Funded by the European Union

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

Description of Ground-truth system - v1

I: Part 1: Design
II: Part 2: Practical Experiences

Deliverable:	D-2.1.7
Due Date:	June 30, 2014
Latest Update:	July 7, 2014
Revision:	1.0

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

Contents

Document Overview

This Deliverable is dedicated to describing the system(s) used by RoCKIn to collect ground truth data. As the following chapters of this document will show, in the contex of RoCKIn this mostly means describing the design, implementation and real-world performance of the RoCKIn system for the collection of pose and trajectory data about robots and objects that robots interact with.

RoCKIn is a project focusing on using robot competitions as benchmarking tools. Whatever methodology is used for benchmarking, it always requires that two conditions are verified, namely that a reference is available, and that the performance of the system under test has been observed. Both the reference and the observations can vary widely in their being more or less precisely defined, and more or less objective. For instance, if we consider the case where the system to be benchmarked is a robot, the reference may be a set of unspecified criteria that a judge (i.e., a person in charge of assessing the performance) formed in her own mind over time about how good a certain class of robot should be at some task; or it may be a set of mathematical properties that some aspect of the robot's performance (e.g., the trajectory) is required to possess. Similarly, the performance of a robot can be observed more or less precisely. For instance, the observation used to assess the performance may be the memory of it that a judge formed while the robot was executing the assigned task; or it may be some type of measured data about the execution of the task (such as its duration as measured with a stopwatch).

The RoCKIn project aims at performing "benchmarking through competitions" while keeping an approach as much objective as possible, the limiting factor being the desire to keep the competitions' key elements of thrill and fun. In particular, such an objective approach requires that the observation of the performance of the participating robot systems is executed as precisely and objectively as possible. Whenever this is feasible, the observation should be based on reliable measurements of physical reality, with which the internal representation of the same reality provided by the robot systems can be compared. Such reference measurements, when available, are called **ground truth**.

This Deliverable explains the structure and setup of the RoCKIn system for collecting ground truth. More generally, it describes how the data needed for the RoCKIn benchmarks is collected, stored, and processed. Such data are identified, collectively, by the term **data for benchmarking**. Deliverable D2.1.7 is the first version of this document (v1), written before the first edition of the RoCKIn Competition (to be held in Toulouse, November 2014). An extended version (v2) of the document, with an analysis of the experience of the Competition, will be published at month 30 as Deliverable D2.1.8.

The remaining part of this document is organized in two parts. Part I describes the design of the RoCKIn system for the collection of data for benchmarking, and particularly of ground truth data. Part II deals with the practical experience in data collection gained by RoCKIn. Part II of the Deliverable is composed of two chapters. Chapter 2 takes into consideration the first RoCKIn Camp (Rome, January 2014), while Chapter

3 refers to the first RoCKIn Competition (Toulouse, November 2014). As the due date for Deliverable D2.1.7 precedes the 2014 RoCKIn Competition, for the time being only includes a description of the data types used to collect data produced by the robots participating to the RoCKIn Competition. The final specifications of the data types used for data collection (which may be modified after the experience of the Competition) will be included in Deliverable D2.1.8 which, as already noted, corresponds to the final version of this document.

Part I

Design

Chapter 1

Data Acquisition for the RoCKIn Competitions

Benchmarking the performance of a robot system requires that suitable *data for benchmarking* are collected. Benchmarks define what data have to be collected, but there are several practical aspects of the data collection process that can influence the quality and significance of the benchmark: these must be carefully considered to obtain satisfactory results. This chapter is dedicated to pointing out such aspects, and to explain briefly the approach of RoCKIn towards them.

1.1 External and internal data for benchmarking

The description of each RoCKIn benchmark include the specification of the required data for benchmarking (also called "benchmarking data" in the following). Such data describe the behaviour of the robot system under test for what concerns the aspects that the benchmark is designed to assess, and comprise two types of data.

- **External benchmarking data** are collected by the testbed or by the referees: for instance, by tracking the trajectory of a physical point of the robot, or by verifying if a given effect has been obtained by the robot. No active participation by the robot (nor by the team running the robot) is required to collect these data, and no requirements are set on the structure and functions of the robot in order to enable the collection of such data. The only required change to the robot may be the installation of distinctive markers or patterns on it to allow external localization, if required.

- **Internal benchmarking data** are collected by the robot system. For benchmarking, such data are streamed by the robot to the testbed, or recorded and later transferred to it. An example of internal benchmarking data is the estimate of the robot's own pose as provided by the robot's own self-localization system.

Internal benchmarking data force the robots under test to have an internal representation and/or data interfaces that comply with RoCKIn's specifications[1]. Constraints

[1]This does not usually force the robot to comply to a given internal representation. The only requirement is that the robot must be able to produce and record information according to a given representation. For instance, to benchmark self-localization it may be necessary for the robot to estimate its own location, which would rule out a robot which moves randomly through the environment.

such as these put an additional burden on participating teams, because they may require modifications to the robots; for some robots, such modifications may even prove to be unfeasible or too disruptive for their normal operation. For these reasons, in RoCKIn the use of internal benchmarking data is kept to a minimum by careful choice and design of the benchmarks.

1.2 Ground truth

The term **ground truth**, or **GT**, is used for reference data about the actual behaviour of a robot system, collected using highly accurate sensors and systems independent from those of the robot. For an ideal robot and environment, the robot's own perceptions and estimates would be in perfect accordance with ground truth; in a real setting, differences between what the robot perceives and estimates and ground truth data will occur, due to imprecisions in sensors and processing. Such imprecisions affect both the robot and the GT collection system, of course; however, if the GT system's imprecisions are much lower than the robot's, the data provided by the former can be used to estimate the quality of the data produced by the latter.

According to the classification of Section 1.1, in the context of RoCKIn, GT data correspond to *external benchmarking data*. For benchmarking, GT data can be used on their own or in conjunction with *internal benchmarking data* (provided by the robot). More precisely, RoCKIn benchmarks assess robot performance by using the following methods:

1. by processing GT data only (either by direct application of suitable metrics or by comparing GT data about the module under test with GT data about a reference module);

2. by jointly processing GT data and internal benchmarking data;

Only the second of these methods requires that the robot under test is compliant with RoCKIn-provided specifications about internal representations and/or interfaces: the other do not put any constraint on the robot, and is, therefore, preferred.

1.3 Trajectory data

Among the ground truth data that RoCKIn will acquire during the RoCKIn Competitions are *trajectory data*. Ground truth about trajectory is required every time a benchmark requires an evaluation of position estimates (either of an element of the environment or of a part of the robot) provided by the robot system under test.

In RoCKIn, trajectory data are collected by making use of a *motion capture system*. Systems belonging to this category are mostly used for cinematography, but their features are such that they are increasingly employed in robotics laboratories as well (e.g., to accurately track flying robots). They employ cameras to acquire video data about a volume of space where the objects to track move. Such objects are fitted with reflective *markers*; the video from the cameras is processed in order to estimate the position in space of the markers. By knowing (thanks to prior calibration) where the markers are placed on the objects, the system is therefore able to provide an estimate of the spatial pose of the objects.

