Computing robot internal/external wrenches by means of inertial, tactile and F/T sensors: theory and implementation on the iCub

S. Ivaldi, M. Fumagalli, M. Randazzo, F. Nori, G. Metta, G. Sandini

***Abstract*—Reliable access to dynamics is one of the central challenges in humanoid robotics. In this paper we consider the problem of computing both internal and external wrenches in open multiple branches kinematic chains, which are not equipped with joint-level torque sensors, but on the contrary are provided with inertial and tactile sensing, and a set of six-axes Force/Torque Sensors (FTS) distributed within the chain. The proposed method is grounded on the Enhanced Oriented Graph, a graph-representation of the kinematic chain, enriched with the information coming from the different sensors. Under suitable conditions, a maximum of $N+1$ external wrenches can be estimated from N six-axes FTS. The graph is built dynamically, and internal and external wrenches are consequently updated. Theoretical results have been implemented in a software library (iDyn) released with an open-source license (GPL) as part of the iCub software project. The proposed method has been applied to 32 of the 53 DOF of the iCub humanoid robot.**

I. INTRODUCTION

When robots share the same unstructured environment with humans, compliance and safety must be addressed during physical human-robot and environment-robot interaction, to prevent the robot from damaging itself and its surrounding. Therefore it is necessary to provide the robot with the most complete representation of the interaction forces arising during contacts to design suitable motor control strategies [18]. In many existing robots actuated by stiff DC motors and not equipped with suitable sensors so as to provide force feedback, force control schemes are hardly feasible.

In this context, robotics research is currently following several directions, the principal being passive and active compliance. *Passive compliance* relies on passive elastic elements typically interposed in between the actuator and the point where interaction occurs [2]. An example of passive compliance is the so-called series elastic actuator [15] where a spring is inserted between the motor and the actuated joint. An emerging technology is *variable passive compliance*, which relies on passive elements whose stiffness can be actively varied [4]. As an alternative, *active compliance* consists in regulating the interaction forces at each time instant by means of a closed-loop force controller [19, 20]. Practically, forces and/or torques are measured with suitable sensors, and are controlled at the actuators level so as to regulate the interaction forces/torques to the desired value. The main advantage of the active regulation over the passive one is the possibility of regulating forces within a wider range of values. One disadvantage is the response delay of the regulator, which typically limits the bandwidth of the controlled system. Classically, Joint Torque Sensors (JTS) are used to provide force feedback and perform fast control loops directly at the joint level. If robots are not equipped with such sensors, torques can be estimated by measuring the current absorbed by motors, but only when most of the motor torque is transmitted to the joint (low friction), which is unfeasible for many existing humanoid robots.

In this paper we present a particularly convenient way to retrieve force feedback from existing robots which are not directly equipped with JTS, by using three different sets of sensors: inertial, force/torque and tactile. The proposed technique provides a complete perceptual representation of the intrinsic dynamics of the robot and its interaction forces arising due to external contacts.

Actually, the tools discussed in this paper allow computing internal forces and torques[1] of generic multiple branches kinematic chains equipped with one or more six-axes Force/Torque Sensors (FTS). Moreover, we prove that under suitable assumptions, there exists a systematic procedure for computing $N+1$ external wrenches from N known internal wrenches (i.e. FTS measures). Theoretical results have been implemented in an open source C++ library called iDyn[2], released under GPL license as part of the iCub open source project [1]. The proposed method has been fruitfully applied to 32 of the 53 Degrees Of Freedom (DOF) of the iCub, which is equipped with four six-axes FTS located proximally on each arms and legs, one inertial sensor located in the head, and an artificial skin covering partially its body. Experiments shows the effectiveness of our approach, which has been tested in a

This research was supported by the European Projects: VIACTORS (FP7-ICT-2007-3, contract N231554), CHRIS (FP7-IST-215805), ROBOSKIN (FP7-231500) and ITALK (ICT-214668). Email: serena.ivaldi@isir.upmc.fr

All the authors are with the Robotics, Brain and Cognitive Sciences Department, Istituto Italiano di Tecnologia, Genoa, Italy.

G. Metta and G. Sandini are also with the Department of Communication, Computer and System Sciences, University of Genoa, Italy.

S. Ivaldi is also with the Institut des Systèmes Intelligents et de Robotique, Université Pierre et Marie Curie, Paris, France.

M. Fumagalli is also with the Department of Control Engineering, University of Twente, Enschede, The Nederlands.

[1]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 $w = \begin{pmatrix} f \\ \mu \end{pmatrix}$. FTS, 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.

[2]iDyn is a C++ set of classes for computation of dynamics, built on top of iKin [14] a library for forward-inverse kinematics of serial-links chains of revolute joints with standard Denavit-Hartenberg notation, adopting SI units.

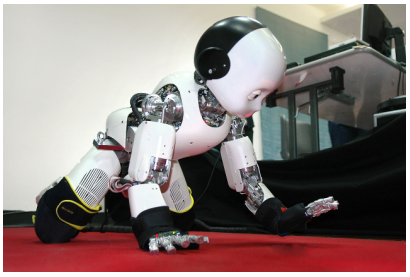

Figure 1. The humanoid robot iCub [21].

variety of tasks and robot configurations: during fixed-waist free motions, floating-base crawling and human interaction scenarios.

