The Fast Research Interface for the KUKA Lightweight Robot

G. Schreiber (KUKA Roboter GmbH)

What‘s new?

- ICRA Presentation – Workshop and Exhibition

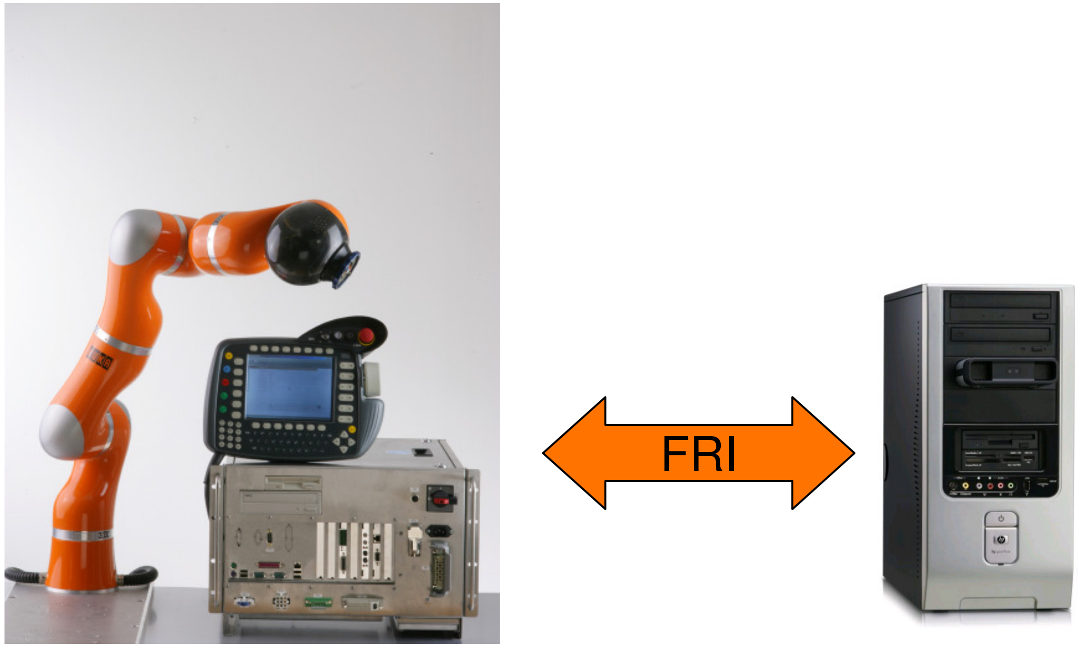
- Release of FRI as product

- Upgrade to KRC5.6
  - Seamless integration into KRL
  - New KRL Variables $friQuality, $friState
  - KRL Functions integrated (friOpen, friOpen2, friStart, friStop, friClose, friShow)
  - Lots of minor bug fixes

- Customer feedback and feature requests
  - Cartesian Impedance control should be commanded by joint angles only
  - Mobile Version of FRI (discussed seperately)

What is “FRI” – KUKA.FastResearchInterface

- Interface for remote control of KUKA Lightweight Robot
- Embedded to industrial commercial controller
- Remote PC-System free of choice by research customer
- Open for research community



KUKA LWR + Controller (KRL)

Remote PC

Use cases – application level

A) standard industrial application

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

B) advanced industrial application with non-continuous feedback control

- pre-programmed task
- external sensors, but only discrete measurements
- no continuous feedback control (i.e., only “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

C) advanced industrial application with continuous feedback control

- pre-programmed task
- external sensors used for continuous feedback control
- examples: cameras, force torque sensor, distance measurement, …
- 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)

D) research outside robotics field:

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

E) robotics research – system / application level:

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

F) 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, visual servoing, …
- control of robot systems at low level (real-time constraints)

G) 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 constr.)

