

Modelling, Estimation and Identification of Humanoid Robots Dynamics

Silvio Traversaro



ISTITUTO ITALIANO
DI TECNOLOGIA

Supervisor: Dott. Francesco Nori

Fondazione Istituto Italiano di Tecnologia
Genova, Italia
2017

Abstract

During the last fifty years, the scientific community has dealt with the problem of designing **humanoid robots**, with the goal of creating artificial machines flexible enough to master any task performed by a human. A prerequisite for such *generic* machines would be the ability to control the forces that robots exchange with the environment, in addition to their own motion. These requirements imply that any *controller* of a humanoid robot requires an *implicit* or *explicit model* for the robot's **dynamics**, i.e. the laws describing the relation between the robot motion and the forces applied on itself, being either the *external* forces that the robot exchange with the environment or the forces provided by its own motors. Additionally, any time-variant quantity present in these models need to be *perceived* by the humanoid robot. If a quantity is not directly measured by a sensor, it needs to be **estimated** using the available measurements of different quantities and the dynamical models that relate them. Both the dynamics models and the sensor models are typically not perfectly known, and need to be **identified** from measured data and using a set of a-priori hypothesis. This thesis focuses on addressing the problems of modelling, estimation and identification of humanoid robots, focusing in particular to the specific characteristics and sensor set of the iCub humanoid robot.

Acknowledgements

*A tutti (scusate se dimentico qualcuno):
... grazie.*

The first group of people I would like to thank are the one that I consider my “scientific mentors”, the one without which this thesis simply would not exists. The first one is Andrea Del Prete, that I first met during my Master’s studies, and with which I had the luck of working while preparing my Master’s Thesis and before starting my PhD. The second one is Francesco Nori, my Master’s and PhD supervisor and the first person in my professional life to trust me with real responsibility. The third one is Daniele Pucci, that was an invaluable companion during my PhD. The fourth one is Alessandro Saccon that I visited in Eindhoven during the end of 2015, and with which I had a lot “open-minding” discussions.

I would like to thank also all my other colleagues at the Istituto Italiano di Tecnologia (IIT), firstly Francesco (Romano), Johrabib and Luca with whom I had the lucky to start to work in the group that would later become the Dynamic Interaction Control group at IIT. Furthermore, I would like to thank all other people that joined the Dynamic Interaction Control lab at different stages and in different ways: Talha, Naveen, Francesca, Yue, Prashanth, Claudia, Nuno, Gabriele, Francisco, Marie, Stefano, Matteo, Ali, Joan, Marta, Maria and Diego. Your feedback, even when just asking question, was invaluable for this thesis.

I would also like to thanks are the amazing crew that before I joined IIT and during the years built and maintained the iCub robot and all the related infrastructure. Ugo, Randaz, Vadim, Lorenzo, Daniele, Ali, Giorgio, Maggia, Stefano, Alberto, Marco, Valentina, Julien, Spada, Gesino, Masa, Marcello were for me an example of dedication. Trying to be at their level has always been a great motivation to always learn more and work harder.

I would like to thank who in this years took care that I was coming to work everyday: Francesco, Raffaello and Cive when commuting by car, and Anand, Giovanni and Emy when living in Genoa.

I would like to thank Benni, Titti and One. I would also like to thank the parish I have grown up in, Santo Stefano del Ponte, and everyone I met related to that. All those experiences thought me the few things I know about trying to care for any human being, a fundamental skill for any job. I would also like to thank my family, my sister Marta, my parents Caterina and Gian Renzo, my aunt Anna and my grandparents Angelina, Guido, Olga and Silvio. In the deepest sense, this thesis would have never been written if it was not for them.

Last, I would like to thank Camilla (or *Cami*, as she prefers to be called). The kind of gratitude I have towards her is simply not fit to print in this thesis.

Contents

1	Introduction	1
1.1	iCub Humanoid Robot	1
1.2	Thesis Organization	6
1.2.1	Modelling	6
1.2.2	Estimation	6
1.2.3	Identification	6
1.3	Technological Outcome	7
2	Rigid Body	8
2.1	Overview of the notation	9
2.2	Math preliminaries	10
2.3	Frame kinematics	11
2.3.1	Rigid Body Assumption	11
2.3.2	Points and coordinate frames	12
2.3.3	6D Velocity vectors	13
2.3.4	Adjoint matrix as a change of frame	16
2.4	Frame acceleration	17
2.4.1	The cross product on \mathbb{R}^6 (\times)	17
2.4.2	Acceleration representations and their relation	18
2.4.3	“Sensor” acceleration	19
2.4.4	“Proper” acceleration	19
2.4.5	Illustrative Example: the Spinning Wheel	22
2.5	Force-Torque covectors	24
2.5.1	The dual cross product on \mathbb{R}^6 ($\bar{\times}^*$)	25
2.6	Rigid Body Dynamics	26
2.6.1	Review of Lagrangian Dynamics: the point mass	27
2.6.2	Rigid Body Lagrangian Dynamics	29
2.7	Newton-Euler equations in the different representations	34
2.7.1	Left-trivialized	34

2.7.2	Right-trivialized	34
2.7.3	Mixed	34
2.7.4	Sensor	35
2.8	Rigid Body Total Momentum and its relation with Dynamics	35
3	Multi Body	37
3.1	Composition of relative and absolute velocity	38
3.2	Joints	39
3.2.1	One Degree of Freedom Joints	40
3.2.2	Fixed joints	42
3.3	Modelling of Multibody Systems	42
3.3.1	Topology	42
3.3.2	Numbering	45
3.3.3	Shape	46
3.4	Relative Forward Kinematics	46
3.5	Multibody Lagrangian Dynamics	48
3.5.1	Absolute State definition	48
3.5.2	Kinematics	49
3.5.3	Multibody Lagrangian	51
3.6	Base Change of Variables	53
3.7	Free Floating Dynamics with different representations or the base velocity	55
3.7.1	Left-trivialized	56
3.7.2	Right-Trivialized	57
3.7.3	Mixed	57
3.8	Free Floating Dynamics with different base links	57
3.9	System state transformation providing centroidal dynamics	59
3.9.1	Frames not rigidly attached to a link	60
3.9.2	Recalls on centroidal dynamics quantities	61
3.9.3	Average 6D Velocity	62
3.9.4	Centroidal change of variables	62
3.10	Proofs	66
3.10.1	Proof of Theorem 3.3	66
3.10.2	Proof of Lemma 3.11	67
4	Contact Force-Torques and Internal Torques Estimation	69
4.1	Introduction	69
4.2	State of art	70
4.2.1	Joint torques estimation	70
4.2.2	External force-torque position estimation	71

4.2.3	External force-torque intensity estimation	71
4.3	Sensor Models	72
4.3.1	Joint Sensors	73
4.3.2	Link Sensors	73
4.4	Estimation of the net force-torque	75
4.4.1	Example: rigid body external force-torque estimation	75
4.4.2	Sensor based estimation	76
4.4.3	Kinematic based estimation	77
4.4.4	Hybrid estimation	78
4.5	Multibody External Force-Torque Estimation	78
4.5.1	Force-Torque Sensors Induced Submodel Decomposition	78
4.5.2	External Force-Torque Estimation	79
4.5.3	Estimation of external forces acting on multiple link in a submodel	81
4.6	Joint Torques Estimation	82
4.6.1	Example of torque estimation	84
4.7	Six-Axis Force-Torque Sensors Model-based Offset Calibration	85
4.7.1	Offset Calibration with the external forces acting on a Single Link	85
4.8	Proofs	86
4.8.1	Proof of Theorem 4.1	86
4.8.2	Proof of Theorem 4.2	86
5	Six Axis Force Torque Sensors In Situ Calibration	88
5.1	Introduction	88
5.2	Problem statement and assumptions	92
5.3	In Situ Calibration Methods	93
5.3.1	The geometry of the raw measurements	93
5.3.2	Method for estimating the sensor's offset	95
5.3.3	Method for estimating the sensor's calibration matrix	96
5.4	Experimental Results	99
6	Rigid Body Inertial Parameters Identification	106
6.1	Introduction	106
6.2	Background on Rigid Body Inertial Parameters	108
6.2.1	Rigid Body Dynamics	108
6.2.2	Inertial parameters	109
6.2.3	Relationship between the inertial parameters and the density function	109
6.2.4	3D Inertia at the Center of Mass and Principal Axes .	110

6.2.5	Inertial Parameters Identification	112
6.3	Full Physical Consistency	112
6.3.1	Full physical consistency	112
6.3.2	Full physical consistent parametrization of inertia pa- rameters	113
6.4	Experimental Results	115
6.4.1	Optimization on Manifolds	115
6.4.2	Experiments	117
7	Multi Body Inertial Parameters Identification	120
7.1	Introduction	120
7.2	Identification of floating base dynamics	122
7.3	Inertial Parameter Identification and Torque Estimation . . .	125
7.4	Experimental results	127
Appendices		145
A Mathematical background on Lie group formalism		146
A.1	Matrix Lie Groups	146
A.1.1	Matrix Lie Group Examples	150
A.2	Multibody Dynamics Notation and its connection to Lie Groups	153
A.2.1	Frame Pose and 6D Velocity	153
A.2.2	Cross Product on \mathbb{R}^6	153
A.2.3	The dual cross product on \mathbb{R}^6 ($\bar{\times}^*$)	154
A.3	Euler-Poincaré Equations and Rigid Body Dynamics	154
A.3.1	Euler-Poincaré Equations	154
A.3.2	Rigid Body Dynamics	156
A.4	Hamel Equations and Multi Body Dynamics	157
A.4.1	Hamel Equations	157
A.4.2	Multi Body Dynamics	158

Chapter 1

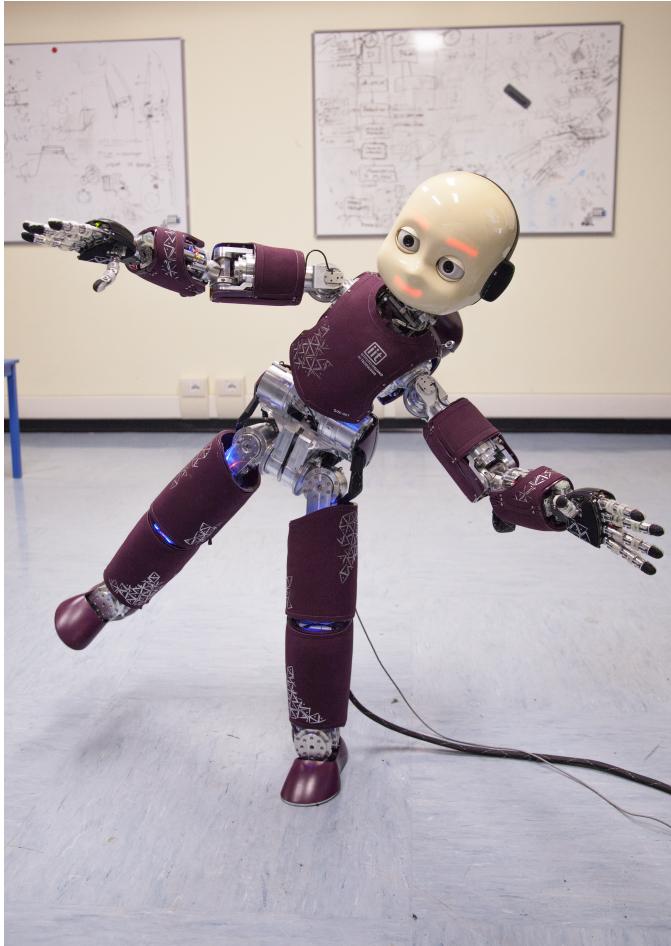
Introduction

1.1 iCub Humanoid Robot

The main robotic experimental platform used in this thesis is the iCub robot, developed by the *iCub Facility* at the *Italian Institute of Technology*. It is a child-sized humanoid robot originally developed by the RobotCub European Project for research in embodied cognition [Sandini et al., 2004]. Since its initial release in 2006, the iCub has been continuously updated with improvements and new features. iCub’s copies it have been distributed to more than 30 partners institution in Europe, Asia and United States. As the improvements are continuously released and integrated into the different iCub’s, all the copies of iCub have different features, depending on their release date, the maintenance’s updates performed during the years and specific customization of each iCub. In the following we discuss the characteristic relevant to this thesis, that apply to the latest “standard” version of iCub, informally referenced hereafter as iCub 2.5, as of the beginning of 2017.

The iCub is a 53 degrees-of-freedom (DOF) humanoid robot. The DOFs are distributed as in the following: 6 for each leg, 3 for the torso, 6 for the head and eyes, 7 for each arm and 9 for each hand. One additional servo motor is used to open and close the eye-lids. In this thesis we consider only a subset of 32 DOFs (legs, torso, arms and neck) that are actuated with Brushless DC electric motor (BLDC) with an Harmonic Drive transmission, making them suitable for joint torque control. More details on the actuation and mechanics of the iCub 2.5 can be found in [Parmiggiani et al., 2012].

Fig. 1.1 The iCub version 2.5, balancing itself on one foot thanks to *contact forces control*.



One of the research goals of the iCub project is to endow humanoids with advanced physical interaction capabilities. This is motivated by the idea that future robots will be required to physically interact with the environment and, in the long run, with humans. A key requirement for this *interaction control* is the capability to measure and control the forces that a robot is exchanging with its environment, i.e. *contact forces control*. While in traditional industrial applications this is achieved by placing a six-axis force-torque sensor between the robot and the environment, this is not feasible in humanoid robotics, in which the contact location is typically not known a-priori. To overcome this limitation, a *unique* set of dynamics-

related sensors have been added during the years to the iCub.

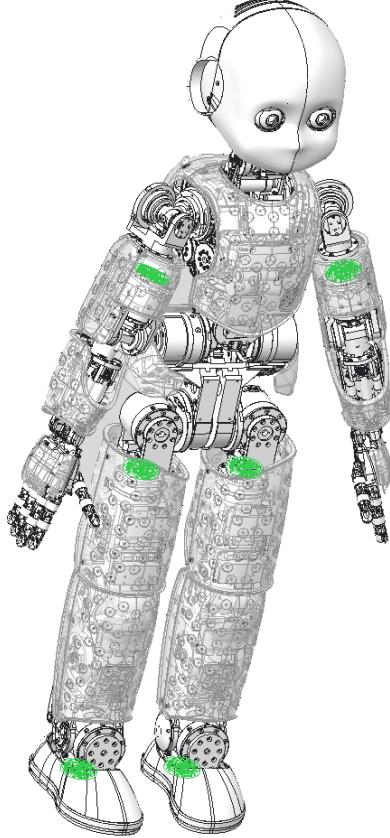


Fig. 1.2 Distribution of the six embedded six-axis force-torque sensors (green) available in iCub 2.5 .

The main force sensors available on the iCub are six *internal* six-axis force-torque sensors. Four of them are mounted at the base of each limb while two of them are mounted in feet right below the ankles. The locations of these sensors are highlighted in Figure 1.2. The placement of these sensors was dictated by practical reasons, as the hand structure would have not been able to support a force-torque sensor mounted in it, but also to enable estimation of internal joint torques and external force-torque, as described in [Fumagalli et al., 2010b] for a single limb and in Chapter 4 of this thesis for the whole-body case.

While internal force-torque sensors are able to measure force-torque due

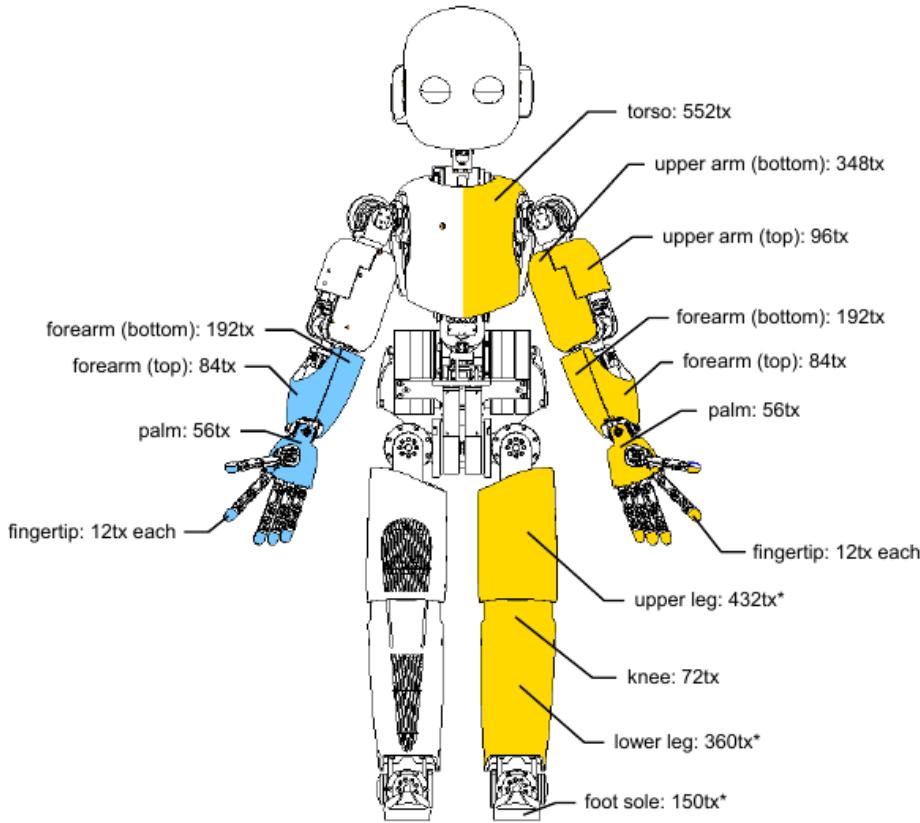


Fig. 1.3 Distribution of the tactile sensors on iCub 2.5.

to a combination of internal dynamics and of contact wrenches, they are in general unable to measure the location of a contact force, or its distribution over a contact surface. For addressing these concerns the iCub body has been covered with a distributed set of capacitive elements acting as tactile sensors [Maiolino et al., 2013]. The entire sensor network acts as an “artificial skin”, constituted by a sandwich of different flexible fabrics mounted on top of a flexible Printed Circuit Board (PCB), so that the entire structure can be conformed on surfaces of different curvatures. The majority of iCub 2.5 external surface has been covered by this “skin”, as shown by Figure 1.3.

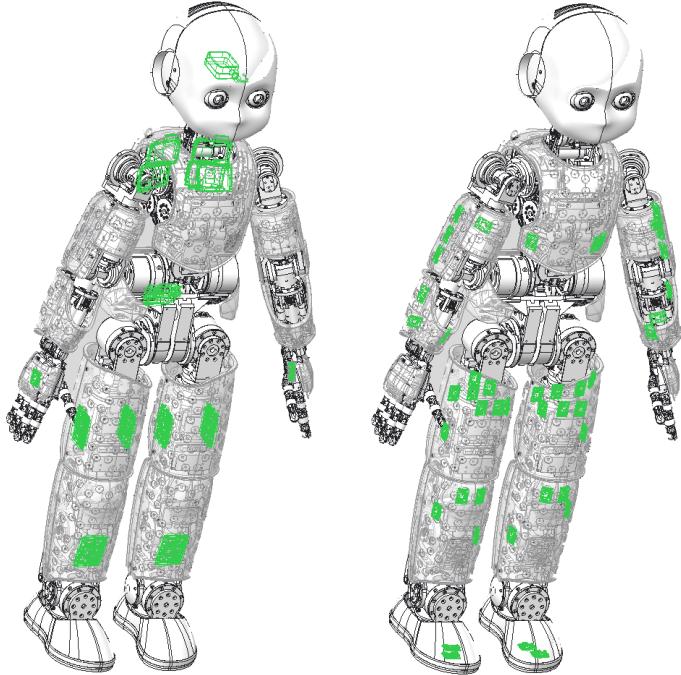


Fig. 1.4 Distribution of the inertial sensors i.e. gyroscopes (left) and accelerometers (right) in iCub 2.5 .

To read the distributed tactile sensors, the iCub has been equipped with distributed network of dedicated electronic boards. Remarkably, this boards are equipped with a 3 Degree-of-Freedom (DOFs) accelerometer. Similarly, several motor control boards are distributed in the robot structure, and each motor control board is equipped with both a 3 DOFs accelerometer and a 3 DOFs gyroscops. Furthermore, an full-fledged Inertial Measurement Unit, equipped with a 3 DOFs magnetometer, accelerometer and gyroscope is mounted on the head of the robot. These arrangements, shown in Figure 1.4 provides the iCub with a vast amount of distributed *inertial* sensing, that has been exploited for fine calibration [Guedelha et al., 2016].

These sensors form the basis for the use of the estimation and identification algorithms presented in the thesis, that are essential technologies to enable *interaction force control*. To describe the presented algorithms, we need to provide a solid background in multibody dynamics, that will be also presented in the thesis.

1.2 Thesis Organization

The following is a brief summary of the organization of the thesis. At the beginning of each chapter a discussion of the contribution with respect to the state of art of the specific chapter is given.

1.2.1 Modelling

This thesis includes in the Chapters 2, 3 a self-contained derivation of the multi body dynamics and in Appendix A its connection with Lie Groups theory. Particular focus is placed on the role of the *base link* in free-floating dynamics, and in the different representation of *base*-related quantities that are used in literature. This chapters form the theoretical basis for the rest of the thesis, devoted to *estimation* and *identification* of humanoid robot dynamics.

1.2.2 Estimation

Chapter 4 describes the algorithms used for the estimation of external forces and internal joint torques using internal six-axis force-torque sensors, distributed inertial sensing and a distributed tactile system, that are the set of sensors available on the iCub robot, as discussed in Section 1.1. This part is mainly a generalization of already existing results [Fumagalli et al., 2010c, Ivaldi et al., 2011, Fumagalli et al., 2012, Del Prete et al., 2012] to the whole-body case.

1.2.3 Identification

The main contribution of Chapter 5 are two new algorithms for calibration of six-axis force-torque sensors that can be performed *in situ*, i.e., without removing the sensor from the hosting system. These algorithms exploit the specific geometric of the gravity force-torque when expressed in the sensor frame.

Chapter 6 introduces the concept of the identification of the inertial parameters of a single rigid body. The main contribution of this chapter is the definition of a new condition, the *fully physical consistency* for a set of inertial parameters to determine if they can be generated by a physical rigid body. The proposed condition ensure both the positive definiteness and the triangular inequality of 3D inertia matrices as opposed to existing techniques in which the triangular inequality constraint is ignored.

Chapter 7 generalize the identification problem to the case of a system composed by multiple rigid bodies, with particular attention of the relations between the identifiability subspaces of the regressor associated with the different set of sensors. The main contribution of this chapter is the adaptation of the existing techniques for inertial parameters identification on humanoids to the specific set of sensors that is available on the iCub robot. While most existing techniques [Ayusawa et al., 2014, Ogawa et al., 2014, Mistry et al., 2009] assume that either the contact forces or the joint torques measurement are available, in the iCub we only have a distributed tactile system and a internal six-axis force-torque sensors. We propose a way of using internal six-axis force-torque sensor for estimation, and furthermore we demonstrate that the set of inertial parameters that are identifiable from the internal six-axis force-torque sensors are a superset of the one that we need to run the estimation algorithms presented in Chapter 4.

1.3 Technological Outcome

The modelling formalism and the estimation algorithms presented in this thesis enabled the development of an whole-body controller on the iCub robot, that has been used to showcase highly dynamical balancing [Pucci et al., 2016b] and has even permitted to the iCub to take part to an Italian Talent show as a participant [Talent, 2016]. The interested reader is referred to [Nori et al., 2015] for an integration paper describing the overall whole-body control architecture.

Furthermore, all the algorithms described in this thesis have been implemented in the open source C++ library iDynTree [Traversaro et al., 2017]. The iDynTree is complete with documentation for installing it and using it. Furthermore, the library is equipped with bindings with MATLAB, Octave, Python, Lua and Java to enable its use in a wide set of platforms, fulfilling all the disparate needs that may arise in a research environment.

Chapter 2

Rigid Body

In this chapter we introduce the dynamics of a *rigid body*. Existing (robot-related) literature on rigid body dynamics can be divided into two main categories. The first category contains work that use almost exclusively *spatial* 6D velocities as defined in Geometrical Mechanics such as [Featherstone, 2008, Featherstone and Orin, 2016, Jain, 2010, Murray et al., 1994]. The second category include work that use the 6D velocities that are a combination of the “classical” linear velocity and the angular velocity of the body [Spong et al., 2006, Siciliano et al., 2008, Chiaverini et al., 2016]. The connection between these two representations is well known to robotics researcher [Murray et al., 1994, Bruyninckx and De Schutter, 1996, Englsberger, 2016], but it was never explored in depth. Furthermore, it is a common source of confusion for newcomers to the field. In this chapter we treat all the commonly used representation of velocity, explaining the respective advantages and limitations, and the effect of choosing one representation or another on dynamics.

Furthermore, while most robotics textbooks on dynamics introduce the Newton-Euler equations as a given, we prefer to derive them from the basic principle of Lagrangian Dynamics. This methodology is consistent with what is typically done for fixed-based robots [Siciliano et al., 2008, Spong et al., 2006], and is useful to get an insight on the structure of Newton-Euler equations and *why* they are different from the classical Euler-Lagrange equations that describe the evolution of a system whose configuration space is a vector space.

The goal of this chapter is two-folded: while introducing the kinematics and dynamics of the rigid body, we also introduce a notation for describing kinematics and dynamics 6D quantities, as well as their coordinate trans-

formations.

The goal of this newly introduced notation is to be compact, not ambiguous, and in harmony with Lie Group formalism. The notation borrows from the well known notation introduced in [Featherstone, 2008], which is also used, with slight modifications, in [Featherstone and Orin, 2016]. This notation is, unfortunately, not fully in accordance with Lie group formalism used in, e.g., [Murray et al., 1994, Park et al., 1995, Kim, 2012], that is, however, less compact than [Featherstone, 2008], leading to long expressions when several rigid bodies are present. This chapter presents the frames notation, while the connection between the notation and the Lie group formalism is presented in Appendix A.

2.1 Overview of the notation

Remark 2.1. *Every chapter will begin with an overview of the notation used in that Chapter. Frequently, the notation used in a given chapter will be a simplified version of the full notation introduced in this chapter, to avoid overloading the text with an extremely complex notation. In some cases, the simplified notation can only be introduced for a single section. In that case, an overview of the simplified notation used in that section will open the section.*

Notation used through the thesis	
A, B	coordinate frames
p	an arbitrary point
o_B	origin of B
$[A]$	orientation frame associated to A
 $B[A]$	frame with origin o_B and orientation $[A]$
${}^A p$	coordinates of p w.r.t. to A
${}^A o_B$	coordinates of o_B w.r.t. to A
${}^A H_B$	homogeneous transformation from B to A
${}^A X_B$	velocity transformation from B to A
${}^C v_{A,B}$	6D velocity expressing the velocity of B wrt to A written in C
${}^C v_{A,B}^\wedge$	4×4 matrix representation of ${}^C v_{A,B}$
${}^C v_{A,B} \times$	6×6 matrix representation of the 6D velocity cross product
${}^C v_{A,B} \bar{\times}^*$	6×6 matrix representation of the dual cross product
${}^B f$	coordinates of the 6D force f w.r.t. B
${}^A X^B$	6D force transformation from B to A
$\langle {}^B f, {}^B v_{A,B} \rangle$	pairing between 6D force and velocity
${}^B \mathbb{M}_L$	6×6 inertia tensor of link (=rigid body) L expressed with respect to frame B

2.2 Math preliminaries

The following notation is used throughout the thesis.

- The set of real numbers is denoted by \mathbb{R} . Let u and v be two n -dimensional column vectors of real numbers, i.e. $u, v \in \mathbb{R}^n$, then their inner product is denoted as $u^T v$, with “ T ” the transpose operator.
- The identity matrix of dimension n is denoted $I_n \in \mathbb{R}^{n \times n}$; the zero column vector of dimension n is denoted $0_n \in \mathbb{R}^n$; the zero matrix of dimension $n \times m$ is denoted $0_{n \times m} \in \mathbb{R}^{n \times m}$.
- The set $\text{SO}(3)$ is the set of $\mathbb{R}^{3 \times 3}$ orthogonal matrices with determinant equal to one, namely

$$\text{SO}(3) := \{ R \in \mathbb{R}^{3 \times 3} \mid R^T R = I_3, \det(R) = 1 \}. \quad (2.1)$$

- The set $\mathfrak{so}(3)$, read *little* $\mathfrak{so}(3)$, is the set of 3×3 skew-symmetric matrices,

$$\mathfrak{so}(3) := \{ S \in \mathbb{R}^{3 \times 3} \mid S^T = -S \}. \quad (2.2)$$

- The set $\text{SE}(3)$ is defined as

$$\text{SE}(3) := \left\{ \begin{bmatrix} R & p \\ 0_{1 \times 3} & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \mid R \in \text{SO}(3), p \in \mathbb{R}^3 \right\}. \quad (2.3)$$

- The set $\mathfrak{se}(3)$ is defined as

$$\mathfrak{se}(3) := \left\{ \begin{bmatrix} \Omega & v \\ 0_{1 \times 3} & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \mid \Omega \in \mathfrak{so}(3), v \in \mathbb{R}^3 \right\}. \quad (2.4)$$

- Given the vector $w = (x; y; z) \in \mathbb{R}^3$, we define w^\wedge (read *w hat*) as the 3×3 *skew-symmetric* matrix

$$w^\wedge = \begin{bmatrix} x \\ y \\ z \end{bmatrix}^\wedge := \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \in \mathfrak{so}(3). \quad (2.5)$$

Given the *skew-symmetric matrix* $W = w^\wedge$, we define $W^\vee \in \mathbb{R}^3$ (read *W vee*) as

$$W^\vee = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}^\vee := \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^3. \quad (2.6)$$

Clearly, the vee operator is the inverse of the hat operator.

- Given a vector $v = (v; \omega) \in \mathbb{R}^6$, v and $\omega \in \mathbb{R}^3$, we define

$$v^\wedge = \begin{bmatrix} v \\ \omega \end{bmatrix}^\wedge := \begin{bmatrix} \omega^\wedge & v \\ 0_{1 \times 3} & 0 \end{bmatrix} \in \mathfrak{se}(3). \quad (2.7)$$

- Similarly to what was done for vectors in \mathbb{R}^3 few lines above, we define the vee operator as the inverse of the hat operator such that

$$\begin{bmatrix} \omega^\wedge & v \\ 0_{1 \times 3} & 0 \end{bmatrix}^\vee := \begin{bmatrix} v \\ \omega \end{bmatrix} = v \in \mathbb{R}^6. \quad (2.8)$$

- Given $A \in \mathbb{R}^{n \times m}$ and $B \in \mathbb{R}^{p \times q}$, we denote with \otimes the Kronecker product $A \otimes B \in \mathbb{R}^{np \times mq}$.

- Given $X \in \mathbb{R}^{m \times p}$, $\text{vec}(X) \in \mathbb{R}^{nm}$ denotes the column vector obtained by stacking the columns of the matrix X . In view of the definition of $\text{vec}(\cdot)$, it follows that

$$\text{vec}(AXB) = (B^\top \otimes A) \text{vec}(X). \quad (2.9)$$

2.3 Frame kinematics

2.3.1 Rigid Body Assumption

A rigid body is an idealization of a physical object, in which the deformation internal to the object are assumed to be negligible, i.e. the object is assumed to be *nondeformable*. While in the real physical world every object that is interacting with the external world is subject to a certain degree of deformation, in the study of certain mechanism such a humanoid robots, the rigid body assumption is useful as it permits to develop model that capture the *dominant dynamics* of the analyzed system, disregarding any aspect related to the internal compression and decompression of the object composing the system.

The kinematics of a rigid body are typically described by attaching to the rigid body a *frame*, defined as the combination of a point (called *origin*) and an *orientation frame* in the 3D space [De Laet et al., 2013, Spong et al., 2006]. For this reason in this section we will describe the kinematics of frames, and this will serve also for describing the kinematics of any rigid body.

2.3.2 Points and coordinate frames

We typically employ a capital letter to indicate a frame. Given a frame A , we will indicate with o_A its origin and with $[A]$ its orientation frame. Formally, we write this as $A = (o_A, [A])$.

Frames can be time varying. They can be used, e.g., to describe the position and orientation in space of a rigid body as time evolves. They are also used to express a coordinate system for a 6D force exchanged by two bodies or used to define a coordinate system to describe a robot task (like a frame attached to the center of mass and oriented as the inertial frame).

Newton's mechanics defines the set of *Inertial* frames. In this document, we usually indicate one inertial frame as with A (the *Absolute* frame). As common practice, for robots operating near the Earth surface, we will assume the frame A to be fixed to the world's surface, disregarding non-inertial effects due to the Earth's motion.

Coordinate vector of a point

Given a point p , its coordinates with respect to a frame $A = (o_A, [A])$ are collected in the *coordinate vector* ${}^A p$. The coordinate vector ${}^A p$ represents the coordinates of the 3D geometric vector $\vec{r}_{o_A, p}$ connecting the origin of frame A with the point p , pointing towards p , expressed in the orientation frame $[A]$, that is

$${}^A p := \begin{bmatrix} \vec{r}_{o_A, p} \cdot \vec{x}_A \\ \vec{r}_{o_A, p} \cdot \vec{y}_A \\ \vec{r}_{o_A, p} \cdot \vec{z}_A \end{bmatrix} \in \mathbb{R}^3, \quad (2.10)$$

with \cdot denoting the scalar product between two vectors and $\vec{x}_A, \vec{y}_A, \vec{z}_A$, the unit vectors defining the orientation frame $[A]$.

Change of orientation frame

Given two frames A and B , we will employ the notation

$${}^A R_B \in \text{SO}(3) \quad (2.11)$$

to denote the coordinate transformation from frame B to frame A . The coordinate transformation ${}^A R_B$ only depends on the relative orientation between the orientation frames $[A]$ and $[B]$, irrespectively of the position of the origins o_A and o_B .

Homogeneous transformation

To describe the position and orientation of a frame B with respect to another frame A , we employ the 4×4 homogeneous matrix

$${}^A H_B := \begin{bmatrix} {}^A R_B & {}^A o_B \\ 0_{1 \times 3} & 1 \end{bmatrix}. \quad (2.12)$$

Given a point p , the homogeneous transformation matrix ${}^A H_B$ can be also used to map the coordinate vector ${}^A p$ to ${}^B p$ as follows. Let ${}^A \bar{p}$ and ${}^B \bar{p}$ denote the *homogenous representation* of ${}^A p$ and ${}^B p$, respectively. That is, let ${}^A \bar{p} := ({}^A p; 1) \in \mathbb{R}^4$ and likewise for ${}^B \bar{p}$ (note that ; indicates row concatenation). Then

$${}^A \bar{p} = {}^A H_B {}^B \bar{p}, \quad (2.13)$$

which is the matrix form of ${}^A p = {}^A R_B {}^B p + {}^A o_B$. We refer the interested readers to [Murray et al., 1994, Chapter 2] for further details on homogeneous representation of rigid transformations.

2.3.3 6D Velocity vectors

In the following, given a point p and a frame A , we define

$${}^A\dot{p} := \frac{d}{dt} ({}^A p). \quad (2.14)$$

In particular, when p is the origin of a frame, e.g., $p = o_B$, we have

$${}^A\dot{o}_B = \frac{d}{dt} ({}^A o_B).$$

It is important to note that, by itself, expressions like \dot{o}_B or \dot{p} have *no* meaning. Similarly to (2.14), we also define

$${}^A\dot{R}_B := \frac{d}{dt} ({}^A R_B) \quad (2.15)$$

and

$${}^A\dot{H}_B := \frac{d}{dt} ({}^A H_B) = \begin{bmatrix} {}^A\dot{R}_B & {}^A\dot{o}_B \\ 0_{1 \times 3} & 0 \end{bmatrix}. \quad (2.16)$$

The relative velocity between a frame B with respect to a frame A can be represented by the time derivative of the homogenous matrix ${}^A H_B \in \text{SE}(3)$. A more *compact* representation of ${}^A\dot{H}_B$ can be obtained multiplying it by the inverse of ${}^A H_B$ on the left or on the right. In both cases, the result is an element of the $\mathfrak{se}(3)$ that will be called a *6D velocity*. Premultiplying on the left, one obtains

$$\begin{aligned} {}^A H_B^{-1} {}^A\dot{H}_B &= \begin{bmatrix} {}^A R_B^T & -{}^A R_B^{TA} o_B \\ 0_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^A\dot{R}_B & {}^A\dot{o}_B \\ 0_{1 \times 3} & 0 \end{bmatrix} \\ &= \begin{bmatrix} {}^A R_B^{TA} \dot{R}_B & {}^A R_B^{TA} \dot{o}_B \\ 0_{1 \times 3} & 0 \end{bmatrix}. \end{aligned} \quad (2.17)$$

Note that ${}^A R_B^{TA} \dot{R}_B$ appearing on the right hand side of (2.17) is skew symmetric. Define ${}^B v_{A,B}$ and ${}^B \omega_{A,B} \in \mathbb{R}^3$ so that

$${}^B v_{A,B} := {}^A R_B^{TA} \dot{o}_B, \quad (2.18)$$

$${}^B \omega_{A,B}^\wedge := {}^A R_B^{TA} \dot{R}_B. \quad (2.19)$$

The *left trivialized* velocity of frame B with respect to frame A is

$${}^B v_{A,B} := \begin{bmatrix} {}^B v_{A,B} \\ {}^B \omega_{A,B} \end{bmatrix} \in \mathbb{R}^6. \quad (2.20)$$

By construction,

$${}^B v_{A,B}^\wedge = {}^A H_B^{-1} {}^A \dot{H}_B. \quad (2.21)$$

Note the slight abuse of notation in using the hat operator \wedge in (2.19) and (2.21) that maps a vector into its corresponding matrix representation (respectively, from \mathbb{R}^3 to $\mathbb{R}^{3 \times 3}$ using (2.5) in (2.19) and from \mathbb{R}^6 to $\mathbb{R}^{4 \times 4}$ using (2.7) in (2.21)).

Specularly to what was done in (2.17), right multiplying ${}^A \dot{H}_B$ by the inverse of ${}^A H_B$ leads to

$$\begin{aligned} {}^A \dot{H}_B {}^A H_B^{-1} &= \begin{bmatrix} {}^A \dot{R}_B & {}^A \dot{o}_B \\ 0_{1 \times 3} & 0 \end{bmatrix} \begin{bmatrix} {}^A R_B^T & -{}^A R_B^{TA} o_B \\ 0_{1 \times 3} & 1 \end{bmatrix} \\ &= \begin{bmatrix} {}^A \dot{R}_B {}^A R_B^T & {}^A \dot{o}_B - {}^A \dot{R}_B {}^A R_B^{TA} o_B \\ 0_{1 \times 3} & 0 \end{bmatrix}. \end{aligned} \quad (2.22)$$

Define ${}^A v_{A,B}$ and ${}^A \omega_{A,B} \in \mathbb{R}^3$ as

$${}^A v_{A,B} := {}^A \dot{o}_B - {}^A \dot{R}_B {}^A R_B^{TA} o_B \quad (2.23)$$

$${}^A \omega_{A,B}^\wedge := {}^A \dot{R}_B {}^A R_B^T. \quad (2.24)$$

The *right trivialized* velocity of B with respect to A is then defined as

$${}^A v_{A,B} := \begin{bmatrix} {}^A v_{A,B} \\ {}^A \omega_{A,B}^\wedge \end{bmatrix} \in \mathbb{R}^6. \quad (2.25)$$

By construction,

$${}^A v_{A,B}^\wedge = {}^A \dot{H}_B {}^A H_B^{-1}. \quad (2.26)$$

The *right trivialized* and *left trivialized* representation of the velocity between two frames are general concepts, that can be applied to any pair of frames in the 3D space. When the frame A is an inertial frame and the B frame is rigidly attached to a body, as we will assume in this chapter, these velocities representations are also called *inertial velocity* and *body velocity*.

The mapping between ${}^A v_{A,B}$ and ${}^B v_{A,B}$ is trivial to express in matrix form, as it follows directly from their definitions that:

$${}^A v_{A,B}^\wedge = {}^A H_B {}^B v_{A,B}^\wedge {}^A H_B^{-1}$$

From this expression, it is clear that the “frame transformation” mapping between ${}^B v_{A,B}$ and ${}^A v_{A,B}$ is a linear mapping. By writing the 6D velocity in linear form:

$${}^A v_{A,B} = {}^A X_B {}^B v_{A,B}$$

with:

$${}^A X_B = {}^A X_B \left({}^A H_B \right) = \begin{bmatrix} {}^A R_B & {}^A o_B^\wedge {}^A R_B \\ 0_{3 \times 3} & {}^A R_B \cdot \end{bmatrix} \quad (2.27)$$

This is usually called the *adjoint* matrix.

Remark 2.2. Through out this thesis, given a quantity such as position, velocity, etc., we define the dot operator $(\dot{})$ as the total time derivative of the quantity. Hence, given a 6D velocity ${}^C v$ expressed in a frame C , the symbol ${}^C \dot{v}$ means

$${}^C \dot{v} = \frac{d}{dt} ({}^C v).$$

While this may be obvious and pedantic, let us recall that the dot operator $(\dot{})$ is defined differently in some of the robotics dynamics literature. For instance, given a 6D velocity ${}^C v$ expressed in a frame C , [Featherstone, 2008, Section 2.10] defines ${}^C \dot{v}$ as

$${}^C X_A \frac{d}{dt} ({}^A v),$$

where A is an arbitrary (and often hidden) inertial frame with respect to (w.r.t.) which the derivative is computed.

2.3.4 Adjoint matrix as a change of frame

The 6×6 transform is ubiquitous in multibody dynamics, and it is usually interpreted as the “change of frame” in which the frame velocity is expressed. Let’s assume that we have a frame \tilde{B} that is rigidly attached to the frame B , i.e. :

$${}^B \dot{H}_{\tilde{B}} = 0 \quad (2.28)$$

Noting that ${}^A H_{\tilde{B}} = {}^A H_B {}^B H_{\tilde{B}}$, ${}^A \dot{H}_{\tilde{B}} = {}^A \dot{H}_B {}^B H_{\tilde{B}}$, we can see that the mapping between the left-trivialized velocity of frame B and the left trivialized velocity of frame \tilde{B} is given by:

$$\tilde{B} v_{A,\tilde{B}}^\wedge = {}^A H_{\tilde{B}}^{-1} {}^A \dot{H}_{\tilde{B}} = \quad (2.29)$$

$$= {}^B H_{\tilde{B}}^{-1} {}^B v_{A,B}^\wedge {}^B H_{\tilde{B}}. \quad (2.30)$$

From which we have:

$$\tilde{B} v_{A,\tilde{B}} = \tilde{B} v_{A,B} = \tilde{B} X_B {}^B v_{A,B}.$$

Similarly, if we imagine to define a new frame \tilde{A} rigidly attached to our previous frame A , we will have:

$$\tilde{A}\mathbf{v}_{A,B} = \tilde{A}X_A{}^A\mathbf{v}_{A,B}. \quad (2.31)$$

This gives us a nice “interpretation” of the linear component of the left-trivialized velocity as the velocity of the origin of a frame fixed to the frame B that is instantaneously coincident with the frame A , that has been popularized by [Featherstone, 2008].

Mixed velocity

In the previous subsection we introduce the left-trivialized and right-trivialized velocity of a frame B with respect to a frame A . In the next chapter we will further discuss their use and properties, but for the moment we want to highlight a “peculiar” feature of both these velocities representation: in neither cases the linear part of the velocity is the derivative of the position vector of the origin of the frame B , i.e. ${}^A\dot{o}_B$. On the other hand, it is quite common for undergraduate physics textbooks or even in robotics [Siciliano et al., 2008] to represent the velocity of a rigid body as a 6D vector given by:

$$\begin{bmatrix} {}^A\dot{o}_B \\ {}^A\omega_{A,B} \end{bmatrix} \quad (2.32)$$

To simplify the description of the algorithms and concepts available in this thesis, we need a way to express this quantity coherently with the rest of the concepts introduced until now.

Fortunately this can be easily interpreted as a left or right trivialized velocity associated with the appropriate change of frame. In particular, we introduce the frame $B[A] := (o_B, [A])$, that is, the frame with the same origin of B and same orientation of A . We have then:

$${}^{B[A]}H_B = \begin{bmatrix} {}^A R_B & 0_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{bmatrix}, \quad (2.33)$$

and by expressing the velocity $\mathbf{v}_{A,B}$ in $B[A]$, we get:

$${}^{B[A]}\mathbf{v}_{A,B} = {}^{B[A]}X_B{}^B\mathbf{v}_{A,B} = \begin{bmatrix} {}^A R_B & 0 \\ 0 & {}^A R_B \end{bmatrix} \begin{bmatrix} {}^B R_A {}^A \dot{o}_B \\ {}^B \omega_{A,B} \end{bmatrix} = \begin{bmatrix} {}^A \dot{o}_B \\ {}^A \omega_{A,B} \end{bmatrix}. \quad (2.34)$$

In [Bruyninckx and De Schutter, 1996, Murray et al., 1994, Englsberger, 2016], (2.34) is referred to as the *hybrid* velocity of frame B with respect to

frame A . To avoid confusion with hybrid systems theory, we will call (2.34) the *mixed velocity* of frame B with respect to frame A (*mixed* as it has both the flavors of a left trivialized velocity for the linear velocity part and of a right trivialized velocity for the angular velocity part).

Table 2.1 collects the definitions of the different representation of the 6D velocity, and the conversion formulas to convert one representation into another.

2.4 Frame acceleration

The acceleration of a frame (attached to a rigid body) is a quantity for which the definition is far from being obvious, as already noted in literature [Featherstone, 2001]. Before introducing the representation for a rigid body acceleration most used in literature, and their properties, we will discuss the cross product for 6D velocities, a useful operator to study the properties of the different accelerations.

