The Fast Research Interface for the KUKA Lightweight Robot

Günter Schreiber, Andreas Stemmer, and Rainer Bischoff

***Abstract*—The KUKA lightweight robot (LWR) provides many unique features for robotic researchers. To give full access to these features, a new interface was developed that gives direct low-level real-time access to the KUKA robot controller (KRC) at high rates of up to 1 kHz. On the other hand, all industrial-strength features, like teaching, motion script features, fieldbus I/O and safety are provided. Using standard UDP socket technology, the user is not limited to one specific runtime system. This paper describes the capabilities of the interface, the practical realization within the LWR control architecture and first applications of the interface.**

I. INTRODUCTION

The KUKA lightweight robot (LWR) (Figure1) is the latest outcome of a bilateral research collaboration between KUKA Roboter GmbH [1] and the Institute of Robotics and Mechatronics at the German Aerospace Center (DLR) [2]. Due to its unique features like high payload ratio, programmable active compliance and torque sensor feedback, it enables researchers and automation engineers to develop new industrial and service robot applications. From the beginning, one important aspect of the LWR product development was to make features available from the KUKA controller and its integrated scripting language (KUKA Robot Language, KRL). This way, every industrial robot programmer who is used to program standard industrial KUKA robots, is able to program the LWR. KRL was extended to make available the LWR features, such as impedance control [3], which is not available for standard robots. Also, the “all-in-a-box” controller hardware was developed, so that power supply, controller board and safety logic are in a common housing. While this kind of approach fits the requirements of industry, researchers have a more elaborate desire w.r.t. to such an arm. To investigate the requirements of the research community KUKA developed a questionnaire within the EC-funded project BRICS – Best practice in robotics [4].

This paper is organized as follows: First the requirement analysis will be presented that motivated the development of the Fast Research Interface (FRI). Section III will explain the FRI in more detail. The control system architecture and its implementation are presented in sections IV and V, respectively. First applications of the FRI are shown in section VI before the paper is concluded in section VII.

Figure 1: KUKA Lightweight Robot (LWR) with KUKA Robot Controller (KRC 2lr) and KUKA Control Panel (KCP).

II. REQUIREMENT ANALYSIS

A. *The BRICS Project*

The EC-funded project BRICS – Best practice in robotics – provided the framework for the requirement analysis. The prime objective of BRICS is to structure and formalize the robot development process itself and to provide tools, models, and functional libraries, which allow reducing the development time by a magnitude. BRICS is working together with academic as well as industrial providers of robotics “components” (hardware and software), to identify and document best practices in the development of complex robotics systems, to refactor (together) the existing components in order to achieve a much higher level of reusability and robustness, and to support the robot development process with a structured tool chain and code repository.

The first target group of the BRICS project is the robotic research community. BRICS aims at offering hardware with a consistent set of harmonized, well-defined and documented, public APIs, and an integrated development environment. BRICS will help researchers to design and make operational complex robotic systems with minimal effort and avoid so-called “from scratch developments”. BRICS involves the community through questionnaires, workshops and research camps.

Günter Schreiber is with KUKA Roboter GmbH, Augsburg, Germany (e-mail: GuenterSchreiber@kuka-roboter.de)

Andreas Stemmer is with DLR, Institute of Robotics and Mechatronics, Oberpfaffenhofen, Germany (e-mail: andreas.stemmer@dlr.de)

Rainer Bischoff is with KUKA Roboter GmbH, Augsburg, Germany (e-mail: RainerBischoff@kuka-roboter.de)

B. Use Cases

In a first step, use cases were defined and discussed with the robotics community using a questionnaire. The use cases were structured according to the demands of typical applications as follows:

1) Standard industrial application

- pre-programmed task
- external interface (if any) only necessary for synchronization

2) Advanced industrial application with non-continuous feedback control

- pre-programmed task
- external sensors, but only discrete measurements
- no continuous feedback control ("look-then-move")
- industrial controller does: path planning, interpolation, inverse kinematics, etc.
- simple interface sufficient (exchange of data without real-time requirements)
- real-time / non-real-time vs. (non-)continuous are two different issues

