Fondazione Istituto Italiano di Tecnologia
University of Genova

PhD Program in Bioengineering and Robotics

Improving Dynamics Estimations and Low Level Torque Control Through Inertial Sensing

by

Nuno Guedelha

Thesis submitted for the degree of *Doctor of Philosophy* (30° cycle)

December 2018

Dott. Francesco Nori	Supervisor
Dott. Daniele Pucci	Supervisor
Dott. Silvio Traversaro	Co-Supervisor
Prof. Giorgio Cannata	Head of the PhD program

PhD Jury, External Examiners, Thesis Reviewers:

Olivier Stasse, *Research Director* *(LAAS-CNRS Toulouse)*	Jury member
Matteo Fumagalli, *Associate Professor* *(Aalborg University Copenhagen)*	Jury member & Reviewer
Andrea Del Prete, *Tenure-track Assistant Professor* *(University of Trento)*	Jury member
Juho Vihonen, *Docent, Doctor of Science* *(Tampere University of Technology)*	External examiner & Reviewer

Dibris

Department of Informatics, Bioengineering, Robotics and Systems Engineering

Abstract

In 1996, professors J. Edward Colgate and Michael Peshkin invented the cobots as robotic equipment safe enough for interacting with human workers. Twenty years later, collaborative robots are highly demanded in the packaging industry, and have already been massively adopted by companies facing issues for meeting customer demands. Meantime, cobots are still making they way into environments where value-added tasks require more complex interactions between robots and human operators. For other applications like a rescue mission in a disaster scenario, robots have to deal with highly dynamic environments and uneven terrains. All these applications require robust, fine and fast control of the interaction forces, specially in the case of locomotion on uneven terrains in an environment where unexpected events can occur. Such interaction forces can only be modulated through the control of joint internal torques in the case of under-actuated systems which is typically the case of mobile robots. For that purpose, an efficient low level joint torque control is one of the critical requirements, and motivated the research presented here. This thesis addresses a thorough model analysis of a typical low level joint actuation sub-system, powered by a Brushless DC motor and suitable for torque control. It then proposes procedure improvements in the identification of model parameters, particularly challenging in the case of coupled joints, in view of improving their control. Along with these procedures, it proposes novel methods for the calibration of inertial sensors, as well as the use of such sensors in the estimation of joint torques.

Acknowledgements

For starting, I apologize to all those I forgot to mention here, specially because every person I met in IIT contributed in some way to make these three years a positive experience.

I would like to thank my first supervisor, Francesco Nori, for giving me the wonderful opportunity to share these three years of PhD with you all, and for his support during my first years at IIT. At the beginning of my PhD, my direct scientific mentor, and my desk neighbor Naveen Kuppuswamy, was also a valuable companion and gave me great support in the first year of my PhD. I really appreciated the mentoring, the availability and kindness, for the ”short” time we shared at IIT. The next person I want to thank is Silvio Traversaro, my most direct scientific mentor after Naveen's departure. By his passion for his work, he encouraged me to always learn more, be curious, and look for challenges. I am grateful for our discussions on specific problems, from which I would always learn something. Even when very busy, he was always available, and that encouraged me to always try to help my colleagues as much as possible. Daniele Pucci became my new supervisor and has ever since been pushing me forward, and I thank him for that, as well as for his sense of humor and our few private jokes. I thank Francesco Romano for his kindness, for his precious support when I met him in Toulouse and was looking for information on Genova, and also for being, with Silvio, one of the guys helping me with the bugs on Mac!

I would like to thank all the team who maintained the iCub firmware, hardware, and kept fixing the robot while we kept breaking it, and for that a special thanks for Julien Jenvrin.. merci Julien! Thank you all for your time, dedication and feedback.

Last but not least, I will never thank enough my wife Diana, who gave me unconditional support, and all her patience, without which I couldn't succeed.

Contents

Chapter 1

Introduction

1.1 Motivation

While classical industrial robots operate in structured and known environments, there is a growing need for robots capable of interacting with humans in shared environments or operating in unstructured and open spaces.

Among human-robot interaction envisioned use cases, we count the cooperative tasks, where we find co-robots (from collaborative robots), or "cobots", working hand in hand with humans. "cobots" were invented in 1996 by J. Edward Colgate and Michael Peshkin, professors at Northwestern University based in Illinois, United States, after an initiative led by General Motors to make robots or robotic equipment safe enough for interacting with people. Presently, human-robot cooperation is highly demanded and adopted in the packaging industry, leading to increased productivity, but is still making its way into assembly lines, typically in the automotive industry. In the same line of interaction, in a relatively near future, more and more cobots will assist people in the office or at home. Making that cooperation possible rises some challenges, among which we find the implementation of collision avoidance strategies for handling the robot operational space, but also the fine control of interaction forces between robotic and human agents during co-manipulation of loads.

Robotics is also investing motion rehabilitation with high-tech adaptive prosthetic devices or exoskeletons. An exoskeleton can typically be an anthropomorphic manipulator reproducing the kinematic chain of a human limb that requires the correction of a postural defect or some mobility strength assistance. In these cases, the harnessing of interaction forces is even more critical for the safety of the patient and the efficiency of the treatment.

In other robotic applications, robots have to operate in unstructured, and eventually, open spaces. In such cases, the robot has to cope with unpredictable and dynamic environments, or even navigate across uneven terrains, like for instance: in a rescue and search mission as part of disaster response measures; in the surveillance of sites against intrusion; in undersea or space exploration.

These applications and operation conditions drive the research in mobile robotics towards safer, compliant systems, thus integrating enhanced sensing capabilities, and more robust control of interaction forces. As a matter of fact, interaction forces are propagated across the robot multi-body articulated system through its joints as cross joint forces and moments. Considering the joint as a kinematic constraint on the relative motion of two bodies [Featherstone, 2014, Chapter 3], the 6-D cross joint force/moment, also referred to as wrench, is the sum of an active wrench and a constraint wrench. The active wrench comes from the transmission damping and elasticity (harmonic drive), and the actuator on that joint, while the passive wrench is the wrench imposing the motion constraint. In other words, the active wrench lies in a subspace T_a aligned with the joint motion subspace S, satisfying $T_a^\top S = 1$, and the constraint wrench lies in the orthogonal subspace $S^\perp$. The active wrench can then be considered as a generalized vector mapped onto the subspace T_a. In the case of a 1-D (one dimension) rotary joint, the active wrench is the total wrench across the joint projected onto the joint axis, and the respective generalized wrench vector has a single dimension and is commonly referred to as the joint torque. Interaction forces can be modulated through the control of joint internal torques, and an efficient low level joint torque control is one of the critical requirements for generating compliant interaction forces. From those requirements we can derive the following: a good measurement of joint positions, velocities and accelerations, an accurate measurement or estimation and good tracking of joint torques, as well as a proper identification of low level joint actuation model parameters, i.e. the motor to joint feed forward parameters. Among these parameters we can find the motor torque model, the friction parameters specific to the motor, gearbox, or other components of the joint actuation system.

Most humanoid robots are equipped with brushless DC motors and zero backlash harmonic drives. These motors, under the adequate control, can deliver a linear voltage to torque response. On the other side, harmonic drives have some friction and elasticity inducing hysteresis in the torque and friction models. Some robots use tendon driven coupled joints which significantly increases the complexity in the transmission model and the uncertainties in the identification of the model parameters.

This thesis addresses a thorough model analysis of a typical low level joint actuation sub-system powered by a Brushless DC motor. It then proposes procedure improvements in the identification of model parameters, particularly challenging in the case of coupled joints, in view of improving their control. The identification process requires accurate joint torques measurement or estimation. Considering the high cost of joint torque sensors, it is worth improving and relying on the estimation of joint torques through inverse dynamics computations. For that purpose we present an improved and optimized algorithm using inertial sensors for estimating link linear and angular accelerations, which in turn can be directly used in the inverse dynamics computations and more accurate joint torques estimation. The efficacy of these computations is strongly dependent on accurate joint position and link inertial sensing. In that context, we propose novel methods for calibrating accelerometers and joint encoder offsets.

1.2 Research platform

For their versatility, high number of degrees of freedom (DoF), high potential efficiency in locomotion capabilities, humanoid robots are often the preferred category of mobile robots when it comes to operate in shared and highly dynamic spaces.

The platform we have used for our experiements is the child-sized humanoid robot “iCub”, developed by the *iCub Facility* at the *Italian Institute of Technology* as part of the European Project *RobotCub*, a fully open source and hardware project which goal was to study cognition through the implementation of a humanoid robot the size of a three and a half year old child: the “iCub” [Tsagarakis et al., 2006] [Tsagarakis et al., 2007]. Since its initial release in 2006, multiple upgrades and improvements, as part of “CoDyCo” among other European Projects, added new features and increased its size to a ten year old child, with stronger legs and the capability to walk. “CoDyCo” aimed at advancing the control concepts and cognitive understanding of robust, goal-directed whole-body motion interaction with multiple contacts, combining planning and compliance to deal with unpredictable events and contacts. Until now, the robot has been adopted by more than 20 laboratories worldwide. The version we have used for the experiments in this thesis is the latest standard version “iCub 2.5”, active since early 2017. We quickly list below its main characteristics relevant for our experiments, and they will be reviewed more in detail in each chapter.

The robot has 53 degrees-of-freedom (DoF) distributed as follows: seven for each arm—shoulder pitch/roll/yaw, elbow, wrist pronation/pitch/roll; six for each leg—hip pitch/roll/yaw, knee, ankle pitch/yaw; three for the torso—torso pitch/roll/yaw; three for the neck—neck pitch/roll/yaw; nine for each hand; and three for the eyes. The many DoF on the legs and torso turned to be very useful for the in-situ calibration of joint encoders and inertial sensors as they allowed to reach a wide range of joint positions and inertial sensor frames orientations required for those procedures.

In the thesis experiments and dissertation we only consider the main DoF: legs, arms and torso. These are actuated by motor groups composed of a Brushless DC motor (BLDC), an harmonic drive and high resolution encoders, suitable for torque control as they can provide a linear voltage to torque response.

The robot is also equipped with a wide range of other sensors: inertial sensors (gyroscopes, accelerometers and IMUs), link force-torque sensors and skin tactile sensors. These are relevant for low level identification of joint actuation model parameters and joint torque estimation.

1.3 Contribution and Thesis outline

In this section we draw the thesis outline through the main theoretical and experimental contributions converging to the targeted joint actuation model identification. Each chapter formulates the addressed problem, highlighting its main contribution with respect to the state of the art on that problem, then describes the methodology and presents the obtained experimental results.

1.3.1 Joint motor and friction model identification

Chapter 2 defines a model of joint low level actuation sub-system, applicable for the main iCub joints on the legs, arms and torso. It then defines a framework and methodology for identifying the model parameters in the case of single joints and coupled joints like the shoulders or the joints connecting the waist to the torso. The identification targets the motor voltage to torque model parameters and the joint state to friction model parameters. The main contributions of this chapter are: the identification of static friction parameters and their correlation with the hysteresis in the motor input voltage to torque characteristics; the improvement of the experiment procedure itself, avoiding the quantization issues affecting the driven motor input voltage Pulse Width Modulation (PWM) duty cycle; the improvement of the procedure specific to coupled joints in view of performing the model identification independently on each motor transmission chain, thus avoiding the coupling effects.

1.3.2 Accelerometers and Joint Encoders Offsets Calibration

Chapter 3 proposes two contributions regarding the identification of joint offsets in a kinematic chain using inertial sensors measurements. The first contribution is about simplifying the identification process by estimating in a single run the offsets of the whole kinematic chain, and doing so from a single batch of accelerometers data. The second contribution is about identifying in-situ the full calibration matrix and axes offsets of any accelerometer mounted on the kinematic chain, without knowing the sensor pose with respect to the link it is attached to. We will show how this allows the calibration of the accelerometers to be independent from the identification of the joint offsets, breaking a dependency loop. Additionally, we give a particular attention to the identification of the cross-axis gains and their dependency with respect to the drifting axes offsets, unlike other existing methods which assume the calibration matrix to have only diagonal terms.

1.3.3 Link Angular Acceleration Estimation

Joint torques can be measured by joint torque sensors or estimated in a framework using embedded force-torque sensors and inverse dynamics algorithms as done on the iCub humanoid robot and presented in [Traversaro, 2017, Chapter 4].

That framework can take directly link accelerations as described in [Traversaro, 2017, Chapter 4 section 4.4.2], referred to as the sensor-based net force-torque estimation, followed by the joint torque estimation. This approach reduces the errors otherwise accumulated when propagating the link accelerations through the composition of joint accelerations from the root to the leafs of a kinematic tree. The link angular accelerations are commonly estimated through the forward kinematics, from the joint accelerations and the model kinematic tree [Khalil and Dombre, 2004, Chapter 5] [Featherstone, 2014, Sections 2.2, 4.4].

Joint accelerations are commonly computed through double numerical differentiation of joint encoders or single numerical differentiation of gyroscope measurements, at the cost of additional filtering and resulting lag. Chapter 4 proposes a causal memoryless algorithm for estimating directly the link angular acceleration from data measured by accelerometers and gyroscopes attached to that same link. We can find in the literature similar solutions, but targeting joint accelerations and depending on the composition of these accelerations for estimating the link accelerations. Our method avoids any propagation of variables across the kinematic chain. On top of that, the method is easily scalable across any number off n accelerometers. We can say that the computational complexity scales linearly with the size of the problem, if we define that size as the number off accelerometers n. It is then common use to characterize the algorithm as having $O(n)$ complexity.

1.4 Publications

[1] N. Guedelha, N. Kuppuswamy, S. Traversaro, and F. Nori, "Self-calibration of joint offsets for humanoid robots using accelerometer measurements," in IEEE-Humanoids, 2016: 1233-1238.

[2] Gabriele Nava, Daniele Pucci, Nuno Guedelha, Silvio Traversaro, Francesco Romano, Stefano Dafarra, Francesco Nori: "Modeling and control of humanoid robots in dynamic environments: ICub balancing on a seesaw" in IEEE-Humanoids, 2017: 263-270.

[3] Stefano Dafarra, Gabriele Nava, Marie Charbonneau, Nuno Guedelha, Francisco Andrade, Silvio Traversaro, Luca Fiorio, Francesco Romano, Francesco Nori, Giorgio Metta, Daniele Pucci: "A Control Architecture with Online Predictive Planning for Position and Torque Controlled Walking of Humanoid Robots", in IEEE-IROS 2018

[4] N. Guedelha, S. Traversaro, and D. Pucci, "Identification of motor and friction parameters on coupled joints," in Intelligent Systems Conference (IntelliSys) 2019.

1.5 Technological outcome

The outcome of this research was:

- a conference paper publication on the 2016 IEEE-RAS International Conference on Humanoid Robots [Guedelha et al., 2016],
- a conference paper publication on the 2019 Intelligent Systems Conference (IntelliSys) [N. Guedelha and Pucci, 2019],

- the implementation of a calibration framework that would later support multiple sensor modalities like accelerometers, gyroscopes and IMUs. The framework was released as an open source project on Github [1],

- the friction estimation feature was integrated in the calibration framework *sensors-calib-inertial*. A *Doxygen* documentation of this framework is in progress and can be found in the Github repository [2],

- the link acceleration algorithm has been implemented on Matlab and a journal paper is under preparation for submission to IEEE-RAL.

I was also involved in analyzing the hardware configuration and interface of the link force-torque sensors used on iCub, as well as the test of the respective calibration tool [3], which was my contribution to the publication [3]. Among the future work goals is collaborating with the author of the mentioned tool in order to integrate the tool's features in the *sensors-calib-inertial* framework.

---

[1] `https://github.com/robotology-playground/sensors-calib-inertial`

[2] `https://github.com/robotology-playground/sensors-calib-inertial/wiki/Framework-documentation`

[3] `https://github.com/robotology-playground/insitu-ft-analysis`

Chapter 2

Joint Motor And Friction Model Identification

2.1 Introduction

Some tasks require highly dynamic motions like running, jumping or even walking on uneven terrains, and for that, the robot has to account for the dynamic properties of the low level joint actuation. In typical torque control architectures composed by two nested control loops, this concern lies within the inner, low level control loop, which guaranties that the desired torque computed by the outer loop is generated at the joint level as expected within a delay that doesn't compromise the stability of the controller. This chapter addresses the identification of the joint low level actuation model parameters which are critical factors for generating and tracking the desired joint torque.

The feed-forward control allows to anticipate the changes in the controller setpoint, which in the case of low level joint torque control, would be the desired joint output torque. At that point, the only errors left to be corrected by the feedback controller are the model errors, the sensors noise and the external disturbancies applied to the system. This improves the stability of the controlled system, and the ability to use a wider range of gains in the feedback control depending on the desired stiffness of the joints. This type of control design is one among other model based control designs which benefit from the accurate identification of the joint low level actuation feed-forward model.

The identification first targets single joints with one DoF, actuated by a single motor. Commonly, identifying the feed-forward parameters goes through the breakdown of the overall transfer function relating the motor input parameter to the joint output torques, by modeling the sub-elements of the actuation chain as well as their interaction. That chain is typically composed of a motor, a reduction drive for multiplying the torque, and a direct or cable driven connection to the joint shaft moving with the actuated link. In this study we will consider electrical joint actuation subsystems powered by a high performance brush-less Direct Current (DC) motor, and equipped with a zero backlash Harmonic Drive. It is actually the case for most of the humanoid robots 40cm tall or above, apart from those equipped with hydraulic actuators.

DC motors, or brush-less motors are typically driven by current or voltage PWM—Pulse Width Modulation—duty cycle. The generated torque is then multiplied in the Harmonic Drive, and transmitted to the load on the output shaft, directly or through a cable system. Some torque is wasted in the process, through friction, mostly in the torque conversion and transmission components. The output torque on the shaft is what we define as the joint torque. Given a parametric model describing the PWM to joint torque dynamic system, the identification process then consists in quantifying the parameters of that model. The resulting final dynamics of the joint, while tracking a desired trajectory or torque, also depends on the load applied on the shaft by the actuated link inertia and the external wrenches. That load appears as an external factor with respect to the analyzed system, and impacts the joint velocity and acceleration which are inputs to the identification process.

A commonly used simple model defines the generated motor torque as a linear function of the motor input voltage duty cycle, which will be detailed in Section 2.3.3.

A large range of models have been defined and studied for describing the friction dynamics, from static models like the Coulomb, the viscous and Stribeck models, to more complex ones, like the switching models (seven parameters model, Dahl model), or the most advanced dynamic models, like the LuGre model, first presented in [Canudas de Wit et al., 1995] [Canudas-de Wit, 1998], accounting also for stick-slip phases transitions and other properties inherent to the interaction of surfaces in contact [Armstrong-Hélouvry et al., 1994] [Olsson et al., 1998] [Van Geffen, 2009].

Previous works proposed improvements on joint torque and external forces estimation, while the joints were not moving [Stolt et al., 2015b] [Capurso et al., 2017], or were moving at very low velocities [Linderoth et al., 2013] [Stolt et al., 2015a], while performing a lead-through programming task. Lead-through programming is a way of programming a desired joint trajectory, in order to perform a given task, by manually guiding the robot joints while the motion sequence is recorded. In this type of task, the detection of very low external forces is required for allowing a smooth external manipulation of the end effector by a human operator. The challenge was to detect external forces lower than the static and Coulomb friction, that could be quite high in industrial robots. The presented methods accounted for the significant torque disturbancies due to static, Coulomb and viscous friction, while estimating the external forces without the use of force-torque sensors. In [Stolt et al., 2015a], the estimation framework integrates a friction compensation based on a simple Coulomb-viscous friction deterministic model, dependent on a Coulomb and a viscous friction coefficients, and velocity thresholds for smoothing the compensation. All these parameters were tuned manually. In [Linderoth et al., 2013] and [Stolt et al., 2015b] the friction is modeled as a uniformly distributed noise, combined to a Gaussian noise of zero mean, with a variance increasing with joint velocity. The uniformly distributed noise is bounded by a range dependent on the velocity and described as the "Coulomb viscous friction band". The band is defined by two shifted sigmoid functions added to a viscous friction term: the area between the two curves, significant only around zero velocity, models the uncertainty of the static friction; the defined band converges to a classical Coulomb-viscous friction model for increasing velocities. The model parameters then include the slope of the sigmoid functions around the zero velocity, the area between the curves, the Coulomb and viscous friction coefficients. Except for the Coulomb friction coefficient, which is averaged for slow velocities, the parameters are identified or tuned manually and few details on the procedure are given, which does not make the whole process fast and easily repeatable.

When analyzing mechanisms that have numerous rolling parts that generate friction, a spectral-based identification of the friction parameters can be performed [Popovic and Goldenberg, 1998]. This approach can identify the contribution of each moving part of the transmission train to the overall friction, accounting for position dependent torque variations due to those contributions [Popovic and Goldenberg, 1998] [Simpson et al., 2002]. In this case, the identified friction model depends on the velocity and position of the joint. The frequency signature of the friction source is considered constant, although their amplitude may vary.

A model-free approach, based on Gaussian networks, more precisely Radial Basis Function Networks (RBFN), can also be used for identifying the non linear terms accounting for uncertainties and stick-slip transitions occurring at low speeds [Hongliu Du and Nair, 1999]. The total friction is split in two linear components, the classical Coulomb and viscous friction terms, and one non-linear component, function of the joint velocity and the joint acceleration sign, more precisely the current and previous samples of the joint velocity. The RBFN identifier estimates the Coulomb constant and the non-linear function parameters, identifying a ten parameters model. But this technique proved efficient on a bench setup, composed of a DC motor with characteristics an order of magnitude higher than the motors used on iCub, which is a child size humanoid robot. The motor is directly connected to the load without a gearbox, which in the case of small or medium robots, is an important source of friction non-linearity.

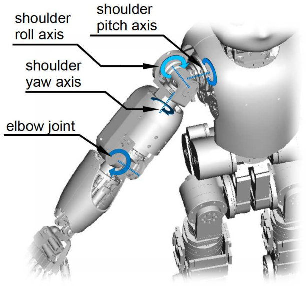
Model-based friction identification has been previously applied to the context of humanoid robots, using a simple Coulomb-viscous friction model in the feed-forward term of the low level torque control [Del Prete et al., 2016]. In the approach presented in this chapter, we extend the Coulomb-viscous static model, adding the Stribeck effect. The overall friction model is an affine function of the joint velocity, does not depend on the joint position and is memory-less, not taking into account stick-slip phase transitions. Further details will be given in Section 2.3.2. This is a first step in including the static friction in a full joint actuation model, and analyzing the its impact on the overall motor torque model identification. Particular attention is also given to how the motor torque model depends on the hardware and the motor control mode (trapeezoidal, sinusoidal, Field Oriented Control), hence the importance of considering the desired high level control scheme and performance at the very early stage of the robot design.

We then focus on the coupled joints case, where the friction effects from the multiple actuation chains are combined and their identification presents a significant challenge. In the multi-body articulated systems we consider, we can find robotic typical "wrist" like mechanisms, or spherical joints, where the three DoF have their respective rotation axis coinciding in a single point. In the case of the humanoid robot "iCub" (Metta et al. [2010]), such joints are coupled through a cable differential drive: the shoulder joints (pitch, roll and yaw) as shown in figure 2.1b and described in Parmiggiani et al. [2009]; the joints connecting the torso to the waist link (pitch, roll, yaw) as shown in figure 2.6 and described in Tsagarakis et al. [2006] and Tsagarakis et al. [2009]. In each of these couplings, the transmission between motors and joints is cable driven. This has many benefits like compactness, reduced inertia, while having the three axis pitch, roll and yaw to intersect in a single point, allowing a motion almost similar to a spherical joint, except that unlike the spherical joint, the coupled joint set has a singular configuration also known as "gimbal lock". The coupling also presents inconvenient properties like increased elasticity in the motor to joint transmission which results in hysteresis once combined with friction, adding complexity to the actuation model. On top of that, the dynamics of the coupled DoF are entangled, and it becomes more challenging to identify the joint actuation parameters in the usual joint DoF space.

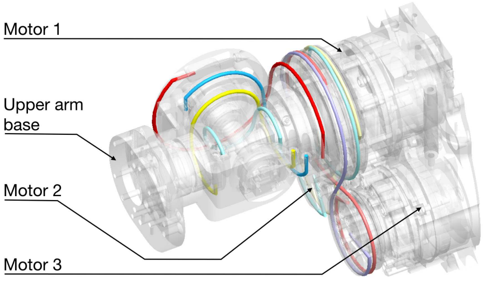
For controlling elastic coupled joints, a MIMO (Multiple Outputs Multiple Inputs) state feedback controller was introduced in [Le Tien et al., 2007] [Le-Tien and Albu-Schäffer, 2017]. The proposed control architecture benefits from joint position measurements both on the link and motor sides, and joint torque sensor measurements on the link side. It integrates a Lugre friction model in the feed forward component. The Coulomb and viscous friction and link load coefficients as well as the Lugre model parameters are identified in closed loop, through a non linear minimization of the expected to measured motor current error. But the measurements acquired near the velocity reversal points were removed from the optimization [Le-Tien and Albu-Schäffer, 2017, Section 5.2], while these transitions are what we are more concerned about in this chapter. In addition, the method relies on a good model of the joint stiffness and damping, which accounts for the elasticity only on the link side (the stiffness matrix is diagonal and depends solely on link side joint positions) [Le-Tien and Albu-Schäffer, 2017, Section 2.2]. Hence, the method does not account for the gearbox elasticity on the motor side.

In our approach, we account for the elasticity and friction on the motor side, i.e. the Harmonic Drive, and the identification is done in open loop. We propose a method for performing the model identification independently on each motor transmission chain, thus avoiding the coupling effects.

This chapter is organized as follows: we first present an overview of the joint actuation architecture implemented on the major joints of the humanoid robot iCub (the platform we used for performing our identification tests); after describing the model of each component composing one typical joint actuation sub-system, we build the model of the full transmission chain and pose the Newton-Euler equations describing its dynamics; based on those equations, we define the identification framework and experimental methodology, highlighting the choice of an open loop identification, which provides improved results compared to a closed loop identification; we then address the case of the coupled joints and present the experiment results.



(a)



(b)

Fig. 2.1 CAD view of the iCub right arm with the three coupled shoulder joints (a) and CAD view of the shoudler joints actuated by the differential drive and three motors. Source: [Parmiggiani et al., 2012, Fig.4 & 5]

2.2 Joint Actuation and Sensors Architecture

The methodology presented in this chapter for identifying the motor torque and joint friction model parameters requires the capability to perform accurate joint position control and motor direct PWM (or current) control. It also requires the feedback on joint torque, motor and joint angular position, velocity and acceleration. The system we have used for our experiments is the humanoid robot “iCub”, with its 53 degrees of freedom as shown in the iCub CAD representation and the kinematic structure depicted in figure 2.2: seven for each arm—shoulder pitch/roll/yaw, elbow, wrist pronation/pitch/roll; nine for each hand, six for each leg—hip pitch/roll/yaw, knee, ankle pitch/yaw; three for the torso—torso pitch/roll/yaw; three for the neck—neck pitch/roll/yaw; and three for the eyes. The robot is also endowed with a wide range of other sensors: high-resolution joint encoders, inertial sensors, link force-torque sensors and skin tactile sensors.

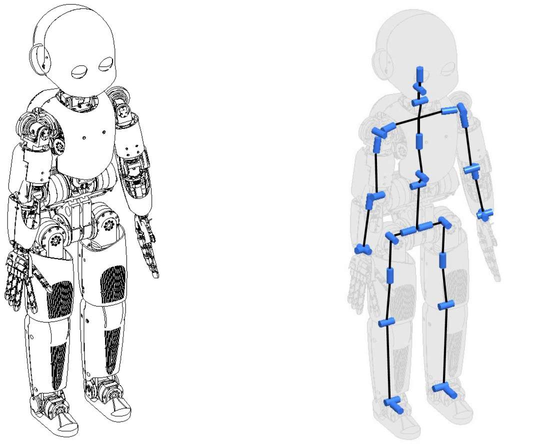

(a) (b)

Fig. 2.2 The iCub kinematic structure. Figure (a) illustrates the iCub CAD representation, and figure (b) its kinematic structure (eyes and hands have been omitted for visual clarity). Source: [Parmiggiani et al., 2012, Fig.2]

2.2.1 Motor control and motor/joint motion sensing

Note. ***Absolute and relative (incremental) encoders:*** *Relative encoders provide the linear or angular displacement measured since the instant they were turned on and the sensing activated. Absolute encoders provide the displacement with respect to a chosen position previously programmed or set by design.*

The motor group: The major joints (arms, legs, torso) are actuated by motor groups composed of a *Moog C2900584* brushless motor and an harmonic reduction drive *Harmoni Drive* in a very compact configuration (figure 2.3). For the joint position sensing, the group contains a 12-bit resolution, absolute hall-effect encoder (AS5045 microchip from Austria Microsystems [Microsystems, 2013] or alternatively an AEA absolute encoder microship). It is located at the joint side, after the harmonic drive, such that it measures the joint child link's rotation angle and not the motor's [Parmiggiani et al., 2012, section 4.3]. Because of space constraints, the encoder was mounted at the rear end of the motor shaft (figure 2.3), and

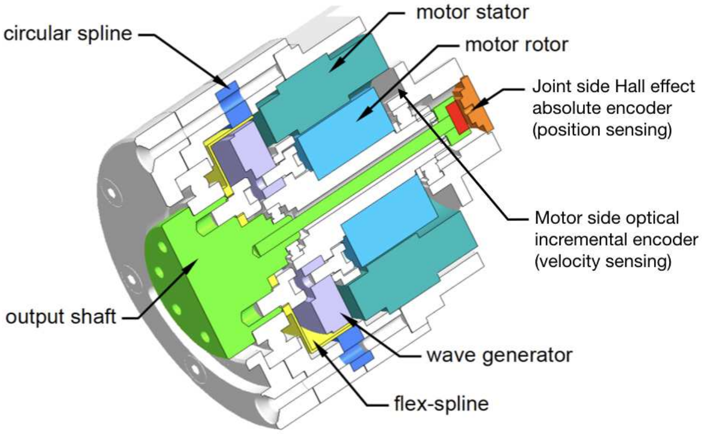


Fig. 2.3 Motor group cross section, showing the brushless motor, the harmonic drive and the angle position encoders. Source: [Parmiggiani et al., 2012, Fig.3]

connected to the output shaft through a stem across the motor's hollow shaft.

Joint position and rotor synchronization: The AS5045 encoder is a contact-less magnetic position sensor providing an accurate angular absolute position measurement over a full turn of 360°. A magnet, diametrically magnetized, is placed above and close to the encoder ship, and rotates with the rotor shaft while the encoder is fixed to the motor housing. An array of Hall effect sensors detects the rotating magnetic field and the magnet angular position is estimated with a typical delay of 384 µs, a resolution of 0.0879° (4096 positions per revolution), a repeatability (Transition Noise) of ±0.03° and an accuracy of ±0.5° (for more details refer to the appendix A).

The joint position is also used for synchronizing the rotating magnetic field (which creates the motor torque) with the rotor position, as explained in more detail in the next section 2.3: the system uses an advanced Field Oriented Control strategy which tracks the rotating magnetic field to keep its position shifted with respect to the rotor's poles by a quarter of electric

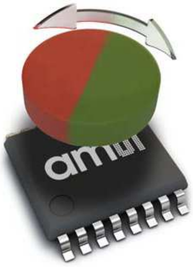


Fig. 2.4 Typical arrangement for placing the rotating magnet close to the sensing surface of the AS5045 chipset. Source: Microsystems [2017]

cycle (cf section 2.3). The resolution provided by the AS5045 encoder turned to be insufficient for such purpose, neither for measuring the velocity of dynamic motions. Indeed, the smallest detectable velocity obtained through pulse counting at 1 kHz sampling rate was $\Delta q/T_s = 0.088^\circ/0.001\text{s} = 88^\circ\,\text{s}^{-1}$, T_s being the encoder sampling period. For improving the velocity measurement accuracy, an optical 13-bit incremental encoder was mounted inside the motor housing as shown in figure 2.3. As the encoder resolution is combined with the reduction drive ratio of 1:100, we get the equivalent resolution at joint level of $360/(2^{13} \times 100) = 0.0004^\circ$, and the smallest detectable velocity becomes $\Delta q/T_s = 0.0004^\circ/0.001\text{s} = 0.4^\circ\,\text{s}^{-1}$ [Tsagarakis et al., 2009, section A].

We can see in appendix A a more detailed description of these two types of rotary encoders—incremental or absolute encoders—with their respective properties like the resolution, the accuracy and how repetitive the measurements are.

Motor driver and low level torque control boards On the robot iCub, the control architecture is designed as two nested control loops: the high level control loop runs in the robot central unit or on a remote computer connected to that unit through a distributed middle-ware called YARP (Yet Another Robot Protocol), over wifi or wired IP network; the high level controller sends through an Ethernet bus the desired joint torque to the inner loop controller which is in charge of obtaining and tracking the desired torque at the joint output shaft. This controller, running at 1 kHz in the Ethernet Motor Supervisor (EMS) board, converts the desired torque into the respective PWM as a percentage of the full-scale voltage feeding the motor, and sends that value to the motor driver board 2FOC.

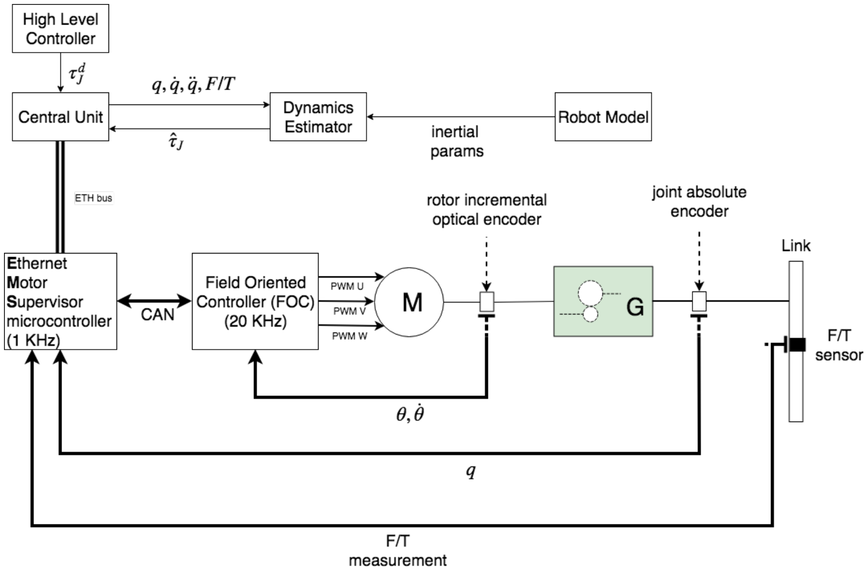


Fig. 2.5 Motor driver and low level torque control hardware design.

Note. *Whatever the selected high level control mode and respective control parameter (desired joint position, velocity or torque), the EMS output parameter sent to the motor driver is always a* PWM *value.*

The 2FOC motor driver board can drive a BLDC motor with a Field Oriented Control algorithm, running at 20 kHz. The driver board feeds directly the PWM power signal to the stator phases, generating a rotating magnetic field that drags the rotor round with it. The motor driver closes the loop on the rotor position feedback (for fine controlling the stator magnetic field and thus maintaining a constant torque) and if required, can close the loop on the stator current for increased robustness against voltage perturbations.

2.2.2 Joint torque estimation

iCub is not equipped with direct joint torque sensing, but instead, has six-axis force-torque sensors mounted in the cross section of the arm [Parmiggiani et al., 2009] and leg links as well as below the ankles. Combined with a tactile skin, these sensors are integrated in a whole-body dynamics estimation framework [Traversaro, 2017, chap. 4] which estimates the external contact forces and all the joint torques, as will be explained more in detail in chapter 4. The joint torque estimate is the feedback in the inner control loop running at 1 kHz.

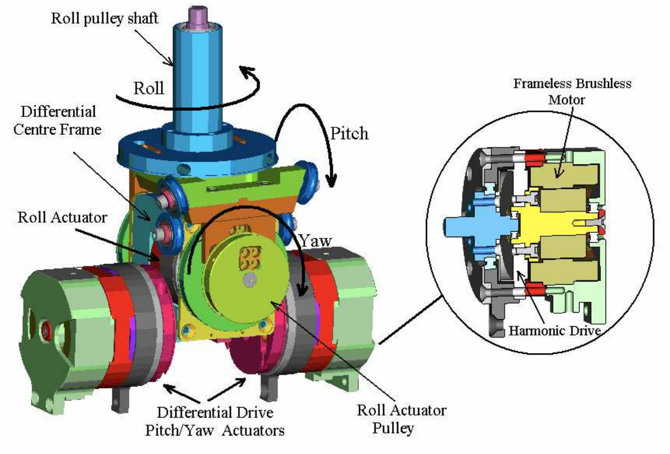


Fig. 2.6 A CAD view of the joints connecting the waist to the torso, actuated by a differential drive and three motors. Source: Tsagarakis et al. [2006]

2.2.3 Coupled joints and differential drives

As mentioned previously, some joints are coupled through a cable differential drive (shoulder and torso joints). We perform our identification experiments on the torso coupled joints, which motor groups and differential drive are depicted in the figure 2.6. The cable driven differential adds significant elasticity to the joints actuation and. combined with friction in the transmission elements, increases the hysteresis in the model relating the input PWM signal to the joint output torque, as well as in the model relating the joint velocity to the friction torque.

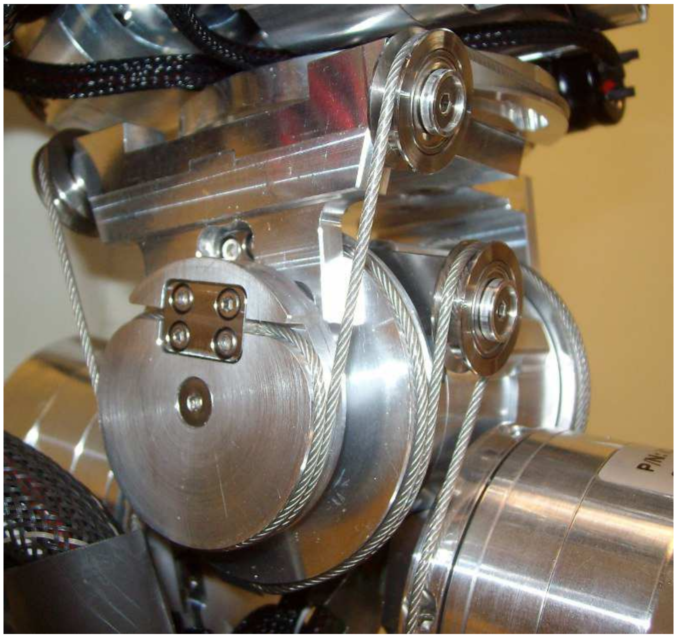


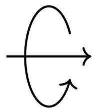
Fig. 2.7 Close-up view of the waist differential drive. Source: Tsagarakis et al. [2006]

2.3 Modelling, problem formulation and assumptions

2.3.1 Notation

We summarize below the notation used for describing the models and the algorithms in this chapter:

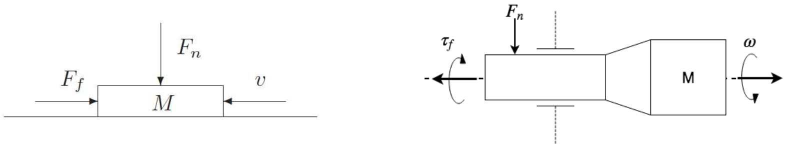
${}_{\mathrm{B}}\tau_{m,l}$	Motor torque applied on point B of link l;
${}_{\mathrm{B}}\tau_{f,l}$	Generalized friction torque applied on point B of link l;
${}_{\mathrm{B}}\tau_{c,l}$	Coulomb friction torque applied on point B of link l;
${}_{\mathrm{B}}\tau_{v,l}$	Viscous friction torque applied on point B of link l;
F_n	Force normal to a contact surface;
μ, σ	Respectively Coulomb and viscous friction coefficients;
$\mathrm{sign}(\omega)$	Sign of the angular velocity;
τ_J	Output joint torque (applied to the joint child link);
$\mathbb{I}_n$	Identity matrix of dimension n.



The notation for describing an arbitrary angular velocity vector ω or torque vector τ about an arbitrary axis x follows the right-hand rule, such that the orientation of ω or τ is defined by the arrow along that axis, the curved arrow being an indication that we are in presence of an oriented vector.

2.3.2 Friction Models

While modeling the multiple components in the joint transmission chain, we will often come across friction torques that can be found in the bearings, reduction drives, coupling systems or result from electromotive forces. Before addressing the friction modeling specific to each component, we present in this section the general concepts and typical models. There is a fair list of candidate friction models: The most simple and common are the static ones (constant parameters and structure) and are not defined for a zero relative velocity between the two surfaces in contact; the "switching" models (seven parameters model, Dahl model described in [Van Geffen, 2009]), and the most advanced dynamic models (LuGre model) switch between two distinct "stick" and "slip" states applying specific parameters for each of them; the dynamic models account for the transition between stick and slip phases in



(a) (b)

Fig. 2.8 Representation of the friction force F_f on an object M moving relative to a flat surface at velocity v (a), or the friction torque τ_f on an object rotating about a rotary joint axis at velocity ω (b). F_n is the force normal to rotation axis.

a continuous function. Some of these models are quite empirical, which is acceptable as long as they match the actual system behavior in the desired range of working conditions: static, slow or fast motion up to a maximum expected joint velocity and acceleration. We describe below the two most commonly used models: the Static-Coulomb-Viscous friction model and the Stribeck-Coulomb-Viscous friction model.

The Static-Coulomb-Viscous friction model

Coulomb developed the concept of a friction (1785) proportional to the load, opposed to the motion, and independent of the contact area. Morin introduced the static friction (1833), and Reynolds the viscous friction (1866). We represent in Figure 2.8 a moving object M, either sliding over a flat surface (Figure 2.8a) or revolving about a rotary joint (Figure 2.8b). In the rotary joint case, we can see the load force F_n on the joint rotation axis (F_n is normal to the axis), the rotation velocity ω and the total friction torque τ_f due to the rotary contact.

The friction torque can be written as the superposition of the three models, Coulomb, viscous and static friction, as follows:

$$\begin{aligned}\tau_f &= \tau_c + \tau_v + \tau_s &\quad (2.1)\\ &= -\mu F_n \operatorname{sign}(\omega) - \sigma\omega - \mathcal{T}_s(\omega), &\quad (2.2)\end{aligned}$$

where the Coulomb friction τ_c is linear with respect to the load force F_n on the rotation axis, and the viscous friction τ_v is a linear function of the axis angular velocity ω. μ and σ are respectively the Coulomb and viscous friction coefficients and are positive, such that the friction and the rotation velocity always have opposite signs. The static friction τ_s, initially

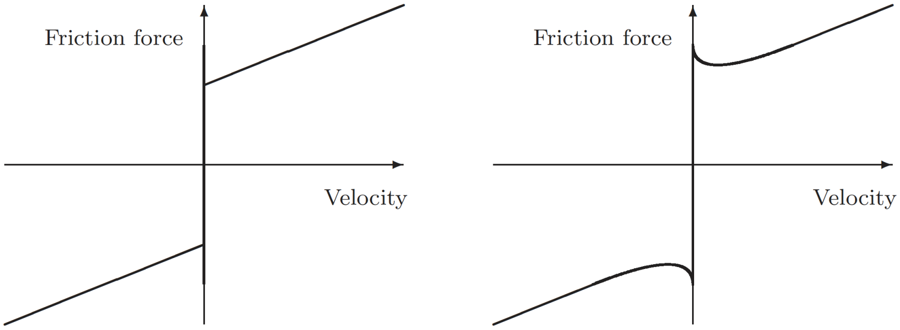


Fig. 2.9 Models of friction force versus angular velocity. (left) Static, Coulomb and viscous friction model. (right) Negative viscous, Coulomb and viscous friction model (Stribeck). Source: Van Geffen [2009].

expressed here as a general function of ω, $\mathcal{T}_s(\omega)$, is the torque required to set the axis into motion starting from a null velocity, and typically is higher than the Coulomb and Viscous added components close to a null velocity. The first simple model would consider the static friction as a constant, usually higher than the Coulomb friction, as described in Van Geffen [2009] and illustrated in Figure 2.9(a). This approximation creates a discontinuity when the system velocity crosses zero, and thus can cause numerical issues and torque instabilities.

The Stribeck-Coulomb-Viscous friction model

In most cases the transition from static to Coulomb-Viscous friction is progressive: the friction first decreases with increasing velocity from rest state (like a negative viscous friction), before increasing again following the Viscous friction model. This is called the Stribeck effect, as shown in Figure 2.9(b). In some cases, the initial friction drop can be significant, and the overall friction appears not to have the Coulomb component.

The Boundary Lubrication: We can find in the literature more detailed Stribeck models characterizing the friction behavior near the null velocity. At very low speeds, when the surfaces are in solid-to-solid contact, we are in a regime called “Boundary Lubrication” [Armstrong-Hélouvry et al., 1994], the sheared solid surfaces being the boundary lubricants. In the case of a

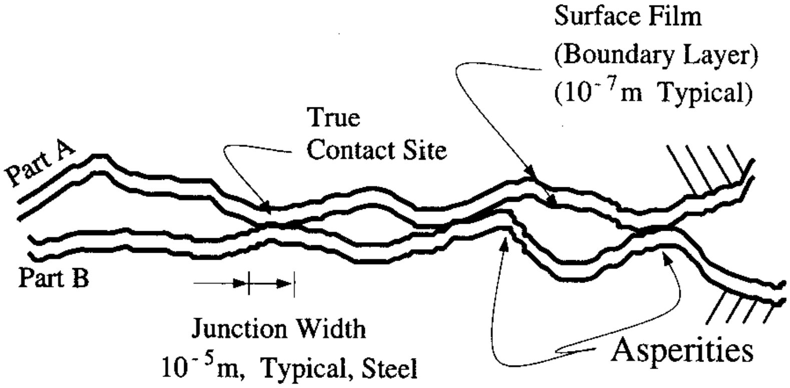


Fig. 2.10 Solid-to-solid contact between microscopic asperities, and formation of the Boundary Layer. Source: [Armstrong-Hélouvry et al., 1994, Figure 4].

solid-to-solid contact, the contact topography can be modeled as a series of asperities, compressed against each other, creating asperity junctions as illustrated in Figure 2.10, even in the case of apparent point or line contacts (between gear teeth), also referred to as nonconformal contacts [Armstrong-Hélouvry et al., 1994, Section 2.1.1, Figure 3]. In the case of steel on steel contact, even in absence of lubricants, an oxide film forms on the surface of the asperities, producing a boundary layer that reduces significantly the friction.

In a system lubricated by oil and grease there are four regimes of lubrication, as illustrated in Figure 2.11: *static friction and presliding displacement* where the asperities behave like springs; *boundary lubrication*, where the velocity is insufficient for building a fluid film between the surfaces (regime at very low speeds); *partial fluid lubrication*, where the fluid is drawn by the motion into the contact load region; at last the *full fluid lubrication*, where the viscous friction classical behavior arises.

Remark 2.1. *The previous models are all static, i.e. the model parameters are the same for all working regimes (joint velocities). They are unable to properly describe what happens at zero velocity (stick phase) or in the transition from stick to slip phase. Considering the joints of a humanoid robot, they operate near zero velocity or cross zero velocity quite often. In this case a dynamic or switching model is preferred. Some models, more complex, deal with hysteresis and have a local memory of previous slip/stick*

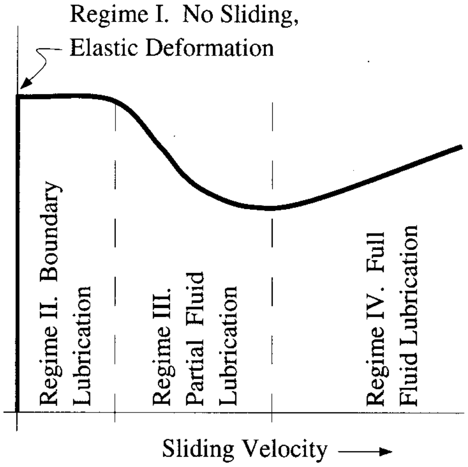


Fig. 2.11 The four dynamic regimes of the generalized Stribeck model. Source: [Armstrong-Hélouvry et al., 1994, Figure 5].

states, but this is out of the scope of this thesis, and will be addressed in a future work.

2.3.3 The Brushless Motor Control and Modelling

A "reversed" DC motor The BrushLess DC motor (BLDC) is a "reversed" DC motor: the rotor is composed of a hub holding permanent magnets arranged in alternating pole pairs (commonly two to eight pairs); the stator is composed by electromagnetic coils, i.e. electric windings wrapped in slots around steel laminated poles. Once excited, the stator generates a magnetic field that can be rotated through electronic commutation or modulation of the current in the windings [Akin et al., 2011]. The motor driver uses the feedback on rotor position for aligning the stator field always along or as close as possible to the rotor quadrature axis, which results in a torque that causes the rotor and field to revolve at the same speed, which classifies this kind of motor as the "synchronous" type. The electronic commutation allows to drop the mechanical brushed commutation, with the following benefits: less friction and less voltage drop from the brushes sliding contacts resistance; less sparks on the same brushes due to the abrupt current switching. Unlike the DC motor where the current commutation and the rotating field generation occur in the rotor, on the brushless motor they occur in the stator, hence the name of "reversed" DC motor.

Generated torque and Back ElectroMotive Force The generated torque can be constant for a given PWM setpoint, and independent from the rotor angular position, as it will be described in the following subsections. This can be achieved through Sinusoidal Commutation Control or Field Oriented Control and for a large range of speeds if the loop is closed on the measured phase currents projected on the Direct-Quadrature frame [Kiran and Swamy, 2014] [Yousef and Abdelmaksoud, 2015].

