

macro@ufmg

Universidade Federal de Minas Gerais

Programa de Pós-Graduação em Engenharia Elétrica

MACRO Research Group - Mechatronics, Control and Robotics

# DYNAMIC MODELING OF ROBOTIC SYSTEMS: A DUAL QUATERNION FORMULATION

**Frederico Fernandes Afonso Silva**

Belo Horizonte, Brazil

2022

Frederico Fernandes Afonso Silva

**DYNAMIC MODELING OF ROBOTIC SYSTEMS:  
A DUAL QUATERNION FORMULATION**

Ph.D. Thesis submitted to the Programa de Pós-Graduação em Engenharia Elétrica of Escola de Engenharia at the Universidade Federal de Minas Gerais, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering.

**Advisor:** Bruno Vilhena Adorno

Belo Horizonte, Brazil

2022

S586d

Silva, Frederico Fernandes Afonso.

Dynamic modeling of robotic systems [recurso eletrônico]: a dual quaternion formulation / Frederico Fernandes Afonso Silva. - 2022.  
1 recurso online (134f. : il., color.) : pdf.

Orientador: Bruno Vilhena Adorno.

Tese (doutorado) - Universidade Federal de Minas Gerais,  
Escola de Engenharia.

Anexos: f. 134.

Apêndices: f. 105-121.

Bibliografia: f. 122-133.

Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia elétrica - Teses. 2. Quatérnios – Teses.  
3. Robótica – Teses. 4. Teoria dos grafos – Teses.  
I. Adorno, Bruno Vilhena. II. Universidade Federal de Minas Gerais.  
Escola de Engenharia. IV. Título.

CDU: 621.3(043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS  
ESCOLA DE ENGENHARIA  
COLEGIADO DO CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

### FOLHA DE APROVAÇÃO

"Dynamic Modeling Of Robotic Systems: A Dual Quaternion Formulation"

FREDERICO FERNANDES AFONSO SILVA

Tese de Doutorado defendida e aprovada, no dia 16 de junho de 2022, pela Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais constituída pelos seguintes professores:

Prof. Dr. Edson Roberto De Pieri - Departamento de Automação e Sistemas (UFSC)

Prof. Dr. João Yoshiyuki Ishihara - ENE (Universidade de Brasília)

Prof. Dr. Leonardo Antônio Borges Tôres - DELT (UFMG)

Prof. Dr. Guilherme Vianna Raffo - DELT (UFMG)

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

Belo Horizonte, 16 de junho de 2022.



Documento assinado eletronicamente por **Bruno Vilhena Adorno, Professor do Magistério Superior**, em 23/06/2022, às 09:22, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site [https://sei.ufmg.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **1533099** e o código CRC **3BF5F153**.

*To my mother, Isabel, who always  
encouraged me, and to the memory  
of my father, Flávio, who could not  
see this journey.*

# AGRADECIMENTOS

Em retrospectiva, só é possível se terminar o doutorado graças a todos que te ajudaram ao longo do caminho. Eu sou especialmente grato aos meus pais, Flávio e Isabel, que sempre entenderam a importância da educação e me incentivaram à vida acadêmica. Eu não estaria aqui sem vocês.

Eu também gostaria de expressar meus sinceros agradecimentos ao Prof. Bruno Adorno, que aceitou me orientar há mais de oito anos. A primeira reunião que tivemos foi quando eu estava procurando um orientador para meu projeto de fim curso durante a graduação. Ele começou me explicando o projeto e me perguntando se eu havia cursado as disciplinas relacionadas ou possuía os conhecimentos necessários, o que, para a maioria das questões, eu respondi que não. Em algum ponto de nossa reunião, eu já estava certo que ele não aceitaria me orientar. Ao invés disso, contudo, ele apenas me fez uma última pergunta: “você está interessado em participar do projeto?”. Eu respondi que sim, e ele girou em sua cadeira, abriu seu armário, pegou um livro (como eu o veria fazendo incontáveis vezes ao longo dos próximos anos) e me explicou o que eu precisaria aprender para o projeto. Todos que conhecem o Prof. Bruno Adorno sabem que sua ética e valores acadêmicos são um exemplo para todos que almejem se tornar bons pesquisadores. Hoje, no entanto, eu sou especialmente grato por ele ter acreditado em mim tantos anos atrás, e ter continuado a acreditar ao longo da minha vida acadêmica.

Um agradecimento especial vai para o Prof. Guilherme Raffo e Prof. Luciano Pimenta, modelos de profissionalismo e sempre dispostos a compartilhar seus conhecimentos e visões sobre a academia. Ao Prof. Guilherme Raffo, um agradecimento extra por sediar o churrasco anual do MACRO, um evento tão necessário ao fim de um ano cansativo de trabalho.

Eu também gostaria de agradecer aos funcionários da UFMG, que fizeram a minha vida na universidade muito mais fácil. Um agradecimento especial para Vera por manter o nosso laboratório em ordem, sempre sorrindo. Ao Jerônimo Coelho, por sua eficiência em todos os seus anos na secretaria do programa, meu sincero reconhecimento.

Eu também sou grato pelos amigos e colegas que compartilharam o laboratório comigo. Obrigado Ana Christina, Stella, Mariana (Mari), Rafael, Juan (Juancho), Marcos, Daniel, Brenner, Marcelo, Edson, Iuro, Gabriel (Enzo), Thales, e Jean por tornar nosso laboratório

---

um local de trabalho tão agradável. Obrigado também aos amigos dos laboratórios do corredor, Arturo, Marcus, Petrus e Wendy, e a uma amiga de um laboratório um pouco mais distante (só uns oito mil quilômetros), Ana Christine.

Por fim, mas igualmente importante, obrigado aos meus primos, Mateus e Cíntia, e a seus respectivos companheiros, Nathália e Ricardo, por sempre serem ótima companhia para cerveja aos fins de semana. Eu também sou grato pelo meu irmão, Lucas, que nunca falha em me fazer rir. Obrigado por me lembrarem que existe mais na vida do que a academia.

# ACKNOWLEDGMENTS

In retrospect, a PhD is only achievable thanks to the help of those who supported you along the way. I am especially grateful to my parents, Flávio and Isabel, for always understanding the importance of education and encouraging my academic life. I would not be here without you.

Also, I would like to express my deepest thanks to Prof. Bruno Adorno, who accepted the task of advising me over eight years ago. The first meeting we ever had was when I was looking for an adviser for my Bachelor's Thesis. He started by explaining a project he had while inquiring me if I had taken the related courses or had the necessary academic skills, which, for the most part, I answered that I did not. At some point during our meeting, I was sure he would not accept me as his student; instead, however, he just asked me one final question: "are you interested in working on this project?" I said yes, and he then spun his chair, opened his filing cabinet, grabbed a book, as I would see him doing countless times in the following years, and told me what I would need to learn for that project. For everyone who knows Prof. Bruno Adorno, it goes without saying that his ethics and academic values are a source of inspiration for anyone striving to become a good researcher. Today, nonetheless, I am most grateful that he believed in me so many years ago and continued to do so throughout my academic life.

A special thanks go to Prof. Guilherme Raffo and Prof. Luciano Pimenta, models of professionalism and always eager to share their knowledge and insights about academia. To Prof. Guilherme Raffo, extra tanks go for hosting the annual Brazilian barbecue of MACRO, a much-needed break at the end of every year.

I am would also like to thank the UFMG's staff for making my life at the university so much easier. A special thanks to Vera for keeping our laboratory in order, always with a smile on her face. A special acknowledgment goes to Jerônimo Coelho for his efficiency during his years as secretary of PPGEE.

I am also grateful for the colleagues and friends that shared our laboratory with me. Thank you to Ana Christina, Stella, Mariana (Mari), Rafael, Juan (Juancho), Marcos, Daniel, Brenner, Marcelo, Edson, Iuro, Gabriel (Enzo), Thales, and Jean for keeping the laboratory this great place to work in. Thanks also to the friends of the neighboring laboratories, Arturo, Marcus, Petrus and Wendy, and to my friend of a not-so-neighboring



---

laboratory (just some eight thousand kilometers away), Ana Christine.

Last but equally important, thanks to my cousins Mateus and Cíntia and their respective significant others, Nathália and Ricardo, for always being good company for a beer. I am also grateful for my brother, Lucas, who never fails to make me laugh. Thank you for reminding me that there is more in life than academia.

# RESUMO

Essa tese propõe uma técnica para a modelagem dinâmica de robôs seriais e ramificados utilizando álgebra de quatérnios duais. O modelo considera tanto todas as juntas do tipo *lower-pair kinematic* quanto juntas de seis graus de liberdade e, adicionalmente, o procedimento permite a composição modular sistemática de modelos dinâmicos compostos de múltiplos subsistemas, cada um deles, por sua vez, composto de diversos corpos rígidos. A estratégia proposta é aplicável ainda que alguns subsistemas se comportem como caixas pretas, exigindo apenas os heligiros e as heliforças do ponto de conexão entre eles. Para auxiliar na composição de modelos, é também proposta uma representação em grafos que codifica a propagação de heligiros e heliforças dentre os subsistemas. As heliforças das juntas são resultado do cálculo da matriz de interconexão do grafo, tornando intuitivo o procedimento de modelagem. O formalismo proposto foi validado utilizando manipuladores robóticos de 6-DoF e 50-DoF, um manipulador móvel de base holonômica de 9-DoF e um robô ramificado de 38-DoF, composto de 9 subsistemas. Os resultados foram comparados com as bibliotecas Robotics Toolbox, desenvolvida pelo Peter Corke, e Spatial V2, desenvolvida pelo Roy Featherstone, além do simulador V-REP/CoppeliaSim, demonstrando que o método proposto é tão preciso quanto as bibliotecas do estado da arte.

# ABSTRACT

This thesis proposes a technique for the dynamic modeling of serial and branched robots using dual quaternion algebra. The modeling accounts for all lower-pair kinematic joints and six-degree-of-freedom joints, and the framework enables the systematic modular composition of dynamic models comprising several subsystems, each, in turn, composed of multiple rigid bodies. The proposed strategy is applicable even if some subsystems are regarded as black boxes, requiring only the twists and wrenches at the connection points between different subsystems. To help in the model composition, a unified graph representation that encodes the propagation of twists and wrenches between the subsystems is also proposed. The joint wrenches result from the calculation of the interconnection matrix of the graph, making the modeling procedure straightforward. The framework was validated using serial manipulators of 6-DoF and 50-DoF, a 9-DoF holonomic mobile manipulator, and a 38-DoF branched robot composed of 9 subsystems. The results were compared with Peter Corke's Robotics Toolbox, Roy Featherstone's Spatial V2, and the robot simulator V-REP/CoppeliaSim, demonstrating that the proposed formalism is as accurate as state-of-the-art libraries.

## CONTENTS

---

<i>List of Figures</i>	xiv
<i>List of Tables</i>	xv
<i>Acronyms</i>	xvi
<i>Notation and Conventions</i>	xvii
1 INTRODUCTION	1
2 LITERATURE REVIEW	5
2.1 Representations used in robot dynamic modeling . . . . .	6
2.1.1 Classic representation using $\mathbb{R}^3$ and $SO(3)$ . . . . .	7
2.1.2 Featherstone's spatial algebra . . . . .	8
2.1.3 Lie algebra associated to $SE(3)$ . . . . .	9
2.1.4 Lie algebra associated to $Spin(3) \times \mathbb{R}^3$ . . . . .	11
2.2 State of the art in dual quaternion dynamic modeling . . . . .	12
2.3 Topological graph and modular dynamic modeling . . . . .	17
2.4 Conclusions . . . . .	19
3 MATHEMATICAL BACKGROUND	22
3.1 Classic Newton-Euler formulation . . . . .	22
3.2 Classic Euler-Lagrange formulation . . . . .	24
3.2.1 Canonical Euler-Lagrange equation from the recursive Newton-Euler formulation . . . . .	25
3.3 Dual quaternion algebra . . . . .	25
3.3.1 Quaternions . . . . .	25
3.3.2 Dual Quaternions . . . . .	27
3.3.2.1 Unit dual quaternions . . . . .	28
3.3.3 Forward Kinematic Model . . . . .	29
3.3.3.1 Differential forward kinematic model of serially coupled kinematic structures . . . . .	30
3.4 Conclusions . . . . .	32
4 THE DUAL QUATERNION NEWTON-EULER FORMULATION	33
4.1 The inertia tensor . . . . .	33
4.1.1 The quaternionic inertia tensor . . . . .	35

4.2	Dynamic modeling of serial kinematic chains . . . . .	38
4.2.1	Fixed-base robot manipulators . . . . .	38
4.2.1.1	Forward Recursion . . . . .	39
4.2.1.2	Backward Recursion . . . . .	42
4.2.1.3	Summary . . . . .	44
4.2.1.4	Cost comparison . . . . .	46
4.2.1.5	Simulations and discussions . . . . .	48
4.2.2	Adding more rigid bodies to the kinematic chain . . . . .	53
4.2.3	Mobile robot manipulators . . . . .	56
4.2.3.1	Holonomic mobile manipulators . . . . .	57
4.2.3.2	Simulations and discussions . . . . .	57
4.3	Dynamic modeling of branched robots . . . . .	60
4.4	Conclusions . . . . .	65
5	MODULAR MODEL COMPOSITION . . . . .	67
5.1	The dual quaternion modular model composition strategy . . . . .	71
5.2	Graph representation . . . . .	76
5.3	Simulations and discussions . . . . .	80
5.4	Conclusions . . . . .	82
6	CONCLUSION AND FUTURE WORKS . . . . .	84
	APPENDICES . . . . .	86
A	DIFFERENTIAL-DRIVE MOBILE BASES . . . . .	87
A.1	Simulations and discussions . . . . .	89
B	DYNAMIC CONTROL . . . . .	92
B.1	Wrench control . . . . .	93
B.1.1	End-effector position control based on potential fields . . . . .	94
B.1.2	Simulations and discussions . . . . .	95
C	DUAL QUATERNION EULER-LAGRANGE FORMULATION . . . . .	97
C.1	Simulations and discussions . . . . .	98
D	SIMULATION DETAILS . . . . .	99
	<i>Bibliography</i> . . . . .	103
	<i>Publications</i> . . . . .	115

LIST OF FIGURES

---

2.1	Atlas performing a backflip. . . . .	6
2.2	Examples of tasks that require dynamic models. . . . .	7
2.3	The dual angle. . . . .	12
3.1	Three-dimensional position represented by a pure quaternion. . . . .	26
3.2	Axis-angle rotation represented by a rotation quaternion. . . . .	26
3.3	Sequence of rigid transformations represented by unit dual quaternions. . .	28
3.4	An $n$ -DoF serial manipulator. . . . .	31
4.1	An $n$ -DoF serial manipulator. . . . .	38
4.2	Twist generated on the mass center of a link due to the application of an angular rotation at its reference frame. . . . .	40
4.3	The 6-DoF JACO robotic arm in the robot simulator V-REP. . . . .	49
4.4	Joint torque waveforms of the 6-DoF JACO robotic arm. . . . .	50
4.5	A 50-DoF serial manipulator in the robot simulator V-REP. . . . .	51
4.6	Joint acceleration waveforms of the 50-DoF robotic manipulator. . . . .	53
4.7	A 2-DoF serial manipulator. . . . .	55
4.8	A 3-DoF serial manipulator. . . . .	56
4.9	X-Terrabot, a holonomic mobile manipulator. . . . .	57
4.10	9-DoF holonomic mobile manipulator in the robot simulator V-REP. . . .	58
4.11	Generalized acceleration waveforms of the 9-DoF mobile manipulator. . . .	59
4.12	A 6-DoF branched robot. . . . .	60
4.13	Simplified upper body of the Poppy Humanoid robot. . . . .	62
4.14	Dual quaternion formalism applied to an open kinematic tree. . . . .	64
5.1	Two serial kinematic chains with $n_1$ and $n_2$ degrees of freedom. . . . .	68
5.2	Wrenches generated at the joints of the $i$ -th subsystem. . . . .	73
5.3	Graph representation of the $(n_1 + n_2)$ -DoF assembled branched mechanism. .	76
5.4	Three serial kinematic chains with $n_1$ , $n_2$ , and $n_3$ degrees of freedom. . . .	79
5.5	Graph representation of the new $(n_1 + n_2 + n_3)$ -DoF assembled mechanism. .	79
5.6	A 38-DoF branched robot in the robot simulator V-REP. . . . .	80
5.7	Graph representation of the 38-DoF branched robot. . . . .	81
5.8	Joint torque waveforms of the 38-DoF branched robot. . . . .	82
A.1	Little John, a nonholonomic mobile manipulator. . . . .	87

A.2	Representation of a differential-drive mobile base. . . . .	88
A.3	Differential-drive mobile robot Pioneer in the robot simulator V-REP. . . . .	90
A.4	Generalized force waveforms of the differential-drive mobile robot Pioneer. . . . .	91
B.1	Norm of the end-effector position error of the 6-DOF JACO robotic arm. . . . .	96
B.2	Input joint torques of the 6-DOF JACO robotic arm. . . . .	96
D.1	Unfiltered joint velocity, acceleration, and torque waveforms of the third joint of the 6-DoF JACO robotic. . . . .	100
D.2	Filtered joint velocity and acceleration and unfiltered torque waveforms of the third joint of the 6-DoF JACO robotic. . . . .	102

## LIST OF TABLES

---

4.1	Twists of some of the most commonly used joints in robotics. . . . .	39
4.2	Cost of operations with quaternions and dual quaternions in terms of multiplication and addition of real numbers. . . . .	46
4.3	Number of operations in the computation of $\underline{\zeta}_{0,c_i}^{c_i}$ . . . . .	47
4.4	Number of operations in different parts of the dual quaternion Newton-Euler algorithm. . . . .	47
4.5	Cost comparison between the proposed method and its classic counterpart for obtaining the dynamical model for an $n$ -DoF serial robot. . . . .	48
4.6	CMC between the joint torque waveforms of the 6-DoF JACO robotic arm. . . . .	51
4.7	CMC between the joint acceleration waveforms of the 50-DoF serial manipulator. . . . .	52
4.8	CMC between the generalized acceleration waveforms of the 9-DoF holo- nomic mobile manipulator. . . . .	59
5.1	CMC between the joint torque waveforms of the 38-DoF branched robot. . . . .	82
A.1	CMC between the generalized force waveforms of the differential-drive mobile robot Pioneer. . . . .	90
C.1	The mean and the variance of the error between the joint wrenches given by the dqEL and the dqNE for the 7-DOF KUKA LWR robotic arm. . . . .	98

# ACRONYMS

ASIMoV	Advanced Systems with Intelligence Mobility and Vision.
CMC	Coefficient of multiple correlation.
CoM	Center of mass.
D-H	Denavit-Hartenberg.
DoF	Degrees of freedom.
dqGP	Dual quaternion Gauss's Principle of Least Constraint.
dqNE	Dual quaternion Newton-Euler.
HTM	Homogeneous transformation matrices.
KUKA	Keller und Knappich Augsburg.
LWR	Lightweight Robot.
MACRO	Mechatronics, Control and Robotics.
rtNE	Robotic Toolbox's Newton-Euler.
Sv2	Spatial V2.
UAV	Unmanned aerial vehicle.



# NOTATION AND CONVENTIONS

$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$	Coriolis and Centrifugal matrix.
$\mathbb{F}^n$	Set of $n$ -dimensional spatial force vectors.
$\mathbf{g}(\mathbf{q})$	Gravity vector.
$\mathbb{H}$	Set of quaternions.
$\mathbb{H}_p$	Set of pure quaternions.
$\mathcal{H}$	Set of dual quaternions.
$\mathcal{H}_p$	Set of pure dual quaternions.
$\bar{\mathbb{I}}$	Three-dimensional inertia tensor.
$\mathbb{I}$	Quaternionic inertia tensor.
$\mathbf{I}_n$	Identity matrix of dimension $n \times n$ .
$\hat{i}, \hat{j}, \hat{k}$	Imaginary units.
$\mathbb{M}^n$	Set of $n$ -dimensional spatial motion vectors.
$\mathbf{M}(\mathbf{q})$	Inertia matrix.
$\mathbb{R}$	Set of real numbers.
$\mathbb{S}^3$	Set of unit-norm quaternions.
$\underline{\mathbf{S}}$	Set of unit-norm dual quaternions.
$SE(n)$	Special Euclidean group of dimension $n$ .
$se(n)$	Lie algebra associated to $SE(n)$ .
$SO(n)$	Special orthogonal group of dimension $n$ .
$so(n)$	Lie algebra associated to $SO(n)$ .

$\underline{\boldsymbol{x}}^*$	Conjugate of dual quaternion $\underline{\boldsymbol{x}}$ .
$\dot{x}$	Time derivative of $x$ .
$\varepsilon$	Dual unit.
$\underline{\boldsymbol{\Gamma}}$	Vector of dual quaternion wrenches.
$\underline{\boldsymbol{\zeta}}$	Pure dual quaternion representing a wrench.
$\underline{\boldsymbol{\Xi}}$	Vector of dual quaternion twists.
$\underline{\boldsymbol{\xi}}$	Pure dual quaternion representing a twist.
$\boldsymbol{\omega}$	Pure quaternion representing an angular velocity.

Lowercase bold letters represent column vectors or quaternions:

$$\boldsymbol{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}, \quad \boldsymbol{a} \in \mathbb{R}^n$$

or

$$\boldsymbol{b} = b_1 + \hat{i}b_2 + \hat{j}b_3 + \hat{k}b_4, \quad \boldsymbol{b} \in \mathbb{H}.$$

Uppercase bold letters represent matrices:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{bmatrix}, \quad \boldsymbol{A} \in \mathbb{R}^{n \times m}.$$

Underlined variables represent dual numbers:

$$\underline{a} = a_1 + \varepsilon a_2, \quad \underline{a} \in \mathcal{H},$$

whereas

$$\underline{\boldsymbol{h}} = \boldsymbol{h}_1 + \varepsilon \boldsymbol{h}_2, \quad \underline{\boldsymbol{h}} \in \mathcal{H}$$

is a dual quaternion.

Uppercase underlined variables represent column vectors or matrices of dual quaternions:

$$\underline{\boldsymbol{\Gamma}} = \begin{bmatrix} \underline{\boldsymbol{\zeta}}_1 \\ \vdots \\ \underline{\boldsymbol{\zeta}}_n \end{bmatrix}, \quad \underline{\boldsymbol{\Gamma}} \in \mathcal{H}_p^n,$$

or

$$\underline{\mathbf{A}} = \begin{bmatrix} \underline{\mathbf{a}}_{11} & \cdots & \underline{\mathbf{a}}_{1m} \\ \vdots & \ddots & \vdots \\ \underline{\mathbf{a}}_{n1} & \cdots & \underline{\mathbf{a}}_{nm} \end{bmatrix}, \quad \underline{\mathbf{A}} \in \mathcal{H}_p^{n \times m}.$$

Variables with hats represent spatial vectors. For example,

$$\hat{\mathbf{v}} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \\ v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega} \\ \mathbf{v} \end{bmatrix}$$

is a spatial velocity, in which  $\boldsymbol{\omega}$  and  $\mathbf{v}$  are, respectively, the angular and linear velocities.

Variables with upside down hats represent elements in the Lie algebras associated to  $SO(3)$  and to  $SE(3)$ :

$$\check{\mathbf{E}} = \begin{bmatrix} \check{\boldsymbol{\omega}} & \mathbf{v} \\ \mathbf{0}_{1 \times 3} & 0 \end{bmatrix}, \quad \in \mathbb{R}^{4 \times 4},$$

where  $\check{\mathbf{E}} \in se(3)$ ,  $\check{\boldsymbol{\omega}} \in so(3)$ , and  $\mathbf{0}_{1 \times 3} \in \mathbb{R}^{1 \times 3}$  is a vector of zeros.

Subscripts and superscripts are used to indicate the reference frames of the variables. For instance,  $\boldsymbol{\omega}_{a,b}^b$  represents the angular velocity of frame  $\mathcal{F}_b$  relative to  $\mathcal{F}_a$ , expressed on frame  $\mathcal{F}_b$ . If no superscript is used, it is assumed the global inertial reference frame.

# 1

## INTRODUCTION

When asked to describe what a robot is, people will often answer something on the lines of a human-like machine, capable of navigating and interacting with objects from our usual environment and performing complex tasks without the need for human intervention. This view present in popular imagination conveys a crucial aspect of robotics; the complexity of a robotic system goes far beyond the complexity of the mechanism itself. A typical robotic system involves motion/force/impedance control, path planning, task planning, and many more higher-level layers. Alas, representations that are very useful for robot modeling, such as homogeneous transformation matrices, are not necessarily easy to use when performing pose control or impedance control, for example. That is precisely the reason why it is common to use homogeneous transformation matrices to obtain the robot kinematics but then indirectly find the geometric Jacobian and, finally, to use quaternions and position vectors to perform pose control in the task-space (Yuan, 1988; Xian et al., 2004). There are several drawbacks to using the aforementioned strategy. The mix of different representations unnecessarily complicates the overall description of the system, and the mapping between them usually introduces mathematical artifacts, such as algorithmic singularities and discontinuities (Silva et al., 2022).

Those are some of the reasons why, in the last thirty years, there have been an expressive amount of papers—published in prestigious journals and conferences—dealing with different representations for robot modeling. Notorious examples can be found in the works of Featherstone (2008, 2010a,b), McCarthy (1990), Dooley & McCarthy (1991), Perez & McCarthy (2004), Selig (2004, 2005), and Selig & Bayro-Corrochano (2010),

among many others.

Amongst the different representations, dual quaternion algebra possesses elements that have strong geometrical meaning, such as in screw theory, and are also represented as coupled entities within single elements. In kinematics, this representation has been extensively explored to obtain the robot kinematics and differential kinematics (Perez & McCarthy, 2004; Adorno, 2011; Gouasmi, 2012; Cohen & Shoham, 2016; Özgür & Mezouar, 2016; Kong, 2017; Dantam, 2020). Furthermore, in recent works, dual quaternions have been used to perform:

- admittance control (Fonseca et al., 2020), which is fundamental in physical human-robot interaction;
- constrained motion control (Marinho et al., 2019; Quiroz-Omana & Adorno, 2019), which takes into account geometrical constraints imposed by the workspace;
- hybrid control, which takes into account the topology of the space of rigid motions (Kussaba et al., 2017) and optimal control, which uses a linear-quadratic optimal tracking controller for robotic manipulators (Marinho et al., 2015);
- distributed pose formation control (Savino et al., 2020) and cooperative manipulation (Adorno et al., 2010; Figueredo et al., 2014), including the ones that involve human-robot collaboration (Adorno et al., 2015);
- and to define high-level geometrical tasks (Lana et al., 2015).

Moreover, elements such as unit dual quaternions and pure dual quaternions, when equipped with standard multiplication and addition operations, form Lie groups with associated Lie algebras. Therefore, a formulation based on dual quaternion algebra offers the geometrical insights of screw theory, the rigor of Lie Algebra, and a simple algebraic treatment of the dynamical model as present in the spatial algebra (Featherstone, 2008). Hence, often reducing the necessity of an extensive geometric analysis of the mechanism, which contrasts with approaches based on the representation of screw theory using matrices (Huang et al., 2015; Renda et al., 2017).

Concerning dynamics, although several works have addressed the study of rigid body dynamics using dual quaternions over the last decades—including modeling (Baklouti & Castelain, 1993; Pennock & Meehan, 2000; Brodsky & Shoham, 1999), estimation (Zu et al., 2018), and control (Wang & Yu, 2010; Zhang et al., 2010; Zhang & Duan, 2011; Filipe & Tsiotras, 2013; Abaunza et al., 2016; Gui & Vukovich, 2016)—the literature still lacks a concise and systematic methodology to attain the dynamic model of robotic systems using dual quaternion algebra. That is even more evident when considering robots with more general joint types (e.g., helical, cylindrical, 6-DoF, etc.) and branched robots (e.g., humanoid robots).

In conclusion, since there are no adequately general methodology to obtain robot dynamic models based on dual quaternion algebra, there is still a theoretical gap that creates the unnecessary need for intermediate mappings when using higher-level algorithms based on dual quaternions to connect them to the low-level dynamic model. The purpose of this thesis is to fill that gap by proposing suitable dynamic models of robotic systems using dual quaternion algebra.

## Contributions

The contributions of this thesis are summarized into the following groups:

1. A novel quaternionic inertia tensor that enables the description of rigid body dynamics exclusively with operations inside the dual quaternion algebra is proposed and used to develop a technique for the dynamic modeling of serial and branched robots that accounts for all lower-pair kinematic joints and six-degrees-of-freedom joints. Thus, extending the systematic application of dual quaternions from kinematic (Adorno, 2011) to dynamic modeling.
2. A systematic modular composition of dynamic models comprising several subsystems, each in turn composed of multiple rigid bodies, is developed. The proposed strategy is applicable even if the complete information of the subsystems' dynamics is unavailable, requiring only the twists and wrenches at the connection points between different subsystems. Moreover, a unified graph representation that encodes the propagation of twists and wrenches between the subsystems aids the model composition. The joint wrenches result from the calculation of the interconnection matrix of the graph, making the modeling procedure straightforward.

## Organization of the thesis

Chapter 2 presents a literature review on different algebras used in the dynamic modeling of multibody systems to motivate the choice of dual quaternions. Furthermore, it assesses the state of the art of dynamic modeling using dual quaternion algebra and strategies based on topological graphs and modular dynamic modeling to contextualize the contributions of this thesis.

Chapter 3 gives a brief revision of the classic recursive Newton-Euler formulation and Euler-Lagrange canonical equation, typically presented on robotics textbooks, and also provides an introduction to dual quaternion algebra. More notably, it reviews the dual quaternion forward kinematic model proposed by Adorno (2011) alongside his strategy for obtaining the differential forward kinematic model of serially coupled kinematic structures, both of which motivated several ideas presented in this thesis.

Chapter 4 proposes a novel quaternionic inertia tensor that allows the description of the dynamic properties of rigid bodies exclusively with operations in the dual quaternion

algebra. This chapter also proposes the dual quaternion Newton-Euler formalism, a systematic strategy for the dynamic modeling of branched robots, which includes fixed-base and mobile manipulators as particular cases, and compares its cost in terms of the number of elementary operations with the classic Newton-Euler recursive algorithm.

Chapter 5 proposes a modular composition methodology and a graph representation of the robotic system that enables its assembling starting from intermediate subsystems, even if the complete information of their dynamics is unavailable, requiring only the twists and wrenches at the connection points between different subsystems.

Chapter 6 presents a final discussion and future works.

# 2

## LITERATURE REVIEW

For most of their existence, robots were primarily designed to work in controlled environments, performing well defined—and repetitive—tasks. Nowadays, however, rising fields of interest on research projects include both more complex mechanisms, such as humanoid robots (Sentis & Khatib, 2006; Schaal, 2019), and more complex tasks, ranging from general human-robot interaction (Goodrich & Schultz, 2007) to medical applications (Chen et al., 2013; Marinho et al., 2019). Regardless of the changes in the work environment and the diversification of the tasks, most of the control techniques remain model-dependent (Siciliano & Khatib, 2008). Therefore, a project in robotics often starts with determining a suitable model for the robot.

Robot models are commonly divided into kinematic and dynamic models. Kinematic models are concerned with the movements of the robot disregarding forces or torques acting on the system. When the robot is rigid, has small inertia and the application does not demand relatively high accelerations, the kinematic model is a well-suited choice. Despite how restrictive these prerequisites may sound, kinematic models have been used in several works employing both mobile manipulators (De Luca et al., 2006; Zhang et al., 2012; Quiroz-Omaña & Adorno, 2017; Silva & Adorno, 2018) and humanoids (Nishiwaki et al., 2005; Gienger et al., 2006; Park & Lee, 2013), even in tasks involving human-robot interactions (Adorno, 2011; Adorno et al., 2011b,a, 2015; Lana et al., 2013; Lasota et al., 2014).

Some tasks, however, demand the consideration of forces and torques present in the activity. For instance, the backflip performed by the Boston Dynamics' Atlas, depicted in



figure 2.1, could not be achieved if only the robot kinematics was used. Other applications that require the robot's dynamic model are dynamic gait (Ott & Hyon, 2019; Reher et al., 2019) and tasks involving unmanned aerial vehicles (UAVs) (Abaunza et al., 2016; Mello et al., 2016; Rego et al., 2016; Zu et al., 2018) (*see* figure 2.2).

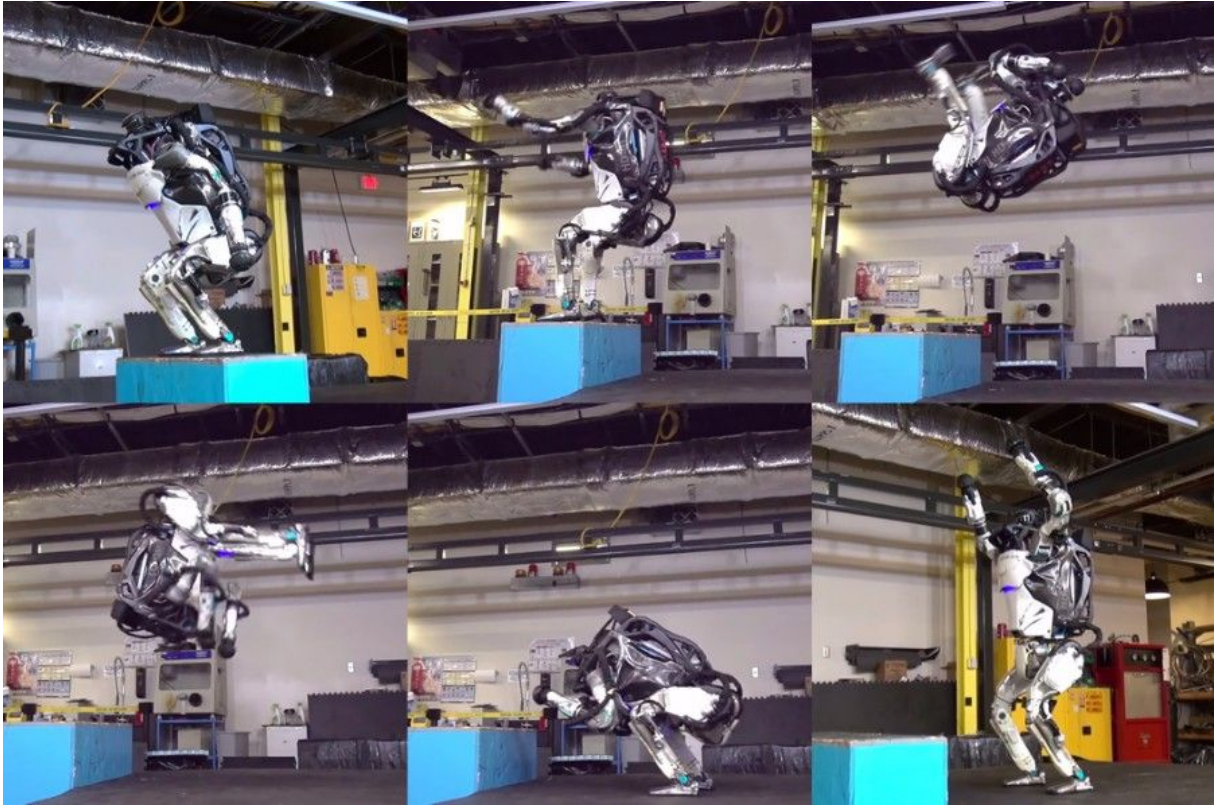


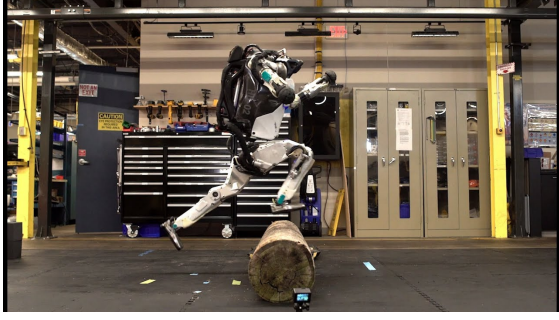
Figure 2.1: Atlas performing a backflip (courtesy of Boston Dynamics).

Similarly to how the choice of a proper model is crucial for the success of the application, so is the choice of an adequate mathematical representation for it. An ill-chosen model representation can inadvertently overcomplicate the control design and, at the worst, even render it unfeasible. For this reason, several different representations have been proposed throughout the history of robotics, and textbooks customarily dedicate their initial chapters to the study of this subject (Spong et al., 2006; Siciliano & Khatib, 2008; Siciliano et al., 2009; Corke, 2011).

In this context, the next section presents a brief review of the representations most commonly used by roboticists for the dynamic modeling of robotic systems.

## 2.1 Representations used in robot dynamic modeling

Representations used in robot dynamic modeling can be categorized into two main groups. The group of those that represent linear and angular velocities and accelerations, and forces and torques, as independent elements, whose relations are given in different equations; and



(a) Atlas jumping over a log (courtesy of Boston Dynamics). (b) UAV delivering a product (courtesy of Amazon).

Figure 2.2: Tasks that require dynamic models. Figure 2.2a presents the robot Atlas jumping over a log, whereas figure 2.2b shows a delivery being performed by a UAV.

the group of those that couple linear and angular components into a single entity, thus presenting their interactions in a single expression. A reader that has taken introductory robotics courses is probably familiarized with the former and may even tend to see it as the most intuitive representation since it often demands only basic knowledge of vector algebra. Nonetheless, a coupled representation can lead to high-level abstractions that aid the development of more general formulations for the dynamic modeling of robotic systems. To further clarify this discussion, the remaining of this section presents four of the most commonly used representations in robotics.

### 2.1.1 Classic representation using $\mathbb{R}^3$ and $SO(3)$

One of the most used dynamic model representations in robotics employs three-dimensional vectors to represent the linear and angular variables and rotation matrices to perform transformations between different reference frames. Take, for instance, the equations of the forces  $\mathbf{f}_i^{i-1} \in \mathbb{R}^3$  and the torques  $\boldsymbol{\tau}_i^{i-1} \in \mathbb{R}^3$  of the  $i$ -th link of an  $n$ -DoF serial manipulator with revolute joints found in the classic recursive Newton-Euler algorithm<sup>1</sup> (Spong et al., 2006, p. 277), given by

$$\begin{aligned}\mathbf{f}_i^{i-1} &= \mathbf{R}_i^{i-1} \mathbf{f}_{i+1}^i + m_i (\mathbf{a}_{0,c_i}^{i-1} + \mathbf{g}^{i-1}), \\ \boldsymbol{\tau}_i^{i-1} &= \mathbf{R}_i^{i-1} \boldsymbol{\tau}_{i+1}^i - \mathbf{f}_i^{i-1} \times \mathbf{p}_{i-1,c_i}^{i-1} + (\mathbf{R}_i^{i-1} \mathbf{f}_{i+1}^i) \times \mathbf{p}_{i,c_i}^{i-1} + \bar{\mathbb{I}}_i^{i-1} \dot{\boldsymbol{\omega}}_{0,c_i}^{i-1} + \boldsymbol{\omega}_{0,c_i}^{i-1} \times (\bar{\mathbb{I}}_i^{i-1} \boldsymbol{\omega}_{0,c_i}^{i-1}),\end{aligned}\tag{2.1}$$

<sup>1</sup>Remember that, as defined in the section of notations (page xix), subscripts and superscripts are used to indicate the reference frames of the variables. For instance,  $\mathbf{a}_{0,c_i}^{i-1}$  represents the linear acceleration of frame  $\mathcal{F}_{c_i}$  relative to  $\mathcal{F}_0$ , expressed on frame  $\mathcal{F}_{i-1}$ . If no superscript is used, it is assumed the global inertial reference frame.

where  $\mathbf{R}_i^{i-1} \in SO(3)$  is the rotation matrix representing the rotation from frame  $\mathcal{F}_{i-1}$  at the end of the  $(i-1)$ -th link to  $\mathcal{F}_i$  at the end of the  $i$ -th link,  $m_i \in (0, \infty)$  and  $\bar{\mathbb{I}}_i^{i-1} \in \mathbb{R}^{3 \times 3}$ , with  $\bar{\mathbb{I}}_i^{i-1} > 0$ , are, respectively, the mass and the inertia tensor of the  $i$ -th link,  $\mathbf{a}_{0,c_i}^{i-1} \in \mathbb{R}^3$  and  $\boldsymbol{\omega}_{0,c_i}^{i-1} \in \mathbb{R}^3$  are the linear and angular acceleration of the center of mass (CoM), respectively,  $\mathbf{g}^{i-1} \in \mathbb{R}^3$  is the gravity acceleration, and  $\mathbf{p}_{i-1,c_i}^{i-1} \in \mathbb{R}^3$  is the position of the  $i$ -th CoM with respect to  $\mathcal{F}_{i-1}$ .

It is evident from (2.1) that the total torque appearing on the body is affected by the lever arms produced by the forces acting on it. Nonetheless, forces and torques are presented in different equations, thus demanding careful geometrical analysis of the mechanism to model it correctly. For fixed-base robotic manipulators with revolute and prismatic joints, the final expressions obtained through this analysis are well known and readily available in different robotics textbooks (Selig, 2005; Siciliano et al., 2009) and conventional computational libraries (Corke, 2017). For more general and complex mechanisms, such as humanoid robots or differential-drive mobile manipulators, however, the roboticist often has to be the one performing this geometrical analysis. Such a task is both time-consuming and prone to errors, as one can easily neglect a lever arm or some influence of the angular velocity on the linear acceleration. Furthermore, when considering joint types other than revolute and prismatic (e.g., helical, cylindrical, 6-DoF, etc.), and conventions other than the Denavit-Hartenberg convention, this vectorial representation leads to convoluted equations of forces and torques. The aforementioned drawback is evident in the following comment from Siciliano and Khatib about the classic Recursive Newton-Euler Algorithm (RNEA) based on three-dimensional vector representation (given by algorithm 3.3 in their book), (Siciliano & Khatib, 2008, p. 54)

*The original version of the RNEA was developed and expressed using 3-D vectors (e.g. [3.2, 4]). Algorithm 3.3 shows a special case of this algorithm, in which the joints are assumed to be revolute, and the joint axes are assumed to coincide with the z axes of the link coordinate systems. (**Without these assumptions, the equations would be a lot longer.**) It also assumes that the external forces are zero.*

In conclusion, although well established and intuitive, the disconnection of linear and angular components in the classic representation based on  $\mathbb{R}^3$  and  $SO(3)$  is the prime source of its drawbacks. Thus, the remainder of this section presents representations that couple those entities into a single element.

### 2.1.2 Featherstone's spatial algebra

Featherstone (2008) proposes one alternative to deal with the previously presented problems replacing the three-dimensional vectors by spatial vectors, which are six-dimensional vectors composed by the concatenation of the angular and linear components acting on

the body. For instance, the classic representation of linear and angular velocities as two three-dimensional vectors

$$\mathbf{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix},$$

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix},$$

is replaced by the spatial velocity

$$\hat{\mathbf{v}} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \\ v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega} \\ \mathbf{v} \end{bmatrix} \in \mathbf{M}^6, \quad (2.2)$$

where  $\mathbf{M}^6$  is the set of the six-dimensional spatial motion vectors.

Featherstone proposed several operators to deal with rigid transformations and deduced the most common algorithms of rigid body dynamics in spatial vector form (Featherstone, 2008), originating reliable and efficient implementations (Carpentier et al., 2019) that have been tested on actual complex robotic platforms (Bouyarmane et al., 2019; Kleff et al., 2021; Sleiman et al., 2021; Ramuzat et al., 2022). Nonetheless, the spatial algebra relies majorly on standard matrix algebra and lacks the algebraic formalism of more thorough approaches, such as the ones based on Lie algebra (Featherstone, 2010a, p. 94).

### 2.1.3 Lie algebra associated to $SE(3)$

One solution towards a more rigorous and elegant algebraic treatment of rigid body motions, while still using matrices, is to introduce an exponential mapping to  $SE(3)$ , such as the one presented by Murray et al. (1994). Alternatively to the conventional rotation matrix, the rotation of a body by  $\theta$  radians around an unit rotation axis  $\boldsymbol{\phi} = [\phi_x \ \phi_y \ \phi_z]^T$  may be described in the exponential form as

$$\mathbf{R}(\boldsymbol{\phi}, \theta) = e^{\check{\boldsymbol{\omega}}\theta}, \quad (2.3)$$

where

$$e^{\check{\boldsymbol{\omega}}\theta} = \mathbf{I} + \check{\boldsymbol{\omega}} \sin(\theta) + \check{\boldsymbol{\omega}}^2 (1 - \cos(\theta)),$$

with  $\mathbf{I} \in \mathbb{R}^{3 \times 3}$  being the identity matrix. The skew-symmetric matrix  $\check{\omega} \in so(3)$  is an element of the Lie algebra associated to  $SO(3)$ , given by (Murray et al., 1994, p. 27)

$$\check{\omega} = \mathbf{S}(\boldsymbol{\phi}) = \begin{bmatrix} 0 & -\phi_z & \phi_y \\ \phi_z & 0 & -\phi_x \\ -\phi_y & \phi_x & 0 \end{bmatrix} \in so(3), \quad (2.4)$$

where  $\boldsymbol{\phi}$  is a unit rotation axis.

Accordingly, elements of the Lie algebra  $se(3)$  associated to  $SE(3)$  are given, in homogeneous coordinates, by

$$\check{\Xi} = \begin{bmatrix} \check{\omega} & \mathbf{v} \\ \mathbf{0}_{1 \times 3} & 0 \end{bmatrix} \in se(3), \quad (2.5)$$

where  $\mathbf{0}_{1 \times 3} \in \mathbb{R}^{1 \times 3}$  is a row vector of zeros and  $\mathbf{v} = -\boldsymbol{\phi} \times \mathbf{t}$ , in which  $\mathbf{t} \in \mathbb{R}^3$  is a point on the axis of rotation. The four-by-four matrix  $\check{\Xi} \in se(3)$  is the *twist* (Murray et al., 1994, p. 39) and, analogously to (2.3), is used to obtain the exponential mapping of  $SE(3)$ , whose homogeneous coordinates representation is given by (Murray et al., 1994, p. 46)

$$e^{\check{\Xi}\theta} = \begin{bmatrix} e^{\check{\omega}\theta} & (\mathbf{I} - e^{\check{\omega}\theta})(\boldsymbol{\phi} \times \mathbf{v}) + \boldsymbol{\phi}\boldsymbol{\phi}^T \mathbf{v}\theta \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}. \quad (2.6)$$

As demonstrated by Murray et al. (1994), every rigid transformation can be written as the exponential of some twist, in the form of (2.6). Also, when changing the reference frames of twists, and wrenches,<sup>2</sup> the corresponding adjoint transformation ensures that all the appropriated lever arms are algebraically found. Nonetheless, this choice of representation has its downsides. For instance, a movement consisting of a rotation of  $\theta \in \mathbb{R}$  radians about an axis  $\boldsymbol{\phi} \in \mathbb{R}^3$ , here no longer considered unitary, followed by a translation  $d \in \mathbb{R}$  along the axis  $\mathbf{v} \in \mathbb{R}^3$ , is given by a screw characterized by a pitch (i.e., the ratio of translational to rotational motion)

$$h = \frac{\boldsymbol{\phi}^T \mathbf{v}}{\|\boldsymbol{\phi}\|^2},$$

a rotation axis

$$\mathbf{l} = \begin{cases} \left\{ \frac{\boldsymbol{\phi} \times \mathbf{v}}{\|\boldsymbol{\phi}\|^2} + \lambda \boldsymbol{\phi} : \lambda \in \mathbb{R} \right\}, & \text{if } \boldsymbol{\phi} \neq 0 \\ \{0 + \lambda \mathbf{v} : \lambda \in \mathbb{R}\}, & \text{if } \boldsymbol{\phi} = 0 \end{cases}$$

and a magnitude

$$M = \begin{cases} \|\boldsymbol{\phi}\|, & \text{if } \boldsymbol{\phi} \neq 0 \\ \|\mathbf{v}\|, & \text{if } \boldsymbol{\phi} = 0. \end{cases}$$

---

<sup>2</sup>Similar to how twists are defined as elements composed of linear and angular velocities, wrenches are defined as elements comprising of forces and torques.

Thus, working with this representation involves not only working with matrix exponential of the form (2.6), which does not have a readily intuitively geometrical meaning, but also involves having to choose among different cases, depending on the existence or not of a pure translation, in the construction of the screw.

