Integration and Assessment of Multiple Mobile Manipulators in a Real-World Industrial Production Facility

Simon Bøgh, Casper Schou
Department of Mechanical & Manufacturing Engineering, Aalborg University, Denmark

Thomas Rühr, Yevgen Kogan
KUKA Laboratories GmbH, Germany

Andreas Dömel, Manuel Brucker
Deutsches Zentrum für Luft- und Raumfahrt, Germany

Christof Eberst, Riccardo Tornese
Convergent Information Technologies GmbH, Austria

Christoph Sprunk, Gian Diego Tipaldi
Department of Computer Science, University of Freiburg, Germany

Trine Hennessy
Grundfos A/S, Denmark

Abstract

This paper presents a large-scale research experiment carried out within the TAPAS[1] project, where multiple mobile manipulators were integrated and assessed in an industrial context. We consider an industrial scenario in which mobile manipulators naturally extend automation of logistic tasks towards assistive ones. In the experiment, we included tasks such as preparatory and post-processing work, e.g. pre-assembly or machine tending with inherent quality control. In the experiment, we deployed the two heterogeneous mobile manipulators Little Helper and omniRob in a production scenario at Grundfos A/S, a manufacturer of water circulation pumps, in Denmark. The experiment showed that mobile manipulation is at a level of technology readiness that will allow industrial application in the near future. Despite challenges indicated later in the paper, the research efforts presented do show that research is on the right track on transferring mobile manipulation from research to industry.

1 Introduction

The increasing need for flexibility in modern industrial production puts higher demands on the design and flexibility of robotics solutions of today and the future. To accommodate this demand a new generation of transformable solutions to automation and logistics for small and large series production is sought for in the TAPAS project. The demand of automation flexibility is investigated through the use of mobile manipulators in real-world industrial settings carrying out industrial tasks. The research and experimental approach throughout the TAPAS project is to have continuous testing and evaluation of the research results in real-world industrial settings, i.e., live running production. The TAPAS mobile manipulators are able to perform tasks that naturally extend traditional logistics towards automating assistive tasks, such as pre-assembly or machine tending. In this paper we present the third and final round of evaluation experiments where an industrial scenario features two mobile manipulators that were set up in only three days followed by an eight-hour consecutive run in a live running production facility.

2 Robot Systems Description

In the experiment two mobile manipulators have been used for assembly and logistics: the Little Helper 3 by Aalborg University and the omniRob by KUKA. In the following an introduction is given to each mobile manipulator's hardware and software setup.

[1] "Robotics-enabled Logistics and Assistive Services for the Transformable Factory of the Future" (TAPAS) is funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no 260026.

2.1 Little Helper

Little Helper 3 (see **Figure 1**) is primarily composed of commercial off-the-self (COTS) components. However, a custom aluminum frame serves as interface between the components. **Table 1** lists the main components of Little Helper 3.

Figure 1: The Little Helper 3 from Aalborg University.

Table 1: Little Helper hardware system

Robot arm	KUKA LWR4+ (7 DoF)
Mobile platform	Neobotix MP-655 (non-holonomic)
Vision system	DMK 31BF03-Z2 FireWire CCD monochrome motorized zoom PrimeSense Carmine 1.09 (short range)
Tool/gripper	Schunk WSG 50 electrical 2-Finger parallel gripper
Computational	Intel i7 2.4GHz quad-core, 16 GB RAM, SSD drive

The software system of Little Helper is based on a distributed architecture using Robot Operating System (ROS) as the underlying infrastructure. This enables the two onboard computers to share the same ROS network. Each device is connected to one of the two computers, on which a device driver handles communication towards the device. A central control node utilizes all the distributed device nodes and thus creates the higher level of control. The central control node also incorporates the user interface including task-level-programming.

Little Helper applies a concept of robot skills [1] [2] for programming and executing tasks. Where traditional robot programming operates on functionalities of each specific device, the concept of skills is used as an abstraction upon the functionality of the mobile manipulator as a whole. This could for instance be “pick <object>", “rotate <object>" or “push <object>". Note, that the skills are formulated to act on objects.

