UNIVERSITÄT DUISBURG-ESSEN
FAKULTÄT FÜR WIRTSCHAFTSWISSENSCHAFTEN
INSTITUT FÜR INFORMATIK UND WIRTSCHAFTSINFORMATIK
LEHRSTUHL FÜR PERVASIVE COMPUTING

Masterarbeit

ORIGNIAL FORMAT for the title of the work

First Name Second Name
Matrikelnummer: 123456

Networked Embedded Systems Group
Institut für Informatik und Wirtschaftsinformatik
Fakultät für Wirtschaftswissenschaften
Universität Duisburg-Essen

25. September 2020

Erstprüfer: Prof. Dr. Pedro José Marrón
Zweitprüfer: Prof. Dr. (First Name)(Second Name)
Zeitraum: 1.Januar 2010 - 1.Juni 2010

Contents

Abstract

The function of the abstract is to summarize, in one or two paragraphs, the major aspects of the entire bachelor or master thesis. It is usually written after writing most of the chapters.

It should include the following:

- Definition of the problem (the question(s) that you want to answer) and its purpose (Introduction).
- Methods used and experiments designed to solve it. Try to describe it basically, without covering too many details.
- Quantitative results or conclusions. Talk about the final results in a general way and how they can solve the problem (how they answer the question(s)).

Even if the Title can be a reference of the work's meaning, the Abstract should help the reader to understand in a quick view, the full meaning of the work. The abstract length should be around 300 words.

Abstracts are protected under copyright law just as any other form of written speech is protected. However, publishers of scientific articles invariably make abstracts publicly available, even when the article itself is protected by a toll barrier. For example, articles in the biomedical literature are available publicly from MEDLINE which is accessible through PubMed. It is a common misconception that the abstracts in MEDLINE provide sufficient information for medical practitioners, students, scholars and patients[citation needed]. The abstract can convey the main results and conclusions of a scientific article but the full text article must be consulted for details of the methodology, the full experimental results, and a critical discussion of the interpretations and conclusions. Consulting the abstract alone is inadequate for scholarship and may lead to inappropriate medical decisions[2].

An abstract[IGM97, **?**, MAdR02, Sal89] allows one to sift through copious amounts of papers for ones in which the researcher can have more confidence that they will be relevant to his research. Once papers are chosen based on the abstract, they must be read carefully to be evaluated for relevance. It is commonly surmised that one must not base reference citations on the abstract alone, but the entire merits of a paper.

Chapter 1

Introduction

[You should answer the question: What is the problem?]

This paragraph should establish the context of the reported work. To do that, authors discuss over related literature (with citations[1]) and summarize the knowledge of the author in the investigated problem.

ToDo: how to make citations

An introduction should answer (most of) the following questions:

- What is the problem that I want to solve?
- Why is it a relevant question?
- What is known before the study?
- How can the study improve the current solutions?

To write it, use if possible active voice:

- We are going to watch a film tonight (Active voice).
- A film is going to be watched by us tonight (Passive voice).

The use of the first person is accepted.

1.1 Motivation

A good introduction usually starts presenting a general view of the topic and continues focusing on the problem studied. Begin it clarifying the subject area of interest and establishing the context (remember to support it with related bibliography).

[1] To cite a work in latex

1.2 Problem definition

Additionally, focuses the text on the relevant points of your investigation and problems that you want to solve, relating them with the first part.

1.3 Thesis/Diplom/Bachelor/Master Structure

Present your work to the reader giving a brief overview of what is going to cover every chapter. Write only general concepts, no more than one or two sentences per chapter should be necessary.

Chapter 2

Materials and Methods

This section is to clarify the pre-existing tools, defining what was developed in this field until now, and why this tool was used instead of others.

The general structure is the following:

- Definition of the specific tool(s) studied (robots, sensor nodes, smart-phones). When relevant, pre-existing experiments.
- Definition of the context of use (indoor/outdoor, humans/animals/robots, with/without connection).
- Definition of used protocols (How the data are collected, when, etc.)

Chapter 3

Approach

In our system there are multiple robots that must handle various tasks. For example, visiting given rooms. To tackle this problem, a communication efficient task scheduling system is designed. This system allocate task according to system resources, including environment factors and robot available battery. Once this information is attained, the task scheduling system assign robot a set of task.

- *Robot.* The robot is responsible for moving in 2-dimensional physical space as well as listen to sensors on its way. It has a rechargeable battery. Robot battery level drops as its moves and rotates.

- *Tasks.* Each task requires one or more robots to traverse a path in the workspace and carry out certain actions.

- *Environment.* We consider robots moving in an office that contains a corridor along the central x-axis and 16 rooms located around the corridor. The environment model is shown in Figure 3.1. The environment factors, such as room locations and occupancy possibilities help task allocation.

3.1 Architecture Design

As shown in Figure 3.2, the architecture of the system consist of several parts: centralized pool, robot controller, navigation stack, charging station and system environment.

- *Centralized Pool.* A centralized pool consist of several modules: multi-robot task allocation module, system environment state, database, execution and monitoring. The database contains the room information such as occupancy possible and the tasks. The multi-robot task allocation module assign task to robots according to both robot status and system environment.