#### 2.1.4 Lie algebra associated to $\text{Spin}(3) \ltimes \mathbb{R}^3$

As yet another way to represent rigid motions, for those willing to leave aside the realm of matrices, dual quaternion algebra presents itself as a compact and elegant solution that allows the representation of angular and linear entities as a single element, while enabling a straightforward algebraic treatment of the dynamic model, hence limiting the necessity of geometric analysis of the system. For instance, a twist may be represented in dual quaternion form as<sup>3</sup>

$$\begin{aligned} \underline{\boldsymbol{\xi}} &= \boldsymbol{\omega} + \varepsilon \boldsymbol{v} \\ &= \boldsymbol{\omega} + \varepsilon (\dot{\boldsymbol{p}} + \boldsymbol{p} \times \boldsymbol{\omega}) \\ &= \omega_x \hat{i} + \omega_y \hat{j} + \omega_z \hat{k} + \varepsilon (v_x \hat{i} + v_y \hat{j} + v_z \hat{k}), \end{aligned} \quad (2.7)$$

where  $\boldsymbol{\omega} \in \mathbb{H}_p$  and  $\boldsymbol{v} \in \mathbb{H}_p$  are, respectively, the angular and linear velocities of the body,  $\boldsymbol{p} \in \mathbb{H}_p$  is the position of its CoM,  $\hat{i}^2 = \hat{j}^2 = \hat{k}^2 = \hat{i}\hat{j}\hat{k} = -1$ , and  $\varepsilon^2 = 0$ . This representation may resemble that given by (2.2), but it has important differences that justify its use instead of a spatial velocity. Differently from what happens to spatial vectors, unit dual quaternions and pure dual quaternions, when equipped with standard multiplication and addition operations, form Lie groups with associated Lie algebras (Adorno, 2017). Furthermore, dual quaternion algebra provides an elegant representation of homogeneous transformations (Funda et al., 1990) and screw theory (Aspragathos & Dimitros, 1998), such that the twist given by (2.7) is an element of the Lie algebra associated with  $\text{Spin}(3) \ltimes \mathbb{R}^3$ .<sup>4</sup> Additionally, unit dual quaternions, as homogeneous transformation matrices (HTM), do not have representational singularities, and they are more compact and have lower computational cost than HTM (Adorno, 2011). Consequently, dual quaternion algebra possesses the thoroughness of the approach presented by Murray et al. (1994) while preserving the intuitiveness grasped by Featherstone (2008) with his spatial algebra. Nonetheless, as will be seen shortly, and further discussed in section 4.1, one of the main limitations of the use of dual quaternions in dynamics lies on how to properly represent inertia tensors within the algebra.

To better contextualize the contributions of this thesis, the next section presents the state of the art in dynamic modeling of robotic systems based on dual quaternion algebra.

<sup>3</sup>More details are given in chapter 3. For now, it is only important to have a general idea of how a dual quaternion representation of a twist looks like.

<sup>4</sup>The symbol  $\ltimes$  represents the semi-direct product between groups (Selig, 2005, p. 22).

## 2.2 State of the art in dual quaternion dynamic modeling

Several works have addressed the study of rigid body dynamics using dual quaternions over the last decades. One of the most common approaches is to use dual angles as generalized coordinates (figure 2.3) (Yang, 1965) and to represent body motions as screw motions<sup>5</sup> in a dual vector form (Pennock & Oncu, 1992), such as

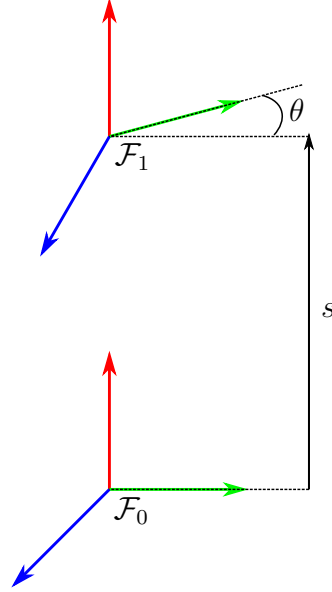


Figure 2.3: The dual angle between  $\mathcal{F}_0$  and  $\mathcal{F}_1$  is given by  $\underline{\theta} = \theta + \varepsilon s$ .

$$\underline{\mathbf{v}} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} + \varepsilon \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \boldsymbol{\omega} + \varepsilon \mathbf{v}, \quad (2.8)$$

where  $\boldsymbol{\omega}$  and  $\mathbf{v}$  are, respectively, the angular and the linear velocity of the body, and  $\varepsilon^2 = 0$ .

Pioneering this methodology, and the use of dual quaternions in mechanics, Yang & Freudenstein (1964) presented the use of dual quaternion algebra in the velocity and static force analysis of a spatial four-link mechanism. A dual vector  $\underline{\mathbf{a}}^o = \mathbf{a}_p^o + \varepsilon \mathbf{a}_d^o$  expressed in a reference frame  $\mathcal{F}_o$ , where  $\mathbf{a}_p^o, \mathbf{a}_d^o \in \mathbb{R}^3$  are, respectively, the primary and the dual part of  $\underline{\mathbf{a}}^o$ , can be expressed in a different reference frame  $\mathcal{F}_p$  as

$$\underline{\mathbf{a}}^p = \mathbf{a}_p^p + \varepsilon \mathbf{t}^p,$$

where

$$\mathbf{t}^p = \mathbf{a}_d^p + \mathbf{d} \times \mathbf{a}_p^p, \quad (2.9)$$

<sup>5</sup>An introduction to the subject, using matrices, can be found in Murray et al. (1994).

in which  $\mathbf{d} \in \mathbb{R}^3$  is the vector connecting  $\mathcal{F}_o$  and  $\mathcal{F}_p$ . Equation (2.9) is known as the Equation of Transformation and has to be considered in every transformation of reference frames, thus demanding a careful geometrical analysis of the system. This work was later extended by Yang (1966), providing equations for the accelerations of the mechanism. However, the application of dual quaternions to the study of rigid body dynamics was only introduced by Yang (1967), when the author combined the linear and the angular momentum of the body in a single dual quaternion, which he named dual momentum, given by

$$\underline{\mathbf{h}} = m\mathbf{v} + \varepsilon\bar{\mathbb{J}}\boldsymbol{\omega}, \quad (2.10)$$

where  $\boldsymbol{\omega}, \mathbf{v} \in \mathbb{R}^3$  are, respectively, the angular and linear velocities of the body. This work was further extended by Yang (1971), combining the equations of the resultant external forces and torques exerted on the body into a single dual vector equation, namely the dual force

$$\underline{\mathbf{f}} = \mathbf{f} + \varepsilon\boldsymbol{\tau},$$

where  $\mathbf{f}, \boldsymbol{\tau} \in \mathbb{R}^3$  are, respectively, the forces and torques in the body, with

$$\mathbf{f} = m(\mathbf{v} + \boldsymbol{\omega} \times \mathbf{d}),$$

in which  $\mathbf{d} \in \mathbb{R}^3$  is the vector connecting the reference frame with the frame attached to the CoM of the body, and

$$\boldsymbol{\tau} = \dot{\underline{\mathbf{h}}} + \boldsymbol{\omega} \times \underline{\mathbf{h}} + \mathbf{v} \times \mathbf{f},$$

where

$$\underline{\mathbf{h}} = \langle \bar{\mathbb{J}}, \boldsymbol{\omega} \rangle + m(\mathbf{d} \times \mathbf{v}),$$

in which  $\langle \bar{\mathbb{J}}, \boldsymbol{\omega} \rangle$  is the dot product between  $\boldsymbol{\omega}$  and the inertia dyadic of the body  $\bar{\mathbb{J}}$ .<sup>6</sup>

Although all these works (Yang & Freudenstein, 1964; Yang, 1966, 1967, 1971) provided an in-depth analysis of the spatial four-link mechanism being studied, they were tailored for that specific mechanism; hence, not concerning with achieving a systematic procedure to attain the dynamic model of a general multibody system. Such procedure was presented by Pennock & Yang (1983) in their analysis of a serial-open chain using matrices of dual numbers. In their work, the authors replaced the dual momentum given by (2.10) with

$$\underline{\mathbf{h}} = \bar{\Phi} \left[ \begin{array}{cccccc} \omega_x & \omega_y & \omega_z & v_x & v_y & v_z \end{array} \right]^T, \quad (2.11)$$

---

<sup>6</sup>For the definition of a dyadic and the dot product between it and a vector, please refer to Gürgöze & Zeren (2012).



where  $\Phi$  is the inertia binor,<sup>7</sup> given by

$$\Phi = \begin{bmatrix} \mathbf{S}^T & m\mathbf{I}_3 \\ \bar{\mathbb{I}} & \mathbf{S} \end{bmatrix} \in \mathbb{R}^{6 \times 6}, \quad (2.12)$$

where

$$\mathbf{S} = m \begin{bmatrix} 0 & -g_3 & g_2 \\ g_3 & 0 & -g_1 \\ -g_2 & g_1 & 0 \end{bmatrix}$$

is called the first moment of mass, in which  $g_1$ ,  $g_2$  and  $g_3$  are the coordinates of the CoM. The aforementioned calculation requires the mapping of the three-dimensional dual velocities given by (2.8) into a six-dimensional real vector in (2.11). Afterward, the obtained dual momentum must be remapped again to a three-dimensional dual vector. Moreover, despite representing velocities and momentum as dual vectors, the inertia binor and tensor are still given in the standard matrix representation, thus, not achieving a dynamic modeling procedure exclusively performed in terms of a dual quaternion formulation.

In order to avoid the dependency on the inertia binor, Shoham & Brodsky (1993) proposed the dual inertia, a mathematical object given given by

$$\underline{\mathbf{M}} = \mathbf{M} \frac{d}{d\varepsilon} + \varepsilon \bar{\mathbb{I}} = \begin{bmatrix} m \frac{d}{d\varepsilon} + \varepsilon i_{xx} & \varepsilon i_{xy} & \varepsilon i_{xz} \\ \varepsilon i_{yx} & m \frac{d}{d\varepsilon} + \varepsilon i_{yy} & \varepsilon i_{yz} \\ \varepsilon i_{zx} & \varepsilon i_{zy} & m \frac{d}{d\varepsilon} + \varepsilon i_{zz} \end{bmatrix}, \quad (2.13)$$

where  $i_{nn}$ ,  $n \in \{x, y, z\}$  are elements of the inertia tensor of the body, and the operator  $d/d\varepsilon$  that extracts the dual part of a dual quaternion; for instance,

$$m \frac{d}{d\varepsilon} (\boldsymbol{\omega} + \varepsilon \mathbf{v}) = m \mathbf{v}.$$

It is worth highlighting that the symbol  $d/d\varepsilon$  is a strong abuse of notation, having no derivative sense since  $\varepsilon$  is a constant element of the algebra. Furthermore,  $\varepsilon^{-1}$  is not defined because  $\varepsilon$  has no inverse, which makes an expression such as  $d/d\varepsilon$  syntactically inadequate. Nonetheless, using (2.13) the authors obtained the dual momentum as

$$\underline{\mathbf{h}} = \underline{\mathbf{M}} \underline{\mathbf{v}} = \begin{bmatrix} m \frac{d}{d\varepsilon} + \varepsilon i_{xx} & \varepsilon i_{xy} & \varepsilon i_{xz} \\ \varepsilon i_{yx} & m \frac{d}{d\varepsilon} + \varepsilon i_{yy} & \varepsilon i_{yz} \\ \varepsilon i_{zx} & \varepsilon i_{zy} & m \frac{d}{d\varepsilon} + \varepsilon i_{zz} \end{bmatrix} \begin{bmatrix} \omega_x + \varepsilon v_x \\ \omega_y + \varepsilon v_y \\ \omega_z + \varepsilon v_z \end{bmatrix},$$

thus avoiding the necessity of mapping  $\underline{\mathbf{v}}$  to  $\mathbb{R}^6$ . To obtain the equations of a multibody system, the authors proposed the use of the virtual-work and D'Alembert principle, thus demanding the time derivative of the Jacobian matrix, which can be an expensive operation

---

<sup>7</sup>More details can be found in the work of Dimentberg (1965).

(Chembrammel & Kesavadas, 2019). The use of the dual inertia was extended to the Lagrangian formulation by Brodsky & Shoham (1994), through the deduction of the equations of dual kinetic energy of a body.

As one of the few works in the literature not relying on dual angles, Dooley & McCarthy (1991) proposed an alternative dynamic model for a multibody system, based on the study of spatial motions and mechanisms presented by Ravani & Roth (1984), where dual quaternions were represented as a pair, given by

$$\underline{\mathbf{h}} = (\mathbf{h}, \mathbf{h}_o) = h_1\hat{i} + h_2\hat{j} + h_3\hat{k} + h_4 + \varepsilon (h_5\hat{i} + h_6\hat{j} + h_7\hat{k} + h_8),$$

where  $\hat{i}^2 = \hat{j}^2 = \hat{k}^2 = \hat{i}\hat{j}\hat{k} = -1$ ;  $h_1, h_2, h_3, h_4 \in \mathbb{R}$  are the Euler parameters of a rotation matrix representing a rotation of a angle of  $\theta \in \mathbb{R}$  around a rotation axis  $\mathbf{n} = (n_x, n_y, n_z)$ , given by

$$\begin{aligned} h_1 &= n_x \sin\left(\frac{\theta}{2}\right), & h_2 &= n_y \sin\left(\frac{\theta}{2}\right), \\ h_3 &= n_z \sin\left(\frac{\theta}{2}\right), & h_4 &= \cos\left(\frac{\theta}{2}\right); \end{aligned} \quad (2.14)$$

and  $h_5, h_6, h_7, h_8 \in \mathbb{R}$  are a combination of those parameters with the corresponding translation, given by

$$\begin{bmatrix} h_5 \\ h_6 \\ h_7 \\ h_8 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -d_z & d_y & d_x \\ d_z & 0 & -d_x & d_y \\ -d_y & d_x & 0 & d_z \\ -d_x & -d_y & -d_z & 0 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} \quad (2.15)$$

To compose successive transformations between reference frames, the authors used Hamilton operators.<sup>8</sup> For instance, given two dual quaternions  $\underline{\mathbf{g}} = (\mathbf{g}, \mathbf{g}_o)$  and  $\underline{\mathbf{h}} = (\mathbf{h}, \mathbf{h}_o)$ , the transformation  $\mathbf{GH}$  is given by

$$\mathbf{GH} = \overset{+}{\mathbf{H}}_8(\underline{\mathbf{g}}) \begin{bmatrix} \mathbf{h} \\ \mathbf{h}_o \end{bmatrix} = \bar{\mathbf{H}}_8(\underline{\mathbf{h}}) \begin{bmatrix} \mathbf{g} \\ \mathbf{g}_o \end{bmatrix},$$

where

$$\begin{aligned} \overset{+}{\mathbf{H}}_8(\underline{\mathbf{g}}) &= \begin{bmatrix} \overset{+}{\mathbf{H}}_4(\mathbf{g}) & \mathbf{0}_4 \\ \overset{+}{\mathbf{H}}_4(\mathbf{g}_o) & \overset{+}{\mathbf{H}}_4(\mathbf{g}) \end{bmatrix}, \\ \bar{\mathbf{H}}_8(\underline{\mathbf{h}}) &= \begin{bmatrix} \bar{\mathbf{H}}_4(\mathbf{h}) & \mathbf{0}_4 \\ \bar{\mathbf{H}}_4(\mathbf{h}_o) & \bar{\mathbf{H}}_4(\mathbf{h}) \end{bmatrix} \end{aligned} \quad (2.16)$$

---

<sup>8</sup>Although Dooley & McCarthy did not use the term *Hamilton operators*, I have chosen to use the name given by Adorno (2017).

with

$$\begin{aligned} \mathbf{H}_4^+(\mathbf{g}) &= \begin{bmatrix} g_4 & -g_3 & g_2 & g_1 \\ g_3 & g_4 & -g_1 & g_2 \\ -g_2 & g_1 & g_4 & g_3 \\ -g_1 & -g_2 & -g_3 & g_4 \end{bmatrix}, \\ \mathbf{H}_4^-(\mathbf{h}) &= \begin{bmatrix} h_4 & h_3 & -h_2 & h_1 \\ -h_3 & h_4 & h_1 & h_2 \\ h_2 & -h_1 & h_4 & h_3 \\ -h_1 & -h_2 & -h_3 & h_4 \end{bmatrix}. \end{aligned}$$

To obtain the dynamic model of a multibody system, the authors applied the method of generalized forces derived by Kane et al. (1983). The major disadvantage of the proposed model is its complicated equations of motion, with a nonintuitive physical meaning of its variables. Furthermore, this dynamic modeling formalism strongly mix dual quaternion algebra and standard matrix algebra.

An elegant study of the use of dual quaternions in the dynamics of multibody systems was presented by Hachicho & Eldin (2000). Representing dual quaternions in an exponential form, they derived iterative equations of propagation of velocities, accelerations, torques, and forces. However, the paper focus on the propagation of these equations, while the extension of the method for the formalism of Newton-Euler was left for future works. To the best of my knowledge, such work was never done. In addition, no validation of the equations was provided, and the mathematical cost of this approach was not discussed.

Valverde & Tsiotras (2018a) used dual quaternions to model the dynamics of a single rigid body, using a variation of the dual inertia matrix proposed by Shoham & Brodsky (1993), which obliges the swapping of the primary and dual parts of the dual velocity in order to obtain the dual momentum. To obtain the dynamics of a multibody system, Valverde & Tsiotras (2018a) formulated the dynamical equations in the form of a linear system (Salazar, 2018),

$$\mathbf{S}\mathbf{Y} = \mathbf{B},$$

in which  $\mathbf{Y}$  is a matrix of the unknown quantities (i.e., the linear and angular accelerations of each of the bodies and the reaction torques and forces experienced by them due to the kinematic coupling in the chain), while  $\mathbf{S}$  and  $\mathbf{B}$  are matrices constructed from the non-reaction wrenches. Thus, this approach involves the inversion of the matrix  $\mathbf{S} \in \mathbb{R}^s$ , where  $s = (8b + r) \times (8b + r)$ , in which  $b$  is the number of rigid bodies and  $r$  is the dimensionality of the reaction wrenches. It is worth highlighting, however, that  $\mathbf{S}$  is a block matrix and can be inverted blockwise from its four sub-blocks. Nonetheless, such sub-blocks are composed of  $8b \times 8b$ ,  $8b \times r$ ,  $r \times 8b$  and  $r \times r$  matrices, which may be still expensive to invert.

Miranda de Farias et al. (2019b), being more concerned with computational efficiency, proposed to use the dual quaternion exponential mapping and Plücker lines to represent screw displacements, as first proposed by Özgür & Mezouar (2016), to obtain the Newton-Euler model of serial manipulators with revolute joints. In their approach, dual quaternions were represented as

$$\underline{\mathbf{x}}_i^{i-1}(\theta_i(t)) = \underline{\delta}_i^{i-1} \exp\left(\underline{\mathbf{a}}_i \frac{\Delta\theta_i}{2}\right),$$

where  $\underline{\delta}_{i-1}^i \in \mathcal{H}$  is the dual quaternion representing the transformation between frames  $\mathcal{F}_i$  and  $\mathcal{F}_{i-1}$ , the pure dual quaternion  $\underline{\mathbf{a}}_i \in \mathcal{H}_p$  is the screw axis of frame  $\mathcal{F}_i$ , and  $\Delta\theta_i \in \mathbb{R}$  is the variation of the  $i$ -th joint with respect to its home configuration. Naturally, their algorithm works only for that specific representation and currently only for revolute joints, although an extension for prismatic joints should be fairly straightforward.

## 2.3 Topological graph and modular dynamic modeling

The approaches discussed on the previous section give a monolithic solution to the dynamic modeling of the robot. That is, using the aforementioned strategies, it is not possible to use the dynamic models of intermediate subsystems to compose the overall dynamic model of the system.

There are plenty of motivations for seeking a systematic dynamic model composition formalism. One could be assembling a robot using existing systems whose dynamic models are already known, such as the limbs of a humanoid robot. In a different scenario, a self-reconfiguring modular robot (Kotay et al., 1998; Neubert & Lipson, 2016; Unsal & Khosla, 2000) could possess the dynamic information of its modules. From a control perspective, one could be interested in applying distributed control strategies to the subsystems comprising a complex dynamic structure. To better contextualize the contributions of this thesis on this subject, the remaining of this section presents a review of the literature on dynamic modular composition strategies.

The application of linear graph theory in mechanism analysis is not a novelty, being used to the dynamic modeling of single rigid bodies (Andrews & Kesavan, 1975; Chou et al., 1986b) and multibody systems composed of open (Chou et al., 1986a) and closed kinematic chains (Sheth & Uicker, 1972; Andrews et al., 1988; Baciu et al., 1990; McPhee, 1996, 1998; Reungwetwattana & Toyama, 2001). Most approaches lead to different graphs for the rotational and translational variables of the mechanism (Andrews & Kesavan, 1975; Chou et al., 1986b,a; Baciu et al., 1990; McPhee, 1998).

Jain (2012) proposed a technique of partitioning and aggregating graphs that allows the consideration of subsystems during the dynamic modeling of multibody systems. The

author used the subgraph elements to compose the mass matrix  $\mathbf{M}(\mathbf{q})$  and the vector  $\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}})$  of nonlinear Coriolis and gyroscopic terms of the aggregated system and then calculated the stacked vector of generalized forces

$$\mathbb{R}^n \ni \boldsymbol{\tau} = \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}),$$

where  $\mathbf{q} \in \mathbb{R}^n$  is the vector of generalized coordinates of the  $n$ -DoF system. Thus, even though composed of individual modules, full knowledge of the masses, inertia tensors, Coriolis accelerations, and gyroscopic terms of the whole system is required.

McPhee (1998) presented a strategy where the terminal equations of individual graph components (e.g., the dynamic equations of rigid bodies, rigid-arms, etc.) were combined with the set of equations representing their connectivity, the so-called *cutset* and *circuit equations*, to obtain the system dynamics. Furthermore, the author also proposed a criterion for selecting the spanning trees of the translational and rotational graphs and a novel set of “branch coordinates” that leads to fewer motion equations than what would be obtained if either absolute or joint coordinates were used. The proposed formalism, however, was restricted to planar mechanisms. McPhee et al. (2004) extended this work to tree-dimensional mechanisms and proposed a strategy that uses individual subsystem models to compose the dynamic model of mechatronic multibody systems. Their approach uses free vectors and rotational matrices, thus decoupling translational and rotational components, which require separated graphs. Moreover, this formalism relies on symbolic implementation; thus, one must symbolically derive the subsystems before the modeling process to reorganize the system of differential-algebraic equations in the form

$$\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_q^T \boldsymbol{\lambda} = \mathbf{Q},$$

where  $\mathbf{M}$  is a mass matrix,  $\mathbf{q}$  is the vector of generalized coordinates,  $\boldsymbol{\Phi}_q$  is the Jacobian matrix of the system’s constraints,  $\boldsymbol{\lambda}$  is the vector of Lagrange multipliers, and  $\mathbf{Q}$  is a vector of external loads and quadratic velocity terms.

Moving away from graph representations, Matarazzo Orsino & Hess-Coelho (2015) proposed a strategy for the modular modeling of multibody systems based on the dynamic invariant equations of its subsystems. Orsino (2017) extended this formalism by proposing a hierarchical description of lumped-parameter dynamic systems that lead to a recursive modeling methodology. Furthermore, the author also explored its relations with the Udwadia-Kalaba equation (Udwadia & Kalaba, 1992). Albeit those formulations (Matarazzo Orsino & Hess-Coelho, 2015; Orsino, 2017) do not impose limitations in how the dynamic equations of each subsystem must be obtained, the final result is not given in terms of the generalized forces of the overall dynamic system but rather by its dynamic invariants. Thus, this strategy is not necessarily convenient for problems of robot dynamic modeling, where one is typically interested in finding the joint forces/torques required to

control the robot.

More recently, Hess-Coelho et al. (2021) presented a dynamic modular modeling methodology for parallel mechanisms. Their strategy uses the hierarchical description proposed by Orsino (2017) but follows a different approach to deduce the model. The authors rely on the Jacobian matrices of the subsystem's angular and linear velocities and apply the Principle of Virtual Power to arrive at the Euler-Lagrange model of the robot. Modularity is achieved by using a library of previously deduced subsystem models. Nonetheless, the process of obtaining such models is highly dependent on geometric analysis of the system (i.e., geometric inverse kinematics) and, if the library was not developed beforehand, the methodology performs a monolithic solution of the robot dynamics. Moreover, due to the free vector representation in this formalism, translational and rotational components are decoupled. Furthermore, this strategy requires information of the inertia matrix of the whole system.

## 2.4 Conclusions

The analysis of rigid body dynamics involves dealing with highly coupled linear and angular components. Nonetheless, classic approaches represent those entities as different vectors (Spong et al., 2006; Siciliano & Khatib, 2008; Siciliano et al., 2009; Corke, 2011), thus demanding a deep geometrical analysis of the system being studied, which might be a simple problem for serial manipulators but becomes increasingly complicated for more complex structures, such as humanoid robots. As a possible way of unification of those coupled components, Featherstone (1984) proposed the spatial algebra. This algebra has appealing utilitarian aspects, such as its intuitiveness and the efficiency achieved by the spatial algebra version of the classic dynamic modeling algorithms. However, the spatial algebra lacks the necessary expressiveness in order to be free from *ad hoc* solutions in some situations.

A more algebraically formal approach to treat those coupled linear and angular components is given by the screw theory, in which they are represented as a single element. The main advantage of this strategy over the spatial algebra is that screws form groups on different Lie algebras, thus possessing strong algebraic properties. Murray et al. (1994) presented the use of screws of the algebra  $se(3)$  through the exponential mapping of  $SE(3)$ . Although this representation mitigates the necessity of a deep geometrical analysis of the systems, it is not as intuitive as the spatial algebra. Rigid transformations on  $SE(3)$  are represented by the exponential of twists in this approach, thus demanding one to deal with exponential of matrices. Additionally, twists are constructed as choices between “case options” for its pitch, rotation axis and magnitude, depending on the value of the angular velocity.

Screws also form a group on dual quaternion algebra. When represented as dual

hypercomplex numbers, they preserve the intuitiveness grasped by Featherstone (1984) while maintaining the thoroughness of the approach presented by Murray et al. (1994). Besides, dual quaternions had their elegance and efficiency proved in kinematic modeling (Perez & McCarthy, 2004; Adorno, 2011; Gouasmi, 2012; Cohen & Shoham, 2016; Özgür & Mezouar, 2016; Kong, 2017; Dantam, 2020) and control (Adorno et al., 2010; Figueredo et al., 2014; Adorno et al., 2015; Lana et al., 2015; Marinho et al., 2015; Kussaba et al., 2017; Quiroz-Omana & Adorno, 2019; Marinho et al., 2019; Fonseca et al., 2020; Savino et al., 2020), thus raising the question if they are an equally suitable mathematical representation for the dynamic modeling of robotic systems.

Some works have addressed the use of dual quaternion algebra on rigid body dynamics over the last decades, although not necessarily concerned with the creation of a formalism for multibody system analysis (Yang & Freudenstein, 1964; Yang, 1966, 1967, 1971). Among the works that sought this formalism, mostly are based on three-dimensional dual vectors and demand some mapping to higher dimensional vectors in order to find the dynamic equations of the system (Pennock & Yang, 1983; Dooley & McCarthy, 1991; Shoham & Brodsky, 1993; Valverde & Tsiotras, 2018a), therefore losing the elegance of a analysis based only on dual quaternion algebra and, at times, incurring in abuses of notation (Dooley & McCarthy, 1991) or demanding artificial swaps on the vectors (Valverde & Tsiotras, 2018b) in order to deal with that mixing of representations. Other works focused on the propagation of dual quaternions (Hachicho & Eldin, 2000) or the computational aspects of algorithms based on this algebra (Miranda de Farias et al., 2019b), rather than on the algebraic and geometrical insights provided by the dual quaternion dynamic modeling of robotics systems, which are especially important when dealing with more complex robots (e.g., mobile manipulators, humanoids, etc.) and more general joint types (e.g., helical, cylindrical, 6-DoF, etc.).

Regarding the use of topological graphs and modular dynamic modeling, the existing model composition strategies oftentimes generate different graph representations for the translational and rotational components (Andrews & Kesavan, 1975; Chou et al., 1986b,a; Baciú et al., 1990; McPhee, 1998; McPhee et al., 2004) and either require the previous construction of subsystem models (McPhee et al., 2004; Hess-Coelho et al., 2021) or demand full knowledge of the dynamic elements of the whole system (Jain, 2012; Orsino, 2017).

In this context, this thesis presents a study on the application of the dual quaternion algebra to the dynamic modeling of complex robotic systems whilst exploring to which extension the expressiveness of this algebra simplifies the study of those systems and what impacts the use of dual quaternions cause on the computational costs of classic formalisms. This thesis proposes a general procedure to attain the dynamic model of branched robotic systems using dual quaternion algebra, which includes serial kinematic chains as a particular case. Moreover, the method allows easily assembling of previously

calculated models into a single model of a more complex robotic system, even if some subsystems are regarded as black boxes. Furthermore, it is expected that suitable dual quaternion dynamic models will allow the development of dual quaternion control strategies without the need for intermediate mappings, which usually introduce mathematical artifacts, such as algorithmic singularities and discontinuities.



# 3

## MATHEMATICAL BACKGROUND

This chapter presents the mathematical concepts and notations used throughout this thesis. More specifically, it briefly reviews the classic<sup>1</sup> Newton-Euler and Euler-Lagrange formulations, commonly found in robotics textbooks, and the basic operations with quaternions and dual quaternions, alongside their applications in kinematics.

Since neither of those topics is a contribution of this thesis, some details will be omitted to avoid overextending the text. For a more complete account on the Newton-Euler and Euler-Lagrange formulations, please refer to Spong et al. (2006); Siciliano et al. (2009). For a thorough introduction on dual algebra and kinematic modeling and control, please refer to Adorno (2017). Nonetheless, as shown in section 2.2, notations used for dual quaternions vary greatly, and this chapter fulfills the crucial role of presenting the ones chosen for this thesis. Readers should, therefore, at least skim section 3.3 of this chapter to ensure that they are familiarized with them, even if they already have a great general understanding of the algebra.

### 3.1 Classic Newton-Euler formulation

In the classic approach, the iterative Newton-Euler formulation for robot manipulators (Luh et al., 1979) consists in the following process. First, in the serial process, the forward recurrence equations are used to calculate the angular velocities (Spong et al., 2006, p.

---

<sup>1</sup>Henceforth, the term *classic* will be used to refer to approaches based on  $\mathbb{R}^3$  and  $SO(3)$ .

275; Siciliano et al., 2009, p. 282)<sup>2</sup>

$$\boldsymbol{\omega}_{0,i}^{i-1} = \mathbf{R}_{i-2}^{i-1} \boldsymbol{\omega}_{0,i-1}^{i-2} + \dot{\theta}_i \mathbf{z}_{i-1}^{i-1},$$

the angular accelerations

$$\dot{\boldsymbol{\omega}}_{0,i}^{i-1} = \mathbf{R}_{i-2}^{i-1} \dot{\boldsymbol{\omega}}_{0,i-1}^{i-2} + \ddot{\theta}_i \mathbf{z}_{i-1}^{i-1} + \left( \mathbf{R}_{i-2}^{i-1} \boldsymbol{\omega}_{0,i-1}^{i-2} \right) \times \left( \dot{\theta}_i \mathbf{z}_{i-1}^{i-1} \right)$$

and the linear accelerations

$$\begin{aligned} \mathbf{a}_{0,i}^{i-1} &= \mathbf{R}_{i-2}^{i-1} \left( \mathbf{a}_{0,i-1}^{i-2} + \dot{\boldsymbol{\omega}}_{0,i-1}^{i-2} \times \mathbf{p}_{i-2,i-1}^{i-2} + \boldsymbol{\omega}_{0,i-1}^{i-2} \times \left( \boldsymbol{\omega}_{0,i-1}^{i-2} \times \mathbf{p}_{i-2,i-1}^{i-2} \right) \right), \\ \mathbf{a}_{0,c_i}^{i-1} &= \mathbf{a}_{0,i}^{i-1} + \dot{\boldsymbol{\omega}}_{0,i}^{i-1} \times \mathbf{p}_{i-1,c_i}^{i-1} + \boldsymbol{\omega}_{0,i}^{i-1} \times \left( \boldsymbol{\omega}_{0,i}^{i-1} \times \mathbf{p}_{i-1,c_i}^{i-1} \right), \end{aligned}$$

where  $\mathbf{R}_{i-2}^{i-1} \in SO(3)$  is the rotation matrix representing the rotation between frames  $\mathcal{F}_{i-1}$  and  $\mathcal{F}_{i-2}$ . The variables  $\theta_i \in \mathbb{R}$  and  $\mathbf{z}_{i-1}^{i-1} \in \mathbb{R}^3$  are, respectively, the angle and the rotation axis of the  $i$ -th joint, and  $\mathbf{p}_{i-2,i-1}^{i-2} \in \mathbb{R}^3$  is its position with respect to the previous joint, expressed in the reference frame of the previous joint.<sup>3</sup>

Then, in the backward recursive process, those recurrence equations are used to calculate the forces

$$\mathbf{f}_{i-1}^{i-1} = \mathbf{R}_i^{i-1} \mathbf{f}_i^i + m_i \left( \mathbf{a}_{c_i}^{i-1} + \mathbf{g}^{i-1} \right),$$

and the torques

$$\boldsymbol{\tau}_{i-1}^{i-1} = \mathbf{R}_i^{i-1} \boldsymbol{\tau}_i^i - \mathbf{f}_{i-1}^{i-1} \times \mathbf{p}_{i-1,c_i}^{i-1} + \mathbf{R}_i^{i-1} \left( \mathbf{f}_i^i \times \mathbf{p}_{i,c_i}^i \right) + \bar{\mathbb{I}}_i^{i-1} \dot{\boldsymbol{\omega}}_{0,i}^{i-1} + \boldsymbol{\omega}_{0,i}^{i-1} \times \left( \bar{\mathbb{I}}_i^{i-1} \boldsymbol{\omega}_{0,i}^{i-1} \right),$$

where  $m_i \in (0, \infty)$  is the mass of the  $i$ -th link,  $\bar{\mathbb{I}}_i^{i-1} \in \mathbb{R}^{3 \times 3}$ , with  $\bar{\mathbb{I}}_i^{i-1} > 0$ , is its inertia matrix expressed in its own reference frame, and  $\mathbf{g}^{i-1} \in \mathbb{R}^3$  is the gravity acceleration expressed in the CoM of the  $i$ -th link.

A possible extension of this method for holonomic mobile manipulators is achieved by considering the mobile base as composed of virtual joints, namely two prismatic joints aligned with the axes  $x, y$  and one rotational joint aligned with the axis  $z$ .

---

<sup>2</sup>Here, the standard D-H convention was used in the definition of the reference frames, although that was not the case in Spong et al., 2006.

<sup>3</sup>As defined in the section of notations (page xix), this convention of subscripts and superscripts is maintained throughout this thesis. If no superscript is used, it is assumed the global inertial frame.

## 3.2 Classic Euler-Lagrange formulation

The canonical form of the Euler-Lagrange model for an  $n$ -DoF serial manipulator is given by (Spong et al., 2006, p. 257)

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}, \quad (3.1)$$

where  $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$  is the inertia matrix,  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$  is the Coriolis and Centrifugal terms matrix,  $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$  is the gravity vector, and  $\boldsymbol{\tau} \in \mathbb{R}^n$  is the vector of non-conservative generalized forces.

Considering the global inertial reference frame, from the kinetic energy associated with each link  $i$ , given by

$$K_i = \frac{1}{2} \dot{\mathbf{q}}^T \left( \underbrace{m_i (\mathbf{J}_{v_i}^T \mathbf{J}_{v_i}) + J_{\omega_i}^T \mathbb{I}_i \mathbf{J}_{\omega_i}}_{\mathbf{M}_i} \right) \dot{\mathbf{q}},$$

where  $\dot{\mathbf{q}} \in \mathbb{R}^n$  is the time derivative of the robot configuration vector and  $\mathbf{J}_{v_i} \in \mathbb{R}^{3 \times 3}$  and  $\mathbf{J}_{\omega_i} \in \mathbb{R}^{3 \times 3}$  are, respectively, the Jacobian matrix of linear and angular velocity. The inertia matrix  $\mathbf{M}(\mathbf{q})$  may be obtained as

$$\mathbf{M}(\mathbf{q}) = \sum_{i=1}^n \mathbf{M}_i.$$

The Coriolis and Centrifugal terms matrix  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  is obtained from  $\mathbf{M}(\mathbf{q})$  through the Christoffel symbols of first kind as

$$c_{kj} = \sum_{i=1}^n \frac{1}{2} \left( \frac{\partial \bar{m}_{kj}}{\partial q_j} + \frac{\partial \bar{m}_{ki}}{\partial q_j} + \frac{\partial \bar{m}_{ij}}{\partial q_k} \right) \dot{q}_i,$$

where  $c_{kj}$  is the element of the  $k$ -th row and  $j$ -th column of  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ ,  $\bar{m}_{ij}$  is the  $ij$ -th element of  $\mathbf{M}(\mathbf{q})$  and  $\dot{q}_i$  is the  $i$ -th element of  $\dot{\mathbf{q}}$ .

Finally, the gravitational vector  $\mathbf{g}(\mathbf{q})$  is obtained from the potential energy  $P$  as

$$\mathbf{g}(\mathbf{q}) = \frac{\partial P}{\partial \mathbf{q}}.$$

### 3.2.1 Canonical Euler-Lagrange equation from the recursive Newton-Euler formulation

Alternatively, the matrices from the canonical form (3.1) can be obtained through the iterative Newton-Euler algorithm<sup>4</sup> as

$$\begin{aligned} \mathbf{g}(\mathbf{q}) &= \text{newton\_euler}(\mathbf{q}, \mathbf{0}, \mathbf{0}), \\ \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} &= \text{newton\_euler}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{0}) - \mathbf{g}(\mathbf{q}), \\ \text{col}_i(\mathbf{M}(\mathbf{q})) &= \text{newton\_euler}(\mathbf{q}, \mathbf{0}, \text{col}_i(\mathbf{I}_n)) - \mathbf{g}(\mathbf{q}), \end{aligned} \quad (3.2)$$

where  $\text{newton\_euler}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$  represents the execution of the algorithm with the inputs of joint positions, velocities, and accelerations,  $\mathbf{0} \in \mathbb{R}^n$  is a vector of zeros, in which  $n$  is the number of rigid bodies in the kinematic chain,  $\text{col}_i(\cdot)$  extracts the  $i$ -th column of a given matrix, and  $\mathbf{I}_n \in \mathbb{R}^{n \times n}$  is the identity matrix.

## 3.3 Dual quaternion algebra

This thesis relies on dual quaternion algebra to derive the dynamical model of robotic systems. This section presents the definitions, basic operations, and geometrical insights of quaternion and dual quaternion algebra. Furthermore, section 3.3.3 briefly recollects the dual quaternion forward kinematic model proposed by Adorno (2011) since this formulation is essential for the dynamic modeling strategies proposed in this thesis.

However, it is important to highlight that the contributions of this thesis lie in the methodology to derive the dynamic models themselves; therefore, there is no novelty in the dual quaternion *algebra* presented here.

### 3.3.1 Quaternions

Quaternions were proposed by Hamilton in the 19th century (Hamilton, 1844, apud Adorno, 2011) with the purpose of extending the algebra of complex numbers, and they are elements of the set given by

$$\mathbb{H} \triangleq \{h_1 + \hat{i}h_2 + \hat{j}h_3 + \hat{k}h_4 : h_1, h_2, h_3, h_4 \in \mathbb{R} \text{ and } \hat{i}^2 = \hat{j}^2 = \hat{k}^2 = \hat{i}\hat{j}\hat{k} = -1\}. \quad (3.3)$$

Addition and multiplication of quaternions are analogous to their counterparts of real and complex numbers; one must only respect the properties of the imaginary units  $\hat{i}, \hat{j}, \hat{k}$  given in (3.3). Quaternions whose real part (i.e., the element not multiplied by any

---

<sup>4</sup>This is an adaptation of the results given in the lecture notes of Prof. Alessandro De Luca (Luca, 2019).

imaginary unit, given by  $\text{Re}(\mathbf{h})$ ) is equal to zero are called pure quaternions and form the set  $\mathbb{H}_p \triangleq \{\mathbf{h} \in \mathbb{H} : \text{Re}(\mathbf{h}) = 0\}$ .

Pure quaternions are used to represent positions  $(p_x, p_y, p_z)$  in the three-dimensional space by means of position quaternions

$$\mathbf{p} = \hat{i}p_x + \hat{j}p_y + \hat{k}p_z, \quad (3.4)$$

as shown in figure 3.1.

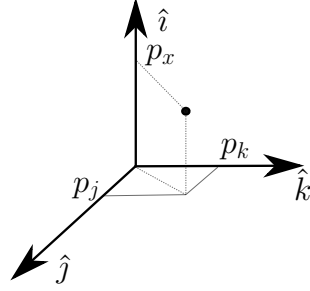


Figure 3.1: Three-dimensional position  $(p_x, p_y, p_z)$  represented by the pure quaternion  $\mathbf{p} = \hat{i}p_x + \hat{j}p_y + \hat{k}p_z$ .

The norm of a quaternion  $\mathbf{h} = h_r + \hat{i}h_x + \hat{j}h_y + \hat{k}h_z$  is defined as (Adorno, 2017)

$$\|\mathbf{h}\| = \sqrt{\mathbf{h}\mathbf{h}^*} = \sqrt{\mathbf{h}^*\mathbf{h}},$$

where  $\mathbf{h}^* = h_r - \hat{i}h_x - \hat{j}h_y - \hat{k}h_z$  is the conjugate of  $\mathbf{h}$ , and  $\|\mathbf{h}\|$  is equivalent to the Euclidean norm. Quaternions whose norm is equal to one are called unit quaternions (rotation quaternions) and are used to represent rotations by means of

$$\mathbf{r} = \cos\left(\frac{\phi}{2}\right) + \mathbf{n} \sin\left(\frac{\phi}{2}\right), \quad (3.5)$$

where  $\phi \in [0, 2\pi)$  is the rotation angle around the rotation axis  $\mathbf{n} \in \mathbb{H}_p \cap \mathbb{S}^3$ , with  $\mathbb{S}^3 = \{\mathbf{h} \in \mathbb{H} : \|\mathbf{h}\| = 1\}$  (Selig, 2005), as illustrated in figure 3.2.

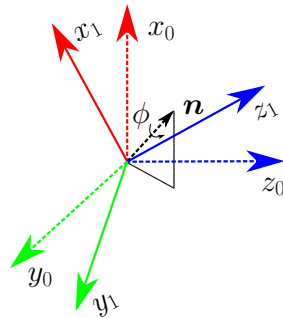


Figure 3.2: Rotation of an angle  $\phi$  around a rotation axis  $\mathbf{n}$ , represented by the rotation quaternion  $\mathbf{r} = \cos(\phi/2) + \mathbf{n} \sin(\phi/2)$ .

Additionally,

$$\mathbf{a} \times \mathbf{b} \triangleq \frac{\mathbf{ab} - \mathbf{ba}}{2}, \quad (3.6)$$

with  $\mathbf{a}, \mathbf{b} \in \mathbb{H}_p$ , is the cross product between pure quaternions, which is analogous to the cross product between vectors in  $\mathbb{R}^3$ ; whereas

$$\langle \mathbf{a}, \mathbf{b} \rangle \triangleq -\frac{(\mathbf{ab} + \mathbf{ba})}{2}, \quad (3.7)$$

is the inner product between pure quaternions (Adorno, 2017).

### 3.3.2 Dual Quaternions

Clifford extended the algebra of quaternions in 1873 giving rise to dual quaternions (Selig, 2005), which are elements of the set

$$\mathcal{H} \triangleq \{\mathbf{h}_{\mathcal{P}} + \varepsilon \mathbf{h}_{\mathcal{D}} : \mathbf{h}_{\mathcal{P}}, \mathbf{h}_{\mathcal{D}} \in \mathbb{H}, \varepsilon \neq 0, \varepsilon^2 = 0\}. \quad (3.8)$$

Addition and multiplication of dual quaternions are analogous to their counterparts of real and complex numbers; one must only respect the properties of imaginary units  $\hat{i}, \hat{j}, \hat{k}$  and dual unit  $\varepsilon$  given by (3.3) and (3.8), respectively.

Given  $\underline{\mathbf{h}} \in \mathcal{H}$  such that

$$\underline{\mathbf{h}} = \underbrace{h_1 + \hat{i}h_2 + \hat{j}h_3 + \hat{k}h_4}_{\mathbf{h}_{\mathcal{P}}} + \varepsilon \underbrace{(h_5 + \hat{i}h_6 + \hat{j}h_7 + \hat{k}h_8)}_{\mathbf{h}_{\mathcal{D}}},$$

the operators

$$\begin{aligned} \mathcal{P}(\underline{\mathbf{h}}) &\triangleq \mathbf{h}_{\mathcal{P}} \\ \mathcal{D}(\underline{\mathbf{h}}) &\triangleq \mathbf{h}_{\mathcal{D}} \end{aligned} \quad (3.9)$$

provide the primary part and dual part of  $\underline{\mathbf{h}}$ , respectively, whereas the operators  $\text{Re}(\underline{\mathbf{h}}) \triangleq h_1 + \varepsilon h_5$  and  $\text{Im}(\underline{\mathbf{h}}) = \hat{i}h_2 + \hat{j}h_3 + \hat{k}h_4 + \varepsilon(\hat{i}h_6 + \hat{j}h_7 + \hat{k}h_8)$  provide the real and the imaginary parts of  $\underline{\mathbf{h}}$ , respectively. The conjugate of  $\underline{\mathbf{h}}$  is defined as  $\underline{\mathbf{h}}^* \triangleq \text{Re}(\underline{\mathbf{h}}) - \text{Im}(\underline{\mathbf{h}})$  and its norm is given by  $\|\underline{\mathbf{h}}\| = \sqrt{\underline{\mathbf{h}}\underline{\mathbf{h}}^*} = \sqrt{\underline{\mathbf{h}}^*\underline{\mathbf{h}}}$  (Adorno, 2017).

To help simplify some of the expressions that will be presented in chapter 4, the following lemma is proposed.

**Lemma 3.1.** *Given  $\underline{\mathbf{a}} \in \mathcal{H}$  and  $\mathbf{b} \in \mathbb{H}$ , then*

$$\text{Ad}(\underline{\mathbf{a}})(\varepsilon \mathbf{b}) = \underline{\mathbf{a}}(\varepsilon \mathbf{b})\underline{\mathbf{a}}^* = \text{Ad}(\mathcal{P}(\underline{\mathbf{a}}))(\varepsilon \mathbf{b}).$$

*Proof.* Expanding  $\text{Ad}(\underline{\mathbf{a}})(\varepsilon\mathbf{b})$  and using the fact that  $\underline{\mathbf{a}} = \mathcal{P}(\underline{\mathbf{a}}) + \varepsilon\mathcal{D}(\underline{\mathbf{a}})$ , we obtain

$$\begin{aligned}
 \text{Ad}(\underline{\mathbf{a}})(\varepsilon\mathbf{b}) &= \underline{\mathbf{a}}(\varepsilon\mathbf{b})\underline{\mathbf{a}}^* \\
 &= (\mathcal{P}(\underline{\mathbf{a}}) + \varepsilon\mathcal{D}(\underline{\mathbf{a}}))(\varepsilon\mathbf{b})(\mathcal{P}(\underline{\mathbf{a}}^*) + \varepsilon\mathcal{D}(\underline{\mathbf{a}}^*)) \\
 &= (\mathcal{P}(\underline{\mathbf{a}})(\varepsilon\mathbf{b}))(\mathcal{P}(\underline{\mathbf{a}}^*) + \varepsilon\mathcal{D}(\underline{\mathbf{a}}^*)) \\
 &= \mathcal{P}(\underline{\mathbf{a}})(\varepsilon\mathbf{b})\mathcal{P}(\underline{\mathbf{a}}^*) \\
 &= \text{Ad}(\mathcal{P}(\underline{\mathbf{a}}))(\varepsilon\mathbf{b}).
 \end{aligned}$$

□

### 3.3.2.1 Unit dual quaternions

The subset  $\underline{\mathbf{S}} = \{\underline{\mathbf{h}} \in \mathcal{H} : \|\underline{\mathbf{h}}\| = 1\}$ , known as the set of *unit* dual quaternions, is used to represent poses (position and orientation) in the three-dimensional space and form the group  $\text{Spin}(3) \ltimes \mathbb{R}^3$  under the multiplication operation. Unit dual quaternions can always be written as (Selig, 2005)

$$\underline{\mathbf{x}} = \mathbf{r} + \varepsilon \frac{1}{2} \mathbf{p} \mathbf{r} = \mathbf{r} + \varepsilon \frac{1}{2} \mathbf{r} \mathbf{p}^b, \quad (3.10)$$

where  $\mathbf{p}$  and  $\mathbf{r}$  are given by (3.4) and (3.5), respectively, and  $\mathbf{p}^b = \mathbf{r}^* \mathbf{p} \mathbf{r}$  is the position with respect to the body frame.

Successive multiplications of unit dual quaternions, given by (3.10), represent a sequence of rigid transformations, as shown in figure 3.3.

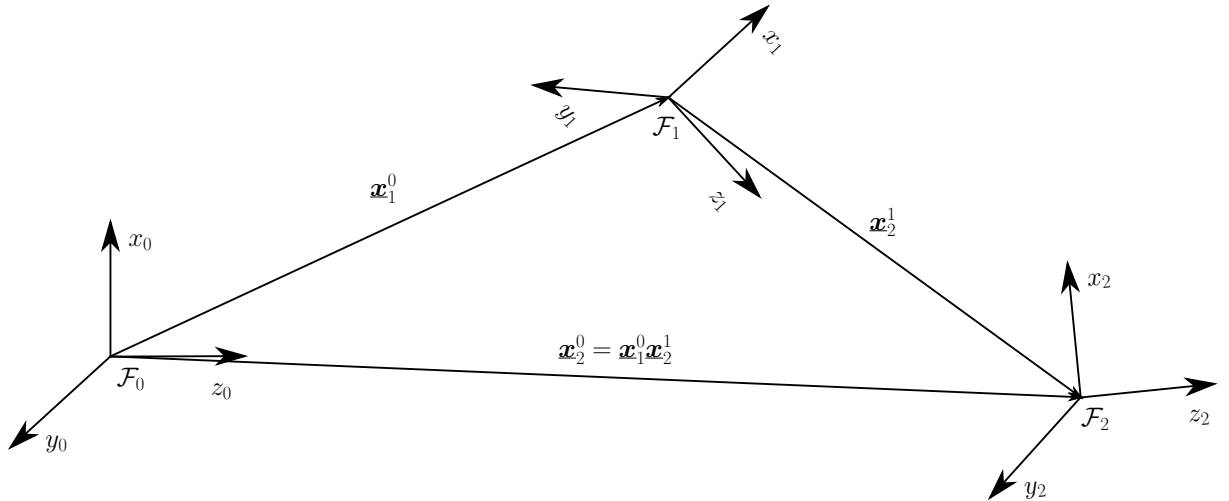


Figure 3.3: Sequence of rigid transformations represented by unit dual quaternions.

