

Bachelor-Thesis Proposal: Multi-Robot Simulation of the RoboCup Logistic League with Fawkes and Gazebo

Frederik Zwilling 304314

June 13, 2013

1 Introduction

Autonomous multi-robot systems are about to play an important role in the future of logistics and production. They are able to save time and cost in logistics and the industrial production process. Existing systems in the warehousing domain, such as [6], already show the potential of autonomous robots to boost efficiency in many ways. Multi-robot-systems have a huge potential because they are able to do work distributed and in parallel, are efficient if they work together and are more reliable than single robots. Though, the development of multi-robot systems for logistics can be difficult because the robots have to handle many complex tasks. They have to detect objects, localize themselves, make a plan of what to do in which order, coordinate with other robots, optimize the work flow and be safe for humans and hardware all the time. *at all times.*

and makes it much harder

A key to effective and time saving development of reliable software for robots is simulation. Simulation makes it possible to test written code fast in different scenarios. This leads to a better quality of the software and better performance of the robot. By simulating the behaviour of a multi-robot system much time can be saved because there are no real robot which have to be set up and the simulation can quickly change between different scenarios. Simulation also enables testing if no real robots are available and can be easily used to compare different versions of the software. For example this can be useful to find unknown parameters such as thresholds.

The proposed bachelor-thesis will develop a multi-robot simulation environment for the *Logistic League Sponsored by Festo (LLSF)* and the robot software framework *Fawkes*¹. The multi-robot simulation primarily requires a simulation, that is able to properly simulate single robots. If this basis simulation works on a high level logic and a nearly physical level as well, most of the problems in the development of the robot system can be simulated. As a basis simulation, the open source simulator *Gazebo* was chosen.

In the following the proposal shows the background of the thesis in section 2. Section 3 gives an overview of related work. Other simulators and multi-robot simulations are shown there and it is argued why gazebo was chosen as simulator for this thesis. Section 4 presents the proposed work in detail. It defines goals, gives a schedule and discusses how the evaluation of the thesis can be done. The conclusion is found in section 5.

¹<http://www.fawkesrobotics.org>

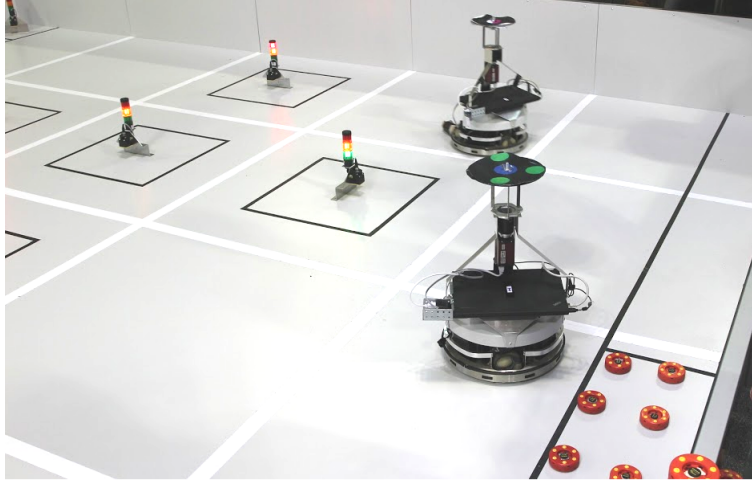


Figure 1: A half of the LLSF field. Two Robotinos of the CaroLogistics team at the RoboCup German Open 2013

