CEA comité español de automática

XLIV Jornadas de Automática 2023

XLIV Jornadas de AUTOMÁTICA 2023 – ZARAGOZA
https://jautomatica.es

A generic controller for teleoperation on robotic manipulators using low-cost devices

Łukawski, B.*, Victores, J.G., Balaguer, C.

Robotics Lab, Department of Systems Engineering and Automation, Universidad Carlos III de Madrid, Avda. Universidad, 30, 28911, Leganés, Spain.

To cite this article: Łukawski, B., Victores, J.G., Balaguer, C. 2023. A generic controller for teleoperation on robotic manipulators using low-cost devices.
XLIV Jornadas de Automática , 785-788. https://doi.org/10.17979/spudc.9788497498609.785

Abstract

A usual form of human-robot interaction is the ability of the former to remotely command the latter through any sort of auxiliary device; this interaction is referred to with the term "teleoperation". Robots are common examples of systems that can be controlled remotely. Depending on the task at hand, said systems can grow in complexity and costs. Specifically, the peripherals devoted to controlling the robot could require costly engineering and even an ad hoc design. However, a range of low-cost, commercial devices and controllers, originally intended for other purposes, can also be a good fit for teleoperation tasks in robotics. This work explores a selected collection of popular devices of this kind, and proposes a unified framework to exploit their capabilities as remote controllers for a set of robotic platforms. Their suitability is proven both on real and simulated versions of these platforms through simple experiments that show how they could be further used in more complex scenarios.

Keywords: Robotics, Teleoperation, Software architectures.

---

1. Introduction

Input devices employed in teleoperation for robotics (i.e. telerobotic control) can be characterized according to a set of general specifications. Attending to design, functional and "fit for use" criteria, some of them are: position and force bandwidth, backdriveability, backlash, cross coupling effects, dexterity, control modes (manual, supervised, shared and negotiated control), complexity or cost (Fischer et al., 1990).

From a cost-wise perspective, other works explored cheaper implementations for a full robotic platform involving teleoperation or telepresence. For instance, a low-cost MechRc humanoid robot was proposed to be remotely controlled through a human suit equipped with potentiometers and accelerometers, in such a way that the robot mimicks the motion of the human operator (Cela et al., 2013). In another work, a ROS-enabled iRobot Create mobile platform equipped with an onboard computer and cameras was proposed (Lazewatsky and Smart, 2011). Both solutions rely on open sourcing for cost reduction, as also do two of our own robotic platforms.

The following sections describe a generic controller architecture tested on actual hardware and the simulated counterparts, when available. Section 2 describes the peripherals and platforms used in this work. Section 3 details the controller architecture, while section 4 provides insights on the experiments and explains a workaround that needed to be applied to overcome some limitations. In section 5, conclusions are drawn and related works stemming from this project are outlined.

2. Materials

To demonstrate the purpose of the controller architecture presented in this work, a set of commercial and low-cost peripherals has been selected. Figure 1 depicts each device.

- SpaceNavigator by 3Dconnexion (rebranded as SpaceMouse): a 6 DoF USB joystick most suitable for CAD applications. Translation and orientation motions can be achieved simultaneously. It exhibits a rather limited range of motion, therefore these motions should be interpreted as small displacements (i.e. velocities) in the pose of the commanded entity in 3D space.
- Leap Motion Controller by Ultraleap: a USB device intended for hand and arm tracking that consists of two

---

*Corresponding author: blukawsk@ing.uc3m.es infrared cameras and three infrared LEDs. It can be used in either tabletop (sensors facing upwards) or virtual reality-oriented (sensor mounted on a VR headset pointing towards the front) configurations. The accompanying motion-capture software can detect and track multiple body markers to generate a representation of their position in 3D space using 2D camera data. It also features gesture recognition.

- Wiimote (Nintendo Wii Remote controller): a wireless (Bluetooth) controller that resembles a TV remote device. An internal 3-axis accelerometer allows to capture inertial data. Feedback can be emulated to some extent thanks to its "rumble" feature.

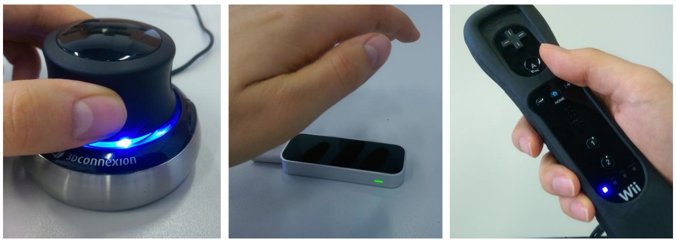

Figure 1: Peripherals. From left to right: SpaceNavigator, Leap Motion Controller, Wiimote.