2.4.1 The cross product on \mathbb{R}^6 (\times)

Equation (2.21) can be rewritten as

$${}^A\dot{H}_B = {}^A H_B {}^B v_{A,B}^\wedge. \quad (2.35)$$

By time differentiation of (2.27), it can be shown that a similar formula holds for ${}^A\dot{X}_B$, namely, that

$${}^A\dot{X}_B = {}^A X_B {}^B v_{A,B} \times \quad (2.36)$$

with ${}^B v_{A,B} \times$ defined as

$${}^B v_{A,B} \times := \begin{bmatrix} {}^B \omega_{A,B}^\wedge & {}^B v_{A,B}^\wedge \\ 0_{3 \times 3} & {}^B \omega_{A,B}^\wedge \end{bmatrix}. \quad (2.37)$$

We will refer to (2.37) as the matrix representation of the *cross product on \mathbb{R}^6* .

Remark 2.3. *The connection between the cross product on \mathbb{R}^6 introduced here and its equivalent concepts in the Lie group language is discussed in the appendix, in particular in Subsection A.2.2.*

Basic properties of the cross product

Equation (2.37) defines a cross product between vectors of \mathbb{R}^6 , with the classical anticommutative property

$${}^C \mathbf{v}_{A,B} \times {}^C \mathbf{v}_{D,E} = - {}^C \mathbf{v}_{D,E} \times {}^C \mathbf{v}_{A,B}. \quad (2.38)$$

As a direct consequence of anticommutativity is

$${}^C \mathbf{v}_{A,B} \times {}^C \mathbf{v}_{A,B} = 0_{6 \times 1}. \quad (2.39)$$

2.4.2 Acceleration representations and their relation

By direct derivation with respect to time of the left-trivialized, right-trivialized and mixed 6D velocity, we obtain the respective accelerations, that are described in detail in Table 2.2. All the conversions between the accelerations are also detailed in Table 2.2, and make an extensive use of the 6D cross product introduced in this section. Please remember that the definition of the dot notation (\cdot) is clarified in Remark 2.2.

In the case that the frame A is an inertial frame, some alternative representation of the acceleration between a frame B and the inertial frame A are used in literature, that are described hereafter.

2.4.3 “Sensor” acceleration

We define the “sensor” acceleration as:

$$\alpha_{A,B} = {}^B X_{B[A]} {}^B \dot{\mathbf{v}}_{A,B} = \begin{bmatrix} {}^B R_A {}^A \ddot{\mathbf{o}}_B \\ {}^B \dot{\omega}_{A,B} \end{bmatrix} \quad (2.40)$$

This acceleration is used for example in [Siciliano et al., 2008]. Its relationship with inertial sensing will be discussed in Subsection 4.3.2.

2.4.4 “Proper” acceleration

It is widespread, in the literature on multibody dynamics applied to robotics, to sometimes refer to a vector of the acceleration minus the gravitational acceleration. The same symbols is sometime used to refer both to the acceleration with respect earth-fixed inertial frame and this acceleration minus gravity. To avoid confusion, in this thesis we will always refer to this as *proper acceleration*, as opposed to the *coordinate acceleration*, consistently with the nomenclature used in relativity theory [Fraundorf, 1996].

Name	Symbol	Definition	Left-Trivialized	Right-Trivialized	Mixed
Left-Trivialized	$B_{\mathbf{v}_A,B}$	$\begin{bmatrix} {}^A R_B^{TA} \dot{o}_B \\ \left({}^A R_B^{TA} \dot{R}_B \right)^\vee \end{bmatrix}$	${}^B \mathbf{v}_{A,B}$	${}^B X_A {}^A \mathbf{v}_{A,B}$	${}^B X_{B[A]} {}^B [A] \mathbf{v}_{A,B}$
Right-Trivialized	$A_{\mathbf{v}_A,B}$	$\begin{bmatrix} {}^A \dot{o}_B - {}^A R_B {}^A R_B^{TA} o_B \\ \left({}^A \dot{R}_B {}^A R_B^T \right)^\vee \end{bmatrix}$	${}^A X_B {}^B \mathbf{v}_{A,B}$	${}^A \mathbf{v}_{A,B}$	${}^A X_{B[A]} {}^B [A] \mathbf{v}_{A,B}$
Mixed	$B[A]_{\mathbf{v}_A,B}$	$\begin{bmatrix} {}^A \dot{o}_B \\ \left({}^A \dot{R}_B {}^A R_B^T \right)^\vee \end{bmatrix}$	${}^B [A] X_B {}^B \mathbf{v}_{A,B}$	${}^B [A] X_A {}^A \mathbf{v}_{A,B}$	${}^B [A] \mathbf{v}_{A,B}$

Table 2.1 Conversion rules between the different representations of frame 6D velocity.

\equiv	${}^B \dot{\mathbf{v}}_{A,B}$	${}^A \dot{\mathbf{v}}_{A,B}$	${}^B [A] \dot{\mathbf{v}}_{A,B}$	${}^A \dot{\mathbf{v}}_{A,B}$	${}^B [A] \dot{\mathbf{v}}_{A,B}$	${}^B [A] \dot{\mathbf{v}}_{A,B}$	α
$B \dot{\mathbf{v}}_{A,B}$	${}^B \dot{\mathbf{v}}_{A,B}$	${}^B X_A {}^A \dot{\mathbf{v}}_{A,B}$	${}^B X_{B[A]} {}^B [A] \dot{\mathbf{v}}_{A,B} + \begin{bmatrix} {}^B R_A {}^A \dot{o}_B \times {}^B \omega_{A,B} \\ 0_{3 \times 1} \end{bmatrix}$	${}^B X_{B[A]} {}^B [A] \dot{\mathbf{v}}_{A,B} + \begin{bmatrix} {}^B R_A {}^A \dot{o}_B \times {}^B \omega_{A,B} \\ 0_{3 \times 1} \end{bmatrix}$	${}^B X_A {}^A \dot{\mathbf{v}}_{A,B} + \begin{bmatrix} {}^B R_A {}^A \dot{o}_B \times {}^B \omega_{A,B} \\ 0_{3 \times 1} \end{bmatrix}$	${}^B R_A {}^A \dot{o}_B \times {}^B \omega_{A,B}$	$\alpha +$
$A \dot{\mathbf{v}}_{A,B}$	${}^A X_B {}^B \dot{\mathbf{v}}_{A,B}$	${}^A \dot{\mathbf{v}}_{A,B}$	${}^A X_{B[A]} {}^B [A] \dot{\mathbf{v}}_{A,B} + \begin{bmatrix} {}^A \dot{o}_B \times {}^A \omega_{A,B} \\ 0_{3 \times 1} \end{bmatrix}$	${}^A \dot{\mathbf{v}}_{A,B}$	${}^A X_B {}^B \dot{\mathbf{v}}_{A,B} + \begin{bmatrix} {}^A \dot{o}_B \times {}^A \omega_{A,B} \\ 0_{3 \times 1} \end{bmatrix}$	${}^A \dot{o}_B \times {}^A \omega_{A,B}$	$\alpha +$
$B[A] \dot{\mathbf{v}}_{A,B}$	$B[A] X_B {}^B \dot{\mathbf{v}}_{A,B} - \begin{bmatrix} {}^A \dot{o}_B \times {}^A \omega_{A,B} \\ 0_{3 \times 1} \end{bmatrix}$	$B[A] X_A {}^A \dot{\mathbf{v}}_{A,B} - \begin{bmatrix} {}^A \dot{o}_B \times {}^A \omega_{A,B} \\ 0_{3 \times 1} \end{bmatrix}$	${}^B [A] \dot{\mathbf{v}}_{A,B}$	${}^B X_A {}^A \dot{\mathbf{v}}_{A,B} - \begin{bmatrix} {}^B R_A {}^A \dot{o}_B \times {}^B \omega_{A,B} \\ 0_{3 \times 1} \end{bmatrix}$	${}^B X_{B[A]} {}^B [A] \dot{\mathbf{v}}_{A,B}$	${}^B X_B \alpha$	$B[A] X_B \alpha$
α	$B \dot{\mathbf{v}}_{A,B} - \begin{bmatrix} {}^B R_A {}^A \dot{o}_B \times {}^B \omega_{A,B} \\ 0_{3 \times 1} \end{bmatrix}$	$B X_A {}^A \dot{\mathbf{v}}_{A,B} - \begin{bmatrix} {}^B R_A {}^A \dot{o}_B \times {}^B \omega_{A,B} \\ 0_{3 \times 1} \end{bmatrix}$	${}^B X_{B[A]} {}^B [A] \dot{\mathbf{v}}_{A,B}$	${}^B X_{B[A]} {}^B [A] \dot{\mathbf{v}}_{A,B}$	${}^B X_{B[A]} {}^B [A] \dot{\mathbf{v}}_{A,B}$	${}^B X_{B[A]} {}^B [A] \dot{\mathbf{v}}_{A,B}$	α

Table 2.2 Conversion rules between the different representations of frame 6D acceleration.

We define first the “proper” acceleration using the “sensor” representation, as:

$$\alpha_{A,B}^g := \alpha_{A,B} - \begin{bmatrix} {}^B R_A {}^A g \\ 0_{3 \times 1} \end{bmatrix} \quad (2.41)$$

Note that this is the acceleration that can be obtained by an inertial measurement unit aligned with B , as the linear part of it is the output of a linear accelerometer, and the angular part is the derivative of the output of a gyroscope. This is thoroughly discussed in Section 4.3.2.

For all other accelerations, we define:

$${}^B a^g := {}^B \dot{v} - \begin{bmatrix} {}^B R_A {}^A g \\ 0_{3 \times 1} \end{bmatrix}, \quad (2.42a)$$

$${}^A a^g := {}^A \dot{v} - \begin{bmatrix} {}^A g \\ 0_{3 \times 1} \end{bmatrix}, \quad (2.42b)$$

$${}^B[A] a^g := {}^B[A] \dot{v} - \begin{bmatrix} {}^A g \\ 0_{3 \times 1} \end{bmatrix} \quad (2.43)$$

as respectively the *left-trivialized*, *right-trivialized* and *mixed* proper acceleration. Note that all the conversion rules presented in Table 2.2 still hold for proper acceleration in the presented form.

2.4.5 Illustrative Example: the Spinning Wheel

To see a concrete example of the definition just presented, we use a spinning wheel, depicted in Figure 2.1. An inertial frame A is rigidly attached to the ground, while there is a moving frame B attached to the external part of a wheel of radius r spinning around the axis:

$${}^A a = {}^B a = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

passing in the point o_C . Furthermore, the system is affected by a uniform gravitational field with the gravity acceleration vector, whose expression in the A orientation is:

$${}^A g = \begin{bmatrix} 0 \\ -|g| \\ 0 \end{bmatrix}.$$

The location of the spinning wheel w.r.t. the inertial frame is given by the angle θ . Assuming that for $\theta = 0$ the orientation of A and B coincide, we

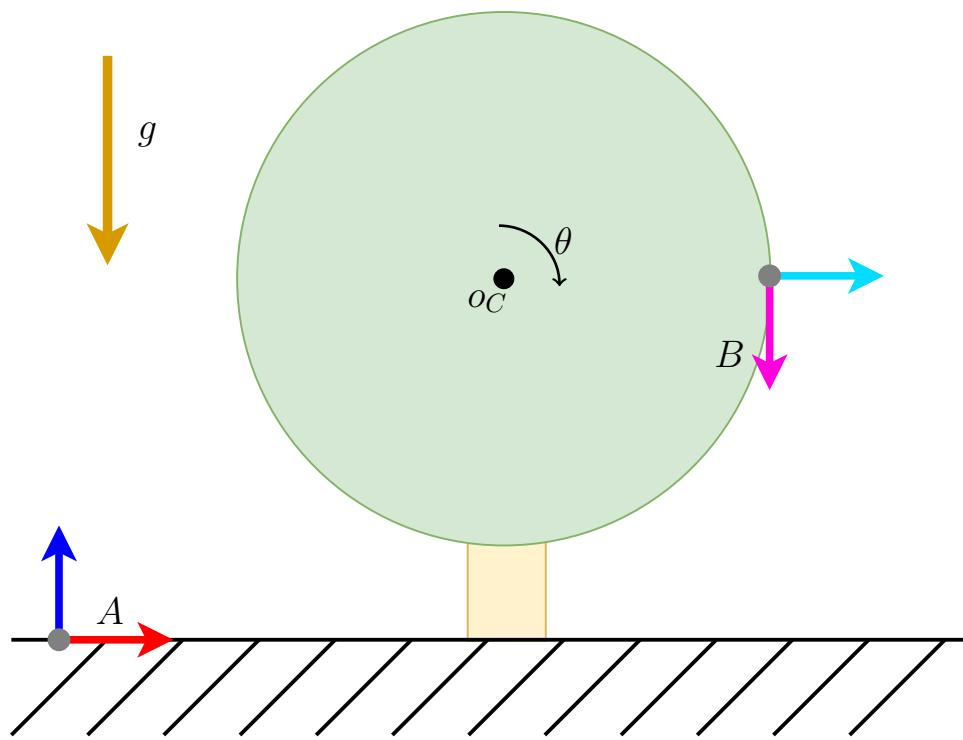


Fig. 2.1 Spinning wheel, with the gravity acceleration g , the inertial frame A and the moving frame B at $\theta = -\frac{\pi}{2}$.

can define the transform between A and B as (where we define $H := {}^A H_B$ and $R := {}^A R_B$ to avoid overloading the notation).

$$H(\theta) = \begin{bmatrix} R(\theta) & {}^A o_C + R(\theta) \begin{bmatrix} 0 \\ r \end{bmatrix} \\ 0_{1 \times 3} & 1 \end{bmatrix}, \quad R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.44)$$

Taking the derivatives w.r.t. to time, we have that:

$$\begin{aligned} \dot{H}(\theta, \dot{\theta}) &= \begin{bmatrix} \dot{R}(\theta, \dot{\theta}) & \dot{R}(\theta, \dot{\theta}) \begin{bmatrix} 0 \\ r \end{bmatrix} \\ 0_{1 \times 3} & 0 \end{bmatrix}, \\ \dot{R}(\theta, \dot{\theta}) &= \begin{bmatrix} -\sin(\theta) & -\cos(\theta) & 0 \\ \cos(\theta) & -\sin(\theta) & 0 \\ 0 & 0 & 0 \end{bmatrix} \dot{\theta}, \\ R^T(\theta) \dot{R}(\theta) &= \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \dot{\theta}, \\ \dot{R}(\theta) R^T(\theta) &= \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \dot{\theta}. \end{aligned}$$

Using the definition of left-trivialized, right-trivialized and mixed veloc-

ity summarized in Table 2.1 we then have:

$$\begin{aligned}
{}^B \mathbf{v}_{A,B} &= \begin{bmatrix} {}^B v_{A,B} \\ {}^B \omega_{A,B} \end{bmatrix} = \begin{bmatrix} R^T A \dot{o}_B \\ (R^T \dot{R})^\vee \end{bmatrix} = \begin{bmatrix} -r \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \dot{\theta}, \\
{}^A \mathbf{v}_{A,B} &= \begin{bmatrix} {}^A v_{A,B} \\ {}^A \omega_{A,B} \end{bmatrix} = \begin{bmatrix} {}^A \dot{o}_B - \dot{R} R^T A o_B \\ (\dot{R} R^T)^\vee \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} -\cos(\theta)r \\ -\sin(\theta)r \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \wedge {}^A o_B \\ 0 \\ 0 \\ 1 \end{bmatrix} \dot{\theta}, \\
{}^{B[A]} \mathbf{v}_{A,B} &= \begin{bmatrix} {}^{B[A]} v_{A,B} \\ {}^{B[A]} \omega_{A,B} \end{bmatrix} = \begin{bmatrix} {}^A \dot{o}_B \\ (\dot{R} R^T)^\vee \end{bmatrix} = \begin{bmatrix} -\cos(\theta)r \\ -\sin(\theta)r \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \dot{\theta} + .
\end{aligned}$$

Similarly, for the accelerations defined in Table 2.2 we then have:

$$\begin{aligned}
{}^B \ddot{\mathbf{v}}_{A,B} &= \begin{bmatrix} -r \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \ddot{\theta} \\
{}^A \ddot{\mathbf{v}}_{A,B} &= \begin{bmatrix} [-\cos(\theta)r] \\ [-\sin(\theta)r] \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} {}^A \omega_B \wedge \begin{bmatrix} [\sin(\theta)r] \\ [-\cos(\theta)r] \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} {}^A \omega_B \wedge \begin{bmatrix} [-\cos(\theta)r] \\ [-\sin(\theta)r] \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \dot{\theta}^2, \\
{}^{B[A]} \ddot{\mathbf{v}}_{A,B} &= \begin{bmatrix} -\cos(\theta)r \\ -\sin(\theta)r \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \ddot{\theta} + \begin{bmatrix} \sin(\theta)r \\ -\cos(\theta)r \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \dot{\theta}^2, \\
\alpha_{A,B} &= \begin{bmatrix} -r \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \ddot{\theta} + \begin{bmatrix} 0 \\ -r \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \dot{\theta}^2 = \begin{bmatrix} -r\ddot{\theta} \\ -r\dot{\theta}^2 \\ 0 \\ 0 \\ 0 \\ \ddot{\theta} \end{bmatrix}.
\end{aligned}$$

2.5 Force-Torque covectors

In classical mechanics, the interaction between a rigid body and the environment is described by *interaction* forces. While these forces can be described as a system of forces acting on a finite number of contact points it is common to represent the effect of these interaction forces as a 6D force-torque vector, in which the first three elements are the sum of all the contact forces, while the last three coordinate are the sum of the moments of these forces with respect to a given point in space. Similarly to the 6D velocity case, also for the 6D force-torque can be said to be represented in different frames, in which the *origin* of the frame is the point with respect to which the moment is taken and the *orientation* is the one in which the forces and moments are

expressed. In particular we indicate the coordinates of a 6D force f with respect to frame B with

$${}_B f := \begin{bmatrix} {}_B f \\ {}_B \tau \end{bmatrix} \in \mathbb{R}^6. \quad (2.45)$$

Note that the frame B is simply used to indicate the coordinate frame with respect to which the 6D force f is expressed in coordinates and there is no necessity for the 6D force f to also be applied to, e.g., the rigid body (if any) to which B is attached. Similarly to what we did for a 6D velocities, we can define a linear map to change the coordinates of a 6D force from a frame B to another frame A . This coordinate transformation is indicated with ${}_A X^B$ and written as

$${}_A f = {}_A X^B {}_B f. \quad (2.46)$$

The mapping ${}_A X^B$ is actually induced by the velocity transformation (2.27) (why this is the case will be explained below) and is related to ${}^B X^A$ via the definition

$${}_A X^B := {}^B X_A^T. \quad (2.47)$$

It is important to realize that (2.47) is such to make the following identity (of power) hold

$$\langle {}_B f, {}^B v_{A,B} \rangle = \langle {}_A f, {}^A v_{A,B} \rangle, \quad (2.48)$$

where f can be interpreted as a 6D force applied to a rigid body to which the moving frame B is rigidly attached and A as the absolute inertial frame.

2.5.1 The dual cross product on \mathbb{R}^6 ($\bar{\times}^*$)

The time derivative of the 6D force coordinate transformation ${}_A X^B$ has an expression that is dual to velocity coordinate transformation ${}_A X_B$ given in (2.36). Indeed, straightforward computations lead to obtain

$${}_A \dot{X}^B = {}_A X^B {}_B v_{A,B} \bar{\times}^* \quad (2.49)$$

where the (matrix representation of the) dual cross product $\bar{\times}^*$ is defined by

$${}^B v_{A,B} \bar{\times}^* := \begin{bmatrix} {}^B \omega_{A,B}^\wedge & 0_{3 \times 3} \\ {}^B v_{A,B}^\wedge & {}^B \omega_{A,B}^\wedge \end{bmatrix}. \quad (2.50)$$

It is worth noting that (2.50) is obtained from (2.37) by simply transposing this latter expression and taking the negative value of the result: a fact that is also encoded in the symbol $\bar{\times}^*$, where the overline sign has been chosen to represent the minus sign and the star the transpose operation (more formally, the adjoint of a linear map, typically indicated with a star). The dual cross product (2.50) takes one 6D velocity and one 6D force and return one 6D force (as opposed to the cross product (2.37) that takes as input two 6D velocity and return one 6D velocity); this is also the reason why the sub- and superscripts in (2.49) makes sense: when ${}_A\dot{X}^B$ is applied to a 6D force ${}_Bf$ expressed in B , the dual cross product between ${}^Bv_{A,B}$ and the 6D force will return a 6D force expressed in B that can then be converted into a 6D force expressed in A via ${}_AX^B$. It is also straightforward to prove that

$${}_AX^{BB}v_{A,B}\bar{\times}^* = {}^Av_{A,B}\bar{\times}^* {}_AX^B. \quad (2.51)$$

Remark 2.4. *The connection between the cross product on \mathbb{R}^6 introduced here and its equivalent concepts in the Lie group language is discussed in the appendix, in particular in Subsection A.2.3.*

2.6 Rigid Body Dynamics

To discuss rigid body dynamics, we tipically only are interested in two frames: an inertial frame A and a body-fixed frame B . To avoid confusion in the readers, throughout this section we then use a simplified notation,

summarized in the next notation table.

Simplified Notation, valid for Section 2.6	
A	Inertial frame.
B	Frame rigidly attached to the body.
$o := {}^A o_B$	Origin of the body w.r.t. to the inertial frame.
i	
$R := {}^A R_B$	Rotation of the body w.r.t. to the inertial frame.
$H := {}^A H_B$	Pose of the body w.r.t. to the inertial frame.
$v = \begin{bmatrix} v \\ \omega \end{bmatrix} := {}^B v_{A,B}$	Left-trivialized velocity of the body.
$p := {}^A p$	Generic point of the body expressed in the inertial frame A .
$r := {}^B p$	Generic point of the body expressed in the body frame B .
$\rho(\cdot)$	Density function, taking in input points expressed in the body frame B .

All the results of system in classical mechanics can be obtained by applying the principle of least action, that states that the trajectories of a mechanical systems are the extremum of a trajectory-dependent quantity called *action*.

2.6.1 Review of Lagrangian Dynamics: the point mass

Before approaching the lagrangian dynamics of the rigid body, we review the basic concepts of lagrangian mechanics using a simple example, a point mass.

The configuration of a point mass with respect to an inertial frame A is given by its position coordinate vector ${}^A p \in \mathbb{R}^3$. To avoid overloading the notation, in the remainder of this subsection we will indicate the position of the point mass simply as $p := {}^A p \in \mathbb{R}^3$. The velocity of the point mass is given by the derivative w.r.t to time of the point position, i.e. $\dot{p} := {}^A \dot{p}$.

Assuming that the point mass lies in a uniform gravitational field, and the effect of the point mass on the gravitational field is negligible, the Lagrangian for a point mass m is the difference between the Kinetic Energy

$K(p, \dot{p})$ and the Potential (Gravitational) Energy $U(p)$:

$$L(p, \dot{p}) = K(p, \dot{p}) - U(p), \quad (2.52a)$$

$$K(p, \dot{p}) = \frac{1}{2}m|\dot{p}|^2, \quad (2.52b)$$

$$U(p) = mg^T p. \quad (2.52c)$$

where $g \in \mathbb{R}^3$ is the gravitational acceleration vector of the uniform field. The action of a given trajectory is defined as:

$$S[p(\cdot)] = \int_{t_0}^{t_1} L(p(t), \dot{p}(t)) dt$$

The Principle of Least Action states that the trajectory performed by such a system is the one that minimize the action. Such a variational problem can be demonstrated to have the same solution of the Euler-Lagrange differential equations [Bullo and Lewis, 2005]:

$$\frac{\partial}{\partial p} L(p, \dot{p}) - \frac{d}{dt} \frac{\partial}{\partial \dot{p}} L(p, \dot{p}) = 0.$$

In the point mass case we have:

$$\frac{\partial}{\partial p} L(p, \dot{p}) = \frac{\partial}{\partial p} U(p) = mg, \quad (2.52d)$$

$$\frac{\partial}{\partial \dot{p}} L(p, \dot{p}) = \frac{\partial}{\partial \dot{p}} K(p, \dot{p}) = m\dot{p}, \quad (2.52e)$$

$$\frac{d}{dt} \frac{\partial}{\partial \dot{p}} L(p, \dot{p}) = m\ddot{p}. \quad (2.52f)$$

From which we have:

$$m\ddot{p} - mg = 0_{3 \times 1}. \quad (2.53)$$

The left term of the equations is 0 only if the only external force acting on the point mass is the gravitational force described by the potential $U(p)$. If forces not described by the potential are acting on the point mass, the equation is modified to include this forces acting on the point:

$$m\ddot{p} - mg = f. \quad (2.54)$$

2.6.2 Rigid Body Lagrangian Dynamics

In the previous subsection we reviewed the Lagrangian Dynamics formalism and the principle of Least Action for a simple point mass. In this section we will see how the same concepts applies, with the appropriate changes, to the Rigid Body case. But first we need to formally introduce the concept of *rigid body*, that was informally presented in Section 2.3.1.

Definition 2.1 (Rigid Body). *A rigid body is a mathematical abstraction describing an arbitrary distribution of mass in the 3D space, that is fixed with respect to a given frame, that we call the body frame B .*

Definition 2.2 (Volumetric Mass Density). *The mass distribution of a rigid body in space is described by a time-invariant density function, that maps each 3D point (expressed in the frame B) to its density.*

$$\rho(\cdot) : \mathbb{R}^3 \mapsto \mathbb{R}_{\geq 0}. \quad (2.55)$$

The mass $m(V)$ enclosed in a given 3D volume V is given by the integral of the density over such a volume:

$$m(V) = \iiint_V \rho(r) dr. \quad (2.56)$$

Note that if the density domain was defined as the points in the 3D space expressed in the *inertial A frame*, the density function would have not been constant, but it would also explicitly depend on time. As the density of the rigid body is constant in the body frame B , the pose of the *rigid body* can be fully represented by the homogeneous transformation between the *body frame B* and the *inertial frame A*, that for simplicity in this section we will indicate as $H = {}^A H_B$, with $R = {}^A R_B$ and $o = {}^A o_B$.

The relation between a point attached to the rigid body expressed in the absolute frame ${}^A p$, that for simplicity we refer as p and in the body frame ${}^B p$ (for simplicity r) is simply given as:

$$\begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} R & o \\ 0_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} r \\ 1 \end{bmatrix} = \begin{bmatrix} Rr + o \\ 1 \end{bmatrix} \quad (2.57)$$

And its velocity is obtained as:

$$\begin{bmatrix} \dot{p} \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{R} & \dot{o} \\ 0_{1 \times 3} & 0 \end{bmatrix} \begin{bmatrix} r \\ 1 \end{bmatrix} = \begin{bmatrix} \dot{R}r + \dot{o} \\ 0 \end{bmatrix} \quad (2.58)$$

Similarly to the point mass case (2.52) the kinetic energy of a rigid body is obtained by integrating all the contributions to the kinetic energy of each point p of the rigid body:

$$\frac{1}{2} \iiint_{\mathbb{R}^3} \rho(R^T(p - o)) |\dot{p}|^2 dp. \quad (2.59)$$

While the potential (gravitational) energy is obtained by integrating all the contributions to the potential (gravitational) energy of each point p of the rigid body:

$$\iiint_{\mathbb{R}^3} \rho(R^T(p - o)) g^T p dr. \quad (2.60)$$

To highlight the dependency of the lagrangian on the state of the rigid body H, \dot{H} , we apply the change of variables $p = Rr + o$, $\dot{p} = \dot{R}r + \dot{o}$ and obtain the following definition of the Rigid Body Lagrangian.

Definition 2.3 (Rigid Body Lagrangian). *The Lagrangian function for a rigid body B with pose $H \in SE(3)$ and velocity \dot{H} is:*

$$L(H, \dot{H}) = K(H, \dot{H}) - U(H), \quad (2.61a)$$

$$K(H, \dot{H}) = \frac{1}{2} \iiint_{\mathbb{R}^3} \rho(r) |\dot{R}r + \dot{o}|^2 dr, \quad (2.61b)$$

$$U(H) = \iiint_{\mathbb{R}^3} \rho(r) g^T (Rr + o) dr. \quad (2.61c)$$

There are two main differences between the Lagrangian of a point particle (2.52) and of a rigid body (2.61). First the configuration of a point particle $p \in \mathbb{R}^3$ is a element of a vector space, while the state of the rigid body is an element of a *matrix Lie group* $H = \begin{bmatrix} R & o \\ 0_{1 \times 3} & 1 \end{bmatrix} \in SE(3)$. The second difference is that for the point mass, the parameter describing the inertia of a particle is a single lumped single parameter, the mass of the point m . For the rigid body the inertia information is contained in the continuous density function $\rho(\cdot)$.

The first difference is addressed using the *left-trivialized velocity* or a rigid body, while the second difference is addressed by introducing the *inertial parameters*, a set of 10 parameters fully describing all the inertial properties of a rigid body.

Between the different ways we could transform \dot{H} in a 6D vector we choose the left-trivialized velocity because we can directly apply the Euler-Poincarè equations as will be necessary in Theorem 2.1.

To reduce the complexity of equations, in the next section we will refer to the left-trivialized velocity ${}^B v_{A,B}$ as v :

$$v = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} R^T \dot{\phi} \\ (R^T \dot{R})^\vee \end{bmatrix}, \quad \dot{H} = H v^\wedge = \begin{bmatrix} R\omega^\wedge & Rv \\ 0_{3 \times 1} & 0 \end{bmatrix}. \quad (2.62)$$

We can express the Lagrangian (2.61) in function of the *left-trivialized* velocity. We refer to the Lagrangian written w.r.t. to the left-trivialized velocity as the *left-trivialized Lagrangian* $l(H, v)$.

$$l(H, v) = L(H, H v^\wedge) = k(H, v) - U(H), \quad (2.62a)$$

$$\begin{aligned} k(H, v) &= \frac{1}{2} \iiint_{\mathbb{R}^3} \rho(r) |R\omega^\wedge r + Rv|^2 dr = \\ &= \frac{1}{2} \iiint_{\mathbb{R}^3} \rho(r) |R(\omega^\wedge r + v)|^2 dr = \\ &= \frac{1}{2} \iiint_{\mathbb{R}^3} \rho(r) |\omega^\wedge r + v|^2 dr. \end{aligned} \quad (2.62b)$$

Remark 2.5. Through the left-trivialization, i.e. introducing a dependency on v rather than on \dot{H} , we remove the dependency on H from the kinetic energy, an so we can write the trivialized Kinetic energy simply as $k(v)$.

Even if simplified, the left-trivialized Lagrangian still depends explicitly on the density function. We can simplify its expression by depending on a fixed number of functional of the density function, that are classically called *inertial parameters*.

Proposition 2.1. The left-trivialized Lagrangian of a rigid body can be written as:

$$l(H, v) = k(v) - U(H), \quad (2.63a)$$

$$k(v) = \frac{1}{2} v^T \mathbb{M} v \quad (2.63b)$$

$$U(H) = [g^T \ 0] H \begin{bmatrix} mc \\ m \end{bmatrix} \quad (2.63c)$$

where $m \in \mathbb{R}$ is the total mass of the body, defined as:

$$m = \iiint_{\mathbb{R}^3} \rho(r) dr, \quad (2.64)$$

$c \in \mathbb{R}^3$ is the center of mass of the body, defined as:

$$c = \frac{\iiint_{\mathbb{R}^3} r \rho(r) dr}{\iiint_{\mathbb{R}^3} \rho(r) dr} = \frac{\iiint_{\mathbb{R}^3} r \rho(r) dr}{m} \quad (2.65)$$

$I \in \mathbb{R}^{3 \times 3}$ is the 3D inertia matrix of the body, defined as:

$$I = - \iiint_{\mathbb{R}^3} \rho(r) (r^\wedge)^2 dr \in \mathbb{R}^{3 \times 3} \quad (2.66)$$

and $\mathbb{M} \in \mathbb{R}^{6 \times 6}$ is the 6D inertia matrix of the body, defined as:

$$\mathbb{M} = \begin{bmatrix} \iiint_{\mathbb{R}^3} \rho(r) dr 1_3 & -(\iiint_{\mathbb{R}^3} r \rho(r) dr)^\wedge \\ (\iiint_{\mathbb{R}^3} r \rho(r) dr)^\wedge & -\iiint_{\mathbb{R}^3} \rho(r) (r^\wedge)^2 dr \end{bmatrix} = \begin{bmatrix} m & -(mc)^\wedge \\ (mc)^\wedge & I \end{bmatrix}. \quad (2.67)$$

Proof. We can isolate the terms that depend on the *density* and the integration variable r . For the gravitational energy from (2.62) we have:

$$U(H) = g^T \left(R \iiint_{\mathbb{R}^3} r \rho(r) dr + o \iiint_{\mathbb{R}^3} \rho(r) dr \right) = \begin{bmatrix} g \\ 0 \end{bmatrix}^T H \begin{bmatrix} \iiint_{\mathbb{R}^3} r \rho(r) dr \\ \iiint_{\mathbb{R}^3} \rho(r) dr \end{bmatrix}.$$

As from (2.64)-(2.65) we have that $\iiint_{\mathbb{R}^3} r \rho(r) dr = mc$ and $\iiint_{\mathbb{R}^3} \rho(r) dr = m$, we obtain a compact expression for the gravitational energy of a rigid body in a uniform gravitational field:

$$U(H) = \begin{bmatrix} g \\ 0 \end{bmatrix}^T H \begin{bmatrix} mc \\ m \end{bmatrix}.$$

Regarding the kinetic energy from (2.62b) one has:

$$k(v) = \frac{1}{2} \iiint_{\mathbb{R}^3} \rho(r) \left| v^\wedge \begin{bmatrix} r \\ 1 \end{bmatrix} \right|^2 dr. \quad (2.68)$$

By noting that the argument of the norm can be written as:

$$v^\wedge \begin{bmatrix} r \\ 1 \end{bmatrix} = [1_3 \quad -r^\wedge] v$$

it is possible to write the kinetic energy as a quadratic form of the left-trivialized velocity v :

$$\begin{aligned} k(v) &= \frac{1}{2} \iiint_{\mathbb{R}^3} \rho(r) v^T \begin{bmatrix} 1_3 \\ r^\wedge \end{bmatrix} \begin{bmatrix} 1_3 & -r^\wedge \end{bmatrix} v dr = \\ &= \frac{1}{2} v^T \iiint_{\mathbb{R}^3} \rho(r) \begin{bmatrix} 1_3 & -r^\wedge \\ r^\wedge & -(r^\wedge)^2 \end{bmatrix} dr v = \\ &= \frac{1}{2} v^T \mathbb{M} v. \end{aligned}$$

□

Remark 2.6. Note that while it is common practice to define the center of mass as in (2.65), $U(H)$ is more easily expressed as a linear function of mc . $mc \in \mathbb{R}^3$ is usually called first moment of mass, and it will be discussed in depth in Chapter 6.

Theorem 2.1 (Euler-Poincaré Equations for Rigid Body Dynamics). *Given a time interval $[0, T]$ the trajectory $H(\cdot)$ of a rigid body in a uniform gravitational field is the one that minimizes the action:*

$$S[H(\cdot)] = \int_0^T L(H(t), \dot{H}(t)) dt \quad (2.69)$$

where the rigid body Lagrangian $L(H, \dot{H})$ is defined in (2.61).

In particular the resulting equations of motion are given by:

$$\dot{H} = H v^\wedge, \quad (2.70a)$$

$$\mathbb{M}\dot{v} + v\bar{\times}^*\mathbb{M}v = \mathbb{M} \begin{bmatrix} R^T g \\ 0_{3 \times 1} \end{bmatrix} \quad (2.71)$$

where \mathbb{M} is the 6D inertia matrix defined in (2.67) and $v\bar{\times}^*$ is defined in (2.50).

The proof to Theorem 2.70 is given in Subsection A.3.2, because it require the necessary background in the matrix Lie group theory.

Lemma 2.1. *The equations of motion of a rigid body in a uniform gravitational field under the influence of additional non-gravitational force-torques are:*

$$\mathbb{M}\dot{v} + v\bar{\times}^*\mathbb{M}v = \mathbb{M} \begin{bmatrix} R^T g \\ 0_{3 \times 1} \end{bmatrix} + {}_B f^x \quad (2.72)$$

where ${}_B f^x$ is the sum of all the external force-torques acting on the body, expressed in the body frame B .

From this computation we can now write the Newton-Euler equations using the *left-trivialized* representation:

$$\mathbb{M}\dot{v} + v\bar{\times}^*\mathbb{M}v = \mathbb{M} \begin{bmatrix} R^T g \\ 0_{3 \times 1} \end{bmatrix} \quad (2.73)$$

Non-conservative forces can be considered in this model by appropriately modifying the Lagrangian [Bullo and Lewis, 2005], to obtain the following equation:

$$\mathbb{M}\dot{v} + v\bar{\times}^*\mathbb{M}v = \mathbb{M} \begin{bmatrix} R^T g \\ 0_{3 \times 1} \end{bmatrix} + {}_B f^x \quad (2.74)$$

Where \mathbf{f}^x is the external force-torque exerted by the environment on the rigid body, expressed in the B frame for consistency with the rest of the equations.

The same equation can be expressed in a more compact way using the *proper* acceleration:

$$\mathbb{M} \left(\dot{\mathbf{v}} - \begin{bmatrix} R^T g \\ 0_{3 \times 1} \end{bmatrix} \right) + \mathbf{v} \bar{\times}^* \mathbb{M} \mathbf{v} = {}_B \mathbf{f}^x \quad (2.75)$$

2.7 Newton-Euler equations in the different representations

Applying the accelerations and velocity transforms presented in Table 2.1 and Table 2.2, we can easily transform the Newton-Euler equation to be expressed with respect to the chosen acceleration representation. Furthermore if velocity and acceleration are written with respect to a frame, we will write also the external force-torque w.r.t. to the same frame.

2.7.1 Left-trivialized

Using the notation introduced in this thesis in extended form, (2.74) can be written as:

$${}_B \mathbb{M}_B {}^B \dot{\mathbf{v}}_{A,B} + {}^B \mathbf{v}_{A,B} \bar{\times}^* {}_B \mathbb{M}_B {}^B \mathbf{v}_{A,B} = {}_B \mathbb{M}_B \begin{bmatrix} {}^A R_B^{TA} g \\ 0_{3 \times 1} \end{bmatrix} + {}_B \mathbf{f}^x \quad (2.76)$$

2.7.2 Right-trivialized

$${}_A \mathbb{M}_B {}^A \dot{\mathbf{v}}_{A,B} + {}^A \mathbf{v}_{A,B} \bar{\times}^* {}_A \mathbb{M}_B {}^A \mathbf{v} = {}_A \mathbb{M}_B \begin{bmatrix} {}^A g \\ 0_{3 \times 1} \end{bmatrix} + {}_A \mathbf{f}^x. \quad (2.77)$$

While this equation may seem similar to (2.76) the main difference is that ${}_B \mathbb{M}_B$ is a fixed quantity while ${}_A \mathbb{M}_B = {}_A X^B {}_B \mathbb{M}_B {}^B X_A$ that is a time-varying quantity that depends on the position of the rigid body.

2.7.3 Mixed

The dynamics using mixed acceleration can be written as:

$$\begin{aligned} {}_{B[A]} \mathbb{M}_B {}^{B[A]} \dot{\mathbf{v}}_{A,B} + \begin{bmatrix} 0_{3 \times 1} \\ {}_A \omega_{A,B} \end{bmatrix} \bar{\times}^* {}_{B[A]} \mathbb{M}_B \begin{bmatrix} 0_{3 \times 1} \\ {}_A \omega_{A,B} \end{bmatrix} = \\ {}_{B[A]} \mathbb{M}_B \begin{bmatrix} {}^A g \\ 0_{3 \times 1} \end{bmatrix} + {}_{B[A]} \mathbf{f}^x. \end{aligned} \quad (2.78)$$

2.7.4 Sensor

The dynamics using sensor acceleration can be written as:

$${}_B\mathbb{M}_B \alpha_{A,B} + \begin{bmatrix} 0_{3 \times 1} \\ {}_B\omega_{A,B} \end{bmatrix} \bar{\times} {}^* {}_B\mathbb{M}_B \begin{bmatrix} 0_{3 \times 1} \\ {}_B\omega_{A,B} \end{bmatrix} = {}_B\mathbb{M}_B \begin{bmatrix} {}^A R_B^T g \\ 0_{3 \times 1} \end{bmatrix} + {}_B f^x. \quad (2.79)$$

The equation in this form can compactly written using the *proper sensor acceleration* $\alpha_{A,B}^g$, defined in (2.41) :

$${}_B\mathbb{M}_B \alpha_{A,B}^g + \begin{bmatrix} 0_{3 \times 1} \\ {}_B\omega_{A,B} \end{bmatrix} \bar{\times} {}^* {}_B\mathbb{M}_B \begin{bmatrix} 0_{3 \times 1} \\ {}_B\omega_{A,B} \end{bmatrix} = {}_B f^x. \quad (2.80)$$

2.8 Rigid Body Total Momentum and its relation with Dynamics

Given a rigid body B with a density $\rho(\cdot)$, we define its *total* momentum w.r.t. to a frame A expressed in a frame C as:

$${}_C h_{A,B} := {}_C X^A \int_{\mathbb{R}^3} \rho({}^A R_B^T ({}^A p - {}^A o_B)) \begin{bmatrix} {}^A \dot{p} \\ {}^A \dot{p} \wedge {}^A p \end{bmatrix} d^A p = \quad (2.81)$$

$$= {}_C X^A \int_{\mathbb{R}^3} \rho(r) \begin{bmatrix} {}^A \omega_{A,B}^\wedge r + {}^A \dot{o}_B \\ \left({}^A \omega_{A,B}^\wedge r + {}^A \dot{o}_B \right)^\wedge ({}^A R_B r + {}^A o_B) \end{bmatrix} dr \quad (2.82)$$

Note that in this definition, while A and C are just frames, B is a rigid body.

A more convenient expression for the total momentum is given in the next theorem.

Lemma 2.2. *The total momentum defined in (2.81) can be equivalently expressed as:*

$${}_C h_{A,B} := {}_C X^B {}_B \mathbb{M}_B {}^B v_{A,B}. \quad (2.83)$$

The momentum is a quantity with several interesting properties. One of it is the following alternative formulation of the Newton Euler equations.

Lemma 2.3. *The time derivative of the total momentum expressed in the inertial frame A is equal to the sum of the potential and contact 6D forces applied on the body, i.e. one can rewrite the right-trivialized rigid body dynamics in (2.77) as:*

$${}_A \dot{h}_{A,B} = {}_A X^B {}_B \mathbb{M}_B \begin{bmatrix} R^T g \\ 0_{3 \times 1} \end{bmatrix} + {}_A f^x \quad (2.84)$$

This is formulation of the Newton-Euler equations as the derivative of the total momentum expressed in the inertial frame is how the Rigid Body Dynamics is typically introduced in robotics dynamics literature [Featherstone, 2008].

Note that the Newton-Euler equations can be defined also using the *mixed* representation, assuming that the origin of the frame L coincides with the center of mass of the robot, as stated in the next property.

Lemma 2.4. *Assuming that the origin of a link frame G is coincident with the center of mass of the body, i.e. ${}^G c_B = 0$, the time derivative of the total momentum expressed in the mixed frame $G[A]$ is equal to the sum of the potential and contact 6D forces applied on the body, i.e.:*

$${}_{G[A]} \dot{h}_{A,B} = {}_{G[A]} X^B{}_B \mathbb{M}_B \begin{bmatrix} {}^A R_G^{T,A} g \\ 0_{3 \times 1} \end{bmatrix} + {}_{G[A]} f^x = \begin{bmatrix} m^A g \\ 0_{3 \times 1} \end{bmatrix} + {}_{G[A]} f^x. \quad (2.85)$$

Chapter 3

Multi Body

In existing control-related robotics literature [Siciliano et al., 2008], the dynamics of fixed base structures is introduced through the Lagrangian formalism and the Euler-Lagrange equations, that is typically used for having a greater insight on the structure of the dynamics. Instead, the dynamics of floating base structure is either derived using the Lagrangian formalism with singular representation of the rotation of the base [Wieber et al., 2016], or by taking the Newton-Euler equation for a rigid body as a given[Featherstone, 2008, Jain, 2010]. In this thesis instead we present also free-floating dynamics equations using the Lagrangian formalism, using known results from the geometrical mechanics literature [Marsden and Ratiu, 2013] such as Euler-Poincaré and Hamel equations. While the resulting equation of motions are (clearly) identical to the one that can be obtained assuming as given the Newton-Euler equations, such a presentation of the Multi Body Dynamics give the interested reader more insight on the underlying mechanical principles.

Another main contribution with respect to most existing literature is the fact that we model floating systems without assuming any preferred base link or frame, and we show how the kinematics and dynamics of the robot arise by choosing different base link. In particular, we show how the equations of motion associated with different *base links* can be obtained as a nonlinear change of variables of the robot state. We also show that this change of variables can be used to express the robot dynamics as a combination of the “internal” and the “centroidal dynamics”, as introduced in [Orin et al., 2013] and extended in [Garofalo et al., 2015]. Attempts to generalize the floating-base systems’ equations of motion irrespective from the base frame choice can be found in [van Oort, 2011, Chapter 3] and in [Jain, 2010, Chapter 17,

Section 3.6]. The main theoretical drawbacks of these works, which have been largely ignored by the humanoid control literature, is the assumption that the base frame is rigidly attached to one of the links of the system, that is instead superseded in this thesis.

While this thesis does not directly include results related to control, its modelling content has been used in several results achieved on balancing control of iCub. Control related results based on the modelling developed in this thesis can be found in [Nava et al., 2016, Pucci et al., 2016a, Dafarra et al., 2016].

3.1 Composition of relative and absolute velocity

A key aspect in multibody systems is how to compute the *absolute* position, velocity and acceleration of a frame D given the *absolute* position, velocity and acceleration of a frame B and the *relative* position, velocity and acceleration of the frame D w.r.t. to B .

At the position level, this problem is simply solved by the composition law of homogeneous transformation: given ${}^A H_B$ and ${}^B H_D$, ${}^A H_D$ can simply be obtained by matrix multiplication:

$${}^A H_D = {}^A H_B {}^B H_D. \quad (3.1)$$

By time differentiation, we obtain a similar relation at velocity and acceleration level:

$${}^A \dot{H}_D = {}^A \dot{H}_B {}^B H_D + {}^A H_B {}^B \dot{H}_D, \quad (3.2)$$

$${}^A \ddot{H}_D = {}^A \ddot{H}_B {}^B H_D + 2 {}^A \dot{H}_B {}^B \dot{H}_D + {}^A H_B {}^B \ddot{H}_D. \quad (3.3)$$

This same relation, but written with respect to the 6D velocity and acceleration in their different representation are given in the next subsection.