Use Cases
(collected and categorised answers)

Attach a novel hand & use it for a project for picking in an industry application	C
Visual Servoing	C
Line-Drawing – Calligraphic text painting	C
Peg in hole, pegs have small clearance, put a key into a lock & take it out	B,C
Vibration damping (→ already solved internally)	F
Like to implement „our own control algorithms“ *(→ on which level? Current control will not be possible)*	F
Advanced assembly and manufacturing, adapt the robot in real-time with additional sensor/process model information	C
Pick and place task in office or home, including simple manipulation	E
instrument carrier, with controllability of redundancy and with 10 Hz mechanical bandwidth, but with 1 KHz sensor interface (read access)	F
10 Hz multi-TCP „force“ control (or other sensor)	G

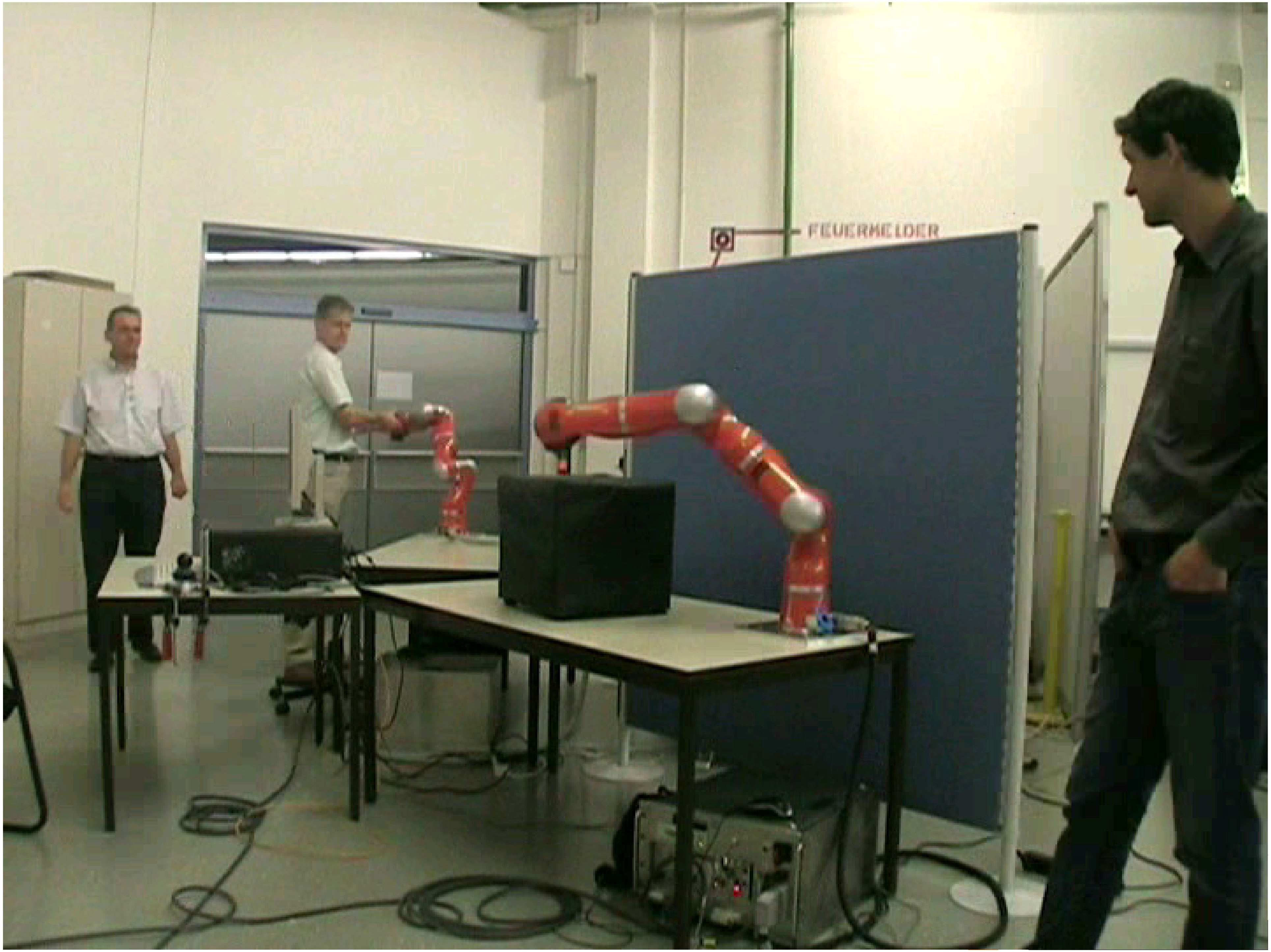
→ Every application is different → access at different levels necessary

Motivation – Why FRI instead RSI

Motivation: The answers to questionnaires (e.g. BRICS) within the research community revealed that for some research purposes a fast interface is needed.