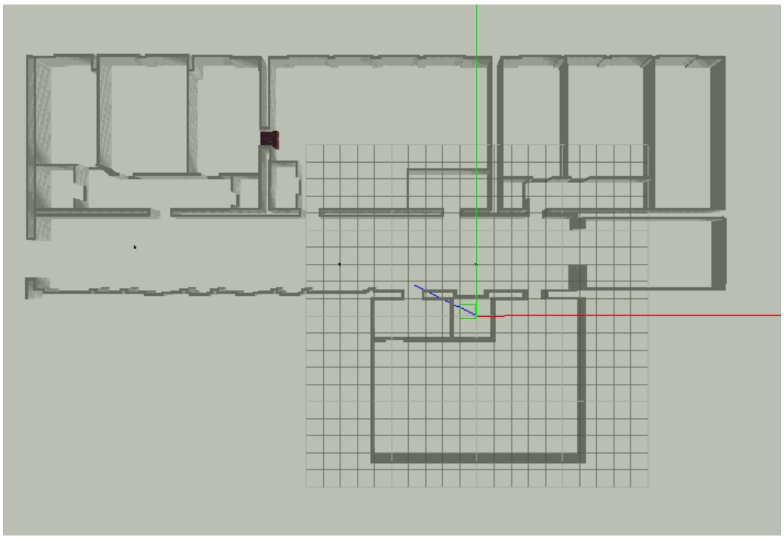

Figure 3.1: Gazebo Model

- *Robot Controller.* A robot controller contains several modules: execute module and robot action. The execute module receive commands from centralized pool and decide when and which task the robot should perform. During performing a task, a robot can send environment information to centralized pool.

- *Navigation stack.* The move_base node provides a ROS interface for configuring, running, and interacting with the navigation stack on a robot. It makes robot move to desired positions using the navigation stack. Its advantages include optionally performing recovery behaviors when the robot perceives itself as stuck.

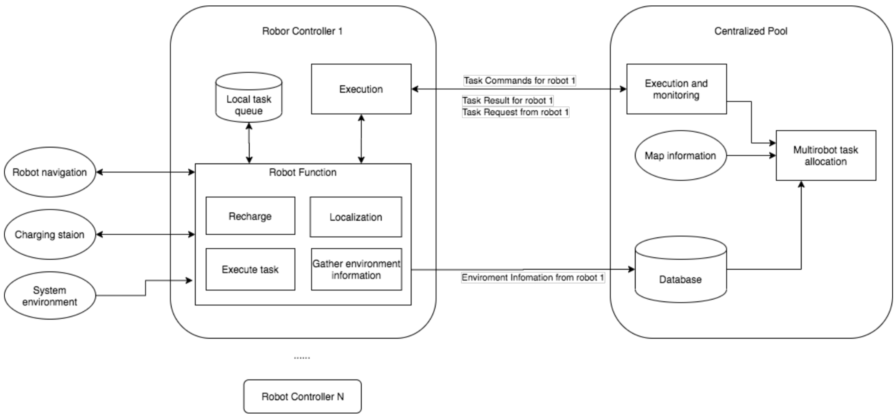

Figure 3.2: Multi-robot task allocation and execution system architecture

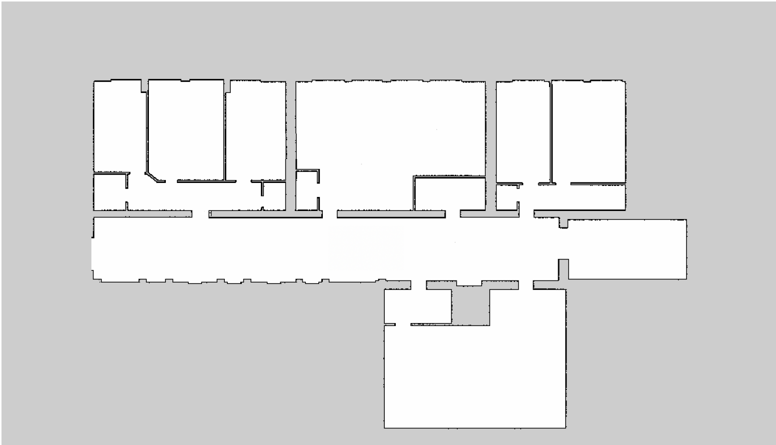

Figure 3.3: Environment occupancy grid

3.2 System Environment

A 3D Gazebo module shown in 3.1. It is assumed that there are no dynamic obstacles. As is shown in 3.3, the black lines are occupied area, which is the wall in 3D-Model. The gray area is unknown are. The white area is unoccupied. In unoccupied area there are following important area and coordinates:

- *Rooms.* The rectangle areas (Figure 3.4) are used to represent Rooms. Each rectangle has its upper and lower limit in x and y coordinates, in order to identify which rooms the robots or targets belong to.
- *Doors* The positions of doors (Figure 3.5) are stored in database. There are used by ROS door simulator nodes, which broadcast thier position and door status periodically. The broadcast messages are received and filtered by robots.
- *Charging Stations* The positions of charging stations are used by ROS charging station nodes. For details please refer to Section 4.3.3.

3.3 Task Allocation

3.3.1 Task Specifications

In order to improve overall execution efficiency, one single robot can carry either small task, which make robot move to one position, or large task, which consist of multiple small task. On one hand, to make robot work long hours in office environment, recharging is necessary. On the other hand, a robot should gather environment information as