We also describe how the motor torque, the amplitude of the rotating magnetic field, the common amplitude of the sinusoidal currents on all three phases of the stator and the motor input parameter PWM are all linearly dependent [Dixneuf and Gilabert, 2005], such that we can write, when the rotor is not moving:

$$\tau_{\text{PWM},m} = k_{\text{PWM},\tau}\ \text{PWM} \tag{2.3}$$

The rotation of the magnetic field generated by the rotor induces an electromotive force (Faraday's law) in the stator windings, E_v , also called Back ElectroMotive Force (Back EMF), and therefore an induced current. we will see in the following subsections that the Back EMF torque can be writen as follows:

$$\tau_{\text{BEMF}} = -k_{\text{BEMF},\tau}\,\dot{\theta} \tag{2.4}$$

Stator windings arrangement The most common design of BLDC motors comprises a stator with three-phase windings, arranged in one of the two common configurations as shown in figure 2.12: the ”delta” configuration where the windings are connected to each other in a triangle shape; the ”Y-shaped”, also referred to as ”star” configuration where the windings are connected to a central contact point, leaving the loose ends connected to the phase interface terminals U, V and W.

There are three control methods for doing the electronic commutation of the phases: the trapezoidal, the sinusoidal and the Field Oriented Control as described further.

Trapezoidal control

The trapezoidal control is the simplest one, where we have always two excited phases (“ON” state phases) and the a third one floating i.e. in open-circuit state (“OFF” state). This technique is well suited for BLDC motors because in a certain range of rotor speeds, the constant current in the phases

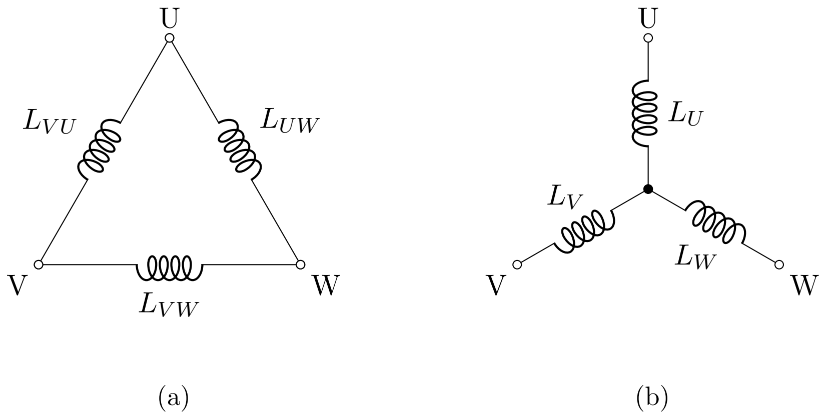


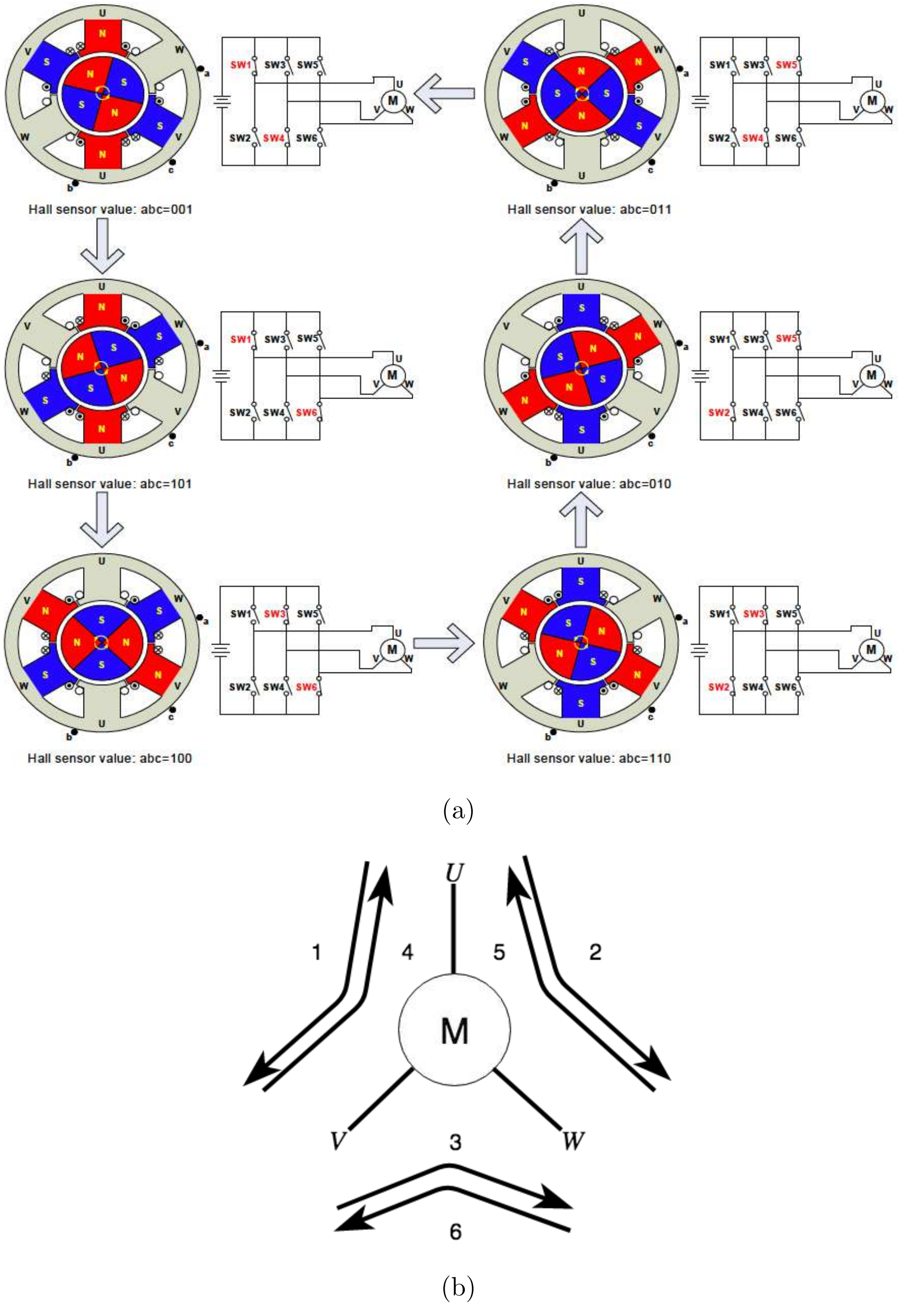
Fig. 2.12 The delta (a) and star i.e. Y-shaped (b) winding types on the stator.

combined with the trapezoidal Back EMF results in a constant torque applied to the rotor. We illustrate in figure 2.13 the commutation sequence happening during an electric cycle, matching a half mechanical cycle (half turn of the rotor).

Definition 2.1. *Electric cycle: a cycle in the oscillation of the current flowing in a stator winding (pole), during which the stator winding changes its polarity twice. Mechanical cycle: a cycle in the rotation of the rotor.*

We can see in the right column of the figure 2.13 the energized phases and the respective current direction for each rotation step. Driving the current in the phases in such way requires a "three-phase inverter" composed of six IGBTs or MOSFETs inter-connected in a bridge structure. Each pair of power transistors sets one of the phase input terminals **U**, **V** or **W** to DC+ or DC-. As an example, for setting the phases U and V as illustrated in step 1, the terminal **U** has to be set to DC+, and terminal **V** to DC-, i.e. the transistors SW1 and SW4 are both set to the same state ON, while all the other transistors are set to OFF (floating state). Actually the way the transistors are driven depends on the PWM or current control mode.

PWM control mode: In the commutation sequence step 1 illustrated in figure 2.13, the transistors SW1 and SW4 are supposed to close the phases **U** and **V** loop circuit, but the current in that loop needs to be regulated to a desired value through a PWM drive signal $\mathrm{PWM_U}$. In such case, either both transistors are driven by the same PWM signal $\mathrm{PWM_U}$ (called the hard



(a)

(b)

Fig. 2.13 Electronic commutation sequence of a Y-shaped winding stator, following a trapezoidal control scheme. Source: [Zhao and Yu, 2014, Figure 11] chopping operation causing high frequency ripples in the current that can damage the motor), either SW4 is set to ON for the whole duration of step 1 while SW1 is driven by $\mathrm{PWM_U}$ (called the soft chopping operation). During the soft chopping operation, the $\mathrm{PWM_U}$ high state sets SW1 to "ON", while the $\mathrm{PWM_U}$ low state sets SW1 to "OFF" and the current flows through the diode LD_V which is then in forward direction. This behavior reduces the high frequency ripples.

Note. *Every inductor (winding round a ferromagnetic core) has a self-capacitance due to the proximity between the wound wire loops [Massarini and Kazimierczuk, 1997]. When an inductor like a phase winding has an electric current flowing through it and the circuit is abruptly opened, a high voltage is induced (Lenz's law) and the coupled inductor and self-capacitance behave like a free oscillator, creating current ripples.*

Hysteresis (current) control mode: If a hysteresis type current regulator is used instead of the PWM control mode, the power transistors in the MOSFET bridge are driven by a ON/OFF signal depending on the error between the current flowing in each phase and the desired reference current determined by the motor supervisor synchronizing the rotor rotation. From this point on we consider only the PWM control mode.

The motor driver control system The Figure 2.14 illustrates an example of motor driver control system: the hall effect sensors mounted in the stator provide the rotor position feedback to the motor supervisor logic circuit; the motor supervisor integrates the single input PWM drive signal with the rotor position feedback and generates a set of switching signals—"ON", "OFF", "PWM"—intended to drive the stator phases as described earlier; the Three Phase Inverter uses these signals to drive the six power transistors that feed the stator phases, generating a rotating magnetic field of desired intensity and synchronized with the rotor position.

From PWM to current and from current to torque At each step of the electric cycle, 2 phases in series (U-V, V-W, W-U) are supplied with a given PWM power signal. As depicted in figures 2.15 and 2.18, the two energized phases can be modeled by an equivalent electric circuit composed by: the windings terminal to terminal (e.g. U to V) resistance r; the respective self-inductance l; the total Back EMF voltage e_v on both phases (U and V). The circuit is a typical "RL" low-pass filter which cut-off frequency depends on the resistance r and inductance l values, and can be inferred by

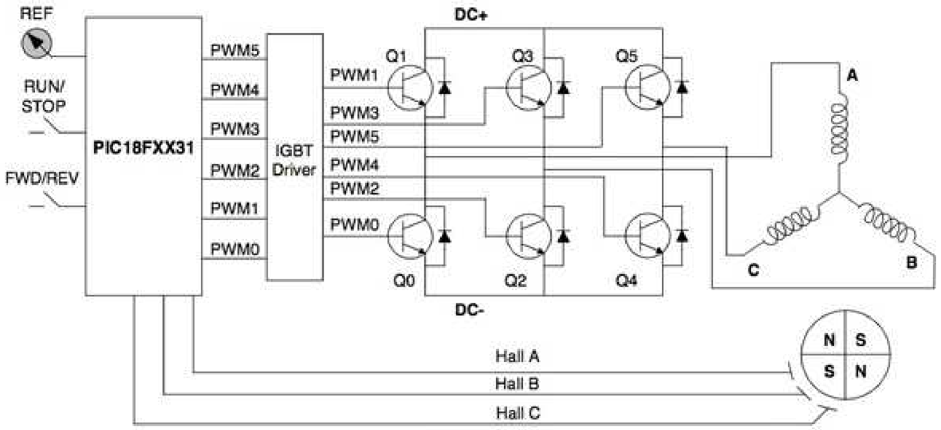


Fig. 2.14 A complete closed-loop control system for a sensored three-phase BLDC motor. The circuit comprises the microcontroller supervising the PWM outputs, a six MOSFET bridge driver, the MOSFET bridge and the BLDC motor. Source: Microchip [1]

computing the gain norm $\|G\|$ in sinusoidal stationary regime, with a stalled rotor ($e_v = 0$):

$$\|G\| = \left\| \frac{u_r}{e_i} \right\| = \left\| \frac{1}{1 + j\omega \frac{l}{r}} \right\| = \frac{1}{\sqrt{1 + \omega^2 \frac{l^2}{r^2}}} \tag{2.5}$$

The gain norm drops to -3dB for $\omega = \omega_c$ which characterizes the cut-off frequency. For the MOOG brushless motor used on the iCub hip pitch or torso joints, the datasheet (appendix C) specifies a terminal to terminal resistance of $0.682\,\Omega$ and self-inductance of $0.452\,\text{mH}$. From those values we get a cut-off frequency of:

$$\|G\| = \frac{1}{\sqrt{2}} \Longleftrightarrow f_c = \frac{r}{2\pi l} \sim 240\,\text{Hz} \tag{2.6}$$

Note. *The two "ON" phases resistance and self-inductance are characterized in the motor datasheet respectively as the "terminal to terminal" resistance and inductance, and are modeled here as the* stator self-inductance *l.*

---

[1] Figure 3 at `https://www.digikey.it/en/articles/techzone/2013/mar/an-introduction-to-brushless-dc-motor-control`.

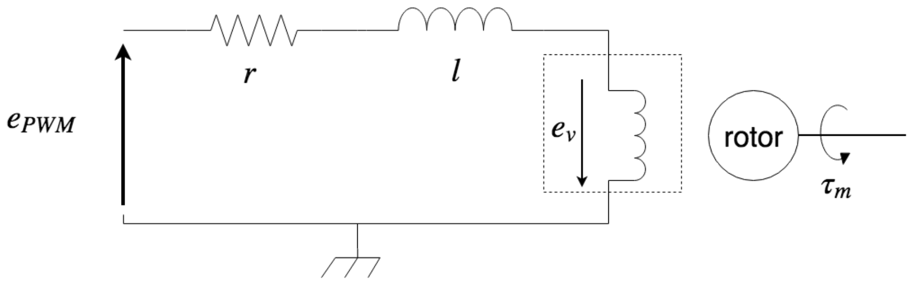


Fig. 2.15 Stator two "ON" phases electric model. e_{PWM} is the PWM input signal, e_v is the Back EMF voltage on the windings, r the windings resistance and l the self-inductance.

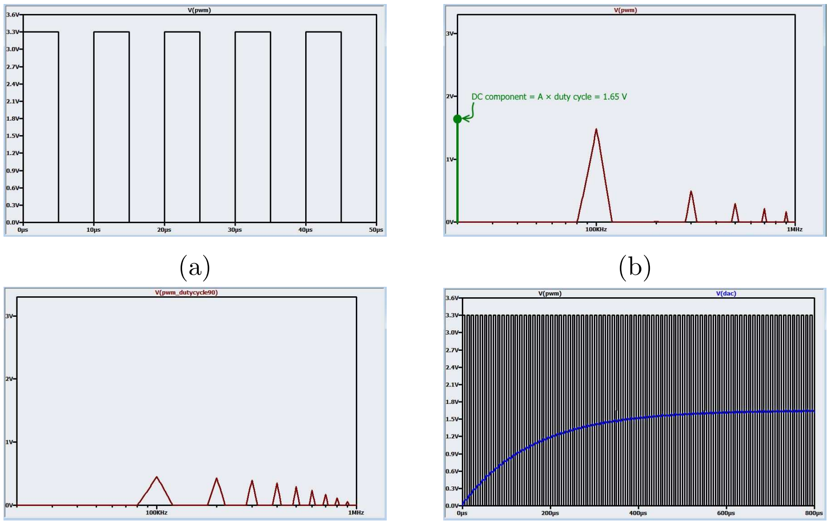
The PWM signal has a typical frequency spectrum as depicted in figure 2.16, obtained through a Fast Fourrier Transform (FFT) of the time-domain signal. The spectrum shows a pattern having a DC component, a first lobe on the PWM carrier frequency and a series of harmonics with decreasing amplitude on higher frequencies. We expect the RL low-pass filter to retain the DC component, and suppress the first lobe and remaining harmonics.

In the frequency domain, the farther the first lobe is i.e. the higher the PWM carrier frequency is, the more it is attenuated by the filter as well as all the harmonics, and the less ripples we get in the time-domain. In these conditions, for a fixed PWM duty-cycle value, the current flowing in the two "ON" phases rises smoothly and reaches an approximately constant limit after a settling time. Actually, increasing the PWM carrier frequency is a good way to smooth the ripples without affecting the settling time which depends only on the filter.

Note. *The FOC motor driver used on the iCub motor control can run and generate a PWM at up to* 40 kHz*, which is two orders of magnitude higher than the stator self-inductance cut-off frequency and allows a good current response in the stator phases.*

The DC component in the FFT sets the target voltage u_r reached after the filter settling time, and is given by: DC component = carrier Amplitude × PWM duty cycle.

Because of the settling time required before the voltage reaches the target limit, there is a ripple in the phase current each time there is a phase switch, i.e. at every step transition of the electronic commutation described in figure 2.13. That effect can be observed in the resulting rotor torque (2.17), which presents ripples at each phase switch—every 60° rotation—instead of the



(a) (b)

(c) (d)

Fig. 2.16 PWM time-domain signal (a), frequency spectrum (FFT) (b), frequency spectrum filtered by the low-pass RL filter (c), time-domain filtered signal (d). Source: [Keim, 2016]

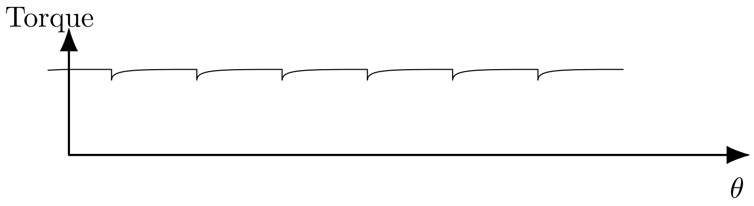


Fig. 2.17 Rotor torque ripple in Trapezoidal Control. Source: [Akin et al., 2011, Figure 3]

expected constant smooth value.

If we ignore the ripple issue, we can now replace the PWM signal in the original model 2.15 by a constant voltage supply E_0 and drop the inductance l as shown in the equivalent electric model 2.18. E_0 is proportional to the PWM duty-cycle: $E_0 = k_{\text{PWM},v}$ PWM. The Back EMF is proportional to fixed parameters like the number of winding turns per phase N, the rotor radius r, length l and magnet flux density B, and to a variable parameter

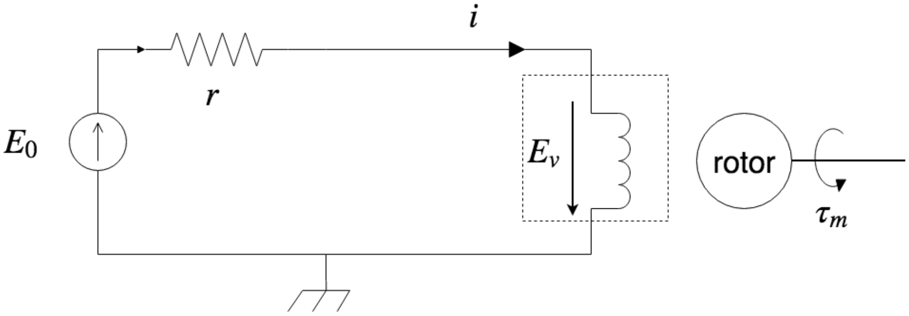


Fig. 2.18 Stator two "ON" phases equivalent model in continuous regime, i.e. for an imput PWM with a frequency $f_e \gg f_c$ where f_c is the cut-off frequency of the phase self-inductance l. Typically $f_e \sim 20\text{kHz}$. E_0 is the equivalent voltage that would generate the same average current as the one generated by the PWM signal.

which is the rotor velocity $\dot{\theta}$ [Akin et al., 2011]:

$$E_v = 2NlrB\dot{\theta} = k_{\text{BEMF},v}\,\dot{\theta} \tag{2.7}$$

The rotor torque depends on two residual reluctance components and a third term which is the main produced torque, proportional to the same fixed parameters mentioned above and to the phase current i:

$$\tau_m \sim \frac{4N}{\pi}Brl\pi i = k_t\, i \tag{2.8}$$

We can then express the current flowing in the stator as in (2.9) and express the resulting torque as in (2.10). The torque constant is given in the motor datasheet ($k_t = 0.047\,\mathrm{N\,m\,A^{-1}}$) and can later be used to cross check the PWM to torque constant $k_{\text{PWM},\tau}$.

$$i = \frac{E_0 + E_v}{r} = \frac{k_{\text{PWM},v}}{r}\ \text{PWM} - \frac{k_{\text{BEMF},v}}{r}\,\dot{\theta} \tag{2.9}$$

$$\tau_m = k_t\, i = \left(k_t\,\frac{k_{\text{PWM},v}}{r}\right)\ \text{PWM} - \left(k_t\,\frac{k_{\text{BEMF},v}}{r}\right)\,\dot{\theta} \tag{2.10}$$

$$k_{\text{PWM},\tau} = k_t\,\frac{k_{\text{PWM},v}}{r}, \quad k_{\text{BEMF},\tau} = k_t\,\frac{k_{\text{BEMF},v}}{r} \tag{2.11}$$

Sinusoidal Commutation Control

The Trapezoidal commutation has two main inconveniences: the abrupt switching of the phase current creates torque ripples; the current and torque are approximately constant only at high speed as the trapezoidal Back EMF, combined with the constant phase current, optimizes the constant torque approximation. For these reasons the Trapezoidal Control fails to provide a smooth torque at low speed.

The Sinusoidal commutation solves that problem by generating a smooth rotating magnetic field always in quadrature direction with respect to the rotor direct axes (in the simple case of a two pole rotor) as illustrated in the figure 2.19, which allows to avoid the current commutation spikes and produce a constant torque free of ripples [Kiran and Swamy, 2014] [Yousef and Abdelmaksoud, 2015].

The controller drives the winding currents following sinusoidal waveforms which phase varies smoothly and synchronously with the rotor position. At each instant, the winding currents are phase shifted by 120 degrees from each other and are placed 120 degrees from each other around the stator. Furthermore the windings are distributed (sometimes referred to as "sinusoidal" windings in the litterature) i.e. for a single stator pole the windings are placed around several slots in the stator armature, and the resulting poles end up being partially overlapped [Akin et al., 2011]. This results in a smoother transition between the stator magnetic poles. As explained in appendix D, these properties allow the three phase currents to combine and create a magnetic field $\vec{B}$ rotating smoothly with the rotor, proportional and aligned with a current Space Vector $\vec{I}$ computed as follows [Dixneuf and Gilabert, 2005]:

$$\vec{I} = \vec{I}_{\mathrm{U}} + \vec{I}_{\mathrm{V}} + \vec{I}_{\mathrm{W}} = \frac{3}{2}\overline{I}, \quad \vec{B}_I \propto \vec{I}, \tag{2.12}$$

$$\text{with} \quad \overline{I} = \vec{u}_{\mathrm{U}} I \sqrt{2} e^{j\omega t} \tag{2.13}$$

Where $\vec{I}_{\mathrm{U}}$, $\vec{I}_{\mathrm{V}}$ and $\vec{I}_{\mathrm{W}}$ are projections of the Time Vector $\overline{I}$. The combined field space vector $\vec{B}_I$ interacts with the rotor magnetic moment $\vec{M}_j$ (D.2) creating a couple that can be computed as follows:

$$C_{em} = \vec{M}_j \times \vec{B}_I \propto I \sin\left(\vec{M}_j, \vec{B}_I\right) \tag{2.14}$$

Where I is the amplitude of the winding current in a single phase. So if the generated rotating field $\vec{B}_I$ is always in quadrature with the rotor direct

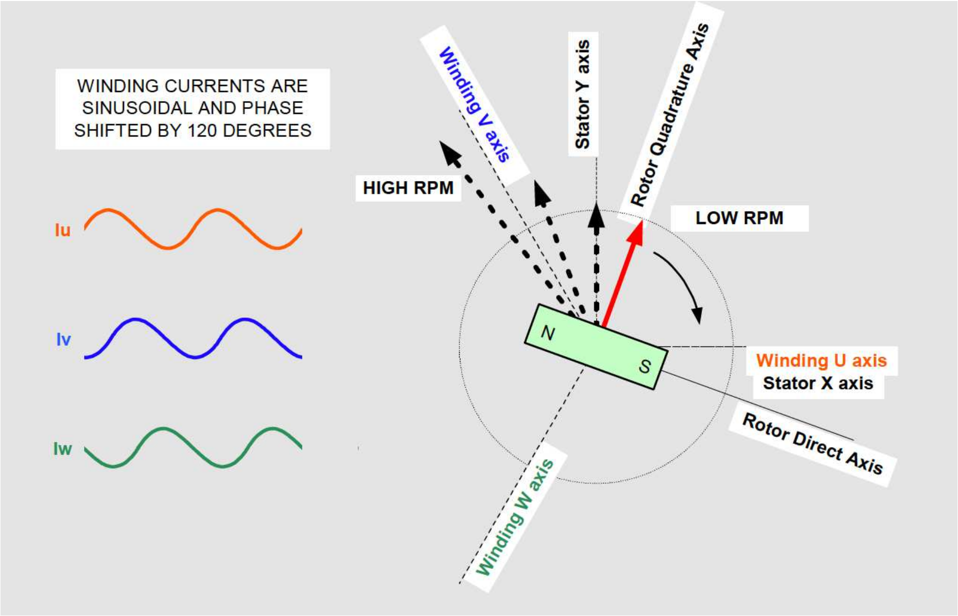


Fig. 2.19 Sinusoidal Control: current space vector rotating synchronized with the rotor position. The dashed arrows show the current space vector lagging away of the quadrature axis as the rotation speed increases. Source: Microchip, 2011 [2].

axis, the torque is constant, maximal and depends only on the phase current amplitude. We can consider the same model defined by (2.9), (2.10) and (2.11).

Tracking the torque through current control: The control loop architecture of a Sinusoidal Commutation has a motor supervisor logic unit that differs from the one from the Trapezoidal Commutation. The motor supervisor is composed of a sinusoidal commutator that synthesizes a sinusoidal voltage signal for each phase, and PWM modulators that generate from those signals the PWM waveforms, which are then routed to the three-phase inverter (MOSFETs bridge power stage). In the case where the motor controller tracks the current in each phase winding, the loop is closed on each phase current i_{U}, i_{V} and i_{W} separately, right after the sinusoidal commutator and the error processed through a PI controller before the PWM modulation and power amplifying stages. This means that as the rotor angular

---

[2] `https://www.microchip.com`.

velocity increases, each PI controller has to track a sinusoidal current and compensate a sinusoidal Back EMF with increasing frequency. Due to the PI controllers bandwidth limitation, the tracking fails for high rotor speeds and the current space vector lags away from the rotor quadrature axis.

Field Oriented Control

The Field Oriented Control solves this issue by applying the current tracking in the reference frame attached to the rotor, commonly called the direct-quadrature (d-q) frame [Kiran and Puttaswamy, 2014]. This is achieved in three main steps: the phase currents $\overline{I}_{\mathrm{U}}$, $\overline{I}_{\mathrm{V}}$ are first transformed from the stator static reference frame (stator X axis and Y axis in figure 2.19) to the rotating d-q reference frame ($\overline{I}_d$, $\overline{I}_q$); the loop is then closed on the desired current, and the error processed by the PI controllers; the output corrected voltages are transformed back to the stator reference frame before being routed to the PWM modulator. In the d-q frame, the current is theoretically constant, and the Back EMF a linear function of the rotor angular velocity ω, and so we stay far from the PI controllers bandwidth limitation. Under these conditions the tracking is straightforward: driving the direct component $\overline{I}_d$ to zero and the quadrature component $\overline{I}_q$ to the desired current i.e. the motor input command.

Conclusion on the motor model definition

Currently the iCub firmware does not have a current loop implemented in the motor driver so we will consider only the Sinusoidal Commutation control use case in the methodology described in further sections. Furthermore we will control the motor in PWM mode, using the feed-forward model defined in (2.10) and (2.11):

$$\tau_{\mathrm{PWM},m} = k_{\mathrm{PWM},\tau}\ \mathrm{PWM} - k_{\mathrm{BEMF},\tau}\,\dot{\theta} + \tau_{f,m} \tag{2.15}$$

Where PWM is the motor driver input parameter, $\dot{\theta}$ the rotor angular velocity, $k_{\mathrm{PWM},\tau}$ is the PWM to torque coefficient and $k_{\mathrm{BEMF},\tau}$ is the velocity to Back EMF torque coefficient. Here we added a residual generalized friction term $\tau_{f,m}$ which could be the friction on the axis bearings. We can assume $\tau_{f,m}$ to be negligible compared to the Back EMF torque which can be considered as a viscous friction, since it acts as a counter torque proportional to the rotor velocity.

2.3.4 Velocity, Torque and Power Conversions by Reduction or Coupling Drives

Harmonic Drives

Close after the motor in the actuation chain, the harmonic drive is the next component to transform the transmission dynamics variables, namely the angular velocity, the torque and the rotor apparent inertia in the sub-system [motor]-[harmonic drive]. We describe in 2.20 a spinning mass m with the respective rotational inertia ${}_{\mathrm{M}}\dot{h}_m$ and angular velocity ${}^{\mathrm{M}}\omega$ both expressed on the point M of the shaft. We can apply this model to an iCub main joint motor group (figure 2.3), where G is the harmonic drive and m is the lumped mass of the fast rotating parts—the rotor and the harmonic drive wave generator. We observe a torque ${}_{\mathrm{M}}\tau$ across the point M of the shaft. The harmonic drive transforms the three dynamics quantities ${}_{\mathrm{M}}\dot{h}_m$, ${}^{\mathrm{M}}\omega$ and ${}_{\mathrm{M}}\tau$ depending on the step-down ratio ρ, such that from its output point of view, i.e. on point G, the mass m is seen rotating at an angular velocity ${}^{\mathrm{G}}\omega = \rho^{-1}\,{}^{\mathrm{M}}\omega$, the output torque is ${}_{\mathrm{G}}\tau = \rho\;{}_{\mathrm{M}}\tau$, and the rotational inertia is ${}^{\mathrm{G}}\dot{h}_m = \rho^2\,{}_{\mathrm{M}}\dot{h}_m$. We give a more detailed explanation on these conversions further on.

Axiom 2.1. *A reduction drive doesn't inject power into a mechanical system: it is conservative with respect to the power of the torques applied to input shaft, except for the power loss due to friction and the potential power stored in the elastic deformation of the drive components. The same principle is applicable to cable driven coupling mechanisms like the differential drives on iCub.*

Note. *In the harmonic drive, the friction lies in the contact between the wave-generator and the flex-spline, and in the contact between the flex-spline and the circular spline gear teeth.*

Conversion of angular velocity and torque: We write the identity between the input power and the output power. We consider the effect of the generalized friction in the reduction drive as a negative power added to the input:

$$W_{in} + W_{friction} = W_{out} \tag{2.16}$$

$$\Leftrightarrow {}_{\mathrm{M}}\tau\,{}^{\mathrm{M}}\omega + {}_{\mathrm{M}}\tau_{f,g}\,{}^{\mathrm{M}}\omega = {}_{\mathrm{G}}\tau\,{}^{\mathrm{G}}\omega, \tag{2.17}$$

and in view of the unaltered input to output velocity ratio, we get:

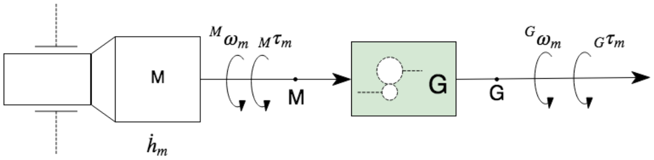


Fig. 2.20 Reduction drive dynamics conversions.

$$
\begin{aligned}
&{}^{\mathrm{G}}\omega = \rho^{-1}\,{}^{\mathrm{M}}\omega \;\Rightarrow\; {}_{\mathrm{G}}\tau = \rho\,{}_{\mathrm{M}}\tau + \rho\,{}_{\mathrm{M}}\tau_{f,g} &(2.18)\\
&\Leftrightarrow {}_{\mathrm{G}}\tau = \rho\,{}_{\mathrm{M}}\tau + {}_{\mathrm{G}}\tau_{f,g} &(2.19)\\
&\Leftrightarrow {}_{\mathrm{G}}\tau = \left(\rho + \frac{{}_{\mathrm{G}}\tau_{f,g}}{{}_{\mathrm{M}}\tau}\right){}_{\mathrm{M}}\tau &(2.20)
\end{aligned}
$$

Where ${}_{\mathrm{M}}\tau_{f,g}$ is the generalized friction in the reduction drive expressed on M and ${}_{\mathrm{G}}\tau_{f,g}$ the generalized friction expressed on G. We can either consider that the resulting input to output torque ratio is changed from ρ to $\left(\rho + \frac{{}_{\mathrm{G}}\tau_{f,g}}{{}_{\mathrm{M}}\tau}\right) < \rho$ or consider an ideal friction-less reduction drive with torque ratio ρ in series with a friction brake applying ${}_{\mathrm{G}}\tau_{f,g}$.

Conversion of angular momenta: Since the derivative of an angular momentum is equivalent to a torque, it is converted by the gearbox ratio the same way, as follows:

$$
\begin{aligned}
&{}_{\mathrm{G}}\dot{h}_m = \rho\,{}_{\mathrm{M}}\dot{h}_m = \rho\,\left({}_{\mathrm{M}}I_m\,{}^{\mathrm{M}}\dot{\omega}\right)\\
\Leftrightarrow\;&{}_{\mathrm{G}}I_m\,{}^{\mathrm{G}}\dot{\omega} = \rho\,\left({}_{\mathrm{M}}I_m\,\rho\,{}^{\mathrm{G}}\dot{\omega}\right)\\
\Leftrightarrow\;&{}_{\mathrm{G}}I_m = \rho^2\,{}_{\mathrm{M}}I_m &(2.21)
\end{aligned}
$$

Which makes ${}_{\mathrm{G}}I_m$ be the apparent inertia of the spinning mass m. We then realize that in the case of a motor group on iCub, even when the standalone inertia of the rotor is negligible, it might not be the case of its apparent inertia if the actuation chain is using a high ratio gearbox and performing fast rotations: ${}_{\mathrm{G}}I_m$ is then four orders of magnitude greater than ${}_{\mathrm{M}}I_m$.

Static friction and stiffness in Harmonic drives In the study [Chedmail and Martineau, 1970], the authors characterized the friction and stiffness in harmonic drives by running load tests on the drives, and observed that the static friction, referred to in the article as “dry friction torque”, depended on the applied load and on the angular position of the rotating shaft. This dependency was highlighted by the hysteresis in the experimental results. They defined multiple hypothetical mechanical models of the harmonic drive in order to match the experimental observations. The models were always composed by three dynamic parameters: the friction between the gear teeth (function F); the stiffness of the flexspline (function S); and the play in the gears (function P). These functions were then combined in different, sequential arrangements: F-P-S; F-S-P; S-P-F; and so on. The model best matching the experimental results turned to be the arrangement S-F-P: stiffness-friction-play, with an almost null play and a friction torque of 2.6N m. We will later see that this value is fairly close to the hysteresis observed in our own experiments.

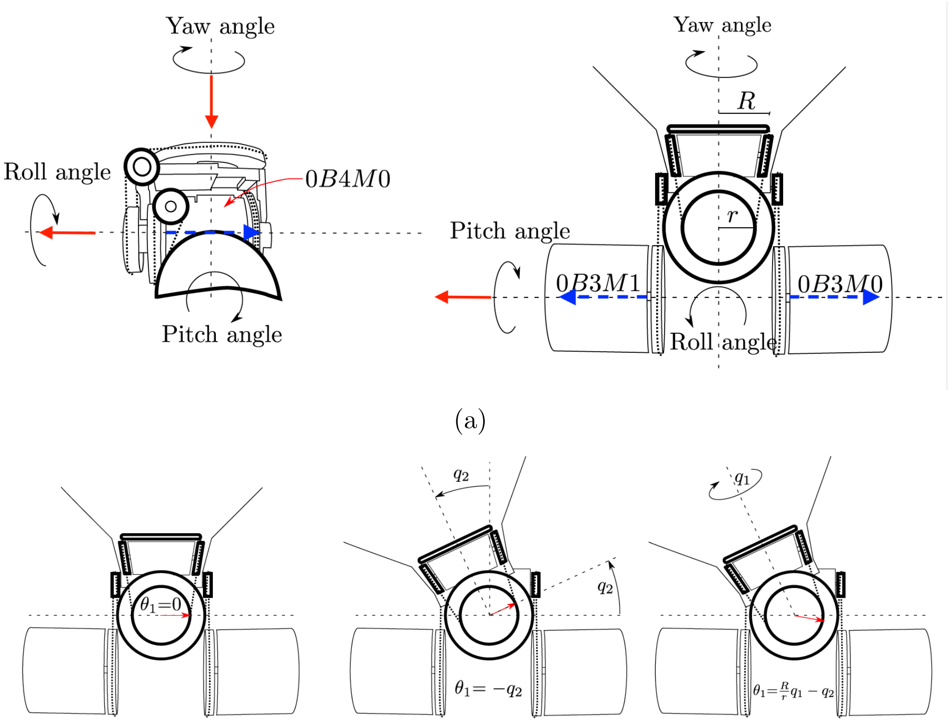
Differential Cable Drives

Power, velocity and torque conversion: Differential drives, when used, are commonly placed at the front end of a transmission chain for routing and coupling the transmission power from a set of motors to a set of joints. They can be found on iCub for coupling the waist yaw roll pitch joints to three motors 0B4M0, 0B3M0 and 0B3M1 as shown in the close-up 2.7 and illustrated in the functional sketch 2.21. The same principles apply here as for the reduction drives. We define the vectors of motor velocities $\omega_{\mathbf{m}}$ and torques τ_m as being the drive input, and the vectors of joint velocities $\omega_{\mathbf{j}}$ and torques τ_j as being the the drive output. The mapping between motor and joint velocities is defined through a bijective linear transformation $T : \omega_{\mathbf{m}} \in \mathbb{R}^N \mapsto \omega_{\mathbf{j}} \in \mathbb{R}^N$, as done in [Nori et al., 2015b, 3.3.3]. T can be expressed as an invertible coupling matrix defined as follows:

$$\omega_{\mathbf{j}} = T\,\omega_{\mathbf{m}}, \tag{2.22}$$

The identity between the input and output mechanical power still holds [Parmiggiani et al., 2009, section 3.3.3] [Nori et al., 2015b, section 3.3.3]:

$$\begin{aligned} & \tau_j^\top\,\omega_{\mathbf{j}} = \tau_m^\top\,\omega_{\mathbf{m}} \quad \forall \omega_{\mathbf{j}}, \omega_{\mathbf{m}} \\ \Longrightarrow\ & \tau_j^\top\,\omega_{\mathbf{j}} = \tau_m^\top\,T^{-1}\,\omega_{\mathbf{j}} \\ \Longleftrightarrow\ & \tau_j^\top = \tau_m^\top\,T^{-1} \\ \Longleftrightarrow\ & \boxed{\tau_j = T^{-\top}\tau_m}. \end{aligned} \tag{2.23}$$



(a)

(b)

Fig. 2.21 Sketch of the differential drive coupling the waist joints: the yaw, roll and pitch joints are actuated by the motors 0B4M0, 0B3M0,and 0B3M1 in differential configuration. The plain red arrows are the joint rotation axes, the dashed blue ones are the motor rotation axes (a). Motor and joint positions are respectively noted $\theta \in \mathbb{R}^3$ and $\mathbf{q} \in \mathbb{R}^3$. A roll rotation by an angle of q_2, with q_1 and q_3 unchanged, implies a rotation of 0B4M0 by $\theta_1 = -q_2$, 0B3M0 and 0B3M1 respectively by $\theta_2 = q_2$ and $\theta_3 = q_2$. A yaw rotation by q_1 implies an equal rotation of 0B4M0 modulo the transmission ratio (b). Source: [Nori et al., 2015b, Fig.5, 6].

Computation of coupling matrix T: In the case of the differential drive on the iCub's waist joints, let the velocity vectors be:

$$\begin{aligned}\omega_{\mathbf{m}} &= \begin{bmatrix}\theta_{\text{0B4M0}} & \theta_{\text{0B3M0}} & \theta_{\text{0B3M1}}\end{bmatrix} = \begin{bmatrix}\theta_1 & \theta_2 & \theta_3\end{bmatrix} \in \mathbb{R}^3,\\ \omega_{\mathbf{j}} &= \begin{bmatrix}torso_{yaw} & torso_{roll} & torso_{pitch}\end{bmatrix} = \begin{bmatrix}q_1 & q_2 & q_3\end{bmatrix} \in \mathbb{R}^3.\end{aligned}$$

We assume the geometry of the differential drive to be known accurately, more specifically the pulley diameters required for computing the transmission ratios: $R = 0.04\text{m}$ and $r = 0.022\text{m}$ as illustrated in the figure 2.21a. In this particular coupling configuration, it is easier to compute the inverse matrix T^{-1} than the direct transformation matrix T. We compute its columns one by one, by "moving" a single joint DoF q_i, keeping the others unchanged, and observing the impact on the motor angular positions. For instance, a roll rotation by an angle of q_2, with q_1 and q_3 unchanged, implies a rotation of 0B4M0 by $\theta_1 = -q_2$, 0B3M0 and 0B3M1 respectively by $\theta_2 = q_2$ and $\theta_3 = q_2$, as shown below:

$$\begin{bmatrix}\theta_1\\ \theta_2\\ \theta_3\end{bmatrix} = \begin{bmatrix}\times & -1 & \times\\ \times & 1 & \times\\ \times & 1 & \times\end{bmatrix}\begin{bmatrix}0\\ q_2\\ 0\end{bmatrix} \tag{2.24}$$

After building the same way the first and third columns, we get:

$$T^{-1} = \begin{bmatrix}\frac{R}{r} & -1 & 0\\ 0 & 1 & -1\\ 0 & 1 & 1\end{bmatrix}, \quad T = \begin{bmatrix}\frac{r}{R} & \frac{r}{2R} & \frac{r}{2R}\\ 0 & 0.5 & 0.5\\ 0 & -0.5 & 0.5\end{bmatrix}, \quad T^{-\top} = \begin{bmatrix}\frac{R}{r} & 0 & 0\\ -1 & 1 & 1\\ 0 & -1 & 1\end{bmatrix}. \tag{2.25}$$

Static friction and stiffness We will not analyze the effect of the cables stiffness or the friction in the differential drive, instead we will assume that effect can be modeled the same way we did for the harmonic drives model, only this time with higher static friction and lower stiffness.

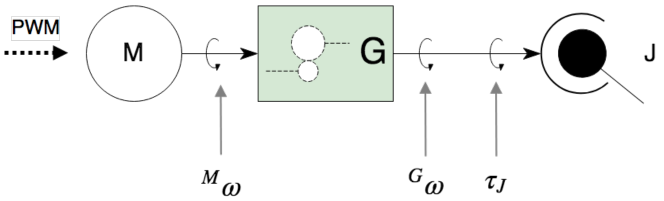


Fig. 2.22 Kinematic chain representing a single joint.

2.4 Methodology

2.4.1 Assembling the Joint Actuation Model

We consider a joint actuation chain composed of a single motor **M**, a gearbox (or harmonic drive) **G** , and a rotary joint, as shown in Fig 2.22. A PWM input voltage feeds the motor, and the actuation chain delivers a joint torque τ_J. For building the full joint transfer function mapping the input PWM to the output joint torque, we concatenate the sub-models described in the previous section—the motor model ((2.15)) and the harmonic drive model ((2.18), (2.19), (2.21))—and integrate them in the system Euler equation.

Angular momenta and Euler equation

We can write the Euler dynamics equation for the system, which equates the sum of torques applied on the system to the time derivative of the added angular momenta of the rotating elements:

$$\sum_{i=1}^{N} \tau_i = \sum_{k=1}^{M} \dot{h}_k \tag{2.26}$$

for N external torques applied to the system, and M rotating bodies. In this analysis we consider the child link rotating with the joint as an external element with respect to the joint actuation system. The motor rotor and the gearbox are the bodies with significant inertia worth accounting for. The child link rotating with the joint is treated as an external element with respect to the joint actuation system:

$$\tau_{\mathrm{PWM},m} + \tau_{f,g} + \tau_{f,\mathrm{J}} - \tau_{\mathrm{J}} = \dot{h}_m + \dot{h}_g \tag{2.27}$$

where I_m and I_g are respectively the rotational inertia of the motor and the gearbox projected on the rotation axis such that $I_m\ \dot{\omega}_g$ and $I_g\ \dot{\omega}_g$ are scalars. In the above equation, all the derivatives are defined with respect to the same inertial frame, and expressed on the same point of the axis G at the reduction drive output.

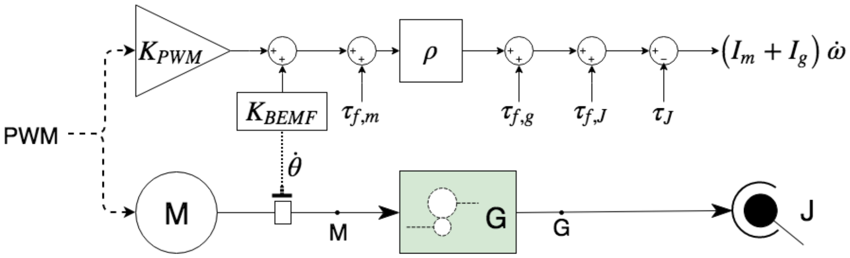


Fig. 2.23 Expanded single joint actuation model: parallel representation of the physical model and the transfer function PWM $\mapsto (I_m + I_g)\dot{\omega}$. The diagram shows the input parameter PWM, all the torques applied to the system and the resulting Angular Momentum. Since τ_J is an output torque, it appears as a negative value.

In view of (2.18), (2.19) and (2.21), we expand all the terms and rewrite the equation solving τ_J:

$$\tau_{\text{PWM},m} + \tau_{f,g} + \tau_{f,\text{J}} - (I_m + I_g)\,\dot{\omega}_{\text{J}} = \tau_{\text{J}} \tag{2.28}$$

$$\begin{aligned}\Longleftrightarrow & \rho k_{\text{PWM},\tau}\ \text{PWM} - \rho^2 k_{\text{BEMF},\tau}\,\omega_{\text{J}} + \rho\tau_{f,m} \\ & + \tau_{f,g} + \tau_{f,\text{J}} - \rho^2\,(I_m + I_g)\,\dot{\omega}_{\text{J}} = \tau_{\text{J}}\end{aligned} \tag{2.29}$$

Equivalent Motor PWM Torque

We define the *motor PWM torque* as the torque produced after the PWM input parameter and applied on the rotor, not accounting for any friction effect of mechanic or electric nature, i.e. the linear term $\rho K_{\text{PWM},\tau}$ PWM $=$ $K^*_{\text{PWM},\tau}$ PWM, $K^*_{\text{PWM},\tau}$ being the equivalent parameter of the [motor + harmonic drive] block:

$$K^*_{\text{PWM}} = \rho K_{\text{PWM},\tau} \tag{2.30}$$

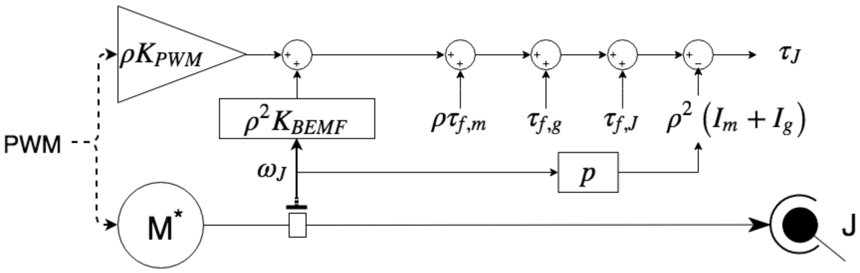


Fig. 2.24 Single joint actuation transfer function PWM ↦ τ_J in the Laplace domain. M^* represents the motor block grouping the brush-less motor and the harmonic drive.

Equivalent Friction model

We can see that a series of terms contribute to the overall friction torque applied to the shaft: the friction on the rotor bearings $\rho\tau_{f,m}$, the Back EMF torque $\rho^2 k_{\text{BEMF},\tau}\,\omega_\text{J}$, the friction in the reduction drive $\tau_{f,g}$, and the friction on the joint bearings $\tau_{f,\text{J}}$. The approach adopted in this methodology consists in grouping all these friction terms in a single friction model, which could be the Static/Coulomb/Viscous or the Stribeck friction model.

All the terms are grouped in a single Static friction term τ_s, a single Coulomb friction term τ_c and a single viscous friction term τ_v: all the friction on the bearings contributes to the the terms τ_c and τ_v, although we could consider them negligible with respect to the friction in the harmonic drive and the Back EMF; the Back EMF in particular is a linear function of ω_J and so has the properties of a viscous friction; the friction on the bearings should be negligible with respect to the other terms; the friction in the harmonic drive contributes to the three terms τ_s, τ_c and τ_v, with a significant static friction component as we will observe in the experiment results.

Assumption 2.1. *For professional grade bearings, or at least in the case of iCub brushless motors and joints bearings, the static friction on those bearings is negligible with respect with all the other friction terms and will be ignored.*

$$\begin{aligned}
\tau_c &= \tau_{c,m} + \tau_{c,g} + \tau_{c,\text{J}} = -K_c \operatorname{sign}(\omega_\text{J}) \\
\tau_v &= \tau_{v,m} + \tau_\text{BEMF} + \tau_{v,g} + \tau_{v,\text{J}} = -K_v\,\omega_\text{J} \\
\tau_s &= \tau_{s,g}(\omega_\text{J}, \sigma_s) \\
\tau_f(\omega_\text{J}, \sigma_s) &= \tau_c(\omega_\text{J}) + \tau_v(\omega_\text{J}) + \tau_s(\omega_\text{J}, \sigma_s),
\end{aligned} \tag{2.31}$$

Where $\tau_{c,\cdot}$, $\tau_{v,\cdot}$ and $\tau_{s,\cdot}$ relate to the nature of the friction—respectively Coulomb, viscous and static—and $\tau_{\cdot,m}$, $\tau_{\cdot,g}$, $\tau_{\cdot,\mathrm{J}}$ relate to the origin of the friction—respectively the motor bearings, harmonic drive (gearbox) and joint bearings. σ_s is the maximum static friction measured at zero velocity.