The paper is organized as follows: Section II describes the theoretical framework for computing the complete dynamics of kinematics chains through the Enhanced Oriented Graphs, exploiting different sensors measurements. Section III describes the application of the proposed method to the iCub humanoid robot, whereas Section IV presents some experimental results obtained with the dynamics library iDyn on the iCub, where the ideas presented in this paper have been implemented for realizing an active compliance control scheme based on the joint torques estimated from proximal FTS. Finally, Sections V and VI discuss the proposed method and draw conclusions.

II. ENHANCED ORIENTED GRAPHS

In this section we introduce the foundation of the framework for computing the internal wrenches acting within a kinematic chain, as well as the external wrenches acting on it. The proposed method is based on the classical Recursive Newton-Euler Algorithm (RNEA) (see [19] Chapter 4 Section 5). Similarly to the classical approach, we represent a kinematic chain as a graph, so that the dynamics can be obtained by applying the well-known equations during pre-order or post-order visits of the graph itself. However, here we overcome some limitations of the classical RNEA, particularly to deal with a floating base chain, different sensory information, and with multiple external wrenches which may be exerted at arbitrary locations (not exclusively at the end-effector). Thus, we enhance the graph with additional elements representing multiple sensory sources, and known/unknown kinematic and dynamic variables, to obtain the so-called resulting Enhanced Oriented Graph (EOG). Since the EOG evolves dynamically, the robot internal dynamics (link wrenches, joint torques) and the external wrenches caused by physical interactions are computed on the fly, without making any constraining hypotheses about the location of the contacts (e.g. contacts at the end-effector only).

If the robot model is known, a single *Inertial Sensor* is sufficient to compute the kinematics of the robot in unconstrained, floating base conditions. *Force/Torque Sensors* (FTS) can be often inserted in kinematics chains, thanks to their compact size, avoiding radical changes in the mechanics [22]. Furthermore, not only they have the great advantage of being relatively cheap, but also they provide a better representation of the interaction forces (six components per each wrench). For this reason, classically robots are equipped with such sensors mounted in their end-effectors, where most of the interactions with the environment occurs. In contrast, our solution is based on a set of *proximal* FTS, meaning that they are close to the base of the kinematic chains and far from the end-effectors: this configuration enables to measure not only interaction forces acting at the end of the chains, but also forces acting between the sensor and distal joints. To the best of our knowledge, this solution has been adopted only once in [11, 12] where a single FTS was used to estimate the joint torques in the first 3 DOFs of a PUMA manipulator. Finally, *Tactile Sensors* are necessary to retrieve dynamically the location of possible contacts between the robot and the environment. Their use is mandatory if there are no a priori assumptions about the locations of the external forces: thus, the best is to cover as much surface of the robot as possible, and particularly the limbs which are more involved in interaction scenarios, such as arms and hands. For a review of the different tactile sensing technologies, see [3].

A. Background

Graph theory has been extensively used to represent mechanical systems [23, 5, 6, 24], and to formalize the computation of the internal wrenches of mechanical systems.

We consider an open (single or multiple branches) kinematic chain with n DOF. Adopting the Denavit-Hartenberg notation [19], we define a set of reference frames $\langle 0 \rangle$, $\langle 1 \rangle$, $\ldots$, $\langle n \rangle$ attached to each link. The i-th link of the chain is then described by its frame $\langle i \rangle$ and a vertex v_i (also called node), represented with the symbol $\textcircled{v_i}$. A hinge joint between link i and link j (i.e. a rotational joint) is represented by an oriented edge $e_{i,j}$ connecting v_i with v_j, represented by an oriented arrow: $\textcircled{v_i} \rightarrow \textcircled{v_j}$. The orientation of the edge can be chosen arbitrarily or follow the exploration of the kinematic tree according to the "regular numbering scheme", which induces parent/child relationships such that each node has a unique input edge (parent) and multiple output edges (children).

The graph can be used to represent both kinematics and dynamics information associated to each link. In kinematics, the edge $e_{i,j}$ is associated to the reference frame $\langle i \rangle$ attached to the i-th link, set accordingly to the Denavit-Hartenberg notation. The associated kinematic variables are the joint angle position $\theta_i \in \mathbb{R}$ and the angular and linear velocity and accelerations of $\langle i \rangle$, denoted by $\omega_i, \dot{p}_i, \dot{\omega}_i, \ddot{p}_i \in \mathbb{R}^3$ respectively. In dynamics, the edge orientation disambiguates between the wrench that i exerts on j and the (equal and opposite) reaction wrench that j exerts on i: therefore, $e_{i,j}$ is associated to $w_{i,j} \in \mathbb{R}^6$, the interaction wrench that the i-th link exerts on the j-th link, projected in the reference frame of the i-th link.

