

# Mobile Robot Transportation in Laboratory Automation

Dissertation

zur

Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

der Fakultät für Informatik und Elektrotechnik

der Universität Rostock



Submitted by:

Hui Liu, born on 7<sup>st</sup>, February 1983 in Xianyou City, China

Rostock, Germany, 2013

Reviewers:

1. Reviewer:

Prof. Dr. -Ing. habil. Kerstin Thurow  
Institute of Automation, University of Rostock, Germany

2. Reviewer:

Prof. Dr. -Ing. Norbert Stoll  
Institute of Automation, University of Rostock, Germany

3. Reviewer:

Prof. Robin A. Felder, Ph.D.  
Medical Automation Research Center, University of Virginia, USA

Data of Submission: 22<sup>nd</sup>, March, 2013

Data of Defense: 13<sup>rd</sup>, September, 2013

## **Acknowledgement:**

This doctoral dissertation would not have been completed without the guidance and the help from the following several persons.

First and foremost, I would like to express my utmost gratitude to my supervisors Prof. Kerstin Thurow and Prof. Norbert Stoll for their elaborate guidance, continuous support and precious knowledge. Their ideas helped me in all the time of research and writing of this dissertation. I could not have imagined having a better advisor and mentor for my doctoral research. Also I really pretty appreciate the great opportunity they offered to continue my work here as a postdoctoral fellow.

Besides my supervisors, I would like to thank all my colleagues of the Institute of Automation and the Center for Life Science Automation (Celisca) at University of Rostock, Germany. Without their kinds of support and help to my work and daily life in Germany, this dissertation would not have been finished on time. Especially I would like to thank Dr. Steffen Junginger, Dr. Thomas Roddelkopf, Dr. Sebastian Neubert, Mr. Hans-Joachim Stiller, Mr. Lars Woinar and Mr. Peter Passow. Their profession and knowledge in automation technology gives me the deepest impression. Moreover, special thank goes to Mrs. Anett Ahrens for her kind assistance at my beginning in Rostock.

Additionally, I would like to take this opportunity to express my sincere gratitude to Dr Steffen Junginger and his family for all the kind help they have offered in Rostock. We are very fortunate to have friends like them.

Last but not least, I would like to dedicate this thesis to my beloved family: my wife Mrs. Yanfei Li, my little son Mr. Xing Liu, my father Mr. Jinzheng Liu and my mother Mrs. Meiqiong Zheng, for their great love, patience, unparalleled support and encouragement throughout my life!

# Contents

<b>Table of Contents</b>	<b>I</b>
<b>List of Figures</b>	<b>VI</b>
<b>List of Tables</b>	<b>VII</b>
<b>List of Algorithms</b>	<b>VII</b>
<b>List of Abbreviations</b>	<b>IX</b>
<b>Chapter 1 Introduction</b>	<b>1</b>
1.1 Background of this Dissertation	1
1.2 Literature Review of Robot Indoor Transportation	2
1.2.1 Supermarket/Shopping transportation	2
1.2.2 Hospital transportation	4
1.2.3 Factory transportation	5
1.2.4 Laboratory transportation/Delivery	6
1.2.5 Discussions	7
1.3 Aims of this Dissertation	8
<b>Chapter 2 Transportation Strategy</b>	<b>12</b>
2.1 H2O Mobile Robots	12
2.2 Control Strategy	13
<b>Chapter 3 Communication Network</b>	<b>18</b>
3.1 Introduction	18
3.2 Designed Network (Internal & External Architectures)	19
3.3 Wireless Bridge Integration	20
<b>Chapter 4 Laboratory Indoor Localization</b>	<b>22</b>
4.1 Introduction	22
4.1.1 Sensor-based Indoor Localization	22
4.1.2 Map-based Indoor Localization	24
4.1.3 Landmark-based Localization	24
4.1.4 Discussions	25
4.2 Hybrid Solution	26



4.2.1 Ceiling Landmark Based Global Localization.....	26
4.2.1.1 StarGazer Module .....	26
4.2.1.2 Precision Discussions of StarGazer.....	28
4.2.2 Floor Landmark Based Local Localization.....	31
4.2.2.1 Microsoft Kinect Sensor .....	32
4.2.2.2 Strategy of Local Localization.....	33
<b>Chapter 5 Collision Avoidance .....</b>	<b>36</b>
5.1 Introduction.....	36
5.2 Original Collision Avoidance of H20 Robots .....	37
5.2.1 Measuring Sensors .....	38
5.2.2 Avoidance Path Planning.....	38
5.2.3 Existent Problems.....	39
5.3 Improved Methods .....	39
5.4 Discussions .....	41
<b>Chapter 6 Map Theory Based P2P Path Planning .....</b>	<b>42</b>
6.1 Introduction.....	42
6.2 Hybrid Strategy .....	42
6.2.1 Floyd Algorithm .....	43
6.2.2 Dijkstra Algorithm .....	44
6.3 Discussions .....	45
<b>Chapter 7 Artificial Intelligence Based Patrol Path Planning .....</b>	<b>46</b>
7.1 Introduction.....	46
7.2 Genetic Algorithm.....	46
7.2.2 Chromosome coding.....	47
7.2.3 Generation of initial populations.....	48
7.2.4 Fitness function.....	48
7.2.5 Selection operation.....	48
7.2.6 Crossover operation .....	48
7.2.7 Mutation operation.....	49
7.3 Artificial Ant Colony Algorithm .....	50
7.4 Comparison of GA and AACA .....	53
7.5 Discussions .....	56