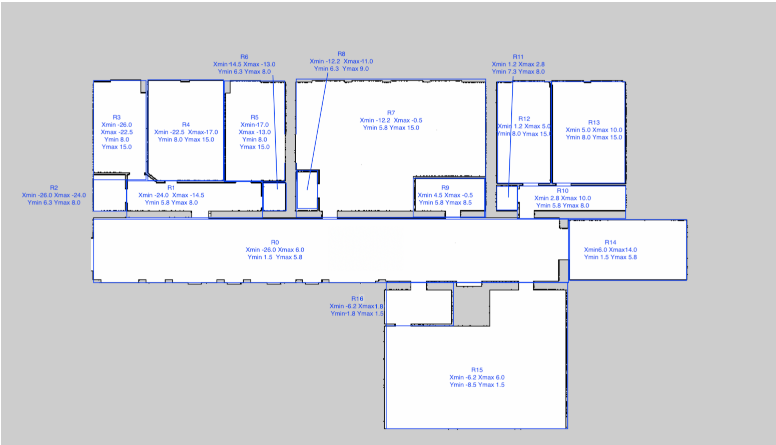

Figure 3.4: Room division

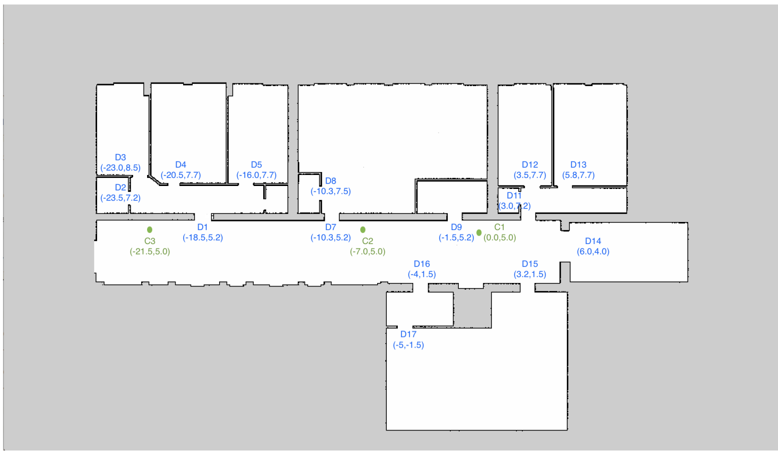

Figure 3.5: Doors and Charging Stations

much as possible, which centralized pool would learn from and make better decision. Therefore, three types of task are defined, which are shown in Table 3.1 and Table 3.2. Task attributes includes task type, target, size, dependency, priority, generator, the way of handle task, start time, finish time.

Task Attribute	Target	Dependency	Priority	Generator
GatherEnvironmentInfo	Door	No	1	Centralized Pool
Execute task	Any point	Dependent on Execute-task	2,3,4	User
Charging	Charging station	No	5	Centralized Pool

Table 3.1: Task arributes part 1

Task Attributes	Start time	Finish time
Explanation	The time when robot start moving	The time when robot finished interacting with the target

Table 3.2: Task attributes part 2

Task Id	Task Type	Start Time	Target Id	Robot Id	Priority	Status	dependency	Finish Time	Description
1	2	10:00:00	10:59:59	0.80	2	RanToCompletion	0	2020-06-01 9:00:00	Succeeded

Table 3.3: Task Table in Database

- *Task Size.* One single robot is able to carry out one Charging-task or gather environment information task, but can carry multiple Execute-task, thereby These tasks with dependencies also referred to as small task. The example of small task includes Charging-task and gather environment info task. Those small tasks form a dependency chain, also referred to as a large task. Execute task can be a large task which let the robot move continuously to several positions.

- *Target.* Targets include door, point and charging station. When a robot run a gather environment information task, it moves to the front of the door and interact with a sensor in door position, without entering the door. When robot run an Execute-task, the robot moves to a given point ether in corridor or in the room. When robot run a Charging-task, the robot moves to charging station and interact with charging station.

3.3.2 Execute Task Allocation

With robot information such as positions and available battery provided by robot controller, the multi-robot task allocation module in the architecture should perform multi-robot task allocation. To select an Execute-task, the following decision variable are considered.

Door Id	Door Status	Date Time
1	1	2020-06-01 09:00:01

Table 3.4: Measurement Result

Door id	Day Of Week	Start Time	End Time	Init Open Possibility	Open Possibility Statistic
1	2	10:00:00	10:59:59	0.80	0.80

Table 3.5: Door Open Possibility

Decision variable

- *Task Priority.* Task priority. Task priority is an important factor that describes task emergency level. The Charging-task has the highest priority of 5. The gather environment task has the lowest priority of 1. The Execute-task is determined by users but must be in the range of [2,4].
- *Product of Door Open Possibility.* Because of the limitation of simulation, the door open possibility is used to represent room occupancy. The door open possibility is based on the statistic of door measurement in a specific time period of each working day. The doors that the robot may pass through can be obtained from the map information module. An example of door measurement result is shown in Table 3.4, an example of door open possibility table is shown in Table 3.5.
- *Waiting Time.* The waiting time is the difference of current system simulation time and start time of the first task to be executed. $T_{waiting} = T_{first\_task} - T_{now}$
- *Battery Consumption.* The Battery Consumption is related to robot trajectory. If a robot get a Large Execute-task that contains n small task, Equation 3.1 can be used to calculate battery consumption. The centralized pool will send the task with the lowest cost to this robot.

$$
\begin{aligned}
&B : Battery\_ConsumptionW : Weight \\
B_{large\_task} &= \sum_{task_0}^{task_n} B_{trajectory} \\
&= \sum_{task_0}^{task_n} \sum_{waypoint_0}^{pont_M} [W_{position} \times position\_variation + W_{angle} \times angle\_variation] \\
&= \sum_{t=task_0}^{task_n} \sum_{p=waypoint_0}^{waypoint_M} [W_{position} \times \sqrt{(x_p - x_{p-1})^2 + (y_p - y_{p-1})^2} \\
&+ W_{angle} \times 2 \times \arccos(w_p)]
\end{aligned}
\tag{3.1}
$$

In conclusion, equation 3.2 can be used to calculate cost of a large Execute-task.

$$\begin{aligned} Cost_{Large\_execute\_task} = & \frac{W_{battery} \times Battery\_consum}{n} + W_{waiting} \times Waiting\_time \\ & + W_{possibility} \times \prod_{i=1}^{n} Door\_open\_possibility + W_{priority} \times Priority \end{aligned} \tag{3.2}$$

3.3.3 Environment Task Allocation

Once robot request a task, The task allocation module select only tasks with cost below the threshold. Once either no task in database or all tasks cost above the threshold, the task allocation module should create a gather environment information task, in order to make more measurement result and further more improve the accuracy of door open possibility table. To create a gather environment information task, which door should robot visit must be considered. The following factor help task allocation module to select the door.