Existing interfaces, e.g. RSI not sufficient for “rapid prototyping” & “research”

- **(External) sample rate too slow** (and inflexible)
- (Specific) lightweight robot features not supported
- **Torque/Impedance control not feasible**
- High protocol overhead due to XML
- Porting to other robot programming systems not easy (fast XML parser required)
- Embedding in other systems difficult
- Researchers like to have their flavor of operating system and real-time frameworks (huge variety)
- Middleware choice free for the customer
- **Low acceptance in research community**


FEUERMELDER

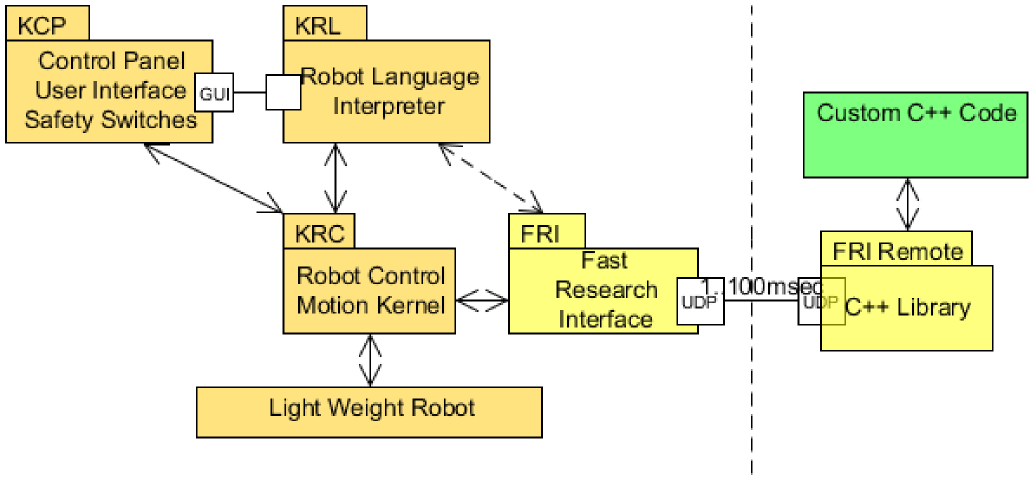
„Take the best of two worlds“ – Features

Requirements

- 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 flavours of operating systems
- no specific (full-grown) (special) middleware

