ECHORD++ Experiment

CoHRoS - Cooperate Programming of Highly Redundant Robot Systems

Deliverable D1.1 - Requirements Analysis

Christian Emmerich

August 10, 2015

1 Introduction

The major goal of the CoHRoS experiment is to develop an interactive teaching procedure for intuitive programming of high-dimensional robot systems, e.g. such as the welding robot systems built by CLOOS. It targets a working prototype in simulation and a real-world application incorporating a CLOOS robot system, as well as a benchmarking with state-of-the-art programming in a reference application scenario. These aims require strong integration efforts of the partners concerning the development of proper software interfaces between hardware, simulation, and the to-be-developed software components for learning and control. In addition, requirements arise from an application point of view. The usability and usefulness of the provided interfaces must be regarded with respect to the users' needs and tested in a realistic scenario.

The purpose of the document at hand is to present the results of an analysis regarding these requirements which was conducted in the first phase of the project. Furthermore, it briefly presents the targeted technology readiness level (TRL) according to the *Robotics 2020 - Multi-Annual Roadmap* [1].

2 Definition of Reference Application

During the workshop and the kick-off meeting (Task 1.1) responsible researchers from CoR-Lab had the opportunity to discuss typical drawbacks of state-of-the-art programming interfaces with application developers from CLOOS. Most of these shortcomings originate in the trade-off between the simplicity of controlling the robot in world-mode (i.e. Cartesian control of

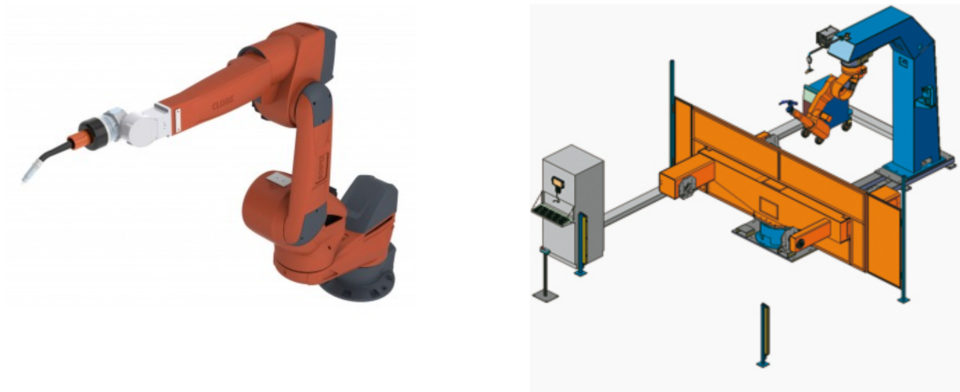

Figure 1: Defined reference scenarios. Left: CLOOS robot with 7 axes for increased workspace reachability. Right: Sketch of second application with a 6-Dof robot mounted on a C-shaped rack.

end-effector) and the user's awareness of the robot posture during joint-mode (i.e. joint by joint movements) - an effect revealed also by our previous experiments on physical human-robot interaction [2]. As a result, the interviewed developers report

- collisions of the robot body with the environment,
- lack of awareness about singular robot configurations and joint limits,
- suboptimal use of additional axes (7th axis).

Based upon this analysis, two reference scenarios have been selected for the further course of the project. The first employs a 7-DoF robot structure with a classical 6-DoF manipulator mounted on an additional external axis for the sake of enlarging the workspace, see Fig. 1 (left). In the second scenario, a 6-DoF robot is mounted on a C-shaped rack and is working in two different working areas separated by a wall as shown in Fig. 1 (right). Both scenarios constitute typical welding applications and address the aforementioned shortcomings of existing programming interfaces. As for the benchmarking, one of these scenarios will be selected to be employed on a real-world prototype and tested with application developers.

3 Software Integration Requirements

3.1 Software Architecture

Regarding the general software architecture the partners decided to re-use software and architecture concepts which have been shown successful in previous projects, the EU-funded project AMARSi and the ECHORD experiment MoFTaG.

The reference systems shown in Fig. 1 will be modeled in the Robot Control Interface (RCI, [3]), a C++-library which provides a set of domain-specific abstractions to represent common features of complex robotics systems. The domain-specific abstractions of RCI are imposed through the domain of motion learning on interactive robot systems. The learning and control components will be organized in the component-based architecture Compliant Control Architecture (CCA, [3]), an event-based component architecture for robotics research focusing on (real-time) control of interactive hardware and machine learning. This C++ library serves as a technology mapping for platforms modeled in RCI as well as component architecture for implementing user applications. The middle-ware utilized for interprocess communication in this project will be RSB, the Robotics Service Bus [4]. RSB is a lightweight, platform-independent and flexible middleware, developed at Bielefeld University in the context of robotics and intelligent systems, providing interfaces amongst others to C++, Java and Python.

3.2 Hierarchical Control Concepts

One design decision of the project is to treat the the whole robotic system including external axes as one highly-redundant kinematic chain, where inverse kinematics redundancy resolution includes all axes. Thereby, hierarchical control concepts such as the control basis framework (CBF) [5] allow to flexibly compose complex motions from individual control aspects, e.g. to synthesize motions using null-space projections from sub-ordinate controllers to a primary controller. The particular implementation which will be used in this project utilizes the CBF C++ library available in [6] and accordingly implements a CCA interface to embed the controller into the data-flow control architecture.