In relation to task programming, the abstraction of skills is closely related to the manipulations done by the operator when carrying out the task manually. As a result, programming the task using skills becomes much more intuitive compared to traditional robot programming. Additionally, by enabling a shop floor operator to program a given task, it will also help preserve his or hers process knowledge [4]. In this experiment, the skill concept is used to both program and execute tasks on Little Helper.

Little Helper 3 is based on a modular architecture and can be configured for different application types. In this experiment, it is configured for assembly operations and machine tending.

2.2 omniRob

The KUKA omniRob (see **Figure 2**) is a mobile robotic platform with omnidirectional wheels and a KUKA LWR4+ mounted on top. Both the robot and the platform are controlled by the same controller and operating system and are treated as one unit. The basic system is equipped with torque sensors in the robot axes for compliant motion execution and laser scanners on the platform for navigation and safety purposes.

Figure 2: The omniRob carrying out part logistics, transporting finished assembled goods from the assembly station to the warehouse.

At DLR, additional sensors (see **Table 2**) were mounted on the mobile platform, on a sensor pole with a Schunk PTU, and on the robot’s end effector. These sensors enable the system to perceive the environment in gray-scale and in depth values. To enable the system to work without external infrastructure, processor boards and a FPGA for semi-global matching stereo processing [5] were integrated in the robot.

The software architecture is based on a modular approach encapsulating functional units into different processes. On top of the KUKA Robotics API which is based on Java, a communication framework based on ROS has been implemented. The top level for programming mobile manipulation applications is made up by a graphical programming framework which allows intuitive programming of the available complex features.

Table 2: omniRob hardware system

Robot arm	KUKA LWR4+ (7 DoF)
Mobile platform	KUKA omni-directional platform (holonomic)
Vision system	Schunk PTU SwissRanger SR4000 Stereo camera system
Tool/gripper	Schunk PG 70 servo-electric 2-finger parallel gripper
Computational	4x I7 processor boards FPGA-board

As the robot interacts with partially unknown environments such as shelves or the robot's working surface, these need to be explored to allow for collision free path planning. Therefore, the PTU is moved and a probabilistic voxel space is updated with each depth measurement applying Bayes' Rule as in [7]. Due to the expected scarcity of texture information in industrial scenarios, the object recognition module of the framework is based on depth images. Specifically, a geometric matching approach [6] featuring a fast GPU based verification step is used.

To cope with the high dimensional configuration space of the robot, a sampling based path planner is used for generating autonomous motions. Using existing CAD models and the sensed environment models, the path planner guarantees collision free motions in partially unknown environments. During manipulation of objects, constraints regarding the orientation have to be met. Our approach is based on a BiRRT-Connect algorithm [8] using sample rejection for configurations violating the pose constraints.

In order to fulfill its tasks, the omniRob has to navigate the production environment autonomously. Therefore, the omniRob is equipped with a navigation system which uses the current laser reading and odometry to localise itself and navigate based on a gridmap recorded during the setup phase of the system.

For planning the trajectories of the omniRob the motion generation approach presented by Sprunk et al. [10] is used. This method generates smooth base trajectories for holonomic omnidirectional robots. It takes as input a set of waypoints and a map of the environment to generate curvature-continuous trajectories. Furthermore, it explicitly accounts for constraints on the velocity and acceleration of the vehicle which allows a feedback controller to track the generated trajectories with high precision.

The approach starts from an initial trajectory inferred from the Voronoi diagram of the environment map. The approach then employs an anytime-optimization of the shape of the trajectory with respect to a user-defined cost function. A schematic example of the optimization process is depicted in **Figure 3**. Through regular replanning and smooth, continuous stitching of updated trajectories the method can react also to unmapped obstacles.

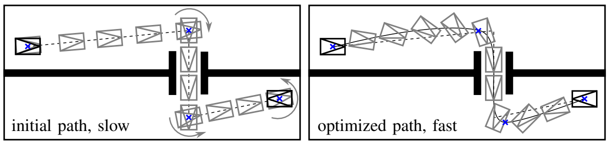

Figure 3: Trajectories before and after optimization using the approach by Sprunk et al. [10]. The generated trajectories account for constraints on velocities and accelerations.

