Robot Competitions Kick Innovation in Cognitive Systems and Robotics
FP7-ICT-601012

RoCKIn@Work

– Innovation in Industrial Mobile Manipulation –

I: RoCKIn@Work in a Nutshell

Revision: 1.2
Date: March 31, 2014

Editors: Jakob Berghofer, Rhama Dwiputra, Gerhard Kraetzschmar
Contributors: Aamir Ahmad, Francesco Amigoni, Iman Awaad, Jakob Berghofer, Rainer Bischoff, Andrea Bonarini, Rhama Dwiputra, Giulio Fontana, Frederik Hegger, Nico Hochgeschwender, Luca Iocchi, Gerhard Kraetzschmar, Pedro Lima, Matteo Matteucci, Daniele Nardi, Viola Schiaffonati, Sven Schneider

1 Introduction

The main purpose of RoCKIn@Work is to *foster innovation in industrial service robotics.* Innovative robot applications for industry call for the capability to work interactively with humans and reduced initial programming requirements. This will open new opportunities to automate challenging manufacturing processes, even for small to medium-sized lots and highly customer-specific production requirements. Thereby, the RoCKIn competitions pave the way for technology transfer and contribute to the continued commercial competitiveness of European industry.

2 The RoCKIn@Work User Story

RoCKIn@Work considers a medium-sized factory RoCKIn'N'RoLLIn trying to optimize its production process to meet the increasing demands of their customers. RoCKIn'N'RoLLIn is specialized in production of small- to medium-sized lots of mechanical parts and assembled mechatronic products. Furthermore, the RoCKIn'N'RoLLIn production line integrates incoming shipments of damaged or unwanted products and raw material. RoCKIn'N'RoLLIn's operation is quite dynamic; each costumer order is unique. To optimize its production process, RoCKIn'N'RoLLIn factory requires a system with two essential capabilities: 1) mobile manipulation for performing different tasks such as assembly processes, quality controls, order handling, and logistics, and 2) autonomy in switching between different tasks.

RoCKIn@Work is looking for ways towards innovative and flexible manufacturing systems such as required by the RoCKIn'N'RoLLIn factory. The challenges for such a system are set in the subsequent scenario description.

3 RoCKIn@Work Scenario

The RoCKIn@Work scenario description is structured into three parts, environment, tasks, and robots, which constitute the first part of the rules for the competition:

- The environment part specifies the environment in which the tasks have to be performed. This information is also relevant for building testbeds and simulators.
- The tasks part provides information on the tasks the participating teams are expected to solve through the use of one or more robots and possibly additional equipment. This information tells teams what to prepare for.
- The robot part specifies some constraints and requirements for participating robots, which mainly arise for practical reasons (size and weight limitations, for example) and/or due to the need to observe safety regulations.

3.1 RoCKIn@Work Environment

The RoCKIn@Work environment uses a scaled-down environment typical for a small- to medium-sized factory production area, including all its environmental aspects like walls, workstation areas, shelves, machinery and supply devices like conveyor belts. The factory depicted in Figure 1 may serve as an illustrating example. More detailed specifications are given in the rulebook of RoCKIn@Work.