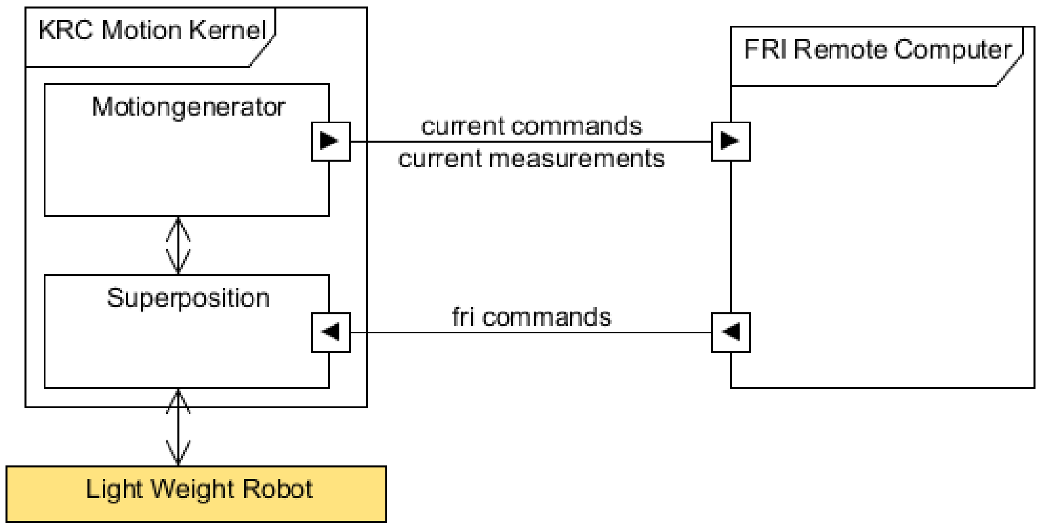
Realised

- Cyclic timeframe: range 1 - 100 ms
- Interplay to KRL Programs (scripting features)
  - Superposition of LIN/PTP/CIRC Motions
- Several control modes of LWR
  - Joint specific position control
  - Joint specific impedance control
  - Cartesian impedance control
- Fieldbus infrastructure
- Safety
  - T1/T2 (Test-Mode) support
  - adjustable safety margins
- Portability facilitated

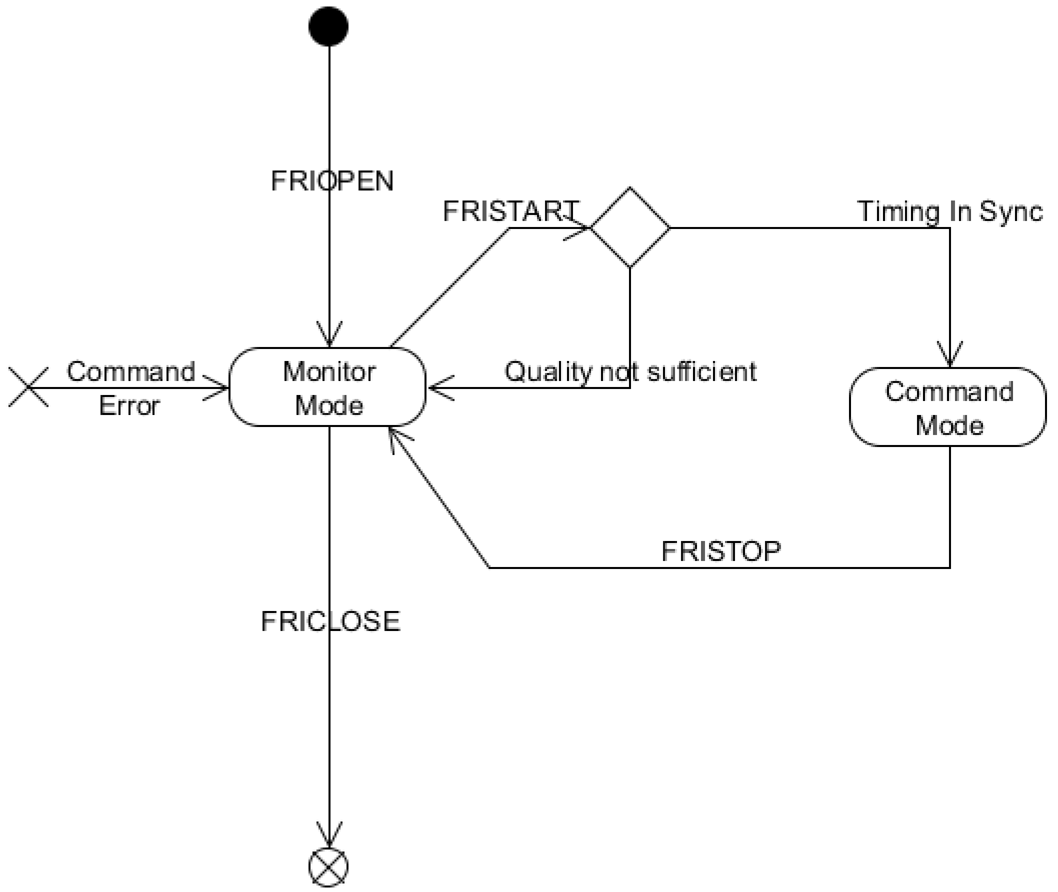


Connectivity & portability

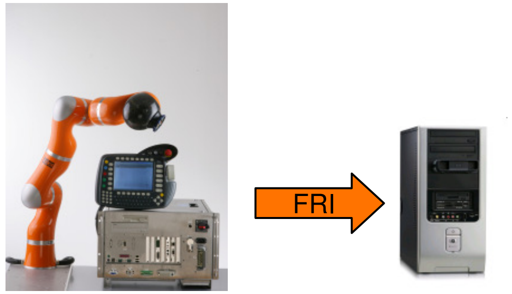
- Simple UDP socket communication, simple binary communication.
- C++ sample code (“SDK”) included to facilitate porting to several different operating systems.
- Operating systems already been used in early tests (also by customers):
  - QNX
  - VxWorks
  - Windows
  - Linux/ RT/ RTAI



