Workshop on Towards Fully Decentralized Multi-Robot Systems: Hardware, Software and Integration
2013 IEEE International Conference on Robotics and Automation, ICRA 2013
Karlsruhe, Germany, May 6 - 10, 2013, **http://icra2013mrs.tuebingen.mpg.de/**

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

Johannes Lächele[1], Martin Riedel, Paolo Robuffo Giordano[2], and Antonio Franchi[1]

In this notes 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.

[1]Max Planck Institute for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany. {`johannes.laechele`, `antonio.franchi`}`@tuebingen.mpg.de`
[2]CNRS at IRISA, Campus Universitaire de Beaulieu, 35042 Rennes cedex, France. `prg@irisa.fr`

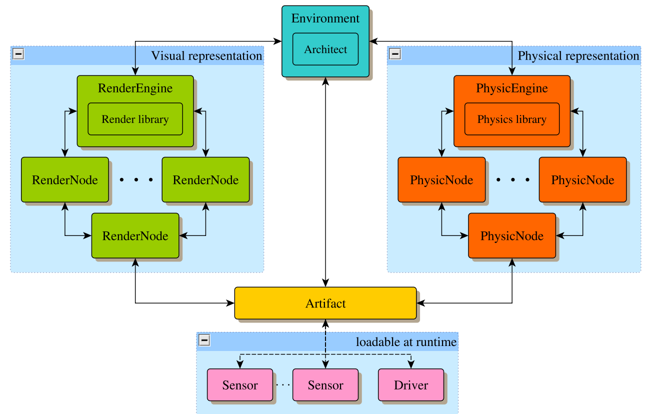

Fig. 1. Overview of the software architecture of SwarmSimX.

The software architecture of SwarmSimX allows new robots, sensors, and other libraries to be loaded at runtime, extending the functionality of the simulation environment significantly. In addition, SwarmSimX allows an easy exchange of the underlying libraries used for the visual and physical simulation to incorporate different libraries (e.g., improved or future versions). A major feature is also the possibility to perform the whole simulation in real-time allowing for human-in-the-loop or hardware-in-the-loop scenarios.

The SSX simulation environment can be divided into three main parts: the visual representation, the physical representation, and *Artifacts*. Figure 1 gives an overview of the elements used in the simulation and how they are related among themselves. In the nomenclature of SSX, the visual representation is managed by the RenderEngine with individual elements being represented by RenderNodes. Symmetrically, the physical representation is managed by the PhysicEngine and the individual parts are called PhysicNodes. Both, Physic- and RenderNodes can be connected to form tree structures. Child nodes are defined w.r.t. the parent node to which they are associated, and may contain information about the position, orientation, mass, and similar quantities.

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.

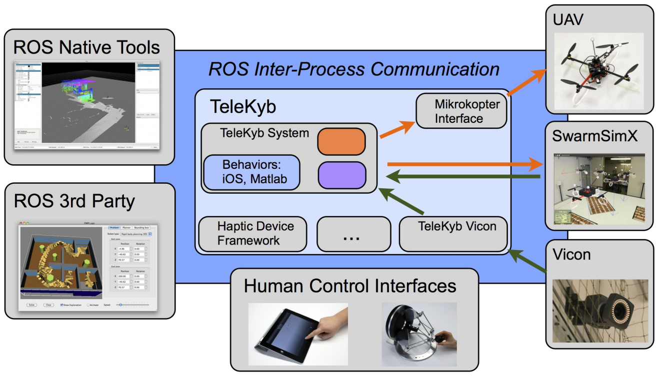

Fig. 2. High-level overview of several TeleKyb components.

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*)[1] 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 estimation, trajectory planning, processing and tracking. It is completely based on the ROS and all of its components are freely available as a public ROS repository.

From a high-level perspective (compare the horizontal structure of Fig. 2), TeleKyb can be divided into a *human interface layer*, representing direct modes of interaction with the human operator, *a control layer* responsible for the 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[2] of 3rd party ROS packages.

References

[1] A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano, "Distributed online leader selection in the bilateral teleoperation of multiple UAVs," in *50th IEEE Conf. on Decision and Control*, Orlando, FL, Dec. 2011, pp. 3559–3565.

[2] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff, "Passivity-based decentralized connectivity maintenance in the bilateral teleoperation of multiple UAVs," in *2011 Robotics: Science and Systems*, Los Angeles, CA, Jun. 2011.

[1] `http://www.ros.org/wiki/telekyb`
[2] `http://www.ros.org/browse`

[3] M. Cognetti, P. Stegagno, A. Franchi, G. Oriolo, and H. H. Bülthoff, "3-D mutual localization with anonymous bearing measurements," in *2012 IEEE Int. Conf. on Robotics and Automation*, St. Paul, MN, May 2012, pp. 791–798.