3) Advanced industrial application with continuous feedback control

- pre-programmed task
- external sensors used for feedback control
- examples: cameras, FT sensor, …
- major part of application is programmed on industrial controller
- sensor data processing is programmed outside robot controller
- low cycle time and minimal dead time of feedback control is important for sensor-based control → real-time interface: exchange of data in fixed time intervals (e.g., interpolation cycle time)

4) Research outside robotics field:

- robot is used for research outside the field of robotics, e.g., robot is used to automate measurements
- use cases 1-3 are applicable

5) Robotics research – system / application level:

- robot is used as part of a larger system to realize and evaluate new applications in the area of cognitive systems, service robotics, etc.
- integration of robot controller in other systems should be easy
- functionality of robot controller should be controllable from outside

6) Robotics research – control level:

- robot is used to implement and evaluate new robotics algorithms in the area of control, e.g., inverse kinematics, dynamics, force control, …
- robot control at low level (real-time constraints)

7) Robotics research – haptics:

- robot is used as haptic input device (e.g., for virtual reality) or slave for tele-presence systems; high sensitivity for force control (< 10 N)
- control of robot systems at lowest level possible (real-time constraints)

C. BRICS Questionnaire and Use Case Analysis

Researchers were asked in what kind of applications they would want to use the LWR. The following table gives an overview of some of the answers along with a classification according to the use cases defined above:

Attach a novel hand & use it for a project for picking in an industry application	2, 3
Visual servoing	3, 6
Line drawing – calligraphic text painting	2, 3
Peg in hole, pegs have small clearance, put a key into a lock & take it out	2, 3
Vibration damping	6
Like to implement „our own control algorithms“	6
Haptic input device for virtual / augmented reality	7
Advanced assembly and manufacturing, adapt the robot in real-time with additional sensor/process model information	3
Pick and place task in office or home, including simple manipulation	5
Instrument carrier, with controllability of redundancy and with 10 Hz mechanical bandwidth, but with 1 kHz sensor interface (read access)	6
10 Hz multi-TCP „force“ control (or other sensor)	6

Table 1: Excerpt from BRICS questionnaire: selected answers from potential LWR users.

The questionnaire revealed that there is a huge variety of ideas on how to use the LWR. Still, the answers could be classified in two major groups: The first group contains those customers with applications in use cases 1-4. These can be satisfied with the current industrial controller. Here, it became clear to KUKA that it is necessary to inform users better on the features of the system and teach them how to use them. The second group contains use cases 5-7. Here, the customers want to control the robot at the lowest level possible. It is interesting to see that some users intend to improve or research on algorithms that are already quite mature and integrated in the KUKA controller, such as standard motion control algorithms or more advanced ones, such as vibration damping. Here, it also seems necessary to better inform users about the features of the system that they do not re-invent the wheel. Nevertheless, to be able to realize the customer ideas – including the most challenging use case 7, haptics, in which nothing can be anticipated, it deemed necessary to develop a generic interface that allows access at different levels and control rates without loosing the strengths of an industrial controller.

III. FEATURES OF THE FAST RESEARCH INTERFACE

Following the requirement analysis three important features of a new interface could be sketched:

- access to the core controller functions at varying control rates with low latency
- robust and beneficial interplay with an industrial-strength controller
- portability to various flavors of operating systems

With respect to the access rate to this new interface the user has to keep in mind that high sampling rates demand a computer with real-time operating system that generates the commands – something not all users can or want to provide. For many robotics research topics, like path planning for example, a slower update rate is good enough and much easier to handle for the user. Therefore, a flexible cyclic timeframe of [1…100] ms was spotted as one of the essential key features of the new interface. This has been realized by letting the user select the desired access rate during the activation of the interface. Once the communication has been established, its quality is constantly monitored by the control system to identify possible problems.