2 Background

2.1 Logistic League Sponsored by Festo

The Logistic League Sponsored by Festo (LLSF) is part of the *RoboCup*, an international robotics competition founded in 1997. The RoboCup provides standard problems as a platform for artificial intelligence and robotics research [9]. There are different leagues in the RoboCup. The main focus is on soccer leagues, but there are more leagues, such as the leagues for rescue, domestic and logistic robots. LLSF is one of those leagues. LLSF aims to foster scientific work on autonomous solutions for logistics. The participants should find new approaches and improve already existing ones to optimize material and information flow in logistics. LLSF takes place in a simplified production hall [17]. Figure 1 shows a part of the 5.6m x 5.6m hall with two CaroLogistic robots, three machines and some material-pucks. The main task of LLSF is to produce and deliver ordered products as efficient as possible by feeding machines with resources and semi-finished products. The participants can use up to three Robotino robots by Festo [8]. They hold omni-directional wheels, a gripper to move pucks, and other sensors added by the participants. The orange pucks represent resources and products. They are equipped with an RFID-chip² which is needed to store the different product states of the pucks. The light-signals on top of the RFID-readers represent the machines. They convert resources or semi-finished products into products and trash. There are different types of machines. They can produce different product-types, produce intermediate products, recycle trash or are used for the delivery of products. The light-signals on top of the machines indicate the machine-status. Beside these visual elements, there is also a referee box (Refbox), which controls the machines and communicates with the robots during the game. The Refbox gives orders which products are to be produced, informs the robots about the game state and rewards points for achieved goals.

²Radio-frequency identification allows the wireless identification of objects with small chips.

2.2 Fawkes

Fawkes is an open source robot software framework developed by the Knowledge-based Systems Group³ at RWTH Aachen University. It is written for Linux and follows a component-based software design [15]. It provides an infrastructure to load and unload binary components, implemented as plugins, at run-time and provides a blackboard for communication between the plugins. A plugin can access the blackboard via interfaces. Furthermore it organizes the plugin activity in threads to make use of multi-core architectures. Because of this design, Fawkes is flexible and dynamic. The interchangeability of the plugins, which is caused by the well defined interfaces, makes hardware abstraction and reuse easy. This is also an important advantage for the development of the simulation because the simulation can easily exchange the lower-level robot control and sense plugins. The robot control plugins on higher levels can operate as usual because the interface is the same.

In comparison to the Robot Operating System ROS [18], which provides operating system like functionality, such as peer-to-peer communication, and a bunch of different libraries, Fawkes has a smaller community. On the other hand Fawkes is more dynamic and extendable because of the blackboard communication and the component-based design. This is an important advantage for this thesis.

2.3 Gazebo

Gazebo is an open source multi-robot simulator [11]. It simulates the behavior of robots and other objects physically in a three dimensional world. The open source engine Ogre⁴ is used for the graphical presentation of the simulation and allows realistic graphical environment. This is important for the simulation of camera sensors. Physics are simulated with the Open Dynamics Engine⁵. Gazebo was originally developed for outdoor applications and aims to simulate robots in a complex and realistic environment. Besides the physical simulation, it does this by generating data for different types of sensors. Gazebo can simulate laser range finders, cameras, sonar sensors, bumpers and more. This enables realistic simulation on a hardware-near level. A simulation in Gazebo is developed by creating robot and world models and developing Gazebo-plugins for behavior. These plugins can control objects in Gazebo and communicate with other software.

Why Gazebo was chosen as the most suitable simulator for this thesis, is shown in the section about related work.

2.4 High-Level Agent Design

The current implementation of the high-level agent of the Carologistics team uses the CLIPS rules engine⁶. CLIPS is a production system with rule-based, object-oriented and procedural programming paradigms. It uses forward chaining inference, can use functions and can be integrated into C and Java. This integration makes it easy to use CLIPS as a subsystem in other software systems. We have chosen the CLIPS rules engine, because using incremental reasoning the robot can take the next best action at any point in time, that is whenever the robot is idle, without costly re-planning (as with approaches using classical planners) and it allows us to cope with incomplete knowledge about the world, required in the LLSF scenario. Additionally, it is computationally inexpensive [16]

³<http://www.kbsg.rwth-aachen.de/>

⁴<http://www.ogre3d.org>

⁵<http://www.ode.org>

⁶<http://clipsrules.sourceforge.net>

that

This was due to the rules of 2012. In ~~this~~ year the robots had to deal with incomplete knowledge. They had to figure out the machine-types by trying to feed them with available pucks and looking for the response. By the rules of 2013, the Refbox provides the robots with the ground truth about the machine-types. Another change since 2013 is that the Carologistic Robotinos can use a mounted laptop for computing. In 2012 only one of up to three Carologistic Robotinos operated in the competition. All these points show that the multi robot coordination and planning is a necessary step to go further in LLSF.