Pure dual quaternions are elements of the set  $\mathcal{H}_p = \{\underline{\mathbf{h}} \in \mathcal{H} : \text{Re}(\underline{\mathbf{h}}) = 0\}$  and are used to represent twists and wrenches. The operator  $\text{Ad}(\underline{\mathbf{x}}) : \mathcal{H}_p \rightarrow \mathcal{H}_p$ , where  $\underline{\mathbf{x}} \in \underline{\mathbf{S}}$ , performs rigid transformations on those entities. For instance, given  $\underline{\xi}^a \in \mathcal{H}_p$ , a twist

expressed in the reference frame  $\mathcal{F}_a$ , and the unit dual quaternion  $\underline{\mathbf{x}}_a^b$  that gives the pose of  $\mathcal{F}_a$  with respect to  $\mathcal{F}_b$ , the same twist is expressed in frame  $\mathcal{F}_b$  as (Adorno, 2017)

$$\underline{\boldsymbol{\xi}}^b = \text{Ad}(\underline{\mathbf{x}}_a^b) \underline{\boldsymbol{\xi}}^a = \underline{\mathbf{x}}_a^b \underline{\boldsymbol{\xi}}^a (\underline{\mathbf{x}}_a^b)^* . \quad (3.11)$$

The time derivative of  $\underline{\mathbf{x}}_b^a$  is given by (Han et al., 2008)

$$\dot{\underline{\mathbf{x}}}_b^a = \frac{1}{2} \underline{\boldsymbol{\xi}}_{ab}^a \underline{\mathbf{x}}_b^a = \frac{1}{2} \underline{\mathbf{x}}_b^a \underline{\boldsymbol{\xi}}_{ab}^b, \quad (3.12)$$

where

$$\underline{\boldsymbol{\xi}}_{ab}^a = \boldsymbol{\omega}_{ab}^a + \varepsilon (\dot{\mathbf{p}}_{ab}^a + \mathbf{p}_{ab}^a \times \boldsymbol{\omega}_{ab}^a) \quad (3.13)$$

is the twist expressed in frame  $\mathcal{F}_a$ , with  $\boldsymbol{\omega}_{ab}^a \in \mathbb{H}_p$  being the angular velocity, and<sup>5</sup>

$$\underline{\boldsymbol{\xi}}_{ab}^b = \text{Ad}(\underline{\mathbf{x}}_a^b) \underline{\boldsymbol{\xi}}_{ab}^a = \boldsymbol{\omega}_{ab}^b + \varepsilon \dot{\mathbf{p}}_{ab}^b \quad (3.14)$$

is the twist expressed in  $\mathcal{F}_b$ . Furthermore,  $\underline{\boldsymbol{\xi}}_{ab}^a$  is an element of the Lie algebra associated to  $\text{Spin}(3) \ltimes \mathbb{R}^3$ . Additionally, given  $\underline{\mathbf{a}}, \underline{\mathbf{b}} \in \mathcal{H}_p$ , in which  $\underline{\mathbf{a}} = \mathbf{a} + \varepsilon \mathbf{a}'$  and  $\underline{\mathbf{b}} = \mathbf{b} + \varepsilon \mathbf{b}'$ ,

$$\langle \underline{\mathbf{a}}, \underline{\mathbf{b}} \rangle \triangleq -\frac{(\underline{\mathbf{a}}\underline{\mathbf{b}} + \underline{\mathbf{b}}\underline{\mathbf{a}})}{2}, \quad (3.15)$$

is the inner product between pure dual quaternions and

$$\underline{\mathbf{a}} \times \underline{\mathbf{b}} \triangleq \frac{\underline{\mathbf{a}}\underline{\mathbf{b}} - \underline{\mathbf{b}}\underline{\mathbf{a}}}{2} = \mathbf{a} \times \mathbf{b} + \varepsilon (\mathbf{a} \times \mathbf{b}' + \mathbf{a}' \times \mathbf{b}), \quad (3.16)$$

is the cross-product between pure dual quaternions (Adorno, 2017).

Moreover, given the twist  $\underline{\boldsymbol{\xi}}'$  expressed in frame  $\mathcal{F}$  and the relative pose  $\underline{\mathbf{x}} \in \underline{\mathcal{S}}$  between this frame and the frame the  $\mathcal{F}$ , the time derivative of  $\underline{\boldsymbol{\xi}} = \text{Ad}(\underline{\mathbf{x}}) \underline{\boldsymbol{\xi}}'$  is given by the following lemma.

**Lemma 3.2** (Silva et al., 2022). *If  $\underline{\mathbf{x}} \in \underline{\mathcal{S}}$ , such that  $\dot{\underline{\mathbf{x}}} = (1/2)\underline{\boldsymbol{\xi}}\underline{\mathbf{x}}$  and  $\underline{\boldsymbol{\xi}}' \in \mathcal{H}_p$ , then*

$$\frac{d}{dt} (\text{Ad}(\underline{\mathbf{x}}) \underline{\boldsymbol{\xi}}') = \text{Ad}(\underline{\mathbf{x}}) \dot{\underline{\boldsymbol{\xi}}}' + \underline{\boldsymbol{\xi}} \times (\text{Ad}(\underline{\mathbf{x}}) \underline{\boldsymbol{\xi}}'). \quad (3.17)$$

### 3.3.3 Forward Kinematic Model

As proposed by Adorno (2011), the forward kinematics of any robot can be represented using dual quaternions by a suitable mapping  $\underline{\mathbf{f}} : \mathbb{R}^n \rightarrow \text{Spin}(3) \ltimes \mathbb{R}^3$  such that

$$\underline{\mathbf{x}}_{\text{eff}} = \underline{\mathbf{f}}(\mathbf{q}), \quad (3.18)$$

---

<sup>5</sup>This result can be demonstrated by direct calculation.



where  $\mathbf{q} \in \mathbb{R}^n$  is the robot configuration vector and  $\mathbf{x}_{\text{eff}} \in \text{Spin}(3) \ltimes \mathbb{R}^3$  is the end-effector pose.

For an  $n$ -DoF serial manipulator, as the one shown in figure 3.4,<sup>6</sup> equation (3.18) can be rewritten as

$$\mathbf{x}_{\text{eff}} = \mathbf{x}_n^0 = \mathbf{x}_1^0 \mathbf{x}_2^1 \dots \mathbf{x}_n^{n-1}, \quad (3.19)$$

where  $\mathbf{x}_i^{i-1}$  is the pose of each intermediate link with respect to its predecessor in the kinematic chain. For the specific case where the D-H convention is used, each  $\mathbf{x}_i^{i-1}$  is given by

$$\mathbf{x}_i^{i-1} = \mathbf{r}_{z,\theta_i} \underline{\mathbf{p}}_{z,d_i} \underline{\mathbf{p}}_{x,a_i} \mathbf{r}_{x,\alpha_i}, \quad (3.20)$$

where  $\mathbf{r}_{z,\theta_i}$  represents a pure rotation of  $\theta_i \in \mathbb{R}$  around the  $z$ -axis, whereas  $\mathbf{r}_{x,\alpha_i}$  represents a pure rotation of  $\alpha_i \in \mathbb{R}$  around the  $x$ -axis, given by

$$\begin{aligned} \mathbf{r}_{z,\theta_i} &= \cos\left(\frac{\theta_i}{2}\right) + \hat{k} \sin\left(\frac{\theta_i}{2}\right), \\ \mathbf{r}_{x,\alpha_i} &= \cos\left(\frac{\alpha_i}{2}\right) + \hat{i} \sin\left(\frac{\alpha_i}{2}\right), \end{aligned}$$

and  $\underline{\mathbf{p}}_{z,d_i}$  represents a pure translation of  $d_i \in \mathbb{R}$  along the  $z$ -axis, whereas  $\underline{\mathbf{p}}_{x,a_i}$  represents a pure translation of  $a_i \in \mathbb{R}$  along the  $x$ -axis, given by

$$\begin{aligned} \underline{\mathbf{p}}_{z,d_i} &= 1 + \varepsilon \frac{d_i}{2} \hat{k}, \\ \underline{\mathbf{p}}_{x,a_i} &= 1 + \varepsilon \frac{a_i}{2} \hat{i}. \end{aligned}$$

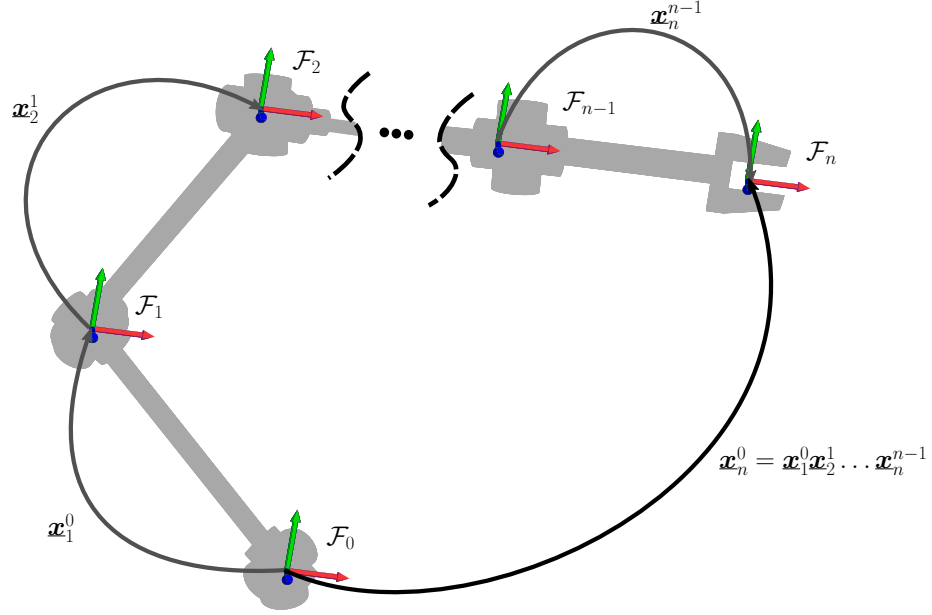
In brief, the processes of obtaining the forward kinematic model of a given robot consist of first finding the poses of each of its joints with respect to the previous one in the chain, which are functions of the current joint configurations, and finally multiplying them to obtain the end-effector pose, as given by (3.19). If using the D-H convention, then each joint pose is given by (3.20); otherwise, they must be found following the specific choices for the orientations of each reference frame and the corresponding positions of their origins.

### 3.3.3.1 Differential forward kinematic model of serially coupled kinematic structures

It is not uncommon in robotics to divide a complex kinematic structure into individual subsystems that are easier to model (e.g., one could divide a humanoid robot into legs, arms, and torso). In this scenario, one would be later interested in assembling the independently modeled subsystems to obtain the kinematic model of the overall robotic system. Adorno

---

<sup>6</sup>I would like to thank my colleague Juan José Quiroz Omaña for this illustration, which was made for a draft of one of our co-authored publications but was replaced for the final version (Silva et al., 2022, figure 1). This figure is a minor adaptation of his drawing.


 Figure 3.4: An  $n$ -DoF serial manipulator.

(2011) proposed a systematic methodology to achieve this goal using dual quaternion algebra.

Consider a serial kinematic system composed of  $s$  subsystems, each of them represented by the rigid transformations  $\underline{\mathbf{x}}_1, \underline{\mathbf{x}}_2, \dots, \underline{\mathbf{x}}_s$ . Let  $\underline{\mathbf{x}}_s^0$  be the transformation between the last frame  $\mathcal{F}_s$  and the inertial frame  $\mathcal{F}_0$ , its time derivative  $\dot{\underline{\mathbf{x}}}_s^0$  is given by (Adorno, 2011)

$$\begin{aligned} \text{vec}_8 \dot{\underline{\mathbf{x}}}_s^0 &= \sum_{i=0}^{s-1} \overset{+}{\mathbf{H}}_8(\underline{\mathbf{x}}_i^0) \bar{\mathbf{H}}_8(\underline{\mathbf{x}}_s^{i+1}) \text{vec}_8 \dot{\underline{\mathbf{x}}}_{i+1}^i, \\ &= \sum_{i=0}^{s-1} \underbrace{\overset{+}{\mathbf{H}}_8(\underline{\mathbf{x}}_i^0) \bar{\mathbf{H}}_8(\underline{\mathbf{x}}_s^{i+1}) \mathbf{J}_{\underline{\mathbf{x}}_{i+1}^i}}_{\mathbf{L}_{i+1}} \boldsymbol{\theta}_{i+1} \\ &= \sum_{i=0}^{s-1} \mathbf{L}_{i+1} \boldsymbol{\theta}_{i+1} \end{aligned}$$

where the operator  $\text{vec}_8 : \mathcal{H} \rightarrow \mathbb{R}^8$  maps the coefficients of a dual quaternion into an eight-dimensional column vector<sup>7</sup>,  $\mathbf{J}_{\underline{\mathbf{x}}_{i+1}^i}$  and  $\boldsymbol{\theta}_{i+1}$  are, respectively, the dual quaternion Jacobian matrix and the joint configurations of the subsystem represented by the rigid transformation  $\underline{\mathbf{x}}_{i+1}^i$ ,  $\mathbf{L}_{i+1} = \overset{+}{\mathbf{H}}_8(\underline{\mathbf{x}}_i^0) \bar{\mathbf{H}}_8(\underline{\mathbf{x}}_s^{i+1}) \mathbf{J}_{\underline{\mathbf{x}}_{i+1}^i}$ , in which the Hamilton operators are given by  $\overset{+}{\mathbf{H}}_8 : \mathcal{H} \rightarrow \mathbb{R}^{8 \times 8}$  and  $\bar{\mathbf{H}}_8 : \mathcal{H} \rightarrow \mathbb{R}^{8 \times 8}$ , such that  $\text{vec}_8(\underline{\mathbf{h}}_1 \underline{\mathbf{h}}_2) = \overset{+}{\mathbf{H}}_8(\underline{\mathbf{h}}_1) \text{vec}_8 \underline{\mathbf{h}}_2 = \bar{\mathbf{H}}_8(\underline{\mathbf{h}}_2) \text{vec}_8 \underline{\mathbf{h}}_1$ .

<sup>7</sup>Given  $\underline{\mathbf{h}} = h_1 + \hat{i}h_2 + \hat{j}h_3 + \hat{k}h_4 + \varepsilon(h_5 + \hat{i}h_6 + \hat{j}h_7 + \hat{k}h_8)$ ,  $\text{vec}_8 \underline{\mathbf{h}} = [h_1 \ \dots \ h_8]^T$ .

## 3.4 Conclusions

This chapter reviewed some fundamental mathematical concepts to this work. Sections 3.1 and 3.2 respectively reviewed the classic recursive Newton-Euler and Euler-Lagrange formulations. Section 3.3 presented the definitions, operations, properties and applications of dual quaternions in the representation of rigid transformations, including a brief revision of the dual quaternion forward kinematic model proposed by Adorno (2011) alongside his strategy for obtaining the differential forward kinematic model of serially coupled kinematic structures.

The next chapter is going to present the dual quaternion Newton-Euler formulation, which is one of the main contributions of this thesis.

# 4

## THE DUAL QUATERNION NEWTON-EULER FORMULATION

This chapter proposes the dynamic modeling of robotic systems using the dual quaternion Newton-Euler formulation, alongside a novel quaternionic inertia tensor that enables the description of rigid body dynamics exclusively with operations inside the dual quaternion algebra. This strategy covers serial kinematic chains and branched robots, works for arbitrary joint types, and does not impose any particular parameterization convention for twist/wrench propagation. The dual quaternion Newton-Euler formalism is one of the main contributions of this thesis.

The formulation proposed in this chapter considers two basic premises: that the full kinematic model of the robot is available using dual quaternion algebra (section 3.3.3; Adorno, 2011) and that the robot is composed of rigid bodies (i.e., the dynamic model disregards link deformations).

### 4.1 The inertia tensor

Before presenting the proposed dual quaternion Newton-Euler formulation, one crucial element to the process deserves special attention: the inertia tensor.

Over the last 60 years (Yang, 1967, 1971; Pennock & Yang, 1983; Shoham & Brodsky, 1993; Miranda de Farias et al., 2019a), several authors have attempted to find a suitable way to represent the inertia tensor in their dual quaternion dynamic modeling of rigid bodies.

For instance, Yang (1967) introduced the classic three-by-three matrix representation,

$$\bar{\mathbb{I}} = \begin{bmatrix} i_{xx} & i_{xy} & i_{xz} \\ i_{yx} & i_{yy} & i_{yz} \\ i_{zx} & i_{zy} & i_{zz} \end{bmatrix} \in \mathbb{R}^{3 \times 3}, \quad (4.1)$$

in his formulations, whereas Yang (1971) worked with the dyadic representation of the inertia tensor,

$$\mathbb{J} = i_{xx} \mathbf{i}\mathbf{i} + i_{xy} \mathbf{i}\mathbf{j} + i_{xz} \mathbf{i}\mathbf{k} + i_{yx} \mathbf{j}\mathbf{i} + i_{yy} \mathbf{j}\mathbf{j} + i_{yz} \mathbf{j}\mathbf{k} + i_{zx} \mathbf{k}\mathbf{i} + i_{zy} \mathbf{k}\mathbf{j} + i_{zz} \mathbf{k}\mathbf{k},$$

where  $\mathbf{i}$ ,  $\mathbf{j}$  and  $\mathbf{k}$  are unit vectors (Gürgöze & Zeren, 2012). In both approaches, dual quaternion algebra was mixed with standard matrix/tensor algebra. Since Yang (Yang, 1967, 1971) represented dual quaternions as dual vectors, the introduction of matrices/dyadics in his formulations did not demand the addition of new operators (*see* section 2.2).

Other authors opted to unify the representation of the mass and the inertia tensor of the body into a single element. Pennock & Yang (1983) explored the use of the inertia binor (Dimentberg, 1965)<sup>1</sup>

$$\Phi = \begin{bmatrix} \mathbf{S}^T & m\mathbf{I}_3 \\ \bar{\mathbb{I}} & \mathbf{S} \end{bmatrix} \in \mathbb{R}^{6 \times 6},$$

where  $\bar{\mathbb{I}}$  is given by (4.1) and  $\mathbf{I}_3$ ,  $\mathbf{S}$ , and  $m$  are given by (2.12). This approach, however, still does not include the inertia tensor as an element of the algebra since  $\Phi \in \mathbb{R}^{6 \times 6}$ . Alternatively, Shoham & Brodsky (1993) proposed the dual inertia

$$\underline{\mathbf{M}} = \begin{bmatrix} m \frac{d}{d\varepsilon} + \varepsilon i_{xx} & \varepsilon i_{xy} & \varepsilon i_{xz} \\ \varepsilon i_{yx} & m \frac{d}{d\varepsilon} + \varepsilon i_{yy} & \varepsilon i_{yz} \\ \varepsilon i_{zx} & \varepsilon i_{zy} & m \frac{d}{d\varepsilon} + \varepsilon i_{zz} \end{bmatrix},$$

which depends on the operator  $d/d\varepsilon$ , which in turn has its notational abuses discussed in section 2.2, and also demands the representation of dual quaternions as dual vectors.

More recently, Miranda de Farias et al. (2019a) proposed operator given by the *Dual Quaternion Inertia Transformation*

$$\mathcal{G}(\underline{\xi}) = m \mathcal{D}(\underline{\xi}) + \varepsilon (\mathbf{i}_x \omega_x + \mathbf{i}_y \omega_y + \mathbf{i}_z \omega_z), \quad (4.2)$$

where  $\mathcal{D}(\underline{\xi}) = v_x \hat{\mathbf{i}} + v_y \hat{\mathbf{j}} + v_z \hat{\mathbf{k}}$  is the linear velocity of the body, whereas  $\mathcal{P}(\underline{\xi}) = \omega_x \hat{\mathbf{i}} + \omega_y \hat{\mathbf{j}} + \omega_z \hat{\mathbf{k}}$  is its angular velocity, and  $\mathbf{i}_x = i_{xx} \hat{\mathbf{i}} + i_{xy} \hat{\mathbf{j}} + i_{xz} \hat{\mathbf{k}}$ ,  $\mathbf{i}_y = i_{yx} \hat{\mathbf{i}} + i_{yy} \hat{\mathbf{j}} + i_{yz} \hat{\mathbf{k}}$ , and  $\mathbf{i}_z = i_{zx} \hat{\mathbf{i}} + i_{zy} \hat{\mathbf{j}} + i_{zz} \hat{\mathbf{k}}$ , with  $i_{nn}$ ,  $n \in \{x, y, z\}$ , are elements of the inertia tensor (4.1).

<sup>1</sup>Originally named *second order inertia motor tensor* by Mises (1924) (quoted in Wohlhart 1995).

In this formulation, despite having its columns represented as pure quaternions, the *inertia tensor* does not appear as an element of the algebra, existing only as an internal part of the operator  $\mathcal{G}(\boldsymbol{\omega})$ . Thus, it is unclear how one could deal with inertia tensors represented in different reference frames.

The following section presents the formulation of a quaternionic inertia tensor that allows the dynamic modeling of rigid bodies to be entirely performed within dual quaternion algebra.

#### 4.1.1 The quaternionic inertia tensor

Let  $\mathbb{H}_p^3$  be the set defined as

$$\mathbb{H}_p^3 \triangleq \mathbb{H}_p \times \mathbb{H}_p \times \mathbb{H}_p,$$

where  $\mathbb{H}_p$  is the set of pure quaternions. That is,  $\mathbb{H}_p^3$  is the set of all ordered triples of pure quaternions—i.e., the set of all triples  $(\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3)$  where  $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3 \in \mathbb{H}_p$ .

Regarding the set  $\mathbb{H}_p^3$ , consider the following definitions.

**Definition 4.1.** Consider the three-dimensional inertia tensor (4.1) of a given rigid body and the set  $\mathbb{H}_p^3$ . The quaternionic inertia tensor  $\mathbb{I}$  of this rigid body is given by

$$\mathbb{I} \triangleq (\mathbf{i}_x, \mathbf{i}_y, \mathbf{i}_z) \in \mathbb{H}_p^3 \subset \mathcal{H}^n, \quad (4.3)$$

where  $\mathbf{i}_x = i_{xx}\hat{i} + i_{xy}\hat{j} + i_{xz}\hat{k}$ ,  $\mathbf{i}_y = i_{yx}\hat{i} + i_{yy}\hat{j} + i_{yz}\hat{k}$ , and  $\mathbf{i}_z = i_{zx}\hat{i} + i_{zy}\hat{j} + i_{zz}\hat{k}$ .

**Definition 4.2.** Given  $\mathbf{A} = (\mathbf{a}_x, \mathbf{a}_y, \mathbf{a}_z) \in \mathbb{H}_p^3$  and  $\mathbf{b} \in \mathbb{H}_p$ , the operator  $\mathcal{L}_3 : \mathbb{H}_p^3 \times \mathbb{H}_p \rightarrow \mathbb{H}_p$ , is defined as

$$\mathcal{L}_3(\mathbf{A})\mathbf{b} = \hat{i}\langle \mathbf{a}_x, \mathbf{b} \rangle + \hat{j}\langle \mathbf{a}_y, \mathbf{b} \rangle + \hat{k}\langle \mathbf{a}_z, \mathbf{b} \rangle, \quad (4.4)$$

where the inner product between quaternions is given by (3.7).

Since only standard quaternion multiplications and additions are used in definition 4.2 (see (3.7)), all elements and operations in (4.4) are from quaternion algebra, which in turn is a subset of dual quaternion algebra.

Considering the definitions 4.1 and 4.2, the following proposition presents how the angular momentum of a rigid body can be obtained within the dual quaternion formulation.

**Proposition 4.1.** *Given the quaternionic inertia tensor  $\mathbb{I} \in \mathbb{H}_p^3$  of a rigid body and its angular velocity  $\boldsymbol{\omega} \in \mathbb{H}_p$ , with  $\boldsymbol{\omega} = \omega_x\hat{i} + \omega_y\hat{j} + \omega_z\hat{k}$ , its angular momentum in dual quaternion algebra is given by*

$$\mathbf{l} = \mathcal{L}_3(\mathbb{I})\boldsymbol{\omega} \in \mathbb{H}_p. \quad (4.5)$$

*Proof.* Expanding (4.5), one finds

$$\begin{aligned} \mathbf{l} &= \hat{i}\langle \mathbf{i}_x, \boldsymbol{\omega} \rangle + \hat{j}\langle \mathbf{i}_y, \boldsymbol{\omega} \rangle + \hat{k}\langle \mathbf{i}_z, \boldsymbol{\omega} \rangle \\ &= (i_{xx}\omega_x + i_{xy}\omega_y + i_{xz}\omega_z)\hat{i} + (i_{yx}\omega_x + i_{yy}\omega_y + i_{yz}\omega_z)\hat{j} + (i_{zx}\omega_x + i_{zy}\omega_y + i_{zz}\omega_z)\hat{k}. \end{aligned} \quad (4.6)$$

Consider now a rigid body whose inertia tensor is given by (4.1) and whose angular velocity is  $\boldsymbol{\omega}_{\text{vec}} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T \in \mathbb{R}^3$ . Its angular momentum is given by

$$\begin{aligned} \mathbf{l}_{\text{vec}} &= \bar{\mathbb{I}}\boldsymbol{\omega}_{\text{vec}} \\ &= \begin{bmatrix} i_{xx} & i_{xy} & i_{xz} \\ i_{yx} & i_{yy} & i_{yz} \\ i_{zx} & i_{zy} & i_{zz} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} i_{xx}\omega_x + i_{xy}\omega_y + i_{xz}\omega_z \\ i_{yx}\omega_x + i_{yy}\omega_y + i_{yz}\omega_z \\ i_{zx}\omega_x + i_{zy}\omega_y + i_{zz}\omega_z \end{bmatrix}. \end{aligned} \quad (4.7)$$

The comparison between (4.6) and (4.7) shows that  $\mathbf{l}_{\text{vec}} = \text{vec}_3(\mathbf{l})$ , where  $\text{vec}_3 : \mathbb{H}_p \rightarrow \mathbb{R}^3$  such that  $\text{vec}_3(a\hat{i} + b\hat{j} + c\hat{k}) = \begin{bmatrix} a & b & c \end{bmatrix}^T$ . Therefore, the angular momentum of a rigid body whose quaternionic inertia tensor is given by  $\mathbb{I} \in \mathbb{H}_p^3$  and whose angular velocity is given by  $\boldsymbol{\omega} \in \mathbb{H}_p$ , is given by  $\mathbf{l} = \mathcal{L}_3(\mathbb{I})\boldsymbol{\omega} \in \mathbb{H}_p$ .  $\square$

The following examples show the applications of the proposed quaternionic inertia tensor  $\mathbb{I} \in \mathbb{H}_p^3$  and operator  $\mathcal{L}_3$  to problems of rigid body dynamics.

**Example 4.1.** Consider a coordinate change for the angular momentum given by (4.7). Given the rotation matrix  $\mathbf{R} \in SO(3)$  between the reference frames, then the angular momentum in the rotated frame is given by

$$\begin{aligned} \mathbf{l}'_{\text{vec}} &= \mathbf{R}\mathbf{l}_{\text{vec}} \\ &= \mathbf{R}\bar{\mathbb{I}}\boldsymbol{\omega}_{\text{vec}} \in \mathbb{R}^3. \end{aligned}$$

Alternatively,  $\mathbf{l}'_{\text{vec}}$  can be given as  $\mathbf{l}'_{\text{vec}} = \bar{\mathbb{I}}'\boldsymbol{\omega}'_{\text{vec}}$ , where  $\bar{\mathbb{I}}' = \mathbf{R}\bar{\mathbb{I}}\mathbf{R}^T$  and  $\boldsymbol{\omega}'_{\text{vec}} = \mathbf{R}\boldsymbol{\omega}_{\text{vec}}$  are the inertia tensor and the angular velocity in the rotated frame, respectively.

Analogously, given the unit quaternion  $\mathbf{r}$  representing the rotation between the reference frames, the angular momentum (4.4) in the rotated frame is given by

$$\begin{aligned} \mathbf{l}' &= \text{Ad}(\mathbf{r})\mathbf{l} \\ &= \text{Ad}(\mathbf{r})\mathcal{L}_3(\mathbb{I})\boldsymbol{\omega}. \end{aligned} \quad (4.8)$$

By direct calculation, it follows that  $\mathbf{l}'_{\text{vec}} = \text{vec}_3(\mathbf{l}')$ . Moreover, the expansion of (4.8)

leads to

$$\begin{aligned} \mathbf{l}' &= \text{Ad}(\mathbf{r}) \left( \hat{i} \langle \mathbf{i}_x, \boldsymbol{\omega} \rangle + \hat{j} \langle \mathbf{i}_y, \boldsymbol{\omega} \rangle + \hat{k} \langle \mathbf{i}_z, \boldsymbol{\omega} \rangle \right) \\ &= \langle \mathbf{i}_x, \boldsymbol{\omega} \rangle \text{Ad}(\mathbf{r}) \hat{i} + \langle \mathbf{i}_y, \boldsymbol{\omega} \rangle \text{Ad}(\mathbf{r}) \hat{j} + \langle \mathbf{i}_z, \boldsymbol{\omega} \rangle \text{Ad}(\mathbf{r}) \hat{k}, \end{aligned} \quad (4.9)$$

since the adjoint transformation is a linear operation and the result of the inner product of pure quaternions is a real number.

This demonstrates that the standard adjoint transformation of quaternions performs the rigid coordinate change of the angular momentum. The following example better illustrates how this transformation is analogous to the similarity transformation used in matrix algebra.

**Example 4.2.** Consider two reference frames,  $\mathcal{F}$  and  $\mathcal{F}'$ , and use the superscript of the variable to indicate its reference (e.g.,  $\mathbf{l}'_{\text{vec}}$  is given with respect to  $\mathcal{F}'$ , whereas  $\mathbf{l}$  is given with respect to  $\mathcal{F}$ ). Given the inertia tensor  $\bar{\mathbb{I}} \in \mathbb{R}^{3 \times 3}$  of a rigid body and its angular velocity  $\boldsymbol{\omega}'_{\text{vec}} \in \mathbb{R}^3$ , its angular momentum  $\mathbf{l}'_{\text{vec}} \in \mathbb{R}^3$  is

$$\mathbf{l}'_{\text{vec}} = \mathbf{R} \bar{\mathbb{I}} \mathbf{R}^T \boldsymbol{\omega}'_{\text{vec}},$$

where  $\mathbf{R} \in SO(3)$  is the rotation matrix between the reference frames  $\mathcal{F}'$  and  $\mathcal{F}$ . Similarly, given  $\mathbb{I} \in \mathbb{H}_p^3$  and  $\boldsymbol{\omega} \in \mathbb{H}_p$ , and using the previously established fact that  $\mathbf{l}' = \text{Ad}(\mathbf{r}) \mathbf{l}$ , the angular momentum  $\mathbf{l}'$  is

$$\begin{aligned} \mathbf{l}' &= \text{Ad}(\mathbf{r}) \mathbf{l} \\ &= \mathbf{r} \mathbf{l} \mathbf{r}^* \\ &= \mathbf{r} (\mathcal{L}_3(\mathbb{I}) \boldsymbol{\omega}) \mathbf{r}^* \\ &= \mathbf{r} [\mathcal{L}_3(\mathbb{I}) (\mathbf{r}^* \boldsymbol{\omega}' \mathbf{r})] \mathbf{r}^* \\ &= \text{Ad}(\mathbf{r}) \mathcal{L}_3(\mathbb{I}) (\mathbf{r}^* \boldsymbol{\omega}' \mathbf{r}) \\ &= \text{Ad}(\mathbf{r}) \mathcal{L}_3(\mathbb{I}) \text{Ad}(\mathbf{r}^*) \boldsymbol{\omega}', \end{aligned} \quad (4.10)$$

where  $\mathbf{r}$  is the rotation quaternion between the reference frames  $\mathcal{F}'$  and  $\mathcal{F}$ .

Conversely, given  $\bar{\mathbb{I}}' \in \mathbb{R}^{3 \times 3}$  and  $\boldsymbol{\omega}_{\text{vec}} \in \mathbb{R}^3$ , the angular momentum  $\mathbf{l}_{\text{vec}} \in \mathbb{R}^3$  is

$$\mathbf{l}_{\text{vec}} = \mathbf{R}^T \bar{\mathbb{I}}' \mathbf{R} \boldsymbol{\omega}_{\text{vec}}.$$

Analogously, given  $\bar{\mathbb{I}}' \in \mathbb{H}_p^3$  and  $\boldsymbol{\omega} \in \mathbb{H}_p$ , the angular momentum  $\mathbf{l}$  is

$$\mathbf{l} = \text{Ad}(\mathbf{r}^*) \mathcal{L}_3(\bar{\mathbb{I}}') \text{Ad}(\mathbf{r}) \boldsymbol{\omega}.$$

Therefore, the change of reference frames for the proposed inertia operator is not only



achieved by the standard adjoint transformation of quaternions, as it is also analogous to the matrix approach. Furthermore, all coordinate changes are performed within the dual quaternion algebra, and there is no need to perform intermediate operations over the three-by-three matrix inertia tensor  $\bar{\mathbb{I}}$ . Neither is there any need to map the angular velocities  $\boldsymbol{\omega} \in \mathbb{H}_p$  to vectors.

## 4.2 Dynamic modeling of serial kinematic chains

Section 3.3.3 presented the process of obtaining the end-effector pose of an  $n$ -DoF serial manipulator through a sequence of multiplications of intermediate rigid transformations. Similarly, twists and wrenches can be propagated throughout the kinematic chain to yield the dynamic information of the robot. From this proposition arises the dual quaternion Newton-Euler (dqNE) formalism.

### 4.2.1 Fixed-base robot manipulators

Consider the  $n$ -DoF serial manipulator presented in figure 4.1.<sup>2</sup> The goal is to find the wrenches acting on the robot joints, given the corresponding joints configurations, velocities, and accelerations. This section presents the dual quaternion Newton-Euler formalism as an iterative algorithm inspired by the classic version based on three-dimensional vectors proposed by Luh et al. (1980). The dqNE, however, works with arbitrary joint types, whereas the classic algorithm was designed to revolute and prismatic joints.

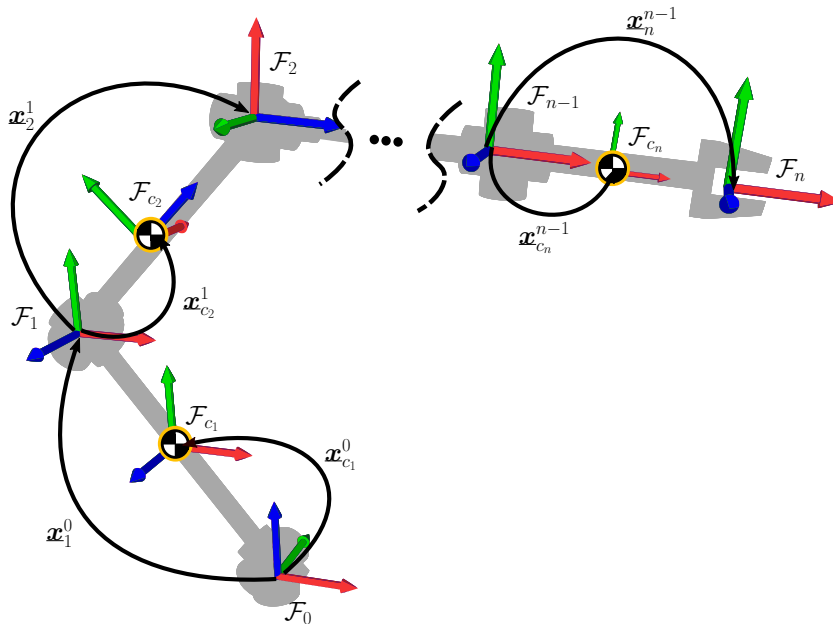


Figure 4.1: An  $n$ -DoF serial manipulator.

<sup>2</sup>I would like to thank my colleague Juan José Quiroz Omaña for this illustration, which was made for a draft of one of our co-authored publications but was replaced for the final version (Silva et al., 2022, figure 1). This figure is a minor adaptation of his drawing.

#### 4.2.1.1 Forward Recursion

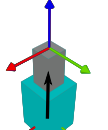
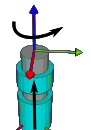
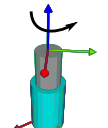
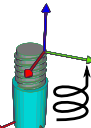
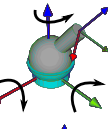
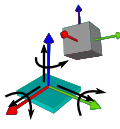
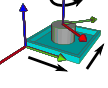
The first process of the iterative algorithm consists of a serial sweeping of the robot joints to calculate the twist of the CoM<sup>3</sup> of each link. The objective is to find the forward recurrence relations that will then be used to iteratively obtain the wrenches acting on the robot joints.

**Twists** The twist of the CoM of the first link with respect to the inertial frame  $\mathcal{F}_0$  is given by the pure dual quaternion

$$\underline{\xi}_{0,c_1}^0 = \omega_{0,c_1}^0 + \varepsilon \mathbf{v}_{0,c_1}^0, \quad (4.11)$$

where  $\omega_{0,c_1}^0 = \omega_x \hat{i} + \omega_y \hat{j} + \omega_z \hat{k}$  and  $\mathbf{v}_{0,c_1}^0 = v_x \hat{i} + v_y \hat{j} + v_z \hat{k}$  are, respectively, the angular and the linear velocities imposed by the movements of the first joint. Table 4.1 presents the twists for some of the most commonly used joints in robotics, where  $\mathbf{l} \in \mathbb{H}_p \cap \mathbb{S}^3$  is a constant unit-norm pure quaternion, which is equivalent to a vector in  $\mathbb{R}^3$ , that is used to define an arbitrary axis. For instance, when using the standard Denavit-Hartenberg convention,  $\mathbf{l} = \hat{k}$ , which is equivalent to the  $z$ -axis. Furthermore,  $\omega, \omega_x, \omega_y, \omega_z \in \mathbb{R}$  and  $v, v_x, v_y, v_z \in \mathbb{R}$  are the scalar components of the angular and linear velocities, respectively. Again, when using the DH convention,  $\omega = \dot{\theta}$  for a revolute joint and  $v = \dot{d}$  for a prismatic joint. For helical joints, the constant  $h \in \mathbb{R}$  is called the *pitch*.

Table 4.1: Twists of some of the most commonly used joints in robotics, where  $\mathbf{l} \in \mathbb{H}_p \cap \mathbb{S}^3$  and  $\omega, \omega_x, \omega_y, \omega_z, v, v_x, v_y, v_z, h \in \mathbb{R}$  (adapted from Silva et al., 2022, table 1).

Joint Type	Twist	Joint Type	Twist
Prismatic	 $\underline{\xi} = \varepsilon v \mathbf{l}$	Cylindrical	 $\underline{\xi} = (\omega + \varepsilon v) \mathbf{l}$
Revolute	 $\underline{\xi} = \omega \mathbf{l}$	Helical	 $\underline{\xi} = (\omega + \varepsilon h \omega) \mathbf{l}$
Spherical	 $\underline{\xi} = \omega_x \hat{i} + \omega_y \hat{j} + \omega_z \hat{k}$	6-DoF	 $\underline{\xi} = \omega_x \hat{i} + \omega_y \hat{j} + \omega_z \hat{k} + \varepsilon (v_x \hat{i} + v_y \hat{j} + v_z \hat{k})$
Planar	 $\underline{\xi} = \omega \mathbf{l} + \varepsilon (v_x \hat{i} + v_y \hat{j})$		

<sup>3</sup>Henceforth, I will use the expression “twist of the CoM” as a shorthand for “the twist of the frame attached to the center of mass.”

Using (3.11) and the fact that  $\underline{\mathbf{x}}_0^{c_1} = (\underline{\mathbf{x}}_0^0)^*$ , the twist (4.11) is expressed in  $\mathcal{F}_{c_1}$  as

$$\underline{\xi}_{0,c_1}^{c_1} = \text{Ad}(\underline{\mathbf{x}}_0^{c_1}) \underline{\xi}_{0,c_1}^0 = \omega_{0,c_1}^{c_1} + \varepsilon \left( \mathbf{v}_{0,c_1}^{c_1} + \omega_{0,c_1}^{c_1} \times \mathbf{p}_{0,c_1}^{c_1} \right), \quad (4.12)$$

where the linear velocity due to the application of an angular velocity in a point displaced from the CoM (i.e., at  $\mathcal{F}_0$ ) arises algebraically. Figure 4.2 illustrates this phenomenon when a purely rotational joint is used (i.e.,  $\underline{\xi}_{0,c_1}^0 = \omega_{0,c_1}^0 = \omega_1 \mathbf{l}_1^0$ , where  $\mathbf{l}_1^0 \in \mathbb{H}_p \cap \mathbb{S}^3$  gives the joint motion axis).

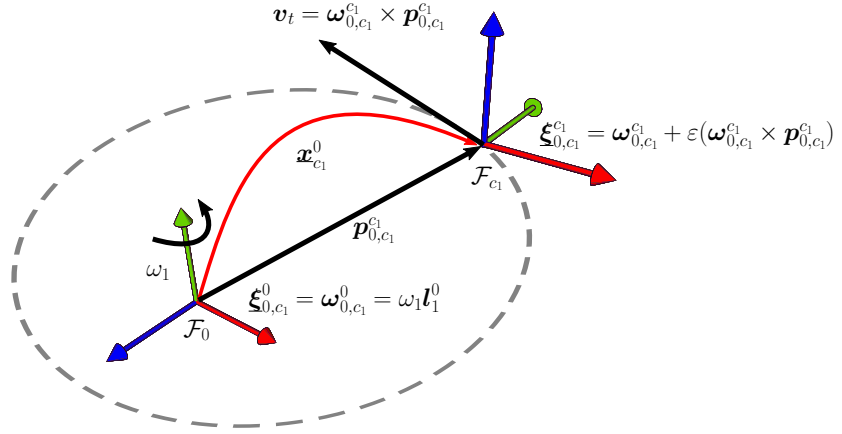


Figure 4.2: Twist  $\underline{\xi}_{0,c_1}^{c_1}$  generated due to the application of an angular velocity  $\omega_1$  around an arbitrary axis of the reference frame  $\mathcal{F}_0$ . The circular trajectory that  $\mathcal{F}_{c_1}$  follows is represented by the *dashed gray* line. The linear velocity due to the application of  $\omega_1$  appears algebraically through the adjoint transformation. Thus, the tangential velocity of the reference frame  $\mathcal{F}_{c_1}$ , represented as a solid black arrow, is given by the dual part of the twist  $\underline{\xi}_{0,c_1}^{c_1}$  (adapted from Silva et al., 2022, figure 2).

The CoM twist of the second link with respect to the inertial frame depends not only on the twist generated by its joint but also on the twist of the first link since those links are physically attached. Therefore, it is given by

$$\begin{aligned} \underline{\xi}_{0,c_2}^{c_2} &= \underline{\xi}_{0,1}^{c_2} + \underline{\xi}_{1,c_2}^{c_2} \\ &= \text{Ad}(\underline{\mathbf{x}}_{c_1}^{c_2}) \underline{\xi}_{0,1}^{c_1} + \text{Ad}(\underline{\mathbf{x}}_1^{c_2}) \underline{\xi}_{1,c_2}^1 \\ &= \text{Ad}(\underline{\mathbf{x}}_{c_1}^{c_2}) \left( \underline{\xi}_{0,c_1}^{c_1} + \underline{\xi}_{c_1,1}^{c_1} \right) + \text{Ad}(\underline{\mathbf{x}}_1^{c_2}) \underline{\xi}_{1,c_2}^1, \end{aligned} \quad (4.13)$$

where  $\underline{\xi}_{1,c_2}^1 = \omega_{1,c_2}^1 + \varepsilon \mathbf{v}_{1,c_2}^1$  is the twist of the CoM of the second link with respect to  $\mathcal{F}_1$ , which is generated by the second joint, and  $\text{Ad}(\underline{\mathbf{x}}_{c_1}^{c_2}) \left( \underline{\xi}_{0,c_1}^{c_1} + \underline{\xi}_{c_1,1}^{c_1} \right)$  is the twist generated by the first joint, but expressed in  $\mathcal{F}_{c_2}$  using the adjoint transformation (3.11). Moreover,  $\underline{\xi}_{c_1,1}^{c_1} = 0$  because  $\underline{\mathbf{x}}_1^{c_1} = 0$  since the link is rigid. Therefore,

$$\underline{\xi}_{0,c_2}^{c_2} = \text{Ad}(\underline{\mathbf{x}}_{c_1}^{c_2}) \underline{\xi}_{0,c_1}^{c_1} + \text{Ad}(\underline{\mathbf{x}}_1^{c_2}) \underline{\xi}_{1,c_2}^1.$$

More generally, the twist in  $\mathcal{F}_{c_i}$  that provides the motion of  $\mathcal{F}_{c_i}$  with respect to  $\mathcal{F}_0$ ,

which arises from the movement of the first  $i$  joints, is given by

$$\underline{\xi}_{0,c_i}^{c_i} = \underline{\xi}_{0,i-1}^{c_i} + \underline{\xi}_{i-1,c_i}^{c_i}, \quad (4.14)$$

where  $\underline{\xi}_{0,i-1}^{c_i}$  is the twist related to the motion of the first  $i-1$  joints and  $\underline{\xi}_{i-1,c_i}^{c_i}$  is the twist related to the motion of  $i$ -th joint. Also,  $\underline{\xi}_{c_{i-1},i-1}^{c_{i-1}} = 0$  because  $\underline{\dot{\mathbf{x}}}_{i-1}^{c_{i-1}} = 0$  for all  $i$  since all links in the robot are rigid bodies. Furthermore,  $\underline{\xi}_{0,0}^a = 0$  for any  $a$ .

Analyzing (4.11), (4.13), and (4.14), by induction, the recurrence equation for the total twist of the  $i$ -th CoM, which has the contribution of all joints up to the  $i$ -th joint, expressed in  $\mathcal{F}_{c_i}$ , is given by

$$\underline{\xi}_{0,c_i}^{c_i} = \text{Ad} \left( \underline{\mathbf{x}}_{c_{i-1}}^{c_i} \right) \left( \underline{\xi}_{0,c_{i-1}}^{c_{i-1}} + \underline{\xi}_{c_{i-1},i-1}^{c_{i-1}} \right) + \text{Ad} \left( \underline{\mathbf{x}}_{i-1}^{c_i} \right) \underline{\xi}_{i-1,c_i}^{i-1},$$

where  $c_0 \triangleq 0$ . Therefore,

$$\underline{\xi}_{0,c_i}^{c_i} = \text{Ad} \left( \underline{\mathbf{x}}_{c_{i-1}}^{c_i} \right) \underline{\xi}_{0,c_{i-1}}^{c_{i-1}} + \text{Ad} \left( \underline{\mathbf{x}}_{i-1}^{c_i} \right) \underline{\xi}_{i-1,c_i}^{i-1}. \quad (4.15)$$

Since the twist  $\underline{\xi}_{i-1,c_i}^{i-1}$  is generated by the  $i$ -th joint, its expression depends on which type the  $i$ -th joint is (*see* table 4.1). The transformation  $\underline{\mathbf{x}}_{c_{i-1}}^{c_i}$  is calculated as  $\underline{\mathbf{x}}_{c_{i-1}}^{c_i} = \left( \underline{\mathbf{x}}_{c_i}^0 \right)^* \underline{\mathbf{x}}_{c_{i-1}}^0$ , where  $\underline{\mathbf{x}}_{c_i}^0 = \underline{\mathbf{x}}_{c_{i-1}}^0 \underline{\mathbf{x}}_{i-1}^{c_{i-1}} \underline{\mathbf{x}}_{c_i}^{i-1}$ ,  $\underline{\mathbf{x}}_0^0 = 1$ , the transformation  $\underline{\mathbf{x}}_{i-1}^{c_{i-1}}$  is constant, and  $\underline{\mathbf{x}}_{c_i}^{i-1}$  is a function of  $q_i$ .