Identification of The Equivalent Model Parameters

In view of (2.29), (2.30) and (2.31), we can rewrite the output joint torque as:

$$ {}_{\mathrm{G}}\tau_{\mathrm{J}} = K^{*}_{\mathrm{PWM}}\ \mathrm{PWM} - \left({}_{\mathrm{G}}I_m + {}_{\mathrm{G}}I_g\right) {}^{\mathrm{G}}\dot{\omega} - K_c\ \mathrm{sign}({}^{\mathrm{G}}\omega) - K_v\ {}^{\mathrm{G}}\omega + \tau_s \quad (2.32)$$

The identification of the motor and gearbox inertial parameters $I_m + I_g$ is out of the scope of this thesis, so we assume they are given. We then need to identify the motor parameter K^{*}_{PWM} and the friction parameters τ_s, K_c and K_v.

The identification will be performed in two phases: the first phase identifies the viscous friction parameters τ_s, K_c and K_v; the second phase identifies the motor parameter K^{*}_{PWM}. This approach allows to simplify the fitting of the model parameters by reducing the problem dimension for each phase of the identification.

2.4.2 First phase - friction parameters identification

Symmetric Coulomb/viscous Friction Model

We initially only consider the Coulomb and viscous friction components, for describing the base estimation algorithm, and will later introduce the static friction component. We need to place the system in a condition where only the friction torques are present and acting on the shaft. This is achieved by setting the motor input PWM to zero, canceling the motor PWM torque and leaving only the motor internal mechanical friction and the counter torque due to the Back EMF. In normal operating conditions, the Back EMF adds a significant contribution to the overall joint actuation friction, and for this reason, it's crucial to account for that contribution in this phase of the identification. The Back EMF creates induced currents in the motor electromagnetic coils circuit [3] which result in a torque opposed to the motor PWM torque. Setting the PWM parameter as mentioned above, instead of turning off the motor, allows to keep the stator circuit closed and the Back

---

[3] located in the stator in the case of brushless motors

EMF current to flow. In view of (2.32) and PWM = 0, we express the joint torque as follows:

$$_{\mathrm{G}}\tau_{\mathrm{J}} + ({}_{\mathrm{G}}I_m + {}_{\mathrm{G}}I_g)\,{}^{\mathrm{G}}\dot{\omega} = -K_c\,\mathrm{sign}({}^{\mathrm{G}}\omega) - K_v\,{}^{\mathrm{G}}\omega \tag{2.33}$$

Where the joint velocity ${}^{\mathrm{G}}\omega$ and acceleration ${}^{\mathrm{G}}\dot{\omega}$ are respectively measured from joint encoders or inertial sensors. The joint torque ${}_{\mathrm{G}}\tau_J$ is measured by a joint torque sensor or estimated from Force-Torque sensors measurements and a modified inverse dynamics algorithm as seen in [Traversaro, 2017, Chapter 4 section 4.4.2]. We get the training data from a set of measurements, and then fit the model (2.33) by defining and solving the over-constrained linear system below:

$$X\,\Theta = \mathbf{y} \tag{2.34}$$

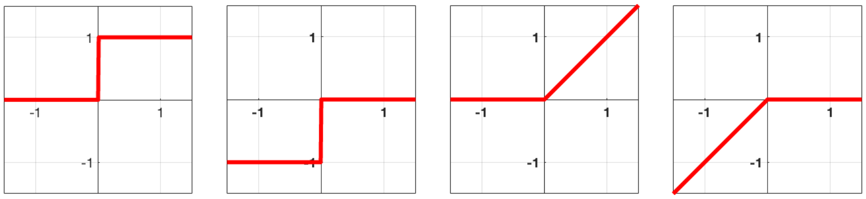
With,

$$\begin{aligned}
\mathbf{y} &= {}_{\mathrm{G}}\tau_{\mathrm{J}} + ({}_{\mathrm{G}}I_m + {}_{\mathrm{G}}I_g)\,{}^{\mathrm{G}}\dot{\mathbf{\Omega}} \\
X &= \begin{bmatrix} \mathrm{sign}(\omega_1) & \mathrm{sign}(\omega_2) & \dots & \mathrm{sign}(\omega_n) \\ \omega_1 & \omega_2 & \dots & \omega_n \end{bmatrix}^\top \\
\Theta &= \begin{bmatrix} -K_c & -K_v \end{bmatrix}^\top
\end{aligned} \tag{2.35}$$

where ω_i is the measured velocity at instant i, ${}^{\mathrm{G}}\dot{\mathbf{\Omega}}$ is the column vector of the joint angular acceleration measurement samples and ${}_{\mathrm{G}}\tau_{\mathrm{J}}$ is the column vector of the joint torque measurement samples. Obtaining the system (2.35) is straightforward as it's just the matrix formulation of (2.33). The linear least squares solution can be computed by inverting the matrix X through a Moore-Penrose left pseudo-inverse:

$$\Theta = \left(X^\top X\right)^{-1} X^\top\,\mathbf{y} \tag{2.36}$$

X is of full column rank since the second column is an evenly distributed set of velocities and so cannot be a multiple of the first column which is a series of ±1 elements. The matrix $X^\top X$ is well conditioned, assuming that we use a reasonable range of joint velocities and have a low level of noise in the encoder measurements. In any case, we verify numerically the condition number of that matrix in the experiments in section 2.5.



(a) (b) (c) (d)

Fig. 2.25 Selection step and ramp functions used for identifying the friction parameters in (2.37). **a:** $u^+(\omega)$, **b:** $u^-(\omega)$, **c:** $r^+(\omega)$, **d:** $r^-(\omega)$

Asymmetric Coulomb/viscous Friction Model with respect to the Rotation Direction

Depending on the properties of the gearbox or harmonic drive, the coupling pulleys for a cable driven transmission, we can consider the friction parameters to change with the rotation direction. We've adapted the linear system (2.35) to account for this property, by using a set of four unknowns K_{c-}, K_{v-}, K_{c+}, K_{v+} instead of two, and by using step and ramp functions instead of the sign functions, as shown in Fig. 2.25. We can now redefine the linear system as:

$$\begin{aligned} \mathbf{y} &= {}_{\mathrm{G}}\tau_{\mathrm{J}} + ({}_{\mathrm{G}}I_m + {}_{\mathrm{G}}I_g)\,{}^{\mathrm{G}}\dot{\boldsymbol{\Omega}} \\ X &= \begin{bmatrix} u^-(\omega_1) & u^-(\omega_2) & \dots & u^-(\omega_n) \\ r^-(\omega_1) & r^-(\omega_2) & \dots & r^-(\omega_n) \\ u^+(\omega_1) & u^+(\omega_2) & \dots & u^+(\omega_n) \\ r^+(\omega_1) & r^+(\omega_2) & \dots & r^+(\omega_n) \end{bmatrix}^\top \\ \Theta &= \begin{bmatrix} -K_{c-} & -K_{v-} & -K_{c+} & -K_{v+} \end{bmatrix}^\top \end{aligned} \tag{2.37}$$

Proof. We will show here that the system (2.37) is equivalent to writing (2.33) for every measurement pair $\{\omega_i, {}_{\mathrm{G}}\tau_{\mathrm{J},i}\}$, $\forall i \in [0, N]$, N being the number of acquired samples, but applying K_{c-}, K_{v-} for $\omega_i < 0$ and K_{c+}, K_{v+}

for $\omega_i > 0$:

$$\begin{aligned}(2.37) \iff \forall i \in & [0, N],\\ \mathbf{y}_i = & - K_{c-}\, u^{-}(\omega_i) - K_{v-}\, r^{-}(\omega_i)\\ & - K_{c+}\, u^{+}(\omega_i) - K_{v+}\, r^{+}(\omega_i)\\ = & \begin{cases} -K_{c-}\,\mathrm{sign}(\omega_i) - K_{v-}\,\omega_i, & \omega_i < 0\\ -K_{c+}\,\mathrm{sign}(\omega_i) - K_{v+}\,\omega_i, & \omega_i > 0 \end{cases}\end{aligned} \tag{2.38}$$

$\square$

This improvement can be systematically applied, adding a minor computation cost, and resulting in a more robust estimation. It is fully compatible with the estimation procedure on coupled joints 2.4.4.

Stribeck/Coulomb/viscous friction model

If the static friction in the reduction drive is significant, for instance around 2 N m or higher, the results obtained with the models proposed above will not result in an accurate fitting, giving a good approximation at higher velocities, and a poor approximation near zero velocities, where the friction torque curve appears to be quite non linear, and right before the state transitions to a "stick" phase, as we will observe in the experiment results (2.5.1). The non linearity and the unaccounted effects of the "slip-stick" phase transitions are aggravated when the joint reaches or crosses the zero velocity.

For that reason we now integrate in the estimation algorithm the model combining the Coulomb and Viscous friction components with the Stribeck effect $\tau_s(\omega)$ Van Geffen [2009] Harnoy et al. [2008]. The Stribeck effect is modeled here with an empiric non linear function, but still linear with respect to the fitting parameters. The function we are looking for has to approximate the torque response in the plot 2.29, which strongly resembles the Stribeck model depicted in 2.9 and analyzed closer in this section. For that and in view of (2.31), the overall friction function $\tau_f(\omega)$ has to meet the following properties:

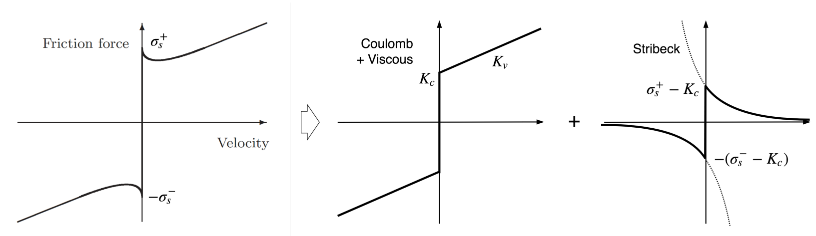


Fig. 2.26 Decomposition of Stribeck model into Coulomb/Viscous + exponential function.

$$\lim_{\omega \to 0^-} \tau_f(\omega) = \sigma_s^- \tag{2.39}$$

$$\lim_{\omega \to 0^+} \tau_f(\omega) = \sigma_s^+ \tag{2.40}$$

$$\lim_{\omega \to +\infty} \tau_f(\omega) = \tau_c(\omega) + \tau_v(\omega) \tag{2.41}$$

$$\lim_{\omega \to -\infty} \tau_f(\omega) = \tau_c(\omega) + \tau_v(\omega) \tag{2.42}$$

By applying (2.31) in each of the above equations we get:

$$\lim_{\omega \to 0^-} \tau_s(\omega) = \sigma_s^- - \tau_c(\omega) - \tau_v(\omega) = \sigma_s^- - \tau_c(\omega) \tag{2.43}$$

$$\lim_{\omega \to 0^+} \tau_s(\omega) = \sigma_s^+ - \tau_c(\omega) \tag{2.44}$$

$$\lim_{\omega \to \infty} \tau_s(\omega) = 0 \tag{2.45}$$

We could have chosen for $\tau_s(\omega)$ an hyperbolic function $\propto \frac{1}{\omega^n}$ but such function would have a singularity at $\omega = 0$ that could be avoided with the shifted function: $\tau_s(\omega) \propto \frac{1}{(\omega - \omega_0)^n}$, only this introduces a new parameter ω_0 with respect to which τ_s would be non-linear. The good candidate we found is an exponential function linearly parameterized. Let's consider the function $g(x, \sigma) = (\sigma - K_c)\, e^{-x}$. $\tau_s(\omega)$ is symmetric with respect to the origin, coincides with $g(\omega, \sigma_s^+)$ in the half-plane $\omega > 0$, and coincides with $-g(-\omega, \sigma_s^-)$ in the half-plane $\omega < 0$, as illustrated in figure 2.26. We can then write:

$$\begin{aligned}\tau_s(\omega) &= u^+(\omega)\left(\sigma_s^+ - K_c\right)\, e^{-\omega} - u^+(-\omega)\left(\sigma_s^- - K_c\right)\, e^{\omega}\\ &= -\left(u^+(\omega)\, e^{-\omega} + u^-(\omega)\, e^{\omega}\right) K_c\\ &\quad + u^+(\omega)\, e^{-\omega}\, \sigma_s^+\\ &\quad + u^-(\omega)\, e^{\omega}\, \sigma_s^-\end{aligned} \tag{2.46}$$

where u^+ and u^- are the step functions depicted in figure 2.25 (a) and (b). So the overall friction can be posed as the linear system:

$$\begin{aligned}\mathbf{y} &= {}_{\mathrm{G}}\tau_{\mathbf{J}} + \left({}_{\mathrm{G}}I_m + {}_{\mathrm{G}}I_g\right)\dot{\mathbf{\Omega}}_{\mathrm{G}}\\ X &= X_1 + X_2\\ \Theta &= \begin{bmatrix}-K_c & -K_v & -\sigma_s^+ & -\sigma_s^-\end{bmatrix}^\top,\end{aligned} \tag{2.47}$$

with X_1 and X_2 defined as follows:

$$\begin{aligned}X_1 &= \begin{bmatrix} u^+(\omega_1)\left(1+e^{-\omega_1}\right) & \dots & u^+(\omega_n)\left(1+e^{-\omega_n}\right)\\ r^+(\omega_1) & \dots & r^+(\omega_n)\\ u^+(\omega_1)\, e^{-\omega_1} & \dots & u^+(\omega_n)\, e^{-\omega_n}\\ 0 & \dots & 0\end{bmatrix}^\top\\ &= \operatorname{diag}\left(u^+(\omega_1)\dots u^+(\omega_n)\right)\begin{bmatrix}\left(1+e^{-\omega_1}\right) & \dots & \left(1+e^{-\omega_n}\right)\\ \omega_1 & \dots & \omega_n\\ e^{-\omega_1} & \dots & e^{-\omega_n}\\ 0 & \dots & 0\end{bmatrix}^\top\end{aligned} \tag{2.48}$$

$$\begin{aligned}X_2 &= \begin{bmatrix} u^-(\omega_1)\left(1+e^{\omega_1}\right) & \dots & u^-(\omega_n)\left(1+e^{\omega_n}\right)\\ r^-(\omega_1) & \dots & r^-(\omega_n)\\ 0 & \dots & 0\\ u^-(\omega_1)\, e^{\omega_1} & \dots & u^-(\omega_n)\, e^{\omega_n}\end{bmatrix}^\top\\ &= \operatorname{diag}\left(u^-(\omega_1)\dots u^-(\omega_n)\right)\begin{bmatrix}\left(1+e^{\omega_1}\right) & \dots & \left(1+e^{\omega_n}\right)\\ \omega_1 & \dots & \omega_n\\ e^{\omega_1} & \dots & e^{\omega_n}\\ 0 & \dots & 0\end{bmatrix}^\top\end{aligned} \tag{2.49}$$

2.4.3 Second phase - Motor Parameter Identification

Algorithm The friction coefficients having been identified, we can now use them in the joint dynamics equation (2.32) for identifying K^*_{PWM}. The approach is then equivalent, defining a system correlating this time the input duty cycle PWM and the output joint torque τ_{J}, where K^*_{PWM} is the only unknown parameter. From (2.32) we can write:

$$X\,\Theta = \mathbf{y} \tag{2.50}$$

With,

$$\begin{aligned}
&\mathbf{y} = {}_{\mathrm{G}}\tau_{\mathrm{J}} + ({}_{\mathrm{G}}I_m + {}_{\mathrm{G}}I_g)\,{}^{\mathrm{G}}\dot{\boldsymbol{\Omega}} + K_c\,\mathrm{sign}({}^{\mathrm{G}}\boldsymbol{\Omega}) + K_v\,{}^{\mathrm{G}}\boldsymbol{\Omega} - {}_{\mathrm{G}}\tau_{\mathbf{s}}\\
&X = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \mathrm{PWM}_1 & \mathrm{PWM}_2 & \dots & \mathrm{PWM}_n \end{bmatrix}^\top\\
&\Theta = \begin{bmatrix} \tau_0 & K^*_{\mathrm{PWM}} \end{bmatrix}^\top,
\end{aligned} \tag{2.51}$$

where $\mathrm{sign}({}^{\mathrm{G}}\boldsymbol{\Omega}) = \begin{bmatrix}\mathrm{sign}(\omega_1)\dots\mathrm{sign}(\omega_n)\end{bmatrix}^\top$. τ_0 accounts for an eventual offset current in the stator windings. A null motor current and torque could then be obtained from an input value $\mathrm{PWM} = -K^{-1}_{\mathrm{PWM}}\,\tau_0$. The system is solved as the previous one, through the computation of a Moore-Penrose left pseudo-inverse.

Note. *As done for the friction parameters identification in 2.4.2, we should verify that X is well conditioned. Again, X is of full column rank since the second column is an evenly distributed set of PWM setpoint values and so cannot be a multiple of the first column. The matrix $X^\top X$ is well conditioned, assuming that we use a reasonable range of PWM values with respect to the noise level of the $\mathbf{y}$ measurements.*

2.4.4 The Coupled Joints Case

The kinematic and dynamic coupling implemented by the differential drives has to be accounted for when correlating joint with motor torques. this aspect significantly impacts the measurement procedures and the estimation results. Let us consider three joints J_1, J_2, J_3 actuated by three motors M_a, M_b, M_c through a differential coupling. We depict in Fig. 2.27 the respective actuation system.

Velocities and torques transformation

We've seen in section 2.3.4 how to transform velocities (2.22) and torques (2.23) between the input and the output of a coupling system like the differential drive on the torso. If we apply (2.22) and (2.23) to the quantities illustrated in the figure 2.27, we get:

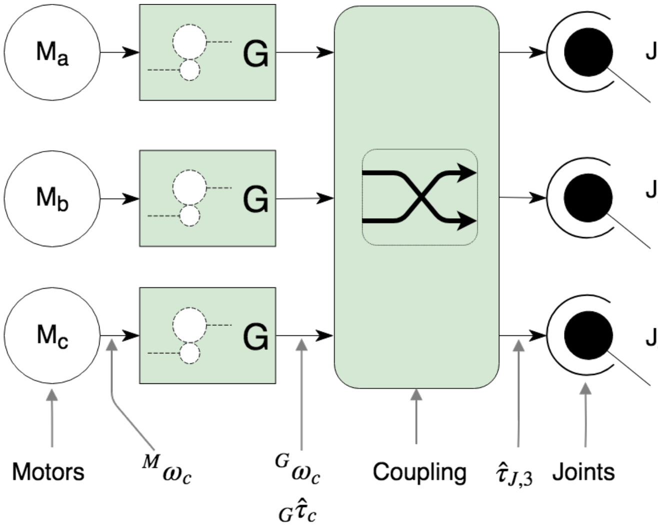


Fig. 2.27 Kinematic chain representing three coupled joints, the gearboxes and motors. $^{\mathrm{M}}\omega_c$, $^{\mathrm{G}}\omega_c$ and $_{\mathrm{G}}\tau_c$ are respectively shaft angular velocities and the measured or estimated torque on the motor C transmission chain. $\tau_{\mathrm{J},3}$ is the output joint torque estimated on the joint J_3.

$$\bar{\omega}_{\mathrm{J}} = T\,{}^{\mathrm{G}}\bar{\omega}, \quad \text{with} \quad \bar{\omega}_{\mathrm{J}} = \begin{bmatrix} \omega_{\mathrm{J},1} \\ \omega_{\mathrm{J},2} \\ \omega_{\mathrm{J},3} \end{bmatrix}, \quad {}^{\mathrm{G}}\bar{\omega} = \begin{bmatrix} {}^{\mathrm{G}}\omega_a \\ {}^{\mathrm{G}}\omega_b \\ {}^{\mathrm{G}}\omega_c \end{bmatrix}, \tag{2.52}$$

$$\dot{\bar{\omega}}_{\mathrm{J}} = T\,{}^{\mathrm{G}}\dot{\bar{\omega}}, \tag{2.53}$$

$$\bar{\tau}_{\mathrm{J}} = T^{-\top}\,{}_{\mathrm{G}}\bar{\tau}, \quad \text{with} \quad \bar{\tau}_{\mathrm{J}} = \begin{bmatrix} \tau_{\mathrm{J},1} \\ \tau_{\mathrm{J},2} \\ \tau_{\mathrm{J},3} \end{bmatrix}, \quad {}_{\mathrm{G}}\bar{\tau} = \begin{bmatrix} {}_{\mathrm{G}}\tau_a \\ {}_{\mathrm{G}}\tau_b \\ {}_{\mathrm{G}}\tau_c \end{bmatrix}, \tag{2.54}$$

Identification Joint Wise or Motor Wise

Regarding the algorithms proposed in this section, like (2.35) and (2.51), we have the choice between two methods: either we express all the quantities at the joint level and we perform an identification joint wise; either we express them at the point G of each motor group shaft (a), (b) and (c), and we perform the identification motor wise, applying the algorithms independently on each motor group M_a^*, M_b^*, M_c^*. The first approach seems simpler from an execution point of view, assuming it is easier to constrain the coupled joints motion to one joint at a time, but more complex regarding the identification system. The second method simplifies the identification algorithm but it requires a more elaborated external actuation of the joints for moving the analyzed motor in the desired range or following a desired approximate velocity. We first need to verify how the identification algorithm unfolds for each method.

Joint wise identification each joint "sees" a virtual motor resulting from its interaction with the three coupled motors. During the data acquisition, each joint should be moved alone, which results in the coupled motors potentially all moving simultaneously, depending on the terms of the coupling matrix T redefined in (2.52) and (2.54). For instance, in the case of joint 1 we would get:

$$
\begin{aligned}
{}^{\mathrm{G}}\bar{\omega} &= T^{-1}\,\bar{\omega}_{\mathrm{J}} = \begin{bmatrix} (T^{-1})_{1,1} & \times & \times \\ (T^{-1})_{2,1} & \times & \times \\ (T^{-1})_{3,1} & \times & \times \end{bmatrix} \begin{bmatrix} \omega_{\mathrm{J},1} & 0 & 0 \end{bmatrix}^\top, \\
{}_{\mathrm{G}}\bar{\tau} &= T^\top \begin{bmatrix} \tau_{\mathrm{J},1} & 0 & 0 \end{bmatrix}^\top,
\end{aligned}
$$

In view of (2.52) and (2.54), we rewrite (2.32) as follows:

$$
\bar{\tau}_{\mathrm{J}} = T^{-\top} \left(\mathbf{K}^*_{\mathrm{PWM}}\, \overline{\mathrm{PWM}} - {}_{\mathrm{G}}\mathbf{I}\; {}^{\mathrm{G}}\dot{\bar{\omega}} - \mathbf{K_c}\, \mathrm{sign}({}^{\mathrm{G}}\bar{\omega}) - \mathbf{K_v}\, {}^{\mathrm{G}}\bar{\omega} + \bar{\tau}_s \right), \qquad (2.55)
$$

where $\mathbf{K}^*_{\mathrm{PWM}}$, $\mathbf{K_c}$, $\mathbf{K_v}$ and ${}_{\mathrm{G}}\mathbf{I}$ are respectively the diagonal matrices of the motor groups torque, Coulomb and viscous friction coefficients and inertia, and $\overline{\mathrm{PWM}}$, $\bar{\tau}_s$ are respectively the vectors of motor groups inputs and static friction:

$$
\begin{aligned}
\mathbf{K}^*_{\mathrm{PWM}} &= \begin{bmatrix} K^*_{\mathrm{PWM},a} & 0 & 0 \\ 0 & K^*_{\mathrm{PWM},b} & 0 \\ 0 & 0 & K^*_{\mathrm{PWM},c} \end{bmatrix}, \quad \mathbf{K_c} = \begin{bmatrix} K_{c,a} & 0 & 0 \\ 0 & K_{c,b} & 0 \\ 0 & 0 & K_{c,c} \end{bmatrix}, \\
\mathbf{K_v} &= \begin{bmatrix} K_{v,a} & 0 & 0 \\ 0 & K_{v,b} & 0 \\ 0 & 0 & K_{v,c} \end{bmatrix}, \quad \overline{\mathrm{PWM}} = \begin{bmatrix} \mathrm{PWM}_a \\ \mathrm{PWM}_b \\ \mathrm{PWM}_c \end{bmatrix}, \quad \bar{\tau}_s = \begin{bmatrix} \tau_{s,a} \\ \tau_{s,b} \\ \tau_{s,c} \end{bmatrix}, \\
\mathrm{sign}({}^{\mathrm{G}}\bar{\omega}) &= \begin{bmatrix} \mathrm{sign}({}^{\mathrm{G}}\omega_a) & \mathrm{sign}({}^{\mathrm{G}}\omega_b) & \mathrm{sign}({}^{\mathrm{G}}\omega_c) \end{bmatrix}^\top,
\end{aligned}
$$

For identifying the friction parameters as done in the previous sections, we set $\overline{\mathrm{PWM}} = 0_{3\times 1}$. In view of (2.55), we can then write the joint torques as follows:

$$
\begin{aligned}
\bar{\tau}_{\mathrm{J}} + T^{-\top}\,{}_{\mathrm{G}}\mathbf{I}\,T^{-1}\,\dot{\bar{\omega}}_{\mathrm{J}} &= T^{-\top}\left(-\mathbf{K_c}\,\mathrm{sign}({}^{\mathrm{G}}\bar{\omega}) - \mathbf{K_v}\,{}^{\mathrm{G}}\bar{\omega} + \bar{\tau}_s\right), \\
&= -T^{-\top}\,\mathbf{K_c}\,\mathrm{sign}(T^{-1}\,\bar{\omega}_{\mathrm{J}}) - T^{-\top}\,\mathbf{K_v}\,T^{-1}\,\bar{\omega}_{\mathrm{J}}, \qquad (2.56)
\end{aligned}
$$

so we get the composite friction coefficients $\tilde{\mathbf{K}}_\mathbf{c}$ and $\tilde{\mathbf{K}}_\mathbf{v}$ such that:

$$\tilde{\mathbf{K}}_\mathbf{c} = T^{-\top}\,\mathbf{K_c}, \qquad (2.57)$$

$$\tilde{\mathbf{K}}_\mathbf{v} = T^{-\top}\,\mathbf{K_v}\,T^{-1}. \qquad (2.58)$$

The matrices $\tilde{\mathbf{K}}_\mathbf{c}$ and $\tilde{\mathbf{K}}_\mathbf{v}$ are not diagonal and we end up with a higher dimension problem, tripling the number of terms to identify (eighteen instead of six). The same way we built the identification linear system (2.35) from (2.33), we build the identification system for a single joint k from the k^{th} line of (2.56) as follows:

$$
\begin{aligned}
\mathbf{y} &= \tau_{\mathrm{J},\mathbf{k}} + \dot{\mathbf{\Omega}}_{\mathrm{J}}\ \mathrm{col}_k\left(T^{-\top}\,{}_{\mathrm{G}}\mathbf{I}\,T^{-1}\right) \\
X &= \begin{bmatrix} \mathrm{sign}(T^{-1}\,\bar{\omega}_{\mathrm{J},1}) & \mathrm{sign}(T^{-1}\,\bar{\omega}_{\mathrm{J},2}) & \dots & \mathrm{sign}(T^{-1}\,\bar{\omega}_{\mathrm{J},n}) \\ \bar{\omega}_{\mathrm{J},1} & \bar{\omega}_{\mathrm{J},2} & \dots & \bar{\omega}_{\mathrm{J},n} \end{bmatrix}^{\top} \qquad (2.59) \\
\Theta &= \begin{bmatrix} \mathrm{row}_k\left(-\tilde{K}_c\right) & \mathrm{row}_k\left(-\tilde{K}_v\right) \end{bmatrix}^{\top}
\end{aligned}
$$

where $\bar{\omega}_{\mathrm{J},n}$ is the vector of all three joints measured velocities at instant n, $\dot{\mathbf{\Omega}}_{\mathrm{J}}$ is the $n \times 3$ matrix of measured angular accelerations from all the three coupled joints, $\tau_{\mathrm{J},\mathbf{k}}$ is the column vector of the joint k torque measured samples, $\mathrm{col}_k(\cdot)$ and $\mathrm{row}_k(\cdot)$ are respectively the k^{th} column and k^{th} row of the argument matrix. The linear least squares solution can be computed the same way as for (2.35).

Identifying the $\tilde{\mathbf{K}}_\mathbf{c}$ and $\tilde{\mathbf{K}}_\mathbf{v}$ terms In order to identify Θ, relative to joint k, we have to move all the joints and get a reasonable range of velocity measurements (and accelerations if we do not neglect the inertial terms) for each joint. If only one joint is moved, like for instance joint j, and the remaining two are constrained in a fixed position, we obtain the velocity to torque curve characterizing the cross-gains $\tilde{K}_{v,kj}$ and $\tilde{K}_{c,kj}$ which define how the motion of joint j affects the torque measured on joint k.

Note. *Even if a single joint is being moved using an external force (moving manually the child link of the joint), reaction torques can appear on the constrained still joints, due to the multiple coupled friction torques coming from the motor groups.*

This analysis highlights the fact that the dynamics of the moved joint depends on the dynamics of all three motor groups. The model identified for a specific joint will inherit the combined uncertainties of the respective motor groups parameters, typically due to slip/stick phase transitions and other dynamics behaviors not accounted in the motor group models.

Motor wise identification This second approach is simpler: we first transform all the joint velocities and torque measurements $\bar{\omega}_{\mathrm{J}}$ and $\bar{\tau}_{\mathrm{J}}$, into the respective transformed quantities ${}^{\mathrm{G}}\bar{\omega}$ and ${}^{\mathrm{G}}\bar{\tau}$ expressed in the point G of the motor groups shafts (a), (b) and (c), as per (2.52) and (2.54) and as illustrated in Figure 2.27. We then just have to apply to each motor separately the dynamics equation (2.32), replacing ${}_{\mathrm{G}}\tau_J$ by ${}_{\mathrm{G}}\tau_x$ and ω_{G} by ${}^{\mathrm{G}}\omega_{g,x}$, x being the motor a, b or c selected for identification. At last we solve the equation for the friction and motor parameters following the algorithms proposed in 2.4.2 and 2.4.3, ignoring the coupling. This converts the problem into three independent problems, each on a single motor actuation chain.

For instance, in the case of motor M_c we would use ${}^{\mathrm{G}}\omega_c$ and ${}_{\mathrm{G}}\tau_c$ in the system for identifying the friction as follows:

$$
\begin{aligned}
\mathbf{y} &= {}_{\mathrm{G}}\tau_{\mathbf{c}} + ({}_{\mathrm{G}}I_{m,c} + {}_{\mathrm{G}}I_{g,c})\,{}^{\mathrm{G}}\dot{\mathbf{\Omega}}_{\mathbf{c}} \\
X &= \begin{bmatrix} \mathrm{sign}({}^{\mathrm{G}}\omega_{c,1}) & \mathrm{sign}({}^{\mathrm{G}}\omega_{c,2}) & \dots & \mathrm{sign}({}^{\mathrm{G}}\omega_{c,n}) \\ {}^{\mathrm{G}}\omega_{c,1} & {}^{\mathrm{G}}\omega_{c,2} & \dots & {}^{\mathrm{G}}\omega_{c,n} \end{bmatrix}^{\top} \\
\Theta &= \begin{bmatrix} -K_{c,c} & -K_{v,c} \end{bmatrix}^{\top}
\end{aligned}
\tag{2.60}
$$

Where ${}^{\mathrm{G}}\dot{\mathbf{\Omega}}_{\mathbf{c}}$ is the column vector of the motor M_c angular acceleration time series, measured or computed using (2.53).

This method requires to block the remaining motors not targeted for identification, in this case, M_a and M_b, in order to properly drive the analyzed motor (M_a) rotation through an external force applied on the coupled joints.

We followed the second method, which presents the following benefits: it is less sensitive to model uncertainties that can arise from bad estimates of the coupling reduction ratios and the cables elongation in case of a cable driven differential; it allows to track unexpected slip/stick phase transitions separately for each of the three motor actuation chains; whatever the complexity of the coupling it will be easier to cover a desired range of velocities on each motor; it allows to leave the identification of the coupling model to a separate task.

Note. *In platforms having Field Oriented Control motor drives, the motor velocity can be estimated from the measurements of a high resolution encoder placed on the motor stator, which benefits from much higher accuracy compared to the typical three hall-effect sensors used in motors driven by a Trapezoidal Control.*

2.5 Experimental Procedure and Results

We have tested the estimation algorithms proposed in section 2.4, on the robot iCub, which actuation system and sensors framework were presented in section 2.2. We have estimated the joint friction and motor parameters on the right leg knee and on the coupled joints—yaw, roll, pitch—connecting the waist to the torso. The sensors data were captured at 100Hz sampling rate: joint positions, velocities, torque and motor velocities.

2.5.1 First Phase - Friction Parameters Identification

Procedure

After setting the PWM to zero, we apply an external torque to the joint in an oscillating motion [4]. It is preferred to avoid significant accelerations in order to minimize and eventually neglect the inertial terms, otherwise we use the inertial sensors measurements for estimating and accounting for the inertial terms. We then apply the method described in 2.4.2 on the acquired data. When applying the model, we verified numerically that the matrix $X^\top X$ is well conditioned, which is a requirement for the pseudo-inverse computation to give an accurate solution.

Experiment Results

We present here the plots from the experiments on the friction estimation on two single joints from the right leg, free of any coupling or cable driven actuation: the right knee (2.29), the right hip roll (2.30a, 2.30b).

---

[4] This can be done manually or with an external actuation.

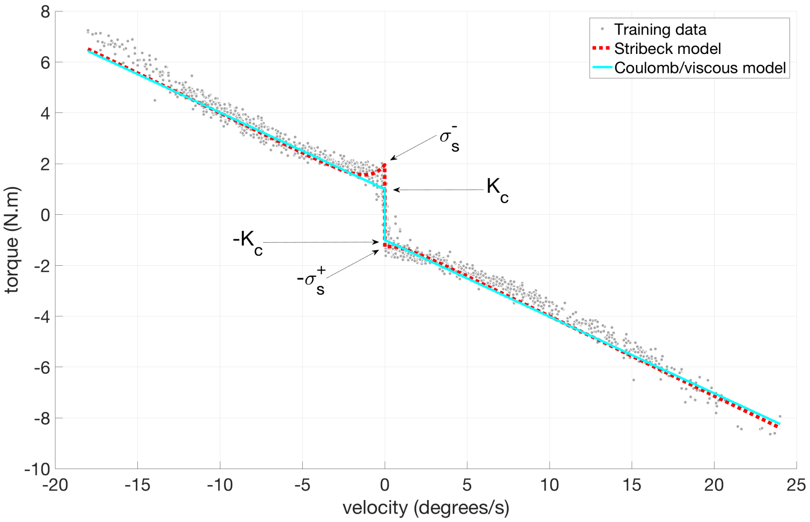


Fig. 2.28 Friction model fitting results on the right leg knee. Grey dots: the acquired samples. Plain cyan: Coulomb/Viscous model fitted with the method proposed in 2.4.2. dashed red line: Stribeck model. The identified parameters are: $K_c \sim 1\mathrm{N\,m}$, $K_v \sim 0.30\mathrm{Nms/\,deg}$.

After testing the procedure with the asymmetric model proposed in 2.4.2, we verified that the friction (K_c, K_v) in the transmission system was rather symmetric at least for the Coulomb and viscous friction parameters. We then tried the symmetric Coulomb/Viscous method 2.4.2 on the same data.

As we can observe in the plot 2.28, the model (cyan color line) doesn't seem to fit well the data near the zero velocity: while at zero velocity the model gives a Coulomb friction $K_c \sim 1\mathrm{N\,m}$, the measured torque is twice as high, rising up to $\sim 2\mathrm{N\,m}$.

Later tests on most of the joints on the right leg of iCub—hip pitch/roll/yaw, knee—revealed the same consistent increase of friction when the joint velocity decreases to zero, similar to the Stribeck effect, as can be observed in all the plots of friction torque in this section. As explained in 2.3.2, at very low speeds, when the surfaces are in solid-to-solid contact, we are in a regime called "Boundary Lubrication", the sheared solid surfaces being the boundary lubricants. For a given extension of that regime and viscous friction coefficient, we can get a friction curve as the one observed on the leg joints: when the angular velocity increases, starting from zero, the friction keeps a constant value equal to the identified static friction σ_s ("Boundary

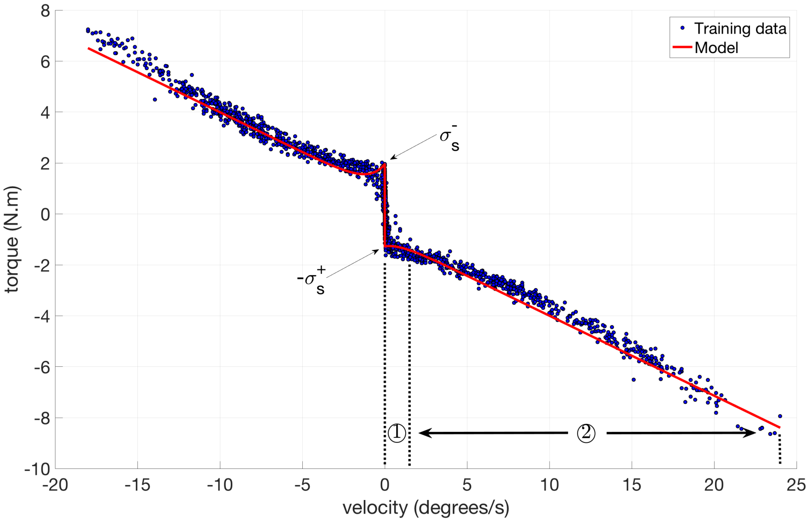


Fig. 2.29 Friction model fitting results on the right leg knee. Blue dots: the acquired samples. Plain red: Stribeck/Coulomb/Viscous model fitted with the method proposed in 2.4.2. The identified parameters are: static frictions $\sigma_s^+ \sim 1.27\mathrm{N\,m}$, $\sigma_s^- \sim 1.95\mathrm{N\,m}$; $K_c \sim 0.85\mathrm{N\,m}$; $K_v \sim 0.31\mathrm{Nms/\,deg}$. ①: Boundary Lubrication regime, ②: Full Fluid Lubrication regime (viscous friction).

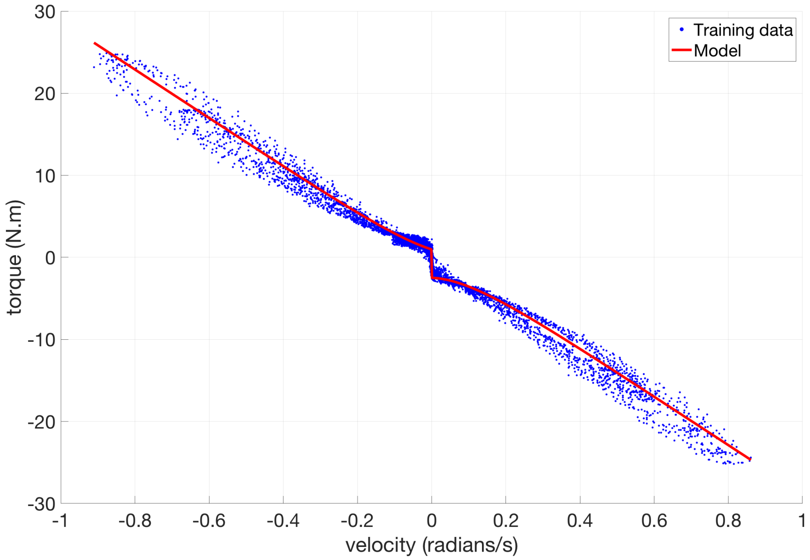
Lubrication") until the viscous friction becomes dominant.

Figure 2.29 illustrates the Stribeck model fitted on the same data as 2.28. Unlike the Coulomb/Viscous model, the Stribeck model gives a good estimate of σ_s^+ and σ_s^-. Although we get approximately the same coefficient K_v, K_c differs significantly between the two models.

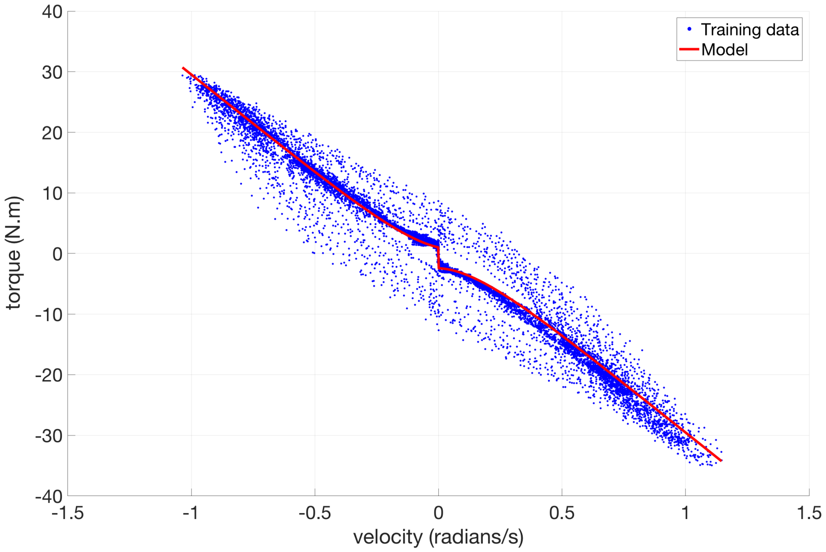
A better fit of the "Boundary Lubrication": The constraint on the derivative of the model function at zero velocity, $[\delta\tau_f/\delta\dot{q}]_{\dot{q}=0} = 0$, ensures that we better fit the "Boundary Lubrication" plateau. we can observe the fitting improvement in figure 2.29, where we applied the constraint on the right derivative, i.e. $[\delta\tau_f/\delta\dot{q}]_{\dot{q}=0^+}$.

Further tests on the right leg hip roll provided comparable results, as illustrated in 2.30a and 2.30b.

In 2.30a the motion was slow when changing direction. We can observe the same Boundary Lubrication around zero velocity as on the knee joint. The identified parameters are: $\sigma_s^+ \sim 2.49\mathrm{N\,m}$, $\sigma_s^- \sim 0.96\mathrm{N\,m}$; $K_c \sim$



(a) Slow change of direction.



(b) Fast change of direction.

Fig. 2.30 Friction model fitting results on right leg hip roll. (a) Slow change of direction. The “Boundary Lubrication” is visible around zero velocity. (b) Fast change of direction (higher accelerations). Results in additional concentric hysteresis cycles.

Joint	σ_s^+ (N m)	σ_s^- (N m)	K_v (Nms/ deg)
right hip pitch	1.31	2.24	1.16
right hip roll	2.49	0.96	0.50
right hip yaw	1.30	2.58	1.22
right knee	1.27	1.95	0.69
right ankle pitch	2.90	0.71	1.32
right ankle roll	2.05	1.29	1.20

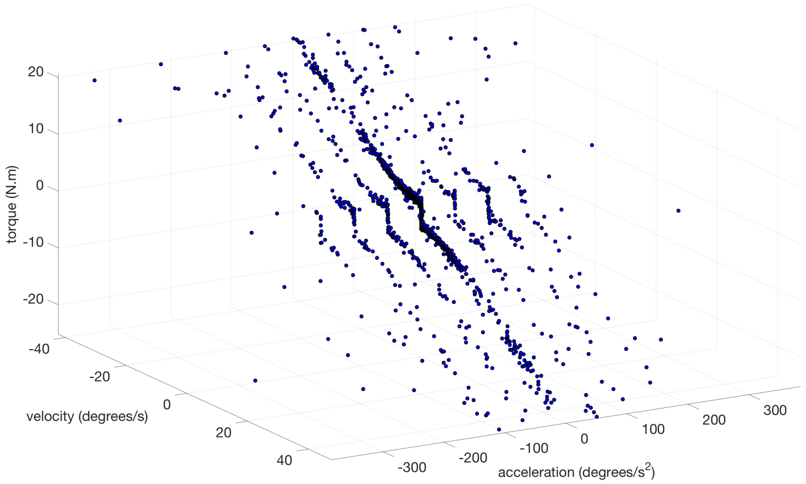
Table 2.1 Friction parameters extimation results on the iCub right leg joints.

0.76N m; $K_v \sim 0.50$Nms/ deg. A higher acceleration motion results in "outlier" samples sitting away from the fitted friction model, in concentric hysteresis cycles. As mentioned in 2.3.4, this can be due to the stiffness of the flex-spline in the Harmonic drive. A spring or a flexible element characterized by a stiffness will transform an input torque like a low pass filter, introducing delay and thus an hysteresis response.

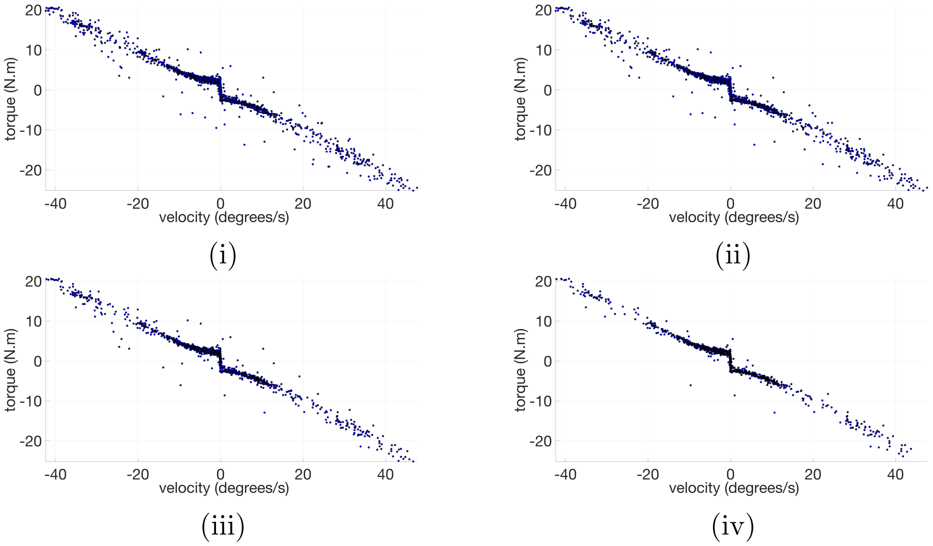
We have run a friction estimation trial on all the right leg joints, listed in the table 2.1. These results haven't been validated yet with the low level controller. On top of that, the present low level controller doesn't integrate yet static nor Coulomb friction compensation.

Using the angular acceleration measurement for removing outliers

As we will see in the experiments done on the right leg hip roll, the friction torque measurements present a significant number of outliers that lower the identifiability of the friction model parameters. We propose here a method for efficiently filtering those outliers, while making the training data acquisition process more reliable. We use the rotation acceleration measurement as a criterion to discard the samples which are not eligible for the static friction fitting. Actually, the hysteresis observed in 2.30b appears and is aggravated as soon as we cross the zero velocity with ever greater acceleration. A simple first approach to avoid that condition would be to discard the samples acquired for which the angular acceleration is above a given threshold, as illustrated in 2.31.



(a) 3-D view with angular acceleration axis.



(i)

(ii)

(iii)

(iv)

(b) Progressive filtering of the outliers by applying an acceleration upper threshold of $200\,\mathrm{deg}\,/\mathrm{s}^2$ (ii), $100\,\mathrm{deg}\,/\mathrm{s}^2$ (iii), $10\,\mathrm{deg}\,/\mathrm{s}^2$ (iv).

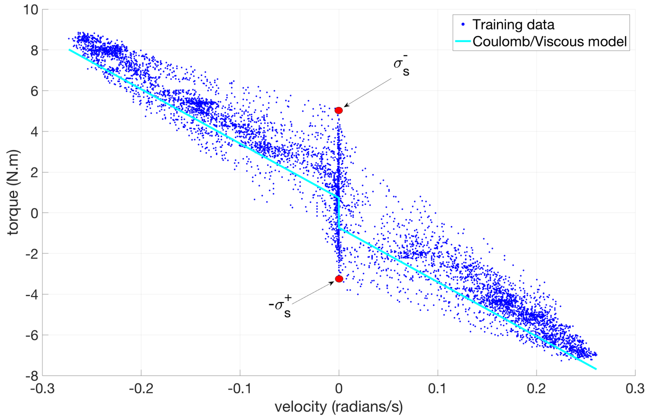
Fig. 2.31 Friction model fitting results on right leg knee, with acceleration axis for filtering the outliers.

The Coupled Joints Case

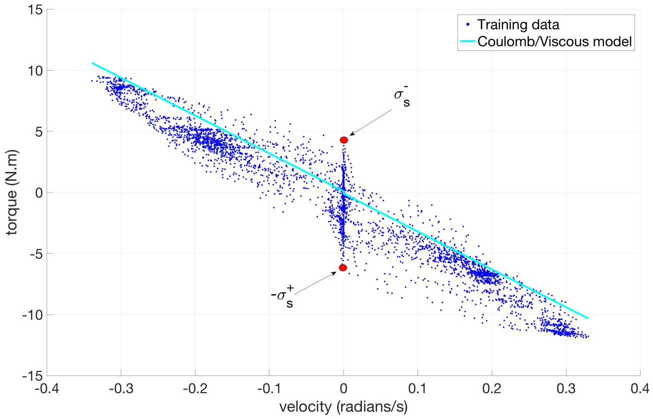
As explained in the methodology section 2.4.4, for identifying the joint parameters of a target motor, e.g. M_c, which is part of a coupling of motors $[M_a, M_b, M_c]$:

- we first block the other two motors M_a and M_b in any desired position. We can do such by using position control with the controller PID gains as high as possible before the low level controller becomes unstable. As on iCub we don't have an interface for setting a single motor in position control, we had to emulate it at high level through PWM control and a custom PID controller using the same gains as the original one.
- we then apply an external torque to the coupled joints, while measuring the joint velocities and torques.