3.3 Exchange of Simulation Models

In order to test the propsed approach in simulation first, the simulation models of the respective CLOOS robot platforms have to be transferred to formats compatible with the CoR-Lab simulation software. The latter consists in the OpenRave[1] simulation environment [7]. The analysis in this context revealed, that the 3D robot models can be exported by CLOOS in the Virtual Reality Modeling Language (VRML, wrl-files) and imported into OpenRave with a custom XML format to compose robot models from simple geometries. Fig. 2 shows a 6-DoF CLOOS Qirox QRC 350 robot model[2] in

[1] `http://openrave.org/docs/latest_stable/`

[2] `http://www.cloos.de/de-de/produkte/qirox/robotermechanik/robotermechanik-classic-handgelenk/roboter-qrc-350-mechanik/`

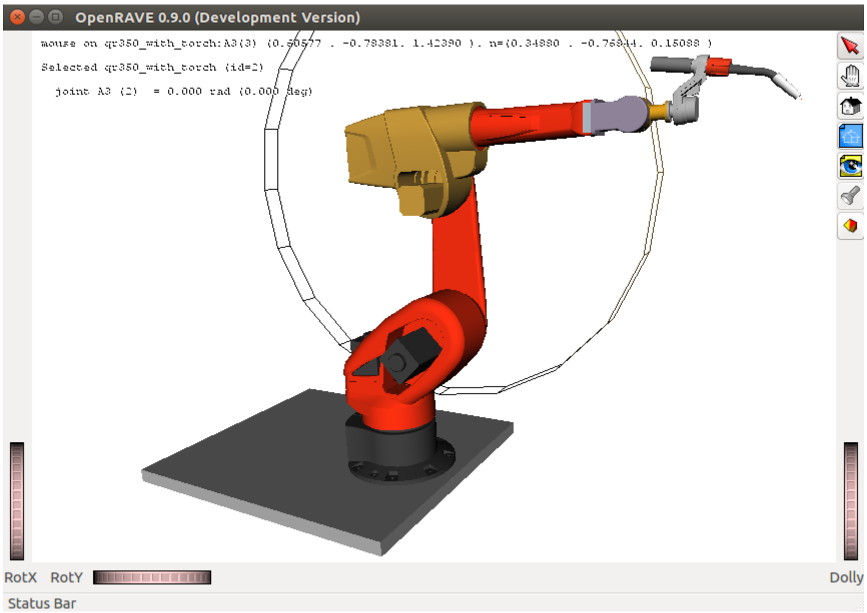

Figure 2: The CLOOS Qirox QRC350 robot simulation model loaded in an OpenRave simulation environment.

an OpenRave simulation environment.

3.4 Interface to Real-World Robot System

Concerning integration of the real-world robot systems, the partners decided on a udp-based communication with a udp-server running on the CLOOS robot controller and a remote client PC running the learning components and controllers. Required information that needs to be transferred between client and server includes:

- robot system configuration/setup
  - number of robot axes and external axes
  - speed and joint limits per axis
  - cycle time
- current robot state/positions
  - current interaction state (assisted yes/no)
  - currently measured joint angles
  - Cartesian coordinates of end-effector in world frame
  - Cartesian coordinates of end-effector in robot base frame
- commanded joint angles to robot

4 Targeted TRL

According to the definitions of the *Robotics 2020 - Multi-Annual Roadmap* [1], currently, the TRL level of multi-stage programming of complex tasks for the Kuka LWR is at level 7 - "System Prototype Demonstration in Operational Environment". As stated in the project proposal, the final demonstration of the experiment and the benchmarking as well TRL 7-8 for ClOOS' reference application and is targeted to be ready for further daily use by the CLOOS developers immediately.

References

[1] Robotics 2020 - Multi-Annual Roadmap. Technical Report Initial Release B 15/01/2014, euRobotics - aisbl, 2014.

[2] Christian Emmerich, Arne Nordmann, Agnes Swadzba, Jochen J. Steil, and Sebastian Wrede. Assisted Gravity Compensation to cope with the complexity of kinesthetic teaching on redundant robots. In *2013 IEEE International Conference on Robotics and Automation*, pages 4322–4328. Ieee, May 2013.

[3] Arne Nordmann, Matthias Rolf, and Sebastian Wrede. Software Abstractions for Simulation and Control of a Continuum Robot. In *SIMULATION, MODELING, and PROGRAMMING for AUTONOMOUS ROBOTS*, pages 113–124, 2012.

[4] Johannes Wienke and Sebastian Wrede. A middleware for collaborative research in experimental robotics. In *2011 IEEE/SICE International Symposium on System Integration (SII)*, pages 1183–1190. IEEE, December 2011.

[5] Roderic Grupen and Manfred Huber. A framework for the development of robot behavior. In *2005 AAAI Spring Symposium Series: Developmental Robotics*. Stanford University, 2005.

[6] Control basis framework. `https://github.com/fps/CBF`. Accessed: 2015-05-25.

[7] Rosen Diankov. *Automated Construction of Robotic Manipulation Programs*. PhD thesis, Carnegie Mellon University, Robotics Institute, August 2010.