The motion capture system chosen by RoCKIn is the OptiTrack system, manufactured by Natural Point, Inc. The choice of this system has first been narrowed down (using performance criteria) to choosing between similar products from Natural Point and its competitor Vicon; subsequently, the final choice was made based on cost-effectiveness. The OptiTrack setup used by RoCKIn includes: 12 infrared cameras (model S250e) with built-in IR LED illuminators[2], the Motive software package[3], as well as the necessary IR markers, mounting gear, network components, and data-acquisition PC. For RoCKIn, only the capability for tracking *rigid bodies* of the OptiTrack system is used: in fact, while the system can be also used to track deformable objects (such as the body parts of human actors), this capability is not sufficiently useful for RoCKIn to justify its considerable additional cost. The RoCKIn motion capture system is able to reliably track multiple rigid bodies at a high frame rate (250 fps) over a volume of up to a few tens of cubic metres (depending on obstacles and camera positioning), with millimetre-level accuracy in position tracking. This system has been chosen because it is a good trade-off, considering both the requirements and the budget of RoCKIn. For the forthcoming 2014 RoCKIn Competition, agreements are in progress to temporarily augment the motion capture system with additional OptiTrack elements (e.g., cameras). According to the experiences described in Part II, such an augmentation would in fact improve the coverage and reliability (in terms of reduction of drop-outs in pose collection) of the system.

1.4 Installation of the motion capture system

Here, we provide some information about how the RoCKIn motion capture system will be set up for the 2014 Competition. What follows is intended as a short guide for researchers interested in similar systems. Chapter 2 will provide additional information about both the setup of the system and its limitations.

Before the motion capture system can be used, it needs to be installed, pointed, and calibrated. Calibration makes use of a special "wand" and a procedure that requires to move the wand over the observed volume, while the Motive software acquires and processes calibration data[4]. Once familiar with such procedure, an operator can calibrate (or recalibrate) the system in a quarter of an hour. Since recalibration is required every time a camera is moved, unless cameras are installed in a very stable way it is advisable to recalibrate fairly frequently (e.g., once a day).

Installing and pointing the cameras requires considerably more time than calibrating the OptiTrack system. To give an idea about this to the reader, if the environment includes many obstacles (such as the RoCKIn testbeds) a motion capture system similar to the one used by RoCKIn requires several hours to be installed and optimally pointed. This is true even if the work is done by people experienced in the setup. For this reason, it is important that the mountings are as stable as possible: repositioning is in fact very time-consuming. Some guidelines about what "optimally pointed" means will be provided by the remainder of this chapter, and especially by Section 2.3.

Each Natural Point S250e camera produces square images of 832 by 832 pixel; the field of view is 56 degrees. For reasons that will be explained shortly, it is absolutely necessary that significant superposition occurs between the fields of view of the cameras.

[2] `http://www.naturalpoint.com/optitrack/products/s250e/`

[3] `http://www.naturalpoint.com/optitrack/products/motive/`

[4] Additional information (including videos) about this calibration procedure is available through the Natural Point website.

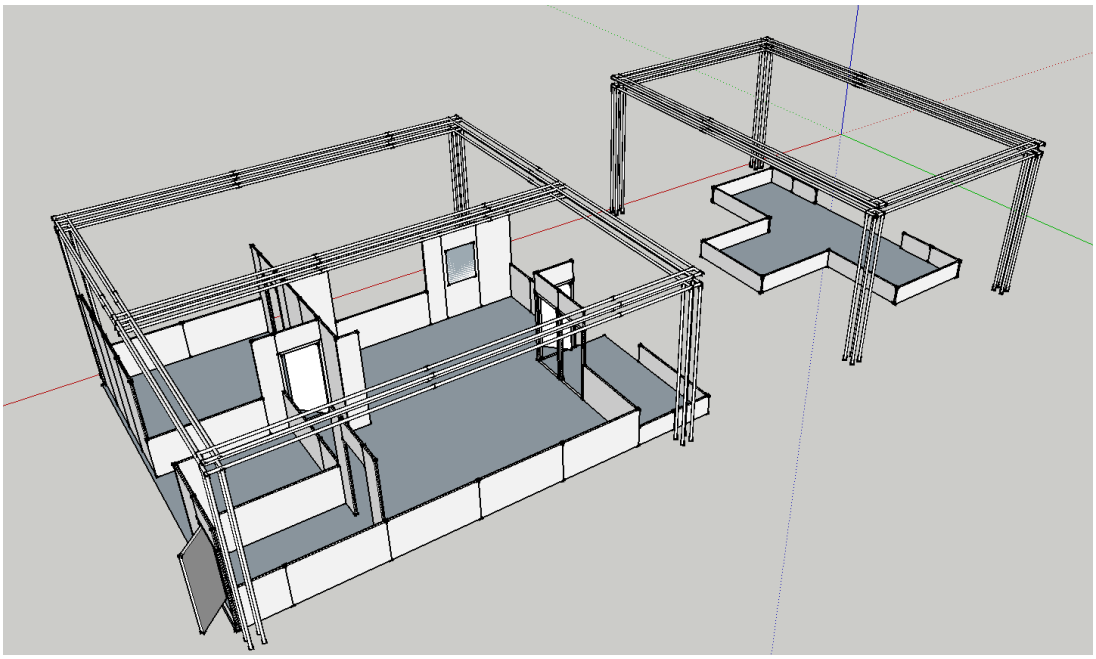

Figure 1.1: CAD sketch of the testbeds, showing the trusses dedicated to supporting the motion capture cameras.

Within limits, superposition can be increased by moving the cameras away from the observed volume. However, maximum distance from the markers is limited by the fairly low resolution of the image sensor: as the image of each marker becomes smaller on the image sensor, the system loses its capability to reliably recognize the marker. Practical experience shows that tracking distances up to 10 m are reasonable. For the RoCKIn Competition, cameras for the acquisition of the trajectory ground truth data will be mounted at around 4 m from the ground, all around the two testbeds (one each for RoCKIn@Home and RoCKIn@Work), on truss-based support structures designed to be stable and vibration-free. A preliminary sketch of the testbed designs[5] with the trusses for the motion capture system is presented in Figure 1.1.

After pointing and calibration, the required rigid bodies must be defined. This corresponds to the operation of mounting markers on the objects to track. Such objects must be rigid: no possibility of relative motion between markers associated to the same rigid body must be present, or the system will cease to recognize it. At least 3 markers per rigid body are required; in practice, given that occlusions occur very often (because the rigid body itself is not transparent, and because of other obstacles) it is advisable to use the maximum allowed number of markers per rigid body, i.e., 7. Even with 7 markers, though, it is not infrequent to lose track of a rigid body unless its shape, the positioning of the markers and the surroundings are favourable.

Markers of different types (adhesive pads or 3D spheres) and sizes are available. We found that adhesive pads lead to poor performance. The best performance for RoCKIn requires use of 3D markers: the larger spheres (diameter of about 18 mm) are the best whenever their size does not prevent their installation.

To be able to estimate the location in space of each marker, at least 2 cameras must be able to see it at any time. This is an absolute minimum, though; in practice, performance

[5]These sketch has been provided by the Dr. Bredenfeld UG company (`http://www.dr-bredenfeld.de/`), which will realize the testbeds for the 2014 RoCKIn Competition.

is very unstable unless at least 3 cameras see each marker. On the other hand, it is not necessary that such 3 cameras remain the same over time: handover from one camera to another occurs with little or no disruption of localization. To estimate the pose of a rigid body, at least 3 of its associated markers must be located by the system. So, to locate a rigid body, at least 3 of its markers must be seen by at least 2 cameras each. Again, this is an absolute minimum: to achieve a robust localization it is best to exceed it significantly.

1.5 Acquisition of data for benchmarking

The metrics of the benchmarks must be applied to the data collected during the benchmarking experiments. This requires that suitable methods to transmit and/or store the data as they are produced must exist. Unfortunately, this problem is not as simple as it can appear: first of all, because the volume of data may be large (for instance, if video data are involved); secondly, because the chosen method to manage the data should be adaptable to the vast majority of the participating robots, whatever their hardware and software architectures.