Lemma 3.1 (Left-Trivialized Velocity and Acceleration Composition). *Given the transforms ${}^A H_B$, ${}^B H_D$ and their left-trivialized velocities ${}^B v_{A,B}$, ${}^D v_{B,D}$ and accelerations ${}^B \dot{v}_{A,B}$, ${}^D \dot{v}_{B,D}$, the left-trivialized velocity and acceleration of the composed transform ${}^A H_D = {}^A H_B {}^B H_D$ are given by:*

$${}^D v_{A,D} = {}^D X_B {}^B v_{A,B} + {}^D v_{B,D} \quad (3.4a)$$

$${}^D \dot{v}_{A,D} = {}^D X_B {}^B \dot{v}_{A,B} + {}^D \dot{v}_{B,D} + {}^D v_{A,B} \times {}^D v_{B,D}. \quad (3.4b)$$

Lemma 3.2 (Right-Trivialized Velocity and Acceleration Composition). *Given the transforms ${}^A H_B$, ${}^B H_D$ and their right-trivialized velocities ${}^A v_{A,B}$, ${}^B v_{B,D}$*

and accelerations ${}^A\dot{\mathbf{v}}_{A,B}$, ${}^B\dot{\mathbf{v}}_{B,D}$, the right-trivialized velocity and acceleration of the composed transform ${}^A H_D = {}^A H_B {}^B H_D$ are given by:

$${}^A \mathbf{v}_{A,D} = {}^A \mathbf{v}_{A,B} + {}^A X_B {}^B \mathbf{v}_{B,D} \quad (3.5a)$$

$${}^A \dot{\mathbf{v}}_{A,D} = {}^A \dot{\mathbf{v}}_{A,B} + {}^A X_B {}^B \dot{\mathbf{v}}_{B,D} + {}^A \mathbf{v}_{A,B} \times {}^A \mathbf{v}_{B,D}. \quad (3.5b)$$

Lemma 3.3 (Mixed Velocity Composition). *Given the transforms ${}^A H_B$, ${}^B H_D$ and their mixed velocities ${}^{B[A]} \mathbf{v}_{A,B}$, ${}^{D[B]} \mathbf{v}_{B,D}$ and accelerations ${}^{B[A]} \dot{\mathbf{v}}_{A,B}$, ${}^{D[B]} \dot{\mathbf{v}}_{B,D}$, the mixed velocity and acceleration of the composed transform ${}^A H_D = {}^A H_B {}^B H_D$ are given by:*

$${}^{D[A]} \mathbf{v}_{A,D} = {}^{D[A]} X_{B[A]} {}^{B[A]} \mathbf{v}_{A,B} + {}^{D[A]} X_{D[B]} {}^{D[B]} \mathbf{v}_{B,D} \quad (3.6a)$$

$$\begin{aligned} {}^{D[A]} \dot{\mathbf{v}}_{A,D} = & {}^{D[A]} X_{B[A]} {}^{B[A]} \dot{\mathbf{v}}_{A,B} + {}^{D[A]} X_{D[B]} {}^{D[B]} \dot{\mathbf{v}}_{B,D} + \\ & + \left[2 {}^A \omega_{A,B} \times {}^A R_B {}^B \dot{o}_D + {}^A \omega_{A,B} \times ({}^A \omega_{A,B} \times {}^A R_B {}^B o_D) \right. \\ & \left. + {}^A \omega_{A,B} \times {}^A R_B {}^B \omega_{B,D} \right]. \end{aligned} \quad (3.6b)$$

Remark 3.1. *The complicated expression of propagation of acceleration in the mixed case (3.6) is one of the reason why the right-trivialized and the left-trivialized representation are typically preferred in describing in a compact way multibody algorithms, as discussed in [Featherstone, 2001, 2010].*

Remark 3.2. *Differently from the left-trivialized and right-trivialized accelerations, the propagation of the mixed acceleration does not require any information on the linear part of the mixed velocity of any couple of frame involved in the kinematic propagation. This property holds also for the sensor acceleration, and is a useful property when performing estimation using only inertial sensing, as will be explained in Subsection 4.4.3.*

Lemma 3.4 (Sensor Acceleration Composition). *Given the transforms ${}^A H_B$, ${}^B H_D$ and their sensor accelerations $\alpha_{A,B}$, $\alpha_{B,D}$, the sensor acceleration of the composed transform ${}^A H_D = {}^A H_B {}^B H_D$ is given by:*

$$\alpha_{A,D} = {}^D X_B \alpha_{A,B} + \alpha_{B,D} + \quad (3.7)$$

$$+ \left[{}^D R_B \left[2 ({}^B \omega_{A,B} \times {}^B \dot{o}_D) + {}^B \omega_{A,B} \times ({}^B \omega_{A,B} \times {}^B o_D) \right] \right]. \quad (3.8)$$

3.2 Joints

The *joint* is a connection that constraints the relative motion between two rigid bodies. The modelling of joints it is a foundation of modern mechanics [Denavit and Hartenberg, 1964], and it is an active field of research even in

the first years of the 21st century [Seth et al., 2010]. In the following section we will review the main properties of the joint necessary in the following of the thesis.

The number of degrees of freedom (dofs) of a joint is the number of relative degrees of freedom between the two body that are left unconstrained by the joint. In general the number of degrees of freedom of a joint can range from 0 to 6, and also be time variant. The relative position of two bodies connected by a joint is in general an element of a manifold whose local dimension is equal to the number of DOFs of the joint.

Assumption 3.1. *For the sake of simplicity, in this thesis we only consider 0-dof and 1-dof joints. In particular we will only considered 1-dof joints, whose configuration can considered an element of \mathbb{R} .*

3.2.1 One Degree of Freedom Joints

The transform between two bodies B and D connected by a joint are is fully determined by the function mapping the joint position $\theta \in \mathbb{R}$ to the relative pose of the two links connected by the joint:

$${}^B H_D(\theta) : \mathbb{R} \mapsto \text{SE}(3).$$

The relative velocity and acceleration of D w.r.t. to B is related to the time derivatives of joint configuration θ :

$$\frac{d {}^B H_D(\theta)}{dt} = \frac{d {}^B H_D(\theta)}{d\theta} \dot{\theta} \quad (3.9)$$

As discussed in the previous chapter, the that typically the relative position between two bodies is expressed using a 6D velocity representation.

Lemma 3.5. *Give a one degree-of-freedom joint described by the transform function ${}^B H_D(\theta)$, the left-trivialized, right-trivialized and mixed relative velocity can be computed as:*

$${}^D v_{B,D} = {}^D s_{B,D}(\theta) \dot{\theta}, \quad {}^D s_{B,D} := \left(({}^B H_D(\theta))^{-1} \frac{d {}^B H_D(\theta)}{d\theta} \right)^{\vee}, \quad (3.10a)$$

$${}^B v_{B,D} = {}^B s_{B,D}(\theta) \dot{\theta}, \quad {}^B s_{B,D} = \left(\frac{d {}^B H_D(\theta)}{d\theta} ({}^B H_D(\theta))^{-1} \right)^{\vee}, \quad (3.10b)$$

$${}^{D[B]} v_{B,D} = {}^{D[B]} s_{B,D}(\theta) \dot{\theta}, \quad {}^{D[B]} s_{B,D} = \left[\begin{array}{c} \frac{d}{d\theta} ({}^B o_D(\theta)) \\ \left(\frac{d {}^B R_D(\theta)}{d\theta} ({}^B R_D(\theta))^T \right)^{\vee} \end{array} \right] \quad (3.10c)$$

The vector ${}^D\mathbf{s}_{B,D} \in \mathbb{R}^6$ obeys the same transformation rules of 6D velocity, as it is the mapping between the relative 6D velocity of the two bodies connected by the joint and the joint velocity and it is known as *joint motion subspace vector*.

As the definition of the joint is symmetric w.r.t. to the two links B and D connected by the joint, they can be inverted in the joint motion subspace definition, regardless in the frame C in which the joint motion subspace is expressed, resulting in:

$${}^C\mathbf{s}_{D,B} = -{}^C\mathbf{s}_{B,D}. \quad (3.11)$$

Assumption 3.2. *In this thesis we will always assume that the left-trivialized, the right-trivialized and the mixed joint motion subspace vector are independent of the joint configuration, i.e. :*

$$\frac{d {}^D\mathbf{s}_{B,D}}{d\theta} = 0_{6 \times 1}, \quad (3.12)$$

$$\frac{d {}^B\mathbf{s}_{B,D}}{d\theta} = 0_{6 \times 1}, \quad (3.13)$$

$$\frac{d {}^{D[B]}\mathbf{s}_{B,D}}{d\theta} = 0_{6 \times 1}. \quad (3.14)$$

$$(3.15)$$

While this may seem an arbitrary complex requirement, it is actually observed by most simple joints used in robotics, such as the *revolute* or the *prismatic* joints.

Under Assumption 3.2 we then can write that the relative accelerations of two bodies connected by a joint are:

$${}^D\ddot{\mathbf{v}}_{B,D} = {}^D\mathbf{s}_{B,D}\dot{\theta}, \quad (3.16)$$

$${}^B\ddot{\mathbf{v}}_{B,D} = {}^B\mathbf{s}_{B,D}\dot{\theta}, \quad (3.17)$$

$${}^{D[B]}\ddot{\mathbf{v}}_{B,D} = {}^{D[B]}\mathbf{s}_{B,D}\dot{\theta}. \quad (3.18)$$

Revolute Joint

A classical example of 1-dof joint is the revolute joint, whose use is widespread in humanoid robotics.

Definition 3.1. *Given a revolute joints with an axis $a \in \mathbb{R}^3, |a| = 1$ that connects two bodies B and D , with ${}^B H_D(0) = 1_4$, the joint transform is given by:*

$${}^B H_D(\theta) = \begin{bmatrix} {}^B R_D(\theta) & 0_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{bmatrix}, \quad {}^B R_D(\theta) = 1_3 + \cos(\theta)a^\wedge + \sin(\theta)(a^\wedge)^2. \quad (3.19)$$

Remark 3.3. For a revolute joint defined as in Definition 3.1, the 3D joint axis is a is expressed in the same way in the frame B and in the frame D , as we have that, given that $a^\wedge a = 0_{3 \times 1}$:

$${}^B a = {}^B R_A {}^A a = 1_3 + \cos(\theta) a^\wedge + \sin(\theta) (a^\wedge)^2 {}^A a = {}^A a.$$

Lemma 3.6. Given a revolute joints defined as in (3.19), the left-trivialized, right-trivialized and mixed joint motion subspaces coincide, are constant and are given by:

$${}^D s_{B,D} = {}^B s_{B,D} = {}^D [B] s_{B,D} = \begin{bmatrix} 0_{3 \times 1} \\ a \end{bmatrix} \quad (3.20)$$

The lemma follows by applying Lemma 3.5 to (3.19).

This result can be generalized to a general revolute joint introducing two constant transforms ${}^{\tilde{B}} H_B$ and ${}^D H_{\tilde{D}}$. The joint connecting the two frames B and D that coincide for $\theta = 0$ can be transformed in the generic revolute joint connecting \tilde{B} and \tilde{D} using the following transformations:

$${}^{\tilde{B}} H_{\tilde{D}}(\theta) = {}^{\tilde{B}} H_B {}^B H_D(\theta) {}^D H_{\tilde{D}}, \quad (3.20a)$$

$${}^{\tilde{D}} s_{\tilde{B}, \tilde{D}} = {}^{\tilde{D}} X_D {}^D s_{B,D}, \quad (3.20b)$$

$${}^{\tilde{B}} s_{\tilde{B}, \tilde{D}} = {}^{\tilde{B}} X_B {}^B s_{B,D}, \quad (3.20c)$$

$${}^{\tilde{D}} [{}^{\tilde{B}}] s_{\tilde{B}, \tilde{D}} = {}^{\tilde{D}} [{}^{\tilde{B}}] X_D {}^D [B] s_{B,D}. \quad (3.20d)$$

3.2.2 Fixed joints

A (0-dof) fixed joint connecting two bodies B and D is fully determined by the (constant) homogeneous transform between the two bodies ${}^B H_D$. Given that the two bodies are rigidly attached, all the time derivatives of this transform are equal to zero, i.e. ${}^B \dot{H}_D \equiv 0_{4 \times 4}$, and one has:

$${}^D v_{B,D} = 0_{6 \times 1}, \quad {}^D \dot{v}_{B,D} = 0_{6 \times 1}, \quad (3.21a)$$

$${}^B v_{B,D} = 0_{6 \times 1}, \quad {}^B \dot{v}_{B,D} = 0_{6 \times 1}, \quad (3.21b)$$

$${}^D [B] v_{B,D} = 0_{6 \times 1}, \quad {}^D [B] \dot{v}_{B,D} = 0_{6 \times 1}. \quad (3.21c)$$

3.3 Modelling of Multibody Systems

This section recalls notation, state definition, and equations of motion associated with *free-floating* mechanical systems.

3.3.1 Topology

To mathematically describe the topology, i.e. how joints interconnect the links, of a multibody system, we need to define use some concepts from *graph* theory. While the use of graph theory to model the multibody topology is a widely studied subject we remark that in the presented formalism we always treat the graph induced by the multibody structure as a *undirected* graph, while existing literature [Jain, 2011, McPhee, 1996] model multibody systems as *directed* graphs.

Definition 3.2 (Multibody System). *A multibody system is composed by a set of n_L rigid bodies –called links– interconnected by n_J mechanisms –called joints– constraining the relative motion of a pair of links. From the mathematical point of view, a multibody system is represented by a undirected graph, a couple $(\mathfrak{L}, \mathfrak{J})$, where \mathfrak{L} is the set of the links (the vertices of the graph), while \mathfrak{J} is the set of joints, modeled as sets containing two distinct links (the undirected edges).*

Definition 3.3 (Path). *A path from a link B to a link D of length d is an ordered sequence of d links (L_1, L_2, \dots, L_d) such that:*

$$L_1 = B, \quad L_d = D \quad \forall i \in 1, 2, \dots, d \quad \{L_i, L_{i+1}\} \in \mathfrak{J}. \quad (3.22)$$

Furthermore, we define with $\pi_B(L)$ the unique set of links that lie on the path connecting L with B .

Assumption 3.3 (Link Frames). *We assume that each link $L \in \mathfrak{L}$ is associated with a link-fixed frame, called link frame. In the sequel, we often refer to the frame attached to link L simply as L .*

Assumption 3.4 (Joint's Degrees of Freedom). *We assume that each joint possess either zero or one degree-of-freedom. For each joint $\{E, F\} \in \mathfrak{J}$, we define a function $DOFs(\cdot)$ that returns the number of degree-of-freedom of the joint:*

$$DOFs(\cdot) : \mathfrak{J} \mapsto \{0, 1\}.$$

Definition 3.4 (Acyclic Graph). *A multibody system is acyclic if there are no loops in its structure. In mathematically terms, a multibody system is acyclic if for every ordered pair of links (B, C) there is a unique path connecting B to C .*

Assumption 3.5 (Acyclic Graph). *All the multibody systems considered in this thesis are acyclic.*

Definition 3.5 (Internal Degrees of Freedom). *The internal degrees of freedom of given multibody system, indicated with n , are defined as the sum of all the degrees-of-freedom of all the joints contained in the multibody system:*

$$n := \sum_{J \in \mathfrak{J}} DOFs(J) \quad (3.23)$$

The assumption of acyclicity of the multibody systems mean that all considered systems will be represented by an *undirected tree*. In general an undirected tree can be represented as a *directed tree* by selecting one link as the *root* or *base* of the tree. If we indicate the selected base as B , the direction of any given joint given the base link B is defined as *toward* the base B .

To fully model this base-induced directionality, we define some helper functions.

Definition 3.6 (Parent Link). *Assuming that $B \in \mathfrak{L}$ is the selected base link, the parent function:*

$$\lambda_B(\cdot) : \mathfrak{L} - B \mapsto \mathfrak{L} \quad (3.24)$$

is defined as the function that maps every link (except the base B) to its unique parent. As, by definition, the base link B is the root of the directed tree, no parent is defined for it.

Definition 3.7 (Ancestor Link). *Given a base link $B \in \mathfrak{L}$, the n th ancestor link of $L \in \mathfrak{L}$ is defined as the power of the parent function:*

$$\lambda_B^n(L) = \underbrace{\lambda_B(\dots \lambda_B(L))}_n. \quad (3.25)$$

Definition 3.8 (Children Links). *Given a base link $B \in \mathfrak{L}$, we define the set of the children of a link L as all links whose parent is L :*

$$\mu_B(L) := \{D \in \mathfrak{L} \mid \lambda_B(D) = L\}. \quad (3.26)$$

Definition 3.9 (Subtree Links). *Given a base link $B \in \mathfrak{L}$, we define the set of the subtree starting at link L as $\gamma_B(L)$:*

$$\gamma_B(L) := \{D \in \mathfrak{L} \mid \exists n \in \mathbb{N} \text{ s.t. } \lambda_B^n(D) = L\} \cup \{L\}. \quad (3.27)$$

Definition 3.10 (Neighbor Links). *We define with $\aleph(L)$ the set of all links that are adjacent to link L , i.e. the neighbors of L :*

$$\aleph(L) = \{D \in \mathfrak{L} \mid \{D, L\} \in \mathfrak{J}\}. \quad (3.28)$$

Note that the set of neighbors does not depend on the definition of a specific base.

Lemma 3.7. *Given a link L , for every $B \neq L$ the set of neighbors of L $\aleph(L)$ is the union of the singleton containing the parent link $\lambda_B(L)$ and the children set $\mu_B(L)$:*

$$\aleph(L) = \{\lambda_B(D)\} \cup \mu_B(L). \quad (3.29)$$

Lemma 3.8. *Given a link L the set of neighbors of L $\aleph(L)$ is also the children set using L as a base $\nu_L(L)$:*

$$\aleph(L) = \mu_L(L). \quad (3.30)$$

3.3.2 Numbering

In the previous section, we introduced the basic elements composing a multi-body systems, i.e. links and joints, as abstract elements. For composing link and joint related quantities such as position, velocity or torques in familiar linear algebra concepts such as matrices or vectors, it is convenient to associate each link and joint to unique natural number, that we call *index* of the Link or Joint, defined in the following.

Definition 3.11 (Link Serialization). *Each link is associated to a natural number by the numbering bijective function $\text{LinkIndex}(\cdot)$:*

$$\text{LinkIndex}(\cdot) : \mathfrak{L} \rightarrow 1, 2, \dots, n_L. \quad (3.31)$$

Definition 3.12 (Joint Serialization). *Each link is associated to a natural number by the numbering bijective function $\text{JointIndex}(\cdot)$:*

$$\text{JointIndex}(\cdot) : \mathfrak{J} \rightarrow 1, 2, \dots, n_J. \quad (3.32)$$

Similarly to the link and joint cases, it is also convenient to associate to each (internal) degree-of-freedom of the system a natural number. However, the fact that each joint can have either 0 or 1 dof complicates the definition, given in the following.

Definition 3.13 (Degree-of-Freedom Serialization). *Each non-fixed joint is associated with a natural number describing the position of the DOF associated with that in vectors with n size describing internal position, velocities of the robot by the $\text{DOFsOffset}(\cdot)$ function.*

$$\text{DOFsOffset}(\cdot) : \mathfrak{J} - \mathfrak{J}_0 \rightarrow 1, 2, \dots \quad (3.33)$$

Where $\mathfrak{J}_0 := \{J \in \mathfrak{J} \mid \text{DOFs}(J) = 0\}$ is the set of fixed joints.

Remark 3.4. In classical approaches this numbering is also used to encode some of the hierarchical properties of a spanning directed tree induced by the choice of an arbitrary floating base link. In particular the regular numbering is usually adopted [Featherstone, 2008, Jain, 2010], in which every parent link has a number lower than the ones of its children. This aid the description of some algorithms for multibody dynamics but has the downside of coupling the chosen numbering with the chosen base link. As in this chapter we explore and use in depth the concept of changing the base frame of the system, we opted to keep the numbering and the topology of the system separated, such that internal quantities such the robot shape are independent of the choice of the base frame.

Definition 3.14 (Degrees of Freedom in a Path). *The set of DOFs that belong to a path $\pi_B(L)$ is defined as $\pi_B^{DOF}(L)$:*

$$\pi_B^{DOF}(L) := \{i \in \mathbb{N} \mid \exists E, F \in \mathfrak{L} \wedge \text{DOFsIndex}(\{E, F\}) = i \wedge (3.34)$$

$$\wedge E = \lambda_L(F) \wedge E, F \in \pi_B(L)\}. \quad (3.35)$$

3.3.3 Shape

We define with $s \in \mathbb{R}^n$ the position of all the joints in the system.

For a given joint $\{E, F\} \in \mathcal{J}$ with $\text{DOF}(\{E, F\}) = 1$, its configuration can be found in the *shape* vector at the location $\text{DOFsOffset}(\{E, F\})$.

Consistently with Assumption 3.1, the velocity and acceleration of the internal joints are given by $\dot{s} \in \mathbb{R}^n$ and $\ddot{s} \in \mathbb{R}^n$.

Remark 3.5. In geometric mechanics literature, the configuration of the internal joints of a multibody system is called the *shape* of the system.

3.4 Relative Forward Kinematics

Using the definition relative to the joints, we can now define the basic relationship between the shape s of a multibody system and the relative position, velocity and accelerations of the body composing the system, i.e. the links.

With the definition of joints as given before and given the shape of the system s , it is possible to compute the *relative forward kinematics*, i.e. the relative position of two arbitrary bodies L and $D : {}^L H_D(s)$. It is defined in the following:

$${}^L H_D(s) = {}^L H_{\lambda_D(L)} \lambda_D(L) H_{\lambda_D^2(L)} \dots \lambda_D^{d-1}(L) H_{\lambda_D^d(L)}. \quad (3.36)$$

In particular, the transform between the two nearby links L and $\lambda_D(L)$ is given by the joint model, as explained in 3.2. This joint transform is constant if $DOF(\{\lambda_D(L), L\}) = 0$ or it is dependent on $s_{DOFsOffset}(\{\lambda_D(L), L\})$ otherwise.

Proposition 3.1 (Left-trivialized Relative Jacobian). *The left-trivialized relative velocity ${}^D v_{L,D}(s, \dot{s})$ can be written as:*

$${}^D v_{L,D}(s, \dot{s}) = {}^D S_{L,D}(s) \dot{s}(s). \quad (3.37)$$

Where ${}^D S_{L,D} \in \mathbb{R}^{6 \times n}$ is the relative left-trivialized Jacobian.

The i -th column of ${}^D S_{L,D}$, i.e. $({}^D S_{L,D})_{(:,i)}$ is given as:

$$({}^D S_{L,D})_{(:,i)} = \begin{cases} {}^D X_F^F s_{E,F} & i \in \pi_L^{DOF}(D) \wedge DOFsOffset(\{E, F\}) = i \\ 0_{6 \times 1} & \text{otherwise} \end{cases} \quad (3.38)$$

Proof. For the sake of readability, in the proof we will drop the explicit dependency on s and \dot{s} .

The left-trivialized velocity ${}^D v_{L,D}$ is defined as:

$${}^D v_{L,D} = \left({}^D H_L^L \dot{H}_D \right)^\vee. \quad (3.39)$$

From the time derivative of (3.36), ${}^D H_L^L \dot{H}_D$ can be written as (assuming that $DOFsOffset(\{E, F\}) = i$):

$${}^D H_L^L \dot{H}_D = \sum_{i \in \pi_L^{DOF}(D)} {}^D H_L^L H_E \frac{\partial^E H_F}{\partial s_i} F H_D \dot{s}_i = \quad (3.40a)$$

$$= \sum_{i \in \pi_L^{DOF}(D)} {}^D H_F^F H_E \frac{\partial^E H_F}{\partial s_i} F H_D \dot{s}_i = \quad (3.40b)$$

$$= \sum_{i \in \pi_L^{DOF}(D)} {}^D H_F^F s_{E,F}^\wedge F H_D \dot{s}_i = \quad (3.40c)$$

$$= \sum_{i \in \pi_L^{DOF}(D)} ({}^D X_F^F s_{E,F})^\wedge \dot{s}_i, \quad (3.40d)$$

where in the derivation we use that ${}^F s_{E,F}^\wedge = {}^F H_E \frac{\partial^E H_F}{\partial s_i}$ from (3.10) and then that ${}^D X_F^F s_{E,F} = \left({}^D H_F^F s_{E,F}^\wedge F H_D \right)^\vee$ from (2.27).

Combining (3.39) and (3.40d) one obtains:

$${}^D v_{L,D} = \sum_{i \in \pi_L^{\text{DOF}}(D)} ({}^D X_E {}^E s_{E,F}) \dot{s}_i, \quad (3.41)$$

that written in matrix form is (3.37). \square

3.5 Multibody Lagrangian Dynamics

In this section we introduce the dynamics of a multibody system using Lagrangian concepts, as done for the rigid body in Section 2.6. To reduce unnecessary complexity and improve readability, in this section we will always use the *left-trivialized* representation for rigid body velocity, acceleration and forces. Expression to transform the dynamics using the different representation are then given in Section 3.7.

3.5.1 Absolute State definition

Being a mechanical system, the equations of motion governing its dynamics are a second-order differential system. Hence, we have to define a *state* of the system composed of a properly defined *position* and *velocity*.

Free-Floating system position

The process of defining the *system position* aims at determining a set of variables from which the position of each point of the multi-body system can be retrieved in the absolute frame A . Being a composition of rigid bodies, each point of the multi-body system can be retrieved from the position-and-orientation – referred to as *pose* – of each link frame $L \in \mathfrak{L}$. Given the topology of the considered multi-body system, however, each link pose can be determined from the pose of the base frame B and the *joint configurations* or *shape* (see Figure 3.1). In light of the above, the configuration of a free-floating system is given by the *base pose* ${}^A H_B \in \text{SE}(3)$ and the *joint positions* $s \in \mathbb{R}^n$, i.e. one can define the configuration set \mathbb{Q} as follows:

$$\mathbb{Q} = \text{SE}(3) \times \mathbb{R}^{n_J}, \quad (3.42)$$

$$q^B = ({}^A H_B, s) \in \mathbb{Q}. \quad (3.43)$$

The B superscript in q^B emphasizes the dependency on B of the representation of the configuration.

Free-Floating system velocity

In view of (3.43), the derivative of the robot position is given by:

$$\dot{q}^B := (^A\dot{H}_B, \dot{s}).$$

As in the case of a single rigid body discussed in detail in Chapter 2 it is more convenient to represent the velocity as a column vector. Using the *left-trivialized* base velocity representation, we can define the system velocity vector $\nu^B \in \mathbb{R}^{6+n_J}$ as follows:

$$\nu^{B/B} = \begin{bmatrix} {}^B v_{A,B} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} {}^A R_B^{T,A} \dot{o}_B \\ ({}^A R_B^{T,A} \dot{R}_B)^\vee \\ \dot{s} \end{bmatrix} \in \mathbb{R}^{n+6}, \quad (3.44)$$

with ${}^B v_{A,B} \in \mathbb{R}^6$ the left-trivialized *base* velocity, and \dot{s} the *joint* velocities.

In a similar way we can represent the system velocity with the *right-trivialized* or *mixed* representation for the base velocity:

$$\nu^{B/A} = \begin{bmatrix} {}^A v_{A,B} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} {}^A R_B^{T,A} \dot{o}_B \\ ({}^A R_B^{T,A} \dot{R}_B)^\vee \\ \dot{s} \end{bmatrix} \in \mathbb{R}^{n+6}, \quad (3.45)$$

Remark 3.6. *The system position and velocity depend upon the choice of base link B . Furthermore, the system velocity vector depends also on the representation used for the base 6D velocity, either left-trivialized, right-trivialized or mixed.*

3.5.2 Kinematics

We recall below how to relate the pose and velocity of an arbitrary link frame $L \in \mathfrak{L}$ to the Free-Floating system position and velocity.

The pose of a link L w.r.t to the inertial frame is a function of the system position q^B (see Figure 3.1):

$${}^A H_L(q^B) : \mathbb{Q} \mapsto \text{SE}(3), \quad (3.46a)$$

$${}^A H_L(q^B) = {}^A H_B {}^B H_L(s) = \begin{bmatrix} {}^A R_B & {}^A o_B \\ 0_{1 \times 3} & 1 \end{bmatrix} {}^B H_L(q). \quad (3.46b)$$

The velocity of a frame L w.r.t. to the inertial frame is the product between a robot position-dependent Jacobian matrix $J_{L,B/B}(q^B) \in \mathbb{R}^{6 \times n+6}$ and the system velocity:

$${}^L v_{A,L}(q^B, \nu^B) = J_{L,B/B}(q^B) \nu^B, \quad (3.47a)$$

$$J_{L,B/B}(q^B) = [{}^L X_B \quad {}^L S_{L,B}(s)]. \quad (3.47b)$$

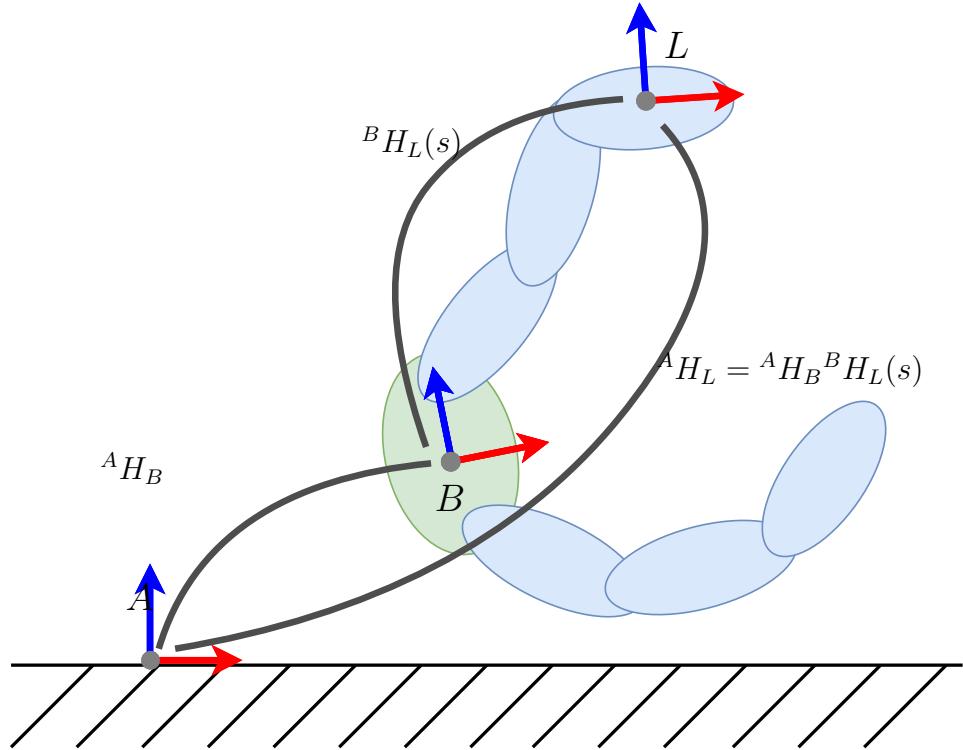


Fig. 3.1 The pose of a frame ${}^A H_L$ is a function of the base pose ${}^A H_B$ and the shape s of the mechanism.

The equations (3.46) and (3.47) represent the so-called *forward kinematics* of the link L using B as the base frame, and using the left-trivialized base velocity, i.e. expressed in the frame B . For the time being, in this chapter we will always assume that the base link B is known and fixed, and consequently we will avoid explicitly indicating the base dependency on the different quantities, i.e. we will simply indicate q^B with q , ν^B with ν , ${}^L v_{A,L}$ with v_L and $J_{L,B/B}$ with $J_{L/B}$. This assumption will be properly discussed and generalized in Section 3.7.

Furthermore, in the next chapter we will discuss how to derive the equation of motions using Lagrangian dynamics. To simplify the notation even more in this section, we will simply assume that we use the left-trivialized representation, i.e. $J_{L/B}$ will be simply indicated J_L . Again, this assumption will be properly discussed and generalized in Section 3.7.

3.5.3 Multibody Lagrangian

Similarly to the Rigid Body case, for a Multibody system in which the configuration is described by an element of $\text{SE}(3) \times \mathbb{R}^n$, a generalization of the Euler-Lagrange equations apply, the Hamel equations. In particular the Hamel equations can be seen as a combination of the Euler-Poincare equations for the *base* part, and the classical Euler-Lagrange equation for the joint.

The Lagrangian for a Free-Floating Mechanical System can be simply obtained as the sum of the Lagrangian of each rigid body.

$$l(q, \nu) = k(q, \nu) - U(q) \quad (3.48a)$$

$$k(q, \nu) = \sum_{L \in \mathbb{L}} {}^L v_{A,L}^T \mathbb{M}_L {}^L v_{A,L} \quad (3.48b)$$

$$U(q) = - \sum_{L \in \mathbb{L}} [{}^A g^T \ 0] {}^A H_L(q) \begin{bmatrix} m_L {}^L c_L \\ m_L \end{bmatrix} \quad (3.48c)$$

One major difference between (2.63) and (3.48) is that in (3.48) the kinetic energy term depends also on the system configuration, rather than just on the velocity.

The following proposition gives us a more convenient form for the left-trivialized lagrangian.

Proposition 3.2. *The left trivialized lagrangian for a multibody defined in*

(3.48) can be equivalent expressed as:

$$k(q, \nu) = \frac{1}{2} \nu^T M(s) \nu \quad (3.49)$$

$$U(q) = - [{}^A g^T \ 0] {}^A H_B \begin{bmatrix} m {}^B c(s) \\ m \end{bmatrix} \quad (3.50)$$

where $M(s) \in \mathbb{R}^{n+6 \times n+6}$ is the system's mass matrix, defined as:

$$M(s) = \sum_{L \in \mathfrak{L}} J_L^T(s) {}_L \mathbb{M} L J_L(s), \quad (3.51)$$

m is the total mass of the multibody system:

$$m := \sum_L m_L \quad (3.52)$$

and ${}^B c(s)$ is the total center of mass of the multibody system, defined as:

$$\begin{bmatrix} {}^B c(s) \\ 1 \end{bmatrix} := \frac{1}{m} \sum_L {}^B H_L(s) \begin{bmatrix} m_L {}^L c_L \\ m_L \end{bmatrix} \quad (3.53)$$

The mass matrix has a specific structure, discussed in the next Theorem.

Theorem 3.1 (Mass Matrix structure, [Wensing and Orin, 2016]). *The mass matrix M_B can be expressed as follows:*

$$M = \begin{bmatrix} {}_B \mathbb{M}(s) & F(s) \\ F^T(s) & H(s) \end{bmatrix}, \quad (3.54)$$

with $F_B \in \mathbb{R}^{6 \times n}$ the upper right block of the mass matrix, $H_B \in \mathbb{R}^{n \times n}$ the joint mass matrix and ${}_B \mathbb{M}^C$ the so-called locked 6D rigid body inertia of the multi-body system, i.e.

$${}_B \mathbb{M}(s) = \sum_{L \in \mathfrak{L}} {}_B X^L(s) {}_L \mathbb{M} L {}^L X_B(s). \quad (3.55)$$

Furthermore, the first six rows of the mass matrix define the jacobian of the articulated body momentum (i.e. the sum of the linear/angular momentum of all the bodies composing the robot), referred to as the Momentum Matrix.

$${}_B h = [{}_B \mathbb{M} \quad F_B] \nu^B = \sum_{L \in \mathfrak{L}} {}_B X^L I_L {}^L v_L. \quad (3.56)$$

Remark 3.7. In the following we indicate with m_L , Lc_L and ${}_L\mathbb{M}_L$ the constant inertial quantities for a specific link L . We indicate instead with m , ${}^Bc(s)$ and ${}_L\mathbb{M}(s)$ without any subscript the inertial quantities for all the robot. Even if by analogy we use the same symbols we used in describing the dynamics of a single rigid body, always remember that for a multibody this quantities (except for the total mass) are shape-dependent.

Plugging this lagrangian in the Hamel equations, we then have that the equations of motions classical form of the equation of a multibody system expressed in the next theorem.

Theorem 3.2. *The equations of motion of a multibody system are given by:*

$$M(s)\dot{\nu} + C(q, \nu)\nu + G(q) = \begin{bmatrix} 0_{6 \times 1} \\ \tau \end{bmatrix} + \sum_{L \in \mathbb{L}} J_L^T L f^x, \quad (3.57)$$

where we have:

$$M(s) = \sum_L J_L^T L \mathbb{M}_L J_L, \quad (3.58a)$$

$$C(q, \nu) = \sum_L J_L^T \left[(\mathbf{v}_L \bar{\times} {}^* L \mathbb{M}_L + {}_L \mathbb{M}_L \mathbf{v}_L \times) J_L + {}_L \mathbb{M}_L \dot{J}_L \right], \quad (3.58b)$$

$$G(q) = -M(s) \begin{bmatrix} {}^A R_B^{TA} g \\ 0_{3 \times 1} \\ 0_{n \times 1} \end{bmatrix}. \quad (3.58c)$$

The proof to this theorem is given in the appendix and in particular in Theorem A.2, as it requires the necessary background in matrix Lie group theory.

The matrix C is chosen such that the following properties hold on the model (3.57):

Proposition 3.3. *The mass matrix M is symmetric and positive definite.*

Proposition 3.4. *The matrix $\dot{M} - 2C$ is skew symmetric.*

Remark 3.8. *To the best of the authors' knowledge, the compact form of the Coriolis matrix (3.58b) was proposed for the first time in [Garofalo et al., 2013].*

Remark 3.9. *The left-trivialized matrix $M_{B/B}(s)$ depends only on the shape of the system, i.e. it is independent on the base pose ${}^A H_B$.*

3.6 Base Change of Variables

In the next sections, we will see that the effect of changing base representation and base link choice can be represented as a change of state. We can pose all this transformations as a change of state variables from (q, ν) to $(\tilde{q}, \tilde{\nu})$. Such transformation can model a change of base representation, a change of base link or additional useful system transformation as discussed in Section 3.9. An important feature of the class of transformations considered is that they only change how *base* part of the state is represented, while the shape variable representation is assumed to never change. For this reason we will call the following class of transformations *Base Changes of Variables*:

$$\tilde{q} = (\tilde{H}, s) = (HH_T(s), s) \quad (3.58d)$$

$$H_T(s) \in \text{SE}(3) \quad (3.58e)$$

$$\tilde{\nu} = T(q)\nu, \quad (3.58f)$$

$$T(q) = \begin{bmatrix} T_{b,b}(q) & T_{b,s}(q) \\ 0_{n \times 6} & 1_n \end{bmatrix} \in \mathbb{R}^{(n+6) \times (n+6)}. \quad (3.58g)$$

Assumption 3.6. *For each q , $T(q)$ is invertible, i.e*

$$\forall q \in \text{SE}(3) \times \mathbb{R}^n \exists T^{-1}(q).$$

Assumption 3.7. *The function $T(\cdot) : \text{SE}(3) \times \mathbb{R}^n \mapsto \mathbb{R}^{6+n \times 6+n}$ is smooth.*

Remark 3.10. *Given the triangular structure of $T(q)$, $T(q)$ is invertible if and only if $T_{b,b}(q)$ is invertible.*

Remark 3.11. *The assumption 3.6 prevents to use as $T(q)$ a transform that maps base angular velocity to a derivative of a minimal representation of the base orientation, such as Euler Angles, as the resulting transformation would be noninvertible in the singularity point of the representation [Stuelpnagel, 1964].*

In the next theorem we see how such a transformation affect the multi-body dynamics.

Theorem 3.3. *Assume that we have a change of variables in the form (3.59) that respect Assumption 3.6, that is applied to the system described by the equations of motion introduced in Theorem 3.2. Then, the following results hold.*

1. The equations of motions (3.57) transform into

$$\dot{\tilde{q}} = \left(\dot{H}(q, \nu) H_T(s) + H(q) \dot{H}_T(s, \dot{s}), \dot{s} \right) \quad (3.59a)$$

$$\tilde{M}(\tilde{q})\dot{\tilde{\nu}} + \tilde{C}(\tilde{q}, \tilde{\nu})\tilde{\nu} + \tilde{G}(\tilde{q}) = \begin{bmatrix} 0_{6 \times 1} \\ \tau \end{bmatrix} + \sum_L \tilde{J}_L f_L^x, \quad (3.59b)$$

with (omitting the dependencies to improve readability)

$$\tilde{M} = T^{-T} M T^{-1}, \quad (3.60a)$$

$$\tilde{C} = T^{-T} \left(M \frac{d}{dt} (T^{-1}) + C T^{-1} \right), \quad (3.60b)$$

$$\tilde{G} = T^{-T} G. \quad (3.60c)$$

2. The free-floating system kinetic energy is given by:

$$K = \frac{1}{2} \tilde{\nu}^T \tilde{M}(\tilde{q}) \tilde{\nu} \quad (3.61)$$

3. Property 3.3 is preserved: if M is symmetric and positive definite then \tilde{M} is symmetric and positive definite.

4. Property 3.4 is preserved: if $\dot{M}(q, \nu) - 2C(q, \nu)$ is skew symmetric, then $\dot{\tilde{M}}(\tilde{q}, \tilde{\nu}) - 2\tilde{C}(\tilde{q}, \tilde{\nu})$ is skew symmetric.

The demonstration of this theorem is given in Subsection 3.10.1.

This theorem is important because it ensures us that, as long as we transforms the dynamics using a transformation in the form (3.59), for the resulting equations of motion the basic properties required by motion control algorithm holds. Consequently, the control design can choose the representation that he prefers, or in which the specific control synthesis is simplified, without having to verify every time that this properties hold.

3.7 Free Floating Dynamics with different representations or the base velocity

Note that (3.57) presents the equation of motion of the multibody system using the left-trivialized velocity. To obtain the equation of motions for other representations such as the right-trivialized and mixed representation of the multibody velocity, we will just apply some nonlinear transformations for the velocity part of the state. As in this section we need to discuss how to change between different representation of the velocity, we will need to

be precise with respect to the representation used, at the expense of the lightness of the notation. Furthermore the equation provided in this section will provide a convenient reference for the rest of the thesis, and for this purpose we need to write them in a non ambiguous way. In particular, we are interested to understand to which state transformation in the form 3.59 change in base velocity representation from *left-trivialized* to *right-trivialized* or *mixed* representation is associated, to exploit the properties enunciated in the previous section.

3.7.1 Left-trivialized

For, we first just rewrite the equations of motion using the complete notation.

Theorem 3.4. *The equations of motions of a multibody system using B as the base link, and using the left-trivialized representation of the base velocity, are given by:*

$$\begin{aligned} M_{B/B}(s)\dot{\nu}^{B/B} + C_{B/B}(q^B, \nu^{B/B})\nu^{B/B} + G_{B,B}(q) = \\ = \begin{bmatrix} 0_{6 \times 1} \\ \tau \end{bmatrix} + \sum_L J_{L,B/BL}^T f^x, \end{aligned} \quad (3.62)$$

where we have:

$$M_{B/B}(s) = \sum_L J_{L,B/B}^T L \mathbb{M}_L J_{L,B/B}, \quad (3.63a)$$

$$\begin{aligned} C_{B/B}(q^B, \nu^{B/B}) = \sum_L J_{L,B/B} \left[\left({}^L v_{A,L} \bar{\times} {}^L \mathbb{M}_L + {}_L \mathbb{M}_L {}^L v_{A,L} \times \right) J_{L,B/B} + \right. \\ \left. + {}_L \mathbb{M}_L \dot{J}_{L,B/B} \right], \end{aligned} \quad (3.63b)$$

$$G_{B/B}(q) = M_{B/B}(s) \begin{bmatrix} {}^A R_B^T A \\ 0_{3 \times 1} \\ 0_{n \times 1} \end{bmatrix}. \quad (3.63c)$$

Remark 3.12. *Similarly to the rigid body case, the left-hand side of (3.62) can be written to depend just on the sensor proper acceleration $\alpha_{A,B}^g$, the body angular velocity ${}^B \omega_{A,B}$ and the shape position, velocity and accelerations s, \dot{s}, \ddot{s} , i.e.:*

$$\begin{aligned} M_{B/B}(s)\dot{\nu}^{B/B} + C_{B/B}(q^B, \nu^{B/B})\nu^{B/B} + G_{B,B}(q) = \\ \Gamma(\alpha_{A,B}^g, {}^B \omega_{A,B}, s, \dot{s}, \ddot{s}) = \begin{bmatrix} 0_{6 \times 1} \\ \tau \end{bmatrix} + \sum_L J_{L,B/BL}^T f^x, \end{aligned} \quad (3.64)$$

Such representation is not convenient when analyzing the multibody system as a dynamical system as it lumps together the linear acceleration of the base with the gravity expressed in body frame, that depends on another part of the state (the rotation between the base frame B and the inertial frame A). However, it is extremely convenient in estimation and identification, because α^G and ${}^B\omega_{A,B}$ can easily be obtained from common inertial sensors, as explained in Chapters 4 and 7.

3.7.2 Right-Trivialized

Lemma 3.9. Assume that the base velocity representation is changed from left-trivialized to right-trivialized. Then, the following results hold.

1. The system position and velocity are subject to the following transformations

$$H_T = 1_4, \quad (3.65)$$

$${}^{B/A}T_{B/B} = \begin{bmatrix} {}^A X_B & 0_{6 \times n} \\ 0_{n \times 6} & 1_n \end{bmatrix}. \quad (3.66)$$

2. The jacobian $J_{L,B/A} \in \mathbb{R}^{6 \times n+6}$ of a link frame L is subject to the following transformation

$$J_{L,B/A} = {}^A X_L J_{L,B/B} {}^{B/B} T_{B/A}. \quad (3.67)$$

3.7.3 Mixed

Lemma 3.10. Assume that the base velocity representation is changed from left-trivialized to mixed. Then, the following results hold.