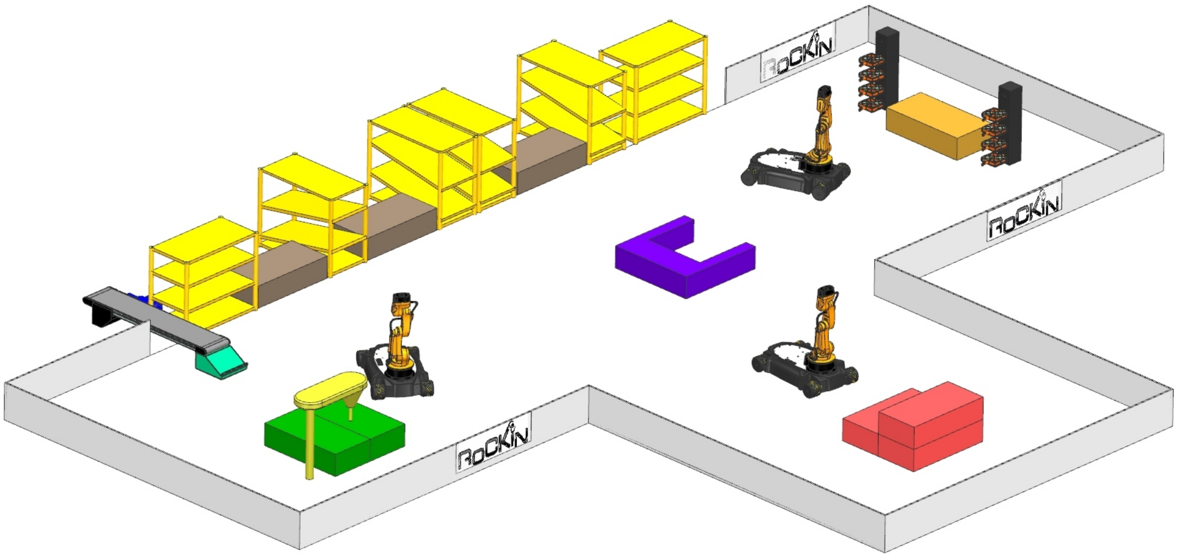


Figure 1: RoCKIn@Work environment with aid tray assembly (orange workstation), plate drilling (green workstation) and collecting for manual assembly (red workstation). The central station (purple) is for quality inspection and parking the robot.

The robot assistant can communicate (via a WLAN network) with various embedded devices which will be installed in the environment:

- A central scheduling system assigning tasks to and receiving reports from robots
- One or more quality control cameras (at conveyor belt and drilling workstation)
- A drilling machine or other machines and robots
- A conveyor belt: an object can be directly picked from the moving belt. If not taken, objects fall from the belt into a drip tray in some random position, from where the robot can take them.
- A force fitting machine; in the first competition this will be done by a human worker.

3.2 RoCKIn@Work Tasks

The first version of the competition will focus on three representative tasks, which illustrate the abilities required for production robot assistants. Together with further tasks described in future versions of the competition, the assembly of a complete drive axle for a mobile robotic platform (Figure 2) can eventually be executed. The assembly consists of the parts listed in Table 1.

1. ***Prepare assembly aid tray for force fitting:***The robot collects the bearing boxes (part-ID: AX-01) from stock and delivers them to workstation #1. For the following production steps the bearing boxes have to be inserted into an assembly aid tray at this workstation. The trays have an identifier (QR-Code), which must be scanned to start with the lot. In the following production steps the identifier is reused to monitor the production process and to quickly react to occurring errors.

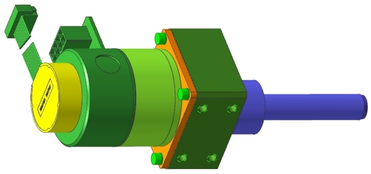


Figure 2: Product of assembly: a drive axle for a mobile robot.

Table 1: List of parts for assembling the drive axle

Part ID	part name
AX-01	Bearing Box
AX-02	Bearing
AX-03	Axis
AX-04	Shaft nut
AX-05	Distance tube
AX-06	Cover Plate
AX-07	Cover Plate Machined
AX-08	Pre Assembled Bearing Box
AX-09	Motor with Gearbox and Encoder
AX-10	Screw Pack 1 (mounting cover plate to bearing box)
AX-11	Screw Pack 2 (mounting cover plate to motor
AX-12	Screw Pack 3 (mounting bearing box to chassis)

Furthermore, the QR code is used as reference number to track the customer-specific assemblies throughout the production process.

The robot should grasp the assembly aid tray and put it on its transportation rig. At workstation #1, the tray has to be plugged into a rig. Then the robot can fill the tray with bearing boxes. The tray has a loose fit for easy filling. The robot has to schedule the order of collecting the items required for filling the tray. The filled

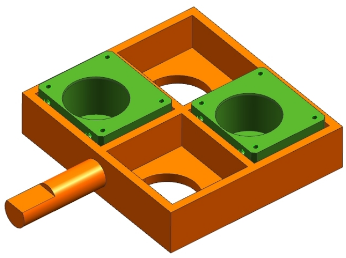
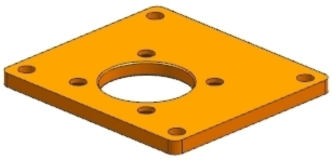


Figure 3: Assembly aid tray for bearing boxes

tray is then inserted into the machine and the bearings are then force fitted into the bearing boxes by the machine (or by a human) at workstation #1. After finishing a tray the machine has to be emptied and the tray has to be transported to the next workstation or to the stock. The bearing boxes with mounted bearings stay in the tray for nearly the rest of the production. Workstation #1 can hold only a limited number of unfinished and finished trays in special rigs.