The programming of the mobile manipulation tasks discussed in Section 4.2 is carried out using the graphical programming framework (see **Figure 4**). This programming scheme has been developed and implemented on the omniRob in order to allow non-programming experts to set up mobile manipulation applications.

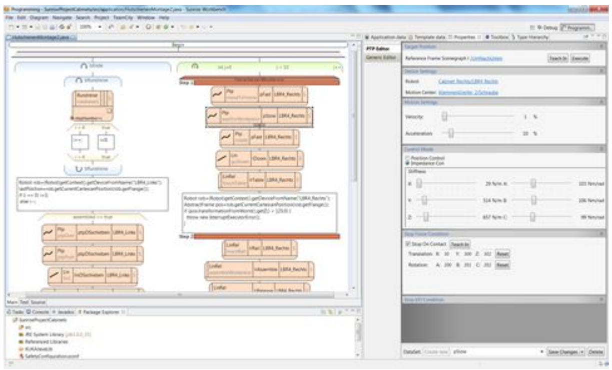

Figure 4: Graphical Programming Framework for setting up mobile manipulation applications.

The programming framework itself is based on a library of function blocks representing the functionalities of the robot. The function blocks can then be arranged in a graphical editor, which also serves as a runtime interface. The features of the graphical programming framework also include facilities for editing parameters and step-wise debugging.

The omniRob in the described configuration has powerful facilities for autonomously operating in only partly known environments. Furthermore it is basically able to safely interact with human operators in the area. In the experiment it is used for logistic tasks.

3 Mission Planning and Control

A central mission planner and controller (MPC) carries out task planning and scheduling for the scenario. It ensures that parts are retrieved, assembled and transported as efficiently as possible between the workstations and the warehouse. The planner accounts for cycle time, travel time, load capacity of the robots, and the overall production goals. It also handles unexpected events in the overall production scenario and notifies a human operator about errors, which the robots cannot autonomously recover from. The MPC runs on a separate computer connected to both mobile manipulators via wireless ethernet. The MPC consists of two main parts, a mission planner and a mission controller (see **Figure 5**) [3].

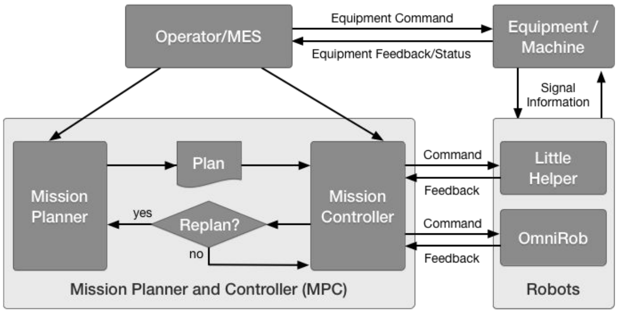

Figure 5: Overview of the Mission Planner and Controller (MPC).

The mission planner generates a plan based on either operator input or information from the Manufacturing Execution System (MES). The planner takes the practical constraints of the scenario into account; for instance, cycle time of the tasks, travel times between tasks and capacity limits. On unexpected events, the planner must be able to replan in a sufficiently short period of time. Thus, for the experiment the maximum allowable computation time is limited to 30 seconds.

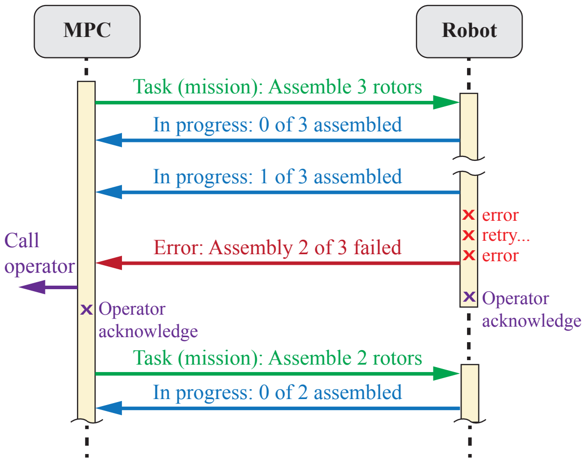

Figure 6: UML sequence diagram of the communication between MPC and mobile manipulator. The diagram illustrates an error in the task.