The controller architecture was tested on a range of manipulator arms and other platforms mainly developed at the Universidad Carlos III de Madrid:

- ASIBOT: a 5 DoF assistive manipulator arm intended for rehabilitation in a household environment (Jardón et al., 2009). Its symmetric layout and the cone-like end effectors on the extremal flanges have been designed so that the robot can attach autonomously to any docking station present in its environment, transferring itself between stations. Also, a wheelchair was adapted and provided with a gear rack to which ASIBOT can be anchored. Motion control is performed on a computer aboard.
- AMOR: a commercial 7 DoF assistive manipulator arm manufactured by Exact Dynamics. It is provided with a gripper tool, and several accessories were added: RGB webcam, depth sensor, and 46 proximity sensors placed all over its casing.
- TEO: a 30 DoF full-sized humanoid robot (Monje et al., 2011) consisting of four limbs (6 DoF each), torso (2 DoF) and head (2 DoF) that can be actuated independently. For the purpose of this paper, any TEO's arm would be effectively treated as a 6 DoF manipulator arm. Motion control is performed on three computers aboard.
- TEO's soft neck prototype: a 3 DoF modular robotic neck intended to replace the current 2 DoF stiff neck by means of a cable mechanism (actuated tendons). It is designed to control the inclination and orientation of a platform attached to the neck's base through a flexible link composed of 3D-printed elements (Nagua et al., 2018a,b; Mena et al., 2020).

Figure 2 depicts the aforementioned robotic platforms.

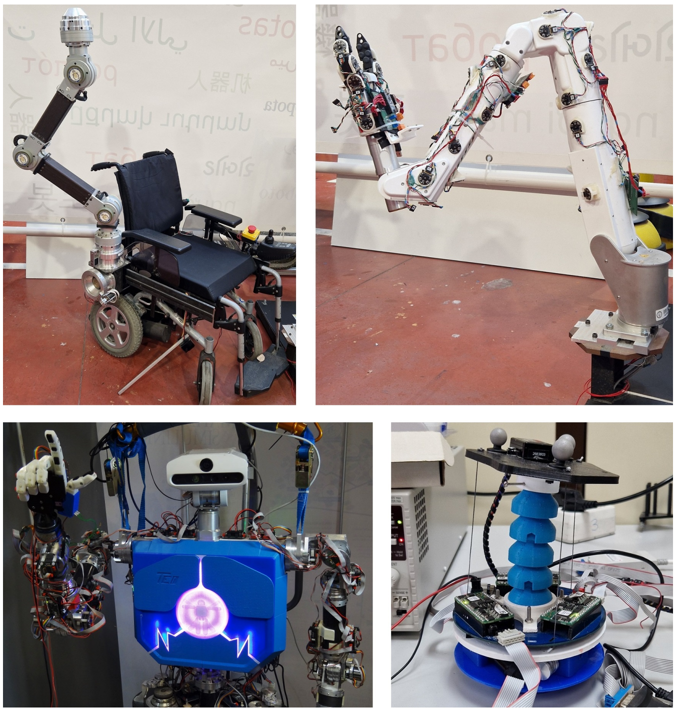

Figure 2: Real platforms. Upper left: ASIBOT, upper right: AMOR, lower left: TEO, lower right: TEO's "soft" neck prototype.

Where applicable, the behavior of the control architecture was first tested in a simulated environment. By taking advantage of the middleware applications involved and modularity, the same controllers and commands can operate either with the simulated robot or the real platform. The availability of a physics engine was not deemed relevant as any motion exerted on the robot would rely on either position or velocity control on the low level (joint controller), not accounting for dynamics.

Two popular robot simulators were selected: OpenRAVE and Gazebo. Figure 3 depicts the virtual models of the three robotic arm platforms in the OpenRAVE simulator.

