Auton Robot
DOI 10.1007/s10514-012-9291-2

Force feedback exploiting tactile and proximal force/torque sensing

Theory and implementation on the humanoid robot iCub

Matteo Fumagalli · Serena Ivaldi · Marco Randazzo · Lorenzo Natale · Giorgio Metta · Giulio Sandini · Francesco Nori

Received: 12 July 2011 / Accepted: 17 March 2012

Abstract The paper addresses the problem of measuring whole-body dynamics for a multiple-branch kinematic chain in presence of unknown external wrenches. The main result of the paper is to give a methodology for computing whole body dynamics by aligning a model of the system dynamics with the measurements coming from the available sensors. Three primary sources of information are exploited: (1) embedded force/torque sensors, (2) embedded inertial sensors, (3) distributed tactile sensors (i.e. artificial skin). In order to cope with external wrenches applied at continuously changing locations, we model the kinematic chain with a graph which dynamically adapts to the contact locations. Classical pre-order and post-order traversals of this dynamically evolving graph allow computing whole-body dynamics and estimate external wrenches. Theoretical results have been implemented in an open-source software library (iDyn) released under the iCub project. Experimental results on the iCub humanoid robot show the effectiveness of the proposed approach.

Keywords Active force control · Proximal sensing · Multi-body dynamics

1 Introduction

Currents trends in robotics foster research in the development of capabilities and skills which can make robots autonomous and safe (i.e. not dangerous). The evolving scenario indeed requires the human and the robot to coexist within a shared unstructured environment, to interact and perform cooperative or independent tasks precisely and safely. Many research fields are addressing safety objectives from different perspectives: some examples are rescue robotics, home robotics, medical and rehabilitation robotics. Humanoid robotics shares as well the goal of guaranteeing safe interactions between robots and external agents, that being a human or the environment.

M. Fumagalli and S. Ivaldi equally contributed to this work.

M. Fumagalli · S. Ivaldi (✉) · M. Randazzo · L. Natale · G. Metta · G. Sandini · F. Nori
Department of Robotics Brain and Cognitive Sciences, Istituto Italiano di Tecnologia, Via Morego 30, 16163 Genova, Italy
e-mail: serena.ivaldi@iit.it

M. Fumagalli
e-mail: matteo.fumagalli@iit.it

M. Randazzo
e-mail: marco.randazzo@iit.it

L. Natale
e-mail: lorenzo.natale@iit.it

G. Metta
e-mail: giorgio.metta@iit.it

G. Sandini
e-mail: giulio.sandini@iit.it

F. Nori
e-mail: francesco.nori@iit.it

M. Fumagalli
Department of Control Engineering, University of Twente, Enschede, The Netherlands
e-mail: m.fumagalli@ewi.utwente.nl

S. Ivaldi
Institut des Systemes Intelligents et de Robotique, University Pierre et Marie Curie, Paris, France
e-mail: serena.ivaldi@isir.upmc.fr

G. Metta · G. Sandini
Department of Informatics, Systems and Communications (DIST), University of Genova, Genova, Italy

Published online: 04 April 2012

When deployed into a human environment, a robot can be considered safe if physical interaction is properly controlled, for example if: it does not damage itself, a human being or the environment; it is compliant to external perturbations and during interactions with the surroundings; it quickly reacts to unpredicted events.

A framework integrating all the aspects of human-robot-environment interaction, within the same physical space, must be adequately defined. To this purpose, all the possible sources of information (cameras, proximity sensors, proprioceptive sensors, etc.) should be included to improve the representation of the robot workspace, in order to fulfill its tasks while preserving safety. Uncertainties and noises may reduce the reliability of its perceptual representation, and lead to misbehaviors. In these cases, unsought and potentially dangerous contacts might occur between the human and the robot.

Clearly, during human-robot physical interaction, intrinsic compliance and force regulation must be addressed. An interesting analysis of the effects of possible impacts of robotic manipulators on humans can be found in Haddadin et al. (2008a, 2008b). The lack of compliance has been traditionally compensated by collision avoidance solutions, where commonly the end- effector trajectory or the manipulator configuration is changed during motion so as to avoid collisions with the surroundings or the self (Minguez et al. 2008; Kulic and Croft 2007; Sisbot et al. 2010). To address the safety issue, physical Human-Robot Interaction (*pHRI*) (Santis et al. 2008) and related disciplines in actuator technologies are focusing on the developments of *passive compliance* actuation. These novel systems are characterized by the intrinsic capacity to reduce the risk of hurting and damaging the surroundings: variable impedance actuators (Eiberger et al. 2010), series elastic actuators (Pratt and Williamson 1995), pneumatics and hydraulics actuators, etc. Nevertheless, recent studies have shown that certain solutions present an undesired effect that may lead to unsafe behaviors. Elastic elements, especially if combined with actuators, can store great amounts of potential energy which, once released, can be extremely unsafe, as recently shown in Haddadin et al. (2010a).

Conversely, when *active compliance* is employed, the robot behavior is grounded on the sensory system: joint torque sensors rather than force sensors (the latter typically placed at the end-effector), are required. Exploiting the sensors information, the robot can carry out force regulation, react safely to contacts and also take decisions about the tasks (Siciliano and Villani 1996; Luca 2006; Haddadin et al. 2010b; Mistry et al. 2010; Calinon et al. 2010; Fumagalli et al. 2010b). However, retrieving measurements using localized torque sensors might not allow a full perceptual representation of the interaction scenario, in terms of forces and torques which rise over the whole structure. Even if torque sensors are distributed over the entire structure of the robot and can measure the internal dynamic as well as the interactions occurring on its links, they can only measure the single component of momentum which works along the axis of rotation of the joints. More specifically, since forces and torques are linearly related by the transposed Jacobian, which might have a non-empty null-space, there exist singular configurations where some interaction forces at sensory level cannot be fully retrieved (e.g. pure forces working on a direction which is parallel to the joint torque sensor axis are *hidden*). A more robust and complete representation of the interaction forces can be retrieved by Force/Torque Sensors (FTS). Classically, robots are equipped with six-axis FTS mounted on their end-effectors, where the most interaction with the environment occurs during manipulation (Sciavicco and Siciliano 2005).

This solution is compact and less invasive with respect to the design of joint torque sensors (Parmiggiani et al. 2009; Luh et al. 1983): it allows a complete representation of the interaction forces and thus a better achievement of active compliance (Caccavale et al. 2005; Chiaverini et al. 1999; Siciliano and Villani 2000). However, this information is localized at the tool level. In other words, a FTS at the end-effector does not allow retrieving neither the information about the manipulator dynamics, nor about the potential interaction occurring on any link of the robotic system. In this situation, this information must be retrieved with other sensors.

With the solution addressed in this paper, a more complete and better representation of the interaction forces over the entire structure of the robot can be obtained. The proposed approach makes use of three sets of sensors, distributed along the kinematic chain: *force/torque*, *inertial* and *tactile* sensors. The focus of the paper is on understanding how to integrate the measurements from these sensors to estimate both internal and externally applied wrenches (i.e. forces and torques). Specifically, the questions addressed in the paper will be the following: given a multiple branch kinematic chain with embedded (force/torque, tactile, inertial) sensors, assuming that a certain number of unknown external wrenches act on the system and assuming that a dynamical model of the system is available:

- is there a systematic procedure to propagate force/torque measurements along the chain in order to estimate internal wrenches?[1] How does this procedure change depending on the location of the externally applied wrenches?
- is there a systematic procedure to measure the external wrenches acting on the chain? How many external

[1] Internal wrenches are of particular interest because their projection on the joints can be used to estimate joint torques (Sciavicco and Siciliano 2005).

wrenches can be estimated? Is there a condition on the localization of these external wrenches (with respect to the FTS locations) to guarantee that they can be estimated?

To the best of our knowledge, a limited number of previous works have considered similar issues. Methods for estimating internal wrenches from FTS embedded in the links have been proposed in Morel and Dubowsky (1996), Morel et al. (2000) where a single FTS placed at the base of a 3 Degrees Of Freedom (DOF) PUMA manipulator was used to estimate the joint torques. Here, the proposed method is generalized to the estimation of both internal and external wrenches on 32 of the 53 DOF of the humanoid robot iCub (Metta et al. 2010). Moreover, we enrich the estimation by also computing external wrenches at arbitrary contact locations. We remark that these locations can be either fixed a priori for particular robot tasks or updated on-the-fly. Here, contact locations are provided by a set of distributed capacitive tactile sensors, constituting a sort of "artificial skin" (Cannata et al. 2008; Maggiali et al. 2008).

We also point out that the proposed approach is particularly convenient from an economic point of view, if active compliance must be enabled on existing robots actuated by electric motors. For example in the iCub, passive solutions were not considered at design stage, and the introduction of joint-level torque sensing in the whole robot would have required a complete re-design of the mechatronics. Conversely, since FT sensors are compact and relatively small, embedding the fours on the platform has been quite easy (no radical changes were made to the structure), not to mention conveniently cheap. However, even if the makeover of the robot structure to include joint torque sensors is more expensive in terms of time and resources, the next versions of the iCub will adopt both solutions, to enhance the perceptual capabilities of the robot and allow different control modalities. A study of the new shoulder equipped with joint torque sensors has been presented in Parmiggiani et al. (2009).

The paper is organized as follows. Section 2 introduces the notation of the variables required for computing the internal and external forces and torques[2] on a given link. The method of the Enhanced Oriented Graph (EOG) is presented in Sect. 3. It will be shown that the method can be applied to both single and multiple branched open kinematic chains with embedded FTS, where the concept of sub-link is also introduced. Section 4 discusses the computations when the *Recursive Newton-Euler Algorithm* (RNEA) (Sciavicco and Siciliano 2005) is employed. It must be pointed out that the RNEA is here proposed as a tool to compute kinematic and dynamic information recursively, but its adoption is not a must and other algorithms could be used (e.g. the Articulated Body Algorithm (ABA), see Featherstone and Orin 2008 for a review). The proposed method is thus shown to be generic and applicable to every open kinematic tree. Force sensors are used to improve the estimation of the internal wrenches and for the computation of external interaction. The number of FTS to employ and their placement along a manipulation structure is not defined. Nevertheless, as it will be clear later on, we suggest spreading them on different links of the system to increase the quality and reliability of the results. The paper indeed presents a systematic procedure for computing $N+1$ external wrenches from N internal wrenches (i.e. measurements from FTS). Remarkably, under some conditions that will be discussed in next sections, not only all link wrenches and joint torques can be computed, but also a certain set of external wrenches applied to the robotic chains. The technique is built on initial work presented in Ivaldi et al. (2011): here, we show further results and provide a more extensive discussion of the proposed techniques. Section 6 presents experimental results obtained on the iCub platform. Experiments were performed considering different interaction scenarios, as a proof of the validity of the approach.

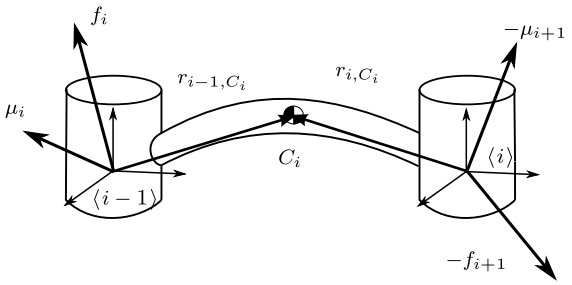


Fig. 1 Notation for the i-th link of a kinematic chain. A more complete description can be found in Sciavicco and Siciliano (2005)

2 Notations

We present here the notations that have been used to generally describe the kinematics and the dynamics of a link (see Fig. 1). For the kinematic we adopted the Denavit-Hartenberg notation, for the dynamics the RNEA.[3] Moreover, we limit the discussion to revolute joints.[4] Here is a list of the adopted symbols:

$\langle \cdot \rangle$ generic Cartesian reference frame

[2]Given a force $f \in \mathbb{R}^3$ and a moment $\mu \in \mathbb{R}^3$, a wrench $w \in \mathbb{R}^6$ is the vector $w = \begin{pmatrix} f \\ \mu \end{pmatrix}$. Force/torque sensors, which actually measure a wrench, are named according to the physics terminology, where μ is called torque. In this paper, to discriminate from the joint torque τ, we call μ moment according to the mechanical terminology.