1. The system position and velocity are subject to the following transformations

$$H_T = 1_4 \quad (3.68)$$

$${}^{B/B[A]}T_{B/B} = \begin{bmatrix} {}^{B[A]} X_B & 0_{6 \times n} \\ 0_{n \times 6} & 1_n \end{bmatrix}. \quad (3.69)$$

2. The Jacobian $J_{L,B/A} \in \mathbb{R}^{6 \times n+6}$ of a link frame L is subject to the following transformation

$$J_{L,B/B[A]} = {}^{B[A]} X_L J_{L,B/B} {}^{B/B} T_{B/B[A]}. \quad (3.70)$$

3.8 Free Floating Dynamics with different base links

Similarly to the change of base velocity representation, also a change of the base link can be represented as *Base Change of Variables*.

The main difference is that a change of base link introduce the transformation matrix T is not block diagonal anymore. Furthermore, the change of variables involves also the base position part of the state, i.e. $H_T \neq 1_4$. To improve the details carried by the complete notation, given a robot velocity $\nu^{B/B}$ and a transformed robot velocity $\nu^{D/D}$ we define ${}^{D/D}T_{B/B}$ as the matrix for which it holds that:

$$\nu^{D/D} = {}^{D/D}T_{B/B}\nu^{B/B}. \quad (3.71)$$

In analogy with the force transformation matrix for 6D forces, we also define ${}^{D/D}T^{B/B}$ as:

$${}^{D/D}T^{B/B} = \left({}^{D/D}T_{B/B} \right)^{-T}. \quad (3.72)$$

Property 3.1. *The state transformation associated with a change of base link from a link B to a link D is of the form 3.59. In particular, it is given, depending on the used velocity representation:*

- *Left-Trivialized Change of Base Link*

$$H_T = {}^B H_D, \quad (3.73)$$

$${}^{D/D}T_{B/B} = \begin{bmatrix} {}^D X_B & {}^D S_{D,B} \\ 0_{n_J \times 6} & 1_{n_J} \end{bmatrix}. \quad (3.74)$$

- *Right-Trivialized Change of Base Link*

$$H_{D/A}T_{B/A} = {}^B H_D, \quad (3.75)$$

$${}^{D/A}T_{B/A} = \begin{bmatrix} 1_6 & {}^A S_{D,B} \\ 0_{n_J \times 6} & 1_{n_J} \end{bmatrix}. \quad (3.76)$$

- *Mixed Change of Base Link*

$$H_T = {}^B H_D, \quad (3.77)$$

$${}^{D/D[A]}T_{B/B[A]} = \begin{bmatrix} {}^{D[A]}X_{B[A]} & {}^{D[A]}S_{D,B} \\ 0_{n_J \times 6} & 1_{n_J} \end{bmatrix}. \quad (3.78)$$

Remark 3.13 (Pure internal joint torques invariance). *Let Γ_B be a free-floating generalized force acting on the multi-body system. If Γ_B has a base force-torque equal to zero, then it is independent from the base frame in which it is expressed, i.e.*

$$T^T \begin{bmatrix} 0_{6 \times 1} \\ \tau_B \end{bmatrix} = \begin{bmatrix} 0_{6 \times 1} \\ \tau_D \end{bmatrix} = \begin{bmatrix} 0_{6 \times 1} \\ \tau \end{bmatrix}.$$

Remark 3.14 (External joint torque base dependency). *Let assume that B is the base link, and the base dynamics is represented using the left-trivialized representation. The effect of a pure base force-torque ${}_B f_B^x$ on a the link B is by definition the generalized force where the first six elements are given by f_B^x and joint part is equal to zero:*

$$\Gamma_{B/B} = \begin{bmatrix} {}_B f_B^x \\ 0_{n \times 1} \end{bmatrix}.$$

The generalized force expressed w.r.t. another base link D is given by:

$$\Gamma_{D/D} = {}_{D/D} T^{B/B} \Gamma_{B/B} = \begin{bmatrix} {}_D X^B & 0_{6 \times n_J} \\ ({}^D S_{D,B})^T {}_D X^B & 1_{n_J} \end{bmatrix} \begin{bmatrix} f_B^x \\ 0_{n \times 1} \end{bmatrix} = \begin{bmatrix} {}_D X^{BB} f_B^x \\ ({}^D S_{D,B})^T {}_D X^B f_B^x \end{bmatrix}.$$

The first six elements of the transformed generalized torques represent the external force-torque f_B^x , but expressed with respect to the origin of D rather than the origin of B . The last n elements are instead a form of external joint torques, and they are non-zero only for the joints on the path from B to D . It is important to note that contrary to the case of fixed base robots, the external joint torques depend on the floating base, and, consequently, they do not describe any meaningful base-invariant physical quantity.

3.9 System state transformation providing centroidal dynamics

In the humanoid robotics literature, it is widespread to control as primary task the position of the center of mass [Ott et al., 2011] and to minimize the angular momentum of the robot [Orin and Goswami, 2008, Orin et al., 2013, Dai et al., 2014, Herzog et al., 2016, Lee and Goswami, 2012, Koolen et al., 2016]. For analysis purposes, it is then convenient to include the center of mass in the state representing the robot position [Ott et al., 2011, Nava et al., 2016]. Such an inclusion can be expressed as a change of variables using the formalism discussed in the previous sections. Expressing such

change of variables in the proposed formalism simplifies the computation of the system dynamics in the new state.

As detailed in the next subsection, the state transformation equations given by using a generic base *frame* rather a base frame rigidly attached to a link are simpler when using the *mixed* representation. Furthermore the works related to multi-task control and centroidal dynamics typically use the *mixed* representation. For this reason in this section we will prefer to use the mixed representation, with the following simplified notation to improve the readability of the section.

Simplified Notation, valid for Section 3.9	
$\overset{A}{\overline{B}} := B[A]$	Inertial frame.
i	Frame with the orientation of the inertial frame A and with the origin at the origin of frame B .
$\nu^C := \begin{bmatrix} {}^{C[A]}v_{A,C} \\ s \end{bmatrix}$	Robot velocity using the mixed velocity of C as the base frame.
$M_C := M_{C/C[A]}$	Mass matrix using C as the base frame and the mixed representation for base quantities.

3.9.1 Frames not rigidly attached to a link

So far, the base frame transformation was meant to be between a frame B and a frame D , and both of these frames were assumed to be attached to a physical link. This assumption, however, is not strict. As a matter of fact, we can assume that the base frame D is a frame whose origin is that of a frame E , and whose orientation that of a frame F :

$$q^{E[F]} := ({}^A H_{E[F]}, s) = \left(\begin{bmatrix} {}^A R_F & {}^A o_E \\ 0_{3 \times 1} & 0 \end{bmatrix}, s \right). \quad (3.79)$$

Then, the associated generalized robot velocity is given by:

$$\nu^{E[F]} = \begin{bmatrix} {}^A \dot{o}_E \\ {}^A \omega_{A,F} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} {}^A \dot{o}_E \\ \left({}^A \dot{R}_F {}^A R_F^T \right)^\vee \\ \dot{s} \end{bmatrix} \quad (3.80)$$

The new *base* velocity reflect the different choices of the *base* position variables. The Jacobian of the velocity of a frame defined in such a compound way is simply given by the combination of the linear part (first three

rows) of $J_{C,B}$, indicated with $J_{C,B}^l \in \mathbb{R}^{3 \times (6+n)}$ and the angular part of (last three rows) of $J_{D,B}$, indicated with $J_{D,B}^a \in \mathbb{R}^{3 \times (6+n)}$:

$$J_{E[F],B} = \begin{bmatrix} J_{E,B}^l \\ J_{F,B}^a \end{bmatrix}. \quad (3.81)$$

Note that the property of (3.81) be the simple stacking of the linear and angular Jacobians is a consequence of the use of *mixed* velocity representation. Indeed the equivalent jacobians using the *inertial* or *body-fixed* representation are related in a more complex way to the Jacobians of frame E and F .

From (3.81), one obtains the transformation matrix ${}^{E[F]}T_B$ to be used in a base variable changed defined as in :

$$H_T = {}^B H_{E[F]} {}^{E[F]}T_B = \begin{bmatrix} J_{E[F],B} \\ 0_{6 \times n} & 1_{n \times n} \end{bmatrix}. \quad (3.82)$$

3.9.2 Recalls on centroidal dynamics quantities

To properly define the *centroidal dynamics*, it is convenient to define a frame \bar{G} whose origin is the center of mass of the multi body system, and whose orientation is that of the inertial frame A . Note that the use of \bar{G} is an abuse of notation, as we *do not* define an orientation for frame G (see Figure 3.2).

Then, the total momentum and the composite rigid body inertia (CRBA) of the system expressed w.r.t. the \bar{G} frame are given by:

$$\bar{G}^h = \bar{G}^X \bar{B}^h, \quad \bar{G}^I = \bar{G}^X \bar{B}^I C^{\bar{B}} X_{\bar{G}}. \quad (3.83)$$

In the robotics literature, these quantities are known as *Centroidal Momentum* and as *Centroidal Composite Rigid Body Inertia (CCRBI)* [Orin et al., 2013], respectively. The structure of these *centroidal* quantities is the following [Orin et al., 2013] :

$$\bar{G}^h = \begin{bmatrix} m^A \dot{c} \\ \bar{G}^h^a \end{bmatrix}, \quad \bar{G}^I = \begin{bmatrix} m I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & L^C \end{bmatrix}, \quad (3.84)$$

with $m \in \mathbb{R}$ the total mass of the robot, ${}^A c \in \mathbb{R}^3$ the center of the mass of the robot expressed in the inertial frame, $\bar{G}^h^a \in \mathbb{R}^3$ the total angular momentum and $L^C \in \mathbb{R}^{3 \times 3}$ the locked 3D inertia of the robot , both expressed in \bar{G} .

Given a base frame B , note that the matrix $A_{G,B} \in \mathbb{R}^{6 \times (n+6)}$ that multiplied by the generalized robot velocities vector ν^B gives the *centroidal*

momentum can be easily obtained from the floating base mass matrix. In fact, by combining (3.56) and (3.83) one has:

$$\bar{G}^h = A_{G,B}\nu^B, \quad A_{G,B} = \bar{G}^X \bar{B}^T [\bar{B}^I C \quad F_B]. \quad (3.85)$$

The $A_{G,B}$ matrix is known as the *Centroidal Momentum Matrix* [Orin and Goswami, 2008, Orin et al., 2013].

3.9.3 Average 6D Velocity

In the humanoids whole-body control literature, it is common to define the *average 6D velocity* v_G of the robot as [Orin et al., 2013]:

$$v_G = (\bar{G}^I)^{-1} \bar{G}^h = \left[\begin{array}{c} \frac{1}{m} (m^A \dot{o}_G) \\ (L^C)^{-1} \bar{G}^h^a \end{array} \right] = \left[\begin{array}{c} {}^A \dot{o}_G \\ \omega_G \end{array} \right]. \quad (3.86)$$

By definition, the linear part of the *average 6D velocity* (i.e. the first three components of v_G) is simply the velocity of the center of mass of the system. Its angular part (i.e. the last three components) is called the *average angular velocity* ω_G [Jellinek and Li, 1989, Essen and Essén, 2004, Morita and Ohnishi, 2003], even if this name is an abuse of the term *angular velocity* because ω_G is not defined as the angular velocity of an orientation frame. In fact, a rotation matrix $R(s) \in SO(3)$ such that $R(s)R^T(s) = \omega_G^\wedge$ exists only for a limited class of multibody systems [Saccon et al., 2017]. The *average angular velocity* has a precise physical meaning if a multibody system is evolving without external forces acting on it. If a generalized impulse blocks all its joint motions instantaneously, the resulting rigid body would evolve with an angular velocity ω_G . For this reason, ω_G is also known as the *locked angular velocity* in geometrical mechanics literature [Marsden and Scheurle, 1993].

The relationship between the generalized robot velocities vector ν^B and the *centroidal velocity* v_G can be easily obtained by combining (3.85) and (3.86):

$$\begin{aligned} v^G &= (\bar{G}^I)^{-1} A_{G,B} \nu^B = (\bar{G}^I)^{-1} \bar{G}^X \bar{B}^T [\bar{B}^I C \quad F_B] \nu^B = \\ &= [\bar{G}^X \bar{B} \quad (\bar{G}^I)^{-1} \bar{G}^X \bar{B}^T F_B] \nu^B. \end{aligned}$$

Noting that this matrix has the same structure of the floating base jacobian (3.48), we borrow the notation we use for the Jacobians of links, even if v_G is not defined as the mixed velocity of a well defined frame. So, we define $J_{G,B}$ and $S_{G,B}$ such that:

$$\nu^G = J_{G,B} \nu^B, \quad (3.87)$$

$$J_{G,B} := \begin{bmatrix} \bar{G}X_{\bar{B}} & S_{G,B} \end{bmatrix} := \begin{bmatrix} \bar{G}X_{\bar{B}} & (\bar{G}I^C)^{-1}\bar{G}X^{\bar{B}}F_B \end{bmatrix}. \quad (3.88)$$

3.9.4 Centroidal change of variables

Including the center of mass in the robot state

For the sake of including the center of mass in the multibody dynamics, we can define a new robot position as follows:

$$q^{G[B]} = (^A c, ^A R_B, s). \quad (3.89)$$

This implies that:

$$\nu^{G[B]} = \begin{bmatrix} v_{G[B]} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} {}^A \dot{c} \\ (^A \dot{R}_B {}^A R_B^T)^\vee \\ \dot{s} \end{bmatrix}. \quad (3.90)$$

The corresponding transformation matrix is given by, as explained in subsection 3.9.1:

$${}^{G[B]} T_B = \begin{bmatrix} J_{G[B],B}^l & \\ 0_{3 \times (3+n)} & 1_{(3+n) \times (3+n)} \end{bmatrix}, \quad (3.91)$$

where $J_{G[B],B}^l$ is the center of mass jacobian, i.e. the matrix such that

$${}^A \dot{c} = J_{G[B],B}^l \nu_B.$$

Note that this is also equal to the first three rows of (3.88), i.e. $J_{G[B],B}^l = J_{G,B}^l$. The change of base introduced by ${}^{G[B]} T_B$ is the one used in [Ott et al., 2011].

Block diagonalization of the mass matrix

Using the definition of *average 6D velocity* of the multibody system, we can define a *centroidal* generalized joint velocities vector in which we combine the average 6D velocity and the joint velocities :

$$\nu^G = \begin{bmatrix} v_G \\ \dot{s} \end{bmatrix}. \quad (3.92)$$

Let us remark (again) that there is no such thing as a G frame: the notation ν^G is an abuse of notation. In particular, it does not make sense to write q^G : the change of variables induced by the use of the average 6D velocity is only

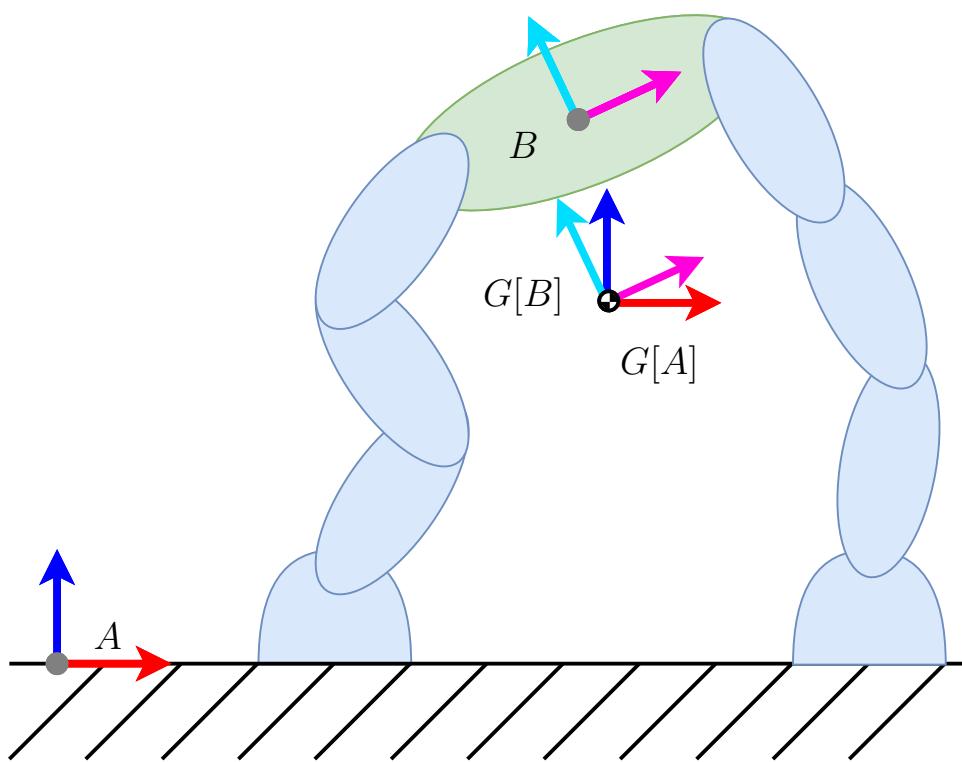


Fig. 3.2 Graphical depiction of frames A , B , $G[A]$ and $G[B]$ for an example two legged robot.

a change of variables in the velocity space, while for the position space it is usually convenient to continue to use the $q^{G[B]}$ position introduced before.

The mapping ${}^G T_B$ from the *floating base* generalized velocities ν^B to ν^G has the same structure of the change of variables introduced in (3.59):

$$\nu^G = {}^G T_B \nu^B, \quad {}^G T_B = \begin{bmatrix} {}^G X_{\bar{B}} & S_{G,B} \\ 0_{n_J \times 6} & 1_{n_J \times n_J} \end{bmatrix}. \quad (3.93)$$

This induces a change of variables also on the generalized robot forces, as detailed in (3.72), i.e. :

$${}^G T^B = {}^G T_B^{-T} = \begin{bmatrix} {}^{\bar{G}} X^{\bar{B}} & 0_{6 \times n_J} \\ -S_{G,B}^T {}^{\bar{G}} X^{\bar{B}} & 1_{n_J} \end{bmatrix}. \quad (3.94)$$

Using the transformation ${}^G T_B$ in (3.60), one can obtain the equations of motion with $(q^{G[B]}, \nu^G)$ as state, i.e. :

$$\dot{q}^{G[B]} = \left(\begin{bmatrix} {}^A \omega_{A,B}^A(\nu^G) {}^A R_B & {}^A \dot{c} \\ 0_{3 \times 1} & 0 \end{bmatrix}, \dot{s} \right), \quad (3.95a)$$

$$M_G(q^{G[B]}) \ddot{\nu}^G + C_G(q^{G[B]}, \nu^G) \nu^G + G_G = \begin{bmatrix} 0_{6 \times 1} \\ \tau \end{bmatrix} + \sum_{L \in \mathfrak{L}} J_{L,G}^T f_L^x, \quad (3.95b)$$

where ${}^A \omega_{A,B}(\nu^G)$ is the angular velocity of the link B as a function of the *centroidal* generalized joint velocity ν_G .

The equations of motion (3.95b) incorporate in the first six rows what is usually called the *centroidal dynamics*, together with the joint dynamics in a single set of equations of motions. The specific features of such a representation of the equations of motion are highlighted in the following theorem, and they have been already exploited in [Nava et al., 2016].

Lemma 3.11. *For the equations of motion given by (3.95b), the following results hold.*

1. *The (centroidal) mass matrix M_G has a block diagonal structure, i.e.*

:

$$M_G(q^{G[B]}) = \begin{bmatrix} m 1_3 & 0_{3 \times 3} & 0_{3 \times n_J} \\ 0_{3 \times 3} & L^C(q^{G[B]}) & 0_{3 \times n_J} \\ 0_{3 \times 3} & 0_{3 \times 3} & H_G(s) \end{bmatrix} \quad (3.96)$$

where $m \in \mathbb{R}$ is the total mass of the system, $L^C \in \mathbb{R}^{3 \times 3}$ is the locked 3D inertia matrix of the system expressed in the center of mass and with the orientation of the absolute frame A and $H_G(s) \in \mathbb{R}^{n \times n}$ is the centroidal joint mass matrix.

2. The (centroidal) gravity term G_G is independent from the robot state and has the following structure:

$$G_G = \begin{bmatrix} -m^A g \\ 0_{3 \times 1} \\ 0_{n_J \times 1} \end{bmatrix} \quad (3.97)$$

where ${}^A g \in \mathbb{R}^3$ is the gravitational acceleration expressed in the frame A .

3. Assuming that Property 3.4 holds for M_B and C_B , the (centroidal) coriolis matrix C_G has a block diagonal structure, i.e. :

$$C_G(q^{G[B]}, \nu^G) = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times (n+3)} \\ 0_{3 \times (n+3)} & C_G^{aj}(q^{G[B]}, \nu^G) \end{bmatrix} \quad (3.98)$$

where $C_G^{aj}(q^{G[B]}, \nu^G) \in \mathbb{R}^{(3+n) \times (3+n)}$ is the centroidal coriolis matrix for the angular and joint part of the dynamics.

The proof for Lemma 3.11 is provided in Subsection 3.10.2.

3.10 Proofs

3.10.1 Proof of Theorem 3.3

1. First, notice that the time derivative of (3.59) yields:

$$\dot{\tilde{\nu}} = T\dot{\nu} + \dot{T}\nu. \quad (3.99)$$

Then, the equations of motion (3.60) can be obtained by substituting $\dot{\nu}$ obtained from (3.99) into (3.57), by substituting ν obtained from (3.59) into (3.57) and by multiplying the obtained equation times T^{-T} .

2. In view of (3.49), i.e.:

$$k = \frac{1}{2} \nu^T M(q) \nu,$$

and of $\tilde{\nu} = T\nu$ one has :

$$k = \frac{1}{2} \tilde{\nu}^T T^{-T} M T^{-1} \tilde{\nu}.$$

From which we obtain that the kinematic energy is also given by:

$$k = \frac{1}{2} \tilde{\nu}^T \tilde{M} \tilde{\nu}, \quad \tilde{M} = T^{-T} M T^{-1}.$$

3. The symmetry and positive definiteness of \tilde{M} follow from the expression of the kinetic energy given in the previous step.
4. The condition that $\dot{M} - 2C$ is skew-symmetric is equivalent to the condition that $\dot{M} = C + C^T$. So, we will demonstrate that $\dot{M} - C - C^T = 0_{(n+6) \times (n+6)}$ implies $\dot{\tilde{M}} - \tilde{C} - \tilde{C}^T = 0_{(n+6) \times (n+6)}$. Let us write $\dot{\tilde{M}} - \tilde{C} - \tilde{C}^T$ using (3.60a) and (3.60b), i.e. :

$$\begin{aligned}
& \dot{\tilde{M}} - \tilde{C} - \tilde{C}^T = \\
&= \underbrace{\frac{d}{dt}(T^{-T})MT^{-1} + T^{-T}\dot{M}T^{-1} + T^{-T}M\frac{d}{dt}(T^{-1})}_{\dot{\tilde{M}}} \\
&\quad \underbrace{-T^{-T}M\frac{d}{dt}(T^{-1}) - T^{-T}CT^{-1}}_{-\tilde{C}} \\
&\quad \underbrace{-\frac{d}{dt}(T^{-T})MT^{-1} - T^{-T}C^TT^{-1}}_{-\tilde{C}^T}
\end{aligned}$$

We can then write:

$$\begin{aligned}
& \dot{\tilde{M}} - \tilde{C} - \tilde{C}^T = \\
& T^{-T}(\dot{M} - C - C^T)T^{-1} + \\
& + \frac{d}{dt}(T^{-T})MT^{-1} - \frac{d}{dt}(T^{-T})MT^{-1} + \\
& \quad + T^TM\frac{d}{dt}(T^{-1}) - T^TM\frac{d}{dt}(T^{-1})
\end{aligned}$$

Using the hypothesis that $\dot{M} - C - C^T = 0_{(n+6) \times (n+6)}$ we can then conclude that $\dot{\tilde{M}} - \tilde{C} - \tilde{C}^T = 0_{(n+6) \times (n+6)}$.

■

3.10.2 Proof of Lemma 3.11

1. The *centroidal* mass matrix can be obtained by applying (3.93) to (3.60), i.e. :

$$M_G = {}_G T^B M_B {}^B T_G. \quad (3.100)$$

By exploiting the structure of M_B and ${}_G T^B$ in (3.54) and (3.94), and recalling that ${}_{\bar{G}} X^{\bar{B}} = {}^{\bar{B}} X_{\bar{G}}^T$ one has:

$$M_G = \begin{bmatrix} {}_{\bar{G}} X^{\bar{B}} & 0_{6 \times n_J} \\ S_{B,G}^T & 1_{n_J \times n_J} \end{bmatrix} \begin{bmatrix} {}^{\bar{B}} I^C & F_B \\ F_B^T & H_B \end{bmatrix} \begin{bmatrix} {}^{\bar{B}} X_{\bar{G}} & S_{B,G} \\ 0_{6 \times n_J} & 1_{n_J \times n_J} \end{bmatrix} \quad (3.101a)$$

$$= \begin{bmatrix} {}_{\bar{G}} X^{\bar{B}} {}^{\bar{B}} I^C {}^{\bar{B}} X_{\bar{G}} & {}_{\bar{G}} X^{\bar{B}} ({}^{\bar{B}} I^C S_{B,G} + F_B) \\ (S_{B,G}^T {}^{\bar{B}} I^C + F_B^T) {}^{\bar{B}} X_{\bar{G}} & H_G \end{bmatrix} \quad (3.101b)$$

with $H_G = S_{B,G}^T {}^{\bar{B}} I^C S_{B,G} + S_{B,G}^T F_B + F_B^T S_{B,G} + H_B$.

In view of (3.83) and (3.84), then ${}_{\bar{G}} X^{\bar{B}} {}^{\bar{B}} I^C {}^{\bar{B}} X_{\bar{G}}$ can be written as the *centroidal composite rigid body inertia*, i.e. :

$${}_{\bar{G}} X^{\bar{B}} {}^{\bar{B}} I^C {}^{\bar{B}} X_{\bar{G}} = {}_{\bar{G}} I^C = \begin{bmatrix} m 1_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & L^C \end{bmatrix}. \quad (3.102)$$

Using (3.88), it is possible to write $S_{B,G}$ as:

$$S_{B,G} = -{}^{\bar{B}} X_{\bar{G}} S_{G,B} = -({}^{\bar{B}} I^C)^{-1} F_B. \quad (3.103)$$

Substituting (3.103) in the off-diagonal terms of (3.101b) it is possible to show that the off-diagonal terms are equal to $0_{6 \times n}$.

2. Exploiting the gravity generalized forces structure given by (3.60c) and the structure of M_G given in (3.96), we can write G_G as:

$$G_G = -M_G \begin{bmatrix} {}^A g \\ 0_{3 \times 1} \\ 0_{n \times 1} \end{bmatrix} = \begin{bmatrix} -m {}^A g \\ 0_{3 \times 1} \\ 0_{n \times 1} \end{bmatrix}.$$

3. From the structure of the centroidal dynamics (3.95b), and from the Newton equation $m^A \ddot{c} - m^A g = \sum_{L \in \mathcal{L}} {}^A f_L^x$ we have that the C_G matrix has the following structure:

$$\begin{bmatrix} 0_{3 \times 3} & 0_{3 \times (n+3)} \\ C_G^{\text{offDiag}} & C_G^{\text{aj}} \end{bmatrix}. \quad (3.104)$$

From the assumption given by Property 3.4 that $(\dot{M}_B - 2C_B)^T = (2C_B - M_B)$ by applying Theorem 3.3 we obtain that:

$$C_G + C_G^T = \frac{\dot{M}_G}{2}. \quad (3.105)$$

Plugging the the sparsity patterns (3.104) and (3.96) in the (3.105), extracting the top left $3 \times n$ subblock one obtains that:

$$0_{3 \times (n+3)} + \left(C_G^{\text{offDiag}} \right)^T = 0_{3 \times (n+3)}$$

from which:

$$C_G^{\text{offDiag}} = 0_{(n+3) \times 3}.$$

■

Chapter 4

Contact Force-Torques and Internal Torques Estimation

4.1 Introduction

In this chapter we will discuss the techniques for whole-body external force-torque and internal torques estimation developed to endow the iCub robot with torque-control capabilities. The developed techniques solved both the problems of *external force-torque* and *joint torques*, while the problem of identifying the *location* of the external force-torque is solved using the distributed tactile system available on the iCub robot.

This chapter extends the results presented in [Fumagalli et al., 2012, Del Prete et al., 2016] and [Del Prete et al., 2012] to the whole-body case. A key idea exploited in this chapter is that the rigid-body dynamics expressed w.r.t. to the *proper sensor acceleration* is particularly well suited for force estimation, as the linear part of the *proper sensor acceleration* is exactly the output of an accelerometer mounted at the origin of the link.

The chapter is structured as in the following. A brief state of the art of how this estimation problems have been solved in the past is presented in Section 4.2. Section 4.3 describe the different sensors used in the estimation algorithms. Section 4.4 discuss the estimation of the *net force-torque* for a single rigid body. Section 4.5 introduces the algorithm for estimation of external force-torques of multibody system equipped with internal force-torques sensors. Section 4.6 discusses the estimation of joint torques, while Section 4.7 discusses the estimation of six-axis force-torque sensor offsets.

4.2 State of art

4.2.1 Joint torques estimation

Strain gauge-based direct sensing of joint torques is one of the main methods used to obtain joint torque feedback [Wu and Paul, 1980, Luh et al., 1983]. Using this approach, the output shaft of the transmission of the motor system is modified to include a strain gauge, that is used to measure the deformation in the shaft due to the torque transmitted by the shaft to the system. The weak points of this techniques are the additional complexity added in the manufacturing of the motor group [Randazzo et al., 2011] and the fact that strain-gauge sensor are quite sensitive to forces that exceed they sensing range.

Another popular technique to estimate joint torque is the use of so-called Series Elastic Actuators (SEA) [Paine et al., 2015]. In SEAs, a linear compliant element (a “spring”) is inserted in series after the transmission output shaft. Optical or magnetic encoders are mounted before and after the compliant element to measure its deformation, that is proportional to the transmitted torques. With respect to strain-gauge based techniques, there are two main advantages. The first one is to exploit explicitly designed compliance rather then the implicit compliance of the shaft, and the second one is to rely on optical or magnetic transducers, that are more robust then strain gauges. A drawback of SEAs is that while the additional compliance is convenient for torque measures, it may be a limitation for higher level controllers, not explicitly design to deal with a compliant system. For this reason the compliance level of a SEAs unit is tipically a tradeoff between the softness required for the torque measure and the rigidity required by classical high level controllers.

An alternative to SEA-based torque estimation is to estimate the torque transmitted by the trasmission by measuring directly the deformation of the trasmission, rather then of a specifically designed elastic elements. In [Zhang et al., 2015] in particular the authors use a pair of encoders to measure the deformation in a Harmonic Drive. With respect to classical SEA-based techniques, in the trasmission-based estimation the compliance model of the trasmission is not simply linear, but it is more complicated.

All the discussed techniques until now (strain gauges mounted on the shaft, SEAs, trasmission-deformation) rely on specific hardware to measure the deformation of part of the motor group. Adding this specific hardware can be too costly, or simply unfeasible if a robot was already designed without any concern for torque control, as it was the case for iCub. An

alternative is to estimate the joint torque from the motor input voltage or current, inverting a given model of relationship between the input variable of the motor and the output torques. This is feasible for low gear ratio motor groups as in [Wensing et al., 2017], but in classical robot system equipped with Harmonic Drives the effect of friction effects degrades the torque estimation quality.

In this thesis we will present a technique for estimating the joint torques without any motor level sensor, but relying instead on six-axis force-torque sensor embedded in some of the robot links, as initially explored in Fumagalli et al. [2012].

4.2.2 External force-torque position estimation

In general, external force-torques are exchanged over a surface at the interface between the robot and the environment, the so-called *contact surface*. Estimating the contact surface is important mainly because it can be used by the robot to produce forces to prevent the contact to be broken.

[Del Prete et al., 2011] considered the problem of estimating the position of an external contact assuming that the contact generates a pure force (i.e. null torque) and its effects are sensed with a force/torque sensor embedded in the robot structure (e.g. at the robot base). The estimation relies on the classical force transformation principle [Siciliano et al., 2009, Section 3.8.3] according to which the spatial transformation of a pure force generates a torque equal to the cross product of the application vector and the applied force. Estimating the application vector (i.e. the contact location) boils down to a linear least squares which identifies a unique solution if at least two non collinear forces. The proposed procedure allows the calibration of a tactile array. In [Manuelli and Tedrake, 2016] the authors propose to solve two problems simultaneously: contact location on the one hand and external force estimation on the other. To reduce the associated computational complexity, the two problems are separated to exploit the fact that the force estimation problem is convex if contact locations are known. The proposed estimation problem is solved with a particle filter.

In this thesis, we will not present techniques for estimating the *position* of an external force-torque, as we rely on a distributed tactile system such as the one present in the iCub robot Maiolino et al. [2013], Del Prete et al. [2011].

4.2.3 External force-torque intensity estimation

[De Luca et al., 2006] proposed an approach to estimate the effect of external forces from joint torque measurements. Their approach is based on the definition of a dynamic quantity, the residual, which acts as a collision identification signal. Its has two main features: first, it can be computed from the motor torques and the joint positions and velocities; second, its dynamics are governed by a linear differential equation and therefore the residual is zero if no external contact force is applied to the manipulator. [Magrini et al., 2014] proposed a contact estimation strategy which is based on the residual concept proposed by [De Luca et al., 2006]. The estimation of the contact location relies on a Kinect to detect the human posture and on a robot three dimensional rendering to predict possible contact location on the robot surfaces. Given the contact location, a least-square estimation is proposed to obtain the contact force magnitude and direction via a Moore-Penrose pseudo-inverse computation. Only the force components which do not lie on null space of the contact Jacobian can be estimated and this is a limitation of the proposed approach. [Jaeheung Park and Khatib, 2005] considered the problem of estimating contact forces assuming a spring contact model for the external forces. A classical task-space inverse dynamics controllers is adopted to control contact forces. The proposed estimation technique, named Active Observer (AOB), consists of a Kalman filter for estimating the difference between the input command force and measured output force. [Petrovskaya et al., 2007] extended the AOB estimator in [Jaeheung Park and Khatib, 2005] to include the geometric model of the interaction. In particular, the proposed approach defines a probabilistic model of the robot and the environment. The estimation strategy aims at estimating the position of the contact point on the robot, while maintaining constant the model of the environment. Estimation is limited to the single contact case even if experiments include some multi-contact scenarios handled with compliant exploration strategy described by [Jaeheung Park and Khatib, 2005].

In this thesis we present techniques for estimating the intensity of a group of external force-torque in the whole-body case relying on a distributed tactile system and on six-axis force-torque sensor embedded in some of the robot links, as initially explored in [Del Prete et al., 2012]. As we assume that no joint torque measurement are available, we cannot use any residual-based technique.

4.3 Sensor Models

In the previous chapter, we presented the theoretical background underling the dynamics of multibody systems. In this section we will do a brief review of all the sensors that will be used in this and in the following chapters, and their relationship to the quantities that we defined in the previous chapters. For a detailed description of the underling physical principles, the interested reader is referred to [Doebelin, 2003, Chapter 4].

4.3.1 Joint Sensors

Six Axis Force Torque Sensor

In the following, we will always model a six axis force torque sensor as a fixed joint, that connects the two sides of the sensors. This choice has the convenient condition that all the inertial information necessary to describe the sensor are still stored in the model description introduced in Chapter 3. Under this assumption, to fully describe a force-torque sensor we need the following information: the fixed joint of which the force-torque sensor measure the transmitted force-torque, the frame in which the force-torque sensors is reporting its measure, and the applied body on which the measured force-torque is assumed to be applied. For example, if two bodies B and D are rigidly attached by a fixed joint, and ${}_B\mathbf{f}_{B,D}$ is the force-torque applied by B on D , the measure force-torque is given by:

$$y_{\text{FT}} = \sigma_{B,D} X^B {}_B\mathbf{f}_{B,D} \quad (4.1)$$

where the constant $\sigma_{B,D} \in \{-1, 1\}$ accounts for the *direction* of the measured force-torque.

Remark 4.1. *In the next chapters, we will assume that joint-related quantities such as joint positions s , joint velocities \dot{s} are measured. As the definition of these quantities is rather unambiguous given a multibody definition as in Chapter 3, we will not discuss their definition in detail.*

4.3.2 Link Sensors

All the sensors in this subsection are assumed to be rigidly connected to a body, whose frame is indicated with B . Furthermore the sensors are assumed to be mounted such that they report their measurement in a frame S , rigidly attached to the body, while A indicates as usual an inertial frame.

3D Accelerometer

A 3D accelerometer is a sensor that measures the linear part of the *proper sensor* acceleration of frame S , as described in Subsection 2.4.4:

$$y_{\text{acc}} = {}^S R_A{}^A \ddot{o}_S - {}^S R_A{}^A g \quad (4.2)$$

Note that this definition is actually independent from the assumed inertial frame A .

3D Gyroscope

A 3D gyroscope is a sensor that measures the 3D angular velocity of the sensor frame S :

$$y_{\text{gyr}} = {}^S \omega_{A,S} = ({}^A R_S^T {}^A \dot{R}_S)^\vee \quad (4.3)$$

This measurement can be identified simply as the angular part of the *left-trivialized* velocity of the sensor frame S w.r.t. to an inertial frame A :

$$y_{\text{gyr}} = [0_{3 \times 3} \ 1_3] {}^S v_{A,S} = {}^S \omega_{A,S}. \quad (4.4)$$

3D Magnetometer

The 3D magnetometer is a sensor that measures the magnetic field intensity in the sensor frame. In typical application scenario, it is assumed that the measured magnetic field is the earth magnetic field ${}^A b \in \mathbb{R}^3$, a quantity that can be assumed constant in the absolute frame A :

$$y_{\text{mag}} = {}^S R_A{}^A b = {}^S R_B{}^B R_A{}^A b. \quad (4.5)$$

Inertial Measurement Unit

An *Inertial Measurement Unit* is a compound sensor in which a gyroscope, an accelerometer and possibly a magnetometer are packed together to estimate the orientation of the sensor frame w.r.t. an arbitrary *earth-fixed* inertial frame A_{IMU} . The output of the IMU estimation is then an orientation, that can be expressed using Euler Angles, Quaternion or some other representation of $\text{SO}(3)$. Assuming that we consider the output of the IMU to be directly a rotation matrix, we have:

$$y_{\text{IMU}} = {}^{A_{\text{IMU}}} R_S = {}^{A_{\text{IMU}}} R_A{}^A R_B{}^A R_S \quad (4.6)$$

It is important to stress that in general the inertial frame A_{IMU} used by the IMU may be different from the one used in the rest of the analysis. The constant ${}^A_{\text{IMU}} R_A$ must be properly taken in account when designing algorithms that use the output of one or more IMUs.

Remark 4.2. *The force-torque sensors, the gyroscope and accelerometer operating principle is actually always based on deformation or vibration: while this may seems a violation of the basic assumption of rigid body systems, the point is that the dynamics of such part of the systems is usually negligible w.r.t. the dynamics of the multibody system. For example, for a humanoid robot with a total mass of the order 10^2 Kg where even smallest links have a mass in the order of 10^{-2} Kg, the proof mass of an accelerometers sensors can be in the order of 10^{-10} Kg [Andrejašić, 2008].*

4.4 Estimation of the net force-torque

4.4.1 Example: rigid body external force-torque estimation

Imagine that you have a rigid body B attached to a bigger structure through a force-torque sensor and subject to an external force-torque due to the interaction with the environment, as depicted in Figure 4.1. Assuming for simplicity that the force-torque sensor returns the constraint force in the body frame B , we have from the rigid body dynamics of the body expressed w.r.t. *proper sensor acceleration*:

$${}_B \mathbb{M}_B {}^g \alpha_{A,B} + \begin{bmatrix} 0_{3 \times 1} \\ {}_B \omega_{A,B} \end{bmatrix} \bar{\times} {}^* {}_B \mathbb{M}_B \begin{bmatrix} 0_{3 \times 1} \\ {}_B \omega_{A,B} \end{bmatrix} = {}_B f^x + {}_B f^s \quad (4.7)$$

So, assuming a perfect model and the knowledge of the inertial parameters of the body, and measurements of ${}^g \alpha_{A,B}$, ${}_B \omega_{A,B}$ and ${}_B f^s$ at a given instant, we can estimate the external force-torque ${}_B f^x$ as:

$${}_B f^x = {}_B \mathbb{M}_B {}^g \alpha_{A,B} + \begin{bmatrix} 0_{3 \times 1} \\ {}_B \omega_{A,B} \end{bmatrix} \bar{\times} {}^* {}_B \mathbb{M}_B \begin{bmatrix} 0_{3 \times 1} \\ {}_B \omega_{A,B} \end{bmatrix} - {}_B f^s \quad (4.8)$$

Conveniently, in this formula the term ${}_B \mathbb{M}_B {}^g \alpha_{A,B} + \begin{bmatrix} 0_{3 \times 1} \\ {}_B \omega_{A,B} \end{bmatrix} \bar{\times} {}^* {}_B \mathbb{M}_B \begin{bmatrix} 0_{3 \times 1} \\ {}_B \omega_{A,B} \end{bmatrix}$ is the only one that depends on acceleration, velocity and the inertial parameters of the body. For convenience, we will indicate this term as:

$${}_B \phi_B({}^g \alpha_{A,B}, {}_B \omega_{A,B}) := {}_B \mathbb{M}_B {}^g \alpha_{A,B} + \begin{bmatrix} 0_{3 \times 1} \\ {}_B \omega_{A,B} \end{bmatrix} \bar{\times} {}^* {}_B \mathbb{M}_B \begin{bmatrix} 0_{3 \times 1} \\ {}_B \omega_{A,B} \end{bmatrix} \quad (4.9)$$

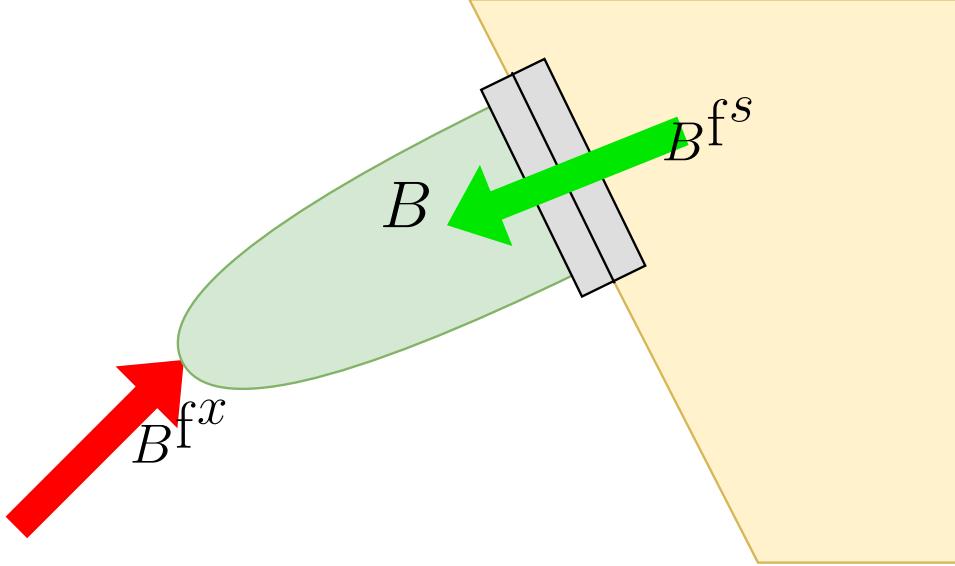


Fig. 4.1 Graphical representation of equation (4.10).

In the following we will omit the dependency on the *proper sensor acceleration* and on the *body angular velocity* and simply indicate this as $B\phi_B$. An interpretation for the physical meaning of $B\phi_B$ is that is the sum of all the force-torque acting on body, both the external ones and the one due to interaction with the other bodies in the system, minus the gravitational force-torque. Even if this term does not include the force-torque due to gravity, to simplify the nomenclature in the following will call it *net force-torque* acting on the body B .

Using the definition of net force-torque, the external force-torque estimation can be compactly expressed as:

$$B^f_x = B\phi_B - B^f_s \quad (4.10)$$

The net force-torque $B\phi^B$ of body B depends on the inertial parameters of the body B , the *proper sensor acceleration* and on the *body angular velocity*. In this chapter, we assume that the inertial parameters of the body are perfectly known. Techniques to estimate those parameters are discussed in Chapter 6 and Chapter 7.

If these inertial parameters are assumed to known, then the *net force-torque* estimation boils down to estimation of the *proper sensor acceleration* $\alpha_{A,B}^g$ and on the *body angular velocity* $B\omega_{A,B}$.

4.4.2 Sensor based estimation

Beside the inertial parameters, to compute the net force-torque it is necessary to estimate the proper sensor acceleration $\alpha_{A,B}^g \in \mathbb{R}^6$ and the *body* angular velocity ${}^B\omega_{A,B} \in \mathbb{R}^3$. In an ideal case, we would want to have a linear accelerometer, and angular accelerometer and a gyroscope rigidly mounted on the body, all reporting their measurements directly in the body frame B . However, typically this is not the case. Firstly there could be an offset between the sensor frame S and the body frame B . Secondly, angular accelerometers are not typically available as off-the-shelf component, and so the information on the angular acceleration is typically obtained by numerical derivative of the gyroscope output. Another strategy used to deal with the absence of direct angular acceleration feedback is to simply ignore the angular acceleration contribution to the net force-torque, assuming its influence in the overall dynamics to be negligible.