Once a plan is generated by the mission planner, it is sent to the mission controller. The mission controller handles the scheduling of tasks to the two mobile manipulators; thus, the mission controller handles IO-communication of the MPC. **Figure 6** shows an UML sequence diagram of the communication between MPC and one of the mobile manipulators.

The communication between the MPC and the mobile manipulators is done via TCP/IP socket connections. The transmitted data is formatted in a custom XML format.

Throughout the experiment, the MPC is operated from a graphical user interface. Production data (number of products to produce) are given manually by an operator; hence, they are not provided by a MES system.

4 Scenario Description

The starting point of the scenario is a real-world production facility spanning 25x15 meters at the industrial project partner Grundfos A/S in Bjerringbro, Denmark. This production environment entails multiple tasks including part retrieval from multiple production cells, assembly station for part assembly, and warehouse for empty containers and finished goods (see **Figure 7**).

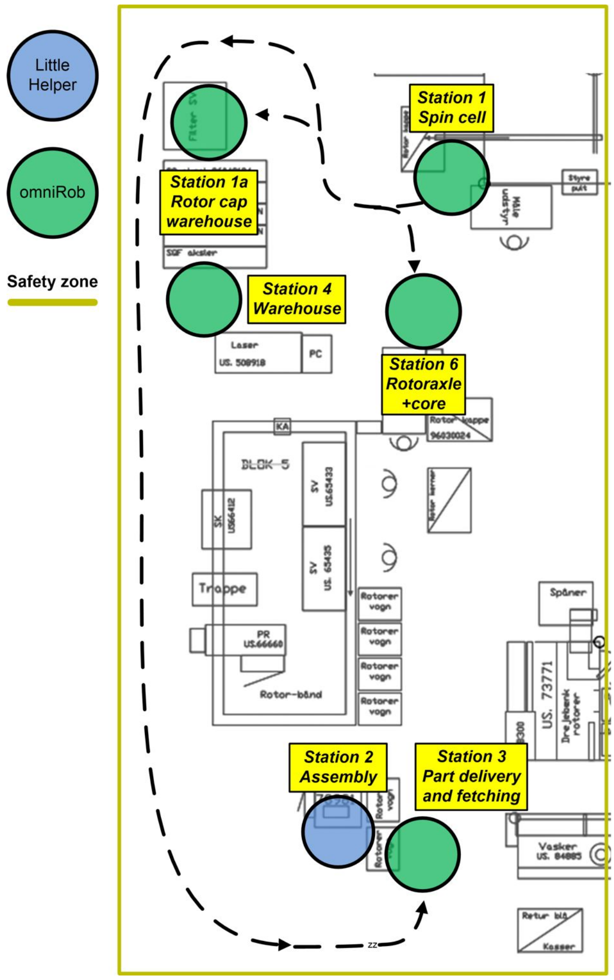

Figure 7: Layout of the production area where the experiment was conducted. The layout also illustrates the six workstations including travel-routes for the logistics tasks (station 5 "charging station" is excluded from the figure).

In the given scenario the omniRob utilizes its mobility for logistics while the Little Helper is stationary at a workstation for assembly. The purpose of this is to show how a real-world usage scenario of mobile manipulators could take place. The logistic task could be part of an everyday need at the production line, while assembly on the other hand could be an ad-hoc task taking place in cases of sickness in the manual workforce or in urgent need of increased capacity in short periods of time. To handle and execute all the different tasks in the scenario, the MPC manages the overall production scenario i.e., scheduling the mobile manipulators for specific tasks.

In **Figure 8** Little Helper and omniRob are shown at two different workstations. The Little Helper carries out the assembly of a rotor at workstation 2. The omniRob carries out logistics with both single parts and Small Load Carriers (SLCs) at workstation 3.