[3]The method we will present in next sections is not strictly dependent on the notation and algorithms employed for the computations. Custom choices can also be adopted.

[4]The method can be easily generalized for revolutionary and linear joints.

v_a is a generic n-dimensional vector $v \in \mathbb{R}^n$ expressed in $\langle a \rangle$
$r_{a,b}$ distance vector r from $\langle a \rangle$ to $\langle b \rangle$
z_i z-axis of $\langle i \rangle$, aligned with the axis of rotation of joint i
θ_i the angle associated to the i-th joint
$\ddot{p}_i \in \mathbb{R}^3$ denotes the linear acceleration of $\langle i \rangle$
$\omega_i, \dot{\omega}_i \in \mathbb{R}^3$ the angular velocity and acceleration of $\langle i \rangle$
m_i mass associated with the i-th link
$\bar{I}_i^i \in \mathbb{R}^{3\times3}$ represents the inertia tensor of the i-th link, defined with respect to the center of mass oriented as the frame $\langle i \rangle$
$C_i \in \mathbb{R}^3$ the coordinate vector of the center of mass of the i-th link with respect to $\langle i \rangle$
$f_i \in \mathbb{R}^3$ represents the forces applied on $\langle i \rangle$, that the $i+1$-th link exerts on the i-th link
$\mu_i \in \mathbb{R}^3$ represents the moment applied on $\langle i \rangle$, that the $i+1$-th link exert on the i-th link
$\tau_i \in \mathbb{R}$ is the joint torque, i.e. the component of μ_i along z_i
$w_i \in \mathbb{R}^3$ is the wrench $w = \binom{f}{\mu}$ applied on $\langle i \rangle$, that the $i+1$-th link exerts on the i-th link

3 Enhanced Oriented Graph, applied to kinematics and dynamics of robots

Graph theory has been extensively used to represent mechanical systems (Wittenburg 1994; Featherstone 2007) and kinematic chains, producing compact and clear models, in matrix forms with beneficial properties (e.g. branch-induced sparsity, Featherstone 2010) when the connectivity among its elements is expressed. There is not a unique choice for a graph representing a chain: for example, in Featherstone (2007) graphs are undirected, nodes and arcs represent bodies and joints respectively; the resulting graph is undirected (i.e. non-oriented), but nodes are "labeled" according to a "regular numbering scheme".

Hereinafter we present the theoretical framework of the *Enhanced Oriented Graphs* (EOG), applied to the computation of both internal and external wrenches for single and multiple branches, generally non-grounded, kinematic chains. The proposed method is an extension of the classical RNEA (Featherstone and Orin 2008; Sciavicco and Siciliano 2005). Similarly to the classical approach we represent a kinematic chain as a graph such that computations of the system dynamics can be obtained performing a pre-order and a post-order traversal visit of the graph itself. However, we enhance the graph with specific nodes representing both known and unknown (kinematic or dynamic) variables. Remarkably, not all the unknowns will be specified a-priori (e.g. contacts at arbitrary locations might appear and other contacts might be removed) and therefore the graph structure will be adapted accordingly.[5] The dynamically evolving graphical description of the chain modifies the way the graph is visited during the Newton-Euler recursion, thus changing in particular the direction along which the recursion is propagated in the graph. In order to cope with this evolving representation we introduced another difference with respect to classical RNEA graphical representations by defining the kinematic chain as an *oriented* graph: the direction along which edges are traversed will determine either the use of the classical Newton-Euler recursion formula or a slightly modified version of it.

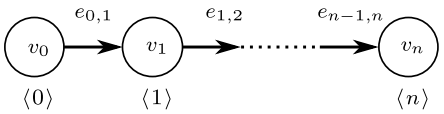


Fig. 2 An open chain represented as a graph

3.1 The enhanced graph representation

We consider an open (single or multiple branches) kinematic chain with n DOF composed of $n+1$ links (see Fig. 2). The i-th link of the chain is represented by a vertex v_i (sometimes called node). A hinge joint between the link i and the link j (i.e. a rotational joint) is represented by an oriented edge $e_{i,j}$ connecting v_i with v_j. The orientation of the edge can be either chosen arbitrarily (it will be clear later on that the orientation simply induces a convention) or it can follow from the exploration of the kinematic tree according to the "regular numbering scheme" (Featherstone and Orin 2008), which induces a parent/child relationship such that each node has an unique input edge and multiple output edges. Following the classical RNEA and the classical Denavit-Hartenberg notation, we assume that each joint has an associated reference frame with the z-axis aligned with the rotation axis; this frame will be denoted $\langle e_{i,j} \rangle$. In kinematics, an edge $e_{i,j}$ from v_i to v_j represents the fact that $\langle e_{i,j} \rangle$ is fixed in the i-th link. In dynamics, $e_{i,j}$ represents the fact that the dynamic equations will compute (and make use of) $w_{i,j}$, i.e. the wrench that the i-th link exerts on the j-th link, and not the equal and opposite reaction $-w_{i,j}$, i.e. the wrench that the j-th link exerts on the i-th link (further details in Sect. 4).

In order to simplify the computations of the inverse dynamics on the graph (see Sect. 4), kinematic and dynamic measurements are explicitly represented. Specifically, the

[5] Within this context, a crucial role is played by the distributed tactile sensor, primarily used to compute the presence and the location of externally applied wrenches. Even if the tactile sensor would be capable of measuring also the component of the force normal to the skin surface, this information is not used in this paper where we focus on computing both the applied force and torque (i.e. the whole externally applied wrench) exploiting the embedded sensors.

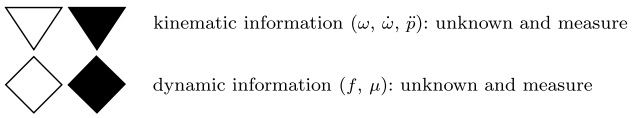


Fig. 3 The notation introduced to represent unknown and known (i.e. measured) variables in graphs

graph representation is enhanced with a new set of graphical symbols: a triangle to represent kinematic quantities (i.e. velocities and acceleration of links), and a rhombus for wrenches (i.e. force sensors measurements on a link), as shown in Fig. 3. Moreover, a color code groups these symbols into *known* to represent sensors measurements, and *unknown* to indicate the quantities to be computed.

Kinematic variables can in general be measured by means of gyroscopes, accelerometers, or simply inertial sensors. When attached on link i-th, these sensors provide angular and linear velocities and accelerations (ω, $\dot{\omega}$, $\dot{p}$ and $\ddot{p}$) at the specific location where the sensor is located. We can represent these measurements in the graph with a *black triangle* (▼) and an additional edge from the proper link where the sensor is attached to.[6] As usual, the edge has an associated reference frame, in this case corresponding to the reference frame of the sensor. Similarly, an unknown kinematic variable is represented with a *white triangle* (▽) with an associated edge going from the link (where the unknown kinematic variable is attached) to the triangle. The reference frame associated to the edge will determine the characteristics of the retrieved unknown kinematic variables, as it will be described in Sect. 4.

Analogously for dynamic variables, we introduce two new types of nodes with a rhomboidal shape (see Fig. 3): *black rhombi* (⧫) to represent known (i.e. measured) wrenches, *white rhombi* (◊) to represent unknown wrenches which need to be computed. The reference frame associated to the edge is the location of the applied or unknown wrench.

Remark 1 There is not a fixed rule to determine the orientation of the edge connecting the rhombi to the graph: according to our convention for representing the wrenches, the edge can be either directed from the rhombus to the link or vice versa depending on the variable we are interested in representing (i.e. the wrench from the link to the external environment or the equal and opposite wrench from the environment to the link).

It is important to point out that, whereas the position of ⧫ is static within the graph (because sensors are fixed in the manipulator), the location of ◊ can be dynamic (contact point locations are dynamically detected by the distributed tactile sensor). If a contact moves along a chain, the graph is accordingly modified. This rule shows a big benefit of the EOG, which dynamically adapts in response to the location of the unknown external wrenches.

[6] According to our kinematic convention an edge $e_{i,j}$ is fixed on the i-th link. Therefore a sensor fixed in the i-th link, will be represented by $e_{i,s}$, i.e. an edge from the link to the sensor.

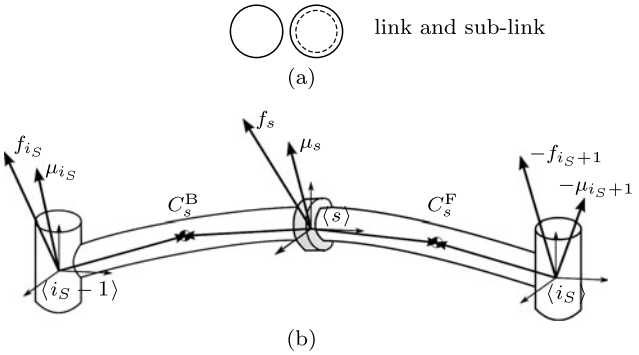


Fig. 4 (**a**) The notation introduced to represent node (vertex, link) and sub-node (sub-link). (**b**) A representation of a FTS within the i_S-th link. Note that the sensor divides the link into two sub-links, each with its own dynamical properties. In particular, it is evident that the center of mass (COM) of the original link, C_{i_S}, differs from C_s^{F}, C_s^{B}, i.e. the COM of the two "sub-links"

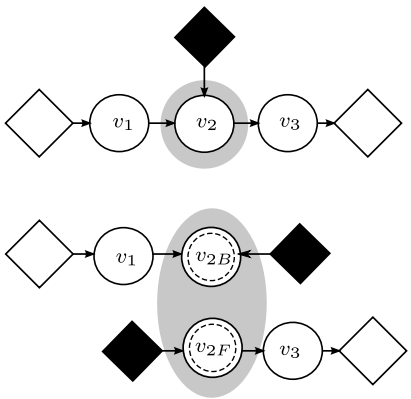


Fig. 5 The graph shows how to insert a FTS in a graph representation of a kinematic chain. The node where the sensor is attached to (*highlight*), is practically divided into two sub-nodes. The graph is divided into two sub-graphs and *two black rhombi* (known wrenches corresponding to the sensor measurement) are connected to the sub-nodes

Within this representation, embedded FTS can be inserted by "cutting" the manipulator chain where the FTS is located and creating two virtual "sub-links" from the link hosting physically the sensor. The EOG is then split into two sub-graphs, where black rhombi (⧫, i.e. known wrenches representing the FTS measures, one per graph) are introduced and attached to the sub-links. In practice, suppose that a FTS is placed in the i_S-th link (see Fig. 4(b)). Let $\langle s \rangle$ be the frame associated to the sensor. The sensor virtually divides link i_S into two "sub-links" and therefore measures the wrench exchanged between the so-called "forward" and

the "backward" sub-links. This wrench will be represented by two rhomboidal nodes, attached to the two sub-links (see Fig. 5). Under these considerations, the FTS within a link is represented by splitting the node associated to the link into two sub-nodes, with suitable dynamical properties. Two known wrenches in the form of black rhombi are then attached to the sub-nodes, with suitable edges whose associated reference frame is $\langle s \rangle$ for both edges.

4 Exploiting the RNEA for EOG

The graphical representation proposed in the previous section can be used to compute the internal dynamics of a (floating) kinematic chain provided with sufficient tactile, force and inertial sensors. In particular, in this section we describe how to compute both kinematic and dynamic variables, associated to the edges of the graphical representation.

A first recursion on the graph (pre-order traversal) computes the linear acceleration ($\ddot{p}$) and the angular velocity and acceleration (ω, $\dot{\omega}$) for each of the reference frames associated to the edges of the graph. This procedure practically allows propagating the information coming from a single inertial sensor to the entire kinematic chain. At each step, the values of ($\ddot{p}$, ω, $\dot{\omega}$) for a given link are propagated to the neighbor links by exploiting the encoder measurements and the kinematic model of the chain.

