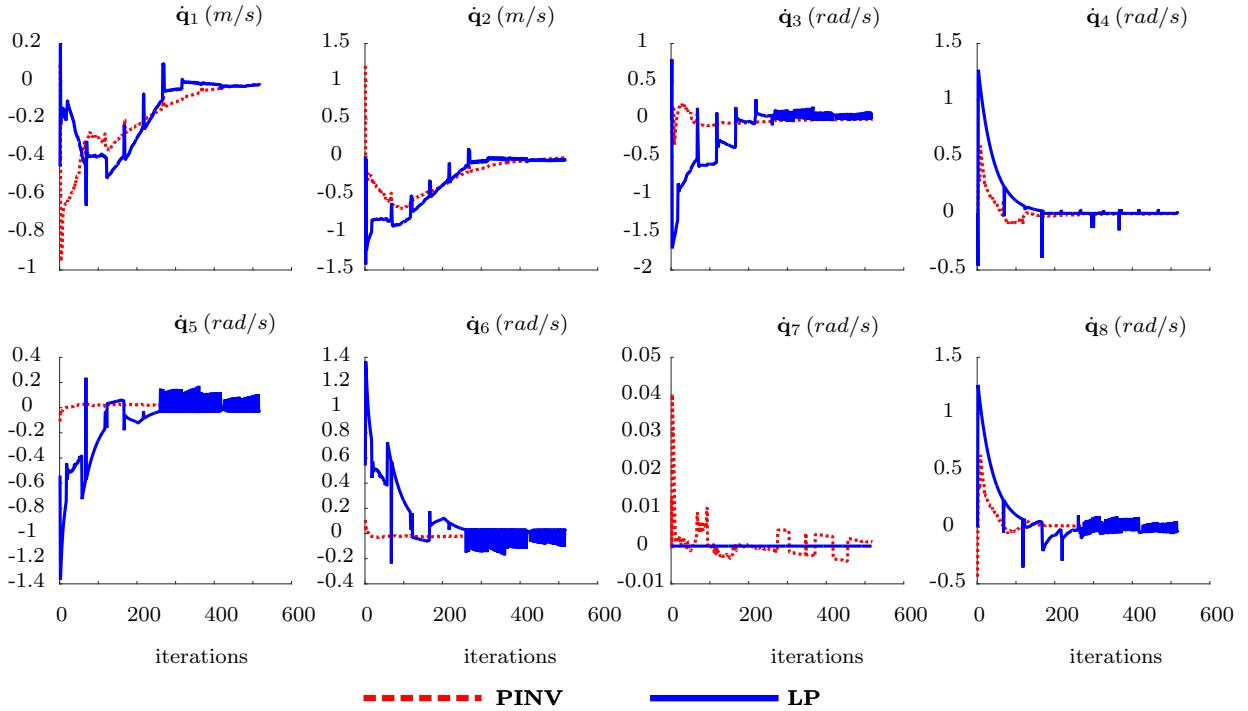


Figure 6.6: Control of the end-effector **pose**: control inputs using the classic kinematic control (PINV) and the parsimonious control using linear programming with Simplex (LP).

in each iteration. This is expected considering the fact that, in addition to the number of DOF required by the task, each equality constraint requires one DOF but the inequality constraints require DOF *only* when they become equalities. This way, since the task requires 6 DOF and there is one equality constraint to impose the nonholonomic constraint, we expect that 7 DOF should be used at all times, unless the inequality constraints take action (the equality constraint (5.28) related to obstacle avoidance is activated only when the robot is close to the obstacle, so it behaves as an inequality constraint). Remarkably, after the 150th iteration LP used only 6 DOF most of the time, thanks to the fact that at this point the end-effector pose was close to the desired one, so the controller needed to generate signals only to maintain the end-effector pose and to make small adjustments until complete stabilization. On the other hand, the classic approach, PINV, used all DOF all the time, as expected.

It is important to highlight that Fig. 6.7 shows the number of entries generated per iteration by the controllers and does not show the entries really used in the experimental test. This is because some generated entries are close to zero and they are considered zero by the low-level drivers of the servomotors and, consequently, a lower number of entries are used. In order to show the number of entries used in the experimental test, we introduce the concept of practical parsimony, where values lower than a threshold $\mathcal{Y} = 0.01$, (which correspond to the minimal command velocity accepted by the the motors) are set to zero. Fig. 6.8 shows the number of entries used in the experimental test.

Fig. 6.9 shows the time response of both approaches for each coefficient of the dual quaternion error given by

$$\underline{e} = \underline{x} - \underline{x}_d = e_1 + e_2\hat{i} + e_3\hat{j} + e_4\hat{k} + \varepsilon (e_5 + e_6\hat{i} + e_7\hat{j} + e_8\hat{k}).$$

All coefficients stabilize for both controllers, but LP has a less smooth response as we can observe mainly in the time response of coefficients e_2 and e_3 . This is also expected because LP is a discontinuous controller whereas PINV is a continuous one; therefore, as the control inputs generated by the former are more abrupt than the ones generated by the latter, we expect that the response will be less smooth for LP.

Fig. 6.10 shows the behavior of both controllers in face of joints limits applied only to the manipulator robot. LP respected those joints limits at all times, as expected, whereas PINV violated the limit of the 5th joint, represented by q_8 . It is important to highlight that limits could also be applied to the mobile base in order to confine it into a specific subset of the workspace.

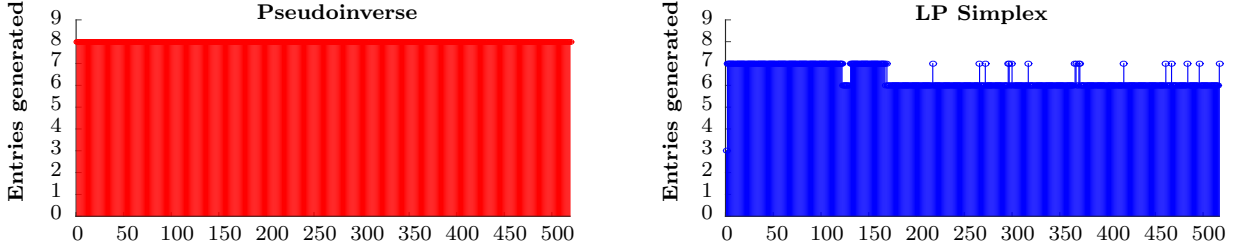


Figure 6.7: Control of the end-effector **pose**: number of nonzero entries generated using the classic kinematic control (PINV) and the parsimonious control using linear programming with Simplex (LP).

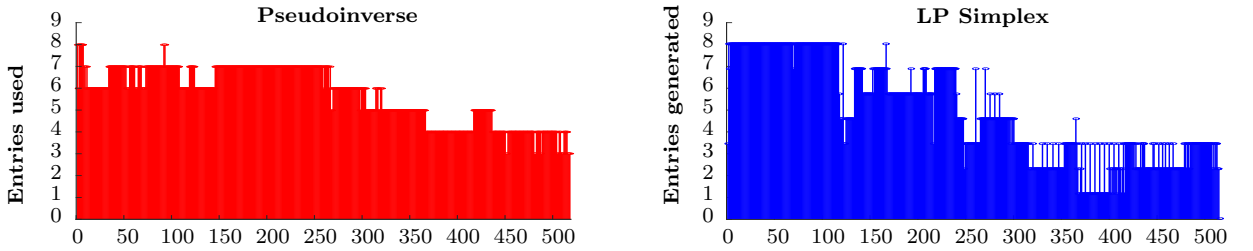


Figure 6.8: Control of the end-effector **pose**: number of nonzero entries used in the experimental test using the classic kinematic control (PINV) and the parsimonious control using linear programming with Simplex (LP).

Table 6.1 presents the comparison with respect to four metrics between the control inputs generated by both methods. As expected, since the classic control law minimizes the 2-norm, it performs better in metrics that use the 2-norm (i.e., metrics 2 and 4); conversely, the parsimonious control using linear programming minimizes the 1-norm and hence performs better in metrics that use this norm (i.e., metrics 1 and 3).