**Time derivative of the twists** Considering (3.17), the time derivative of (4.15) is given by

$$\begin{aligned} \dot{\underline{\xi}}_{0,c_i}^{c_i} = \text{Ad} \left( \underline{\mathbf{x}}_{c_{i-1}}^{c_i} \right) \dot{\underline{\xi}}_{0,c_{i-1}}^{c_{i-1}} + \underline{\xi}_{c_i,c_{i-1}}^{c_i} \times \left( \text{Ad} \left( \underline{\mathbf{x}}_{c_{i-1}}^{c_i} \right) \underline{\xi}_{0,c_{i-1}}^{c_{i-1}} \right) + \text{Ad} \left( \underline{\mathbf{x}}_{i-1}^{c_i} \right) \dot{\underline{\xi}}_{i-1,c_i}^{i-1} \\ + \underline{\xi}_{c_i,i-1}^{c_i} \times \left( \text{Ad} \left( \underline{\mathbf{x}}_{i-1}^{c_i} \right) \underline{\xi}_{i-1,c_i}^{i-1} \right). \end{aligned}$$

Since  $\underline{\xi}_{i-1,c_i}^{c_i} = -\underline{\xi}_{c_i,i-1}^{c_i}$ , then

$$\underline{\xi}_{c_i,i-1}^{c_i} \times \left( \text{Ad} \left( \underline{\mathbf{x}}_{i-1}^{c_i} \right) \underline{\xi}_{i-1,c_i}^{i-1} \right) = -\underline{\xi}_{c_i,i-1}^{c_i} \times \underline{\xi}_{c_i,i-1}^{c_i} = 0.$$

Therefore,

$$\dot{\underline{\xi}}_{0,c_i}^{c_i} = \text{Ad} \left( \underline{\mathbf{x}}_{c_{i-1}}^{c_i} \right) \dot{\underline{\xi}}_{0,c_{i-1}}^{c_{i-1}} + \text{Ad} \left( \underline{\mathbf{x}}_{i-1}^{c_i} \right) \dot{\underline{\xi}}_{i-1,c_i}^{i-1} + \underline{\xi}_{c_i,c_{i-1}}^{c_i} \times \left[ \text{Ad} \left( \underline{\mathbf{x}}_{c_{i-1}}^{c_i} \right) \underline{\xi}_{0,c_{i-1}}^{c_{i-1}} \right], \quad (4.16)$$

where  $\dot{\underline{\xi}}_{0,c_0}^{c_0} \triangleq 0$ . Also, since  $\underline{\xi}_{c_i,c_{i-1}}^{i-1} = \underline{\xi}_{c_i,i-1}^{i-1} + \underline{\xi}_{i-1,c_{i-1}}^{i-1}$  and  $\underline{\xi}_{i-1,c_{i-1}}^{i-1} = 0$ , then

$$\underline{\xi}_{c_i,c_{i-1}}^{c_i} = \text{Ad} \left( \underline{\mathbf{x}}_{i-1}^{c_i} \right) \underline{\xi}_{c_i,i-1}^{i-1} = -\text{Ad} \left( \underline{\mathbf{x}}_{i-1}^{c_i} \right) \underline{\xi}_{i-1,c_i}^{i-1}. \quad (4.17)$$

As previously discussed, the twist  $\underline{\xi}_{i-1,c_i}^{i-1}$  depends on the type of the  $i$ -th joint and,

therefore, so does the term  $\underline{\xi}_{i-1,c_i}^{i-1}$ . For instance, if the  $i$ -th joint is revolute, then  $\underline{\xi}_{i-1,c_i}^{i-1} = \dot{\omega}_i \mathbf{l}_i^{i-1}$ . If it is prismatic, then  $\underline{\xi}_{i-1,c_i}^{i-1} = \varepsilon \dot{v}_i \mathbf{l}_i^{i-1}$ . Analogously, if it is helical, then  $\underline{\xi}_{i-1,c_i}^{i-1} = (\dot{\omega}_i + \varepsilon h \dot{\omega}_i) \mathbf{l}_i^{i-1}$ , etc.

*Remark 4.1.* Although (4.16) is written in recursive form, one can always write twists as in (3.13) and (3.14). Thus, as  $\underline{\xi}_{0,c_i}^{c_i} = \text{Ad}(\underline{\mathbf{x}}_0^{c_i}) \underline{\xi}_{0,c_i}^0$ , with  $\underline{\xi}_{0,c_i}^0 = \boldsymbol{\omega}_{0,c_i}^0 + \varepsilon (\dot{\mathbf{p}}_{0,c_i}^0 + \mathbf{p}_{0,c_i}^0 \times \boldsymbol{\omega}_{0,c_i}^0)$ , from (3.17) one obtains

$$\underline{\dot{\xi}}_{0,c_i}^{c_i} = \text{Ad}(\underline{\mathbf{x}}_0^{c_i}) \underline{\dot{\xi}}_{0,c_i}^0 \quad (4.18)$$

because  $\underline{\xi}_{c_i,0}^{c_i} \times \text{Ad}(\underline{\mathbf{x}}_0^{c_i}) \underline{\xi}_{0,c_i}^0 = -\underline{\xi}_{0,c_i}^{c_i} \times \underline{\xi}_{0,c_i}^{c_i} = 0$ . Expanding (4.18) leads to

$$\begin{aligned} \underline{\dot{\xi}}_{0,c_i}^{c_i} &= \text{Ad}(\underline{\mathbf{x}}_0^{c_i}) \left( \boldsymbol{\omega}_{0,c_i}^0 + \varepsilon (\dot{\mathbf{p}}_{0,c_i}^0 + \dot{\mathbf{p}}_{0,c_i}^0 \times \boldsymbol{\omega}_{0,c_i}^0 + \mathbf{p}_{0,c_i}^0 \times \dot{\boldsymbol{\omega}}_{0,c_i}^0) \right) \\ &= \text{Ad}(\underline{\mathbf{x}}_0^{c_i}) \boldsymbol{\omega}_{0,c_i}^0 + \text{Ad}(\underline{\mathbf{x}}_0^{c_i}) \varepsilon (\dot{\mathbf{p}}_{0,c_i}^0 + \dot{\mathbf{p}}_{0,c_i}^0 \times \boldsymbol{\omega}_{0,c_i}^0 + \mathbf{p}_{0,c_i}^0 \times \dot{\boldsymbol{\omega}}_{0,c_i}^0), \end{aligned}$$

which, using lemma 3.1, simplifies to

$$\begin{aligned} \underline{\dot{\xi}}_{0,c_i}^{c_i} &= \text{Ad}(\underline{\mathbf{x}}_0^{c_i}) \boldsymbol{\omega}_{0,c_i}^0 + \text{Ad}(\mathbf{r}_0^{c_i}) \varepsilon (\ddot{\mathbf{p}}_{0,c_i}^0 + \dot{\mathbf{p}}_{0,c_i}^0 \times \boldsymbol{\omega}_{0,c_i}^0 + \mathbf{p}_{0,c_i}^0 \times \dot{\boldsymbol{\omega}}_{0,c_i}^0) \\ &= \boldsymbol{\omega}_{0,c_i}^{c_i} + \varepsilon (\dot{\boldsymbol{\omega}}_{0,c_i}^{c_i} \times \mathbf{p}_{0,c_i}^0) + \varepsilon (\ddot{\mathbf{p}}_{0,c_i}^{c_i} + \dot{\mathbf{p}}_{0,c_i}^{c_i} \times \boldsymbol{\omega}_{0,c_i}^{c_i} + \mathbf{p}_{0,c_i}^{c_i} \times \dot{\boldsymbol{\omega}}_{0,c_i}^{c_i}) \\ &= \boldsymbol{\omega}_{0,c_i}^{c_i} + \varepsilon (\ddot{\mathbf{p}}_{0,c_i}^{c_i} + \dot{\mathbf{p}}_{0,c_i}^{c_i} \times \boldsymbol{\omega}_{0,c_i}^{c_i}), \end{aligned} \quad (4.19)$$

since  $\dot{\boldsymbol{\omega}}_{0,c_i}^{c_i} \times \mathbf{p}_{0,c_i}^0 + \mathbf{p}_{0,c_i}^{c_i} \times \dot{\boldsymbol{\omega}}_{0,c_i}^{c_i} = -\mathbf{p}_{0,c_i}^{c_i} \times \dot{\boldsymbol{\omega}}_{0,c_i}^{c_i} + \mathbf{p}_{0,c_i}^{c_i} \times \dot{\boldsymbol{\omega}}_{0,c_i}^{c_i} = 0$ . Moreover, since  $\mathcal{D}(\underline{\dot{\xi}}_{0,c_i}^{c_i}) = \ddot{\mathbf{p}}_{0,c_i}^{c_i} + \dot{\mathbf{p}}_{0,c_i}^{c_i} \times \boldsymbol{\omega}_{0,c_i}^{c_i}$ , then

$$\ddot{\mathbf{p}}_{0,c_i}^{c_i} = \mathcal{D}(\underline{\dot{\xi}}_{0,c_i}^{c_i}) - \mathcal{D}(\underline{\xi}_{0,c_i}^{c_i}) \times \mathcal{P}(\underline{\xi}_{0,c_i}^{c_i}). \quad (4.20)$$

#### 4.2.1.2 Backward Recursion

The second process of the iterative algorithm consists of sweeping the serial robot from the last to the first joint to calculate the wrenches applied at each one of them. To that aim, let us use the twists previously obtained and their time derivatives.

Before obtaining the general expression for the backward recursion, let us continue with the example from the previous section; that is, consider the  $n$ -DoF serial manipulator shown in figure 4.1. The wrench at the CoM of the last link, expressed in  $\mathcal{F}_{c_n}$ , is given by the pure dual quaternion

$$\underline{\xi}_{0,c_n}^{c_n} = \underline{\xi}_{0,c_n}^{c_n} - m_n \mathbf{g}^{c_n}, \quad (4.21)$$

where  $m_n \mathbf{g}^{c_n}$  is the gravitational component, in which  $\mathbf{g}^{c_n} \in \mathbb{H}_p$  is the gravity acceleration expressed in  $\mathcal{F}_{c_n}$ , and  $\underline{\xi}_{0,c_n}^{c_n} = \mathbf{f}_{0,c_n}^{c_n} + \varepsilon \boldsymbol{\tau}_{0,c_n}^{c_n}$ , with  $\mathbf{f}_{0,c_n}^{c_n} = f_x \hat{i} + f_y \hat{j} + f_z \hat{k}$  being the force at the CoM of the last link, given by Newton's second law  $\mathbf{f}_{0,c_n}^{c_n} = m_n \ddot{\mathbf{p}}_{0,c_n}^{c_n}$ . Taking (4.20)

into consideration, the force  $\mathbf{f}_{0,c_n}^{c_n}$  is given by

$$\mathbf{f}_{0,c_n}^{c_n} = m_n \left( \mathcal{D} \left( \dot{\underline{\boldsymbol{\xi}}}_{0,c_n}^{c_n} \right) + \mathcal{P} \left( \underline{\boldsymbol{\xi}}_{0,c_n}^{c_n} \right) \times \mathcal{D} \left( \underline{\boldsymbol{\xi}}_{0,c_n}^{c_n} \right) \right). \quad (4.22)$$

Furthermore,  $\boldsymbol{\tau}_{0,c_n}^{c_n}$  is the torque about the CoM of the last link due to the change of its angular momentum, given by the Euler's rotation equation (Siciliano et al., 2009, p. 284)

$$\begin{aligned} \boldsymbol{\tau}_{0,c_n}^{c_n} &= \mathcal{L}_3 \left( \mathbb{I}_n^{c_n} \right) \dot{\boldsymbol{\omega}}_{0,c_n}^{c_n} + \boldsymbol{\omega}_{0,c_n}^{c_n} \times \left( \mathcal{L}_3 \left( \mathbb{I}_n^{c_n} \right) \boldsymbol{\omega}_{0,c_n}^{c_n} \right) \\ &= \mathcal{L}_3 \left( \mathbb{I}_n^{c_n} \right) \mathcal{P} \left( \dot{\underline{\boldsymbol{\xi}}}_{0,c_n}^{c_n} \right) + \mathcal{P} \left( \underline{\boldsymbol{\xi}}_{0,c_n}^{c_n} \right) \times \left( \mathcal{L}_3 \left( \mathbb{I}_n^{c_n} \right) \mathcal{P} \left( \underline{\boldsymbol{\xi}}_{0,c_n}^{c_n} \right) \right), \end{aligned} \quad (4.23)$$

where  $\mathcal{L}_3$  is given by (4.4) and  $\mathbb{I}_n^{c_n} \in \mathbb{H}_p^3$  is the quaternionic inertia tensor of the last link, expressed at its CoM (see definition 4.1). Because (4.23) is calculated with respect to the CoM, the gravity acceleration does not contribute to the torque.

Using the adjoint transformation (3.11) in (4.21), the wrench at the last joint, resulting from the wrench at the CoM of the last link, is given by

$$\underline{\boldsymbol{\zeta}}_{0,n}^{n-1} = \text{Ad} \left( \mathbf{x}_{c_n}^{n-1} \right) \underline{\boldsymbol{\zeta}}_{0,c_n}^{c_n}. \quad (4.24)$$

The resultant wrench at the penultimate joint includes the effects of the wrenches from the penultimate and the last links as they are rigidly attached to each other. Therefore, the resultant wrench at the penultimate joint is given by

$$\underline{\boldsymbol{\zeta}}_{0,n-1}^{n-2} = \text{Ad} \left( \mathbf{x}_{c_{n-1}}^{n-2} \right) \underline{\boldsymbol{\zeta}}_{0,c_{n-1}}^{c_{n-1}} + \text{Ad} \left( \mathbf{x}_{n-1}^{n-2} \right) \underline{\boldsymbol{\zeta}}_{0,n}^{n-1}, \quad (4.25)$$

where  $\underline{\boldsymbol{\zeta}}_{0,c_{n-1}}^{c_{n-1}} = \underline{\boldsymbol{\zeta}}_{0,c_{n-1}}^{c_{n-1}} - m_{n-1} \mathbf{g}^{n-1}$ , with  $\underline{\boldsymbol{\zeta}}_{0,c_{n-1}}^{c_{n-1}} = \mathbf{f}_{0,c_{n-1}}^{c_{n-1}} + \varepsilon \boldsymbol{\tau}_{0,c_{n-1}}^{c_{n-1}}$ , is the wrench at the CoM of the penultimate link expressed in  $\mathcal{F}_{c_{n-1}}$ .

Thus, analyzing (4.21), (4.24), and (4.25), the backward recurrence relation for the total wrench at the  $i$ -th joint, which includes the contribution of all wrenches starting at the CoM of the last link up to the wrench at the CoM of the  $i$ -th one, expressed in  $\mathcal{F}_{i-1}$ , is given by

$$\underline{\boldsymbol{\zeta}}_{0,i}^{i-1} = \text{Ad} \left( \mathbf{x}_{c_i}^{i-1} \right) \underline{\boldsymbol{\zeta}}_{0,c_i}^{c_i} + \text{Ad} \left( \mathbf{x}_i^{i-1} \right) \underline{\boldsymbol{\zeta}}_{0,i+1}^i, \quad (4.26)$$

with  $i \in \{1, \dots, n\}$ , where  $n$  is the number of rigid bodies in the kinematic chain,  $\underline{\boldsymbol{\zeta}}_{0,n+1}^n = 0$ , and  $\underline{\boldsymbol{\zeta}}_{0,c_i}^{c_i} = \underline{\boldsymbol{\zeta}}_{0,c_i}^{c_i} - m_i \mathbf{g}^{c_i}$ , with  $\underline{\boldsymbol{\zeta}}_{0,c_i}^{c_i} = \mathbf{f}_{0,c_i}^{c_i} + \varepsilon \boldsymbol{\tau}_{0,c_i}^{c_i}$ , is the wrench at the  $i$ -th CoM,<sup>4</sup>  $\mathbf{f}_{0,c_i}^{c_i} = m_i \left( \mathcal{D} \left( \dot{\underline{\boldsymbol{\xi}}}_{0,c_i}^{c_i} \right) + \mathcal{P} \left( \underline{\boldsymbol{\xi}}_{0,c_i}^{c_i} \right) \times \mathcal{D} \left( \underline{\boldsymbol{\xi}}_{0,c_i}^{c_i} \right) \right)$ , and  $\boldsymbol{\tau}_{0,c_i}^{c_i} = \mathcal{L}_3 \left( \mathbb{I}_i^{c_i} \right) \mathcal{P} \left( \dot{\underline{\boldsymbol{\xi}}}_{0,c_i}^{c_i} \right) + \mathcal{P} \left( \underline{\boldsymbol{\xi}}_{0,c_i}^{c_i} \right) \times \left( \mathcal{L}_3 \left( \mathbb{I}_i^{c_i} \right) \mathcal{P} \left( \underline{\boldsymbol{\xi}}_{0,c_i}^{c_i} \right) \right)$ . Moreover, the transformation  $\mathbf{x}_{c_i}^{i-1}$  is a function of the joint configuration  $q_i$ .

In the case of robots with revolute and/or prismatic joints, which are the most common

---

<sup>4</sup>If an *external* wrench is applied at the end-effector, then  $\underline{\boldsymbol{\zeta}}_{0,n+1}^n \neq 0$ .

ones, the wrenches given by (4.26) are projected onto the joints motion axes through<sup>5</sup>

$$\langle \underline{\zeta}_{0,i}^{i-1}, \mathbf{l}_i^{i-1} \rangle = f_i + \varepsilon \tau_i, \quad (4.27)$$

where  $f_i, \tau_i \in \mathbb{R}$  and  $\langle \underline{\zeta}_{0,i}^{i-1}, \mathbf{l}_i^{i-1} \rangle$  is the inner product between the wrench  $\underline{\zeta}_{0,i}^{i-1} = \mathbf{f}_{0,i}^{i-1} + \varepsilon \boldsymbol{\tau}_{0,i}^{i-1} \in \mathcal{H}_p$  and the motion axis  $\mathbf{l}_i^{i-1} \in \mathbb{H}_p \cap \mathbb{S}^3 \subset \mathcal{H}_p$  of the  $i$ -th joint, given by (3.15). Therefore, if the  $i$ -th joint is revolute, then the corresponding torque is given by  $\tau_i = \mathcal{D}(\langle \underline{\zeta}_{0,i}^{i-1}, \mathbf{l}_i^{i-1} \rangle)$ . If it is prismatic, then the corresponding force along the axis  $\mathbf{l}_i^{i-1}$  is given by  $f_i = \mathcal{P}(\langle \underline{\zeta}_{0,i}^{i-1}, \mathbf{l}_i^{i-1} \rangle)$ .

#### 4.2.1.3 Summary

Algorithms 4.1, 4.2, 4.3, and 4.4 summarize the proposed dual quaternion Newton-Euler formalism.

---

**Algorithm 4.1** Dual Quaternion Newton-Euler Algorithm.

---

- 1: **function** NEWTON\_EULER( $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ )
  - 2:      $(\Xi, \dot{\Xi}) \leftarrow$  FORWARD\_RECURSION( $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ )
  - 3:      $\Gamma \leftarrow$  BACKWARD\_RECURSION( $\Xi, \dot{\Xi}$ )
  - 4:     **return**  $\Gamma$
  - 5: **end function**
- 

---

**Algorithm 4.2** Forward recursion to obtain the twists and their derivatives for the CoM of all robot links.

---

- 1: **function** FORWARD\_RECURSION( $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ )
  - 2:      $\underline{\xi}_{0,c_0}^{c_0} \leftarrow 0$  and  $\dot{\underline{\xi}}_{0,c_0}^{c_0} \leftarrow 0$
  - 3:     **for**  $i = 1$  to  $n$  **do**
  - 4:          $(\underline{\xi}_{i-1,c_i}^{i-1}, \dot{\underline{\xi}}_{i-1,c_i}^{i-1}) \leftarrow$  JOINT\_TWIST( $\dot{\mathbf{q}}, \ddot{\mathbf{q}}$ )
  - 5:         ▷ Calculation of the  $i$ -th CoM twist
  - 6:          $\underline{\xi}_{0,c_i}^{c_i} \leftarrow \text{Ad}(\mathbf{x}_{c_{i-1}}^{c_i}) \underline{\xi}_{0,c_{i-1}}^{c_{i-1}} + \text{Ad}(\mathbf{x}_{i-1}^{c_i}) \underline{\xi}_{i-1,c_i}^{i-1}$
  - 7:         ▷ Calculation of the  $i$ -th CoM twist derivative
  - 8:          $\dot{\underline{\xi}}_{c_i,c_{i-1}}^{c_i} \leftarrow -\text{Ad}(\mathbf{x}_{i-1}^{c_i}) \dot{\underline{\xi}}_{i-1,c_i}^{i-1}$
  - 9:          $\dot{\underline{\xi}}_{0,c_i}^{c_i} \leftarrow \text{Ad}(\mathbf{x}_{c_{i-1}}^{c_i}) \dot{\underline{\xi}}_{0,c_{i-1}}^{c_{i-1}} + \text{Ad}(\mathbf{x}_{i-1}^{c_i}) \dot{\underline{\xi}}_{i-1,c_i}^{i-1} + \underline{\xi}_{c_i,c_{i-1}}^{c_i} \times [\text{Ad}(\mathbf{x}_{c_{i-1}}^{c_i}) \underline{\xi}_{0,c_{i-1}}^{c_{i-1}}]$
  - 10:     **end for**
  - 11:      $\Xi \leftarrow (\underline{\xi}_{0,c_1}^{c_1}, \dots, \underline{\xi}_{0,c_n}^{c_n})$
  - 12:      $\dot{\Xi} \leftarrow (\dot{\underline{\xi}}_{0,c_1}^{c_1}, \dots, \dot{\underline{\xi}}_{0,c_n}^{c_n})$
  - 13:     **return**  $(\Xi, \dot{\Xi})$
  - 14: **end function**
- 

<sup>5</sup>Notice that for joints with more than one axis of movement (e.g., spherical joints, planar joints, etc.), a similar procedure applies, where the wrenches are projected onto all the axes of motion of the joint.

**Algorithm 4.3** Function to obtain the twists of some of the most commonly used joints in robotics.

---

```

1: function JOINT_TWIST( $\dot{\mathbf{q}}, \mathbf{q}$ )
2:   if revolute joint then
3:      $\underline{\xi}_{i-1,c_i}^{i-1} \leftarrow \omega_i \mathbf{l}_i^{i-1}$  and  $\dot{\underline{\xi}}_{i-1,c_i}^{i-1} \leftarrow \dot{\omega}_i \mathbf{l}_i^{i-1}$ 
4:   else if prismatic joint then
5:      $\underline{\xi}_{i-1,c_i}^{i-1} \leftarrow \varepsilon v_i \mathbf{l}_i^{i-1}$  and  $\dot{\underline{\xi}}_{i-1,c_i}^{i-1} \leftarrow \varepsilon \dot{v}_i \mathbf{l}_i^{i-1}$ 
6:   else if spherical joint then
7:      $\underline{\xi}_{i-1,c_i}^{i-1} \leftarrow \omega_{i_x} \hat{i} + \omega_{i_y} \hat{j} + \omega_{i_z} \hat{k}$  and  $\dot{\underline{\xi}}_{i-1,c_i}^{i-1} \leftarrow \dot{\omega}_{i_x} \hat{i} + \dot{\omega}_{i_y} \hat{j} + \dot{\omega}_{i_z} \hat{k}$ 
8:   else if planar joint then
9:      $\underline{\xi}_{i-1,c_i}^{i-1} \leftarrow \omega_i \mathbf{l}_i^{i-1} + \varepsilon (v_{i_x} \hat{i} + v_{i_y} \hat{j})$  and  $\dot{\underline{\xi}}_{i-1,c_i}^{i-1} \leftarrow \dot{\omega}_i \mathbf{l}_i^{i-1} + \varepsilon (\dot{v}_{i_x} \hat{i} + \dot{v}_{i_y} \hat{j})$ 
10:  else if cylindrical joint then
11:     $\underline{\xi}_{i-1,c_i}^{i-1} \leftarrow (\omega_i + \varepsilon v_i) \mathbf{l}_i^{i-1}$  and  $\dot{\underline{\xi}}_{i-1,c_i}^{i-1} \leftarrow (\dot{\omega}_i + \varepsilon \dot{v}_i) \mathbf{l}_i^{i-1}$ 
12:  else if helical joint then
13:     $\underline{\xi}_{i-1,c_i}^{i-1} \leftarrow (\omega_i + \varepsilon h_i \omega_i) \mathbf{l}_i^{i-1}$  and  $\dot{\underline{\xi}}_{i-1,c_i}^{i-1} \leftarrow (\dot{\omega}_i + \varepsilon h_i \dot{\omega}_i) \mathbf{l}_i^{i-1}$ 
14:  else if 6-DoF joint then
15:     $\underline{\xi}_{i-1,c_i}^{i-1} \leftarrow \omega_{i_x} \hat{i} + \omega_{i_y} \hat{j} + \omega_{i_z} \hat{k} + \varepsilon (v_{i_x} \hat{i} + v_{i_y} \hat{j} + v_{i_z} \hat{k})$  and  $\dot{\underline{\xi}}_{i-1,c_i}^{i-1} \leftarrow \dot{\omega}_{i_x} \hat{i} + \dot{\omega}_{i_y} \hat{j} + \dot{\omega}_{i_z} \hat{k} + \varepsilon (\dot{v}_{i_x} \hat{i} + \dot{v}_{i_y} \hat{j} + \dot{v}_{i_z} \hat{k})$ 
16:  end if
17:  return ( $\underline{\xi}_{i-1,c_i}^{i-1}, \dot{\underline{\xi}}_{i-1,c_i}^{i-1}$ )
18: end function

```

---

**Algorithm 4.4** Backward recursion to obtain the wrenches at the robot joints.

---

```

1: function BACKWARD_RECURSION( $\Xi, \dot{\Xi}$ )
2:    $\underline{\zeta}_{0,n+1}^n \leftarrow$  external_wrench
3:   for  $i = n$  to 1 do
4:      $\underline{\xi}_{0,c_i}^{c_i} \leftarrow \Xi[i]$  and  $\dot{\underline{\xi}}_{0,c_i}^{c_i} \leftarrow \dot{\Xi}[i]$ 
5:      $\mathbf{f}_{0,c_i}^{c_i} \leftarrow m_i (\mathcal{D}(\underline{\xi}_{0,c_i}^{c_i}) + \mathcal{P}(\underline{\xi}_{0,c_i}^{c_i}) \times \mathcal{D}(\dot{\underline{\xi}}_{0,c_i}^{c_i}))$ 
6:      $\boldsymbol{\tau}_{0,c_i}^{c_i} \leftarrow \mathcal{L}_3(\mathbb{I}_i^{c_i}) \mathcal{P}(\dot{\underline{\xi}}_{0,c_i}^{c_i}) + \mathcal{P}(\underline{\xi}_{0,c_i}^{c_i}) \times (\mathcal{L}_3(\mathbb{I}_i^{c_i}) \mathcal{P}(\underline{\xi}_{0,c_i}^{c_i}))$ 
7:      $\underline{\zeta}_{0,c_i}^{c_i} \leftarrow \mathbf{f}_{0,c_i}^{c_i} + \varepsilon \boldsymbol{\tau}_{0,c_i}^{c_i} - m_i \mathbf{g}^{c_i}$ 
8:      $\triangleright$  Let  $\Gamma[i] = \underline{\zeta}_{0,i}^{i-1}$ 
9:      $\Gamma[i] \leftarrow \text{Ad}(\mathbf{x}_{c_i}^{i-1}) \underline{\zeta}_{0,c_i}^{c_i} + \text{Ad}(\mathbf{x}_i^{i-1}) \underline{\zeta}_{0,i+1}^i$ 
10:  end for
11:  return  $\Gamma$ 
12: end function

```

---



#### 4.2.1.4 Cost comparison

The comparison between the proposed dqNE and its classic counterpart is made in terms of number of multiplications and additions involved in each method. To that aim, consider the cost operator  $C(\text{op})$  that is used to calculate the cost of the operation  $\text{op}$  as a function of the cost of simpler operations (Adorno, 2011). For instance, given  $\underline{\mathbf{a}}, \underline{\mathbf{b}} \in \mathcal{H}$ , the cost of their multiplication is given by  $C(\underline{\mathbf{a}}\underline{\mathbf{b}})$  and, since  $\text{Ad}(\underline{\mathbf{a}})\underline{\mathbf{b}} = \underline{\mathbf{a}}\underline{\mathbf{b}}\underline{\mathbf{a}}^*$  then  $C(\text{Ad}(\underline{\mathbf{a}})\underline{\mathbf{b}}) = 2C(\underline{\mathbf{a}}\underline{\mathbf{b}}) + C(\underline{\mathbf{a}}^*)$ . In other words, the cost of one adjoint operation is equivalent to the cost of two dual quaternion multiplications plus one dual quaternion conjugation. Table 4.2 summarizes the cost of elementary operations used throughout this section.

Table 4.2: Cost of operations with quaternions and dual quaternions in terms of multiplication and addition of real numbers.

	Multiplications	Additions
<hr/> <hr/>		
Quaternions ( $\mathbb{I} \in \mathbb{H}_p^3$ , $\mathbf{a}, \mathbf{b} \in \mathbb{H}$ , and $\lambda \in \mathbb{R}$ )		
$C(\mathcal{L}_3(\mathbb{I})\mathbf{a})$ (see equation (4.4))	9	6
$C(\lambda\mathbf{a})$	4	0
$C(\mathbf{a}^*)$	3	0
$C(\mathbf{a} + \mathbf{b})$	0	4
$C(\mathbf{a}\mathbf{b})$	16	12
$C(\mathbf{a} \times \mathbf{b}) = 2C(\mathbf{a}\mathbf{b}) + C(\mathbf{a} + \mathbf{b}) + C(\lambda\mathbf{a})$	36	28
$C(\text{Ad}(\mathbf{a})\mathbf{b}) = 2C(\mathbf{a}\mathbf{b}) + C(\mathbf{a}^*)$	35	24
<hr/> <hr/>		
Dual quaternions ( $\underline{\mathbf{a}}, \underline{\mathbf{b}} \in \mathcal{H}$ )		
$C(\lambda\underline{\mathbf{a}})$	8	0
$C(\underline{\mathbf{a}}^*)$	6	0
$C(\underline{\mathbf{a}} + \underline{\mathbf{b}})$	0	8
$C(\underline{\mathbf{a}}\underline{\mathbf{b}})$	48	40
$C(\text{Ad}(\underline{\mathbf{a}})\underline{\mathbf{b}}) = 2C(\underline{\mathbf{a}}\underline{\mathbf{b}}) + C(\underline{\mathbf{a}}^*)$	102	80
$C(\underline{\mathbf{a}} \times \underline{\mathbf{b}}) = 2C(\underline{\mathbf{a}}\underline{\mathbf{b}}) + C(\underline{\mathbf{a}} + \underline{\mathbf{b}}) + C(\lambda\underline{\mathbf{a}})$	104	88
<hr/> <hr/>		

The cost of calculating the twists throughout the kinematic chain is equivalent to calculating  $\underline{\xi}_{0,c_n}^{c_n}$ , the twist of the  $n$ -th and last center of mass, thanks to the recursive formulation. Therefore, from (4.15),

$$C(\underline{\xi}_{0,c_n}^{c_n}) = 2nC(\text{Ad}(\underline{\mathbf{a}})\underline{\mathbf{b}}) + nC(\underline{\mathbf{a}} + \underline{\mathbf{b}}). \quad (4.28)$$

From (4.16) and (4.17), the cost of calculating the time derivative of twists along the whole kinematic chain, similarly to the calculation of the twists, is given by

$$C(\dot{\underline{\xi}}_{0,c_n}^{c_n}) = n \left( 2C(\text{Ad}(\underline{\mathbf{a}})\underline{\mathbf{b}}) + C(\underline{\mathbf{a}} \times \underline{\mathbf{b}}) \right) + n \left( 2C(\underline{\mathbf{a}} + \underline{\mathbf{b}}) + C(\lambda\underline{\mathbf{a}}) \right), \quad (4.29)$$

considering the fact that  $\text{Ad}(\underline{\mathbf{x}}_{c_{i-1}}^{c_i}) \underline{\boldsymbol{\xi}}_{0,c_{i-1}}^{c_i-1}$  and  $\text{Ad}(\underline{\mathbf{x}}_{i-1}^{c_i}) \underline{\boldsymbol{\xi}}_{i-1,c_i}^{i-1}$  were already calculated for the twists.

Again, thanks to the recursive formulation, the cost of calculating the wrenches throughout the kinematic chain is equivalent to calculating  $\underline{\boldsymbol{\zeta}}_{0,1}^0$ , the wrench of the first joint. Therefore, from (4.26),

$$C(\underline{\boldsymbol{\zeta}}_{0,1}^0) = n \left( C(\underline{\boldsymbol{\zeta}}_{0,c_i}^{c_i}) + 2C(\text{Ad}(\underline{\mathbf{a}}) \underline{\mathbf{b}}) + C(\underline{\mathbf{a}} + \underline{\mathbf{b}}) \right), \quad (4.30)$$

where  $C(\underline{\boldsymbol{\zeta}}_{0,c_i}^{c_i})$  is given in table 4.3.

Table 4.3: Number of operations in the computation of  $\underline{\boldsymbol{\zeta}}_{0,c_i}^{c_i}$  (see line 7 in the dual quaternion Newton-Euler algorithm 4.4).

	Multiplications	Additions
$C(\underline{\mathbf{f}}_{0,c_i}^{c_i}) = C(\lambda \underline{\mathbf{a}}) + C(\underline{\mathbf{a}} \times \underline{\mathbf{b}}) + C(\underline{\mathbf{a}} + \underline{\mathbf{b}})$	40	32
$C(\underline{\boldsymbol{\tau}}_{0,c_i}^{c_i}) = 2C(\mathcal{L}_3(\mathbb{I}) \underline{\mathbf{a}}) + C(\underline{\mathbf{a}} \times \underline{\mathbf{b}}) + C(\underline{\mathbf{a}} + \underline{\mathbf{b}})$	54	44
$C(\underline{\boldsymbol{\zeta}}_{0,c_i}^{c_i}) = C(\underline{\mathbf{f}}_{0,c_i}^{c_i}) + C(\underline{\boldsymbol{\tau}}_{0,c_i}^{c_i})$	94	76
$C(\underline{\boldsymbol{\zeta}}_{0,c_i}^{c_i}) = C(\underline{\boldsymbol{\zeta}}_{0,c_i}^{c_i}) + C(\underline{\mathbf{a}} + \underline{\mathbf{b}}) + C(\lambda \underline{\mathbf{a}})$	98	80

Lastly, the total cost of the dual quaternion Newton-Euler algorithm also includes the cost  $C(\underline{\mathbf{x}}_n^0)$  of calculating the forward kinematics. Once more, since the forward kinematics is calculated iteratively throughout the kinematic chain (Adorno, 2011), all intermediate transformations are calculated at once. Therefore, the total cost of the dqNE algorithm is given by

$$C_{\text{total}}(\text{dqNE}) = C(\underline{\mathbf{x}}_n^0) + C(\underline{\boldsymbol{\xi}}_{0,c_n}^{c_n}) + C(\underline{\dot{\boldsymbol{\xi}}}_{0,c_n}^{c_n}) + C(\underline{\boldsymbol{\zeta}}_{0,1}^0), \quad (4.31)$$

where the intermediate costs are summarized in table 4.4 with their explicit values in terms of additions and multiplications of real numbers.

Table 4.4: Number of operations in different parts of the dual quaternion Newton-Euler algorithm.

	Multiplications	Additions
$C(\underline{\mathbf{x}}_n^0)$ (Adorno, 2011)	$60n - 48$	$44n - 40$
$C(\underline{\boldsymbol{\xi}}_{0,c_n}^{c_n})$ (equation (4.28))	$204n$	$168n$
$C(\underline{\dot{\boldsymbol{\xi}}}_{0,c_n}^{c_n})$ (equation (4.29))	$316n$	$264n$
$C(\underline{\boldsymbol{\zeta}}_{0,1}^0)$ (equation (4.30))	$302n$	$248n$

The cost comparison between the dual quaternion Newton-Euler formalism for obtaining the dynamical model of an  $n$ -DoF serial robot and its classic counterpart, in terms of number of multiplications and additions of real numbers, is summarized in table 4.5.

For the classic Newton-Euler algorithm, it was considered the version based on three-dimensional vectors proposed by Luh et al. (1980), whose mathematical cost was calculated by Balafoutis (1994), and is, to the best of my knowledge, one of the most efficient implementations in the literature.

Table 4.5: Cost comparison between the proposed method and its classic counterpart for obtaining the dynamical model for an  $n$ -DoF serial robot.

Method	Multiplications	Additions
Dual quaternion Newton-Euler algorithm (cost for arbitrary joints)	$882n - 48$	$724n - 40$
Classic Newton-Euler algorithm (cost for revolute and prismatic joints) (Balafoutis, 1994)	$150n - 48$	$131n - 48$

The algorithm presented by Luh et al. (1980) costs less than the dual quaternion Newton-Euler formalism. The cost presented for the proposed method is, however, fairly conservative and is given as an upper bound. For instance, the calculations could be further optimized by exploring the fact that several operations involve pure dual quaternions, which have six elements instead of eight.<sup>6</sup> Additionally, the cost presented by Balafoutis (1994) does not include the costs of obtaining the robot kinematic model (which would be equivalent to the cost  $C(\underline{\mathbf{x}}_n^0)$  in (4.31)). Also, the proposed method works for *any* type of joint and the calculations were not optimized for any particular type of joint, differently from Luh et al. (1980) who only consider prismatic and revolute joints, which are exploited to optimize the computational cost. In contrast, the costs (4.28) and (4.29) take into account an *arbitrary* twist  $\underline{\xi}_{i-1,c_i}^{i-1}$  and corresponding twist derivative  $\dot{\underline{\xi}}_{i-1,c_i}^{i-1}$ . Nonetheless, both the proposed algorithm and the one of Luh et al. (1980) have linear costs in the number of degrees of freedom, with coefficients of the same order of magnitude.

#### 4.2.1.5 Simulations and discussions

This section presents two simulations of robot manipulators with revolute joints to illustrate the application of the proposed dual quaternion Newton-Euler (dqNE) formalism. The first one uses the dqNE to obtain the joint torques of a 6-DoF robotic manipulator, whereas the second one utilizes the dqNE to find the joint accelerations of a 50-DoF serial manipulator. Both simulations compare the proposed formalism with state-of-the-art libraries and are performed in the robot simulator V-REP PRO EDU V3.6.2 (Rohmer et al., 2013) with the Vortex Studio Academic<sup>7</sup> physics engine using an interface with MATLAB 2020a and the computational library DQ Robotics (Adorno & Marques Marinho, 2021) for dual quaternion algebra on a computer running Ubuntu 18.04 LTS 64 bits equipped with an

<sup>6</sup>For the cost of the cross product of dual quaternions, one of the most expensive operation presented in table 4.2, the reduction would be from  $C(\underline{\mathbf{a}} \times \underline{\mathbf{b}}) = \{104, 88\}$  to  $C(\underline{\mathbf{a}} \times \underline{\mathbf{b}}) = \{60, 48\}$ .

<sup>7</sup>Available at: <https://www.cm-labs.com/vortex-studio/>

Intel Core i7 6500U with 8GB RAM.

**Obtaining joint torques** This simulation demonstrates the application of the dqNE to obtain the joint torques of the 6-DoF JACO robotic arm shown in figure 4.3. The robot follows sinusoidal joint velocity trajectories  $\dot{\mathbf{q}}_d \in \mathbb{R}^6$  using V-REP’s standard low-level controllers and its joint configurations  $\mathbf{q} \in \mathbb{R}^6$ , joint velocities  $\dot{\mathbf{q}} \in \mathbb{R}^6$ , and joint torques  $\boldsymbol{\tau} \in \mathbb{R}^6$  are stored. Since the simulator does not allow the direct reading of accelerations,  $\dot{\mathbf{q}}$  is used to obtain the joint accelerations  $\ddot{\mathbf{q}} \in \mathbb{R}^6$  by means of numerical differentiation based on Richardson extrapolation (Gilat & Subramaniam, 2014, p. 322).<sup>8</sup>



Figure 4.3: The 6-DoF JACO robotic arm in the robot simulator V-REP.

Using the values of  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$ , the joint torques  $\boldsymbol{\tau}_{\text{dqNE}} \in \mathbb{R}^6$  are obtained by projecting the joint wrenches yielded by the dqNE onto the joints motion axes through (4.27). Then, both  $\boldsymbol{\tau}_{\text{dqNE}}$  and  $\boldsymbol{\tau}$  are filtered with a 6nd-order low-pass digital Butterworth filter with normalized cutoff frequency of 10Hz<sup>9</sup> and the comparison between those two joint torque waveforms is made considering the coefficient of multiple correlation (CMC) (Ferrari et al., 2010) between them. The CMC measures the similarity of waveforms, taking into account, among other things, effects of different offsets. Furthermore, the CMC formulation proposed by Ferrari et al. (2010) was designed to access the similarity of waveforms acquired synchronously through different media (e.g., different dynamic modeling strategies) when one is interested in the effect of the media on the waveform. The CMC provides a coefficient ranging between zero and one that indicates how similar two given waveforms are. Identical waveforms have CMC equal to one, whereas completely different waveforms have CMC equal to zero.<sup>10</sup>

Furthermore, the joint torques  $\boldsymbol{\tau}_{\text{rtNE}} \in \mathbb{R}^6$  are obtained from the Robotics Toolbox’s Newton-Euler (rtNE) classic recursive algorithm (Corke, 2017), whereas  $\boldsymbol{\tau}_{\text{sv2NE}} \in \mathbb{R}^6$  are

<sup>8</sup>I would like to thank my colleague Juan José Quiroz Omaña for providing me with the MATLAB implementation of the numerical differentiation class used in this thesis.

<sup>9</sup>A discussion about the filter and its necessity is presented in appendix D.

<sup>10</sup>I would like to thank my colleague Ana Christine de Oliveira for providing me with the MATLAB implementation of the CMC used in this thesis.

obtained from its spatial vector counterpart implemented on Roy Featherstone’s Spatial V2 (sv2NE) library.<sup>11</sup> Accordingly, those joint torques are filtered using the same discrete low-pass Butterworth filter applied to  $\tau_{\text{dqNE}}$  and  $\tau$ . Finally, the CMCs between the joint torque waveforms  $\tau_{\text{rtNE}}$  and  $\tau$  and between  $\tau_{\text{sv2NE}}$  and  $\tau$  are calculated. The Robotics Toolbox and the Spatial V2 are widely used libraries whose accuracy has been verified throughout the years; thus, they are an appropriate baseline for the evaluation of the CMCs obtained by using the proposed dqNE on robotic manipulators. As no widespread library for dynamic modeling using the Lie algebra  $se(3)$  is available, this representation is not considered in the comparison.<sup>12</sup>

The joint torque waveforms from the different dynamic model strategies ( $\tau_{\text{dqNE}}$ ,  $\tau_{\text{rtNE}}$ , and  $\tau_{\text{sv2NE}}$ ) and the values from V-REP ( $\tau$ ) are shown in figure 4.4, whereas their relative CMCs are presented in table 4.6. All models present mean (mean) and minimum (min) CMC close to one, with small standard deviation (std) and high maximum (max) CMC; thus, indicating high similarity between their joint torque waveforms and the values from V-REP. Moreover, the dqNE is equivalent to both the rtNE and the sv2NE, which demonstrates that the proposed strategy is as accurate as state-of-the-art libraries. The small discrepancies between the joint torques calculated by the models and the ones read from V-REP arise from both discretization effects, such as the approximation of the joint accelerations  $\ddot{\mathbf{q}}$ , and uncertainties in the kinematic/dynamic parameters. Adaptive control strategies could be used to mitigate this problem, and have already been explored in classic recursive Newton-Euler formulations in the literature (Walker, 1988; Walker et al., 1989; Jone Hann Jean & Li-Chen Fu, 1993), but are outside the scope of this thesis.

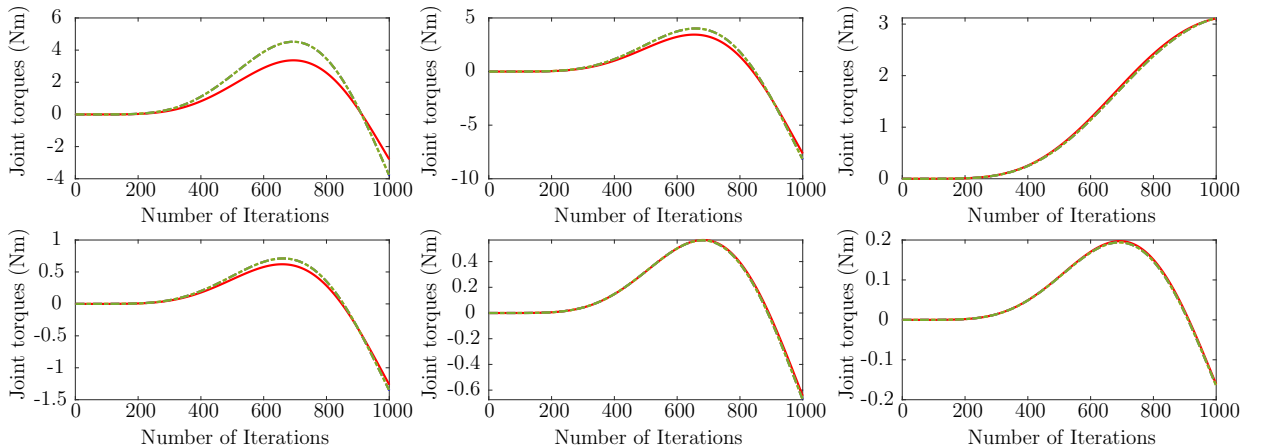


Figure 4.4: Joint torque waveforms of the 6-DoF JACO robotic arm. Solid curves correspond to the joint torques  $\tau$ , whereas dashed curves correspond to the joint torques  $\tau_{\text{dqNE}}$  and dotted curves correspond to the joint torques  $\tau_{\text{rtNE}}$  and  $\tau_{\text{sv2NE}}$ . Since all the model strategies (dqNE, rtNE, and sv2NE) obtained almost identical results, their joint torque waveforms overlap in the graph.

<sup>11</sup>Available at: <http://royfeatherstone.org/spatial/v2/>

<sup>12</sup>The (Mathematica) packages “Screws.m” and “RobotLinks.m,” once maintained by Murray and Sur (Murray et al., 1994, p. 435), are no longer available at [avalon.caltech.edu](http://avalon.caltech.edu).

Table 4.6: CMC between the joint torque waveforms obtained through the different dynamic model strategies ( $\boldsymbol{\tau}_{\text{dqNE}}$ ,  $\boldsymbol{\tau}_{\text{rtNE}}$ , and  $\boldsymbol{\tau}_{\text{sv2NE}}$ ) and the values obtained from V-REP ( $\boldsymbol{\tau}$ ) for the 6-DoF JACO robotic arm. The closer to one, the more similar the waveforms are.

Method	CMC for the 6-DoF JACO robotic arm			
	min	mean	std	max
dqNE	0.9667	0.9930	0.0130	0.9999
rtNE	0.9666	0.9930	0.0131	0.9999
sv2NE	0.9666	0.9930	0.0131	0.9999

Although not apparent from an illustrative example with a robotic manipulator, the advantage of using the proposed dual quaternion Newton-Euler formalism is the unification of the joint variables into twists and wrenches whilst maintaining a formal algebraic description of the robot dynamics. This will become more evident in the sections dedicated to the modular composition strategy, an idea introduced in chapter 5.

**Obtaining joint accelerations** This simulation demonstrates the application of the dqNE to obtain the joint accelerations of a 50-DoF serial manipulator shown in figure 4.5.<sup>13</sup> The setup is similar to one adopted in the previous example; that is, the robot uses the V-REP’s standard low-level controllers to follow the joint velocity trajectories  $\dot{\boldsymbol{q}}_d \in \mathbb{R}^{50}$ , have its joint configurations  $\boldsymbol{q} \in \mathbb{R}^{50}$  and velocities  $\dot{\boldsymbol{q}} \in \mathbb{R}^{50}$  stored, and its joint accelerations  $\ddot{\boldsymbol{q}} \in \mathbb{R}^{50}$  numerically estimated from  $\dot{\boldsymbol{q}}$ .

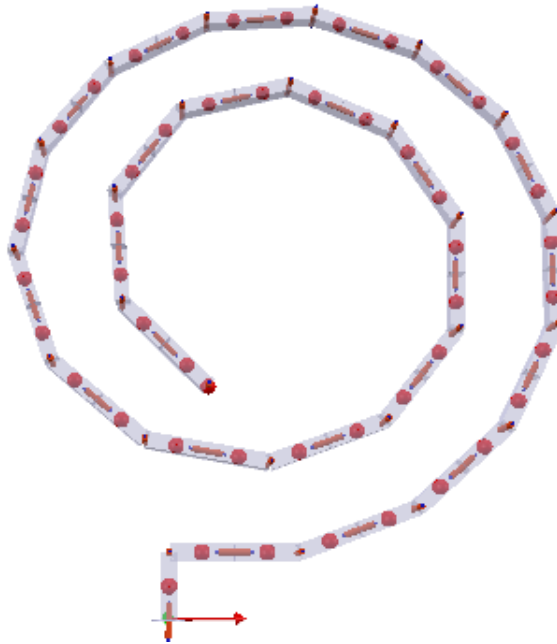


Figure 4.5: A 50-DoF serial manipulator in the robot simulator V-REP.

<sup>13</sup>I would like to thank my colleague Juan José Quiroz Omaña for providing me with the V-REP implementation of the 50-DoF serial manipulator used in this thesis.

