Whole Body Interface Toolbox (WBI-T): A Simulink Wrapper for Robot Whole Body Control

Jorhabib Eljaik, Andrea del Prete, Silvio Traversaro, Marco Randazzo, Francesco Nori

Istituto Italiano di Tecnologia. Department of Robotics, Brain and Cognitive Sciences (RBCS)
Overview

- Motivations
- Features and description
- Dependencies and System Requirements
- Whole Body Interface
- Implementing a controller (Demo)
- Working with the real platform
Under the Hood

The YARP Middleware

"It is a set of libraries, protocols, and tools to keep modules and devices cleanly decoupled. It is reluctant middleware, with no desire or expectation to be in control of your system"
Motivations

■ "Rapid prototyping" of controllers.
■ Better alternative to the use of YARP JAVA bindings
■ Exploitation of Simulink and MATLAB toolboxes
■ Higher level of abstraction for humanoid robot interfaces.
■ Motivate non-programmers roboticists to approach the real platform.
Description & Features

Simulink blocks wrapping a YARP-based implementation of the Whole-Body Interface (WBI) [1] C++ library. WBI acts as an abstraction layer for any interaction with the robot, making code robot-independent.

- Easy interface with YARP based humanoid robots.
- Instantaneous transfer of simulation results onto the real platform.
- Deals with both fixed and floating base humanoids.
- Synchronization with YARP! Important aspect in the design and use of controllers in simulation.
- Supported OS: Linux, Mac OS X, Windows.

Whole Body Interface Toolbox

MEX-Files dynamically linked C/C++ code and libraries → MATLAB Interpreter load and executes → SIMULINK S-function
Whole Body Interface Toolbox

YARP

wholeBodyInterface

gazebo_yarp_plugins

WBI-Toolbox
**Dependencies and System Requirements**

- **YARP** - Robotics middleware.
- **iCub Software** - Not strictly necessary
- **CoDyCo Software**
  - **iDynTree Library**
    - YARP based Robot dynamics library
  - **wholeBodyInterface Library**
    - Library defining a general interface for communicating with a floating-base rigid robot
- **Gazebo** or **iCubSim** simulator.
Dependencies and System Requirements

- **YARP** - Robotics middleware.
- **iCub Software** - Not strictly necessary
- **CoDyCo Software**
  - **iDynTree Library**
    - YARP based Robot dynamics library
  - **wholeBodyInterface Library**
    - Library defining a general interface for communicating with a floating-base rigid robot
- **Gazebo** or **iCubSim** simulator.
Under the Hood

wholeBodyInterface Library

C++ template library defining a general interface for communicating with a floating-base rigid robot.
Under the Hood

wholeBodyInterface Library

- **iWholeBodyActuators**
  - Class
  - send commands to the low-level motor controllers

- **iWholeBodyModel**
  - Class
  - access the kinematic/dynamic model of the robot
  - The robot has \( n \) joints and \( n+6 \) DoFs

- **iWholeBodyStates**
  - Class
  - read estimations of the state of the robot

- **iWholeBodySensors**
  - Class
  - read sensor data (e.g., encoders, force/torque sensors, IMUs)

- **wholeBodyInterface**
  - Class
  - public
  - includes:
    - iWholeBodyStates
    - iWholeBodyModel
    - iWholeBodyActuators

WBI-Toolbox 11
Implementing a Controller

Toolbox's main screen and subsections.
Implementing a Controller

Whole Body Impedance Controller quickly implemented on Matlab

\[-K_p(q_j - q_{j0}) - K_d \dot{q}_j + g = \tau_j\]
The Real Platform

The Whole Body Controller in the following video is being run on Simulink
http://www.youtube.com/watch?v=jaTEbCsFp_M
More Information

Installation instructions:
http://github.com/robotology/codyco/tree/master/src/simulink

Contact:
jorhabib.eljaik@iit.it
francesco.nori@iit.it