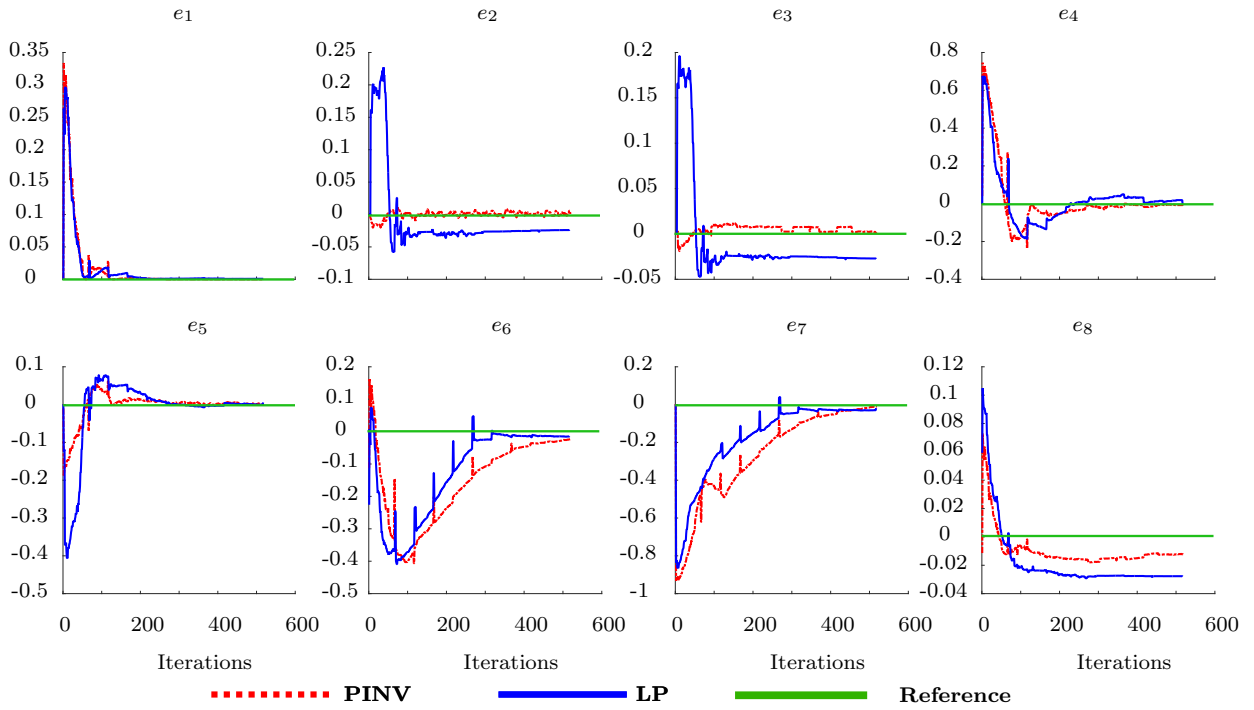


Figure 6.9: Control of the end-effector **pose**: time response of each coefficient of the dual quaternion error.

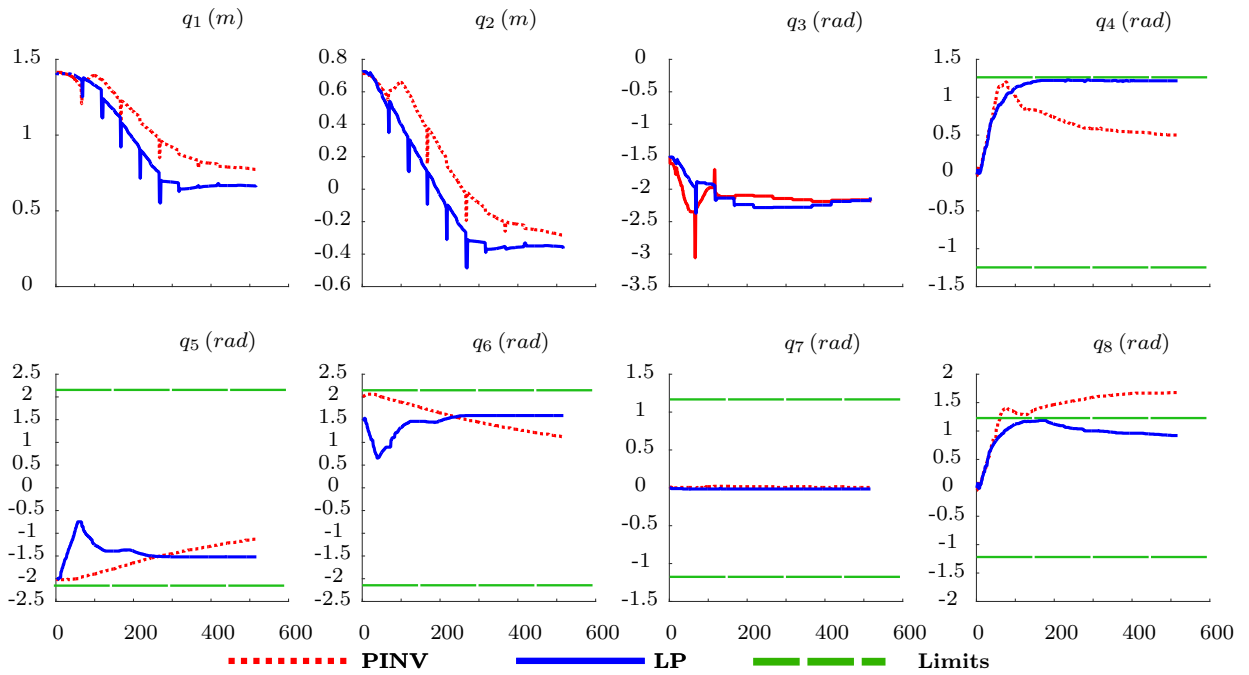


Figure 6.10: Control of the end-effector **pose**: coefficients of the robot configuration \mathbf{q} (see (4.9)) using the classic kinematic control (PINV) and the parsimonious control using linear programming with Simplex (LP). The joints limits are represented by *dashed green* lines.

Table 6.1: Control effort: pseudoinverse versus linear programming.

	Metric	Pose control		Position Control	
		PINV	LP Simplex	PINV	LP Simplex
1	$\int_0^\infty \ \dot{\mathbf{q}}(\mathbf{t})\ _1 dt$	1.9796	1.4944	0.3907	0.3431
2	$\sqrt{\int_0^\infty \ \dot{\mathbf{q}}(\mathbf{t})\ _2^2 dt}$	0.2960	0.3267	0.0583	0.0654
3	$\int_0^\infty \ \ddot{\mathbf{q}}(\mathbf{t})\ _1 dt$	0.0026	0.0012	0.0047	0.0021
4	$\sqrt{\int_0^\infty \ \ddot{\mathbf{q}}(\mathbf{t})\ _2^2 dt}$	0.00013	0.00004	0.00084	0.00085

6.2.2 Control of the end-effector position

In order to increase the functional redundancy of the system, a second experiment was performed where the robot had to control only the end-effector position, which requires only three DOF. The control laws are essentially the same, with the difference that the error is calculated between the current position \mathbf{p} and the desired position \mathbf{p}_d , and the corresponding task Jacobian \mathbf{J}_p is calculated algebraically from the original Jacobian \mathbf{J} as shown in 4.4 (Adorno et al., 2010). The parameter $\eta = 1$ was used for both controllers and $\beta = 10$ was used for LP. For the controller based on LP $k = 1$ and $\beta_l = 0.9$. Fig. 6.11 shows the control inputs for both methods and again the parsimonious control inputs were more abrupt.

Fig. 6.12 shows an important fact when using Simplex to solve the linear program: more redundancy implies more parsimony; that is, less DOF are used. Whereas in the classic controller all DOF were used, LP used only five DOF most of the time. Since the main task requires 3 DOF and the nonholonomic constraint requires 1 DOF, we conclude that other inequality constraints were activated. Fig. 6.13 shows the number of entries used per iteration in the experimental test.

Again, Table 6.1 shows that metrics based on the 1-norm favored LP whereas the ones based on the 2-norm favored the classic controller.

Fig. 6.14 shows the end effector translation from the initial to the desired position. It can be observed that the robot stabilizes very close to the desired position respecting all the constraints imposed.

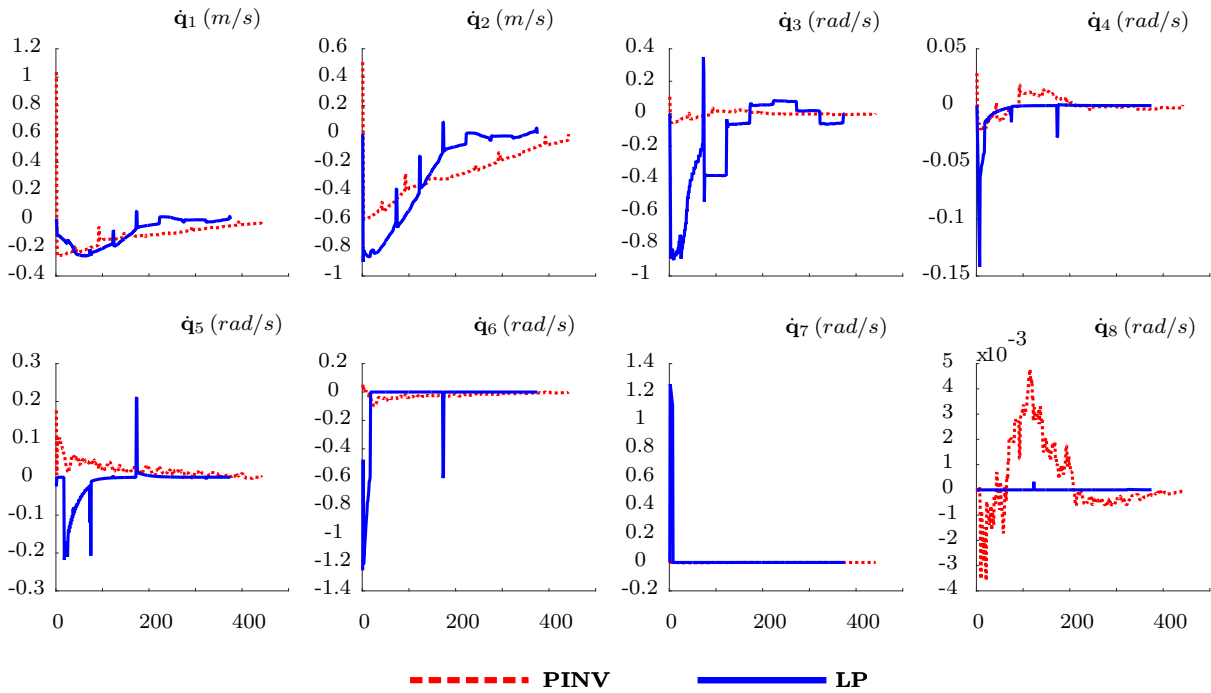


Figure 6.11: Control of the end-effector **position**: control inputs using the classic kinematic control (LEFT) and the parsimonious control using linear programming with Simplex (RIGHT).