The Recursive Newton-Euler Algorithm (RNEA) can be used to propagate known kinematic and dynamic variables from the extremities to all the links of the kinematics chain (see [19], chapter on Dynamics). If the position, velocity and acceleration of a given node v_i are known, then this

information can be propagated through the Newton-Euler kinematic step to all the nodes connected to v_i exploiting the joint position/velocity measurements/acceleration measurements. The well-known equations can be iterated to retrieve the i-th link angular velocity and acceleration (ω_i, $\dot{\omega}_i$) and linear acceleration ($\ddot{p}_i$) from the analogous base information (ω_0, $\dot{\omega}_0$, $\ddot{p}_0$):

$$\begin{aligned}
\omega_{i+1} &= \omega_i + \dot{\theta}_{i+1} z_i, \\
\dot{\omega}_{i+1} &= \dot{\omega}_i + \ddot{\theta}_{i+1} z_i + \dot{\theta}_{i+1} \omega_i \times z_i, \\
\ddot{p}_{i+1} &= \ddot{p}_i + \dot{\omega}_i \times r_{i,i+1} + \omega_{i+1} \times \big(\omega_{i+1} \times r_{i,i+1}\big),
\end{aligned} \tag{1}$$

where z_i is the z-axis of $\langle i \rangle$. That is, we propagate the kinematic information from the base to the end-effector, visiting all the nodes moving from one node to the next following the edges. The internal dynamics of the manipulator can be studied as well: if the dynamical parameters of the system are known (mass m_i, inertia $\bar{I}_i$, center of mass C_i), we can propagate knowledge of the wrenches applied at the end-effector (f_{n+1}, μ_{n+1}) to the base frame of the manipulator so as to retrieve forces and moments f_i, μ_i up to f_0, μ_0.

$$\begin{aligned}
f_i &= f_{i+1} + m_i \ddot{p}_{C_i}, \\
\mu_i &= \mu_{i+1} - f_i \times r_{i-1,C_i} + f_{i+1} \times r_{i,C_i} + \bar{I}_i \dot{\omega}_i + \\
&\quad + \omega_i \times \big(\bar{I}_i \omega_i\big),
\end{aligned} \tag{2}$$

where: $\ddot{p}_{C_i} = \ddot{p}_i + \dot{\omega}_i \times r_{i,C_i} + \omega_i \times \big(\omega_i \times r_{i,C_i}\big)$.

Analogous considerations can be extended to multiple branches open kinematic chains (also known as kinematic trees [6]). In that case, the kinematic Newton-Euler computation can be performed by assuming that angular velocity and acceleration (ω_i, $\dot{\omega}_i$) and linear acceleration ($\ddot{p}_i$) are known for at least one node of the graph. This information can then be propagated to the other nodes, by progressively moving from that node (usually the base) to all the other nodes of the graph, i.e. from v_j to v_k using the oriented edge $e_{j,k}$ (or $e_{k,j}$).

Similarly, the wrenches of the system can be computed by propagating the information from one edge to the other. However, in the classical RNEA initialization is always performed at the end-effector, thus following a unique order in visiting the graph. With the EOG framework, we overcome these limitations, and furthermore we give the possibility to include different sources of dynamic information, with a generic procedure to build dynamically the spanning tree from the graph description of the dynamics, taking into account knowns (e.g. measured wrenches) and unknowns (e.g. wrenches to be estimated) of the system.

B. Enhanced Oriented Graphs

Whatever the orientation of the graph and the set of recursive equations used to propagate kinematics and dynamics, some initial information must be set to initialize the computations. When we have measurements of kinematic variables (by means of suitable gyroscopes and/or accelerometers) and applied wrenches (by means of 6-axes FT sensors), the graph representation of the kinematic chain can be enriched, using a specific set of symbols. The graph with this additional information is called *Enhanced Oriented Graph* (EOG).

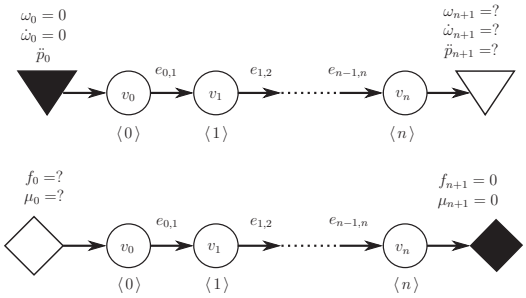

Figure 2. A representation of the classical Newton-Euler computations for an open n-links chain, with the EOG notation. Kinematics are assumed known at the base (▼). Wrenches are assumed known at the end-effector (⧫, e.g. zero if there is no interaction with the environment) and propagated to the base.

Kinematic variables representation: when the kinematic variables ω_i, $\dot{\omega}_i$ and $\ddot{p}_i$ of the i-th link can be measured (e.g. the link is a fixed base or an inertial sensor is attached to the link) we represent this information with a *black triangle* (▼) connected to v_i by a n additional edge. Remarkably, knowing ω_i, $\dot{\omega}_i$ and $\ddot{p}_i$ for a single node v_i is sufficient to propagate the kinematic information to the entire graph, where the unknown kinematics variables are represented with a *white triangle* (▽).

Dynamic variables representation: within a kinematic chain, we can always consider the problem of computing wrenches applied at specific locations given the knowledge of wrenches at (other) locations. *Black rhombi* (⧫) are used to represent known (i.e. measured) wrenches. *White rhombi* (◊) represent unknown wrenches which need to be computed. The reference frame associated to its edge represents the location of the applied or unknown wrench. The latter can be located anywhere in the chain, and generally are dynamically changing, being due to contacts of the manipulator with the environment. Thus, the edge associated to the ◊ and the node to whom is attached are (without constraints or further hypotheses) dynamically changing too.