2. ***Plate Drilling:*** This task simulates handling an incomplete or faulty delivery from an external component supplier. The cover plate of the bearing box has eight holes for connecting the motor with the bearing box. The four central holes need to have a cone sink. Unfortunately the supplier forgot to drill these sinks. The robot has



(a) Cover plate

(b) Cone sink driller

Figure 4: Task "*plate drilling*"

to transport a bundle of cover plates in a customized box to workstation #4. The plates are continuously delivered in an unsorted manner by the conveyor belt next to the shelves. They have to be picked up by the robot and presented to a quality control camera. Depending on the feedback received by quality control, the robot has to sort the plates into one of three categories of file card boxes: unusable, faulty and perfect. Faulty (meaning: all holes are in, only cone sinks are missing) cover plates are carried to workstation #4, then taken out from the box and put onto the workstation table. The cover plate is inserted into a rig for drilling the four missing cone sinks. After the drilling process, the robot has to take the cover plate out of the rig and into a new box for machined cover plates.

3. ***Fill a box with parts for manual assembly:*** The robot composes boxes with parts for the manual, final assembly of the drive axle. The boxes have no special subdivisions (as shown in Figure 5); they only have foam material at the bottom to guarantee safe transport. Therefore, the robot has to plan the order of collecting the parts to arrange the parts next to each other. The parts required for a "drive

Figure 5: Assembly box with subdivisions

axle" include:

- a motor with gearbox and encoder, part ID AX-09
- a machined cover plate, part ID AX-07
- a pre-assembled bearing box (taken from the aid tray now) (part-ID: AX-08)

In addition, screws are needed for this assembly step. These commodity items have to be always available in sufficient quantity. More part sets can be defined for increasing the complexity level of scheduling. The robot picks up a new empty box, scans its code and associates the code with the ordering number. Then it collects the requested parts and delivers the boxes to workstation #5. At the workstation, a worker takes care of assembly. After finishing the drive axle, the worker puts it back to the same box and triggers the status to the central scheduler. The robots scans the box and takes it back to the stock. During the collection process, errors may occur: stock boxes may be empty, parts may be in the wrong shelf and may have to be searched. These errors have to be reported and/or solved by the robot. To increase complexity and to have more variety the set of required parts may vary in number and configuration.

3.3 RoCKIn@Work Robots

The RoCKIn@Work competition is designed for robot platforms with mobile manipulation capabilities such as the KUKA youBot (Figure 6). The robot must be operated in fully autonomous mode when completing each task, i.e. neither power supply via cable nor any kind of teleoperation is permitted. Additionally, the robot used for the competition should consider the competition arena which is a scaled down version of the typical small- to medium-sized factory production area. Networked sensor systems will be provided by the organizer to provide additional input for the robot perception in completing different tasks. The participating teams may use an external PC to provide additional computational power to the robot.

Figure 6: The youBot robot platform

4 RoCKIn@Work Task Benchmarks

In this section, more details on the environment settings, benchmark procedure and the performance criteria for the tasks in Section 3.2 are presented. For all task benchmarks described in this Section applies: The environment dimensions are known to the teams beforehand, but on site mapping is recommended to get accurate data.

4.1 Assembly Aid Tray Task Benchmark

Task: The robot prepares an assembly aid tray for force fitting of bearings to bearing boxes. The robot is presented with a labeled assembly aid tray (QR code). The label allows the robot to understand which bearing box is suitable for the assembly aid tray. The robot proceeds with collecting the required bearing box and places them in the assembly aid tray. In this task, another robot is also moving inside the environment which needs to be avoided.

Information Provided to the Team: Teams are provided with the IDs and 3D models of the aid-trays to be processed. The bearing boxes are placed in storage areas which are known to the team. The navigation details of any other robot in the environment (target location, planned paths, etc.) are unknown to the team.

Expected Output or Behavior: The robot is expected to fill an assembly aid tray with the appropriate bearing boxes. The ID of the assembly aid tray and the number of bearing boxes successfully placed into the tray have to be reported to the competition logging system.

Performance Criteria: Robot performance will be evaluated according to the accomplishment of the task (e.g. objects correctly identified and manipulated) and its complexity (to account for different levels of difficulty in the possible assembly). Other performance criteria used to rank teams are referred to the time required to perform the task (which is both affected by task scheduling efficiency and navigation ability) and safety of navigation (i.e., the robot should not bump into obstacles being those fixed or mobile).

4.2 Plate Drilling Task Benchmark

Task: The robot performs quality control for the cover plate of the bearing box. By utilizing an external camera the robot examines each bearing plate for a possible defect. The cover plates will be delivered on a continuously moving conveyor belt. The robot should pick up one plate at a time from the conveyor and present it to the external camera. The image received from the external camera is then used to determine the condition of the plate. The team's robot has to evaluate the image. This inspection will result in a decision of the plate on hand being either unusable, faulty or perfect. Based on the examination result, the robot places the plates into an appropriate file card box and the robot will proceed with the examination of the next plate.