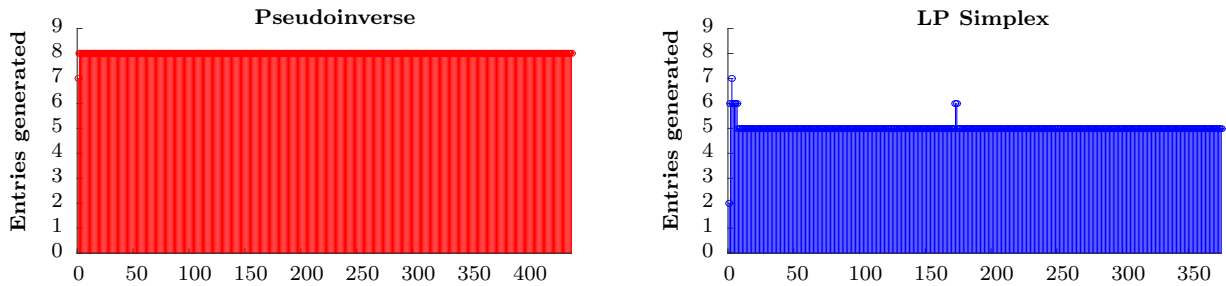


Figure 6.12: Control of the end-effector **position**: number of nonzero entries generated using the classic kinematic control (PINV) and the parsimonious control using linear programming with Simplex (LP).

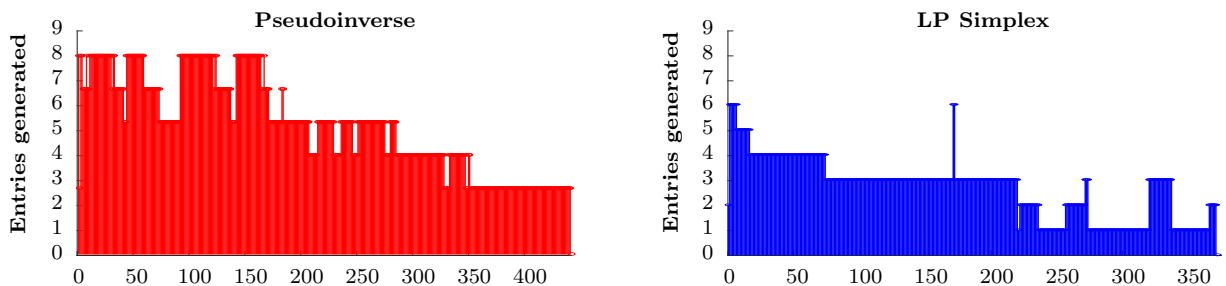


Figure 6.13: Control of the end-effector **position**: number of nonzero entries used in the experimental test using the classic kinematic control (PINV) and the parsimonious control using linear programming with Simplex (LP).

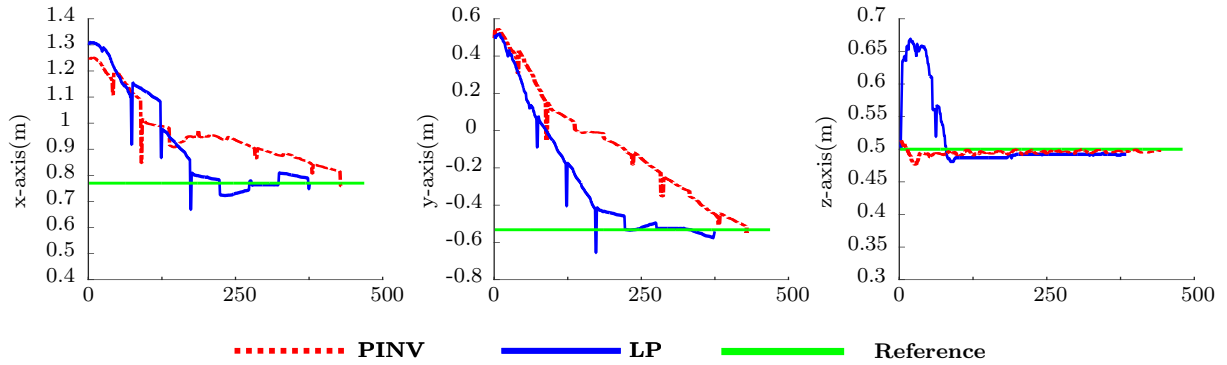


Figure 6.14: Control of the end-effector **position**: End effector translation using the classic kinematic control (PINV) and the parsimonious control using linear programming with Simplex (LP). The reference is represented by the straight green line.

6.2.3 Obstacle avoidance

A third experiment was executed to show the performance of LP in the task of controlling the end-effector pose while avoiding an obstacle, as shown in Fig. 6.16. This experiment is the more sophisticated in the sense that more constraints are imposed. As in the aforementioned experiments, all constraints are implemented in order to consider the nonholonomic constraint of the mobile platform and prevent violation of the joints limits in the arm manipulator. In addition, the constraint (5.28) is activated when the robot approaches to an obstacle in the plane in order to prevent collisions with the mobile base. There is no a comparison with respect to the classic method because it does not allow to include inequality constraints. We can see in Fig. 6.17 that the robot used all DOF throughout most part of the experiment. This was also expected because the task requires 6 DOF, the nonholonomic constraint requires 1 DOF, and the constraint for obstacle avoidance—when activated—requires 1 DOF, which in total corresponds to 8 DOF. Other inequalities constraints, if activated, can make the end-effector stabilize before reaching the desired set point, but this was not the case for this particular experiment. However, note that this method may fall into local minima. There is no guarantee of reaching the desired set point, but at least there is guarantee of preventing collisions between the mobile base and obstacles in the plane. In order to avoid local minima, a good alternative would be the use of probabilistic motion planing algorithms.

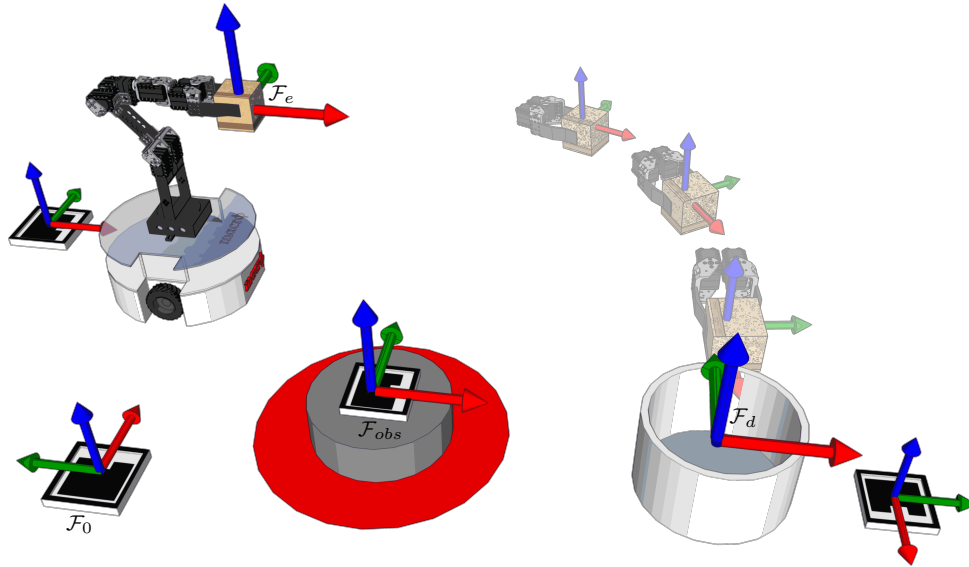


Figure 6.15: Description of the task: The robot must to grasp a box in \mathcal{F}_e and put it inside a trash bin located in \mathcal{F}_d . There is an obstacle in the plane located in \mathcal{F}_{obs} and the controller must to prevent a collision between the mobile platform and the obstacle. The pose of all frames are computed with respect to \mathcal{F}_0 by using the vision system.

