Dissemination Level: PU

610917 - STAMINA

Sustainable and reliable robotics for part handling in manufacturing

Project no.:	**610917**
Project full title:	**Sustainable and reliable robotics for part handling in manufacturing**
Project Acronym:	**STAMINA**
Deliverable no.:	**D2.1**
Title of the deliverable:	**Report on the Localization and Mapping System**

Contractual Date of Delivery to the CEC:	**30.04.2016**
Actual Date of Delivery to the CEC:	**30.04.2016**
Organisation name of lead contractor for this deliverable:	**Albert Ludwigs Universität Freiburg**
Author(s):	**Jörg Röwekämper, Alexander Schiotka**
Work package contributing to the deliverable:	**WP2**
Nature:	**R**
Version:	**1.0**
Total number of pages:	**14**
Start date of project:	**01.10.2013**
Duration:	**42 months – 31.03.2017**

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 610917

Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Abstract:

This deliverable is an outcome of Task 2.1 "Localization and Mapping" and provides a description of all software components for the mapping and localization part.

Keyword list: Mapping, Localization, Integration

Document History

Version	Date	Author (Unit)	Description
0.9	23.03.2016	Jörg Röwekämper, Alexander Schiotka	Initial version of document
1.0	11.04.2016	Jörg Röwekämper, Alexander Schiotka	Completed version of document

Table of Contents

1 Introduction

The ability to localize precisely is a fundamental skill in mobile robotics which is heavily relied on. Components such as a navigation module of a mobile robot build on the accurate pose estimate in a model of the environment. In the Stamina project we adapted previously developed software and focused on the robustness and usability of the software by an external expert user. We also developed new techniques to deal with dynamic environments. In particular we developed a module to update parts of a map and techniques to localize w.r.t. a single object.

Task 2.1 started in PM 1 and this deliverable reports on the work carried out. It provides details about the work regarding the deployment of the software, robustness tests and further research to handle dynamic environments, such as reorganization of the pallets or shelves in the warehouse.