The second important feature to be realized was the robust interplay of the interface with an industrial-strength controller, which means that all "normal" features of the KUKA LWR controller should still be accessible, such as:

- standard robot programming and operation
- easy realization of LIN/PTP/CIRC motions
- touchup - teaching
- selection of control mode (position or impedance control, impedance parameters)
- safety system and switches (test / automatic modes)
- fieldbus infrastructure for extensions

With the FRI these features are still available and can be accessed through a powerful programming language, the KUKA Robot Language (KRL). Since the new interface is embedded in the KUKA controller, bidirectional communication between KRL applications and client applications are possible. This means that the user can use existing infrastructure and features of KRL for the "body" of the application and concentrate on the real application/research issues. Subtasks like prepositioning the robot to a taught-in pose or controlling an industrial gripper can be realized with existing technology without effort. Even in those subtasks that use FRI, the user can choose to keep some of the commands from the KRL layer and override others. It is possible to just monitor the sensor signals, to alter part of the KRL-generated trajectory, or to completely take over the commands to the robot.

The third important feature of the FRI enables users to choose for their client application any operating system they need as far as it is able to establish a reliable UDP socket communication, using simple binary datagrams. With such a simple communication protocol the porting/integrating to all flavors of operating systems should be straight forward. An "SDK" sample implementation is included with the FRI documentation, which facilitates the integration and usage of the interface. In principle, one simple header file containing the data structures will be sufficient for the integration. Several beta test customers succeeded already in the task of integrating the interface into their systems (see section VI).

IV. CONTROL SYSTEM ARCHITECTURE

The interplay between an industrial robot controller and the research interface provides the researcher with several levels of autonomy. From a user point of view, the controller itself is equipped with a control panel (KUKA Control Panel – KCP), which enables the user to move and teach the robot. The KCP provides a three-way engage switch and various operating modes enabling safe application development and testing. The panel can also be used to display, execute and edit the KRL scripts (Figure 2).

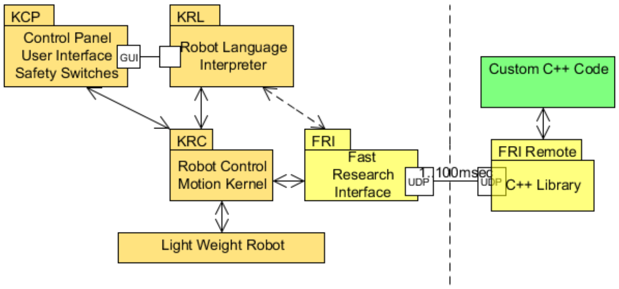

Figure 2: Overview of the FRI control system architecture.

The KUKA motion kernel already provides different methods for generating smooth trajectories (PTP, LIN, CIRC) by simple KRL commands. FRI interacts directly with the motion kernel (Figure 3). The generated trajectory is sent to the remote computer and modifications or complete replacement of the trajectories are possible. As the LWR is more than just a position controlled industrial robot, the possible commands also include stiffness and damping parameters for impedance control or desired joint torques.

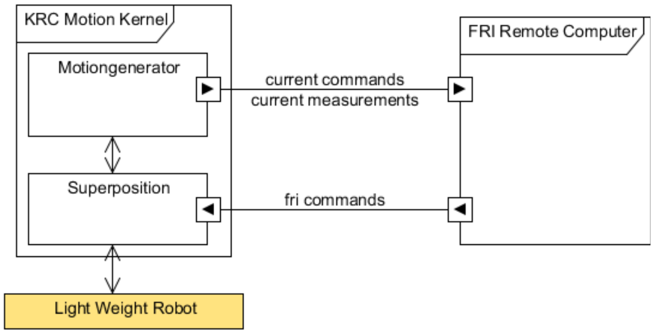

Figure 3: Interaction of the FRI with the KRC motion kernel.