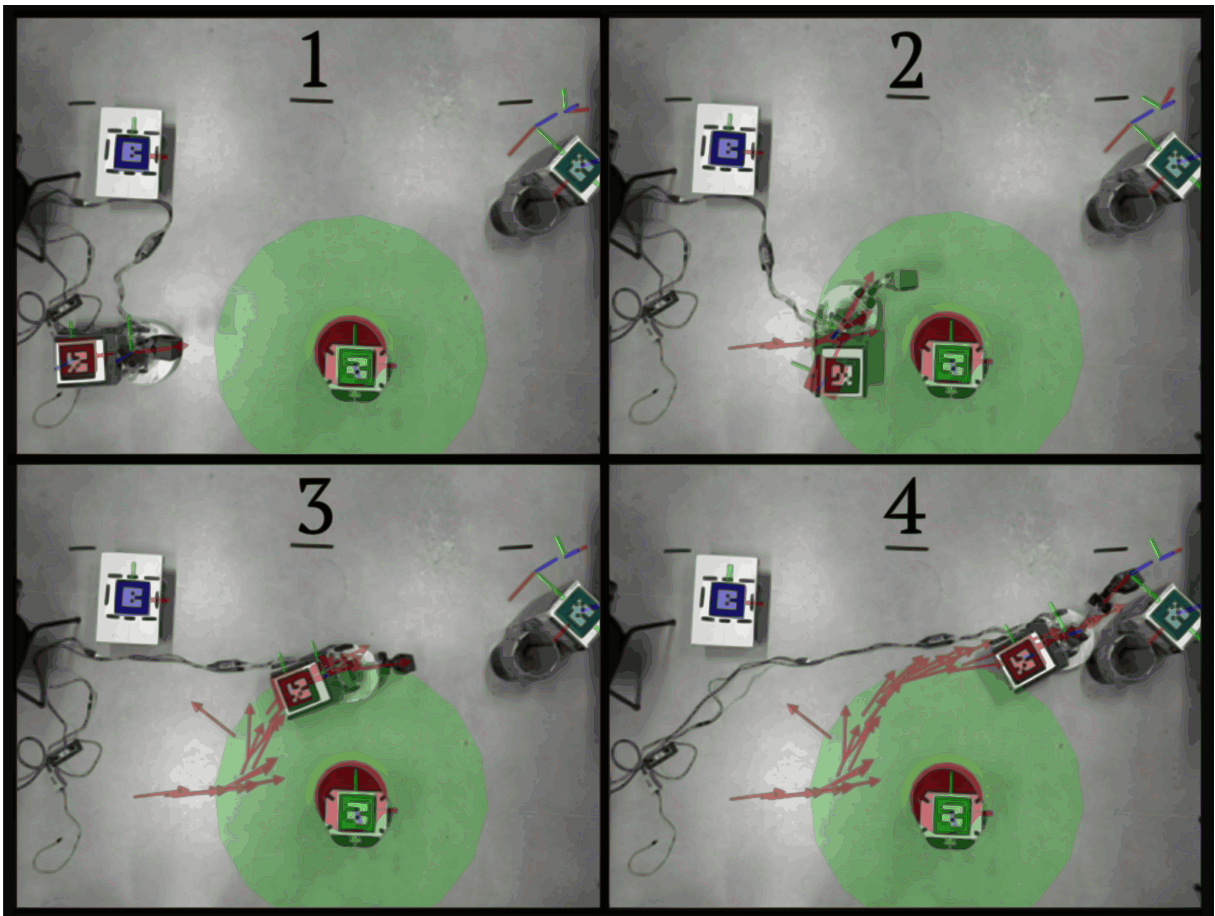


Figure 6.16: Control of the end-effector **pose** while avoiding an obstacle: snapshots from the experiment.

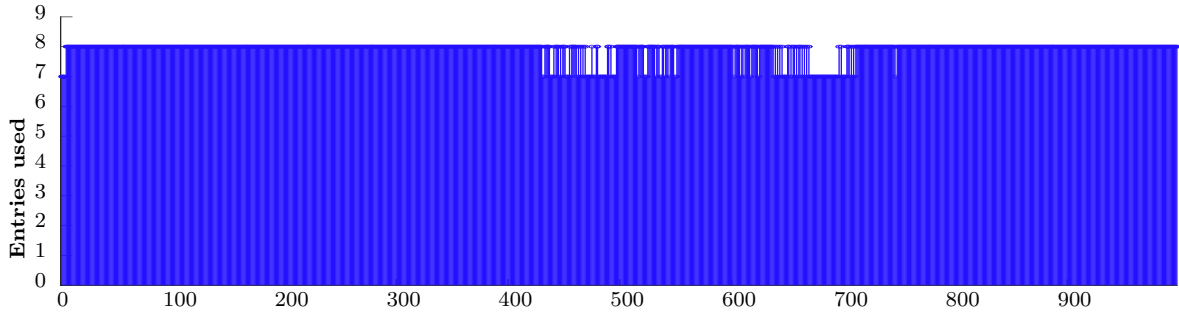


Figure 6.17: Control of the end-effector **pose** while avoiding an obstacle: number of nonzero entries using linear programming with Simplex (LP).

6.3 Computational Efficiency Test

The experimental results were performed on ROS and C++, where each kinematic control node was set to 50Hz or 20ms per iteration, which is the maximum value supported by the ROS library Dynamixel motors. Both approaches had similar performance as the convergence rate was the same for both controllers and each iteration time was always less than 20ms. In order to show the computational efficiency two tests were performed in the same computer running Xubuntu 16.04 64bits and equipped with Intel Core i7-4712HQ and 16Gb RAM. Each test consisted in executing the same pose control task 10000 times.

6.3.1 Test 1: Holonomic mobile manipulator case.

In this first test, the mobile manipulator was considered as a holonomic mobile manipulator. Consequently, no cascade scheme was used in the classic controller and no additional constraints were imposed to the controller based on linear programming.. The goal was to control the end-effector pose. For both controllers $\eta = 5$ and for the linear program $\beta = 40$. It was considered that a trial or simulation converged when the error norm was less than 0.001. Fig. 6.18 shows the convergence time used in each trial when the pose task required 46 iterations for both controllers. The average convergence time was 158.96ms with a standard deviation $\sigma_{pinv} = 5.23\text{ms}$ for the classic controller and 173.42ms with a standard deviation $\sigma_{lp} = 2.79\text{ms}$ for the controller based in linear programming.

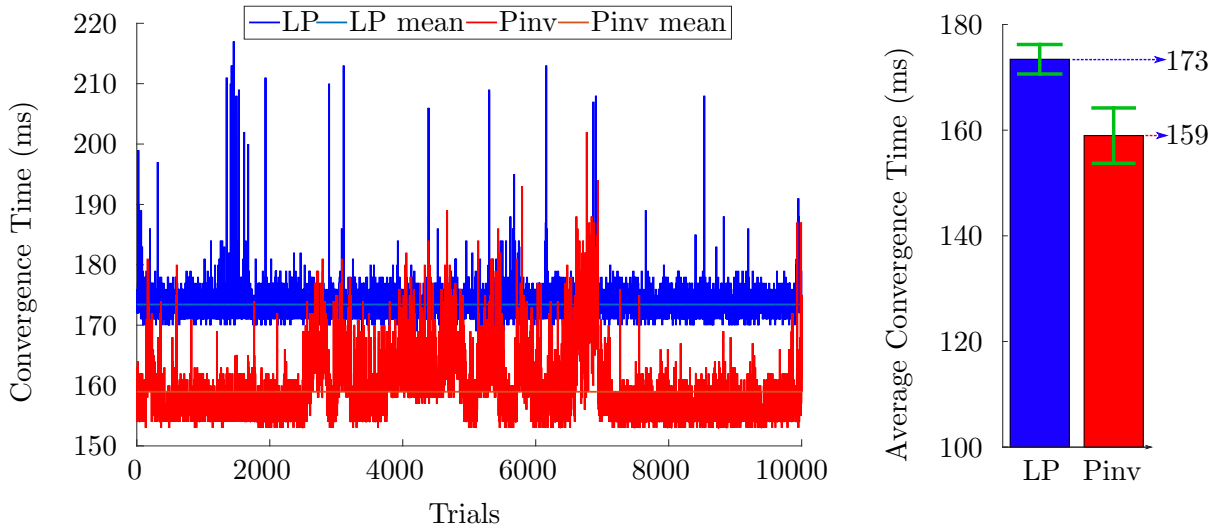


Figure 6.18: Convergence time in a pose task that required 46 iterations for both controllers. In the *left*, it is presented the convergence time in each trial performed. In the *right*, it is presented the average convergence time with the errors bar, which denote the respective standard deviation.

6.3.2 Test 2: Nonholonomic mobile manipulator case

In this second test, the mobile manipulator was considered as a nonholonomic mobile manipulator. It was used a cascade scheme in the controller based on the pseudoinverse of the Jacobian matrix in order to deal with the nonholonomic constraint. In the case of the controller based on linear programming, the nonholonomic constraint was imposed as an equality constraint in the linear programming formulation. The goal was to control the end-effector pose. The desired pose was different with respect to the first test. For both controllers $\eta = 5$ and for the linear program $\beta = 40$, $k = 1$ and $\beta_l = 0.99$. It was considered that a trial or simulation ended when the error norm was less than 0.001. Fig. 6.19 shows the convergence time used in each trial when the pose task required 77 iterations for both controllers, the pseudoinverse and the linear programming approach. The average convergence time were 251.03ms with a standard deviation $\sigma_{pinv} = 7.58\text{ms}$ and 290.14ms with a standard deviation $\sigma_{lp} = 4.30\text{ms}$, respectively.