FRI Concept: State machine



FRI Concept: KRC to Remote Computer



KUKA LWR + Controller (KRL) Remote PC

Administrative data

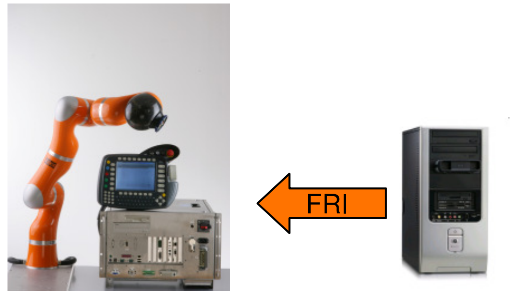
- Timing statistics
- Control strategy
- Timestamp
- State of the drives (on/off)
- User defined variables
- „House Keeping Data“, like temperatures etc.

Robot data

- Measured position
- Commanded position (interpolated from controller) for motion superposition
- Measured torque
- Estimated external torque
- Numerical dynamic model (Massmatrix, Gravitational effects,Coriolis..)

FRI Concept: Remote to KRC – Position Control

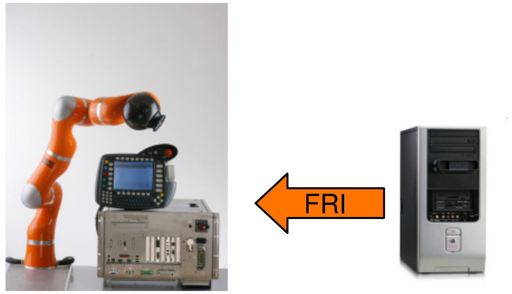
- Desired position
  - Absolute q_{FRI}



KUKA LWR + Controller (KRL) Remote PC

- User defined variables

FRI Concept: Remote to KRC – Joint stiffness control



KUKA LWR + Controller (KRL)   Remote PC

- Desired position
  - Absolute q_{FRI}

- Superposed torque τ_{FRI}
- Stiffness spring factor k_j
- Stiffness damping factor d_j

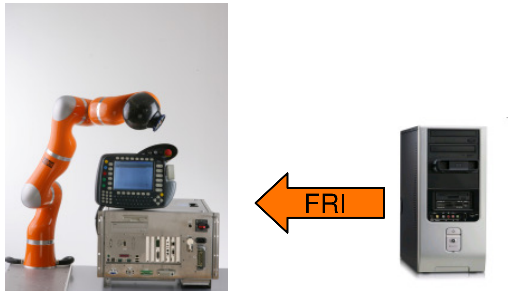
Control law structure:

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

- User defined variables

User can select, which parameters to override via the interface

FRI Concept: Remote to KRC – Cartesian stiffness control



KUKA LWR + Controller (KRL) Remote PC

- Desired Cartesian position
  - Absolute x_{FRI}

- Superposed exerted force/torque at the endeffector FT_{FRI}
- Cartesian stiffness spring factor k_c
- Cartesian stiffness damping factor d_c

Control law will become something like

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

- User defined variables

User can select, which parameters to override via the interface

DEMO: peer-to-peer haptics

Realisation

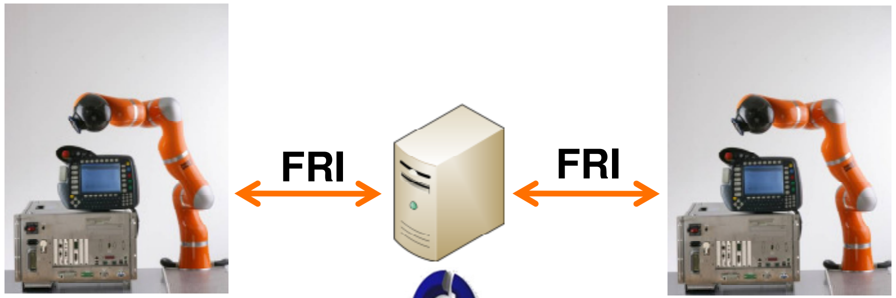
- Orocos on RTAI-Linux platform
- exchange joint positions between two LWRs