Assuming that one of the above strategies was used to compute ${}^S\dot{\omega}_{A,S}$, we will have a *proper sensor acceleration* $\alpha_{A,S}^g$ and an angular velocity expressed ${}^S\omega_{A,S}$, and we want to transform them in the frame B , in which the rest of the model information (such as inertial parameters and joint information) is expressed. From the definition of sensor acceleration and from the propagation rules for mixed acceleration (3.6) we have that:

$${}^B\omega_{A,B} = {}^B R_S {}^S\omega_{A,B}, \quad (4.11a)$$

$$\alpha_{A,B}^g = {}^B X_S \alpha_{A,S}^g + \begin{bmatrix} {}^B\omega_{A,B}^\wedge & {}^B\omega_{A,B}^\wedge & {}^B o_S \\ 0_{3 \times 1} \end{bmatrix}. \quad (4.11b)$$

4.4.3 Kinematic based estimation

In humanoid robotics, it is actually unusual to have more than one IMU mounted on the robot. Typically only one IMU is mounted on one *central* link of the robot, and information on the angular velocity and proper sensor acceleration of all the other links can be obtained through kinematic propagation, using information about joint velocity \dot{s} and acceleration \ddot{s} . This information can be obtained thought high-frequency numerical derivation of high-precision joint encoders.

Under this assumption, we can compute recursively the angular velocity and proper sensor acceleration, considering the *base link* B to be the one in which the IMU is available. In particular we assume that we can compute $\alpha_{A,B}^g$ and ${}^B\omega_{A,B}$ for the link containing the IMU using equations (4.11). Then, given an arbitrary link L , its sensor proper acceleration and body

angular velocity can be computed from (3.7) and (3.4a):

$$\begin{aligned} {}^L\omega_{A,L} &= {}^L R_B(s) {}^B\omega_{A,B} + {}^L\omega_{B,L}(s, \dot{s}), \\ \alpha_{A,L}^g &= {}^L X_B(s) \alpha_{A,B}^g + \alpha_{B,L}(s, \dot{s}, \ddot{s}) + \end{aligned} \quad (4.12)$$

$$\begin{aligned} &\left[{}^L R_B(s) [2({}^B\omega_{A,B} \times {}^B\dot{o}_L(s, \dot{s})) + {}^B\omega_{A,B} \times ({}^B\omega_{A,B} \times {}^B o_L(s))] \right] \\ &\quad ({}^L R_B(s) {}^B\omega_{A,B}) \wedge {}^L\omega_{B,L}(s, \dot{s}) \end{aligned} \quad (4.13)$$

$$= {}^L X_B(s) \alpha_{A,B}^g + \alpha_{B,L}(s, \dot{s}, \ddot{s}) + \quad (4.14)$$

$$\begin{bmatrix} 2{}^L R_B(s) {}^B\omega_{A,B}^\wedge & 0_{3 \times 3} \\ 0_{3 \times 3} & {}^L R_B(s) {}^B\omega_{A,B}^\wedge \end{bmatrix} {}^{L[B]} v_{L,B} + \quad (4.15)$$

$$\begin{bmatrix} {}^L R_B(s) [{}^B\omega_{A,B} \times ({}^B\omega_{A,B} \times {}^B o_L(s))] \\ 0_{3 \times 1} \end{bmatrix}$$

where ${}^L R_B(s)$, ${}^L X_B(s)$, ${}^L\omega_{B,L}(s, \dot{s})$, ${}^B\dot{o}_L(s, \dot{s})$ and $\alpha_{B,L}(s, \dot{s}, \ddot{s})$ are given by the relative forward kinematics.

4.4.4 Hybrid estimation

If only n_{IMU} out of n_L links of the robot are equipped with IMUs, it is possible to mix the two approaches by splitting the multibody system in n_{IMU} submodels, each of it containing several connected links, of which only one is equipped with an *IMU*. For all other links in the subgraph, the net force-torque of each link can be computed by propagating the kinematic information using the joint information, using the equations presented in the previous subsection.

4.5 Multibody External Force-Torque Estimation

4.5.1 Force-Torque Sensors Induced Submodel Decomposition

In this section we consider the generalization of single-body external force-torque estimation depicted in Fig 4.1 to the multibody case, depicted in Figure 4.2.

For exploiting the measures of internal force-torque sensors, it is convenient to consider independently each submodel induced by cutting the multibody-model along the fixed joints that contain an internal force-torque sensor. If we have n force-torque sensors, we can define $n + 1$ submodels. We indicate with \mathfrak{M} the set of different submodels, with $sm \in \mathfrak{M}$ a specific submodel, and with \mathcal{L}_{sm} the set of the links belonging to submodel sm .

Furthermore, for each link $L \in \mathfrak{L}_{sm}$ we indicate with $\mathfrak{D}_{sm}(L)$ the set of links that are connected with L in the full model, but that belong to a different submodel, i.e.:

$$\mathfrak{D}_{sm}(L) := \{D \in \mathfrak{L} \mid \{L, D\} \in \mathfrak{J} \wedge D \notin \mathfrak{L}_{sm}\} \quad (4.16)$$

4.5.2 External Force-Torque Estimation

In the following, we will use this assumptions.

Assumption 4.1. *The contact surface and the link to which it belongs of the external force-torques are both known, either from a-priori assumptions or from the measurements from a distributed tactile skin. The a-priori assumptions or the skin do not provide information on the intensity of the external force-torque.*

Assumption 4.2. *The net force-torque ${}_L\phi_L$ of each link is assumed to be known, estimated using one of the techniques described in Section 4.4.*

Property 4.1. *Given a link $L \in \mathfrak{L}$ the relation between the net force-torque ${}_L\phi_L$, the net external force-torque ${}_L\mathbf{f}_B^x$ and the joint internal force-torques is given by:*

$${}_L\phi_L = {}_L\mathbf{f}_L^x + \sum_{D \in \mathfrak{N}(L)} {}_L\mathbf{f}_{D,L}. \quad (4.17)$$

This equation is just a generalization of (4.7) to all the forces interacting with a given link in a multibody system.

A similar result holds for the multibody system as an whole, as stated in the next Proposition.

Theorem 4.1. *Given a multibody system in which the external force-torques are acting on only a subset of the links $\mathfrak{C} \subseteq \mathfrak{L}$, the sum of all the net force-torques of each link is equal to the sum of all the external force-torques acting on the multibody system, provided that all quantities are transformed in a common frame B , i.e. :*

$$\sum_{L \in \mathfrak{L}} {}_B X^L {}_L\phi_L = \sum_{L \in \mathfrak{C}} {}_B X^L {}_L\mathbf{f}_L^x. \quad (4.18)$$

The proof of this theorem is given in Subsection 4.8.1.

With respect to the single body case using Theorem 4.1 we can only estimate the sum of all the external force-torque acting on the multibody

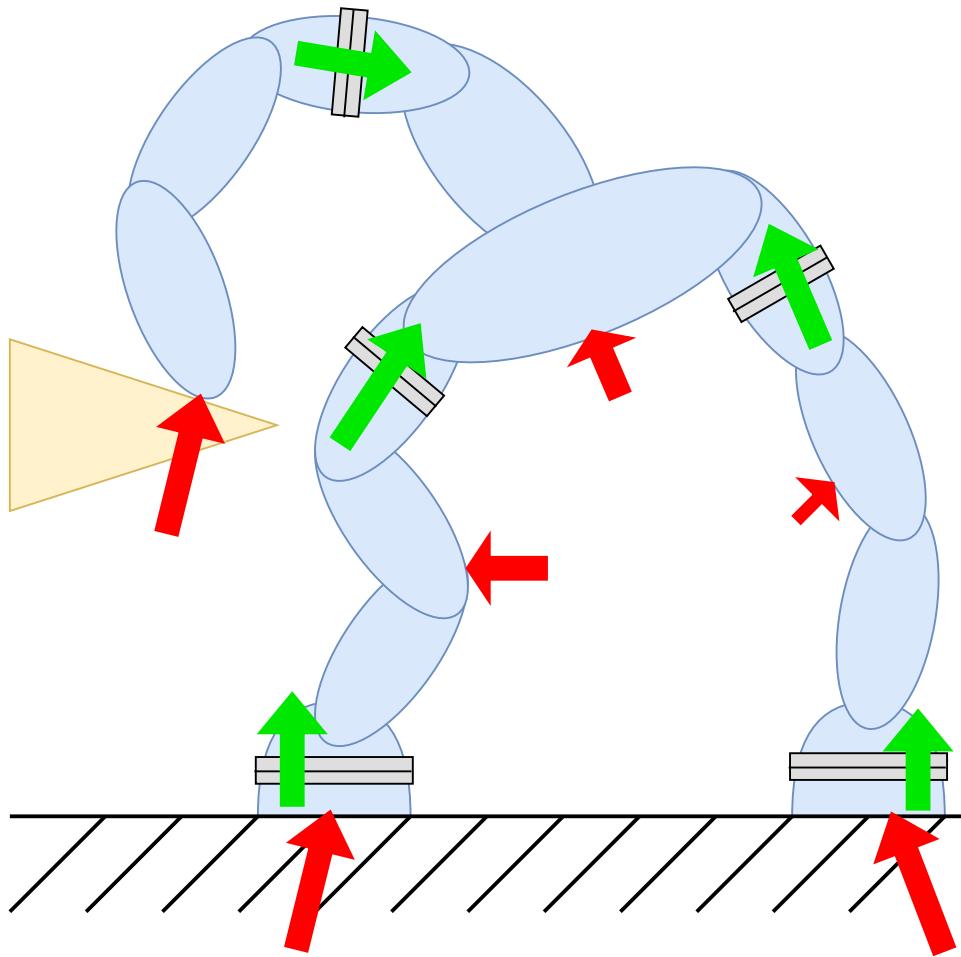


Fig. 4.2 Example of a multibody system with internal six-axis force-torque sensors. Measured force-torques are indicated in green, while unknown contact force-torques are drawn in red. There are $n = 5$ force-torque sensor in the system, that is then decomposed in $n + 1 = 6$ submodels for external force-torque estimation.

system. However it is typically of interest to estimate the intensity of each contact force-torque independently from the others, for example for force control. On the other hand, we need to use in some way the information coming from the force-torque sensor, that measure *internal* force-torque, to estimate *external* force-torque.

A way of obtaining an equation similar to Theorem 4.1 that contains the available sensors measures, is to split the multibody model in several submodels, along the joints in which force-torque sensors are mounted, as described in Subsection 4.5.1. We can then apply Theorem 4.1 to each submodel $sm \in \mathfrak{M}$, that we can rewrite as:

$$\sum_{L \in \mathfrak{L}_{sm}} {}_B X^L L \phi_L = \sum_{L \in (\mathfrak{C} \cap \mathfrak{L}_{sm})} {}_B X^L L f_L^x + \sum_{L \in \mathfrak{L}_{sm}} \sum_{D \in \mathfrak{J}_{sm}(L)} {}_B X^D D f_{D,L}. \quad (4.19)$$

The term $\sum_{L \in \mathfrak{L}_{sm}} \sum_{D \in \mathfrak{J}_{sm}(L)} {}_B X^D D f_{D,L}$ represents the effect on the submodel of the force-torques transmitted by the six-axis sensors. For a given force-torque sensor embedded in the joint $\{D, L\} \in \mathfrak{J}$, the value of $D f_{D,L}$ can be obtained from the sensor measurement y_{FT} by inverting equation (4.1):

$$D f_{D,L} = {}_D X^S \sigma_{D,L} y_{FT}. \quad (4.20)$$

In the rest of the chapter we will assume that the joint force-torque $D f_{\{D,L\}}$ is always available for each joint in which a six-axis force-torque sensors is mounted, without manually indicating its value from equation (4.20).

Given a submodel $sm \in \mathfrak{M}$, if we assume that on the submodel the external force-torques are acting only on a single link called C (i.e. $(\mathfrak{C} \cap \mathfrak{L}_{sm}) = \{C\}$) then this external force-torque can be computed exactly from (4.19) as:

$$C f_C^x = \sum_{L \in \mathfrak{L}_{sm}} {}_C X^L L \phi_L - \sum_{L \in \mathfrak{L}_{sm}} \sum_{D \in \mathfrak{J}_{sm}(L)} {}_C X^D D f_{D,L}. \quad (4.21)$$

4.5.3 Estimation of external forces acting on multiple link in a submodel

If the external forces are acting on more than one link, the estimation problem given by (4.19) is indeterminate, as the number of unknowns is greater than 6, the number of given equations. A possible solution in this case is to solve the equation in the least-square sense as proposed in Del Prete et al. [2012], but in this case there is no guarantee that the resulting estimated force-torques are the “real” ones. In this sense, the least-square estimation

should be seen as fallback solution, rather than a sound estimation technique.

4.6 Joint Torques Estimation

Once we obtained a set of external force-torques, a strictly related problem is how to estimate the internal torques, i.e. the component of the inter-link forces acting along the degree of freedom of the joints. The joint torques estimation is crucial because this quantity is directly related to the motors that are actuating those joints, and the input to the motors is in the end the ultimate control input available in robots.

We have that the torques of the joint connecting links E and F are just the projection of the joint force-torque on the joint motion subspace:

$$\tau_{\{E,F\}} = \langle {}^F s_{E,F}, {}^F f_{E,F} \rangle = \langle {}^E s_{F,E}, {}^E f_{F,E} \rangle. \quad (4.22)$$

The joint torque estimation problem is then solved if an estimation of the joint internal force-torque is available. The internal force-torque estimation problem has a trivial solution using the following property.

Definition 4.1. A set of estimated net force-torques ${}_L \phi_L$, $L \in \mathfrak{L}$ and of external force-torques ${}_L f_L^x$, $L \in \mathfrak{L}$ is consistent if they follow equation (4.18).

Property 4.2. The outcome of the estimation of force-torques algorithm presented in Section 4.5 is always consistent, as defined in Definition 4.1.

Given a set of net force-torques for each body and a set of *consistent* external force-torques, all the internal joint force-torques are totally determined. In particular, the internal force-torque can be written in two equivalent ways:

$${}^F f_{E,F} = - {}_E f_{F,E} \quad (4.23a)$$

$${}^F f_{E,F} = \sum_{L \in \gamma_E(F)} {}_F X^L ({}_L \phi_L + {}_L f_L^x) \quad (4.23b)$$

$${}_E f_{F,E} = \sum_{L \in \gamma_F(E)} {}_E X^L ({}_L \phi_L + {}_L f_L^x) \quad (4.23c)$$

where $\gamma_E(F)$ is the set of the links belonging to the subtree starting at link F , given E as a base link, defined originally in Definition 3.9.

The two sums are equivalent thanks to the fact the the net force-torques and the external force-torques are *consistent*.

Combining equations (4.23) and (4.22) is possible to estimate the joint torques using net force-torques as estimated in Section 4.4 and external force-torques estimated in Section 4.5 :

$$\tau_{\{E,F\}} = \left\langle {}^F s_{E,F}, \sum_{L \in \gamma_E(F)} {}_F X^L ({}_L \phi_L + {}_L f_L^x) \right\rangle = \quad (4.24a)$$

$$= \left\langle {}^E s_{F,E}, \sum_{L \in \gamma_F(E)} {}_E X^L ({}_L \phi_L + {}_L f_L^x) \right\rangle. \quad (4.24b)$$

Note that (4.24) is just an alternative formulation for the line of (3.64) relative to $\tau_{\{E,F\}}$, and it is usually computed using the algorithms for computing *Inverse Dynamics* [Featherstone, 2008].

Relying on (4.24) for internal force-torque estimation may seem risky, as it creates a dependency of the torques on the net force-torque and external force-torque of the link of all the robot, and this could be subject to accumulation error. In the next theorem we see how most of the torques computed as in (4.24) are actually function of a much more “local set of measurements”.

Theorem 4.2. *Assume that:*

- (E, F) is a 1-dof joint,
- (G, H) is a fixed joint that contains a force-torque sensor,
- F is closer to G than E (i.e. $G \in \gamma_E(F)$).
- the path connecting (E, F) to (G, H) is branchless,
- the path connecting (E, F) to (G, H) is free of external force-torques and other force-torque sensors,
- E, F and G belong to the submodel $sm \in \mathfrak{M}$.

Then $\tau_{\{E,F\}}$ given by (4.24) can be equivalently written as:

$$\tau_{\{E,F\}} = \left\langle {}^F s_{E,F}, \sum_{\gamma_E(F) \cap \mathfrak{L}_{sm}} {}_F X^L {}_L \phi_L + {}_F X^G {}_G f_{G,H} \right\rangle. \quad (4.25)$$

Remark 4.3. This means that the torque estimate is just a linear function of the force-torque sensor measure and of the net force-torque of a limited set of links, the one connecting the joint $\{E, F\}$ to the joint $\{G, H\}$. This ensures that any error in all the other force-torque sensors measurements or in the estimation of net force-torque of all the other links in the robot will not influence the estimation of $\tau_{\{E,F\}}$.

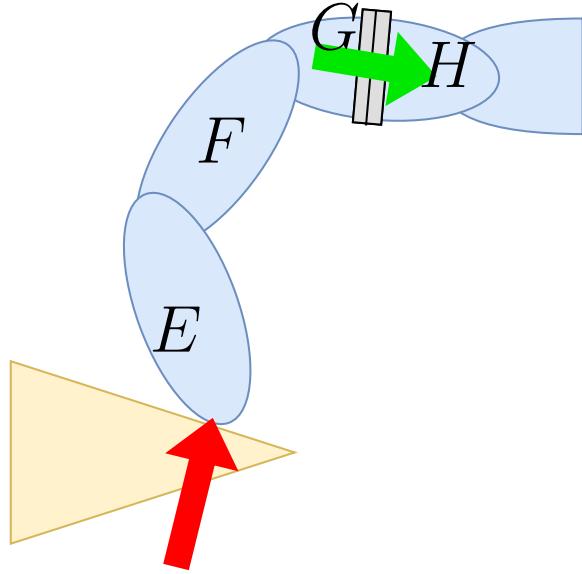


Fig. 4.3 Example of torque estimation.

4.6.1 Example of torque estimation

Let's focus on the upper part of the model depicted in Figure 4.2, depicted in 4.3.

As explained in this section, the joint torques can simply be estimated by (4.24), i.e. *inverse dynamics*, once a *consistent* set of net force-torques and external force-torques have been estimated. However, it is interesting to see on which measurements the estimate of the joint $\{E, F\}$ actually depends on, using Theorem 4.2. In this case we have that $\mathfrak{L}_{sm} = \{D, E, F, G\}$ and $\gamma_E(F) \cap \mathfrak{L}_{sm} = \{F, G\}$. Writing (4.25) in this specific case we have:

$$\tau_{\{E, F\}} = \langle {}^F s_{E, F}, {}_F \phi_F + {}_F X {}^G G \phi_G + {}_F X {}^G G f_{G, H} \rangle, \quad (4.26)$$

where $G f_{G, H}$ is obtained directly from the force-torque sensor measurements. In the end the estimates of the joint $\tau_{\{E, F\}}$ depends only on the net force-torque of links F and G ($_F \phi_F, {}_G \phi_G$), on the measurement of the position of joint $\{F, G\}$ and its geometrical model (${}_F X {}^G$) and on the measurement of the nearby force-torque sensor mounted on joint $\{G, H\}$ (${}_G f_{G, H}$). Then, all other estimation errors that could be present on the robot do not affect the estimate of $\tau_{\{E, F\}}$, even if they are present in equation (4.24).

4.7 Six-Axis Force-Torque Sensors Model-based Offset Calibration

As will be described in detail in 5, the six-axis force-torque sensors bias needs to be determined during the startup of the system. For this reason, we need to have a way find the expected value of the measured sensor assuming the perfect knowledge of the model and some additional assumptions on the external force-torque.

4.7.1 Offset Calibration with the external forces acting on a Single Link

Assumption 4.3. *During force-torque bias sensor calibration, we assume that only one external force-torque is acting on the multibody system. We call C the link on which the unique external 6D force is acting.*

Remark 4.4. *Assumption 4.3 can be automatically checked using a distributed tactile system.*

If only one external force-torque is acting on the system, its value can be computed from Newton-Euler equations, using Theorem 4.1:

$$f_C^x = \sum_{L \in \mathfrak{L}} {}_B X^L {}_L \phi_L \quad (4.27)$$

Once the unique external force-torque has been determined using, joint force-torque for each joint can be computed using the equations in (4.23). In particular only one of the two equations (4.23c)-(4.23b) will contain the term related to external force-torque. By choosing the other expression for all the joints, the joint force-torque can be written as a function of the net force-torques. In particular given a joint $\{E, F\}$, using C as the base assume that $\lambda_C(F) = E$. Then $\gamma_E(F) = \gamma_C(F)$ and:

$${}_F f_{E,F} = \sum_{L \in \gamma_C(E)} {}_F X^L \phi_L. \quad (4.28)$$

4.8 Proofs

4.8.1 Proof of Theorem 4.1

We can sum the equations given by (4.17) for each link L by multiplying them for BX^L , resulting in:

$$\begin{aligned} \sum_{L \in \mathfrak{L}} BX^L L \phi_L &= \sum_{L \in \mathfrak{L}} BX^L \left(L f_L^x + \sum_{D \in \mathfrak{N}(L)} L f_{D,L} \right) = \\ &= \sum_{L \in \mathfrak{L}} BX^L L f_L^x + \sum_{L \in \mathfrak{L}} \sum_{D \in \mathfrak{N}(L)} B f_{D,L}. \end{aligned} \quad (4.29)$$

Focusing on the last addend of the right term of the equation we have:

$$\begin{aligned} \sum_{L \in \mathfrak{L}} \sum_{D \in \mathfrak{N}(L)} B f_{D,L} &= \\ &= \sum_{\{E,F\} \in \mathfrak{J}} B f_{E,F} + B f_{F,E} = \\ &= \sum_{\{E,F\} \in \mathfrak{J}} B f_{E,F} - B f_{E,F} = 0_{6 \times 1} \end{aligned} \quad (4.30)$$

Plugging (4.30) in (4.29) one obtains (4.18). ■

4.8.2 Proof of Theorem 4.2

From (4.24) we have:

$$\tau_{\{E,F\}} = \langle {}^F s_{E,F}, {}^F f_{E,F} \rangle = \left\langle {}^F s_{E,F}, \sum_{L \in \gamma_E(F)} {}^F X^L (L \phi_L + L f_L^x) \right\rangle$$

From (4.23b) instead we have that:

$${}^G f_{G,H} = \sum_{L \in \gamma_G(H)} {}^G X^L (L \phi_L + L f_L^x), \quad (4.31)$$

and that:

$$\gamma_E(F) = (\gamma_E(F) \cap \mathfrak{L}_{sm}) \cup \gamma_G(H).$$

We can then decompose the expression of ${}^F\mathbf{f}_{E,F}$ as:

$$\begin{aligned} {}^F\mathbf{f}_{E,F} &= \sum_{L \in \gamma_E(F)} {}_F X^L ({}_L \phi_L + {}_L \mathbf{f}_L^x) = \\ &= \sum_{L \in \gamma_E(F) \cap \mathfrak{L}_{sm}} {}_F X^L ({}_L \phi_L + {}_L \mathbf{f}_L^x) + \sum_{L \in \gamma_G(H)} ({}_F X^L {}_L \phi_L + {}_L \mathbf{f}_L^x) = \\ &= \sum_{L \in \gamma_E(F) \cap \mathfrak{L}_{sm}} {}_F X^L ({}_L \phi_L) + {}_F X^G {}_G \mathbf{f}_{G,H}, \end{aligned}$$

where we made use of the assumption that no external force-torques are present on the links in $\gamma_E(F) \cap \mathfrak{L}_{sm}$ and the expression of ${}_G \mathbf{f}_{G,H}$ given in (4.31). Substituting this expression of ${}_F \mathbf{f}_{E,F}$ in (4.24) gives the result. ■

Chapter 5

Six Axis Force Torque Sensors In Situ Calibration

5.1 Introduction

The importance of sensors in a control loop goes without saying. Measurement devices, however, can seldom be used *sine die* without being subject to periodic calibration procedures.

This is in particularly true for six-axis force-torque sensors, whose calibration procedures may require to move the sensor from the hosting system to specialized laboratories, which are equipped with the tools for performing the calibration of the measuring device. This chapter presents techniques to calibrate strain gauges six-axis force-torque sensors *in situ*, i.e. without the need of removing the sensor from the hosting system, by exploiting the structure of rigid body dynamics.

Calibration of six-axis force-torque sensors has long attracted the attention of the robotic community [Braun and Wörn, 2011]. The commonly used model for predicting the force-torque from the raw measurements of the sensor is an affine model. This model is sufficiently accurate since these sensors are mechanically designed and mounted so that the strain deformation is (locally) linear with respect to the applied forces and torques. Then, the calibration of the sensor aims at determining the two components of this model, i.e. a six-by-six matrix and a six element vector. These two components are usually referred to as the sensor's *calibration matrix* and *offset*, respectively. In standard operating conditions, relevant changes in the calibration matrix may occur in months. As a matter of fact, manufacturers such as ATI [ATI Industrial Automation, 2014] and Weiss Robotics [Weiss

Robotics, 2013] recommend to calibrate force-torque sensors at least once a year. Preponderant changes in the sensor’s offset can occur in hours, however, and this in general requires to estimate the offset before using the sensor. The offset models the sensitivity of the semiconductor strain gauges with respect to temperature.

Classical techniques for determining the offset of a force-torque sensor exploit the aforementioned affine model between the raw measurements and an *a-priori* knowledge of the load attached to the sensor, as described in Section 4.7. In particular, if no load is applied to the measuring device, the output of the sensor corresponds to the sensor’s offset. This offset identification procedure, however, cannot be always performed since it may require to take the hosting system apart in order to unload the force-torque sensor. Another widely used technique for offset identification is to find two sensor’s orientations that induce equal and opposite loads with respect to the sensor. Then, by summing up the raw measurements associated with these two orientations, one can estimate the sensor’s offset. The main drawback of this technique is that the positioning of the sensor at these opposite configurations may require to move the hosting system beyond its operating domain.

Assuming a preidentified offset, non-*in situ* identification of the calibration matrix is classically performed by exerting on the sensor a set of force-torques known *a priori*. This usually requires to place sample masses at precise relative positions with respect to the sensor. Then, by comparing the known gravitational force-torque with that measured by the sensor, one can apply linear least square techniques to identify the sensor’s calibration matrix. For accurate positioning of the sample masses, the use of robotic positioning devices has also been proposed in the specialized literature [Uchiyama et al., 1991] [Watson and Drake, 1975].

To reduce the number of sample masses, one can apply constrained forces, e.g. constant norm forces, to the measuring device. Then these constraints can be exploited during the computations for identifying the calibration matrix [Voyles et al., 1997]. To avoid the use of added masses, one can use a supplementary already-calibrated measuring device that measures the force-torque exerted on the sensors [Faber et al., 2012] [Oddo et al., 2007]. On one hand, this calibration technique avoids the problem of precise positioning of the added sample masses. On the other hand, the supplementary sensor may not always be available. All above techniques, however, cannot be performed *in situ*, thus being usually time consuming and expensive.

To the best of our knowledge, the first *in situ* calibration method for force-torque sensors was proposed in [Shimano and Roth, 1977]. But this

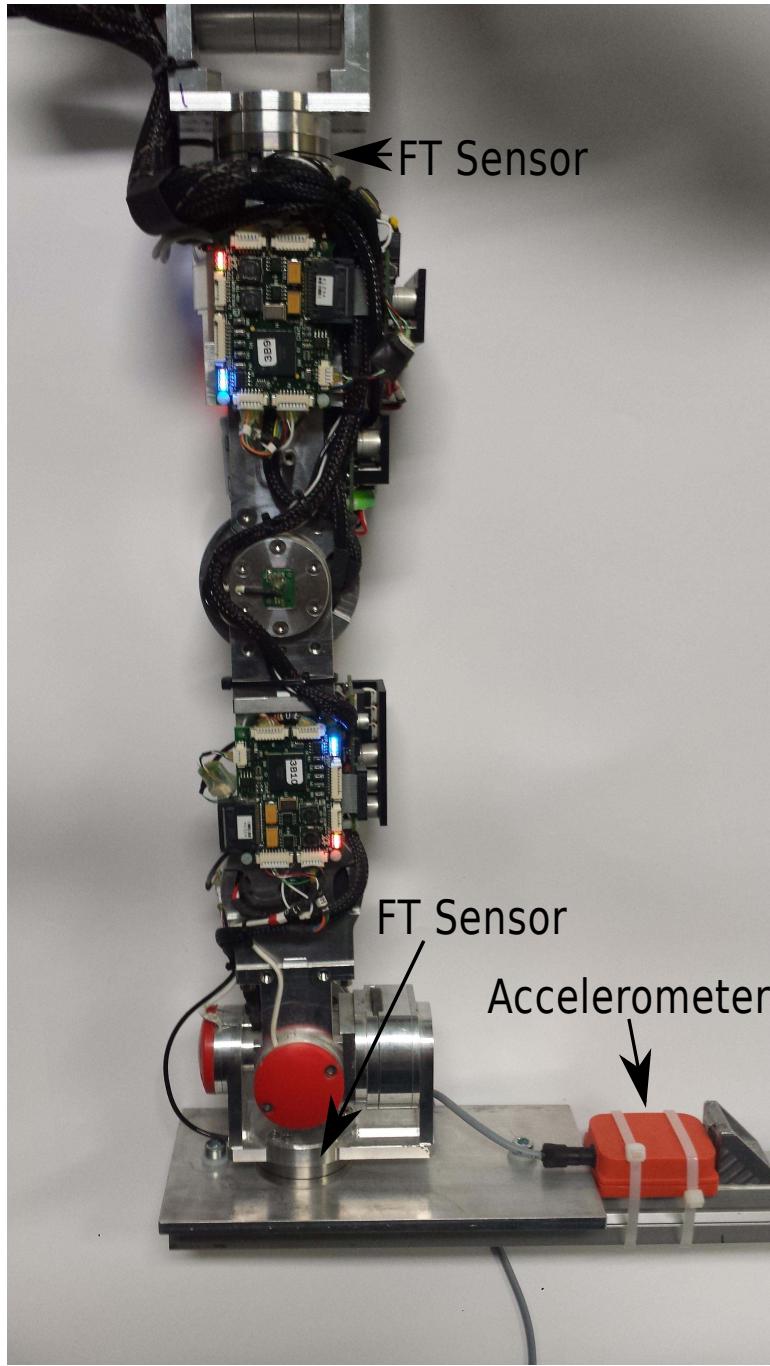


Fig. 5.1 iCub's leg with the two force/torque sensors and an additional accelerometer.

method exploits the topology of a specific kind of manipulators, which are equipped with joint torque sensors then leveraged during the estimation. A recent result [Gong et al., 2013] proposes another *in situ* calibration technique for six-axis force-torque sensors. The technical soundness of this work, however, is not clear to us. In fact, we show that a necessary condition for identifying the calibration matrix is to take measurements for at least three different added masses, and this requirement was not met by the algorithm [Gong et al., 2013]. Another *in situ* calibration technique for force-torque sensors can be found in [Roozbahani et al., 2013]. But the use of supplementary already-calibrated force-torque/pressure sensors impairs this technique for the reasons we have discussed before.

This chapter presents *in situ* calibration techniques for six-axis force-torque sensors using accelerometer measurements. The proposed method exploits the geometry induced by the affine model between the raw measurements and the gravitational force-torque applied to the sensor. In particular, it relies upon the properties that all gravitational raw measurements belong to a three-dimensional space, and that in this space they form an ellipsoid. We first propose a method for estimating the sensor's offset, and then a procedure for identifying the calibration matrix. The latter is independent from the former, but requires to add sample masses to the rigid body attached to the sensor. Both methods are independent from the inertial characteristics of the rigid body attached to the sensor. The proposed algorithms are validated on the iCub platform by calibrating two force-torque sensors embedded in the robot leg.

The simplified notation used in this chapter is listed in the following box.

Notation used in Chapter 5	
$f := {}_S f \in \mathbb{R}^6$	Force-Torque measured by the Force-Torque sensor.
i	Calibration matrix.
$C \in \mathbb{R}^{6 \times 6}$	Raw measurement of the strain gauges.
$r \in \mathbb{R}^6$	Offset of the FT sensor, in the raw measurement space.
$b \in \mathbb{R}^6$	Mass of the body attached to the FT sensor.
$m \in \mathbb{R}$	Center of mass, expressed w.r.t. to sensor frame S .
$c := {}^S c \in \mathbb{R}^3$	Gravitation acceleration, expressed w.r.t. to inertial frame A .
$g := {}^A g \in \mathbb{R}^3$	Gravitation acceleration, expressed w.r.t. to sensor frame S .
$\bar{g} := {}^S R_A {}^A g \in \mathbb{R}^3$	

The chapter is organized as follows. Section 5.2 presents the problem statement with the assumptions. Section 5.3 details the proposed method for the calibration of six-axis force-torque sensors. Validations of the approach are presented in Section 5.4.

5.2 Problem statement and assumptions

We assume that the model for predicting the force-torque (also called wrench) from the raw measurements is an affine model, i.e.

$$f = C(r - b), \quad (5.1)$$

where $f \in \mathbb{R}^6$ is the wrench exerted on the sensor expressed in the sensor's frame, $r \in \mathbb{R}^6$ is the raw output of the sensor, $C \in \mathbb{R}^{6 \times 6}$ is the invertible calibration matrix, and $b \in \mathbb{R}^6$ is the sensor's bias or offset. The calibration matrix and the offset are assumed to be constant.

We assume that the sensor is attached to a rigid body of (constant) mass $m \in \mathbb{R}^+$ and with a center of mass whose position w.r.t. the sensor frame S is characterized by the vector $c \in \mathbb{R}^3$.

The gravity 3D force applied to the body is given by

$$m\bar{g} = m{}^A R_B^T g, \quad (5.2)$$

with $\bar{g}, g \in \mathbb{R}^3$ the gravity acceleration expressed w.r.t. the inertial and sensor frame, respectively. The gravity acceleration \bar{g} is constant, so the vector g does not have a constant direction, but has a constant norm.

Finally, we make the following main assumption.

Assumption 5.1. *The raw measurements r are taken for static configurations of the rigid body attached to the sensor, i.e. the angular velocity of the frame S is always zero. Also, the gravity acceleration g is measured by an accelerometer installed on the rigid body. Furthermore, no external force-torque, but the gravity force, acts on the rigid body. Hence*

$$\mathbf{f} = M(m, c)g, \quad (5.2a)$$

$$M(m, c) := m \begin{pmatrix} 1_3 \\ c \times \end{pmatrix}. \quad (5.2b)$$

Remark 5.1. *We implicitly assume that the accelerometer frame is aligned with the force-torque sensor frame. This is a convenient, but non necessary, assumption. In fact, if the relative rotation between the sensor frame S and the accelerometer frame is unknown, it suffices to consider the accelerometer frame as the sensor frame S .*

Under the above assumptions, what follows proposes a new method for estimating the sensor's offset o and for identifying the sensor's calibration matrix C without the need of removing the sensor from the hosting system.

5.3 In Situ Calibration Methods

The proposed methods rely on the the geometry induced by the models (5.3) and (5.1).

5.3.1 The geometry of the raw measurements

First, observe that

$$\text{rank}(M) = 3.$$

As a consequence, all wrenches \mathbf{f} belong to the three-dimensional subspace given by the $\text{span}(M) \subset \mathbb{R}^6$. In view of this, we can state the following lemma.

Lemma 5.1. All raw measurements r belong to a three dimensional affine space, i.e. there exist a point $r_m \in \mathbb{R}^6$, an orthonormal basis $U_1 \in \mathbb{R}^{6 \times 3}$, and for each $r \in \mathbb{R}^6$ a vector $\lambda \in \mathbb{R}^3$ such that

$$r = r_m + U_1\lambda. \quad (5.3)$$

Also, the vector λ belongs to a three-dimensional ellipsoid.

Proof. From (5.1) and (5.3), one has:

$$r = |g|C^{-1}M\hat{g} + b, \quad (5.4)$$

where $\hat{g} := g/|g|$. The matrix $C^{-1}M \in \mathbb{R}^{6 \times 3}$ is of rank 3. Consequently, all raw measurements r belong to an affine three-dimensional space defined by the point b and the basis of $\text{span}(C^{-1}M)$. Now, define $P \in \mathbb{R}^{3 \times 6}$ as the projector of r onto $\text{span}(C^{-1}M)$. Then, the projection $p \in \mathbb{R}^3$ of r onto this span is given by

$$p = Pr = |g|PC^{-1}M\hat{g} + Pb. \quad (5.5)$$

By considering all possible orientations of the sensor's frame \mathcal{S} , then the gravity direction \hat{g} spans the unit sphere. Consequently, the vector p belongs to the *span of the unit sphere applied to the linear transformation $|g|PC^{-1}M$* , i.e. an *ellipsoid centered at the point Po* . This in turn implies that when decomposing the vector r as in (5.3), the vector λ necessarily belongs to a three-dimensional ellipsoid. \square

To provide the reader with a better comprehension of the above lemma, assume that $r \in \mathbb{R}^3$ and that the affine subspace is a plane, i.e. a two-dimensional space. As a consequence, all measurements belong to an ellipse lying on this plane – see Figure 5.2. Observe also that given a point $\lambda \in \mathbb{R}^2$ on the plane and expressed w.r.t. the basis U_1 , the relationship (5.3) provides with the components of this point in the space \mathbb{R}^3 . By leveraging on the above lemma, the next two sections propose a method to estimate the sensor's offset and calibration matrix.

5.3.2 Method for estimating the sensor's offset

Assume that one is given with a set of measurements (r_i, g_i) with $i \in \{1 \dots N\}$ corresponding to several body's static orientations. Then, let us show how we can obtain the basis (r_m, U_1) in (5.3) associated with all measurements r_i , and how this basis can be used for estimating the offset o .

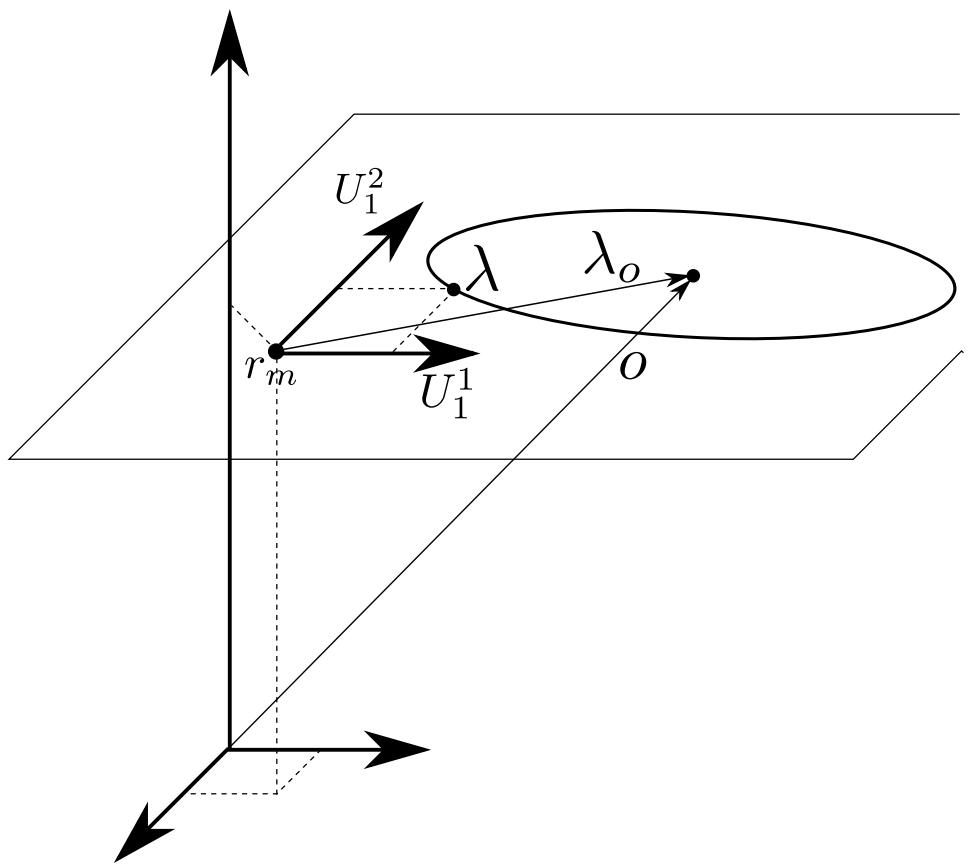


Fig. 5.2 Example when $r \in \mathbb{R}^3$ and $U_1 = (U_1^1, U_1^2) \in \mathbb{R}^{3 \times 2}$.

Observe that the point r_m can be chosen as any point that belongs to the affine space. Then, a noise-robust choice for this point is given by the mean value of the measurements r_i ,

$$r_m = \frac{1}{N} \sum_{i=1}^N r_i. \quad (5.6)$$

An orthonormal basis U_1 can be then obtained by applying the singular value decomposition on the matrix resulting from the difference between all measurements and r_m , i.e.

$$(\tilde{r}_1, \dots, \tilde{r}_n) = USV^\top, \quad (5.7)$$

where

$$\tilde{r}_i := r_i - r_m,$$

and $U \in \mathbb{R}^{6 \times 6}$, $S \in \mathbb{R}^{6 \times N}$, $V \in \mathbb{R}^{N \times N}$ are the (classical) matrices obtained from the singular value decomposition. Note that only the first three elements on the diagonal of S are (significantly) different from zero since all measurements r_i belong to a three dimensional subspace. Consequently, (an estimate of) the orthonormal basis U_1 is given by the first three columns of the matrix U .

With (r_m, U_1) in hand, the offset o can be easily estimated. First, note that equation (5.3) holds for all points belonging to the three dimensional space. Hence, it holds also for the offset o being the center of the ellipsoid (see Figure 5.2), i.e.

$$o = r_m + U_1 \lambda_b. \quad (5.8)$$

Then to estimate the offset o belonging to \mathbb{R}^6 , we can estimate the coordinates λ_b in the subspace \mathbb{R}^3 . In view of $U_1^\top U_1 = I_3$, multiplying the above equation times U_1^\top yields

$$\lambda_b := U_1^\top (o - r_m). \quad (5.9)$$

Now, by subtracting r_m from (5.4) and multiplying the resulting equation by U_1^\top , one has

$$U_1^\top \tilde{r}_i = K g_i + \lambda_b, \quad (5.10)$$

where

$$K := U_1^\top C^{-1} M \quad \text{and} \quad \lambda_b$$

are the unknowns in the above equation. In view of (2.9) the equation (5.10) can be written by stacking the obtained vectors for all measurements as

$$\bar{r} = \Gamma x, \quad (5.10a)$$

$$\bar{r} := (\tilde{r}_1^\top U_1, \dots, \tilde{r}_N^\top U_1)^\top \in \mathbb{R}^{3N \times 1}, \quad (5.10b)$$

$$\Gamma := \begin{pmatrix} g_1^\top \otimes 1_3, 1_3 \\ \vdots \\ g_N^\top \otimes 1_3, 1_3 \end{pmatrix} \in \mathbb{R}^{3N \times 12}, \quad (5.10c)$$

$$x := \begin{pmatrix} \text{vec}(K) \\ \lambda_b \end{pmatrix} \in \mathbb{R}^{12 \times 1}. \quad (5.10d)$$

Solving the equation (5.10a) for the unknown x in the least-square sense provides with an estimate $\hat{\lambda}_o \in \mathbb{R}^3$ of λ_b . To obtain the coordinates of this point w.r.t. the six-dimensional space, i.e. the raw measurements space, we apply the transformation (5.8) as follows:

$$\hat{b} = r_m + U_1 \hat{\lambda}_b.$$

5.3.3 Method for estimating the sensor's calibration matrix

In this section, we assume no offset, i.e. $o = 0$, which means that this offset has already been estimated by using one of the existing methods in the literature or by using the method described in the previous section. Consequently, the relationship between the set of measurements (r_i, g_i) and the body's inertial characteristics, i.e. mass and center of mass, is given by

$$Cr_i = Mg_i.$$

In addition, we also assume that the body's inertial characteristics can be modified by adding sample masses at specific relative positions w.r.t. the sensor frame \mathcal{S} . As a consequence, the matrix M in the above equation is replaced by M_j , i.e.

$$Cr_i^j = M_j g_i^j, \quad (5.11)$$

where j indicates that new inertial characteristics have been obtained by adding sample masses. Observe that M_j can then be decomposed as follows

$$\begin{aligned} M_j &:= M_b + M_a^j \\ &= m \begin{pmatrix} 1_3 \\ c \times \end{pmatrix} + m_a^j \begin{pmatrix} 1_3 \\ c_a^j \times \end{pmatrix}, \end{aligned}$$

where (m_a^j, c_a^j) are the mass and the vector of the center of mass, expressed w.r.t. the sensor frame \mathcal{S} , of the added mass. In the above equation, M_b is unknown but M_a^j is assumed to be known.

In light of the above, we assume to be given with several data sets

$$R_j := (r_1^j, \dots, r_{N_j}^j) \in \mathbb{R}^{6 \times N_j}, \quad (5.11a)$$

$$G_j := (g_1^j, \dots, g_{N_j}^j) \in \mathbb{R}^{3 \times N_j}, \quad (5.11b)$$

associated with N_D different (m_a^j, c_a^j) . Given (5.11) and (5.12), the measurements associated with the j th data set can be compactly written as

$$CR_j - M_b G_j = M_a^j G_j.$$

The matrices C and M_b are unknown. Then, in view of (2.9) the above equation can be written for all data sets as follows

$$\Theta x = \beta, \quad (5.11c)$$

$$x := \begin{pmatrix} \text{vec}(C) \\ m \\ mc \end{pmatrix}, \quad \in \mathbb{R}^{40 \times 1}, \quad (5.11d)$$

$$\Theta := \begin{pmatrix} R_1^\top \otimes 1_6, -(G_1^\top \otimes 1_6)H \\ \vdots \\ R_{N_D}^\top \otimes 1_6, -(G_{N_D}^\top \otimes 1_6)H \end{pmatrix}, \quad \in \mathbb{R}^{6N_T \times 40}, \quad (5.11e)$$

$$\beta := \begin{pmatrix} \text{vec}(M_a^1 G_1) \\ \vdots \\ \text{vec}(M_a^{N_D} G_1) \end{pmatrix}, \quad \in \mathbb{R}^{40 \times 1}. \quad (5.11f)$$

with

$$N_T = \sum_{j=1}^{N_D} N_j,$$

i.e. the number of all measurements, and the matrix $H \in \mathbb{R}^{18 \times 4}$ a properly chosen permutator such that

$$\text{vec}(M_b) = H \begin{pmatrix} m \\ mc \end{pmatrix}.$$

To find the calibration matrix C , we have to find the solution x to the equation (5.11c). The uniqueness of this solution is characterized by the following lemma.