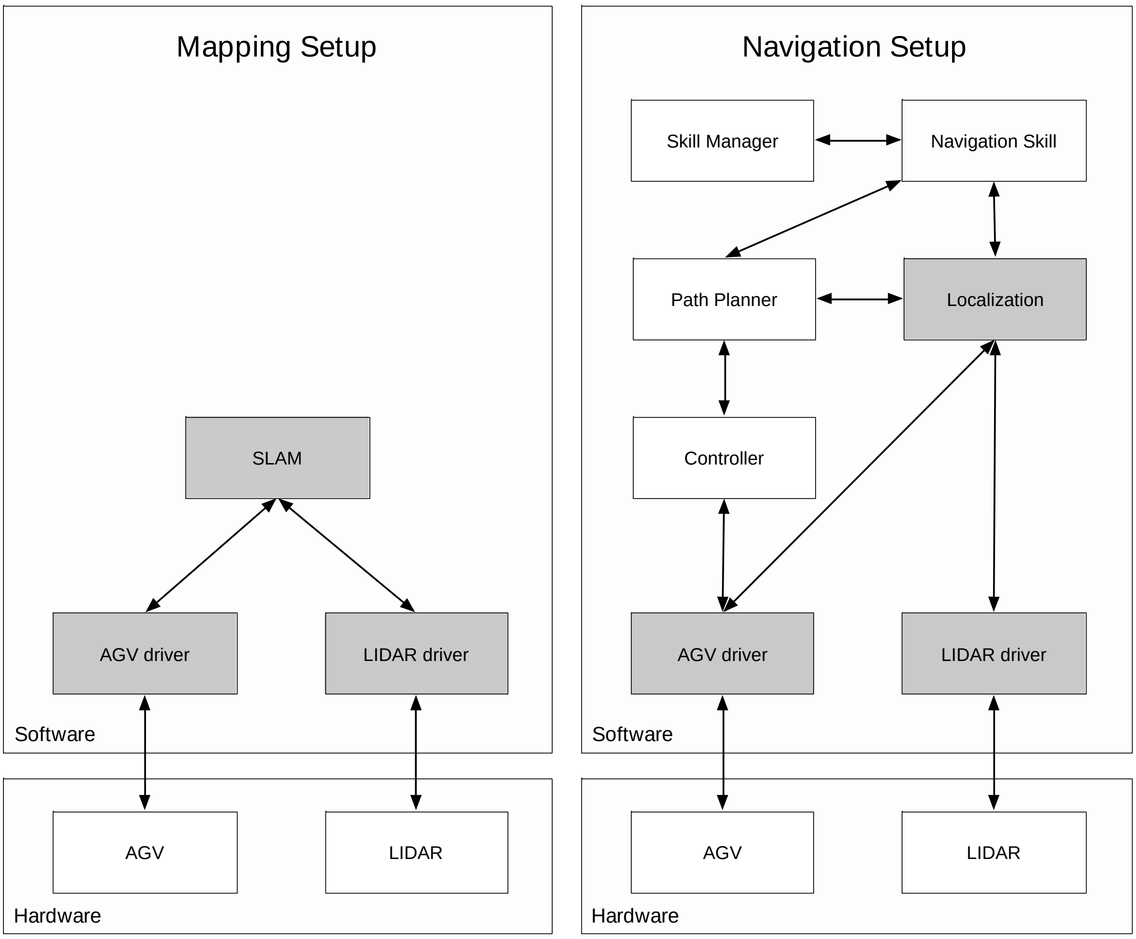

Figure 1: This deliverable embraces all colored modules of the illustration. They are part of the mapping process (left) where we build a global map of the environment in order to localize and navigate (right). Both setups have a hardware part which consists of the AGV and LIDAR (bottom). The drivers in both setups act as an interface for all other software components in the mapping and navigation modules.

2 Mapping and Localization

The Stamina Project requires a safe and robust navigation of the mobile platform in the warehouse in order to collect parts. For this the mobile robot has to stop in front of shelves and pallets so that the manipulator mounted on the robot can pick parts. For this it needs a model of the physical environment and relate it to the world model. In addition, the mobile platform needs the ability to localize in such a model.

Task 2.1 addresses the problem of modeling the physical environment and localizing the robot. We build up on the core technology developed in the EUROPA (http://europa.informatik.uni-freiburg.de) project for mapping the environment. The focus was on the usability by an expert user and the interfacing components with the Stamina hardware. The second part of Task 2.1 addresses the localization of the vehicle in the 2D map of the environment. For this we used the localization technology developed in the FIRST-MM Project (http://www.first-mm.eu). Within the Stamina Project we modified the localizer to be more robust at increased performance. We also performed experiments to test the robustness of the system when dealing with dynamics in the environment.

Further more we developed a technology to localize the robot with respect to a single object. The motivation for this was that pallets in the warehouse are roughly always in the same location but could be shifted. For example, if a pallet got replaced with a new pallet, then the new one would be placed at the same location as the old pallet but with some variation. We also developed a tool that allows to re-map parts of an area. This is useful if only one part of the warehouse gets reconfigured or if the environment has significant dynamics which makes remapping necessary.

In the following section we present first the results of a preliminary robustness test that we performed. Second, we explain the preparation and actions carried out for the software deployment. Then we present the results of our robustness experiments. In the end we present the results of our research to deal with dynamics in the environment.

2.1 Preliminary robustness test

As a preliminary test we analyzed the robustness of the mapping and localization software, to determine their suitability for the Stamina project. We used the benchmark protocol for indoor robot navigation proposed by Sprunk et al. [2]. The used benchmark specifies the type of the environment and the dynamics. It was designed to mimic an indoor office environment including common indoor dynamics. The Stamina robot has to navigate in an industrial setting. Even though the environments differ in the usage, they still have similarities in dynamics, e.g., groups of people moving in the environment, people standing in the path, objects standing around and blocking a path, or re-configurations of the environment (furniture). We used the omniRob as a platform for theses preliminary tests which is equipped with similar laser range finders. The reference system is a Pioneer P3-DX and used the ARNL 1.7.5.1 and BaseARNL 1.7.5.2 software stack. As base line we used the reference robot and compared it to our results which are presented in Tab. 1. We met our goal of zero localization failures while the robot traveled 1.4 km in 1.19 h during the experiment. The localization system was able to deal with all dynamics in the environment. The positioning accuracy is below 0.02 m. Therefore we expect that the mapping and localization system is well suited for the use within the STAMINA project. Further details can be found in [2].