We performed the experiment on the torso motors 0B4M0, 0B3M0, 0B3M1 that drive the joints torso-yaw/pitch/roll. The external force was applied by pushing the upper arms, making the torso rotate following a trajectory constrained by two blocked motors over three. The results are depicted in the figures 2.32b and 2.32b. The fitting problem using the Stribeck model wasn't solvable for the estimation on 0B3M0 or 0B3M1, due to large hysteresis distortion of the measurements caused by the cable driven differential stiffness. We could still identify the maximum observed static friction $\sigma_s \sim 5\mathrm{N\,m}$, which will have a great impact in the motor parameter K^*_{PWM} identification.



(a) Motor 0B3M0 friction model.



(b) Motor 0B3M1 friction model.

Motor	K_c (N m)	K_v (Nms/ deg)
torso motor 3M0	0.73	0.47
torso motor 3M1	0.10	0.54

(c) Identified Coulomb and viscous friction parameters.

Fig. 2.32 Friction model fitting results on the torso motors 0B3M0 and 0B3M1. The fitting was done using a Coulomb/Viscous model.

2.5.2 Second Phase - Motor Parameter Identification

Two alternative procedures were used for generating the training data mapping a range of PWM values to a range of outputs y mentioned in (2.51).

Procedure A - PWM indirect control

This procedure is the one requiring the simplest implementation: we set the target joint in position control, with a fixed, arbitrary, desired position; any external torque about the joint axis results in a rotation detected by the joint encoder; this triggers the PID controller to adjust the PWM in order to bring the joint back to the initial position. While exploiting this behavior, we apply an oscillating external torque covering an even distribution of torques within a reasonable portion of the working range of the actuation chain. We've chosen for the presented experiments a range of moderate torques often up to $30\,\mathrm{N\,m}$ (10% of the PWM fullscale).

On single joints, we obtained results which were good enough for identifying the joint torque model parameter but the accuracy was affected by some quantization and hysteresis effects, as we can see in the experiment results that follow.

Nevertheless the results turned to be worse for cable driven coupled joints: the training data, was inadequate for model fitting, being affected by two main aggravated iissues: quantization and hysteresis resulting in sloppiness in the identified parameters. On top of that the method also suffers from poor repeatability.

Quantization: We can observe in the plots a significant quantization along the PWM axis, as illustrated in the diagram 2.33 representing a typical plot of the function PWM $\mapsto \tau_{\mathrm{J}}$ from data acquired using this method. This problem originated in the PID controller. Its response has actually the form:

$$\Delta\mathrm{PWM} = \mathrm{PID}(q - q^*) \tag{2.61}$$

where ΔPWM is the corrective term from the PID, q and q^* are respectively the measured and the desired joint positions. An update of the correction term ΔPWM is triggered by 1 LSB step of the joint encoder measurement, and so, depends on the encoder resolution (refer to Appendix A for further details). The minimal ΔPWM step would then be $K_p\delta q$ and produce a visible quantization depending on the PID gain K_p and the encoder resolution δq, as we can see in figure

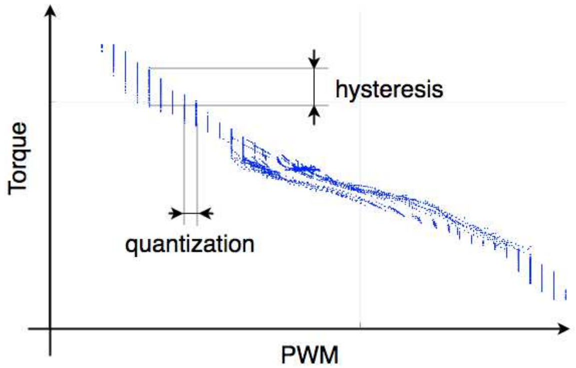


Fig. 2.33 Quantization and hysteresis in the PWM to torque characteristics.

We can estimate the expected quantization as follows. We assume that for the hip roll we have a PID gain $K_p = 2066.66\,\text{pwm.mach.units/deg}$. The resolution of the joint encoders being approximately $\frac{360}{Fullscale} \sim 0.09\text{deg}$, we get a PWM step of $\Delta\text{PWM} \sim 206\,\text{pwm.mach.units}$, which would give, in $\%\,\text{PWM}$, $\frac{206}{FULLSCALE_{\text{PWM}}}100 \sim \frac{206}{32000}100 \sim 0.64\,\%\,\text{PWM}$. On top of that, there is the PID K_i integral term which probably explains why we can have regular quantization and a continuous cloud of points at the same time: the quantization should appear if the integral term of the PID is null ($K_i = 0$) or if it saturates.

Hysteresis: In some cases this can be aggravated by the coupled effect of the transmission elasticity and the static friction generating an alternate succession of stick/slip phases which results in hysteresis.

The combined hysteresis and quantization induced in the training data made it very difficult to fit a simple model, even linear by parts, or a linear combination of simple non-linear functions.

Poor repeatability: The procedure is not repeatable, as the external force is being applied manually with a subjective intensity and velocity that depend on the tester and can vary from one trial to another for the same tester. This makes it difficult to detect anomalies in the data or to use the data as a reference in Key Performance Indicator tests.

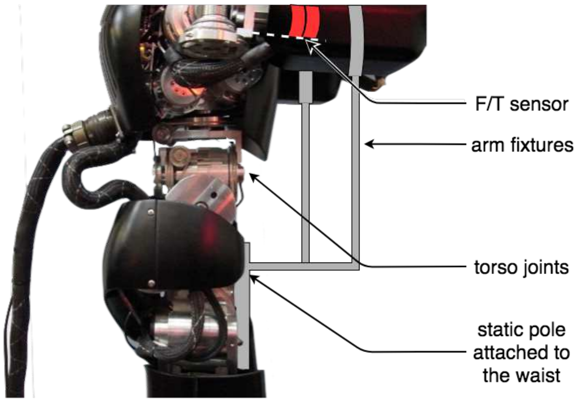


Fig. 2.34 Fixture attached to the static base pole, keeping the upper arms fixed, during the waist/torso joints direct PWM control procedure B. The F/T sensors are located between the fixture attach point and the waist/torso joints. The shoulders are in position control such that the upper arms and the torso can be considered as a lumped single rigid body.

Procedure B - PWM direct control

With this procedure: we set the PWM directly while keeping the joint still by applying an external reaction force; that force is either applied manually by holding the joint child link, either fixing that link to an external fixture. The fixture is the ideal solution, but even in the case of the manual operation we expect some improvements in the results, more specifically getting rid of the quantization, and having repeatable results.

Remark 2.2. *it is important for the Force/Torque sensor providing the measurements—used in the joint torque estimation—to be located between the target joint and the point on the respective child link where the external force is applied. For instance in the case of the torso joints calibration on the iCub robot, since the robot is fixed on a static pole, and the Force/Torque sensors at play are located in the upper arms right below the shoulders, the tester should grab or fix with an external fixture the elbows or forearms of the robot, as illustrated in figure 2.34.*

The actual procedure consists in the followings steps:

- we fix the actuated link as explained earlier,
- the PWM is then set following a periodic pattern, of constant magnitude and frequency, in our case a triangle waveform of amplitude

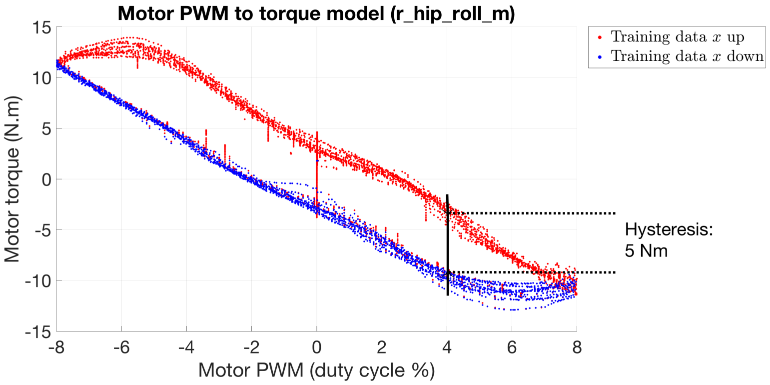


Fig. 2.35 Hysteresis in the PWM to torque characteristics due to static friction in the harmonic drive.

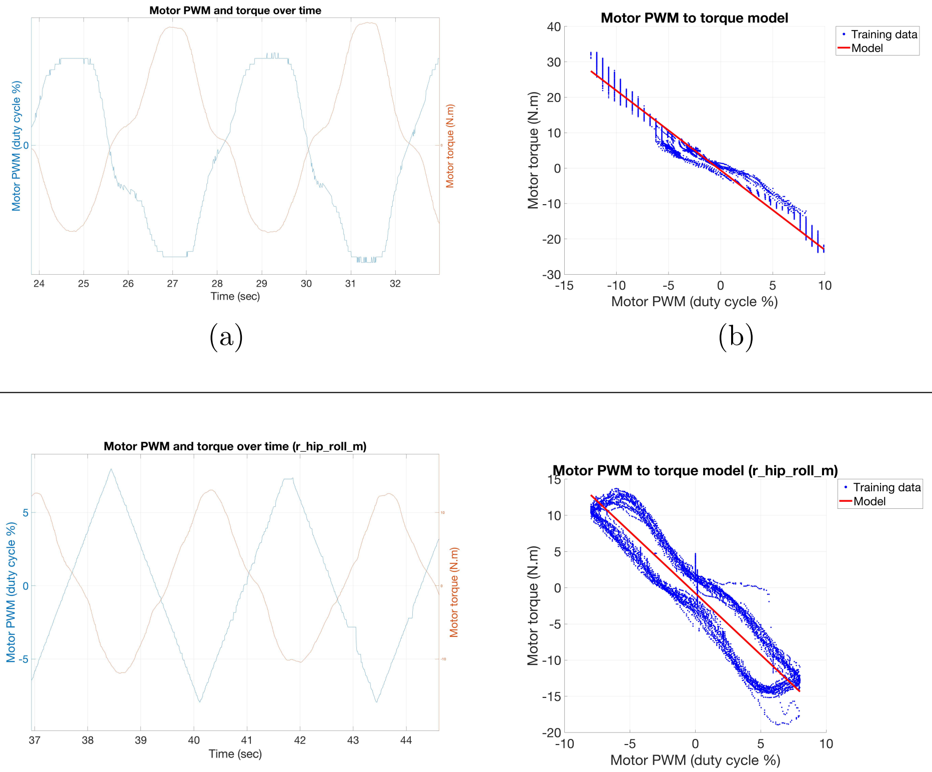
$PWM = 8\%$ PWM, and frequency $f = 0.3$Hz,

- the PWM signal is sampled at 100hertz,
- the pattern stops after 60 seconds, and the PWM final value is 0% PWM.

We perform the test again on the same single joint, the right hip roll, free of any coupling or cable driven actuation. We can see the result in figure 2.35: a torque curve free of quantization. We compare in figure 2.36 the old against the new procedure results. The joint output torque obtained through the direct PWM control is now free of quantization, presents an hysteresis, as expected, but has a a rather clear shape. A zoom in of the PWM time series (2.36a & 2.36c) shows that in the case of the PID controlled PWM, the resulting PWM signal is quantized and oscillating due to the encoder vibrations. There is some quantization though with the new procedure but this is due to delays in the Matlab application, which is not running a real-time process, and is considered negligible in the identification context.

Hysteresis in the Motor Torque

This procedure provides results which are repeatable and free of quantization, but still exhibit hysteresis effects. This is mainly due to the static friction in the reduction drive: we've described in section 2.3.4 the friction model for harmonic drives and how the static friction between the circular spline and the flex-spline, combined with the flex-spline stiffness, can add significant hysteresis to the drive output torque.and we can actually match the level of hysteresis with the value of the static friction at play.



(a) (b)

(c) (d)

Fig. 2.36 (a) PWM set by the PID controller, quantized. (b) Respective quantized motor group torque. (c) PWM directly set to the desired pattern. (d) Respective improved motor group torque. The motor group refers to the brush-less motor and the harmonic drive (section 2.4.3).

We can see highlighted in figure 2.35 the amplitude of the hysteresis $\sim$ 5N m. As explained more in detail in 2.3.4, the output torque of the harmonic drive has an hysteresis due to the static friction between its moving parts and also the flex-spline stiffness. That is the main component of the hysteresis observed in the motor output torque visible in 2.35. We can't measure directly that hysteresis but we could measure the static friction causing it in the experiments we led in section 2.5.1 and in the plot 2.29: σ_s^+ or σ_s^- are usually around 2 N m. That would lead to the same amount of hysteresis.

Model Dependency on The PWM duty cycle level and Frequency

We now wish to evaluate how much the model is sensitive to the PWM duty cycle level and frequency considering the simple case where the duty cycle varies following a triangular waveform.

Waveform Amplitude: We first analyze the torque response to a change in the PWM duty cycle waveform amplitude (figure 2.37):

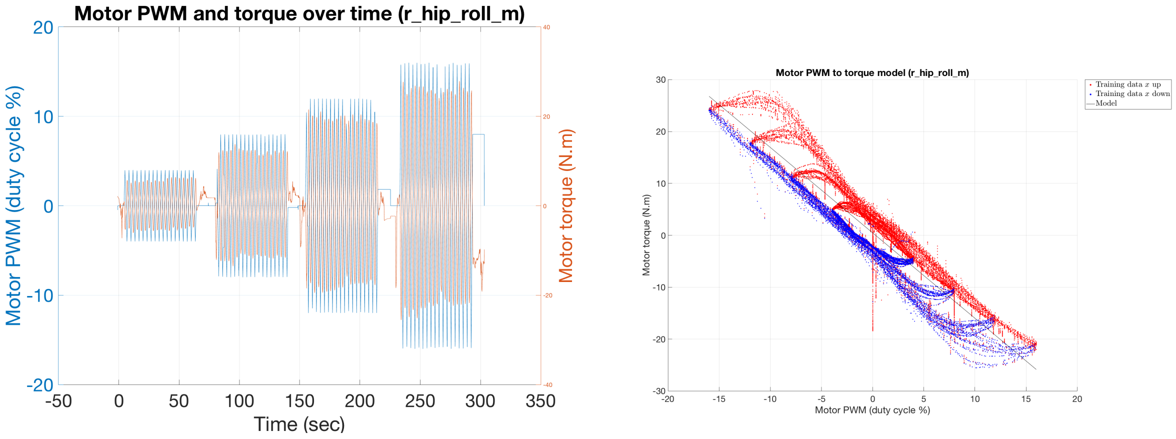
- the waveform has a constant frequency of 0.3 Hz,
- we increase the input waveform amplitude in four steps: $4, 8, 12$ and 16 % PWM,
- each step of constant amplitude lasts 60 s.

The results have a good repeatability with respect to the signal amplitude. The hysteresis level seems to increase linearly with respect to the amplitude, from 5 N m to 10 N m, while we don't see a significant change in the K^*_{PWM}.

Waveform Frequency: We then analyze the torque response to a change in the PWM duty cycle waveform frequency (figure 2.38):

- the waveform has a constant amplitude of 12 % PWM,
- we increase the input waveform frequency in four steps: $0.1, 0.3, 0.6$ and 0.9 Hz,
- each step of constant frequency lasts 60 s.

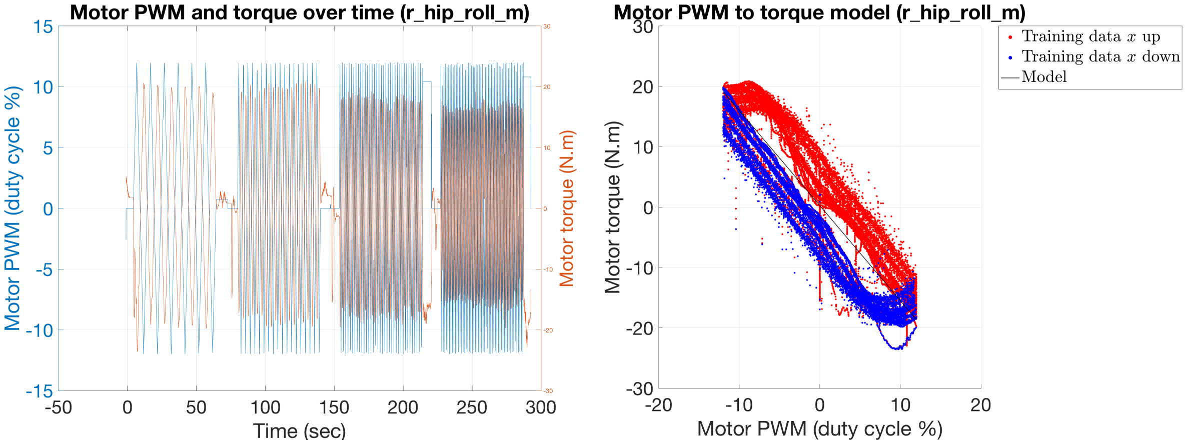
The overall PWM to torque function seems to be invariant (except for a small offset of $\sim$ 2 % PWM and a torque amplitude attenuation of about 20 % with respect to the frequency.



(a) PWM and torque time series.

(b) PWM to torque function.

Fig. 2.37 PWM to Joint torque response for increasing PWM duty cycle amplitudes. The PWM duty cycle frequency is 0.3 Hz.



(a) PWM and torque time series.

(b) PWM to torque function.

Fig. 2.38 PWM to Joint torque response for increasing PWM duty cycle frequencies.

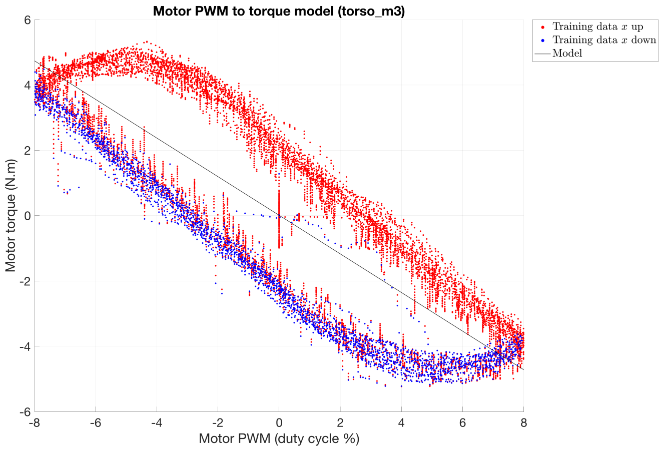


Fig. 2.39 PWM to joint torque response, characterizing a linear model with an hysteresis of 5 N m, clearly visible at both ends of the curve.

The Coupled Joints Case

Unlike the procedure followed in 2.5.1, here we don't have to block the coupled motors not being characterized. If the moving link is tightly blocked by a fixture, and the coupling transformation matrix T (2.3.4) is invertible, we can drive the motor to characterize with a desired PWM waveform, and send a null PWM to the other motors in the coupling (or even switch them off). No power is exchanged between the idle motors and the joints nor between idle motors and the active motor.

Proof. If the transform matrix T is invertible, we can compute the vector of motor velocities $\dot{\theta}$ from the vector of joint velocities $\dot{\mathbf{q}}$ writing $\dot{\theta} = T^{-1}\,\dot{\mathbf{q}} = T^{-1}\,\mathbf{0_{3\times1}} = \mathbf{0_{3\times1}}$. □

We performed the experiment on the torso motors 0B4M0, 0B3M0, 0B3M1. In our case we didn't have a fixture available for rigidly holding the torso as described in 2.2, so we applied the same method used on the friction estimation, controlling the "inactive" motors in position and sending a fixed position. We present the estimation results on one of the motors driving the torso roll DoF, 0B3M0, in figure 2.39. The curve characterizes a linear model with an hysteresis of 5 N m, due again to the harmonic drive static friction and elasticity as for the non-coupled joints (2.5.2).

2.6 Conclusion

Whether the joint torques tracking is done by an inner loop, or integrated in the high level controller, any model based control scheme has its stability improved by a feed-forward term accounting for the joint low level actuation dynamics. This chapter addresses the modeling of that dynamics and the identification of respective parameters.

We first defined a detailed model of a typical joint low level actuation sub-system used on a humanoid robot with electrical actuators, considering the following: the motor PWM-to-torque function, the torque and rotor inertia conversion in the harmonic drive, the Back EMF and the friction in the Harmonic Drive as the main sources of friction. We described the requirements for having a linear PWM-to-torque response and a velocity-to-Back EMF linear model, free of ripples: (i) a Field Oriented Control (FOC) motor driver, for tracking the current in the Direct-Quadrature frame, allowing to keep a constant torque at all speeds; (ii) overlapped stator winding; (iii) a high frequency PWM carrier for smoothing the current ripples without affecting the torque bandwidth. This demonstrates the importance of considering the desired high level control scheme and performance at the very early stage of the robot design. The friction contributions from all components were regrouped in a single Stribeck model, which is a superposition of static-negative viscous (Stribeck effect)-Coulomb-viscous models.

We then defined a methodology for identifying the model parameters from motor PWM, joint veloccity and torque measurements, in the case of single joints. The identifier was based on a least squares fitting algorithm. The methodology was validated on the right leg joints of the humanoid robot iCub. The Stribeck model with three of its usual four dynamic regimes—(I) elastic deformation, (II) Boundary Lubrication, (IV) Full Fluid Lubrication—proved to be a good fit for the experiment results. We have added a constraint on the friction derivative at zero velocity as the observed friction profile exhibits a long Boundary Lubrication regime and no negative viscous friction Stribeck effect, which seems to be a mechanical property specific to the Harmonic Drives. We could correlate the identified static friction with the hysteresis observed in the motor PWM-to-torque function.

The next step was to extended the joint modeling and identification to coupled joints. The same method, used on single joints, is applied to each motor independently, thus decoupling the parameters to be identified.

Unlike previous works on friction identification, the methods presented here were performed in open loop control, relying on the joint torque estimations based on force-torque sensors instead of motor currents adjusted by a closed loop control. This guarantees a simpler and more reliable control of the motor input, avoiding any quantization in the motor torque response, and better revealing the hysteresis effect due to the static friction and elasticity in the transmission components, which would appear as noise on the results we obtained in closed loop identification.

In a future work, the identified friction parameters shall be used on a dynamic friction model (LuGre) and account for other position dependent defects in the Harmonic Drive. In addition, we shall benefit from a fast closed loop current control which would isolate Back EMF from pure mechanical friction. Some friction compensation approaches, although model-free, like the ones based on Gaussian networks, or Radial Basis Function Networks (RBFN), rely on velocity and differential velocity information. As acceleration gives an information on velocity increase of decrease, it should be handy for modeling stick-slip phase transitions. For such use, the accurate direct estimation of angular accelerations would be an attractive feature.

Direct joint torque sensors are expensive. Multiple estimation frameworks are proposed as an alternative to such sensors. Cheaper link force-torque sensors, combined with a tactile skin, can be integrated in a whole-body dynamics estimation framework [Traversaro, 2017, chap. 4] which estimates the external contact forces and all the joint torques, using direct measurement of link accelerations, instead of their estimation from joint accelerations and Forward Kinematics, thus avoiding: propagation errors across the robot kinematic tree; the amplification of high frequency noise due to the numerical differentiation of joint velocities.

For that purpose, we present in Chapter 4 a direct, local estimation of link linear and angular accelerations using inertial sensors. This method is dependent on accurate measurements from accelerometers. The calibration of such sensors is addressed in Chapter 3.

Chapter 3

Accelerometers and Joint Encoders Offsets Calibration

3.1 Introduction

During the first two decades of the twenty first century, the research in humanoid robotics has been greatly facilitated by advances in model-based control approaches. The effectiveness of such approaches depends on the accuracy of the kinematic and dynamic models of a robot as well as the reliability and accuracy of the sensors, and particularly the sensing of joint angles. More specifically, when torque sensors are not available, the joint position sensing may play a critical role in the estimation of joint torques within a multi-link submodel, propagating the link angular and linear accelerations in the submodel [Traversaro, 2017, Chapter 4 section 4.4.2].

The forward kinematics requires an accurate parametric model, which, for standard robot platforms, or even robot units produced in small series, are typically acquired from a CAD model [Rajeevlochana et al., 2012] as well as painstaking calibration procedures. Regardless of how much care is put to this process, a key issue remains in the model and sensor deviations due to wear and tear as well as reassembly/repair work on such robots. The effort being invested in repeated calibration becomes increasingly relevant and has motivated research in self-calibration methods as in [D'Amore et al., 2015], [Mittendorfer and Cheng, 2012], [Roncone et al., 2014]. A related subproblem is that sensor calibration parameters usually have to be estimated a lot more frequently than model kinematic or dynamic parameters.

Joint angle measurements from an encoder are typically subject to an offset [Hollerbach et al., 2008]. Unlike model kinematic parameters, encoder offsets cannot be extracted from detailed CAD diagrams as they are assembly dependent, and so, require a fine calibration procedure once installed and after any kind of mechanical repair tasks on the robot.

While in some cases, the accurate a priori knowledge of joint hardware limits can mitigate this problem, in several advanced humanoid robot platforms the physical joint limits cannot always be reached on all the joints due to constraints induced by wiring, casings or even the complexity of the joint structure. This problem is equally present with absolute encoders, but it is aggravated with relative encoders which require a re-calibration at each power up of the robot system. Therefore, a fast, accurate and automatic joint offset identification method is a real necessity towards testing and deploying humanoid robots in general, an more specifically for accurate joint torque estimation and control, which is our current main concern.

In the context of industrial robots, approaches have been proposed for joint offset calibration using kinematic constraints [Hollerbach et al., 2008] [Liu et al., 2009], some form of kinematic ground truth [Chen et al., 2008] or the effect of added known masses on FT sensors measurements [Ma et al., 1994]. Such approaches may turn to be unfeasible in real-world conditions, due to the difficulties like anchoring the floating base of a humanoid robot, or mounting every DoF (Degrees of Freedom) within fixed constraints. An alternative is to perform model calibration involving alternative sensor modalities, such as inertial sensors [Canepa and Hollerbach] [Wieser et al., 2011]. In particular, the low costs of MEMS accelerometer devices have made it feasible to incorporate multiple such devices on a robot, eventually exploiting the presence of sensor processing boards for other modalities such as tactile sensing [Maiolino et al., 2013].

Such multimodal distributed sensors have already been used to develop techniques, as in [Mittendorfer and Cheng, 2012], [Mittendorfer et al., 2014], where the authors focused on the full kinematic calibration in order to obtain the model forward kinematics. In that case, the calibration problem is split in two: the 3-D shape, i.e. *"body schema"* reconstruction, and the joint axis reconstruction (joint axis position and orientation) in the accelerometer frame. For the two complementary methods, the prior knowledge of the exact pose of the inertial sensors with respect to the link frame was not required, but was instead estimated.

The method in [Mittendorfer and Cheng, 2012] was based on the minimization of an error function with two components: the standard deviation of the measured gravity vector; the error between measured and computed accelerations. No clear proof was given regarding the existence of a global minimum of the error function, which would guarantee the uniqueness of the solution. Furthermore, the used cost function was badly conditioned when including the full set of offsets, which compelled the authors to identify the joint DH (Denavit-Hartenberg) parameters one joint at a time. The main objective of the method was to demonstrate the possibility to discriminate the structural dependencies of a kinematic chain using inertial sensors, along with a first approximation of the DH parameters.

In [Mittendorfer et al., 2014], the same authors built upon their prior work, focusing on identifying the joint axis position and orientation, one joint after the other, through a circle point analysis. Three distinct optimization problems were solved for identifying the joint axis unit vector, the rotation tangential unit vector and the radial distance from the joint axis to the accelerometer frame, which are the critical vectors of the circle point analysis. Despite the improvements achieved by this approach with respect to [Mittendorfer and Cheng, 2012] or older studies using circle point analysis [Canepa and Hollerbach], the obtained accuracy was still not satisfactory for an accurate positioning in the task space, with significant angular deviations of 2.8° and 5.15° respectively for the joint axis tilt and the joint offset.

Another method, using arbitrary distributed and positioned accelerometers, was lead by the same research group. It was based on the global optimization of a rotation quaternion [Wieser et al., 2011], for identifying the joint axis orientation estimation, but leaving out the joint offset estimation.

We propose here a joint offset identification methodology which also uses a distributed set of accelerometers, but in a simpler procedure with respect to existing methods. Our approach focuses on estimating all the joint offsets of a serial chain, although the method could be applied seamlessly to the whole kinematic tree. All the other model kinematic parameters are assumed to be known: the link kinematic parameters, the joint orientations and rough offsets. The joint offsets are known with an error of $\pm 5°$, and the identification procedure goal is to decrease that error to a value within $\pm 1°$. The estimation is done in a single step after acquiring the inertial measurements while executing a single slow motion trajectory. Our approach is based on the following requirements: (i) There is at least one accelerometer every 3 DOFs of the chain, as depicted in Fig. 3.4; (ii) The relative pose of each accelerometer with respect to its support link frame is known accurately a priori; (iii) The joints are moving slowly enough for the sensors to measure only the gravity; (iv) Either the floating base link of the robot is fixed to a static pole, or it is equipped with at least one accelerometer and in this case doesn't have to be static.

There are no requirements on the 3-axis accelerometers defects compensation: a prior full calibration can be performed if the one from the vendors is not sufficient. A method is presented here for that purpose, in addition to the joint offsets calibration. Its novelty is based on the combination of the following three features: (i) it calibrates each accelerometer independently from the joint encoders offsets and without knowing the sensor pose with respect to the link it is attached to, hence avoiding a loop dependency between the two calibration procedures; (ii) in addition to identifying the individual axes offsets and gains, it identifies the cross-axis sensitivity present in single proof-mass MEMS accelerometers [Serrano et al., 2014] [Sun et al., 2010, Table III], i.e. the full three by three calibration matrix; (iii) it performs the identification in-situ, dropping the need to unmount the sensor.

Considering the listed requirements are met and the accelerometers calibrated, we pose the problem of joint offsets identification as an unconstrained nonlinear least squares optimization as we will explain further on. We can apply this optimization independently on each kinematic serial chain, and for proof of concept we applied it to one of the robot iCub legs. We will define a simple trajectory with a minimal set of joint configurations for capturing the training data set. Furthermore, we will compare the obtained offsets against the offsets measured manually (with a level or a laser). Regarding the accelerometers calibration, we will detail the method iterations that were required for improving the results repeatability.

This chapter is organized as follows : Section 3.2 sets the background concepts; Section 3.3 details separately both calibration methods respectively for the accelerometers and the joint offsets; Section 3.4 details the experiments that were conducted and the obtained results; Section 3.5 draws the conclusions.

3.2 Background

3.2.1 Notation specific to this chapter

W	Inertial reference frame or world frame.
F_i	Reference frame associated with link i.
S_j	Reference frame associated with sensor j.
${}^{A}R_{B}$	For any frame A or B, rotation matrix that applied to a 3D vector expressed in frame B, return the same vector expressed in frame A. ${}^{A}R_{B} \in SO(3)$.
α_i^g	Body i proper classical acceleration expressed in body frame S_i. This would be the same acceleration measured by an accelerometer fixed to body i, with a sensor frame coinciding with S_i (refer to Appendix section B.3.3). $\alpha_i^g \in \mathbb{R}^3$.
$\hat{x}$	Estimated variable. This could be an estimation of any acceleration, proper acceleration, the gravity vector, and so on.
${}^{A}g$	Ground truth gravitational field vector expressed in an arbitrary frame A, ${}^{A}g \in \mathbb{R}^3$.
$x^\wedge$	cross product in $\mathbb{R}^3$, 3×3 skew symmetric matrix of vector x ($x \in \mathbb{R}^3$).
C	Accelerometer's calibration matrix. $C \in \mathbb{R}^{3\times3}$.
o	Accelerometer's offset. $o \in \mathbb{R}^3$.
r	Accelerometer's raw measurement. $r \in \mathbb{R}^3$.

3.2.2 Problem statement

Let's highlight the fact that a joint encoder offset is a relative parameter which definition depends on the sensors assembly procedure and on the modeling process that generates the URDF (Universal Robot Definition File describing the robot's kinematic tree model) model from the real robot CAD drawings as illustrated in the diagram 3.1. That process goes through a series of steps: we start from the CAD model initially generated by Creo [1]; a shrink-wrapped version of the CAD model is created (the Creo assembly file), grouping the parts into lumped parts moving together (the sub-assemblies); from there, an XML model describing the links and joint axes frames is created, along with the meshes (STL format), through the Mathworks tool "Simmechanics Link"; finally an open source tool developed in Python at the Italian Institute of Technology converts the XML model and meshes into an URDF. Let's consider one of the 1 DoF joints defined in

---

[1] CREO is a commercial software for 3D CAD modelling.

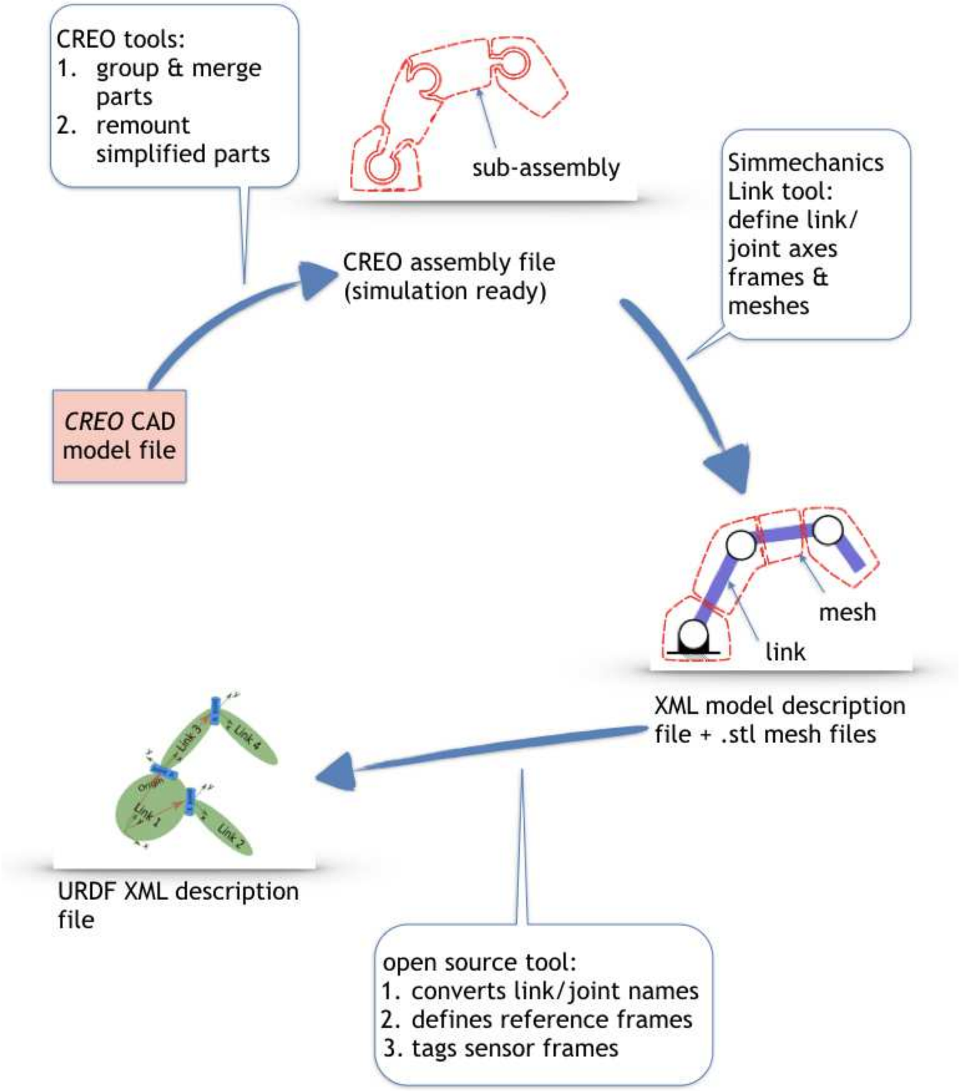


Fig. 3.1 Generation process of a Universal Robot Definition File (URDF) from a Creo CAD file, through a series of conversions from CREO feature tool, Mathworks "Simmechanics Link" tool, and an open source tool developed in Python at IIT.

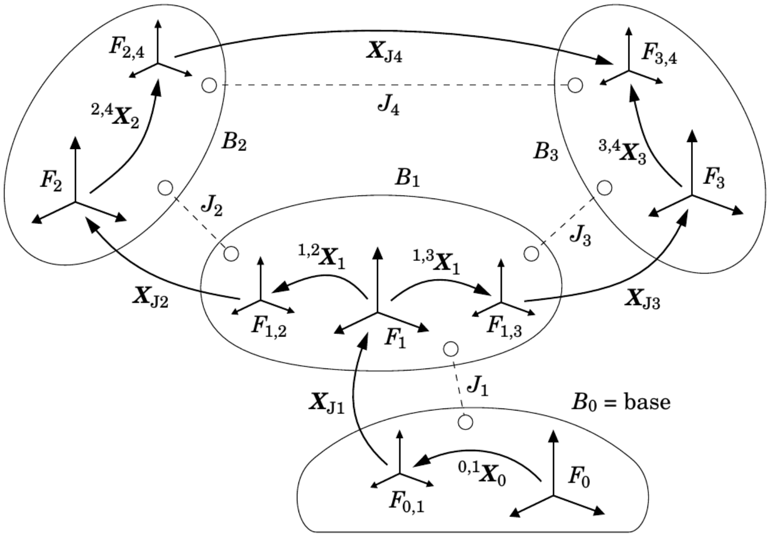


Fig. 3.2 Kinematic tree model of an articulated system of rigid bodies (links) B_0, B_1, B_2 and B_3 connected by joints J_1, J_2 and J_3. Source: [Featherstone, 2014, section 4.2, figure 4.7].

the URDF, that we name J_i, and the formalism described in [Featherstone, 2014, section 4.2]: the joint J_i connects its parent link $B_{\lambda(i)}$ to its child link B_i; the joint frame $F_{\lambda(i),i}$ pose is defined with respect to the parent link frame $F_{\lambda(i)}$; one of the axes of the joint frame gives the orientation of the joint rotation axis; the child link frame F_i rotates around that axis by an angle defined as q_i, and coincides with the joint frame for the initial joint position ($q_i = 0$). In 3.2, $B_{\lambda(i)}$, B_i and J_i could be respectively B_1, B_2 and J_2. In short, the joint frame pose $F_{\lambda(i),i}$ gives the reference orientation of the link frame F_i with respect to the parent link frame $F_{\lambda(i)}$ for $q_i = 0$, and in that configuration, a joint encoder properly mounted on the joint axis should read $\theta_i = 0$.

The accuracy of robot parts dimensions, their positioning and assembling, as well as the mounting of joint encoders are often subject to uncertainties, also referred to as the manufacturing tolerance. These uncertainties usually result in an offset in the encoder measurements. Typically, for absolute joint magnetic encoders like the ones installed on the iCub (AS5045) [2], there is a rotating magnet mounted on the shaft and a Hall array sensing chipset mounted on the motor group frame, as shown in Figure A.1. The positioning of each of these parts is a source of non-linearity errors and an offset error. The offset is the most significant error, which the method presented in this chapter can fix.

For most kinds of encoders, we can derive a simple relationship between the joint angle and the encoder measurement as,

$$\theta_i = q_i + \delta\tilde{\theta}_i \tag{3.1}$$

where θ_i is the reading from the encoder on the joint i, q_i is the joint rotation angle defined by the robot CAD or URDF model, and $\delta\tilde{\theta}_i$ is the encoder offset. If we define a reset reference joint configuration (vector of joint angles $q = 0, q \in \mathbb{R}^N$) matching a given alignment of the joint axis and links, the offsets vector $\delta\tilde{\theta}$ is then defined by the encoders reading in that configuration ($\delta\tilde{\theta} \in \mathbb{R}^N$). For instance, let's consider the manual calibration of the iCub right leg: we align the hip pitch, knee and ankle pitch joint axes as illustrated in figure 3.3; as per the URDF model, the knee joint expected position is $q_{knee} = 0$; the reported joint encoder offset is then $\delta\tilde{\theta}_{knee} = \theta_{knee} - q_{knee}$. There exists many techniques for achieving that specific joints alignment, using external tools or kinematic constraints as already mentioned. Instead, our method uses the gravitation field vector as

---

[2] These encoders are used on the main joints for sensing the joint position of the output shaft.

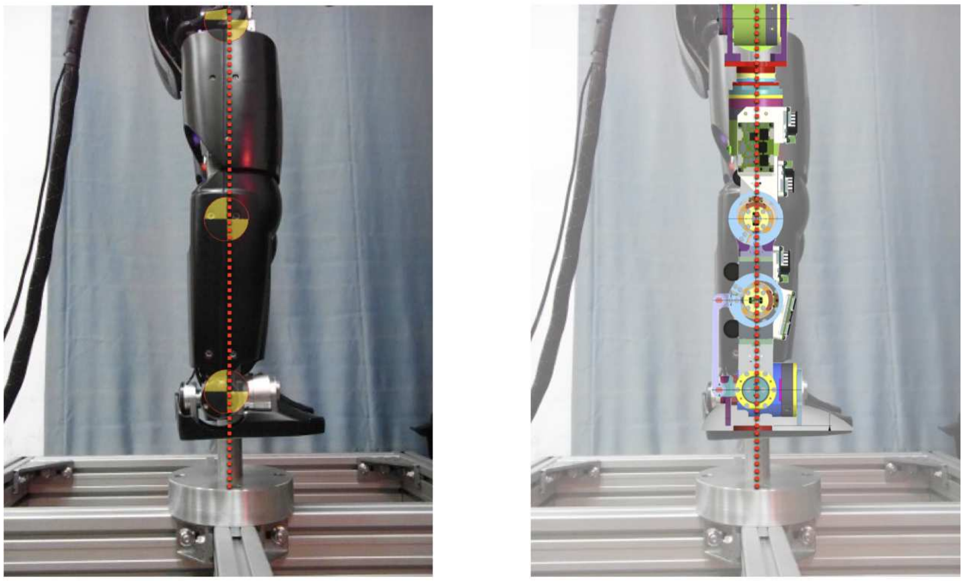

(a) (b)

Fig. 3.3 Calibration of the iCub right leg with a laser (red dotted line) aided alignment of the hip pitch, knee and ankle pitch joint axes. The knee joint expected position $q_{knee} = 0$. (b) actual leg under the covers, exposing the carvings that can be used as the laser aim.

a reference direction, which is measured by the distributed accelerometers as a proper acceleration (as defined in Appendix B.3). As described further, aligning those measurements allows us to compute the joint offsets.

3.2.3 Cost function for a Least Squares optimization

Let us consider a N DoF kinematic chain within the kinematic tree of the robot, including the set of links L_0 to L_N and respective link frames F_0 to F_N. For a given instantaneous joint configuration vector $q = [q_1, \ldots, q_N]^\top$, and an inertial sensor S_j located on link L_i, with $1 \leq i \leq N$, the *expected* link proper acceleration $\hat{\alpha}^g_{j(i)}$ as it would be measured by the sensor j, expressed in the sensor frame, can be computed from the gravity vector and the joint encoder readings as follows: we account for the composition ${}^{F_i}R_{F_0}(q_1, \ldots, q_i)$ of all the rotations transforming the coordinates from the base (waist) frame F_0 to the frame F_i of link L_i; we replace every parameter q_i by its definition

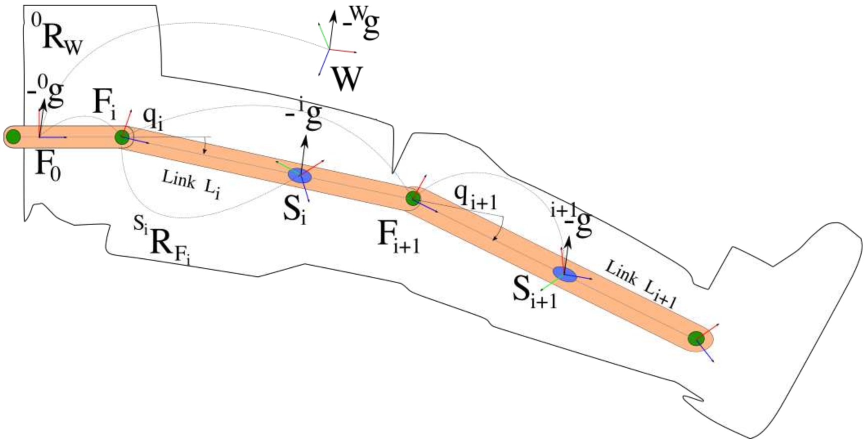


Fig. 3.4 Link, inertial and sensor frames depicted on the links of a kinematic chain: the iCub left leg. For the joint offsets optimization problem, the hip joints can be grouped into a 3-DoF single joint (here q_i).

(3.1):

$$\hat{\alpha}^g_{j(i)}(\theta, \delta\tilde{\theta}) = -\,{}^{S_j}R_{F_i}\,{}^{F_i}R_{F_0}(\theta_1 - \delta\tilde{\theta}_1, \ldots, \theta_i - \delta\tilde{\theta}_i)\,{}^{F_0}g \tag{3.2}$$

$$= -\,{}^{S_j}R_{F_0}(\theta, \delta\tilde{\theta})\,{}^{F_0}g \tag{3.3}$$

Where $(-{}^{F_0}g)$ is the proper acceleration, either measured by an accelerometer mounted on the base link, either computed from the theoretical gravitation field expressed on the base frame.

Note. *For instance, assuming that iCub is fixed on a perfectly vertical pole, we can write* ${}^{F_0}g = \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^\top$.

All the expected accelerations $\hat{\alpha}^g_{j(i)}$ and the actual measured accelerations $\alpha^g_{j(i)}$ should match, but $\delta\tilde{\theta}$ is unknown. So we start with any initial hypothesis $\delta\theta^{(0)}$ and define the global error function e_T as the distance between expected and measured accelerations, resulting from the gap $\delta\tilde{\theta} - \delta\theta$. The function e_T sums the error across all the M sensors and all the joint configurations $p \in [1 \ldots P]$ reached through the whole trajectory of the serial kinematic chain. The problem of finding the correct joint offsets $\delta\tilde{\theta}_k$ can then be posed as a non linear least squares optimization problem where the optimal offsets vector $\delta\theta^*$ minimizes the total error e_T:

$$\delta\theta^* = \arg\min_{\delta\theta} e_T, \quad e_T = \sum_{p=1}^{P}\sum_{j=1}^{M} \left\| \alpha^g_{p,j} - \hat{\alpha}^g_{p,j} \right\|^2, \tag{3.4}$$

where $\alpha^g_{p,j}$ and $\hat{\alpha}^g_{p,j}$ are respectively the measured and expected accelerations of sensor j for the joint configuration p. This approach has been used in Mittendorfer and Cheng [2012] for estimating the full forward kinematic Denavit-Hartenberg parameters of a robot, for which they are constrained to use specific movement patterns on one single joint at a time, and as many optimization cycles. In our approach, one single optimization is run for the complete set of joint offsets. This is possible due to the following choices:

- we focus on estimating only the joint offsets while assuming all the other kinematic parameters to be known, so the problem to solve is simpler with less unknown variables,
- this should reduce the chances for parameter sloppiness and improve the identifiability of the joint offsets,
- the only measured acceleration is the constant proper acceleration $(-^{S_j}g)$, as we perform very slow motions, so we stay in the trained manifold of the calibrated accelerometers (calibration done measuring the gravitation field only).

3.3 Proposed Method

3.3.1 Error function minimization: Unicity and convexity

As mentioned in Section 3.1, we need to verify that the error function minimization has a global minimum. We now consider a serial kinematic chain with $N+1$ links, L_0 to L_N, connected by N 1-DoF joints, with N=1,2 or 3, and the equivalent kinematic chain composed of the two end links L_0 and L_N and a single N-DoF joint connecting the two. At least one accelerometer is attached to each link L_0 and L_N. In the following subsections, the analysis on the equivalent 2-link kinematic chain shows that the computation of the joint position from the gravity measurements gives a unique solution, i.e. the global minimum, if the following requirements are met:

- **for 1-DoF joints:** acquire measurements for at least one joint position

- **for 2 or 3-DoF joints:** acquire measurements for at least two positions
- **in all cases:** none of the axis should ever be aligned with the gravity vector, or mutually aligned (singularity).

The actual multidimensional joint position $q_i \in \mathbb{R}^N$ and measurements $\theta_i \in \mathbb{R}^N$ would match a unique joint offset $\delta\tilde{\theta}_i \in \mathbb{R}^N$. This can easily be extended to a generalized kinematic chain having at least one accelerometer attached to each link, and joints with no more than three DoF, or an equivalent serial kinematic chain with only 1-DoF joints and having at least one accelerometer every three links. Moreover, the optimization problem might not be convex depending on the distribution of the inertial sensors and the measurements noise. A full observability and convexity study is out of the scope of this thesis. We detail below the offset computation analysis on the 2-link kinematic chain.

