Comparison of robots: Fruitcore HORST and Franka Emika Panda

Florian Brunen, Simon Kramer, Johannes Lutz and Valentin Weber
Faculty of Computer Science
Furtwangen University, Furtwangen im Schwarzwald, Germany
{brunenfl, kramersi, lutzjoha, weberval}@hs-furtwangen.de

***Abstract*—This work compares the robots Franka Emika Panda and Fruitcore HORST. It presents a unified application programming interface and a common sense of workspace. The work shows that not only the hardware capabilities of a robot are important, but instead the software is equally important.**

***Index Terms*—robots, DOF, degrees of freedom, api, robot programming methods, carthesian coordinate systems**

I. Introduction

Robotics are important in many industries. The Furtwangen University (HFU) robotic laboratory holds two small, table mounted robots. The objective of this work is to compare the robots. Additionally, a common way to operate the robots shall be developed.

II. Design and physical capabilities

a) Physical design and appearance: HORST and Franka differ in their intended use-cases. While HORST is most often reffered to as a small industrial robot, Franka is commonly called a "Cobot" (collaborating robot) [1]. Industrial robots are meant to perform preprogrammed and precise actions independently to assist e.g. in industrial assembly processes. In contrast, cobots, as the name suggests are designed to be used as an assistance to humans. They are made to work on the same workpiece or process as a human, assisting it with its precise motion capabilities [2]. Industrial robots are not designed with the safety standards in mind required for humans to operate within its workspace.

Both robots are equipped with simple gripper tool heads. For this work, the gripper of both Horst and Panda was customized to allow gripping smaller objects. Moreover, the tool head of Horst can be unmounted and replaced with another tool.

b) Degrees of freedom and range of motion: One of the differences of the two robots is the number of joints. While Franka Emika Panda has 7 joints, and therefore 7 degrees of freedom (7-DOF), HORST has only 6 joints, meaning 6 degrees of freedom (6-DOF). This poses interesting questions: What is the advantage of this additional joint? In their work "Robots with seven degrees of freedom: Is the additional DoF worth it?" [3] pose this question. They analyze the dexterity of a robot with 7-DOF, and compare it the dexterity of the same robot, but with one joint disabled. Their results show that the 7-DOF version is more dexterous, but only by 16.8%. Moreover, when comparing these results with another 6-DOF robot that has significantly larger joint ranges, this value is again outperformed by 7.6%. While the advantage of the 7-DOF is clear in a direct comparison, the shortcoming of the missing degrees of freedom (DOF) seems to be possible to be counteracted by choosing larger joint ranges in a 6-DOF robot.

Searching on IEEE Xplore, ACM Digital Library and ArXiv using the search term "7dof AND robot" only returns a total of 164 scientific works. As this is very little literature, it is no surprise that only one paper performs [3] a full comparison of 7-DOF and 6-DOF.

Another major difference is the effective workspace available with each robot. The workspace of a robot can be described with two spheres subtracted from each other. For simplicity reasons and because the 10 cm from the top of the working table are the most important, the effective workspace is reduced to two cylinders, which are subtracted from each other. These cylinders are described by two circles and measured by operating the respecting robot near its workspace boundary. While the workspace of HORST is overall smaller than the workspace of Panda, HORST is able to operate very close to its own base. This is not the case with Panda, which is instead capable of reaching further. In numbers: The inner circle of HORST is 42 cm in diameter, the outer one 102 cm. The inner circle of Panda is 54 cm in diameter, the outer one 159 cm. The exact alignment is shown in figure 1.

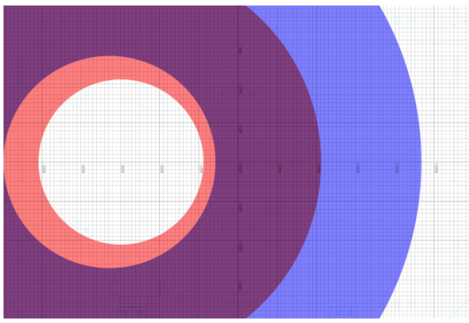


Fig. 1. Effective Workspace of HORST (red), Panda (blue) and both (purple)

III. Programming and Control

a) Robot programming method: Robot interaction can be classified in two methods: online and offline programming [4]. In online programming, the programmer has direct access to the robot, while in offline programming some level of abstraction is present. Both Franka and HORST have interfaces for offline, traditional text based programming. HORST supports receiving commands via the XMLRPC, making it easy to be controlled from many programming environments. Additionally, HorstScript, a language similar to JavaScript, may be used to program Horst. Both methods are well documented on the horstcosmos [5] website and allow the writing of sophisticated programs.

Franka, on the other hand offers the libfranka [6] C++ library to interact with the robot. However, this library is fairly technical and not very well documented, making it particularly difficult to get started with for those not intimately familiar with the Panda robot. There exists an unofficial Python wrapper around librfranka, called frankx, however this project does not expose all of the functionality of libfranka, has some critical bugs and is no longer maintained. Additionally, a visual, drag-and-drop method of programming is supported by Panda. This set of available actions can be expanded by purchase, or by creating with Lingua Franka [6], but this functionality was not yet publicly available during the time of this writing.