Fig. 2 shows the enhanced graphs associated with the classical RNEA kinematic and dynamic steps: in this case kinematic variables are known at the base (▼), wrenches are known at the end-effector, since they are assumed to be null (⧫). It is important to point out that, while the positions of ⧫ is static within the graph (because FT sensors are fixed in the mechanical structure of the manipulator), the location of ◊ instead can be dynamic, since contact point locations are dynamically detected by the distributed tactile sensors. If a contact moves along a chain, the graph is modified accordingly. This rule shows a great benefit of the EOG; which dynamically adapts in response to the location of the unknown wrenches. It should be noticed that in absence of the artificial skin, the EOG rule can still be applied, if other methods for estimating external forces locations are used [8]. However, if it is not possible to retrieve the point of contact, the latter can be assumed fixed at certain positions, e.g. the end-effector, where the most interaction is more likely to happen.

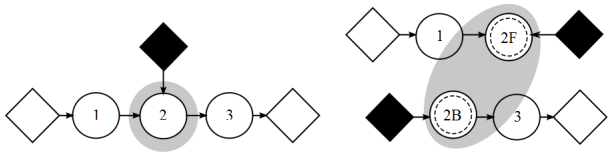

Figure 3. The graph shows how to insert a FTS into an EOG. The node on which the sensor is attached (highlighted), is practically divided into two sub-nodes, each corresponding to the physical sub-link created after cutting the link with the FTS. The graph is then divided into two sub-graphs and two ⧫ (known wrenches corresponding to the sensor measurement) are connected to the sub-nodes.

C. Representing FT measurements in a graph

The dynamic information from a 6-axes FTS can be inserted in the EOG representing the kinematic chain: the graph is divided into two sub-graphs, and two ⧫ (i.e. two known wrenches), one on each sub-graph, are used to represent the FT measure. In detail, suppose that an FTS is placed in the i_S-th link. Let $\langle s \rangle$ be the frame associated to the sensor. The sensor "divides" the link i_S into two "sub-links" (hereafter denoted forward and backward sub-links). Therefore the sensor measures the wrench exerted by the "forward" sub-link to the "backward" sub-link (this will be represented by a first rhomboidal node). However, a wrench equal and opposite to the sensor measurement is also exerted by the "backward" sub-link to the "forward" sub-link (this will be represented by a second rhomboidal node). Under these considerations, the FTS within a link is represented by splitting the node associated to the link into two sub-nodes, represented by a double circled node in the EOG symbolism. A sub-node has suitable dynamical properties, since it corresponds to a "sub-link", i.e. to a portion of the original link which is physically cut after the insertion of the FTS. As an example, consider the i_S-th link divided in two halves by an FTS: mass m_{i_S} is divided in $m_{i_S}^F$ and $m_{i_S}^B$, and the original Center Of Mass (COM) C_{i_S} is replaced by two COM, $C_{i_S}^F$ and $C_{i_S}^B$ for the two sub-links. The overall procedure is shown in Fig. 3.

D. Computing the system dynamics from FTS measures

We now describe how to solve the dynamics of an open (multiple branches) chain exploiting the information from a single inertial sensor and multiple FTS. The fundamental step consists in enhancing the oriented graph adding the known and unknown kinematics and dynamics variables within the chain, and in particular to split the graph into two subgraphs anytime a FTS is inserted. External forces can be computed accurately if their contact location is measured by suitable tactile sensors or otherwise known a priori. Given a graph description of the chain, we perform the following operations to enhance the graph:

1) insert a ▼ and an edge directed to the link where the inertial sensor is attached to, to represent its measurements
2) insert a ▽ and an edge directed to each "terminal"node
3) for each FTS define the sub-links nodes (with suitable dynamical properties); define two ⧫ associated to the sensor measurements, and connect them to the sub-nodes as described in the procedure in Fig. 3;
4) for each unknown wrench, acting on the generic i-th link, insert a ◊ with an associated edge connected to v_i;
5) rearrange each subgraph so that each ◊ is the root of a tree: for each leaf node (each node connected to only another node) insert a ⧫ with null wrench associated (notice that this node is not influencing the system dynamics since it is associated to a null wrench). At this point, leaves are all ⧫.

The EOG is now complete. The first two steps define the *kinematic EOG*, which is used to compute the kinematics of chain. Precisely, given the measurements of ▼, linear and angular velocities and accelerations can be computed for each joint recursively following a *pre-order* traversal of the tree, and simply applying the kinematic step of the RNEA.

The last three steps define the *dynamic EOG*, used to compute either externally applied wrenches and internal wrenches and joint torques. Considering each EOG subgraph independently, wrenches can be propagated from the leaves (which at this point can be only ⧫, i.e. measured or known wrenches) to the root (which is the unknown wrench). More precisely, the propagation of the dynamics information follows a *post-order* traversal of the tree, with the elementary operations defined by the dynamic step of the RNEA. Given N FTS distributed on the chain, $N+1$ graphs are produced and therefore a maximum of $N+1$ external wrenches can be estimated (one for each sub-graph). Remarkably, in the considered case (one ◊ per subgraph at maximum) each edge in the subgraph is visited during the tree visit, thus all the internal wrenches are computed and therefore a complete characterization of the system dynamics is retrieved. Indeed, once the i-th wrench is known, then the i-th joint torque τ_i can be computed with the usual formula [19]:

$$\tau_i = \mu_i^\top z_{i-1} \tag{3}$$

where z_{i-1} is the z-axis of the reference frame $\langle i-1 \rangle$.