Decision variable

- *Door Last Update Time.* The last update time is decision variable related to gather environment information task.
- *Battery Consumption.* Similar to Execute-task allocation, the battery consumption is related to trajectory from robot to the door. Equation 3.1 can be used to calculate battery consumption.
- *Whether door is used.* If another robot is going to this door, the value is 0, otherwise the value is 1.

$$\begin{aligned} Cost_{door} = & \frac{W_{battery} \times Battery\_consum}{n} + W_{time} \times (T_{last\_update} - T_{now}) \\ & + W_{possibility} \times \prod_{i=1}^{n} Door\_open\_possibility + W_{is\_used} \times Is\_used \end{aligned} \tag{3.3}$$

3.3.4 Charging Task Allocation

Once the robot send task request to the centralized pool, and the centralized pool figure out this robot need charging, it should create a Charging-task for robot. Since there are multiple charging station in the system environment (As is shown in Figure 3.5), the centralized pool select a charging station for this robot using the following decision variable.

Decision variable

- *Remain Time.* It describes how long the robot is charging now will take.
- *Battery Consumption.* Similar to Execute-task allocation, the battery consumption is related to trajectory from robot to the door. Equation 3.1 can be used to calculate battery consumption.

In conclusion, equation 3.4 can be used to calculate cost of a charging station. The centralized pool will generate a Charging-task to charging station with the lowest cost to robot which requests a task.

$$Cost_{charging\_station} = \frac{W_{battery} \times Battery\_consum}{n} + W_{time} \times T_{remain} \tag{3.4}$$

Chapter 4

Implementation

4.1 Communication Protocols

Centralized pool and robots need to share task information with each other. There are some basic requirements for communication: firstly, robot should initiate the communication once it has finished all task in task queue and get free. This is solved by assigning robot controller as ROS service client and centralized pool as ROS service server. This method saves unnecessary communication cost by avoiding keep tracking the current position, availability and states of all robots. Secondly, robot should forward sensor data to centralized pool while processing a task. This is solved by assigning robot controller as ROS action server and centralized pool as ROS action client. As is shown in 4.1, an efficient communication protocols is designed.

4.1.1 Message about Measurement

When a robot pass by a door, it will receive messages from sensor. In order to save resources, instead of a system enviroment in real life, a door simulator ROS node is used to publish all measurement result for all doors. The message formart is shown in Table 4.1 The measurement results(opened and closed) are created according to the "open possible" column in the "open possibilities" table (Figure 4.2).

Door ID	Position	Timestamp	Measurement Result
1	(-18.5,5.2)	2020-06-01 9:00:02	Door opened

Table 4.1: Measurement Message Format and Example

4.1.2 Message about Task

Four types of message are defined: (1)Task request message(Table4.2); (2) Task goal messages(Table 4.3); (3) Task feedback message (Table 4.4); (4) Task result message (Table 4.5).

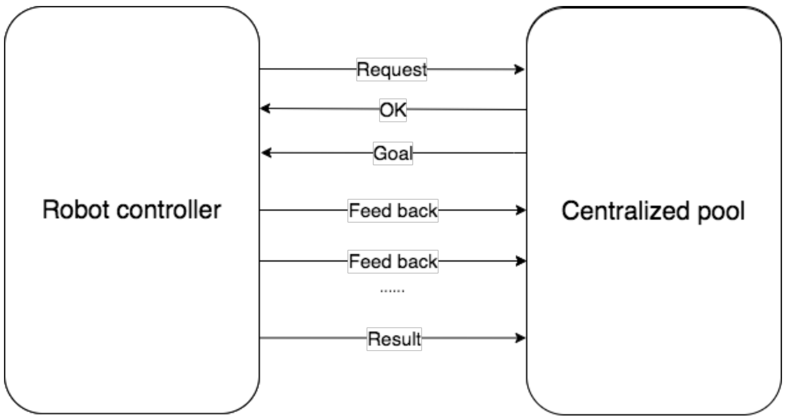

Figure 4.1: Communication between Robot and Centralized Pool

Battery	Pose	Robot ID
93	(2,4)	1

Table 4.2: Request Message Format and Example

Task id[]	Task type	Target id	Goal[]
1	Gather Environment Info	9	(-1.5,5.2) 2020-06-01 9:00:00
[3,4]	Execute task	21, 22	(-24.0,12.0), 2020-06-01 9:02:00 (-21.0,12.0) 2020-06-01 9:02:00
5	Charging	17	(0.0,5.0), 2020-06-01 9:04:00

Table 4.3: Action Goal Message Format and Example

4.1.3 Message about Charging

Figure 4.6 shows the message a robot sends to Charging station when it arrives charging station's position.

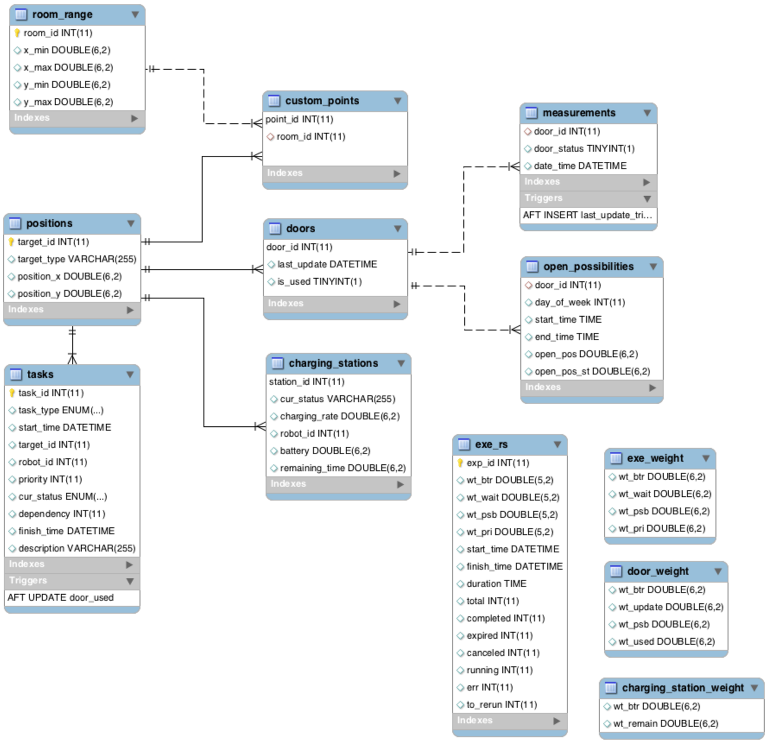

Figure 4.2: Database Entity Relationship Diagram

Robot id	Door id	Measurement time	Measurement result
1	3	2020-06-01 9:00:03	Door open