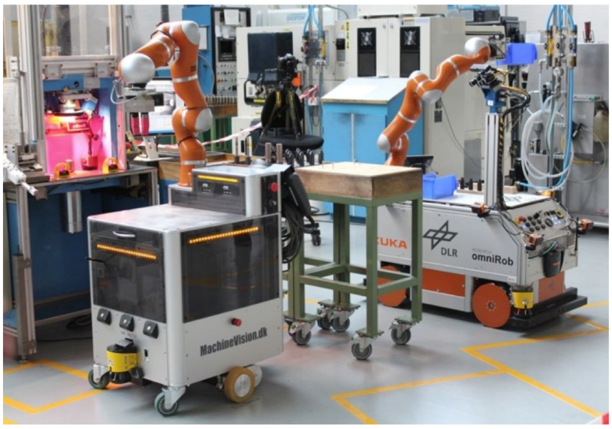

Figure 8: The Little Helper (left) working at the assembly station and the omniRob (right) performing part logistics at workstation 3.

4.1 Assembly Task

The scenario includes an assembly task, which is carried out at workstation 2 (see Figure 7). In this task the "rotor" for a Grundfos pump is assembled from the components shown in **Figure 9**.

The assembly is normally carried out as a manual task, where a worker will attend the task for 1-2 hours a day. The task is conducted by picking the components from Figure 9 and placing them in a fixture inside a hydraulic press (see **Figure 10**). After the components have been placed, the press is activated using a two-hand trigger. The finished product is then placed in a trolley.

In the manual task the components are located next to the hydraulic press. The rotor shafts are positioned ordered in a trolley, whereas the rest of the components are placed randomly in bins. In order to conduct the task using a mobile manipulator, the task has been adapted accordingly. Two feeders have been introduced, one for the rings and one for the magnets (see Figure 10). In order to limit the complexity of the environment adaptation, the feeders are designed to be actuated by the robot itself; thus, the feeders do not require electronic or communication components. For safety reasons, a moveable gate is fitted to the hydraulic press to prevent human injuries. The described adaptations of the machinery in the assembly task have been carried out in such a way, that the workstation is still easily and safely accessible by human workers.

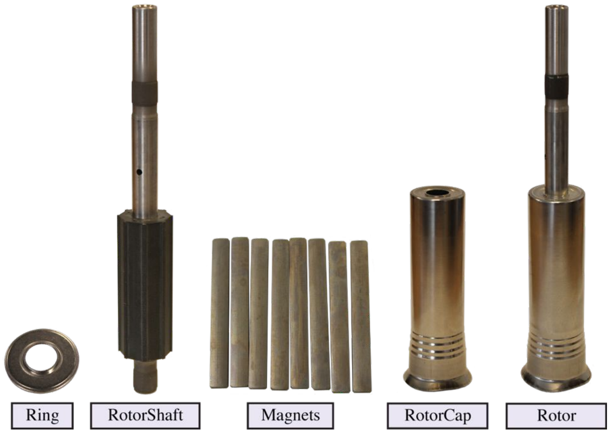

Figure 9: Parts used for the assembly process of the "Rotor" component. "Magnet" and "Ring" are fed via dispensers, "RotorCap" and "RotorShaft" sub-assembly are delivered by the omniRob.

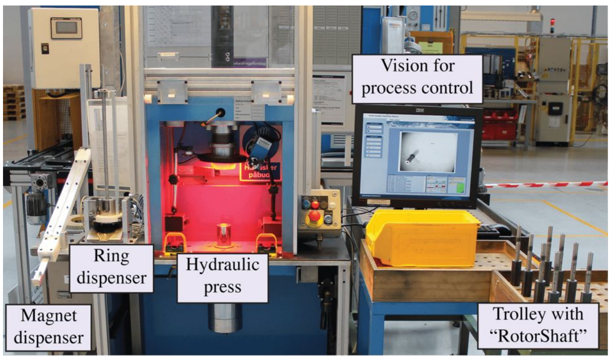

Figure 10: Custom part feeders integrated as part of the assembly task at workstation 2. One feeder for magnets and one for rings. In the middle is a hydraulic press with an assembly fixture, where the assembly process takes place.