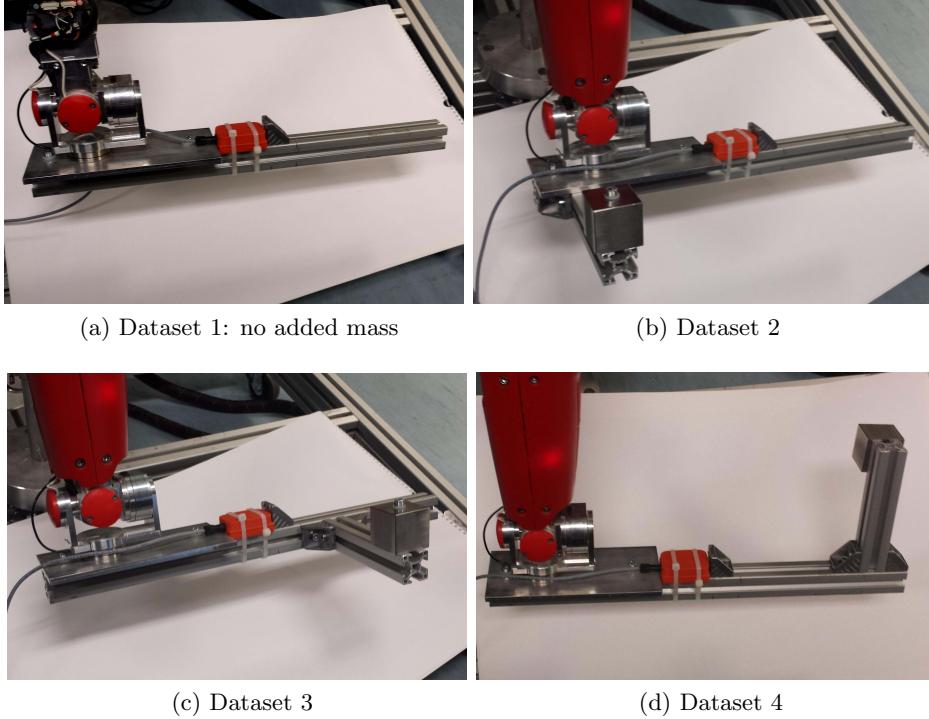


Fig. 5.3 Added mass configurations for calibration datasets.

Lemma 5.2. *A necessary condition for the uniqueness of the solution x to the equation (5.11c) is that the number of data sets must be greater than two, i.e.*

$$N_D \geq 3. \quad (5.12)$$

Proof. This is a proof by contradiction. Assume $N_D = 2$. In addition, assume, without loss of generality, also that the matrix M_j in equation (5.11) is perfectly known (adding unknowns to the considered problem would only require a larger number of data sets). Then, in view of (5.11) and (5.12), one has

$$C(R_1, R_2) = (M_1 G_1, M_2, G_2).$$

The matrix C is the unknown of the above equation. By applying (2.9) one easily finds out that there exists a unique C only if the rank of the matrix

(R_1, R_2) is equal to six, i.e.

$$\text{rank}((R_1, R_2)) = 6.$$

Recall that the matrix C is invertible by assumption, and thus with rank equal to six. Consequently

$$\begin{aligned}\text{rank}((R_1, R_2)) &= \text{rank}((M_1 G_1, M_2, G_2)) \\ &= \text{rank}\left((M_1, M_2) \begin{pmatrix} G_1 & 0 \\ 0 & G_2 \end{pmatrix}\right) \\ &\leq \min(\text{rank}(M_1, M_2), 6).\end{aligned}$$

In view of (5.3), one easily verifies that $\det(M_1, M_2) \equiv 0$, which implies that

$$\text{rank}((R_1, R_2)) \leq 5.$$

□

Establishing a sufficient condition for the uniqueness of the solution x to the equation (5.11c) is not as straightforward as proving the above necessary condition, and is beyond the scope of this thesis. Clearly, this uniqueness is related to the rank of the matrix Θ , and this rank condition can be verified numerically on real data. Then, the solution x can be found by applying least-square techniques, thus yielding estimates of the calibration matrix C and of the inertial characteristics of the rigid body.

5.4 Experimental Results

To test the proposed method, we calibrated the two force-torque sensors embedded in the leg of the iCub humanoid robot – see Figure 5.1. The mass and the center of mass of this leg are unknown.

To apply the method described in section 5.3, we need to add sample masses to the iCub’s leg. For this purpose, we installed on the robot’s foot a beam to which samples masses can be easily attached. This beam also houses a XSens MTx IMU. The supplementary accelerometer will no longer be required when using the iCub version 2, which will be equipped with 50 accelerometers distributed on the whole body. The iCub’s knee is kept fixed with a position controller, so we can consider the robot’s leg as a unique rigid body. Consequently, the accelerometer measures the gravity

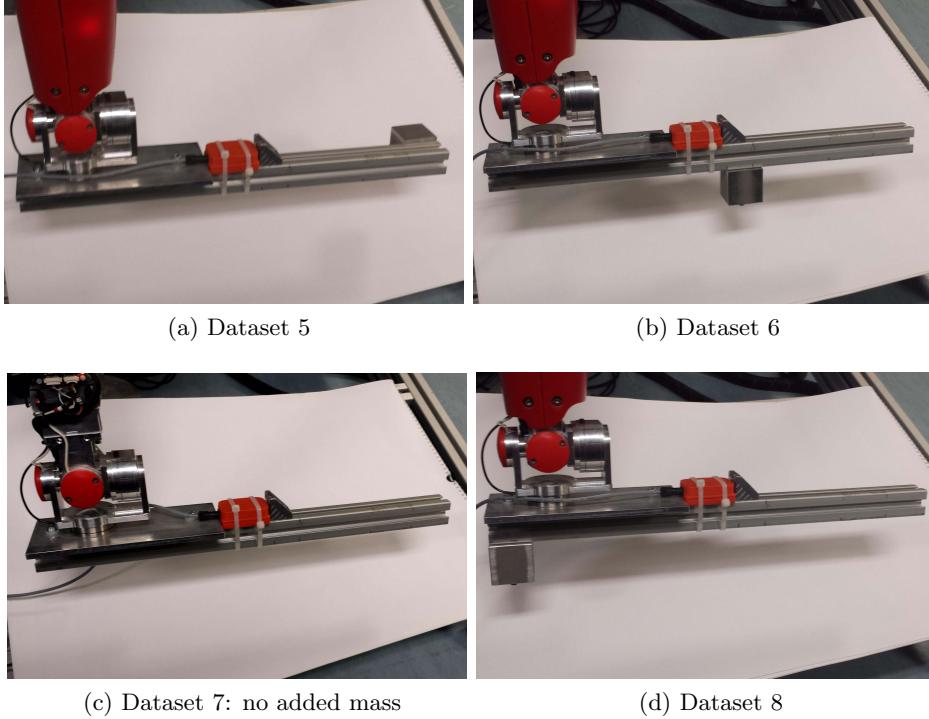


Fig. 5.4 Added mass configurations for validation datasets.

force w.r.t. both sensors. Recall also that to apply the proposed methods, we need to modify the orientations of the sensors' frames. To do so, we modified the *front-back* and *lateral* angles associated with the robot's hip.

We collected data associated with eight different added mass configurations, each of which is characterized by a mass placed at a different location with respect to the beam. In other words, we collected eight different data sets. Figures 5.3 and 5.4 show the configurations of these data sets.

For each of these data sets, we slowly¹ moved the *front-back* and *lateral* angles of the robot hip, spanning a range of 70 deg for the *front-back* angle, and a range of 90 deg for the lateral angle. We sampled the two F/T sensors and the accelerometer at 100 Hz, and we filtered the obtained signals with a Savitzky-Golay filter of third order with a windows size of 301 samples.

We estimated the sensors' offsets by applying the method described in section 5.3.2 on all eight data sets. Figure 5.5 verifies the statement of

¹The iCub front-back and lateral angles were moved with peak velocities of 2 deg /s.

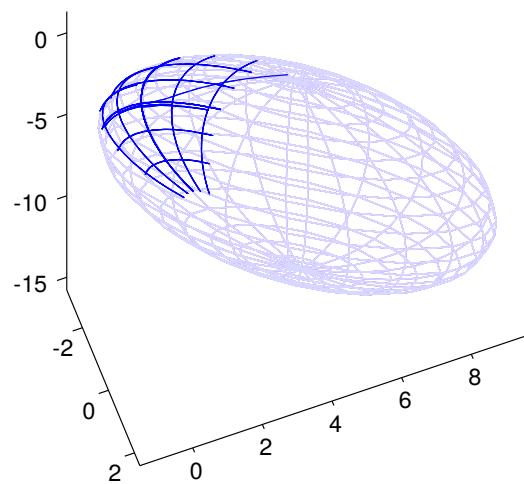


Fig. 5.5 Dark blue: raw measurements of the sensor embedded in the leg for dataset 1 projected in the 3D subspace through U_1 . In light blue an ellipsoid fitted to the measured points is added, to highlight the fact that the measured data lie on an ellipsoid. The o' point estimated with the method described in 5.3.2 is the center of this ellipsoid.

Lemma 1, i.e. the raw measurements belong to a three dimensional ellipsoid. In particular, this figure shows the measurements r_i projected onto the three dimensional space where the ellipsoid occurs, i.e. the left hand side of the equation (5.10). Then, we removed the offset from the raw measurements to apply the estimation method for the calibration matrix described in section 5.3.3.

The two sensors' calibration matrices were identified by using only four data sets (see Figure 5.3). The other four were used to validate the obtained calibration results (see Figure 5.4). The qualitative validation of the calibration procedure is based on the fact that the weight of the leg is constant for each data sets. Consequently, if we plot the force measured by the sensors, i.e. the first three rows of left hand side of the sensor's equation

$$\mathbf{f} = C(r - b),$$

these forces must belong to a sphere, since they represent the (constant norm) gravity force applied to the sensors. As for elements of comparisons, we can also plot the first three rows of the above equation when evaluated with the calibration matrix that was originally provided by the manufacturer of the sensors.

Figure 5.6 depicts the force measured by the sensor with the estimated calibration matrix (in green) and with the calibration matrix provided by the manufacturer (in red). It is clear to see that the green surfaces are much more spherical than the red ones. As a matter of fact, Table 5.1 lists the semi axes of the ellipsoids plotted in Figures 5.6, and clearly shows that the green surfaces represent spheres much better than the red ones. Interestingly enough, the force-torque sensor embedded in the iCub leg is much more mis-calibrated than that embedded in the foot. In fact, by looking at the data sheets describing the technological lives of these sensors, we found out that the force-torque sensor embedded in the leg is much older than that in the foot, which means that the calibration matrix of the leg's sensor is much older than that of the foot's sensor.

The quantitative validation of the proposed calibration procedure is performed by comparing the known weights of the added masses with the weights estimated by the sensors. Table 5.2 shows that the estimated weights obtained after performing the proposed calibration are better than those estimated by using the calibration matrix provided by the sensor manufacturer. A similar comparison was conducted on the estimation of the sample mass positions (Table 5.3). It has to be noticed that the errors in estimating the mass position are relatively high, but this is due to the choice of using

relatively small masses with respect to the sensor range and signal to noise ratio.

Table 5.1 Qualitative calibration evaluation on validation dataset: ellipse semiaxes after calibration

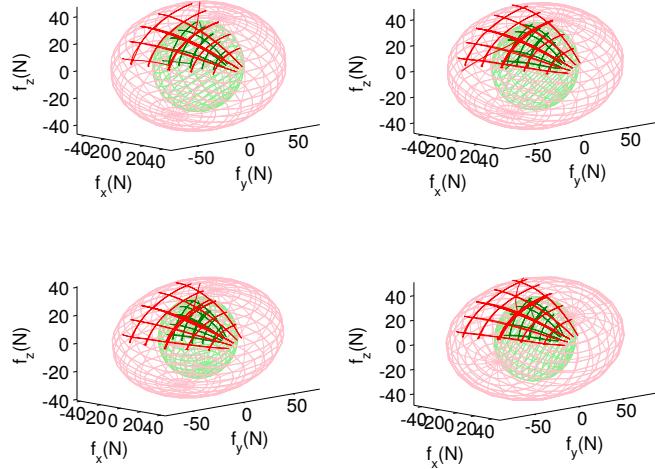
<i>Sensor</i>	<i>Dataset</i>	<i>Added mass (Kg)</i>	<i>Semiaxes length [N] with proposed calibration</i>			<i>Semiaxes length [N] with manufacturer calibration</i>		
Foot	5	0.51	13.6	13.1	12.9	13.5	10.2	4.6
Foot	6	0.51	13.5	12.9	12.7	13.6	10.5	9.4
Foot	7	0	8.4	7.9	7.4	8.5	6.9	6.2
Foot	8	0.51	13.7	12.5	12.0	15.7	13.6	10.4
Leg	5	0.51	34.4	33.3	32.5	76.5	49.4	45.6
Leg	6	0.51	34.8	33.5	32.8	82.4	49.3	47.3
Leg	7	0	30.9	28.2	26.9	77.0	44.5	40.0
Leg	8	0.51	35.52	33.30	32.2	88.9	49.5	48.3

Table 5.2 Qualitative calibration evaluation on validation dataset: sample mass estimations

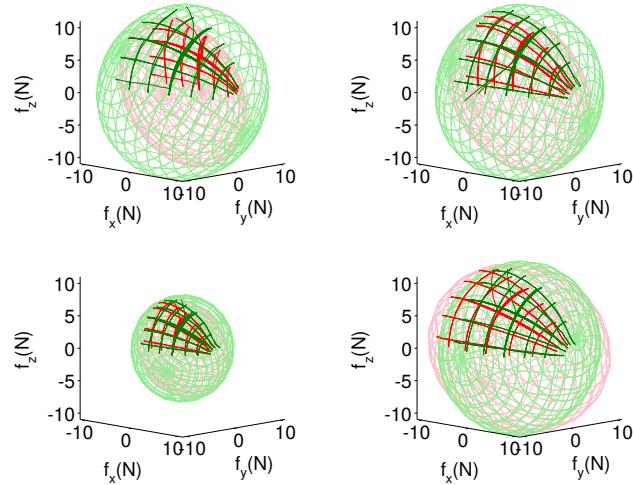
<i>Sensor</i>	<i>Dataset</i>	<i>Added mass (Kg)</i>		
		Ground truth	Proposed calibration	Manufacturer calibration
Foot	5	0.51	0.53	0.06
Foot	6	0.51	0.52	0.27
Foot	7	0	-0.03	-0.13
Foot	8	0.51	0.46	0.45
Leg	5	0.51	0.51	2.77
Leg	6	0.51	0.54	3.04
Leg	7	0	-0.04	2.39
Leg	8	0.51	0.51	3.25

Table 5.3 Qualitative calibration evaluation on validation dataset: center of mass estimations

Sensor	Dataset	Center of mass position for the added mass [cm]								
		Ground truth			Proposed calibration			Manufacturer calibration		
Foot	5	39	-3.5	2.9	31	8.8	-8.3	273	-83	81
Foot	6	21	0	6.3	19	9.9	-5	30	-18	18
Foot	7	-	-	-	-	-	-	-	-	-
Foot	8	-4	0	6.3	5.5	9	-3.2	-8.6	-12	10
Leg	5	39	-3.5	39	28	11	29	16	6.5	29
Leg	6	20	0	43	17	9.7	30	12	5.1	-27
Leg	7	-	-	-	-	-	-	-	-	-
Leg	8	-4.3	0	43	3.8	8.8	32	7.5	4.9	-26



(a) Validation results for leg F/T sensor.



(b) Validation results for foot F/T sensor.

Fig. 5.6 Dark green: force measurements obtained through the calibration matrix estimated using the proposed technique. Dark red: force measurements obtained through the calibration matrix provided with the sensor. Light red and light green surfaces: ellipsoids fitted to the measured forces.

Chapter 6

Rigid Body Inertial Parameters Identification

6.1 Introduction

A large part of existing robotic systems are modeled as a system of multiple rigid bodies. The knowledge of the dynamical characteristics of these rigid bodies is a key assumption of model-based control and estimation techniques, such as the one presented in Chapter 4. The dynamics of a rigid body, i.e. how the acceleration of a rigid body is related to the forces applied on it, is completely described by the mass distribution of the body in the 3D space. The mass distribution itself is completely described by 10 *inertial parameters* [Hollerbach et al., 2016]. These parameters may be available if a good Computer-Aided Design (CAD) model of the robot is available, but often such models are either not available, or the mass distribution of the rigid bodies in the robot changes during operation, as in the case of an end effector that grabs and heavy object.

Inverse robot dynamics models can be written linearly with respect to the inertial parameters of the rigid bodies composing the robot. Classical identification techniques [Hollerbach et al., 2016, Ayusawa et al., 2014] consider the parameters of each body to be an element of the Euclidean space \mathbb{R}^{10} . Exploiting this fact, the inertial parameters identification problem has been classically posed as a *Linear Least Square* optimization problem [Hollerbach et al., 2016]. The resulting problem is convenient from a computational point of view, but it neglects the fact that not all vectors in \mathbb{R}^{10} can be generated by a physical rigid body, i.e. it is possible that some inertial parameters are identified even if no physical rigid body could generate them.

A necessary condition for the inertial parameters to be generated by a physical rigid body was first proposed in [Yoshida et al., 1994]: the *physical consistency* condition. This condition is important for control purposes because it ensures, if it is valid for all the links of a robot, the positive definiteness and the invertibility of the joint mass matrix [Yoshida and Khalil, 2000]. This property is a key assumption in proving the stability of model-based control laws. The *physical consistency* has been enforced in identification of inertial parameters using several techniques: [Yoshida and Khalil, 2000, Mata et al., 2005, Gautier et al., 2013, Gautier and Venture, 2013, Sousa and Cortesao, 2014, Jovic et al., 2015]. However this condition is not *sufficient*: it is possible that some inertial parameters that respect this condition do not correspond to any physical body: in particular this condition does not encode the *triangle inequalities* of the 3D inertia matrix [Wittenburg, 2007, Chapter 3], as it will be explained in the remainder of the chapter.

The main contribution of this paper is a new necessary and *sufficient* condition for the inertial parameters to be generated by rigid body: the *full physical consistency* condition. We show that this condition implies the already proposed *physical consistency* condition and that the triangle inequalities are respected. Furthermore, we propose a nonlinear optimization formulation that takes into consideration this constraint by using state of the art optimization techniques on non-Euclidean manifolds [Brosssette et al., 2015]. The proposed optimization technique is validated with a rigid body inertial identification experiment on the arm of the iCub humanoid robot.

For the sake of simplicity, in this chapter, we only consider the problem of identifying the inertial parameters of a single rigid body. However, the *full physical consistency* condition and the optimization on manifolds are general contributions, that could be applied to the case of the identification of inertial parameters in generic multibody structures.

The chapter is organized as follows. Section 6.2 presents the notations used in the chapter and the background on rigid body dynamics. Section 6.3 details the proposed *full physical consistency* condition, the proposed nonlinear parametrization of the inertial parameters that ensures that this condition is always satisfied and the optimization technique on the manifold of the proposed parametrization. Section 6.4 describe the experiments used for validation.

The notation used in the chapter is summarized in the next table.

Notation used in Chapter 6	
A	Inertial frame.
B	Body frame.
$\mathbb{M} := {}_B\mathbb{M}_B$	6D inertia matrix of body B expressed in the B frame.
i	
m	Mass of the rigid body.
$c := {}^B c$	Center of mass of the rigid body, expressed in the B frame.
I_B	3D inertia of the body B , expressed with the orientation of the body frame B and w.r.t. the origin of B .
$\alpha^g := {}^B \alpha_{A,B}^g \in \mathbb{R}^3$	Angular velocity of the body expressed in body frame.
$\omega := {}^B \omega_{A,B} \in \mathbb{R}^3$	Angular velocity of the body expressed in body frame.

6.2 Background on Rigid Body Inertial Parameters

6.2.1 Rigid Body Dynamics

The Newton-Euler equations using the *proper sensor acceleration* (2.80) are given by:

$$\mathbb{M}\alpha^g + \begin{bmatrix} 0_{3 \times 1} \\ \omega \end{bmatrix} \bar{\times}^* \mathbb{M} \begin{bmatrix} 0_{3 \times 1} \\ \omega \end{bmatrix} = \phi \quad (6.1)$$

where $\alpha^g := \alpha_{A,B}^g \in \mathbb{R}^6$ is the *sensor proper acceleration*, $\omega := {}^B \omega_{A,B} \in \mathbb{R}^3$ is the angular velocity of the body expressed in body frame, $\phi := {}_B \phi_B$ is the net force-torque acting on the robot expressed in B frame and $\mathbb{M} \in \mathbb{R}^{6 \times 6}$ is the 6D inertia matrix (also known as *spatial inertia* in Featherstone [2008]) expressed in body frame B

$$\mathbb{M} = \begin{bmatrix} mI_3 & -mc^\wedge \\ mc^\wedge & I_B \end{bmatrix}. \quad (6.2)$$

Where:

- $m \in \mathbb{R}$ is the mass of the rigid body,
- $c \in \mathbb{R}^3 := {}^B c$ is the center of mass of the rigid body, expressed in the frame B ,

- $I_B \in \mathbb{R}^{3 \times 3}$ is the 3D inertia matrix of the rigid body, expressed with the orientation of frame B and with respect to the frame B origin.

6.2.2 Inertial parameters

The 6D inertia matrix is parametrized by 10 parameters, usually called the *inertial parameters* of the rigid body [Hollerbach et al., 2016], that are defined as $\pi \in \mathbb{R}^{10}$:

$$\pi = \begin{bmatrix} m \\ mc \\ \text{vech}(I_B) \end{bmatrix}. \quad (6.3)$$

The product of a generic vector $\begin{bmatrix} v \\ \omega \end{bmatrix} \in \mathbb{R}^6$ by the 6D inertia matrix \mathbb{M} can be written as a product of a matrix in $\mathbb{R}^{6 \times 10}$ for the vector of inertial parameters π :

$$\mathbb{M} \begin{bmatrix} v \\ \omega \end{bmatrix} = D \left(\begin{bmatrix} v \\ \omega \end{bmatrix} \right) \pi = \begin{bmatrix} v & \omega^\wedge & 0_{3 \times 6} \\ 0_{3 \times 1} & -v^\wedge & \omega \bullet \end{bmatrix} \pi \quad (6.4)$$

where the matrix $\omega \bullet$ is defined such that $\omega \bullet \text{vech}(I_B) = I_B \omega$:

$$\omega \bullet = \begin{bmatrix} \omega_x & \omega_y & \omega_z & 0 & 0 & 0 \\ 0 & \omega_x & 0 & \omega_y & \omega_z & 0 \\ 0 & 0 & \omega_x & 0 & \omega_y & \omega_z \end{bmatrix}. \quad (6.5)$$

Proposition 6.1 (Newton-Euler equations linearity in the inertial parameters). *The Newton-Euler equations (6.1) can be written linearly [Garofalo et al., 2013, Hollerbach et al., 2016] in the inertial parameters (6.3):*

$$Y(\alpha^g, \omega)\pi = \mathbb{M}\alpha^g + \begin{bmatrix} 0_{3 \times 1} \\ \omega \end{bmatrix} \bar{\times}^* \mathbb{M} \begin{bmatrix} 0_{3 \times 1} \\ \omega \end{bmatrix} = \phi, \quad (6.6)$$

with:

$$Y(\alpha^g, \omega) = D(\alpha^g) + \begin{bmatrix} 0_{3 \times 1} \\ \omega \end{bmatrix} \bar{\times}^* D \left(\begin{bmatrix} 0_{3 \times 1} \\ \omega \end{bmatrix} \right). \quad (6.7)$$

6.2.3 Relationship between the inertial parameters and the density function

The mass distribution of a rigid body in space is described by its density function:

$$\rho(\cdot) : \mathbb{R}^3 \mapsto \mathbb{R}_{\geq 0}. \quad (6.8)$$

The domain of this function is the points of body expressed in the body-fixed frame B . We consider the density equal to zero for the points outside the volume of the rigid body, so we can define the domain of $\rho(\cdot)$ to be all the points in the 3D space \mathbb{R}^3 .

The inertial parameters are obviously a functional of the density $\rho(\cdot)$, in particular, we can define the functional $\pi_d(\cdot) : (\mathbb{R}^3 \mapsto \mathbb{R}_{\geq 0}) \mapsto \mathbb{R}^{10}$ that maps the density function to the corresponding inertial parameters:

$$\begin{aligned}\pi_d(\rho(\cdot)) &= \begin{bmatrix} m(\rho(\cdot)) \\ mc(\rho(\cdot)) \\ \text{vech}(I_B(\rho(\cdot))) \end{bmatrix} = \\ &= \begin{bmatrix} \iiint_{\mathbb{R}^3} \rho(r) dr \\ \iiint_{\mathbb{R}^3} r \rho(r) dr \\ \text{vech} \left(\iiint_{\mathbb{R}^3} (r^\wedge)^T r^\wedge \rho(r) dr \right) \end{bmatrix}. \quad (6.9)\end{aligned}$$

6.2.4 3D Inertia at the Center of Mass and Principal Axes

The 3D inertia at the center of mass is defined as

$$I_C = \iiint_{\mathbb{R}^3} ((r - c)^\wedge)^T (r - c)^\wedge \rho(r) dr. \quad (6.10)$$

Exploiting the fact that $(\cdot)^\wedge$ is linear, the inertia matrix with respect to the center of mass can be written as:

$$I_C = I_B + mS(c)S(c). \quad (6.11)$$

This result is known as *parallel axis theorem*.

As I_C is symmetric, it can be diagonalized with an orthogonal matrix $Q \in SO(3)$:

$$I_C = Q \text{diag}(J) Q^T. \quad (6.12)$$

Using (6.10) the diagonal matrix $\text{diag}(J)$ (with $J \in \mathbb{R}^3$) can be written as:

$$\begin{aligned}\text{diag}(J) &= \iiint_{\mathbb{R}^3} Q^T ((r - c)^\wedge)^T (r - c)^\wedge Q \rho(r) dr = \\ &= \iiint_{\mathbb{R}^3} ((Q^T(r - c))^\wedge)^T (Q^T(r - c))^\wedge \rho(r) dr. \quad (6.13)\end{aligned}$$

The operation mapping the body point r to $Q^T(r - c)$ can be interpreted as a change of reference frame, from the body frame B to a frame C (a *principal axes* frame) whose origin is the center of mass of the body and whose orientation is one in which the I_C matrix is diagonal.

By expressing with $\tilde{r} = Q^T(r - c)$ the generic point of the body expressed in the C frame, and with $\tilde{\rho}(\tilde{r})$ the density with respect to the C frame, we can write $\text{diag } J$ as:

$$\text{diag } J = \iiint_{\mathbb{R}^3} (\tilde{r}^\wedge)^T \tilde{r}^\wedge \tilde{\rho}(\tilde{r}) d\tilde{r}. \quad (6.14)$$

The elements of the J vector are:

$$J_x = \iiint_{\mathbb{R}^3} (\tilde{y}^2 + \tilde{z}^2) \tilde{\rho}(\tilde{r}) d\tilde{r}, \quad (6.14a)$$

$$J_y = \iiint_{\mathbb{R}^3} (\tilde{x}^2 + \tilde{z}^2) \tilde{\rho}(\tilde{r}) d\tilde{r}, \quad (6.14b)$$

$$J_z = \iiint_{\mathbb{R}^3} (\tilde{x}^2 + \tilde{y}^2) \tilde{\rho}(\tilde{r}) d\tilde{r}. \quad (6.14c)$$

We can write them as:

$$J_x = L_y + L_z, \quad J_y = L_x + L_z, \quad J_z = L_x + L_y. \quad (6.15)$$

$$L_x = \iiint_{\mathbb{R}^3} \tilde{x}^2 \tilde{\rho}(\tilde{r}) d\tilde{r}, \quad L_y = \iiint_{\mathbb{R}^3} \tilde{y}^2 \tilde{\rho}(\tilde{r}) d\tilde{r}, \quad (6.16)$$

$$L_z = \iiint_{\mathbb{R}^3} \tilde{z}^2 \tilde{\rho}(\tilde{r}) d\tilde{r}. \quad (6.17)$$

Where L_x, L_y, L_z are the *central second moments of mass* of the density $\tilde{\rho}(\tilde{r})$. It is clear that the non-negativity of $\tilde{\rho}(\tilde{r})$ constraints $L = [L_x \ L_y \ L_z]^T$ to be non-negative as well. Furthermore, it is possible to see that the non-negativity of L induces the *triangular inequalities* on $\text{diag}(J)$:

$$J_x \leq J_y + J_z, \quad J_y \leq J_x + J_z, \quad J_z \leq J_x + J_y. \quad (6.18)$$

6.2.5 Inertial Parameters Identification

Assuming that N values for F, A_g and V are measured, equation (6.6) can be used to estimate π solving the following optimization problem:

$$\hat{\pi} = \arg \min_{\pi \in \mathbb{R}^{10}} \sum_{i=1}^N \|Y(\alpha_i^g, \omega_i)\pi - f_i\|^2. \quad (6.19)$$

However, this optimization does not take into account the physical properties of the inertial parameters π . For this reason, the following definition was introduced.

Definition 6.1. *A vector of inertial parameters π is called physical consistent [Yoshida et al., 1994, Yoshida and Khalil, 2000] if:*

$$m(\pi) \geq 0, \quad I_C(\pi) \succeq 0. \quad (6.20)$$

This condition has nice properties (it ensures that the matrix M is always invertible), but is still possible to find some *physical consistent* inertial parameters that can't be generated by a physical density.

6.3 Full Physical Consistency

6.3.1 Full physical consistency

In this subsection, we propose a new condition for assessing if a vector of inertial parameters can be generated from a physical rigid body. We will show that all the constraints that emerge due to this *full physical consistency* condition are due to the non-negativity on the density function.

Definition 6.2. *A vector of inertial parameters $\pi^* \in \mathbb{R}^{10}$ is called fully physical consistent if:*

$$\exists \rho(\cdot) : \mathbb{R}^3 \mapsto \mathbb{R}_{\geq 0} \text{ s.t. } \pi^* = \pi_d(\rho(\cdot)). \quad (6.21)$$

This definition extends the concept of *physical consistent* inertial parameters to include also all possible constraints of inertial parameters, such as the triangular inequalities (6.18) of the diagonal elements of the inertia matrix.

Lemma 6.1. *If a vector of inertial parameters $\pi \in \mathbb{R}^{10}$ is fully physical consistent it follows that it is physical consistent, according to Definition 6.1.*

Proof. If π is *fully physical consistent*, then it follows that there exists $\rho(\cdot)$ such that the corresponding 3D inertia at the center of mass I_C can be written as a function of $\rho(\cdot)$. The positive semi-definiteness of m and I_C then follows from the classical properties of mass and the inertia matrix of a rigid body, see for example subsection 3.3.3 of Wittenburg [2007]. \square

Lemma 6.2. *If a vector of inertial parameters $\pi \in \mathbb{R}^{10}$ is fully physical consistent, the associated inertia matrices at the body origin $I_B(\pi)$ and at the center of mass $I_C(\pi)$ respect the triangular inequalities (6.18).*

Proof. This lemma can be proved by writing I_B or I_C as a functional of the density function $\rho(\cdot)$, as in the proof of Lemma 6.1. Once I_B or I_C are written as a functional of $\rho(\cdot)$, the demonstration that they respect the triangle inequality can be found in any rigid body mechanics textbook, see for example subsection 3.3.4 of Wittenburg [2007]. \square

To get a hint of the demonstration of Lemma 2, consider that the diagonal elements of the 3D inertia matrix with respect to an arbitrary frame can still be written as the sum of two non-negative *second moments of mass*. The triangle inequality then arises in a way similar to the case of the inertia expressed in the principal axes.

6.3.2 Full physical consistent parametrization of inertia parameters

In this subsection, we introduce a novel nonlinear parametrization of inertial parameters that ensures the *full physical consistency* condition.

We choose to parametrize the mass as an element of the spaces of non-negative numbers $m \in \mathbb{R}_{\geq 0}$.

The center of mass do not have any constraints on its location, so we choose to parametrize it as an element of the 3D space $c \in \mathbb{R}^3$.

For parametrize the 3D inertia matrix ensuring the properties described in subsection 6.2.4 we choose the *second moments of mass* $L \in \mathbb{R}_{\geq 0}^3$ to be components of our parametrization. In the following, we will show how this choice ensures the *full physical consistency* of the resulting inertial parameters.

The inertial parameters of a rigid body can then be parametrized by an element $\theta \in \mathfrak{P} = \mathbb{R}_{\geq 0} \times \mathbb{R}^3 \times SO(3) \times \mathbb{R}_{\geq 0}^3$. In particular the components of θ are:

- $m \in \mathbb{R}_{\geq 0}$ the mass of the body

- $c \in \mathbb{R}^3$ the center of mass of the body
- $Q \in SO(3)$ the rotation matrix between the body frame and the frame of principal axis at the center of mass
- $L \in \mathbb{R}_{\geq 0}^3$ the second central moment of mass along the principal axes

In other terms, there is a function $\pi_p(\theta) : \mathfrak{P} \mapsto \mathbb{R}^{10}$ that maps this new parametrization to the corresponding inertial parameters:

$$\pi_p(\theta) = \begin{bmatrix} m(\theta) \\ mc(\theta) \\ \text{vech}(I_B(\theta)) \end{bmatrix} = \begin{bmatrix} m \\ mc \\ \text{vech}(Q \text{diag}(PL)Q^T - mS(c)S(c)) \end{bmatrix}$$

Where $P = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$ is a matrix that maps L to J .

Theorem 6.1. *For each $\theta \in \mathfrak{P}$, there exists a density function $\rho(\cdot) : R^3 \mapsto R_{\geq 0}$ such that $\pi_d(\rho(\cdot)) = \pi_p(\theta)$, i.e. every $\theta \in \mathfrak{P}$ generates fully physical consistent inertial parameters.*

Proof. We prove the statement in a constructive way: given an arbitrary element $\theta = (m, c, Q, L) \in \mathfrak{P}$ we build a density function $\rho(\cdot) : \mathbb{R}^3 \mapsto \mathbb{R}_{\geq 0}$ such that $\pi_p(\theta) = \pi_d(\rho(\cdot))$. For example we can think of a cuboid of uniform unit density, with the center of the cuboid coincident with the center of mass of the inertial parameters (given by c), with the orientation of its symmetry axis aligned with the C principal axes frame defined by the Q rotation matrix and the cuboid sides of lengths $2d_x$, $2d_y$ and $2d_z$, with:

$$d = [d_x \ d_y \ d_z]^\top = \left[\sqrt{3 \frac{L_x}{m}} \ \sqrt{3 \frac{L_y}{m}} \ \sqrt{3 \frac{L_z}{m}} \right]^\top \quad (6.22)$$

Its density function in the C frame is given as:

$$\tilde{\rho}(\tilde{r}) = \begin{cases} 1 & \text{if } -d \geq \tilde{r} \geq d \\ 0 & \text{otherwise} \end{cases}$$

while the density function in the B frame is given by:

$$\rho(r) = \begin{cases} 1 & \text{if } -Qd + c \geq r \geq Qd + c \\ 0 & \text{otherwise} \end{cases}. \quad (6.23)$$

The density defined in (6.23) and (6.22) can be seen as a function $\gamma(\cdot) : \mathfrak{P} \mapsto (\mathbb{R}^3 \mapsto \mathbb{R}_{\geq 0})$. The theorem is then demonstrated by using (6.9) and (6.22) to verify that:

$$\pi_d(\gamma(\theta)) = \pi_p(\theta)$$

is true $\forall \theta \in \mathfrak{P}$. \square

Using the parametrization presented in Theorem 6.3.2, it is possible to recast the identification optimization problem (6.19) as:

$$\hat{\pi} = \pi(\hat{\theta}) \quad (6.24)$$

$$\hat{\theta} = \arg \min_{\theta \in \mathfrak{P}} \sum_{i=1}^N \|Y(\alpha_i^g, \omega_i)\pi(\theta) - f_i\|^2 \quad (6.25)$$

The main advantage of (6.24) with respect to (6.19) is that thanks to Theorem 6.1 the identified inertial parameters $\hat{\pi}$ are ensured to be *fully physically consistent*. However, the optimization variable θ does not live anymore in a Euclidean space, because \mathfrak{P} includes $SO(3)$, so to solve this optimization problem in this need to either modify or to exploit specific techniques related to the *optimization on manifolds*, as we did in the experiments provided in Section

6.4 Experimental Results

6.4.1 Optimization on Manifolds

For this chapter, we focus on $SO(3)$, a 3-dimensional manifold. As such, it can be parametrized *locally* by 3 variables, for example, a choice of Euler angles, but any such parametrization necessarily exhibits singularities when taken as a global map (e.g. gimbal lock for Euler angles), which can be detrimental to our optimization process.

For this reason, when addressing $SO(3)$ with classical optimization algorithms, it is often preferred to use one of the two following parametrizations:

- unit quaternion, *i.e.* an element q of \mathbb{R}^4 with the additional constraint $\|q\| = 1$,
- rotation matrix, *i.e.* an element R of $\mathbb{R}^{3 \times 3}$ with the additional constraints $R^T R = I$ and $\det R \geq 0$.

The alternative is to use optimization software working natively with manifolds [Brossette et al., 2015][Absil et al., 2008] and solve

$$\arg \min_{\theta \in \mathbb{R} \times \mathbb{R}^3 \times SO(3) \times \mathbb{R}^3} \sum_{i=1}^N \|Y(a_i^g, v_i)\pi(\theta) - f_i\|^2 \quad (6.26)$$

$$\text{subj. to } m \geq 0, L_x \geq 0, L_y \geq 0, L_z \geq 0 \quad (6.27)$$

This alternative has an immediate advantage: we can write directly the problem (6.24) without the need to add any parametrization-related constraints. Because there are fewer variables and fewer constraints, it is also faster to solve. To check this, we compared the resolution of (6.24) formulated with each of the three parametrizations (native $SO(3)$, unit quaternion, rotation matrix). We solved the three formulations with the solver presented in [Brossette et al., 2015], and the two last with an off-the-shelf solver (CFSQP [Lawrence et al., 1997]), using the dataset presented later in this section. The formulation with native $SO(3)$ was consistently solved faster. We observed timings around 0.5s for it, and over 1s for non-manifold formulations with CFSQP. The mean time for an iteration was also the lowest with the native formulation (at least 30% when compared to all other possibilities).

Working directly with manifolds has also an advantage that we do not leverage here, but could be useful for future work: at each iteration, the variables of the problem represent a fully physical consistent set of inertial parameters. This is not the case with the other formulations we discussed, as the (additional) constraints are guaranteed to be satisfied only at the end of the optimization process. Having physically meaningful intermediate values can be useful to evaluate additional functions that presuppose it (additional constraints, external monitoring ...). It can also be leveraged for real-time applications where only a short time is allocated repeatedly to the inertial identification, so that when the optimization process is stopped after a few iterations, the output is physically valid. With non-manifold formulations, at any given iteration, the parametrization-related constraints can be violated, thus, the variables might not lie in the manifold. It is then needed to project them on it. Denoting π the projection (for example $\pi = \frac{q}{\|q\|}$ in the unit quaternion formulation), to evaluate a function f on a manifold, we need to compute $f \circ \pi$. If further the gradient is needed, that projection must also be accounted for ([Bouyarmane and Kheddar, 2012] explains this issue in great details for free-floating robots).

In this study, we use the same solver and approach as presented in [Brossette et al., 2015] which was inspired from [Absil et al., 2008]. The driving

idea of the optimization on manifold is to change the parametrization at each iteration. The problem at iteration k becomes:

$$\min_{z_k \in \mathbb{R}^n} f \circ \varphi_{x_k}(z) \quad \text{s.t.} \quad c \circ \varphi_{x_k}(z) = 0. \quad (6.28)$$

Then $x_{k+1} = \varphi_{x_k}(z_k)$ is guaranteed to belong to \mathcal{M} . The next iteration uses the same formulation around x_{k+1} .

The smooth maps φ_x are built-in and are used automatically by the solver while the user only has to implement the functions of the optimization problem without the burden of worrying about the parametrization.

6.4.2 Experiments

The iCub is a full-body humanoid with 53 degrees of freedom, thoroughly described in Section 1.1. For validating the presented approach, we used the six-axis force/torque (F/T) sensor embedded in iCub's right arm to collect experimental F/T measurements. We locked the elbow, wrist and hands joints of the arm, simulating the presence of a rigid body directly attached to the F/T sensor, a scenario similar to the one in which an unknown payload needs to be identified [Kubus et al., 2008].

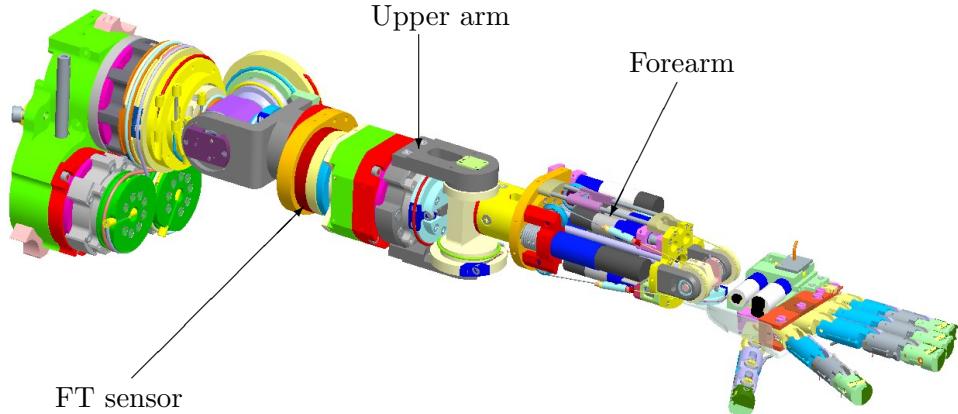


Fig. 6.1 CAD drawing of the iCub arm used in the experiments. The used six-axis F/T sensor is visible in the middle of the upper arm link.

We generated five 60 seconds joint positions paths in which the three shoulder joints were reaching random joint position using point to point minimum-jerk like trajectories. The point to point trajectory completion times were 10s, 5s, 2s, 1s and 0.5s for the different paths. We played these joint paths on the robot and we sampled at 100Hz the F/T sensors and joint encoders output. We filtered the joint positions and obtained joint velocities and accelerations using a Savitzky-Golay filtering of order 2 and with a windows size of 499, 41, 21, 9, 7 samples. We used joint positions, velocities and accelerations with the kinematic model of the robot to compute a^g and v of the F/T sensor for each time sample. We removed the unknown offset from the F/T measurements using the offset removal technique described in [Traversaro et al., 2015]. We then solved the inertial identification problem using the classical linear algorithm (6.19) and the one using the proposed *fully physical consistent* parametrization (6.26). We report the identified inertial parameters in Table 6.1. It is interesting to highlight that for slow datasets (trajectory time of 10s or 5s) the unconstrained optimization problem (6.19) results in inertial parameters that are not fully physical consistency. In particular, this is due to the low values of angular velocities and acceleration, that do not properly excites the inertial parameters, which are then *numerically not identifiable*. The proposed optimization problem clearly cannot identify these parameters anyway, as the identified parameters are an order of magnitude larger than the ones estimated for faster datasets, nevertheless, it always estimates inertial parameters that are fully physical consistent. For faster datasets (trajectory time of 1s or 0.5s) the results of the two optimization problems are the same because the high values of angular velocities and accelerations permit to identify all the parameters perfectly. While this is possible to identify all the inertial parameters of a single rigid body, this is not the case when identifying the inertial parameters of a complex structure such as a humanoid robot, for which both structural [Ayusawa et al., 2014] and numerical [Pham and Gautier, 1991] not identifiable parameters exists. In this later application, the enforcement of full physical consistency will always be necessary to get meaningful results.

Table 6.1 Inertial parameters identified with the different datasets and the different optimization problems.

	10 s		5 s		2 s		1 s		0.5 s	
	\mathbb{R}^{10}	\mathfrak{P}								
m	1.836	1.836	1.842	1.842	1.852	1.852	1.820	1.820	1.843	1.844
mc_x	0.062	0.062	0.061	0.060	0.060	0.060	0.060	0.060	0.060	0.059
mc_y	0.001	0.001	0.000	0.000	0.001	0.001	0.002	0.002	0.005	0.004
mc_z	0.208	0.208	0.206	0.206	0.206	0.206	0.205	0.205	0.204	0.204
I_{xx}	0.580	0.215	0.128	0.166	0.065	0.067	0.032	0.034	0.033	0.037
I_{xy}	0.593	0.012	-0.02	0.001	0.001	0.001	0.001	0.001	0.003	0.001
I_{xz}	-0.54	-0.06	-0.13	-0.09	-0.04	-0.03	-0.02	-0.02	-0.02	-0.02
I_{yy}	1.022	0.227	0.125	0.216	0.066	0.086	0.036	0.042	0.035	0.039
I_{yz}	0.190	0.038	0.026	0.001	0.006	0.003	0.002	0.001	0.000	0.000
I_{zz}	-0.13	0.028	-0.00	0.050	0.007	0.014	0.008	0.009	0.008	0.008

Inertial parameters identified on \mathbb{R}^{10} optimization manifold that are not fully physical consistent are highlighted.

Masses are expressed in kg, first moment of masses in kg m, inertia matrix elements in kg m².

Chapter 7

Multi Body Inertial Parameters Identification

7.1 Introduction

A fundamental problem in controlling torque-actuated robots is the accurate modeling of their dynamics. Depending on the performed task (e.g. control, simulation, contact detection) we can distinguish two possible approaches [Hollerbach et al., 2016]: in *structural modeling* the interest is on identifying the real inertial parameters of the robot, while in *predictive modeling* the interest is only on replicating the input-output behavior of the system, the input and output being some measured quantities. In structural modeling the usual approach is to excite the robot with trajectories chosen so as to be the optimal for identifying the identifiable (i.e. base) parameters [Armstrong, 1988]. In predictive modeling, the only concern is to accurately model the input-output response of the dynamical system. This has a significant implication: the interest is not in estimating the “real” parameters, but in getting parameters capable of generalizing predictions across the whole work space. Interestingly, within this context different regression techniques can be adopted, ranging from parametric [Hollerbach et al., 2016], semi-parametric [Duy Nguyen-Tuong and Peters, 2010] and machine learning approaches [Fumagalli et al., 2010a]. A common task that falls within the predictive modeling category is learning inverse dynamics: inputs are positions, velocities and accelerations while outputs are joint torques.

