iTaSC as a unified framework for task specification, control, and coordination, demonstrated on the PR2

Dominick Vanthienen*, Tinne De Laet*, Wilm Decré*, Ruben Smits*, Markus Klotzbücher*, Koen Buys*, Steven Bellens*, Luca Gherardi†, Herman Bruyninckx*, and Joris De Schutter*

Keywords: mobile manipulation, task specification, shared control, software, PR2, iTaSC

I. INTRODUCTION

This paper describes the implementation of the instantaneous Task Specification using Constraints (iTaSC)-Skill on a PR2. The framework allows easy specification and code-generation for robot tasks. Its power will be demonstrated by a mobile co-manipulation task of a human and a PR2 robot. The PR2 has to follow the instructions of the human and help him carrying an object, while avoiding obstacles and unnatural poses. Although the framework and its implementation are demonstrated on a PR2, it can be used for any robotic system. The software, including the demonstration is publicly available under an open-source license.

Robots have evolved from simple six degree-of-freedom industrial robots in a cage, to mobile platforms with redundant arms and multiple sensors, for example the PR2. These mobile platforms have escaped from their cages to a domestic, cluttered and populated environment. The resulting increase in complexity and uncertainty shows the need for a highly modular methodology for motion and task specification and coordination.

This paper demonstrates the capabilities of such a highly modular methodology, applied on the PR2. The demonstration includes a mobile co-manipulation task of a human and a PR2 robot, in which they handle an object together trough a cluttered and populated environment. The PR2 should follow the instructions of the human and assist him carrying the object, while avoiding obstacles. Figure 1 shows an artist's impression of the co-manipulation task.

II. iTA-SC-SKILLS FRAMEWORK IN A NUTSHELL

The unified framework presented is based on iTaSC, or Instantaneous Task Specification using Constraints, which is developed at the K.U.Leuven during the last years [3], [4], [7]. The framework generates motions by specifying constraints in geometric, dynamic, or sensor-space between the robots and their environment. These motion specifications constrain the relationships between objects (object frames) and their features (feature frames). Established robot motion specification formalisms such as the Operational Space Approach [5], the Task Function Approach [8], the Task Frame Formalism [6], Cartesian Space control, and Joint Space control are special cases of iTaSC and can be specified using the generic iTaSC methodology.

The key advantages of iTaSC over traditional motion specification methodologies are: (i) composability of constraints: multiple constraints can be combined, hence the constraints can be partial, i.e. they do not have to constrain the full 6D relation between two objects; (ii) reusability of constraint specification: the constraints specify a relation between feature frames, which have a semantic meaning in the context of a task, implying that the same task specification can be reused on different objects; (iii) automatic derivation of the control solution: the iTaSC methodology generates a robot motion that optimizes the constraints by automatically deriving the controllers from the constraint specification.

While the framework will be demonstrated on the PR2, it can be used for any robotic system, with a wide variety of sensors.

Skills are responsible for the coordinated execution of tasks and the parameter configuration of different instantaneous
motion specifications. Consequently, the framework separates the continuous level motion specification and discrete level coordination. One skill coordinates a limited set of constraints, which together form a functional motion. Finite State Machines implement the skill functionality.

III. SOFTWARE

The iTaSC software implements the aforementioned framework in Orocos, which is integrated in ROS by the Orocos-ROS-integration [9]. The Real-Time Toolkit (RTT) of the Orocos project enables the control of robots on a hard-realtime capable operating system, e.g., Xenomai-Linux or RTAI-Linux [2]. The rFSM subproject of Orocos allows scripted Finite State Machines, hence Skills, to be executed in hard realtime. Figure 2 shows the software architecture, mentioning (as a clarification) the key formulas for the resolved velocity case without prioritization. The key advantages of the software design include: (i) the modular design, allowing users to implement their own solver, scene graph, motion generators, . . . ; (ii) the modular task specification, allowing users to reuse tasks, and enabling a future task-web application to down- or upload tasks; and (iii) the flexible user interface, allowing users to change the weights and priorities of different constraints, and to add or remove constraints. Furthermore, the Bayesian Filtering Library (BFL) and Kinematics and Dynamics Library (KDL) of the Orocos project are used to retrieve stable estimates out of sensor data and to specify the kinematic chains (the ones of the robots and the virtual ones of the task specification), respectively.

IV. DEMONSTRATION

We will demonstrate a human-PR2 co-manipulation task consisting of four Skills: (i) co-manipulation of an object with the operator, (ii) dynamic and static obstacle avoidance, (iii) maintain visual contact with the operator, and (iv) unnatural pose of the PR2 prevention. In order to demonstrate the aforementioned capabilities of the iTaSC-Skill framework and software, everyone will be allowed to perform the co-manipulation with the PR2, change the weights and priorities of constraints, enable and disable constraints . . . A Java-based Graphical User Interface (GUI) will be at the operator’s disposal, to interact with and monitor the state of the robot. This GUI demonstrates the possibilities of the Java-Orocos integration, JORocos [1].

A beta version of the software, including the demonstration is publicly available under an open-source license on http: //git.mech.kuleuven.be/robotics/itasc.git.

V. LESSONS LEARNED

The results of the demonstration on the PR2 suggest that the performance and functionality of the real time motion control of arms and mobile base of the PR2 arms can be increased. The iTaSC-Skill software support is available under an open-source license. Its capabilities are demonstrated on a PR2, but can be used for any robotic system. We are eager to share our code and appreciate feedback on it.

Additional information including a discussion on the iTaSC-Skill software design and a teaser movie is available at http://people.mech.kuleuven.be/~dvanthienen/IROS2011PR2/.

REFERENCES