3 Related Work

3.1 Existing Context

The work of the proposed thesis is embedded in an already existing context of work. The Carologistics team [2] participates in the LLSF and is a joint team consisting of the Knowledge-based Systems Group at RWTH Aachen University, the IMA/ZLW & IFU Institute Cluster at RWTH Aachen University and the Department for Electrical Engineering and Information Technology, Robotics Group at FH Aachen. The team has developed a working system for LLSF and participated in RoboCup 2012 in Mexico and RoboCup GermanOpen 2013. The next milestone will be the RoboCup 2013 in Eindhoven. Thus, there are many things to do. The skills and movement of the robot need to be made more robust. Communication and coordination between the Robotinos is necessary to produce with multiple robots at the same time and full system tests need to be done to ensure that all components work together. The work of Carologistics is based on Fawkes. It additionally uses ROS⁷ for navigation.

Some work was already done on Gazebo for Fawkes. Bastian Klingen developed a scene reconstruction for fault analysis in his diploma thesis [10]. The scenario was a mobile robot which has to pick up a colored cup with a gripper arm. During the execution of the picking-order, the motions, sensor data and belief of the robot are stored in a database. If the process fails, the scene reconstruction visualizes what happened from the robot's point of view to make fault analysis possible. The fault analysis is supported by a tree of calculation steps, which are needed to compute the knowledge of the system, and their dependencies. The visualization uses Gazebo and shows the robots movements and where he detected the cup. In the implementation Klingen also made a general Gazebo-aspect for Fawkes. This aspect provides basic access to the communication objects of the Gazebo API and instantiates them if a Gazebo instance is running. This aspect can be reused in the proposed thesis and may be enchanted if it is necessary or useful.

connected

There already is a primitive simulation, which is used by Carologistics for testing the high-level-agent. This simulation is implemented within the Clips agent and provides the agent with ground truth. For example the simulation tells the agent that a called skill, such as delivering a puck, succeeded after a few seconds after calling the skill. This is useful to test the behavior of the agent in the best-case scenario. Though, this simulation is only able to simulate the behavior on the highest level and can therefore not indicate many important real world problems. Furthermore the simulated situation is not visualized and it is difficult to evaluate the coordination between multiple robots because their position and actions are not visible and they can never block each others way.

?

⁷<http://www.ros.org/>

3.2 Comparison of Simulators

This subsection presents an overview of the mainly used robot simulators that are able to simulate LLSF. Afterwards, it is reasoned why Gazebo is chosen as the most appropriate simulator for the thesis.

3.2.1 Other Robot Simulators

An often used robot simulator is *Stage*⁸. It is part of the *Player/Stage* project [5] and provides an easy to use simulation environment, which can quickly be linked from Player, a robot control interface. Stage is an open source 2D-simulator and supports large populations of robots.

*Robotino Sim Professional*⁹ is a 3D-simulator for the Robotino developed by its manufacturer Festo. It is a commercial software and only usable on Microsoft Windows.

*Webots*¹⁰ is a commercial 3D-simulator [14]. It is platform independent and can simulate several robots at the same time. Its API functions are available in many programming languages and the simulator has many different robot and environment objects models out of the box.

USARSim [3] was originally intended as an urban search and rescue simulation but it can also be used for other domains. It is platform independent, free of charge for research and education, and used in the Rescue Virtual Robot Competition within the RoboCup.

*SIMRobot*¹¹ is an open source 3D-simulator and platform independent. It is used by multiple teams in the RoboCup [12]. It has been developed by the University Bremen since 1994.