Performance	ALU-FR	Reference	Ratio
Average speed	0.33 m/s	0.26 m/s	1.27
Average positioning error	$0.005\,\mathrm{m} \pm 0.007\,\mathrm{m}$	$0.05\,\mathrm{m} \pm 0.04\,\mathrm{m}$	0.10
Number of failures	0	0	-
Maximum time to failure	4343 s	5125 s	0.85
Maximum distance to failure	1423 m	1349 m	1.05

Table 1: Results for a preliminary robustness test of the navigation software.

2.2 Software Deployment

One of the main goals of Task 2.1 is to provide robust and easy to use software by an external expert user. The external expert user should be able to use the software without having to know the under lying algorithms such as the optimization framework g2o [3]. The mapping software has a user friendly GUI instead of a command line interface. Also the localizer can be initialized using the ROS RViz interface. The software has to run reliably and stably. Therefore we focused on robustness and usability of the software components for deployment. Parts of the localization software were parallelized to increase the efficiency. The number of external libraries, such as the Point Cloud Library (PCL) [4] including its dependencies on other libraries, is reduced to increase the stability and facilitate the future maintenance of the software. Before deployment we tested the software components on the stamina demonstrator in Freiburg. During the integration of the software we faced several challenges which are explained in more detail in section 3 (Lessons Learned). We also performed usability tests in Freiburg and at BA Systèmes in Rennes. For this we instructed users which never worked with the mapping and localization software before in 30 min. The users task was then to create a map of the environment and localize the robot in the map. We tested it with two users in Freiburg and one BA Systèmes employee.

All users were able to create a consistent map and localize the robot in the map without any external help. This demonstrates the ease of use of the software components. Further on we prepared a user guide which explains the installation and use of the software components. Figure 2 shows the outcome of the mapping and localization experiments in Freiburg. The red dots represent the particles of the localizer while the sensor readings from the front laser range finder are projected into the map in blue.

Figure 2: The map was created during a mapping experiment in Freiburg. The robot is localized in the map using the Monte Carlo Localization (MCL). The particle cloud is shown in red and current sensor readings are visualized in blue and reprojected into the map using the estimated robot pose.

2.3 Experiments

During the integration phase we performed multiple experiments to test the robustness of the mapping tool. Therefore we mapped a designated test area at BA Systèmes and the test sprint area at PSA with the STAMINA demonstrator. We repeated the experiment multiple times and checked consistency of the maps. The results are shown in Table 1. The first experiment (I) took place at a test area at BA Systèmes and failed once which was due to wrongly adjusted parameters. After adjusting these parameters we had a success rate of 100 % for the experiments II (test area at BA Systèmes) and III (test area at PSA) shown in Table 1.

In order to test the robustness of the localization system we performed extensive experiments in Freiburg. We used the AGV produced by BA Systèmes and recorded the duration of the experiments, how often a robot travelled to a target location, and how often it failed to reach a target location. Since the path planning and path execution module is still under development, we are especially interested in failures caused by inaccurate localization. Figure 3 shows the environment used for the experiments. The task of the robot was to navigate along the row of pallets and stop at potential picking positions. The target locations were chosen randomly and the robot navigated to 1285 location in total. The results of the experiments are presented in Table 3. The difference between the Static I, II, and III experiments is that we used a bigger safety margin to obstacles for the Static I experiment compared to Static II and III. This led to fewer failures in the path planning and execution software. For the dynamic experiments we modified the position of the seven pallets and checked if the robot is still able to navigate in the environment. The localization module failed zero times and also no other failures were related to inaccurate localization. This coincides with the results from the preliminary experiments from section 2.1. Therefore we expect that the robot should be able to localize robustly during the test runs in Rennes.