Offset computation for a 2-link / one 3-DoF joint chain

As an example for understanding the requirements stated above, we will analyze here the problem of identifying the joint offsets in a much simpler case of a 2-link kinematic chain, with an accelerometer attached to each link, and with a 3-DoF joint composing the 3 Euler angles rotation ϕ_i,θ_i,ψ_i. As shown in Fig. 3.5, we consider the sensor frames S_{i-1}, S_i, the link frames F_{i-1}, F_i, the joint frame $F_{i-1,i}$. The joint frame is defined by the condition $[\phi_i, \theta_i, \psi_i] = [0, 0, 0] : F_{i-1,i} = F_i$, meaning that at the joint original position, the joint frame coincides with the link frame F_i. As the sensors are firmly attached to the respective links, and their pose is accurately known, for this particular analysis we simplify the problem by assuming $S_{i-1} = F_{i-1}$ and $S_i = F_i$. We now can define the overall rotation projecting any vector from frame F_{i-1} to frame F_i as:

$$^{F_i}R_{F_{i-1}} = R_z(\psi_i) \cdot R_y(\theta_i) \cdot R_x(\phi_i) \tag{3.5}$$

Where ϕ_i, θ_i, ψ_i are unknown but can be easily estimated from the rotation matrix $^{F_{i-1}}R_{F_i}$ using a known method as the one described in [Diebel, 2006, sec.5.6.3]. This method gives a unique solution, as long as we avoid the representation singularities of Euler angles [Siciliano, 2009, Section 2.4]. Let us focus on computing the rotation matrix $^{F_i}R_{F_{i-1}}$. It can be done using the Orthogonal Procrustes algorithm introduced in [Schönemann, 1966]. We need to define a pair of free vectors $[u, v]$, with fixed orientation with respect to the world frame, expressed in a frame F_{i-1} as $A = \left[{}^{F_{i-1}}u, {}^{F_{i-1}}v\right]$, and expressed in another frame F_i as $B = \left[{}^{F_i}u, {}^{F_i}v\right]$, and define the transformation of F_{i-1} coordinates to F_i coordinates as follows:

$$\begin{aligned} A &= \left[{}^{F_{i-1}}u, {}^{F_{i-1}}v\right] \\ B &= \left[{}^{F_i}u, {}^{F_i}v\right] \\ B &= {}^{F_i}R_{F_{i-1}}\, A \end{aligned} \tag{3.6}$$

This approach was used in [Mittendorfer et al., 2014] for the automatic identification of a kinematic model, but was using tangential acceleration vectors and rotation axis as reference directions in space for doing the mapping between transformed frames. Another method was using the same principle for calibrating the frame poses of the IMU sensors attached to a robot, using a global angular velocity as the reference vector [Rotella et al., 2016, Section V]. The whole robot was rotated in the air, with all its joints locked, hence was considered like a single rigid body, and all the IMU sensors mounted on the robot would measure the same angular velocity vector. In our case, we only use the gravity as the reference direction, and on top of that, our goal here is to define a minimal simple trajectory still guaranteeing the uniqueness of the optimization solution. The estimation itself is done by the optimizing algorithm.

The gravity vector can be used as a fixed vector with respect to the world frame. Each sensor measures the gravity and provides the respective coordinates in the sensor frame. But we only have one fixed vector available. We show below how we can overcome this problem in three steps:

(i) we first match the gravity measurement in F_{i-1}, with the gravity measurement in F_i as follows:

$${}^{F_i}g_{(1)} = R_z(\psi_i) \cdot R_y(\theta_i) \cdot R_x(\phi_i) {}^{F_{i-1}}g, \tag{3.7}$$

(ii) we then rotate link i of an angle β about the joint axis closer to the link $i-1$ (axis x), such that the gravity measurement vector appears rotated in F_i, and we get a new match for the measurements:

$$\begin{aligned} {}^{F_i}g_{(2)} &= R_z(\psi_i) \cdot R_y(\theta_i) \cdot R_x(\phi_i) \cdot R_x(\beta) {}^{F_{i-1}}g && (3.8) \\ &= R_z(\psi_i) \cdot R_y(\theta_i) \cdot R_x(\phi_i) \cdot {}^{F_{i-1}}g_\beta, && (3.9) \end{aligned}$$

where ${}^{F_{i-1}}g_\beta = R_x(\beta) {}^{F_{i-1}}g$ is an apparent vector resulting from the rotation of the gravitational vector ${}^{F_{i-1}}g$ by β about the axis x. We now redefine the

2 pairs of vectors A and B as follows:

$$A = [{}^{F_{i-1}}g, {}^{F_{i-1}}g_{\beta}], \quad B = [{}^{F_i}g_{(1)}, {}^{F_i}g_{(2)}], \tag{3.10}$$

and considering the transformation:

$$B = {}^{F_i}R_{F_{i-1}}\, A, \tag{3.11}$$

we get:

$$A^{\top}\, {}^{F_{i-1}}R_{F_i} = B^{\top} \tag{3.12}$$

(iii) as mentioned earlier, this can be posed as a minimization problem and solved with the Orthogonal Procrustes algorithm. Although our currently used methodology is not solving the problem this way, the demonstration above shows that by measuring the gravitational vector in two different positions of ϕ_i, apart by an angle β, allows to identify from those measurements a unique joint offset solution.

Offset computation for a 2-link / one 1-DoF joint chain

The 2-DoF joint offset problem is similar to the 3-DoF joint problem, but the 1-DoF problem requires only one single joint position for finding a unique solution, as explained further. We consider a parent link L_{i-1}, a child link L_i, the respective link frames F_{i-1} and F_i, the joint i connecting both links and the respective joint rotation axis Δ_i.

As seen previously, we need to define a pair of free vectors $[u, v]$, fixed with respect to the world frame, for defining the transformation (3.6). We can chose the rotation axis unit vector $\Delta_i \in \mathbb{R}^3$, considered fixed, as the vector u, and the gravity vector as the vector v. A single joint position q_i, resulting in the rotation matrix $R_{\Delta_i}(q_i) = {}^{F_i}R_{F_{i-1}}$, will give us a set of vector coordinates A and B, as defined in (3.6), required for solving the minimization problem and Orthogonal Procrustes problem. In the presence of significant noise, we build a data set of M stacked measurements captured over P positions instead of just two, for more accuracy and stable results.

Note. *for both problems, If the gravity vector g is aligned with the joint axis* Δ_i*, the rotation has no effect on the vector and the equation* (3.12) *has an infinity of solutions. As we can see in the figure 3.5, only the component* $g_{\perp}$ *is rotated by* $R_{\Delta_i}(q_i)$*. The greater the angle between* Δ_i *and g, the better the problem is conditioned for an accurate solution. The ideal angle would be* $\pi/2$ *but for avoiding collisions we chosed an angle around 30 degrees.*

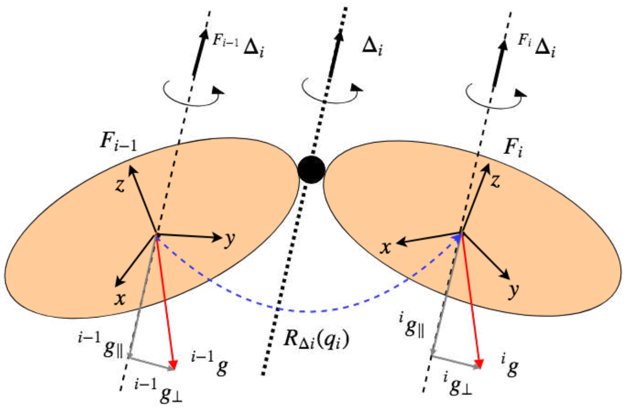


Fig. 3.5 2-link, 1DoF-joint chain: computing the rotation matrix transforming the gravitational vector from the world to the each sensor frame.

3.3.2 Defining joint trajectories

We define here the joint trajectories to follow when capturing the training data set for the joint offsets calibration. This is performed while meeting the requirements listed in the previous section and measuring only the gravity acceleration.

Minimal Motion: We now assume we selected a set of joint configurations $(\bigcup_{p=1}^{P} q_{(p)})$ to use for capturing the training data, as per the requirements defined in section 3.3.1. For avoiding oscillations or simply the higher linear accelerations due to the motion stop and start, a minimum jerk [Kyriakopoulos and Saridis, 1988] [Huang et al.] slow trajectory is used for reaching those selected positions:

- for 1-DoF joints, we select a fixed position for the whole data capture,
- for 2-DoF and 3-DoF joints, rather than cycling through a sequence of purely static joint configurations, we rotate one of the axis in its partial or full range, while keeping the two other axis fixed, in order to span a set of positions that include the two required positions for 3-DoF joints. This will improve the accuracy of the global minimum estimation as well as the convexity of the cost function, reducing the risk of falling into local minima.

By these means, we can define an acquisition trajectory running through a minimal set of positions required for solving the optimization problem.

Remark 3.1. *In any case, the rotating axis must never be aligned with the gravity vector, since in that condition the accelerometers are not sensitive to a change in the respective joint position.*

Measuring only gravity For the acceleration due to the joints motion, also known as the coordinate acceleration (B.3), to be negligible with respect to the gravity, we need to limit its normal and tangential components (α_{tan} and α_{cp}) that could be measured by any of the inertial sensors in the system. So for a given joint J, its maximum angular velocity ($\dot{q}_{J,max}$) and acceleration ($\ddot{q}_{J,max}$) depend on the greatest distance d_J to any accelerometer moved by that joint during the calibration, and on the desired gravity measurement accuracy Δ_α. So these acceleration components are tolerated if $\| \alpha_{tan} \| \leq \Delta_\alpha$ and $\| \alpha_{cp} \| \leq \Delta_\alpha$. Let O be a point on the joint rotation axis, S_i the origin of the accelerometer frame and $\omega_J \in \mathbb{R}^3$ the joint angular velocity (refer to definition in Appendix section B.3.5), and $\dot{\omega}_J \in \mathbb{R}^3$ its acceleration, all expressed in the joint J frame. We then have:

$$\| \alpha_{tan} \| = \| \omega_{J,max}{}^\wedge \left(\omega_{J,max}{}^\wedge OS_i \right) \| \leq \Delta_\alpha \quad \Rightarrow \ddot{q}_{J,max} \leq \frac{\Delta_\alpha}{d_J} \tag{3.13}$$

$$\| \alpha_{cp} \| = \| \dot{\omega}_J{}^\wedge OS_i \| \leq \Delta_\alpha \quad \Rightarrow \dot{q}_{J,max} \leq \sqrt{\frac{\Delta_\alpha}{d_J}} \tag{3.14}$$

On top of that, we can verify, for each trajectory, that the norm of the acceleration is constant and equal to the approximated norm of earth's gravitational field as defined in Appendix section B.3.4.

3.3.3 Error function optimization

The error function is a sum of squares subject to bounding constraints on the offsets $\delta\theta$. Considering that the described procedure is a fine calibration, the lower and upper boundaries can be set to ±20° [3]. We are tuning the $\delta\theta$ offsets vector as an input parameter to error function optimization, so the dimension of the problem is no greater than the number of DoF of the robot. If we assume to have one or more sensors for each link, a Model-Fitting algorithm for over-constrained systems, like for instance a *trust-region-reflective* algorithm, is suited for this problem [Conn et al., 2000] [Berghen, 2004]. Other algorithms were tested, like the *Levenberg-Marquardt* algorithm, but produced less accurate results.

---

[3] It turned out that the algorithm could consistently converge without the use of boundaries for the initial offset.

Note. *The* trust-region-reflective *algorithm does not require the gradient of the error function, although this would potentially improve the accuracy and the processing performances [Conn et al., 2000].*

3.3.4 Accelerometers calibration

The performance of the optimization problem previously described depends directly on the accuracy of the accelerometers measurements. Even high performance 3-axis accelerometers can have different gains and bias (offset) for each axis, as well as some cross-axis sensitivity due to the interaction between the axes, intrinsic to a single-proof mass design, and to the manufacturing tolerance [Serrano et al., 2014] [Sun et al., 2010, Table III].

Sensor model fitting

For canceling out the axes bias and gain differences, we need to fit a model to the sensor behaviour, relating the 3-axis raw measurements vector r to the actual sensor proper acceleration α^g expressed in the sensor frame. In the current section 3.3.4, α^g will be denoted a for the sake of simplification. In general we assume that this model is affine [Serrano et al., 2014] [Maj and Napieralski, 2017], having an offset o accounting for the biases and an invertible calibration matrix C accounting for the axis gains and cross-axis sensitivity. These calibration parameters are also assumed to be constant:

$$a = C(r - o) \tag{3.15}$$

where r is the sensor output already converted to acceleration units ($\mathrm{m\,s^{-2}}$) with a default gain. If the acceleration α^g spans a unit sphere in the 3 dimensional space, the raw measurement r spans an ellipsoid that can be rotated with respect to the sensor frame S_j.

Proof. Assuming that (3.15) is a good model of the inertial sensor, we can easily show that the model is equivalent to an ellipsoid equation. An ellipsoid implicit equation can be written as [Olver and of Standards and Technology , U.S.]:

$$\frac{{}^{\mathrm{E}}x_1^2}{s_1^2} + \frac{{}^{\mathrm{E}}x_2^2}{s_2^2} + \frac{{}^{\mathrm{E}}x_3^2}{s_3^2} = {}^{\mathrm{E}}\mathbf{x}^\top T\, {}^{\mathrm{E}}\mathbf{x} = 1 \tag{3.16}$$

$${}^{\mathrm{E}}\mathbf{x} \in \mathbb{R}^3, T = \begin{bmatrix} s_1^{-2} & 0 & 0 \\ 0 & s_2^{-2} & 0 \\ 0 & 0 & s_3^{-2} \end{bmatrix} \tag{3.17}$$

where ${}^{\mathrm{E}}\mathbf{x}$ is the position of any point of the ellipsoid surface, expressed in a frame which axes e_1, e_2 and e_3 are aligned respectively with the semi-axes s_1, s_2 and s_3 of the ellipsoid. We can express x in any frame A where the ellipsoid is tilted and which orientation is given by $\mathrm{A}^{R}_{\mathrm{E}}$. We then write (3.17) as:

$$\begin{aligned} & {}^{\mathrm{A}}\mathbf{x}^{\top} \left({}^{\mathrm{A}}R_{\mathrm{E}} T \, {}^{\mathrm{E}}R_{\mathrm{A}}\right) {}^{\mathrm{A}}\mathbf{x} = 1 \\ \Longleftrightarrow \; & {}^{\mathrm{A}}\mathbf{x}^{\top} \, M \, {}^{\mathrm{A}}\mathbf{x} = 1 && (3.18) \\ & \text{with } M = \left({}^{\mathrm{A}}R_{\mathrm{E}} T \, {}^{\mathrm{E}}R_{\mathrm{A}}\right). && (3.19) \end{aligned}$$

(3.19) is equivalent to a singular value decomposition of M where T is the real square diagonal matrix and the singular values are the squared ellipsoid semi-axes s_1^2, s_2^2 and s_3^2 [Meyer, 2000]. M is positive definite. So if for a sensor converting a physical variable (the sensor proper acceleration a) into an measurement r, we can write $r = D\,a$ and D is invertible, such that $a = D^{-}1\,r$, then we get:

$$\begin{aligned} & a = D^{-1}\,r \\ & a^{\top} a = r^{\top} D^{-\top} D\,r \\ & r^{\top} \left(\frac{D^{-\top} D}{||a||} \right) r = r^{\top} M\,r = 1, \end{aligned}$$

and if $||a||$ is constant, the space of r is an ellipsoid which is skewed with respect to the sensor frame depending on the values of D (actually the cross-axis gains). This will be observed in the experiment results in section 3.4. $\square$

Unlike the *6-point tumble calibration* procedure which consists in measuring the gravity in six complementary orientations (x, $-x$, y, $-y$, z, $-z$) [STMicroelectronics, 2015], we opted for an in-situ procedure, thus avoiding removing and remounting the devices and thus better cope with the eventual drift of sensor properties. The drawback is that we cannot anymore estimate C and o from the raw measurements of gravity along all the 6 semi-axes of the sensor frame: $+x, -x, +y, -y, +z, -z$. For that reason we've used an alternative method based on an ellipsoid-fitting algorithm from [Hunyadi, 2013]. Through this fitting we can reconstruct the ellipsoid manifold from a limited span of the raw measurements, captured using reachable joint configurations.

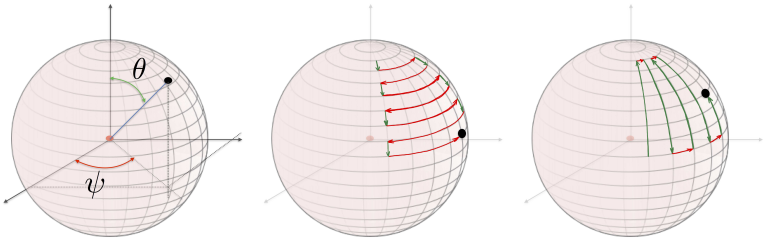


Fig. 3.6 Generating a grid dataset for spanning the accelerometer ellipsoid manifold. ψ and θ give the orientation of the gravity vector in the represented link frame. The link frame z axis is always aligned with the joint axis setting ψ.

Generating a “grid” training data set

We wish to rotate the accelerometers frames while measuring the gravity, for spanning a fraction of the raw measurements manifold as broad as possible. If the robot kinematic architecture allows, we move as a single body all the links supporting the sensors. We perform slow motion rotations, using two orthogonal revolute joints in order to cover a grid of orientations as the ones defined by the two angle parameters ψ and θ, of a polar coordinate system, similar to what was done in [Traversaro et al., 2015], and illustrated in figure 3.6. We then run the ellipsoid-fitting algorithm on the generated training data set.

Extracting the calibration parameters

The ellipsoid-fitting algorithm returns the parameters of the implicit equation defining the ellipsoid, $\mathcal{F}(x, y, z) = 0$, but also the explicit parameters of the ellipsoid: its centre, $centre \in \mathbb{R}^3$; the lengths of its 3 semi-axis, $s_1, s_2, s_3 \in \mathbb{R}^3$; the quaternion $quat \in \mathbb{R}^4$ and the rotation matrix ${}^{\mathrm{E}}R_{\mathrm{S}_j} \in \mathbb{R}^{3\times 3}$ defining the skew of the ellipsoid. Actually, ${}^{\mathrm{E}}R_{\mathrm{S}_j}$ describes the rotation transform from the sensor frame S_j coordinates into the coordinates on a frame E aligned with the skewed ellipsoid semi-axis. For building the calibration matrix we follow the sequence transforming the space of raw vectors (ellipsoid manifold) into the space where the actual ”calibrated” accelerations span a spherical manifold: (i) remove offset; (ii) rotate to the frame E (skewed ellipsoid frame); (iii) apply the semi-axes gains compensa-

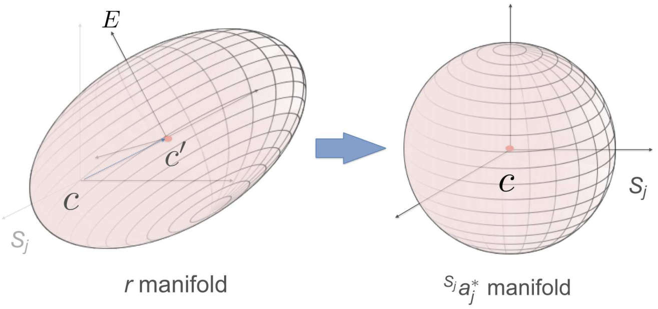


Fig. 3.7 Affine transformation $r \mapsto C\,(r - o)$ transforming vectors from the ellipsoid space (r) into vectors of the sphere space (calibrated acceleration $^{\mathrm{S}_j}a_j^*$).

tion; (iv) rotate back to sensor frame.
We can now extract the calibration parameters from the model:

$$o = centre \quad \in \mathbb{R}^3 \tag{3.20}$$

$$C = {}^{\mathrm{E}}R_{\mathrm{S}_j}^{\top} g\ \mathrm{diag}(s_1, s_2, s_3)^{-1}\ {}^{\mathrm{E}}R_{\mathrm{S}_j} \tag{3.21}$$

With g the norm of earth's gravitational field (refer to Appendix section B.3.4). Using these values on (3.15) allows to generate a corrected measurement from any raw measurement, following the affine transform $r \mapsto C\,(r - o)$ as illustrated in figure 3.7.

3.4 Experiments

The proposed method was tested on the iCub humanoid robot (for more details refer to Section 1.2). As we initially stated, the calibration procedure is done independently on each kinematic serial chain: one full leg, one full arm or the torso along with the head. We tested the method on both legs, but for brevity, we report here only the results on the left leg, which had the most complete set of working inertial sensors. The leg has 6 chained DoF: hip pitch / roll / yaw, knee, ankle pitch / roll. All of the controlled DoF use absolute joint encoders which specifications on the accuracy and repeatability are detailed in the appendix section A.1. The robot is equipped

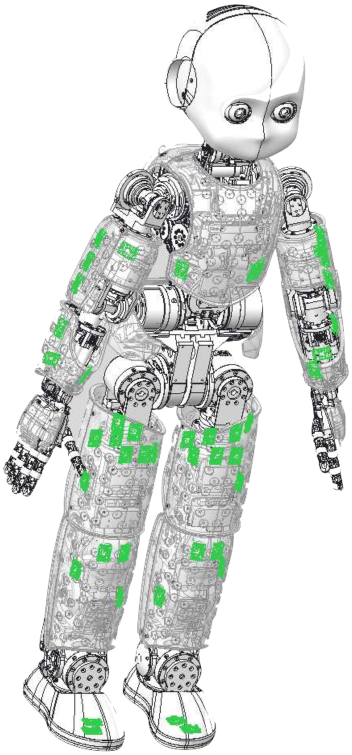


Fig. 3.8 Locations of the accelerometers (in green) integrated into the sensor processing modules of the iCub tactile skin.

with skin sensor units distributed all over the arms, legs and torso. Each of these units has a processing module that embeds a 3-axis high performance linear accelerometer (STMicroelectronics LIS331DLH). The leg holds 13 of a total 44 of these modules distributed as depicted in Fig. 3.8. Additional details on the tactile sensing electronic boards may be found in [Maiolino et al., 2013].

3.4.1 Sensors and setup for data acquisition

Joint configuration and trajectory control For the calibration of the accelerometers and the joint encoders, the robot is mounted on a pole, the base link (waist) being tightly screwed to the pole such that the feet don't touch the ground and so legs and arms can move freely. This is a comfortable choice, although not mandatory, that also requires less supervision. All the joints are controlled through a high gain position control loop. While the leg is moving, all other joints are configured to a fixed position. As per the requirement defined in paragraph 3.3.2, the position control is run through the minimum jerk trajectory generator from the iCub low-level control interface, parametrized with the start and end positions of the joint angle, and a maximum linear velocity of $2\,\mathrm{deg}/\mathrm{s}$, which is actually $\leq v_{max}$. v_{max} was computed from the robot geometry (hip to foot distance $d_{hf} \sim 0.5\,\mathrm{m}$) and the desired gravity measurement accuracy ($\Delta_{\alpha} = 10^{-2}\,\mathrm{m\,s^{-2}}$, so $\frac{\Delta_{\alpha}}{\alpha} \sim 0.1\,\%$): $v_{max} \sim 8\,\mathrm{deg}/\mathrm{s}$. In the experiment, we move the joints at $4\,\mathrm{deg}/\mathrm{s}$.

Joint encoders accuracy The magnetic absolute encoders (AS5045) installed on the main joints of iCub have two moving parts: a rotating magnet fixed on the shaft and a Hall array sensing chipset fixed on the motor group frame, as shown in Figure A.1. The alignment between the magnet and the sensing chipset is subject to tolerance requirements, and any misalignment beyond the defined tolerance can cause non-linearity errors, as described in Appendix Sections A.1 and A.1.1. For the experiments, we rely on the compensation of eventual errors at a firmware level, and can consider the encoder measurements to be linear.

Inertial data acquisition The accelerometers support user selectable full scales of +-2g/+-4g/+-8g, and the current selection is +-2g. The sensors are running at 100 Hz, and the data was collected by the calibration application at that same rate. Prior to any calibration procedure, the sensors data need to be filtered. While the joint encoders signal has very low noise, the accelerometers measurements have a noise level around $0.2 m.s^{-2}$, which is significant against a signal of $10 m.s^{-2}$. The signal also presents some outliers due to motor vibrations that we wish to eliminate. We are using a Savitzky-Sgolay filter offline, whose parameters are tuned manually. A simple tool was implemented for this tuning, and integrated in the main joint offset calibration tool. The selected optimal parameters are: a large window size F (600 samples) and a high polynomial order K of 5, which is actually the order used by the low jerk trajectory generator.

3.4.2 Accelerometers calibration

We are using 3-axis high performance linear accelerometers, but their offsets and the gain differences across the 3 axis are significant enough for generating a badly shaped measurement error distribution as we can see in the results of Fig. 3.9.

Non-linear optimization algorithm for fitting the ellipsoid manifold

We apply the method presented in paragraph 3.3.4 for fitting each accelerometer model and extracting the respective calibration parameters. The used ellipsoid-fitting algorithm, *ellipsoidfit*, defined in [Hunyadi, 2013], was implemented in the free Matlab package *quadfit* [4]. This algorithm fits an ellipsoid by minimizing the point-to-surface distance between the training samples and the ellipsoid model. It defines a cost function based on that distance and uses the Matlab optimization solver *lsqnonlin* parameterized with the *Levenberg-Marquardt* algorithm for minimizing that cost. The starting point for the optimization is an initial solution from a simple least-squares fitting. The algorithm can take boundary constraints for the fitted variables. Our initial approach uses an unconstrained optimization.

Trajectory for generating a "grid" training data set

As described in the section 3.3.4, we have to span the raw measurements manifolds along a grid by rotating the iCub's leg. We configure the leg joints as follows: the hip pitch, knee and ankle joints are fixed to 0 deg; the hip roll joint $[min, max] = [20, 80]$ deg, the grid step = 15 deg; the hip yaw joint $[min, max] = [-60, 60]$ deg, the grid step = 20 deg. In figure 3.6: the upper leg link frame matches the frame depicted in the figure, such that the hip yaw axis is always aligned with the z axis of that frame; the hip roll matches the θ angle and the hip yaw matches the ψ angle.

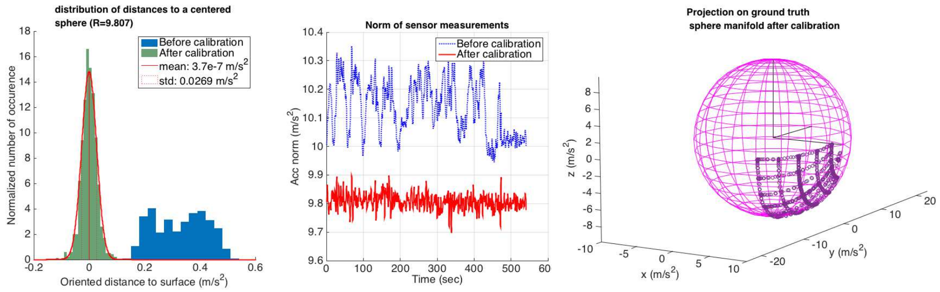
In fig. 3.10 we depict the results on the calibration of six of the lower leg accelerometers. In fig. 3.9 we can see a plot showing the projection of the measurements on the spherical manifold of the ground truth gravity vectors (a), and the distribution of the measurement errors with respect to the ground truth gravity norm. The distribution of those errors, i.e. standard deviation and the mean value, is clearly improved by the calibration, converging to a residual Gaussian noise (fitting of the red curve).

Calibration procedure validation

For validating the calibration procedure, we have analyzed the variance of the identified parameters over a series of consecutive calibration iterations:

---

[4] `http://www.mathworks.com/matlabcentral/fileexchange`, file "fitting-quadratic-curves-and-surfaces"



(a) (b) (c)

Fig. 3.9 Calibration of one accelerometer on the left lower leg (MTB 11). The error distribution of the gravitational vector measurements ($||a_j|| - |g|$), and the norm time series of the vector are depicted, before and after the calibration.

five iterations with intervals of five minutes, and with intervals of one week, as illustrated in figure 3.11 for two of the accelerometers attached to the left leg of iCub: MTB_4 and MTB_9. Tables 3.1 and 3.2 give the mean values and standard deviations of the three axis offsets and the six parameters in the lower triangular part of the symmetric calibration matrix C, for all the tested acceleroemeters on the leg link:

$$\mathbf{o} = \begin{bmatrix} x \\ o_y \\ o_z \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} c_{xx} & \cdot & \cdot \\ c_{yx} & c_{yy} & \cdot \\ c_{zx} & c_{zy} & c_{zz} \end{bmatrix} \tag{3.22}$$

The results show that for consecutive calibration iterations giving the same quality of fitting with a low residual error equivalent to a Gaussian noise ($\mu < 10^{-3}\,\mathrm{m\,s^{-2}}, \sigma < 0.1\,\mathrm{m\,s^{-2}}$), the identification of the calibration parameters is subject to high uncertainties. This characterizes the parameters as being "sloppy".

Fixing the loose-fitting with a constrained ellipsoid offset

The most probable causes for these uncertainties are the high level of noise in the measurements and the fact that the portion of accelerations manifold we are using for fitting the ellipsoid is too small. We have two ways to overcome this problem: either we increase the training manifold, by changing the joint configurations trajectory; either we add constraints to the fitting algorithm.

Sensors	**Offsets**			**matrix C gains**					
	o_x	o_y	o_z	g_{xx}	g_{yy}	g_{zz}	g_{yx}	g_{zx}	g_{zy}
MTB 10b1	-0.0734	0.613	-0.566	0.969	0.952	0.959	0.0234	-0.00591	0.0105
MTB 10b2	-0.206	1.14	0.194	0.964	0.92	0.961	0.034	-0.00483	0.00923
MTB 10b3	-0.292	0.566	0.446	0.977	0.939	0.969	0.0398	0.00186	-0.0036
MTB 10b4	0.218	0.216	0.887	0.991	0.956	0.999	0.0226	0.0018	0.00144
MTB 10b8	-1.66	2.04	0.845	0.916	0.887	0.935	0.0379	0.0122	-0.0158
MTB 10b9	-2.61	3.85	-0.517	0.871	0.833	0.875	0.0662	-0.00753	0.0205
MTB 10b10	-2.7	0.947	-4.51	0.866	0.879	0.79	0.0241	-0.0435	0.0119
MTB 10b11	0.0252	-0.0728	0.525	0.997	1.02	1.01	0.0176	0.00157	-0.000396

Table 3.1 Mean values of the identified accelerometers offsets and gains parameters: old procedure. Refer to figure 3.11.

Sensors	**Offsets**			**matrix C gains**					
	o_x	o_y	o_z	g_{xx}	g_{yy}	g_{zz}	g_{yx}	g_{zx}	g_{zy}
MTB 10b1	0.528	0.412	0.6	0.0477	0.0429	0.0542	0.0102	0.0169	0.0152
MTB 10b2	0.424	0.68	0.297	0.0396	0.0538	0.0326	0.0159	0.00722	0.0103
MTB 10b3	0.579	1.03	0.118	0.0579	0.0819	0.0411	0.0231	0.00355	0.00456
MTB 10b4	0.585	0.773	0.0956	0.0534	0.0637	0.0337	0.0194	0.00309	0.00362
MTB 10b8	4.81	4.44	1.71	0.173	0.164	0.129	0.0384	0.0149	0.0177
MTB 10b9	4.55	7.9	2.33	0.163	0.215	0.148	0.0377	0.0102	0.02
MTB 10b10	4.75	1.87	7.87	0.173	0.149	0.218	0.00844	0.0441	0.0145
MTB 10b11	0.171	0.113	0.302	0.0191	0.0152	0.0297	0.00395	0.0087	0.00618

Table 3.2 Uncertainties (standard deviation) in the identification of the accelerometers offsets and gains parameters: old procedure. Refer to figure 3.11.

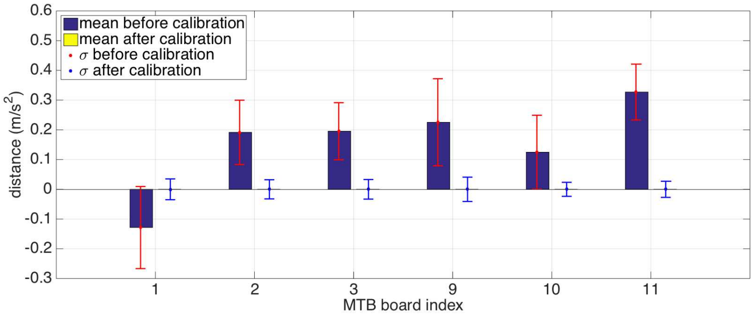


Fig. 3.10 Calibration of the accelerometers MTB 10B1 to 10B3 on the upper leg and 10B9 to 10B11 on the lower leg. We depict the mean value and standard deviation of the distributions as described in fig. 3.9, before and after the calibration.

adding a constraint on the offset turned to be the most efficient solution, considering the limits in the iCub leg mobility. We perform the calibration in two steps: in the first step we calibrate the offset of the accelerometer; in the second step we use the identified offset as a constraint in the original full ellipsoid-fitting algorithm.

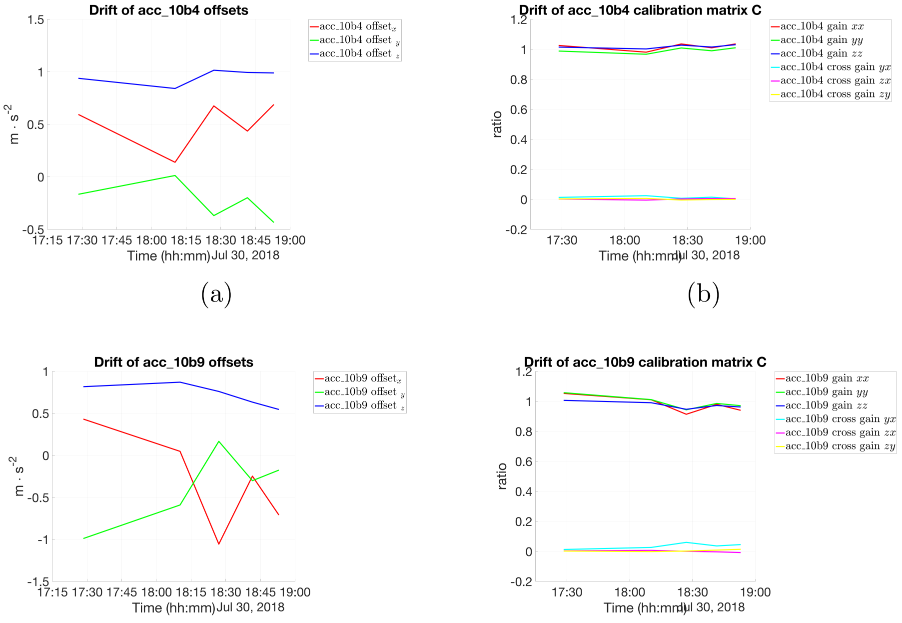
First step - standalone offset o^* identification: At this point, we exploit a method similar to the *6-point tumble calibration*, but just for calibrating the offset calibration. We perform two measurements of the same acceleration vector (for instance the gravitation vector $-g$ in static conditions), for two orientations ${}^{\mathrm{S}_1}R_{\mathrm{W}}$ and ${}^{\mathrm{S}_2}R_{\mathrm{W}}$ of the sensor frame S, with respect to the world frame W, such that ${}^{\mathrm{S}_1}R_{\mathrm{S}_2} = {}^{\mathrm{S}_1}R_{\mathrm{W}}\,{}^{\mathrm{S}_2}R_{\mathrm{W}}^{\top} = -\mathbb{I}_3$. The respective projections of the same acceleration vector in the sensor frame are then parallel with opposite orientations:

$$\left.\begin{aligned} {}^{\mathrm{S}_1}g &= -{}^{\mathrm{S}_1}R_{\mathrm{W}}\,{}^{\mathrm{W}}g \\ {}^{\mathrm{S}_2}g &= -{}^{\mathrm{S}_2}R_{\mathrm{W}}\,{}^{\mathrm{W}}g \end{aligned}\right\} \iff {}^{\mathrm{S}_1}g = {}^{\mathrm{S}_1}R_{\mathrm{W}}\,{}^{\mathrm{S}_2}R_{\mathrm{W}}^{\top}\,{}^{\mathrm{S}_2}g = {}^{\mathrm{S}_1}R_{\mathrm{S}_2}\,{}^{\mathrm{S}_2}g$$

$${}^{\mathrm{S}_1}g = -{}^{\mathrm{S}_2}g \tag{3.23}$$

${}^{\mathrm{S}_1}R_{\mathrm{S}_2} = -\mathbb{I}_3$ represents a rotation of $\pi\,\mathrm{rad}$ about a unit rotation axis denoted by $n_{\mathrm{R}} \in \mathbb{R}^3$, orthogonal to the vector g.

Proof. Euler's rotation theorem states that in three-dimensional space, any displacement of a rigid body such that a point on the rigid body remains fixed,



(a) (b)

(c) (d)

Fig. 3.11 Uncertainty analysis of the identified accelerometer parameters through a series of five calibration iterations. These results were obtained from the old procedure where o and C are fitted simultaneously. The uncertainty affects mostly the offsets.

is equivalent to a single rotation about some axis that runs through the fixed point. This means that any composition of rotations resulting in a rotation matrix R represents a single rotation about an axis n_{R} (unit vector in $\mathbb{R}^3$) of a given angle ψ. The rotation matrix can then be defined with respect to the rotation vector $\mathbf{\Psi} \coloneqq \psi n_{\mathrm{R}}$ as follows [Mäkinen, 2008, Section 3]:

$$R \coloneqq \mathbb{I}_3 + \frac{\sin\psi}{\psi}\Psi^{\wedge} + \frac{1-\cos\psi}{\psi^2}\left(\Psi^{\wedge}\right)^2 \tag{3.24}$$

For $\psi = \pi$, we get:

$$R = \mathbb{I}_3 + 0_{3\times 3} + 2\left(n_{\mathrm{R}}^{\wedge}\right)^2, \tag{3.25}$$

and if the rotation axis n_{R} is orthogonal to the rotated vector $x \in \mathbb{R}^3$, we get:

$$R\,x = \mathbb{I}_3\,x - 2\,\mathbb{I}_3\,x = -x \tag{3.26}$$

□

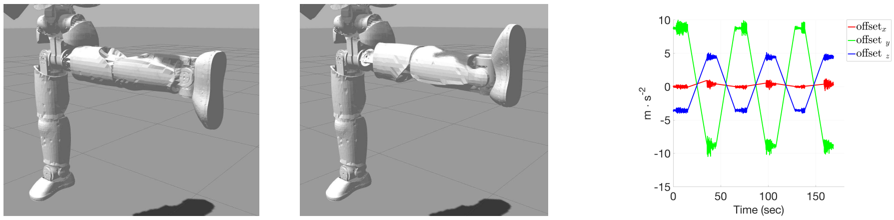
Note. *For an accelerometer fixed to a body link L_i of a given kinematic chain, rotating with a parent joint J_i, these measurements can be achieved through the following steps: (1) align the joint rotation axis with the horizontal plane, i.e. in a direction perpendicular to the gravity vector g; (2) acquire the first measurement r_1; (3) apply the defined rotation ${}^{\mathrm{S}_1}R_{\mathrm{S}_2}$ by rotating the joint J_i by πrad; (4) acquire the second measurement r_2.*

We consider r_1 and r_2 the respective raw measurements of ${}^{\mathrm{S}_1}g$ and ${}^{\mathrm{S}_2}g$. In view of (3.15) and (3.23), and all vectors being expressed in the sensor frame, we can write:

$$\left.\begin{array}{l} r_1 = C^{-1}\,{}^{\mathrm{S}_1}g + o \\ r_2 = C^{-1}\left(-{}^{\mathrm{S}_1}g\right) + o \end{array}\right\} \iff \boxed{o = \frac{1}{2}\left(r_1 + r_2\right)} \tag{3.27}$$

where $r_i \in \mathbb{R}^3$, $a \in \mathbb{R}^3$, $o \in \mathbb{R}^3$, and $C \in \mathbb{R}^{3\times 3}$.

We run those measurements on the same iCub left leg. The total angle rotation required between the two orientations is greater than the range allowed (between hard limits or cover limits) for each of the joints considered alone. Despite that limitation, we can combine two joints in singular configuration for meeting that requirement: in the case of the iCub leg, we combine the hip pitch with the hip yaw when both joints axes are aligned, i.e. the hip roll is set to 90 deg. In that configuration the two axes rotation vectors have the same orientation. For the two opposite sensor frame orientations S_1 and S_2 we set the respective two joint configurations $q_{(1)}$ and $q_{(2)}$, as illustrated in 3.12.



(a) (b) (c)

Fig. 3.12 Visualization (a)(b) of the left leg accelerometers offsets ”tumble” like calibration: joint configurations (a) $q_{(1)} = [-30, +90, -60, 0, 0, 0]^\top$ and (b) $q_{(2)} = [+30, +90, +60, 0, 0, 0]^\top$. The respective plot (c) of the proper acceleration of accelerometer MTB 11, result of the actual experiment on the real robot.

Second step - calibration matrix C^* identification: As described in Section 3.4.2, the fitting of the ellipsoid model is done through a non linear optimization using the *Levenberg-Marquardt* algorithm, which can take boundary constraints for the fitted variables. We now exploit this feature applying upper and lower boundaries, **lb_center** and **ub_center**, to the ellipsoid centre fitting. These boundaries are computed from the offset calibrated in the first step **o**:

$$\begin{aligned} \textbf{lb\_center} &= \mathbf{o} - 0.01 \\ \textbf{ub\_center} &= \mathbf{o} + 0.01 \end{aligned} \tag{3.28}$$

and then set in the parameterization of the *lsqnonlin* solver in the *ellipsoidfit* function which was modified for that purpose with minor changes.

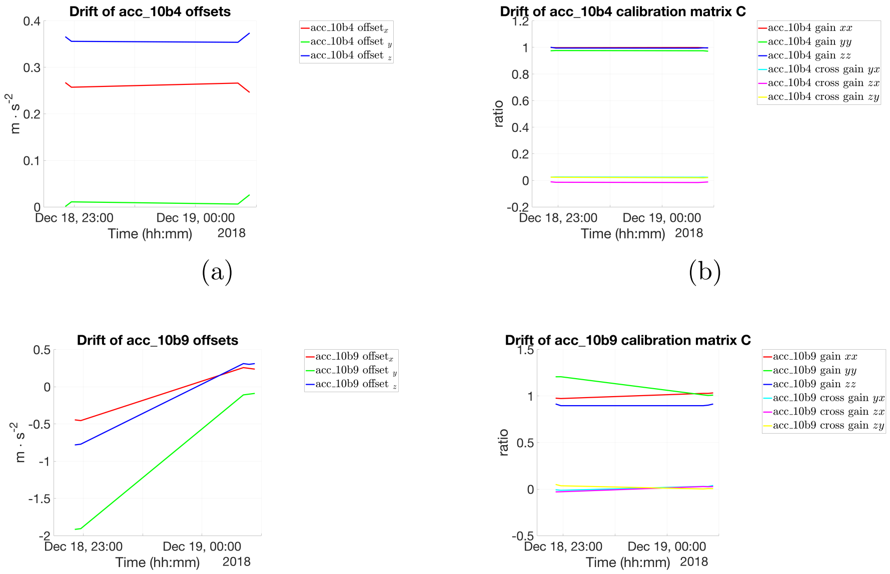
Experiment Results: For validating the new calibration procedure, we first verified that the identified offsets were consistent over several trials. Actually, a single iteration of the new standalone offset calibration repeats three times the measurement of the pair r_1 and r_2, and averages the measurements. We compared four iterations of the offset calibration: at day one two consecutive iterations were performed, a few minutes apart; seven days later, we repeated the procedure. The iterations performed the same day a few minutes apart give identical results (the differences were below $10^{-2}\,\mathrm{m\,s^{-2}}$). When comparing any pair of iterations seven days apart from each other, the differences can be one order of magnitude of the identified offsets. So we can consider that the accelerometer offsets drift quite fast, and probably require an online slow compensation.

We now have to check if the axes gains and cross-sensitivity gains drift as well. For that purpose we ran the full calibration using the identified offsets as a constraint. Once more we analyzed the variance of the identified parameters over a series of three consecutive calibration iterations. Two sets of iterations were run eight days apart: a first set of two iterations, a second with three. For each iteration, both the offsets then the gains were calibrated. The results, displayed in figure 3.13 and tables 3.3 and 3.4, show significant improvements, with a standard deviation reduced by an order of magnitude for most of the accelerometers. Accelerometers MTB_1 to MTB_4 show small variations in the offsets and constant gains, while accelerometers MTB_8 to MTB_11 show significant offset changes and some variability in the gains between the two sets of iterations. But this could be a side effect of the significant and permanent vibrations of the motors on the leg inducing additional noise in the inertial measurements. We can see such vibrations in figure 3.12 when calibrating the offsets with the *tumble like* procedure.

3.4.3 Joint encoders calibration

Trajectory for generating the training data set The leg joints are equipped with absolute joint encoders. We remind that the kinematic model gives us the encoders pose with an uncertainty up to 3 or 4 degrees due to the assembly inaccuracy, so we are doing fine calibration here. As proposed in 3.3.2, we run a minimal trajectory for capturing the training data set from the leg sensors (joint encoders and link accelerometers), staying in the same working conditions as during the calibration of the accelerometers, i.e. the measurements lie in the manifold of constant gravity norm $\mid g \mid$. For more simplicity, the joints are numbered from the base to the foot, $J_1 \longrightarrow J_6$, matching respectively the leg joints hip pitch / roll / yaw, knee, ankle pitch / roll in that order. The requirements described in 3.3.2 apply to the iCub leg as follows. First we group the leg DoFs depending on the position of the accelerometers:

- the gravity vector is assumed to be known in the base link frame
- the accelerometers are located on the links *upper leg*, *lower leg*, *foot*.
- So the grouped DoFs are: hip pitch / roll / yaw, (1x 3-DoFs joint); knee, (1x 1-DoF); ankle pitch / roll, (1x 2-DoF) joint



(a) (b)

(c) (d)

Fig. 3.13 Uncertainty analysis of the accelerometer parameters identified with the new procedure: o and C fitted separately. These are reprocessed data. Real dates are: 06/12/2018 (first two iterations); 14/12/2018 (Three last iterations). The offsets were calibrated the same day and hour as the gains.

Sensors	**Offsets**			**matrix C gains**					
	o_x	o_y	o_z	g_{xx}	g_{yy}	g_{zz}	g_{yx}	g_{zx}	g_{zy}
MTB 10b1	0.238	0.188	-0.0756	0.999	0.979	0.995	0.0157	0.00772	-0.0109
MTB 10b2	0.269	0.248	0.154	1.01	0.995	0.983	0.015	-0.00702	0.00565
MTB 10b3	0.244	0.0417	0.228	1.03	0.986	0.977	0.0253	-0.00733	0.00365
MTB 10b4	0.259	0.0121	0.363	0.998	0.974	0.997	0.024	-0.0137	0.0212
MTB 10b8	-0.0498	-0.821	-0.237	0.98	1.07	1.01	-0.00127	0.015	0.0447
MTB 10b9	-0.032	-0.824	-0.126	1.01	1.09	0.905	0.0152	0.00426	0.0195
MTB 10b10	0.00598	-0.759	-0.111	1	0.968	0.973	0.00478	0.012	-0.06
MTB 10b11	0.00519	-0.689	-0.0562	0.999	1.04	1.06	-0.00895	-0.0106	0.0426

Table 3.3 Mean values of the identified accelerometers offsets and gains parameters: new procedure. Refer to figure 3.13.

Sensors	**Offsets**			**matrix C gains**					
	o_x	o_y	o_z	g_{xx}	g_{yy}	g_{zz}	g_{yx}	g_{zx}	g_{zy}
MTB 10b1	0.0165	0.0137	0.0124	0.00267	0.00322	0.00275	0.00371	0.00321	0.00196
MTB 10b2	0.0138	0.0139	0.00802	0.00377	0.00474	0.00276	0.00524	0.00205	0.00159
MTB 10b3	0.00982	0.00562	0.00496	0.00136	0.00243	0.00439	0.00149	0.00143	0.00106
MTB 10b4	0.00768	0.00869	0.00722	0.00141	0.00163	0.00229	0.00075	0.00187	0.00178
MTB 10b8	0.383	1.02	0.59	0.0311	0.107	0.0506	0.0227	0.0182	0.0417
MTB 10b9	0.341	0.887	0.531	0.0271	0.0968	0.00818	0.0211	0.0273	0.0197
MTB 10b10	0.322	0.789	0.467	0.0302	0.00758	0.0407	0.0247	0.02	0.0363
MTB 10b11	0.272	0.719	0.405	0.0196	0.0333	0.047	0.0279	0.00647	0.0494

Table 3.4 Uncertainties (standard deviation) in the identification of the accelerometers offsets and gains parameters: new procedure. Refer to figure 3.13.

We then apply the control rules described in 3.3.2:

- **3-DoF hip joint:** select fixed appropriate positions for the first two joints (*pitch*,*roll*), for instance $J_1 = 0\,deg$, $J_2 = 20\,deg.$. Rotate J_3 spanning its reachable range. Whatever the angular position of J_3, the axis of J_4 (knee) will never be aligned with the gravity vector.

- **1-DoF knee joint:** select one fixed position, for instance $J_4 = 0$, which reduces the operation space.

- **2-DoF ankle joint:** select one fixed position for the joint pitch, for instance $J_5 = 0$. Rotate J_6 (*roll*) over its reachable range.

Checking the sanity and accuracy of the obtained results The obtained data across all the poses was distributed randomly into 5 buckets of 200 samples, each bucket then being an input to the least squares optimization. The mean and the standard deviation across the 5 solution vectors $\delta\theta^*$ were computed for testing their accuracy.