The first, important technical decision to be taken is between *transmission* and *storage* of data for benchmarking. This is especially relevant for *internal* benchmarking data, i.e. those produced by the robot under test (see Section 1.1). The choice between transmission and storage mainly depends on practical issues. In fact, storage can be managed locally (e.g., by writing on a USB stick) by the device that produces the data, e.g., by the robot system under test. On the other hand, transmission requires that the communication network between robot and testbed has sufficient bandwidth, it is not subject to drop-outs, and so on. Considering that (at least for the 2014 RoCKIn Competition) this network will be a wireless LAN based on commercial technology, operating in an environment where sources of interference are both numerous and unpredictable, it is evident how choosing transmission over storage may introduce significant uncertainties and difficulties. There is a single, important aspect where transmission is preferable to storage: only if the data are transmitted it is possible to apply the benchmark metrics to the data online, *while the benchmarking experiment is running*. Depending on the benchmark, this may provide additional possibilities to make the RoCKIn Competition more interesting to the audience and/or more scientifically significant, besides making the benchmarking process more quick and "dynamic" and therefore more attuned to a quick-paced event. The 2014 RoCKIn Competition does not involve online benchmarking procedures.

The problems related to data transmission do not occur, or can be solved before the benchmarking experiment occurs, for *external* benchmarking data (as defined by Section 1.1). External data for benchmarking include those produced by the motion capture system described by Section 1.3 and all data produced by the RoCKIn testbed itself (including human referees). For these data, every feature (such as format or bitrate) is known beforehand, and additionally wireless transmission can be avoided.

Of course, while choosing between transmission and storage is a key issue, it is not the only one. Some of the other issues will be outlined in the rest of this section. A first problem is that whenever internal data for benchmarking are required, the robot system under test has to perform additional operations in order to prepare and store/transmit them. This (leaving aside the possibilities for cheating that it offers) means that benchmarking imposes an additional resource load on onboard hardware, and this in turn can cause problems or alter the behaviour of the robot system. Therefore, a careful balance between richness of data and system load to prepare them must be sought.

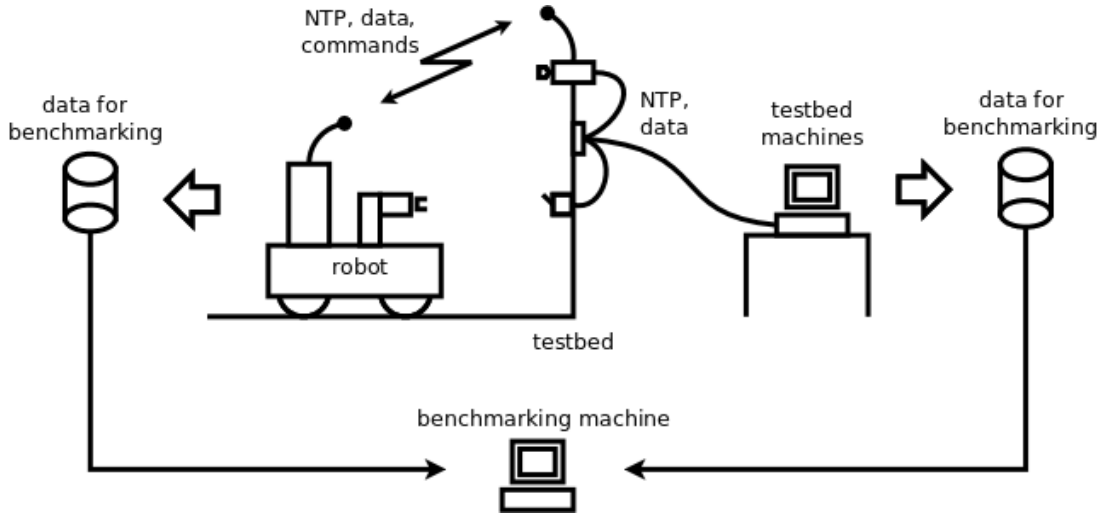

Figure 1.2: Simplified view of the architecture of the RoCKIn data collection and processing system for benchmarking. Among exchanged data, synchronization data ("NTP" refers to the Network Time Protocol used for the 2014 RoCKIn Competition) have been highlighted; Section 1.6 will be dedicated to synchronization.

Another issue is that of format conversions. In fact, data for benchmarking must have a predefined format, that the robot (if the data is prepared internally) has to comply with. Depending on what data formats it uses for its own computations, the amount of work and the system load associated to conversions must be carefully taken into account on a case-by-case basis.

Then there is the problem of interfaces. This has many facets, from the ones that are more easily solved (e.g., use industry standards such as 802.11n for wireless transmission, USB drives for storage, and standard file formats like CSV over text files) to much more complex situations where the execution of the experiment requires that the software of the robot and the benchmarking systems of the testbed interact in real time[6]. For instance, if the software of a robot is based on ROS (Robot Operating System) it may seem a natural choice to configure the benchmarking system as a ROS (sub)system interacting with the robot. However, this approach raises a series of issues, including: (i) How can we set up such interaction? (ii) How much additional work is required from the robot developers to enable it? (iii) What effects the additional elements introduced to provide the interaction have on the functioning of the robot? (iv) Finally, and most importantly: what about robots that do *not* use ROS? The final choice will result from experience; in particular, it will depend on the actual breadth (or lack of it) of the range of technical solutions chosen by the teams participating to RoCKIn.

Taking into consideration the issues described above, as well as the available hardware and software, we can now describe the architecture of the RoCKIn system for the collection and processing of data for benchmarking, as it will be used for the first RoCKIn Competition of November 2014. This architecture, which keeps into consideration the practical experience in data collection done at the 2014 RoCKIn Camp (described by Chapter 2) is shown in Figure 1.2. For the reasons discussed before, the transfer (either network-based or via physical devices moved from one computer to another) of locally logged data is the chosen method to transport benchmarking data to the benchmarking

[6] As already observed, this is not foreseen for the forthcoming 2014 RoCKIn Competition.