Experiment	Successful	Total	Success rate
I	2	3	66.7 %
II	3	3	100.0 %
III	3	3	100.0 %

Table 2: Results of mapping experiments during the integration phase.

Figure 3: Grid-map used for the long term localization experiments in Freiburg. We set up an aisle of a warehouse with four pallets on one side and three pallets on the other side.

Experiment	Duration	Targets	Failures in Total	Localization Failures
Static I	5.25 h	565	2	0
Static II	3.5 h	260	5	0
Static III	3 h	200	3	0
Dynamic I	3 h	200	2	0

Table 3: Results of localization at Freiburg performed with the AGV from BA Systèmes. In total zero localization failures occurred, which demonstrates the robustness of the Monte Carlo Localization (MCL).

2.4 Developments to handle dynamic environments

The experiments showed that the localization system is robust and able to localize the mobile platform in a dynamic environment. Nevertheless, we developed further techniques to handle significant dynamics or changes in the environment. The first deals with a partial reconfiguration of the environment and the second deals with displacements of new pallets at old locations.

Updating Maps

The first technique which we present performs mapping with known poses in a specified area. This has the advantage that it is not necessary to remap the whole area. Instead, small areas can be easily remapped. A user can specify this area or it can be specified by higher level software components. For example, if the world model registers that a pallet got replaced or shelves got reconfigured, it can request the remapping of an area. We create a local grid map and each map-cell represents the probability of occupancy of the environment. We update the map by integrating new sensor measurements from the LIDAR (Light Detection and Ranging) sensor. For this we perform ray casting for each laser beam on the grid map. When saving the map we merge the original and the local map. An example of an original map and an updated map is shown in Figure 4 with the remapped area indicated in red on the right.

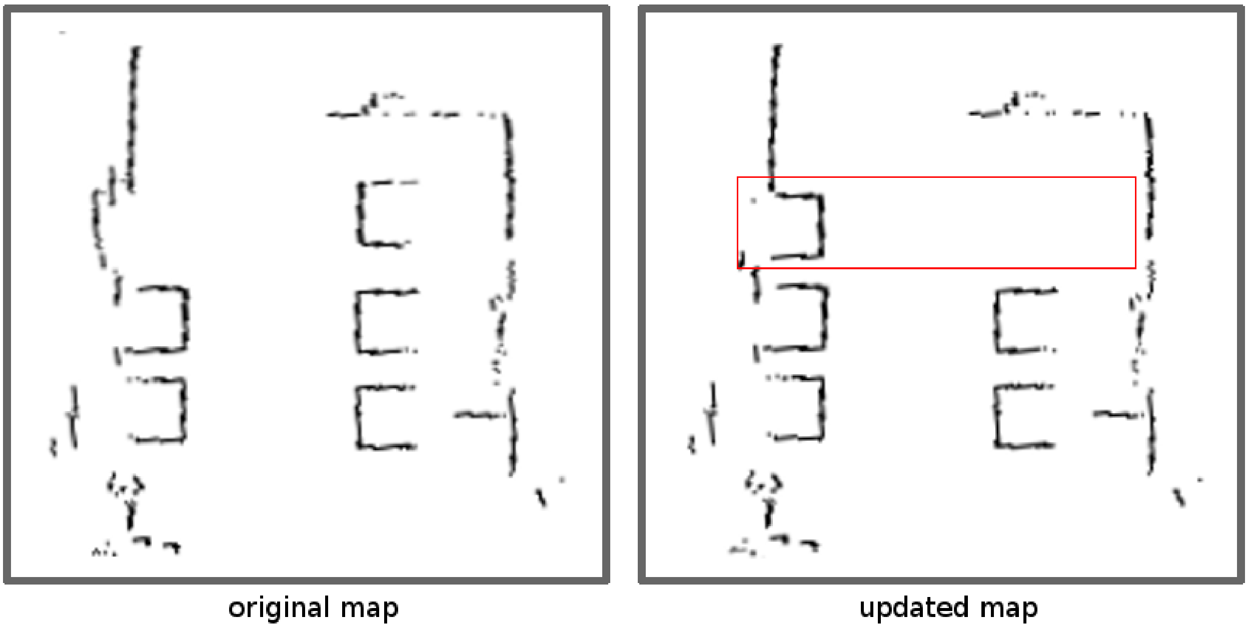

Figure 4: Original map (left) and the updated map (right) with a specified area (red border) which has been remapped