Results after 2 days

- One Orocos component to communicate with each LWR using FRI
  - With non-real-time Linux sockets
  - 2 ms to interact with one LWR
  - 5 ms to interact with two LWRs

Multiplication

- FRI on Orocos to be published on www.orocos.org



The Orocos Project
Smarter control in robotics & automation!

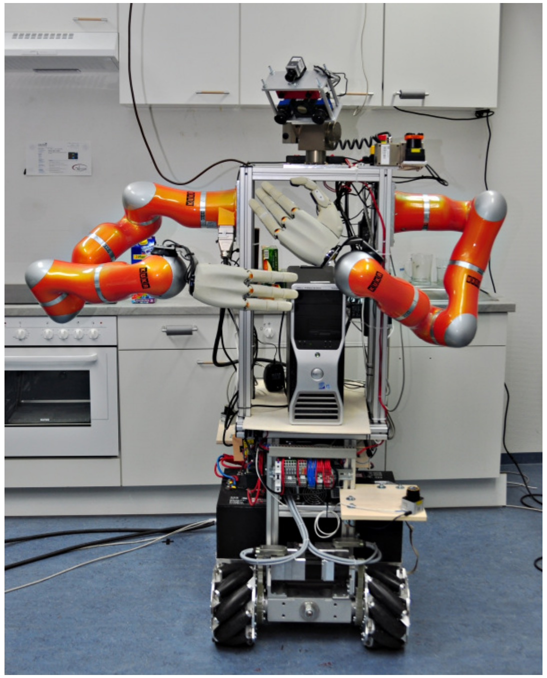
Statement of Prof. Dr. Herman Bruyninckx

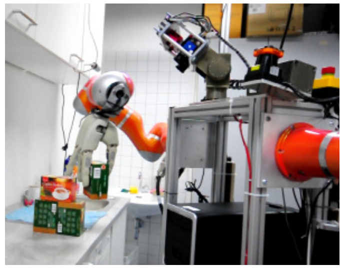
This LWR comes with a **revolutionary** new "constraint based" interface for industrial robot arms: it allows the programmers and users of the arm to not only use the robot in the traditionally proven geometrical way, but also to **specify the desired interaction dynamics** between the robot and the environment or the human. In addition, the arm's motion and dynamic behaviour can be influenced by external third-party control software, via a **fast, deterministic, but cheap and easy** Ethernet connection.