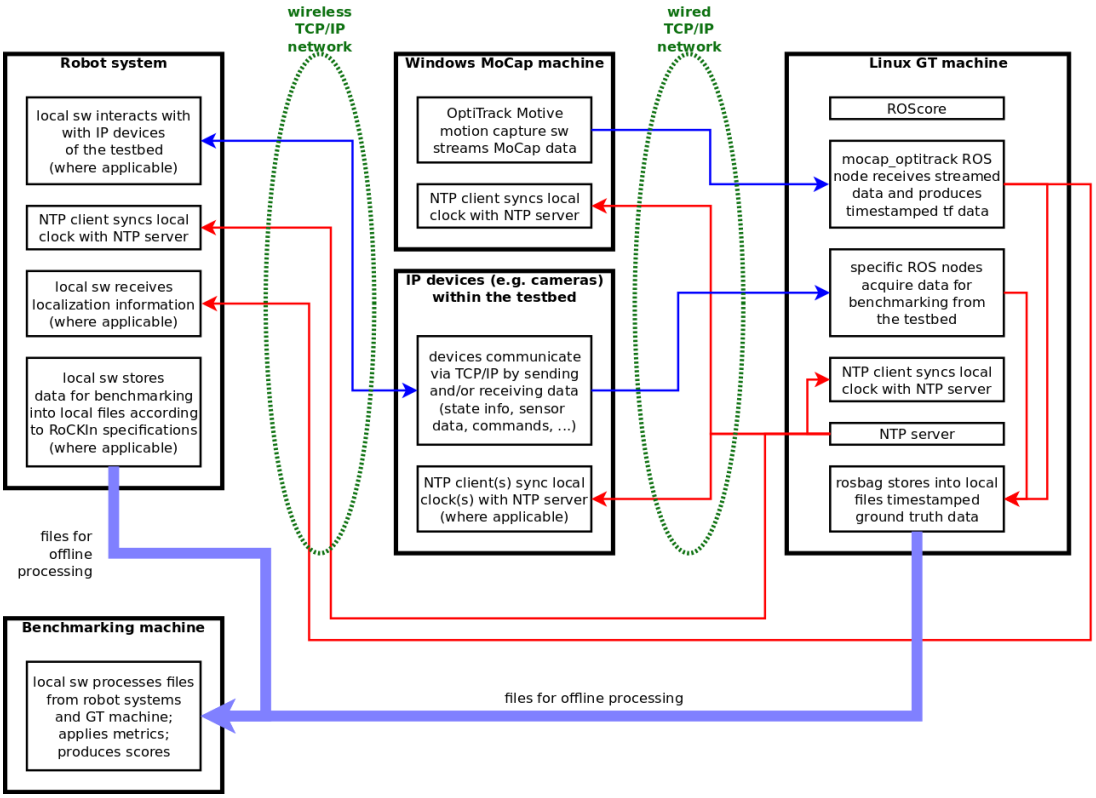

Figure 1.3: Architecture of the RoCKIn data collection and processing system for benchmarking. The modules in this Figure are the same of Figure 1.2.

machine. Figure 1.2 does not contain, for reasons of clarity, any details about the modules involved in the process of generating and using benchmarking data, including how these are distributed over physical machines. Figure 1.3 provides additional information about these modules and their interconnections.

1.6 Time synchronization

For benchmarks that make use of both *external benchmarking data* and *internal benchmarking data* (see Section 1.1), it is important that external and internal data are time-synchronized, since only in this case data for benchmarking can be meaningfully compared. As external data are produced by the testbed while internal data are provided by the robot, time synchronization requires that the clock(s) of the robot are synchronized with the clock of the testbed.

Time synchronization between robots and testbeds is ensured by using established network sync protocols: namely, NTP (Network Time Protocol - `http://www.ntp.org/`) or PTP (Precision Time Protocol - `http://ieee1588.nist.gov/`). Both are networking protocols for clock synchronization between computer systems over packet-switched, variable-latency data networks, and both have a client-server structure.

For RoCKIn, NTP is the preferred choice. Being a more "mainstream" protocol, NTP is very well supported over the widest range of systems. The main difference between NTP and PTP lies in their accuracy, which is much higher for the second. However, for reasons that will be explained shortly, PTP will be taken into consideration only if

NTP will have proved to be not sufficiently accurate over the course of the first RoCKIn Competition.

Within a LAN (i.e., without relying on the internet), the typical time accuracy of NTP is below the millisecond. External benchmarking data deal with the macroscopic actions of the robot: for the type of actions required to participate to the RoCKIn Competitions, observable changes over 1 ms time intervals are negligible. In fact, the RoCKIn benchmarks are designed in such a way that extremely high-resolution over time in observing robot actions is not required. For this reason, NTP is expected to fulfill the needs of RoCKIn comfortably. As an example, a robot moving at a speed of 2 m/s (which corresponds to around 7 km/h, i.e., the speed of a human walking briskly) changes its position by 2 mm over 1 ms. Given that the tracking system has an accuracy of millimeter level, a time sincronization within 1 ms is considered fully acceptable.

PTP significantly improves on the performance of NTP. Over a LAN, its typical accuracy is below the nanosecond, thus greatly exceeding the expected needs of RoCKIn. However, being much less widespread than NTP, PTP is not so widely supported. More importantly yet, PTP has a more aggressive approach towards manipulating the system clock with respect to NTP. This can be a drawback when fluctuating network links are involved: a typical situation when wireless networks are used, especially in the very noisy context of a robot competition where several wireless LANs are active over the same, small, area. For these reasons, as anticipated, PTP will be considered for RoCKIn only if NTP fails.

Synchronization requires that each robot participating to a RoCKIn competition, where both internal and external benchmark data are used, runs a NTP (or PTP) client. Fortunately, well-established off-the-shelf clients (with small computing power footprint) are already available for all major operating systems. For this reason, forcing teams to install such clients on their robot's computers will require a very limited effort and is not expected to compromise their functionality. The NTP or PTP time server providing time correction data will be a part of the testbed, and the robots under test will be required to connect to it through the Competition's wireless network.

Part II

Practical Experiences

Chapter 2

Data Acquisition at the 2014 RoCKIn Camp

This chapter has been written after the end of the 2014 Rockin Camp (Rome, January 26th-30th, 2014). One of the objectives of the Camp was to investigate in a real-world setting the feasibility and effectiveness of the benchmarking procedures envisioned for the RoCKIn Competitions. This chapter is concerned with the results of such investigation.

2.1 Camp and competitions

The 2014 Camp was an excellent setting to test some aspects of the benchmarking procedures: in particular, those concerning set up, calibration and data acquisition. While not fully possessing the frantic pace of a competition (where every procedure must be painstakingly prepared beforehand in order not to disrupt the flow of the event), under many points of view the Camp provided a similar environment and similar challenges. In particular, the Camp included:

- competition-like testbeds;
- competition-like interference problems due to the concurrent access to the testbeds by the teams and the benchmarking personnel (in fact, the testbeds also acted as testing areas for the teams);
- competition-like resource allocation problems due to the need to perform simultaneous benchmarking activities on two separate testbeds (RoCKIn@Work and RoCKIn@Home);
- competition-like data acquisition problems (e.g., set up and calibration in real-world conditions, interference from objects different from the robot under test, unexpected disruptions);
- competition-like limitations to the quantity and quality of ground truth data that could actually be collected (e.g., due to limitations in the number and/or positioning of motion capture cameras).

What could not be tested at the Camp is the *processing* of ground truth data, as this was neither defined nor implemented at the time of the Camp. Nonetheless, the Camp

offered valuable lessons regarding the way such processing should be organized in order to proceed smoothly without interfering with the other activities.

It is important to stress that, while the robotics community (and the RoCKIn Consortium in particular) has collected a large experience in running robot competitions, experience in performing objective benchmarking of robot performance in the context of competitions has to be built from scratch. RoCKIn hopes to provide a valuable starting point for that.

2.2 Experience at the camp

As explained in Chapter 1, in principle, RoCKIn benchmarking is based on the processing of data collected in two ways:

- external benchmarking data, collected by the testbed;
- internal benchmarking data, collected by the robot system under test.

During the Camp, both these types of data have been collected. The following of this section describes the aspects of this data collection activity that are relevant to describe the experience of the Camp in terms of data collection for benchmarking.

2.2.1 Data acquisition and logging

During the Camp, external benchmarking data consisted of the trajectories of rigid elements of the robots under test (such as the base or the wrist) and/or of objects that the robots interacted with. For this, the elements to be tracked had to be fitted with IR-reflecting markers which are part of the motion capture system. Figure 2.1 shows some examples of marker installation at the Camp.

Trajectory data have been generated using the camera-based commercial motion capture system already described in Chapter 1 (i.e., Natural Point OptiTrack). Each trajectory had the form of a time series of poses of the relevant robot element. Pose data generated by the OptiTrack system have been acquired and logged by a customized external software system based on ROS (Robot Operating System: the same middleware used by all participating robot systems). More precisely, logged data were saved as *bagfiles* created with the *rosbag* utility provided by ROS. Figure 2.2 shows the architecture of the logging system.

The choice of using a ROS-based system allowed translation of OptiTrack data into geometric transforms published on the `tf` topic of ROS: i.e., precisely the same form taken by the robot's estimate of its own pose. This enabled (subsequent) direct comparison between ground truth trajectories and trajectories estimated by the robots.