A second recursion (post-order traversal) computes all the (internal and external) wrenches acting on the chain at the reference frames associated with all the edges in the graph. In this case, Newton-Euler equations are exploited to propagate force information along the chain. At each step, all but one wrench acting on a link are assumed to be known and the remaining unknown wrench is computed exploiting a dynamic model of the link and the output from the kinematic recursion.

4.1 Kinematics

In the following, we describe the basic equations for propagating the kinematic information within the graph. The proposed notation might seem fairly too general, especially if compared with the classical computations where the major simplification is the assumption that kinematics variables are propagated across the kinematic tree along a constant path. In our case instead, we are interested in a formulation capable of exploiting multiple (dynamically inserted) sensors to propagate the kinematic information from the sensors to the surrounding links. Therefore the flow of kinematics cannot be predefined but needs to be dynamically adapted to the current structure of the EOG.

The basic step here described consists in propagating the kinematic information associated to an edge connected to a node v to all the other edges connected to it. As usual, for each edge i we consider the associated reference frame $\langle i \rangle$. Referring to Figs. 6(a)–(c) we assume that knowing the linear acceleration ($\ddot{p}_j$) and the angular velocity and acceleration ($\omega_j, \dot{\omega}_j$) of the reference frame $\langle j \rangle$ we want to compute the same quantities for the frame $\langle i \rangle$ sharing with $\langle j \rangle$ a common node v. Figure 6(a) represents the case where the edge i exits v but the edge j enters v; recalling the kinematic meaning of the edge directions, the sketch in Fig. 6(a) represents a situation where $\langle i \rangle$ is attached to v while $\langle j \rangle$ is rotated by the joint angle θ_j around z_j. The situation is exactly the one we have in the classical Denavit-Hartenberg forward kinematic description and therefore we have (Sciavicco and Siciliano 2005):[7]

$$\begin{aligned} \omega_i &= \omega_j + \dot{\theta}_j z_j, \\ \dot{\omega}_i &= \dot{\omega}_j + \ddot{\theta}_j z_j + \dot{\theta}_j \omega_j \times z_j, \\ \ddot{p}_i &= \ddot{p}_j + \dot{\omega}_i \times r_{j,i} + \omega_i \times (\omega_i \times r_{j,i}), \end{aligned} \tag{1}$$

where z_j and θ_j indicate the rotational axis and the angular position of the joint associated to the edge j. Similarly,

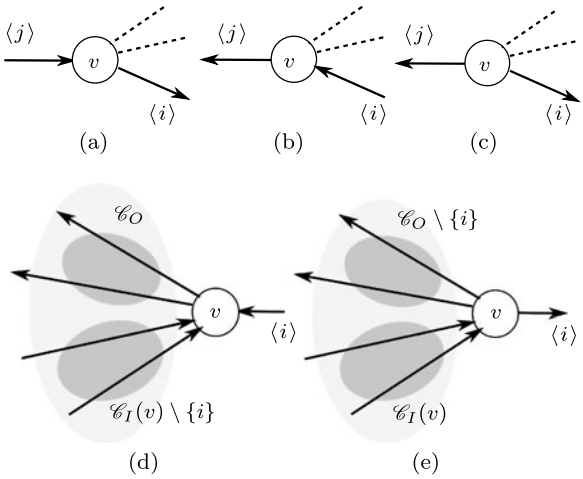


Fig. 6 (**a**)–(**c**) The three cases accounting for the exchange of kinematic information. (**d**)–(**e**) The two cases accounting for the exchange of dynamic information

[7] In the classical recursive kinematic computation (Sciavicco and Siciliano 2005) there is a one-to-one correspondence between links and joints (see Fig. 1) thus resulting in a set of kinematic equations slightly different from the ones of Eq. (1). Classically, the i-th link has two joints and associated reference frames $\langle i \rangle$ and $\langle i-1 \rangle$, respectively. Only $\langle i \rangle$ is attached to the i-th link while $\langle i-1 \rangle$ is attached to the link $i-1$. The rotation between these two links is around the z-axis of $\langle i-1 \rangle$ by an angle which is denoted by θ_i and therefore the analogous of Eq. (1) in Sciavicco and Siciliano (2005) refer to $\dot{\theta}_i$ in place of $\dot{\theta}_j$ and z_{i-1} in place of z_i. In our notation, we get rid of this common labeling for joints and links by explicitly distinguishing the link represented with the node v and the attached joints represented with the edges $i, j, \ldots$ and associated frames $\langle i \rangle, \langle j \rangle, \ldots$ whose axes are therefore $z_i, z_j, \ldots$ with associated angles θ_i, θ_j.

Fig. 6(b) represents the case where the edge i enters v but the edge j exits the node; therefore Fig. 6(b) represents a situation where $\langle j \rangle$ is attached to v while $\langle i \rangle$ is rotated by the joint angle θ_i. The situation is exactly the opposite encountered in classical Denavit-Hartenberg so that we have:

$$\begin{aligned} \omega_i &= \omega_j - \dot{\theta}_i z_i, \\ \dot{\omega}_i &= \dot{\omega}_j - \ddot{\theta}_i z_i - \dot{\theta}_i \omega_j \times z_i, \\ \ddot{p}_i &= \ddot{p}_j - \dot{\omega}_j \times r_{i,j} - \omega_j \times (\omega_j \times r_{i,j}). \end{aligned} \tag{2}$$

Finally, Fig. 6(c) represents the case where both $\langle i \rangle$ and $\langle j \rangle$ are attached to the link represented by v. In this case, continuity formulas are obtained putting $\dot{\theta}_i = 0$ and $\ddot{\theta}_i = 0$ in Eq. (1) (or equivalently Eq. (2)):

$$\begin{aligned} \omega_i &= \omega_j, \\ \dot{\omega}_i &= \dot{\omega}_j, \\ \ddot{p}_i &= \ddot{p}_j + \dot{\omega}_i \times r_{j,i} + \omega_i \times (\omega_i \times r_{j,i}). \end{aligned} \tag{3}$$

These rules can be used to propagate kinematic information across different edges connected to the same node. The only situation which cannot be solved is the one where all edges enter the node v, i.e. none of the associated reference frames is fixed to the link v. We can handle these cases a posteriori by defining a new arbitrary reference frame $\langle v \rangle$ attached to the link. In our formalism, this is achieved by adding a kinematic unknown ($\triangledown$) and an edge from v to $\triangledown$ with associated frame $\langle v \rangle$.

Remark 2 If the edge directions are chosen according to a "regular numbering scheme" as proposed in Sect. 3.1, each edge will have a unique ingoing edge and multiple outgoing edges.

The only nodes with no outgoing edges will be the ones corresponding to the leaves of the kinematic tree (typically the end-effectors). For these nodes, we will add a kinematic unknown ($\triangledown$) and an edge from v to $\triangledown$ with associated frame $\langle v \rangle$ (typically the end-effector reference frame of the classical Denavit-Hartenberg notation).

4.2 Dynamics

We here describe the basic equations for propagating the dynamic information within the graph. As seen before, the flow of dynamical information cannot be predefined because the graph structure continuously changes according to the position of the applied external wrenches, detected by the distributed tactile sensor. The basic step proposed in this section assumes that all the wrenches acting on a link but one are known. The remaining unknown wrench is computed by the Newton-Euler equations. Using the graph representation, a node v with all its edges represents a link with all its joints. As proposed in Sect. 3, at each edge $e_{u,v}$, we can associate the wrench $w_{e_{u,v}}$ that u exerts on v. At each edge $e_{v,u}$ we can associate the wrench $w_{e_{v,u}}$ that v exerts on u. The Newton-Euler equations for the link v can therefore be written as follows (Sciavicco and Siciliano 2005):

$$\begin{aligned} &\sum_{e_I \in \mathcal{C}_I(v)} f_{e_I} - \sum_{e_O \in \mathcal{C}_O(v)} f_{e_O} = m_v \ddot{p}_{C_v}, \\ &\sum_{e_I \in \mathcal{C}_I(v)} (\mu_{e_I} + f_{e_I} \times r_{e_I,C_v}) \\ &\quad - \sum_{e_O \in \mathcal{C}_O(v)} (\mu_{e_O} + f_{e_O} \times r_{e_O,C_v}) \\ &= \bar{I}_i \dot{\omega}_i + \omega_i \times (\bar{I}_i \omega_i), \end{aligned} \tag{4}$$

where:[8]

$$\ddot{p}_{C_v} = \ddot{p}_i + \dot{\omega}_i \times r_{i,C_v} + \omega_i \times (\omega_i \times r_{i,C_v}), \tag{5}$$

and where $\mathcal{C}_I(v)$ is the set of ingoing edges, $\mathcal{C}_O(v)$ is the set of outgoing edges and where the index i refers to any edge in $\mathcal{C}_O(v)$ (necessarily non-empty in consideration of what we discussed in Sect. 4.1). In other terms, recalling the kinematic meaning of outgoing edges, i is an edge associated with any of the arbitrary reference frames $\langle i \rangle$ fixed with respect to the link v. As anticipated, Eq. (4) can be used to propagate the dynamic information across the graph. Assuming that all but one wrench acting on a link are known, the remaining unknown wrench can be computed with Eq. (4). Let us denote with i the edge associated with the unknown wrench. If $i \in \mathcal{C}_I(v)$, then the situation is the one represented in Fig. 6(d) and we have:

$$\begin{aligned} f_i &= -\sum_{\substack{e_I \in \mathcal{C}_I(v) \\ e_I \neq i}} f_{e_I} + \sum_{e_O \in \mathcal{C}_O(v)} f_{e_O} + m_v \ddot{p}_{C_v}, \\ \mu_i &= -f_i \times r_{i,C_v} - \sum_{\substack{e_I \in \mathcal{C}_I(v) \\ e_I \neq i}} (\mu_{e_I} + f_{e_I} \times r_{e_I,C_v}) \\ &\quad + \sum_{e_O \in \mathcal{C}_O(v)} (\mu_{e_O} + f_{e_O} \times r_{e_O,C_v}) \\ &\quad + \bar{I}_i \dot{\omega}_i + \omega_i \times (\bar{I}_i \omega_i). \end{aligned} \tag{6}$$

If $i \in \mathcal{C}_O(v)$, then the situation is the one represented in Fig. 6(e) and we have:

[8] With slight abuse of notation we indicated with $r_{\star,C_v}$ the vector connecting the generic frame $\langle \star \rangle$ to the one placed on the center of mass C_v of the v-th link.

$$
\begin{aligned}
f_i &= \sum_{e_I \in \mathcal{C}_I(v)} f_{e_I} - \sum_{\substack{e_O \in \mathcal{C}_O(v) \\ e_O \neq i}} f_{e_O} - m_v \ddot{p}_{C_v}, \\
\mu_i &= -f_i \times r_{i,C_v} + \sum_{e_I \in \mathcal{C}_I(v)} (\mu_{e_I} + f_{e_I} \times r_{e_I,C_v}) \\
&\quad - \sum_{\substack{e_O \in \mathcal{C}_O(v) \\ e_O \neq i}} (\mu_{e_O} + f_{e_O} \times r_{e_O,C_v}) \\
&\quad - \bar{I}_i \dot{\omega}_i - \omega_i \times (\bar{I}_i \omega_i).
\end{aligned} \tag{7}
$$

Remark 3 With reference to Eqs. (6)–(7), it must be noted that if only one edge is connected to the generic node v, then $\mathcal{C}_I(v) \cup \mathcal{C}_O(v) = \{i\}$. Hence, the sums $\sum f_k$, $\sum(\mu_k + f_k \times r_{k,C_v})$ (where k is the generic index for the edge) do not contribute to the computations of f_i and μ_i which, in this particular case, correspond to the inertial components only. This case is peculiar, and its significance will be clear later on when the solution of the EOG is discussed in detail.