Localization with respect to a movable object

The second approach focuses on fine localization w.r.t. a movable object, especially if a pallet in the warehouse got replaced and the new pallet position deviates from the old position. To be able to localize w.r.t. a single pallet we developed a multi-body registration algorithm which uses a labeled model of the environment and object of interest. This model can be created from sensor data or from CAD data and is provided by the user. Figure 5 on the left shows a grid map of an environment with the particle cloud of the localization system in front of a box. The pose estimate is used as an initial guess for the multi-body registration. The results of the multi-body registration are shown on the right. The current sensor readings are aligned with the box and with the background.

A detailed description of the approach is presented in [1]. The multi body registration approach was tested in simulated and real world experiments. Figure 6 shows the results of our experiments. Set 1 through 6 are simulation experiments and set 7 through 9 are real world experiments. The maximum localization error was below 0.01 m for the all simulation experiments and below 0.02 m for all real world experiments. A detailed description and a discussion of the experiments are also presented in [1].

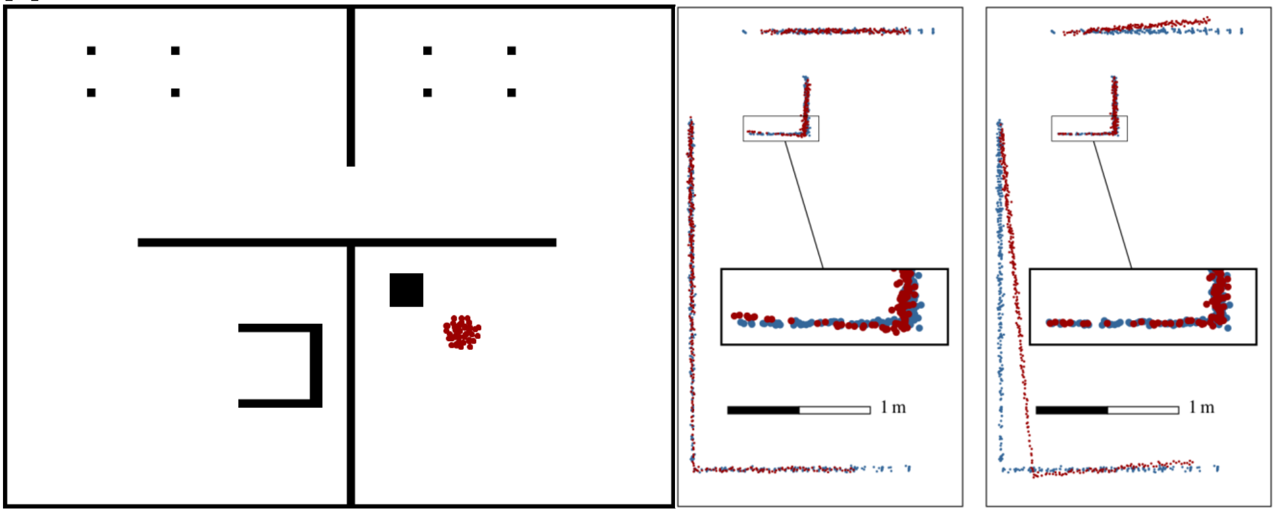

Figure 5: Grid map on the left with a particle cloud of the MCL and the result of the multi body registration on the right. The current sensor readings are aligned to the background (middle) and to the object (right) using multi body registration.