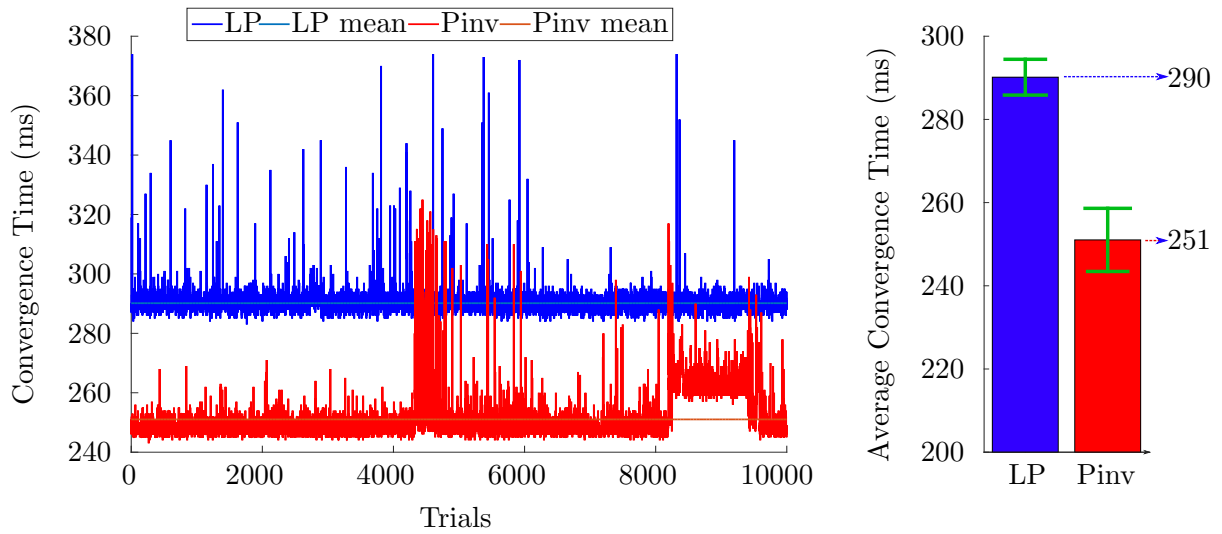


Figure 6.19: Convergence time in a pose task that required 77 iterations for both controllers. In the *left*, it is presented the convergence time in each trial performed. In the *right*, it is presented the average convergence time with the errors bar, which denote the respective standard deviation.

The time required to perform an iteration can be estimated by dividing the average convergence time into the number of iterations. Fig. 6.20 shows the average time per iteration for both controllers for both tests. The controller based on the pseudoinverse of the Jacobian matrix is faster than the controller based on the linear programming. The former needs about 3.4ms to perform an iteration and the latter needs about 3.7ms. However, both are very efficient and fast enough to perform feedback control.

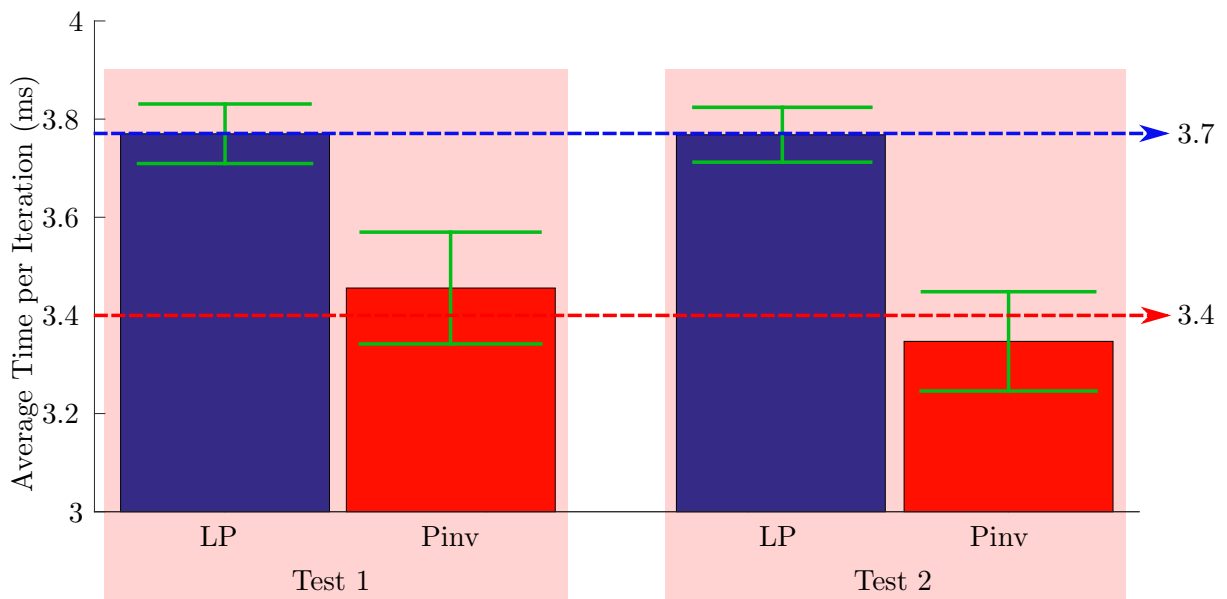


Figure 6.20: Average time per iteration for both controllers.

6.4 Chapter Conclusions

This chapter presented the experiments performed as well as the obtained results.

Section 6.1 detailed the specification of the experimental environment where all the experiments were performed. In addition the vision system implemented was detailed as well its architecture used. Section 6.2 showed the experiments performed. A first experiment was performed where consisted in controlling the end-effector pose. We noticed that when the linear program formulation is used the control inputs generated are more abrupt in comparison to the traditional approach, which is based on the pseudoinverse of the whole-body Jacobian matrix, although the former used fewer actuators. A second experiment was performed in order to increase the functional redundancy of the robot. The goal was to control the end-effector position and we verified a higher parsimony, less actuators were used and the joint limits were respected when the linear programming was applied. A third experiment was performed with an obstacle in the plane. The robot not only prevented the collision between the robot and the obstacle, but also performed movements around the obstacle and stabilized near the desired position.

The linear programming approach allows to include inequalities constraints easily in the optimization formulation. This feature is advantageous because allows take into account some physical limitations of the robot as limit joints or avoiding obstacles. The implementation of the controllers can be performed in C++ by using a very robust solver as CPLEX studio. This approach is very efficient and its performance is similar to the pseudoinverse performance. Furthermore, there are several open-source solvers that can be used in different languages programming. In addition, the vision system can be implemented using open-source ROS packages and the library of dual quaternion algebra DQ Robotics, both easy and ready to use.

7

Conclusion and Future Works

7.1 Conclusions

This master thesis presented a study and implementation of the robot kinematic control applied to a nonholonomic mobile manipulator by using linear programming proposed by Goncalves et al. (2016), which is computationally efficient, and has a guarantee formal of stability. Furthermore some improvements to the original formulation were proposed.

The kinematic control strategy is based on an optimization problem where the 1-norm is used, a convex positive definite error metric, in order to reduce the optimization problem to a linear program. In this formulation, a constraint that depends on the error was imposed by Goncalves et al. (2016) in order to avoid joint movement when the robot is asymptotically stabilized to the desired configuration. In cases when the robot does not stabilize asymptotically, the constraint imposed by Goncalves et al. (2016) does not avoid joint movement. A new constraint, that depends on the time-derivative of the error is proposed in order to solve this issue.

The nonholonomic constraint of the mobile base was imposed as an equality constraint in the linear program formulation. This approach avoids the necessity to use a cascade control scheme in order to deal with the nonholonomic constraint. On the other hand, the addition of this constraint requires always 1-DOF all the time.

In order to avoid violation of the joint limits were imposed inequality constraints. Because the linear programming formulation minimizes the joint velocities, the constraints were imposed in the joint velocities taking into account the joints positions. The possibility

of including inequality constraints is the main feature of the linear programming formulation with respect to the classic approach. When n inequality constraints take actions the controller requires n additional DOF.

An additional constraint was imposed in order to prevent collisions between the mobile base and obstacles in the plan, where the obstacles were modeled as circles. First a conservative solution was proposed to prevent collision that is simple to implement in the linear program formulation. A second one was proposed in order to enhance the robot behavior. The former forces the robot to stop before colliding with an obstacle and the latter prevent collision, and in most cases, the robot could may move around the obstacle and stabilizes in the desired configuration. This constraints to prevent collisions between the mobile base and obstacles in the plane were imposed as a low level protection.

In order to verify the control strategy and the proposed final formulation, experiments were performed on a real mobile manipulator. A Microsoft Kinect sensor was placed in the ceiling in order to enhance the localization of the mobile base, to detect the desired end-effector pose, and to detect obstacles in the workspace. The kinematic control was implemented using C++ on ROS Indigo in order to enhance the performance. A low pass filter using dual quaternion algebra was implemented in order to decrease the measurement noises of the camera.

In the first experiment performed the goal was to control the end-effector pose. The robot had to grasp a box and put it inside a trash bin and the end-effector must arrive at the goal in a certain orientation and there were no obstacles in the plane. The results showed that the robot stabilizes while respecting all imposed constraints. The linear programming controller was solved with the Simplex method and generated more abrupt control signals than the classic one, which is based on the pseudoinverse of the robot Jacobian matrix. This was expected because the former is a discontinuous controller whereas the latter is a continuous one. On other hand, the linear programming controller used fewer joints, specifically between 6 and 7 DOF. This was expected, because the task required 6-DOF and the nonholonomic constraint required 1-DOF.

A second experiment was performed in order to increase the functional redundancy of the robot. The goal was to control the end-effector position, which requires only 3-DOF. In this experiment, the robot had to grasp a box and put it inside a trash bin, but the orientation of the end-effector was not relevant and it was not controlled. The results showed that the robot stabilized in the desired position, but nevertheless, the linear programming controller generated more abrupt control signals than the classic approach and it respected the joints limits imposed, as expected. This experiment showed an important and expected fact: when the Simplex method is used, more redundancy implies a smaller number of DOF used.