The method was implemented in MATLAB using the Matlab Optimization Toolbox solver *lsqnonlin* along with the *trust-region-reflective* algorithm, with a termination tolerance of 10^{-7} on the function and of $0.1deg$ on $\delta\theta$ since that corresponds to the joint encoders resolution (refer to the appendix A.1 for more details). This method has been made freely available under an open source license[5].

Note. *Although the resolution of the joint absolute encoder AS5045 (used on the main iCub joints) is quite high for a magnetic encoder, bringing the LSB value down to* $0.1\,\mathrm{deg}$*, its accuracy is much lower, around* $0.5\,\mathrm{deg}$*, as it depends on the non-linearity due to the sensor signal processing and amplification, and to the magnet sensor misalignment. As a result, the measurements are subject to the Integral Non-Linearity error, which typical value is* $0.5\,\mathrm{deg}$ *(A.1.1, table A.1).*

The *trust-region-reflective* algorithm was preferred to the *levenberg-marquardt* since our example system is an over-constrained problem: indeed the iCub leg is equipped with 13 accelerometers providing as many measured acceleration vectors that the expected accelerations have to match, while the leg has only 6 DoF. On top of that, the *trust-region-reflective* algorithm doesn't require the gradient of the error function for searching its minimum, although this would potentially improve the accuracy and the processing performances.

---

[5] `https://github.com/robotology-playground/sensors-calib-inertial.git`

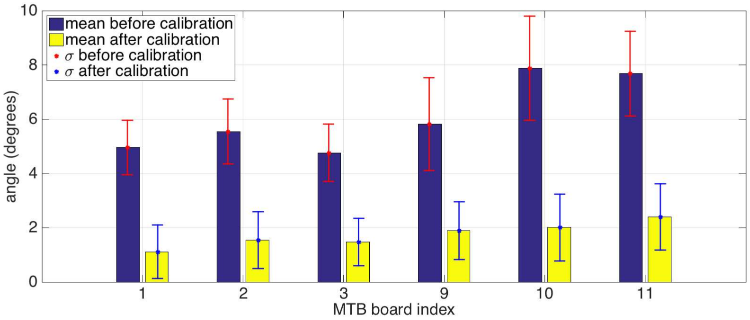


Fig. 3.14 The Angle of sensors measurement VS estimation.

3.4.4 Encoder calibration results

We depict in table 3.5 the joint offsets obtained from the least squares optimization equation (3.4). We can see that there is no major difference with the offsets measured manually. The gap is always below 1 degree, except for joints J_3 and J_6. Further work is ongoing for analyzing the accuracy of these results, as well as a thorough observability study. The standard deviation across 5 buckets of the capture data is under or equal 0.1 degrees. The low magnitude of the standard deviation is indicative of the robustness of the proposed method to the pose of the robot, i.e. regardless of what poses are considered for the minimization, in the range of reachable positions for a 3-DoF joint (3.4.3), the resulting optimal joint offsets are nearly identical. This robustness is essential in order to be able to generalize the approach across different robot morphologies and taken into account practical considerations in their deployment. For validating the calibration results, we evaluate the angle error between the measured and the estimated gravity vector across all the sensors, but this time running a different joint trajectory, for instance following the same type of grid shaped trajectory defined in Section 3.3.4 for calibrating the accelerometers. As we can see in Fig. 3.14, this error has been overall reduced by an average factor of 3.

3.5 Conclusion

In this chapter, we propose two contributions in the context of the calibration of joint position and link inertial sensors. We first propose a joint offset identification methodology which uses a distributed set of accelerometers,

Left leg joints		$\mathbf{J_1}$	$\mathbf{J_2}$	$\mathbf{J_3}$	$\mathbf{J_4}$	$\mathbf{J_5}$	$\mathbf{J_6}$
Offsets (deg)	Manual calibration	0.1	-4.0	-0.5	-3.8	-4.3	4.3
	Auto-calibration	-0.1	-4.7	2.5	-4.5	-4.6	6.0
	Gap	0.2	0.7	-3	0.7	0.3	-1.7

Table 3.5 Example of joint offsets (degrees) estimated from manual calibration or from automatic calibration using accelerometers.

but in a simpler procedure with respect to prior existing methods. All the model kinematic parameters but the joint encoder fine offsets are assumed to be known: the link kinematic parameters (orientation and position between two consecutive joint axes), the joint orientations and rough offsets. The joint offsets are known with an error of $\pm 5°$, and the identification procedure goal is to decrease that error to a value within $\pm 1°$. The estimation is done in a single step after acquiring the inertial measurements while executing a single slow motion trajectory.

The problem of identifying the joint offsets was then posed as an unconstrained nonlinear least squares optimization minimizing the gap between expected and measured accelerations, acquired in different joint configurations. We demonstrated the requirements for getting a global minimum solution. For that purpose, a single slow trajectory was defined, running through a minimal set of joint configurations, while acquiring accelerometers data, satisfying two main requirements: the measured acceleration was always only due to the gravitational acceleration; the joint axis was never aligned with the gravity vector, which would make the measurement insensitive to any rotation offset. The method was validated on one kinematic chain of the robot iCub, and the accuracy of the offset computation was demonstrated by: comparing the results to offsets measured manually, although manual calibration cannot be considered a ground truth; measuring the angle deviation between the expected and measured accelerations; performing a qualitative test where the calibrated kinematic chain is controlled in torque, gravity compensation is applied and joint friction is partially compensated (in this configuration, the joints are subject to drift in case the encoder offsets are not properly compensated).

The algorithm we have presented can work in a chain-wise fashion, thus each limb of a humanoid can be calibrated either independently or simultaneously depending on the need. there is no need to reach the joint limits. This in particular is ideal for robots that cannot reach joint limits due to collisions with plastic casings and problems in wiring. Two additional requirements for our method are however: the availability of multiple accelerometers with at least one every three links—given the low-cost easy availability of MEMS inertial sensors, our approach makes a strong case for incorporation of such sensors on all links of a humanoid robot; the need for accurate knowledge of link to sensor frame pose—this may not be an issue for commercial humanoids as accurate CAD diagrams can be typically utilized to extract this information.

The second contribution is about identifying in-situ the full calibration matrix and axes offsets of any accelerometer mounted on the kinematic chain, without knowing the sensor pose with respect to the link it is attached to, nor the joint configurations. Additionally, we give a particular attention to the identification of the cross-axis gains and their dependency with respect to the drifting axes offsets, unlike other existing methods which assume the calibration matrix to have only diagonal terms.

We defined a grid shaped trajectory for generating the spherical manifold of a constant vector (Earth's gravitation field vector) spanning all the directions. The obtained manifold is a fraction of the sphere due to the joint limits. The location of the fraction on the sphere depends on the orientation of the sensor frame with respect to the world frame. But since the physical quantity to be measured is isotropic, any sheer or stretch effect due to the sensor defects (offsets, differences between the axes gains and the cross-axis sensitivity) can be characterized w.r.t. the sensor frame alone, i.e. the fitted model can be expressed in the sensor frame regardless of the frame orientation in space. Hence the model parameters do not depend on the pose of the sensor in the link frame, and even less on the joint configuration. The model mapping the raw measurements to the measured accelerations was assumed affine and identified with an ellipsoid fitting algorithm from an open source library. Further we defined the conversion from the ellipsoid explicit parameters to the sensor calibration parameters. The method was validated on one kinematic chain of the robot iCub, which is equipped with a dozen of single proof-mass MEMS accelerometers The method required an improvement consisted of a prior, separate identification of the offsets, still not dependent on the sensor frame orientation, but requiring the first joint in the kinematic chain to be orthogonal to Earth's gravitational field vector. Once identified, the offsets were used as an additional constraint in the fitting algorithm, which improved significantly the results repeatability.

As a possible future extension of this work, the source of the accelerometers offsets drift could be analyzed, in order to implement some compensation process. In addition, the algorithm could be improved in order to relax some constraints on the trajectories that can otherwise raise issues with respect to the joint limits. Among these constraints we can find the following: the rotation axis has to be orthogonal to the gravitational field vector; the angle between the two extreme positions used in the accelerometers offsets calibration has to be of 180° for removing the dependency on the calibration matrix.

Regarding the validation of the identified encoders offsets, the qualitative tests in torque control with gravity compensation can be improved by reducing further the friction, hence increasing the sensitivity to the encoders offsets. This can be achieved typically by adding dithering like it was done in [Capurso et al., 2017], and using the friction parameters identified in Chapter 2.

In order to run the joint calibration procedure faster and anywhere, we need to get rid of any external fixture previously required as a support while moving the legs. For that intent, an accelerometer should be mounted on the base, and the joint trajectories should be integrated in a balancing task, for just relying on the ground contact, and eventually a third temporary contact as a support. On top of that, the algebraic computation of angular accelerations would improve the Forward Kinematics and allow to include the accelerometer coordinate acceleration in (3.3). This would make the algorithm applicable for faster motions.

The calibrated accelerometers can be used along with gyroscopes for estimating the joint accelerations algebraically, i.e. without numerical differenciation, as presented in Chapter 4. Once the angular and linear link accelerations are available, along with the accurate joint positions, these can be used directly in inverse dynamics algorithms [Traversaro, 2017, Chapter 4, Section 4.4.2] decomposed in a “sensor-based net force-torque estimation, followed by a joint torque estimation. This approach reduces the errors otherwise accumulated when propagating the link accelerations through the composition of joint accelerations from the root to the leafs of a kinematic tree.

Chapter 4

Link Angular Acceleration Estimation

4.1 Introduction

The link angular acceleration can be used in dynamics computations that estimate: the motion compensated inclination with respect to the gravitation vertical vector; the joint positions, velocities and accelerations without relying on joint encoders; the rate of the angular momentum of a humanoid robot in a whole-body controller loop; or the external wrenches and internal joint torques.

Joint torques can be measured by joint torque sensors, or estimated in a framework using embedded force-torque sensors and inverse dynamics [Nori et al., 2015b] [Traversaro, 2017], or just joint encoders measurements and the robot dynamic model [Capurso et al., 2017], [Linderoth et al., 2013] [Stolt et al., 2015a]. For the third option, a model observer estimates the external forces based on the deviation of the measured link momentum from the value predicted by the model. The force is estimated by solving a convex optimization problem. Because of the high cost of joint torque sensors, the mentioned alternatives are often preferred. In [Traversaro, 2017, Chapter 4], the framework estimating joint torques using force-torque sensors and inverse dynamics can take directly link accelerations as described in [Traversaro, 2017, Chapter 4 section 4.4.2], referred to as the sensor-based net force-torque estimation, followed by the joint torque estimation. This approach reduces the errors propagation, while speeding up the computations, since we relate only on local data from sensors attached to the link closest to the target joint. Hence, the first step is to improve the estimation of link linear and angular accelerations. These are commonly estimated through the forward kinematics and joint position derivatives [Khalil and Dombre, 2004, chap 5] [Featherstone, 2014, sections 2.2, 4.4]. Joint positions can be directly measured by rotary encoders of two possible distinct types: absolute or incremental. Absolute encoders produce a unique digital code for each distinct angle of the shaft, while most common designs of incremental encoders produce two cyclical quadrature signals. The joint velocity can be obtained by measuring the digital signal change frequency in the first case or the cyclic output signals frequency in the second case, or even by computing a numerical derivative approximation of consecutive positions. As the numerical differentiation is very sensitive to high frequency noise affecting the encoder measurements, it requires filtering and thus introduces lag, which can compromise the system controller performance. In any case, these methods produce measurements with a limited resolution. For this reason, in this chapter we discuss on an alternative method for estimating link accelerations using other sensor modalities. Gyroscopes and inertial sensors like accelerometers, as well as a combination of these sensors as we can find in IMUs, are good candidates, providing more accurate and high resolution measurements.

4.2 Related Work and Innovation

4.2.1 An alternative to direct sensing

The research on the estimation of link velocities and accelerations directly from IMUs mounted on these links is getting a growing attention, in particular MEMS (MicroElectroMechanical Systems) based IMUs. MEMS based inertial sensors are low cost, robust and immune to local magnetic disturbances. The estimation of joint velocities and accelerations from the respective link kinematics is quite trivial. But while angular velocities are directly measurable by gyroscopes, it is not the case for link angular accelerations. It's hard to find sensors that measure directly angular accelerations, as small as MEMS. For instance, Columbia research laboratories provide angular accelerometers, like the *SR 100FR* [1] shown in figure 4.1, based on a fluid rotor concept, highly accurate, that provide good bias stability and linear acceleration rejection. The drawback is its size and power consumption inappropriate for light duty robotics systems or humanoid robots: the *SR 100FR*

---

[1] `https://www.crlsensors.com/product.cfm?cat=force-balance&sub=angular-accelerometers&prod=sr-100fr`


(a) (b)

Fig. 4.1 Three-axes direct sensing angular acclerometers: (a) fluid rotor model SR-100FR by Columbia Research Laboratories, providing high sensitivity from 0.1. Source: https://`www.crlsensors.com`; (b) piezoresistive model 7302BM4 by MEGGITT, of much smaller dimensions, but with lower sensitivity (5mV per $\mathrm{krad\,s^{-2}}$). Source: http://www.akron.be.

weights 1.58kg, has a diameter of 16cm, which is for instance, respectively 5% of a iCub's total weight and almost as large as iCub's head. Compared to a MEMS sensor, it can draw thirty times more power than a single of the MEMS gyroscopes used on iCub (ST Microelectronics L3GD20H). Another sensors manufacturer, MEGGIT, proposes a smaller model, the 7302BM4 [2], piezoresistive, of much smaller dimensions, but far less sensitive than the *SR-100FR*. These are a couple of examples to illustrate the usual compromise between small size and high sensitivity.

The current alternative is to use the fused measurements from accelerometers and gyroscopes and a model relating the angular acceleration with the linear accelerations across a distributed set of body points, and this way, sensing the body full motion state.

In [Vihonen et al., 2013] a configuration with single-axis and biaxial accelerometers are combined into a set of 6 axes, and completed with one triaxial gyroscope, as shown in Fig. 4.2, for estimating the full joint motion state, i.e. position, angular velocity and acceleration, of all three joints in a coplanar kinematic chain. The authors first propose an algebraic, fast method (with a lag below 1ms) for estimating each joint angular acceleration using the multi-MEMS configuration instead of any differentiation technique, and for that reason referred to as "direct sensing". When compared to the double differentiation of the joint encoders or the direct differentiation of a gyroscope's measurements, the method scores improved accuracy, with up to five times less error, even if still subject to significant high frequency noise.

---

[2] http://www.akron.be/pdf/endevco/7302BM4.pdf

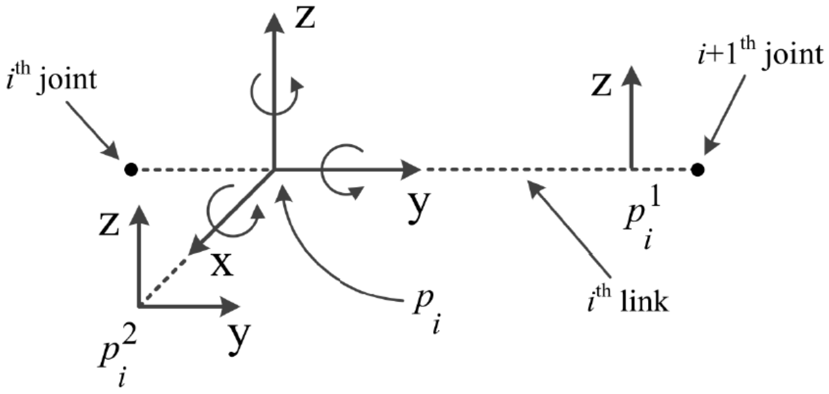


Fig. 4.2 Configuration of six linear accelerometers (single axis) for the direct sensing of angular acceleration. Straight arrows are accelerometer axis, arced arrows are gyroscope axis. Source: Vihonen et al. [2013]

The authors then use the estimated angular acceleration for distinguishing the link coordinate acceleration from the gravitation acceleration in the inertial sensor's measurements, in the context of sensing the inclination of a link with respect to the gravity vertical. Although the result turns to be too noisy for the inclination sensing itself, the author used it effectively in the estimation of the joint positions, achieving a position error below $\pm 1\,\text{deg}$. For that purpose, they implemented a discretized PI-type complementary filter where the motion compensated inclination was used for bias cancellation. The filter gave very good results compared to encoders benchmarks. Regarding the joints angular acceleration, the estimation was done iteratively on one DoF at a time, from the base to the end effector, as it depended on the composition of accelerations in the supporting tree [3].

That approach was improved later by the same authors in [Vihonen et al., 2015].

In Rotella et al. [2016] the authors proposed a similar solution formulated in a simpler way, also using six accelerometer axes for each link, hence the equivalent of two IMUs, and applicable to floating base systems. Each joint acceleration $\ddot{\theta}_i$ is estimated from the joint velocity $\dot{\theta}_i$ and the linear accelerations measurements from one IMU on the parent link $i-1$, and from two IMUs on the link i [4], respectively denoted by $\overline{a}_{i-1}$, $\overline{a}_i$, and $\tilde{\overline{a}}_i$ in the article.

---

[3] Joints in the kinematic path between the base and the current processed joint.

[4] The link $i-1$ is the parent link of joint i, which supports the link of same index i.

4.2.2 Towards a local acceleration estimation

Our approach is similar to the first two, is also applicable for floating base systems, but directly estimates the link angular acceleration without requiring the computation of joint velocities $\dot{\theta}_i$, the joint positions defining the rotation ${}^{i}R_{i-1}$, nor the measurements from any IMU fixed to the parent link $\overline{a}_{i-1}$ nor any other link. All the measurements are local to the link being estimated. We pose the problem as a linear system, similarly to what is done in [Rotella et al., 2016]. Each sensor proper acceleration [5] depends on the link's angular velocity $\omega_{A,L}$ and acceleration $\dot{\omega}_{A,L}$, the linear coordinates acceleration of the link itself ${}^{A}\ddot{p}_{A,L}$, the pose of the sensor frame with respect to the inertial frame A, and the gravitation acceleration $\mathbf{g}$. We compute and compare the linear accelerations of two sensors, which allows us to remove the dependency to the link orientation ${}^{L}R_{A}$, the link acceleration ${}^{A}\ddot{p}_{A,L}$, as well as the gravity, and depend on the relative poses of the sensor frames w.r.t. each other, instead of the absolute poses. As for [Rotella et al., 2016], we relate these two IMU with a third one fixed on the link for getting a unique solution for $\dot{\omega}_i$. A minimal filtering can be used for canceling or at least attenuating the inertial sensors noise, as the sensors distribution in a small sized link might make the algorithm very sensitive to measurement errors. It is crucial to define a filtering that would not introduce too much lag. The system is easily scalable, as the algorithm update is trivial for each new added IMU to the predefined set.

Note. *We don't assume that the accelerometers are assembled as three single axis accelerometers, but rather single proof-mass MEMS, as it is the case in the iCub sensor framework. Hence we consider the worst case scenario where the accelerometers have a cross-axis sensitivity, which we can compensate by applying the calibration procedure seen in chapter 3, section 3.3.4.*

We will address how the relative IMU positions condition the angular acceleration observability. As a first intuition, it is likely that the IMU positions should be as far as possible from each other, as for a given rotation velocity, the contribution of the noise to the final estimation error should decrease with that distance . For instance, we can use the configuration depicted in fig. 4.2 replacing p_i^1 and p_i^2 by one IMU each.

This paper is organized as follows : in Section 4.3 we present the sensor framework and the background concepts; in Section 4.4 we describe the improved acceleration estimation method, the filtering done on the measurements and the estimated quantities, and the application of the method to

---

[5] appx:rigidBodyDynamics-frameKinematics-properAcc

the whole-body dynamics computations; Section 4.5 details the experiments that were conducted and the obtained results; finally, Section 4.6 draws the conclusions.

4.3 Background

4.3.1 Notation and Definitions

The notation used for describing the models and the algorithms in this chapter is based on the notation presented in section B. We list below the simplified notation specific to this chapter:

A, B, S_k	cartesian coordinate frames
S_k	cartesian coordinate frame of sensor k
${}^A p_B$	coordinates of origin of frame B expressed in frame A
${}^B \alpha_{S_k}$	proper acceleration of sensor k projected on a given frame B
α_{S_k}	proper acceleration of sensor k expressed in sensor frame
$[A]$	orientation frame associated to A
$B[A]$	frame with same origin as frame B and same orientation as frame A
${}^A R_B$	rotation transformation ($\in SO(3)$) of a 3D vector from B to A
$x^\wedge$	3 x 3 skew symmetric matrix of vector x ($x \in \mathbb{R}^3$)
$Y^\vee$	vector such that $(Y^\vee)^\wedge = Y$ ($Y \in \mathbb{R}^3$)
${}^C \overline{v}_{A,B}$	linear component of ${}^C v_{A,B}$
${}^C v_{A,B} \times$	matrix representation of the twist cross product
${}^C v_{A,B} \times^*$	matrix representation of the wrench (twist dual) cross product
$\hat{x}$	estimate of vector x
${}^A g$	gravitational acceleration vector (${}^A g \in \mathbb{R}^3$) written in frame A
$\mathbb{I}_n$	identity matrix

We choose A as the world inertial frame.

4.3.2 Problem formulation and assumptions

We consider a single floating link L or a link from a kinematic chain but for which we don't have an accurate measurement of the support joints [6] positions, velocities and accelerations. The link is equipped with at least one

---

[6] joints in the sub-chain connecting the link to the robot base

gyroscope measuring the angular velocity, and two accelerometers S_1 and S_2 fixed to that link. We wish to define a linear system of kinematic equations allowing to estimate the link angular acceleration from the accelerometers and gyroscopes measurements.

The first step is to compute the linear part of the coordinate acceleration of any of the accelerometers sensor frames S_k as a function of the link coordinate acceleration, the link angular velocity and the sensor frames poses.

From now on all the velocities and accelerations are defined with respect to the world inertial frame denoted A, which will be omitted in the notation for the sake of clarity. We write the position of the sensor frame origin expressed in the world frame as follows:

$$
{}^{A}p_{S_k} = {}^{A}p_L + {}^{A}R_L\,{}^{L}r_{L,S_k}, \tag{4.1}
$$

where ${}^{A}p_{S_k}$ is the sensor frame origin position in the world frame and ${}^{L}r_{L,S_k}$ is its respective fixed position in the link reference frame. We then compute the sensor frame velocity through the first time derivative:

$$
{}^{A}\dot{p}_{S_k} = {}^{A}\dot{p}_L + {}^{A}\dot{R}_L\,{}^{L}r_{L,S_k} = {}^{A}\dot{p}_L + {}^{A}\dot{R}_L\,{}^{A}R_L^{\top}\,{}^{A}r_{L,S_k}, \tag{4.2}
$$

and, using the definition of the *right trivialized* angular velocity given in B.3.5, ${}^{A}\omega_L := {}^{A}\dot{R}_L\,{}^{A}R_L^{\top}$, we get:

$$
{}^{A}\dot{p}_{S_k} = {}^{A}\dot{p}_L + {}^{A}w_L{}^{\wedge}\,{}^{A}r_{L,S_k}. \tag{4.3}
$$

We now write the sensor frame acceleration as the second time derivative, applying the distributive property of the cross product in $\mathbb{R}^3$ with respect to the differentiation:

$$
\begin{aligned}
{}^{A}\ddot{p}_{S_k} &= {}^{A}\ddot{p}_L + {}^{A}\dot{w}_L{}^{\wedge}\,{}^{A}R_L\,{}^{L}r_{L,S_k} + {}^{A}w_L{}^{\wedge}\,\frac{d}{dt}\left({}^{A}R_L\,{}^{L}r_{L,S_k}\right) \\
&= {}^{A}\ddot{p}_L + {}^{A}\dot{w}_L{}^{\wedge}\,{}^{A}R_L\,{}^{L}r_{L,S_k} + {}^{A}w_L{}^{\wedge}\left({}^{A}w_L{}^{\wedge}\,{}^{A}R_L\,{}^{L}r_{L,S_k}\right)
\end{aligned} \tag{4.4}
$$

Given the definition of the accelerometer measurement in B.3, as being the sensor proper acceleration expressed in the sensor frame, we can write:

$$
\alpha_{S_k} := {}^{S_k}R_A\left({}^{A}\ddot{p}_{S_k} - {}^{A}g\right) \tag{4.5}
$$

In view of (4.4) and (4.5), we rewrite the same measurement expressed in the link frame:

$$\begin{aligned} {}^{L}\alpha_{S_k} ={}& {}^{L}R_{S_k}{}^{S_k}R_A \left({}^{A}\ddot{p}_L + {}^{A}\dot{w}_L{}^{\wedge}\,{}^{A}R_L{}^{L}r_{L,S_k}\right. \\ &\left. + {}^{A}w_L{}^{\wedge}\left({}^{A}w_L{}^{\wedge}\,{}^{A}R_L{}^{L}r_{L,S_k}\right) - {}^{A}g\right) \end{aligned} \tag{4.6}$$

$$\begin{aligned} {}^{L}\alpha_{S_k} ={}& {}^{L}R_A{}^{A}\ddot{p}_L + {}^{L}R_A{}^{A}\dot{w}_L{}^{\wedge}\,{}^{L}r_{L,S_k} \\ &+ {}^{L}w_L{}^{\wedge}\left({}^{L}w_L{}^{\wedge}\,{}^{L}r_{L,S_k}\right) - {}^{L}g \end{aligned} \tag{4.7}$$

Remark 4.1. *We did not simplify on purpose the elements* ${}^{L}R_A{}^{A}\ddot{p}_L$ *and* ${}^{L}R_A{}^{A}\dot{w}_L$*. We recall the meaning of* ${}^{B}\ddot{m}$ *for any position or motion vector* m *and any arbitrary frame* B*:*

$${}^{B}\ddot{m} := \frac{d^2}{dt^2}\left({}^{B}m\right).$$

Following the above definition, ${}^{L}\ddot{p}_L = \mathbf{0}$ *and so* ${}^{L}R_A{}^{A}\ddot{p}_L \neq {}^{L}\ddot{p}_L$*. The case of* ${}^{L}R_A{}^{A}\dot{w}_L$ *is less trivial. We can write:*

$$\begin{aligned} {}^{A}\omega_L &= {}^{A}R_L{}^{L}\omega_L \\ {}^{A}\dot{\omega}_L &= {}^{A}\dot{R}_L{}^{L}\omega_L + {}^{A}R_L{}^{L}\dot{\omega}_L \\ &= {}^{A}\dot{R}_L{}^{A}R_L^{\top}{}^{A}\omega_L + {}^{A}R_L{}^{L}\dot{\omega}_L \\ &= {}^{A}\omega_L{}^{\wedge}\,{}^{A}\omega_L + {}^{A}R_L{}^{L}\dot{\omega}_L\ , \end{aligned}$$

and since the first term is null, we get:

$${}^{A}\dot{\omega}_L = {}^{A}R_L{}^{L}\dot{\omega}_L\ ,$$

hence:

$${}^{L}R_A\,{}^{A}\dot{\omega}_L = {}^{L}\dot{\omega}_L\ , \tag{4.8}$$

We specify in section B.1, remark B.1, the definition of the dot operator $(\dot{-})$.

So (4.7) can be rewritten as follows:

$$\boxed{{}^{L}\alpha_{S_k} = {}^{L}R_A{}^{A}\ddot{p}_L + {}^{L}\dot{w}_L{}^{\wedge}\,{}^{L}r_{L,S_k} + {}^{L}w_L{}^{\wedge}\left({}^{L}w_L{}^{\wedge}\,{}^{L}r_{L,S_k}\right) - {}^{L}g} \tag{4.9}$$

From here, the common base idea is to define a series of relations like (4.7), relating other accelerometers and gyroscopes measurements, either from the same link L, either from multiple links, building up a linear system and solving it for ${}^{L}\dot{\omega}_L$.

4.4 Methodology

4.4.1 Base Algorithm: Local Estimation

System Definition

Starting from (4.9), it is a known approach to pose the difference between the measurements from two sensors ($\alpha_{S_0} - \alpha_{S_1}$), in order to cancel out the gravitational field and the link coordinate acceleration (and with it the dependency on the link pose with respect to the world frame), which is equivalent to what was done in [Rotella et al., 2016], [Vihonen et al., 2013] and [Vihonen et al., 2015], only these previous works use a composition of angular velocities and linear accelerations from neighbor links while here we will use kinematic variables and measurements solely from a single rigid body link L. We rewrite (4.9) as:

$$\left({}^{L}r_{S_1,S_2}\right)^{\wedge}\,{}^{L}\dot{\omega}_L = {}^{L}\alpha_{S_1} - {}^{L}\alpha_{S_2} + ({}^{L}\omega_L^{\wedge})^2\,{}^{L}r_{S_1,S_2}\ , \tag{4.10}$$

allowing us to get rid of the gravitational field ${}^{L}g$ and the link acceleration ${}^{A}\ddot{p}_L$. The goal is to solve the system for ${}^{L}\dot{\omega}_L$, but it does not have a unique solution since the matrix $\left({}^{L}r_{S_1,S_2}\right)^{\wedge}$ has rank two.

Proof. We consider the Hermitian inner product on a complex vector space $\mathbb{C}^n$ denoted by $\langle\cdot,\cdot\rangle$, verifying the following properties:

$$\begin{aligned}&\langle \lambda u, v\rangle = \lambda\langle u, v\rangle\\ &\langle u, \lambda v\rangle = \bar{\lambda}\langle u, v\rangle\\ &\text{with } \lambda\in\mathbb{C} \text{ and } u,v\in\mathbb{C}^n\ ,\end{aligned} \tag{4.11}$$

If a real matrix A is skew symmetric, it verifies $A^\top = -A$, and we can write:

$$\begin{aligned}\forall u,v\in\mathbb{C}^n,\quad &\langle Au, v\rangle = (A\,u)^\top\,v = u^\top A^\top v = -u^\top A\,v\\ &\langle u, -Av\rangle = -u^\top A\,v = \langle Au, v\rangle\\ \text{so,}\quad &\langle Au, v\rangle = \langle u, -Av\rangle\end{aligned} \tag{4.12}$$

Considering that every eigen value and respective eigen vector of A verify $Av = \lambda v$, and considering the inner product on $\mathbb{C}^n$ as well as the properties (4.11) and (4.12), we can write:

$$\lambda\langle v, v\rangle = \langle \lambda v, v\rangle = \langle Av, v\rangle = \langle v, -Av\rangle = \langle v, -\lambda v\rangle = -\bar{\lambda}\langle v, v\rangle\ ,$$

and we conclude that $\lambda = -\bar{\lambda}$, i.e. that λ is purely imaginary. For a real matrix, complex eigenvalues come in conjugate pairs, so the rank must be even, i.e. if $A \in \mathbb{R}^{3\times3}$, then $\text{rank}(A) = 2$. ☐

So we need at least a third accelerometer for obtaining a unique solution, hence we add an additional line to (4.10) as shown below:

$$\begin{bmatrix} ({}^L r_{S_0,S_1})^\wedge \\ ({}^L r_{S_0,S_2})^\wedge \end{bmatrix} {}^L\dot{\omega}_L = \begin{bmatrix} {}^L\alpha_{S_0} - {}^L\alpha_{S_1} + ({}^L\omega_L^\wedge)^2 {}^L r_{S_0,S_1} \\ {}^L\alpha_{S_0} - {}^L\alpha_{S_2} + ({}^L\omega_L^\wedge)^2 {}^L r_{S_0,S_2} \end{bmatrix} , \tag{4.13}$$

The augmented linear system, now over constrained, defines a linear least-squares optimization problem. If the matrix of sensor positions multiplying ${}^L\dot{\omega}_L$ has a rank three, i.e. if the accelerometers S_0,S_1 and S_2 positions are well distinct and well distributed, we can solve the problem for ${}^L\dot{\omega}_L$ through the computation of a Moore-Penrose pseudoinverse.

minimal number of sensors and required relative positions

For finding a unique solution to the system, the matrix of sensor positions needs to be full rank, i.e. rank three. In [Zappa et al., 2001] a similar analysis was done on a system with twelve single axis accelerometers and no gyroscopes, but computing the determinant of the square matrix of sensor positions. The additional accelerometers were required for compensating the absence of angular velocity measurement, which is not our case since we are using gyroscopes. The first trivial condition is to avoid singular cases (${}^L r_{S_0,S_1} \sim {}^L r_{S_0,S_2}$), placing the three accelerometers S_0, S_1 and S_2 at three distinct positions, as far as possible from each other. This condition is clearly mentioned in previous studies like [Rotella et al., 2016], but for identifying further conditions, we need to actually compute the rank. Let the sensor positions be defined as follows:

$${}^L r_{S_0,S_1} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} , \quad {}^L r_{S_0,S_2} = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} , \quad C = \begin{bmatrix} ({}^L r_{S_0,S_1})^\wedge \\ ({}^L r_{S_0,S_2})^\wedge \end{bmatrix} \tag{4.14}$$

We then perform a rank computation through a Gaussian Elimination (row reduction):

$$C = \begin{bmatrix} 0 & -z_1 & y_1 \\ z_1 & 0 & -x_1 \\ -y_1 & x_1 & 0 \\ 0 & -z_2 & y_2 \\ z_2 & 0 & -x_2 \\ -y_2 & x_2 & 0 \end{bmatrix} \underset{(1)}{\rightarrow} \begin{bmatrix} z_1 & 0 & -x_1 \\ -y_1 & x_1 & 0 \\ 0 & -z_1 & y_1 \\ 0 & -z_2 & y_2 \\ z_2 & 0 & -x_2 \\ -y_2 & x_2 & 0 \end{bmatrix} \underset{(2)}{\rightarrow} \begin{bmatrix} z_1^* & 0 & 0 \\ 0 & x_1^* & 0 \\ 0 & 0 & y_1^* \\ 0 & -z_2 & y_2 \\ z_2 & 0 & -x_2 \\ -y_2 & x_2 & 0 \end{bmatrix},$$

by applying the operations (1) and (2) specified below:

(1) swap rows 1 and 2, then 2 and 3 (counting from the top starting at 1),

(2) subtract row 5 $\times \frac{x_1}{x_2}$ from row 1, then subtract row 6 $\times \frac{y_1}{y_2}$ from row 2, then subtract row 4 $\times \frac{z_1}{z_2}$ from row 3.

From the Gaussian Elimination, we get a matrix with three pivots x_1^*, y_1^* and z_1^*. For C to have rank three, all pivots must be non null:

$$\left.\begin{array}{ll} z_1 x_2 - z_2 x_1 & \neq 0 \\ x_1 y_2 - x_2 y_1 & \neq 0 \\ y_1 z_2 - y_2 z_1 & \neq 0 \end{array}\right\} \iff \left({}^L r_{S_0,S_1}\right)^\wedge \left({}^L r_{S_0,S_2}\right). \tag{4.15}$$

(4.15) states that for the system to have a unique solution (rank$(C) = 3$), the sensors S_0, S_1 and S_2 should never be aligned. The analysis would be similar for a system scaled to an arbitrary number N of three-axis accelerometers, with $N \geq 3$.

4.4.2 A more Scalable Formulation

Although the minimum set of accelerometers for estimating the angular acceleration is three, we might have many more inertial sensors available on the estimated link. In the previous formulation of the estimation system, we realize that the measurements from one of the sensors, namely S_0, are used more often than the others, which impacts the weight of the respective data in the least squares optimization. That difference might become significant typically if we select systematically S_0 for the same purpose, i.e. as a reference to the others, while increasing the number of accelerometers used in the estimation process. In order to cope with this inconvenience, we rewrite (4.9), replacing the link coordinate acceleration and the gravity by the link *proper acceleration* as it would be measured by a virtual accelerometer which sensor frame is the frame L: $\alpha_L = {}^L R_A \left({}^A\ddot{p}_L - {}^A g\right)$. We rewrite (4.9) as follows:

$$\begin{bmatrix} ({}^L r_{L,S_1})^\wedge & -\mathbb{I}_3 \\ ({}^L r_{L,S_2})^\wedge & -\mathbb{I}_3 \end{bmatrix} \begin{bmatrix} {}^L\dot{\omega}_L \\ \alpha_L \end{bmatrix} = \begin{bmatrix} -{}^L\alpha_{S_1} + ({}^L\omega_L^\wedge)^2 {}^L r_{L,S_1} \\ -{}^L\alpha_{S_2} + ({}^L\omega_L^\wedge)^2 {}^L r_{L,S_2} \end{bmatrix} \tag{4.16}$$

Solving this new system for $[{}^L\dot{\omega}_L \; \alpha_L]^\top$ requires the matrix multiplying it to have rank six (full rank), while as is, the matrix has at most rank four. For that reason we need to add at least an additional accelerometer to the system, as with the base algorithm in the previous section. (4.16) can then be re-written as follows:

$$\begin{bmatrix} ({}^L r_{L,S_1})^\wedge & -\mathbb{I}_3 \\ ({}^L r_{L,S_2})^\wedge & -\mathbb{I}_3 \\ ({}^L r_{L,S_3})^\wedge & -\mathbb{I}_3 \end{bmatrix} \begin{bmatrix} {}^L\dot{\omega}_L \\ \alpha_L \end{bmatrix} = \begin{bmatrix} -{}^L\alpha_{S_1} + ({}^L\omega_L^\wedge)^2 {}^L r_{L,S_1} \\ -{}^L\alpha_{S_2} + ({}^L\omega_L^\wedge)^2 {}^L r_{L,S_2} \\ -{}^L\alpha_{S_3} + ({}^L\omega_L^\wedge)^2 {}^L r_{L,S_3} \end{bmatrix} \tag{4.17}$$

$$\iff \begin{bmatrix} {}^L\hat{\dot{\omega}}_L \\ \hat{\alpha}_L \end{bmatrix} = C^+ \begin{bmatrix} -{}^L\alpha_{S_1} \\ -{}^L\alpha_{S_2} \\ -{}^L\alpha_{S_3} \end{bmatrix} + C^+ \begin{bmatrix} ({}^L\omega_L^\wedge)^2 {}^L r_{L,S_1} \\ ({}^L\omega_L^\wedge)^2 {}^L r_{L,S_2} \\ ({}^L\omega_L^\wedge)^2 {}^L r_{L,S_3} \end{bmatrix} \tag{4.18}$$

$$\text{Where } C = \begin{bmatrix} ({}^L r_{L,S_1})^\wedge & -\mathbb{I}_3 \\ ({}^L r_{L,S_2})^\wedge & -\mathbb{I}_3 \\ ({}^L r_{L,S_3})^\wedge & -\mathbb{I}_3 \end{bmatrix} \text{ and } C^+ = (C^\top C)^{-1} C^\top. \tag{4.19}$$

Where ${}^L\hat{\dot{\omega}}_L$ and $\hat{\alpha}_L$ are respectively the estimated angular and linear link accelerattions. Note that the gravity doesn't appear in the system above as also happens with (4.13), such that gravity compensation is not necessary. Note that ${}^L\dot{\omega}_L$ is the time derivative of ${}^A\omega_L$ projected into the frame L. This quantity can differ from the result of the numerical derivative of the measurement from a gyroscope mounted on the link. Actually, this is not an issue since the time derivative of ${}^A\omega_L$ projected into the frame L is the quantity required by the internal joints torque estimation algorithm described in [Traversaro, 2017, Chapter 4 section 4.4.2]. All other quantities are measured, like the accelerometers measurements α_{S_i}, the gyroscopes measurements ω_L, or known, like the sensor relative poses extracted from the robot CAD model or calibrated offline.

minimal set of sensors

As we need a rank six matrix for solving the system, we require at least one three-axis gyroscope and three distinct three-axis accelerometers.

Remark: As six independent equations in the system are required, we only need two axis per accelerometer, so we could have defined a system with six single axis accelerometers.

4.4.3 Optimizing the acceleration computation

In order to regroup the measured dynamic variables apart from the constants, we wish to rewrite (4.17) as follows:

$$\begin{bmatrix} {}^{L}\hat{\omega}_L \\ \hat{\alpha}_L \end{bmatrix} = C^{+}R \begin{bmatrix} \alpha_{S_1} \\ \alpha_{S_2} \\ \alpha_{S_3} \end{bmatrix} + C^{+}D \ \text{vec}\left({}^{L}\omega_L \otimes {}^{L}\omega_L^{\top}\right) + C^{+}P\left({}^{L}\omega_L^{\top}{}^{L}\omega_L\right) \tag{4.20}$$

Where $[\alpha_{S_1}^{\top}\ \alpha_{S_2}^{\top}\ \alpha_{S_3}^{\top}]^{\top}$ and the terms in ${}^{L}\omega_L$ are the measured quantities, and R, D and P are the constant values that depend on the sensor poses and can be computed offline. This should reduce significantly the computation cost at runtime.

Computing R

$$\begin{aligned} \begin{bmatrix} -{}^{L}\alpha_{S_1} \\ -{}^{L}\alpha_{S_2} \\ -{}^{L}\alpha_{S_3} \end{bmatrix} &= \begin{bmatrix} -{}^{L}R_{S_1} & 0 & 0 \\ 0 & -{}^{L}R_{S_2} & 0 \\ 0 & 0 & -{}^{L}R_{S_3} \end{bmatrix} \begin{bmatrix} \alpha_{S_1} \\ \alpha_{S_2} \\ \alpha_{S_3} \end{bmatrix} \\ &= -\,\text{diag}\left({}^{L}R_{S_1}, {}^{L}R_{S_2}, {}^{L}R_{S_3}\right) \begin{bmatrix} \alpha_{S_1} \\ \alpha_{S_2} \\ \alpha_{S_3} \end{bmatrix} . \end{aligned} \tag{4.21}$$

We defined the *diag(-)* operator in the section B.2.

Computing D and P

Computing the squared skew matrix $({}^{L}\omega_L^{\wedge})^2$ once and using it in a single matrix multiplication is more efficient than executing a cross product twice for each sensor position ${}^{L}r_{L,S_k}$. We convert the squared skew matrix $({}^{L}\omega_L^{\wedge})^2$ into a combination of simple matrix inner product:

$$
\begin{aligned}
(\omega^{\wedge})^2\, r &= \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} r \\
&= \begin{bmatrix} -z^2 - y^2 & yx & zx \\ xy & -z^2 - x^2 & zy \\ xz & yz & -y^2 - x^2 \end{bmatrix} r \\
&= \left(\omega \otimes \omega^{\top} - \omega^{\top}\omega\, \mathbb{I}_3\right) r \\
&= \omega \otimes \omega^{\top}\, r - \omega^{\top}\omega\, r
\end{aligned} \tag{4.22}
$$

We can re-write the first term of (4.22) as:

$$
\begin{aligned}
\omega \otimes \omega^{\top}\, r &= \left(\left(\omega \otimes \omega^{\top}\, r\right)^{\top}\right)^{\top} \\
&= \left(r^{\top}\left(\omega \otimes \omega^{\top}\right)\right)^{\top} \\
&= \operatorname{vec}\left(r^{\top}\left(\omega \otimes \omega^{\top}\right)\right) \\
&= \left(I_3 \otimes r^{\top}\right) \operatorname{vec}\left(\omega \otimes \omega^{\top}\right)
\end{aligned}
$$

The last line is obtained using the vectorization property, demonstrated in section B.2, that states: $\operatorname{vec}(AB) = (I_m \otimes A)\operatorname{vec}(B)$, for any matrices A, B and where m is the number of columns of B.

Since $\omega^{\top}\omega$ is a scalar, we can re-write the second term of (4.22) as $r\ \omega^{\top}\omega$. We then get the following decomposition:

$$
\begin{aligned}
\begin{bmatrix} ({}^{L}\omega_L^{\wedge})^{2\,L} r_{L,S_1} \\ ({}^{L}\omega_L^{\wedge})^{2\,L} r_{L,S_2} \\ ({}^{L}\omega_L^{\wedge})^{2\,L} r_{L,S_3} \end{bmatrix} &= \begin{bmatrix} \mathbb{I}_3 \otimes {}^{L}r_{L,S_1}^{\top} \\ \mathbb{I}_3 \otimes {}^{L}r_{L,S_2}^{\top} \\ \mathbb{I}_3 \otimes {}^{L}r_{L,S_3}^{\top} \end{bmatrix} \operatorname{vec}\left({}^{L}\omega_L \otimes {}^{L}\omega_L^{\top}\right) \\
&\quad - \begin{bmatrix} {}^{L}r_{L,S_1} \\ {}^{L}r_{L,S_2} \\ {}^{L}r_{L,S_3} \end{bmatrix} \left({}^{L}\omega_L^{\top}\,{}^{L}\omega_L\right)
\end{aligned} \tag{4.23}
$$

In view of (4.21) and (4.23) we can rewrite (4.18) as (4.20), and generalize to N sensors as follows:

$$
\begin{bmatrix} {}^{L}\hat{\omega}_L \\ \hat{\alpha}_L \end{bmatrix} = J\left({}^{L}r_{L,S_i}\right)\ y\left(\alpha_{S_i}, {}^{L}\omega_L\right), \tag{4.24}
$$

with:

$$\begin{aligned} J\left({}^{L}r_{L,S_i}\right) &= \begin{bmatrix} C^{+}R & C^{+}D & C^{+}P \end{bmatrix}, \\ y\left(\alpha_{S_i}, {}^{L}\omega_L\right) &= \begin{bmatrix} \alpha_{S_1} \\ \vdots \\ \alpha_{S_N} \\ \operatorname{vec}\left({}^{L}\omega_L \otimes {}^{L}\omega_L^{\top}\right) \\ \left({}^{L}\omega_L^{\top}\,{}^{L}\omega_L\right) \end{bmatrix}, \end{aligned} \tag{4.25}$$

and:

$$C^{+} = (C^{\top}C)^{-1}C^{\top}, \quad C = \begin{bmatrix} ({}^{L}r_{L,S_1})^{\wedge} & -\mathbb{I}_3 \\ \vdots & \\ ({}^{L}r_{L,S_N})^{\wedge} & -\mathbb{I}_3 \end{bmatrix}, \tag{4.26}$$

$$\begin{aligned} R &= -\operatorname{diag}\left({}^{L}R_{S_1}, \cdots, {}^{L}R_{S_N}\right), \\ D &= \begin{bmatrix} \mathbb{I}_3 \otimes {}^{L}r_{L,S_1}^{\top} \\ \vdots \\ \mathbb{I}_3 \otimes {}^{L}r_{L,S_N}^{\top} \end{bmatrix}, \; P = -\begin{bmatrix} {}^{L}r_{L,S_1} \\ \vdots \\ {}^{L}r_{L,S_N} \end{bmatrix}, \end{aligned} \tag{4.27}$$

where $J\left({}^{L}r_{L,S_i}\right)$ is what we will call the “configuration matrix” of the accelerometers set, and $y\left(\alpha_{S_i}, {}^{L}\omega_L\right)$ is the vector of all accelerometers and gyroscopes measurements.

4.4.4 Sensors Anisotropic Sensitivity and bias compensation

The MEMS inertial sensors are characterized by a slow time varying bias on each axis, different axis gains, cross axis sensitivity. We addressed in chapter 3 the analysis and compensation of those defects in the case of the accelerometers, and here we assume them to have been calibrated offline.

In this thesis we consider the gyroscopes to be affected only by an offset, having an isotropic sensitivity along the three axes. The gyroscopes offsets can be compensated online but that is out of the scope of this Thesis, and will be addressed in a future work. Here we consider the offsets calibrated offline.

4.4.5 Impact of the Sensor Noise on the Estimated Quantities

The MEMS inertial sensors are affected by an additive high frequency noise. Unlike the accelerometers used in our experiments, the gyroscopes have some filtering integrated in the sensor chipset for canceling the measurement noise. For this reason, the accelerometers noise will be the only perturbation considered throughout this chapter. Their measurements can then be expressed as follows:

$$\tilde{\alpha}_{S_i} = \alpha_{S_i} + \eta_{S_i} \ , \tag{4.28}$$

where α_{S_i} is the sensor measurement without any perturbation, and η_{S_i} the high frequency noise affecting the sensor. If we integrated that perturbation in the link acceleration estimate (4.24) we get:

$$\begin{aligned}
\begin{bmatrix} {}^L\tilde{\dot{\omega}}_L \\ \tilde{\alpha}_L \end{bmatrix} &= C^+R \begin{bmatrix} \tilde{\alpha}_{S_1} \\ \vdots \\ \tilde{\alpha}_{S_N} \end{bmatrix} + C^+D \ \operatorname{vec}\left({}^L\omega_L \otimes {}^L\omega_L^\top\right) + C^+P\left({}^L\omega_L^\top {}^L\omega_L\right) \\
&= C^+R \begin{bmatrix} \alpha_{S_1} \\ \vdots \\ \alpha_{S_N} \end{bmatrix} + C^+D \ \operatorname{vec}\left({}^L\omega_L \otimes {}^L\omega_L^\top\right) + C^+P\left({}^L\omega_L^\top {}^L\omega_L\right) \\
&+ C^+R \begin{bmatrix} \eta_{S_1} \\ \vdots \\ \eta_{S_N} \end{bmatrix} \\
\begin{bmatrix} {}^L\tilde{\dot{\omega}}_L \\ \tilde{\alpha}_L \end{bmatrix} &= \begin{bmatrix} {}^L\hat{\dot{\omega}}_L \\ \hat{\alpha}_L \end{bmatrix} + C^+R \begin{bmatrix} \eta_{S_1} \\ \vdots \\ \eta_{S_N} \end{bmatrix} = \begin{bmatrix} {}^L\hat{\dot{\omega}}_L \\ \hat{\alpha}_L \end{bmatrix} + {}^L\eta_L, \qquad (4.29)
\end{aligned}$$

where $\tilde{x}$ and $\hat{x}$ are the quantities estimated respectively with and without noisy measurements, and ${}^L\eta_L$ is the cumulated contributions of the sensors noise to the estimation.