A second layer of interaction is possible using a set of KRL variables that can be accessed by the interface. It is the user's choice what these variables do on KRL level and they can be used to set or read digital I/Os of the KRC, to talk to any fieldbus component attached to the KRC, to control grippers, to set or read internal states of the KRL program, or to change the program flow of KRL scripts (Figure 4).

Figure 4: Interplay of the FRI with the KRL interpreter through a user-defined set of variables.

V. IMPLEMENTATION

A. *UDP Communication*

The major challenge in utilizing a UDP socket communication is reliability – especially w.r.t real-time requirements. That means that not only packet loss needs to be detected, but also communication delays (roundtrip) and variations (jitter) have to be measured and monitored. Packet loss could be avoided by using TCP/IP streams instead of UDP packets, however the retransmission of the data in case of transmission errors is of no use in a real-time environment. So, the interface itself uses UDP sockets, and implements an own packet loss detecting and recovery mechanism (Figure 5).

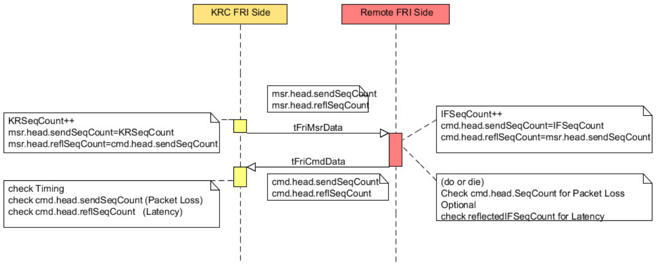

Figure 5: UDP socket communication with packet loss detection.

The basic idea is to send a continuous sequence count on each data packet. The sequence count itself has to be continuous and monotonously growing. On top of this sequence count mechanism, the roundtrip time – including the delay w.r.t the remote side – is computed, since the received sequence count has to be mirrored on each side of communication. Single packet dropouts up to a certain extent can be bridged by extrapolation of the previous commands whereas more severe communication problems are considered as error.

B. *Interface Modes*

The interface itself has two modes of activity and is realized as a state machine (Figure 6):

- Monitor mode: all data is provided to the remote side, no cyclic commanding is feasible, the cyclic timing is not required to meet real-time conditions
- Command mode: as monitor mode, but commanding is feasible and the cyclic timing is required to meet real-time conditions.

On activating the interface, the FRI state machine enters monitor mode and the KUKA controller (acting as the client) starts transmitting measurement data packets to the configured remote computer (acting as the server) in the desired intervals. The remote computer is expected to send an answer packet with commands within the given timeframe. Only if the timing (receiving data) of the remote side is appropriate for some time, i.e. only if the remote side shows that it is capable of meeting the desired timing specifications, the command mode may be entered.

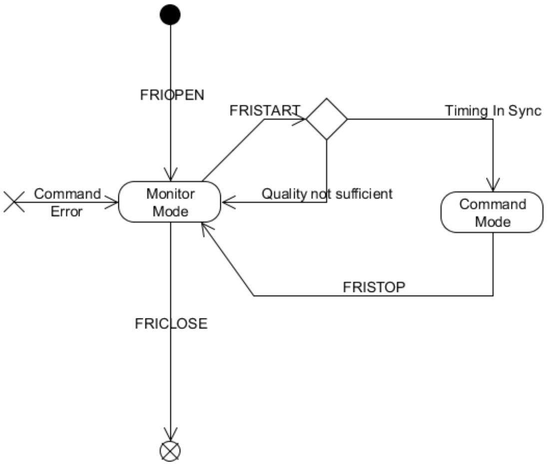

Figure 6: FRI state machine with monitor and command modes.

In command mode, the desired commands from the remote computer are handed over to the robot and executed. Fine-interpolation is done if the user selected a sample rate that is longer than 1 ms. If the communication quality should not be sufficient any more, the interface falls back to monitor mode, stopping the robot if necessary. During runtime, the modes “monitor mode” and “command mode” may be switched on user demand. Since the switching of LWR control modes raises several handshaking and convergence issues, it is feasible in monitor mode only.