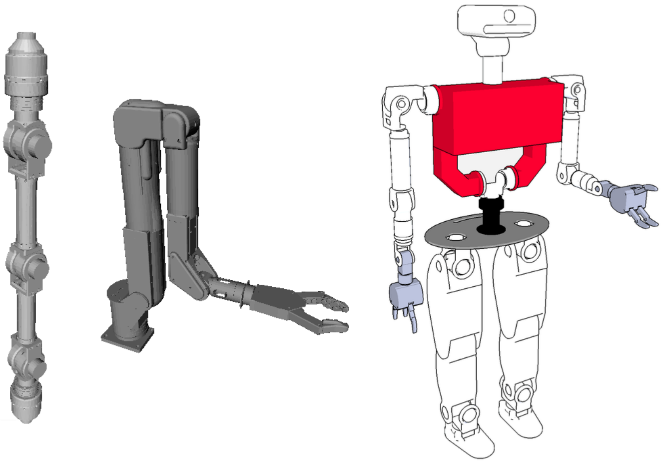

Figure 3: Simulated platforms. From left to right: ASIBOT, AMOR, TEO.

3. Architecture

The proposed teleoperation scheme has been envisioned following the diagram represented in Figure 4. The robot platform is the central actor in the task space along with its (optional) accessories such as cameras and sensors. In the user space, a human operator exerts motion on the selected peripheral and (optionally) monitors the progress of the task on a display if either external or on-robot cameras are available.

On the controller side, a data acquisition layer captures the readings from the peripheral and interprets them as commands conveniently, depending on the desired result (could it be position or velocity commands, in the base or tool frame). Additional information can be queried from the robot, such as signals from the proximity sensors if available. It is also possible to return some sort of feedback (e.g. "rumble" as in the Wiimote device) to the user through the commanding peripheral.

Further down the control layers, a high level controller and a low level controller must interpret the incoming commands to translate them to the Cartesian space (high level control) and joint space (low level control). The output of the low level controller is fed directly to the robot, thus closing the loop. On this stage, setpoints or other sort of information could be displayed in a graphical UI to the user for monitoring purposes.

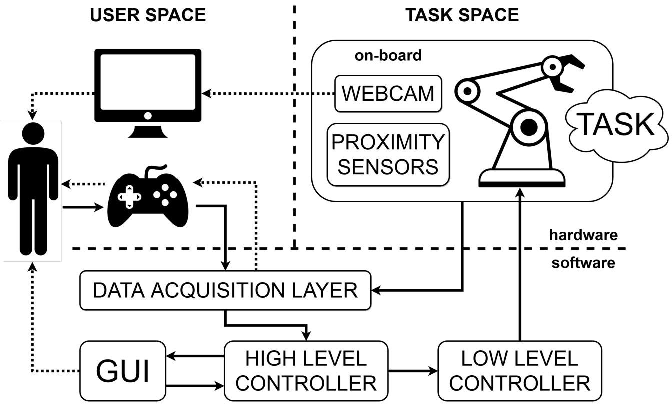

Figure 4: Main components in our teleoperation control scheme.

On the software side, all aforementioned components have been linked through the YARP middleware and robotics framework (Metta et al., 2006). To exploit modularity, a library exposing a set of relevant C++ interfaces (this is referred to as "device" in the YARP slang) was developed for each peripheral, while a common and generic (where possible) controller architecture was designed for the layer that processes data from the peripherals and generates output robot commands. On each level of this architecture, the YARP framework provides a convenient interface exposing relevant methods, e.g. for commanding in joint or Cartesian space, or reading data from a sensor or peripheral. Following this premise, usually a pair of devices (client and server) is also provided exposing those methods on the YARP network, which enables devices to run on different machines and even different operating systems, if compatible. Wherever a custom interface or device pair was not provided by YARP, it was designed from scratch in the context of this work.

Figure 5 represents the software modules involved in the proposed architecture.

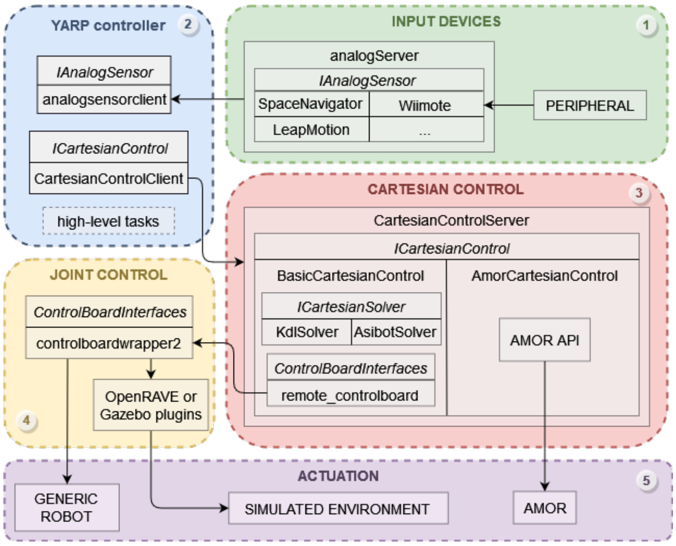

Figure 5: Proposed software control architecture.