Internal benchmarking data have been collected as ROS *bagfiles* as well. In this case the files had to be created by running *rosbag* on the robot. This, as will be better explained later in this chapter, did not prove to be a viable solution for the competition. Firstly, because many teams forgot to perform the manual running of *rosbag*; secondly, because (differently from what happened at the Camp, where all robots were ROS-based) at the Competition it is not possible to take for granted that *rosbag* will be available on all robots.

Subsequent comparison between OptiTrack-generated (i.e., ground truth) and robot-generated poses requires that they are aligned in time, as already explained in Chapter

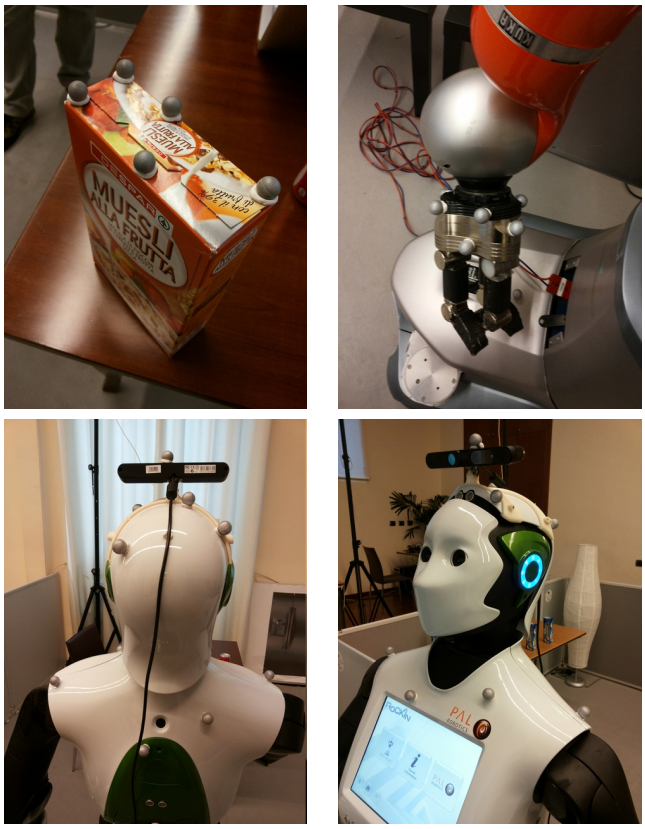

Figure 2.1: Examples of installation of markers (gray spheres) at the 2014 RoCKIn Camp. Clockwise from top left: on a box (the box had to be grasped by robots); on the wrist of a robot; on the back and front of the head and torso of a robot.

1. For this reason, a special ROS node (i.e., a software module compatible with the ROS middleware) was developed and distributed to teams. Such node, which was required to be running both on the robots and on the machine that acquired ground truth data, simply generated (on a specific ROS topic) periodic messages including both machine time ("wall clock") and ROS time, thus enabling subsequent synchronization of the data in the bagfiles with ground truth data. Participating teams were required, before running their Camp demos, to perform manual alignment of their robot's clock with an external NTP time server[1]. The latter had been set up as part of the ground truth acquisition system. In practice, this manual alignment procedure proved to be unreliable, since in many cases teams forgot to apply it.

2.2.2 Physical setup of the motion capture system

For the 2014 RoCKIn Camp the RoCKIn@Home and RoCKIn@Work testbeds were installed in two separate halls. This led to the decision to split the 12-camera OptiTrack system into two 6-camera systems, one for each testbed. Figure 2.3 shows the two motion capture systems as installed at the Camp.

Considering the dimensions of the testbeds, using 6 cameras per testbed lies at the very lowest limit of what -in our practical experience- is feasible with the OptiTrack system.

[1]For Linux-based computers such alignment was done by running the *ntpdate* command; on Windows-based machines a graphical tool is available as part of the clock system; while not tested at the Camp, *ntpdate* should be available in MacOS as well.

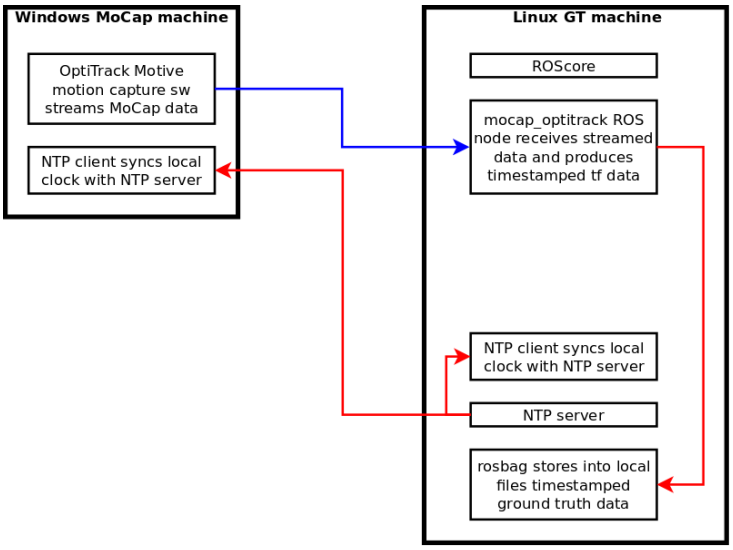

Figure 2.2: System used at the 2014 RoCKIn Camp to acquire and log trajectory data from the motion capture system. This is a subset of the system shown in Figure 1.3.

Below we will describe the issues that arose from that, and what compromises had to be made.

The first issue encountered is related to physical camera installation. The positioning of the testbeds within their respective halls proved to be very challenging from this point of view. For the Camp, cameras were expected to be installed on (3m-high, and therefore having a large footprint) tripods around the capture volume; however, in practice the available space around the testbeds proved to be insufficient for a good installation. The worst situation occurred with the RoCKIn@Work testbed, which -being adjacent to a wall and a passage area- only allowed cameras on 2 of its 4 sides (no space suitable for tripods was available inside the testbed). For RoCKIn@Home things were better (all four sides of the testbed were usable, with some limitation, thanks to the possibility of installing tripods inside the testbed). However, the presence of working areas directly adjacent to the testbed (and thus to the tripods) led to accidental motions of the tripods, and consequent need for recalibration.

Another issue related to camera positioning is the difficulty in ensuring a satisfactory capture volume. Capture volume is the region of space where reliable motion capture is possible, which strongly depends on the number and positions of cameras. As explained in Chapter 1, the OptiTrack system uses markers on the objects (rigid bodies) to be tracked. Figure 2.4 shows two of the rigid bodies fitted with markers which were built at the 2014 Camp.

In order to provide pose data, a minimum set of requirements must be fulfilled by each tracked object, in terms of number of separate markers perceived by the system and number of cameras that perceive each marker. These requirements proved to be problematic for a real-world setting (such as the Camp) where the tracked objects are not optimized for motion capture and the number of cameras is low. This resulted in reduced capture volumes and occasional loss of tracking, especially close to the edges of such volumes.

One interesting thing that has been noted at the Camp is that accidental (small) movements of the motion capture cameras after calibration do not severely impair the precision of the OptiTrack system in localizing objects within the capture volume; conversely, such

Figure 2.3: Motion capture cameras as installed at the 2014 RoCKIn Camp. On the left, camera setup around the RoCKIn@Home (up) and RoCKIn@Work (down) testbed. On the right, closer views of a single RoCKIn@Home tripod with camera (up) and of part of the RoCKIn@Work setup (down).

movements have a strong impact on the capture volume, reducing it. Both effects can be verified by placing the calibration "wand" in the location where the check has to be performed. For what concerns precision such verification is easy: the software provides a specific function, and (most importantly) only spot checks are needed, as precision is quite constant over the capture volume. On the other hand, the only effective way to check if the capture volume has reduced is to explore its boundaries with the "wand": an error-prone, time-consuming process. The final consequence of this situation is that it is necessary to make specific efforts to carefully prevent camera movements. Alternatively, a tight schedule of periodic recalibrations need to be established.

