wholeBodyInterface*, an open-source software abstraction layer for whole-body motion control

S. Traversaro, F. Romano, N. Kuppuswamy,
A. Del Prete, J. Eljaik, N. Lorenzo, F. Nori

September 25, 2015

We propose a C++ open-source software API that aims at standardising the software development for control, estimation and identification of free-floating robots. The key idea is to provide a software abstraction layer that can be used on any robot for which the layer has been implemented. Our API is called `iWholeBodyInterface` and it is divided into four parts: `iWholeBodySensors`, `iWholeBodyActuators`, `iWholeBodyModel` and `iWholeBodyStates`. The first interface, `iWholeBodySensors`, contains methods to access the sensors of the robot. Similarly, `iWholeBodyActuators` offers methods to command the actuators. Different low-level control modes are supported: position control, velocity control, torque control. The third interface, `iWholeBodyModel`, provides access the the kinematic and dynamic model of the robot, for instance: forward kinematics, Jacobians, inverse dynamics, mass matrix. Finally, `iWholeBodyStates` is the API for an estimator: since controllers typically need quantities for which no sensor may be available (e.g. joint velocities) this interface provides a standard way to get them.

YARP based implementation. We provided a C++ implementation[1] of the `iWholeBodyInterface` introduced in the previous section. The `iWholeBodyActuators` and `iWholeBodySensors` classes have been implemented to interface with YARP controlboards. The `iWholeBodyModel` class, instead, computes kinematic and dynamic quantities by using the iDynTree[2] dynamic library. The user is required to provide a URDF model of the robot, and to write a configuration file specifying how the joints are mapped to the YARP controlboards. It is also possible to specify the presence of inertial measurement units and force/torque sensors. The configuration file is already provided for different humanoids including iCub, CoMan and Armar4.

*Code `https://github.com/robotology-playground/wholeBodyInterface` and documentation `http://wiki.icub.org/codyco/dox/html/classwbi_1_1wholeBodyInterface.html`.

[1] `https://github.com/robotology-playground/yarp-wholebodyinterface`

[2] `https://github.com/robotology/idyntree`

iWholeBodyInterface and Matlab. Two sets of C++ files have been created to compile MEX files to be called as built-in functions in MATLAB[3] and SIMULINK[4], respectively. Utilising the same structure as the other iWholeBodyInterface components, the toolbox is able to automatically initialise the interface and the rapid visual prototyping of different control schemes is enabled through the natural flexibility of MATLAB SIMULINK; other pre-existing toolbox components may easily be used in concurrence as well. These tools have so far played a critical role in advancing the state-of-art in the development of a stable whole body balancing controller and state estimator on the iCub humanoid robot [5]. The WBI-Toolbox[6] wraps the iWholeBodyInterface C++ implementation, acting as a higher abstraction layer that simplifies the design, debugging and final implementation of whole-body controllers. We like to call it *rapid controllers' prototyping*. Synchronization with YARP time is also possible when using the Gazebo simulator or the real platform. The user has the advantage of using online the numerous Simulink Toolboxes.

Estimation as implemented in iWholeBodyStates. We use an instance of the iWholeBodySensors interface to implement multi-sensor fusion techniques for whole body control (as the one described in [1]), to abstract from the middleware-specific way in which sensor data are published. For the iCub robot this permits to access transparently a given accelerometer or gyroscope sensor, regardless if their measure is published as part of an IMU measure or as a standalone measure. The proposed solution enables great flexibility in defining the estimation algorithm independently from the set of available sensors.

wholeBodyInterface at Humanoids 2015. We will introduce the basic concepts of the wholeBodyInterface dividing the presentation[7] in a theoretical and a practical session, the latter including some live demos and step-by-step tutorials on simulated robots.

References

[1] Francesco Nori, Naveen Kuppuswamy, and Silvio Traversaro. Simultaneous state and dynamics estimation in articulated structures. In *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2015)*, 2015.

[3] https://github.com/robotology-playground/mex-wholebodymodel

[4] https://github.com/robotology-playground/WBI-Toolbox

[5] https://www.youtube.com/watch?v=VrPBSSQEr3A

[6] https://github.com/robotology-playground/WBI-Toolbox

[7] https://github.com/iron76/wholeBodyInterface-abstract/raw/master/slides/wbi_and_toolbox.pdf