DEMO: Dual-arm robot system

Realisation
- RTAI-Linux platform
- ROS as middleware

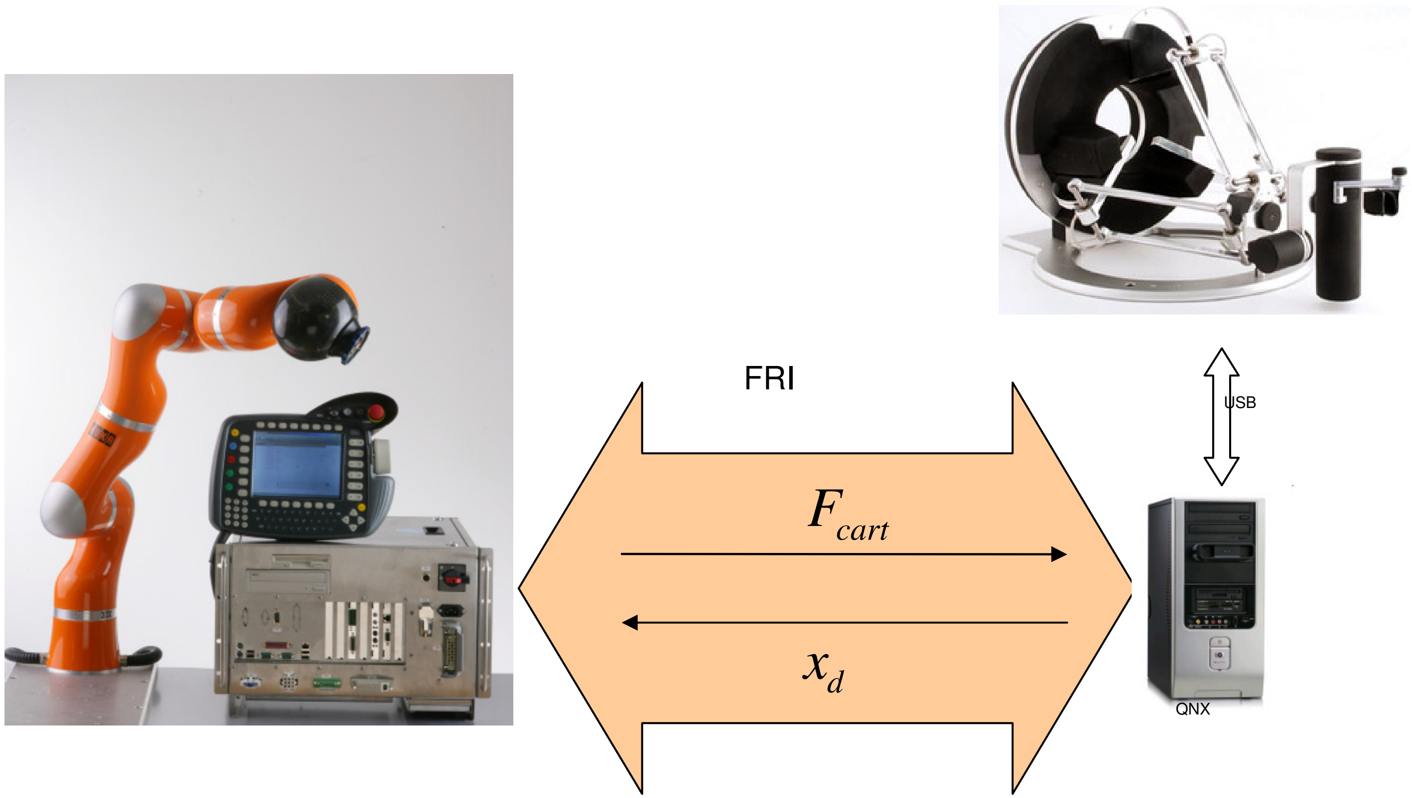
Results
- Mobile manipulation in dynamic non-structured environments → Kitchen
- Advantages of high frequency/low latency control:
  - external control loops become simpler (more linear)
  - extracting features out of the sensor stream
- system can cope with uncertainty of positions of objects and the robot





Multiplication
- via Willow Garage / ROS community

DLR: Omega7 haptic device



- QNX drivers for Omega7 available
- Scaling of motions

KUKA Lightweight Robot with Fast Research Interface: Demo Application

A. Stemmer - Institute of Robotics and Mechatronics, DLR
andreas.stemmer@dlr.de

Conclusions

- The new Fast Research Interface in combination with the KUKA LWR provides researchers with a unique reference platform for robotics research.
- There is no need any more to reinvent the wheel!
- The KUKA Lightweight Robot in combination to FRI is (one of) the best commercially available robotics research platforms.

Thank you for your attention!

Visit us in the exhibition area
Booth 24

Exhibition in Anchorage/AK – ICRA 2010

Good response and attractivity

Good contacts to research customers

- Features of the lightweight robot
- FRI essential for the research customer
- Lot of potential customers are fed up building and maintaining their own hardware – which was developped, since they could not buy some adequate machine

Michael Gerstenberger as new technichal support for the US was presented


HEALTH AND
METALS
PAPER
KUKA


KUKA
AEROSPACE AGRICULTURE APPLIANCE
ELECTRONICS
GLASS
MEDICAL METALS
WIND ENERGY
AUTOMOTIVE
KUKA
CONSUMER
MACHINE
PHARMACEUTICAL
WIND ENERGY