The *Microsoft Robotics Developer Studio*¹² is free for noncommercial use and bound to Microsoft Windows. Besides the simulation it offers standardization for robotic control and several tools for robot development [7].

3.2.2 Advantages of the Gazebo Simulator

The Gazebo simulator has many advantages for this thesis [11]. It is open source what makes it free to use and easy to edit if something is missing. For example [10] extended Gazebo to be able to draw a coordinate system, which shows the location of found cups. It is important that Gazebo is a 3D-simulator because camera sensors can not be simulated in a 2D simulator. The physics engine and realistic graphics are indispensable for a realistic simulation. It is possible in Gazebo to assign for example friction and reflection values to an object. So, even the problem of a reflecting ground can be simulated if you are developing a light-seeking vision program. However, it has to be figured out which level of realism is best for the simulation. For example the behavior of the omni-directional wheels might differ in a physically correct simulation. Other advantages are that Gazebo runs in Linux and has an active community and a beginner friendly documentation. Furthermore, it already includes many commonly used robot-models and sensors. Therefore, it is not often necessary to implement own sensor plugins. In addition to that, Gazebo seems to be well suited for future work. It is a general multi-robot simulator and is well funded

⁸<http://playerstage.sourceforge.net/index.php?src=stage>

⁹<http://www.festo-didactic.com/int-en/learning-systems/software-e-learning/robotino-sim-view/robotino-sim-professional.htm>

¹⁰<http://www.cyberbotics.com>

¹¹<http://www.informatik.uni-bremen.de/simrobot>

¹²<http://www.microsoft.com/robotics>

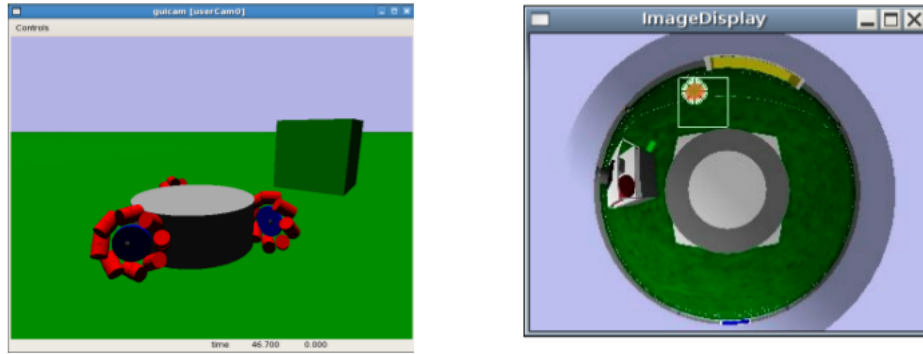


Figure 2: An omni-directional wheel prototype built in Gazebo (left) and the simulated picture taken of an omni-vision camera (right) [1].

by the Open Source Robotics Foundation¹³ and Willow Garage¹⁴. Besides that Gazebo was chosen as the basis for the *DARPA Robotics Challenge Simulator 2013*¹⁵. A more practical advantage of Gazebo is that some work was already done on Gazebo in Fawkes by [10].

There are not many disadvantages of Gazebo. One is that Gazebo can not handle a larger number of robots [11]. The complexity of the simulations limits the number of robots to the order of ten. This is no problem for this work because in LLSF there are only up to three robots and a control plugin for the machines at a time.