Further, Franka offers the option to be controlled via Robot Operating System (ROS). This is done by using the MoveIt motion planning framework [7] for example. MoveIt relies on C++ code to move the robot and define collision objects.

As for online programming methods, there are two relevant distinctions: In lead through programming, a teach pendant is used to move the robot. The operator moves the robot's arm to the required positions and saves them for later playback. The HorstFX software on the control tablet attached to Horst allows for lead though programming. In walk through programming, the operator moves the arm directly without the need for a teach pendant, then the robot replays the motion. This very intuitive programming method is only supported by Panda.

b) Unified robot API: To simplify the interaction with the robots further, a simplified, common robots Application Programming Interface (API) was developed. This API supports all basic operations to operate a robot with a single gripper tool head. Most importantly, it specifies the operation of the robot with Cartesian coordinates and Euler angles. The exact specification is shown in figure 2.

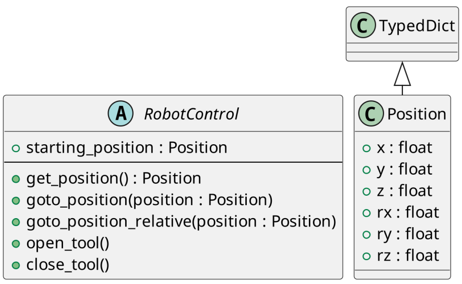


Fig. 2. UML class diagram of the common robots API

c) Common sense of workspace: To enable the use of a single API efficiently, a common sense of space is required.

This was achieved by developing a common robot coordinate system. As the working table is the same model for the two robots, it was decided that the common coordinate systems anchor point (0,0,0) is in the center of the tables top surface.

To use the common coordinate system with HORST, common coordinates are translated to robot specific coordinates with the following formula:

$$f\left(\begin{pmatrix} x & y & z \end{pmatrix}\right) = \begin{pmatrix} x + 301.03 & y \times -1 & z + 2.5 \end{pmatrix} \tag{1}$$

The Euler angles are already in the correct format.

The transformation function for the Panda robot looks as follows:

$$f(\begin{pmatrix} x & y & z \end{pmatrix}) = \begin{pmatrix} x + 23 & -(y - 4) & -(z + 445) \end{pmatrix} \tag{2}$$

IV. Conclusion

In conclusion, this work shows that both robots are similar in terms of hardware. HORSTs missing joint didn't limit its capabilities in our use case. However, this work also shows the importance of good software and API support. In comparison to Panda, HORSTs APIs are well-documented and easy to learn, but still powerful. Missing documentation relativizes the hardware advantage of Panda further.

References

[1] G. Lefranc, I. López, R. Osorio-Comparán, and M. Peña, "Cobots in automation and at home," in *2022 IEEE International Conference on Automation/XXV Congress of the Chilean Association of Automatic Control (ICA-ACCA)*, Oct. 2022, pp. 1–6. DOI: 10/gsbjwc.

[2] C. Cichosz and H. Gurocak, "Collision avoidance in human-cobot work cell using proximity sensors and modified bug algorithm," in *2022 10th International Conference on Control, Mechatronics and Automation (ICCMA)*, Nov. 2022, pp. 53–59. DOI: 10/gsbjwn.

[3] I. Kuhlemann, P. Jauer, F. Ernst, and A. Schweikard, "Robots with seven degrees of freedom: Is the additional DoF worth it?" In *2016 2nd International Conference on Control, Automation and Robotics (ICCAR)*, Hong Kong: IEEE, Apr. 2016, pp. 80–84. DOI: 10/gsbjwf.

[4] O. Heimann and J. Guhl, "Industrial robot programming methods: A scoping review," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vienna, Austria: IEEE, Sep. 2020, pp. 696–703. DOI: 10.1109/ETFA46521.2020.9211997.

[5] "horstCOSMOS by fruitcore robotics." (2023), [Online]. Available: https://horstcosmos.com/documentation (visited on 06/15/2023).

[6] Franka Emika GmbH. "Libfranka — franka control interface (FCI) documentation." (2017), [Online]. Available: https://frankaemika.github.io/docs/libfranka.html# (visited on 04/20/2023).

[7] "MoveIt motion planning framework." (2023), [Online]. Available: https://moveit.ros.org/ (visited on 06/15/2023).

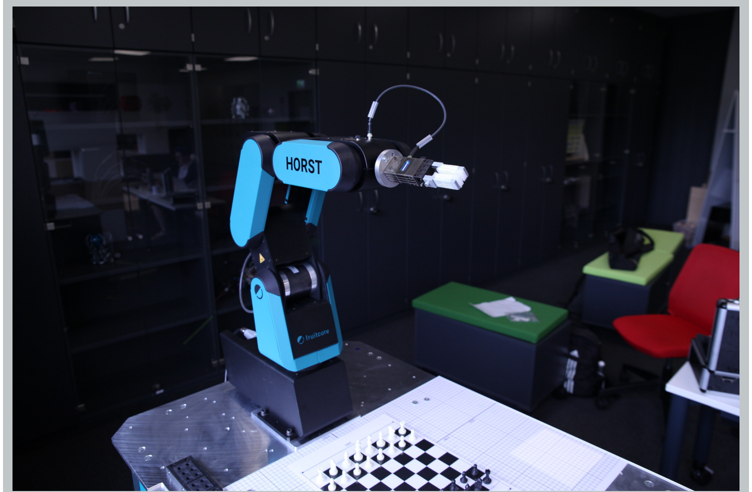
Comparison of robots: Fruitcore HORST and Franka Emika Panda

Introduction

Robotics are important in many industries. The HFU robotic laboratory holds two small, table mounted robots. The objective of this work is to compare the robots. Additionally, a common way to operate the robots shall be developed.