Applying the strategy presented in section 3.2.1, the Euler-Lagrange model of the robot was obtained from the dqNE and the rtNE and then used to calculate the joint accelerations  $\ddot{\mathbf{q}}_{\text{dqNE}} \in \mathbb{R}^{50}$  and  $\ddot{\mathbf{q}}_{\text{rtNE}} \in \mathbb{R}^{50}$ , respectively, through

$$\ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})^{-1} (\boldsymbol{\tau} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} - \mathbf{g}(\mathbf{q})),$$

in which the joint wrenches obtained from the dqNE must be projected onto the joints motion axes through (4.27) to yield the vectors  $\mathbf{g}(\mathbf{q})$ ,  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}$  and the columns of  $\mathbf{M}(\mathbf{q})$  given by (3.2). For instance,

$$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} \langle \underline{\boldsymbol{\zeta}}_{0,1}^0, \mathbf{l}_1^0 \rangle \\ \vdots \\ \langle \underline{\boldsymbol{\zeta}}_{0,50}^{49}, \mathbf{l}_{50}^{49} \rangle \end{bmatrix} \in \mathbb{R}^{50},$$

where  $\mathcal{H}_p^{50} \ni \left[ \underline{\boldsymbol{\zeta}}_{0,1}^0 \quad \dots \quad \underline{\boldsymbol{\zeta}}_{0,50}^{49} \right]^T = \text{newton\_euler}(\mathbf{q}, \mathbf{0}, \mathbf{0})$  is the stacked vector of joint wrenches and  $\left[ \mathbf{l}_1^0 \quad \dots \quad \mathbf{l}_{50}^{49} \right]^T \in (\mathbb{H}_p \cap \mathbb{S}^3)^{50}$  is the stacked vector of all the motion axes of the robot's joints. Finally, the joint accelerations  $\ddot{\mathbf{q}}$ ,  $\ddot{\mathbf{q}}_{\text{dqNE}}$ , and are  $\ddot{\mathbf{q}}_{\text{rtNE}}$  filtered, using a 6nd-order low-pass digital Butterworth filter with normalized cutoff frequency of 100Hz, and the CMCs between each model's joint acceleration waveforms and  $\ddot{\mathbf{q}}$  are calculated. The sv2NE is arbitrarily not used in the comparison since its almost identical accuracy with the rtNE was established on the previous simulation; thus, its inclusion was deemed unnecessary.

Table 4.7 presents the CMC between the joint acceleration waveforms obtained through the different dynamic model strategies ( $\ddot{\mathbf{q}}_{\text{dqNE}}$  and  $\ddot{\mathbf{q}}_{\text{rtNE}}$ ) and the values obtained from V-REP ( $\ddot{\mathbf{q}}$ ) for the 50-DoF robot manipulator. For qualitative analysis, figure 4.6 presents the joint accelerations which correspond to the minimum and maximum CMCs found during simulations. Once again, the results indicate that the proposed dual quaternion Newton-Euler formulation yields results that are as accurate as state-of-the-art libraries.

Table 4.7: CMC between the joint acceleration waveforms obtained through the different dynamic model strategies ( $\ddot{\mathbf{q}}_{\text{dqNE}}$  and  $\ddot{\mathbf{q}}_{\text{rtNE}}$ ) and the values obtained from V-REP ( $\ddot{\mathbf{q}}$ ) for the 50-DoF serial manipulator. The closer to one, the more similar the waveforms are (adapted from Silva et al., 2022, table 2).

Method	CMC for the 50-DoF serial manipulator			
	min	mean	std	max
dqNE	0.9044	0.9893	0.0182	0.9993
rtNE	0.9044	0.9893	0.0182	0.9993

As a final remark, although applicable to the problem of forward dynamics through (3.2), obtaining the Euler-Lagrange model through the Newton-Euler formalism requires several executions of the algorithm. One execution to obtain the gravitational vector, one

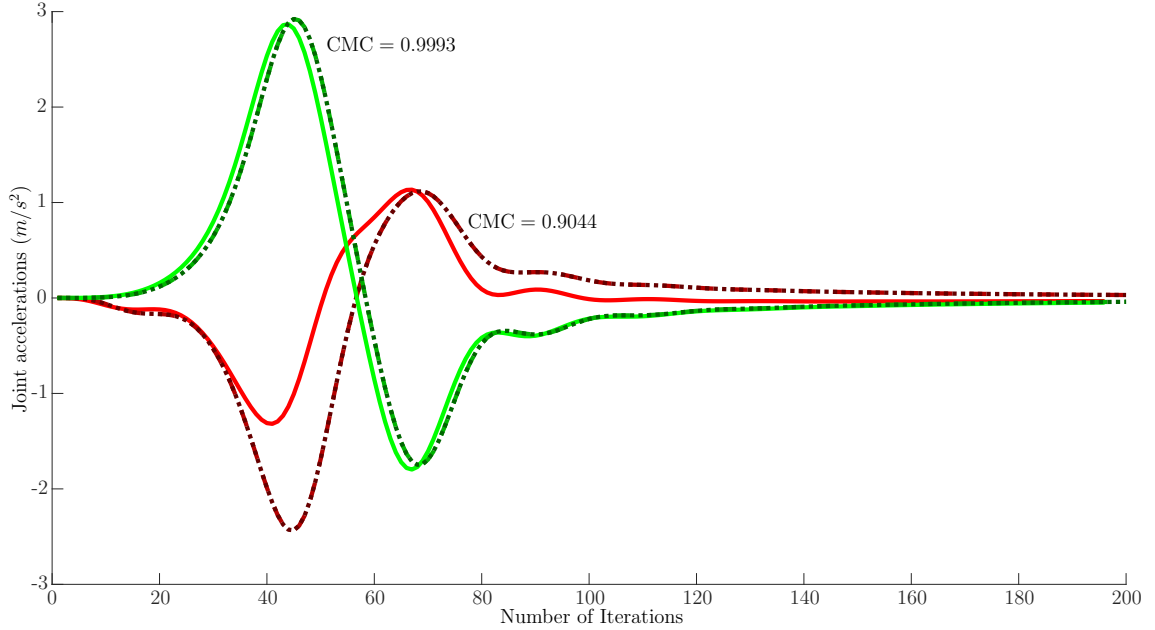


Figure 4.6: Joint acceleration waveforms of the 50-DoF robotic manipulator. Solid curves correspond to the V-REP values, whereas dashed curves correspond to the values obtained using the dqNE and dotted curves correspond to the values obtained using the rtNE for the joint acceleration waveforms of the third (CMC = 0.9044) and twenty-seventh (CMC = 0.9993) joints, respectively. Since both the model strategies (dqNE and rtNE) obtained almost identical results, their joint torque waveforms overlap in the graph.

to obtain the vector of Coriolis and centrifugal terms and one for each column of the inertia matrix. Considering the 50-DoF manipulator robot discussed in this example, this results in 52 executions of each algorithm (dqNE and rtNE) for each simulation step. Therefore, recursive Newton-Euler formulations are better suited for inverse dynamics applications where one is interested in finding the joint torques; thus, requiring only one execution of either the dqNE or the rtNE.

## 4.2.2 Adding more rigid bodies to the kinematic chain

This section introduces the idea of considering the dual quaternion Newton-Euler formalism not as a recursive algorithm but as a—mathematical—function of the joint twists. Consider the following definition.

**Definition 4.3.** Consider the  $i$ -th rigid body of a serial kinematic chain composed of  $n$  rigid bodies. Given  $\bar{\Xi}_{1,i}$ , the stacked vector of twists imposed by the movements of joints 1 to  $i$ , and  $\bar{\Xi}_{i,n}$ , the stacked vector of twists at CoMs  $i$  to  $n$ , in which  $\bar{\Xi}_{a,b}, \bar{\Xi}_{a,b} \in \mathcal{H}_p^{b-a+1}$ , the twist at the  $i$ -th CoM and the wrench at the  $i$ -th joint of the robot are respectively



given by the functions  $\underline{\mathcal{F}} : \mathcal{H}_p^i \rightarrow \mathcal{H}_p$  and  $\underline{\mathcal{B}} : \mathcal{H}_p^{n-i+1} \rightarrow \mathcal{H}_p$ . That is,

$$\begin{aligned}\underline{\xi}_{0,c_i}^{c_i} &= \underline{\mathcal{F}}_i \left( \bar{\underline{\Xi}}_{1,i} \right), \\ \underline{\zeta}_{0,i}^{i-1} &= \underline{\mathcal{B}}_i \left( \check{\underline{\Xi}}_{i,n} \right),\end{aligned}\quad (4.32)$$

where  $\bar{\underline{\Xi}}_{1,i} = \left[ \underline{\xi}_{0,c_1}^0 \quad \underline{\xi}_{1,c_2}^1 \quad \cdots \quad \underline{\xi}_{i-1,c_i}^{i-1} \right]^T$  and  $\check{\underline{\Xi}}_{i,n} = \left[ \underline{\xi}_{0,c_i}^{c_i} \quad \underline{\xi}_{0,c_{i+1}}^{c_{i+1}} \quad \cdots \quad \underline{\xi}_{0,c_n}^{c_n} \right]^T$ .

From the recurrence equations of the twists, given by (4.15), it is straightforward to notice that the total twist at the  $i$ -th CoM of the robot is a function of the joint motions from the first up to the  $i$ -th joint. As for the joint wrench, from the recurrence equation given by (4.26) one can notice that the  $i$ -th joint wrench is a function of the wrenches from the last joint up to the  $i$ -th joint. Moreover, although Newton's second law and the Euler's rotation equation demand the expression of  $\dot{\underline{\xi}}_{0,c_i}^{c_i}$  (see equations (4.22) and (4.23)), this is the analytical time derivative of  $\underline{\xi}_{0,c_i}^{c_i}$ , given by (3.17). Therefore,  $\dot{\underline{\xi}}_{0,c_i}^{c_i}$  can be explicitly calculated in the function  $\underline{\mathcal{B}}$  using the twist  $\underline{\xi}_{0,c_i}^{c_i}$ .

Considering (4.32), (4.15), and (4.26) the recurrence equations of the dual quaternion Newton-Euler formalism, for the  $i$ -th rigid body in a kinematic chain, can be rewritten as

$$\underline{\xi}_{0,c_i}^{c_i} = \underline{\mathcal{F}}_i \left( \bar{\underline{\Xi}}_{1,i} \right) = \text{Ad} \left( \mathbf{x}_{c_{i-1}}^{c_i} \right) \underline{\mathcal{F}}_{i-1} \left( \bar{\underline{\Xi}}_{1,i-1} \right) + \text{Ad} \left( \mathbf{x}_{i-1}^{c_i} \right) \underline{\xi}_{i-1,c_i}^{i-1}, \quad (4.33)$$

where  $\underline{\mathcal{F}}_0 \left( \bar{\underline{\Xi}}_{1,0} \right) \triangleq 0$  and

$$\underline{\zeta}_{0,i}^{i-1} = \underline{\mathcal{B}}_i \left( \check{\underline{\Xi}}_{i,n} \right) = \underline{\zeta}_{0,c_i}^{i-1} + \text{Ad} \left( \mathbf{x}_i^{i-1} \right) \underline{\mathcal{B}}_{i+1} \left( \check{\underline{\Xi}}_{i+1,n} \right), \quad (4.34)$$

where  $\underline{\mathcal{B}}_{n+1} \left( \check{\underline{\Xi}}_{n+1,n} \right) \triangleq 0$ .

The next examples demonstrate how to apply equations (4.33) and (4.34) to obtain the model of the simple two-link robot shown in figure 4.7 and how to consider the inclusion of a third link at the end of the kinematic chain.

**Example 4.3.** Consider the simple two-link robot presented in figure 4.7. Taking into account the functions  $\underline{\mathcal{F}}$  and  $\underline{\mathcal{B}}$  given by definition 4.3, the dynamic model of this robot is given by

$$\begin{aligned}\underline{\zeta}_{0,2}^1 &= \underline{\mathcal{B}}_2 \left( \check{\underline{\Xi}}_{2,2} \right) = \text{Ad} \left( \mathbf{x}_{c_2}^1 \right) \underline{\zeta}_{0,c_2}^{c_2}, \\ \underline{\zeta}_{0,1}^0 &= \underline{\mathcal{B}}_1 \left( \check{\underline{\Xi}}_{1,2} \right) = \underline{\zeta}_{0,c_1}^0 + \text{Ad} \left( \mathbf{x}_1^0 \right) \underline{\mathcal{B}}_2 \left( \check{\underline{\Xi}}_{2,2} \right) = \underline{\zeta}_{0,c_1}^0 + \underline{\zeta}_{0,2}^0,\end{aligned}$$

where  $\check{\underline{\Xi}}_{2,2} = \underline{\xi}_{0,c_2}^{c_2}$ ,  $\check{\underline{\Xi}}_{1,2} = \left[ \underline{\xi}_{0,c_1}^{c_1} \quad \underline{\xi}_{0,c_2}^{c_2} \right]^T$ , and  $\underline{\zeta}_{0,c_1}^0 = \text{Ad} \left( \mathbf{x}_{c_1}^0 \right) \underline{\zeta}_{0,c_1}^{c_1}$ . Moreover,

$$\begin{aligned}\underline{\xi}_{0,c_1}^{c_1} &= \underline{\mathcal{F}}_1 \left( \bar{\underline{\Xi}}_{1,1} \right) = \text{Ad} \left( \mathbf{x}_0^{c_1} \right) \underline{\xi}_{0,c_1}^0, \\ \underline{\xi}_{0,c_2}^{c_2} &= \underline{\mathcal{F}}_2 \left( \bar{\underline{\Xi}}_{1,2} \right) = \text{Ad} \left( \mathbf{x}_{c_1}^{c_2} \right) \underline{\mathcal{F}}_1 \left( \bar{\underline{\Xi}}_{1,1} \right) + \underline{\xi}_{1,c_2}^{c_2} = \underline{\xi}_{0,c_1}^{c_2} + \underline{\xi}_{1,c_2}^{c_2},\end{aligned}\quad (4.35)$$

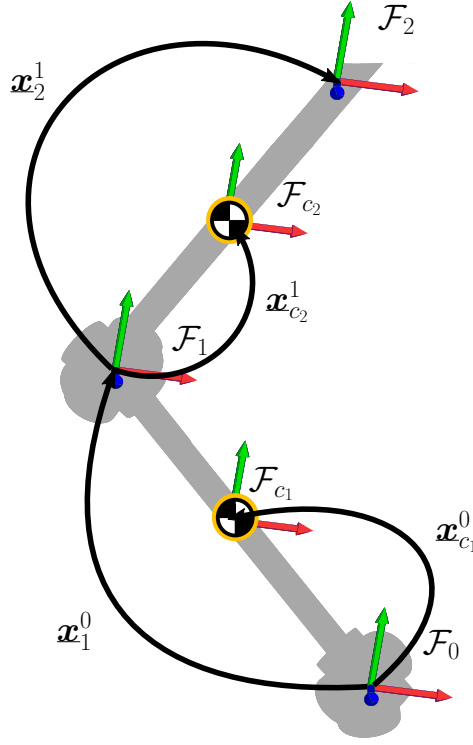


Figure 4.7: A 2-DoF serial manipulator.

where  $\bar{\Xi}_{1,1} = \underline{\xi}_{0,c_1}^0$ ,  $\bar{\Xi}_{1,2} = \begin{bmatrix} \underline{\xi}_{0,c_1}^0 & \underline{\xi}_{1,c_2}^1 \end{bmatrix}^T$ , in which the twist  $\underline{\xi}_{i-1,c_i}^{i-1}$  is generated by the  $i$ -th joint (see table 4.1), and  $\underline{\xi}_{1,c_2}^{c_2} = \text{Ad}(\mathbf{x}_1^{c_2}) \underline{\xi}_{1,c_2}^1$ .

**Example 4.4.** Consider the addition of a third link at the end of the simple two-link robot presented in figure 4.7, thus, yielding the three-link serial manipulator shown in figure 4.8. The dynamic model of this newly-formed robot is given by

$$\begin{aligned} \underline{\zeta}_{0,3}^2 &= \mathcal{B}_3(\bar{\Xi}_{3,3}) = \text{Ad}(\mathbf{x}_{c_3}^2) \underline{\zeta}_{0,c_3}^{c_3}, \\ \underline{\zeta}_{0,2}^1 &= \mathcal{B}_2(\bar{\Xi}_{2,3}) = \underline{\zeta}_{0,c_2}^1 + \text{Ad}(\mathbf{x}_2^1) \mathcal{B}_3(\bar{\Xi}_{3,3}) = \underline{\zeta}_{0,c_2}^1 + \underline{\zeta}_{0,3}^2, \\ \underline{\zeta}_{0,1}^0 &= \mathcal{B}_1(\bar{\Xi}_{1,3}) = \underline{\zeta}_{0,c_1}^0 + \text{Ad}(\mathbf{x}_1^0) \mathcal{B}_2(\bar{\Xi}_{2,3}) = \underline{\zeta}_{0,c_1}^0 + \underline{\zeta}_{0,2}^1, \end{aligned}$$

where  $\bar{\Xi}_{3,3} = \underline{\xi}_{0,c_3}^{c_3}$ ,  $\bar{\Xi}_{2,3} = \begin{bmatrix} \underline{\xi}_{0,c_2}^{c_2} & \underline{\xi}_{0,c_3}^{c_3} \end{bmatrix}^T$ ,  $\bar{\Xi}_{1,3} = \begin{bmatrix} \underline{\xi}_{0,c_1}^{c_1} & \underline{\xi}_{0,c_2}^{c_2} & \underline{\xi}_{0,c_3}^{c_3} \end{bmatrix}^T$ ,  $\underline{\zeta}_{0,c_2}^1 = \text{Ad}(\mathbf{x}_{c_2}^2) \underline{\zeta}_{0,c_2}^{c_2}$ , and  $\underline{\zeta}_{0,c_1}^0 = \text{Ad}(\mathbf{x}_{c_1}^0) \underline{\zeta}_{0,c_1}^{c_1}$ . Moreover,

$$\begin{aligned} \underline{\xi}_{0,c_1}^{c_1} &= \mathcal{F}_1(\bar{\Xi}_{1,1}) = \text{Ad}(\mathbf{x}_0^{c_1}) \underline{\xi}_{0,c_1}^0, \\ \underline{\xi}_{0,c_2}^{c_2} &= \mathcal{F}_2(\bar{\Xi}_{1,2}) = \text{Ad}(\mathbf{x}_{c_1}^{c_2}) \mathcal{F}_1(\bar{\Xi}_{1,1}) + \underline{\xi}_{1,c_2}^{c_2} = \underline{\xi}_{0,c_1}^{c_2} + \underline{\xi}_{1,c_2}^{c_2}, \\ \underline{\xi}_{0,c_3}^{c_3} &= \mathcal{F}_3(\bar{\Xi}_{1,3}) = \text{Ad}(\mathbf{x}_{c_2}^{c_3}) \mathcal{F}_2(\bar{\Xi}_{1,2}) + \underline{\xi}_{2,c_3}^{c_3} = \underline{\xi}_{0,c_2}^{c_3} + \underline{\xi}_{2,c_3}^{c_3}, \end{aligned}$$

where  $\bar{\Xi}_{1,1} = \underline{\xi}_{0,c_1}^0$ ,  $\bar{\Xi}_{1,2} = \begin{bmatrix} \underline{\xi}_{0,c_1}^0 & \underline{\xi}_{1,c_2}^1 \end{bmatrix}^T$ ,  $\bar{\Xi}_{1,3} = \begin{bmatrix} \underline{\xi}_{0,c_1}^0 & \underline{\xi}_{1,c_2}^1 & \underline{\xi}_{2,c_3}^2 \end{bmatrix}^T$ , in which the twist  $\underline{\xi}_{i-1,c_i}^{i-1}$  is generated by the  $i$ -th joint (see table 4.1),  $\underline{\xi}_{1,c_2}^{c_2} = \text{Ad}(\mathbf{x}_1^{c_2}) \underline{\xi}_{1,c_2}^1$ , and

$$\underline{\xi}_{2,c_3}^{c_3} = \text{Ad}(\underline{x}_2^{c_3}) \underline{\xi}_{2,c_3}^2.$$

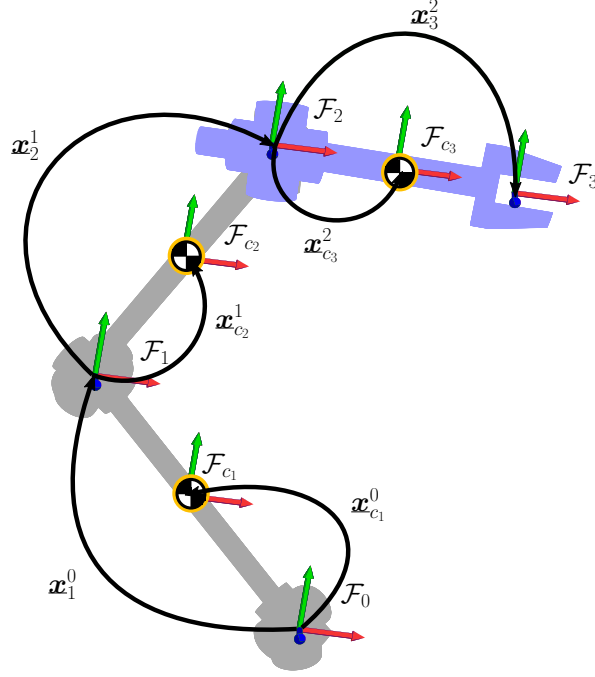


Figure 4.8: A 3-DoF serial manipulator composed by the addition of a third link at the end of the simple two-link robot presented in figure 4.7.

If the usefulness of functions  $\underline{\mathcal{F}}$  and  $\underline{\mathcal{B}}$  is not yet clear, the next sections demonstrate how they simplify the explanation of the dynamic modeling of mobile manipulators and branched robots. Nonetheless, the major advantage of seeing the dual quaternion Newton-Euler formulation as a function (of the twists) is shown on chapter 5, which deals with the composition of dynamic models of complete kinematic chains.

### 4.2.3 Mobile robot manipulators

Consider a fixed-base robotic manipulator composed of  $n$  links. When this serial kinematic chain is attached to a mobile base, the dynamic model of the complete mobile manipulator is given by (4.33) and (4.34), as demonstrated on the previous section. One needs only to find, then, the expression  $\underline{\mathcal{F}}_1(\underline{\Xi}_{1,1})$  of the mobile base twist. This process is illustrated in figure 4.9, in which the CoMs of the mobile base and of the first link of the robotic manipulator are, respectively, represented by the frames  $\mathcal{F}_{c_1}$  and  $\mathcal{F}_{c_2}$ .

The next section demonstrate how to obtain the analytical expressions of the twists for holonomic mobile bases. Notice, however, that this is presented as an example and the same procedure could be used to find the twist of arbitrary mobile bases. For instance, a UAV mobile base could be considered on the model as the addition of a 6-DoF joint at the beginning of the serial kinematic chain, and its twist is already given in table 4.1.

### 4.2.3.1 Holonomic mobile manipulators

Consider a holonomic mobile base, such as the one of the holonomic mobile manipulator presented in figure 4.9, whose coordinates are given by  $(x, y, \phi)$ , where  $\phi \in [0, 2\pi)$  is its angle of rotation.

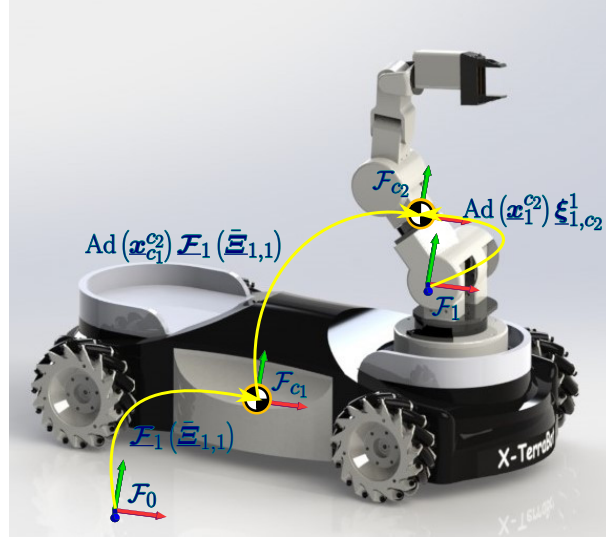


Figure 4.9: X-Terrabot, a holonomic mobile manipulator (courtesy of ASIMoV Robotics). The mobile base twist  $\mathcal{F}_1(\underline{\Xi}_{1,1})$  is propagated throughout the robotic manipulator through (4.33).

The twist of this holonomic mobile base is given by

$$\begin{aligned}\underline{\xi}_h &= \omega_h + \varepsilon \mathbf{v}_h \\ &= \dot{\phi} \hat{k} + \varepsilon (\dot{x} \hat{i} + \dot{y} \hat{j}),\end{aligned}\quad (4.36)$$

whose time derivative yields

$$\dot{\underline{\xi}}_h = \ddot{\phi} \hat{k} + \varepsilon (\ddot{x} \hat{i} + \ddot{y} \hat{j}).$$

The dynamical model of the holonomic mobile manipulator is then given by (4.33) and (4.34), in which  $\mathcal{F}_1(\underline{\Xi}_{1,1})$  is given by (4.36). Additionally, notice how the twist  $\underline{\xi}_h$  has the same expression of the twist of a planar joint (*see* table 4.1) since that type of joint is kinematically equivalent to a holonomic mobile base.

### 4.2.3.2 Simulations and discussions

Using the simulation setup described in section 4.2.1.5, the dqNE is applied to obtain the generalized accelerations of the 9-DoF holonomic mobile manipulator shown in figure 4.10, a robot composed of a JACO robotic arm attached to a KUKA's youBot holonomic base. The robot follows the generalized velocities trajectories  $\dot{\mathbf{q}}_d \in \mathbb{R}^9$ , have its generalized

configurations  $\mathbf{q} \in \mathbb{R}^9$  and velocities  $\dot{\mathbf{q}} \in \mathbb{R}^9$  stored, and its generalized accelerations  $\ddot{\mathbf{q}} \in \mathbb{R}^9$  numerically estimated from  $\dot{\mathbf{q}}$ . The analysis is made in terms of generalized accelerations because V-REP does not allow the direct reading of the mobile base's generalized forces.

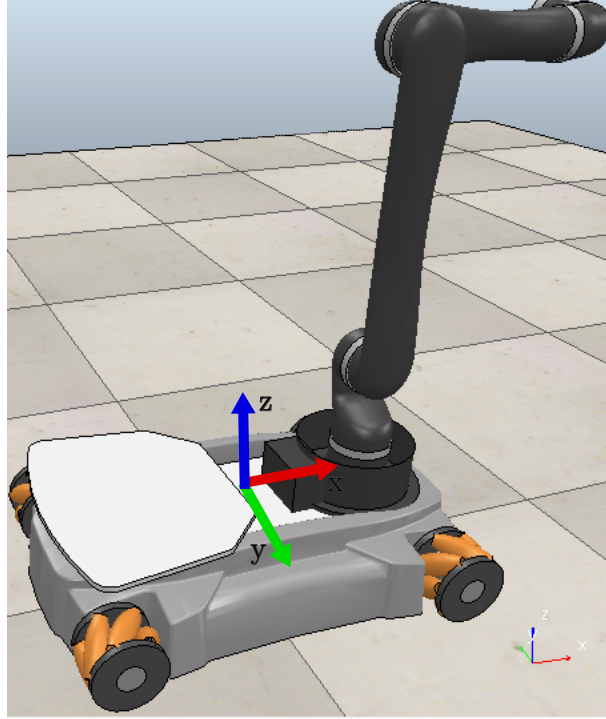


Figure 4.10: 9-DoF holonomic mobile manipulator composed of a JACO robotic arm attached to a KUKA's youBot holonomic base in the robot simulator V-REP.

The joint accelerations  $\ddot{\mathbf{q}}_{\text{dqNE}} \in \mathbb{R}^9$  are obtained from the dqNE applying the same strategy presented in the simulation of the 50-DoF robotic manipulator (see section 4.2.1.5), that is

$$\ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})^{-1} (\boldsymbol{\tau} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} - \mathbf{g}(\mathbf{q})).$$

For the holonomic mobile base, the projections onto the motion axes through (4.27) are made considering

$$\begin{bmatrix} f_x \\ f_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} \mathcal{P}(\langle \underline{\boldsymbol{\zeta}}_{0,c_1}^{c_1}, \text{Ad}(\mathbf{r}_0^{c_1}) \hat{i} \rangle) \\ \mathcal{P}(\langle \underline{\boldsymbol{\zeta}}_{0,c_1}^{c_1}, \text{Ad}(\mathbf{r}_0^{c_1}) \hat{j} \rangle) \\ \mathcal{D}(\langle \underline{\boldsymbol{\zeta}}_{0,c_1}^{c_1}, \text{Ad}(\mathbf{r}_0^{c_1}) \hat{k} \rangle) \end{bmatrix}, \quad (4.37)$$

where  $\underline{\boldsymbol{\zeta}}_{0,c_1}^{c_1}$  is the wrench at the CoM of the holonomic mobile base. Then, both  $\ddot{\mathbf{q}}_{\text{dqNE}}$  and  $\ddot{\mathbf{q}}$  are filtered, using a 6nd-order low-pass digital Butterworth filter with normalized cutoff frequency of 100Hz, and the CMCs between them are calculated and presented in table 4.8. The results indicate that the proposed dual quaternion Newton-Euler formulation yields accurate results, and, for qualitative analysis, figure 4.11 presents the joint accelerations which correspond to the minimum and maximum CMCs found during simulations. Since the almost identical accuracy between the dqNE, the rtNE, and the sv2NE was established on the previous simulations, the inclusion of those extra libraries was deemed unnecessary.

Table 4.8: CMC between the joint acceleration waveforms obtained through the dqNE ( $\ddot{\mathbf{q}}_{\text{dqNE}}$ ) and the values obtained from V-REP ( $\ddot{\mathbf{q}}$ ) for the 9-DoF holonomic mobile manipulator. The closer to one, the more similar the waveforms are (adapted from Silva et al., 2022, table 2).

Method	CMC for the 9-DoF holonomic mobile manipulator			
	min	mean	std	max
dqNE	0.9938	0.9977	0.0022	0.9999

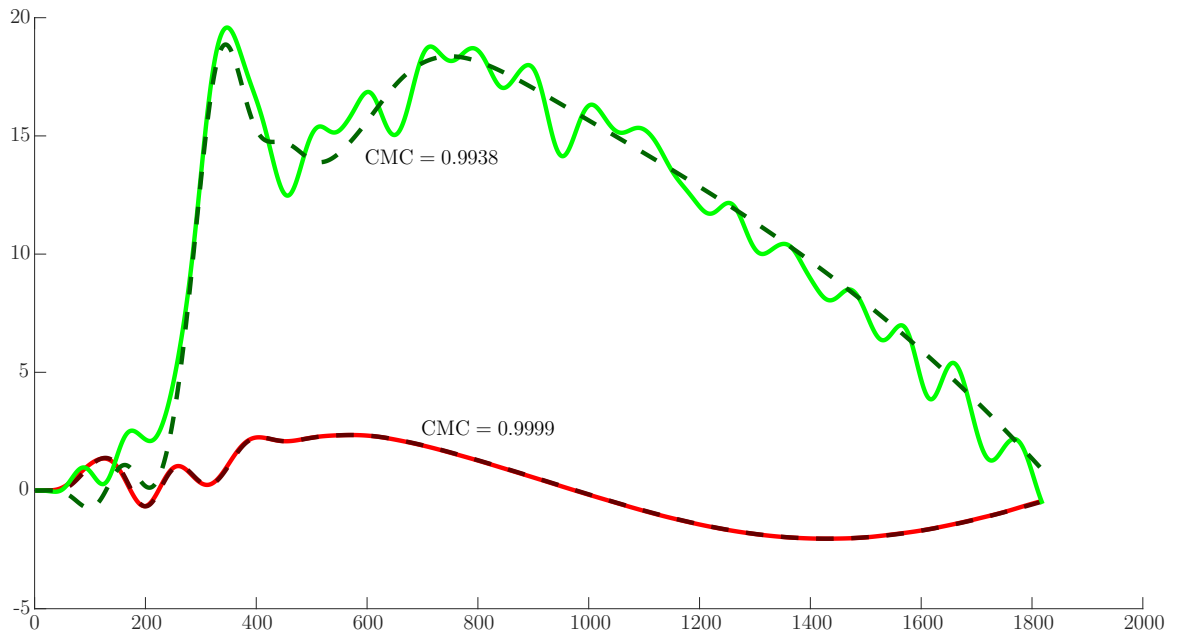


Figure 4.11: Generalized acceleration waveforms of the 9-DoF mobile manipulator. Solid curves correspond to the V-REP values, whereas dashed curves correspond to the values obtained using the dqNE for the generalized acceleration waveforms of the ninth (CMC = 0.9938) and third (CMC = 0.9999) joints, respectively.

### 4.3 Dynamic modeling of branched robots

This section presents the dynamical modeling of branched robots. The process is a generalization of the dual quaternion Newton-Euler formulation presented in section 4.2. To illustrative purposes, consider the 6-DoF branched robot presented in figure 4.12.

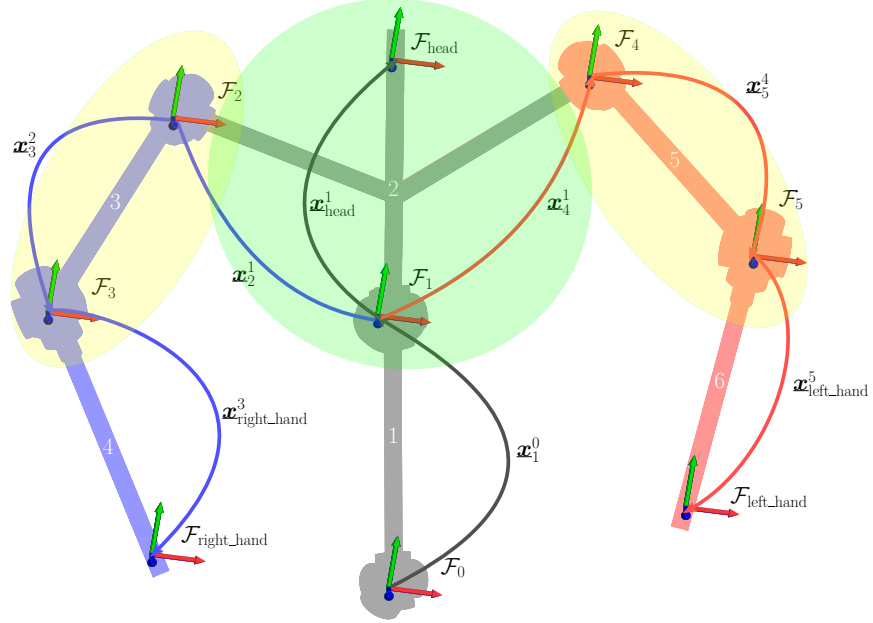


Figure 4.12: A 6-DoF branched robot. The links of the robot are numbered from 1 to 6. For equation (4.38), the second link, marked by the green ellipsis, is simultaneously labeled as  $p_3$  and  $p_5$ . For equation (4.39), the set  $S_2$  is composed by the links marked by the yellow ellipses; that is,  $S_2 = \{3, 5\}$ .

Since branched robots do not contain loops, they form an open kinematic tree. On this sort of mechanism, as happens with robot manipulators, each link of the robot is only *preceded* by a single link. This means that the forward recursion of the Newton-Euler formalism is the same when modeling the dynamics of an individual serial kinematic chain or a branched robot. However, equation (4.33) needs to be adapted since link labels are not sequential anymore. For instance, notice how frame  $\mathcal{F}_4$  is preceded by frame  $\mathcal{F}_1$ . Therefore, for a branched robot where the  $i$ -th link is preceded by the link  $p_i$ , the twist of the  $i$ -th CoM is given by

$$\underline{\xi}_{0,c_i}^{c_i} = \text{Ad} \left( \mathbf{x}_{c_{p_i}}^{c_i} \right) \mathcal{F}_{p_i} \left( \bar{\underline{\xi}}_{1,p_i} \right) + \text{Ad} \left( \mathbf{x}_{i-1}^{c_i} \right) \underline{\xi}_{i-1,c_i}^{i-1}. \quad (4.38)$$

On the other hand, the links of a branched robot can be *succeeded* by more than one element, which means that the backward recursion of the Newton-Euler formalism needs to be adapted. As wrenches belong to an additive group, the wrench of the  $i$ -th joint of the robot is given by the summation of all wrenches acting on the  $i$ -th link. Those are: the wrench at the CoM of the  $i$ -th link, obtained by using Newton's second law and the Euler's rotation equation, and the wrenches propagating backwards from all branches that

succeed this link on the robot. Therefore, for a branched robot composed of  $s$  links, where the link  $i \in \{1, 2, \dots, s\} \triangleq S$  is succeeded by the links  $j \in \{j_{i,1}, \dots, j_{i,m_i}\} \triangleq S_i \subset S$ , the wrench of the  $i$ -th joint is given by

$$\underline{\zeta}_{0,i}^{i-1} = \underline{\zeta}_{0,c_i}^{i-1} + \sum_{j \in S_i} \left( \text{Ad} \left( \mathbf{x}_{j-1}^{i-1} \right) \mathcal{B}_j \left( \bar{\underline{\xi}}_{j,n_j} \right) \right), \quad (4.39)$$

where  $n_j$  is the last link of the branch starting from link  $j \in S_i$ .

**Example 4.5.** Consider the simplified<sup>14</sup> upper body of the Poppy Humanoid, which forms the 8-DoF branched robot shown in figure 4.13. In this representation, frames  $\mathcal{F}_i$  and  $\mathcal{F}_{c_i}$  correspond to the  $i$ -th joint and  $i$ -th CoM, respectively, whereas the frames  $\mathcal{F}_{\text{head}}$ ,  $\mathcal{F}_{\text{right\_hand}}$ , and  $\mathcal{F}_{\text{left\_hand}}$ , indicate the head, right hand, and left hand of the robot, respectively. The robot comprises three branches going from frames  $\mathcal{F}_0$  to  $\mathcal{F}_{\text{head}}$ , represented by black arrows;  $\mathcal{F}_2$  to  $\mathcal{F}_{\text{right\_hand}}$ , depicted by green arrows; and  $\mathcal{F}_4$  to  $\mathcal{F}_{\text{left\_hand}}$ , illustrated by red arrows. A simplified drawing of this robot is given in figure 4.12.

The forward propagation of the dual quaternion Newton-Euler formalism is shown in figure 4.14a, where, for simplification, only the most relevant reference frames are displayed. The propagation of the twists starts at the first joint of the branched robot, represented by frame  $\mathcal{F}_0$ , and proceeds independently on each branch of the open kinematic tree (i.e., to the left and to the right) since there is no direct connection between the rigid bodies of each branch. The twists of CoMs of this branched robot are given by (4.38). For instance, the twists of the third and fifth CoMs are given by

$$\begin{aligned} \underline{\xi}_{0,c_3}^{c_3} &= \text{Ad} \left( \mathbf{x}_{c_{p_3}}^{c_3} \right) \underline{\mathcal{F}}_{p_3} \left( \bar{\underline{\xi}}_{1,p_3} \right) + \text{Ad} \left( \mathbf{x}_2^{c_3} \right) \underline{\xi}_{2,c_3}^2 = \text{Ad} \left( \mathbf{x}_{c_1}^{c_3} \right) \underline{\mathcal{F}}_1 \left( \bar{\underline{\xi}}_{1,1} \right) + \underline{\xi}_{2,c_3}^{c_3}, \\ \underline{\xi}_{0,c_5}^{c_5} &= \text{Ad} \left( \mathbf{x}_{c_{p_5}}^{c_5} \right) \underline{\mathcal{F}}_{p_5} \left( \bar{\underline{\xi}}_{1,p_5} \right) + \text{Ad} \left( \mathbf{x}_4^{c_5} \right) \underline{\xi}_{4,c_5}^4 = \text{Ad} \left( \mathbf{x}_{c_1}^{c_5} \right) \underline{\mathcal{F}}_1 \left( \bar{\underline{\xi}}_{1,1} \right) + \underline{\xi}_{4,c_5}^{c_5}, \end{aligned} \quad (4.40)$$

where the preceding link of both the third and fifth links is  $p_3 = p_5 = 1$ , which has the twist of its CoM given by  $\underline{\mathcal{F}}_{p_3} \left( \bar{\underline{\xi}}_{1,p_3} \right) = \underline{\mathcal{F}}_{p_5} \left( \bar{\underline{\xi}}_{1,p_5} \right) = \underline{\mathcal{F}}_1 \left( \bar{\underline{\xi}}_{1,1} \right) = \underline{\xi}_{0,c_1}^{c_1}$ , and the twists  $\underline{\xi}_{2,c_3}^{c_3}$  and  $\underline{\xi}_{4,c_5}^{c_5}$  are imposed by the movements of the third and fifth seventh joints, respectively.

Accordingly, the propagation of wrenches on each branch occurs independently during the backward recursion of the dual quaternion Newton-Euler formalism. This process is shown in figure 4.14b, where, for simplification, only the most relevant reference frames are displayed. For instance, the wrenches propagating from branches  $\mathcal{F}_{\text{right\_hand}}$  to  $\mathcal{F}_2$  and  $\mathcal{F}_{\text{left\_hand}}$  to  $\mathcal{F}_4$  converge at the second link of the robot. Considering (4.39), the wrench of

<sup>14</sup>To avoid cluttering the illustration, several of the robot joints are omitted. The real platform has 5-DoF on its torso and 4-DoF on each of its arms.



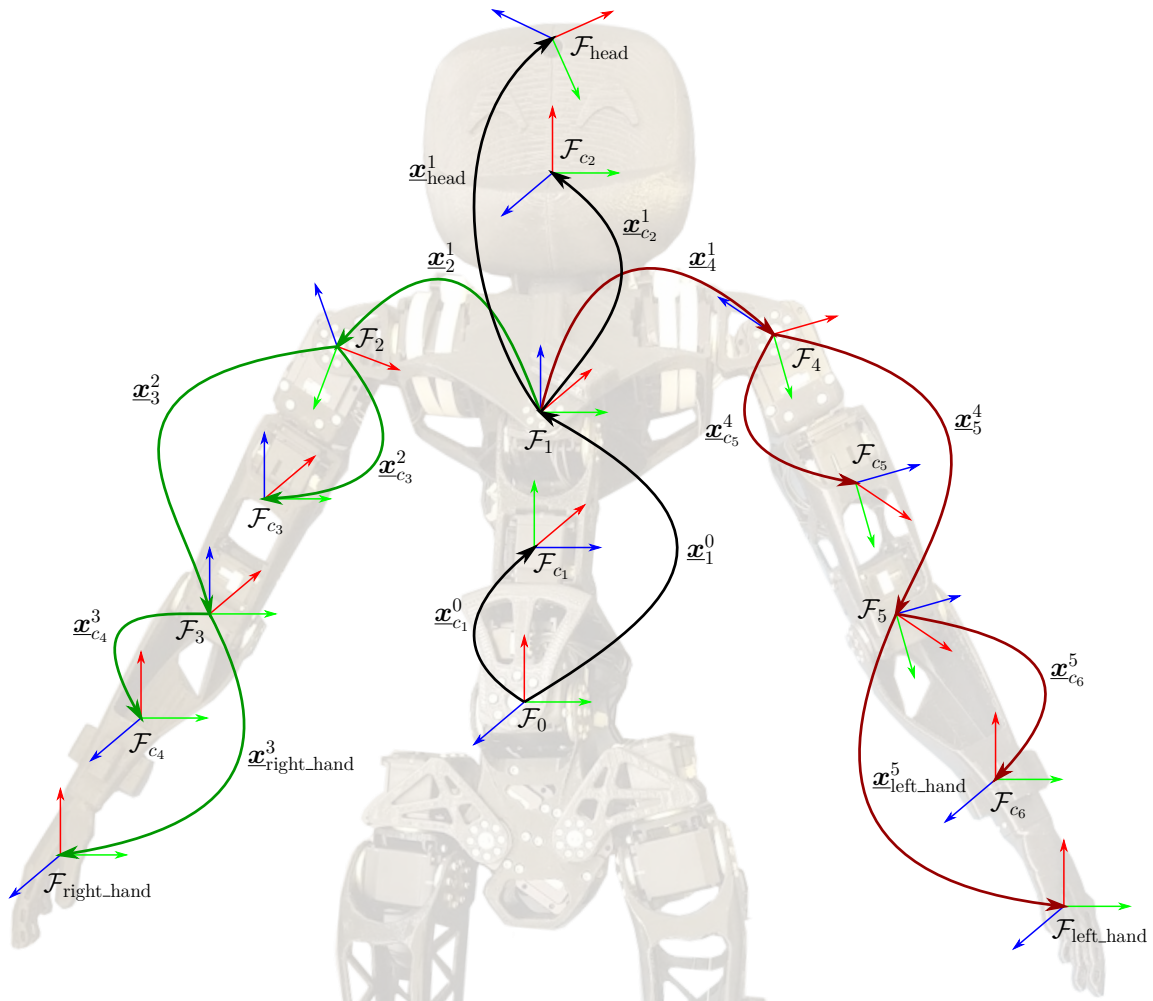
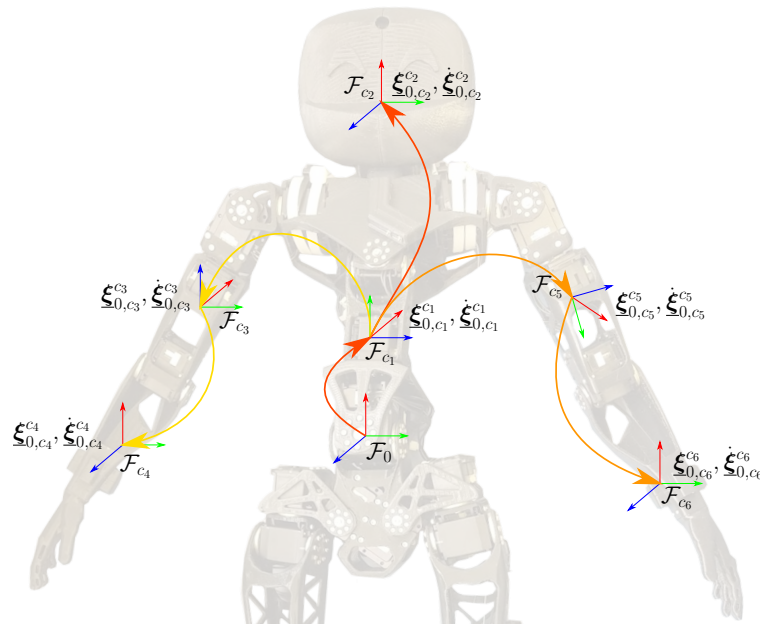


Figure 4.13: Simplified upper body of the Poppy Humanoid robot, which forms a 6-DoF branched robot. The frames  $\mathcal{F}_i$  and  $\mathcal{F}_{c_i}$  correspond to the  $i$ -th joint and  $i$ -th CoM, respectively, whereas the frames  $\mathcal{F}_{\text{head}}$ ,  $\mathcal{F}_{\text{right\_hand}}$ , and  $\mathcal{F}_{\text{left\_hand}}$ , indicate the head, right hand, and left hand of the robot, respectively. The robot comprises three branches going from frames  $\mathcal{F}_0$  to  $\mathcal{F}_{\text{head}}$ , represented by black arrows;  $\mathcal{F}_2$  to  $\mathcal{F}_{\text{right\_hand}}$ , depicted by green arrows; and  $\mathcal{F}_4$  to  $\mathcal{F}_{\text{left\_hand}}$ , illustrated by red arrows. A simplified drawing of this robot is given in figure 4.12.

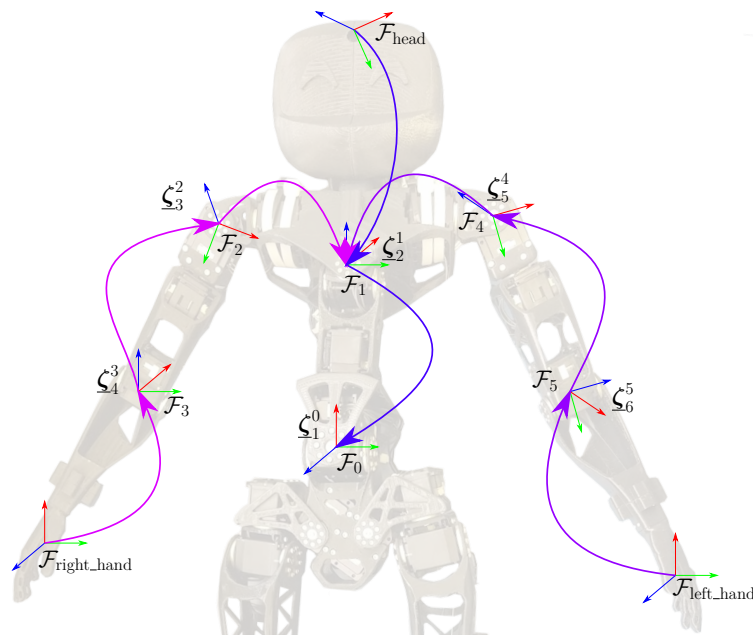
the second joint of the branched robot, which is represented by the frame  $\mathcal{F}_1$ , is given by

$$\begin{aligned}\underline{\zeta}_{0,2}^1 &= \underline{\zeta}_{0,c_2}^1 + \sum_{j \in S_2} \left( \text{Ad} \left( \underline{\mathbf{x}}_{j-1}^1 \right) \underline{\mathcal{B}}_j \left( \underline{\tilde{\mathbf{E}}}_{j,n_j} \right) \right) \\ &= \underline{\zeta}_{0,c_2}^1 + \left( \text{Ad} \left( \underline{\mathbf{x}}_2^1 \right) \underline{\mathcal{B}}_3 \left( \underline{\tilde{\mathbf{E}}}_{3,4} \right) + \text{Ad} \left( \underline{\mathbf{x}}_4^1 \right) \underline{\mathcal{B}}_5 \left( \underline{\tilde{\mathbf{E}}}_{5,6} \right) \right),\end{aligned}$$

where  $\underline{\zeta}_{0,c_2}^1$  is the wrench of the second CoM given by Newton's second law and the Euler's rotation equation (*see* equations (4.22) and (4.23)),  $S_2 = \{3, 5\}$ , and  $\underline{\mathcal{B}}_3 \left( \underline{\tilde{\mathbf{E}}}_{3,4} \right) = \underline{\zeta}_{0,3}^2$  and  $\underline{\mathcal{B}}_5 \left( \underline{\tilde{\mathbf{E}}}_{5,6} \right) = \underline{\zeta}_{0,5}^4$  are the total wrenches of the third and fifth joints, respectively.