Table 4.4: Action Feedback Message Format and Example

Task id	Task type	Result
1	Gather Enviroment Info	Success

Table 4.5: Action Result Message Format and Example

4.2 Database

The centralized pool keep information it requires, to make decisions. Figure 4.2 shows the relationship between entities. Following are explanations of some tables.

- *Table doors.* Table doors stores enviroment information. In table doors, the is_used column will be updated when an Enviroment-task to this door is updated. The last update will be updated when the centralized pool receive a new measurement result.

- *Table open_ossibilities.* Table open_ossibilities is based on the statistic of door measurement in a specific time period of each working day. These two table will be updated when centralized pool receive a new measurement result.

- *Table exe_rs.* Table exe_rs stores experiment result while table exe_weight, table door_weight and table charging_station_weight stores weight values for experiment. Chapter 5 introduces the details of experiment.

- *Table costum_points.* When user gives the system a task, the target point of this task will be stored in position table, and a target_id will be generated. This target_id will be stored in custum_point table. Additionally, with the information in room_range table, the system will recognize which room does the point belong to and write room_id column in costom_points table.

Robot ID	Battery Level
1	93

Table 4.6: Message to Charging Station

4.3 Procedure

As stated in the Chapter 3, the goal of task scheduling is finishing all tasks as soon as possible while keep the cost as low as possible. The task assignment and execution happends at two level. [IGS17] the task and the path planner solves a planning problem. It takes and occupancy grid, a specific robot and a set of task specifications, and generates trajectorys for each task. According to those trajectorys and task specifications, the task with the lowest cost is assigned to robot. At the dynamic level, after each robot receive a task, it runs a navigation stack to execute this task stepwise. Each robot computes a local trajectory but takes into account dynamic obstacles. The process of the robot task allocation system is as follows.

4.3.1 Centralized Pool

Task Allocation When the centralized pool receives a task request (Table 4.2) from robot, it performs task allocation. The task allocation algorithm is discussed in Section 3.3.2. The process of task allocation is shown in Figure 4.3.

1. When the battery of robot belows 10%, the centralized pool create a charging task to the charging staion with the lowest cost.
2. When the battery of robot aboves 10%, the centralized pool loads execute-tasks in database, then combine small tasks with dependencies into large tasks, finally calculates task costs and select one large task with the lowest cost.
3. If there are no suitable tasks, a gather-enviroment-task to the door with the lowest cost is generated.

The difference between task is discussed in Section 3.3.1. The output of the task allocation includes: task ID, goal coodinate, timestamp and selected robot ID. The task is sent to the selected robot, and the robot performs the tasks.

Handle Task Feedback. When the centralized pool receives a task feedback (Table 4.4) that contains a new measurement result from robot, it will add a record in measurement table and update "open possibilities" table in database.

Handle Task Result. When the centralized pool receives a task result (Table 4.5), it updates "tasks" table (Figure 4.2). The failed "execute tasks" will be reused while the failed others will be marked as "Cancel" or "Error"(Figure 4.4).