The peripherals (block 1) are being treated as analog sensor devices, i.e. independent modules that publish data in a continuous one-way stream with a configurable period, thus resembling a generic sensor. All these modules implement a common interface that suits all sensor devices. The YARP framework provides the "analogServer" device to expose said interface commands over the network. This methodology is extensible to any device and therefore more peripherals could be adapted to this scheme in the future, as long as a new device is created and the relevant interface methods are implemented for it.

Block 2 depicts the main YARP controller, that is, the module that orchestrates data acquisition from the input device and generates high level commands that flow down the controller layers. In this step, specific high level tasks can be added to the controller depending on the working scenario, for instance: vision feedback for visual servoing applications, or proximity sensor feedback for shared control in teleoperation (Stoelen et al., 2015).

Block 3 groups the high level cartesian controller implementations. A generic one is introduced to handle open-sourced platforms (TEO, ASIBOT) developed at Universidad Carlos III de Madrid. It also embeds an additional layer of indirection to perform solver operations (forward and inverse kinematics): a generic one using the Kinematics and Dynamics Library (KDL), and a specific one for the simpler case of the ASIBOT arm. The AMOR platform required a separate implementation to accomodate (in terms of object-oriented programming and the adapter design pattern, to "wrap") C++ calls to its closed-source API. In the former scenario, the resulting output of the solver operations is fed to the next block to become joint-level commands.

In block 4, joint-space controllers can be divided into two groups to accomodate both robot realms this work encompasses: simulated robots and real robots. Similarly to the previous block, a set of common interfaces devoted to joint control is implemented separately for the real platforms, which run this controller aboard, and the simulated platforms (Gazebo or OpenRAVE).

Block 5 depicts the actuation stage on the actual platform, be it real or simulated. Joint control commands flow into this phase and the robot performs motion as requested by the user and according to any additional processing steps enforced by the controller (blocks 2, 3 or 4).

4. Experiments and Results

The proposed architecture has been tested on all available platforms, simulated (ASIBOT, AMOR, TEO) and real (the soft neck prototype in addition to the previous three).

Table 1 summarizes the output command type generated by each input device and the robot frame they refer to. The Cartesian controller which accepts the acquired data is transparent to the chosen frame as long as it is properly configured on start.

Table 1: Command type and reference frame mappings per peripheral.

peripheral	command type	reference frame
SpaceNavigator	velocity	base
Leap Motion	position	base
Wiimote	velocity	tool

It was observed that, on the real platforms, velocity-based commands cannot compensate the effects of gravity on the manipulator arm. Because of that, the arm tends to bend towards the ground during the execution of the trajectory, depending on its initial position and distribution of masses. In order to sort that problem out, the Cartesian controller can also accept position commands resulting from the time-integration of the original velocity commands. In the SpaceNavigator case, a virtual reference point is tracked by the main controller in order to represent a point in space which corresponds to an “ideal” position of the commanded TCP. This helps to maintain a steady trajectory and avoid a similar issue caused by gravity, while in fact a position control loop is closed. Since fast and precise inverse kinematics need to be performed in the position command case instead of iteratively obtaining differential IK (as in velocity commands), a geometric closed-form solver based on Lie’s algebra was used (Łukawski et al., 2022).

The experiments have been recorded and published for: the SpaceNavigator on TEO, `https://youtu.be/fL69GH1_IE0`; the Leap Motion Controller on TEO, `https://youtu.be/ZcUHZ9aGKeA`; the SpaceNavigator on the soft neck prototype, `https://youtu.be/8AsrKjzjGpg`.

The source code of the main controller and solvers has been published on GitHub: `https://github.com/roboticslab-uc3m/kinematics-dynamics/`.

5. Conclusions

In this paper, a control architecture for teleoperation on generic platforms driven by YARP has been proposed. It can be extended for potentially any peripheral to be considered in the future, as long as a convenient implementation is provided, while the remaining controller blocks remain the same. Thus, modularity and reusability is on the focus of this work.