C. *Safety Issues*

In the context of usability, also safety issues have to be considered, which includes both machine safety and user safety. Since the system originates from industrial controllers, the common operating modes “test” and “automatic” are supported. In test mode, a three-way engage switch is active, which enables the user to test its software and stop the robot immediately on obviously erroneous behavior of the robot. The remote side must obey those safety modes, and is informed of the activity of the system. Additionally, the joint speeds are strictly observed and limited. For testing the interface and gaining experience, the user can shrink the allowed limits for FRI, and thus be sure that the robot will not execute erroneous commands. If limits of the commanded values are reached or the timing requirements are not met, the KRC will enforce a fallback to “monitor mode” and bring the robot into a safe state.

D. Control Modes

In the following, the available cyclic control modes are sketched briefly:

1) Joint specific position control

The command is interpreted as $q_{cmd} = q_{FRI}$, i.e., the original command from the KRC is replaced by the external command q_{FRI}. For a smooth transition between monitor mode and command mode, the remote computer has to mirror the commands of the KRC during monitor mode.

2) Joint specific impedance control

The control law of the joint specific impedance control is in principle something like

$$\tau_{Cmd} = k_j(q_{FRI} - q_{msr}) + D(d_j) + \tau_{FRI} + f_{dynamics}(q, \dot{q}, \ddot{q})$$

With this control law, a virtual spring $k_j\,(q_{FRI} - q_{msr})$ is realized in joint space. Depending on the configuration of the interface, some of the following parameters may be modified by the remote side: the spring stiffness k_j, the normalized damping parameter d_j, the desired position q_{FRI} and/or a superposed control torque τ_{FRI} are modifiable by the remote side. If not specified by the FRI remote side, the common defaults (specified by the user’s KRL program) are taken. The other components are depended on the control system, and are computed inside the motion kernel: The damping term $D(d_j)$ and the dynamic model itself $f_{dynamics}(q, \dot{q}, \ddot{q})$. The same rules for having a smooth transition apply as for position control.

3) Cartesian impedance control

The control law is something like

$$\tau_{Cmd} = J^T\left(k_c(x_{FRI} - x_{msr}) + D(d_c) + F_{FRI}\right) + f_{dynamics}(q, \dot{q}, \ddot{q})$$

The resulting joint torque is computed by a Cartesian law defined by the transposed Jacobian J^T. The control law represents a Cartesian virtual spring $k_c\,(x_{FRI} - x_{msr})$. As in the joint specific case, several parameters are modifiable by the remote side, if desired: the Cartesian stiffness k_c, the Cartesian normalized damping parameter d_c, the desired Cartesian position x_{FRI} and the superposed Cartesian force/torque term F_{FRI}. The terms for the damping design $D(d_c)$ and the dynamic model $f_{dynamics}(q, \dot{q}, \ddot{q})$ are taken care of in the motion kernel itself.

E. Transmitted Data

To make the implementation as easy as possible, the same data structures are used for transmission independent on control modes. A short overview of the transmitted data is presented in the following listing.

1) From KRC to Remote Computer:

- joint sensor data (position, torque)
- Cartesian measured data (TCP frame, estimated force/torque at TCP)
- information about interface state
- information about robot state
- current commanded values (joint specific and Cartesian)
- current Jacobian and mass matrix
- KRL interaction variables

2) From Remote Computer to KRC

- bitfield for desired commands
- set joint commands (position, additional torque)
- set Cartesian commands (TCP frame, additional force/torque)
- set impedance parameters (joints specific and Cartesian)
- KRL interaction variables

VI. EXAMPLE APPLICATIONS

Several example applications using the new interface were already developed by KUKA, by DLR and by customers of KUKA. The general feedback showed that the usage of the interface was easy and intuitive and simplified the work with the robot in research environments a lot.