5 Building EOG for computing dynamics and external wrenches

In Sects. 4.1 and 4.2 we presented the basic steps for propagating kinematic and dynamic information across a graph representing a kinematic tree. In this section we describe how to use these basic steps to compute the whole-body dynamics, with specific attention at getting estimates for the externally applied wrenches (denoted with ◊). During these computations the graph structure is assumed static but it might change from one computation to the next. Initially, the graph structure needs to be defined.

1. Create the graph representing the kinematic tree; define a node for each link and an edge for each joint connecting two links. The edge orientation is arbitrary and in particular it can be defined according to a "regular numbering scheme".
2. For each inertial sensor[9] (measuring the linear acceleration and the angular velocity and acceleration) insert a *black triangle* (▼) and an edge from the node v to the triangle, where v represents the link to which the sensor is attached. Associate to the edge the reference frame $\langle s \rangle$ corresponding to the sensor frame.
3. For any node v with only ingoing edges, add a *white triangle* (▽) and an edge from v to the triangle. Associate to the edge an arbitrary reference frame $\langle v \rangle$.[10]

These steps define the kinematic EOG which can be used to compute the kinematics of the entire chain. Specifically, if this graph contains a single inertial sensor (represented by a ▼ node), the associated measurements can be used to compute the linear acceleration and angular acceleration and velocity for all the edges of the graph. Computations can be performed following the procedure in Algorithm 1, that is a *pre-order*[11] traversal of the tree with elementary operations defined by Eq. (1), Eq. (2) or Eq. (3). If multiple ▼ nodes (i.e. inertial sensors) are present in the graph, each path between two of these nodes corresponds to a set of three equations containing the measurements: one for the linear accelerations, one for the angular velocity and one for the angular accelerations. These equations can be used to refine the sensor measurements or to give better estimates of the joint velocities and accelerations (typically derived numerically from the encoders and therefore often noisy).

Remark 4 In this respect, a possible algorithm for computing the better estimate of the kinematics, given the multiple sources, is briefly reported in Algorithm 2. Basically, given a set of K kinematics sources ▼, which for brevity we name $\kappa_1, \ldots, \kappa_K$, Algorithm 1 is solved K times. At each time k, κ_k is the only kinematic source which is not being removed from the EOG, and then the only ▼ in the graph. The solution of the EOG K times yields a set of conditional estimates $\omega_{j|\kappa_1}, \ldots, \omega_{j|\kappa_K}$, $\forall j$ (analogous considerations hold for $\dot{\omega}$ and $\ddot{p}$), which can be used by classical filters to provide the better estimate (e.g. maximum likelihood filters, Kalman filters etc.). The analysis and evaluation of the possible filters is outside the scope of the paper, but the interested reader could refer for example to Mutambara (1998).

A clarifying example is shown in Fig. 7(a): notice that the visit order is not related to the edge direction, since the latter only affects the recursive equations that must be used to propagate the variables. Once velocities and accelerations have been computed for all edges, a new series of steps needs to be performed on the EOG to obtain the dynamic enhanced sub-graphs.

4. For each FTS embedded in the link v, cut the graph into two sub-graphs as shown in Fig. 5. Divide v into two nodes v_B and v_F representing the sub-links (with suitable dynamic properties); define two *black rhombi* (♦)

[9] Kinematic chains are often grounded and therefore there exists a base link with null angular kinematics, $\omega = [0,0,0]^\top$, $\dot{\omega} = [0,0,0]^\top$ and gravitational linear acceleration $\ddot{p} = g$, being g the vector representing the gravity force. This situation is mathematically equivalent to an inertial sensor attached to the base link and measuring constantly $\omega = 0$, $\dot{\omega} = 0$ and $\ddot{p} = g$.

[10] See also Remark 2.

[11] *pre-* and *post-order* refer to different classical graph visiting algorithms (Cormen et al. 2002).

Algorithm 1 Solution of kinematic EOG exploiting a tree

Require: EOG, $\omega_0, \dot{\omega}_0, \ddot{p}_0$
Ensure: $\omega_i, \dot{\omega}_i, \ddot{p}_i, \forall v_i$
1: Attach a node ▼ for every kinematic source (e.g. inertial sensor)
2: Set $\omega_0, \dot{\omega}_0, \ddot{p}_0$ in ▼
3: Re-arrange the graph with a ▼ as the root of a tree
4: *KinVisit*(EOG,v_{root})

KinVisit(EOG,v_i)
1: Compute $\omega_i, \dot{\omega}_i, \ddot{p}_i$ with Eq. (1) or (2) or (3) according to direction of the edges i, j connected to v
2: **for all** child v_k of v_i **do**
3: *KinVisit*(EOG,v_k)
4: **end for**

Algorithm 2 Fusion of multiple kinematic sources

Require: EOG, $\kappa_k = [\omega_k, \dot{\omega}_k, \ddot{p}_k], k = 1, \ldots, K$
Ensure: $\hat{\omega}_i, \hat{\dot{\omega}}_i, \hat{\ddot{p}}_i, \forall i$
1: **for all** $k = 1 : K$ **do**
2: Attach a node ▼ for κ_k
3: Compute $\omega_{i|\kappa_k}, \forall i$
4: **end for**
5: Compute $\hat{\omega}_i$ = filter* $(\omega_{i|\kappa_1}, \ldots, \omega_{i|\kappa_K})$, analogously compute $\hat{\dot{\omega}}_i, \hat{\ddot{p}}_i$

* filter is a generic filter for data fusion from multiple sensors

and add two edges from the rhombi to the nodes. Associate to both the edges the same reference frame $\langle s \rangle$ corresponding to the sensor frame.

5. If there are other known wrenches acting on a link (e.g. sensors attached at the end-effector), insert a *black rhombus* (⧫) and an edge from the rhombus to v, where v represents the link to which the wrench is applied. Associate to the edge the reference frame $\langle s \rangle$ corresponding point where the external wrench is applied.
6. If the distributed tactile sensor is detecting externally applied wrenches, insert a *white rhombus* (◊) for each externally applied unknown wrench. Add an edge connecting the rhombus with v, where v represents the link to which the wrench is applied. The edge orientation is arbitrary depending on the wrench to be computed (i.e. the wrench from the link to the external environment or the equal and opposite wrench from the environment to the link). Associate to the edge the reference frame $\langle c \rangle$ corresponding to the location where the external wrench is applied.

After these steps have been performed, we basically obtained the dynamic enhanced sub-graphs, each of which can be considered independently.

Wrenches can be propagated to *all* the unknown nodes (◊) if and only *if there exists at maximum an unknown for each sub-graph*. If a sub-graph contains no unknown, we can always define it arbitrarily. Then for each sub-graph we can define a tree with the node ◊ as root. Wrenches can be propagated from the leaves to the root following the procedure in Algorithm 3, which is basically a *post-order* traversal of a tree (Cormen et al. 2002) with elementary operations defined by Eq. (6) or Eq. (7). Remarkably, when there is exactly one ◊ per sub-graph, each edge in the sub-graph is visited during the *post-order* traversal. As a result, all internal wrenches are computed and therefore a complete characterization of the whole-body dynamics is retrieved. In this particular case, given N FT sensors distributed on a chain, $N + 1$ sub-graphs are produced and therefore a maximum of $N + 1$ external wrenches can be estimated (one for each sub-graph). If there is no ◊ node in a sub-graph (i.e. no external forces are acting on the sub-graph), then the *post-order* traversal of this graph produces two equations (one for forces and the other for wrenches) with no unknowns.[12] These equations can be used to estimate on-line the dynamical parameters of the corresponding kinematic sub-tree exploiting the linearity of these parameters in the equations (Sciavicco and Siciliano 2005).

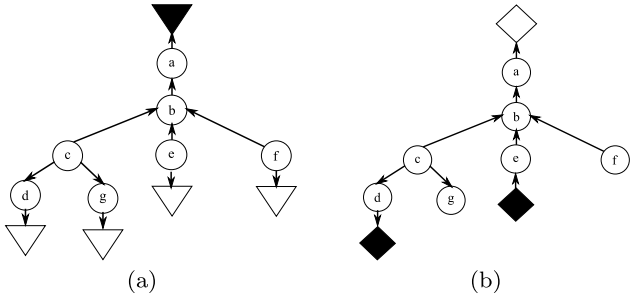


Fig. 7 (**a**) An example of kinematic EOG with multiple branches. Starting from the root ▼, the propagation of kinematics information follows the pre-order traversal of the tree. Thus, the order of "visiting" nodes is: a, b, c, d, g, e, f. (**b**) An example of dynamic EOG with multiple branches. The propagation of dynamics information follows the post-order traversal of the tree: starting from leaves, information is propagated from children to parents, until the root ◊. Thus, the order of "visiting" nodes is: d, g, c, e, f, b, a. It must be noted that leaves are not necessarily ⧫, as explained in Remark 3

If a sub-graph contains more than a single unknown node (◊), then wrenches can be propagated only to certain parts of the graph. These computations can be performed by choosing any of the unknown nodes as root and following a slightly modified procedure of Algorithm 3. The only additional condition consists of labeling a node as "not computable" whenever it is directly connected to an unknown node; this condition is then propagated to all its parent nodes

[12] Practically, these equations can be obtained by defining an arbitrary ◊ connected to an arbitrary node. A *post-order* traversal of the graph with ◊ as root determines the equations by simply assuming that the wrench associated to the edge connected to ◊ is null.

and edges which will be correspondingly labeled "not computable".

Algorithm 3 Solution of dynamic EOG exploiting a tree

Require: EOG, $w_s \forall$ FTS
Ensure: $w_i, \forall v_i$
1: For every FTS, attach a node ♦ to the corresponding link
2: Set w_s in each ♦
3: For each ♦, split the graph and create two sub-graphs (see text for details)
4: Attach a node ♢ to each link where a contact is detected: if there is no contact in a sub-graph, choose an arbitrary position and attach a fictitious ♢ (in this case, usually the terminal link or the end-effector is selected)
5: Re-arrange each sub-graph with a ♢ as the root of a tree
6: **for all** sub-graph **do**
7: *DynVisit*(EOG,v_{root})
8: **end for**

DynVisit(EOG,$v = v_i$)
1: **if** v has children **then**
2: **for all** child $e_{v,h} \in \mathcal{C}(v), e_{v,h} \neq i$ **do**
3: $w_{e_{v,h}} = DynVisit(\text{EOG},h)$
4: **end for**
5: Compute w_i with Eq. (6) or Eq. (7) according to the direction of the edges
6: **else**
7: Return w_i
8: **end if**

Once the i-th wrench is known, the associated i-th joint torque can be computed by the following:

$$\tau_i = \mu_i^\top z_{i-1} \tag{8}$$

where z_{i-1} is the z-axis of the reference frame $\langle i-1 \rangle$ as in Fig. 1 (see Sciavicco and Siciliano 2005). Estimated joint torques can thus be used, successively, for joint torque control.[13] Of course, the more six-axis FTS are used, the more accurate can be the estimation. We remark that joint torques basically provide a single component describing the dynamics flowing through the edges, whereas wrenches provide a more complete representation of forces and moments of the structure: this point highlights again the benefit of the proposed method, which allows a full representation of the dynamics of the system and its interaction with the environment.

[13]Moreover, this is not the only information that it is possible to extract from the method. Joint torques are here found as one component of the wrenches flowing through the edges. These wrenches allow having a better representation of the possible contact situation, which can be used as a virtual measurement, to perform every kind of tasks involving force detection and control.

6 A case study: the iCub humanoid robot

6.1 Software

Theoretical results have been implemented in iDyn (Ivaldi et al. 2011), a library for dynamics of single and multiple-branched serial-links kinematic chains. iDyn is built on top of iKin (Pattacini 2011; Pattacini et al. 2010), a library for forward-inverse kinematics of serial-links chains of revolute joints with standard Denavit-Hartenberg notation. Both libraries are part of an open source software project, released under a GPL license. Though being tailored for the iCub, remarkably iKin and iDyn are generic, cross-platform and portable C++ libraries (relying on CMake and YARP middleware, see Fitzpatrick et al. 2008, 2010) that can be used to study kinematics and dynamics of potentially any robotic device.