3.3 Other Multi-Robot Simulations

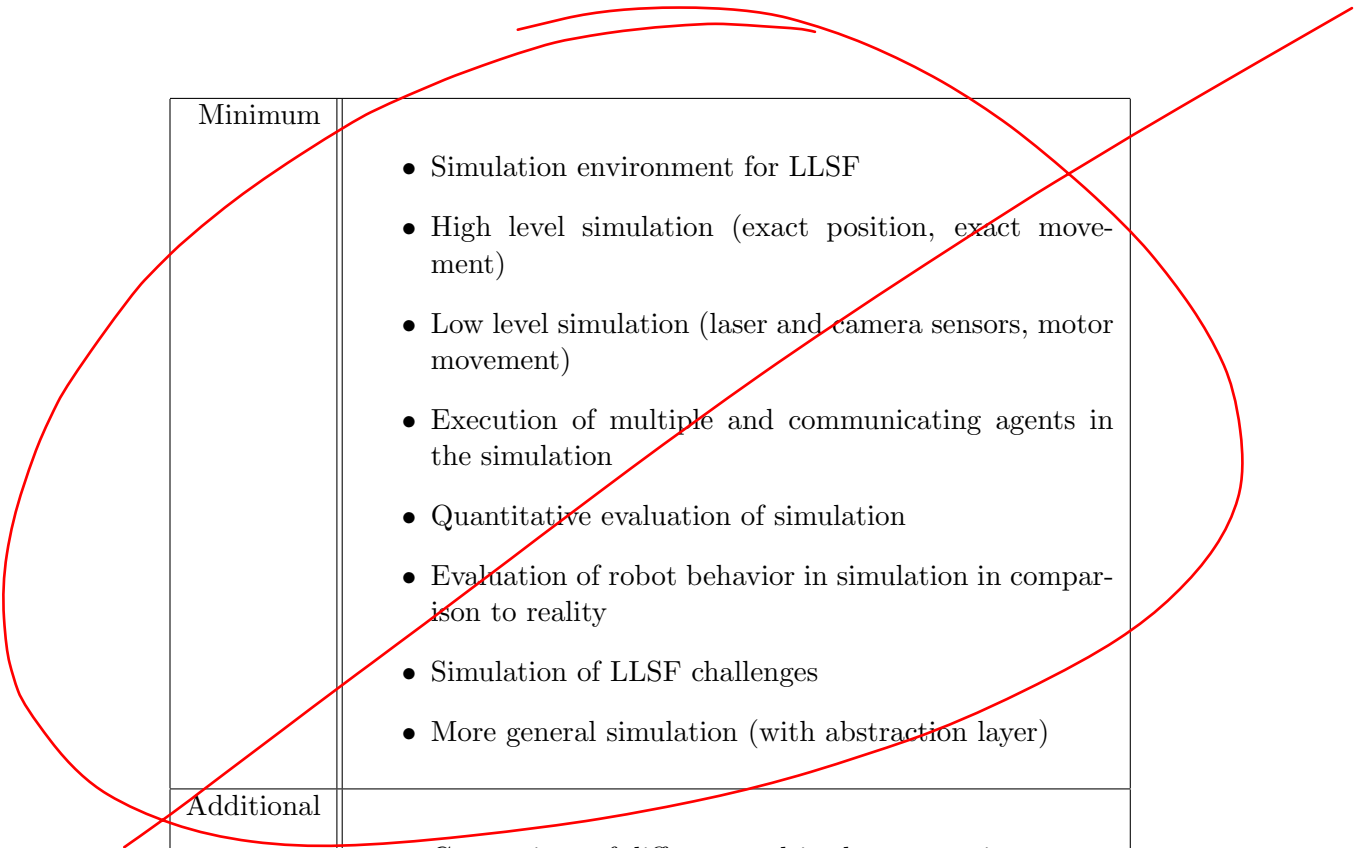
There are several other multi-robot simulation environments. [1] simulates the Middle-size Soccer League (MSL) of RoboCup and also works with Gazebo, but the communication is done with Player. In the MSL there are two teams with five robots under the size of 90 cm. Therefore, the simulation has to handle up to ten robots at once. In this work, similar equipment was simulated as used by Carologistics. [1] simulated prototypes of omni-directional wheels and an omni-vision camera. Figure 2 shows these two devices in the simulation. The omni-directional wheels of the Robotino work in the same way, but have a different design that needs to be modeled. The omni-vision camera mounted on the Robotino of Carologistics is very similar to the camera on the soccer-robots. The Gazebo-plugin for this camera might be a help for this thesis. The main point of [1] is the multi-level abstraction of the simulation. That means that the simulation is able to simulate with different levels of complexity. For example, the simulation can simulate on the physical level, where the robot software has to do movements and sensing as in the real world, and on a high logical level, where the robot moves exactly like intended and the simulation tells the robot exactly where it is located. These different abstraction levels are also important for the proposed thesis because they make the simulation more flexible and are well suited as milestones of the development of the simulation. [1] also shows that the speed of the simulation is an important aspect. If the simulation does not run fast enough to simulate in real-time, the behavior of the simulated robots may change because many processes running on the agent measure the time themselves. [1] dealt with