Specific applications that benefit from this project have been proposed and tested in medical rehabilitation and assistance contexts, for instance, guiding a robotic arm in a shared control scenario with proximity sensors for obstacle detection (Oña et al., 2019, 2020).

Acknowledgments

This research has been financed by ALMA, “Human Centric Algebraic Machine Learning”, H2020 RIA under EU grant agreement 952091; ROBOASSET, “Sistemas robóticos inteligentes de diagnóstico y rehabilitación de terapias de miembro superior”, PID2020-113508RB-I00, financed by AEI/10.13039/501100011033; “RoboCity2030-DIH-CM, Madrid Robotics Digital Innovation Hub”, S2018/NMT-4331, financed by “Programas de Actividades I+D en la Comunidad de Madrid”; “iREHAB: AI-powered Robotic Personalized Rehabilitation”, ISCIII-AES-2022/003041financed by ISCIII and EU; and EU structural funds.

References

Cela, A., Yebes, J. J., Arroyo, R., Bergasa, L. M., Barea, R., López, E., 2013. Complete low-cost implementation of a teleoperated control system for a humanoid robot. Sensors 13 (2), 1385–1401.
DOI: `10.3390/s130201385`

Fischer, P., Daniel, R., Siva, K. V., 1990. Specification and design of input devices for teleoperation. In: Int. Conf. on Robotics and Automation. pp. 540–545.
DOI: `10.1109/ROBOT.1990.126036`

Jardón, A., Victores, J. G., Stoelen, M., Martínez, S., Balaguer, C., 2009. Asibot assistive robot in a domestic environment. In: Int. Conf. on PErvasive Technologies Related to Assistive Environments. pp. 1–4.
DOI: `10.1145/1579114.1579175`

Lazewatsky, D. A., Smart, W. D., 2011. An inexpensive robot platform for teleoperation and experimentation. In: Int. Conf. on Robotics and Automation. pp. 1211–1216.
DOI: `10.1109/ICRA.2011.5980230`

Łukawski, B., Montesino, I., Victores, J. G., Jardón, A., Balaguer, C., 2022. An inverse kinematics problem solver based on screw theory for manipulator arms. In: XLIII Jornadas de Automática, Logroño, Spain. pp. 864–869.
DOI: `10.17979/spudc.9788497498418.0864`

Mena, L., Monje, C. A., Nagua, L., Balaguer, C., 2020. Test bench evaluation for a soft robotic link. Frontiers in Robotics and AI 7.
DOI: `10.3389/frobt.2020.00027`

Metta, G., Fitzpatrick, P., Natale, L., 2006. Yarp: Yet another robot platform. Int. Journal of Advanced Robotic Systems 3 (1).
DOI: `10.5772/5761`

Monje, C. A., Pierro, P., Balaguer, C., 2011. A new approach on human-robot collaboration with humanoid robot rh-2. Robotica 29, 949–957.
DOI: `10.1017/S026357471100018X`

Nagua, L., Monje, C. A., Muñoz, J., Balaguer, C., 2018a. Design and performance validation of a cable-driven soft robotic neck. In: Jornadas Nacionales de Robótica, Valladolid, Spain. pp. 1–6.

Nagua, L., Muñoz, J., Monje, C. A., Balaguer, C., 2018b. A first approach to a proposal of a soft robotic link acting as a neck. In: XXXIX Jornadas de Automática, Badajoz, Spain. pp. 522–529.
DOI: `10.17979/spudc.9788497497565.0522`

Oña, E. D., Łukawski, B., Jardón, A., Balaguer, C., 2019. Hacia una estrategia asistida por robot para la recuperación de función motora de extremidad superior con aspectos cognitivos. In: XL Jornadas de Automática, Ferrol, Spain. pp. 756–763.
DOI: `10.17979/spudc.9788497497169`

Oña, E. D., Łukawski, B., Jardón, A., Balaguer, C., 2020. A modular framework to facilitate the control of an assistive robotic arm using visual servoing and proximity sensing. In: IEEE Int. Conf. on Autonomous Robot Systems and Competitions (ICARSC), Ponta Delgada, Portugal. pp. 28–33.
DOI: `10.1109/ICARSC49921.2020.9096146`

Stoelen, M. F., Tejada, V. F., Jardón, A., Balaguer, C., Bonsignorio, F., 2015. Distributed and adaptive shared control systems: methodology for the replication of experiments. IEEE Robotics & Automation Magazine 22 (4), 137–146.
DOI: `10.1109/MRA.2015.2460911`