6.2 Robotic platform

Experiments have been performed on the 53 DOF robot iCub (Metta et al. 2008, 2010). iCub is a full-body humanoid composed[14] of one head (3 DOF), two arms (7 DOF each), torso (3 DOF), two leg (6 DOF each) and 2 hands (9 DOF each), which has been developed by the Italian Institute of Technology within the European Project *RobotCub*, as an open-source platform for research in embodied cognition. Excluding the hands from the model, 32 DOF have been taken into account for computing the inverse dynamics of the robot.

Joints positions are retrieved directly from encoders' measurements, while joints velocities and accelerations are derived from position measurements through a least-squares algorithm based on an adaptive window (Janabi-Sharifi et al. 2000; Fumagalli et al. 2010a).

As shown in Fig. 8(a) and 8(b), iCub is equipped with one inertial sensor (Xsens MTx-28A33G25 (Xsens 2012)) located on the head, providing measurements of ω and $\ddot{p}$ of its final link; $\dot{\omega}$ is again found using an adaptive window filter. Four custom-made six-axes FTS (Fumagalli et al. 2010b), one per leg and arm, are placed proximally with respect to the end-effectors (hands and feet), as shown in Fig. 8(a).

Sets of distributed capacitive tactile sensing elements are integrated on most of the plastic shells covering the robot limbs (Cannata et al. 2008), and provide a tactile feedback for possible contacts with the environment. This sort of "artificial skin" is constituted by a layer of capacitive pressure sensors included on a flexible Printed Circuit Board (PCB),

[14]The description of the iCub kinematics can be found online (iCub Project 2011).

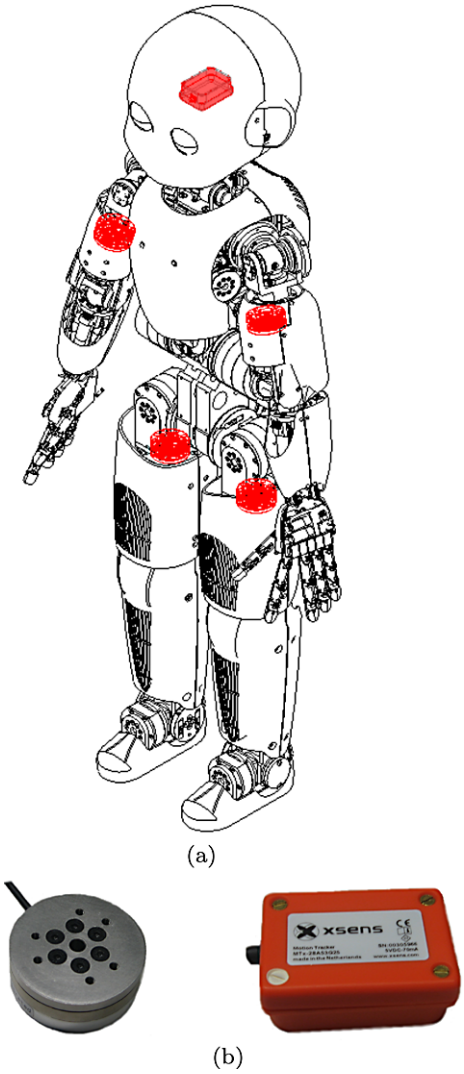


Fig. 8 (**a**) A mechanical scheme of the humanoid robot iCub: in evidence, the four proximal six-axes FTS (legs and arms) and the inertial sensor (head). (**b**) The force/torque and inertial sensors used in iCub

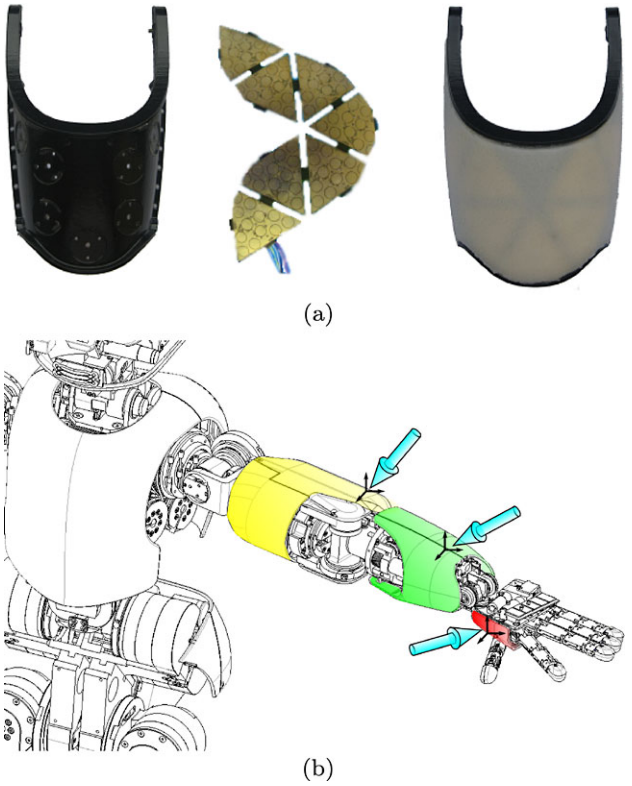


Fig. 9 (**a**) Distributed tactile elements constitute a sort of artificial "skin". The plastic cover, the elements and the final device for the fore-arm are shown. Details about skin fabrication and how iCub has been covered with it can be found in Roboskin (2010). (**b**) Some possible application points (marked with a *frame*) for external forces (*arrows*) arising during contact of the iCub arm with the environment. Upper, fore-arm and palm are covered with plastic shells, providing the base for the tactile elements

with embedded electronics, covered by silicone foam to protect each *taxel* (i.e. tactile element) and make the skin also more compliant. An example of the device for the forearm is shown in Fig. 9(a).

Sensor measures are acquired through local boards, and sent through CAN bus or RS232 bus to a PC104 board, the local CPU on the robot (located on its head). An interface module running on the PC104 replicates collected measures to a local Gigabit Ethernet network, exploiting YARP middleware (Fitzpatrick et al. 2008). On a Blade (i.e. a PC in the cluster) connected to the same local network of the robot, the information of the sensors, sent through YARP ports, is collected and used to compute the robot dynamics as presented in Sect. 3 and Sect. 4, exploiting the iDyn library on a YARP module. The resulting wrenches are then shared again on the YARP network: they can be used for high level tasks, but also for low level joint torque control.

6.3 Experiments

Given the sensors position and the description of the robot kinematics, it is quite easy to build the kinematic and dynamic EOG.

The structure of the EOG needed for the computation of the robot kinematic variables for all joints is shown in Fig. 10(a):[15] the inertial sensor is the unique absolute source of kinematic information (▼) (encoders are relative sources, and their information is considered as a property of the links); unknowns (▽) are placed by default at the end-effectors, so that kinematics variables are propagated through all the graph nodes.

Since the complete knowledge of the kinematic information is a prerequisite for the computation of the dynamics,

[15]Each vertex is named as $X - k$, where $X = \{H, LA, RA, RL, LL, T\}$ is a code for the limb (head, torso, right/left arm/leg) and k means that the corresponding link is the k-th for that specific limb.

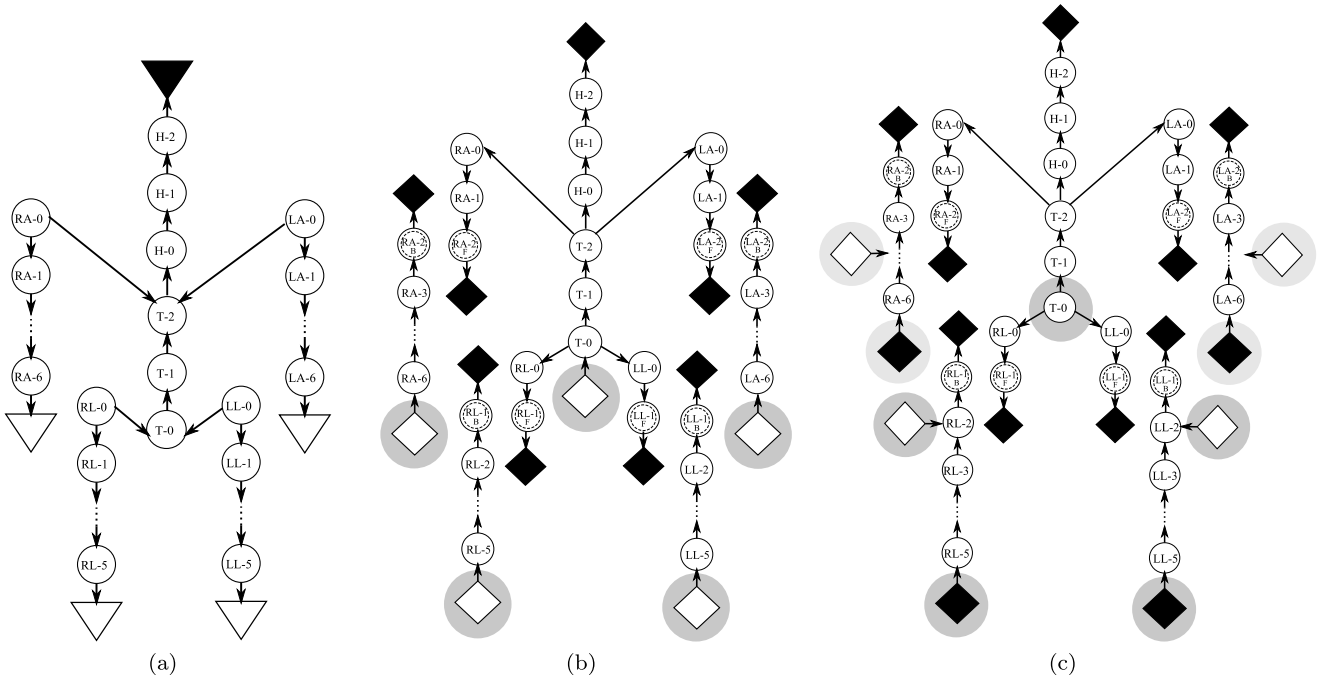


Fig. 10 Representation of iCub's kinematic and dynamic EOG, using the notation of Fig. 3. (**a**) iCub's kinematic EOG. It is noticeable that the inertial sensor measure (▼) is the unique source of kinematic information for the whole branched system. (**b**) iCub's dynamic EOG, when iCub is standing on the mainstay and moving freely in the space, as shown in Fig. 11(a). Given the four FTS, the main graph is cut by the four links hosting the sensors, and a total of five sub-graphs are finally generated. The unknowns are the external wrenches at the end-effector: if the robot does not collide with environment, they must be zero, whereas if a collision happens, an external wrench must arise. The displacement between the expected and the estimated wrenches allows detecting contacts with the environment. Of course, the hypothesis holds that interactions can only occur at the end-effectors. The external wrench on top of the head is assumed to be null. Notice that the mainstay is represented by an unknown wrench ◊. (**c**) iCub's dynamic EOG, when the iCub is crawling like a baby, as shown in Fig. 11(b). As in the previous case, five sub-graphs have been generated after the insertion of the four FTS measurements, but unlike the free-standing case, here the mainstay wrench is missing, being the iCub floating (unfixed) on the floor. Specific locations for the contacts with the environment are specified as being part of the task: thus, the unknown external wrenches (◊) are placed at wrists and knees, while wrenches at the feet and palms are assumed known and null (⧫). Interestingly, while moving on the floor the contact with the upper part could be varying (e.g. wrists, palms, elbows), so the unknown wrenches could be placed in different locations than the ones shown in the graph