After finishing inspection of all plates, the robot continues with the faulty plates to the drilling machine. The robot takes out a plate from the file card box, inserts it to the drilling rig and starts the drilling machine by sending the required command. After drilling has finished, the robot presents the result to the camera fixed at the workstation for a second quality inspection.

Information Provided to the Team: The teams will be provided with all possible states of a cover plate (e.g. no cone sinks, only a few cone sinks, no holes at all).

The external camera and its interface description, and the drilling machine and its software interface description will be provided by the organizers.

Expected Output or Behavior: The robot is expected to sort the bearing plates based on their condition. The robot is expected to process the faulty plates together with the drilling machine. The robot needs to report the number of total plates examined and the number of plates in each category (unusable, faulty, perfect).

Performance Criteria: Robot performance will be evaluated according to the accomplishment of the task (e.g. objects correctly identified and manipulated) and the accuracy in doing this (e.g. precision and recall in faults). Other performance criteria used to rank teams are referred to the time required in to performing the task (which is both affected by task scheduling efficiency and manipulation ability).

4.3 Fill a Box Task Benchmark

Task: The robot task is to support a human operator in assembling a product. The operator provides the name of the product through a computer console and the robot collects the parts required to assemble the product. The required parts are stored in different storage areas and the robot should determine an effective path for collecting these parts. The robot can use its transportation platform for temporarily storing the parts before returning to the human operator. In this task, some of the storage areas will be occupied by other robots and the robot needs to decide whether to wait or to return after collecting other parts. This can be simulated through a temporary blocking by an obstacle of the particular storage area to keep the task predictable.

Information Provided to the Team: The team will be provided with a list of products and the parts required to assemble each product. The storage area of each part will be known to the teams.

Expected Output or Behavior: The robot is expected to collect the set of needed parts based on the order. The robot is expected to bring the box with the set of parts to the manual assembly station. The robot needs to report the number of boxes collected and delivered, and its plan for collecting the parts.

Performance Criteria: Robot performance will be evaluated according to the accomplishment of the task (e.g. objects correctly collected). Other performance criteria used to rank teams are the time required to perform the task (which is both affected by task scheduling efficiency and manipulation ability) and the navigation safety.

4.4 Benchmarking Data

The robot notifies the benchmarking infrastructure when it has completed subtasks or steps and it is required to log task-related data to perform task execution benchmarking. These data are described in details in the competition rule book; some examples include: the plan and the scheduling of the activities, the trajectory executed by the robot manipulator during the whole task, the list of recognized objects, the images or point clouds used to perform such object recognition, and so on. For the all of these the desired format will be specified together with tools for logging and time synchronization.

5 RoCKIn@Work Functionality Benchmarks

As part of RoCKIn@Work, we will also perform various benchmarks that assess the robot's performance with respect to particular functionalities. These benchmarks aim to evaluate the functionalities embedded in a fully working system and on a real robot. In order to achieve this, the task to be performed will be narrowed to include, if possible, only a single functionality or the minimal subset of functionalities needed for performing the task. The focus is on repeating the task several times, often with varying the object playing a central role (the object to be perceived, manipulated, etc.) or the environment (varying lighting conditions, for example).

5.1 Object Perception Functionality Benchmark

A certain number of objects, selected from the list of RoCKIn@Work items, will be positioned, one at the time, on a work area located directly in front of the robot. For each object presented to it, the robot has to perform the following activities:

- Object detection: Perception of the presence of an object on the table and association between the perceived object and one of the object classes.
- Object recognition: Association between the perceived object and one of the object instances belonging to the selected class.
- Object localization: Estimation of the 3D pose of the perceived object with respect to the surface of the table.

5.2 Visual Servoing Functionality Benchmark

A number of objects will be positioned, one at the time, on a table located directly in front of the robot. The set of objects that will be presented to the robot is known to the robot. For each object presented, the robot has to perform the following activities:

- Identify which object has been presented, and provide this information.
- Position the end effector with respect to the object in order to grasp the object. Limits on possible grasps may be specified for some classes of objects.
- Grasp the object, lift it, and notify that grasping has occurred.
- Keep the grip on the object for a given time, then set the object down, release the object and move the end effector away from it.

Note: For this functional benchmark, the use of any perception except vision and (if available) touch is excluded.

5.3 Planning and Scheduling Functionality Benchmark

For this functional benchmark, teams are required to provide and use their own planning and scheduling software to generate the basic tasks to be assigned to two youBots that are provided as part of the test bed. The basic tasks executed must, collectively, lead to the solution of the following assembly problem: a set of components has to be fetched from their locations and prepared in an predefined way.

The software provided by the team also has to monitor and verify the correct execution of the basic tasks by the robots that have been assigned basic tasks. When necessary, the system should modify the plan and schedule.