¹³<http://osrfoundation.org/>

¹⁴<http://www.willowgarage.com/>

¹⁵<http://spectrum.ieee.org/automaton/robotics/robotics-software/darpa-robotics-challenge-simulation-software-open-source-robotics-foundation>

make ref



Minimum	<ul style="list-style-type: none"> • Simulation environment for LLSF • High level simulation (exact position, exact movement) • Low level simulation (laser and camera sensors, motor movement) • Execution of multiple and communicating agents in the simulation • Quantitative evaluation of simulation • Evaluation of robot behavior in simulation in comparison to reality • Simulation of LLSF challenges • More general simulation (with abstraction layer)
Additional	<ul style="list-style-type: none"> • Comparison of different multi-robot strategies

Table 1: Goals defined for the thesis. The additional goal will be worked on last depending on how much time is left.

this problem by changing the source code of the robot-software to use the simulation time instead of the system time. This is also possible in Fawkes because all plugins can use the time provided by Fawkes.

Another related work is [4]. It uses Gazebo and Player to simulate a surveillance system with Pioneer3AT robots. In this system several robots have to monitor an outdoor area and find possible intruders. Therefore, the robots have to spread over the area and coordinate their movements. The simulation was important for the development of the system because it could quickly simulate different areas and situations.

The 3D-Soccer-Simulation League in the RoboCup uses Simspark as simulator¹⁶. In this league two teams of eleven NAO robots each play in the simulation. The competition and comparison between the different approaches of multi-agent strategies make this league interesting for the thesis. There are some works such as [13] which show the capability of the simulator to simulate complex movements and agent architectures.

4 Details of Work

4.1 Goals

This section presents the goals of the thesis and how they will be reached in detail. Table 1 shows an overview of all goals and their level.

Developing a simulation of LLSF means in detail to create a realistic environment in

¹⁶http://www.robocupgermanopen.de/sites/default/files/rules_G0_2013_3Dsim.pdf

the simulation similar to Figure 1. The communication between Fawkes and Gazebo will be realized by using Google's Protocol Buffers¹⁷. Google Protocol Buffers is commonly known as Protobuf. It provides a flexible and easy to use data exchange format. It is possible to define Messages in *.proto* files. These file can be compiled to generate code files in different languages, such as C++ or Java. This makes it simple to use Protobuf in the own code. The main reason for coosing Protobuf for communication is that it is used in the Gazebo API and in [10].

On the one hand the movement orders from Fawkes will be executed in Gazebo and on the other hand the Gazebo plugins will simulate the sensing and will send the data to Fawkes. These two sides of the simulation will be developed in different levels. In the higher level, Gazebo tells Fawkes the exact position of the simulated Robotino and all movement commands are executed exactly. In the lower level, the movement is realized by using the physics engine and only the rotation of the wheels is simulated. Fawkes receives only raw laser-data and captured images.

An important task of the thesis is the simulation of multiple robots controlled by separately running Fawkes instances in the same Gazebo environment at the same time. A global administrator plugin or a mechanism to map the Fawkes instances to the simulated Robotino plugins is necessary. The communication between the Fawkes instances has not to be simulated because the local network is available¹⁸. The details of evaluation are discussed after the next section.