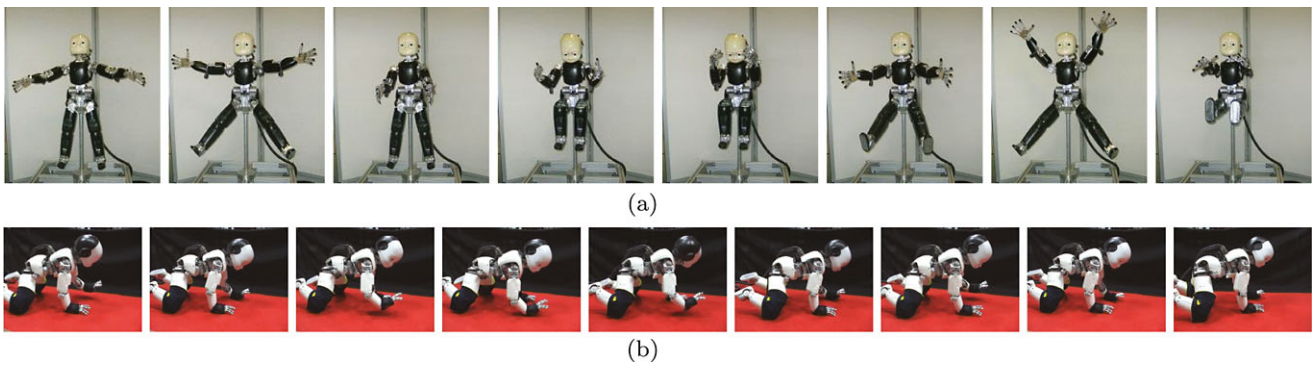


Fig. 11 (**a**) Some snapshots of the "Yoga" demo, where all limbs are moving freely in the space without colliding with the robot own body or the environment. Though having a fixed base (the robot is supported by a metallic mainstay mechanically inserted into its hip), remarkably also the head is moving: thus, the presence of the inertial sensor is crucial for the computation of joint torques. (**b**) Some snapshots of iCub crawling on a carpet. Black straps are used to protect knees and wrists and simultaneously improve the friction of the plastic covers with the floor. Limbs motion is orchestrated by a controller based on central pattern generators (Degallier et al. 2008). Self-body collision is prevented a priori. Interaction with the environment occurs on knees and wrists. The base frame is floating, and the inertial sensor in the head is again crucial, since it provides the linear and angular velocity and acceleration of the head: these components change continuously as an effect of the progression of the robot on the floor, combined with the head movements

the kinematic EOG shown in Fig. 10(a) is adequate for all applications. The dynamic EOG is instead task-dependent.

As the iCub is provided with a set of four FTS, the dynamic EOG is divided in five sub-graphs, each containing a wrench measure (⧫). The head terminal wrench is usually set to zero, so it is treated as a known variable (again ⧫).

The choice of the nodes where unknown wrenches (◊) are applied is instead totally arbitrary and depends on the application point of an interaction force.

For example, if the robot is moving unconstrained in the space, without incurring into contacts with itself or the surrounding as in Fig. 11(a), unknown wrenches (◊) can be statically attached to the end-effectors of the main limbs, hands and feet. Whereas in an interaction scenario, such as the robot crawling on the floor (see Fig. 11(b)), external wrenches must be assumed on wrists and knees.[16]

More in general, unknown wrenches due to any sort of contact cannot be statically attached to a specific link, since the application point of the external force (i.e. the centroid of the contact) is unknown and generally difficult to predict (unless visual feedback is exploited to predict possible contact situations, but it is not always reliable). However, thanks to the artificial tactile skin it is possible to retrieve such information dynamically: therefore the EOG structure can be defined on the fly based on the contact position at each time instant. In such cases, as a consequence of the fact that only one unknown is allowed per each sub-graph, the external force due to contact is the unknown ◊, while wrenches located at the end-effectors are assumed to be known and null (⧫).

Examples of sub-graphs are reported in Fig. 12, which correspond to the contacts shown in Fig. 9(b), where three different contact locations in the left arm are presented.

A rigid-body dynamic model has been used to describe the whole robot. Kinematics and dynamics parameters were retrieved from the CAD model of the robot. Two experiments prove the reliability of the approach:

1. both arms and legs FTS measurements were compared with their model-based prediction, during unconstrained movements (i.e. null external wrenches);
2. measurements from an external FTS, applied at a given position on the end-effectors, were compared with their estimation.

6.4 FTS predictions

During unconstrained, contact-free motion, the measurements w_s from the four six-axes FTS embedded in the limbs

[16]This application also highlights the importance of the inertial sensor, which allows performing the Newton-Euler computations without a fixed base frame (as it is usually assumed in its classical applications).

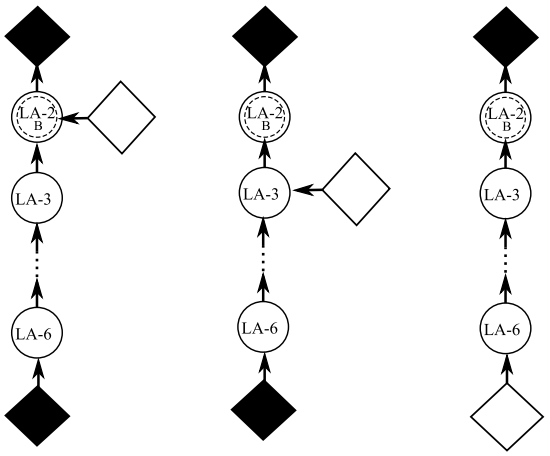


Fig. 12 A sketch of different situations in case of contacts occurring at different locations in the left arm, as shown in Fig. 9(b). The external wrench to estimate (◊) is attached to different nodes. In the *first and second sub-graphs*, the wrench at the end-effector is assumed to be known (⧫), typically null, since only one unknown per graph is allowed (see text for details)

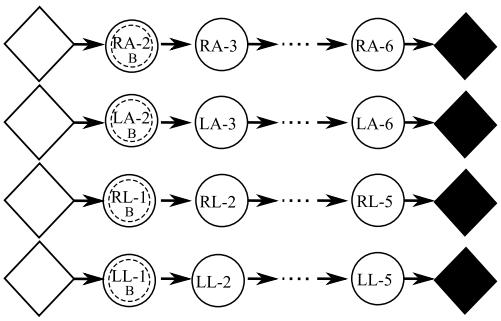


Fig. 13 Enhanced graphs for predicting the four FTS measurements, $\hat{w}_s$, when the external wrench acting at the end-effectors (hands and feet) is known, typically null

have been compared with the analogous quantities $\hat{w}_s$ predicted by the dynamical model. Sensor measurements w_s can be predicted assuming known wrenches at the limbs extremities (hands or feet) and then propagating forces up to the sensors. In this case, null wrenches were assumed, because of the absence of contact with the environment. The EOG in this case is shown in Fig. 13. Table 1 summarizes the statistics of the errors $w_s - \hat{w}_s$ for each limb during the sequence of movements in Fig. 11(a). In particular, the table shows the mean and the standard deviation of the errors between measured and predicted sensor wrench during the movements. Figure 14 plots the error between w_s and $\hat{w}_s$ for the right arm during the same sequence of movements (only one limb out of four is shown without loss of generality).

6.5 External wrench estimation

When solving the dynamic EOG in Fig. 10(b), it is possible to retrieve one external wrench per sub-graph. Thus, we

Table 1 Errors in predicting FTS measures (see text for details)

Right arm: $\epsilon \triangleq \hat{w}_{s,RA} - w_{s,RA}$

	ϵ_{f_0}	ϵ_{f_1}	ϵ_{f_2}	ϵ_{μ_0}	ϵ_{μ_1}	ϵ_{μ_2}
$\bar{\epsilon}$	−0.3157	−0.5209	0.7723	−0.0252	0.0582	0.0197
σ_ϵ	0.5845	0.7156	0.7550	0.0882	0.0688	0.0364

Left arm: $\epsilon \triangleq \hat{w}_{s,LA} - w_{s,LA}$

	ϵ_{f_0}	ϵ_{f_1}	ϵ_{f_2}	ϵ_{μ_0}	ϵ_{μ_1}	ϵ_{μ_2}
$\bar{\epsilon}$	−0.0908	−0.4811	0.8699	0.0436	0.0382	0.0030
σ_ϵ	0.5742	0.6677	0.7920	0.1048	0.0702	0.0332

Right leg: $\epsilon \triangleq \hat{w}_{s,RL} - w_{s,RL}$

	ϵ_{f_0}	ϵ_{f_1}	ϵ_{f_2}	ϵ_{μ_0}	ϵ_{μ_1}	ϵ_{μ_2}
$\bar{\epsilon}$	0.1090	0.0923	0.0121	−0.0912	−0.0004	0.0060
σ_ϵ	0.9740	0.3615	0.2211	0.1429	0.2318	0.0110

Left leg: $\epsilon \triangleq \hat{w}_{s,LL} - w_{s,LL}$

	ϵ_{f_0}	ϵ_{f_1}	ϵ_{f_2}	ϵ_{μ_0}	ϵ_{μ_1}	ϵ_{μ_2}
$\bar{\epsilon}$	0.1100	0.1089	0.0451	−0.0449	0.0022	−0.0386
σ_ϵ	0.9517	0.3297	0.2916	0.1245	0.2542	0.0052

Notes: $\epsilon \triangleq \hat{w} - w = [\epsilon_{f_0}, \epsilon_{f_1}, \epsilon_{f_2}, \epsilon_{\mu_0}, \epsilon_{\mu_1}, \epsilon_{\mu_2}]$
SI Units: f: [N], μ: [Nm]

compared the estimation of an external wrench applied at the end-effector with a direct measure of it, through a free-standing six-axes FTS which was manually "pushed" on the terminal link. In particular, a wrench w^E was exerted on the left hand and measured with the external FTS. Its value was then compared with $\hat{w}^E$, the estimation of the external wrench obtained by propagating the embedded FTS measure in the sub-graph until the frame corresponding to the application point of w^E. A plot of w^E and $\hat{w}^E$ is reported in Fig. 15.

As a counter evidence of the reliability of the method we compared the torques $\hat{\tau}$, determined with (8) with the ones corresponding to the projection on joints of an external wrench applied at the end-effector $\tau^E = J_E^\top w^E$, where $J_E \in \mathbb{R}^{6\times n}$ is the Jacobian (here referred to the frame of the node connecting torso, head and arms).

During this experiment the arm is not moving, while the external force is applied on the hand. Joint torques measured with the *virtual torque sensors* are $\hat{\tau} = \hat{\tau}^I + \hat{\tau}^E$, being τ^I the internal joint torque, i.e. the torque which is due to the intrinsic dynamic of the system. τ^E, i.e. the external force projected on joints, instead is not affected by the internal dynamics (e.g. the gravitational component in this specific static case). Figure 16 shows a comparison of the variation of torque, due to an external wrench application. In particular, we show the comparison between τ^E and $\hat{\tau}^E = \hat{\tau} - \hat{\tau}^I$.