(a) Forward recursion of the dual quaternion formalism in an open kinematic tree. Colored arrows represent the propagation of twists through different branches.



(b) Backward recursion of the dual quaternion formalism in an open kinematic tree. Colored arrows represent the propagation of wrenches through different branches.

Figure 4.14: Dual quaternion formalism applied to an open kinematic tree. Figure 4.14a presents the forward recursion, where twists and their time derivatives are propagated throughout the branches of the robot, whereas figure 4.14b illustrates the backward recursion and the propagation of the wrenches of the robot.

## 4.4 Conclusions

Section 4.1.1 proposed a novel quaternionic inertia tensor that allows the description of the dynamic properties of rigid bodies exclusively with operations inside the dual quaternion algebra, which was then used in section 4.2 for the deduction of the dual quaternion Newton-Euler formalism. The proposed technique consists in a forward sweeping of the kinematic chain to find the twists and their time derivatives, which are then used in a backward recursive sweeping of the robot to attain the wrenches. This unified representation has a straightforward treatment on dual quaternion algebra, in which the naturally coupled linear and angular components algebraically appear on the application of the adjoint transformation. Thus, the dual quaternion Newton-Euler formalism removes the necessity of a deep geometrical analysis of the system, which contrasts with the classic approach. Granted, for the specific case of robotic manipulators with revolute or prismatic joints, the classic Newton-Euler algorithm (Luh et al., 1980) consists of iterative equations that are easily implemented and readily found on undergraduate robotics textbooks (Spong et al., 2006; Siciliano & Khatib, 2008; Siciliano et al., 2009) and computational libraries (Corke, 2017). However, for more complex systems (e.g., tree structures, such as humanoids), this free-vector formalism demands geometrical analyses where all the coupled components must be carefully found by the roboticist performing the dynamic modeling.

Furthermore, compared to previous works, the proposed dual quaternion Newton-Euler formalism is more general because it works for arbitrary joint types and does not impose any particular parameterization convention for the propagation of twists. In contrast, take, for instance, the work of Miranda de Farias et al. (2019a), where the dual quaternion equations, albeit easily extendable to prismatic joints, currently only cover robotic manipulators with revolute joints and require screw displacements to be represented as proposed by Özgür & Mezouar (2016).

Moreover, both the proposed method and the classic Newton-Euler algorithm proposed by Luh et al. (1980) have linear costs in the number of degrees of freedom, with coefficients of the same order of magnitude. However, the dqNE is much more general. That is, the dual quaternion algebra allowed a generalization of the classic algorithm, while maintaining its overall order of magnitude for the computational cost.

One straightforward extension of the proposed dual quaternion Newton-Euler formalism was presented to the dynamic modeling of mobile robots. For those robot types, one only needs to obtain the twist of the mobile base and include it at the beginning of the iterative process, namely at (4.15) and (4.16). Section 4.2.3 presented the calculation of the twist of holonomic mobile bases.

Section 4.3 demonstrated the application of the proposed dual quaternion Newton-Euler formalism on branched robots. The only difference with regards to serial kinematic chains occurs at the connection points of the branches, where twists propagate independently

throughout each of them, whereas wrenches merge at the connection point and follow the main branch.

At this point, it is important to recall and critically compare the present work with the existing dynamic modeling approaches found in the literature. Representations aside, to be fair, all dynamic models based on screw theory present similar final equations (Pennock & Yang, 1983; Murray et al., 1994; Hachicho & Eldin, 2000; Spong et al., 2006; Featherstone, 2008; Siciliano & Khatib, 2008; Siciliano et al., 2009; Miranda de Farias et al., 2019a). One of the major advantages of this type of formulation is the capacity of algebraically find the coupled dynamical and kinematic elements caused by linear and angular components, namely through the adjoint transformation. Being a structural and intrinsic feature of the theory, this naturally extends to its different representations. Dynamic modeling strategies that take advantage of the adjoint transformation exist in approaches using matrices (Murray et al., 1994) and dual quaternions, where they go as far as the works of Dimentberg (1965).<sup>15</sup> In those regards, an interesting discussion concerning screw theory does not revolve around the final equations of the model but rather is concerned with how intuitive the chosen representation makes the process of deduction of the dynamic model and what insights it provides along the way. Thus, one of the major advantages of the proposed dual quaternion Newton-Euler formalism is its versatility and applicability to the dynamic modeling of different robotic structures, including serial kinematic robots with fixed and mobile bases and branched robots.

The next chapter is going to present a strategy for the dynamic modeling of branched robots using modular composition, which is one of the main contributions of this thesis.

---

<sup>15</sup>Dimentberg (1965) did not use exactly an adjoint transformation but an operation based on matrices of dual quaternions that yield the same results.

# 5

## MODULAR MODEL COMPOSITION

Chapter 4 presented the dual quaternion Newton-Euler formalism at the level of each rigid body in the robotic structure, notably, the “ $i$ -th link/joint/CoM.” Thus yielding a monolithic solution that allows the easy concatenation of a mobile base to a serial manipulator, as shown in section 4.2.3, and the analysis of the ramifications of a branched robot, as presented in section 4.3. The goal of this chapter is, however, to find a systematic formulation of a higher abstraction order, which will allow the composition of entire dynamic models, which are, in turn, comprised of several rigid bodies each. Moreover, section 5.2 presents a graph representation of the robotic system that encodes the propagation of twists and wrenches between the subsystems and aids the model composition. Furthermore, the proposed strategy is applicable even if some subsystems are regarded as black boxes, with complete dynamic information unavailable. The proposed modular composition formulation, alongside the dual quaternion Newton-Euler formalism presented in chapter 4, constitutes the main contributions of this thesis.

To illustrate the intuition behind the proposed formalism, let us consider the two serial kinematic chains with  $n_1$  and  $n_2$  degrees of freedom each, shown in figure 5.1. If we were to join both kinematic chains connecting the first joint of the  $n_2$ -DoF kinematic chain with the  $k$ -th link of the  $n_1$ -DoF kinematic chain, as indicated by the gray dashed line in figure 5.1, the dynamics of the newly-formed branched mechanism can still be given, at the low modeling level, by the procedure described in section 4.3 for branched robots. The objective of this chapter is, however, to find a strategy that gives the joint wrenches  $\underline{\mathbf{\Gamma}}_t \in \mathcal{H}_p^{n_1+n_2}$  of the overall system as a function of the dynamics of its subsystems.

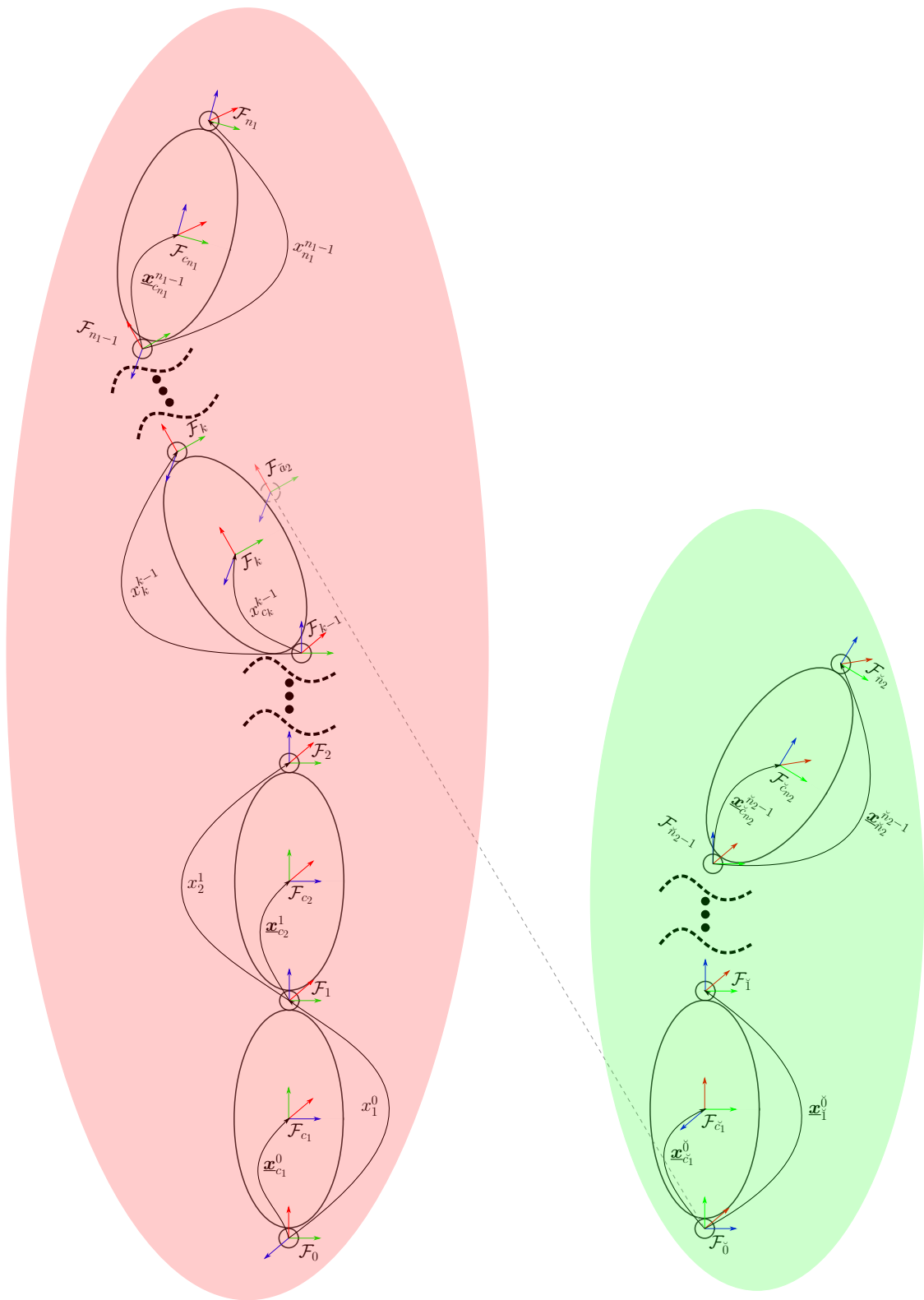


Figure 5.1: Two serial kinematic chains with  $n_1$  and  $n_2$  degrees of freedom.

To aid in this pursue, the next definition extends the adjoint operation of dual quaternions to the set  $\mathcal{H}_p^n$ .

**Definition 5.1.** Given a vector of unit dual quaternions  $\underline{\mathbf{X}} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_n]^T \in \underline{\mathbf{S}}^n$  and a pure dual quaternion  $\underline{\mathbf{a}} \in \mathcal{H}_p$ , the operator  $\text{Ad}_n : \underline{\mathbf{S}}^n \times \mathcal{H}_p \rightarrow \mathcal{H}_p^n$  is defined as

$$\text{Ad}_n(\underline{\mathbf{X}}) \underline{\mathbf{a}} \triangleq \text{diag}(\underline{\mathbf{X}}) \underline{\mathbf{a}} \begin{bmatrix} \mathbf{x}_1^* \\ \mathbf{x}_2^* \\ \vdots \\ \mathbf{x}_n^* \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{x}}_1 \underline{\mathbf{a}} \underline{\mathbf{x}}_1^* \\ \underline{\mathbf{x}}_2 \underline{\mathbf{a}} \underline{\mathbf{x}}_2^* \\ \vdots \\ \underline{\mathbf{x}}_n \underline{\mathbf{a}} \underline{\mathbf{x}}_n^* \end{bmatrix} = \begin{bmatrix} \text{Ad}(\mathbf{x}_1) \underline{\mathbf{a}} \\ \text{Ad}(\mathbf{x}_2) \underline{\mathbf{a}} \\ \vdots \\ \text{Ad}(\mathbf{x}_n) \underline{\mathbf{a}} \end{bmatrix}, \quad (5.1)$$

where  $\text{diag}(\underline{\mathbf{X}}) \in \mathcal{H}_p^{n \times n}$  is a diagonal matrix with the elements of  $\underline{\mathbf{X}}$  in its main diagonal.

Before being connected with the  $n_1$ -DoF serial kinematic chain, the CoM twists of the subsystem composed of  $n_2$  rigid bodies shown in figure 5.1, henceforth labeled as “subsystem 2,” was initially given by

$$\underline{\Xi}_{2,2} = \begin{bmatrix} \check{\xi}_{0,\check{c}_1}^{\check{c}_1} \\ \check{\xi}_{0,\check{c}_2}^{\check{c}_2} \\ \check{\xi}_{0,\check{c}_3}^{\check{c}_3} \\ \vdots \\ \check{\xi}_{0,\check{c}_{n_2}}^{\check{c}_{n_2}} \end{bmatrix} = \begin{bmatrix} \mathcal{F}_{\check{1}}(\check{\Xi}_{\check{1},\check{1}}) \\ \mathcal{F}_{\check{2}}(\check{\Xi}_{\check{1},\check{2}}) \\ \mathcal{F}_{\check{3}}(\check{\Xi}_{\check{1},\check{3}}) \\ \vdots \\ \mathcal{F}_{\check{n}_2}(\check{\Xi}_{\check{1},\check{n}_2}) \end{bmatrix}, \quad (5.2)$$

where  $\underline{\Xi}_{2,2} \in \mathcal{H}_p^n$  is the stacked vector of twists at all the CoMs of subsystem 2 and the symbol “ $\check{\cdot}$ ” indicate reference frames of the second subsystem. It is important to highlight here the difference between the double subscript in  $\check{\Xi}_{i,j}$  and in  $\Xi_{i,j}$ . For the monolithic solution, the subscripts in  $\check{\Xi}_{i,j}$  referred to the twists at the CoM from links  $i$  to  $j$ . However, in the context of modular composition, the subscripts in  $\Xi_{i,j}$  refer to the CoM twists in subsystem  $j$  that originated from its interaction with subsystem  $i$ . Thus, for instance,  $\underline{\Xi}_{2,2}$  is the stacked vector of twists at all the CoMs of subsystem 2 that originated from subsystem 2 itself.

Considering the recurrence (4.38), equation (5.2) can be expanded as

$$\underline{\Xi}_{2,2} = \begin{bmatrix} \mathcal{F}_{\check{1}}(\check{\Xi}_{\check{1},\check{1}}) \\ \check{\xi}_{\check{1},\check{c}_2}^{\check{c}_2} + \text{Ad}(\mathbf{x}_{\check{c}_1}^{\check{c}_2}) \mathcal{F}_{\check{1}}(\check{\Xi}_{\check{1},\check{1}}) \\ \check{\xi}_{\check{2},\check{c}_3}^{\check{c}_3} + \text{Ad}(\mathbf{x}_{\check{c}_2}^{\check{c}_3}) \mathcal{F}_{\check{2}}(\check{\Xi}_{\check{1},\check{2}}) \\ \vdots \\ \check{\xi}_{\check{n}_2-1,\check{c}_{n_2}}^{\check{c}_{n_2}} + \text{Ad}(\mathbf{x}_{\check{c}_{n_2-1}}^{\check{c}_{n_2}}) \mathcal{F}_{\check{n}_2-1}(\check{\Xi}_{\check{1},\check{n}_2-1}) \end{bmatrix},$$

which, considering that  $\text{row}_i(\underline{\mathbf{A}}) \triangleq \underline{\mathbf{A}}_i \in \mathcal{H}^n$  gives the  $i$ -th row of the matrix  $\underline{\mathbf{A}} =$



$[\underline{\mathbf{A}}_1 \ \dots \ \underline{\mathbf{A}}_m] \in \mathcal{H}^{n \times m}$ , can then be further expanded to isolate the term  $\underline{\mathcal{F}}_{\check{1}}(\bar{\underline{\Xi}}_{\check{1},\check{1}})$ ,

$$\begin{aligned}
 \text{row}_1(\underline{\Xi}_{2,2}) &= \underline{\mathcal{F}}_{\check{1}}(\bar{\underline{\Xi}}_{\check{1},\check{1}}) \\
 \text{row}_2(\underline{\Xi}_{2,2}) &= \underline{\xi}_{\check{1},\check{c}_2}^{\check{c}_2} + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_1}^{\check{c}_2}) \underline{\mathcal{F}}_{\check{1}}(\bar{\underline{\Xi}}_{\check{1},\check{1}}) \\
 \text{row}_3(\underline{\Xi}_{2,2}) &= \underline{\xi}_{\check{2},\check{c}_3}^{\check{c}_3} + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_2}^{\check{c}_3}) (\underline{\xi}_{\check{1},\check{c}_2}^{\check{c}_2} + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_1}^{\check{c}_2}) \underline{\mathcal{F}}_{\check{1}}(\bar{\underline{\Xi}}_{\check{1},\check{1}})) \\
 &\vdots \\
 \text{row}_n(\underline{\Xi}_{2,2}) &= \underline{\xi}_{\check{n}_2-1,\check{c}_{n_2}}^{\check{c}_{n_2}} + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_{n_2-1}}^{\check{c}_{n_2}}) (\underline{\xi}_{\check{n}_2-2,\check{c}_{n_2-1}}^{\check{c}_{n_2-1}} + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_{n_2-2}}^{\check{c}_{n_2-1}}) (\underline{\xi}_{\check{n}_2-3,\check{c}_{n_2-2}}^{\check{c}_{n_2-2}} + \dots \\
 &\quad + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_2}^{\check{c}_3}) (\underline{\xi}_{\check{1},\check{c}_2}^{\check{c}_2} + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_1}^{\check{c}_2}) \underline{\mathcal{F}}_{\check{1}}(\bar{\underline{\Xi}}_{\check{1},\check{1}})) \dots)) . \tag{5.3}
 \end{aligned}$$

Once connected with the  $k$ -th link of subsystem 1 (i.e., the  $n_1$ -DoF kinematic chain), as indicated by the gray dashed line in figure 5.1, the twist  $\underline{\xi}_{0,c_k}^{c_k}$  of the  $k$ -th CoM of subsystem 1 will be added to the twist of the first CoM of subsystem 2, represented by the frame  $\underline{\mathcal{F}}_{\check{c}_1}$ , and, consequently, propagated throughout subsystem 2. This will result in<sup>1</sup>

$$\begin{aligned}
 \text{row}_1(\underline{\Xi}_2) &= \underline{\mathcal{F}}_{\check{1}}(\bar{\underline{\Xi}}_{\check{1},\check{1}}) + \text{Ad}(\underline{\mathbf{x}}_{c_k}^{\check{c}_1}) \underline{\xi}_{0,c_k}^{c_k} \\
 \text{row}_2(\underline{\Xi}_2) &= \underline{\xi}_{\check{1},\check{c}_2}^{\check{c}_2} + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_1}^{\check{c}_2}) (\underline{\mathcal{F}}_{\check{1}}(\bar{\underline{\Xi}}_{\check{1},\check{1}}) + \text{Ad}(\underline{\mathbf{x}}_{c_k}^{\check{c}_1}) \underline{\xi}_{0,c_k}^{c_k}) \\
 \text{row}_3(\underline{\Xi}_2) &= \underline{\xi}_{\check{2},\check{c}_3}^{\check{c}_3} + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_2}^{\check{c}_3}) (\underline{\xi}_{\check{1},\check{c}_2}^{\check{c}_2} + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_1}^{\check{c}_2}) (\underline{\mathcal{F}}_{\check{1}}(\bar{\underline{\Xi}}_{\check{1},\check{1}}) + \text{Ad}(\underline{\mathbf{x}}_{c_k}^{\check{c}_1}) \underline{\xi}_{0,c_k}^{c_k})) \\
 &\vdots \\
 \text{row}_n(\underline{\Xi}_2) &= \underline{\xi}_{\check{n}_2-1,\check{c}_{n_2}}^{\check{c}_{n_2}} + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_{n_2-1}}^{\check{c}_{n_2}}) (\underline{\xi}_{\check{n}_2-2,\check{c}_{n_2-1}}^{\check{c}_{n_2-1}} + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_{n_2-2}}^{\check{c}_{n_2-1}}) (\underline{\xi}_{\check{n}_2-3,\check{c}_{n_2-2}}^{\check{c}_{n_2-2}} + \dots \\
 &\quad + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_2}^{\check{c}_3}) (\underline{\xi}_{\check{1},\check{c}_2}^{\check{c}_2} + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_1}^{\check{c}_2}) (\underline{\mathcal{F}}_{\check{1}}(\bar{\underline{\Xi}}_{\check{1},\check{1}}) + \text{Ad}(\underline{\mathbf{x}}_{c_k}^{\check{c}_1}) \underline{\xi}_{0,c_k}^{c_k})) \dots)) . \tag{5.4}
 \end{aligned}$$

Rearranging the terms of (5.4), such that all the elements  $\underline{\xi}_{0,c_k}^{c_k}$  are isolated, leads to

$$\begin{aligned}
 \text{row}_1(\underline{\Xi}_2) &= \underline{\mathcal{F}}_{\check{1}}(\bar{\underline{\Xi}}_{\check{1},\check{1}}) + \text{Ad}(\underline{\mathbf{x}}_{c_k}^{\check{c}_1}) \underline{\xi}_{0,c_k}^{c_k} \\
 \text{row}_2(\underline{\Xi}_2) &= \underline{\xi}_{\check{1},\check{c}_2}^{\check{c}_2} + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_1}^{\check{c}_2}) \underline{\mathcal{F}}_{\check{1}}(\bar{\underline{\Xi}}_{\check{1},\check{1}}) + \text{Ad}(\underline{\mathbf{x}}_{c_k}^{\check{c}_1}) \underline{\xi}_{0,c_k}^{c_k} \\
 \text{row}_3(\underline{\Xi}_2) &= \underline{\xi}_{\check{2},\check{c}_3}^{\check{c}_3} + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_2}^{\check{c}_3}) (\underline{\xi}_{\check{1},\check{c}_2}^{\check{c}_2} + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_1}^{\check{c}_2}) \underline{\mathcal{F}}_{\check{1}}(\bar{\underline{\Xi}}_{\check{1},\check{1}})) + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_1}^{\check{c}_2}) \text{Ad}(\underline{\mathbf{x}}_{c_k}^{\check{c}_1}) \underline{\xi}_{0,c_k}^{c_k} \\
 &\vdots \\
 \text{row}_n(\underline{\Xi}_2) &= \underline{\xi}_{\check{n}_2-1,\check{c}_{n_2}}^{\check{c}_{n_2}} + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_{n_2-1}}^{\check{c}_{n_2}}) (\underline{\xi}_{\check{n}_2-2,\check{c}_{n_2-1}}^{\check{c}_{n_2-1}} + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_{n_2-2}}^{\check{c}_{n_2-1}}) (\underline{\xi}_{\check{n}_2-3,\check{c}_{n_2-2}}^{\check{c}_{n_2-2}} + \dots \\
 &\quad + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_2}^{\check{c}_3}) (\underline{\xi}_{\check{1},\check{c}_2}^{\check{c}_2} + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_1}^{\check{c}_2}) \underline{\mathcal{F}}_{\check{1}}(\bar{\underline{\Xi}}_{\check{1},\check{1}})) \dots)) \\
 &\quad + \text{Ad}(\underline{\mathbf{x}}_{\check{c}_{n_2-1}}^{\check{c}_{n_2}}) \dots \text{Ad}(\underline{\mathbf{x}}_{\check{c}_2}^{\check{c}_3}) \text{Ad}(\underline{\mathbf{x}}_{\check{c}_1}^{\check{c}_2}) \text{Ad}(\underline{\mathbf{x}}_{c_k}^{\check{c}_1}) \underline{\xi}_{0,c_k}^{c_k} . \tag{5.5}
 \end{aligned}$$

Then, considering that the first half of the expression of  $\underline{\Xi}_2$  is identical to (5.3), which, in turn, is an expansion of (5.2), and applying the  $\text{Ad}_n$  operator (*see* (5.1)) to simplify the

<sup>1</sup>Notice the single subscript in  $\underline{\Xi}_2$ . This indicate that this is the stacked vector of the total twists at the CoMs of subsystem 2, originated from all the interactions between the it and its succeeding subsystem. The final expression of  $\underline{\Xi}_2$ , that will be shown shortly, will further clarify this meaning.

isolated vector containing  $\underline{\xi}_{0,c_k}^{c_k}$ , equation (5.5) can be further organized as

$$\begin{aligned}
 \underline{\Xi}_2 &= \begin{bmatrix} \underline{\xi}_{0,\check{c}_1}^{\check{c}_1} \\ \underline{\xi}_{0,\check{c}_2}^{\check{c}_2} \\ \underline{\xi}_{0,\check{c}_3}^{\check{c}_3} \\ \vdots \\ \underline{\xi}_{0,\check{c}_{n_2}}^{\check{c}_{n_2}} \end{bmatrix} + \begin{bmatrix} \text{Ad} \left( \underline{\mathbf{x}}_{c_k}^{\check{c}_1} \right) \underline{\xi}_{0,c_k}^{c_k} \\ \text{Ad} \left( \underline{\mathbf{x}}_{c_k}^{\check{c}_2} \right) \underline{\xi}_{0,c_k}^{c_k} \\ \text{Ad} \left( \underline{\mathbf{x}}_{c_k}^{\check{c}_3} \right) \underline{\xi}_{0,c_k}^{c_k} \\ \vdots \\ \text{Ad} \left( \underline{\mathbf{x}}_{c_k}^{\check{c}_{n_2}} \right) \underline{\xi}_{0,c_k}^{c_k} \end{bmatrix} \\
 &= \underline{\Xi}_{2,2} + \underbrace{\text{Ad}_{n_2} \left( \underline{\mathbf{X}}_{1,\check{z}} \right) \underline{\xi}_{0,c_k}^{c_k}}_{\underline{\Xi}_{1,2}} \\
 &= \underline{\Xi}_{2,2} + \underline{\Xi}_{1,2}, \tag{5.6}
 \end{aligned}$$

where  $\underline{\Xi}_{1,2} \triangleq \text{Ad}_{n_2} \left( \underline{\mathbf{X}}_{1,\check{z}} \right) \underline{\xi}_{0,c_k}^{c_k}$  and  $\underline{\mathbf{X}}_{1,\check{z}} = \left[ \underline{\mathbf{x}}_{c_k}^{\check{c}_1} \quad \underline{\mathbf{x}}_{c_k}^{\check{c}_2} \quad \underline{\mathbf{x}}_{c_k}^{\check{c}_3} \quad \cdots \quad \underline{\mathbf{x}}_{c_k}^{\check{c}_{n_2}} \right]^T \in \underline{\mathbf{S}}^{n_2}$ . Thus, the total twists at the CoMs of subsystem 2 ( $\underline{\Xi}_2$ ) are a combination of the twists originated from its own motion ( $\underline{\Xi}_{2,2}$ ) and from its interaction of subsystem 1 ( $\underline{\Xi}_{1,2}$ ).

Equation (5.6) shows an important aspect of twist propagation between dynamic systems. There are two ways of performing the process. The direct approach is to add the twist of the preceding subsystem directly to the first joint of its successor and, subsequently, let the Newton-Euler formalism's forward recursion propagate it throughout the system, which leads to (5.4). Alternatively, however, the new twist can be propagated independently, through the  $\text{Ad}_n$  operation, and then have its effects added to all the succeeding subsystem's joints at once, resulting in (5.6). The same reasoning applies to the backward recursion and wrench propagation.

Albeit arguably not the most straightforward solution to twist/wrench propagation between subsystems, the latter approach allows a clear separation between the subsystem's own dynamics (e.g.,  $\underline{\Xi}_{2,2}$ ) and the dynamic information that originates from a different subsystem (e.g.,  $\underline{\Xi}_{1,2}$ ). This, as will be shown shortly, leads to more compact high-level dynamic equations of the overall integrated system.

## 5.1 The dual quaternion modular model composition strategy

Consider a branched robot composed of a set of  $s$  coupled subsystems, such as the one partially shown in figure 5.2, in which each subsystem  $i \in \{1, \dots, s\} \triangleq S$  is composed of  $n_i$  joints/links, is preceded by a subsystem  $p_i \in S$ , and is immediately succeeded by the subsystems  $j \in \{j_{i,1}, \dots, j_{i,m_i}\} \triangleq S_i$ . Consider also that the subsystem  $i$  is connected to the subsystem  $p_i$  at the point  $a_i$  and to each of the subsystems  $j$  at the connection points  $b_{i,j}$ .

Finally, consider the following definition, which presents a function that maps the vector of total twists at the CoM of each link of a subsystem to the corresponding vector of wrenches at its joints. This can be seen as a generalization of the previously defined function  $\underline{\mathcal{B}}$ , given by (4.32), to the set  $\mathcal{H}_p^n$ .

**Definition 5.2.** For a subsystem  $i$ , the function  $\underline{\mathcal{W}}_i : \mathcal{H}_p^{n_i} \rightarrow \mathcal{H}_p^{n_i}$ , that gives the wrenches  $\underline{\Gamma}_i \in \mathcal{H}_p^{n_i}$  at the  $n_i$  joints of robot as a function of the vector  $\underline{\Xi}_i = \begin{bmatrix} \underline{\xi}_{0,c_1}^{c_1} & \cdots & \underline{\xi}_{0,c_{n_i}}^{c_{n_i}} \end{bmatrix}^T \in \mathcal{H}_p^{n_i}$  of total twists at the CoM of each link, is defined as<sup>2</sup>

$$\underline{\Gamma}_i = \underline{\mathcal{W}}_i(\underline{\Xi}_i). \quad (5.7)$$

As discussed in section 4.3, the composed branched robot is an open kinematic tree, which means that each subsystem can only be preceded by one subsystem but can be succeeded by several kinematic chains. Therefore, twists generated by the subsystem  $i$  will be propagated to each  $j \in S_i$  and all other subsequent subsystems. On the other hand, the combined wrenches from all  $j \in S_i$  will affect  $i$  and all its preceding subsystems. Therefore, the wrenches at the joints of each subsystem  $i \in S$  originate from three sources: the twists (and their derivatives) at the CoMs of each link in the  $i$ -th subsystem; the twist (and its derivative) at the connection point with  $a_i$ ; and the wrenches at the connection points  $b_{i,j}$ .

The wrenches generated at the joints of the  $i$ -th subsystem as a result of its own motion, the motion of its predecessor, and the wrenches from its successors are shown in figure 5.2. The connection point between  $p_i$  and the  $i$ -th subsystem, denoted by 1 in figure 5.2, contains the *resultant* twist generated by all moving joints from the connection point up to the root node. On the other hand, the connection points  $j \in S_i$  between subsequent kinematic chains connected to the  $i$ -th subsystem, denoted by 2 and 3, contain the *resultant* wrenches generated by those systems and all their successors.

Therefore, to calculate the wrenches at the joints of the  $i$ -th subsystem, one needs the information of the twists (and its derivatives) at its CoMs, the twist at the connection point with its predecessor, and the wrenches at the connection points with its successors. Thus, the dynamic model of the composed branched robot is given by

$$\underline{\Gamma}_t = \begin{bmatrix} \underline{\bar{\Gamma}}_1 \\ \underline{\bar{\Gamma}}_2 \\ \vdots \\ \underline{\bar{\Gamma}}_k \end{bmatrix} \in \mathcal{H}_p^n, \quad (5.8)$$

---

<sup>2</sup>Similarly to what was previously discussed for the function  $\underline{\mathcal{B}}$  (see (4.32)), the derivatives of the twists used to calculate the wrenches can be explicitly calculated in the function  $\underline{\mathcal{W}}_i$ .

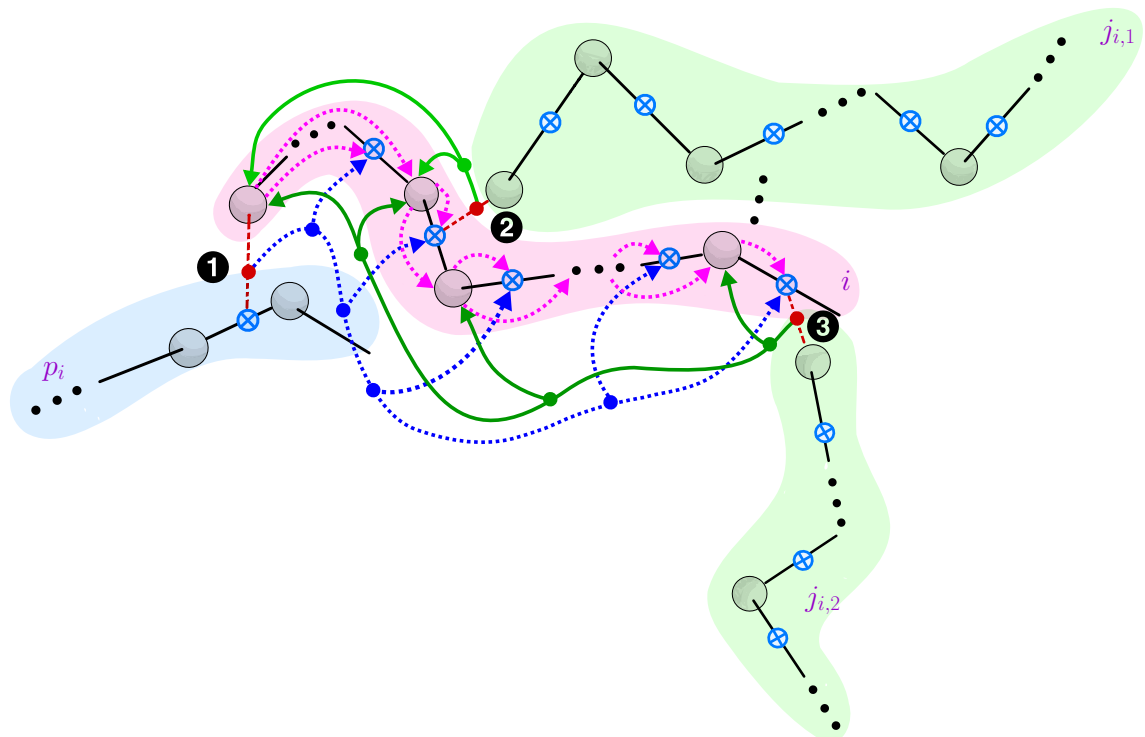


Figure 5.2: Wrenches generated at the joints of the  $i$ -th subsystem. For each subsystem, the large gray circles represent joints, solid black lines represent links, and blue circles with crosses represent CoMs. Subsystem  $i$  is represented in pink, subsystem  $p_i$  is given in blue, subsystems  $j \in \{j_{i,1}, j_{i,2}\} \triangleq S_i$  are colored in green, and red circles on the red dashed lines, numbered from 1 to 3, indicate the connection points  $a_i$ ,  $b_{i,j}$ , with  $j \in S_i$ , respectively. Dotted arrows represent the forward propagation of twists, whereas solid arrows represent the backward propagation of wrenches.

where  $n = \sum n_i$  and

$$\mathbf{\Gamma}_i = \mathbf{W}_i(\mathbf{\Xi}_i) + \sum_{j \in S_i} \mathring{\mathbf{\Gamma}}_{j,i}, \quad \text{with } \mathbf{\Xi}_i = \mathbf{\Xi}_{p_i,i} + \mathbf{\Xi}_{i,i}, \quad (5.9)$$

in which the function  $\mathbf{W}_i$  is given by (5.7);  $\mathbf{\Xi}_{i,i} \in \mathcal{H}_p^{n_i}$  is the stacked vector of twists at the  $n_i$  CoMs of subsystem  $i \in S$  originated from its own motion; the elements of  $\mathbf{\Xi}_{p_i,i} \in \mathcal{H}_p^{n_i}$  are the twist at the connection point with  $a_i$  expressed in each of the  $n_i$  CoMs of subsystem  $i$ , given by

$$\mathbf{\Xi}_{p_i,i} = \text{Ad}_{n_i} \left( \mathbf{X}_{\check{p}_i,i} \right) \mathbf{\xi}_{0,\check{a}_i}^{\check{a}_i}, \quad (5.10)$$

where  $\mathbf{X}_{\check{p}_i,i} = \left[ \mathbf{x}_{\check{a}_i}^{c_1} \quad \dots \quad \mathbf{x}_{\check{a}_i}^{c_{n_i}} \right]^T \in \mathbf{S}^{n_i}$  is the vector of the relative poses between the connection point  $a_i$  and the CoMs of the  $i$ -th subsystem and  $\mathbf{\xi}_{0,\check{a}_i}^{\check{a}_i}$  is the twist of the connection point  $a_i$  given with respect to its own reference frame  $\mathcal{F}_{\check{a}_i}$ ; and the elements of  $\mathring{\mathbf{\Gamma}}_{j,i} \in \mathcal{H}_p^{n_i}$  are the wrenches of the connection point  $b_{i,j}$ , with  $j \in S_i$ , expressed in each of the  $\eta \leq n_i$  joints of subsystem  $i$  that precedes  $b_{i,j}$ , given by

$$\mathring{\mathbf{\Gamma}}_{j,i} = \text{Ad}_{n_i} \left( \mathbf{X}_{\bar{j},i} \right) \mathbf{\zeta}_{0,\bar{b}_{i,j}}^{\bar{b}_{i,j}}, \quad (5.11)$$

in which  $\mathbf{X}_{\bar{j},i} = \left[ \mathbf{X}_{\bar{j},i} \quad \mathbf{0}_{n_i-\eta} \right]^T \in \mathbf{S}^{n_i}$ , with  $\mathbf{X}_{\bar{j},i} = \left[ \mathbf{x}_{\bar{b}_{i,j}}^0 \quad \dots \quad \mathbf{x}_{\bar{b}_{i,j}}^\eta \right]^T \in \mathbf{S}^\eta$  being the vector of the relative poses between the connection point  $b_{i,j}$  and each of the  $\eta \leq n_i$  joints of subsystem  $i$  that precede this connection point,  $\mathbf{0}_{n_i-\eta} \in \mathbb{R}^{n_i-\eta} \subset \mathcal{H}_p^{n_i-\eta}$  is a vector of zeros, and  $\mathbf{\zeta}_{0,\bar{b}_{i,j}}^{\bar{b}_{i,j}}$  is the wrench at the connection point  $b_{i,j}$  given with respect to its own reference frame  $\mathcal{F}_{\bar{b}_{i,j}}$ .

The proposed modular composition strategy is applicable to two different scenarios. Strictly speaking, besides its own information, each subsystem  $i$  only requires the twist  $\mathbf{\xi}_{0,\check{a}_i}^{\check{a}_i}$  and wrench  $\mathbf{\zeta}_{0,\bar{b}_{i,j}}^{\bar{b}_{i,j}}$  of its connection points with other subsystems in the branched structure (see (5.10) and (5.11)). Therefore, to calculate the joint wrenches of the  $i$ -th subsystem, there is no necessity of knowing the remaining kinematic (e.g., current configuration, generalized velocities, etc.) or dynamic (e.g., masses, inertia tensors, etc.) properties of the neighboring subsystems. Thus, even if some subsystems are presented as black boxes, the joint wrenches of the overall system can still be obtained, as long as we have  $\mathbf{\xi}_{0,\check{a}_i}^{\check{a}_i}$  and  $\mathbf{\zeta}_{0,\bar{b}_{i,j}}^{\bar{b}_{i,j}}$  (e.g., through sensor readings). Furthermore, this characteristic of the proposed modular composition makes it a viable candidate for parallelization.

Alternatively, we could be dealing with the situation described in the beginning of this chapter, where the dynamic models of the subsystems are known from previous modeling processes but we do not necessarily have direct access to  $\mathbf{\xi}_{0,\check{a}_i}^{\check{a}_i}$  and  $\mathbf{\zeta}_{0,\bar{b}_{i,j}}^{\bar{b}_{i,j}}$  (see figure 5.1). In this scenario, using the information of the CoM twists of subsystem  $p_i$  and the joint wrenches of succeeding subsystem  $j \in S_i$ , it would still be possible to calculate the joint wrenches of the  $i$ -th subsystem. Considering that the subsystem  $i$  is connected to the

point  $a_i$  at the  $k$ -th link of subsystem  $p_i$ , the twist at the connection point  $a_i$ , given with respect to its own reference frame  $\mathcal{F}_{\check{a}_i}$ , can be calculated as  $\xi_{0,\check{a}_i}^{\check{a}_i} = \text{Ad} \left( \mathbf{x}_{\check{c}_k}^{\check{a}_i} \right) \mathcal{F}_{\check{c}_k} \left( \bar{\Xi}_{\check{1},\check{k}} \right)$  (e.g., see (5.6)), and the wrenches at the connection points  $b_{i,j}$ , given with respect to its own reference frame  $\mathcal{F}_{\check{b}_{i,j}}$ , can be calculated as  $\xi_{0,\check{b}_{i,j}}^{\check{b}_{i,j}} = \text{Ad} \left( \mathbf{x}_{\check{\eta}-1}^{\check{b}_{i,j}} \right) \mathcal{B}_{\check{1}} \left( \bar{\Xi}_{\check{1},\check{n}_j} \right)$ , where functions  $\mathcal{F}$  and  $\mathcal{B}$ , given by (4.32), respectively give the twist at the  $i$ -th CoM and the wrench at the  $i$ -th joint of a subsystem. Furthermore,  $\mathcal{B}_{\check{1}} \left( \bar{\Xi}_{\check{1},\check{n}_j} \right)$  is also given by the first element of  $\bar{\Gamma}_{\check{j}}$  and, consequently, equation (5.9) can be calculated iteratively.

The following example illustrates the application of the proposed modular composition strategy to derive the dynamic model of the  $(n_1 + n_2)$ -DoF branched mechanism shown in figure 5.1.

**Example 5.1.** Consider the two independent serial kinematic chains with  $n_1$ -DoF and  $n_2$ -DoF shown in figure 5.1, whose joint wrenches are given by

$$\begin{aligned} \Gamma_1 &= \mathcal{W}_1(\Xi_1) \in \mathcal{H}_p^{n_1}, \\ \Gamma_2 &= \mathcal{W}_2(\Xi_2) \in \mathcal{H}_p^{n_2}, \end{aligned}$$

respectively. Now, let us join both kinematic chains connecting the first joint of the  $n_2$ -DoF kinematic chain at the point  $a_2$  located on the  $k$ -th link of the  $n_1$ -DoF kinematic chain, as indicated by the gray dashed line in figure 5.1. The reference frames of the joints of subsystems 1 (the  $n_1$ -DoF kinematic chain) and 2 (the  $n_2$ -DoF kinematic chain) are, respectively, given by  $\mathcal{F}_i$  and  $\mathcal{F}_{\check{i}}$ , and the reference frame of the connection point  $a_2$  between them is given by  $\mathcal{F}_{\check{a}_2}$ .

From (5.8), the joint wrenches vector  $\Gamma_t$  of the assembled branched mechanism is given by

$$\Gamma_t = \begin{bmatrix} \bar{\Gamma}_1 \\ \bar{\Gamma}_2 \end{bmatrix} \in \mathcal{H}_p^{n_1+n_2},$$

where the set of succeeding kinematic chains are  $S_1 = \{2\}$  and  $S_2 = \emptyset$  and

$$\begin{aligned} \bar{\Gamma}_1 &= \mathcal{W}_1(\Xi_1) + \dot{\Gamma}_{2,1} = \mathcal{W}_1(\Xi_{1,1}) + \dot{\Gamma}_{2,1}, \\ \bar{\Gamma}_2 &= \mathcal{W}_2(\Xi_2) = \mathcal{W}_2(\Xi_{1,2} + \Xi_{2,2}), \end{aligned} \quad (5.12)$$

in which  $\mathcal{W}_1(\Xi_1) = \mathcal{W}_1(\Xi_{1,1})$  is the vector of wrenches generated at the joints of subsystem 1 as a result of its own motion,  $\dot{\Gamma}_{2,1}$  is the vector of wrenches generated by its interaction with subsystem 2, its successor in the branched structure, and  $\mathcal{W}_2(\Xi_2) = \mathcal{W}_2(\Xi_{1,2} + \Xi_{2,2})$  is the vector of wrenches generated at the joints of subsystem 2 by the combination of its own motion ( $\Xi_{2,2}$ ) and the motion of subsystem 1 ( $\Xi_{1,2}$ ), its predecessor in the branched structure. Moreover,

$$\Xi_{1,2} = \text{Ad}_{n_2} \left( \mathbf{X}_{\check{1},\check{2}} \right) \xi_{0,\check{a}_2}^{\check{a}_2}, \quad (5.13)$$

where  $\underline{\xi}_{0,\bar{a}_2}^{\bar{a}_2} = \text{Ad}(\underline{x}_{c_k}^{\bar{a}_2}) \mathcal{F}_k(\underline{\Xi}_{1,k})$  is the twist at the connection point  $a_2$  that connects the two subsystems, and

$$\underline{\dot{\Gamma}}_{2,1} = \text{Ad}_{n_1}(\underline{X}_{2,1}) \underline{\xi}_{0,\bar{b}_{1,2}}^{\bar{b}_{1,2}}, \quad (5.14)$$

where  $\underline{\xi}_{0,\bar{b}_{1,2}}^{\bar{b}_{1,2}} = \text{Ad}(\underline{x}_{k-1}^{\bar{b}_{1,2}}) \mathcal{B}_1(\underline{\Xi}_{1,\bar{n}_2})$  is the wrench at  $b_{1,2} = a_2$  (represented in figure 5.1 by the frame  $\mathcal{F}_{\bar{a}_2}$ ). Furthermore, the pose transformation vectors are given by

$$\underline{X}_{\bar{1},\bar{2}} = \begin{bmatrix} \underline{x}_{\bar{a}_2}^{\check{c}_1} & \underline{x}_{\bar{a}_2}^{\check{c}_2} & \cdots & \underline{x}_{\bar{a}_2}^{\check{c}_{n_2}} \end{bmatrix}^T \in \underline{S}^{n_2},$$

which contains the transformation between the frames at the CoMs of each link of subsystem 2 and the frame at the connection point  $a_2$ , and

$$\underline{X}_{\bar{2},1} = \begin{bmatrix} \bar{X}_{\bar{2},1} & \mathbf{0}_{n_1-\eta} \end{bmatrix}^T \in \underline{S}^{n_1},$$

where

$$\bar{X}_{\bar{2},1} = \begin{bmatrix} \underline{x}_{\bar{b}_{1,2}}^0 & \underline{x}_{\bar{b}_{1,2}}^1 & \cdots & \underline{x}_{\bar{b}_{1,2}}^{k-1} \end{bmatrix} \in \underline{S}^k,$$

which contains the transformation between the frames at each joint of subsystem 1 and the frame at the connection point  $b_{1,2} = a_2$ , and  $\mathbf{0}_{n_1-k} \in \mathbb{R}^{n_1-k} \subset \mathcal{H}_p^{n_1-k}$  is a vector of zeros.

The next section presents a graph representation of the dynamic system assembling that facilitates the application of the proposed modular composition strategy.

## 5.2 Graph representation

Each subsystem in a branched robot can be considered as a vertex in a graph, in which directed, weighted edges represent the propagation of wrenches and twists. The advantage of such representation is that in addition to visually depicting the model composition, it provides the joint wrenches from the calculation of the graph interconnection matrix. For instance, the weighted graph in figure 5.3 represents the  $(n_1 + n_2)$ -DoF assembled branched mechanism shown in figure 5.1, where dashed edges correspond to the propagation of twists and solid edges represent the propagation of wrenches.

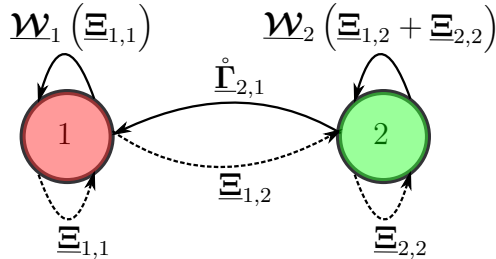


Figure 5.3: Graph representation of the  $(n_1 + n_2)$ -DoF assembled branched mechanism shown in figure 5.1. The  $n_1$ -DoF and the  $n_2$ -DoF kinematic chains are labeled as subsystems 1 and 2, respectively.

The interconnection matrix of the graph presented in figure 5.3 is given by

$$\underline{\mathbf{A}} = [\underline{\mathbf{A}}_{ij}] \triangleq \begin{bmatrix} \underline{\mathbf{W}}_1(\underline{\Xi}_{1,1}) & \mathring{\underline{\Gamma}}_{2,1} \\ \mathbf{0}_{n_2} & \underline{\mathbf{W}}_2(\underline{\Xi}_{1,2} + \underline{\Xi}_{2,2}) \end{bmatrix} \in \mathcal{H}_p^{(n_1+n_2) \times 2}. \quad (5.15)$$

The vectors  $\underline{\mathbf{A}}_{ij} \in \mathcal{H}_p^{n_i}$  of the partitioned matrix  $\underline{\mathbf{A}}$  indicates the propagation of wrenches from vertex  $j$  to vertex  $i$ , represented as a solid edge, in which  $n_i$  is the number of rigid bodies of the  $i$ -th subsystem. The interconnection matrix  $\underline{\mathbf{A}}$  is analogous to the adjacency matrix of the graph. However, instead of simply indicating the existence of a connection between the vertexes, the element  $\underline{\mathbf{A}}_{ij}$  gives the joint wrenches imposed by one subsystem onto the other. For instance, since subsystem 1 has wrenches generated at its joints as a result of its own motion, there is a self-loop in vertex 1 in the graph on figure 5.3 with a solid edge whose weight, as denoted in (5.15), is  $\underline{\mathbf{A}}_{1,1} = \underline{\mathbf{W}}_1(\underline{\Xi}_{1,1}) \in \mathcal{H}_p^{n_1}$ . Additionally, because there is no wrench propagation from subsystem 1 to 2, the corresponding weight is  $\underline{\mathbf{A}}_{2,1} = \mathbf{0}_{n_2} \in \mathbb{R}^{n_2} \subset \mathcal{H}_p^{n_2}$ , in which  $\mathbf{0}_{n_2}$  is a vector of zeros and there is no solid edge from 1 to 2 in the graph. Conversely, as there is wrench propagation from subsystem 2 to 1, there is a solid edge with weight  $\underline{\mathbf{A}}_{1,2} = \mathring{\underline{\Gamma}}_{2,1} \in \mathcal{H}_p^{n_1}$  in the graph. Finally, since subsystem 2 has wrenches generated at its joints by the combination of its own motion and the motion of subsystem 1, there is a self-loop in vertex 2 in the graph with a solid edge whose weight is  $\underline{\mathbf{A}}_{2,2} = \underline{\mathbf{W}}_2(\underline{\Xi}_{1,2} + \underline{\Xi}_{2,2}) \in \mathcal{H}_p^{n_2}$ .