Achieving abstraction in the implementation is important, so other developments beside LLSF can benefit from this thesis. The approach is to build additional abstraction layers for Fawkes and Gazebo plugins which encapsulates the communication between Fawkes and Gazebo as well as other functionality that is not domain specific. This is done as a second step to achieve a working simulation quicker and to benefit from the knowledge gained in the first run. The multi-level abstraction similar to [1] follows after the implementation of the high and low level simulation. Afterwards, it should be possible to choose the detail-level of the simulation. Another goal is to simulate the technical challenges of LLSF. For example there is a *Whac-A-Mole* challenge. A robot has to reach machines, which give a light signals, as fast as possible. This can be simulated by changing the behavior of the machines in the simulation environment. The comparison of different multi-robot strategies to solve the LLSF task is a long term goal of Carologistics. A efficient strategy has to be found and optimized to get good results in LLSF. The thesis can compare two strategies (e.g. Master-Slave job assigning or a job blackboard) to prepare further work and show the capabilities of the simulator. ~~In which detail this can be done depends on the time left for the thesis.~~

4.2 Implementation Structure

As said before, the work will structurally consist of plugins for Gazebo and plugins for Fawkes. Similar to [10] the Gazebo-plugins will be a robot-control plugin and an object-creator plugin. In addition, robot-sensor plugins and an LLSF-control plugin, which controls the machines and pucks to provide the LLSF game logic, are needed. On the other side, there will be Fawkes plugins for the sensors and for the Robotino to control the movements.

¹⁷<http://code.google.com/p/protobuf/>

¹⁸This allows to run Fawkes and Gazebo on different machines and can be useful to test a already set up Robotino before a competition.

	Weeks															
Tasks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Basis and high level simulation	■	■														
Building LLSF environment			■													
Physical level simulation				■	■											
Refactoring / abstraction					■	■										
Simulation of multiple agents						■										
Multi robot coordination							■	■								
Evaluation								■	■							
Writing										■	■	■	■	■	■	■

Table 2: Time-schedule

4.3 Schedule

The time-schedule of the thesis is shown in Table 2. There is enough buffer-time to handle unexpected problems or to go further, implement more features and reach higher goals. There are two weeks buffer on the practical level and two weeks after the writing. These buffers are this large due to the difficulty of estimating the needed time for the defined goals.

4.4 Evaluation

This section covers the evaluation of the proposed thesis. On the one hand a qualitative analysis on how well the simulation works from an obvious point of view and on the other hand a quantitative analysis of the performance and scalability of the simulation will be given.

The qualitative evaluation analyzes the functionality of the simulation and the realized features. It will evaluate which components and problem scenarios can be well simulated and where there are difficulties in the other problem scenarios. The aim of the evaluation is to compare the robot's behavior in the simulation to that in reality. To do this, it is necessary to look at the difference in the sensor data first because this may be the main cause for different behavior. Furthermore, the difference of multi-robot behavior needs to be analyzed. This will be more challenging because of the greater complexity.

In the quantitative evaluation the speed of the simulation is important because a difference between real-time and simulation-time can also cause differences in the behavior. The network latency can be a cause for difference in the coordination between the robots. Another part of the quantitative evaluation is to measure the CPU and memory usage. Because the thesis is about a multi-robot simulation, the scalability of the simulation has to be measured, too. This means to measure the simulation speed, CPU and memory usage in relation to the number of robots simulated.

5 Conclusion

The thesis proposed here will develop a multi-robot simulation environment for LLSF with Fawkes and Gazebo. It is the basis for efficient and quality-assured work of current and future developments. The simulation environment will concentrate on the LLSF league, which promotes research on logistic robotics to prepare a new logistical and industrial revolution. The proposal showed that Gazebo is well suited as simulator for this thesis

and future work. A part of the thesis will be built on previous work by [10] and [1]. The proposal also defined the main goals in Table 1. The evaluation will look at qualitative and quantitative aspects to ensure the quality and performance of the simulation. An important advantage of the thesis is the support future work. Although the simulation is primarily made for LLSF, it is also an important step for other developments because Gazebo can simulate almost every robot and other simulations can base on this thesis. Furthermore the proposal showed that a multi-robot simulator is a key to an effective development of multi-robot coordination.