In this chapter we tackle a problem that lies in between *structural* and *predictive modeling*. We aim to relax the modeling assumptions in the procedure illustrated in Chapter 4 to estimate joint torques from embedded

6-axis force/torque sensors. This estimation procedure allows us to implement torque control on robots without joint torque sensing. Since most (humanoid) robots are not equipped with joint torque sensors, but have 6-axis F/T sensors, this approach opens the possibility to implement inverse-dynamics control on these “old-generation” robots. Moreover, this is interesting also for new-generation robots, which could be easily equipped with 6-axis F/T sensors, without going through the hassle of redesigning the joints to include torque sensing.

The main drawback of this method is that it relies on the inertial parameters to estimate the joint torques. The goal of this paper is to understand if and to what extent this knowledge is necessary and if we can partially retrieve it through identification procedures similar to the one proposed in [Hollerbach et al., 2016]. The major technical obstacle lies in the following consideration. We can use F/T measurements to estimate certain inertial parameters (known in literature as *base parameters*): what is the relationship between these parameters and the ones used in [Fumagalli et al., 2012] to estimate (internal) joint torques and (external) contact forces? In this framework non-parametric techniques have limited appeal and therefore we pursue a parametric approach.

Consistently with the rest of the thesis, in this chapter we will discuss the problem of estimating inertial parameters in the context of free floating robots. In particular we will discuss the case of iCub and its specific set of sensors, as introduced in Section 1.1.

In this chapter, we will use several concepts related to the change of base link that we introduced in Chapter 3. The notation used in the chapter is summarized in the next table.

Notation used in Chapter 7

$\text{LinkIndex}(\cdot)$: Link serialization function.

$$\mathcal{L} \mapsto \{1, \dots, n_L\}$$



$$\pi_L \in \mathbb{R}^{10}$$

$$\psi \in \mathbb{R}^{10n_L}$$

$$\alpha_B^g := \alpha_{A,B}^g$$

$$\omega_B := {}^B\omega_{A,B}$$

$$s$$

$$\dot{s}$$

$$\ddot{s}$$

Vector of inertial parameters of link L .

Vector of all the inertial parameters of a multibody system.

Sensor proper acceleration of body B .

Angular velocity of the link B in the B frame.

Internal joint (shape) positions.

Internal joint (shape) velocities.

Internal joint (shape) accelerations.

7.2 Identification of floating base dynamics

Similarly to the rigid body case presented in Chapter 6, the inertial parameters of the a multibody system can be reprented by an inertial parameters vector, defined as in the following.

Definition 7.1 (Multibody Inertial Parameters Vector). *Given a multibody system, its inertial parameters vector $\psi \in \mathbb{R}^{10n_L}$ is defined as:*

$$\psi = \begin{bmatrix} \pi_{LinkIndex(1)} \\ \pi_{LinkIndex(2)} \\ \vdots \\ \pi_{LinkIndex(n_L)} \end{bmatrix} \in \mathbb{R}^{10n_L},$$

where $\pi_L \in \mathbb{R}^{10}$ is the vector of the inertial parameters of link L , defined in (6.3).

Several dynamics-related quantities, starting from the lagrangian itself, can be written linearly w.r.t. to this vector. While in the past methods to use *energy-based* [Gautier and Khalil, Gautier] or *center of pressure-based* [Baelemans et al., 2013, Baelemans, 2013] have been proposed, the most used method in the literature of the identification of humanoids inertial parameters is the floating base dynamics regressor [Jovic et al., 2015, Ayusawa et al., 2014, Ogawa et al., 2014, Mistry et al., 2009], introduced in the next proposition.

Proposition 7.1 (Floating Base Regressor [Ayusawa et al., 2014]). *The right-hand side of (3.64) can be rearranged linearly with respect to a vector of inertial parameters ψ , i.e. :*

$$\Gamma(\alpha_B^g, \omega_B, s, \dot{s}, \ddot{s}) = \begin{bmatrix} {}_B Y_b \\ {}_B Y_s \end{bmatrix} \psi = \begin{bmatrix} 0_{6 \times 1} \\ \tau \end{bmatrix} + \sum_{L \in \mathfrak{L}} J_L^\top f_l^x. \quad (7.1)$$

The typical assumption in literature is that both joint torques and contact force measurements are available [Mistry et al., 2009, Ogawa et al., 2014], and in that case all the lines of (7.1) can be used for identification. The alternative assumption is that only contact forces measurements are available [Ayusawa et al., 2014, Jovic et al., 2015] and in that case only the first 6 rows of (7.1) are used for estimation, disregarding the *shape dynamics*. However both this hypothesis are not matched by the assumptions of this thesis explained in Chapter 4: in our case neither joint torques nor contact

forces are available as measurements, and the estimates presented in Chapter 4 depend them-self on the inertial parameters of the assumed model, so they cannot be used for inertial parameters estimation. Assuming that the considered models is equipped with internal six-axis force-torque sensor, the solution is again to consider the submodels introduced in Subsection 4.5.1 separately. In particular, also the left-hand term of (4.19) can be rewritten in function of ψ :

$$Y_{\phi_{sm}} \psi = \sum_{L \in \mathfrak{L}_{sm}} {}_B X^L L \phi_L = \sum_{L \in (\mathfrak{C} \cap \mathfrak{L}_{sm})} {}_B X^L L f_L^x + \sum_{L \in \mathfrak{L}_{sm}} \sum_{D \in \mathfrak{D}_{sm}(L)} {}_B X^D D f_{D,L} \quad (7.2)$$

In general there are two classes of unknowns in this equation: the inertial parameters ψ and the external force-torque $L f_L^x$. If at a given instant we know that no external force-torques (from a-priori information or from the distributed tactile system) are acting on the submodel sm , the equation can be rewritten as:

$$Y_{\phi_{sm}} \psi = \sum_{L \in \mathfrak{L}_{sm}} {}_B X^L L \phi_L = \sum_{L \in \mathfrak{L}_{sm}} \sum_{D \in \mathfrak{D}_{sm}(L)} {}_B X^D D f_{D,L} \quad (7.3)$$

and this equation can be used directly for the identification of the inertial parameters ψ .

The regressor obtained by combining the base regressors of all submodels $\mathfrak{M} = 1, 2, \dots, n+1$ is defined as:

$$Y_{\phi_{\mathfrak{M}}} \psi = \begin{bmatrix} Y_{\phi_1} \\ Y_{\phi_2} \\ \vdots \\ Y_{\phi_{n+1}} \end{bmatrix} \psi = \begin{bmatrix} \sum_{L \in (\mathfrak{C} \cap \mathfrak{L}_1)} {}_B X^L L f_L^x \\ \sum_{L \in (\mathfrak{C} \cap \mathfrak{L}_2)} {}_B X^L L f_L^x \\ \vdots \\ \sum_{L \in (\mathfrak{C} \cap \mathfrak{L}_{n+1})} {}_B X^L L f_L^x \end{bmatrix} + \begin{bmatrix} \sum_{L \in \mathfrak{L}_1} \sum_{D \in \mathfrak{D}_1(L)} {}_B X^D D f_{D,L} \\ \sum_{L \in \mathfrak{L}_2} \sum_{D \in \mathfrak{D}_2(L)} {}_B X^D D f_{D,L} \\ \vdots \\ \sum_{L \in \mathfrak{L}_{n+1}} \sum_{D \in \mathfrak{D}_{n+1}(L)} {}_B X^D D f_{D,L} \end{bmatrix} \quad (7.4)$$

The actual part of $Y_{\phi_{\mathfrak{M}}}$ that can be used for estimation depends on the instantaneous set of links that are in contact with the environment \mathfrak{C} . However, assuming that for each submodel there exists data samples in which they are not in contact with the environment all parts of $Y_{\phi_{\mathfrak{M}}}$ can be used for identification, sooner or later.

The regressor for joint torques can be written easily due to Theorem 4.2. In particular (4.25) can be written as:

$$\tau_{E,F} = \left\langle {}^F s_{E,F}, \sum \phi_L + {}^F f_{G,H} \right\rangle = Y_{\hat{\tau}_{E,F}} \psi + \left\langle {}^F s_{E,F}, {}^F f_{G,H} \right\rangle. \quad (7.5)$$

Identifiable subspaces

The different parametric representation of the robot dynamics such as (7.1) or the submodel representation (7.2) can be always rearranged as $Y\phi = m$, where both Y and m can be computed given the available measurements. An estimation of ϕ can be obtained by considering repeated measurements Y^1, \dots, Y^N and the associated values of the regression matrix Y^1, \dots, Y^N , related as follows:

$$\begin{bmatrix} Y^1 \\ Y^2 \\ \vdots \\ Y^N \end{bmatrix} \phi = \begin{bmatrix} m^1 \\ m^2 \\ \vdots \\ m^N \end{bmatrix}. \quad (7.6)$$

The matrix that multiplies ϕ is rank deficient regardless of the number of measured samples N [Hollerbach et al., 2016].

More specifically, the following vector subspaces of \mathbb{R}^{10n_L} can be defined.

Definition 7.2 (Non-Identifiable Subspace [Sheu and Walker, 1991]). *Given a regressor Y , the inertial parameters non-identifiable subspace N_Y is defined as:*

$$N_Y = \{\phi \in \mathbb{R}^{10n_L} : Y(\omega_B, \alpha_B^g, s, \dot{s}, \ddot{s})\phi = \mathbf{0}, \\ \forall \omega_B \in \mathbb{R}^3, \alpha_B^g \in \mathbb{R}^6, s, \dot{s}, \ddot{s} \in \mathbb{R}^n\}. \quad (7.7)$$

Definition 7.3 (Identifiable Subspace [Sheu and Walker, 1991]). *Given a regressor Y , the inertial parameters identifiable subspace I_Y is defined as the subspace orthogonal to N_Y , i.e.:*

$$I_Y = N_Y^\perp \quad (7.8)$$

In general the space N_Y is non-empty as a consequence of fact that the columns of Y are linearly dependent for any choice of the robot position, velocity and acceleration. Only certain linear combinations of the elements of ϕ influence the measurements and these combinations can be obtained as:

$$\psi = B\pi \quad (7.9)$$

being B a matrix whose columns are an orthonormal base of the identifiable subspace I_Y .

It is then possible to reformulate (7.6) as:

$$\begin{bmatrix} Y^1 B \\ Y^2 B \\ \vdots \\ Y^N B \end{bmatrix} \pi = \begin{bmatrix} Y^1 \\ Y^2 \\ \vdots \\ Y^N \end{bmatrix} \rightarrow \mathbf{G}_N \pi = \mathbf{g}_N, \quad (7.10)$$

with obvious definition for the matrix \mathbf{G}_N and the vector \mathbf{g}_N . Classically, equation (7.10) has been used for the estimation of the base parameters associated with a certain measurement m . This is suitable if the goal of the parametric identification was to improve the prediction of the measurement m itself. In the case of the estimation algorithm presented in Chapter 4, however the measurement available for the inertial parameters identification (i.e. the sum of the measured force-torque acting on a submodel) are different from some measurements we are interested in estimating. For this reason in the next section we investigate the relation between the identifiable subspace of the quantities that we can measure (internal six-axis force-torque sensor) and the identifiable subspace of the measurements that we want to predict (joint torques, estimated as in (7.5)).

7.3 Inertial Parameter Identification and Torque Estimation

In this section we demonstrate that the inertial parameters used for torque estimation are a subset of the inertial parameters that can be estimated from the submodel regressors given in (7.4).

Property 7.1 (Identifiable subspace of sum of two regressors). *Given the regressors Y_a, Y_b and $Y_c = Y_b + Y_a$ the identifiable subspace of Y_c is given by:*

$$I_{Y_c} \subseteq I_{Y_a} + I_{Y_b}. \quad (7.11)$$

Proof. From the definition of non-identifiable subspace (7.7) we have:

$$N_{Y_a} \cap N_{Y_b} \subseteq N_{Y_c}.$$

Using the definition of identifiable subspace (7.8) and the De Morgan laws for vector spaces one obtains (7.11). \square

Property 7.2 (Identifiable subspace of combination of two regressors). *Given two regressors Y_a, Y_b and the combined regressor $Y_c = \begin{bmatrix} Y_a \\ Y_b \end{bmatrix}$ the identifiable subspace of the combined regressor is given by:*

$$I_{Y_c} = I_{Y_a} + I_{Y_b}. \quad (7.12)$$

Proof. From the definition of non-identifiable subspace (7.7) we have:

$$N_{Y_a} \cap N_{Y_b} = N_{Y_c}.$$

Using (7.8) and the De Morgan laws for vector spaces one obtains (7.12). \square

Lemma 7.1 (Identifiable subspace of a regressor multiplied by a full rank square matrix). *Given a regressor Y_a and the regressor obtained by multiplying Y_a by a always full-rank square matrix M : $Y_b = MY_a$ the identifiable subspace of Y_b is equal to the one of Y_a , i.e. :*

$$Y_b = Y_a \quad (7.13)$$

Theorem 7.1. *The identifiable subspace of the base dynamics regressor Y_b is equal to the identifiable subspace of the full dynamics regressor Y , both defined in (7.1), i.e.:*

$$I_Y = I_{Y_b} \quad (7.14)$$

furthermore the shape dynamics regressors Y_s is a subspace of the base dynamics regressor Y_b , i.e.:

$$I_{Y_s} \subseteq I_{Y_b}. \quad (7.15)$$

Proof. The proof for the first part of this theorem is provided in [Ayusawa et al., 2014]. For the second part, it is a consequence of Property 7.2 that $I_{Y_s} \subseteq I_Y$, by combining this with (7.14) one obtains (7.15). \square

Theorem 7.2. *Given a multibody system equipped with internal six-axis force-torque sensors, the identifiable subspace of the dynamics regressor I_Y is a subspace of the identifiable subspace of the combined regressor of all submodel base dynamics $I_{Y_{\mathfrak{M}}}$, i.e.:*

$$I_Y \subseteq I_{Y_{\phi_{\mathfrak{M}}}} \quad (7.16)$$

Proof. From Property 7.2 we have that $I_{Y_{\phi_{\mathfrak{M}}}}$ is given by:

$$I_{Y_{\phi_{\mathfrak{M}}}} = \sum_{sm \in \mathfrak{M}} I_{Y_{\phi_{sm}}} \quad (7.17)$$

while from Property 7.1 and from the structure of Y_b we have that

$$I_{Y_b} \subseteq \sum_{sm \in \mathfrak{M}} I_{Y_{\phi_{sm}}}. \quad (7.18)$$

Combining these two statements with (7.14) one obtains (7.16). \square

Theorem 7.3. *Given a multibody system equipped with internal six-axis force-torque sensors, and one submodel $sm \in \mathfrak{M}$ induced by the force-torque sensors for every joint J belonging to sm the identifiable subspace of the regressor used for torque estimation (7.5) is a subspace of the identifiable subspace of the regressor of the submodel base dynamics $I_{Y_{\mathfrak{M}}}$, i.e.:*

$$\forall J \in \mathfrak{J} - \mathfrak{J}_0 \quad I_{Y_{\hat{\tau}_J}} \subseteq I_{Y_{\phi_{sm}}}. \quad (7.19)$$

Proof. If we consider the submodel sm as an independent multibody system rather than as a submodel, we would have that the base submodel dynamics regressor $Y_{\phi_{sm}}$ would be exactly Y_b , while, by choosing an appropriate base link $I_{Y_{\hat{\tau}_J}}$ would be a line of Y_s . Then, the theorem is demonstrated as a consequence of (7.15). \square

7.4 Experimental results

In this section we present the results obtained on a special version of the iCub humanoid robot, in which some joints are equipped with joint torque sensors and can provide us with a *ground truth* for the estimation of joint torques and the corresponding identification of the inertial parameters.

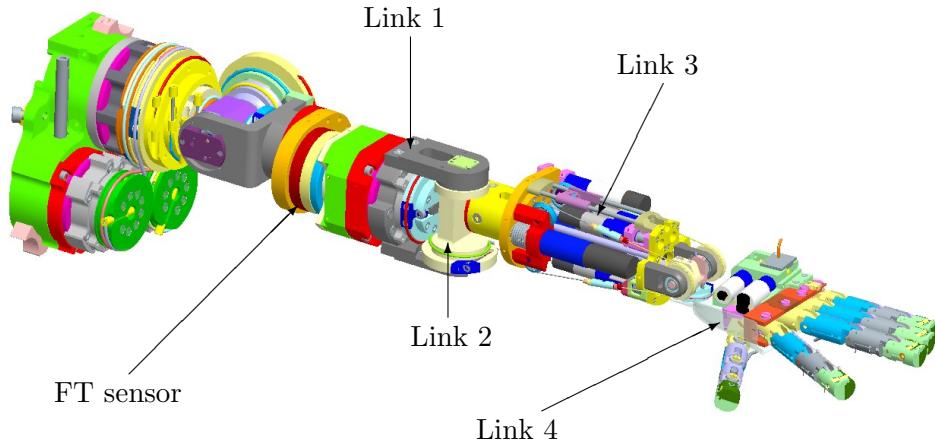


Fig. 7.1 CAD drawing of the seven degree-of-freedom iCub arm used in the experiments. Three out of the four joints (two in the shoulder and one in the elbow) are sensorized with joint level torque sensors. These joints are the ones considered in the proposed experiments.

Experiments were conducted on three joints (pitch and yaw in the shoulder and elbow)¹ of the iCub left arm (see Fig. 7.1). These joints are equipped with joint level torque sensors. Additionally a single F/T sensor is positioned in the middle of the upper arm as represented in Fig. 7.1. In the experiment,

¹http://wiki.icub.org/wiki/ICub_joints

data from the F/T sensor were used to estimate the associated base parameters. Thanks to the theoretical results presented in the previous sections, these parameters coincides with the one used by the method in the rRNEA to obtain an estimation of the joint torques. These estimations have been compared with direct joint torque measurements, used in this framework as a ground truth. Results are presented in Fig. 7.2 where we reported in blue direct joint torque measurements, in red predictions using CAD parameters and in green predictions from the estimation in Fumagalli et al. [2012] supplied with the on-line estimation of the base parameters (presented in Fig. 7.3). At the beginning of the simulation estimated parameters are clearly not sufficiently well estimated to predict with sufficient accuracy the joint torques. This condition holds true until the arm starts moving (vertical solid black line in both Fig. 7.2 and Fig. 7.3).

During the testing trajectories, the end-effector randomly moved in Cartesian space, without any interaction with the environment. Literature on suitable choices of the exciting trajectories is extensive, but such an implementation is out of the scope of the present paper. This simpler choice is also motivated by two factors: first, we need to avoid self-collision of the robot in a simple way; second, we need to generate trajectories similar to those produced during standard operation of the humanoid robot and our goal is to on-line estimate the parameters during standard operations. Joint velocities and accelerations have been estimated using an adaptive-window fitting algorithm Janabi-Sharifi et al. [2000].

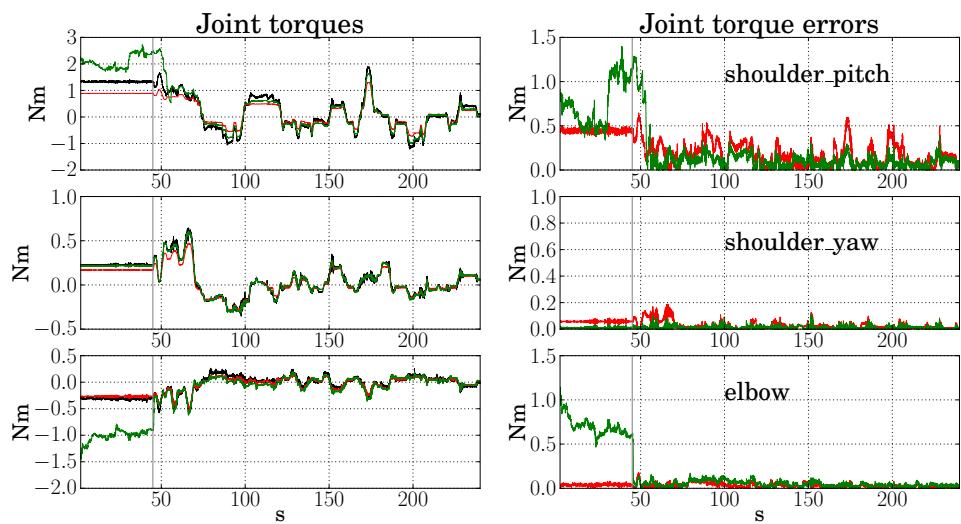


Fig. 7.2 Joint level torques (left part) and errors (right part): measured (black), estimated with CAD parameters (red) and estimated with the procedure in Fumagalli et al. [2012] supplied with identified parameters (green). The vertical solid gray line indicate the movement onset.

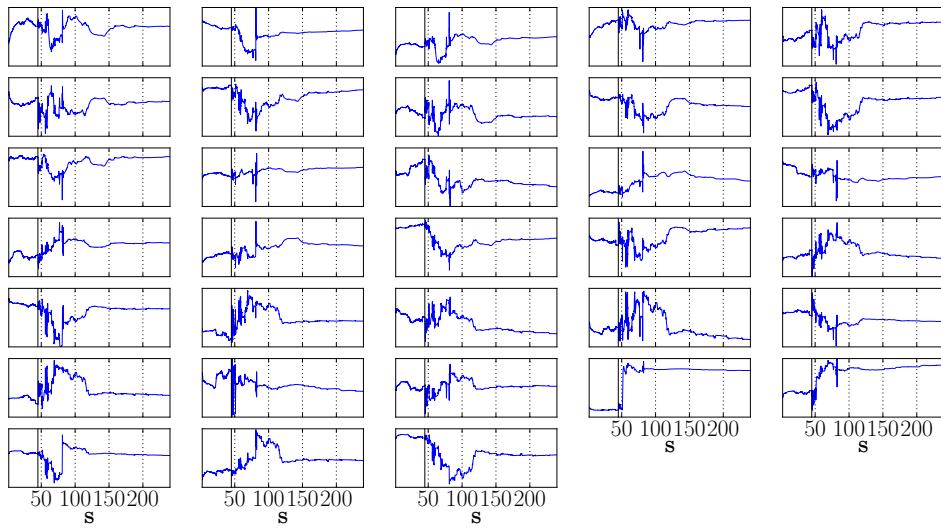


Fig. 7.3 The picture shows the time behavior of the base parameters estimation. The estimation is executed on-line in an iterative fashion. The onset of the movement (vertical solid black line) determines the instant at which the data from the F/T sensor become informative for the estimation problem. Convergence is quite fast and mirrors the behavior of the torques estimation in Fig. 7.2.

Bibliography

- P.-A. Absil, R Mahony, and R Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008. ISBN 9780691132983.
- Matej Andrejašic. MEMS accelerometers. In *University of Ljubljana. Faculty for mathematics and physics, Department of physics, Seminar*. University of Ljubljana, 2008.
- Brian Stewart Randall Armstrong. *Dynamics for Robot Control: Friction Modeling and Ensuring Excitation During Parameter Identification*. PhD thesis, Stanford University, 1988.
- ATI Industrial Automation. Six-Axis Force/Torque Transducer Installation and Operation Manual, 2014.
- Ko Ayusawa, Gentiane Venture, and Yoshihiko Nakamura. Identifiability and identification of inertial parameters using the underactuated base-link dynamics for legged multibody systems. *The International Journal of Robotics Research*, 33(3):446–468, mar 2014. ISSN 0278-3649. doi: 10.1177/0278364913495932. URL <http://ijr.sagepub.com/cgi/doi/10.1177/0278364913495932>
<http://journals.sagepub.com/doi/10.1177/0278364913495932>.
- Jos Baeleman. *Parameter estimation of humanoid robots using the center of pressure*. PhD thesis, Eindhoven University of Technology, 2013.
- Jos Baeleman, Pieter van Zutven, and Henk Nijmeijer. Model parameter estimation of humanoid robots using static contact force measurements. In *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 1–6. IEEE, oct 2013. ISBN 978-1-4799-0880-6. doi: 10.1109/SSRR.2013.6719328. URL <http://ieeexplore.ieee.org/document/6719328/>.
- K Bouyarmane and A Kheddar. On the dynamics modeling of free-floating-base articulated mechanisms and applications to humanoid whole-body dynamics and control. In *Humanoid Robots (Humanoids), 2012 12th*

IEEE-RAS International Conference on, pages 36–42, nov 2012. doi: 10.1109/HUMANOIDS.2012.6651496.

Daniel Braun and H. Wörn. Techniques for Robotic Force Sensor Calibration. *13th International Workshop on Computer Science and Information Technologies (CSIT2011)*, pages 218–223, 2011. ISSN 2225-6024. URL <http://journal.ugatu.ac.ru/index.php/systems/article/view/74>.

Stanislas Brossette, Adrien Escande, Gregoire Duchemin, Benjamin Chretien, and Abderrahmane Kheddar. Humanoid posture generation on non-Euclidean manifolds. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 352–358. IEEE, IEEE, nov 2015. ISBN 978-1-4799-6885-5. doi: 10.1109/HUMANOIDS.2015.7363574. URL <http://ieeexplore.ieee.org/document/7363574/>.

Herman Bruyninckx and Joris De Schutter. Symbolic differentiation of the velocity mapping for a serial kinematic chain. *Mechanism and Machine Theory*, 31(2):135–148, feb 1996. ISSN 0094114X. doi: 10.1016/0094-114X(95)00069-B. URL <http://linkinghub.elsevier.com/retrieve/pii/0094114X9500069B>.

Francesco Bullo and Andrew D. Lewis. *Geometric Control of Mechanical Systems*, volume 49 of *Texts in Applied Mathematics*. Springer New York, New York, NY, 2005. ISBN 978-1-4419-1968-7. doi: 10.1007/978-1-4899-7276-7. URL <http://link.springer.com/10.1007/978-1-4899-7276-7>.

Stefano Chiaverini, Giuseppe Oriolo, and Anthony A Maciejewski. *Redundant Robots*, pages 221–242. Springer International Publishing, Cham, 2016. ISBN 978-3-319-32552-1. doi: 10.1007/978-3-319-32552-1_10. URL http://link.springer.com/10.1007/978-3-319-32552-1_10.

Stefano Dafarra, Francesco Romano, and Francesco Nori. Torque-controlled stepping-strategy push recovery: Design and implementation on the iCub humanoid robot. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 152–157. IEEE, nov 2016. ISBN 978-1-5090-4718-5. doi: 10.1109/HUMANOIDS.2016.7803271. URL <http://ieeexplore.ieee.org/document/7803271/>.

Hongkai Dai, Andres Valenzuela, and Russ Tedrake. Whole-body motion planning with centroidal dynamics and full kinematics. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 295–302. IEEE, IEEE, nov 2014. ISBN 978-1-4799-7174-9. doi: 10.1109/HUMANOIDS.2014.7041375. URL <http://ieeexplore.ieee.org/document/7041375/>.

- T. De Laet, Steven Bellens, Ruben Smits, E. Aertbelien, Herman Bruyninx, and Joris De Schutter. Geometric Relations Between Rigid Bodies (Part 1): Semantics for Standardization. *IEEE Robotics & Automation Magazine*, 20(1):84–93, mar 2013. ISSN 1070-9932. doi: 10.1109/MRA.2012.2205652. URL <http://ieeexplore.ieee.org/document/6476696/>.
- A De Luca, A Albu-Schaffer, S Haddadin, and G Hirzinger. Collision Detection and Safe Reaction with the DLR-III Lightweight Manipulator Arm. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1623–1630, oct 2006. doi: 10.1109/IROS.2006.282053.
- Andrea Del Prete, Simone Denei, Lorenzo Natale, Fulvio Mastrogiovanni, Francesco Nori, Giorgio Cannata, and Giorgio Metta. Skin spatial calibration using force/torque measurements. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3694–3700. IEEE, 2011.
- Andrea Del Prete, Lorenzo Natale, Francesco Nori, and Giorgio Metta. Contact Force Estimations Using Tactile Sensors and Force / Torque Sensors. In *Human Robot Interaction*, pages 0–2, 2012.
- Andrea Del Prete, Nicolas Mansard, Oscar E Ramos, Olivier Stasse, and Francesco Nori. Implementing torque control with high-ratio gear boxes and without joint-torque sensors. *International Journal of Humanoid Robotics*, 13(01):1550044, 2016.
- Jacques Denavit and Richard Scheunemann Hartenberg. *Kinematic Synthesis of Linkages*. 1964. ISBN 0070269106.
- Ernest Doebelin. *Measurement Systems*. McGraw-Hill Science/Engineering/Math, 2003. ISBN 0072990724.
- Duy Nguyen-Tuong and Jan Peters. Using model knowledge for learning inverse dynamics. In *2010 IEEE International Conference on Robotics and Automation*, pages 2677–2682. IEEE, may 2010. ISBN 978-1-4244-5038-1. doi: 10.1109/ROBOT.2010.5509858. URL <http://ieeexplore.ieee.org/document/5509858/>.
- Johannes Adalbert Englsberger. *Combining reduced dynamics models and whole-body control for agile humanoid locomotion*. PhD thesis, 2016. URL <http://d-nb.info/112178027X/34>.
- H. Essen and Hanno Essén. Average Angular Velocity. *European journal of physics*, 14(1993):11, jan 2004. doi: 10.1088/0143-0807/14/5/002. URL <http://arxiv.org/abs/physics/0401146>.

Gert S Faber, Chien-Chi Chang, Idsart Kingma, H Martin Schepers, Sebastiaan Herber, Peter H Veltink, and Jack T Dennerlein. A force plate based method for the calibration of force/torque sensors. *Journal of Biomechanics*, 45(7):1332–1338, apr 2012. ISSN 00219290. doi: 10.1016/j.jbiomech.2012.01.024. URL <http://linkinghub.elsevier.com/retrieve/pii/S0021929012000668>.

Roy Featherstone. The Acceleration Vector of a Rigid Body. *The International Journal of Robotics Research*, 20(11):841–846, nov 2001. ISSN 0278-3649. doi: 10.1177/02783640122068137. URL <http://ijr.sagepub.com/cgi/doi/10.1177/02783640122068137>.

Roy Featherstone. *Rigid Body Dynamics Algorithms*, volume 49. Springer US, Boston, MA, 2008. ISBN 978-0-387-74314-1. doi: 10.1007/978-0-387-74315-8. URL <http://www.springerlink.com/index/10.1007/978-0-387-74315-8>.

Roy Featherstone. A Beginner’s Guide to 6-D Vectors (Part 1). *IEEE Robotics & Automation Magazine*, 17(3):83–94, sep 2010. ISSN 1070-9932. doi: 10.1109/MRA.2010.937853. URL <http://ieeexplore.ieee.org/document/5569032/>.

Roy Featherstone and David E. Orin. Dynamics. In *Springer Handbook of Robotics*, pages 37–66. Springer International Publishing, Cham, 2016. doi: 10.1007/978-3-319-32552-1_3. URL http://link.springer.com/10.1007/978-3-319-32552-1{_}3.

P Fraundorf. A one-map two-clock approach to teaching relativity in introductory physics. 8, 1996.

Matteo Fumagalli, Arjan Gijsberts, Serena Ivaldi, Lorenzo Jamone, Giorgio Metta, Lorenzo Natale, Francesco Nori, and Giulio Sandini. From Motor to Interaction Learning in robotics. In *From Motor Learning to Interaction Learning in Robots*, chapter Learning h, pages 159–177. springer-verlag, 2010a. doi: 10.1007/978-3-642-05181-4_7. URL http://link.springer.com/10.1007/978-3-642-05181-4{_}7.

Matteo Fumagalli, Arjan Gijsberts, Serena Ivaldi, Lorenzo Jamone, Giorgio Metta, Lorenzo Natale, Francesco Nori, and Giulio Sandini. Learning to Exploit Proximal Force Sensing: A Comparison Approach. In *From Motor Learning to Interaction Learning in Robots*, pages 149–167. Springer, 2010b. doi: 10.1007/978-3-642-05181-4_7. URL http://link.springer.com/10.1007/978-3-642-05181-4{_}7.

Matteo Fumagalli, Marco Randazzo, F Nori, Lorenzo Natale, Giorgio Metta, and Giulio Sandini. Exploiting proximal F/T measurements for the

iCub active compliance. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1870–1876. IEEE, oct 2010c. ISBN 978-1-4244-6674-0. doi: 10.1109/IROS.2010.5651421. URL <http://ieeexplore.ieee.org/document/5651421/>.

Matteo Fumagalli, Serena Ivaldi, Marco Randazzo, Lorenzo Natale, Giorgio Metta, Giulio Sandini, and Francesco Nori. Force feedback exploiting tactile and proximal force/torque sensing. *Autonomous Robots*, 33(4):381–398, nov 2012. ISSN 0929-5593. doi: 10.1007/s10514-012-9291-2. URL <http://link.springer.com/10.1007/s10514-012-9291-2>.

Gianluca Garofalo, Christian Ott, and Alin Albu-Schaffer. On the closed form computation of the dynamic matrices and their differentiations. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2364–2359. IEEE, IEEE, nov 2013. ISBN 978-1-4673-6358-7. doi: 10.1109/IROS.2013.6696688. URL <http://ieeexplore.ieee.org/document/6696688/>.

Gianluca Garofalo, Bernd Henze, Johannes Englsberger, and Christian Ott. On the inertially decoupled structure of the floating base robot dynamics. *IFAC-PapersOnLine*, 48(1):322–327, 2015. ISSN 24058963. doi: 10.1016/j.ifacol.2015.05.189. URL <http://linkinghub.elsevier.com/retrieve/pii/S2405896315001901>.

M. Gautier. Dynamic identification of robots with power model. In *Proceedings of International Conference on Robotics and Automation*, volume 3, pages 1922–1927. IEEE. ISBN 0-7803-3612-7. doi: 10.1109/ROBOT.1997.619069. URL <http://ieeexplore.ieee.org/document/619069/>.

M. Gautier and W. Khalil. On the identification of the inertial parameters of robots. In *Proceedings of the 27th IEEE Conference on Decision and Control*, pages 2264–2269. IEEE. doi: 10.1109/CDC.1988.194738. URL <http://ieeexplore.ieee.org/document/194738/>.

Maxime Gautier and Gentiane Venture. Identification of standard dynamic parameters of robots with positive definite inertia matrix. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5815–5820. IEEE, IEEE, nov 2013. ISBN 978-1-4673-6358-7. doi: 10.1109/IROS.2013.6697198. URL <http://ieeexplore.ieee.org/document/6697198/>.

Maxime Gautier, Sébastien Briot, and Gentiane Venture. Identification of consistent standard dynamic parameters of industrial robots. In *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 1429–1435. IEEE, jul 2013. ISBN 978-1-4673-5320-5. doi: 10.1109/AIM.2013.6584295. URL <http://ieeexplore.ieee.org/document/6584295/>.

Daoxiong Gong, Yunyi Jia, Yu Cheng, and Ning Xi. On-line and simultaneous calibration of wrist-mounted Force/Torque sensor and tool Forces/Torques for manipulation. In *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, number December, pages 619–624. IEEE, dec 2013. ISBN 978-1-4799-2744-9. doi: 10.1109/ROBIO.2013.6739528. URL <http://ieeexplore.ieee.org/document/6739528/>.

Nuno Guedelha, Naveen Kuppuswamy, Silvio Traversaro, and Francesco Nori. Self-calibration of joint offsets for humanoid robots using accelerometer measurements. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 1233–1238. IEEE, nov 2016. ISBN 978-1-5090-4718-5. doi: 10.1109/HUMANOIDS.2016.7803427. URL <http://ieeexplore.ieee.org/document/7803427/>.

Brian C Hall. *Lie Groups, Lie Algebras, and Representations*, volume 222 of *Graduate Texts in Mathematics*. Springer New York, New York, NY, 2003. ISBN 978-1-4419-2313-4. doi: 10.1007/978-0-387-21554-9. URL <http://link.springer.com/10.1007/978-0-387-21554-9>.

Alexander Herzog, Nicholas Rotella, Sean Mason, Felix Grimmerger, Stefan Schaal, and Ludovic Righetti. Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid. *Autonomous Robots*, 40(3):473–491, mar 2016. ISSN 0929-5593. doi: 10.1007/s10514-015-9476-6. URL <http://link.springer.com/10.1007/s10514-015-9476-6>.

John Hollerbach, Wisama Khalil, and Maxime Gautier. Model Identification. In *Springer Handbook of Robotics*, pages 113–138. Springer International Publishing, Cham, 2016. doi: 10.1007/978-3-319-32552-1_6. URL http://link.springer.com/10.1007/978-3-319-32552-1_6.

Serena Ivaldi, M Fumagalli, M Randazzo, F Nori, G Metta, and G Sandini. Computing robot internal/external wrenches by means of inertial, tactile and F/T sensors: Theory and implementation on the iCub. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 521–528, oct 2011. doi: 10.1109/Humanoids.2011.6100813.

Jaeheung Park and Oussama Khatib. Multi-Link Multi-Contact Force Control for Manipulators. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, number April, pages 3613–3618. IEEE, apr 2005. ISBN 0-7803-8914-X. doi: 10.1109/ROBOT.2005.1570670. URL <http://ieeexplore.ieee.org/document/1570670/>.

Abhinandan Jain. *Robot and Multibody Dynamics: Analysis and Algorithms*. Springer, 2010.

Abhinandan Jain. Graph theoretic foundations of multibody dynamics. *Multibody System Dynamics*, 26(3):307–333, oct 2011. ISSN 1384-5640.

doi: 10.1007/s11044-011-9266-7. URL <http://link.springer.com/10.1007/s11044-011-9266-7>.

Farrokh Janabi-Sharifi, Vincent Hayward, and C.-S.J. Chen. Discrete-time adaptive windowing for velocity estimation. *IEEE Transactions on Control Systems Technology*, 8(6):1003–1009, nov 2000. ISSN 10636536. doi: 10.1109/87.880606. URL <http://ieeexplore.ieee.org/document/880606/>.

Julius Jelinek and D. H. Li. Separation of the Energy of Overall Rotation in Any N-Body System. *Physical Review Letters*, 62(3):241–244, jan 1989. ISSN 0031-9007. doi: 10.1103/PhysRevLett.62.241. URL <http://link.aps.org/doi/10.1103/PhysRevLett.62.241>.

Jovana Jovic, Franck Philipp, Adrien Escande, Ko Ayusawa, Eiichi Yoshida, Abderrahmane Kheddar, and Gentiane Venture. Identification of dynamics of humanoids: Systematic exciting motion generation. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2173–2179. IEEE, IEEE, sep 2015. ISBN 978-1-4799-9994-1. doi: 10.1109/IROS.2015.7353668. URL <http://ieeexplore.ieee.org/document/7353668/>.

J Kim. Lie Group Formulation of Articulated Rigid Body Dynamics. 2012. URL <http://www.cs.cmu.edu/~junggon/tools/liegroupdynamics.pdf>.

Twan Koolen, Sylvain Bertrand, Gray Thomas, Tomas de Boer, Tingfan Wu, Jesper Smith, Johannes Englsberger, and Jerry Pratt. Design of a Momentum-Based Control Framework and Application to the Humanoid Robot Atlas. *International Journal of Humanoid Robotics*, 13(01):1650007, mar 2016. ISSN 0219-8436. doi: 10.1142/S0219843616500079. URL <http://www.worldscientific.com/doi/abs/10.1142/S0219843616500079>.

Daniel Kubus, T. Kroger, and F.M. Wahl. On-line estimation of inertial parameters using a recursive total least-squares approach. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3845–3852. IEEE, IEEE, sep 2008. ISBN 978-1-4244-2057-5. doi: 10.1109/IROS.2008.4650672. URL <http://ieeexplore.ieee.org/document/4650672/>.

Craig Lawrence, Jian L Zhou, and André L Tits. User’s Guide for {CFSQP} Version 2.5, 1997.

Sung-Hee Lee and Ambarish Goswami. A momentum-based balance controller for humanoid robots on non-level and non-stationary ground. *Autonomous Robots*, 33(4):399–414, nov 2012. ISSN 0929-5593. doi: 10.

1007/s10514-012-9294-z. URL <http://link.springer.com/10.1007/s10514-012-9294-z>.

J. Luh, W. Fisher, and R. Paul. Joint torque control by a direct feedback for industrial robots. *IEEE Transactions on Automatic Control*, 28(2):153–161, feb 1983. ISSN 0018-9286. doi: 10.1109/TAC.1983.1103215. URL <http://ieeexplore.ieee.org/document/1103215/>.

E Magrini, F Flacco, and A De Luca. Estimation of contact forces using a virtual force sensor. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2126–2133, sep 2014. doi: 10.1109/IROS.2014.6942848.

Perla Maiolino, Marco Maggiali, Giorgio Cannata, Giorgio Metta, and Lorenzo Natale. A Flexible and Robust Large Scale Capacitive Tactile Sensor for Robots. *IEEE Sensors Journal*, 13(10):3910–3917, oct 2013. ISSN 1530-437X. doi: 10.1109/JSEN.2013.2258149. URL <http://ieeexplore.ieee.org/document/6502183/>.

Lucas Manuelli and Russ Tedrake. Localizing external contact using proprioceptive sensors: The contact particle filter. *Under review*, 2016.

J E Marsden and T Ratiu. *Introduction to mechanics and symmetry: a basic exposition of classical mechanical systems*. Springer, 2nd edition, 1999.

Jerrold E Marsden and Tudor Ratiu. *Introduction to mechanics and symmetry: a basic exposition of classical mechanical systems*, volume 17. Springer Science & Business Media, 2013.

Jerrold E. Marsden and J. Scheurle. The reduced euler-lagrange equations. *Fields Institute Communications*, 1:139–164, 1993.

Vicente Mata, Francesc Benimeli, Nidal Farhat, and Angel Valera. Dynamic parameter identification in industrial robots considering physical feasibility. *Advanced Robotics*, 19(1):101–119, jan 2005. ISSN 0169-1864. doi: 10.1163/1568553053020269. URL <http://www.tandfonline.com/doi/abs/10.1163/1568553053020269>.

John J McPhee. On the use of linear graph theory in multibody system dynamics. *Nonlinear Dynamics*, 9(1-2):73–90, feb 1996. ISSN 0924-090X. doi: 10.1007/BF01833294. URL <http://link.springer.com/10.1007/BF01833294>.

Michael Mistry, Stefan Schaal, and Katsu Yamane. Inertial parameter estimation of floating base humanoid systems using partial force sensing. In *2009 9th IEEE-RAS International Conference on Humanoid Robots*, pages 492–497. IEEE, dec 2009. ISBN 978-1-4244-4597-4. doi: 10.1109/

ICHR.2009.5379531. URL <http://ieeexplore.ieee.org/document/5379531/>.

Yuichi Morita and Kouhei Ohnishi. Attitude control of hopping robot using angular momentum. In *IEEE International Conference on Industrial Technology, 2003*, volume 1, pages 173–178. IEEE, IEEE, 2003. ISBN 0-7803-7852-0. doi: 10.1109/ICIT.2003.1290263. URL <http://ieeexplore.ieee.org/document/1290263/>.

R.M. Richard M Murray, Zexiang Li, and S.S. Sastry. *A mathematical introduction to robotic manipulation*. CRC, 1994. ISBN 9780849379819.

Gabriele Nava, Francesco Romano, Francesco Nori, and Daniele Pucci. Stability analysis and design of momentum-based controllers for humanoid robots. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 680–687. IEEE, oct 2016. ISBN 978-1-5090-3762-9. doi: 10.1109/IROS.2016.7759126. URL <http://ieeexplore.ieee.org/document/7759126/>.

Francesco Nori, Silvio Traversaro, Jorhabib Eljaik, Francesco Romano, Andrea Del Prete, and Daniele Pucci. iCub Whole-Body Control through Force Regulation on Rigid Non-Coplanar Contacts. *Frontiers in Robotics and AI*, 2:6, mar 2015. ISSN 2296-9144. doi: 10.3389/frobt.2015.00006. URL http://www.frontiersin.org/Humanoid{_}Robotics/10.3389/frobt.2015.00006/abstract.

C M Oddo, P Valdastri, L Beccai, S Roccella, M C Carrozza, and P Dario. Investigation on calibration methods for multi-axis, linear and redundant force sensors. *Measurement Science and Technology*, 18(3):623–631, mar 2007. ISSN 0957-0233. doi: 10.1088/0957-0233/18/3/011. URL <http://stacks.iop.org/0957-0233/18/i=3/a=011?key=crossref.3dae6b56cc5ec22def5f10498b687075>.

Yusuke Ogawa, Gentiane Venture, and Christian Ott. Dynamic parameters identification of a humanoid robot using joint torque sensors and/or contact forces. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 457–462. IEEE, nov 2014. ISBN 978-1-4799-7174-9. doi: 10.1109/HUMANOIDS.2014.7041401. URL <http://ieeexplore.ieee.org/document/7041401/>.

David E. Orin, Ambarish Goswami, and Sung-Hee Lee. Centroidal dynamics of a humanoid robot. *Autonomous Robots*, 35(2-3):161–176, jun 2013. ISSN 0929-5593. doi: 10.1007/s10514-013-9341-4. URL <http://link.springer.com/10.1007/s10514-013-9341-4>.