The Camp offered an excellent occasion for building up experience in the optimal positioning both of the motion capture cameras and of the markers on the robots. With careful set up, dropouts in pose data were reduced to a minimum. However, to reach this result it was necessary to reduce the capture volume with respect to the overall volume of the testbeds. With the best available setup, covered testbed volume was around one half both for RoCKIn@Work and RoCKIn@Home. The capture volume for RoCKIn@Home was actually much larger than for RoCKIn@Work, as the overall testbed volume for @Home is around 6 times larger than that for @Work. This better result is due to a larger base area and, mostly, to the possibility to install the cameras all around the

Figure 2.4: Two examples of rigid bodies fitted with markers, and a robot equipped with one of them.

capture volume. This was not possible for RoCKIn@Work, as two out of four borders of the RoCKIn@Work testbed were not usable (one was a passage area, the other a wall.

A final issue related to the physical setup of the motion capture system was the preparation of the rigid bodies that the system had to track. When possible, special 3D "marker sets" were built and fitted to the robots. However, no standardized marker set was used: indeed, in most cases changes to the sets and/or their construction was done on the spot, during the demos. In some cases, and particularly with larger robots, it was found that the only possible approach was to fit the markers (with removable putty) directly on the robot's body. In all cases, the OptiTrack system had to be used during the demos to define the just-realized marker sets as rigid bodies, in order to track them. A side effect of this approach has been that the roto-translation of the intrinsic reference system of each rigid body w.r.t. the reference system(s) of the robot was unknown, and had to be deduced indirectly from acquired data. Such an *ad hoc* approach, while acceptable in a Camp, is not feasible in a Competition where the time schedule is tight and where it is not acceptable that the performance of the benchmarking system changes from robot to robot. Therefore, a general and standardized approach to marker positioning will be used for the Competition.

All in all, the 2014 RoCKIn Camp allowed experimental verification of a wide range of aspects and procedures related to the collection and logging of benchmarking data. Section 2.3 will translate such experience into a list of practical recommendations. These have been already taken into account in the design of the 2014 RoCKIn Competition, and will be in the design of the subsequent 2015 Competition. Moreover, some of the recommendations are not relevant to the benchmarks actually used for the 2014 Competitions. In any case, all recommendations are provided here for the benefit of other research groups planning similar efforts.

2.3 Lessons learned

This Section is dedicated to outlining the lessons that the benchmarking experience done at the 2014 Camp has provided. What follows is strongly focused on the necessity of performing benchmarking in the context of the RoCKIn Competition: therefore it will be expressed in the form of a list of recommendations for the final design and setup

of the Competition. Of course, such recommendations have already been taken into consideration in the design of the forthcoming 2014 RoCKIn Competition.

2.3.1 General recommendations

- It is infeasible to make benchmarking rely on explicit actions that a participating team must take when their robot is subject to test. Any activity required by the test must be automatically executed by the robot, without any intervention from the team. In other words, benchmarking aspects of the Competition should be made as *transparent* as possible to the teams.

Such "transparent benchmarking" can be supported by providing each team with a single piece of software (e.g., a ROS node) and by requiring that such software is running on the robot during the Competition. The testbed should be capable of detecting if the software is actually running: in this way each benchmarking experiment will be started only if and when the software is active, i.e., if it is certain that all the activities required to the robot to participate to the experiment will actually be executed.

For instance, the piece of software described above may: (i) connect to an NTP server on the testbed's wireless network to synchronize the robot's clock with that of the server; (ii) notify the testbed (again through the wireless network) that the node is actually running and synchronization has been achieved; (iii) start the logging of the data required for benchmarking.

The software provided by RoCKIn should be robust, lightweight in terms of required computing resources and possibly also provided well in advance w.r.t. the Competition, in order to let teams check that the performance of their robots is not affected when the software is running on the robot.

Issues raised by the above recommendation:

1. Forcing a robot to run a piece of software means forcing the internal architecture of the robot to be compatible with such software: therefore, it is a decision that cannot be taken lightly.

2. It is likely that the requirement of being lightweight in terms of resources requested to the RoCKIn software will set constraints on the benchmarking process. For instance, logging large quantities of data (e.g., video) may be useful for benchmarking, but incompatible with such requirement. A careful balance will have to be sought.

3. One issue to be managed is how logged data should be stored and/or transmitted. Data streaming over the testbed's wireless network is probably best avoided to avoid reliability issues interfering with the logging (and therefore with benchmarking): therefore storage should be done onboard. To reduce operative problems and the risk of cheating, the best solution is that RoCKIn provides the teams with an external storage device (e.g., USB key): however, this introduces new requirements on the robots (such as the availability of a suitable free socket), which should be carefully considered.

2.3.2 Recommendations on testbeds

- A lightweight overhead truss (such as those used for theatre or concerts) should be available above the whole perimeter of each testbed area where motion capture has

to be performed. The truss should be at 4m from the ground, and should be affixed to the ground (*not* to the testbed or its boundaries) in order to make it impossible to move it even in the occurrence of serious collisions with people, robots or other heavy objects. The horizontal bars of the truss will support the motion capture cameras, and should therefore not be subject to vibrations; moreover, sufficient space should be present around such bars to allow free positioning, moving and aiming of cameras and associated mounting systems.

2.3.3 Recommendations on the motion capture system

- Objects contained in the testbeds, as well as testbed features, that can interfere with motion capture (e.g. bright metal surfaces, reflecting surfaces, light sources) should be avoided whenever possible. As it is impossible to guarantee that such interfering objects will not be present in the vicinity of the testbeds, positioning and pointing of motion capture cameras should ensure that out-of-testbed objects cannot be perceived[2]).

- During the Competition, only the robot subjected to a test should be allowed in the testbed for the duration of the test and its preparation. No people or extraneous material (which can create occlusions and/or otherwise interfere with the collection of motion capture data) should be present.

- Due to the tight time schedule of the Competition, it is necessary to have a means to immediately identify what was happening in each testbed at any arbitrary time (e.g., what robot was in there). Therefore, (conventional) video coverage of the whole volume of the testbeds should be acquired and saved for the whole duration of their opening hours. Time (synchronized with the NTP server of the relevant testbed) should appear superimposed on video frames.

- Any processing operation on benchmarking data to be done during the Competition should be implemented as a highly-automated script, written and tested well in advance of the Competition. In a competition setting, manually performing data processing is not feasible in terms of time and leads to errors.

- As already observed, the number of cameras required to observe the testbeds strongly depends on the size and shape of the testbeds. For this reason, testbed dimensions and shape should carefully matched to the available motion capture hardware.

- For testbeds similar to those used for the 2014 RoCKIn Camp, the 12 cameras available to the POLIMI partner proved to be insufficient (good coverage over the whole volume of the testbeds would have required at least 20 cameras). If similar testbeds will be used and full coverage of them will be required, it will be necessary to restrict the capture volume (as done at the Camp). Otherwise, a higher number of cameras should be acquired (possibly only temporarily).

[2]The OptiTrack camera management software allows masking of stationary video disturbances. This is done by defining masking shapes covering (small) areas of the field of view of single cameras. However, though possible, masking should be avoided. As cameras subjected to masking become partially "blind", masking introduces "blind spots" in the volume of space observed by the motion capture system, in the form of subvolumes that are perceived only by part of the cameras or, in the worst case, by none of them.