A third experiment was performed in order to validate the method proposed to prevent collision between the mobile base and obstacles in the plane. In this experiment, the robot

had to grasp a box and put it inside a trash bin and the end-effector must arrive at the goal in a certain orientation. Also, all aforementioned constraint were used in order to prevent violations of joints limits and take into account the nonholonomic constraint of the mobile base. The results showed that the robot performed the specified task (i.e., control of the end-effector pose) preventing a collision between the mobile base and a obstacle in the plane. In this experiment the linear programming controller used, in most cases, all joints available, as expected.

7.2 Future Works

Future works will be focused on improving the implementation as well as validating the control strategy in hyper-redundant systems. There are some shortcomings that might be improved:

- The Simplex method is computationally efficient and provides a parsimonious behavior (i.e., a minimum amount of actuators are used to perform a particular task) in the robot. However, it generates abrupt control inputs. These abrupt control inputs generated could be attenuated by modifying the Simplex algorithm.
- The vision system implemented is very limited with respect to the visual area range of the camera. A second Microsoft Kinect Sensor is required in order to increase such area. However, because of the high computational cost of each Kinect sensor, the frequency of acquisition should be reduced and a more sophisticated estimator must be implemented in order to improve the performance and quality of the measurements.
- The constraints imposed to prevent collisions were oriented to obstacles in the 2-D plane. Additional constraints can be imposed in order to avoid collision with obstacles in the 3-D workspace. An interesting focus is to model the robot links and 3-D objects using Plücker lines.
- The parsimonious kinematic control strategy is oriented mainly to hyper-redundant system. An interesting work would be to test such kinematic control in a real humanoid robot.
- In this work did not performed a comparison between the linear programming and nonlinear programming approaches using numerical solvers for both approaches. This should be done in future works because additional constraints can be easily imposed for both controllers.
- The equality constraint proposed in this work to prevent collisions between the mobile base and obstacles in the plane violates some of the stability assumptions.

Future works will be focused on finding inequality constraints that do not violate the stability assumptions while enabling the robot to avoid obstacles in an efficient way.

Bibliography

- Adorno, B. V. (2011). *Two-arm Manipulation: From Manipulators to Enhanced Human-Robot Collaboration [Contribution à la manipulation à deux bras : des manipulateurs à la collaboration homme-robot]*. Phd thesis, Université Montpellier 2.
- Adorno, B. V. (2012). Manipulação Cooperativa Descentralizada Usando o Espaço Dual de Cooperação. In *XIX Congresso Brasileiro de Automática*, pages 1436–1443, Campina Grande. Sociedade Brasileira de Automática.
- Adorno, B. V. and Fraisse, P. (2016). The cross-motion invariant group and its application to kinematics. *IMA Journal of Mathematical Control and Information*, page dnw032.
- Adorno, B. V., Fraisse, P., and Druon, S. (2010). Dual position control strategies using the cooperative dual task-space framework. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3955–3960, Taipei. IEEE.
- Baillieul, J. (1985). Kinematic programming alternatives for redundant manipulators. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, pages 722–728.
- Bayle, B., Fourquet, J. Y., and Renaud, M. (2003). Manipulability of Wheeled Mobile Manipulators: Application to Motion Generation. *The International Journal of Robotics Research*, 22(7-8):565–581.
- Carlson, J. and Murphy, R. (2005). How UGVs physically fail in the field. *IEEE Transactions on Robotics*, 21(3):423–437.
- Cha, E., Forlizzi, J., and Srinivasa, S. S. (2015). Robots in the Home. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction - HRI '15*, pages 319–326, New York, New York, USA. ACM Press.
- 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.

- Chiaverini, S., Oriolo, G., and Maciejewski, A. A. (2016). Redundant Robots. In *Springer Handbook of Robotics*, pages 221–242. Springer International Publishing, Cham.
- Chin, K.-Y., Hong, Z.-W., and Chen, Y.-L. (2014). Impact of Using an Educational Robot-Based Learning System on Students’ Motivation in Elementary Education. *IEEE Transactions on Learning Technologies*, 7(4):333–345.
- Dai, H. and Tedrake, R. (2016). Planning robust walking motion on uneven terrain via convex optimization. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 579–586. IEEE.
- De Luca, A., Lanari, L., and Oriolo, G. (1992). Control of redundant robots on cyclic trajectories. In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pages 500–506. IEEE Comput. Soc. Press.
- De Luca, A. and Oriolo, G. (1991). The reduced gradient method for solving redundancy in robot arms. *Robotersysteme*, 7(May 1991):117–122.
- De Luca, A., Oriolo, G., and Giordano, P. (2006). Kinematic modeling and redundancy resolution for nonholonomic mobile manipulators. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 1867–1873. IEEE.
- De Luca, A., Oriolo, G., and Giordano, P. R. (2010). Kinematic control of nonholonomic mobile manipulators in the presence of steering wheels. In *2010 IEEE International Conference on Robotics and Automation*, pages 1792–1798. IEEE.
- De Santis, A., Siciliano, B., De Luca, A., and Bicchi, A. (2008). An atlas of physical human-robot interaction. *Mechanism and Machine Theory*, 43(3):253–270.
- Dietrich, A., Wimbock, T., Albu-Schaffer, A., and Hirzinger, G. (2011). Singularity avoidance for nonholonomic, omnidirectional wheeled mobile platforms with variable footprint. In *2011 IEEE International Conference on Robotics and Automation*, pages 6136–6142. IEEE.
- Escande, A., Mansard, N., and Wieber, P. (2012). Hierarchical Quadratic Programming Part 1: Fundamental Bases. *Projects.Laas.Fr*.
- Escande, A., Mansard, N., and Wieber, P.-B. (2010). Fast resolution of hierarchized inverse kinematics with inequality constraints. In *2010 IEEE International Conference on Robotics and Automation*, number 4, pages 3733–3738. IEEE.
- Escande, A., Mansard, N., and Wieber, P.-B. (2014). Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research*, 33(7):1006–1028.

- Fan-Tien Cheng, Tsing-Hua Chen, and York-Yih Sun (1994). Resolving manipulator redundancy under inequality constraints. *IEEE Transactions on Robotics and Automation*, 10(1):65–71.
- Figueredo, L., Adorno, B., Ishihara, J., and Borges, G. (2013). Robust kinematic control of manipulator robots using dual quaternion representation. In *2013 IEEE International Conference on Robotics and Automation*, pages 1949–1955, Karlsruhe. IEEE.
- Flacco, F. and De Luca, A. (2013). Fast redundancy resolution for high-dimensional robots executing prioritized tasks under hard bounds in the joint space. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2500–2506. IEEE.
- Fonseca, M. and Adorno, B. V. (2016). Whole-body Modeling and Hierarchical Control of a Humanoid Robot Based on Dual Quaternion Algebra. In *Latin American Robotics Symposium/Brazilian Symposium on Robotics (LARS/SBR)*, pages 103–108.
- Gardner, J. F. and Velinsky, S. A. (2000). Kinematics of mobile manipulators and implications for design. *Journal of Robotic Systems*, 17(6):309–320.
- Giordano, P. R., Fuchs, M., Albu-Schaffer, A., and Hirzinger, G. (2009). On the kinematic modeling and control of a mobile platform equipped with steering wheels and movable legs. In *2009 IEEE International Conference on Robotics and Automation*, pages 4080–4087. IEEE.
- Goncalves, V. M., Fraitse, P., Crosnier, A., and Adorno, B. V. (2016). Parsimonious Kinematic Control of Highly Redundant Robots. *IEEE Robotics and Automation Letters*, 1(1):65–72.
- Hamilton, W. R. (1844). II. On quaternions; or on a new system of imaginaries in algebra. *Philosophical Magazine Series 3*, 25(163):10–13.
- Ho, E. S. L., Komura, T., and Lau, R. W. H. (2005). Computing inverse kinematics with linear programming. In *Proceedings of the ACM symposium on Virtual reality software and technology - VRST '05*, page 163, New York, New York, USA. ACM Press.
- Jia, Y., Xi, N., and Nieves, E. (2014). Coordination of a nonholonomic mobile platform and an on-board manipulator. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4356–4361. IEEE.
- Kanoun, O., Lamiroux, F., and 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.
- Khatib, O. (1986). Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *The International Journal of Robotics Research*, 5(1):90–98.