Inside the hydraulic press a 2D monochrome camera is mounted. This is used to perform quality checks to ensure that the assembly process is carried out correctly and no parts are misplaced. The vision camera is connected to a vision processing system developed by Grundfos. The quality check is wirelessly triggered from the Little Helper using a custom protocol over TCP/IP. The Grundfos vision system processes the image and sends the outcome to Little Helper. A similar camera is mounted on the robot tool, which allows the robot to locate objects for the assembly

process and gain knowledge about the quality of the assembly process. Images from this camera are wirelessly transferred to the Grundfos vision system for image analysis. Throughout the assembly process, continuous quality control is carried out to ensure final product quality.
Due to the close relation between task operations and skills, the skill sequence constituting the robot task is very similar to the manual assembly task. The skill sequence is listed below:

1. Press <button> on ring feeder
2. QC to verify ring is dispensed
3. Pick <ring>
4. Place <ring> in fixture
5. QC to verify ring
6. Pick <rotor shaft> from ordered pallet
7. PegInHole <rotor shaft> into fixture
8. Align <rotor shaft>
9. Actuate <magnet dispenser handle> 8 times
10. Repeat x8
    (a) Pick <magnet> from feeder
    (b) PegInHole <magnet> into fixture
    (c) Rotate <rotor shaft> 45 deg
    (d) QC to verify magnet
11. Pick <rotor cap> from ordered pallet
12. Place <rotor cap> on top of rotor shaft
13. Close <safety gate> to activate hydraulic press
14. QC to verify press completed
15. Open <safety gate>
16. Pick <rotor> from fixture
17. Place <rotor> in SLC

4.2 Logistic Task

In the given scenario, the omniRob carries out two logistic tasks: part feeding of components to the assembly workstation and transportation of SLCs with assembled rotors to the warehouse.
In the part feeding task, the omniRob will supply both rotor caps and rotor shafts to workstation 2. The rotor caps are picked from a conveyor at a production cell (see **Figure 11**). The pickup operation requires operating a switch for deactivating the conveyor belt, localizing the correct number of parts to pick up and the pickup-operation of the parts. The retrieved parts are stored for transport on a fixture in the loading bay area of the omniRob.

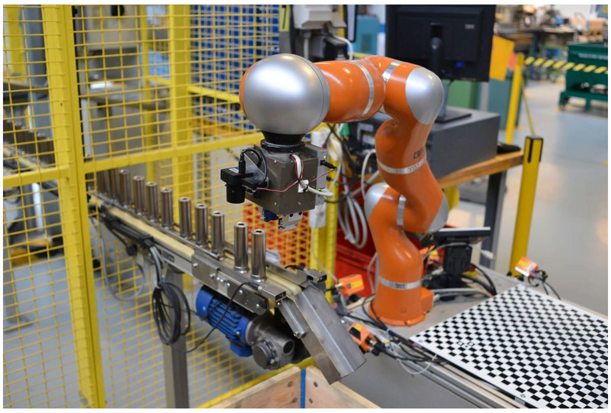

Figure 11: omniRob picking up rotor caps at the spin cell machine.

If no or not enough rotor caps are available, the remaining rotor caps will be picked from a warehouse. The rotor shafts are picked from a warehouse located at workstation 6.
The assembled rotors are placed in an SLC at workstation 2 by Little Helper. Once the SLC is full, the omniRob will be scheduled to transport the full SLC to the warehouse and replace it with an empty SLC.
The position of the various operational areas for the omniRob are registered in the navigation system during the setup phase. Additional visual markers have been applied to some workstation to provide more positional accuracy. Between the workstations the robot travels autonomously. Once the robot has reached the workstation, a docking procedure that accurately positions the robot at the workstation is performed. The method determines the offset to the target position by matching the current readings of the robot's laser scanners against reference readings previously recorded at the workstation location. More details on this fine positioning and its combination with the navigation approach are presented by Röwekämper et al. [9].
The subsequent object manipulation steps are then carried out using the perception, path planning and compliance control features of the robot described in Section 2.2. In that way, objects can be safely handled even if the environment changes (e.g., the position of mobile fixtures on the shop floor change) or previously unknown obstacles appear in the area of operation.