- D.E. Orin and Ambarish Goswami. Centroidal Momentum Matrix of a humanoid robot: Structure and properties. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 653–659. IEEE, IEEE, sep 2008. ISBN 978-1-4244-2057-5. doi: 10.1109/IROS.2008.4650772. URL <http://ieeexplore.ieee.org/document/4650772/>.
- Christian Ott, Maximo A Roa, and Gerd Hirzinger. Posture and balance control for biped robots based on contact force optimization. In *2011 11th IEEE-RAS International Conference on Humanoid Robots*, pages 26–33. IEEE, IEEE, oct 2011. ISBN 978-1-61284-868-6. doi: 10.1109/Humanoids.2011.6100882. URL <http://ieeexplore.ieee.org/document/6100882/>.
- Nicholas Paine, Joshua S. Mehling, James Holley, Nicolaus A. Radford, Gwendolyn Johnson, Chien-Liang Fok, and Luis Sentis. Actuator Control for the NASA-JSC Valkyrie Humanoid Robot: A Decoupled Dynamics Approach for Torque Control of Series Elastic Robots. *Journal of Field Robotics*, 32(3):378–396, may 2015. ISSN 15564959. doi: 10.1002/rob.21556. URL <http://doi.wiley.com/10.1002/rob.21556>.
- F.C. Park, J.E. Bobrow, and S.R. Ploen. A Lie Group Formulation of Robot Dynamics. *The International Journal of Robotics Research*, 14(6):609–618, dec 1995. ISSN 0278-3649. doi: 10.1177/027836499501400606. URL <http://ijr.sagepub.com/cgi/doi/10.1177/027836499501400606>.
- Alberto Parmiggiani, Marco Maggiali, Lorenzo Natale, Francesco Nori, Alexander Schmitz, Nikos G Tsagarakis, Jose Santos Victor, Francesco Bechi, Giulio Sandini, and Giorgio Metta. The Design of the iCub Humanoid Robot. *International Journal of Humanoid Robotics*, 09(04), dec 2012. ISSN 0219-8436. doi: 10.1142/S0219843612500272. URL [http://apps.webofknowledge.com/full{_\]record.do?product=UA{&}search{_\]mode=GeneralSearch{&}qid=2{&}SID=P1x6BurusM6MbHu4gG1{&}page=2{&}doc=13http://www.worldscientific.com/doi/abs/10.1142/S0219843612500272](http://apps.webofknowledge.com/full{_]record.do?product=UA{&}search{_]mode=GeneralSearch{&}qid=2{&}SID=P1x6BurusM6MbHu4gG1{&}page=2{&}doc=13http://www.worldscientific.com/doi/abs/10.1142/S0219843612500272).
- Anna Petrovskaya, Jaeheung Park, and Oussama Khatib. Probabilistic Estimation of Whole Body Contacts for Multi-Contact Robot Control. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, number c, pages 568–573. IEEE, apr 2007. ISBN 1-4244-0602-1. doi: 10.1109/ROBOT.2007.363047. URL <http://ieeexplore.ieee.org/document/4209151/>.
- C.M. Pham and Maxime Gautier. Essential parameters of robots. In *[1991] Proceedings of the 30th IEEE Conference on Decision and Control*, pages 2769–2774. IEEE, IEEE, 1991. ISBN 0-7803-0450-0. doi: 10.1109/CDC.1991.261862. URL <http://ieeexplore.ieee.org/document/261862/>.

Daniele Pucci, Gabriele Nava, and Francesco Nori. Automatic gain tuning of a momentum based balancing controller for humanoid robots. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 158–164. IEEE, nov 2016a. ISBN 978-1-5090-4718-5. doi: 10.1109/HUMANOIDS.2016.7803272. URL <http://ieeexplore.ieee.org/document/7803272/>.

Daniele Pucci, Francesco Romano, Silvio Traversaro, and Francesco Nori. Highly dynamic balancing via force control. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 141–141. IEEE, nov 2016b. ISBN 978-1-5090-4718-5. doi: 10.1109/HUMANOIDS.2016.7803266. URL <http://ieeexplore.ieee.org/document/7803266/>.

Marco Randazzo, Matteo Fumagalli, F Nori, Lorenzo Natale, Giorgio Metta, and Giulio Sandini. \hhref{http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6048660}{A comparison between joint level torque sensing and proximal F/T sensor torque estimation: implementation on the iCub}. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4161–4167. IEEE, 2011.

Hamid Roozbahani, Alireza Fakhrizadeh, Heikki Haario, and Heikki Handroos. Novel Online Re-Calibration Method for Multi-Axis Force/Torque Sensor of ITER Welding/Machining Robot. *IEEE Sensors Journal*, 13(11):4432–4443, nov 2013. ISSN 1530-437X. doi: 10.1109/JSEN.2013.2274195. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6564411>.

Alessandro Saccon, Silvio Traversaro, Francesco Nori, and Hendrik Nijmeijer. On Centroidal Dynamics and Integrability of Average Angular Velocity. *IEEE Robotics and Automation Letters*, pages 1–1, 2017. ISSN 2377-3766. doi: 10.1109/LRA.2017.2655560. URL <http://ieeexplore.ieee.org/document/7827045/>.

Giulio Sandini, Giorgio Metta, and David Vernon. RobotCub: an open framework for research in embodied cognition. In *4th IEEE/RAS International Conference on Humanoid Robots, 2004.*, volume 1, pages 13–32. IEEE, 2004. ISBN 0-7803-8863-1. doi: 10.1109/ICHR.2004.1442111. URL <http://ieeexplore.ieee.org/document/1442111/>.

J M Selig. *Geometric Fundamentals of Robotics*. Monographs in Computer Science. Springer New York, New York, NY, 2005. ISBN 978-0-387-20874-9. doi: 10.1007/b138859. URL <http://link.springer.com/10.1007/b138859>.

- Ajay Seth, Michael Sherman, Peter Eastman, and Scott L. Delp. Minimal formulation of joint motion for biomechanisms. *Nonlinear Dynamics*, 62(1-2):291–303, oct 2010. ISSN 0924-090X. doi: 10.1007/s11071-010-9717-3. URL <http://link.springer.com/10.1007/s11071-010-9717-3>.
- S.-Y. Sheu and Michael W Walker. Identifying the Independent Inertial Parameter Space of Robot Manipulators. *The International Journal of Robotics Research*, 10(6):668–683, dec 1991. ISSN 0278-3649. doi: 10.1177/027836499101000606. URL <http://ijr.sagepub.com/cgi/doi/10.1177/027836499101000606>.
- B Shimano and B Roth. On force sensing information and its use in controlling manipulators. In *Proceedings of the IFAC International Symposium on Information-Control Problems in Manufacturing Technology*, pages 119–126, 1977.
- Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: modelling, planning and control*. Advanced Textbooks in Control and Signal Processing. Springer Publishing Company, Incorporated, 1st edition, 2008. ISBN 1846286417, 9781846286414. URL <http://books.google.com/books?hl=en&lr={}&id=jPCAFmE-logC&oi=fnd&pg=PR8&dq=Robotics+-+Modelling+Planning+and+Control&ots=3TMi0jIsuu&sig=VoMpDiIASZGcrAg7juLabQszoqE>.
- Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: modelling, planning and control*. Advanced Textbooks in Control and Signal Processing. Springer, 2009. ISBN 9781846286414. URL <http://books.google.com/books?hl=en&lr={}&id=jPCAFmE-logC&oi=fnd&pg=PR8&dq=Robotics+-+Modelling+Planning+and+Control&ots=3TMi0jIsuu&sig=VoMpDiIASZGcrAg7juLabQszoqE>.
- C. D. Sousa and R. Cortesao. Physical feasibility of robot base inertial parameter identification: A linear matrix inequality approach. *The International Journal of Robotics Research*, 33(6):931–944, feb 2014. ISSN 0278-3649. doi: 10.1177/0278364913514870. URL <http://ijr.sagepub.com/cgi/doi/10.1177/0278364913514870>.
- M W Spong, S Hutchinson, and M Vidyasagar. *Robot modeling and control*, volume 3. Wiley New York, 2006.
- John Stillwell. *Naive Lie Theory*. Undergraduate Texts in Mathematics. Springer New York, New York, NY, 2008. ISBN 978-0-387-78214-0. doi: 10.1007/978-0-387-78214-0. URL <http://link.springer.com/10.1007/978-0-387-78214-0>.

- John Stuelpnagel. On the Parametrization of the Three-Dimensional Rotation Group. *SIAM Review*, 6(4):422–430, 1964. ISSN 0036-1445. doi: 10.1137/1006093. URL <http://pubs.siam.org/doi/abs/10.1137/1006093>.
- Italia’s Got Talent. iCub, concorrente artificiale, 2016. URL <https://www.youtube.com/watch?v=5-1VcTmJk0k>.
- Silvio Traversaro, Daniele Pucci, and Francesco Nori. In situ calibration of six-axis force-torque sensors using accelerometer measurements. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2111–2116. IEEE, IEEE, may 2015. ISBN 978-1-4799-6923-4. doi: 10.1109/ICRA.2015.7139477. URL <http://ieeexplore.ieee.org/document/7139477/>.
- Silvio Traversaro, Francesco Romano, Andrea Del Prete, Daniele Pucci, and Francesco Nori. iDynTree : Dynamics Library designed for Free Floating Robots, 2017. URL <https://github.com/robotology/idyntree>.
- M Uchiyama, E Bayo, and E Palma-Villalon. A systematic design procedure to minimize a performance index for robot force sensors. *Journal of Dynamic Systems, Measurement, and Control*, 113(3):388–394, 1991.
- Gijs van Oort. *Analysis, control and design of walking robots*. PhD thesis, University of Twente, Enschede, The Netherlands, oct 2011. URL <http://purl.org/utwente/doi/10.3990/1.9789036532648>.
- R. M. Voyles, J. D. Morrow, and P. K. Khosla. The Shape From Motion Approach to Rapid and Precise Force/Torque Sensor Calibration. *Journal of Dynamic Systems, Measurement, and Control*, 119(2):229, 1997. ISSN 00220434. doi: 10.1115/1.2801238. URL <http://dynamicsystems.asmedigitalcollection.asme.org/article.aspx?articleid=1407475>.
- P C Watson and S H Drake. Pedestal and wrist force sensors for automatic assembly. In *Proceedings of the 5th International Symposium on Industrial Robots*, pages 501–511, 1975.
- Weiss Robotics. KMS 40 Assembly and Operating Manual, 2013.
- Patrick M. Wensing and David E Orin. Improved Computation of the Humanoid Centroidal Dynamics and Application for Whole-Body Control. *International Journal of Humanoid Robotics*, 13(01):1550039, mar 2016. ISSN 0219-8436. doi: 10.1142/S0219843615500395. URL <http://www.worldscientific.com/doi/10.1142/S0219843615500395>.

Patrick M. Wensing, Albert Wang, Sangok Seok, David Otten, Jeffrey Lang, and Sangbae Kim. Proprioceptive Actuator Design in the MIT Cheetah: Impact Mitigation and High-Bandwidth Physical Interaction for Dynamic Legged Robots. *IEEE Transactions on Robotics*, pages 1–14, 2017. ISSN 1552-3098. doi: 10.1109/TRO.2016.2640183. URL <http://ieeexplore.ieee.org/document/7827048/>.

Pierre-Brice Wieber, Russ Tedrake, and Scott Kuindersma. Modeling and Control of Legged Robots. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, pages 1203–1234. Springer International Publishing, Cham, 2016. doi: 10.1007/978-3-319-32552-1_48. URL http://link.springer.com/10.1007/978-3-319-32552-1{_}48.

Jens Wittenburg. *Dynamics of multibody systems*. Springer Science & Business Media, 2007.

Chi-haur Wu Chi-haur Wu and Richard P Paul Richard P Paul. Manipulator compliance based on joint torque control. *1980 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, 19, 1980. doi: 10.1109/CDC.1980.272025.

Koji Yoshida and Wisama Khalil. Verification of the Positive Definiteness of the Inertial Matrix of Manipulators Using Base Inertial Parameters. *The International Journal of Robotics Research*, 19(5):498–510, may 2000. ISSN 0278-3649. doi: 10.1177/02783640022066996. URL <http://journals.sagepub.com/doi/10.1177/02783640022066996>.

Koji Yoshida, Koichi Osuka, Hirokazu Mayeda, and Toshiro Ono. When is the set of base parameter values physically impossible? In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, volume 1, pages 335–342. IEEE, IEEE, 1994. ISBN 0-7803-1933-8. doi: 10.1109/IROS.1994.407372. URL <http://ieeexplore.ieee.org/document/407372/>.

H Zhang, S Ahmad, and G Liu. Torque Estimation for Robotic Joint With Harmonic Drive Transmission Based on Position Measurements. *IEEE Transactions on Robotics*, 31(2):322–330, apr 2015. ISSN 1552-3098. doi: 10.1109/TRO.2015.2402511.

Appendices

Appendix A

Mathematical background on Lie group formalism

In this appendix we will briefly define the concepts necessary to derive the equations of motions from the principle of least action. To ensure that this section is accessible to readers without an extensive background in differential geometry, we will only introduce concepts related to matrices of real elements, rather than more abstract representations.

The reader interested in Lie Groups is referred to [Hall, 2003] and [Stillwell, 2008] for a basic overview of Matrix Lie Group theory, to [Selig, 2005] and [Murray et al., 1994, Appendix] for Lie Group applications in robotics and to [Marsden and Ratiu, 1999] for an advanced approach to Lie Group and their connection to mechanical systems.

A.1 Matrix Lie Groups

Definition A.1 (Group). *A set G associated with a binary operation $\cdot : G \times G \mapsto G$ is called a group (G, \cdot) if the following properties are respected:*

- Closure: *For every $g_1, g_2 \in G$ their operation $g_1 \cdot g_2$ belongs to G , i.e. G is the domain of the group operation.¹*
- Associativity: *For every $g_1, g_2, g_3 \in G$, $(g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$.*
- Existence of the identity: *There exist an element of the group, called identity $I \in G$, such that for every $g \in G$ we have $I \cdot g = g \cdot I = g$.*

¹This condition is implied in the definition of the binary operation, but it is typically included in the group definition in literature.

- Existence of the inverse: *For each element $g \in G$ there exist another element, called inverse of g : $g^{-1} \in G$ such that $g \cdot g^{-1} = g^{-1} \cdot g = I$.*

Definition A.2 (Matrix Group). *A group (G, \cdot) in which:*

- *the set is a subset of $\mathbb{R}^{n \times n}$,*
- *the group operation is the matrix multiplication,*
- *the inverse is the matrix inversion,*

is called Matrix Group.

Definition A.3 (Matrix Lie Group, [Hall, 2003] Definition 1.4). *Given a Matrix Group G , and if the following property holds for G :*

- *if A_m is any sequence of matrices in G , and A_m converges to some matrix A , then either A is in G or A is not invertible,*

then G is Matrix Lie Group.

Remark A.1. *A group in which the group elements are matrices, but in which the group operation is not the matrix multiplication is not a matrix group. Similarly, a Lie group in which the group elements are matrices but in which the group operation is not the matrix multiplication is not a matrix Lie group.*

Definition A.4 (Matrix Exponential). *Given a Matrix $X \in \mathbb{R}^{n \times n}$, the exponential pf the matrix X , indicated with $\exp(X)$, is defined as:*

$$\exp(X) = \sum_{i=0}^{\infty} \frac{X^i}{i!}. \quad (\text{A.1})$$

Definition A.5 (Vector Derivative of Scalar-Valued Function). *Given a scalar-valued vector function $l(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}$, the vector derivative of $l(u)$, indicated with $\frac{\partial l}{\partial u} \in \mathbb{R}^n$, is defined as:*

$$\left(\frac{\partial l}{\partial u} \right)_i = \frac{\partial l}{\partial u_i}. \quad (\text{A.2})$$

Definition A.6 (Vector Derivative of Vector-Valued Function). *Given a vector-valued vector function $f(\cdot) : \mathbb{R}^n \mapsto \mathbb{R}^m$, the vector derivative of $f(u)$, indicated with $\frac{\partial f}{\partial u} \in \mathbb{R}^{n \times m}$, is defined as:*

$$\left(\frac{\partial f}{\partial u} \right)_{ij} = \frac{\partial f_j}{\partial u_i}. \quad (\text{A.3})$$

If the argument u depends itself a scalar variable $t \in \mathbb{R}$, the total derivative of the composite function $f(u(t))$ can be written as:

$$\frac{d}{dt}(f) = \left(\frac{\partial f}{\partial u} \right)^T \frac{du}{dt}. \quad (\text{A.4})$$

Definition A.7 (Matrix Derivative). Given a function scalar-valued matrix function $l(\cdot) : \mathbb{R}^{n \times n} \mapsto \mathbb{R}$, the matrix derivative of $l(X)$, indicated with $\frac{\partial l}{\partial X} \in \mathbb{R}^{n \times n}$, is defined as:

$$\left(\frac{\partial l}{\partial X} \right)_{ij} = \frac{\partial l}{\partial X_{ij}}. \quad (\text{A.5})$$

Definition A.8 (Tangent Space of a Matrix Lie Group). Given a matrix Lie group element $g \in G$, the tangent space $T_g G$ is the space of all matrices $X \in \mathbb{R}^{n \times n}$ such that there exist a smooth function $\gamma(t) : \mathbb{R} \mapsto \mathbb{R}^{n \times n}$ for which:

$$X = \left(\frac{d\gamma(t)}{dt} \Big|_{t=T} \right), \quad \gamma(T) = G \quad (\text{A.6})$$

Definition A.9 (Dual of Tangent space). Let $T_g G$ be the tangent space of a matrix Lie group G in its point g . Its dual space $T_g^* G$ is defined as the space of all the linear functions from $T_g G$ to \mathbb{R} :

$$T_g^* G = \{f(\cdot) : T_g G \mapsto \mathbb{R} \mid \forall X, Y \in T_g G \quad f(X + Y) = fX + fY\}. \quad (\text{A.7})$$

Remark A.2. If a $G \subseteq \mathbb{R}^{n \times n}$, then a generic element of $T_g^* G$ can be written as a matrix $f \in \mathbb{R}^{n \times n}$, and its application to an element $X \in T_g G$ can be expressed as

$$\langle f, X \rangle = \sum_{ij} f_{ij} X_{ij}. \quad (\text{A.8})$$

Definition A.10 (Lie algebra of a matrix Lie group, [Hall, 2003] Definition 3.18). Let G be a Matrix Lie Group. The Lie Algebra \mathfrak{g} of G is the set of all matrices $X \in \mathbb{R}^{n \times n}$ such that $\exp(Xt)$ is in G for all $t \in \mathbb{R}$, i.e. :

$$\mathfrak{g} := \{X \in \mathbb{R}^{n \times n} \mid \exp(Xt) \in G \quad \forall t \in \mathbb{R}\}. \quad (\text{A.9})$$

Furthermore, for every X and Y that belong to \mathfrak{g} , we have that $XY - YX$ belongs to \mathfrak{g} . We then define the Lie Brackets $[\cdot, \cdot] : \mathfrak{g} \times \mathfrak{g} \mapsto \mathfrak{g}$ as:

$$[X, Y] = XY - YX. \quad (\text{A.10})$$

Definition A.11 (Dual space of the Lie algebra). *Let \mathfrak{g} be a Lie algebra of a Matrix Lie Group. Its dual space \mathfrak{g}^* is defined as the space of all the linear functions from \mathfrak{g} to \mathbb{R} :*

$$\mathfrak{g}^* = \{f(\cdot) : \mathfrak{g} \mapsto \mathbb{R} \mid \forall X, Y \in \mathfrak{g} \quad f(X + Y) = fX + fY\}. \quad (\text{A.11})$$

Lemma A.1 (Lie algebra as the Tangent Space of the Identity). *Let \mathfrak{g} be a Lie algebra of a matrix Lie group G . We have that (as a set) the Lie algebra coincides with the Tangent space of G at the identity.*

$$\mathfrak{g} = T_{1_n} G. \quad (\text{A.12})$$

Furthermore for the dual space \mathfrak{g}^* we have that:

$$\mathfrak{g}^* = T_{1_n}^* G. \quad (\text{A.13})$$

Lemma A.2 (Trivialization of Tangent Space). *Let $\delta g \in T_g G$ be an element of the tangent space of the matrix Lie group G . Then $g^{-1}\delta g$ belongs to \mathfrak{g} and is called left-trivialization of δg . Furthermore δgg^{-1} belongs to \mathfrak{g} and is called right-trivialization of δg .*

Lemma A.3 (Trivialization of the dual Tangent Space). *Let $f \in T_g^* G$ be an element of the dual tangent space of the matrix Lie group G . Then the left-trivialization of f , indicated with $g^{-1}f \in \mathfrak{g}^*$ is defined as:*

$$\langle g^{-1}f, \xi \rangle = \langle f, g\xi \rangle \quad (\text{A.14})$$

with $\xi \in \mathfrak{g}$.

Similarly the right-trivialization of $f \in T_g^* G$, indicated with $fg^{-1} \in \mathfrak{g}^*$ is defined as:

$$\langle fg^{-1}, \xi \rangle = \langle f, \xi g \rangle \quad (\text{A.15})$$

with $\xi \in \mathfrak{g}$.

Remark A.3. Note that while $f \in T_g^* G$ can be represented by a $n \times n$ matrix as discussed in Remark A.2, the left trivialization $g^{-1}f$ is not the matrix multiplication of $g^{-1} \in \mathbb{R}^{n \times n}$ times the matrix representing f .

Definition A.12 (Adjoint action of a matrix Lie group on its Lie algebra, [Hall, 2003] Definition 3.32). *Let G be a matrix Lie group, with \mathfrak{g} its Lie algebra. Then for each $A \in G$, the Adjoint Map of the group $\text{Ad}_A : \mathfrak{g} \mapsto \mathfrak{g}$ is defined as:*

$$\text{Ad}_A X = AXA^{-1}. \quad (\text{A.16})$$

Definition A.13 (Adjoint action of a matrix Lie group on the dual space of its Lie algebra). *Let G be a matrix Lie group, with \mathfrak{g} its Lie algebra and \mathfrak{g}^* its dual space. Then for each $A \in G$, the Adjoint action of the group on the dual space $\text{Ad}_A : \mathfrak{g} \mapsto \mathfrak{g}$ is defined as:*

$$\langle \text{Ad}_A^* f, X \rangle = \langle f, \text{Ad}_A Y \rangle. \quad (\text{A.17})$$

Definition A.14 (Adjoint action of a Lie algebra on itself, [Hall, 2003] Definition 3.7). *Let \mathfrak{g} be a Lie algebra of a matrix Lie group G . Then for each $X \in \mathfrak{g}$, the Adjoint Map of the algebra $\text{ad}_X : \mathfrak{g} \mapsto \mathfrak{g}$ is defined as:*

$$\text{ad}_X Y = [X, Y]. \quad (\text{A.18})$$

Definition A.15 (Adjoint action of a Lie algebra on its dual space). *Let \mathfrak{g} be a Lie algebra of a matrix Lie group G , and let \mathfrak{g}^* be its dual space. Then given $X, Y \in \mathfrak{g}$ and $f \in \mathfrak{g}^*$, the Adjoint action of the algebra on the dual space $\text{ad}_X^* : \mathfrak{g}^* \mapsto \mathfrak{g}^*$ is defined as:*

$$\langle \text{ad}_X^* f, Y \rangle = \langle f, \text{ad}_X Y \rangle. \quad (\text{A.19})$$

A.1.1 Matrix Lie Group Examples

Rotation matrices

The set $\text{SO}(3)$ is the set of $\mathbb{R}^{3 \times 3}$ orthogonal matrices with determinant equal to one, namely

$$\text{SO}(3) := \{ R \in \mathbb{R}^{3 \times 3} \mid R^T R = I_3, \det(R) = 1 \}. \quad (\text{A.20})$$

When endowed with matrix multiplication, $\text{SO}(3)$ becomes a Lie group, the *Special Orthogonal* group of dimension three.

The Lie Algebra of $\text{SO}(3)$ is $\mathfrak{so}(3)$, i.e. the set of skew-symmetric matrices of dimension 3:

$$\mathfrak{so}(3) := \{ S \in \mathbb{R}^{3 \times 3} \mid S^T = -S \}. \quad (\text{A.21})$$

In particular any element $S \in \mathfrak{so}(3)$ has the following structure:

$$S = \begin{bmatrix} 0 & s_z & -s_y \\ -s_z & 0 & s_x \\ s_y & -s_x & 0 \end{bmatrix}. \quad (\text{A.22})$$

Consequently, an alternative representation for S is the $s \in \mathbb{R}^3$, defined as:

$$s = S^\vee = \begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix}. \quad (\text{A.23})$$

We will call s the representation of the element $S \in \mathfrak{so}(3)$ as a *vector* in \mathbb{R}^3 , with $^\vee$ the operator to map an element of $\mathfrak{so}(3)$ to the corresponding vector in \mathbb{R}^3 . The representation of $\mathfrak{so}(3)$ as \mathbb{R}^3 simplifies the representation of the covector in $\mathfrak{so}^*(3)$ and all the adjoint actions presented in Definitions A.12, A.13, A.14 and A.15. In particular, if the element of $\mathfrak{so}(3)$ is represented as a vector, then also the element of $\mathfrak{so}^*(3)$ can be represented as 3D vector, with the application of the element $\mathfrak{so}^*(3)$ to the element $v^\wedge \in \mathfrak{so}(3)$ is simply the vector dot product:

$$\langle f, v \rangle = f^T v. \quad (\text{A.24})$$

Furthermore, all the adjoint actions on $\mathfrak{so}(3)$ and $\mathfrak{so}^*(3)$ can be represented as 3×3 matrices. In particular for $R \in \text{SO}(3)$, $v, u \in \mathbb{R}^3 \approx \mathfrak{so}(3)$ and we have:

$$\text{Ad}_R = R, \quad (\text{A.25})$$

$$\text{Ad}_R^* = R^T, \quad (\text{A.26})$$

$$\text{ad}_v = v^\wedge, \quad (\text{A.27})$$

$$\text{ad}_v^* = (v^\wedge)^T = -v^\wedge. \quad (\text{A.28})$$

Homogeneous transformation matrices

The set $\text{SE}(3)$ is defined as

$$\text{SE}(3) := \left\{ \begin{bmatrix} R & p \\ 0_{1 \times 3} & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \mid R \in \text{SO}(3), p \in \mathbb{R}^3 \right\}. \quad (\text{A.29})$$

When endowed with matrix multiplication, it becomes the *Special Euclidean* group of dimension three, a Lie group that can be used to represent rigid transformations and their composition in the 3D space.

The Lie Algebra of $\text{SE}(3)$ is $\mathfrak{se}(3)$, the set of the matrices defined as following:

$$\mathfrak{se}(3) := \left\{ \begin{bmatrix} \Omega & v \\ 0_{1 \times 3} & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \mid \Omega \in \mathfrak{so}(3), v \in \mathbb{R}^3 \right\}. \quad (\text{A.30})$$

As in the case of $\mathfrak{so}(3)$, we can also identify $\mathfrak{se}(3)$ with \mathbb{R}^6 using the following mapping:

$$v = \begin{pmatrix} \Omega & v \\ 0_{1 \times 3} & 0 \end{pmatrix}^\vee = \begin{bmatrix} v \\ \Omega^\vee \end{bmatrix}. \quad (\text{A.31})$$

Using this 6D vector representation, the adjoint action assume the form of 4×4 real matrices, in particular:

$$\text{Ad}_H = \begin{bmatrix} R & o^\wedge R \\ 0_{3 \times 3} & R \end{bmatrix} = X, \quad (\text{A.32})$$

$$\text{Ad}_H^* = X^T = \begin{bmatrix} R^T & 0_{3 \times 3} \\ -R^T o^\wedge R & R^T \end{bmatrix}, \quad (\text{A.33})$$

$$\text{ad}_v = \begin{bmatrix} \omega^\vee & v^\vee \\ 0_{3 \times 3} & \omega^\vee \end{bmatrix} = v \times, \quad (\text{A.34})$$

$$\text{ad}_v^* = (v \times)^T = \begin{bmatrix} -\omega^\vee & 0_{3 \times 3} \\ -v^\vee & -\omega^\vee \end{bmatrix} \quad (\text{A.35})$$

where the 6D cross product $v \times$ is defined in (2.37).

Real Vector Spaces endowed with vector sum

The vector space or real vectors of dimension n can be seen as a Matrix Lie Group. In particular the vector $u \in \mathbb{R}^m$ can be mapped to a specific group of matrices in $\mathbb{R}^{(n+1) \times (n+1)}$:

$$\begin{bmatrix} 1_n & u \\ 0_{1 \times n} & 1 \end{bmatrix} \quad (\text{A.36})$$

It is trivial to verify that a group multiplication in this space is equivalent to a sum in the vector space:

$$\begin{bmatrix} 1_n & u \\ 0_{1 \times n} & 1 \end{bmatrix} \begin{bmatrix} 1_n & v \\ 0_{1 \times n} & 1 \end{bmatrix} = \begin{bmatrix} 1_n & u + v \\ 0_{1 \times n} & 1 \end{bmatrix} \quad (\text{A.37})$$

Thanks to such a mapping, we can consider the real vector space \mathbb{R}^n to be *equivalent* to a matrix Lie group.

The Lie Algebra of such a matrix Lie group is the set of matrices of format:

$$\begin{bmatrix} 0_{n \times n} & u \\ 0_{1 \times n} & 0 \end{bmatrix}. \quad (\text{A.38})$$

A convenient vector representation of such a matrix is obviously the u vector itself.

Using this vector representation, the adjoint actions are defined as matrices in $\mathbb{R}^{n \times n}$:

$$\text{Ad}_v = 1_n, \quad (\text{A.39})$$

$$\text{Ad}_v^* = 1_n, \quad (\text{A.40})$$

$$\text{ad}_u = 0_{n \times n}, \quad (\text{A.41})$$

$$\text{ad}_u^* = 0_{n \times n}. \quad (\text{A.42})$$

A.2 Multibody Dynamics Notation and its connection to Lie Groups

In this section, the connection between the notation introduced in Chapter 2 and the concepts of matrix Lie groups reviewed in this appendix.

A.2.1 Frame Pose and 6D Velocity

Given two frames A and B , their relative pose can be represented by the homogeneous transform ${}^A H_B$, that is an element of the matrix Lie group SE(3). Given a trajectory of the rigid body ${}^A H_B(t) : \mathbb{R} \mapsto \text{SE}(3)$, the time derivative ${}^A \dot{H}_B(t)$ belongs to the tangent space $T_{{}^A H_B} \text{SE}(3)$. The *left-trivialized* velocity ${}^B v_{A,B}^\wedge$ and the *right-trivialized* ${}^A v_{A,B}^\wedge$ both belong to the Lie algebra of SE(3), i.e. $\mathfrak{se}(3)$.

A.2.2 Cross Product on \mathbb{R}^6

In the language of Lie groups, the \mathbb{R}^6 cross product $v \times$ introduced in (2.37) is nothing else than the matrix representation of the adjoint action of \mathbb{R}^6 on itself, indicated with ad, when thinking at \mathbb{R}^6 as the Lie algebra *induced* by the Lie algebra homeomorphism (2.7) between \mathbb{R}^6 and $\mathfrak{se}(3)$. Defining $g = {}^A H_B \in \text{SE}(3)$, (2.36) is then usually written as (cf. [Marsden and Ratiu, 1999, Chapter 9, equation (9.3.4)])

$$\frac{d}{dt} \text{Ad}_g = \text{Ad}_g \text{ad}_{g^{-1}\dot{g}}, \quad (\text{A.43})$$

where $\text{Ad}_g = {}^A X_B$ and $\text{ad}_{g^{-1}\dot{g}} = {}^A v_{A,B} \times$, with $g^{-1}\dot{g} = {}^B v_{A,B}$. This notation is used in the robotic literature in, e.g., [Garofalo et al., 2013] and [Park et al., 1995]. This connection with Lie group theory allows to obtain immediately useful algebraic equalities such as, e.g., the identity $(v \times w)^\wedge = v^\wedge w^\wedge - w^\wedge v^\wedge =: [v^\wedge, w^\wedge]$, valid for arbitrary vectors v and $w \in \mathbb{R}^6$, that derives

from the fact that the adjoint operator ad is nothing else than the matrix commutator $[\cdot, \cdot]$ when using the matrix representations (v^\wedge and w^\wedge) of the Lie algebra elements.

A.2.3 The dual cross product on \mathbb{R}^6 ($\bar{\times}^*$)

In the language of matrix Lie groups, the dual space of $\mathfrak{se}(3)$ (i.e., the space of linear applications from $\mathfrak{se}(3)$ to \mathbb{R}) is indicated with $\mathfrak{se}(3)^*$ and is the space where 6D forces belong (as opposed to $\mathfrak{se}(3)$ where 6D velocity belong). In terms of Lie group theory, the 6D force coordinate trasformation $_A X^B$ is written

$$_A X^B = \text{Ad}_{g^{-1}}^* \quad (\text{A.44})$$

with $g = {}^A H_B \in \text{SE}(3)$. Recall that $\text{Ad}_g = {}^A X_B$ and $\text{Ad}_{g^{-1}} = {}^B X_A$. Then, posing $\xi^\wedge = {}^B v_{A,B}^\wedge \in \mathfrak{se}(3)$, one sees that

$${}^B v_{A,B} \bar{\times}^* = - \text{ad}_\xi^*. \quad (\text{A.45})$$

Once again, note how the symbol $\bar{\times}^*$ appearing in (2.50) has been explicitly chosen to remind the fact that (2.50) is obtained from the product (\times) given in (2.37), by computing its adjoint (*) and changing its sign (-). Finally, (2.49) is simply

$$\frac{d}{dt} \text{Ad}_{g^{-1}}^* = - \text{Ad}_{g^{-1}}^* \text{ad}_\xi^* \quad (\text{A.46})$$

for $\dot{g} = g\xi$, with $g = {}^A H_B$ and $\xi = {}^B v_{A,B}^\wedge$.

A.3 Euler-Poincaré Equations and Rigid Body Dynamics

The Euler-Poincaré Equations are the generalization of the Euler-Lagrange equations to a system whose configuration space is a Lie Group.

A.3.1 Euler-Poincaré Equations

Theorem A.1. *Let G be a Matrix Lie group and let $L : TG \mapsto \mathbb{R}$ be a Lagrangian function. Let $l : G \times \mathfrak{g} \mapsto \mathbb{R}$ be the left-trivialization of the Lagrangian L , that is defined $l(g, \xi) := L(g, g\xi)$. Then, the variational principle*

$$\delta \int_0^T L(g, \dot{g}) dt = 0 \quad (\text{A.47})$$

with variations δq with fixed end points is equivalent to the Euler-Poincarè equations

$$\frac{d}{dt} \frac{\partial l}{\partial \xi} = \text{ad}_\xi^* \frac{\partial l}{\partial \xi} + g^{-1} \frac{\partial l}{\partial g} \quad (\text{A.48})$$

with reconstruction equation $\dot{g} = g\xi$ and where $g^{-1} \frac{\partial l}{\partial g}$ is the left-trivialization of $\frac{\partial l}{\partial g}$, as defined in Lemma A.3.

Proof. To prove the theorem we first transform (A.47) on a variational equation expressed with respect to l . We express the variation $\delta\xi$ as a function of variations $\delta g = \frac{dg}{d\epsilon} \Big|_{\epsilon=0}$ and $\delta\dot{g} = \frac{d\dot{g}}{d\epsilon} \Big|_{\epsilon=0}$, obtaining

$$\begin{aligned} \delta\xi &= \frac{d}{d\epsilon} (g^{-1}\dot{g}) \Big|_{\epsilon=0} = - (g^{-1}\delta gg^{-1}) \dot{g} + g^{-1}\delta\dot{g} \\ &= -\eta\xi + g^{-1}\delta\dot{g} \end{aligned}$$

Computing the time derivative of $\eta = g^{-1}\delta g$, recalling that $\frac{d}{dt}\delta g = \delta\dot{g}$, we get

$$\begin{aligned} \dot{\eta} &= \frac{d}{dt} (g^{-1}\delta g) = - (g^{-1}\dot{g}g^{-1}) \delta g + g^{-1}\delta\dot{g} \\ &= -\xi\eta + g^{-1}\delta\dot{g}. \end{aligned}$$

We therefore obtain, combining the two equations above, that

$$\delta\xi = \xi\eta - \eta\xi + \dot{\eta} = \text{ad}_\xi \eta + \dot{\eta}. \quad (\text{A.49})$$

The variational principle (A.47) is equivalent to

$$\delta \int_0^T l(g, \xi) dt = 0$$

with variations of the form (A.49) fixed at the end points. Expliciting, the above reads

$$\begin{aligned} &\int_0^T \left(\frac{\partial l}{\partial g} \cdot \delta g + \frac{\partial l}{\partial \xi} \cdot \delta\xi \right) dt = \\ &\int_0^T \left\langle \frac{\partial l}{\partial g}, g\eta \right\rangle + \left\langle \frac{\partial l}{\partial \xi}, \text{ad}_\xi \eta + \dot{\eta} \right\rangle dt = \\ &\int_0^T \left\langle g^{-1} \frac{\partial l}{\partial g}, \eta \right\rangle + \left\langle \text{ad}_\xi^* \frac{\partial l}{\partial \xi}, \eta \right\rangle - \left\langle \frac{d}{dt} \frac{\partial l}{\partial \xi}, \eta \right\rangle dt = \\ &\int_0^T \left\langle g^{-1} \frac{\partial l}{\partial g} + \text{ad}_\xi^* \frac{\partial l}{\partial \xi} - \frac{d}{dt} \frac{\partial l}{\partial \xi}, \eta \right\rangle dt = 0, \quad (\text{A.50}) \end{aligned}$$

where we used integration by parts to go from the second to the third step, recalling that η is a variation with fixed end points. Given that δg (and hence η) is arbitrary, we finally get

$$\frac{d}{dt} \frac{\partial l}{\partial \xi} - \text{ad}_\xi^* \frac{\partial l}{\partial \xi} - g^{-1} \frac{\partial l}{\partial g} = 0 \quad (\text{A.51})$$

with $\dot{g} = g\xi$. \square

A.3.2 Rigid Body Dynamics

For obtaining the Rigid Body equation of motions we can write the Euler-Poincaré equations with $g = H, \xi = v$, and the left-trivialized lagrangian is given in Proposition 2.1.

Expliciting the different terms of the Euler-Poincaré equations, we have:

$$-\text{ad}_v^* = v \bar{x}^*, \quad (\text{A.52a})$$

$$\frac{\partial l}{\partial v} = Mv, \quad (\text{A.52b})$$

$$\frac{d}{dt} \frac{\partial l}{\partial v} = M\dot{v}. \quad (\text{A.52c})$$

The first equivalence is consequence from the definition of $v \bar{x}^*$, while the other two come from the fact that M is constant and hence independent from v .

For the term that depends on the potential energy $-H^{-1} \frac{\partial l}{\partial H} = -H^{-1} \frac{\partial U}{\partial H}$, from the definition of left-trivialization in Lemma A.3 and given $\eta = \begin{bmatrix} \eta_l \\ \eta_a \end{bmatrix} \in \mathbb{R}^6$, $\eta^\wedge \in \mathfrak{se}(3)$ we have:

$$\begin{aligned} & \left\langle -H^{-1} \frac{\partial U(H)}{\partial H}, \eta^\wedge \right\rangle = \\ &= \left\langle -\frac{\partial U(H)}{\partial H}, H\eta \right\rangle = \\ &= -U(H\eta^\wedge) = \\ &= m \begin{bmatrix} g \\ 0 \end{bmatrix}^T \left(\begin{bmatrix} R\eta_a^\wedge & R\eta_l \\ 0_{3 \times 1} & 0 \end{bmatrix} \begin{bmatrix} c \\ 1 \end{bmatrix} \right) = \\ &= m \begin{bmatrix} g \\ 0 \end{bmatrix}^T \left(\begin{bmatrix} R & -Rc^\wedge \\ 0_{1 \times 3} & 0_{1 \times 3} \end{bmatrix} \right) \begin{bmatrix} \eta_l \\ \eta_a \end{bmatrix} = \\ &= (R^T g)^T [m1_3 \quad -mc^\wedge] \eta. \end{aligned}$$

From the last equation, we can write $-H^{-1} \frac{\partial l}{\partial H}$ in vector form as:

$$-H^{-1} \frac{\partial l}{\partial H} = \begin{bmatrix} m1_3 \\ mc^\wedge \end{bmatrix} R^T g = \mathbb{M} \begin{bmatrix} R^T g \\ 0_{3 \times 1} \end{bmatrix}. \quad (\text{A.53})$$

Combining equations and (A.53), we obtain equation.

A.4 Hamel Equations and Multi Body Dynamics

A.4.1 Hamel Equations

If the configuration space of a mechanical system is a combination of a Lie Group and of a vector space, then the Euler-Poincaré can be specialized in the Hamel equations.

Theorem A.2. *Let Q be a Matrix Lie group defined as the direct product of another Matrix Lie Group G and of a real vector space \mathbb{R}^n , i.e. $Q = G \times \mathbb{R}^n$. Furthermore let $L : G \times \mathbb{R}^n \times TG \times \mathbb{R}^n \mapsto \mathbb{R}$ be a Lagrangian function. Let $l : G \times \mathbb{R}^n \times \mathfrak{g} \times \mathbb{R}^n \mapsto \mathbb{R}$ the left-trivialization (relative just to G) of the Lagrangian L , defined as $l(g, s, \xi, \dot{s}) := L(g, s, g\xi, \dot{s})$. Then, the variational principle*

$$\delta \int_0^T l(g, s, \dot{g}, \dot{s}) dt = 0 \quad (\text{A.54})$$

with variations $(\delta g, \delta s)$ with fixed end points is equivalent to the Hamel equations

$$\frac{d}{dt} \frac{\partial l}{\partial \xi} - \text{ad}_\xi^* \frac{\partial l}{\partial \xi} - g^{-1} \frac{\partial l}{\partial g} = 0 \quad (\text{A.55a})$$

$$\frac{d}{dt} \frac{\partial l}{\partial \dot{s}} - \frac{\partial l}{\partial s} = 0 \quad (\text{A.55b})$$

with reconstruction equation $\dot{g} = g\xi^\wedge$ and where $g^{-1} \frac{\partial l}{\partial g}$ is the left-trivialization of $\frac{\partial l}{\partial g}$, as defined in Lemma A.3.

Proof. The variational principle (A.54) is equivalent to

$$\delta \int_0^T l(g, s, \xi, \dot{s}) dt = 0$$

with variations of the base part of the form (A.49) fixed at the end points. Expliciting, the above reads:

$$\int_0^T \left(\frac{\partial l}{\partial g} \cdot \delta g + \frac{\partial l}{\partial \xi} \cdot \delta \xi + \frac{\partial l}{\partial s} \cdot \delta s + \frac{\partial l}{\partial \dot{s}} \cdot \delta \dot{s} \right) dt.$$

It is possible to separate the first two term from the last two terms. Proceeding for the first two terms as in (A.50), and similarly for the last two terms we can (A.54) as:

$$\int_0^T \left\langle g^{-1} \frac{\partial l}{\partial g} + \text{ad}_\xi^* \frac{\partial l}{\partial \xi} - \frac{d}{dt} \frac{\partial l}{\partial \xi}, \eta \right\rangle + \left\langle \frac{\partial l}{\partial s} - \frac{d}{dt} \frac{\partial l}{\partial s}, \delta s \right\rangle dt = 0.$$

Given that η and δs are arbitrary, we get that both (A.55a) and (A.55b) need to be satisfied. \square

A.4.2 Multi Body Dynamics

The Hamel equations (A.55) can be written for multibody dynamics with $g = H, \xi = v$ and using the lagrangian defined in (3.48). First, we can combine the first terms of the base and shape equations in a single term:

$$\begin{aligned} \left[\begin{array}{c} \frac{d}{dt} \frac{\partial l}{\partial v} \\ \frac{d}{dt} \frac{\partial l}{\partial \dot{s}} \end{array} \right] &= \frac{d}{dt} \frac{\partial l}{\partial v} = \\ &= \frac{d}{dt} \frac{\partial (M(s)v)}{\partial v} = \\ &= M(s)\dot{v} + \left(\frac{d}{dt} M(s) \right) v. \end{aligned}$$

The second term and third terms of the base equation of motion, recalling that $-\text{ad}_v^* = v\bar{x}^*$ and the equivalent derivations in Subsection A.3.2 are:

$$-\text{ad}_v^* \frac{\partial l}{\partial v} = v\bar{x}^* (\mathbb{M}(s)v + F(s)\dot{s}), \quad (\text{A.56})$$

$$-H^{-1} \frac{\partial l}{\partial H} = - \begin{bmatrix} m1_3 \\ mc^\wedge(s) \end{bmatrix} R^T g = -\mathbb{M}(s) \begin{bmatrix} R^T g \\ 0_{3 \times 1} \end{bmatrix}. \quad (\text{A.57})$$

The second term of the shape Hamel equations is given as:

$$-\frac{\partial l}{\partial s} = -\frac{\partial}{\partial s} \left(\frac{1}{2} \nu^T M(s) \nu \right) - \frac{\partial c}{\partial s} R^T g. \quad (\text{A.58})$$

Combining the different terms, we obtain:

$$M(s)\dot{v} + \left(\frac{d}{dt} M(s) \right) v - \begin{bmatrix} v\bar{x}^* (\mathbb{M}(s)v + F(s)\dot{s}) \\ 0_{n \times 1} \end{bmatrix} + \quad (\text{A.59})$$

$$- \begin{bmatrix} 0_{6 \times 1} \\ \frac{\partial}{\partial s} \left(\frac{1}{2} \nu^T M(s) \nu \right) \end{bmatrix} - \begin{bmatrix} m1_3 \\ mc^\wedge(s) \\ \frac{\partial c}{\partial s} \end{bmatrix} R^T g = 0_{(n+6) \times 1} \quad (\text{A.60})$$

The first term is the only one that depends on the acceleration of the system, while the last one is the only one that depends on the gravitation acceleration.

From straightforward algebraic manipulations we then have that:

$$\left[\frac{\partial}{\partial s} \left(\frac{d}{dt} M(s) \right) \nu \right] = \sum_L J_L^T \left[(\mathbf{v}_L \bar{\times} {}^* L \mathbb{M}_L + {}^* L \mathbb{M}_L \mathbf{v}_L \times) J_L + {}^* L \mathbb{M}_L J_L \right] \quad (\text{A.61})$$

$$\begin{bmatrix} m\mathbf{1}_3 \\ mc^\wedge(s) \\ \frac{\partial c}{\partial s} \end{bmatrix} R^T g = M(s) \begin{bmatrix} R^T g \\ 0_{3+n_J} \end{bmatrix}. \quad (\text{A.62})$$