- Kim, I. and Oh, J.-H. (2013). Inverse Kinematic Control of Humanoids under Joint Constraints. *International Journal of Advanced Robotic Systems*, 10(1):74.
- Kingston, Z. K., Dantam, N. T., and Kavraki, L. E. (2015). Kinematically constrained workspace control via linear optimization. In *IEEE-RAS International Conference on Humanoid Robots*, volume 2015-Decem, pages 758–764. IEEE.
- Kussaba, H. T., Figueredo, L. F., Ishihara, J. Y., and Adorno, B. V. (2017). Hybrid Kinematic Control for Rigid Body Pose Stabilization using Dual Quaternions. *Journal of the Franklin Institute*.
- Latombe, J.-C. (1991). *Robot Motion Planning*, volume 54. Springer US, Boston, MA.
- Laumond, J.-P., Mansard, N., and Lasserre, J. B. (2015). Optimization as motion selection principle in robot action. *Communications of the ACM*, 58(5):64–74.
- Li, H.-L. (1998). Solve least absolute value regression problems using modified goal programming techniques. *Computers & Operations Research*, 25(12):1137–1143.
- 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.
- Liu, Y. and Li, Y. (2006). A New Method of Executing Multiple Auxiliary Tasks by Redundant Nonholonomic Mobile Manipulators. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1–6. IEEE.
- Luenberger, D. G. and Ye, Y. (2016). *Linear and Nonlinear Programming*, volume 228 of *International Series in Operations Research & Management Science*. Springer International Publishing, Cham, third edition.
- Mansard, N., Khatib, O., and Kheddar, A. (2009). A Unified Approach to Integrate Unilateral Constraints in the Stack of Tasks. *IEEE Transactions on Robotics*, 25(3):670–685.
- Marani, G. (2004). A general singularity avoidance framework for robot manipulators: task reconstruction method. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, number April, pages 4809–4814 Vol.5. IEEE.
- Marinho, M. M., Figueredo, L. F. C., and Adorno, B. V. (2015). A dual quaternion linear-quadratic optimal controller for trajectory tracking. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2015-Decem, pages 4047–4052. IEEE.
- Murty, K. (1983). *Linear Programming*. John Wiley & Sons.

- Nakamura, Y. and Hanafusa, H. (1987). Optimal Redundancy Control of Robot Manipulators. *The International Journal of Robotics Research*, 6(1):32–42.
- Nakamura, Y., Hanafusa, H., and Yoshikawa, T. (1987). Task-Priority Based Redundancy Control of Robot Manipulators. *The International Journal of Robotics Research*, 6(2):3–15.
- Nakamura, Y., Yamane, K., Fujita, Y., and Suzuki, I. (2005). Somatosensory computation for man-machine interface from motion-capture data and musculoskeletal human model. *IEEE Transactions on Robotics*, 21(1):58–66.
- Oliveira, A. C. and Adorno, B. V. (2015). Balance Control of a Humanoid Robot Based on the Cooperative Dual Task-Space Framework. In *XII Simpósio Brasileiro de Automação Inteligente (SBAI)*, pages 485 – 490.
- Perez, A. and McCarthy, J. M. (2004). Dual Quaternion Synthesis of Constrained Robotic Systems. *Journal of Mechanical Design*, 126(3):425.
- Pham, H.-L., Perdereau, V., Adorno, B. V., and Fraitse, P. (2010). Position and orientation control of robot manipulators using dual quaternion feedback. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 658–663, Taipei. IEEE.
- Quiroz-Omana, J. J. and Adorno, B. V. (2016). Parsimonious Kinematic Control of Nonholonomic Mobile Manipulators. In *2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR)*, pages 73–78, Recife. IEEE.
- Rauscher, M., Kimmel, M., and Hirche, S. (2016). Constrained robot control using control barrier functions. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 279–285. IEEE.
- Reister, D. and Unseren, M. (1993). Position and constraint force control of a vehicle with two or more steerable drive wheels. *IEEE Transactions on Robotics and Automation*, 9(6):723–731.
- Salazar-Sangucho, F. R. (2014). *Modelagem e Controle de Corpo Completo Usando Quatérnios Duais para um Manipulador Móvel*. PhD thesis, Universidade Federal de Minas Gerais.
- Salazar-Sangucho, F. R. and Adorno, B. V. (2014). Modelagem e Controle de Corpo Completo Usando Quatérnios Duais para um Manipulador Móvel. In *Congresso Brasileiro de Automática*, pages 1544–1551, Belo Horizonte.
- Sciavicco, L. and Siciliano, B. (1988). A solution algorithm to the inverse kinematic problem for redundant manipulators. *IEEE Journal on Robotics and Automation*, 4(4):403–410.

- Selig, J. (2005). *Geometric Fundamentals of Robotics*. Monographs in Computer Science. Springer New York, New York, NY.
- Sentis, L. and Khatib, O. (2005). Control of Free-Floating Humanoid Robots Through Task Prioritization. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1718–1723. IEEE.
- Seraji, H. (1998). A Unified Approach to Motion Control of Mobile Manipulators. *The International Journal of Robotics Research*, 17(2):107–118.
- Siciliano, B. (1990). Kinematic control of redundant robot manipulators: A tutorial. *Journal of Intelligent and Robotic Systems*, 3(3):201–212.
- Siciliano, B. and Khatib, O., editors (2016). *Springer Handbook of Robotics*. Springer International Publishing, Cham.
- Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2009). *Robotics. Modelling Planning and Control*. Advanced Textbooks in Control and Signal Processing. Springer London, London.
- Siciliano, B. and Slotine, J.-J. (1991). A general framework for managing multiple tasks in highly redundant robotic systems. In *Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*, pages 1211–1216 vol.2. IEEE.
- Silva, F. F. A. and Adorno, B. V. (2016). Whole-Body Control of a Mobile Manipulator Using Feedback Linearization Based on Dual Quaternions. In *2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR)*, pages 293–298. IEEE.
- Skaar, S. B. (2007). Robot Modeling and Control-[Book review; M. Spong, S. Hutchinson, and M. Vidyasagar]. *IEEE Transactions on Automatic Control*, 52(2):378–379.
- Snell-Rood, E. (2016). Interdisciplinarity: Bring biologists into biomimetics. *Nature*, 529(7586):277–278.
- Spong, M. W., Hutchinson, S., and M., V. (2006). *Robot Modeling and Control*, volume 141. John Wiley & Sons.
- Stoger, C., Muller, A., and Gattringer, H. (2015). Kinematic analysis and singularity robust path control of a non-holonomic mobile platform with several steerable driving wheels. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4140–4145. IEEE.
- Taylor, R. (2006). A Perspective on Medical Robotics. *Proceedings of the IEEE*, 94(9):1652–1664.

- Thuilot, B., D'AAndrea-Novel, B., and Micaelli, A. (1996). Modeling and feedback control of mobile robots equipped with several steering wheels. *IEEE Transactions on Robotics and Automation*, 12(3):375–390.
- Tin Lun Lam, Huihuan Qian, Yangsheng Xu, and Guoqing Xu (2009). Omni-directional steer-by-wire interface for four wheel independent steering vehicle. In *2009 IEEE International Conference on Robotics and Automation*, pages 1383–1388. IEEE.
- Vanderbei, R. J. (1998). *Linear Programming: Foundations and Extensions*, volume 49.
- Wang, X. and Yu, C. (2013). Unit dual quaternion-based feedback linearization tracking problem for attitude and position dynamics. *Systems & Control Letters*, 62(3):225–233.
- Xiangke Wang, Changbin Yu, and Zhiyun Lin (2012). A Dual Quaternion Solution to Attitude and Position Control for Rigid-Body Coordination. *IEEE Transactions on Robotics*, 28(5):1162–1170.
- Zhang, H., Jia, Y., and Xi, N. (2012). Sensor-based redundancy resolution for a nonholonomic mobile manipulator. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5327–5332. IEEE.

A

Dimensions of the Mobile Manipulator

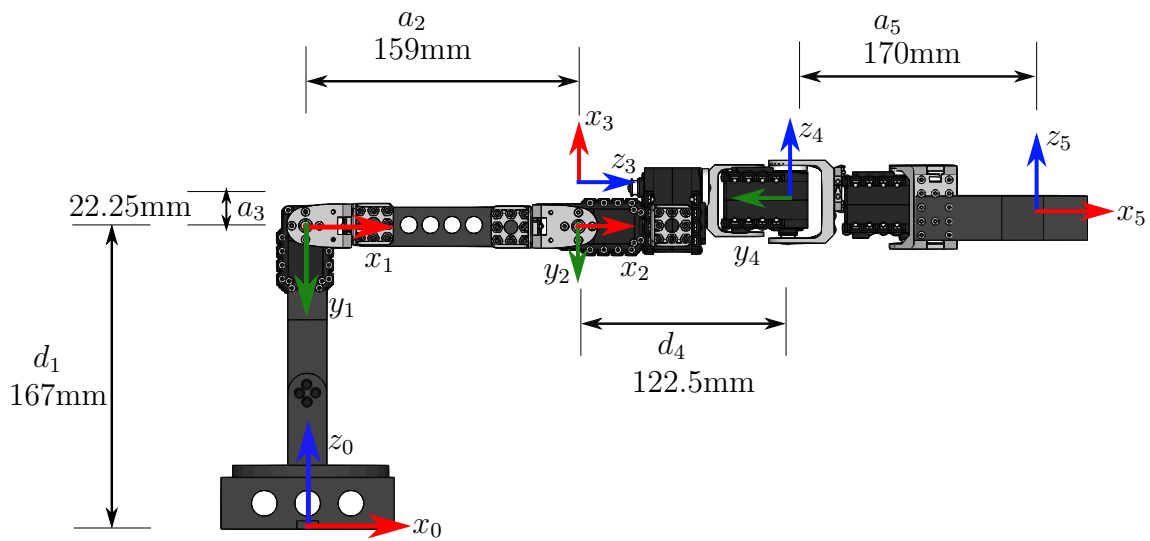


Figure A.1: AX18 arm manipulator.