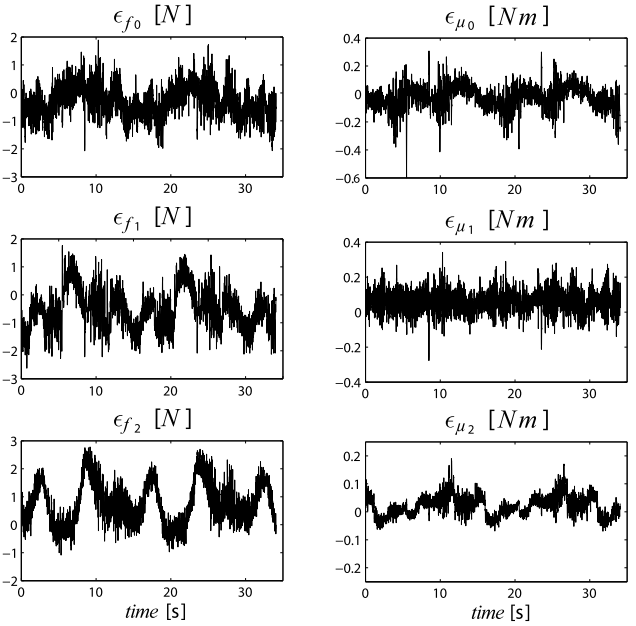


Fig. 14 Right arm: error between the wrenches measured by the FT sensor $w_{s,RA}$ and the one predicted with the model $\hat{w}_{s,RA}$, during the "Yoga" demo

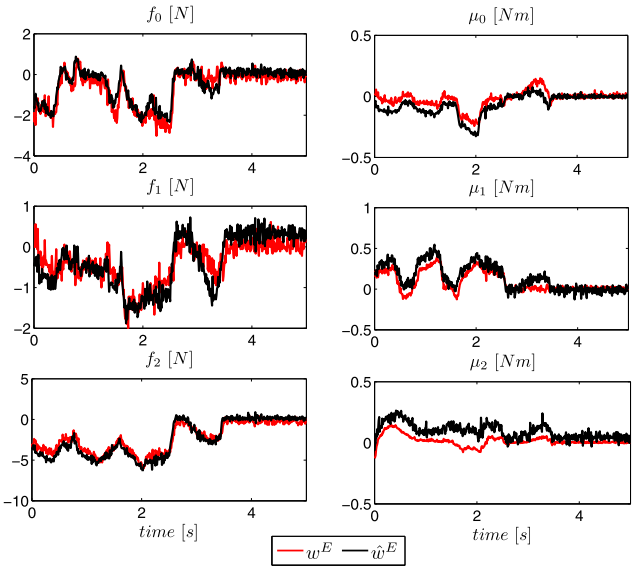


Fig. 15 Left arm: comparison between the external wrench estimated after the FT sensor measurements and the one measured by an external FT sensor, placed on the palm of the left hand

6.6 Exploiting the tactile feedback

As anticipated, the iCub artificial "skin" (Cannata et al. 2008; Roboskin 2010) allows retrieving information about the location of possible contact points (i.e. location of externally applied wrenches) practically on the most of the robot body. Figure 12 shows how the dynamism of the EOG

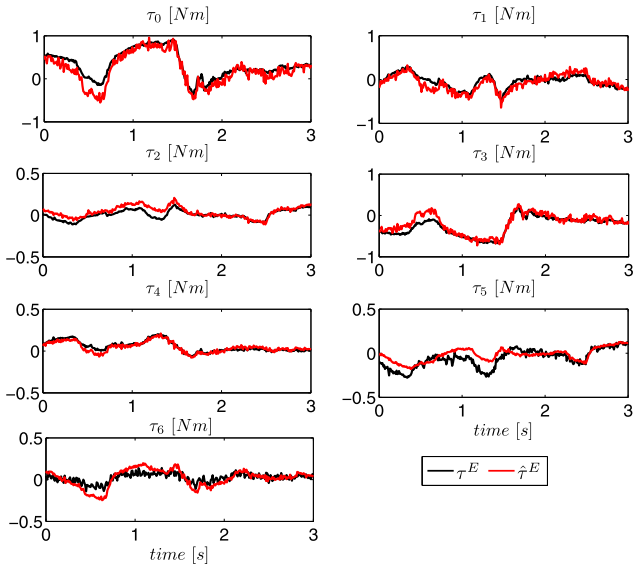


Fig. 16 Left arm: comparison between the torques computed exploiting the embedded FTS and the ones obtained by projecting the external FTS on the joints through the Jacobian (see text)

method can be fully exploited when the link where the contact occurs is known through tactile measurements.

We remark again that Figs. 11(a)–(b) are only possible instances of the EOG, and that the graph is continuously re-created along with the update of the sensory information coming from the tactile skin, indicating the contact locations.

6.7 Comparison with joint torque measures

Further experiments have been conducted on an experimental setup, to evaluate the estimation accuracy of the joint torques with their measure through joint torque sensors. More precisely a prototype of the new iCub arm (Parmiggiani et al. 2009) has been used, which was constituted by 4 DOF arm (shoulder and elbow only), equipped with two six axes-FTS (one placed proximally as in the iCub, the other one placed at the end of the fore-arm, as the wrist) and four JTS.

In the first experiment, during nearly static movements of the arm, we compared the JTS measures τ_j with the torques τ_{FT} estimated through the EOG exploiting the proximal FTS, and also the torques τ_{m+e} composed by adding the gravity torques predicted by the model with the projection of the external wrench (due to the wrist FTS) on the links through the Jacobian. The prediction error is reported in Table 2, while experimental results are shown in Fig. 17. Remarkably, the estimation error is smaller than 0.2 Nm for all the four joints.

In a second experiment, we validated the estimation during fast repetitive movements of the arm, which was also loaded with a known weight attached at the end-effector.

Table 2 Mean and standard deviation of the prediction errors, computed between the measured joint torques and the ones estimated exploiting the FTS or projecting a known external wrench

$\epsilon_{FT} \triangleq \tau_j - \tau_{FT}$

	Joint 0	Joint 1	Joint 2	Joint 3
$\bar{\epsilon}_{FT}$	0.127	−0.049	−0.002	−0.032
$\sigma_{\epsilon FT}$	0.186	0.131	0.013	0.042

$\epsilon_{m+e} \triangleq \tau_j - \tau_{m+e}$

	Joint 0	Joint 1	Joint 2	Joint 3
$\bar{\epsilon}_{m+e}$	0.075	−0.098	−0.006	0.006
$\sigma_{\epsilon m+e}$	0.191	0.173	0.020	0.032

Note: SI Unit: τ: [Nm]

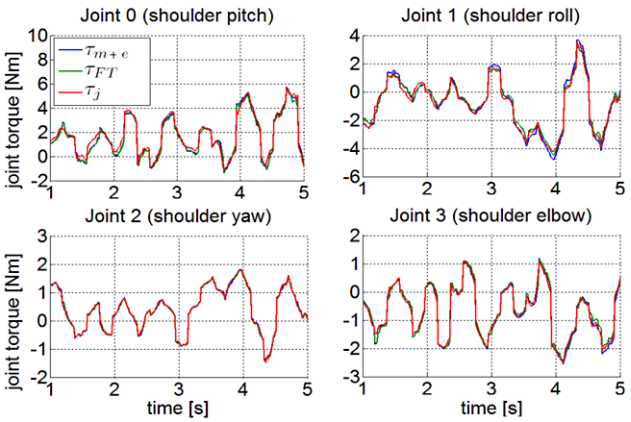


Fig. 17 Comparison between the measured joint torques τ_j and the ones estimated exploiting the FTS, τ_{FT}, or projecting a known external wrench, τ_{m+e}, during quasi-static movements of an arm prototype equipped with JTS. Statistics on the error are reported in Table 2

The purpose was to reveal the discrepancies between measure and estimation due to inaccuracies in the dynamics model to the inevitable errors due to filtering noise (which is the case, for example, of filtering encoders' measurements to retrieve joint velocities and accelerations). As shown in Fig. 18, for repetitive movements performed at 90 and 180 deg/s, the estimation error of the elbow joint torque increases with the acceleration. Moreover, as discussed by Randazzo et al. (2011), the dynamic errors increase with the kinematic distance between the FTS and the joint, because of the propagation of the noise affecting the estimation of velocity and acceleration, which is propagated in the recursive computations.

7 Conclusions

We presented a method for exploiting measurements from multiple sensors distributed along a multiple branches kine-

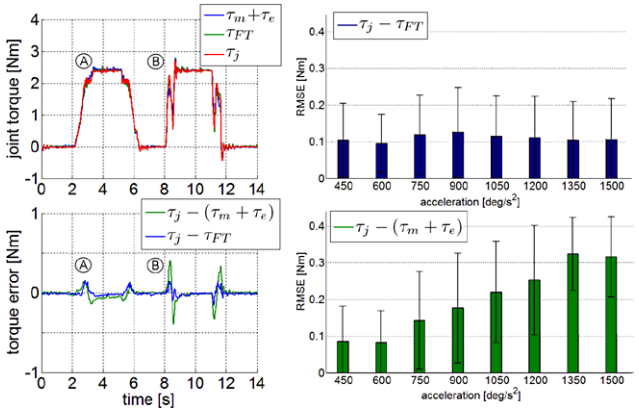


Fig. 18 *Left*: measured and predicted joint torques during fast movements of the elbow; trajectories were performed at *A*: 90 deg/s, *B*: 180 deg/s. *Right*: regularized mean squared error (RMSE) of the prediction errors, increasing with the acceleration

matic chain, to compute a complete representation of the robot dynamics. Force/torque and tactile sensors are the crucial sources of information. With the additional information of a single inertia sensor placed anywhere in the chain, the proposed method can be used to propagate wrenches within the robot exploiting a graph representation of its structure. In presence of unknown wrenches (at locations detected by the tactile sensor), we proposed a systematic procedure for propagating force/torque measurements to get an estimation of the unknowns. The proposed procedure consists on a post-order traversal of a tree which is obtained by an online rearrangement of the graph, according to the contact locations. Conditions on the graph structure have been given in order to guarantee the propagation of force/torque measurements to the entire kinematic chain and, in particular, to the unknowns. It was shown that given N-FTS, a maximum of $N+1$ unknown external wrenches can be estimated, if there is exactly one unknown per each of the sub-graphs induced by the force/torque sensors locations. When all external wrenches can be computed, then all joint torques can be computed too. An artificial tactile skin, covering most of the surface of the robot body, was used to provide the application points of the external wrenches. A software library for performing these computations has been released with an open source license (Ivaldi et al. 2011). Results on the implementation on the 53-DOFs humanoid robot iCub have been presented.

The effectiveness of the proposed algorithms has been proved in a variety of experiments with the iCub platform: by means of iDyn, active force control strategies were introduced to enhance existing motion control and provide essential safety during interactions. Numerous videos of realized applications can be found at http://www.youtube.com/robotcub.

Acknowledgements The authors wish to thank Dr. Ugo Pattacini for the iKin software library, upon which iDyn was built. The authors acknowledge the support for the CHRIS, ITALK, Viactors and Roboskin Projects provided by the European Commission grant agreement number FP7-ICT-215805, FP7-ICT-214668, FP7-ICT-231554, FP7/2007-2013 No. 231500.

References

Caccavale, F., Natale, C., Siciliano, B., & Villani, L. (2005). Integration for the next generation: embedding force control into industrial robots. *IEEE Robotics & Automation Magazine*, *12*(3), 53–64.

Calinon, S., Sardellitti, I., & Caldwell, D. (2010). Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies. In *IEEE/RSJ int. conf. on intelligent robots and systems*, Taipei, Taiwan.

Cannata, G., Maggiali, M., Metta, G., & Sandini, G. (2008). An embedded artificial skin for humanoid robots. In *IEEE int. conf. on multisensor fusion and integration*, Seoul, Korea.

Chiaverini, S., Siciliano, B., & Villani, L. (1999). A survey of robot interaction control schemes with experimental comparison. *IEEE/ASME Transactions on Mechatronics*, *4*(3), 273–285.

Cormen, T., Leiserson, C., Rivest, R., & Stein, C. (2002). *Introduction to algorithms* (2nd ed.). New York: McGraw-Hill.

Degallier, S., Righetti, L., Natale, L., Nori, F., Metta, G., & Ijspeert, A. (2008). A modular bio-inspired architecture for movement generation for the infant-like robot icub. In *IEEE RAS/EMBS int. conf. on biomedical robotics and biomechatronics*, Scottsdale, Arizona.

Eiberger, O., Haddadin, S., Weis, M., Albu-Schäffer, A., & Hirzinger, G. (2010). On joint design with intrinsic variable compliance: derivation of the DLR QA-joint. In *IEEE int. conf. on robotics and automation* (pp. 1687–1694).

Featherstone, R. (2007). *Rigid body dynamics algorithms*. New York: Springer.

Featherstone, R. (2010). Exploiting sparsity in operational-space dynamics. *The International Journal of Robotics Research*, *29*, 1353–1368.

Featherstone, R., & Orin, D. E. (2008). Dynamics. In B. Siciliano & O. Khatib (Eds.), *Handbook of robotics* (pp. 35–65). Berlin: Springer.

Fitzpatrick, P., Metta, G., & Natale, L. (2008). Towards long-lived robot genes. *Robotics and Autonomous Systems*, *56*, 29–45.

Fitzpatrick, P., Natale, L., & Metta, G. (2010). The Cmaking of a humanoid. *The Kitware Source: Software Developer's Quarterly*, *13*, 7–9.