More generally, the graph representation of the complete system composed of  $s$  kinematic chains is constructed as follows:

1. Create a vertex for each kinematic chain;
2. Add the edges, according to the following rules:
  - (a) Each vertex  $j$  has a dashed edge self-loop weighted by its own vector of twists  $\underline{\Xi}_{j,j}$ ;
  - (b) Except for the vertex representing the root subsystem of the branched kinematic chain, each vertex  $j$  has an incoming dashed edge from the vertex  $i$  representing its predecessor, “weighted” by the vector of twists  $\underline{\Xi}_{i,j} \in \mathcal{W}^{n_j}$ , and a solid edge self-loop weighted by its own vector of wrenches given by the  $\underline{\mathbf{W}}_j(\underline{\Xi}_{j,j} + \underline{\Xi}_{i,j})$  function;
    - i. If the  $j$ -th vertex represents the root subsystem of the branched kinematic chain, then the solid edge self-loop is weighted by  $\underline{\mathbf{W}}_j(\underline{\Xi}_{j,j})$ .
  - (c) Except for the vertex representing the root subsystem of the branched kinematic chain, each vertex  $j$  has an outgoing solid edge that goes to the vertex  $i$  representing its predecessor, “weighted” by the vector of wrenches  $\mathring{\underline{\Gamma}}_{j,i} \in \mathcal{W}^{n_i}$ .



The proposition below shows how the adjacency matrix fully encapsulates the model of the complete assembled system.

**Proposition 5.1.** *Let a branched kinematic system be composed of  $n$  rigid bodies divided into a set of  $s$  coupled subsystems, each one containing  $n_1, n_2, \dots, n_s$  rigid bodies, respectively. Considering the weighted graph representation proposed in section 5.2, the vector of joint wrenches  $\underline{\Gamma}_t$  of the complete system, given by (5.8), can be found as*

$$\underline{\Gamma}_t = \begin{bmatrix} \bar{\underline{\Gamma}}_1 \\ \bar{\underline{\Gamma}}_2 \\ \vdots \\ \bar{\underline{\Gamma}}_s \end{bmatrix} = \underline{\mathbf{A}} \mathbf{1}_s \in \mathcal{H}_p^n, \quad (5.16)$$

where  $\bar{\underline{\Gamma}}_i \in \mathcal{W}^{n_i}$  is the vector of the total joint wrenches of the  $i$ -th subsystem,  $\underline{\mathbf{A}} \in \mathcal{H}_p^{n \times s}$  is the interconnection matrix, and  $\mathbf{1}_s \in \mathbb{R}^s$  is a vector of ones.

*Proof.* Each element  $\underline{\mathbf{A}}_{ij} \in \mathcal{W}^{n_j}$  of the matrix  $\underline{\mathbf{A}}$  represents the edge from vertex  $j$  to vertex  $i$  in the interconnection graph and, therefore, the propagation of wrenches from subsystem  $j$  to  $i$ . Hence, each row of  $\underline{\mathbf{A}}$  contains all the wrenches acting upon the joints of the  $i$ -th subsystem. Therefore, the vector  $\bar{\underline{\Gamma}}_i$  of the total joint wrenches of the  $i$ -th subsystem is given by  $\bar{\underline{\Gamma}}_i = \sum_{j=1}^s \underline{\mathbf{A}}_{ij} = \sum_{j=1}^s \underline{\mathbf{A}}_{ij} \cdot \mathbf{1}$ . Thus,  $\underline{\Gamma}_{\text{total}} = \underline{\mathbf{A}} \mathbf{1}_s$ .  $\square$

The following examples illustrate the application of the weighted graph representation of a composed system to obtain its dynamic model through the proposition 5.1.

**Example 5.2.** Consider the  $(n_1 + n_2)$ -DoF branched mechanism shown in figure 5.1, whose graph representation is given by figure 5.3. The joint wrenches vector  $\underline{\Gamma}_t$  of the complete system is given by

$$\underline{\Gamma}_t = \underline{\mathbf{A}} \mathbf{1}_2 = \begin{bmatrix} \underline{\mathcal{W}}_1(\underline{\Xi}_{1,1}) & \dot{\underline{\Gamma}}_{2,1} \\ \mathbf{0}_{n_2} & \underline{\mathcal{W}}_2(\underline{\Xi}_{1,2} + \underline{\Xi}_{2,2}) \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \underline{\mathcal{W}}_1(\underline{\Xi}_{1,1}) + \dot{\underline{\Gamma}}_{2,1} \\ \underline{\mathcal{W}}_2(\underline{\Xi}_{1,2} + \underline{\Xi}_{2,2}) \end{bmatrix} \in \mathcal{H}_p^{n_1+n_2}, \quad (5.17)$$

where the interconnection matrix  $\underline{\mathbf{A}}$  is given by (5.15), which is the same result previously obtained in (5.12).

**Example 5.3.** Consider connection of a new  $n_3$ -DoF kinematic chain into the system described in example 5.2. The new subsystem, labeled as 3, branches from the  $n_2$ -DoF serial manipulator (i.e., the subsystem 2), as shown in figure 5.4.

The corresponding graph of the newly-assembled robot is shown in figure 5.5 and has its interconnection matrix given by

$$\underline{\mathbf{A}}' = \begin{bmatrix} \underline{\mathcal{W}}_1(\underline{\Xi}_{1,1}) & \dot{\underline{\Gamma}}_{2,1} & \mathbf{0}_{n_1} \\ \mathbf{0}_{n_2} & \underline{\mathcal{W}}_2(\underline{\Xi}_{1,2} + \underline{\Xi}_{2,2}) & \dot{\underline{\Gamma}}_{3,2} \\ \mathbf{0}_{n_3} & \mathbf{0}_{n_3} & \underline{\mathcal{W}}_2(\underline{\Xi}_{2,3} + \underline{\Xi}_{3,3}) \end{bmatrix} \in \mathcal{H}_p^{(n_1+n_2+n_3) \times 3},$$

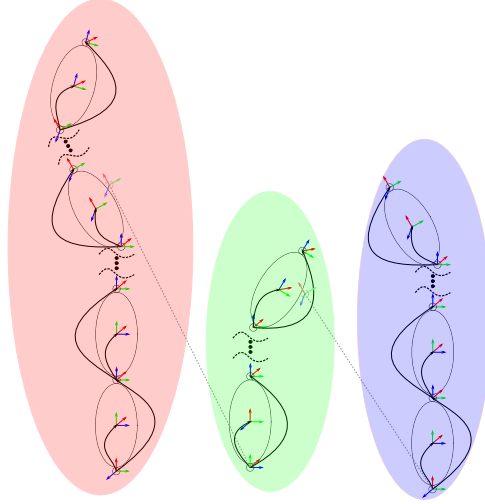


Figure 5.4: Three serial kinematic chains with  $n_1$ ,  $n_2$ , and  $n_3$  degrees of freedom.

where  $\mathbf{0}_{n_3} \in \mathbb{R}^{n_3} \subset \mathcal{H}_p^{n_3}$  is a vector of zeros.

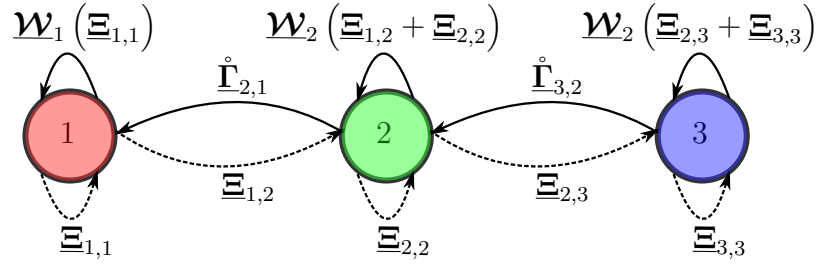


Figure 5.5: Graph representation of the new  $(n_1 + n_2 + n_3)$ -DoF assembled mechanism composed of three serial kinematic chains shown in figure 5.4.

Following the proposition 5.1, the dynamic model of the newly-assembled system is given by

$$\begin{aligned} \underline{\Gamma}_t = \underline{A}' \mathbf{1}_3 &= \begin{bmatrix} \underline{\mathcal{W}}_1(\underline{\Xi}_{1,1}) & \dot{\underline{\Gamma}}_{2,1} & \mathbf{0}_{n_1} \\ \mathbf{0}_{n_2} & \underline{\mathcal{W}}_2(\underline{\Xi}_{1,2} + \underline{\Xi}_{2,2}) & \dot{\underline{\Gamma}}_{3,2} \\ \mathbf{0}_{n_3} & \mathbf{0}_{n_3} & \underline{\mathcal{W}}_2(\underline{\Xi}_{2,3} + \underline{\Xi}_{3,3}) \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} \underline{\mathcal{W}}_1(\underline{\Xi}_{1,1}) + \dot{\underline{\Gamma}}_{2,1} \\ \underline{\mathcal{W}}_2(\underline{\Xi}_{1,2} + \underline{\Xi}_{2,2}) + \dot{\underline{\Gamma}}_{3,2} \\ \underline{\mathcal{W}}_2(\underline{\Xi}_{2,3} + \underline{\Xi}_{3,3}) \end{bmatrix} \in \mathcal{H}_p^{n_1+n_2+n_3}, \end{aligned} \quad (5.18)$$

where  $\underline{\Xi}_{2,3} \in \mathcal{H}_p^{n_3}$  and  $\dot{\underline{\Gamma}}_{3,2} \in \mathcal{H}_p^{n_2}$  are given by (5.10) and (5.11), respectively.

Notice how even though now connected into a more complex structure, the first two subsystems still have the same graph representation in figure 5.5 that they had before the addition of the third subsystem (see figure 5.3). Accordingly, notice how the expressions of (5.18) closely resemble those of (5.17) for the first and second subsystems. It is crucial to highlight, however, that the *numerical value* of  $\dot{\underline{\Gamma}}_{2,1}$  differs in examples 5.2 and 5.3, since in the later  $\dot{\underline{\Gamma}}_{2,1}$  carries the information of the wrenches propagating from subsystem 3,

which did not exist in the former example. Nonetheless, from the perspective of the first subsystem, the *numerical value* of  $\dot{\mathbf{T}}_{2,1}$  is not relevant, as it is but a vector of wrenches to be added to its nominal wrench values.

### 5.3 Simulations and discussions

Consider the 38-DoF branched robot shown in figure 5.6 and let us divide it into nine subsystems, as indicated by the colored ellipses. This simulation demonstrates the application of the dqNE to obtain the joint torques of this branched robot by means of the proposed modular composition and graph representation. Additionally, the monolithic solution implemented in the sv2NE is also presented for comparison. The simulation setup is the same described in section 4.2.1.5. The robot follows sinusoidal generalized velocities trajectories  $\dot{\mathbf{q}}_d \in \mathbb{R}^{38}$  using V-REP's standard low-level controllers, have its joint configurations  $\mathbf{q} \in \mathbb{R}^{38}$ , joint velocities  $\dot{\mathbf{q}} \in \mathbb{R}^{38}$ , and joint torques  $\boldsymbol{\tau} \in \mathbb{R}^{38}$  stored, and its generalized accelerations  $\ddot{\mathbf{q}} \in \mathbb{R}^{38}$  numerically estimated from  $\dot{\mathbf{q}}$ .

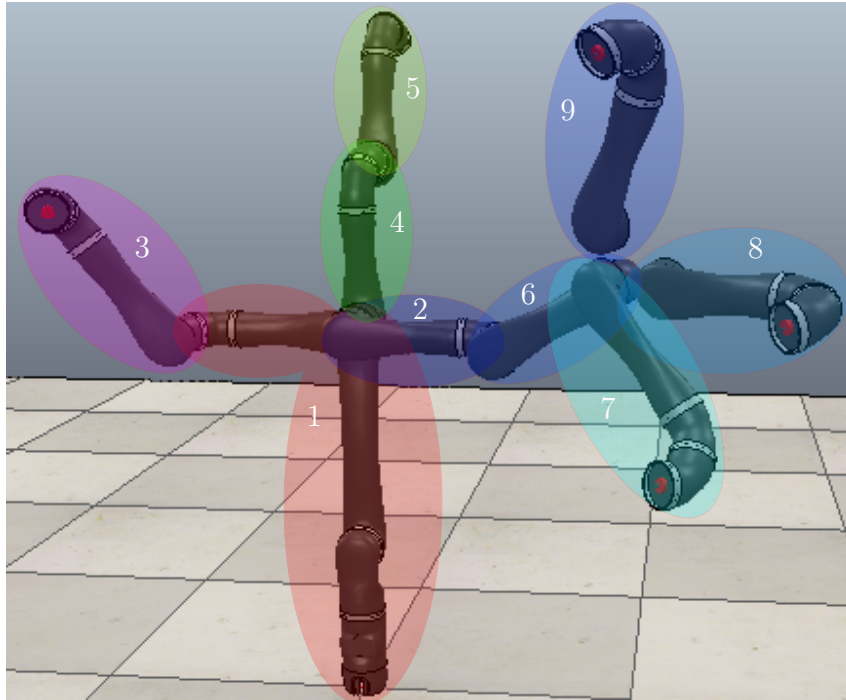


Figure 5.6: A 38-DoF branched robot in the robot simulator V-REP. Different colored ellipses highlight individual subsystems.

The weighted graph representing the 38-DoF branched robot is shown in figure 5.7, and its interconnection matrix  $\mathbf{A} \in \mathcal{H}_p^{38 \times 9}$  is given by

$$\underline{\mathbf{A}} = \begin{bmatrix} \underline{\mathbf{W}}_1 & \dot{\underline{\Gamma}}_{2,1} & \dot{\underline{\Gamma}}_{3,1} & \dot{\underline{\Gamma}}_{4,1} & \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 & \mathbf{0}_6 \\ \mathbf{0}_4 & \underline{\mathbf{W}}_2 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \dot{\underline{\Gamma}}_{6,2} & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 \\ \mathbf{0}_4 & \mathbf{0}_4 & \underline{\mathbf{W}}_3 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 \\ \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \underline{\mathbf{W}}_4 & \dot{\underline{\Gamma}}_{5,4} & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 \\ \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \underline{\mathbf{W}}_5 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 \\ \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \underline{\mathbf{W}}_6 & \dot{\underline{\Gamma}}_{7,6} & \dot{\underline{\Gamma}}_{8,6} & \dot{\underline{\Gamma}}_{9,6} \\ \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \underline{\mathbf{W}}_7 & \mathbf{0}_4 & \mathbf{0}_4 \\ \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \underline{\mathbf{W}}_8 & \mathbf{0}_4 \\ \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \underline{\mathbf{W}}_9 \end{bmatrix},$$

where  $\underline{\mathbf{W}}_i = \underline{\mathcal{W}}_i (\underline{\Xi}_{p_i,i} + \underline{\Xi}_{i,i})$ , in which  $i \in \{1, \dots, 9\}$ ,  $p_2 = p_3 = p_4 = 1$ ,  $p_5 = 4$ ,  $p_6 = 2$ , and  $p_7 = p_8 = p_9 = 6$ , and  $\mathbf{0}_6 \in \mathbb{R}^6 \subset \mathcal{H}_p^6$  and  $\mathbf{0}_4 \in \mathbb{R}^4 \subset \mathcal{H}_p^4$  are vectors of zeros.

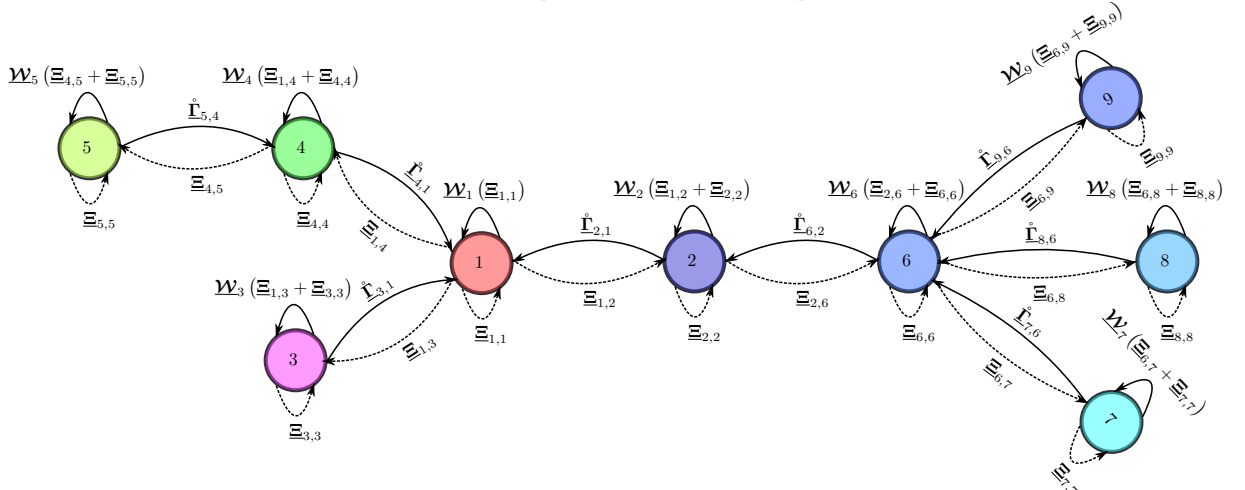


Figure 5.7: Graph representation of the 38-DoF branched robot. The colored nodes follow the color scheme adopted in figure 5.6.

The joint wrenches  $\underline{\Gamma}_{\text{dqNE}}$  are given by (5.16), that is,  $\underline{\Gamma}_{\text{dqNE}} = \underline{\mathbf{A}}\mathbf{1}_{38} \in \mathcal{H}_p^{38}$ , where  $\mathbf{1}_{38} \in \mathbb{R}^{38}$  is a vector of ones, and then projected onto the joints motion axes through (4.27), yielding  $\boldsymbol{\tau}_{\text{dqNE}} \in \mathbb{R}^{38}$ . Then, the comparison between the joint torque waveforms  $\boldsymbol{\tau}_{\text{dqNE}}$  and  $\boldsymbol{\tau}$ , filtered using a 6nd-order low-pass digital Butterworth filter with normalized cutoff frequency of 10Hz, is made considering the CMCs between them. Moreover, the joint torques  $\boldsymbol{\tau}_{\text{sv2NE}} \in \mathbb{R}^{38}$  are obtained from the sv2NE, filtered using the same low-pass Butterworth filter, and the CMCs between them and  $\boldsymbol{\tau}$  are also calculated.

Table 5.1 presents the CMC between the joint torque waveforms obtained through the different dynamic model strategies ( $\boldsymbol{\tau}_{\text{dqNE}}$  and  $\boldsymbol{\tau}_{\text{sv2NE}}$ ) and the values obtained from V-REP ( $\boldsymbol{\tau}$ ). The results demonstrate the accuracy of the proposed dqNE modular model composition and its similarity with the sv2NE. However, it is worth highlighting that the dqNE performs the modular dynamic modeling of the robot, whereas the sv2NE obtains the joint torques through a monolithic solution (i.e., without considering the existence of subsystems or performing modular composition). Furthermore, for qualitative analysis, figure 5.8 presents the joint torques obtained using dqNE and sv2NE, alongside the V-REP

values, for the minimum, maximum, and intermediate CMCs found during simulations. Even for the smallest value of CMC (i.e., 0.9850), the joint torques obtained using the proposed composition formulation match closely the V-REP values.

Table 5.1: CMC between the joint torque waveforms obtained through the different dynamic model strategies ( $\tau_{dqNE}$  and  $\tau_{sv2NE}$ ) and the values obtained from V-REP ( $\tau$ ) for the 38-DoF branched robot. The closer to one, the more similar the waveforms are.

Method	38-DOF branched robot			
	min	mean	std	max
dqNE	0.9850	0.9968	0.0037	1.0000
sv2NE	0.9850	0.9968	0.0037	1.0000

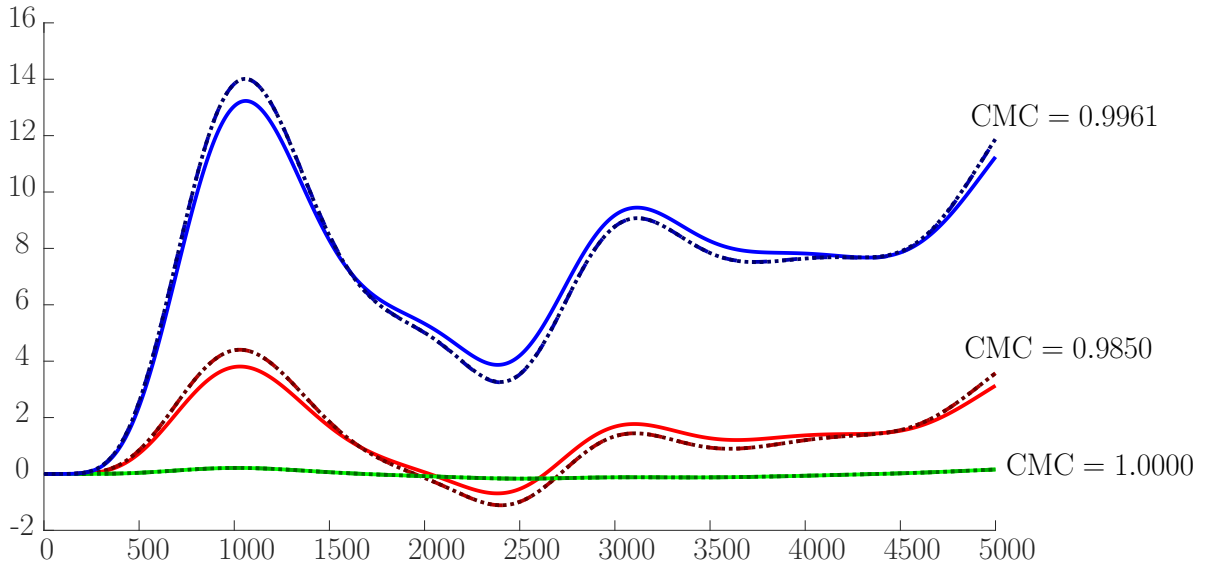


Figure 5.8: Joint torque waveforms of the 38-DoF branched robot. Solid curves correspond to the V-REP values, whereas dashed curves correspond to the values obtained using the dqNE and dot-dashed curves correspond to the values obtained using the sv2NE for the joint torque waveforms of the eleventh (CMC = 0.9850), sixth (CMC = 1.0000), and third (CMC = 0.9961) joints, respectively. Since both the model strategies (dqNE and rtNE) obtained almost identical results, their joint torque waveforms overlap in the graph.

## 5.4 Conclusions

This chapter proposed a modular model composition based on the dual quaternion Newton-Euler formalism. This strategy is applicable even if some subsystems are regarded as black boxes, requiring only the twists and wrenches at the connection points between different subsystems. Furthermore, the framework presented in section 5.1 is general, has a high level of abstraction, and thus can be instantiated into different representations, such as spatial algebra (Featherstone, 2008) or the Lie algebra  $se(3)$  (Murray et al., 1994), as long as the representations capture the high-level operations described in the section (*see*

equation (5.9)). Moreover, section 5.2 presented a unified graph representation of the system that, in addition to visually depicting the model composition, provides the joint wrenches from the calculation of the graph interconnection matrix.

# 6

## CONCLUSION AND FUTURE WORKS

This thesis has presented a study of robot dynamic modeling using dual quaternion formulations. A novel quaternionic inertia tensor has been proposed, allowing the description of the dynamic properties of rigid bodies exclusively with operations inside the dual quaternion algebra; thus, overcoming one of the deterrents on the applications of this algebra to dynamic modeling (Yang, 1967, 1971; Pennock & Yang, 1983; Shoham & Brodsky, 1993; Miranda de Farias et al., 2019a).

Inspired by previous developments in kinematic modeling (Adorno, 2011) and classic approaches based on three-dimensional vectors (Luh et al., 1980), the Newton-Euler iterative algorithm has been rewritten using dual quaternions. In this unified formulation, twists replace the decoupled representation of linear and angular velocities, whereas wrenches unify forces and torques into a single element.

Simulations have demonstrated the applicability of the dual quaternion Newton-Euler formalism to the dynamic modeling of branched robots and its accuracy when compared to state-of-the-art libraries (Featherstone, 2008; Corke, 2017). One of the main limitations of the proposed strategy lies in its computational cost, a problem that, nonetheless, was outside the scope of this thesis. An initial step to optimize the current implementation would be to consider that some operations involve quaternions instead of dual quaternions.<sup>1</sup>

The dynamic modeling of branched robots using modular composition (McPhee, 1998; MCPhee et al., 2004; Jain, 2012; Matarazzo Orsino & Hess-Coelho, 2015; Orsino, 2017; Hess-Coelho et al., 2021) has also been incorporated into the dual quaternion formulation.

---

<sup>1</sup>Initial analyses show that this would result in a cost reduction of over a half for some calculations.

A graph representation of the system further simplifies obtaining the joint wrenches of the assembled robot. Intermediate subsystems can be black boxes or have dynamic models obtained with different strategies or representations. Simulations have demonstrated equivalent accuracy of the proposed formalism to monolithic solutions (Featherstone, 2008), as expected.

## Future works

Thus far, only the expression of the twist for holonomic mobile bases has been deduced. Future works will need to formalize the expressions of a broader range of bases. Appendix A presents an initial study of how this could be done considering the nonholonomic constraints directly during the deduction of the twists for differential-drive mobile bases. This process of modeling new mobile bases is equivalent to including new joint types in table 4.1. Nonetheless, and more interestingly, future works should address the inclusion of nonholonomic constraints into the dynamic models directly as dual quaternion algebra elements.

Insofar as this thesis has presented the dynamic modeling of branched robots using dual quaternion algebra, hence enabling the direct integration between the low-level dynamic model and higher-level algorithms based on dual quaternions, no such applications have been demonstrated. Future works are required to address the development of control strategies that will use the proposed dynamic models. An early study in this regard is presented in appendix B, which consists of a wrench control strategy that takes advantage of the recurrence equations of the dual quaternion Newton-Euler formalism. Furthermore, a simple application for end-effector position control based on potential fields (Khatib, 1986) illustrates the method.

Despite exploring the Newton-Euler formalism using dual quaternion algebra, this thesis has not ventured into other dynamic modeling strategies. Going in this direction, appendix C presents some initial thoughts on the dual quaternion Euler-Lagrange equation, which requires more development.

Finally, the formalism presented in this thesis is applicable to branched robots composed of rigid links. That is, rigid body mechanisms that form an open kinematic tree. Future works are required to extend the dual quaternion Newton-Euler and the modular composition strategy to robots with kinematic loops (e.g., parallel robots) and soft robotics (i.e., robots with flexible links).



# APPENDICES

# A

## DIFFERENTIAL-DRIVE MOBILE BASES

Consider a differential-drive mobile base, such as, for instance, the one of the mobile manipulator shown in figure A.1, whose coordinates are given by  $(x, y, \phi)$ , where  $\phi \in [0, 2\pi)$  is its angle of rotation around the vertical axis. The goal is to find the CoM twist of this differential-drive mobile base, which will then be considered in  $\mathcal{F}_1(\bar{\mathcal{E}}_{1,1})$  for (4.33).



Figure A.1: Little John, a nonholonomic mobile manipulator (courtesy of MACRO Research Group).

To that end, consider that the mobile base's CoM reference frame is given by the frame  $\mathcal{F}_d$ , which is located at the intersection of the wheel axis and the axis of symmetry, and displaced by  $d \in \mathbb{R}$  along the  $x$ -axis of  $\mathcal{F}_r$ , as shown in figure A.2.

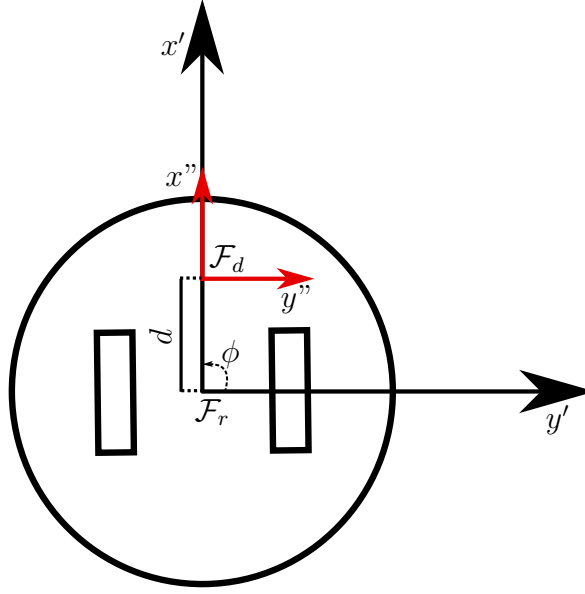


Figure A.2: Representation of a differential-drive mobile base.

The pose of the nonholonomic mobile base with respect to the inertial frame is given by

$$\underline{\mathbf{x}}_r^0 = \mathbf{r}_r^0 + \varepsilon \frac{1}{2} \mathbf{p}_{0,r}^0 \mathbf{r}_r^0,$$

where  $\mathbf{r}_r^0 = \cos(\phi/2) \hat{i} + \hat{k} \sin(\phi/2)$  and  $\mathbf{p}_{0,r}^0 = x\hat{i} + y\hat{j}$ . Furthermore,

$$\underline{\mathbf{x}}_d^r = 1 + \varepsilon \frac{1}{2} \mathbf{p}_{r,d}^d,$$

with  $\mathbf{p}_{r,d}^d = d\hat{i}$ . Thus,

$$\underline{\mathbf{x}}_d^0 = \underline{\mathbf{x}}_r^0 \underline{\mathbf{x}}_d^r, \quad (\text{A.1})$$

where

$$\mathbf{r}_d^0 = \cos(\phi/2) \hat{i} + \hat{k} \sin(\phi/2) \quad (\text{A.2})$$

and

$$\mathbf{p}_{0,d}^0 = (x + d \cos(\phi)) \hat{i} + (y + d \sin(\phi)) \hat{j}. \quad (\text{A.3})$$

The time derivative of (A.2) is given by

$$\dot{\mathbf{r}}_d^0 = -\frac{\dot{\phi}}{2} \sin(\phi/2) \hat{i} + \hat{k} \frac{\dot{\phi}}{2} \cos(\phi/2). \quad (\text{A.4})$$

Analyzing (A.2) and (A.4), and considering the fact that

$$\dot{\mathbf{r}}_d^0 = \frac{1}{2} \boldsymbol{\omega}_{0,d}^0 \mathbf{r}_d^0,$$

the angular velocity of the mobile base, with respect to  $\mathcal{F}_0$ , is given by

$$\boldsymbol{\omega}_{0,d}^0 = \dot{\phi} \hat{k}. \quad (\text{A.5})$$

Additionally, the time derivative of (A.3) is given by

$$\dot{\boldsymbol{p}}_{0,d}^0 = (\dot{x} - d\dot{\phi} \sin(\phi)) \hat{i} + (\dot{y} + d\dot{\phi} \cos(\phi)) \hat{j}. \quad (\text{A.6})$$

From (A.5), (A.6) and (A.3), the twist  $\underline{\boldsymbol{\xi}}_{0,d}^0$  is given by

$$\underline{\boldsymbol{\xi}}_{0,d}^0 = \boldsymbol{\omega}_{0,d}^0 + \varepsilon (\dot{\boldsymbol{p}}_{0,d}^0 + \boldsymbol{p}_{0,d}^0 \times \boldsymbol{\omega}_{0,d}^0). \quad (\text{A.7})$$

However, this twist is given with respect to the inertial frame, whereas the robot's inertia tensor is usually given with respect to a reference frame attached to its body. Thus, the twist given by (A.7) needs to be changed from  $\mathcal{F}_0$  to  $\mathcal{F}_d$ . Considering (3.11), this is given by

$$\begin{aligned} \underline{\boldsymbol{\xi}}_{0,d}^d &= \text{Ad}(\boldsymbol{x}_0^d) \underline{\boldsymbol{\xi}}_{0,d}^0 \\ &= \boldsymbol{\omega}_{0,d}^d + \varepsilon (\dot{\boldsymbol{p}}_{0,d}^d). \end{aligned} \quad (\text{A.8})$$

Furthermore, the time derivative  $\underline{\dot{\boldsymbol{\xi}}}_{0,d}^d$  is given by (3.17), yielding

$$\begin{aligned} \underline{\dot{\boldsymbol{\xi}}}_{0,d}^d &= \text{Ad}(\boldsymbol{x}_0^d) \underline{\dot{\boldsymbol{\xi}}}_{0,d}^0 + \underline{\boldsymbol{\xi}}_{d,0}^d \times (\text{Ad}(\boldsymbol{x}_0^d) \underline{\boldsymbol{\xi}}_{0,d}^0) \\ &= \boldsymbol{\omega}_{0,d}^d + \varepsilon (\ddot{\boldsymbol{p}}_{0,d}^d + \dot{\boldsymbol{p}}_{0,d}^d \times \boldsymbol{\omega}_{0,d}^d), \end{aligned}$$

since  $\underline{\boldsymbol{\xi}}_{d,0}^d \times (\text{Ad}(\boldsymbol{x}_0^d) \underline{\boldsymbol{\xi}}_{0,d}^0) = -\underline{\boldsymbol{\xi}}_{0,d}^0 \times \underline{\boldsymbol{\xi}}_{0,d}^0 = 0$ .

The dynamical model of the differential-drive mobile manipulator is then given by (4.33) and (4.34), in which  $\underline{\mathcal{F}}_1(\underline{\boldsymbol{\Xi}}_{1,1})$  is given by (A.8).

## A.1 Simulations and discussions

The simulation environment is the same described in section 4.2.1.5. The dqNE is applied to obtain the generalized forces of the differential-drive mobile robot Pioneer shown in figure A.3, considering the twist expressions (4.33) and (4.34). The robot is actuated with arbitrary wheel torques  $\boldsymbol{\tau}_w = [\tau_l \ \tau_r]^T \in \mathbb{R}^2$ , in which  $\tau_l$  and  $\tau_r$  are respectively the left and right wheel torques, have its generalized configurations  $\boldsymbol{q} \in \mathbb{R}^3$  and velocities  $\dot{\boldsymbol{q}} \in \mathbb{R}^3$  stored, and its generalized accelerations  $\ddot{\boldsymbol{q}} \in \mathbb{R}^3$  numerically estimated from  $\dot{\boldsymbol{q}}$ .

The generalized forces  $\boldsymbol{\tau}_{\text{dqNE}} \in \mathbb{R}^3$  are obtained from the dqNE after projecting the mobile base wrench onto the motion axes through (4.27) (see (4.37)). Moreover, the generalized forces  $\boldsymbol{\tau}_a \in \mathbb{R}^3$  are calculated from the analytical solution given by (Fierro &



Figure A.3: Differential-drive mobile robot Pioneer in the robot simulator V-REP.

(Lewis, 1997)

$$\mathbf{M}(\mathbf{q}) \mathbf{q}_d + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau}_a,$$

where

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} m & 0 & md \sin(\phi) \\ 0 & m & -md \cos(\phi) \\ md \sin(\phi) & -md \cos(\phi) & i \end{bmatrix}$$

is the inertia matrix, in which  $m$  is the mass of the robot,  $\phi$  its rotation angle,  $i = i_{33} + md^2$ , where  $i_{33}$  is the element of the third row and third column of the inertia tensor of the robot, and  $d \in \mathbb{R}$  is the displacement along the  $x$ -axis of  $\mathcal{F}_r$  (see figure A.2); and

$$\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} md\dot{\phi}^2 \cos(\phi) \\ md\dot{\phi}^2 \sin(\phi) \\ 0 \end{bmatrix}$$

is the Coriolis and centripetal forces vector.<sup>1</sup> Finally, both  $\boldsymbol{\tau}_{\text{dqNE}}$  and  $\boldsymbol{\tau}_a$  are filtered, using a 6nd-order low-pass digital Butterworth filter with normalized cutoff frequency of 10Hz, and the CMCs between them are calculated and presented in table A.1. The results indicate that both models are similar, which is qualitatively shown in figure A.4.

Table A.1: CMC between the generalized force waveforms  $\boldsymbol{\tau}_{\text{dqNE}}$  and  $\boldsymbol{\tau}_a$  of the differential-drive mobile robot Pioneer. The closer to one, the more similar the waveforms are.

Method	CMC for the differential-drive mobile robot Pioneer			
	min	mean	std	max
dqNE	1.0000	1.0000	$1.6385 \times 10^{-05}$	1.00000

<sup>1</sup>The simulation considers  $d = 0.001$  and reads from V-REP the values of  $m = 16\text{Kg}$ ,  $i_{33} = 2.0906$ , and  $\phi = \phi(t)$ , with  $\phi(t)$  given by the simulation trajectory, which is generated by the Pioneer being actuated with constant wheel torques of  $\tau_l = 5\text{Nm}$  and  $\tau_r = 8\text{Nm}$  for 2000 iterations.

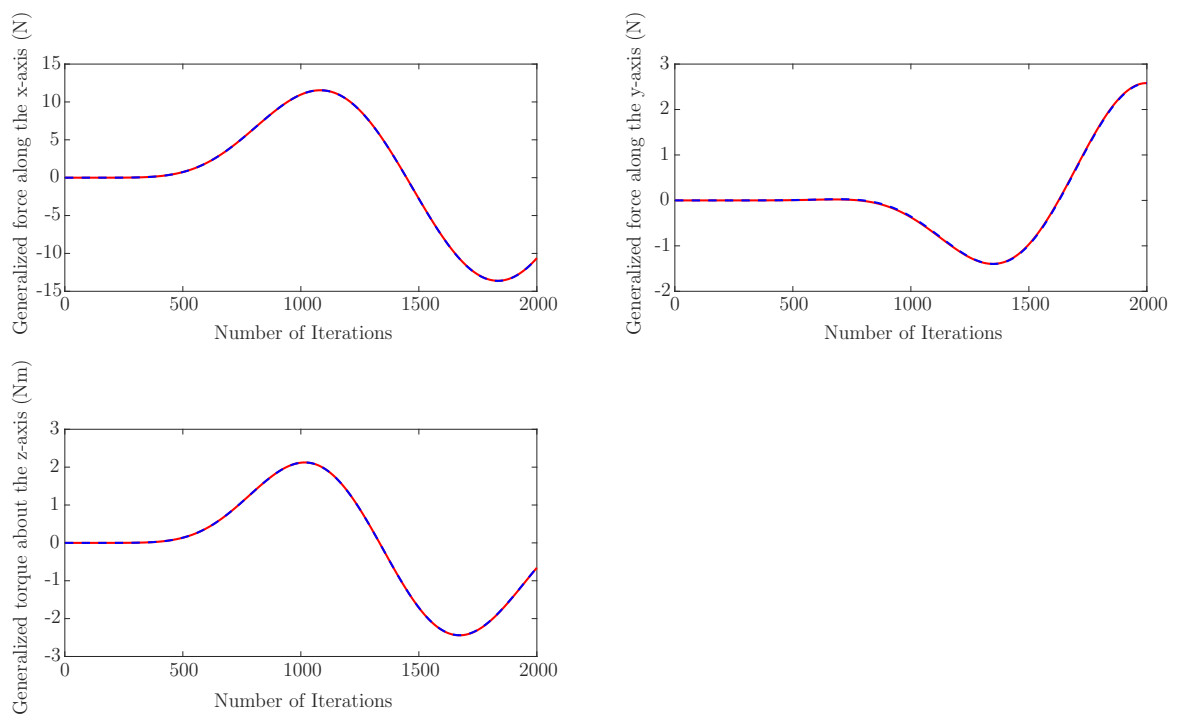


Figure A.4: Generalized force waveforms for the differential-drive mobile robot Pioneer. Solid curves correspond to the analytical solution  $\tau_a$ , whereas dashed curves correspond to the generalized force waveforms  $\tau_{dqNE}$ .

# B

## DYNAMIC CONTROL

Although the objective of this thesis is to propose dynamic models for robotic systems based on dual quaternion algebra, as discussed in chapter 1, one of the motivations for it is to find suitable models that allow the development of dual quaternion control strategies without the need for intermediate mappings, which usually introduce mathematical artifacts, such as algorithmic singularities and discontinuities. In this regard, section B.1 presents a novel wrench control architecture based on one of the main contributions of this thesis, the dual quaternion Newton-Euler formalism presented in chapter 4. Therefore, to give the proposed controller a proper context, this section presents the state of the art in force control strategies based on the recursive Newton-Euler algorithm.

The recursive Newton-Euler algorithm (Luh et al., 1979) is well known and widely used by roboticists as a highly efficient tool for robot modeling (Spong et al., 2006; Siciliano et al., 2009; Felis, 2017). In control applications, however, the usage of the Euler-Lagrange model is far more common (Murray et al., 1994; Siciliano & Khatib, 2008). The Newton-Euler algorithm, if used, is applied either to obtain the canonical Euler-Lagrange equations (Ploen, 1999; De Luca & Ferrajoli, 2009) or as an intermediate step to find the desired joint torques given the desired joint positions, velocities, and accelerations (Lee & Chung, 1984). Some researches, nonetheless, have been trying to explore the recursive nature of the Newton-Euler formulation directly in the control scheme.

Going in that direction, Walker (1988) uses Featherstone's spatial algebra (Featherstone, 1983) to model a robot that may contain closed kinematic loops and proposes an adaptive control strategy that estimates mass-related parameters and viscous friction coefficients

of the robot. First, the author uses the desired position, velocity, and acceleration of the independent joints to calculate the corresponding values of the unconstrained joints. Then, he uses those values on the forward recursion of the Newton-Euler algorithm so that the backward recursion equations, which include the equations to update the parameter estimates, yield the control input joint torques.

Walker's (Walker, 1988) strategy is further explored by Walker et al. (1989) to propose an adaptive compliant control to solve the problem of two robotic manipulators carrying an object of unknown mass while following the desired position and internal force trajectory. Jone Hann Jean & Li-Chen Fu (1993) also apply that structure to design an adaptive hybrid controller for constrained robots. Their control scheme can track both joint positions and constrained forces. To deal with the latter, however, they characterize them in terms of a multiplier function of generalized joints forces. The authors then insert those values into the backward recursion equations. More recently, Hanlei (2010) has proposed a similar strategy to perform a composite adaptation, where, as Jone Hann Jean & Li-Chen Fu (1993), he uses on the backward recursion forces given on each joint reference frame.

The wrench control strategy presented in section B.1 explores the recursive aspect of the dual quaternion Newton-Euler formulation to enable reasoning directly at the task level, hence simplifying the control problem. Moreover, the proposed control formulation allows the high-level definition of desired wrenches at arbitrary points of the robot. Furthermore, the dual quaternion algebra ensures the coupled treatment of forces and torques, whose desirability was previously discussed in this chapter.

## B.1 Wrench control

The dual quaternion Newton-Euler formalism can be seen as the function  $\mathcal{N} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathcal{H}_p^n$  given by

$$\underline{\Gamma} = \mathcal{N}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \in \mathcal{H}_p^n, \quad (\text{B.1})$$

where  $n$  is the number of rigid bodies in the kinematic chain,  $\underline{\Gamma} \in \mathcal{H}_p^n$  is the vector of wrenches acting on the robot joints,  $\mathbf{q} \in \mathbb{R}^n$  is the vector of joint configurations, and  $\dot{\mathbf{q}} \in \mathbb{R}^n$  and  $\ddot{\mathbf{q}} \in \mathbb{R}^n$  are its first and second-order time derivatives, respectively.

One direct application of this method in control is to use it to find the gravity compensation torques for a  $n$ -DOF robot. To do so, one has to set the joint velocities  $\dot{\mathbf{q}} \in \mathbb{R}^n$  and the joint accelerations  $\ddot{\mathbf{q}} \in \mathbb{R}^n$  as vectors of zeros  $\mathbf{0}_n \in \mathbb{R}^n$ , such that

$$\underline{\Gamma}_g = \mathcal{N}(\mathbf{q}, \mathbf{0}_n, \mathbf{0}_n) \in \mathcal{H}_p^n, \quad (\text{B.2})$$

where  $\underline{\Gamma}_g \in \mathcal{H}_p^n$  is the vector of joint wrenches. That is, the joint wrenches  $\underline{\Gamma}_g$  will keep the robot on its current configuration  $\mathbf{q}$  and with joint accelerations equal to  $\mathbf{0}_n$ ; thus, compensating the gravity torques.



Let us consider a more general application of the dual quaternion Newton-Euler formalism in control. The wrenches returned by (B.1) are given by the recurrence equation (4.26). For the  $n$ -th joint, it is considered that there is no wrench acting on the end-effector of the robot (i.e.,  $\underline{\zeta}_n^n = 0$ ) and the method works as previously described. Let us now assume a desired end-effector wrench  $\underline{\zeta}_d^n \in \mathcal{H}_p$  and modify (4.26) to consider  $\underline{\zeta}_n^n = \underline{\zeta}_d^n$ . To simplify further explanations, consider the function  $\bar{\mathcal{N}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \underline{\zeta}_d^n) : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \times \mathcal{H}_p \rightarrow \mathcal{H}_p^n$ , such that

$$\underline{\Gamma} = \bar{\mathcal{N}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \underline{\zeta}_d^n).$$

That is, revisiting (B.2),  $\underline{\Gamma}_g = \mathcal{N}(\mathbf{q}, \mathbf{0}_n, \mathbf{0}_n) = \bar{\mathcal{N}}(\mathbf{q}, \mathbf{0}_n, \mathbf{0}_n, 0)$ .

Let us now assume, however, that we want to impose an arbitrary desired end-effector wrench  $\underline{\zeta}_d^n \in \mathcal{H}_p$ . In this scenario, consider the joint wrenches

$$\underline{\Gamma}_u = \bar{\mathcal{N}}(\mathbf{q}, \mathbf{0}_n, \mathbf{0}_n, \underline{\zeta}_d^n) = \underline{\Gamma}_g + \underline{\Gamma}_{\underline{\zeta}_d^n}, \quad (\text{B.3})$$

where  $\underline{\Gamma}_{\underline{\zeta}_d^n} \in \mathcal{H}_p^n$  is the vector of wrenches generated by the backward propagation of the end-effector wrench  $\underline{\zeta}_d^n$  throughout the kinematic chain. In other words, the vector  $\underline{\Gamma}_{\underline{\zeta}_d^n}$  is formed by the projection of  $\underline{\zeta}_d^n$  into all the robot joints. Naturally, once in the presence of the gravity acceleration, and disregarding numerical and modeling errors, the wrenches  $\underline{\Gamma}_g$  will cancel themselves out with the wrenches caused by the gravity. Thus, resulting in  $\underline{\Gamma}_u = \underline{\Gamma}_{\underline{\zeta}_d^n}$ , which, by construction, are the joint wrenches that impose any arbitrarily desired end-effector wrench  $\underline{\zeta}_d^n$ .

It is worth highlighting that the definition of the desired wrench on the end-effector was arbitrary and made to simplify the explanation of the method. The proposed strategy may be used to impose a desired wrench at any arbitrary point of the robot (e.g., arms or legs for a humanoid climbing on a surface).

### B.1.1 End-effector position control based on potential fields

Consider an attractive force given by (Khatib, 1986)

$$\mathbf{f}_a^n = -k_p(\mathbf{p}^n - \mathbf{p}_d^n) - k_v\dot{\mathbf{p}}^n, \quad (\text{B.4})$$

where  $k_p \in \mathbb{R}$  and  $k_v \in \mathbb{R}$  are positive real gains and  $\mathbf{p}_d^n, \mathbf{p}^n \in \mathbb{H}_p$  are, respectively, the desired and the current end-effector positions expressed in the end-effector frame. This force is then set as the desired end-effector wrench as  $\underline{\zeta}_d^n = \mathbf{f}_a^n$  and the control input is given by

$$\underline{\Gamma}_u = \underbrace{\bar{\mathcal{N}}(\mathbf{q}, \mathbf{0}_n, \mathbf{0}_n, \underline{\zeta}_d^n)}_{\underline{\Gamma}_{\underline{\zeta}_d^n}} + \underline{\Gamma}_{\text{diss}}, \quad (\text{B.5})$$

with  $\underline{\mathbf{\Gamma}}_{\text{diss}} = -k_{\text{diss}} \left[ \varepsilon(\dot{q}_1 \hat{k}) \ \cdots \ \varepsilon(\dot{q}_n \hat{k}) \right]^T \in \mathcal{H}_p^n$ , in which  $k_{\text{diss}} \in \mathbb{R}$  is a positive real gain and  $\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_1 & \cdots & \dot{q}_n \end{bmatrix} \in \mathbb{R}^n$  is the vector of joint velocities.