Remark 4.2. *If we look closer to the impact of the sensors positions on the estimation sensitivity to the measurements noise, we write:*

$$\left\|{}^L\eta_L\right\| \leq \left\|C^+R\right\| \left\| \begin{bmatrix} \eta_{S_1} \\ \vdots \\ \eta_{S_N} \end{bmatrix} \right\|, \tag{4.30}$$

and,

$$\lim_{\forall i, {}^L r_{L,S_i} \to +\infty} \left\|C^+R\right\| = 0,$$

so,

$$\lim_{\forall i, {}^{L}r_{L,S_i} \to +\infty} \left\| {}^{L}\eta_L \right\| = 0.$$

This result shows that, in the simple case of a pure rotation about a given axis, and that axis runs through the link frame origin, the contribution of the inertial sensors noise to the estimated angular acceleration noise decreases with the distance between the sensors and the rotation axis.

Note. *A more accurate analysis of the propagation of uncertainties due to noise measurements would require a proper analysis of the properties of the given noise distribution, in order to infer which probability tools to use. For instance, if the measurements noise could be modeled by a Gaussian distribution, we could address the problem with the Bayesian estimation on linear models. The analysis of the noise distribution and the derived uncertainties propagation analysis are out of scope of this thesis dissertation.*

4.5 Experiments

4.5.1 Sensors framework

For validating the algorithm presented in this chapter and evaluating its performance, we have run the experiments on the robot iCub, which architecture we already presented in the section 2 and 3, along with the sensors framework.

4.5.2 Simulations

Prior to running the experiments on iCub, a series of tests were performed on Matlab for assessing the impact of the sensors positions and sensors measurements noise on the estimation algorithm performance.

The Cube Link Model: The first model we have tested was a cube, 1m large, with eight triaxial accelerometers placed at its vertices, roughly 80cm from the centre, and two gyroscopes at its centre. The cube was attached to a fixed base through a rotary actuated joint with vertical axis. Several joint motion trajectories were tested: fixed position; constant velocity; constant acceleration; sinusoidal acceleration; random smooth trajectory.

Note. *The random trajectory was obtained through a sum of sinusoidal trajectories of random amplitudes and frequencies. This allowed to define the velocity and acceleration analytically.*

The joint positions, velocities and accelerations, were defined analytically and sampled at 100Hz. The sensor measurements were then emulated through forward kinematics from the input joint trajectory variables, by using a dynamics library [7] for the computations. For simulating the measurement noise, we added to the generated samples an uncorrelated white noise, i.e. a random signal with white frequency spectrum and Gaussian amplitude distribution, also referred to as Normal distribution. The noise component, denoted $\eta = \mathcal{N}\left(0, \sigma^2\right)$, has zero mean and a standard deviation $sigma \sim 0.29$.

The link angular acceleration is estimated and expressed at the centre of the cube. A noise level of 1% of the measurement full scale applied to the accelerometers will not affect the estimated angular acceleration, as the distance from the centre of the cube to the accelerometers frames is significant with respect to the noise level. It will not be the case for a model with a distribution of sensors closer to the link centre.

iCub upper leg Link Model: The following simulation tests were run on a model of iCub left thigh. The model is defined by a URDF (Unified Robot Description Format [8]) and related meshes. In Chapter 3 we addressed more in detail the protocol used for describing a kinematic tree and how a URDF is generated from a CAD model.

An iCub thigh (also referred to as "left/right upper leg") is equipped with two gyroscopes (*gyro_eb6*, *gyro_eb7*) integrated in interface boards (EMS6, EMS7) mounted on the link, and seven triaxial accelerometers mounted on the interior side of the thigh covers, more specifically five on the front cover (*acc_10b1* to *acc_10b5*) and two on the rear cover (*acc_10b6* and *acc_10b7*). The Matlab test script imports the URDF model [9] and retrieves from it the sensors frames positions and orientations. Figure 4.3 illustrates the spatial distribution of the sensors.

We describe below the test procedure. We wish to have the exact same trajectory run on the simulation and on the real robot:

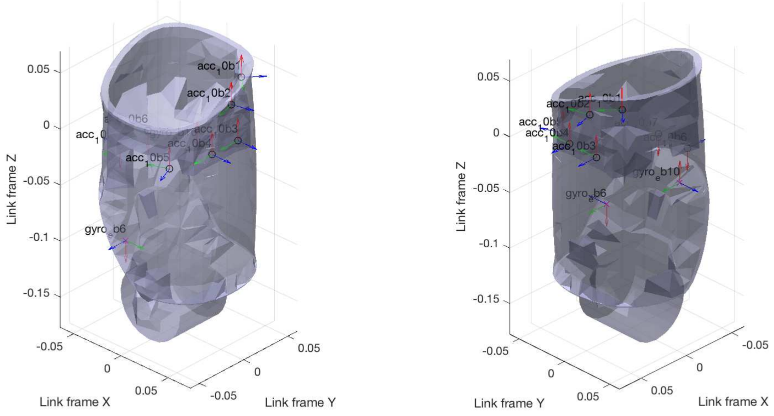
1. we initially run a test on the real robot, which is fixed on a pole. The initial left leg joints position is such that the leg is straight, making an angle of 30deg w.r.t. the gravitation field. We then rotate the hip

---

[7] *iDynTree*: `https://github.com/robotology/idyntree`.

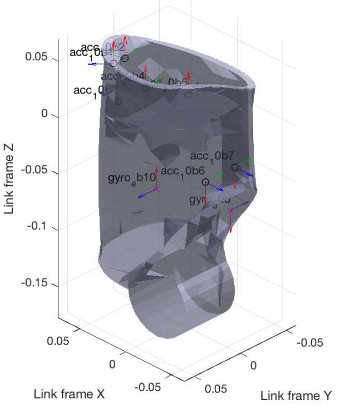
[8] `wiki.ros.org/urdf`

[9] `https://github.com/robotology/icub-models/blob/master/iCub/robots/iCubGenova04/model.urdf`



(a) Front right view

(b) Front left view



(c) Rear view

Fig. 4.3 URDF model with mesh of iCub left upper leg (thigh). Front (a,b) and rear (c) views. The accelerometers MTB 1 to 5 are mounted on the front cover, MTB 6 and 7 on the rear cover. They can be seen through the covers made transparent for clarity.

yaw DoF from limit to limit in fast position control motion in order to get a significant acceleration about the yaw axis,

2. the leg joint encoders and inertial sensors data are acquired and fed to the Matlab script,
3. we then filter the joint encoder positions q using a Finite Impulse Response (FIR) filter defined with coefficients generated from the Savitsky-Golay algorithm [Press, 1992, chapter 14], and used the same polynomial approximation for computing the derivatives $\dot{q}$ and $\ddot{q}$,
4. we compute the expected sensor measurements from the filtered $q, \dot{q}, \ddot{q}$ using a Forward Kinematics algorithm from the *iDynTree* dynamics library, and add to those the same uncorrelated white noise used on the previous cube model,
5. we compute the same way as in the previous step the expected link linear and angular accelerations, by adding a virtual angular accelerometer and a virtual linear accelerometer to the leg link model depicted in figure 4.3. Note that this step uses Forward Kinematics as well to compose and propagate the link velocities and accelerations.
6. we estimate the link linear and angular accelerations using the algorithm proposed in 4.4.3, and compare them against the respective expected values.

First we started with noise free measurements and verified that the estimation is perfectly tracking the expected acceleration computed by the dynamics library, as we can observe in figure 4.4.

Note. *The oscillations observed in the first and last seconds of the time frame are due to the* Savitsky-Golay *filter. Actually, we have implemented a simpler version of the filter, without the "buffer-in" , "buffer-out" sub filters allowing to smooth in and out the signal when the main filter buffer window is partially filled. This has no impact on our analysis, as we will just ignore the edges of the plots time-frames, i.e. the time-frame will be considered valid from* 3s *to* 20s*.*

We can actually already compare in 4.4 the result of our angular acceleration estimation algorithm with a classic numerical derivative defined by

(4.31), which has already some noise (numerical computations noise) even with noise free measurements.

$$\dot{\omega}^*(t_k) = \frac{\omega(t_k) - \omega(t_{k-1})}{t_k - t_{k-1}} \tag{4.31}$$

Remark 4.3. *If for a given application a delay of one sampling period* $T_s = t_k - t_{k-1}$ *is acceptable before getting the derivative result, we use an alternate definition of the numerical derivative as follows, more reliable than* (4.31)*:*

$$\dot{\omega}^*(t_{k-1}) = \frac{\omega(t_k) - \omega(t_{k-2})}{t_k - t_{k-2}} \tag{4.32}$$

Adding Estimation Filters

We now introduce noise in the measurements: we size the sensors noise through a signal-to-noise (SN) ratio which is then applied to the each accelerometer or gyroscope sensor full-scale value.

We set a SN of 1% and will use that level of noise throughout all the experiments in this section. We have tried four different types of estimators and compared their performance achieved with noisy measurements:

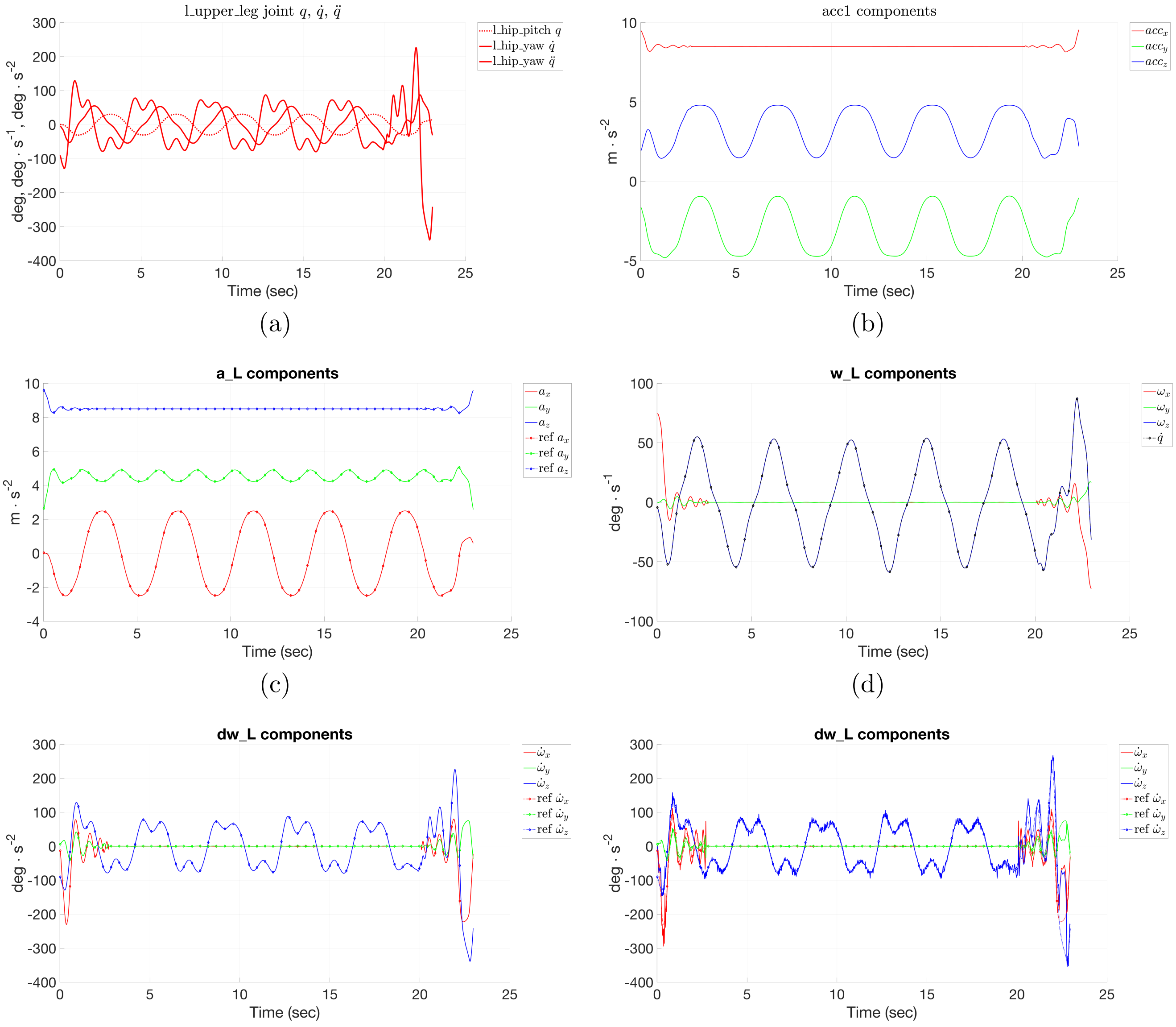
estimator 1: the numerical derivative method applied on ${}^L\omega_L$ (4.31),

estimator 2: our method proposed in this chapter (4.24),

estimator 3: *estimator 2* extended with a regularization term minimizing the distance between consecutive estimations of ${}^L\hat{\dot{\omega}}_L$,

estimator 4: *estimator 3* regularized by the numerical derivative (*estimator 1*).

We can see in figure 4.5 how estimators 1 and 2 perform almost equally bad with 1% SNR. But if we look closer, we notice that the noise spikes generated by the *estimator 1* can be twice as high as those from *estimator 2.* We plotted the probability distribution of the errors with respect to the reference expected link accelerations. Both estimators results exhibit a Gaussian like error distribution with zero mean and a high variance. The measurements noise has significantly been amplified in the angular acceleration estimate ${}^L\hat{\dot{\omega}}_L$, while it barely affects the estimated link linear acceleration $\hat{\alpha}_L$. Actually, the condition number of the matrix C defined in (4.26) is quite high, which is probably due to the proximity of the accelerometers



(a) (b)

(c) (d)

(e) (f)

Fig. 4.4 Simulation results when rotating the hip yaw between two limit joint positions (fast motion in minimum jerk position control): (a) filtered q, $\dot{q}$ and $\ddot{q}$; (b) simulated accelerometer *acc_10b1* measurements; (c) estimated against expected link linear acceleration; (d) mean of simulated gyroscopes *gyro_eb6* and *gyro_eb7*; (e) estimated against expected link angular acceleration; (f) numerical derivative of the angular velocity.

with the rotation axis and the origin of the frame where we estimate the link acceleration.

For compensating that perturbation we wish to regularize the least squares estimation of the 6-D acceleration $\left[{}^{L}\hat{\dot{\omega}}_{L}^{\top}, \hat{\alpha}_{L}^{\top}\right]_{t_k}^{\top}$, from the *estimator 2*, with respect to the 6-D acceleration estimated at the previous sampling time t_{k-1}. We follow an approach similar to the TRLS (Tikhonov-Regularized Least Squares) problem [Golub et al., 1999].

Regularization with respect to previous estimate based on TRLS (*estimator 3*): We wish to solve an over-constrained system,

$$Ax = b,$$

through a Least Square optimization problem. So we pose the cost function to minimize:

$$\Gamma = \|Ax - b\|_2^2 + \lambda \|x - x_{k-1}\|_2^2 ,$$

where the second l^2-norm term is the regularization term forcing the LS (Least Squares) minimization problem to converge to an optimum x close, in euclidean distance, to x_{k-1} which is a known input to the problem. As the l^2-norm is differentiable, we can solve the problem with the gradient descent:

$$x^* = \arg\min_x \Gamma(x) ,$$

the the optimal vector x^* is the one for which the gradient of Γ with respect to x is 0. so we can write:

$$\begin{aligned}
\nabla\Gamma(x) &= \frac{1}{n}(Ax - b)^\top (Ax - b) + \lambda \|x - x_{k-1}\|_2^2 \\
&= \frac{2}{n}A^\top (Ax - b) + 2\lambda (x - x_{k-1}) \\
&= A^\top (Ax - b) + \lambda n (x - x_{k-1}) \\
&= \left(A^\top A + \lambda n \mathbb{I}_n\right) x - \left(A^\top b + \lambda n x_{k-1}\right) ,
\end{aligned}$$

so,

$$\begin{aligned}
&\nabla\Gamma(x^*) = 0 \\
\Longleftrightarrow\, &x^* = \left(A^\top A + \lambda n \mathbb{I}_m\right)^{-1} \left(A^\top b + \lambda n x_{k-1}\right) ,
\end{aligned} \tag{4.33}$$

where m is the dimension of x. If we transpose this new LS formulation to our inital problem, and in view of (4.24), (4.25), (4.26) and (4.27), we get:

$$A := C, \quad x := \begin{bmatrix} {}^L\hat{\dot{\omega}}_L \\ \hat{\alpha}_L \end{bmatrix}, \quad b := \begin{bmatrix} R & D & P \end{bmatrix} y\left(\alpha_{S_i}, {}^L\omega_L\right) \ ,$$

and we redefine the initial algorithm as follows:

- C, R, D and P are still defined as in (4.26) and (4.27),
- we redefine $C^+ = \left(C^\top C + \lambda n \mathbb{I}_6\right)^{-1} C^\top$,
- we define $C_R^+ = \lambda n \left(C^\top C + \lambda n \mathbb{I}_6\right)^{-1}$,
- and:

$$\begin{bmatrix} {}^L\hat{\dot{\omega}}_L \\ \hat{\alpha}_L \end{bmatrix}^* = C^+ \begin{bmatrix} R & D & P \end{bmatrix} y\left(\alpha_{S_i}, {}^L\omega_L\right) + C_R^+ \begin{bmatrix} {}^L\hat{\dot{\omega}}_L \\ \hat{\alpha}_L \end{bmatrix}_{k-1} . \tag{4.34}$$

Regularization with respect to the numerical derivative (*estimator 4*): We now start from the *estimator 3* and add a regularization term defined from the numerical derivative of the measured angular velocity (simulated measurement). We recall the *estimator 3* system definition:

$$C \begin{bmatrix} {}^L\hat{\dot{\omega}}_L \\ \hat{\alpha}_L \end{bmatrix} = \begin{bmatrix} R & D & P \end{bmatrix} y\left(\alpha_{S_i}, {}^L\omega_L\right) + C\, C_R^+ \begin{bmatrix} {}^L\hat{\dot{\omega}}_L \\ \hat{\alpha}_L \end{bmatrix}_{k-1} .$$

We extend the system by adding the line about the constraint on ${}^L\dot{\omega}_L$, i.e. $dt\, \mathbb{I}_3 {}^L\dot{\omega}_L = {}^L\omega_{L,k} - {}^L\omega_{L,k-1}$:

- C is redefined to,

$$C = \begin{bmatrix} ({}^L r_{L,S_1})^\wedge & -\mathbb{I}_3 \\ \vdots & \\ ({}^L r_{L,S_N})^\wedge & -\mathbb{I}_3 \\ dt\, \mathbb{I}_3 & 0_{3\times 3} \end{bmatrix} ,$$

- R, D, P, C^+ and C_R^+ stay unchanged with respect to C,
- and:

$$\begin{aligned} \begin{bmatrix} {}^L\hat{\dot{\omega}}_L \\ \hat{\alpha}_L \end{bmatrix} =& C^+ \begin{bmatrix} R & D & P \\ 0_{3\times N} & 0_{3\times 9} & 0_{3\times 1} \end{bmatrix} y\left(\alpha_{S_i}, {}^L\omega_L\right) + C_R^+ \begin{bmatrix} {}^L\hat{\dot{\omega}}_L \\ \hat{\alpha}_L \end{bmatrix}_{k-1} \\ &+ C^+ \begin{bmatrix} 0_{3N\times 1} \\ {}^L\omega_{L,k} - {}^L\omega_{L,k-1} \end{bmatrix} . \end{aligned} \tag{4.35}$$

Estimation Filters Results

We can see in figure 4.6 that we improved significantly the accuracy with the *estimators 3 and 4*, reducing the estimation error standard deviation respectively by 2 and 3. The estimator 4, combining both regularizations, is the best performer, constraining the estimation variation between consecutive sampling times, and regularizing with respect to the numerical derivative of ${}^{L}\omega_L$. Each of them is less effective than this combination.

4.5.3 Tests on the Real Robot

Filters on the sensor measurements

As the link acceleration estimator is expected to work online, we did not consider non-causal filters, usually used offline on the sensors measurements, like the Savitsky-Golay filter, and typically applied with a filtering window of hundreds of samples for instance in the case of accelerometers or joint offsets calibration. On top of that, we preferred to filter or regularize the estimated quantity instead of a large set of sensors, which would probably reduce significantly the computation cost.

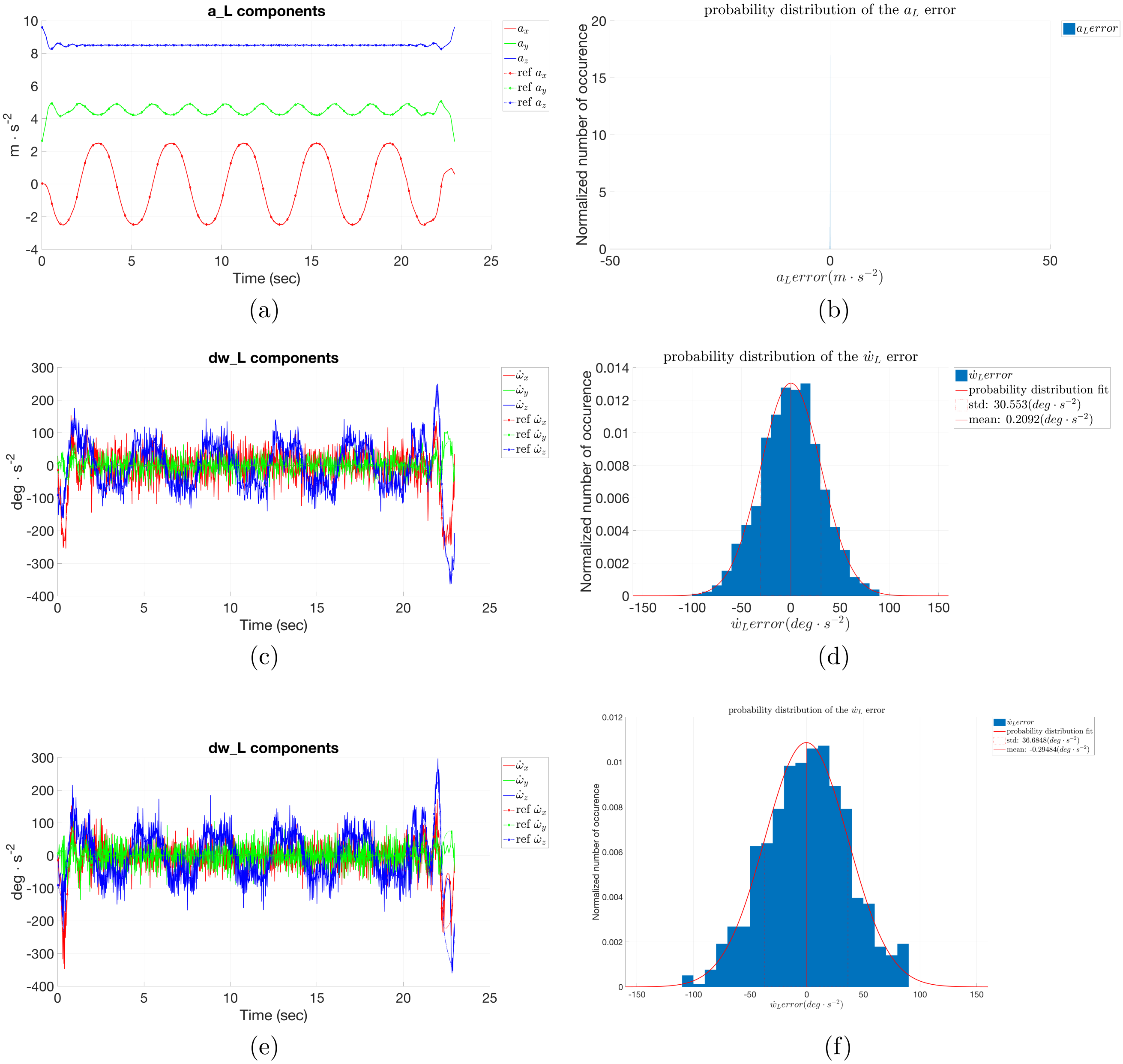
Note. *The gyroscopes already have a low pass filter at the hardware level, such that no noticeable noise is observed in the measurements. The sensor has integrated low-pass and high-pass filters with user-selectable bandwidth spanning from* 16Hz *to* 110Hz *(refer to table 21 of the datasheet). We currently do not have access to those parameters through the robot interface*

Estimation Filters Results

The results on the robot with any of the estimators 1 to 4 are affected by significant noise. Figure 4.7 depicts the results obtained on the same link used in the simulations. Figure (a) and (b) show the joint positions and accelerations, comparing the raw against the filtered ones. We would expect the angular acceleration estimation in the figure (d) to match or track the filtered joint acceleration (hip yaw) in figure (b), black thick line.

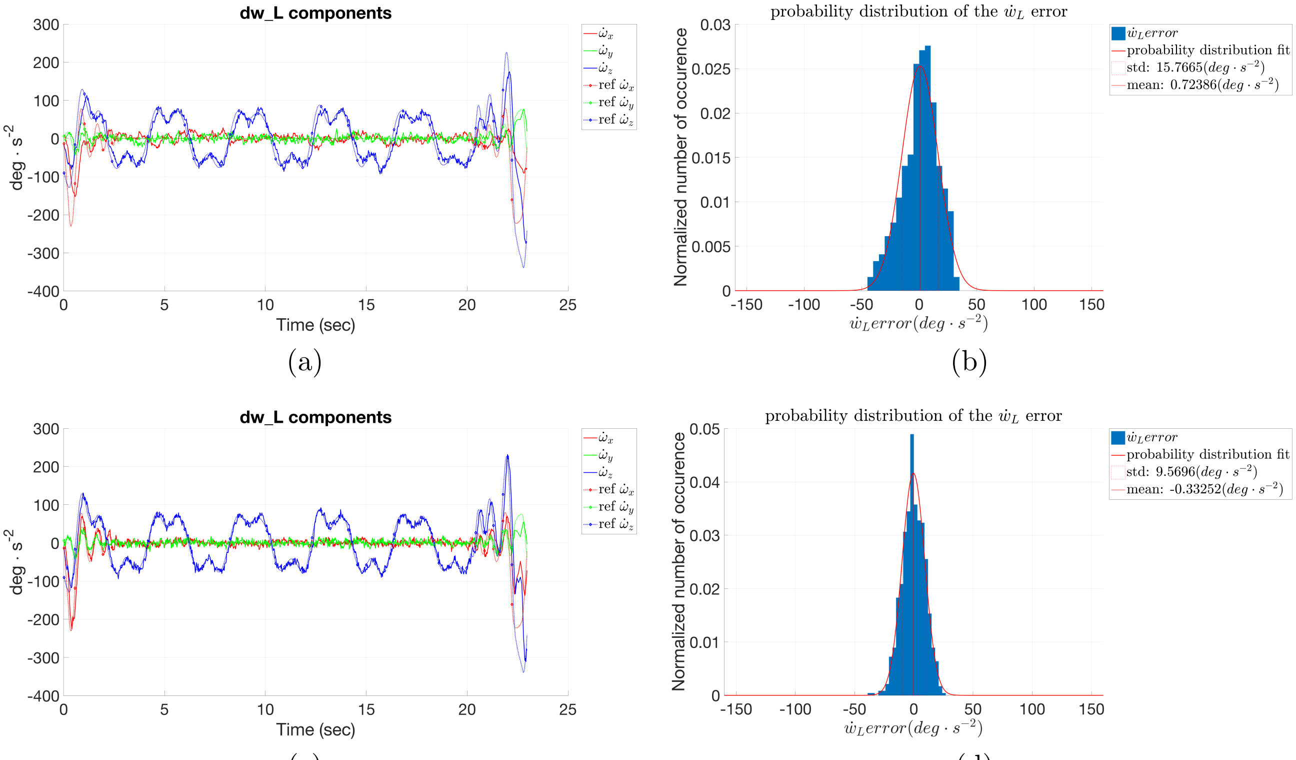
Results on sensor data filtered offline

In view of the link acceleration estimation results obtained on the real robot, we checked if the measurement noise affecting the accelerometers was the main perturbation deteriorating the estimation accuracy. For that purpose we filtered the sensors data offline using a non-causal Finite Response Filter



(a) (b) (c) (d) (e) (f)

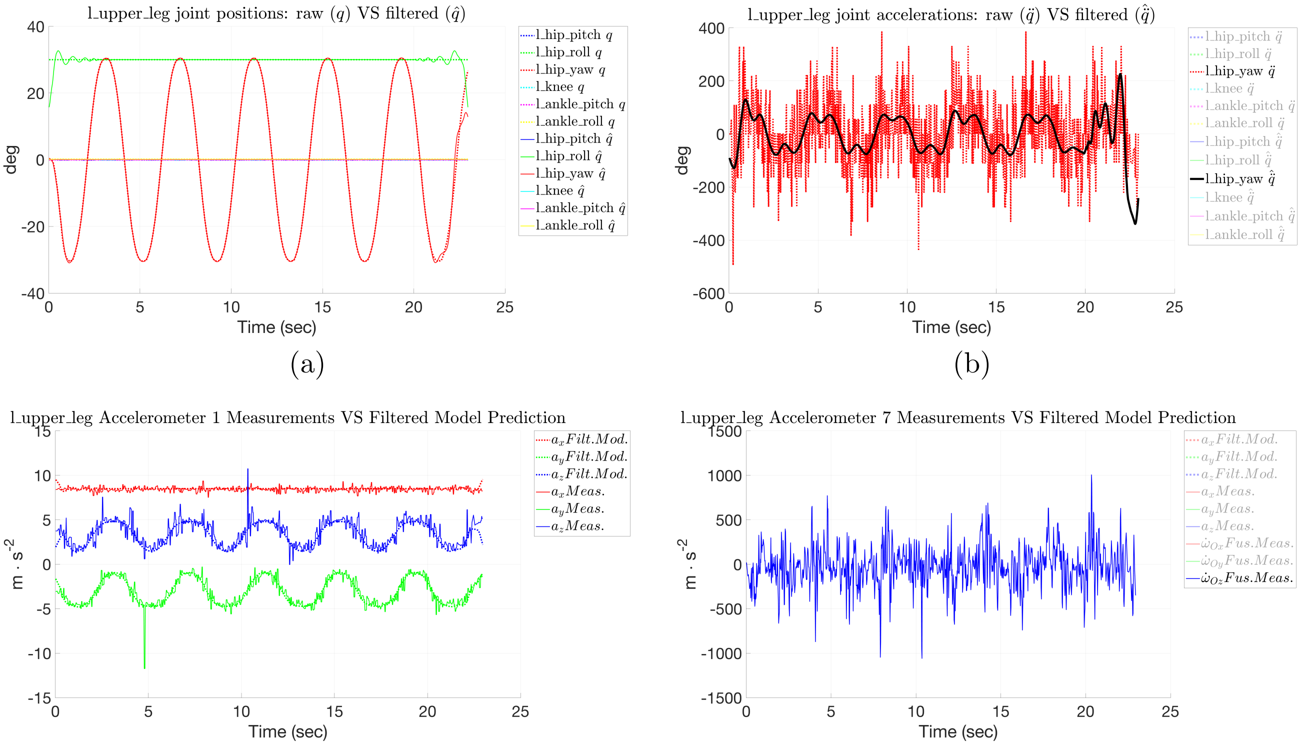
Fig. 4.5 Simulation results from measurements affected by noise with SNR = 1%. Time series signal on the left, respective probability distribution on the right: (a)-(b)-(c)-(d) Link acceleration computed by *estimator 2* (our proposed algorithm); (e)-(f) Link acceleration computed by *estimator 1* (numerical derivative). The estimated link linear acceleration a_L is not affected by noise.



(a) (b)

(c) (d)

Fig. 4.6 Simulation results from measurements affected by noise with SNR = 1%. Time series signal on the left, respective probability distribution on the right: (a)-(b) Link angular acceleration computed by *estimator 3* (minimizes $\dot{\omega} - \dot{\omega}_{k-1}$, $\lambda = 0.01$); (c)-(d) *estimator 3* regularized with the numerical derivative.



(a)

(b)

(c)

(d)

Fig. 4.7 Test results on the real robot. (c) we can see a high level frequency noise, of great amplitude, with punctual spikes in the accelerometers measurements. (d) The angular acceleration estimation is not tracking properly the expected link acceleration (upper leg rotating with the hip yaw oscillations), while the double numerical derivative of the encoder (b) gives better results.

(FIR) defined from the Savitsky-Golay algorithm, and verified that it was the case.

4.6 Conclusion

This chapter proposes a algorithm for estimating algebraically the angular and linear acceleration of a floating body from the measurements of a set of MEMS inertial sensors attached to that body, namely gyroscopes and accelerometers. It is still hard to find small sized sensors, for robotic applications, measuring directly the angular acceleration, with a good bias stability, high sensitivity and linear acceleration rejection. The classical alternative is to use the double numerical derivative of the joint encoders measurements combined with forward kinematics for estimating link accelerations and propagating the, but numerical derivation is subject to high level noise which is then propagated through the Forward Kinematics. Noise can be filtered but this introduces lag, and does not avoid propagation of errors. The algebraic direct computation of the angular acceleration from MEMS sensors data avoids these issues. It should be less sensitive to noise, introduce less lag since low pass filtering is not required, and propagate less errors since it uses only local sensor data. It is derived from higher resolution measurements.

Our approach is based on a model relating the angular velocity and acceleration with the linear accelerations across a distributed set of body points. It can be applied to any link of a fixed or floating base system as long as it has at least three attached accelerometers and one gyroscope. Previous works have used a similar model of the link angular acceleration but were either dependent on the composition of accelerations across the supporting tree or chain, or depended on measurements from neighbor links in the kinematic chain, thus depending on the configuration of the connecting joints which are subject to uncertainties. Our approach rely only on local data from the attached link sensors, and does not depend on any joint configuration measurement.

As done in previous studies, we pose the problem as a linear system to solve for the link angular and proper linear accelerations as it would be measured respectively by a virtual angular accelerometer and linear accelerometer, both sharing a sensor frame coinciding with the root link frame.

The first step is to compute the proper acceleration of a given accelerometer k sensor frame S_k as a function of the link coordinate acceleration, the link angular velocity and acceleration, the sensor frames poses with respect to the link frame, and the link pose with respect to the world frame, all projected in the link frame. It is a known approach to pose the difference between the measurements from two sensors ($\alpha_{S_0} - \alpha_{S_1}$), in order to cancel out the gravitational field and the link coordinate acceleration (and with it the dependency on the link pose with respect to the world frame). The system can then be solved through a linear least squares optimization (pseudo-inverse), as a causal memory-less estimation. We demonstrated that for the system scaled to N sensors to have a unique solution, the sensors configuration matrix, i.e. the vertical concatenation of the skew symmetric matrices derived from the sensor frame positions, needs to be full rank, i.e. rank three. That condition is met if the minimal number of accelerometers is three and they must not be aligned. We have shown that scaling the problem to N sensors from this formulation results in an uneven contribution of the different sensors to the link acceleration estimation.

To cope with this problem the initial formulation was revisited. Instead of canceling the link coordinate acceleration and the gravity, these are replaced by the link *proper acceleration* α_L as it would be measured by a virtual accelerometer which sensor frame coincides with the link frame L, such that we now solve the new system for both angular and linear components of the link acceleration. We get a scalable new system (4.16), for which a new line is added for every additional sensor. All the information on sensors poses in the link acceleration computation (4.24) can be grouped in a single configuration matrix computed offline, hence optimizing the estimation.We analyzed the system sensitivity to noise and showed that, in the simple case of a pure rotation about a given axis, the contribution of the inertial sensors noise to the angular acceleration estimation uncertainty decreases with the distance of the sensors to the axis. This is typically the case for large sized industrial robots, but for a child size humanoid robot like iCub, we get the opposite effect as the links are quite small: the sensors noise is amplified.

We performed simulation tests on Matlab on a simple model for verifying the impact of the sensors distribution on the estimation accuracy. Furthermore we validated the algorithm on an iCub simulation model. Normal (Gaussian) distribution noise was added to the sensors data for analyzing the robustness of the algorithm with respect to measurement noise. While the original algorithm (Section (4.24)) only showed slightly better performance than the numerical derivative of the gyroscopes measurements, an additional regularization based on the TRLS (Tikhonov-Regularized Least Squares), limiting the gap between estimation iterations, significantly reduced the estimation noise.

The estimation results on the real robot exhibit a noise level significantly higher than the one observed in simulation, preventing the angular acceleration to be tracked properly, even when the regularization on the output step is applied. We verified that the measurement noise affecting the accelerometers was the main perturbation deteriorating the estimation accuracy, but we wish to avoid using any low pass filter that would add significant lag to the measurements and the estimation.

As a possible future extension of this work, we consider a few directions for improving the estimation results and make it suitable for being integrated in the joint torque estimation framework. Most probably, a simple exponential or averaging filter will not remove spikes or wavelet like perturbations in the measurements. For that, a wavelet filter, or else a Savitsky-Golay [Press, 1992] filter with a short filtering window including only past samples could be used. Actually, the Savitsky-Golay algorithm generates smoothing polynomials to be applied as FIR (Finite Impulse Response) filters on the sliding window of past, present and future samples (ahead of the processed time instant). When the processing reaches the last time series sample, the actual used polynomial was generated for the case where only past samples exist. That is the typical polynomial we would use for filtering the accelerometers measurements.

In order to better choose the best strategy to follow for improving the estimation accuracy, an accurate analysis of the measurements noise distribution is required, as well as the according noise propagation trough the linear model used by the presented estimation algorithm. In case the measurements noise has a Gaussian component, a Bayesian estimation on linear models could be used as done in [Nori et al., 2015a] [Latella and Nori, Section 4.6.1]: the link acceleration variable to be estimated and the input inertial measurements would be represented by two random Gaussian vectors; the linear model used by the link acceleration estimator would relate both random Gaussian vectors and we would use the *Gauss-Markov* theorem [Latella and Nori, Section 4.6.1] for computing the joint variable distribution and the conditional distribution holding the variance of the noise in the estimation result.

The research presented in this chapter on the accurate and fast estimation of link angular and linear accelerations was motivated by the following objectives: the estimation, in dynamic conditions, of the error between the expected sensor proper acceleration and the respective measurements, for identifying model kinematic parameters, like joint encoder offsets, as an extension of the research presented in Chapter 3; the joint accelerations as an additional input variable to a dynamic friction model used in identification or in control; the accurate estimation of net forces on each body of a kinematic tree, using local inertial sensors data only and thus avoiding propagation errors; the use of these net forces in the estimation of joint torques which in turn are a critical feedback in the joint friction parameters identification presented in Chapter 2.

Bibliography

Bilal Akin, Manish Bhardwaj, and Jon Warriner. Trapezoidal Control of BLDC Motors Using Hall Effect Sensors. page 34, April 2011.

Brian Armstrong-Hélouvry, Pierre Dupont, and Carlos Canudas De Wit. A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica*, 30(7):1083 – 1138, 1994. ISSN 0005-1098. doi: https://doi.org/10.1016/0005-1098(94)90209-7. URL `http://www.sciencedirect.com/science/article/pii/0005109894902097`.

Frank Vanden Berghen. *CONDOR: a constrained, non-linear, derivative-free parallel optimizer for continuous, high computing load, noisy objective functions.* PhD, IRIDIA, Université Libre de Bruxelles, 2004. URL `http://www.applied-mathematics.net/mythesis/index.html`.

Gaetano Canepa and John M Hollerbach. Kinematic Caiibration by Means of a Triaxial Accelerometer. page 7.

C. Canudas-de Wit. Comments on "A new model for control of systems with friction". *IEEE Transactions on Automatic Control*, 43(8):1189–1190, August 1998. ISSN 0018-9286, 1558-2523. doi: 10.1109/9.704999. URL `http://ieeexplore.ieee.org/document/704999/`.

C. Canudas de Wit, H. Olsson, K.J. Astrom, and P. Lischinsky. A new model for control of systems with friction. *IEEE Transactions on Automatic Control*, 40(3):419–425, March 1995. ISSN 00189286. doi: 10.1109/9.376053. URL `http://ieeexplore.ieee.org/document/376053/`.

Martino Capurso, M. Mahdi Ghazaei Ardakani, Rolf Johansson, Anders Robertsson, and Paolo Rocco. Sensorless kinesthetic teaching of robotic manipulators assisted by observer-based force control. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 945–950, Singapore, Singapore, may 2017. IEEE. ISBN 978-1-5090-4633-1. doi: 10.1109/ICRA.2017.7989115. URL `http://ieeexplore.ieee.org/document/7989115/`.

P Chedmail and J-P Martineau. Characterization of the friction parameters of harmonic drive actuators. *WIT Transactions on The Built Environment*, 22, 1970.

Heping Chen, Thomas Fuhlbrigge, Sang Choi, Jianjun Wang, and Xiongzi Li. Practical industrial robot zero offset calibration. In *Automation Science and Engineering, 2008. CASE 2008. IEEE International Conference on*, pages 516–521. IEEE, 2008.

A.R. Conn, N.I.M. Gould, P.L. Toint, Society for Industrial, and Applied Mathematics. *Trust Region Methods*. MPS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, 2000. ISBN 978-0-89871-460-9. URL `https://books.google.it/books?id=5kNC4fqssYQC`.

N. D'Amore, C. Ciarleglio, and D.L. Akin. Imu-based manipulator kinematic identification. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1437–1441, May 2015. doi: 10.1109/ICRA.2015.7139378.

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, March 2016. ISSN 0219-8436, 1793-6942. doi: 10.1142/S0219843615500449. URL `http://www.worldscientific.com/doi/abs/10.1142/S0219843615500449`.

James Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, 58(15-16):1–35, 2006.

D. Dixneuf and D.G. Gilabert. *Principes D'électrotechnique: Cours Et Exercices Corrigés*. Dunod, 2005. URL `https://books.google.it/books?id=lVmtAQAACAAJ`.

Roy Featherstone. *Rigid body dynamics algorithms*. Springer, 2014.

Gene H. Golub, Per Christian Hansen, and Dianne P. O'Leary. Tikhonov Regularization and Total Least Squares. *SIAM Journal on Matrix Analysis and Applications*, 21(1):185–194, jan 1999. ISSN 0895-4798, 1095-7162. doi: 10.1137/S0895479897326432. URL `http://epubs.siam.org/doi/10.1137/S0895479897326432`.

Nuno Guedelha, Naveen Kuppuswamy, Silvio Traversaro, and Francesco Nori. Self-calibration of joint offsets for humanoid robots using accelerometer measurements. In *Humanoid Robots (Humanoids), 16th IEEE-RAS International Conference on*, 2016.

A. Harnoy, B. Friedland, and S. Cohn. Modeling and measuring friction effects. *IEEE Control Systems Magazine*, 28(6):82–91, Dec 2008. ISSN 1066-033X. doi: 10.1109/MCS.2008.929546.

John Hollerbach, Wisama Khalil, and Maxime Gautier. Model identification. In *Springer Handbook of Robotics*, pages 321–344. Springer, 2008.

Hongliu Du and S.S. Nair. Modeling and compensation of low-velocity friction with bounds. *IEEE Transactions on Control Systems Technology*, 7 (1):110–121, jan 1999. ISSN 10636536. doi: 10.1109/87.736763. URL `http://ieeexplore.ieee.org/document/736763/`.

Panfeng Huang, Yangsheng Xu, and Bin Liang. Global Minimum-Jerk Trajectory Planning of Space Manipulator. page 9.

Levente Hunyadi. *Estimation methods in the errors-in-variables context.* PhD thesis, Budapest University of Technology and Economics, 2013.

Robert Keim. Low-pass filter a pwm signal into an analog voltage. Technical report, 2016.

Wisama Khalil and Etienne Dombre. *Modeling, identification and control of robots.* Butterworth-Heinemann, 2004.

Yadu Kiran and Dr. P.S. Puttaswamy. A Review of Brushless Motor Control Techniques. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 03(08):10963–10971, aug 2014. ISSN 23203765, 22788875. doi: 10.15662/ijareeie.2014.0308009. URL `http://www.ijareeie.com/upload/2014/august/9_AReview.pdf`.

Yadu Kiran and Dr.P.S.Putta Swamy. Field Oriented Control of a Permanent Magnet Synchronous Motor using a DSP. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 03(10):12364–12378, oct 2014. ISSN 23203765, 22788875. doi: 10.15662/ijareeie.2014.0310008. URL `http://www.ijareeie.com/upload/2014/october/13_Field.pdf`.

K.J. Kyriakopoulos and G.N. Saridis. Minimum jerk path generation. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pages 364–369, Philadelphia, PA, USA, 1988. IEEE Comput. Soc. Press. ISBN 978-0-8186-0852-0. doi: 10.1109/ROBOT.1988.12075. URL `http://ieeexplore.ieee.org/document/12075/`.

Claudia Latella and Dott Francesco Nori. Human Whole-Body Dynamics Estimation for Enhancing Physical Human-Robot Interaction. page 149.

Luc Le-Tien and Alin Albu-Schäffer. Decoupling and tracking control for elastic joint robots with coupled joint structure. *Advanced Robotics*, 31(4):184–203, feb 2017. ISSN 0169-1864, 1568-5535. doi: 10.1080/01691864.2016.1264886. URL `https://www.tandfonline.com/doi/full/10.1080/01691864.2016.1264886`.

Luc Le Tien, Alin Albu Schaffer, and Gerd Hirzinger. MIMO State Feedback Controller for a Flexible Joint Robot with Strong Joint Coupling. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3824–3830, Rome, Italy, apr 2007. IEEE. ISBN 978-1-4244-0602-9 978-1-4244-0601-2. doi: 10.1109/ROBOT.2007.364065. URL `http://ieeexplore.ieee.org/document/4209683/`.

Magnus Linderoth, Andreas Stolt, Anders Robertsson, and Rolf Johansson. Robotic force estimation using motor torques and modeling of low velocity friction disturbances. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3550–3556, Tokyo, nov 2013. IEEE. ISBN 978-1-4673-6358-7 978-1-4673-6357-0. doi: 10.1109/IROS.2013.6696862. URL `http://ieeexplore.ieee.org/document/6696862/`.

Yong Liu, Ning Xi, George Zhang, Xiongzi Li, Heping Chen, Chi Zhang, Michael J Jeffery, and Thomas A Fuhlbrigge. An automated method to calibrate industrial robot joint offset using virtual line-based single-point constraint approach. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 715–720. IEEE, 2009.

Donghai Ma, John M Hollerbach, and Yangming Xu. Gravity based autonomous calibration for robot manipulators. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pages 2763–2768. IEEE, 1994.

P. Maiolino, M. Maggiali, G. Cannata, G. Metta, and L. Natale. A flexible and robust large scale capacitive tactile system for robots. *Sensors Journal, IEEE*, 13(10):3910–3917, Oct 2013. ISSN 1530-437X. doi: 10.1109/JSEN.2013.2258149.

Cezary Maj and Andrzej Napieralski. Mechanical simulation of 3-axis accelerometer using a single proof-mass. In *2017 XIIIth International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH)*, pages 47–50, Lviv, Ukraine, April 2017. IEEE. ISBN 978-1-5386-4001-2. doi: 10.1109/MEMSTECH.2017.7937530. URL `http://ieeexplore.ieee.org/document/7937530/`.

Jerrold E Marsden and Tudor S Ratiu. *Introduction to mechanics and symmetry: a basic exposition of classical mechanical systems*, volume 17. Springer Science & Business Media, 2013.

A. Massarini and M. K. Kazimierczuk. Self-capacitance of inductors. *IEEE Transactions on Power Electronics*, 12(4):671–676, July 1997. ISSN 0885-8993. doi: 10.1109/63.602562.

Giorgio Metta, Lorenzo Natale, Francesco Nori, Giulio Sandini, David Vernon, Luciano Fadiga, Claes Von Hofsten, Kerstin Rosander, Manuel Lopes, José Santos-Victor, et al. The icub humanoid robot: An open-systems platform for research in cognitive development. *Neural Networks*, 23(8-9):1125–1134, 2010.

C. D. Meyer. *Matrix analysis and applied linear algebra*. Society for Industrial and Applied Mathematics, Philadelphia, 2000. ISBN 978-0-89871-454-8.

Austria Microsystems. *AS5311 - High Resolution Magnetic Linear Encoder*. Austria Microsystems, revision 1.13 edition, 09 2013. URL `https://ams.com/documents/20143/36005/AS5311_DS000200_2-00.pdf/856857e8-ca4a-31df-4c75-2fb9224bab8c`.

Austria Microsystems. *AS5045 - 12-Bit Programmable Magnetic Rotary position sensor*. Austria Microsystems, revision 2.01 edition, 07 2017. URL `https://ams.com/documents/20143/36005/AS5045_DS000101_2-00.pdf/5a2a30ca-e323-82dd-63a9-8fc3d5f7b6c4`.

Philipp Mittendorfer and Gordon Cheng. Open-loop self-calibration of articulated robots with artificial skins. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4539–4545. IEEE, 2012.

Philipp Mittendorfer, Emmanuel Dean, and Gordon Cheng. Automatic robot kinematic modeling with a modular artificial skin. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 749–754. IEEE, 2014.

Jari Mäkinen. Rotation manifold SO(3) and its tangential vectors. *Computational Mechanics*, 42(6):907–919, nov 2008. ISSN 0178-7675, 1432-0924. doi: 10.1007/s00466-008-0293-z. URL `http://link.springer.com/10.1007/s00466-008-0293-z`.

S. Traversaro N. Guedelha and D. Pucci. Identification of motor parameters on coupled joints. *Intelligent Systems Conference (IntelliSys)*, 2019.

Francesco Nori, Naveen Kuppuswamy, and Silvio Traversaro. Simultaneous state and dynamics estimation in articulated structures. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3380–3386, Hamburg, Germany, September 2015a. IEEE. ISBN 978-1-4799-9994-1. doi: 10.1109/IROS.2015.7353848. URL `http://ieeexplore.ieee.org/document/7353848/`.