Results of the ICRA Workshop

- No „agreed standard“ middleware-approach
- FRI approach as „light-weight“ interface a feasible solution

- Importance of a scripting language (like KRL)
  - Standard operations, like „PTP StartPoint“ can be executed
  - Researcher can focus on just his special interest

New KRL Features in KR C LR 5.6#5

- State of FRI and Connection Quality as System variable available

```
ENUM FRISTATE OFF, MONITOR, COMMAND, INVALID
DECL FRISTATE $friState

ENUM FRIQUALITY BAD, UNACCEPTABLE, GOOD, PERFECT, INVALID
DECL FRIQUALITY $friQuality
```

- State effective calls return the state of the interface

```
FRISTATE friOpen(); FRISTATE friOpen2()
FRISTATE friStart()
FRISTATE friStop()
FRISTATE friClose()
FRISTATE friShow()
```

- KRL Systemfunction (seamless integration)

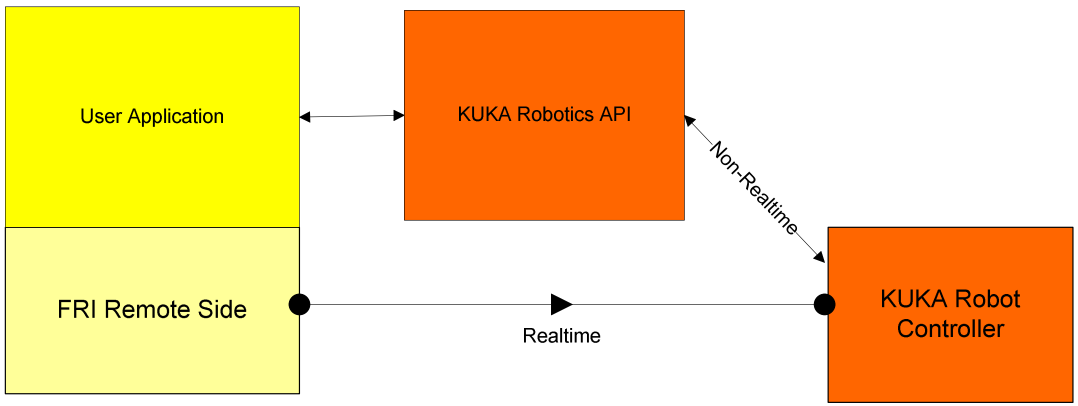
Robotics API & FRI

- Functionality with slight overlap
  - „both can move the robot“
- Robotics API
  - Acyclic Motion commands
  - Full choice and selection of all control modes
  - Program-sequences can be done very easy
  - Proper handshake to robot controller in all (control) state changes
  - Soft-Realtime Option (speed interface)

- FRI
  - Highspeed cyclic motion commands
  - Hard-Realtime Connection up to (guaranteed) 1msec
  - Lightweight UDP socket protocol
  - Superposition of control and motion
  - No program-sequences supported
  - Handshakes for control state changes difficult and very error prone therefore MUST be covered by Robotics API to reduce the pain

Sketch – FRI goes Mobile

- Mobility requires different Hard- and Softwarestructure
- Interplay between RoboticsAPI and FRI
- RoboticsAPI – (Non-Realtime)
  - Selection of Controlmode
  - Basic Motion of Robot (e.g. PTP Home)
  - Selection of FRI
  - Interplay analogous to KRL (-> Userdata, start, stop)
- FRI: Realtime Control from remote
  - Superposition of Basic Motion Features
  - Superposition of Control Modes (within natural physical limits)
- FRI Remote not required to stay in the same process context (or even computer) than Robotics API



FRI: New features?

Speed as command interface?

- Essential?
- Nice to have?

How do we check (from product quality view) the intentions of the user?
Idea: User would specify speed AND final destination position.
Question: Is there a fine-interpolation function, which covers the following properties?

- Choice of initial position and speed
- Choice of commanded (intermediate) speed (-> tangent)
- Choice of goal position (and maybe speed)
- Guaranteed convergence within time period (take me within 100msec to goal)
- Continuous – (monotonous?) – well behaving „convex“
- Bounded w.r.t. position, speed (and acceleration)