4.3 Experiment Setup

The experiment was conducted over a period of five days according to the following schedule at the production facility:

- Monday: Initial setup and testing.
- Tuesday: Functionality testing of the mobile manipulators.
- Wednesday: Integration and first testing with the mission planner. First test-run of the scenario.

- Thursday: Full day running of the scenario. Presentation for visitors.
- Friday: Presentation for company management. Evaluation and pack up.

The target of the experiment was to conduct an eight hour test of the scenario. The eight hours match one shift in the production. The initial state of the scenario when starting the 8 hour experiment was: Little Helper is assembling rotors at workstation 2. It has an empty SLC for finished parts and two rotor shafts and two rotor caps available. The omniRob is idle and thus charging and waiting for incoming logistic tasks from the mission planner.
Once Little Helper started its last assembly relative to the parts available at the assembly station, the mission planner schedules the omniRob to retrieve parts from the production cell and warehouse. After Little Helper has finished the last part, the omniRob is scheduled to deliver the parts and replace the full SLC. To prevent potential collisions between the robots at the assembly station, the Little Helper is idle until the omniRob has finished its task. After delivering parts for yet two assemblies, the Little Helper starts assembly and omniRob travels to charging station. The batch size is chosen to just two to increase omniRob activity.

5 Results

During the eight hour experiment all activities were logged. This includes production numbers, task activities and errors. As the experiment was conducted over a period of eight hours with numerous activities going on, the results presented here will reflect the overall achieved production results. Additionally, a summary of the most common error types logged will be presented.
The eight hour experiment was setup in the three preceding days. This included equipment unpacking and setup, communication setup and testing, adapting workstations, navigation setup, task programming, and testing.

5.1 Production Measures

During the eight hour experiment a total of 10 rotors were produced and transported to the warehouse. A total of 34 rotors were produced throughout the five days including setup and testing.

Table 3: Average cycle times for the tasks

Assembly task	11 minutes
Logistic tasks	22 minutes

Table 3 presents the average cycle times for the tasks. The average cycle times are based on measurements from the eight our shift including only successfully executed tasks. In the scenario the omniRob conducted both the part feeding task and the SLC logistic task consecutively. As a result, the travel times of the tasks are hard to separate and thus a combined average time is listed. Please note, that the cycle time for the logistic task is very dependent on travel time; thus, the cycle time would not increase proportionally to the number of parts carried.

5.2 Errors

For each error logged, the cause and associated downtime were logged. The errors presented here all needed operator assistance to solve. In the assembly process a total of 22 errors were recorded resulting in 44 minutes of downtime. In the logistic task 35 errors occurred resulting in 132 minutes of downtime. The operating time thus becomes 91.8% and 72.5%, respectively.
The three most frequent errors in the assembly task were:

- Manipulation task
- Internal communication
- Communication with mission planner

The most frequent errors in the logistic task were:

- Manipulation task
- Navigation / laser scanner
- Communication with mission planner

The errors concerned with the navigation were mainly due to the laser scanners build in safety-trigger when docking at workstations. Both robots experienced communication errors with the mission planner. When one of the robots was restarted due to an error, the mission planner had to be manually reinitialized. Most of the mission planner errors logged occurred in relation to these reinitializations. Thus, the errors could also be related to human factors.

5.3 Integration Challenges

Safety: Human-Robot and Machine-Robot Safety

During the experiment we found that safety is a major issue when implementing mobile manipulators in industrial environments. In human-robot safety it is essential that no humans come in harms way during production. Since we are in general working with expensive equipment on the actual production line, machine-robot safety is also a concern. Damages to production equipment could result in lost production time and ultimately loss of customer orders.
During the experiment we took the following safety measures:

- Fences and signs to mark the robots operating area
- Warning light indicators on robots
- Remote emergency stops for both robots
- Keep humans out of the work environment
- Proper instruction of personnel

- Appointing a safety responsible to make sure no unauthorized people enter the area
- Setting safety parameters for the on-board laser-range scanners on both robots
- Lowering speed and acceleration on the robot arms and platform navigation
- Fitting a safety gate to the assembly station

Integration with Existing Production System

One integration challenge often encountered throughout the experiments in the TAPAS project has been integration and interaction with existing production equipment. Some equipment might be relatively old, have limited or no interfaces for communication (i.e., I/O), or is so deeply integrated into the production line that it can be critical if any changes are made to the setup.
In the experiment, communication with the assembly station and the spin cell was not possible. Instead mechanical interactions were developed. The hydraulic press at the assembly station is actuated using a two-hand switch. In the experiment the safety gate was designed to activate these switches when being closed. At the spin cell the conveyor needed to be stopped. This was done by implementing a robot-operated mechanical slider to block a sensor.