- As already explained, fitting motion capture markers to robots in a per-robot way is infeasible under the conditions of the Competition. Previously prepared *3D marker sets* (e.g., suitable non-deformable lattice-like objects provided with markers on vertexes) should be used instead.

Issues raised by the above recommendations:

1. 3D marker sets are a good solution to track mobile bases, but their use is infeasible for most end effectors. This strongly restricts the feasibility of collecting motion capture data for manipulation. Not wanting to return to customized marker positioning, two approaches to this problem are possible: (i) affix the marker set on the manipulated object instead (which should therefore be designed right from the start with this requirement in mind); (ii) restricting the collection of motion capture data to the tasks and cases where the marker sets can be used.

2. Robots participating to the Competition are expected to be very heterogeneous in their shape and structure. Devising 3D marker sets that can be successfully fitted to all participating robots is not a trivial task. In addition to mechanical problems and occlusions, issues of other type could arise: for instance, some teams at the Camp were unwilling to accept any mounting that could pose a risk of aestethic damage to their robot.

2.3.4 Recommendation on robots

The recommendations provided in the preceding part of this section lead to requirements that each robot system participating to the RoCKIn Competitions should comply with. Such requirements are listed here.

1. Each robot should possess the hardware and software necessary to interact with the testbed's wireless network. Additionally, the robot should be capable of keeping such network connection active for the whole duration of the benchmarking experiment (which prevents using the same hardware or software for other connections).

2. Each robot should be mechanically compatible with the *3D marker sets* prepared by RoCKIn, both in terms of providing suitable fixtures for their attachment to the robot and in terms of ensuring visibility of the markers by the motion capture cameras (i.e., avoiding that they get occluded by elements of the robot).

3. Each robot should provide a suitable free USB socket (possibly USB 3.0) to connect an external USB disk or stick provided by RoCKIn, for logging data.

4. Each robot should be endowed with sufficient processing power to manage task execution and logging at the same time.

5. Each robot should be capable to run any piece of software that RoCKIn provides to the teams, both in terms of processing resources and in terms of onboard software environment. As the RoCKIn project desires to be as inclusive as possible, this turns into a recommendation on RoCKIn software, which has to be multi-platform and require few resources.

Chapter 3

Data Acquisition at the 2014 RoCKIn Competition

Most of the contents of this chapter will be written after the first RoCKIn Competition, to be held in Toulouse (France) in November 2014. Deliverable D2.1.7, which has to be submitted well in advance of the Competition, only includes a list of the data types used in the forthcoming Competition. While not being part of the required content for D2.1.7, the information about data types has been considered useful to provide a wider view on the topics of D2.1.7.

3.1 List of data types for the RoCKIn competition

This section is dedicated to providing specifications for the data types used by RoCKIn for benchmarking. The main use of these data types is that of specifying the format of the data that the robot systems participating to the RoCKIn Competitions have to provide. With reference to the terminology defined in Section 1.1, these data are both *external benchmarking data* and *internal benchmarking data*.

The contents of this section must be intended as preliminary: in fact, we expect to revise them in the light of the experience of the 2014 RoCKIn Competition. This chapter is intended as an inspiration and a starting point for the implementation details (such as those concerning data formatting) that are actually part of the rulebooks for the 2014 RoCKIn@Home and RoCKIn@Work Competitions. After the 2014 Competitions, the contents of this chapter will be updated; the result will be available in Deliverable D2.1.8 (i.e., the final version of this document).

Whenever it is feasible, the data types will be kept as close as possible to those associated to the built-in message types of the ROS (Robot Operating System) middleware for robotics[1]. This design choice has multiple reasons. First of all, behind the readily available *corpus* of message types dedicated to robotics provided by ROS lies valuable work that can be exploited by RoCKIn as well. Secondly, compliance with such message types makes it very easy for developers using ROS to provide data according to the specifications of RoCKIn: as usage of ROS is quite common nowadays, this is a definite advantage. Finally, this design choice makes it easier for robots to comply with the logging requirements set forth by the rulebooks of the RoCKIn Competition. In fact, robots are required to log data for benchmarking in the form of *rosbags*, i.e., by using the ROS

[1] `http://www.ros.org/`

logging standard format.[2]

This section is built upon the content of the Rulebooks for the 2014 RoCKIn Competition. Version 1.0 for RoCKIn@Home and version 1.1 for RoCKIn@Work have been used.

The RoCKIn Rulebooks define 6 benchmarks each for RoCKIn@Home and RoCKIn@Work. Each group of 6 comprises 3 Task Benchmarks and 3 Functionality Benchmarks. In the following we will specify, for each of the benchmarks, what is the data for benchmarking and who is in charge of collecting such data. We will use the following notations:

- (B) = data collected by the testbed (either automatically or through the actions of the human referees);
- (R) = data collected by the robot under test;
- (M) = data collected by the motion capture system.

Additionally, for data marked (R) we will specify data types.

Specifications for data marked (B) and (M) are less relevant, as such data is neither generated nor used other than by RoCKIn personnel and/or equipment. For what concerns the (M) data, they use a version of the `http://docs.ros.org/api/geometry_msgs/html/msg/Pose.html` message type, augmented with information identifying the rigid body (defined using the OptiTrack software) that the pose refers to. Regarding (B) data, they will often be generated manually (at least at the first RoCKIn Competition), and thus their format is not interesting.

For what concerns string messages, string formats will be defined elsewhere.

3.1.1 Task Benchmarks - RoCKIn@Home

3.1.1.1 Task: catering for Granny Annie's comfort

During the execution of the benchmark, the following data will be collected:

- (R) on the robot, the audio signals of the conversations between Annie and the robot [format: `http://docs.ros.org/hydro/api/audio_common_msgs/html/msg/AudioData.html`]
- (R) the final command produced during the natural language analysis process [format: `http://docs.ros.org/api/std_msgs/html/msg/String.html`]
- (M) the pose of the robot while moving in the environment
- (R) the pose of the robot while moving in the environment, as perceived by the robot [format: `http://docs.ros.org/api/geometry_msgs/html/msg/Pose2D.html` or `http://docs.ros.org/api/geometry_msgs/html/msg/Pose.html`]
- (R) the sensorial data of the robot when recognizing the object to be operated [format: `http://docs.ros.org/api/sensor_msgs/html/msg/Image.html`, `http://docs.ros.org/api/sensor_msgs/html/msg/PointCloud2.html`]
- (B) the results of the robot's attempts to execute Annie's commands

[2] Please note that this does *not* force teams to use ROS. While for ROS-based robots producing *rosbags* is a straightforward process, suitable stand-alone software tools to produce *rosbags* are being developed by RoCKIn and will be distributed to teams not using ROS.

3.1.1.2 Task: welcoming visitors

During the execution of the benchmark, the following data will be collected:

- (R) the event/command causing the activation of the robot [format: `http://docs.ros.org/api/std_msgs/html/msg/String.html`]
- (B) the video signal from the door camera
- (M) the position and movements of the robot during the execution of the task
- (R) the pose of the robot while moving in the environment, as perceived by the robot [format: `http://docs.ros.org/api/geometry_msgs/html/msg/Pose2D.html` or `http://docs.ros.org/api/geometry_msgs/html/msg/Pose.html`]
- (R) the results of any attempts by the robot to detect and classify a visitor [format: `http://docs.ros.org/api/std_msgs/html/msg/String.html`]
- (R) the audio signals of the conversations with the visitors [format: `http://docs.ros.org/hydro/api/audio_common_msgs/html/msg/AudioData.html`]
- (R) any notifications from the robot (e.g., alarm if a visitor shows anomalous behavior) [format: `http://docs.ros.org/api/std_msgs/html/msg/String.html`]
- (B) the results of any actions taken by the robot, including opening or closing the front door, guiding visitors into and around the apartment, manipulating objects, etc.

