

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

Martin Huber<sup>1¶</sup>, Christopher E. Mower<sup>1</sup>, Sebastien Ourselin<sup>1</sup>, Tom Vercauteren<sup>1\*</sup>, and Christos Bergeles<sup>1\*</sup>

<sup>1</sup> School of Biomedical Engineering and Image Sciences, King's College London, United Kingdom ¶

Corresponding author \* These authors contributed equally.

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

## Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Adi Singh](#) ↗

## Reviewers:

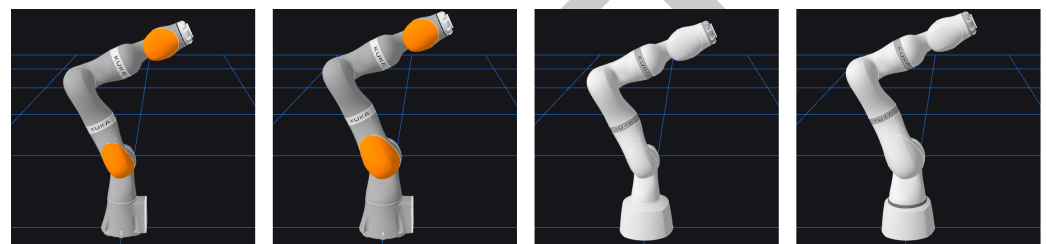
- [@CameronDevine](#)
- [@bmagyar](#)

Submitted: 12 June 2023

Published: unpublished

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).



**Figure 1:** Supported robots in the LBR-Stack. From left to right: KUKA LBR IIWA7, IIWA14, Med7, Med14. Visualizations made using Foxglove<sup>1</sup>.

## 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 Med7/14 and KUKA LBR IIWA7/14 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 are two packages:

- **fri**: Integration of KUKA's original FRI client library into CMake.
- **fri\_vendor**: Vendor library that integrates the **fri** into the ROS 2 build system.

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

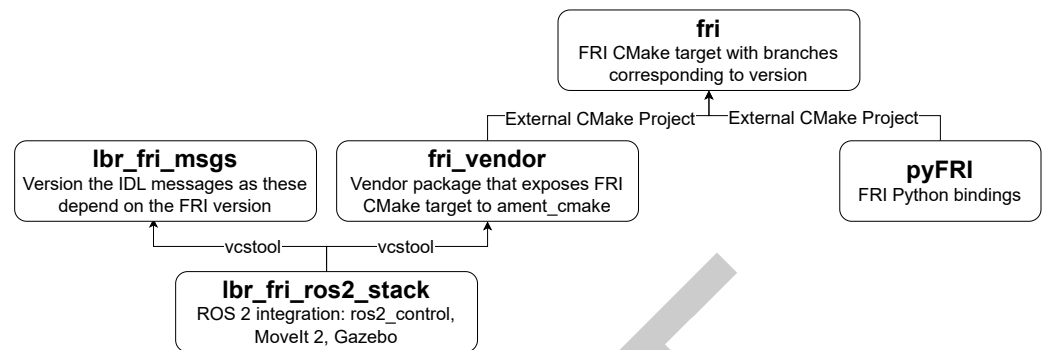
- **pyFRI**: Python bindings for the **fri**.
- **lbr\_fri\_ros2\_stack**: ROS 1/2 integration of the KUKA LBRs through the **fri\_vendor**.

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\_msgs**: 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](#)).

<sup>1</sup>Foxglove: <https://foxglove.dev/ros>.

- 29     ■ **lbr\_moveit\_config**: MoveIt 2 configurations ([Coleman et al., 2014](#)).



**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`<sup>2</sup>.

## 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](#); [Safaea & 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 IIWA7/14 robots, not Med7/14.
2. Don't provide Python bindings.
3. Maintainability:
  - Modified client source code `iiwa_ros`.
  - FRI client library tangled into 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 Med7/14 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 maintainability by outsourcing the FRI into the separate `fri` and `fri_vendor` packages, which leaves the FRI's source code untouched and simply provides build support. 4. is solved by defining an IDL message to KUKA's nanopb command and state protocol buffers in `lbr_fri_msgs`. These messages can then be interfaced from ROS 1/2 topics or from the ROS 1/2 hardware abstraction layer.

<sup>2</sup>`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					Pos	Imp	Cart Imp	HW IF
			ROS 2	RT	FRI	pyFRI					
<a href="#">lbr-stack</a>	•	•	•	•	•	•	•	•	•	•	•
<a href="#">iiwa_ros</a>	•		•		•	•		•	•		•
<a href="#">iiwa_ros2</a>	•			•	•	•		•	•		•
<a href="#">iiwa-stack</a>	•		•					•	•	•	
<a href="#">libiiwa</a>	•		•	•				•	•	•	
<a href="#">KST-KUKA</a>	•							•	•	•	

## 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]. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101016985 (FAROS project).

## 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](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](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*.
- Hennersperger, C., Fuerst, B., Virga, S., Zetting, 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.
- 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](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/>

90 [MRA.2018.2877776](#)

91 Schreiber, G., Stemmer, A., & Bischoff, R. (2010). The fast research interface for the  
92 kuka lightweight robot. *IEEE Workshop on Innovative Robot Control Architectures for*  
93 *Demanding (Research) Applications How to Modify and Enhance Commercial Controllers*  
94 *(ICRA 2010)*, 15–21.

95 Serrano-Muñoz, A., Elguea-Aguinaco, Í., Chrysostomou, D., Bøgh, S., & Arana-Arexolaleiba, N.  
96 (2023). A scalable and unified multi-control framework for KUKA LBR iiwa collaborative  
97 robots. *2023 IEEE/SICE International Symposium on System Integration (SII)*, 1–5.  
98 <https://doi.org/10.1109/SII55687.2023.10039308>

DRAFT