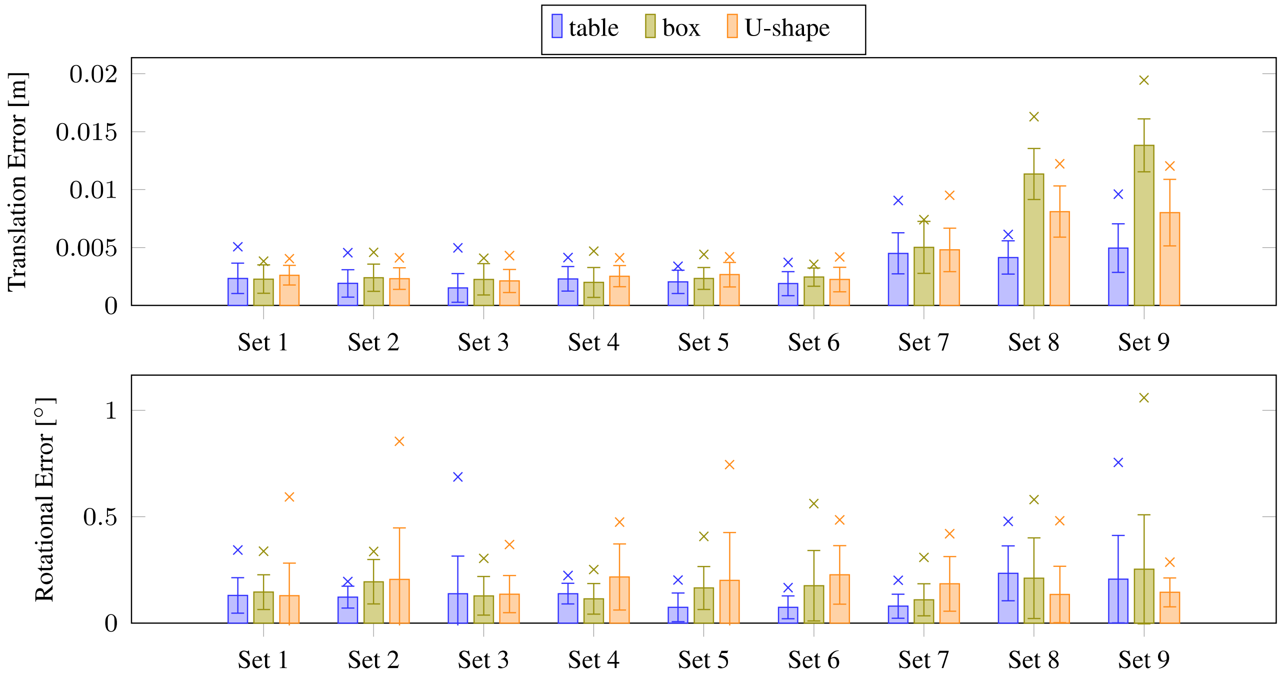

Figure 6: Results of the simulation real world experiments. The simulation experiments with the object shifted by 0.05 m (Set 1), 0.1 m (Set 2), rotated by 5° (Set 3), 10° (Set 4), simultaneously translated by 0.05 m and rotated by 5° (Set 5), and simultaneously translated by 0.1 m and rotated by 10° (Set 6). Results of the real world experiments with the object shifted by 0.1 m (Set 7), rotated by 10° (Set 8), and simultaneously rotated and translated (Set 9). The boxes indicate the mean error over all runs and the bars its standard deviation. The crosses represent the maximum errors.

3 Lessons learned

During the integration we encountered several issues which were caused by software bugs or by different understanding of definitions from project partners. In the following we describe the issues in detail: the appearance, the cause, and how we solved it.

The first issue was that the LIDAR driver did not send out any sensor readings. This was caused by two issues simultaneously. First, the LIDAR was not configured to put out a constant data stream of range measurements. We solved this by changing the internal sensor parameters. The second issue was that the ROS driver of the S300 sensor reported checksum errors with each package received from the sensor. After contacting the LIDAR manufacturer, we got the information that the checksum computation is depends on the internal sensor parameters, which is different from the description in the sensor manual. We solved this issue by adding an automatically checksum protocol detection. Hence, at startup the sensor driver determines first which out of four checksum protocols is used and applies the correct protocol.