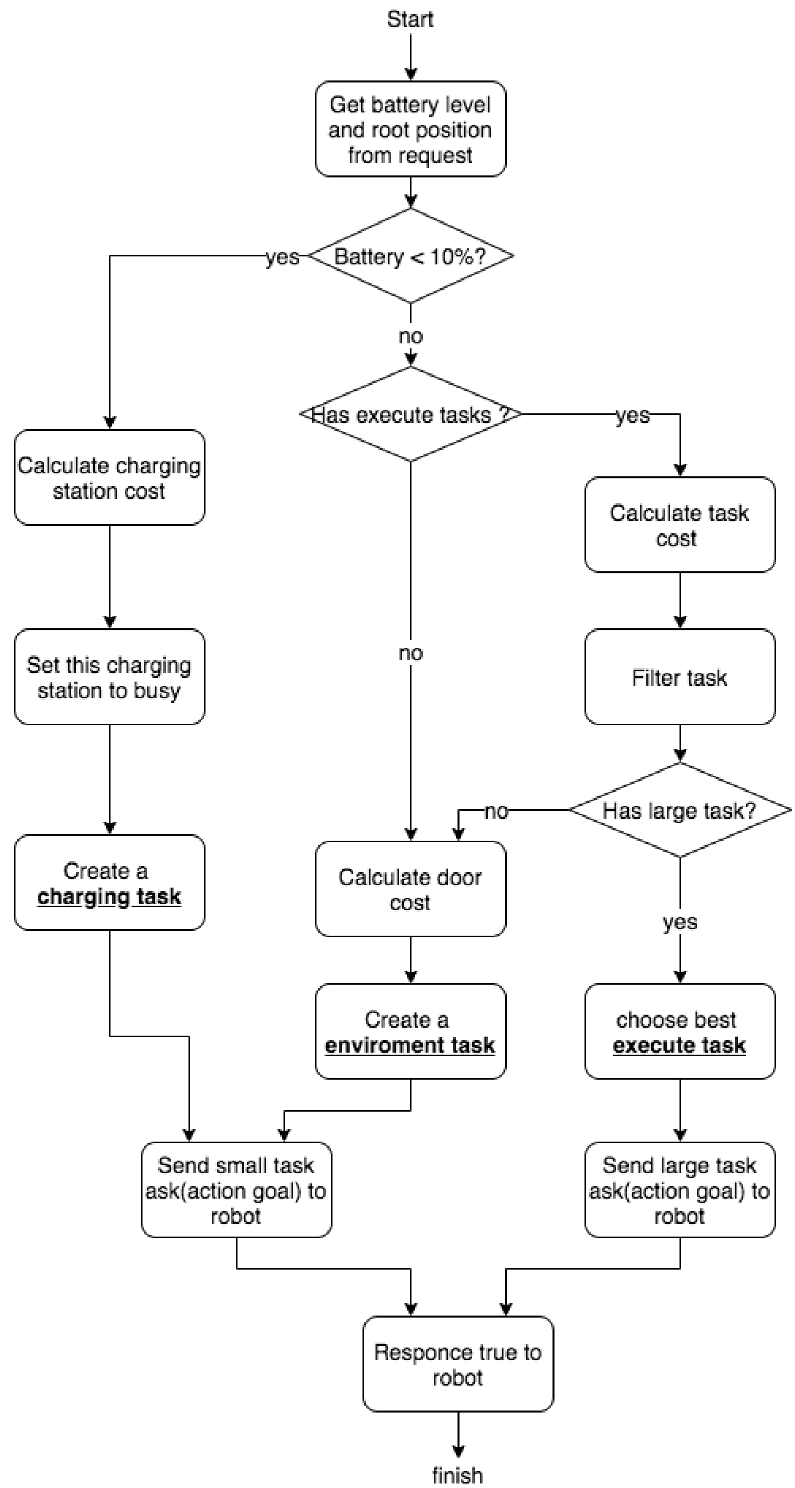

Figure 4.3: Centralized Pool Task allocation

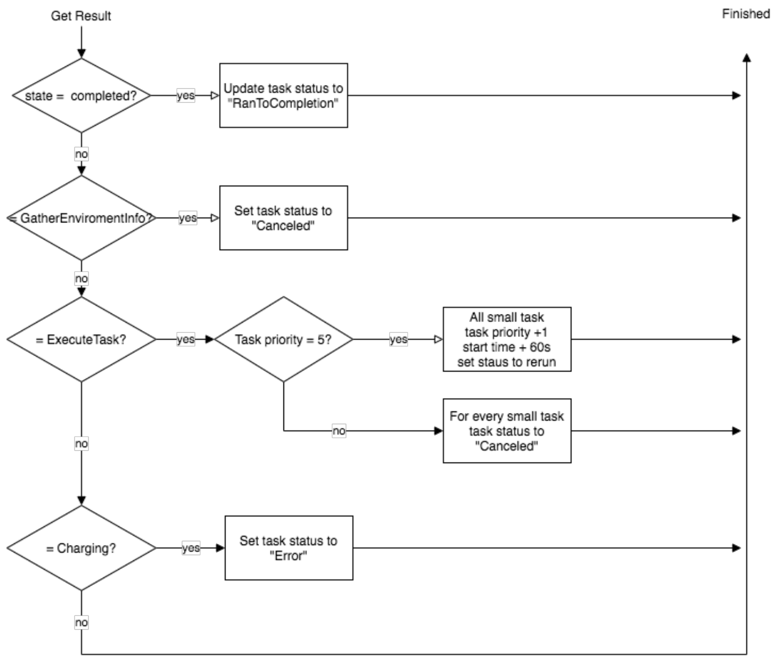

Figure 4.4: Centralized Pool Handle Task Result

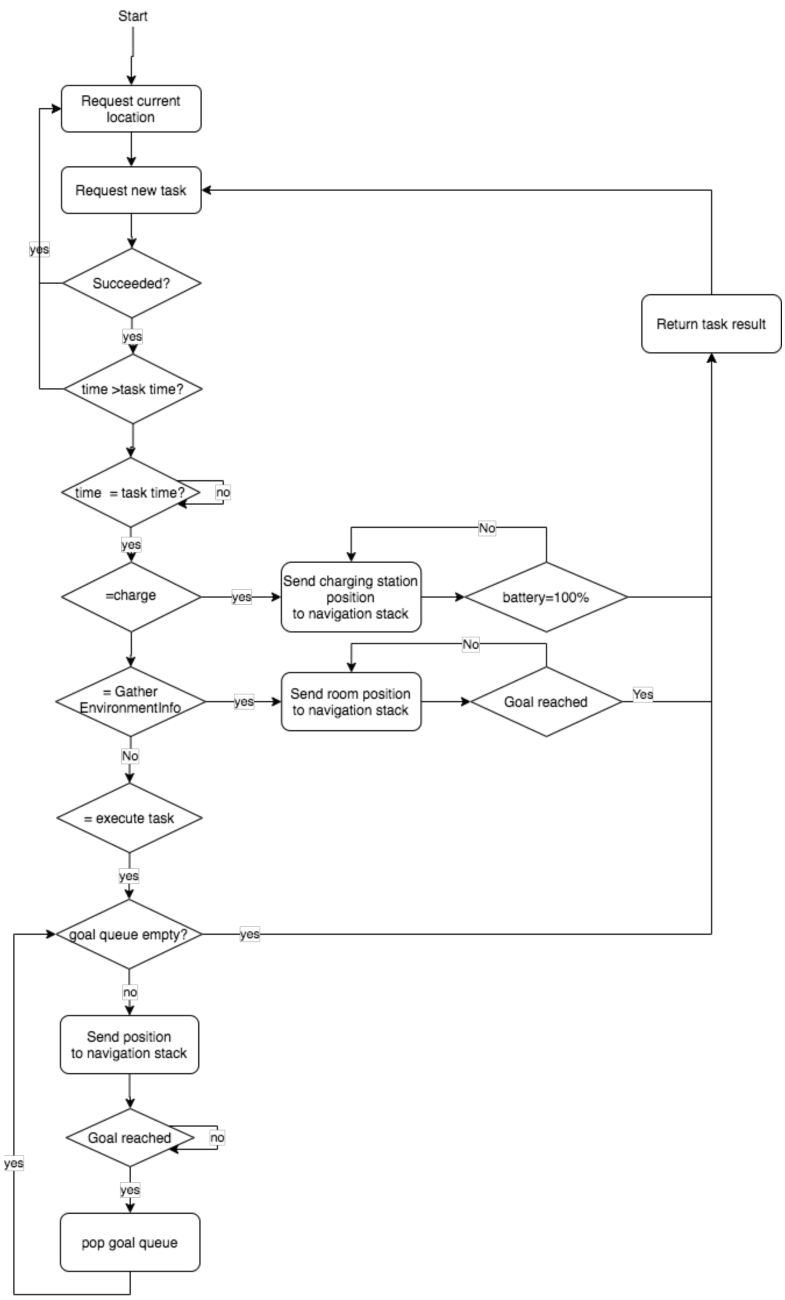

Figure 4.5: Robot Process Task

4.3.2 Robot

Robot Process Tasks When the task queue(Figure 3.2) in a robot is empty, robot requests a new task. If the robot gets a "charging task", it will move to the position of charging staion(Figure 3.5) and interact with charging station node (Section 4.3.3). When a robot gets an "execute task" which is a large task, it will move to all goals in order. When a robot gets a "gather enviroment information" task, it will move to the door's position. During task processing, the timer checks periodically the status of navigation stack. If any errors occurs, the robot send a "failed" result with description to the centralized pool. When task compltes without error, the robot will send "Succedded" result to the centralized pool.

