The Journal of Open Source Software

LBR-Stack: ROS 2 and Python Integration of KUKA FRI for Med and IIWA Robots

Martin Huber [1¶], **Christopher E. Mower** [1], **Sebastien Ourselin** [1], **Tom Vercauteren** [1*], **and Christos Bergeles** [1*]

1 School of Biomedical Engineering & Imaging Sciences, King's College London, United Kingdom ¶ Corresponding author * These authors contributed equally.

DOI: 10.21105/joss.06138

Software

- Review
- Repository
- Archive

Editor: Adi Singh

Reviewers:

- @CameronDevine
- @dmronga

Submitted: 12 June 2023
Published: 15 November 2024

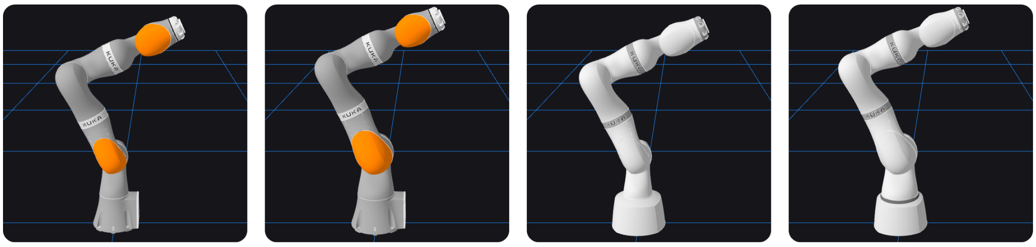

Figure 1: Supported robots in the LBR-Stack. From left to right: KUKA LBR IIWA 7 R800, IIWA 14 R820, Med 7 R800, Med 14 R820. Visualizations made using Foxglove [1].

Summary

The `LBR-Stack` is a collection of packages that simplify the usage and extend the capabilities of KUKA's Fast Robot Interface (FRI) (Schreiber et al., 2010). It is designed for mission critical hard real-time applications. Supported are the `KUKA LBR Med 7/14` and `KUKA LBR IIWA 7/14` robots in the Gazebo simulation (Koenig & Howard, 2004) and for communication with real hardware. A demo video can be found here. An overview of the software architecture is shown in Figure 2.

At the `LBR-Stack`'s core is the following package:

- **fri**: Integration of KUKA's original FRI client library into CMake: link.

All other packages are built on top. These include Python bindings and packages for integration into the Robot Operating System (ROS) and ROS 2:

- **pyfri**: Python bindings for the **fri**: link.
- **lbr_fri_ros2_stack**: ROS 1/2 integration of the `KUKA LBRs` through the **fri**: link.

For brevity, and due to the architectural advantages over ROS (Macenski et al., 2022), only ROS 2 is considered in the following. The **lbr_fri_ros2_stack** comprises the following packages:

- **lbr_bringup**: Python library for launching the different components.
- **lbr_description**: Description files for the `Med7/14` and `IIWA7/14` robots.
- **lbr_demos**: Demonstrations for simulation and the real robots.
- **lbr_fri_idl**: Interface Definition Language (IDL) equivalent of FRI protocol buffers.
- **lbr_fri_ros2**: FRI ROS 2 interface through `realtime_tools` (Chitta et al., 2017).
- **lbr_ros2_control**: Interface and controllers for `ros2_control` (Magyar et al., 2023).
- **lbr_moveit_config**: MoveIt 2 configurations (Coleman et al., 2014).

[1] Foxglove: https://foxglove.dev/ros.

The Journal of Open Source Software

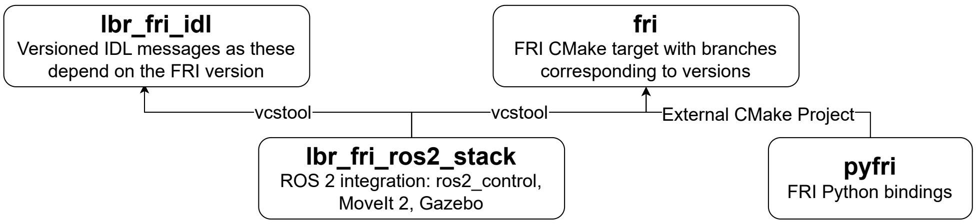

Figure 2: An overview of the overall software architecture. There exists a single source for KUKA's FRI. This design facilitates that downstream packages, i.e. the Python bindings and the ROS 2 package, can easily support multiple FRI versions. The ROS 2 side utilizes vcstool[2].

Statement of need

An overview of existing work that interfaces the KUKA LBRs from an external computer is given in Table 1. We broadly classify these works into custom communication solutions (Hennersperger et al., 2017; Safeea & Neto, 2019; Serrano-Muñoz et al., 2023) and communication solutions through KUKA's FRI UDP channel (Bednarczyk & Guzmán, 2023; Chatzilygeroudis et al., 2019). The former can offer greater flexibility while the latter offer a well defined interface and direct software support from KUKA. Contrary to the custom communication solutions, the FRI solutions additionally enable hard real-time communication, that is beneficial for mission critical development. Stemming from translational medical research, this work therefore focuses on the FRI.