3.1.1.3 Task: getting to know my home

During the execution of the benchmark, the following data will be collected:

- (R) the output files produced by the robot [format: please see the RoCKIn@Home Rulebook]
- (M) the trajectories of the robot
- (R) the pose of the robot while moving in the environment, as perceived by the robot [format: `http://docs.ros.org/api/geometry_msgs/html/msg/Pose2D.html` or `http://docs.ros.org/api/geometry_msgs/html/msg/Pose.html`]
- (B) the result (success/failure) of the command issued to the robot

3.1.2 Task Benchmarks - RoCKIn@Work

3.1.2.1 Task: prepare assembly aid tray for force fitting

During the execution of the benchmark, the following data will be collected:

- (R) ID of the assembly aid tray, provided by the robot (by analyzing the QR code) [format: `http://docs.ros.org/api/std_msgs/html/msg/String.html`]
- (R) ID of the container, provided by the robot (by analyzing the QR code) [format: `http://docs.ros.org/api/std_msgs/html/msg/String.html`]

- (R) images used by the robot to analyze the QR) codes [format: `http://docs.ros.org/api/sensor_msgs/html/msg/Image.html`,]
- (B) number of the bearing boxes successfully fitted to the assembly aid tray
- (R) trajectory planned by the robot and their execution (as perceived by the robot) including additional ones due to replanning [format: `http://docs.ros.org/api/nav_msgs/html/msg/Path.html`]

3.1.2.2 Task: plate drilling

During the execution of the benchmark, the following data will be collected:

- (B) number and condition (unusable, faulty, perfect) of all plates provided to the robot via the conveyor belt
- (R) trajectory planned by the robot and their execution (as perceived by the robot) including additional ones due to replanning [format: `http://docs.ros.org/api/nav_msgs/html/msg/Path.html`]
- (R) condition of each plate, as evaluated by the robot, after receiving the plate [format: `http://docs.ros.org/api/std_msgs/html/msg/String.html`]
- (R) drilling commands issued by the robot [format: `http://docs.ros.org/api/std_msgs/html/msg/String.html`]
- (R) condition of each plate, as evaluated by the robot, after drilling [format: `http://docs.ros.org/api/std_msgs/html/msg/String.html`]
- (B) effect of the robot's sorting actions
- (R) sensor data used by the robot to perform plate analysis (images) [format: `http://docs.ros.org/api/sensor_msgs/html/msg/Image.html`]

3.1.2.3 Task: fill a box with parts for manual assembly

During the execution of the benchmark, the following data will be collected:

- (R) initial plan, as defined by the robot after receiving the information about the product to be assembled [format: `http://docs.ros.org/api/std_msgs/html/msg/String.html`]
- (R) trajectory planned by the robot, including additional ones due to replanning [format: `http://docs.ros.org/api/nav_msgs/html/msg/Path.html`]
- (R) notifications from the robot concerning obstructions preventing it from following the plan [format: `http://docs.ros.org/api/std_msgs/html/msg/String.html`]
- (R) any new plan defined by the robot during part collection (to manage obstructions) [format: `http://docs.ros.org/api/std_msgs/html/msg/String.html`]
- (M) ground truth pose of the base of the robot while it is collecting parts

- (R) pose of the base of the robot as estimated by the robot, while it is collecting parts [format: `http://docs.ros.org/api/geometry_msgs/html/msg/Pose2D.html` or `http://docs.ros.org/api/geometry_msgs/html/msg/Pose.html`]
- (B) number and identity of the parts provided by the robot to the human worker at the end of the collection

3.1.3 Functionality Benchmarks - RoCKIn@Home

3.1.3.1 Object perception functionality

During the execution of the benchmark, the following data will be collected:

- (B) number of objects presented to the robot
- (R) detection, recognition and localization data associated to the objects, provided by the robot [format: `http://docs.ros.org/api/std_msgs/html/msg/String.html`, `http://docs.ros.org/api/geometry_msgs/html/msg/Pose.html`]
- (B+M) ground truth for object pose, object class, and object instance
- (R) sensor data used by the robot to perform classification (e.g., images, point clouds, etc.) [format: `http://docs.ros.org/api/sensor_msgs/html/msg/Image.html`, `http://docs.ros.org/api/sensor_msgs/html/msg/PointCloud2.html`]

3.1.3.2 Object manipulation functionality

During the execution of the benchmark the following benchmarking data will be collected:

- (R) notifications issued by the robot [format: `http://docs.ros.org/api/std_msgs/html/msg/String.html`]
- (B) initial and final setting of all controls on the test panel
- (R) internal robot data referring to end effector position and target object position [format: `http://docs.ros.org/api/geometry_msgs/html/msg/Pose.html`]
- (M) external ground truth about panel position and end effector position

3.1.3.3 Speech understanding functionality

During the execution of the benchmark, the following data will be collected (please see the Rulebook for additional information):

- (R) sensor data (in the form of audio files) used by the robot to perform speech recognition [format: `http://docs.ros.org/hydro/api/audio_common_msgs/html/msg/AudioData.html`]
- (R) the set of all possible transcription for each user utterance [format: `http://docs.ros.org/api/std_msgs/html/msg/String.html`]
- (R) the final command produced during the natural language analysis process [format: `http://docs.ros.org/api/std_msgs/html/msg/String.html`]

- (R) intermediate information produced or used by the natural language understanding system during the analysis as, for example, syntactic information [format: `http://docs.ros.org/api/std_msgs/html/msg/String.html`]

3.1.4 Functionality Benchmarks - RoCKIn@Work

3.1.4.1 Object perception functionality

During the execution of the benchmark, the following data will be collected:

- (B) number of objects presented to the robot
- (R) detection, recognition and localization data associated to the objects, provided by the robot [format: `http://docs.ros.org/api/std_msgs/html/msg/String.html`, `http://docs.ros.org/api/geometry_msgs/html/msg/Pose.html`]
- (B+M) ground truth for object pose, object class, and object instance
- (R) sensor data used by the robot to perform classification (e.g., images, point clouds, etc.) [format: `http://docs.ros.org/api/sensor_msgs/html/msg/Image.html`, `http://docs.ros.org/api/sensor_msgs/html/msg/PointCloud2.html`]

3.1.4.2 Visual servoing functionality

During the execution of the benchmark, the following data will be collected:

- (B) number of objects presented to the robot
- (R) identification of the objects, provided by the robot [format: `http://docs.ros.org/api/std_msgs/html/msg/String.html`]
- (R) grasp notifications issued by the robot [format: `http://docs.ros.org/api/std_msgs/html/msg/String.html`]
- (B) ground truth concerning the fact that the object does not touch the table
- (R) sensor data used by the robot to perform identification (images) [format: `http://docs.ros.org/api/sensor_msgs/html/msg/Image.html`, `http://docs.ros.org/api/sensor_msgs/html/msg/PointCloud2.html`]

3.1.4.3 Planning and scheduling functionality

During the execution of the benchmark, the following data will be collected:

- (R) original plan generated by the scheduler and each new plan generated [format: `http://docs.ros.org/api/std_msgs/html/msg/String.html`]
- (B) the current situation in terms of collected items and blocked robots
- (B) time at which each object of each list has been picked up
- (B) time at which each object of each list has been delivered to the destination
- (B) time at which the last item of each list has been delivered to the destination