<b>Chapter 8</b>	<b>Multi-Robot Charging Management .....</b>	<b>57</b>
8.1	Introduction.....	57
8.2	Power Measurement System.....	57
8.3	Automated Charging Station.....	58
8.4	Optimization of Charging Station Installation .....	59
8.4.1	Artificial Immune Algorithm.....	59
8.4.2	Standard Computational Process .....	59
8.4.3	Improved Computational Steps.....	60
8.4.4	Results and Analysis.....	62
<b>Chapter 9</b>	<b>Software Development .....</b>	<b>66</b>
9.1	Development Environments.....	66
9.2	Software Architecture .....	66
9.2.1	Robot On-board Center (RBC).....	68
9.2.1.1	Communication .....	69
9.2.1.2	Data Acquisition.....	71
9.2.1.3	Motion Control .....	72
9.2.1.4	Power Control .....	72
9.2.1.5	Developed GUIs of RBC.....	73
9.2.2	Robot Arm Component (RAC).....	78
9.2.2.1	Communication .....	78
9.2.2.2	Arm Joint Mapping.....	79
9.2.3	Robot Remote Center (RRC).....	79
9.2.3.1	Communication .....	80
9.2.3.2	Data Acquisition.....	81
9.2.3.3	Path Planning Computation .....	83
9.2.3.4	PMS Command Mapping .....	84
9.2.3.5	Developed GUIs of RRC .....	87
9.3	Command Protocol .....	89
9.3.1	Commands between the RRC and the RBC .....	89
9.3.2	Commands between the PMS and the RRC .....	90
9.3.2.1	Transportation Command.....	92
9.3.2.2	Resource Command.....	93
9.4	Database Design.....	94
9.5	Conclusions.....	96

<b>Chapter 10 Experiments.....</b>	<b>97</b>
10.1 Robot Arm Movements.....	97
10.2 Robot Transportation .....	99
10.3 Charging Management.....	102
<b>Chapter 11 Conclusions and Outlook.....</b>	<b>104</b>
11.1 Conclusions.....	104
11.2 Outlook .....	106
<b>References .....</b>	<b>108</b>
<b>Appendixes .....</b>	<b>121</b>
<b>Declaration .....</b>	<b>131</b>
<b>Curriculum Vitae .....</b>	<b>132</b>
<b>Theses .....</b>	<b>137</b>
<b>Abstract .....</b>	<b>139</b>
<b>Zusammenfassung .....</b>	<b>141</b>

## List of Figures

Figure 1.1:	PMS (Future Lab) at Celisca, Germany	1
Figure 1.2:	Laboratory transportation	2
Figure 1.3:	Supermarket guiding and carting robots	3
Figure 1.4:	Remote shopping robotic application	3
Figure 1.5:	MKR robots for hospital transportation tasks	4
Figure 1.6:	OzTug robots for manufacturing environments	5
Figure 1.7:	Assistive mobile robots for factory transportation	5
Figure 1.8:	iCART transportation system	6
Figure 1.9:	Laboratory local transportation system	7
Figure 2.1:	H20 mobile robots	12
Figure 2.2:	System Strategy	13
Figure 2.3:	PMS presented tasks using explicit naming rule	14
Figure 2.4:	Dataflow among robots, RBCs and RRC	15
Figure 2.5:	Dataflow between RRC and PMS	16
Figure 2.6:	Dataflow between RBC and RAC	16
Figure 3.1:	Communication network	19
Figure 3.2:	Wireless bridge communication: (a) Connection structure of wireless bridges; and (b) Real cases of installing wireless bridges in H20 robots	21
Figure 4.1:	UBIRO robot and its floor RFID-based localization method	23
Figure 4.2:	Ultrasonic-based SESKON positioning system	23
Figure 4.3:	Computer vision-based robot localization using the indoor features	23
Figure 4.4:	Working mechanism of StarGazer module	27
Figure 4.5:	Coding format of ceiling landmarks	27
Figure 4.6:	A case of installing ceiling landmarks at Celisca, Germany	27
Figure 4.7:	StarGazer precision experiment in reference [1]	28
Figure 4.8:	Experimental environment with strong lighting	30
Figure 4.9:	Measured errors of direction	30
Figure 4.10:	Measured errors of position X	30
Figure 4.11:	Measured errors of position Y	31