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, 2015b. ISSN 2296-9144. doi: 10.3389/frobt.2015.00006. URL `https://www.frontiersin.org/article/10.3389/frobt.2015.00006`.

H. Olsson, K.J. Åström, C. Canudas de Wit, M. Gäfvert, and P. Lischinsky. Friction Models and Friction Compensation. *European Journal of Control*, 4(3):176–195, jan 1998. ISSN 09473580. doi: 10.1016/S0947-3580(98)70113-X. URL `https://linkinghub.elsevier.com/retrieve/pii/S094735809870113X`.

Frank W. J. Olver and National Institute of Standards and Technology (U.S.), editors. *NIST handbook of mathematical functions*. Cambridge University Press : NIST, Cambridge ; New York, 2010. ISBN 978-0-521-19225-5 978-0-521-14063-8. OCLC: ocn502037224.

A. Parmiggiani, M. Randazzo, L. Natale, G. Metta, and G. Sandini. Joint torque sensing for the upper-body of the icub humanoid robot. In *2009 9th IEEE-RAS International Conference on Humanoid Robots*, pages 15–20, Dec 2009. doi: 10.1109/ICHR.2009.5379525.

Alberto Parmiggiani, Marco Maggiali, Lorenzo Natale, Francesco Nori, Alexander Schmitz, Nikos Tsagarakis, José Santos Victor, Francesco Becchi, Giulio Sandini, and Giorgio Metta. The design of the icub humanoid robot. *International Journal of Humanoid Robotics*, 09 (04):1250027, dec 2012. ISSN 0219-8436, 1793-6942. doi: 10.1142/S0219843612500272. URL `http://www.worldscientific.com/doi/abs/10.1142/S0219843612500272`.

M.R. Popovic and A.A. Goldenberg. Modeling of friction using spectral analysis. *IEEE Transactions on Robotics and Automation*, 14(1):114–122, feb 1998. ISSN 1042296X. doi: 10.1109/70.660854. URL `http://ieeexplore.ieee.org/document/660854/`.

William H. Press, editor. *Numerical recipes in C: the art of scientific computing*. Cambridge University Press, Cambridge ; New York, 2nd ed edition, 1992. ISBN 978-0-521-43108-8 978-0-521-43720-2.

CG Rajeevlochana, SK Saha, and S Kumar. Automatic extraction of dh parameters of serial manipulators using line geometry. In *The 2nd International Conference on Multibody System Dynamics*, 2012.

Alessandro Roncone, Marco Hoffmann, Ugo Pattacini, and Giorgio Metta. Automatic kinematic chain calibration using artificial skin: self-touch in the icub humanoid robot. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2305–2312. IEEE, 2014.

Nicholas Rotella, Sean Mason, Stefan Schaal, and Ludovic Righetti. Inertial sensor-based humanoid joint state estimation. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1825–1831. IEEE, 2016.

Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

D. E. Serrano, Y. Jeong, V. Keesara, W. K. Sung, and F. Ayazi. Single proof-mass tri-axial pendulum accelerometers operating in vacuum. In *2014 IEEE 27th International Conference on Micro Electro Mechanical Systems (MEMS)*, pages 28–31, San Francisco, CA, USA, January 2014. IEEE. ISBN 978-1-4799-3509-3. doi: 10.1109/MEMSYS.2014.6765565. URL `http://ieeexplore.ieee.org/document/6765565/`.

Bruno Siciliano, editor. *Robotics: modelling, planning and control*. Advanced textbooks in control and signal processing. Springer, London, 2009. ISBN 978-1-84628-641-4 978-1-84628-642-1. OCLC: ocn144222188.

John W L Simpson, Chris D Cook, and Zheng Li. Sensorless Force Estimation for Robots with Friction. page 6, November 2002.

STMicroelectronics. *6-point tumble sensor calibration*. STMicroelectronics, 12 2015. DT0053 Rev. 1.

Andreas Stolt, Fredrik Bagge Carlson, M. Mahdi Ghazaei Ardakani, Ivan Lundberg, Anders Robertsson, and Rolf Johansson. Sensorless friction-compensated passive lead-through programming for industrial robots. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3530–3537, Hamburg, Germany, sep 2015a. IEEE. ISBN 978-1-4799-9994-1. doi: 10.1109/IROS.2015.7353870. URL `http://ieeexplore.ieee.org/document/7353870/`.

Andreas Stolt, Anders Robertsson, and Rolf Johansson. Robotic force estimation using dithering to decrease the low velocity friction uncertainties. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3896–3902, Seattle, WA, USA, may 2015b. IEEE. ISBN 978-1-4799-6923-4. doi: 10.1109/ICRA.2015.7139742. URL `http://ieeexplore.ieee.org/document/7139742/`.

Chih-Ming Sun, Ming-Han Tsai, Yu-Chia Liu, and Weileun Fang. Implementation of a Monolithic Single Proof-Mass Tri-Axis Accelerometer Using CMOS-MEMS Technique. *IEEE Transactions on Electron Devices*, 57(7):1670–1679, July 2010. ISSN 0018-9383, 1557-9646. doi: 10.1109/TED.2010.2048791. URL `http://ieeexplore.ieee.org/document/5467179/`.

Avago Technologies. *Avago AEDR-850x - 3 Channel Reflective Incremental Encoders*. Avago Technologies, av02-2790en edition, February 2011. URL `https://www.broadcom.com/products/motion-control-encoders/incremental-encoders/reflective-encoders/aedr-8501-100`.

S. Traversaro, D. Pucci, and F. 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, May 2015. doi: 10.1109/ICRA.2015.7139477.

Silvio Traversaro. Modelling, estimation and identification of humanoid robots dynamics (phd thesis). Italian Institute of Technology. Retrieved from https://traversaro.github.io/preprints/traversaro-phd-thesis.pdf, 2017.

N. G. Tsagarakis, B. Vanderborght, M. Laffranchi, and D. G. Caldwell. The mechanical design of the new lower body for the child humanoid robot 'icub'. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4962–4968, Oct 2009. doi: 10.1109/IROS.2009.5354498.

Nikolaos G. Tsagarakis, Martin Sinclair, Francesco Becchi, Giorgio Metta, Giulio Sandini, and Darwin G. Caldwell. Lower body design of the 'icub' a human-baby like crawling robot. *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pages 450–455, 2006.

Nikolaos G Tsagarakis, Giorgio Metta, Giulio Sandini, David Vernon, Ricardo Beira, Francesco Becchi, Ludovic Righetti, Jose Santos-Victor, Auke Jan Ijspeert, Maria Chiara Carrozza, et al. icub: the design and realization of an open humanoid platform for cognitive and neuroscience research. *Advanced Robotics*, 21(10):1151–1175, 2007.

V Van Geffen. A study of friction models and friction compensation. *DCT*, 118:24, 2009.

J. Vihonen, J. Honkakorpi, J. Mattila, and A. Visa. Geometry-aided angular acceleration sensing of rigid multi-body manipulator using mems rate gyros and linear accelerometers. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2514–2520, Nov 2013. doi: 10.1109/IROS.2013.6696710.

Juho Vihonen, Janne Honkakorpi, Jouni Mattila, and Ari Visa. Novel pairwise coupled kinematic solution for algebraic angular acceleration estimation of serial link manipulators. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 809–814, Seattle, WA, USA, may 2015. IEEE. ISBN 978-1-4799-6923-4. doi: 10.1109/ICRA.2015.7139271. URL `http://ieeexplore.ieee.org/document/7139271/`.

Erhard Wieser, Philipp Mittendorfer, and Gordon Cheng. Accelerometer based robotic joint orientation estimation. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 67–74. IEEE, 2011.

Ayman Y Yousef and S M Abdelmaksoud. Review on Field Oriented Control of Induction Motor. 2(7):13, 2015.

Bruno Zappa, Giovanni Legnani, Anton J Van Den Bogert, and Riccardo Adamini. On the number and placement of accelerometers for angular velocity and acceleration determination. *TRANSACTIONS-AMERICAN SOCIETY OF MECHANICAL ENGINEERS JOURNAL OF DYNAMIC SYSTEMS MEASUREMENT AND CONTROL*, 123(3):552–553, 2001.

Prepared Jian Zhao and Yangwei Yu. Brushless DC Motor Fundamentals Application Note. page 19, 2014.

Appendices

Appendix A

Incremental and Absolute Rotary Encoders Integrated in iCub

The iCub humanoid robot is equipped with different types of rotary encoders for measuring the state of its 53 DoF. The major joints (arms, legs, torso) are actuated by motor groups composed of a Moog brushless motor and an harmonic drive in a very compact configuration. For the joint position sensing, the group contains a 12-bit resolution, absolute hall-effect encoder (AS5045 microchip from Austria Microsystems [Microsystems, 2013]). It is located at the joint side, after the harmonic drive, such that it measures the links relative angular position and not the motor's rotation [Parmiggiani et al., 2012, section 4.3]. This encoder's resolution turned to be insufficient for measuring the velocity of dynamic motions as the smallest detectable velocity obtained through pulse counting was $\Delta q/T_s = 0.088^\circ/0.001\text{s} = 88^\circ\,\text{s}^{-1}$ T_s being the encoder sampling period. For improving the velocity measurement accuracy, an optical 13-bit incremental encoder was mounted inside the motor housing as shown in figure A.1. As the encoder resolution is combined with the reduction drive ratio of 1:100, we get the equivalent resolution at joint level of $360/(2^{13} \times 100) = 0.0004^\circ$, and the smallest detectable velocity becomes $\Delta q/T_s = 0.0004^\circ/0.001\text{s} = 0.4^\circ\,\text{s}^{-1}$ [Tsagarakis et al., 2009, section A].

Because of space constraints, the absolute encoder (joint side) was mounted behind the rear end of the motor shaft (figure A.1), but still being connected to the output shaft across the hollow shaft of the motor.

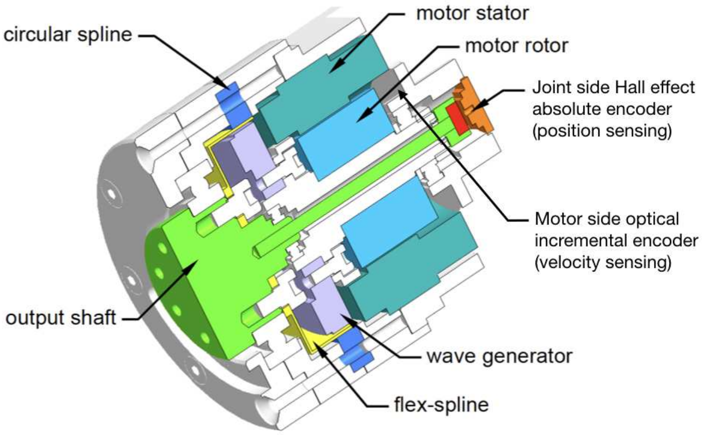


Fig. A.1 Motor group cross section, showing the Moog brushless motor, the harmonic drive and the angle position encoders. Source: [Parmiggiani et al., 2012, Fig.3]

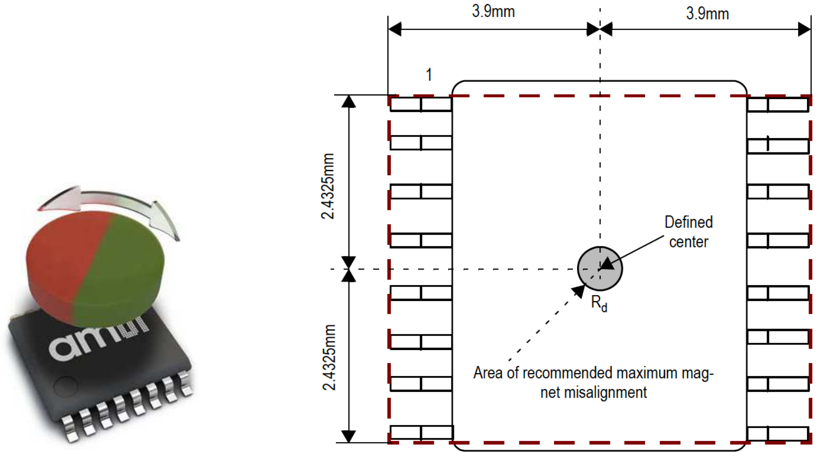
Definition - Absolute and relative (incremental) encoders: Relative encoders provide the linear or angular displacement measured since the instant they were turned on and the sensing activated. Absolute encoders provide the displacement with respect to a chosen position previously programmed or set by design.

A.1 Joint Side Absolute Encoders

In this section, we describe the key features, application interface and performance indicators of the AS5045 magnetic absolute encoder mounted on the major iCub joints (legs, torso, shoulders and elbows), measuring the output shaft position. All this information can be found in the datasheet [Microsystems, 2017].

The AS5045 is a contactless magnetic position sensor for accurate angular measurement over a full turn of 360°. It is a system-on-chip, combining integrated Hall elements, analog front end amplifier, digital signal processing and a serial interface in a single device.

To measure the angle, only a simple two-pole magnet, rotating over the



(a) (b)

Fig. A.2 Typical arrangement for placing the rotating magnet close to the sensing surface of the AS5045 chipset. Source: [Microsystems, 2017]

center of the chip, is required. The magnet (diametrically magnetized) may be placed above or below the IC as the typical arrangement shown in figure A.2. On a typical motor group configuration from iCub, the magnet is fixed on the output shaft tip appearing at the back of the motor as shown in Figure A.1.

The Coordinate Rotation Digital Computer (CORDIC) combines the magnitudes of the Hall sensors array signals for estimating the angular position of the rotating magnet. The estimation is provided with a typical delay of 384µs (in slow mode) and a resolution of $0.0879° = 4096$ positions per revolution. As illustrated in the block diagram A.4, this digital data is available: as a serial bit stream through a serial interface; or as a Pulse Width Modulated (PWM) signal.

A.1.1 Accuracy and Repeatability

The accuracy is defined by the error between the measured angle (digital output of the sensor) and the real angular position of the magnet rotating over the sensing area, and depends on several factors: - the non-linearity of the analog signal processing and amplification, - the non-linearity due to the magnet misalignment with respect to the ideal central position.

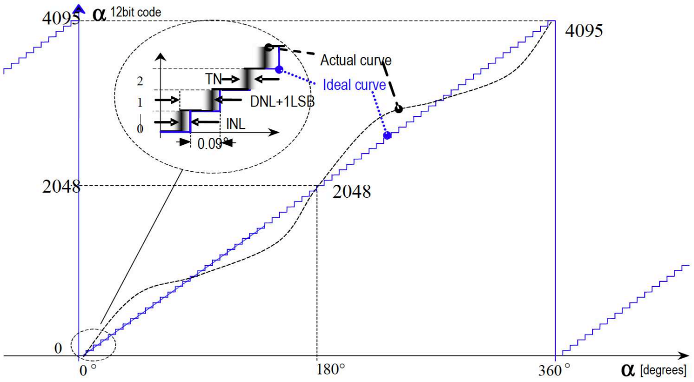


Fig. A.3 Integral and differential non-linearity errors. Source: [Microsystems, 2017, Fig.9]

These factors cause mainly three types of errors illustrated in Figure A.3:

- the Integral Non-Linearity (INL) error is the maximum deviation between the estimated position (”actual curve” in the plot) and the real position (”ideal curve” in the plot). For a perfectly centered magnet, this error is ±0.5 degrees at 25 °C and further increases for any misalignment,
- the Differential Non-Linearity (DNL) is the maximum deviation of the step length on the ”actual curve” with respect to the step length on the ”ideal curve” from one position to the next (1 LSB),
- the Transition Noise (TN) is the repeatability of an estimated position, i.e. the standard deviation of the INL for any fixed chosen real position over many measurement trials.

We describe in tables A.1 and A.2 some of the system specifications—digital output resolution, error values due to the system non-linearity, transition noise.

Symbol	Parameter	Conditions	Max	Units
RES	Resolution		12	bit
1LSB	Less Significant Bit step length		0.088	deg
INL$_{opt}$	Integral non-linearity (optimum)	Maximum error with respect to the best line fit. Centered magnet without calibration, TAMB = 25 °C	±0.5	deg
INL$_{temp}$	Integral non-linearity (optimum)	Maximum error with respect to the best line fit. Centered magnet without calibration, TAMB = -40 °C to 125 °C	±0.9	deg
INL	Integral non-linearity	Best line fit = (Errmax − Errmin) / 2 Over displacement tolerance with 6mm diameter magnet, without calibration, TAMB = -40 °C to 125 °C	±1.4	deg
DNL	Differential non-linearity	12-bit, No missing codes	±0.044	deg
TN	Transition noise	1 sigma, fast mode (MODE = 1)	±0.06	deg RMS
		1 sigma, slow mode (MODE=0 or open)	±0.03	deg RMS
$F_s(T_s)$	Sampling rate (period)	1 sigma, fast mode (MODE = 1)	10.42	kHz
		1 sigma, slow mode (MODE=0 or open)	2.61	kHz

Table A.1 Resolution and accuracy characteristics. Source: [Microsystems, 2017, Fig.8]

Symbol	Parameter	Conditions	Max	Units
$\dot{q}_{max}$	Max. joint velocity @RES 12 bit	1 sigma, fast mode (MODE = 1)	16	$\mathrm{rad\,s^{-1}}$
		1 sigma, slow mode (MODE=0 or open)	3.97	$\mathrm{rad\,s^{-1}}$

Table A.2 Maximum recommended speed at slow and fast modes. Source: [Microsystems, 2017, Fig.12]

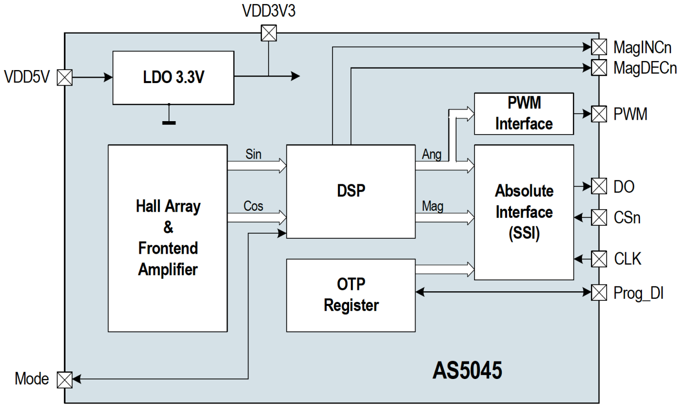


Fig. A.4 AS5045 block diagram. Source: [Parmiggiani et al., 2012, Fig.2]

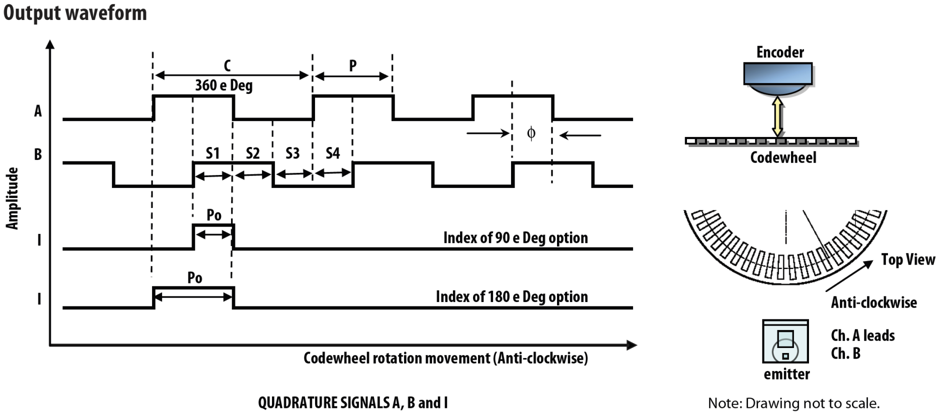


Fig. A.5 AEDR-850x encoder output waveform. Source: [Technologies, 2011, Page 2]

A.2 Motor Side Incremental Encoders

In this section, we describe the key features, application interface and performance indicators of the Avago AEDR-850x high resolution, three channel optical incremental encoder mounted on the major iCub joint motors, measuring the motor angular position as illustrated in Figure A.1. The encoder provides two channel quadrature digital outputs for position and direction sensing, and an index digital output. It has a built-in interpolation with a factor of 1x, 2x and 4x selectable via external pinouts, and an encoding resolution of 12 lines/mm.

We can see in Figure A.5 a typical waveform readout for an anti-clockwise codewheel rotation, having a leading channel A. The typical codewheel, shown in Figure A.6, is a reflective disk with a circular pattern of opaque radial tracks (lines). Each track defines a relative angular position and each transition between two consecutive tracks is detected as a sequence of four events in the channel waveforms A and B: rising edge of A → rising edge of B → falling edge of A → falling edge of B. This sequence generates four counts within the total amount of counts in a full 360° rotation. Hence, the Counts Per Rotation (CPR) defines the encoder resolution and is computed as described in Figure A.7. On iCub motors, the codewheel operation radius $R_{\mathrm{OP}} = 13.298\mathrm{mm}$, so we get $\mathrm{CPR} = 1000\,\mathrm{lines} \times 4\,\mathrm{counts/line} = 4000$. An interpolation factor of 2x in the actual iCub sensors configuration rises the CPR to 8000, with a resulting resolution of around 0.05°.

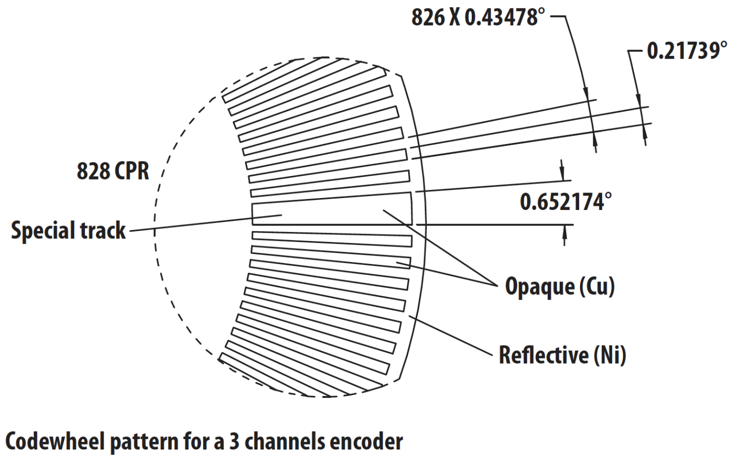
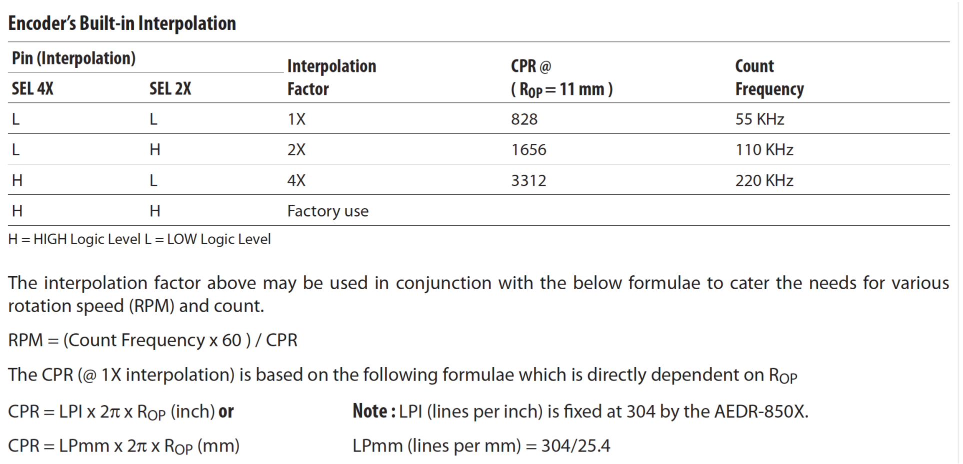


Fig. A.6 Typical codewheel with a special index track. Source: [Technologies, 2011, Page 5]



Encoder's Built-in Interpolation

Pin (Interpolation)				
SEL 4X	SEL 2X	Interpolation Factor	CPR @ (R_{OP} = 11 mm)	Count Frequency
L	L	1X	828	55 KHz
L	H	2X	1656	110 KHz
H	L	4X	3312	220 KHz
H	H	Factory use		

H = HIGH Logic Level L = LOW Logic Level

The interpolation factor above may be used in conjunction with the below formulae to cater the needs for various rotation speed (RPM) and count.

RPM = (Count Frequency x 60) / CPR

The CPR (@ 1X interpolation) is based on the following formulae which is directly dependent on R_{OP}

CPR = LPI x 2π x R_{OP} (inch) **or**

CPR = LPmm x 2π x R_{OP} (mm)

Note : LPI (lines per inch) is fixed at 304 by the AEDR-850X.

LPmm (lines per mm) = 304/25.4

Fig. A.7 AEDR-850x built-in interpolation. Source: [Technologies, 2011, Page 6]

Appendix B

Overview on the Rigid Body Dynamics

B.1 Overview of the Notation

The notation used for describing the models and the algorithms in this thesis is mostly based on spatial vectors, as defined in [Featherstone, 2014] and in [Traversaro, 2017], as shown below. The notation presented below is the general notation, but each chapter can present its own simplified notation when necessary.

A, B	cartesian coordinate frames
p	an arbitrary point
o_B	origin of frame B
$[A]$	orientation frame associated to A
$B[A]$	frame with origin o_B and orientation $[A]$
${}^A p$	coordinates of p written in A
${}^A o_B$	coordinates of o_B written in A
${}^A R_B$	rotation transformation ($\in SO(3)$) of a 3D vector from B to A
${}^A H_B$	homogeneous transformation ($\in SE(3)$) of a 4 x 1 homogeneous vector from B to A
${}^A X_B$	transformation of a twist (or spatial motion vector) from B to A
${}_A X^B$	wrench transformation from B to A
${}^C v_{A,B}$	twist ($\in \mathbb{R}^3$) velocity of body frame B w.r.t. body frame A written in frame C. Same notation for an acceleration (${}^C a_{A,B}$)
${}^k v_{i,j}$	equivalent to the notation ${}^C v_{A,B}$ but having the A, B and C frames respectively indexed as i, j and k
${}_B f$	coordinates of the wrench f written in B
$y^\wedge$	cross product in $\mathbb{R}^3$, 3×3 *skew-symmetric* matrix of vector y ($y \in \mathbb{R}^3$)
$Y^\vee$	vector such that $(Y^\vee)^\wedge = Y$ ($Y \in \mathbb{R}^{3\times 3}$)
${}^C \overline{v}_{A,B}$	linear component of ${}^C v_{A,B}$
${}^C v_{A,B}\times$	matrix representation of the twist cross product in $\mathbb{R}^6$ (refer to B.3)
${}^C v_{A,B}\times^*$	matrix representation of the wrench (twist dual) cross product in $\mathbb{R}^6$ (refer to B.3)
$\hat{x}$	estimated twist or wrench $\in \mathbb{R}^3$
${}^A g$	ground truth gravity vector ($\in \mathbb{R}^3$) written in frame A
$\mathbb{R}$	set of real numbers
$\mathbb{I}_n$	identity matrix $\in \mathbb{R}^{n\times n}$
0_n	zero column vector $\in \mathbb{R}^n$
$0_{n\times m}$	zero matrix $\in \mathbb{R}^{n\times n}$

Remark B.1. *One important difference with the notation used in [Featherstone, 2014] is the dot operator* $(\dot{-})$. *Throughout this thesis, given a position vector* $\in \mathbb{R}^3$ *or a velocity vector in* $\in \mathbb{R}^3$ *or* $\in \mathbb{R}^6$, *we define the* dot operator $(\dot{-})$ *as the total time derivative of the quantity. Thus, given a 6-D vector*

${}^{C}\mathrm{V}$ *expressed in the frame* C*, we have*

$${}^{C}\dot{\mathrm{V}} = \frac{d}{dt}\left({}^{C}\mathrm{V}\right)$$

B.2 Math Preliminaries

- For n matrices of arbitrary dimensions $M_1, M_2, \cdots, M_n$, we define the operator $\operatorname{diag}(M_1, \cdots, M_n)$ as follows:

  $$\operatorname{diag}(M_1, \cdots, M_n) = \begin{bmatrix} M_1 & & \\ & \ddots & \\ & & M_n \end{bmatrix} ,$$

  with the matrices M_1 to M_n being the only non null elements.

- Given two matrices $A \in \mathbb{R}^{p\times q}$, $B \in \mathbb{R}^{q\times m}$, and the identity matrix $\mathbb{I}_m$, we then have the property $\operatorname{vec}(AB) = (\mathbb{I}_m \otimes A)\operatorname{vec}(B)$.

  $$\begin{aligned} \operatorname{vec}(AB) &= \operatorname{vec}\left(\begin{bmatrix} Ab_1 & Ab_2 & \cdots & Ab_m \end{bmatrix}\right) && \text{(B.1)} \\ &= \begin{bmatrix} A & 0 & \cdots & 0 \\ 0 & A & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & A \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} && \text{(B.2)} \\ &= (\mathbb{I}_m \otimes A)\operatorname{vec}(B) , && \text{(B.3)} \end{aligned}$$

  where b_i is a column vector of dimensions $q \times 1$.

B.3 Frame Kinematics

B.3.1 Inertial Frame

As per Newton's laws of mechanics, an inertial frame is a frame where a point mass, which is not subject to an external force, runs a trajectory at constant velocity and following a straight line.

In a formalism where the gravity effect is considered as a force, which is a common practice in robotics problems near the earth's surface, i.e. in low atmosphere or on the ground, we refer to an inertial absolute frame, or world frame W, a frame fixed to the lab ground. this concept disregards any non-inertial effect due to the Earth's motion in space, more specifically the Coriolis effect due to the rotation around its own axis. An inertial frame would then be any frame which motion, with respect to the absolute frame W, follows a straight line at constant velocity. This is the definition that we will use throughout this and the following chapters.

B.3.2 Coordinate Acceleration

The coordinate acceleration of a rigid body is the total time derivative of its velocity with respect to the world frame W (or any inertial frame). For a moving point P whose position in the world frame is given by the coordinates vector p, the linear acceleration of P with respect to frame W, expressed in frame W, would be:

$$ {}^{\mathrm{W}}a_{\mathrm{W,P}} = \frac{d^2p}{dt^2} = \ddot{p}\ . $$

B.3.3 Proper Acceleration

The proper acceleration is the apparent acceleration measured by a linear accelerometer, such that if the sensor is fixed to an inertial frame, the sensor will measure $-g$, with g being the vector defining the gravitation field. If the sensor is fixed in a moving body B, has a sensor frame denoted S, and the sensor frame origin position in the world frame is given by $o_S \in \mathbb{R}^3$, the sensor measures the quantity:

$$ \alpha^g = {}^{\mathrm{S}}R_{\mathrm{W}}\frac{d^2 o_S}{dt^2} - {}^{\mathrm{S}}g = {}^{\mathrm{S}}R_{\mathrm{W}}\,\ddot{o}_{\mathrm{S}} - {}^{\mathrm{S}}g\ . $$

Note. *Here the sensor is assumed flawless, i.e. the measurements are isotropic: all axes have the same gain and zero offset; the cross-axis gains are null.*

B.3.4 Gravitational Field Numerical Approximated Value

In this thesis, we will consider Earth's gravity as a locally uniform field with the approximated value of $9.81\,\mathrm{m\,s^{-2}}$.

B.3.5 Body Angular Velocity

We define two coordinate frames A and B with the same origin O. Let P be a fixed point in B, p_A and p_B the respective coordinates in A and B.

The coordinates transformation from A to B can be described by a rotation operator that belongs to the Special non-commutative Lie-group of proper Orthogonal linear transformations, $SO(3)$, defined as:

$$SO(3) := \left\{ R : \mathbb{R}^3 \to \mathbb{R}^3 | R^\top R = \mathbb{I}_3,\ \det\ R = 1 \right\} \tag{B.4}$$

where R is the 3×3 matrix representing the rotation transformation [Marsden and Ratiu, 2013, Section 9.2]. We have chosen here the notation R instead of ${}^{A}R_{B}$ for the sake of simplicity. We can then write:

$$p_A = R\, p_B \tag{B.5}$$

An angular motion of the frame B being reflected in a change of R, the angular velocity could be represented by the time derivative of $\dot{R}$:

$$\dot{p}_A = \dot{R}\, p_B \tag{B.6}$$

But we can define a more compact representation of the angular velocity, with a 3×1 vector operator instead of the 3×3 matrix $\dot{R}$. In view of (B.6), we can write:

$$\dot{p}_A = \dot{R} R^\top\, p_A, \tag{B.7}$$

and we can show that $\dot{R}R^\top$ is *skew-symmetric*:

$$\begin{aligned} RR^\top &= \mathbb{I}_3 \Rightarrow \dot{R}R^\top + R\dot{R}^\top = 0_3, \\ \dot{R}R^\top &= -R\dot{R}^\top, \\ \dot{R}R^\top &= -\left(\dot{R}R^\top\right)^\top, \end{aligned}$$

which is the property of the set $\mathfrak{so}(3)$ of 3×3 *skew-symmetric* matrices [Marsden and Ratiu, 2013, Section 9.2]:

$$\mathfrak{so}(3) := \left\{ S \in \mathbb{R}^{3\times3} | S^T = -S \right\} \tag{B.8}$$

Given a vector $w \in \mathbb{R}^3$, we define the operator "hat", mapping w to the *skew-symmetric* matrix $w^\wedge$:

$$w^\wedge = \begin{bmatrix} x \\ y \\ z \end{bmatrix}^\wedge := \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \in \mathfrak{so}(3) \tag{B.9}$$

Given a *skew-symmetric* matrix $W = w^\wedge$, we define "vee", the inverse operator of "hat", mapping W to the vector $w = W^\vee$:

$$W^\vee = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}^\vee := \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^3 \tag{B.10}$$

Remark B.2. *For any given vectors* $w = \begin{bmatrix} x & y & z \end{bmatrix}^\top \in \mathbb{R}^3$ *and* $m = \begin{bmatrix} m_1 & m_2 & m_3 \end{bmatrix}^\top \in \mathbb{R}^3$*, the cross product between these vectors can be written as:*

$$\begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}^\wedge \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} = w^\wedge m \tag{B.11}$$

We can then rewrite (B.6) as follows:

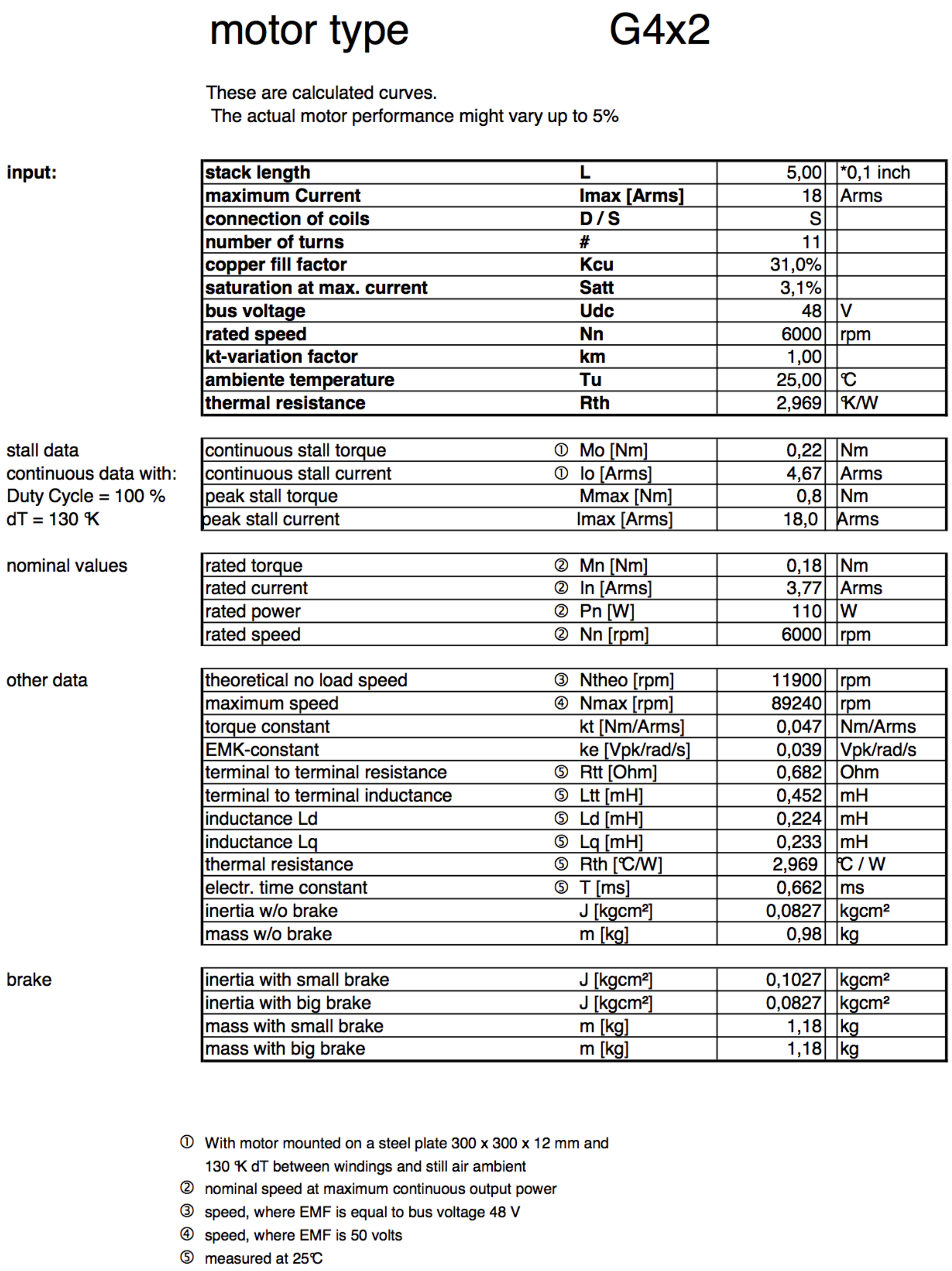
$$\dot{p}_A = {}^A\omega^\wedge_{A,B}\, p_A \tag{B.12}$$

where ${}^A\omega^\wedge_{A,B} = \dot{R}R^\top$ represents the angular velocity of frame B with respect to frame A expressed in A (*right trivialized*), or instead write:

$$R^\top \dot{p}_A = {}^B\omega^\wedge_{A,B}\, p_B \tag{B.13}$$

where ${}^B\omega^\wedge_{A,B} = R^\top \dot{R}$ represents the angular velocity of frame B with respect to frame A, expressed in B (*left trivialized*).

Appendix C
iCub Main Motors Datasheet



motor type G4x2

These are calculated curves.
The actual motor performance might vary up to 5%

input:

stack length		**L**	5,00	*0,1 inch
maximum Current		**Imax [Arms]**	18	Arms
connection of coils		**D / S**	S	
number of turns		**#**	11	
copper fill factor		**Kcu**	31,0%	
saturation at max. current		**Satt**	3,1%	
bus voltage		**Udc**	48	V
rated speed		**Nn**	6000	rpm
kt-variation factor		**km**	1,00	
ambiente temperature		**Tu**	25,00	℃
thermal resistance		**Rth**	2,969	°K/W

stall data
continuous data with:
Duty Cycle = 100 %
dT = 130 °K

continuous stall torque	①	Mo [Nm]	0,22	Nm
continuous stall current	①	Io [Arms]	4,67	Arms
peak stall torque		Mmax [Nm]	0,8	Nm
peak stall current		Imax [Arms]	18,0	Arms

nominal values

rated torque	②	Mn [Nm]	0,18	Nm
rated current	②	In [Arms]	3,77	Arms
rated power	②	Pn [W]	110	W
rated speed	②	Nn [rpm]	6000	rpm

other data

theoretical no load speed	③	Ntheo [rpm]	11900	rpm
maximum speed	④	Nmax [rpm]	89240	rpm
torque constant		kt [Nm/Arms]	0,047	Nm/Arms
EMK-constant		ke [Vpk/rad/s]	0,039	Vpk/rad/s
terminal to terminal resistance	⑤	Rtt [Ohm]	0,682	Ohm
terminal to terminal inductance	⑤	Ltt [mH]	0,452	mH
inductance Ld	⑤	Ld [mH]	0,224	mH
inductance Lq	⑤	Lq [mH]	0,233	mH
thermal resistance	⑤	Rth [℃/W]	2,969	℃ / W
electr. time constant	⑤	T [ms]	0,662	ms
inertia w/o brake		J [kgcm²]	0,0827	kgcm²
mass w/o brake		m [kg]	0,98	kg

brake

inertia with small brake		J [kgcm²]	0,1027	kgcm²
inertia with big brake		J [kgcm²]	0,0827	kgcm²
mass with small brake		m [kg]	1,18	kg
mass with big brake		m [kg]	1,18	kg

① With motor mounted on a steel plate 300 x 300 x 12 mm and
130 °K dT between windings and still air ambient
② nominal speed at maximum continuous output power
③ speed, where EMF is equal to bus voltage 48 V
④ speed, where EMF is 50 volts
⑤ measured at 25℃

Fig. C.1 MOOG C2900584 Datasheet

Appendix D

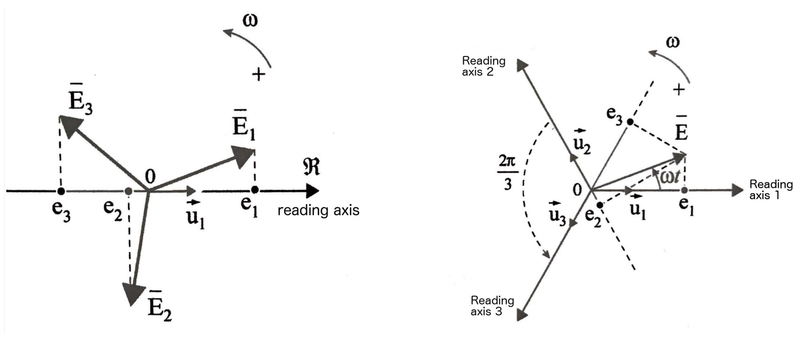
From Phase Winding Currents to a Rotating Magnetic Field

In order to understand how the phase currents combine in space for generating a rotating magnetic field space vector, we first consider the initial model for a single phase, let it be the phase U, as shown in figure 2.15, and express the current and voltage quantities in a Fresnel diagram.

Axiom D.1. *Any scalar time varying sinusoidal quantity of angular velocity ω can be expressed in the complex plane $(O,\mathfrak{R},\mathfrak{I})$ as the projection of a temporal vector rotating clockwise at the angular velocity ω. A voltage $e(t) = A\,\cos(\omega\,t + \phi)$ would be the projection of $\overline{E} = A\,e^{j\omega t}\,e^{\phi}$. We will refer to the Real ($\mathfrak{R}$) axis as the Reading Axis, to $\overline{E}$ as the Voltage Time Vector, to $e(t)$ as the Voltage Time Scalar. The described complex plane is a Time Fresnel plane.*

Such representation of vectors are used in Fresnel diagrams. So we represent the phase model current and voltages as complex vectors rotating at the same velocity ω as the rotor: namely the phase input voltage $\overline{E}_{\mathrm{U}}$ and current $\overline{I}_{\mathrm{U}}$, the voltages on the winding resistance $r\,\overline{I}_{\mathrm{U}}$ and self-inductance voltage $jL\omega\,\overline{I}_{\mathrm{U}}$. For the sake of simplicity we just consider $\overline{E}_{\mathrm{U}}$ and $\overline{I}_{\mathrm{U}}$ from now on.

We now consider any three Time Vectors $\overline{E}_1$, $\overline{E}_2$ and $\overline{E}_3$ of equal amplitude, phase-shifted by 120 degrees such that $\overline{E}_1 + \overline{E}_2 + \overline{E}_3 = 0$ in the Time Fresnel plane. The respective projections e_1, e_2 and e_3 on the "reading" axis $\mathfrak{R}$ (Time Fresnel diagram D.1a) can alternatively be expressed as the projec-



(a) (b)

Fig. D.1 Time Fresnel diagram (a): instantaneous tri-phase voltages e_1, e_2 and e_3 as projections of complex rotating Time Vectors $\overline{E}_1$, $\overline{E}_1$ and $\overline{E}_3$ on a single Reading Axis, the Real axis $\mathcal{R}_e$. Space Fresnel diagram (b): e_1, e_2 and e_3 as projections of a single complex rotating Time Vector $\overline{E}$ on three Reading Axes phase-shifted by 120 degrees. Source: [Dixneuf and Gilabert, 2005]

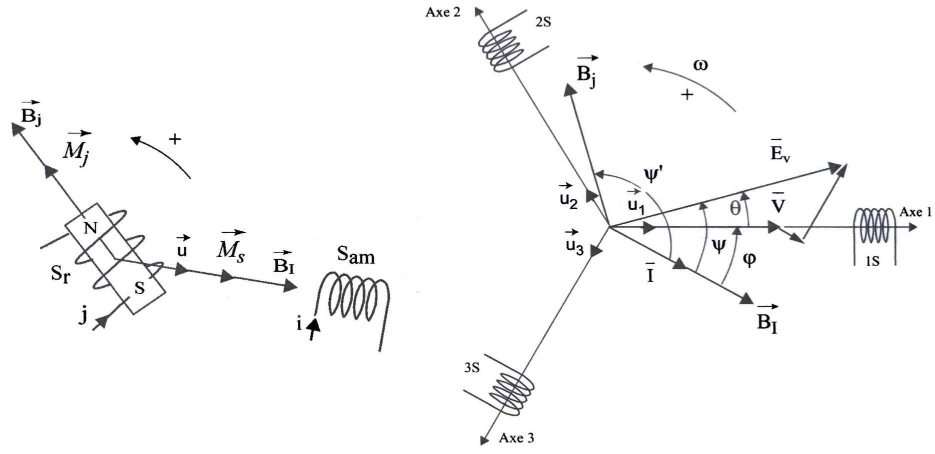


Fig. D.2 Interaction between the rotor magnetic moment $\vec{M}_j$ and the combined stator magnetic rotating field $\vec{B}_I$: $C_{em} = \vec{M}_j \wedge \vec{B}_I$ or $C_{em} = \vec{B}_j \wedge \vec{M}_s$ with $\vec{M}_s = \vec{u}\, i\, S_{am}$ where S_{am} is the winding apparent surface (a). Time and Space Fresnel diagram describing the combined rotating field $\vec{B}_I$. Source: [Dixneuf and Gilabert, 2005] tions of a single rotating vector $\overline{E}$ on three "reading" axes phase-shifted by 120 degrees [Dixneuf and Gilabert, 2005] as shown below and in the Space Fresnel diagram D.1b:

$$\overline{E}_1 = \vec{u}_1\, E\,\sqrt{2}\, e^{j\omega t} \qquad e_1 = E\,\sqrt{2}\,\cos(\omega t) \tag{D.1}$$

$$\overline{E}_2 = \vec{u}_1\, E\,\sqrt{2}\, e^{j\omega t}\, e^{-j\frac{2\pi}{3}} \qquad e_1 = E\,\sqrt{2}\,\cos(\omega t - \frac{2\pi}{3}) \tag{D.2}$$

$$\overline{E}_3 = \vec{u}_1\, E\,\sqrt{2}\, e^{j\omega t}\, e^{-j\frac{4\pi}{3}} \qquad e_1 = E\,\sqrt{2}\,\cos(\omega t - \frac{4\pi}{3}) \tag{D.3}$$

$$\boxed{\overline{E}_1 + \overline{E}_2 + \overline{E}_3 = 0} \qquad \boxed{e_1 + e_2 + e_3 = 0} \tag{D.4}$$

But if we define the space vectors $\vec{E}_1 = \vec{u}_1\, e_1$, $\vec{E}_2 = \vec{u}_2\, e_2$ and $\vec{E}_3 = \vec{u}_3\, e_3$, we can compute their sum using Leblanc's and Ferraris's theorems and we get:

$$\boxed{\vec{E}_1 + \vec{E}_2 + \vec{E}_3 = \frac{3}{2}\overline{E}} \tag{D.5}$$

We can do this computation on the three winding current space vectors $\vec{i}_{\mathrm{U}}$, $\vec{i}_{\mathrm{V}}$ and $\vec{i}_{\mathrm{W}}$, as well as the respective produced field space vectors $\vec{B}_{\mathrm{U}}$, $\vec{B}_{\mathrm{V}}$ and $\vec{B}_{\mathrm{W}}$. Our Fresnel frame used in the above demonstration can be defined by the Real axis ($\Re$) coinciding with the winding U axis or stator X axis, and the Imaginary axis ($\Im$) coinciding with the stator Y axis (figure 2.19). The combination of the phase currents then results in a rotating current space vector $\vec{I}$ and the respective field $\vec{B}_I$ space vector aligned to it, as follows:

$$\vec{I}_{\mathrm{U}} + \vec{I}_{\mathrm{V}} + \vec{I}_{\mathrm{W}} = \frac{3}{2}\overline{I}, \quad \vec{B}_I \propto \vec{I}, \tag{D.6}$$

$$\text{with} \quad \overline{I} = \vec{u}_{\mathrm{U}} I \sqrt{2} e^{j\omega t} \tag{D.7}$$

The combined field space vector $\vec{B}_I$ interacts with the rotor magnetic moment $\vec{M}_j$ (D.2) creating a couple that can be computed as follows:

$$C_{em} = \vec{M}_j \times \vec{B}_I \propto I\,\sin\left(\vec{M}_j, \vec{B}_I\right) \tag{D.8}$$

Where I is the amplitude of the winding current in a single phase. So if the generated rotating field $\vec{B}_I$ is always in quadrature with the rotor direct axis, the torque is constant, maximal and depends only on the phase current amplitude.