III. iCub whole body dynamics

The method described in the previous sections is generic, i.e. can be applied to a variety of robotic platforms. In this work, it has been specifically implemented for the 53 DOF humanoid robot iCub. iCub is equipped with a 3D Orientation Tracker (Xsens MTx-28A33G25) at the top of the head, and four custom-made 6-axes FTS [7], one per leg and arm, each placed proximally. Excluding the hands, 32 DOF have been taken into account. Given the sensors position and the description of the robot kinematics, it is quite easy to build the kinematics and dynamics EOG: 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, the

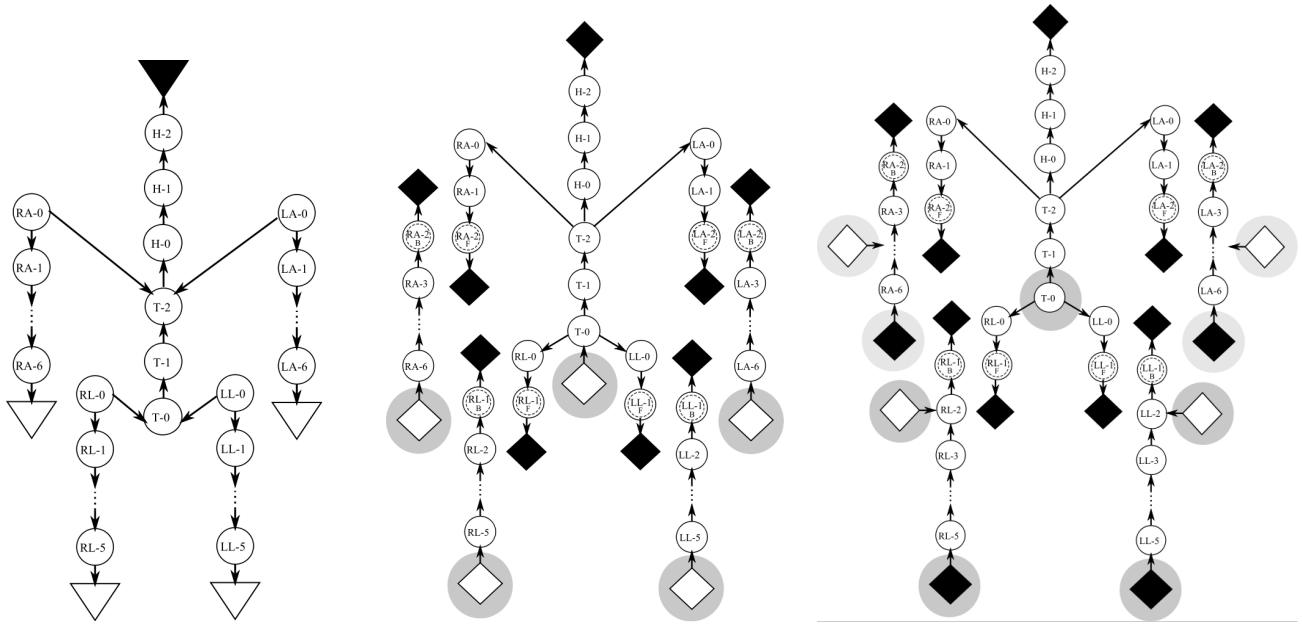

Figure 4. Representation of iCub's kinematic and dynamic EOG. Left(4(a)): iCub's kinematic EOG. It is noticeable that the inertial sensor measure (▼) is the unique source of kinematic information for the whole branched system. Center (4(b)): iCub's dynamic EOG, when iCub is standing on a mainstay and moving freely in the space. Given the four FTS, the main graph is cut by the four links hosting the sensors, and a total of five subgraphs 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 ◊. Right (4(c)): iCub's dynamic EOG, when the iCub is crawling like a baby, as shown in Figure 1. As in the previous case, five subgraphs 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.

kinematic EOG shown in Figure 4(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 subgraphs, 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, 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. 1), external wrenches must be assumed on wrists and knees[3].

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 (visual feedback can be exploited to predict possible contact situations, but not for generic applications). However, thanks to the artificial tactile skin [16] 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 subgraph, the external force due to contact is the unknown ◊, while wrenches located at the end-effectors are assumed to be known and null (⧫).

[3]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).

IV. EXPERIMENTS

Three experiments are reported hereinafter. First, the dynamical model is validated by comparing measurements from the FTS with their model-based prediction. Second, an external commercial FTS has been used to produce known external wrenches, by manually pushing the sensor itself on specific locations (e.g. the wrist): then the external sensor measurements have been compared with the estimation predicted by the EOG. Finally, joint torques computed by (3), after the solution of the EOG, have been compared with a joint torque estimate obtained by projecting a known wrench (once again measured with the external FTS) on the joints.