Regarding the AGV driver we had three main issues. The first one is related to control commands. Whenever we sent a zero velocity command to the robot, the AGV did not stop. In fact the robot continued to drive with the last non-zero velocity command. This issue was solved by a firmware update by BA Systèmes.

The second issue was that any mapping attempt ended with an inconstant map. The reason was that we and BA Systèmes had a different understanding of odometry information. The AGV driver provided the transitional $\dot{x}_{ego}(t)$ and rotational $\dot{\theta}_{ego}(t)$ velocity in the robot ego coordinate system and the odometry position as

$$x(t_1) = \int_0^{t_1} \dot{x}_{ego}(t)dt$$
$$y(t_1) = 0$$
$$\theta(t_1) = \int_0^{t_1} \dot{\theta}_{ego}(t)dt$$

which is the integral of the transitional velocity in the ego coordinate system as x coordinate while y was at any time zero. The orientation of the robot was the integral of the rotational velocity. But we assumed that the odometry information would be computed in a SO(2) space as

$$x(t_1) = \int_0^{t_1} \dot{x}_{ego}(t)\ cos(\theta(t))dt$$
$$y(t_1) = \int_0^{t_1} \dot{x}_{ego}(t)\ sin(\theta(t))dt$$
$$\theta(t_1) = \int_0^{t_1} \dot{\theta}_{ego}(t)dt$$

where x and y describe the position of the robot in 2D plain and θ the orientation of the robot. This issue was solved with an update of the embedded AGV software.

The third issue occurred only after a few hours of operation and caused the localizer to delocalize. The reason for this was a time drift between the embedded clock of the AGV and the clock of the navigation PC. This led to wrongly time-stamped odometry information. To avoid this problem we already used software to learn the mapping between the two clocks and correct odometry time stamps. For this we ran a time server software on the embedded PC of the AGV and a client software on the navigation PC. The client connected to the server and estimated the mapping between the clocks. This was not working correctly because the embedded AGV PC had a Linux operating system with a real time layer which has a separate clock. The time server software was running in the Linux environment and used the Linux system clock and the software which computed odometry information was executed in the real time layer and used the clock of the real time layer. Hence, we learned the mapping to the wrong clock. After an update of the embedded AGV software, the time server software was also executed in the real time layer environment.

4 Conclusion

This document reported the results achieved in Task 2.1. The goal was to develop a robust and accurate mapping and localization system for the AGV. In detail, we adapted the mapping tool from the EUROPA project to fit to the hardware used within the STAMINA project. Also the localizer used from the First-MM project has been fitted and optimized. The mapper is easy to use by an external expert user. The experiments showed that it is robustly working after its customization to the new hardware. A grid-map of the environment is the output of the mapping software. The

system localizes the robot robustly in the grid-map. Experiments showed that the accuracy is sufficient for long-term navigation in dynamic environments.
Additionally we developed an approach for updating parts of the existing grid-map. This is useful for changes in the environment due to reconfiguration of pallets or containers. We also developed a method for localizing the mobile robot with respect to movable objects, e.g., pallets, containers, or shelves.

Parts of the work have been published in the following papers [1, 2].

References

[1] J. Röwekämper, B. Suger, W. Burgard, and G. D. Tipaldi. Accurate localization with respect to moving objects via multiple-body registration. In Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems (IROS), 2015.

[2] C. Sprunk, J. Röwekämper, G. Parent, L. Spinello, G.D. Tipaldi, W. Burgard, and M. Jalobeanu. An experimental protocol for benchmarking robotic indoor navigation. In Proc. of the International Symposium on Experimental Robotics (ISER), Marrakech/Essaouira, 2014.

[3] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, W. Burgard. g2o: A general framework for graph optimization. In Robotics and Automation (ICRA), 2011 IEEE International Conference o*n* (pp. 3607-3613). IEEE.

[4] R. B. Rusu and S. Cousins. 3D is here: Point cloud library (PCL). In IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, May 2011.