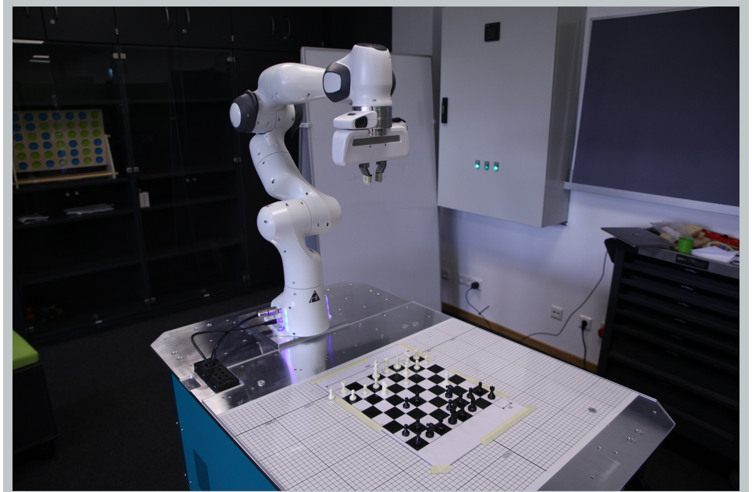
Horst

HORST is a small industrial robot developed by fruitcore robotics GmbH. It possesses six joints and a gripper tool head. Among others, it offers an XML-RPC programming interface for remote control.



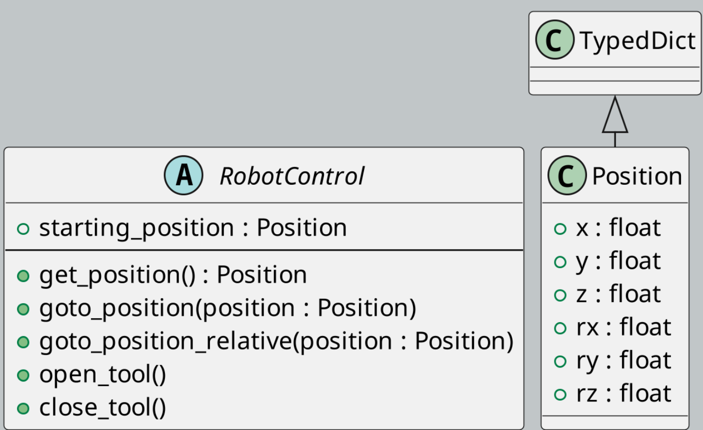
Franka Emika Panda

Franka Emika Panda is a collaborative robot by FRANKA EMIKA GmbH. It is designed to operate together with humans in its work area. Programming it is possible through a visual interface, but also with a C++ and Python API.



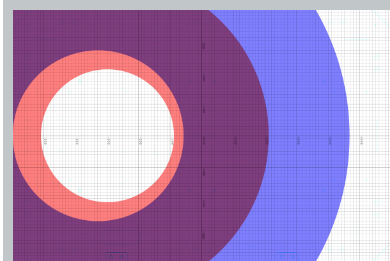
Unified API

To simplify the interaction with the robots further, a simplified, unified robots API was developed. This API supports all basic operations to operate a robot with a single gripper tool head. Most importantly, it specifies the operation of the robot with Cartesian coordinates Euler angles. The image shows a class diagram of this API.



Effective & common workspace

To enable the use of a single API efficiently, a common sense of space is required.
This was achieved by developing a common robot coordinate system. As the working table is the same model for the two robots, it was decided that the common coordinate systems anchor point (0,0,0) is in the center of the tables top surface.



Blue: Franka Emika Panda
Red: Fuitcore HORST
Purple: common

To use the common coordinate system with HORST (1) and Panda (2), common coordinates are translated to robot specific coordinates. The Euler angles are already in the correct format.

$$f\left(\begin{pmatrix} x & y & z \end{pmatrix}\right) = \begin{pmatrix} x + 301.03 & -y & z + 2.5 \end{pmatrix} \quad (1)$$

$$f\left(\begin{pmatrix} x & y & z \end{pmatrix}\right) = \begin{pmatrix} x + 23.0 & -y + 4.0 & -z - 445.0 \end{pmatrix} \quad (2)$$

Florian Brunen **Simon Kramer** **Johannes Lutz** **Valentin Weber**
{brunenfl, kramersi, lutzjoha, weberval}@hs-furtwangen.de
Furtwangen University