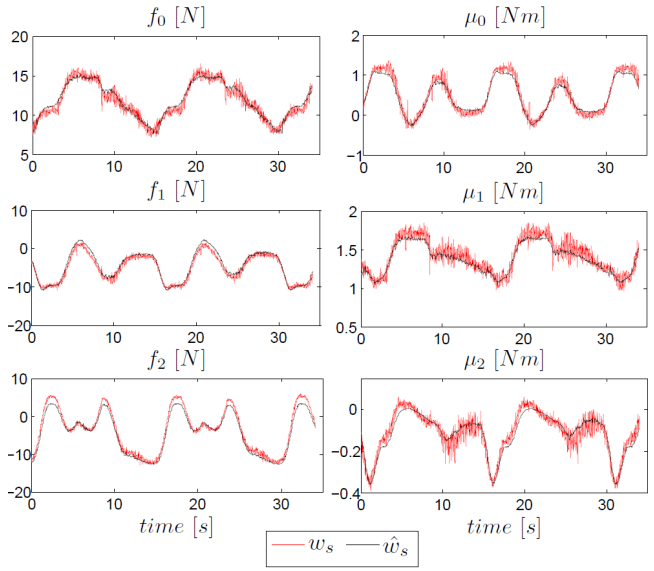

Figure 5. Left arm: comparison between the wrench measured by the FT sensor and the one predicted with the model, during the "Yoga" demo.

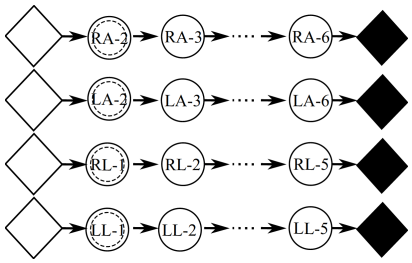

Figure 6. Enhanced graphs for predicting the four FT sensor measurements, $\hat{w}_s$ (◊), when the external wrench acting at the end-effectors (hands and feet) is known (⧫), typically null.

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}
$\bar{\epsilon}$	-1.6678	3.4476	-1.5505	0.4050	-0.7340	0.0171
σ_ϵ	3.3146	2.7039	1.7996	0.3423	0.7141	0.0771

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}
$\bar{\epsilon}$	0.2941	-5.1476	-1.9459	-0.3084	-0.8399	0.0270
σ_ϵ	1.8031	1.8327	2.3490	0.3365	0.8348	0.0498

*: $\epsilon \triangleq \hat{w} - w = [\epsilon_{f_0}, \epsilon_{f_1}, \epsilon_{f_2}, \epsilon_{\mu_0}, \epsilon_{\mu_1}, \epsilon_{\mu_2}]$
*: SI Unit: $f : [N], \mu : [Nm]$.

Table I
ERRORS IN PREDICTING FT MEASUREMENT (SEE TEXT FOR DETAILS)

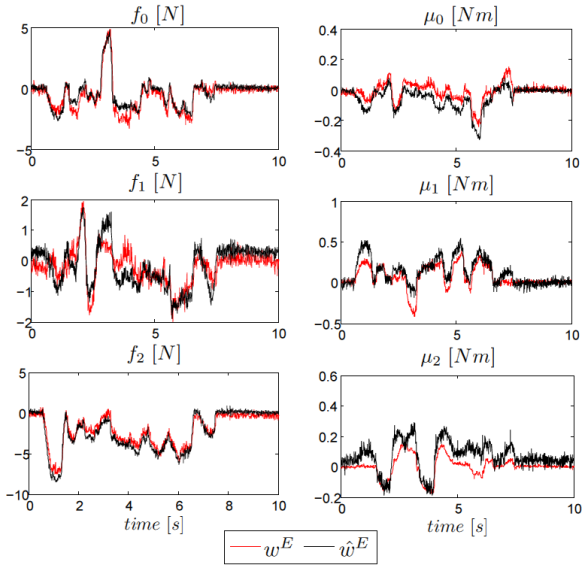

Figure 7. Left arm: comparison between the external wrench estimated after the FT sensor measurements and the one measured by an external FT sensor, manually placed on the palm of the left hand. Besides the obvious measurement noise, the displacement between the estimate and the measure, evident in some peaks, must be attributed to the fact that the human was manually pushing the external FT sensor on the palm while the end-effector was moving, thus the alignment between the axis of the two sensors may have changed because of human error, and so the position of the frame representing the external contact.

A. Validation of the Dynamical Model

To validate the dynamical model, the wrenches w_s from the four six-axes FTS embedded in the limbs are compared with the analogous quantities $\hat{w}_s$ predicted by the dynamical model, during unconstrained movements (i.e. null external wrenches). A rigid-body dynamic model is used to describe the whole robot. Kinematics and dynamics parameters are retrieved from the CAD model of the robot. 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 are assumed, because of the absence of contact with the environment. In practice, this operation can be represented with the EOG in Fig. 6. Table I summarizes the statistics of the errors $w_s - \hat{w}_s$ for each limb during a standard, periodic sequence of movements, during which the robot is supported by a rigid metallic mainstay, and all the limbs are moving freely without collision with the robot own body or the environment. Table I shows the mean and the standard deviation of the errors between measured and predicted sensor wrench during the movements. Fig. 5 shows a comparison between w_s and $\hat{w}_s$ for the left arm (only one limb out of four is shown without loss of generality).