Fumagalli, M., Gijsberts, A., Ivaldi, S., Jamone, L., Metta, G., Natale, L., Nori, F., & Sandini, G. (2010a). Learning to exploit proximal force sensing: a comparison approach. In O. Sigaud & J. Peters (Eds.), *From motor learning to interaction learning in robots* (pp. 149–167). Berlin: Springer.

Fumagalli, M., Randazzo, M., Nori, F., Natale, L., Metta, G., & Sandini, G. (2010b). Exploiting proximal F/T measurements for the iCub active compliance. In *IEEE/RSJ int. conf. on intelligent robots and systems*, Taipei, Taiwan.

Haddadin, S., Albu-Schäffer, A., Frommberger, M., & Hirzinger, G. (2008a). The role of the robot mass and velocity in physical human-robot interaction—part II: Constrained blunt impacts. In *IEEE int. conf. on robotics and automation*, Pasadena, CA, USA.

Haddadin, S., Albu-Schäffer, A., & Hirzinger, G. (2008b). The role of the robot mass and velocity in physical human-robot interaction—part I: Non-constrained blunt impacts. In *IEEE int. conf. on robotics and automation*, Pasadena, CA, USA.

Haddadin, S., Albu-Schaffer, A., Eiberger, O., & Hirzinger, G. (2010a). New insights concerning intrinsic joint elasticity for safety. In *IEEE/RSJ int. conf. on intelligent robots and systems*.

Haddadin, S., Urbanek, H., Parusel, S., Burschka, D., Rossmann, J., Albu-Schaffer, A., & Hirzinger, G. (2010b). Real-time reactive motion generation based on variable attractor dynamics and shaped velocities. In *2010 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 3109–3116).

iCub Project, T. (2011). Documentation of icub kinematics. http://eris.liralab.it/wiki/ICubForwardKinematics.

Ivaldi, S., Fumagalli, M., Randazzo, M., Nori, F., Metta, G., & Sandini, G. (2011). Computing robot internal/external wrenches by means of inertial, tactile and F/T sensors: theory and implementation on the iCub. In *IEEE-RAS int. conf. humanoid robots (HUMANOIDS)* (pp. 521–528).

Ivaldi, S., Fumagalli, M., & Pattacini, U. (2011). Doxygen documentation of the iDyn library. http://eris.liralab.it/iCub/main/dox/html/group__iDyn.html.

Janabi-Sharifi, F., Hayward, V., & Chen, C. S. J. (2000). Discrete-time adaptive windowing for velocity estimation. *IEEE Transactions on Control Systems Technology*, *8*(6), 1003–1009.

Kulic, D., & Croft, E. (2007). Pre-collision safety strategies for human-robot interaction. *Autonomous Robots*, *22*, 149–164.

Luca, A. D. (2006). Collision detection and safe reaction with the DLR-III lightweight manipulator arm. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 1623–1630).

Luh, J., Fisher, W., & Paul, R. (1983). Joint torque control by a direct feedback for industrial robots. *IEEE Transactions on Automatic Control*, *28*(2), 153–161.

Maggiali, M., Cannata, G., Maiolino, P., Metta, G., Randazzo, M., & Sandini, G. (2008). Embedded distributed capacitive tactile sensor. In *Mechatronics 2008*, Limerick, Ireland.

Metta, G., Sandini, G., Vernon, D., Natale, L., & Nori, F. (2008). The iCub humanoid robot: an open platform for research in embodied cognition. In *PerMIS: performance metrics for intelligent systems workshop*, Washington DC, USA, Aug. 19–21.

Metta, G., Natale, L., Nori, F., Sandini, G., Vernon, D., Fadiga, L., von Hofsten, C., Rosander, K., Santos-Victor, J., Bernardino, A., & Montesano, L. (2010). The iCub humanoid robot: an open-systems platform for research in cognitive development. *Neural Networks*, *23*, 1125–1134, Special issue on social cognition: from babies to robots.

Minguez, J., Lamiraux, F., & Laumond, J. P. (2008). Motion planning and obstacle avoidance. In B. Siciliano & O. Khatib (Eds.), *Handbook of robotics* (pp. 827–852). Berlin: Springer.

Mistry, M., Buchli, J., & Schaal, S. (2010). Inverse dynamics control of floating base systems using orthogonal decomposition. In *IEEE int. conf. on robotics and automation*.

Morel, G., & Dubowsky, S. (1996). The precise control of manipulators with joint friction: a base force/torque sensor method. In *IEEE int. conf. on robotics and automation* (pp. 360–365).

Morel, G., Iagnemma, K., & Dubowsky, S. (2000). The precise control of manipulators with high joint friction using base force/torque sensing. *Automatica*, *36*(7), 931–941.

Mutambara, A. (1998). *Decentralized estimation and control for multisensory systems*. Boca Raton: CRC Press.

Parmiggiani, A., Randazzo, M., Natale, L., Metta, G., & Sandini, G. (2009). Joint torque sensing for the upper-body of the iCub humanoid robot. In *Int. conf. on humanoid robotics* (p. 2009). France: Paris.

Pattacini, U. (2011). Doxygen documentation of the iKyn library. http://eris.liralab.it/iCub/main/dox/html/group__iKin.html.

Pattacini, U., Nori, F., Natale, L., Metta, G., & Sandini, G. (2010). An experimental evaluation of a novel minimum-jerk Cartesian controller for humanoid robots. In *IEEE/RSJ int. conf. on intelligent robots and systems (IROS)*, Taipei, Taiwan.

Pratt, G., & Williamson, M. (1995). Series elastic actuators. In *IEEE/RSJ int. conf. on intelligent robots and systems*, Los Alamitos, CA, USA (pp. 399–406).

Randazzo, M., Fumagalli, M., Nori, F., Natale, L., Metta, G., & Sandini, G. (2011). A comparison between joint level torque sensing and proximal F/T sensor torque estimation: implementation on the iCub. In *IEEE-RSJ int. conf on intelligent robots and systems (IROS)*, San Francisco, USA.

Roboskin, E. P. I. F. (2010). http://www.roboskin.eu.

Santis, A. D., Siciliano, B., Deluca, A., & Bicchi, A. (2008). An atlas of physical human-robot interaction. *Mechanism and Machine Theory*, *43*(3), 253–270.

Sciavicco, L., & Siciliano, B. (2005). *Advanced textbooks in control and signal processing*. *Modelling and control of robot manipulators* (2nd ed.). Berlin: Springer.

Siciliano, B., & Villani, L. (1996). A passivity-based approach to force regulation and motion control of robot manipulators. *Automatica*, *32*(3), 443–447.

Siciliano, B., & Villani, L. (2000). *Robot force control*. Norwell: Kluwer Academic.

Sisbot, E., Marin-Urias, L., Broquère, X., Sidobre, D., & Alami, R. (2010). Synthesizing robot motions adapted to human presence. *International Journal of Social Robotics*, *2*, 329–343.

Wittenburg, J. (1994). Topological description of articulated systems. In *NATO ASI series: Vol. 268. Computer-aided analysis of rigid and flexible mechanical systems, Part I* (pp. 159–196). http://www.springerlink.com/content/l364j72l49103vv2/.

Xsens (2012). The MTx orientation tracker user manual. http://www.xsens.com/en/general/mtx.

Matteo Fumagalli was born in 1982 in Rochester, Minnesota. He studied mechanical engineering at Politecnico di Milano where, on July 2006 he got his D.Eng. degree in Mechatronics. His master thesis was on force control of cooperative robots. He arrived at the Italian Institute of Technology on January 2007, as a Ph.D. student of the University of Genoa. There he studied interaction control exploiting proximal force sensors on the James and iCub humanoid robot. In 2011 he received his Ph.D. in Humanoid Technologies, and then he moved to the University of Twente (The Netherlands) where he is currently a PostDoc in the Control Engineering group.

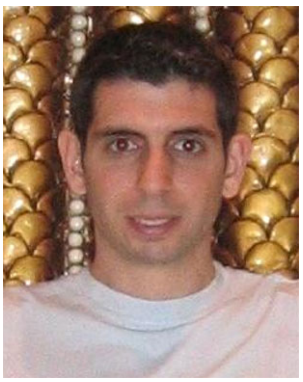
Serena Ivaldi was born in Genoa in 1982. She received her B.S. and M.S. degree in Computer Engineering, both with highest honors, at the University of Genoa, Italy, in 2004 and 2006 respectively. In 2007 she started her Ph.D. in Humanoid Technologies jointly at the University of Genoa and Italian Institute of Technology, where she also held a research fellowship. From 2011 she is a PostDoc researcher in the Institut des Systemes Intelligents et de Robotique, in the University Pierre et Marie Curie in Paris. Her research interests include optimal control, whole-body dynamics and humanoid robotics.

Marco Randazzo was born in Genova in 1981. He graduated in Computer Science Engineering at the University of Genova in 2005. He received his Ph.D. degree at the Italian Institute of Technology (IIT) in 2009, working on the development of multilayer Dielectric Elastomer Actuators for robotic applications. He is currently postdoc researcher at the Italian Institute of Technology. His research interests include force control and compliant actuation on humanoids robots, smart materials, electro active polymers and their potential applications as artificial muscles for the next-generation humanoid robots.

Lorenzo Natale received his degree in Electronic Engineering (with honors) in 2000 and Ph.D. in Robotics in 2004 from the University of Genoa. He worked in the Laboratory for Integrated Advanced Robotics (LIRA-Lab), at the University of Genoa, and was later on postdoctoral researcher at the MIT Computer Science and Artificial Intelligence Laboratory. At the moment he is Team Leader at the Italian Institute of Technology. In the past ten years Lorenzo Natale worked on several humanoid platforms. His research interests range from sensorimotor learning and perception to software architectures for robotics. He has collaborated in several EU-funded projects (MIRROR, CogVis, ADAPT, RobotCub and CHRIS).

Giorgio Metta is senior scientist at the IIT and assistant professor at the University of Genoa where he teaches courses on anthropomorphic robotics and intelligent systems for the bioengineering curricula. He holds a M.S. with honors (in 1994) and Ph.D. (in 2000) in electronic engineering both from the University of Genoa. From 2001 to 2002 he was postdoctoral associate at the MIT AI-Lab where he worked on various humanoid robotic platforms. He is assistant professor at the University of Genoa since 2005 and with IIT since 2006. Giorgio Metta research activities are in the fields of biologically motivated and humanoid robotics and in particular in developing life-long developing artificial systems that show some of the abilities of natural systems. His research developed in collaboration with leading European and international scientists from different disciplines like neuroscience, psychology, and robotics. Giorgio Metta is author of approximately 100 publications. He has been working as principal investigator and research scientist in several international and national funded projects. He has been reviewer for international journals and the European Commission.

Giulio Sandini is director of research at the Italian Institute of Technology and full professor of bioengineering at the University of Genoa. His main research interests are in the fields of Computational and Cognitive Neuroscience and Robotics with the objective of understanding the neural mechanisms of human sensorimotor coordination and cognitive development from a biological and an artificial perspective. He graduated in Electronic Engineeering (Bioengineering) at the University of Genova. He has been assistant professor at the Scuola Normale Superiore in Pisa and Visiting Scientist at the department of neurology of the Harvard Medical School and at the Artificial Intelligence lab at MIT. Since 2006 he is Director of Research at the Italian Institute of Technology where he leads the Department of Robotics, Brain and Cognitive Science.



Francesco Nori was born in Padova in 1976. He received his D.Eng. degree (highest honors) from the University of Padova (Italy) in 2002. During the year 2002 he was a member of the UCLA Vision Lab as a visiting student under the supervision of Prof. Stefano Soatto, University of California Los Angeles. In the year 2006 he moved to the University of Genova and started his PostDoc at the laboratory for integrated advanced robotics, beginning a fruitful collaboration with Prof. Giorgio Metta and Prof. Giulio Sandini. In 2007 Francesco Nori has moved to the Italian Institute of technology where he is currently hired as a team leader.