SwarmSimX and TeleKyb: Two ROS-integrated Software Frameworks for Single- and Multi-Robot Applications

Johannes Lächele¹, Martin Riedel, Paolo Robuffo Giordano², and Antonio Franchi¹

In this note we briefly review two software frameworks that have been developed within the Autonomous Robotics and Human-machine Systems group (formerly, Human-Robot Interaction group) at the Max Plank Institute for Biological Cybernetics. Both frameworks, starting from the early versions up to the most recent releases, have been successfully employed in several works of the group, including, e.g., [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12]. For a detailed description of SwarmSimX, we refer the interested reader to [13], [14]. The TeleKyb framework is instead described thoroughly in [15] and available at http://www.ros.org/wiki/telekyb.

I. SwarmSimX

Software frameworks simulating the behavior of virtual environments are an indispensable tool in most engineering sciences. Within the robotics scope, simulation environments are of paramount importance for fast development and testing of new control algorithms for single robots, or of complex behaviors for multiple interacting robots.

In this latter case, several software suites able to simulate multiple robots at the same time have been developed and are widely used in research. Simulators like ARGoS [16] are capable of handling multiple robots with a pure modular software design that allows for assigning different physics engines to different areas of the simulation. A simulation example involving thousands of robots is discussed, albeit only in a 2D environment. Also, the design of ARGoS is not specialized for real-time (RT) simulation, an essential feature for hardware-in-the-loop scenarios and for all those situations involving strict constraints on the inner simulation timing (e.g., whenever requiring online processing/filtering of signals acquired from the external world).

The crucial requirements that we identified for a robotics simulator have been the following: real-time execution, physical realism, exchangeable visual and physical representation, extendable software architecture, and full control over inherent information of all simulated robots (see, e.g., [17], [18], [19]). To the best of our knowledge, we were unable to find a solution meeting all of the requirements.

SwarmSimX is a simulation environment with the ability to simulate dozens of robots in a realistic 3D environment.

II. TeleKyb

The challenges accompanied during software development for a robotic platform have constantly changed over the last couple of years. Initially, robotic code for sensors, actuators or controllers was developed only for a specific hardware architecture, which made the reuse of software unnecessary.
However, the design of current robotic systems has moved to a more modular setup, where individual hardware components can easily be added or exchanged. This development required a fundamental paradigm shift in the software architecture that drives these robots. Robotic middleware solutions have since moved to a thin design that supports the development of modular components and increases the ability to reuse existing code [20]. Several frameworks follow this paradigm [21], [22] with the Robot Operating System (ROS) [23] being one of the most popular.

The Tele-Operation Platform of the MPI for Biological Cybernetics (TeleKyb) is a collection of software frameworks, libraries and tools that provides a standardized interface for developing and testing bilateral teleoperation systems between human interfaces (e.g., haptic devices or touch screens) and (groups of) mobile robots. TeleKyb implements a high-level closed-loop robotic controller for mobile robots that can be extended dynamically with modules for state sensing, planning and actuation of the robot, and a hardware interface layer that abstracts the underlying hardware into the overall framework. Additionally, TeleKyb is able to integrate with higher level controllers and can be directly interfaced through the TeleKyb control layer. Since TeleKyb is completely based on ROS, it can easily utilize functionality provided from thousands of 3rd party ROS packages.

Fig. 2. High-level overview of several TeleKyb components.

References