B. Estimation of an external wrench

When solving the dynamic EOG, it is possible to retrieve one external wrench per subgraph. Thus, in the second experiment, we show the effectiveness of our procedure for computing unknown external wrenches due to contacts. Thus,

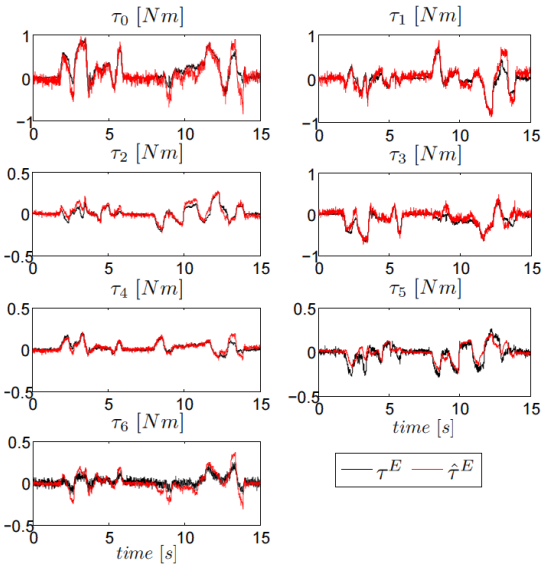

Figure 8. Left arm: comparison between the torques computed exploiting the FT sensor and the ones obtained by projecting the external FT sensor on the joints.

we compare 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 is manually "pushed" on the terminal link. In particular, a wrench w^E is exerted on the left hand and measured with the external FTS. Its value is then compared with $\hat{w}^E$, the estimation of the external wrench obtained by propagating the embedded FTS measure in the subgraph until the frame corresponding to the application point of w^E. A plot of w^E and $\hat{w}^E$ is reported in Fig. 7.

C. Estimation of torques

As a counter evidence of the reliability of the method, we compare the torques $\hat{\tau}$, determined with (3) 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. The external force projected on joints, τ^E, instead is not affected by the internal dynamics (e.g. the gravitational component in this specific static case). Figure 8 shows a comparison of the variation of torque, due to an external wrench application. In particular, we show the comparison between τ^E and $\hat{\tau}^E = \hat{\tau} - \hat{\tau}^I$.

V. Discussion

1) Virtual JTS: The proposed method is particularly convenient because it can be successfully applied to existing robots, which are not equipped with Joint Torque Sensors (JTS), without requiring a significant redesign of their mechanical structure, at a limited cost (at minimum the FTS). In iCub, passive solutions were not considered at design stage [21], therefore active strategies are the only solutions to achieve force control. Analogously to other humanoid robots, iCub is not equipped with JTS, and the joint torques cannot be either estimated by measuring the current absorbed by motors[4]. In such cases, the most adopted solution to provide a force feedback consists in modifying the motor/joint group in order to insert strain-gauge based torque sensors (e.g. in [10], for a Stanford manipulator). Unfortunately, providing pervasively JTS in the existing platform is an expensive solution, which would require a complete makeover of its mechanical structure. Indeed, the new mechanical structure featuring such sensors is currently under design, and so far only preliminary studies on the arm have been presented [13]. With the proposed method, only few sensors are needed to provide iCub with access to its dynamics, which can be added on the existing platform while preserving its original mechanical structure. Further analysis comparing the pros/cons of the two solutions for a prototype of the arm can be found in [17].

2) On computation: The main advantage of the active regulation over the passive one is the possibility of regulating forces within a wider range of values. One disadvantage is the response delay of the regulator, which typically limits the bandwidth of the controlled system. Once the i-th wrench is known, the i-th joint torque τ_i can be computed by (3). Estimated joint torques can be then used in different active force control strategies, but of course the EOG must be solved first in order to compute the link wrenches first, and typically these operations are performed in a remote PC and not directly on the boards controlling the joints.

3) A better representation: We remark that JTSs alone do not provide a complete perceptual representation of the forces arising in consequence of contacts. Indeed, they yield a single component of the manipulator dynamics and are affected by structural singularities. This highlights again the benefit of the EOG approach exploiting 6 axis FTS, which provide a full representation of the internal dynamics of the system and its interaction with the environment.

4) On tactile feedback: Thanks to the tactile feedback, the main feature of the EOG is that it is built (and solved) dynamically: thus, the robot internal dynamics and the external wrenches caused by physical interactions can be computed on the fly, without making any hypotheses about the location of the contacts. If a tactile feedback is missing, than the EOG can be typically built *a priori*, since inertial and FTS are embedded in the mechanical structure of the manipulator (i.e. fixed). In this case, unknown external wrenches can be manually set in the structure, depending on the task: e.g. in hands during manipulation tasks, wrists and knees during crawling.

5) Open-source library: A software library (iDyn) for performing these computations has been released with an open

[4]This solution is feasible only when most of the motor torque is transmitted to the joint (low friction). Therefore, highly efficient transmissions need to be employed (see for example [9]) which is intrinsically incompatible with compact solutions such as planetary or harmonic drive gear trains.

source license [1], and included in the iCub software.

6) Applications: The effectiveness of the algorithm has been proved in several experiments with the iCub: in fact, the computation of the so called "virtual joint torques" by means of the iDyn library enabled different forms of active force control, ranging from torque to impedance control, not to mention the possibility to detect external forces, thus allowing a compliant interaction with the environment and the humans cooperating with the robot. Numerous videos of realized applications can be found at http://www.youtube.com/robotcub.