[4] A. Franchi, C. Secchi, H. I. Son, H. H. Bülthoff, and P. Robuffo Giordano, "Bilateral teleoperation of groups of mobile robots with time-varying topology," *IEEE Trans. on Robotics*, vol. 28, no. 5, pp. 1019–1033, 2012.

[5] A. Franchi, C. Secchi, M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano, "Shared control: Balancing autonomy and human assistance with a group of quadrotor UAVs." *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 57–68, 2012.

[6] D. Zelazo, A. Franchi, F. Allgöwer, H. H. Bülthoff, and P. Robuffo Giordano, "Rigidity maintenance control for multi-robot systems," in *2012 Robotics: Science and Systems*, Sydney, Australia, Jul. 2012.

[7] M. Riedel, A. Franchi, H. H. Bülthoff, P. Robuffo Giordano, and H. I. Son, "Experiments on intercontinental haptic control of multiple UAVs," in *12th Int. Conf. on Intelligent Autonomous Systems*, Jeju Island, Korea, Jun. 2012, pp. 227–238.

[8] R. Spica, A. Franchi, G. Oriolo, H. H. Bülthoff, and P. Robuffo Giordano, "Aerial grasping of a moving target with a quadrotor UAV," in *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vilamoura, Portugal, Oct. 2012, pp. 4985–4992.

[9] C. Masone, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano, "Interactive planning of persistent trajectories for human-assisted navigation of mobile robots," in *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vilamoura, Portugal, Oct. 2012, pp. 2641–2648.

[10] H. I. Son, A. Franchi, L. L. Chuang, J. Kim, H. H. Bülthoff, and P. Robuffo Giordano, "Human-centered design and evaluation of haptic cueing for teleoperation of multiple mobile robots," *IEEE Trans. on Systems, Man, & Cybernetics. Part B: Cybernetics*, vol. 43, no. 2, pp. 597–609, 2013.

[11] D. J. Lee, A. Franchi, H. I. Son, H. H. Bülthoff, and P. Robuffo Giordano, "Semi-autonomous haptic teleoperation control architecture of multiple unmanned aerial vehicles," *IEEE/ASME Trans. on Mechatronics*, In Press.

[12] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff, "A passivity-based decentralized strategy for generalized connectivity maintenance," *International Journal of Robotics Research*, vol. 32, no. 3, pp. 299–323, 2013.

[13] J. Lächele, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano, "SwarmSimX: Real-time simulation environment for multi-robot systems," in *3rd Int. Conf. on Simulation, Modeling, and Programming for Autonomous Robots*, Tsukuba, Japan, Nov. 2012.

[14] J. Lächele, "Development of a real-time simulation environment for multiple robot systems," Master Thesis, Eberhard Karls Universität Tübingen, http://laechele.eu/SwarmSimX/, 2012.

[15] M. Riedel, "Telekyb: A modular software framework for bilateral teleoperation scenarios and its applications in robotics research," Master Thesis, Eberhard Karls Universität Tübingen, 2012.

[16] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, a. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, T. Stirling, a. Gutierrez, L. M. Gambardella, and M. Dorigo, "ARGoS: A modular, multi-engine simulator for heterogeneous swarm robotics," *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5027–5034, Sep. 2011.

[17] J. Craighead, R. Murphy, J. Burke, and B. Goldiez, "A survey of commercial & open source unmanned vehicle simulators," in *Proc. IEEE Int Robotics and Automation Conf*, April 2007, pp. 852–857.

[18] A. L. Alex, T. Bruny, J. Sidman, S. A. Weil, A. Inc, and W. Ma, "From gaming to training: A review of studies on fidelity, immersion, presence, and buy-in and their effects on transfer in PC-based simulations and games," November 2005.

[19] A. Boeing and T. Bräunl, "Evaluation of real-time physics simulation systems," in *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, ser. GRAPHITE '07. New York, NY, USA: ACM, 2007, pp. 281–288.

[20] A. Makarenko, A. Brooks, and T. Kaupp, "On the benefits of making robotic software frameworks thin," in *International Conference on Intelligent Robots and Systems*, vol. 2, 2007.

[21] A. Brooks, T. Kaupp, A. Makarenko, S. Williams, and A. Orebäck, "Orca: a component model and repository," *Software engineering for experimental robotics*, pp. 231–251, 2007.

[22] G. Metta, P. Fitzpatrick, and L. Natale, "Yarp: Yet another robot

platform," *International Journal on Advanced Robotics Systems*, vol. 3, no. 1, pp. 43–48, 2006.

[23] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, vol. 3, no. 3.2, 2009.