In the following, three sample applications are presented:

- bidirectional coupling of a LWR to a commercial haptic input device (Force Dimension “omega.7”),
- two-robot bidirectional coupling for haptic interaction,
- complex path planning demonstration for a two-arm system.

A. Sample Application: remote control with the omega.7 device

One example application shown on the LWR information day at DLR is the bidirectional coupling of a LWR to a commercial haptic input device from Force Dimension – the “omega.7” [5], which provides seven degrees of freedom whereof the three translational degrees are active and can display interaction forces (Figure 7).

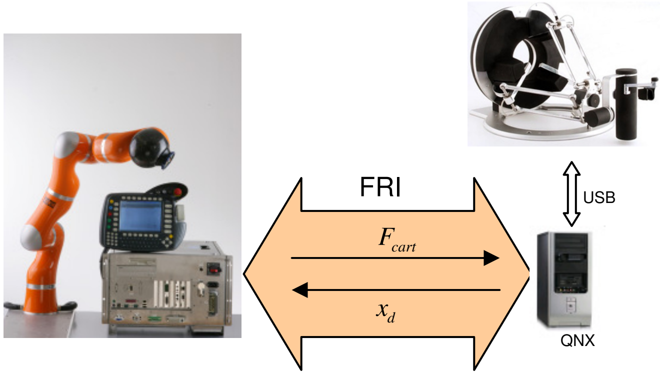

Figure 7: FRI sample application: haptic interplay of an input device with the KUKA LWR.

The omega.7 delivers a Cartesian desired position that is mapped to the robot workspace by appropriate rotation. As the workspace of the omega.7 is much smaller than that of the LWR, a relative Cartesian interface is used that produces a relative, scaled motion of the robot only when an attached footswitch is pressed. Without the footswitch, no coupling between LWR and omega.7 is active. The device can then be moved without affecting the robot, bringing it back to a suitable position at the center of its workspace. The robot uses Cartesian impedance control and is fed with the omega.7 commands directly. To give the user feedback about the contact forces, the estimated external forces at the TCP are commanded to the omega.7. Force as well as the relative motions are scaled to accommodate the different device properties of LWR and omega.7.

The LWR is equipped with an industrial two-finger gripper which is accessible from KRL. The seventh degree of freedom of the omega.7 device (a kind of "trigger" that is moved with the index finger) is used to open and close the gripper by transmitting its position to a KRL variable and evaluating it in an endless loop. Drivers for the real-time operating system QNX [6] used at DLR were readily available from Force Dimension, so that the complete interface could be implemented within one day.

B. *Sample Application: telepresence with two LWRs*

Within the framework of the BRICS project and in cooperation with the KU Leuven, a telepresence application was realized. Two LWR controllers were connected to copy the motion and to display the forces exerted at the other end by exchanging joint positions. In this setting, Orocos [7] was used as the real-time platform, based on RTAI-Linux. Within two days of work, the FRI was integrated to provide real-time communication from one Orocos PC to two LWR controllers (Figure 8) [8].

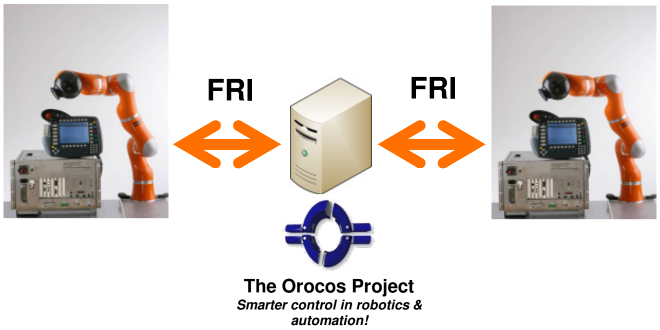

Figure 8: FRI sample application: haptic interaction with two LWRs.