VI. Conclusions and Future Works

We presented a method for exploiting measurements from multiple sensors distributed on an open (multiple branches) kinematic chain, which provides a complete representation of the dynamics of a robot. The theoretical framework presented in this paper can take into account information coming from different sets of sensors, such as force/torque, inertial, tactile. A graphical representation of the sensory information of the robot, called EOG, combined with standard dynamics algorithms allows propagating wrench measures and computing both joints torques and external wrenches due to contacts, applied at known locations. It was also shown that given N-FT sensors, a maximum of $N + 1$ external wrenches can be estimated. In the future, we plan to integrate the numerous additional sensors (inertial measures from the boards, JTS from the main joints in the limbs [17]) in order to have a more accurate representation of the dynamics of the robot, particularly by means of sensor fusion algorithms. Moreover, having more measurements would notably improve the estimation of the dynamic parameters of the rigid-body dynamics model of the robot.

References

[1] Doxygen documentation of the iDyn library. http://eris.liralab.it/iCub/main/dox/html/group__iDyn.html.

[2] E. Colgate and N. Hogan. *The Interaction of Robots with Passive Environments: Application to Force Feedback Control.* Advanced Robotics. Springer-Verlag, 1989.

[3] R.S. Dahiya, G. Metta, M. Valle, and G. Sandini. Tactile sensing from humans to humanoids. *IEEE Trans. on Robotics*, 26:1–20.

[4] O. Eiberger, S. Haddadin, M. Weis, A. Albu-Schäffer, and G. Hirzinger. On joint design with intrinsic variable compliance: derivation of the DLR QA-joint. In *IEEE Int. Conf. Rob. Autom.*, 2010.

[5] R. Featherstone. Exploiting sparsity in operational-space dynamics. *Int. Journ. Robotics Research*, 2010.

[6] R. Featherstone and D.E. Orin. Dynamics. In *Springer Handbook of Robotics*, pages 35–65. 2008.

[7] M. Fumagalli, M. Randazzo, F. Nori, L. Natale, G. Metta, and G. Sandini. Exploiting proximal F/T measurements for the iCub active compliance. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2010. Taipei, Taiwan.

[8] S. Haddadin, A. Albu-Schäffer, A. De Luca, and G. Hirzinger. Collision detection and reaction: A contribution to safe physical human-robot interaction. In *IEEE/RSJ Int. Conf. Intell. Robots and Systems*, 2008.

[9] Barret Technology Inc. The WAM arm from barret technology. http://www.barrett.com/robot/products-arm.htm.

[10] J.Y.S. Luh, W.D. Fisher, and R.P.C. Paul. Joint torque control by a direct feedback for industrial robots. *IEEE Trans. on Automatic Control*, 28(2):153–161, 1983.

[11] G. Morel and S. Dubowsky. The precise control of manipulators with joint friction: A base force/torque sensor method. In *IEEE Int. Conf. on Robotics and Automation*, pages 360–365, 1996.

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

[13] A. Parmiggiani, M. Randazzo, L. Natale, G. Metta, and G. Sandini. Joint torque sensing for the upper-body of the iCub humanoid robot. In *IEEE-RAS Int. Conf. on Humanoid Robotics*, France, 2009.

[14] U. Pattacini, F. Nori, L. Natale, G. Metta, and G. Sandini. An experimental evaluation of a novel minimum-jerk cartesian controller for humanoid robots. In *IEEE/RSJ Int. Conf. Intell. Robots Syst.* Taiwan, 2010.

[15] G. Pratt and M. Williamson. Series elastic actuators. In *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, USA, 1995.

[16] A. Del Prete, S. Denei, L. Natale, F. Mastrogiovanni, F. Nori, G. Cannata, and G. Metta. Skin spatial calibration using force/torque measurements. In *IEEE/RSJ Int. Conf. Intell. Robots Sys.*, San Francisco, CA, USA, 2011.

[17] M. Randazzo, M. Fumagalli, F. Nori, L. Natale, G. Metta, and G. Sandini. A comparison between joint level torque sensing and proximal f/t sensor torque estimation: implementation on the icub. In *IEEE/RSJ Int. Conf. Intell. Robots Sys.*, San Francisco, CA, USA, 2011.

[18] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi. An atlas of physical human-robot interaction. *Mechanism and Machine Theory*, 43(3):253–270, March 2008.

[19] L. Sciavicco and B. Siciliano. *Modelling and Control of Robot Manipulators*. Springer, 2005.

[20] B. Siciliano and L. Villani. *Robot Force Control*. Kluwer Academic Publishers, Norwell, MA, USA, 2000.

[21] N.G. Tsagarakis, G. Metta, G. Sandini, D. Vernon, R. Beira, J. Santos-Victor, M.C. Carrazzo, F. Becchi, and D.G. Caldwell. iCub - the design and realization of an open humanoid platform for cognitive and neuroscience research. *Int. Journal of Advanced Robotics*, 21(10): 1151–75, 2007.

[22] 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 *IEEE/RSJ Int. Conf. Intell. Robots Sys.*, St. Louis, MO, USA, 2009.

[23] J. Wittenburg. Topological description of articulated systems. *Computer-Aided Analysis of Rigid and Flexible Mechanical Systems*, pages 159–196, 1994.

[24] K. Yamane and Y. Nakamura. Parallel o(log n) algorithm for dynamics simulation of humanoid robots. In *IEEE Int. Conf. Humanoids Rob.*, 2006.