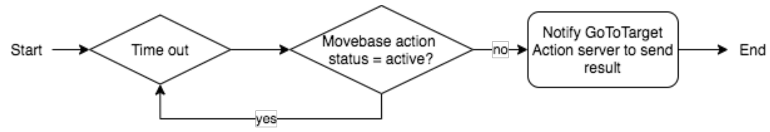

Figure 4.6: Robot Timer

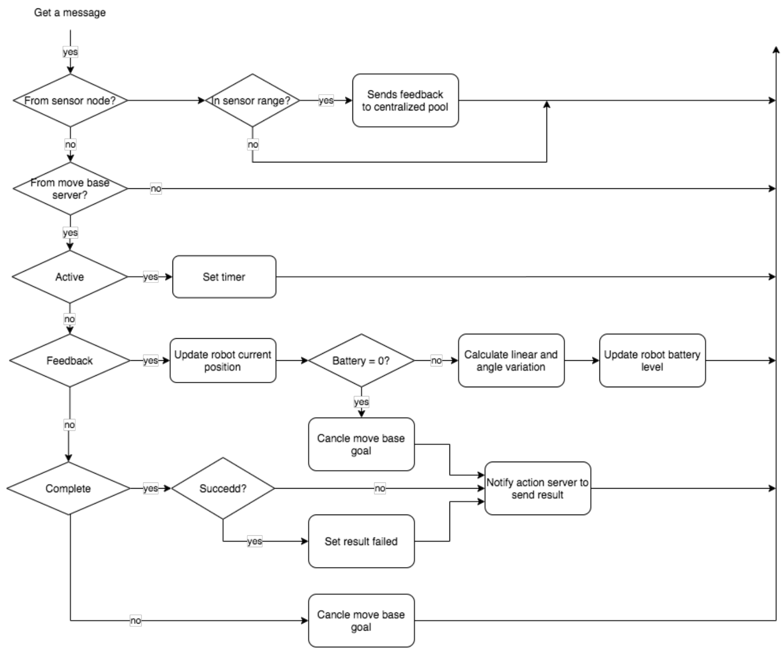

Figure 4.7: Robot Hanlde Message

Robot Handle Messages While a robot is processing a task, it listens to door sensors and forwards measurement result to the centralized pool. Besides sensor messages, it receives messages from move_base node. The details of robot message handling is shown in Figure 4.7.

4.3.3 Charging Station

The charging station consist of a charging station node and "charging station" table in database (Table 4.2). When a robot arrives the position of charging staion, it will start interacting with charging station node. Once the charging station receives robot's information(Figure 4.9), it changes its state to "charging"(Figure 4.8) and increase on its value on "battery level"colum and increase its value on "remaining time"colum in database(Figure 4.10). Once charging finished, its status will be set "charging finish". Once robot leaves charging station, its status will be set "free".