References

- [1] Daniel Beck, Alexander Ferrein, and Gerhard Lakemeyer. A Simulation Environment for Middle-Size Robots with Multi-level Abstraction. In *RoboCup 2007: Robot Soccer World Cup XI*, pages 136–147. Springer, 2008.
- [2] Carologistics. Carologistics Team Description. ~~<http://www.fawkesrobotics.org/teams/carologistics/>~~, 2013. *Tech Report, KBS6*
- [3] Stefano Carpin, Mike Lewis, Jijun Wang, Stephen Balakirsky, and Chris Scrapper. USARSim: A robot Simulator for Research and Education. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 1400–1405. IEEE, 2007.
- [4] E Folgado, M Rincón, JR Álvarez, and J Mira. A Multi-robot Surveillance System Simulated in Gazebo. In *Nature Inspired Problem-Solving Methods in Knowledge Engineering*, pages 202–211. Springer, 2007.
- [5] Brian Gerkey, Richard T Vaughan, and Andrew Howard. The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In *Proceedings of the 11th International conference on Advanced Robotics*, volume 1, pages 317–323, 2003.
- [6] Erico Guizzo. Three Engineers, Hundreds of Robots, One Warehouse. *Spectrum, IEEE*, 45(7):26–34, 2008.
- [7] Jared Jackson. Microsoft Robotics Studio: A Technical Introduction. *Robotics & Automation Magazine, IEEE*, 14(4):82–87, 2007.
- [8] Ulrich Karras. Robotino—An Open Learning Mobile Robot System for Robocup, 2009.
- [9] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, and Eiichi Osawa. Robocup: The Robot World Cup Initiative. In *Proceedings of the First International Conference on Autonomous agents*, pages 340–347. ACM, 1997.
- [10] Bastian Klingens. Scene Reconstruction for post mortem Fault Analysis using the Gazebo Simulator from an on-line recording Database on a mobile Robot. Master’s thesis, RWTH Aachen, 2013.
- [11] Nathan Koenig and Andrew Howard. Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2149–2154. IEEE, 2004.

- [12] Tim Laue and Thomas Röfer. SimRobot-Development and Applications. In *The Universe of RoboCup Simulators-Implementations, Challenges and Strategies for Collaboration. Workshop Proceedings of the International Conference on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR 2008). LNCS (LNAI). Springer, Heidelberg. Citeseer, 2008.*
- [13] Patrick MacAlpine, Daniel Urieli, Samuel Barrett, Shivaram Kalyanakrishnan, Francisco Barrera, Adrian Lopez-Mobilia, Nicolae Ştiurcă, Victor Vu, and Peter Stone. UT Austin Villa 2011: A Champion Agent in the RoboCup 3D Soccer Simulation Competition. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 129–136. International Foundation for Autonomous Agents and Multiagent Systems, 2012.
- [14] Olivier Michel. WebotsTM: Professional Mobile Robot Simulation. *arXiv preprint cs/0412052*, 2004. ?
- [15] Tim Niemueller, Alexander Ferrein, Daniel Beck, and Gerhard Lakemeyer. Design Principles of the Component-Based Robot Software Framework Fawkes. In *Simulation, Modeling, and Programming for Autonomous Robots*, pages 300–311. Springer, 2010.
- [16] Tim Niemueller, Gerhard Lakemeyer, and Alexander Ferrein. Incremental Task-level Reasoning in a Competitive Factory Automation Scenario. 2013. full
- [17] ~~Organizing Committee of the RoboCup Logistic League. Rules of the Logistic League sponsored by Festo. <http://www.robocup-logistics.org/rules>, 2013.~~
- [18] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*, volume 3, 2009.