i	a_i	α_i (rad)	d_i	θ_i (rad)
1	0	$-\frac{\pi}{2}$	d_1	θ_1
2	a_2	0	0	θ_2
3	a_3	$-\frac{\pi}{2}$	0	$\theta_3 - \frac{\pi}{2}$
4	0	$-\frac{\pi}{2}$	d_4	$\theta_4 - \frac{\pi}{2}$
5	a_5	0	0	$\theta_5 - \frac{\pi}{2}$

Table A.1: D-H parameters

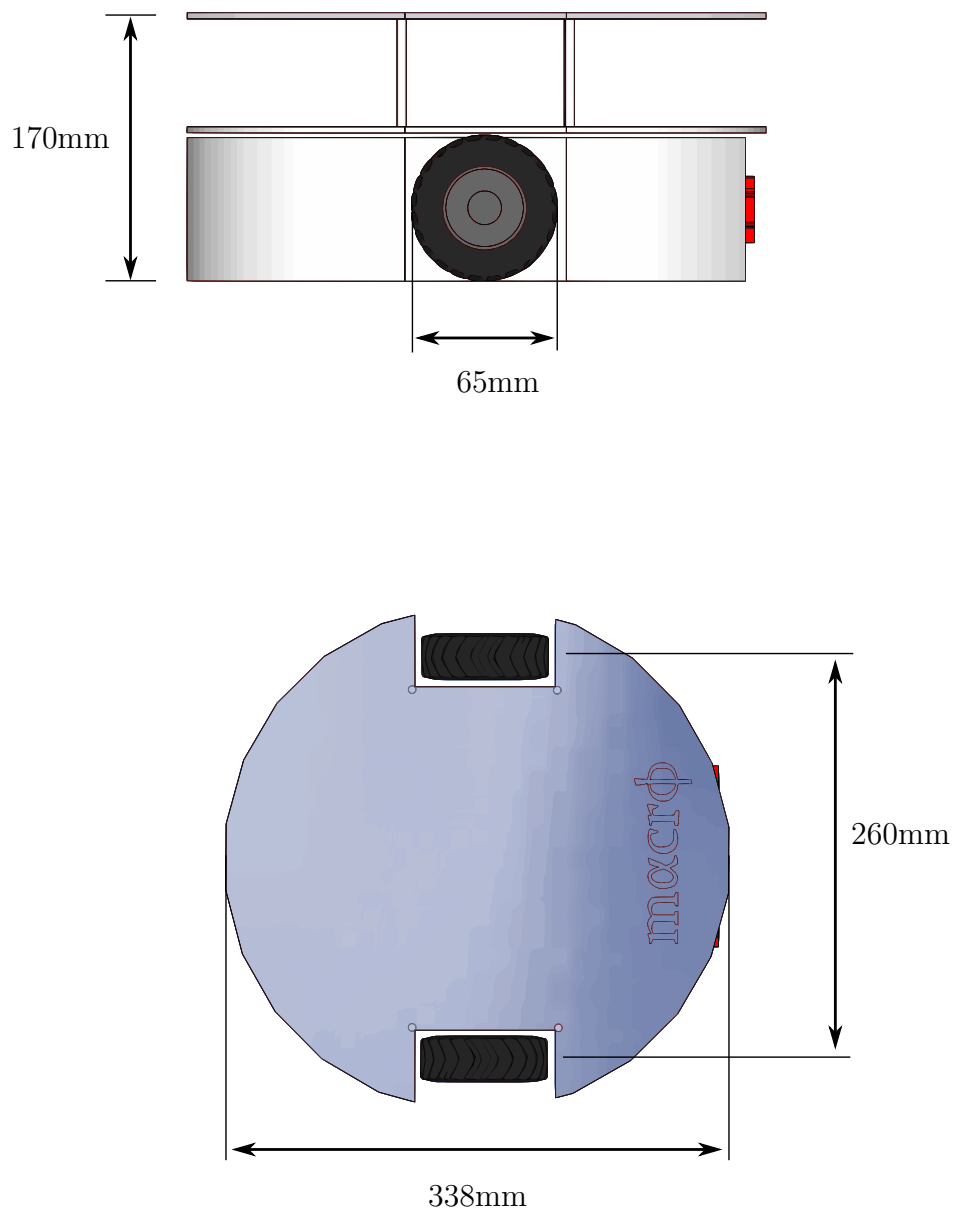


Figure A.2: Differential mobile base.

B

Whole-body kinematic control based on the Pseudoinverse

In order to control the end-effector pose by using whole-body motions (i.e., motions that can potentially—and automatically—use all available DOF), taking into account the nonholonomic constraint of the mobile base, the kinematic control strategy used in this work (for comparison with the kinematic control based on linear programming) is based on the cascade control scheme proposed in Salazar-Sangucho (2014), where the outer loop takes into account all DOF and the inner loop explicitly deals with the nonholonomic constraints of the mobile base by means of an input-output linearizing controller. This cascade scheme is shown in Fig. B.1.

The controller input \mathbf{u} is calculated using the classic approach, it was proposed in Adorno et al. (2010) and is given by

$$\mathbf{u} = -\mathbf{J}^\dagger \eta \text{vec}(\underline{\mathbf{x}} - \underline{\mathbf{x}}_d), \quad (\text{B.1})$$

where \mathbf{J}^\dagger is the generalized Moore-Penrose pseudoinverse, $\eta > 0$ is a scalar gain and $\underline{\mathbf{x}}_d$ and $\underline{\mathbf{x}}$ are the desired and measured end-effector poses, respectively. The control signal \mathbf{u} is partitioned as

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}_b^T & \mathbf{u}_a^T \end{bmatrix}^T,$$

where $\mathbf{u}_b = \begin{bmatrix} \dot{x} & \dot{y} & \dot{\phi} \end{bmatrix}^T$ and $\mathbf{u}_a = \dot{\mathbf{q}}_a$.

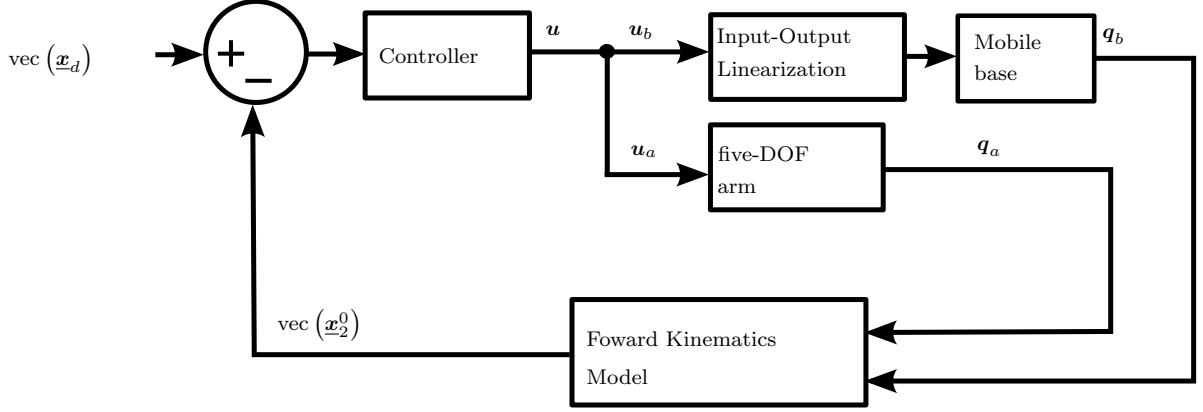


Figure B.1: Whole-body kinematic control scheme.

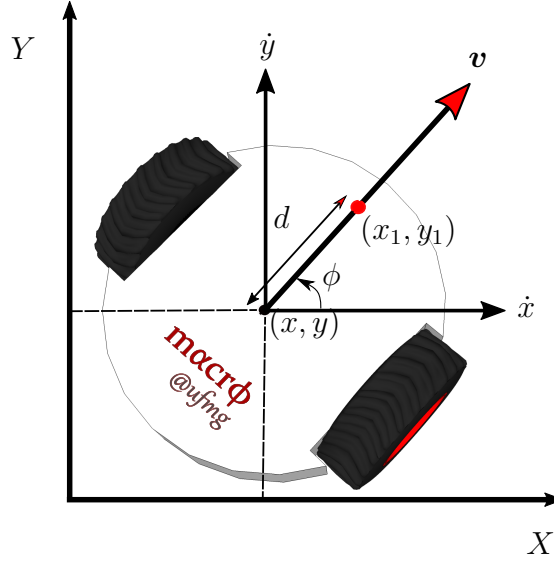


Figure B.2: Point offset used in the control of the nonholonomic mobile base.

The vector \mathbf{u}_b generated in the outer loop is used as reference for the input-output linearizing controller (Skaar, 2007) in the inner loop. Given a point (x_1, y_1) , as shown in Fig. B.2, its time derivative is given by

$$\dot{x}_1 = \dot{x} - (d \sin \phi) \dot{\phi}, \quad \dot{y}_1 = \dot{y} + (d \cos \phi) \dot{\phi}.$$

Let $u_1 \triangleq \dot{x}_1$, $u_2 \triangleq \dot{y}_1$, and $\dot{\phi} = (u_2 \cos \phi - u_1 \sin \phi) / d$, a feedback linearization is obtained (Skaar, 2007) and the linear controller

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \dot{x}_{1d} \\ \dot{y}_{1d} \end{bmatrix} + \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} x_{1d} - x_1 \\ y_{1d} - y_1 \end{bmatrix},$$

guarantees exponential convergence error for $k_1 > 0$ and $k_2 > 0$.