The vector  $\underline{\mathbf{\Gamma}}_{\zeta_d^n}$  is, by construction, the vector of wrenches that imposes the desired end-effector wrench  $\zeta_d^n = -k_p(\mathbf{p}^n - \mathbf{p}_d^n) - k_v \dot{\mathbf{p}}^n$ . Furthermore,  $\underline{\mathbf{\Gamma}}_{\text{diss}}$  is a vector of dissipative joint wrenches that is necessary since the robot is redundant to the task of position regulation (i.e., it has more than 3-DOF), which means that even after end-effector convergence it is still possible to have non-zero joint velocities. In other words, if we had only  $\underline{\mathbf{\Gamma}}_u = \underline{\mathbf{\Gamma}}_{\zeta_d^n}$ , the robot would be free to present joint motions at the null space of the solution. As this behavior is undesirable in most situations, the term  $\underline{\mathbf{\Gamma}}_{\text{diss}}$  ensures that the robot remains still at the desired end-effector position.

The formalization of the proposed control strategy is still an open problem. For instance, the proof that closed-loop system composed of (B.1) under (B.5) is stable for the end-effector position regulation is yet to be presented. Furthermore, equation (B.5) considers the application of the input wrench  $\mathbf{\Gamma}_u$  on the actuators of the robot, whereas in reality the joint inputs are given by the projected values of the joint wrenches onto their motion axes through (4.27).

Nonetheless, the next section presents a simulation demonstrating the application of (B.5) on the the 6-DOF JACO robotic arm.

### B.1.2 Simulations and discussions

The proposed control strategy is used to perform the end-effector control of the 6-DOF JACO robotic arm shown in figure 4.3. The initial and desired end-effector position are  $\mathbf{p} = 0.2828\hat{i} + 0.1177\hat{j} + 0.6855\hat{k}$  and  $\mathbf{p}_d = -0.4619\hat{i} - 0.0675\hat{j} + 0.6277\hat{k}$ , respectively. The controller gains are  $k_p = 30$ ,  $k_v = 15$ , and  $k_{\text{diss}} = 0.1$ . Considering those values, the joint wrenches  $\underline{\mathbf{\Gamma}}_u$  from (B.5) are projected onto the joints motion axes through (4.27) and sent to V-REP as input torques. The simulation runs in synchronous mode and stops after the norm of the time derivative of the error remains lower than  $10^{-5}$  for 2000 consecutive iterations.

Fig B.1 presents the norm of the error of the end-effector position, whereas Fig B.2 presents the input joint torques. Although the system converges to the desired end-effector position, there is a steady-state error that, nonetheless, could be removed with the addition of an integral term on the control law (B.5). Moreover, the average execution time of the controller is 40.87 ms. This value is expected to decrease to around 1.02 ms in a C++ implementation (Adorno & Marques Marinho, 2021), thus allowing application on a real JACO.

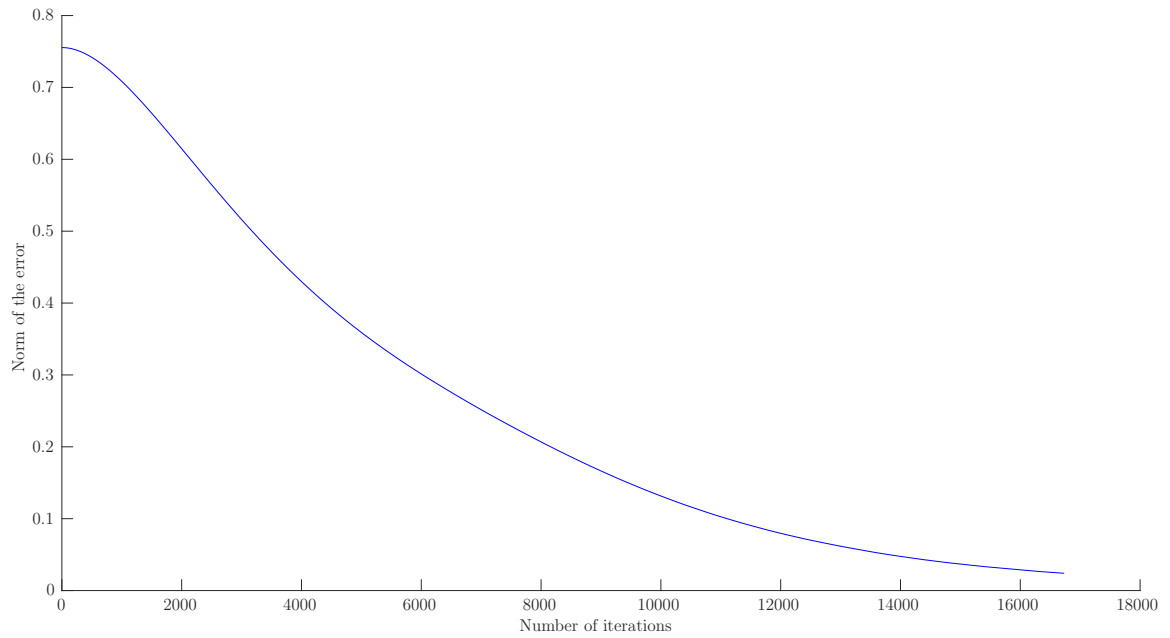


Figure B.1: Norm of the end-effector position error of the 6-DOF JACO robotic arm.

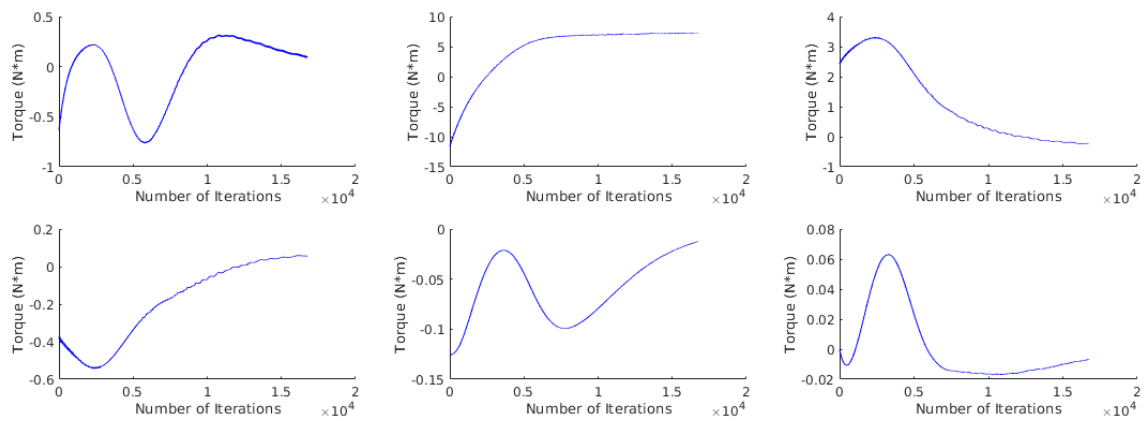


Figure B.2: Input joint torques  $\underline{\mathbf{T}}_u$  of the 6-DOF JACO robotic arm.

# C

## DUAL QUATERNION EULER-LAGRANGE FORMULATION

Equation (3.1) presents the canonical form of the classic Euler-Lagrange model. Accordingly, a similar dual quaternion Euler-Lagrange (dqEL) equation is given by

$$\underline{\mathbf{M}}(\mathbf{q}) \ddot{\mathbf{q}} + \underline{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) + \underline{\mathbf{G}}(\mathbf{q}) = \underline{\mathbf{\Gamma}}, \quad (\text{C.1})$$

where  $\underline{\mathbf{\Gamma}} \in \mathcal{H}_p^n$  is the vector of joint wrenches,  $n$  is the number of rigid bodies in the kinematic chain, and  $\underline{\mathbf{M}} \in \mathcal{H}_p^n \times \mathcal{H}_p^n$ ,  $\underline{\mathbf{C}} \in \mathcal{H}_p^n$ , and  $\underline{\mathbf{G}} \in \mathcal{H}_p^n$  are given by

$$\begin{aligned} \underline{\mathbf{G}}(\mathbf{q}) &= \mathcal{N}(\mathbf{q}, \mathbf{0}, \mathbf{0}), \\ \underline{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) &= \mathcal{N}_0(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{0}) - \underline{\mathbf{G}}(\mathbf{q}), \\ \text{col}_i(\underline{\mathbf{M}}(\mathbf{q})) &= \mathcal{N}_0(\mathbf{q}, \mathbf{0}, \text{col}_i(\mathbf{I})) - \underline{\mathbf{G}}(\mathbf{q}), \end{aligned} \quad (\text{C.2})$$

where  $\mathcal{N}$  is given by (B.1),  $\text{col}_i(\cdot)$  extracts the  $i$ -th column of a given matrix,  $\mathbf{I} \in \mathbb{R}^{n \times n}$  is the identity matrix,  $\dot{\mathbf{q}} \in \mathbb{R}^n$  is the first time derivative of the vector of joint configurations  $\mathbf{q} \in \mathbb{R}^n$ , and  $\mathbf{0} \in \mathbb{R}^n$  is a vector of zeros.

## C.1 Simulations and discussions

This simulation validates the dual quaternion Euler-Lagrange equation on the 7-DOF KUKA LWR robotic arm. A set of 1000 random configurations ( $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$ ) are generated and the left side of (C.1) is calculated considering the matrices given by (C.2), resulting on the joint wrenches  $\underline{\mathbf{\Gamma}}_{\text{dqEL}}$ . Furthermore, the dual quaternion Newton-Euler formalism is used to obtain the joint wrenches  $\underline{\mathbf{\Gamma}}_{\text{dqNE}}$ . Then, the wrench error  $\underline{\mathbf{\Gamma}}_E = \underline{\mathbf{\Gamma}}_{\text{dqNE}} - \underline{\mathbf{\Gamma}}_{\text{dqEL}}$  is calculated and have its Euclidean norm  $\tau_e = \text{norm}(\text{vec}_8(\underline{\mathbf{\Gamma}}_E))$  stored.

Table C.1 presents the mean (mean( $\tau_e$ )) and the variance (var( $\tau_e$ )) of the error  $\tau_e$  for each joint of the 7-DOF KUKA LWR robotic arm, indicating a high similarity between both modeling formalisms.

Table C.1: The mean and the variance of the error  $\tau_e$  comparing the joint wrenches  $\underline{\mathbf{\Gamma}}_{\text{dqEL}}$  generated by the dqEL and the joint wrenches  $\underline{\mathbf{\Gamma}}_{\text{dqNE}}$  obtained from dqNE for a set of 1000 randomly generated configurations ( $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ , and  $\ddot{\mathbf{q}}$ ) for the 7-DOF KUKA LWR robotic arm.

Joint number	mean ( $\tau_e$ )	var ( $\tau_e$ )
1	$6.605303 \times 10^{-14}$	$1.741932 \times 10^{-27}$
2	$5.491231 \times 10^{-14}$	$1.234560 \times 10^{-27}$
3	$4.508591 \times 10^{-14}$	$8.430729 \times 10^{-28}$
4	$3.272564 \times 10^{-14}$	$4.812145 \times 10^{-28}$
5	$2.405790 \times 10^{-14}$	$2.775287 \times 10^{-28}$
6	$1.235095 \times 10^{-14}$	$7.839918 \times 10^{-29}$
6	$1.987228 \times 10^{-15}$	$2.114408 \times 10^{-30}$

A potential application of the dual quaternion Euler-Lagrange to control would be to use it to find the joint accelerations of the robot, which are calculated as

$$\ddot{\mathbf{q}} = (\underline{\mathbf{M}}(\mathbf{q}))^{-1} (\underline{\mathbf{\Gamma}} - \underline{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) - \underline{\mathbf{G}}(\mathbf{q})). \quad (\text{C.3})$$

However, there are a few questions here that are yet to be answered. First, the inversion of dual quaternion matrices is not properly defined; thus, the solution of  $(\underline{\mathbf{M}}(\mathbf{q}))^{-1}$  is unclear. Second, even if such inversion is obtainable, all elements on the right side of (C.3) are pure dual quaternions, which leads to  $\ddot{\mathbf{q}} \in \mathcal{H}^n$ .<sup>1</sup> Obtaining physically meaningful real values for  $\ddot{\mathbf{q}}$  is still an open problem.

---

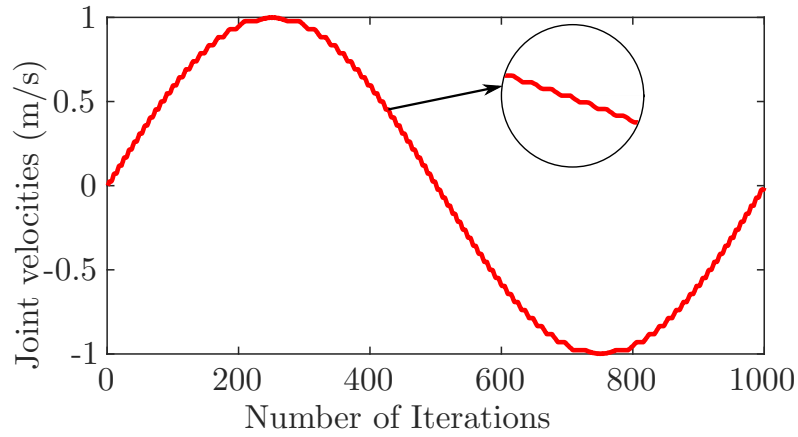
<sup>1</sup>It is straightforward to demonstrate that the multiplication of elements of  $\mathcal{H}_p$  results in elements of  $\mathcal{H}$ , which directly extends to  $\mathcal{H}_p^n$  and  $\mathcal{H}^n$ .

# D

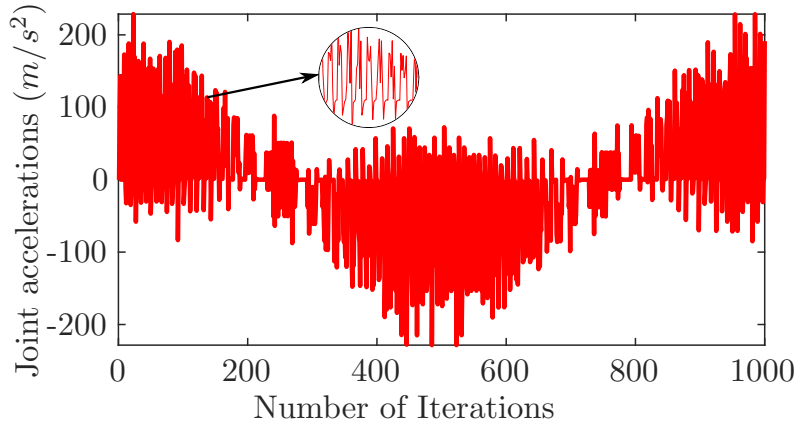
## SIMULATION DETAILS

For the simulations presented in this thesis, the joint velocities  $\dot{\mathbf{q}}$  are obtained from the V-REP and used to calculate the joint accelerations  $\ddot{\mathbf{q}}$  by means of numerical differentiation based on Richardson extrapolation (Gilat & Subramaniam, 2014, p. 322). To illustrate the process, figure D.1b shows the joint velocity of the third joint of the 6-DoF JACO robotic arm (*see* figure 4.3) obtained from the V-REP from a simulation where the robot follows sinusoidal joint velocity trajectories  $\dot{\mathbf{q}}_d \in \mathbb{R}^6$  using V-REP’s standard low-level controllers. When numerically differentiated, this joint velocity has its noises amplified, thus resulting in the noisier joint acceleration presented in figure D.1b. Naturally, the joint torques obtained from a dynamic modeling strategy using those noise joint velocities and accelerations are equally noise, such as the unfiltered joint torque of the third joint of the 6-DoF JACO robotic arm shown in figure D.1c, which was arbitrarily obtained from the dqNE. For this reason, the joint torques/accelerations obtained from the dqNE, the rtNE, and the sv2NE are always filtered with a discrete low-pass Butterworth filter before being compared with the—also filtered—results obtained from the V-REP.

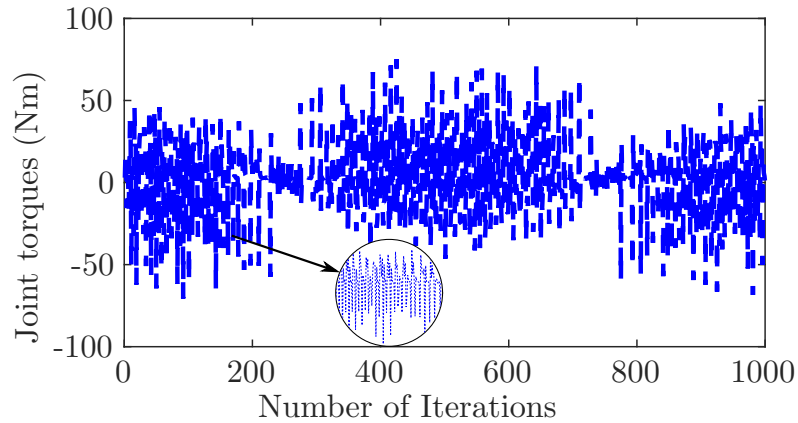
Alternatively to the procedure adopted in this thesis, the joint velocities  $\dot{\mathbf{q}}$  could have been filtered before being used to calculate the joint accelerations  $\ddot{\mathbf{q}}$ . Then, the obtained joint accelerations  $\ddot{\mathbf{q}}$  could also have been filtered. Finally, the filtered  $\dot{\mathbf{q}}$  and  $\ddot{\mathbf{q}}$  could have been given to all the different dynamic modeling strategies. The results of such approach, using a 2nd-order low-pass digital Butterworth filter with normalized cutoff frequency of 500Hz and 100Hz to respectively filter the joint velocities  $\dot{\mathbf{q}}$  and the joint accelerations  $\ddot{\mathbf{q}}$ , are shown in figure D.2. Although less noisier than the joint torque presented in figure D.1c,



(a) Unfiltered joint velocity waveform of the third joint of the 6-DoF JACO robotic arm obtained from the V-REP.



(b) Unfiltered joint acceleration waveform of the third joint of the 6-DoF JACO robotic arm obtained from the numerical differentiation of the unfiltered joint velocity shown in figure D.1a.

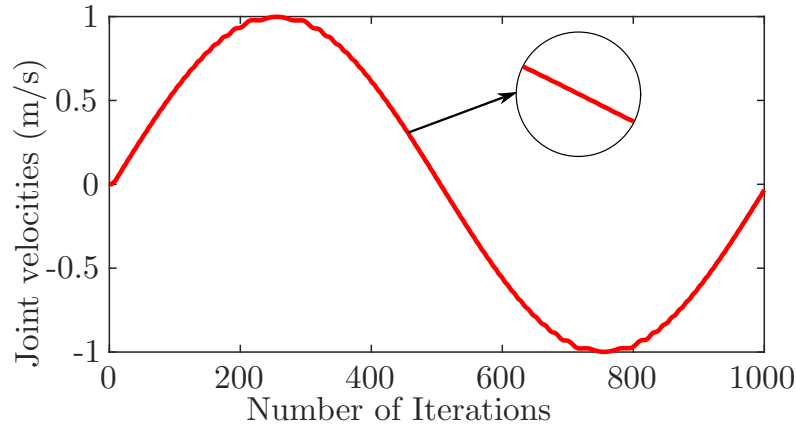


(c) Unfiltered joint torque waveform of the third joint of the 6-DoF JACO robotic arm obtained by the dqNE considering the unfiltered joint velocity and the joint acceleration given in figures D.1a and D.1b, respectively.

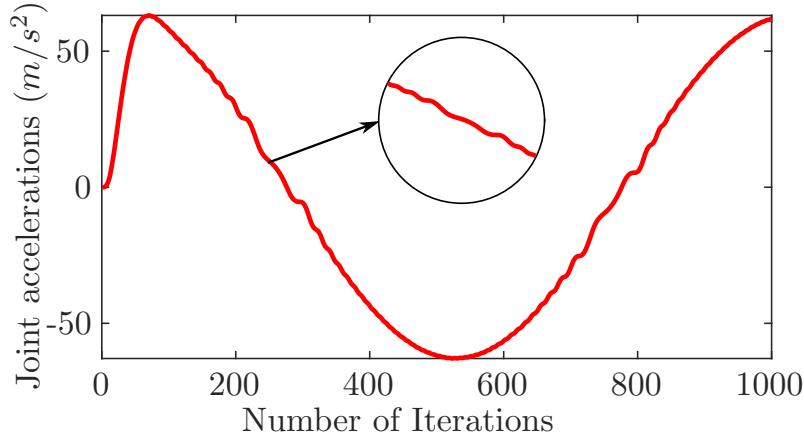
Figure D.1: Unfiltered joint velocity, acceleration, and torque waveforms of the third joint of the 6-DoF JACO robotic. Figure D.1c presents the unfiltered joint torque obtained by the dual quaternion Newton-Euler formalism considering the unfiltered joint acceleration given in figure D.1b, which was obtained from the numerical differentiation of the unfiltered joint velocity shown in figure D.1a.

the joint torque shown in figure D.2c would still requires filtering. Thus, this thesis adopted the first approach described in this appendix, as it only applies the filter once, to the final results of joint torques/accelerations.

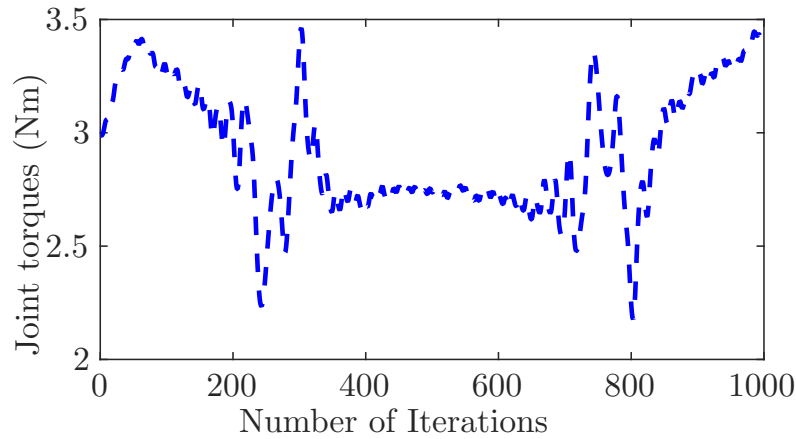




(a) Filtered joint velocity waveform of the third joint of the 6-DoF JACO robotic arm obtained from the V-REP.



(b) Filtered joint acceleration waveform of the third joint of the 6-DoF JACO robotic arm obtained from the numerical differentiation of the filtered joint velocity shown in figure D.2a.



(c) Unfiltered joint torque waveform of the third joint of the 6-DoF JACO robotic arm obtained by the dqNE considering the filtered joint velocity and the joint acceleration given in figures D.2a and D.2b, respectively.

Figure D.2: Filtered joint velocity and acceleration and unfiltered torque waveforms of the third joint of the 6-DoF JACO robotic. Figure D.2c presents the unfiltered joint torque obtained by the dual quaternion Newton-Euler formalism considering the filtered joint acceleration given in figure D.2b, which was obtained from the numerical differentiation of the filtered joint velocity shown in figure D.2a.

# BIBLIOGRAPHY

- Abaunza, H., Castillo, P., Lozano, R., & Victorino, A. (2016). Quadrotor aerial manipulator based on dual quaternions. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 152–161).: IEEE.
- 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]*. Ph.d thesis, Université de Montpellier.
- Adorno, B. V. (2017). Robot Kinematic Modeling and Control Based on Dual Quaternion Algebra – Part I: Fundamentals. Available: <https://hal.archives-ouvertes.fr/hal-01478225v1>.
- Adorno, B. V., Bó, A. P., & Fraitse, P. (2015). Kinematic modeling and control for human-robot cooperation considering different interaction roles. *Robotica*, 33(2), 314–331.
- Adorno, B. V., Bo, A. P. L., & Fraitse, P. (2011a). Interactive manipulation between a human and a humanoid: When robots control human arm motion. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 4658–4663).: IEEE.
- Adorno, B. V., Bo, A. P. L., Fraitse, P., & Pognet, P. (2011b). Towards a cooperative framework for interactive manipulation involving a human and a humanoid. In *2011 IEEE International Conference on Robotics and Automation* (pp. 3777–3783).: IEEE.
- Adorno, B. V., Fraitse, P., & 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* (pp. 3955–3960). Taipei, Taiwan: IEEE.
- Adorno, B. V. & Marques Marinho, M. (2021). DQ Robotics: A Library for Robot Modeling and Control. *IEEE Robotics & Automation Magazine*, 28(3), 102–116.
- Andrews, G. & Kesavan, H. (1975). The vector-network model: A new approach to vector dynamics. *Mechanism and Machine Theory*, 10(1), 57–75.

- Andrews, G. C., Richard, M. J., & Anderson, R. J. (1988). A general vector-network formulation for dynamic systems with kinematic constraints. *Mechanism and Machine Theory*, 23(3), 243–256.
- Aspragathos, N. A. & Dimitros, J. K. (1998). A comparative study of three methods for robot kinematics. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 28(2), 135–145.
- Baciu, G., Chou, J., & Kesavan, H. (1990). Constrained multibody systems: graph-theoretic Newton-Euler formulation. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(5), 1025–1048.
- Baklouti, M. & Castelain, J. M. (1993). Dynamic model of robot manipulator in explicit form formulation with dual vectors and the notion of augmented body. *Journal of Robotic Systems*, 10(2), 271–298.
- Balafoutis, C. A. (1994). A survey of efficient computational methods for manipulator inverse dynamics. *Journal of Intelligent & Robotic Systems*, 9(1-2), 45–71.
- Bouyarmane, K., Chappellet, K., Vaillant, J., & Kheddar, A. (2019). Quadratic Programming for Multirobot and Task-Space Force Control. *IEEE Transactions on Robotics*, 35(1), 64–77.
- Brodsky, V. & Shoham, M. (1994). The Dual Inertia Operator and Its Application to Robot Dynamics. *Journal of Mechanical Design*, 116(4), 1089.
- Brodsky, V. & Shoham, M. (1999). Dual numbers representation of rigid body dynamics. *Mechanism and Machine Theory*, 34(5), 693–718.
- Carpentier, J., Saurel, G., Buondonno, G., Mirabel, J., Lamiroux, F., Stasse, O., & Mansard, N. (2019). The Pinocchio C++ library : A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives. In *2019 IEEE/SICE International Symposium on System Integration (SII)* (pp. 614–619).: IEEE.
- Chembrammell, P. & Kesavadas, T. (2019). A new implementation for online calculation of manipulator Jacobian. *PLOS ONE*, 14(2), e0212018.
- Chen, T. L., Ciocarlie, M., Cousins, S., Grice, P. M., Hawkins, K., Kaijen Hsiao, Kemp, C. C., Chih-Hung King, Lazewatsky, D. A., Leeper, A. E., Hai Nguyen, Paepcke, A., Pantofaru, C., Smart, W. D., & Takayama, L. (2013). Robots for humanity: using assistive robotics to empower people with disabilities. *IEEE Robotics & Automation Magazine*, 20(1), 30–39.
- Chou, J., Singhal, K., & Kesavan, H. (1986a). Multi-body systems with open chains: Graph-theoretic models. *Mechanism and Machine Theory*, 21(3), 273–284.

- Chou, J. C. K., Kesavan, H. K., & Singhal, K. (1986b). A Systems Approach to Three-Dimensional Multibody Systems Using Graph-Theoretic Models. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(2), 219–230.
- Cohen, A. & Shoham, M. (2016). Application of Hyper-Dual Numbers to Multibody Kinematics. *Journal of Mechanisms and Robotics*, 8(1), 2–5.
- Corke, P. (2011). *Robotics, Vision and Control*, volume 73 of *Springer Tracts in Advanced Robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Corke, P. (2017). *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, volume 118 of *Springer Tracts in Advanced Robotics*. Cham, Switzerland: Springer International Publishing, 2nd edition.
- Dantam, N. T. (2020). Robust and efficient forward, differential, and inverse kinematics using dual quaternions. *The International Journal of Robotics Research*, (pp. 027836492093194).
- De Luca, A. & Ferrajoli, L. (2009). A modified newton-euler method for dynamic computations in robot fault detection and control. In *2009 IEEE International Conference on Robotics and Automation* (pp. 3359–3364).: IEEE.
- De Luca, A., Oriolo, G., & 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.*, volume 2006 (pp. 1867–1873).: IEEE.
- Dimentberg, F. M. (1965). *Screw Theory and its Application in Mechanics*. Moskva: Fiziko-Matematicheskoy Literatury.
- Dooley, J. & McCarthy, J. (1991). Spatial rigid body dynamics using dual quaternion components. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, number April (pp. 90–95). Sacramento, California: IEEE Comput. Soc. Press.
- Featherstone, R. (1983). The Calculation of Robot Dynamics Using Articulated-Body Inertias. *The International Journal of Robotics Research*, 2(1), 13–30.
- Featherstone, R. (1984). *Robot dynamics algorithms*. Phd thesis, University of Edinburgh.
- Featherstone, R. (2008). *Rigid body dynamics algorithms*. New York, NY: Springer.
- Featherstone, R. (2010a). A Beginner’s Guide to 6-D Vectors (Part 1). *IEEE Robotics & Automation Magazine*, 17(3), 83–94.

- Featherstone, R. (2010b). A Beginner's Guide to 6-D Vectors (Part 2). *IEEE Robotics & Automation Magazine*, 17(4), 88–99.
- Felis, M. L. (2017). RBDL: an efficient rigid-body dynamics library using recursive algorithms. *Autonomous Robots*, 41(2), 495–511.
- Ferrari, A., Cutti, A. G., & Cappello, A. (2010). A new formulation of the coefficient of multiple correlation to assess the similarity of waveforms measured synchronously by different motion analysis protocols. *Gait & Posture*, 31(4), 540–542.
- Fierro, R. & Lewis, F. L. (1997). Control of a nonholomic mobile robot: Backstepping kinematics into dynamics. *Journal of Robotic Systems*, 14(3), 149–163.
- Figueredo, L., Adorno, B. V., Ishihara, J., & Borges, G. (2014). Switching strategy for flexible task execution using the cooperative dual task-space framework. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, number Iros (pp. 1703–1709). Chicago, IL, USA: IEEE.
- Filipe, N. & Tsiotras, P. (2013). Simultaneous position and attitude control without linear and angular velocity feedback using dual quaternions. In *2013 American Control Conference*, number June (pp. 4808–4813): IEEE.
- Fonseca, M. d. P. A., Adorno, B. V., & Fraisse, P. (2020). Coupled Task-Space Admittance Controller Using Dual Quaternion Logarithmic Mapping. *IEEE Robotics and Automation Letters*, 5(4), 6057–6064.
- Funda, J., Taylor, R. H., & Paul, R. P. (1990). On Homogeneous Transforms, Quaternions, and Computational Efficiency. *IEEE Transactions on Robotics and Automation*, 6(3), 382–388.
- Gienger, M., Janben, H., & Goerick, C. (2006). Exploiting Task Intervals for Whole Body Robot Control. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2484–2490): IEEE.
- Gilat, A. & Subramaniam, V. (2014). *Numerical Methods for Engineers and Scientists: An Introduction with Applications using MATLAB*. Hoboken, USA: John Wiley & Sons, 3rd edition.
- Goodrich, M. A. & Schultz, A. C. (2007). Human-Robot Interaction: A Survey. *Foundations and Trends® in Human-Computer Interaction*, 1(3), 203–275.
- Gouasmi, M. (2012). Robot Kinematics using Dual Quaternions. *IAES International Journal of Robotics and Automation (IJRA)*, 1(1).

- Gui, H. & Vukovich, G. (2016). Dual-quaternion-based adaptive motion tracking of spacecraft with reduced control effort. *Nonlinear Dynamics*, 83(1-2), 597–614.
- Gürgöze, M. & Zeren, S. (2012). Understanding Dyadics and Their Applications in Mechanical Engineering. *Scottish Journal of Arts, Social Sciences and Scientific Studies*, 3(1), 95–110.
- Hachicho, O. & Eldin, H. N. (2000). Dual Hypercomplex Quaternions Based Recursions for Generalized Velocities, Accelerations and Forces in Robot Dynamics. *System and Control: Theory and Applications*, (pp. 85–89).
- Hamilton, W. R. (1844). II. On quaternions; or on a new system of imaginaries in algebra. *Philosophical Magazine Series 3*, 25(163), 10–13.
- Han, D.-P., Wei, Q., & Li, Z.-X. (2008). Kinematic control of free rigid bodies using dual quaternions. *International Journal of Automation and Computing*, 5(3), 319–324.
- Hanlei, W. (2010). On the Recursive Implementation of Adaptive Control for Robot Manipulators. *Proceedings of the 29th Chinese Control Conference*, (pp. 2154–2161).
- Hess-Coelho, T. A., Orsino, R. M. M., & Malvezzi, F. (2021). Modular modelling methodology applied to the dynamic analysis of parallel mechanisms. *Mechanism and Machine Theory*, 161, 104332.
- Huang, T., Yang, S., Wang, M., Sun, T., & Chetwynd, D. G. (2015). An Approach to Determining the Unknown Twist/Wrench Subspaces of Lower Mobility Serial Kinematic Chains. *Journal of Mechanisms and Robotics*, 7(3), 1–9.
- Jain, A. (2012). Multibody graph transformations and analysis. *Nonlinear Dynamics*, 67(4), 2779–2797.
- Jone Hann Jean & Li-Chen Fu (1993). Adaptive hybrid control strategies for constrained robots. *IEEE Transactions on Automatic Control*, 38(4), 598–603.
- Kane, T. R., Likins, P. W., Levinson, D. A., & Huston, R. L. (1983). *Spacecraft Dynamics*. Ithaca: McGraw-Hill Book Company.
- Khatib, O. (1986). *The Potential Field Approach And Operational Space Formulation In Robot Control*. Boston, MA: Springer US, 1st editio edition.
- Kleff, S., Meduri, A., Budhiraja, R., Mansard, N., & Righetti, L. (2021). High-Frequency Nonlinear Model Predictive Control of a Manipulator. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, number Icra (pp. 7330–7336).: IEEE.

- Kong, X. (2017). Reconfiguration Analysis of Multimode Single-Loop Spatial Mechanisms Using Dual Quaternions. *Journal of Mechanisms and Robotics*, 9(5), 1–8.
- Kotay, K., Rus, D., Vona, M., & McGray, C. (1998). The self-reconfiguring robotic molecule. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, volume 1 (pp. 424–431).: IEEE.
- Kussaba, H. T., Figueredo, L. F., Ishihara, J. Y., & Adorno, B. V. (2017). Hybrid kinematic control for rigid body pose stabilization using dual quaternions. *Journal of the Franklin Institute*, 354(7), 2769–2787.
- Lana, E. P., Adorno, B. V., & Maia, C. A. (2015). A new algebraic approach for the description of robotic manipulation tasks. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, volume 2015-June (pp. 3083–3088). Seattle, Washington: IEEE.
- Lana, E. P., Adorno, B. V., & Tierra-Criollo, C. J. (2013). Assistance Task Using a Manipulator Robot and User Kinematics Feedback. *Simpósio Brasileiro de Automação Inteligente*, (pp. 1–6).
- Lasota, P. A., Rossano, G. F., & Shah, J. A. (2014). Toward safe close-proximity human-robot interaction with standard industrial robots. In *2014 IEEE International Conference on Automation Science and Engineering (CASE)* (pp. 339–344). Taipei, Taiwan: IEEE.
- Lee, C. & Chung, M. (1984). An adaptive control strategy for mechanical manipulators. *IEEE Transactions on Automatic Control*, 29(9), 837–840.
- Luca, A. D. (2019). Dynamic model of robots : Newton-Euler approach.
- Luh, J., Walker, Í., & Paul, R. (1979). Newton - Euler Formulation of Manipulator Dynamics for Computer Control. *IFAC Proceedings Volumes*, 12(10), 165–172.
- Luh, J. Y. S., Walker, M. W., & Paul, R. P. C. (1980). On-Line Computational Scheme for Mechanical Manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 102(2), 69–76.
- Marinho, M. M., Adorno, B. V., Harada, K., & Mitsuishi, M. (2019). Dynamic Active Constraints for Surgical Robots Using Vector-Field Inequalities. *IEEE Transactions on Robotics*, 35(5), 1166–1185.
- Marinho, M. M., Figueredo, L. F. C., & 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)* (pp. 4047–4052). Hamburg, Germany: IEEE.

- Matarazzo Orsino, R. M. & Hess-Coelho, T. A. (2015). A contribution on the modular modelling of multibody systems. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2183), 20150080.
- McCarthy, J. (1990). *Introduction to Theoretical Kinematics*. Cambridge, Massachusetts: The MIT Press, first edit edition.
- McPhee, J., Schmitke, C., & Redmond, S. (2004). Dynamic Modelling of Mechatronic Multibody Systems With Symbolic Computing and Linear Graph Theory. *Mathematical and Computer Modelling of Dynamical Systems*, 10(1), 1–23.
- McPhee, J. J. (1996). On the use of linear graph theory in multibody system dynamics. *Nonlinear Dynamics*, 9(1-2), 73–90.
- McPhee, J. J. (1998). Automatic generation of motion equations for planar mechanical systems using the new set of "branch coordinates". *Mechanism and Machine Theory*, 33(6), 805–823.
- Mello, L. S., Raffo, G. V., & Adorno, B. V. (2016). Robust whole-body control of an unmanned aerial manipulator. In *2016 European Control Conference (ECC)* (pp. 702–707).: IEEE.
- Miranda de Farias, C., da Cruz Figueredo, L. F., & Yoshiyuki Ishihara, J. (2019a). A Novel Dual Quaternion Based Cost Efficient Recursive Newton-Euler Inverse Dynamics Algorithm. *International Journal of Robotic Computing*, 1(2), 144–168.
- Miranda de Farias, C., da Cruz Figueredo, L. F., & Yoshiyuki Ishihara, J. (2019b). Performance Study on dqRNEA - A Novel Dual Quaternion Based Recursive Newton-Euler Inverse Dynamics Algorithms. In *2019 Third IEEE International Conference on Robotic Computing (IRC)* (pp. 94–101). Naples, Italy: IEEE.
- Mises, R. V. (1924). Motorrechnung, ein neues Hilfsmittel der Mechanik. *ZAMM - Zeitschrift für Angewandte Mathematik und Mechanik*, 4(2), 155–181.
- Murray, R. M., Li, Z., & Sastry, S. S. (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press, first edit edition.
- Neubert, J. & Lipson, H. (2016). Soldercubes: a self-soldering self-reconfiguring modular robot system. *Autonomous Robots*, 40(1), 139–158.
- Nishiwaki, K., Kagami, S., & Inoue, H. (2005). Object manipulation by hand using whole-body motion coordination. In *IEEE International Conference Mechatronics and Automation, 2005*, volume 4 (pp. 1778–1783).: IEEE.



- Orsino, R. M. M. (2017). Recursive modular modelling methodology for lumped-parameter dynamic systems. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2204), 20160891.
- Ott, C. & Hyon, S.-h. (2019). Torque-Based Balancing. In *Humanoid Robotics: A Reference* (pp. 1361–1386). Dordrecht: Springer Netherlands.
- Özgül, E. & Mezouar, Y. (2016). Kinematic modeling and control of a robot arm using unit dual quaternions. *Robotics and Autonomous Systems*, 77, 66–73.
- Park, H. A. & Lee, C. S. G. (2013). Cooperative-Dual-Task-Space-based whole-body motion balancing for humanoid robots. In *2013 IEEE International Conference on Robotics and Automation* (pp. 4797–4802).: IEEE.
- Pennock, G. R. & Meehan, P. J. (2000). Geometric insight into the dynamics of a rigid body using the theory of screws. In *Proceedings of the Ball Symposium 2000* (pp. 9–11).
- Pennock, G. R. & Oncu, B. A. (1992). Application of Screw Theory to Rigid Body Dynamics. *Journal of Dynamic Systems, Measurement, and Control*, 114(2), 262.
- Pennock, G. R. & Yang, A. T. (1983). Dynamic Analysis of a Multi-Rigid-Body Open-Chain System. *Journal of Mechanisms Transmissions and Automation in Design*, 105(1), 28.
- Perez, A. & McCarthy, J. (2004). Dual Quaternion Synthesis of Constrained Robotic Systems. *Journal of Mechanical Design*, 126(3), 425.
- Ploen, S. R. (1999). A skew-symmetric form of the recursive Newton-Euler algorithm for the control of multibody systems. In *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)*, volume 6 (pp. 3770–3773 vol.6).: IEEE.
- Quiroz-Omaña, J. J. & Adorno, B. V. (2017). Whole-Body Kinematic Control of Non-holonomic Mobile Manipulators Using Linear Programming. *Journal of Intelligent & Robotic Systems*, (pp.91).
- Quiroz-Omana, J. J. & Adorno, B. V. (2019). Whole-Body Control With (Self) Collision Avoidance Using Vector Field Inequalities. *IEEE Robotics and Automation Letters*, 4(4), 4048–4053.
- Ramuzat, N., Stasse, O., & Boria, S. (2022). Benchmarking Whole-Body Controllers on the TALOS Humanoid Robot. *Frontiers in Robotics and AI*, 9.
- Ravani, B. & Roth, B. (1984). Mappings of Spatial Kinematics. *Journal of Mechanisms Transmissions and Automation in Design*, 106(3), 341.

- Rego, B. S., Adorno, B. V., & Raffo, G. V. (2016). Formation Backstepping Control Based on the Cooperative Dual Task-Space Framework: A Case Study on Unmanned Aerial Vehicles. In *2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR)* (pp. 163–168).: IEEE.
- Reher, J., Ma, W.-l., & Ames, A. D. (2019). Dynamic Walking with Compliance on a Cassie Bipedal Robot.
- Renda, F., Cianchetti, M., Abidi, H., Dias, J., & Seneviratne, L. (2017). Screw-Based Modeling of Soft Manipulators With Tendon and Fluidic Actuation. *Journal of Mechanisms and Robotics*, 9(4).
- Reungwetwattana, A. & Toyama, S. (2001). An Efficient Dynamic Formulation for Multibody Systems. *Multibody System Dynamics*, 6(3), 267–289.
- Rohmer, E., Singh, S. P. N., & Freese, M. (2013). V-REP: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1321–1326).: IEEE.
- Salazar, A. V. (2018). *Dynamic Modeling and Control of Spacecraft Robotic Systems using Dual Quaternions*. PhD thesis.
- Savino, H. J., Pimenta, L. C., Shah, J. A., & Adorno, B. V. (2020). Pose consensus based on dual quaternion algebra with application to decentralized formation control of mobile manipulators. *Journal of the Franklin Institute*, 357(1), 142–178.
- Schaal, S. (2019). Historical Perspective of Humanoid Robot Research in the Americas. In *Humanoid Robotics: A Reference* (pp. 9–17). Dordrecht: Springer Netherlands.
- Selig, J. M. (2004). Lie Groups and Lie Algebras in Robotics. In *Computational Non-commutative Algebra and Applications* (pp. 101–125). Dordrecht, Netherlands: Kluwer Academic Publishers.
- Selig, J. M. (2005). *Geometric Fundamentals of Robotics*. Monographs in Computer Science. New York, NY: Springer New York, 2nd edition.
- Selig, J. M. & Bayro-Corrochano, E. (2010). Rigid Body Dynamics Using Clifford Algebra. *Advances in Applied Clifford Algebras*, 20(1), 141–154.
- Sentis, L. & Khatib, O. (2006). A whole-body control framework for humanoids operating in human environments. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, number May (pp. 2641–2648).: IEEE.

- Sheth, P. N. & Uicker, J. J. (1972). IMP (Integrated Mechanisms Program), A Computer-Aided Design Analysis System for Mechanisms and Linkage. *Journal of Engineering for Industry*, 94(2), 454–464.
- Shoham, M. & Brodsky, V. (1993). Analysis of Mechanisms by the Dual Inertia Operator. In J. Angeles, G. Hommel, & P. Kovács (Eds.), *Computational Kinematics*, volume 28 of *Solid Mechanics and Its Applications* (pp. 129–138). Dordrecht: Springer Netherlands.
- Siciliano, B. & Khatib, O. (2008). *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2009). *Robotics*. Advanced Textbooks in Control and Signal Processing. London: Springer London.
- Silva, F. F. A. & Adorno, B. V. (2018). Whole-body Control of a Mobile Manipulator Using Feedback Linearization and Dual Quaternion Algebra. *Journal of Intelligent & Robotic Systems*, 91(2), 249–262.
- Silva, F. F. A., Quiroz-Omaña, J. J., & Adorno, B. V. (2022). Dynamics of Mobile Manipulators using Dual Quaternion Algebra. *Journal of Mechanisms and Robotics*, (pp. 1–24).
- Sleiman, J. P., Farshidian, F., Minniti, M. V., & Hutter, M. (2021). A Unified MPC Framework for Whole-Body Dynamic Locomotion and Manipulation. *IEEE Robotics and Automation Letters*, 6(3), 4688–4695.
- Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2006). *Robot Modeling and Control*. New York, NY: Wiley.
- Udwadia, F. E. & Kalaba, R. E. (1992). A new perspective on constrained motion. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1906), 407–410.
- Unsal, C. & Khosla, P. (2000). Mechatronic design of a modular self-reconfiguring robotic system. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 2 (pp. 1742–1747).: IEEE.
- Valverde, A. & Tsiotras, P. (2018a). Dual Quaternion Framework for Modeling of Spacecraft-Mounted Multibody Robotic Systems. *Frontiers in Robotics and AI*, 5(November).
- Valverde, A. & Tsiotras, P. (2018b). Modeling of Spacecraft-Mounted Robot Dynamics and Control Using Dual Quaternions. In *2018 Annual American Control Conference (ACC)* (pp. 670–675). Wisconsin Center, Milwaukee, USA: IEEE.

- Walker, M. (1988). An efficient algorithm for the adaptive control of a manipulator. In *Proceedings, 1988 IEEE International Conference on Robotics and Automation* (pp. 682–690).: IEEE Comput. Soc. Press.
- Walker, M., Kim, D., & Dionise, J. (1989). Adaptive coordinated motion control of two manipulator arms. In *Proceedings, 1989 International Conference on Robotics and Automation* (pp. 1084–1090).: IEEE Comput. Soc. Press.
- Wang, X. & Yu, C. (2010). Feedback linearization regulator with coupled attitude and translation dynamics based on unit dual quaternion. In *2010 IEEE International Symposium on Intelligent Control* (pp. 2380–2384).: IEEE.
- Wohllhart, K. (1995). Motor Tensor Calculus. In *Angewandte Chemie International Edition*, 6(11), 951-952. (pp. 93–102).
- Xian, B., DeQueiroz, M., Dawson, D., & Walker, I. (2004). Task-Space Tracking Control of Robot Manipulators via Quaternion Feedback. *IEEE Transactions on Robotics and Automation*, 20(1), 160–167.
- Yang, A. T. (1965). Static Force and Torque Analysis of Spherical Four-Bar Mechanisms. *Journal of Engineering for Industry*, 87(2), 221.
- Yang, A. T. (1966). Acceleration Analysis of Spatial Four-Link Mechanisms. *Journal of Engineering for Industry*, 88(3), 296.
- Yang, A. T. (1967). Application of Dual Quaternions to the Study of Gyrodynamics. *Journal of Engineering for Industry*, 89(1), 137.
- Yang, A. T. (1971). Inertia Force Analysis of Spatial Mechanisms. *Journal of Engineering for Industry*, 93(1), 27.
- Yang, A. T. & Freudenstein, F. (1964). Application of Dual-Number Quaternion Algebra to the Analysis of Spatial Mechanisms. *Journal of Applied Mechanics*, 31(2), 300.
- Yuan, J. (1988). Closed-loop manipulator control using quaternion feedback. *IEEE Journal on Robotics and Automation*, 4(4), 434–440.
- Zhang, F. & Duan, G. (2011). Robust Integrated Translation and Rotation Finite-Time Maneuver of a Rigid Spacecraft Based on Dual Quaternion. In *AIAA Guidance, Navigation, and Control Conference*, number 92 (pp. 1–17). Reston, Virginia: American Institute of Aeronautics and Astronautics.
- Zhang, H., Jia, Y., & Xi, N. (2012). Sensor-based redundancy resolution for a nonholonomic mobile manipulator. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 5327–5332).: IEEE.

- Zhang, H., Wang, X., & Han, D. (2010). Dual-Quaternion-Based Variable Structure Control: A New Approach and Application. In H. Liu, H. Ding, Z. Xiong, & X. Zhu (Eds.), *Intelligent Robotics and Applications*, volume 6425 of *Lecture Notes in Artificial Intelligence* (pp. 75–86). Shanghai, China: Springer Berlin Heidelberg NewYork.
- Zu, Y., Lee, U., & Dai, R. (2018). Distributed estimation for spatial rigid motion based on dual quaternions. *Optimal Control Applications and Methods*, 39(4), 1371–1392.

# PUBLICATIONS

Partial results of this thesis have been published in the following papers:

## Journals

1. Silva, F. F. A., Quiroz-Omaña, J. J., & Adorno, B. V. (2022). Dynamics of Mobile Manipulators using Dual Quaternion Algebra. *Journal of Mechanisms and Robotics*, 1–24.
2. Silva, F. F. A. & Adorno, B. V. (2022). Dynamic Modeling of Branched Robots using Modular Composition (to be submitted). *IEEE Robotics and Automation Letters*.

## Extended abstracts in international conferences

1. Silva, F. F. A. & Adorno, B. V. (2019). Wrench Control based on Dual Quaternion Algebra. *Workshop on Applications of Dual Quaternion Algebra to Robotics*. Belo Horizonte, Brasil.