C. *Sample Application: path planning in cluttered environments*

In the framework of the CoTeSys cluster of excellence at the Technical University Munich [9] investigations w.r.t. to mobile manipulation in dynamic non-structured environments are currently undertaken. A kitchen has been chosen as test environment as it provides a number of challenges:

- need to react to sensor data (torques/camera/laser/...), cannot execute pre-planned trajectories,
- uncertainty of positions of objects and the robot,
- clutter/small objects,
- operating in contact with a partially unknown environment,
- possible humans present/interacting.

These challenges are tackled with an omnidirectional mobile platform with two LWR arms and four-finger hands able to perform various tasks (Figure 9). The two LWRs are remotely controlled through the FRI from a standard Linux system with a high priority thread. Both position control and joint impedance control are currently used. The software system is based on the ROS middleware [10].

TU Munich reports that the advantages of high frequency/low latency control led to the advantage, that external control loops become simpler (more linear) – since the high bandwidth inside the motion kernel takes care of the rest. Feature extraction out of the sensor stream remained feasible, due to the high information bandwidth (high measurement frequency) [11].

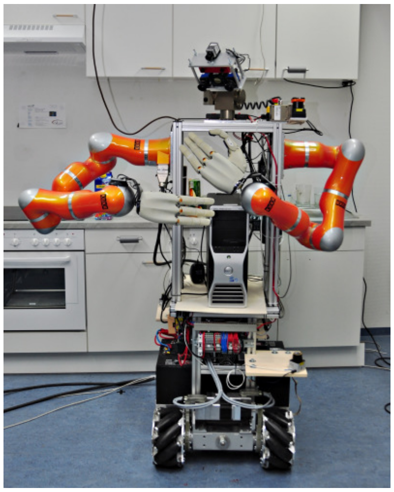

Figure 9: FRI sample application: path planning for dual-arm omnidirectional robot.

VII. CONCLUSION

To identify best practice in robotics requires the comparison of algorithms and methodologies on standardized robot hardware with powerful interfaces. The development of the Fast Research Interface (FRI) for the KUKA Lightweight Robot (LWR) was motivated by the need to provide researchers with such an interface and was realized in close cooperation of KUKA and DLR. The FRI gives access to an industrial-strength controller at very high control rates up to 1 kHz from any client PC. The FRI is based on the KUKA Robot Controller KRC2 lr, which means that developing controller features that already exist, such as sequence control, interpolation and safety features, will be avoided. The first application examples are very encouraging because of the little time needed to integrate the FRI in independent client systems and the high performance achieved even in haptics applications. Since the FRI in combination with the KUKA LWR provides researchers with a unique reference platform for robotics research the authors hope that the LWR will become a central element in robotics research worldwide.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. FP7-ICT-231940-BRICS (Best Practice in Robotics).

Additional funding was made available through the European Community's Sixth Framework Programme (FP6) under grant agreement no. FP6-ICT-045359-PHRIENDS (Physical Human-Robot Interaction: Dependability and Safety).

REFERENCES

[1] (KUKA) www.kuka-roboter.de
[2] (DLR) www.dlr.de/rm/en/
[3] „A unified passivity-based control framework for position, torque and impedance control of flexible joint robots", A. Albu-Schäffer, Ch. Ott, and G. Hirzinger in Int. Journal of Robotics Research, vol. 26, no. 1, pp. 23–39, 2007
[4] (BRICS) www.best-of-robotics.org "Best practices in robotics research"
[5] (Force Dimension) www.forcedimension.com
[6] (QNX) www.qnx.com
[7] (OROCOS) www.orocos.org "The Orocos Project, Smarter control in robotics & automation"
[8] Youtube Video www.youtube.com/watch?v=N7c_D4XWmk8 "Two KUKA Lightweight Robots connected using Orocos RTT"
[9] (Cotesys) cotesys.in.tum.de "Cluster of excellence"
[10] (ROS) www.ros.org/wiki/
[11] "Autonomous grasping of unmodeled objects using 3D sensor data and tactile information", Alexis Maldonado, Uli Klank and Michael Beetz. Submitted to the International Conference on Intelligent Robots and Systems (IROS) 2010. Taiwan.