Error Handling on Mission Level

During the experiment we encountered a number of errors in the communication between robots and mission planner. If one of the robots encountered an error and had to be restarted, the communication with the mission planner had to be reinitialized manually. The errors in the communication often occurred after reinitializing the mission planner. Given that robot errors cannot be completely avoided, it is necessary to overcome this on a mission controlling level. Thus, in future experiments the communication should be developed to encompass some level of error handling.

6 Conclusion and Future Work

In this paper we have presented an experiment where two mobile manipulators have been implemented in a real-world production to, in collaboration, carry out production of a component for a pump. The experiment demonstrates that mobile manipulators are capable of operating for prolonged periods of time in a real-world manufacturing environment. However, despite the significant accomplishments presented in this paper, few challenges still remain in order to bring mobile manipulators to a maturity level where the technology can be successfully commercialized: safety, robustness, performance, ease of use, and equipment costs. These challenges are not straightforward to overcome and are part of present and future work.

Acknowledgement This work has been partly supported by the European Commission under contract number FP7-ICT-260026-TAPAS.

References

[1] Pedersen, M. R.; Nalpantidis, L.; Bobick, A.; Krüger, V.: *On the Integration of Hardware-Abstracted Robot Skills for use in Industrial Scenarios*, 2nd Int. Workshop on Cognitive Robotics Systems, 2013, pp. 6

[2] Bøgh, S.; Nielsen, O. S.; Pedersen, M. R.; Krüger, V.; Madsen, O.: *Does your Robot have Skills?*, Proc. of 43rd Int. Symposium on Robotics, 2012, pp. 6

[3] Dang, V.Q.; Nielsen, I.E.; Bøgh, S.; Bocewicz, G.: *Modelling and Scheduling Autonomous Mobile Robot for a Real-World Industrial Application*, Proc. of IFAC Conf. on Manufacturing Modelling, Management and Control, 2013, pp. 2098-2103

[4] Schou, C.; Damgaard, J.S.; Bøgh, S.; Madsen, O.: *Human-Robot Interface for Instructing Industrial Tasks using Kinesthetic Teaching*, Proc. of 44th Int. Symposium on Robotics (ISR), 2013, pp. 1-6

[5] Hirschmüller, H.: *Stereo Processing by Semiglobal Matching and Mutual Information*, IEEE Trans. Pattern Anal. Mach. Intelligence, 2008, vol 30, no. 2, pp. 328-341

[6] Kriegel, S.; Brucker, M.; Marton, Z.C.; Bodenmüller, T.; Suppa, M.: *Combining Object Modeling and Recognition for Active Scene Exploration*, Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2013, pp. 2384-2391

[7] Kriegel, S.; Rink, C.; Bodenmüller, T.; Narr, A.; Suppa, M.; Hirzinger, G.: *Next-Best-Scan Planning for Autonomous 3D Modeling*, Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2012, pp. 2850-2856

[8] Kuffner, J.J; LaValle, M.: *RRT-Connect: An Efficient Approach to Single-Query Path Planning*, Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA), 2000, pp. 781-787

[9] J. Röwekämper, C. Sprunk, G.D. Tipaldi, C. Stachniss, P. Pfaff, and W. Burgard. On the position accuracy of mobile robot localization based on particle filters combined with scan matching. Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2012

[10] C. Sprunk, B. Lau, P. Pfaff, and W. Burgard. Online generation of kinodynamic trajectories for non-circular omnidirectional robots. Proc. of IEEE Int. Conf. on Robotics & Automation (ICRA), 2011