Limitations with the current FRI solutions are:

1. Only support `IIWA 7/14` robots, not `Med 7/14`.
2. Don't provide Python bindings.
3. Don't support multiple FRI versions:
   - Modified FRI client source code iiwa_ros.
   - FRI client library tangled into the source code iiwa_ros2.
4. Partial support of FRI functionality. Both, iiwa_ros and iiwa_ros2, exclusively aim at providing implementations of the ROS 1/2 hardware abstraction layer. This does not support:
   - FRI's cartesian impedance control mode.
   - FRI's cartesian control mode (FRI version 2 and above).

The first original contribution of this work is to add support for the `KUKA LBR Med 7/14` robots, which, to the best author's knowledge, does not exist in any other work. The second novel contribution of this work is to provide Python bindings. This work solves the support for multiple FRI versions by treating the FRI library as an externally provided library by separating it into the **fri** package, which leaves the FRI's source code untouched and simply provides build support. The partial support for the FRI functionality is solved by defining an IDL message to KUKA's `nanopb` command and state protocol buffers in **lbr_fri_idl**. These messages can then be interfaced from ROS 1/2 topics via simple controllers or from the ROS 1/2 hardware abstraction layer.

[2]vcstool: https://github.com/dirk-thomas/vcstool.

Table 1: Overview of existing frameworks for interfacing the KUKA LBRs. A square indicates support for the respective feature. List of abbreviations: Hard Real-time (**RT**), Position Control (**Pos**), Impedance Control (**Imp**), Cartesian Impedance Control (**Cart Imp**), Hardware Interface (**HW IF**).

Framework	IIWA	Med	ROS	ROS 2	RT	FRI	pyfri	Pos	Imp	Cart Imp	HW IF
lbr-stack	•	•	•	•	•	•	•	•	•	•	•
iiwa_ros	•		•		•	•		•	•		•
iiwa_ros2	•			•	•	•		•	•		•
iiwa-stack	•		•					•	•	•	
libiiwa	•		•	•				•	•	•	
KST-KUKA	•							•	•	•	

Acknowledgement

We want to acknowledge the work in Hennersperger et al. (2017), as their MoveIt configurations were utilized in a first iteration of this project.

This work was supported by core funding from the Wellcome/EPSRC [WT203148/Z/16/Z; NS/A000049/1], the European Union's Horizon 2020 research and innovation programme under grant agreement No 101016985 (FAROS project), and EPSRC under the UK Government Guarantee Extension (EP/Y024281/1, VITRRO).

References

Bednarczyk, M., & Guzmán, J. H. G. (2023). ROS 2 Stack for KUKA IIWA Collaborative Robots. In *GitHub repository*. GitHub. https://github.com/ICube-Robotics/iiwa_ros2

Chatzilygeroudis, K., Mayr, M., Fichera, B., & Billard, A. (2019). *iiwa_ros: A ROS Stack for KUKA's IIWA Robots Using the Fast Research Interface*. http://github.com/epfl-lasa/iiwa_ros

Chitta, S., Marder-Eppstein, E., Meeussen, W., Pradeep, V., Rodríguez Tsouroukdissian, A., Bohren, J., Coleman, D., Magyar, B., Raiola, G., Lüdtke, M., & Fernández Perdomo, E. (2017). ros_control: A Generic and Simple Control Framework for ROS. *The Journal of Open Source Software*. https://doi.org/10.21105/joss.00456

Coleman, D., Sucan, I., Chitta, S., & Correll, N. (2014). Reducing the Barrier to Entry of Complex Robotic Software: A MoveIt! Case Study. *arXiv Preprint arXiv:1404.3785*. https://doi.org/10.6092/JOSER_2014_05_01_p3

Hennersperger, C., Fuerst, B., Virga, S., Zettinig, O., Frisch, B., Neff, T., & Navab, N. (2017). Towards MRI-Based Autonomous Robotic US Acquisitions: A First Feasibility Study. *IEEE Transactions on Medical Imaging*, *36*(2), 538–548. https://doi.org/10.1109/TMI.2016.2620723

Koenig, N., & Howard, A. (2004). Design and Use Paradigms for Gazebo, an Open-Source Multi-Robot Simulator. *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, *3*, 2149–2154 vol.3. https://doi.org/10.1109/IROS.2004.1389727

Macenski, S., Foote, T., Gerkey, B., Lalancette, C., & Woodall, W. (2022). Robot Operating System 2: Design, Architecture, and Uses in the Wild. *Science Robotics*, *7*(66). https://doi.org/10.1126/scirobotics.abm6074

Magyar, B., Stogl, D., Knese, K., & Community. (2023). Generic and Simple Controls Framework for ROS 2. In *GitHub repository*. GitHub. https://github.com/ros-controls/

ros2_control

Safeea, M., & Neto, P. (2019). KUKA Sunrise Toolbox: Interfacing Collaborative Robots With MATLAB. *IEEE Robotics Automation Magazine*, *26*(1), 91–96. https://doi.org/10.1109/MRA.2018.2877776

Schreiber, G., Stemmer, A., & Bischoff, R. (2010). The Fast Research Interface for the KUKA Lightweight Robot. *IEEE Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications How to Modify and Enhance Commercial Controllers (ICRA 2010)*, 15–21.

Serrano-Muñoz, A., Elguea-Aguinaco, Í., Chrysostomou, D., BØgh, S., & Arana-Arexolaleiba, N. (2023). A Scalable and Unified Multi-Control Framework for KUKA LBR IIWA Collaborative Robots. *2023 IEEE/SICE International Symposium on System Integration (SII)*, 1–5. https://doi.org/10.1109/SII55687.2023.10039308