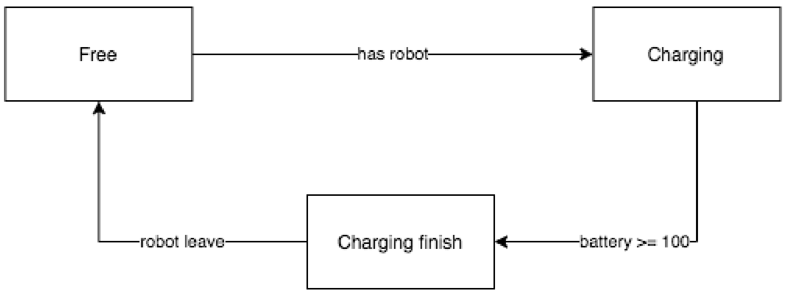

Figure 4.8: Charging Station State Machine

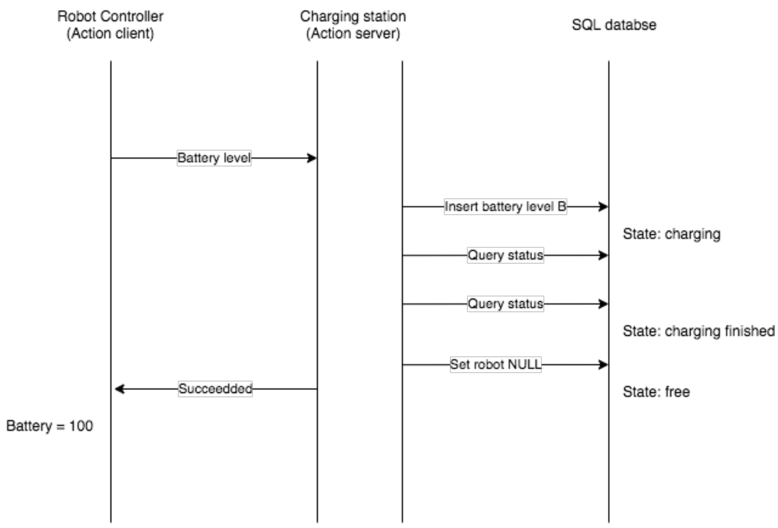

Figure 4.9: Charging Station Message

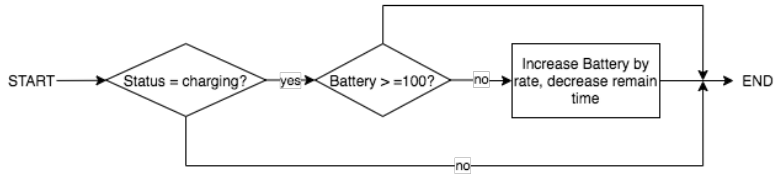

Figure 4.10: Charging Station Scheduled Charging Event

4.4 Unit Test

Unit tests test each part of the program that the individual parts are working correctly. In this program, Google test is used to test system module such as task allocation module, execution modules and so on.

Chapter 5

Evaluation

In this chapter you should describe the previous (if possible) and final experiments performed on the implementation.

Every single experiment should be explained individually, providing to the reader information about the meaning of the experiment, the expected (theoretical) results, the final results, the comparison between them and others (if possible) and the conclusions.

Each experiment should include a description, covering (when possible) the following information:

- Significant physical features (obstacles present on the environment, human presence, temperature, humidity, possible noise sources, computational speed of the machine, etc.)
- The precise location of the experiment (latitude and longitude, room number or citation to a description of the used laboratory).
- Sampling design (variable(s) measured, transformation performed to the data, samples collected, replication, comparative with a Ground Truth system, collecting data protocol).
- Analysis design (how the data are processed, statistical procedures used, statistical level to determine significance).

The provided information should be sufficient to allow other scientists to repeat your experiment in the same conditions. Thus, the use of standard and well-known equipment could only be represented by a simple sentence, but the non-standard equipment should be described in detail, citing the source (vendor) and most important characteristics.

To write it, try to use the third person when describing the experiments and results. Avoid to use first person. Past tense should be the dominant conjugation (the work is done and was performed in the past).

Note: Graphics represent really well data, use them! (Matlab or Octave could be useful for that).

Chapter 6

Discussion

The meaning of this paragraph is to interpret the results of the performed work. It will always connect the introduction, the postulated hypothesis and the results of the thesis/bachelor/master.

It should answer the following questions:

- Could your results answer your initial questions?
- Did your results agree with your initial hypothesis?
- Did you close your problem, or there are still things to be solved? If yes, what will you do to solve them?

Chapter 7

Acknowledgements

(This part is optional, and it could be completely excluded by deleting
\include {content/chapters/chapter7}
from the Firstname_Lastname_Diplom_Master_arbeit.tex file)

This paragraph could mention people or institutions that supported you to some extent with your work or friends and relatives that supported you during your study period.

Bibliography

[IGM97] IKEDA, MITSURU, SHOGO GO and RIICHIRO MIZOGUCHI: *Opportunistic Group Formation*. In *Proceedings of the Conference on Artificial Intelligence in Education (AI-ED)*, pages 167–174, Amsterdam, 1997. IOS Press.

[IGS17] IVAN GAVRAN, RUPAK MAJUMDAR and INDRANIL SAHA: *Antlab: a Multi-Robot Task Server*. ACM Transactions on Embedded Computing Systems, 9:39–58, 2017.

[MAdR02] MIDDLETON, S. E., H. ALANI and D. C. DE ROURE: *Exploiting Synergy between Ontologies and Recommender Systems*. In *Proceedings of the Semantic Web Workshop at the WWW Conference*, Honolulu, HI, 2002.

[Sal89] SALTON, GERARD: *Automatic text processing — the transformation, analysis, and retrieval of information by computer*. Addison-Wesley series in computer science. Addison-Wesley, 1989.

Versicherung an Eides Statt

German

Hiermit versichere ich, dass ich die vorliegende Bachelor(Master)arbeit selbständig - mit Ausnahme der Anleitung durch die Betreuer - verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, sowie Zitate kenntlich gemacht habe. Ich versichere dass ich alle entsprechenden Angaben nach bestem Wissen und Gewissen vorgenommen habe, dass sie der Wahrheit entspechen und dass ich nichts verschwiegen habe. Mir ist bekannt, dass eine falsche Versicherung an Eides Statt nach §156 und nach §163 Abs. 1 des Strafgesetzbuches mit Freiheitsstrafe oder Geldstrafe bestraft wird.

English

I hereby declare that I have written this Bachelor (Master) thesis - except for the guidance of the supervisor - independently, using no other than the specified sources and resources, and that all quotations have been indicated. I declare that I have reported to the best of my knowledge all the relevant information, that it is true and that I concealed nothing. I am aware that a false declaration will be punished according to §156 and §163 par. 1 of the criminal code with a prison sentence or a monetary penalty.

Essen, September 25, 2020
(Ort, Datum)

First Name Second Name