Figure 4.12:	Hybrid method of indoor localization for laboratory transportation	31
Figure 4.13:	Graphical chart of local localization	33
Figure 4.14:	Kinect application in laboratory indoor local localization	34
Figure 4.15:	Main GUI of Kinect local localization function	34
Figure 4.16:	Flowchart of robot Kinect component	35
Figure 5.1:	Sensing modules in H20 mobile robots for the collision avoidance	38
Figure 5.2:	Mechanism of APF	39
Figure 5.3:	Vibrating phenomenon of APF	39
Figure 5.4:	Human obstacle shape & skeleton recognition	40
Figure 5.5:	Obstacle depth recognition	40
Figure 6.1:	A demonstrated map with five working waypoints	43
Figure 7.1:	Computational steps of the GA	47
Figure 7.2:	A robot path and its encoding format	47
Figure 7.3:	An example of the crossover operation	49
Figure 7.4:	A topology graph with nine waypoints	50
Figure 7.5:	An example of the mutation operation	50
Figure 7.6:	Working mechanism of AACA	51
Figure 7.7:	Computational steps of the AACA	51
Figure 7.8:	The random planning result without GA computation	53
Figure 7.9:	The planning result with GA computation	54
Figure 7.10:	The iterative process of the GA computation	54
Figure 7.11:	The planning result with AACA computation	55
Figure 7.12:	The iterative process of the AACA computation	55
Figure 8.1:	Architecture of robot charging management	58
Figure 8.2:	GUI of charging control in the RBC	58
Figure 8.3:	Automated charging station	58
Figure 8.4:	Charging station installed at Celisca, Germany	59
Figure 8.5:	AIA optimization	59
Figure 8.6:	Standard AIA computational process	60
Figure 8.7:	New AIA computational process	61
Figure 8.8:	Experimental position inputs	62
Figure 8.9:	Experimental task distribution inputs	62

Figure 8.10: Iterative steps in Case One	63
Figure 8.11: Calculated results in Case One	63
Figure 8.12: Experimental task distribution inputs	64
Figure 8.13: Iterative steps in Case Two	64
Figure 8.14: Calculated results in Case Two	65
Figure 9.1: Software structure	67
Figure 9.2: Software workflow	68
Figure 9.3: Architecture of TCP/IP sockets in the RRC	69
Figure 9.4: Designed methods for socket communication classes in the RBC	70
Figure 9.5: Workflow of sockets in the RBC	70
Figure 9.6: Architecture of robot hardware communication in the RRC	71
Figure 9.7: Acquisition function of motion modules	71
Figure 9.8: Acquisition function of ultrasonic sensors	71
Figure 9.9: Architecture of robot charging measurement	72
Figure 9.10: Acquisition function of laptop batteries	73
Figure 9.11: IP configuration GUI	73
Figure 9.12: XML for IP configuration	74
Figure 9.13: Sensor monitoring GUI	74
Figure 9.14: Waypoint definition GUI	75
Figure 9.15: Waypoint edition GUI	76
Figure 9.16: XML for waypoint definition	76
Figure 9.17: Socket monitoring GUI in the RRC to receive the robot sensor data and send the selected transportation path	77
Figure 9.18: Socket monitoring GUI in the RBC to send the robot sensor data and receive the given transportation path	77
Figure 9.19: Socket monitoring GUI in the RRC to send the chosen arm control commands	77
Figure 9.20: Socket monitoring GUI in the RAC to receive the given arm control commands	77
Figure 9.21: Architecture of TCP/IP sockets in the RRC	78
Figure 9.22: GUI of RAC	79
Figure 9.23: Architecture of TCP/IP sockets in the RRC	80
Figure 9.24: Programmed methods for socket communication classes in the RRC	81
Figure 9.25: Workflow of the path socket in the RRC	81
Figure 9.26: Framework of hybrid path planning	83

Figure 9.27:	PMS position parsing: (a) matching process; and (b) a demo of built robot path with a series of waypoints	85
Figure 9.28:	GUI of the PMS	85
Figure 9.29:	Robot connection GUI	87
Figure 9.30:	Sensor monitoring GUI	88
Figure 9.31:	Path planning GUI	89
Figure 9.32:	PMS handling GUI	89
Figure 9.33:	XML-based commands between the RRC and the RBC	90
Figure 9.34:	XML-based ' <i>GetWayPoint</i> ' command and its replied data	90
Figure 9.35:	XML-based commands between the PMS and the RRC	91
Figure 9.36:	Command workflow between the PMS and the RRC	91
Figure 9.37:	XML-based transportation command	92
Figure 9.38:	Format of transportation command	92
Figure 9.39:	XML-based transportation estimation time reply	93
Figure 9.40:	XML-based transportation error reply	93
Figure 9.41:	XML-based estimation time reply	93
Figure 9.42:	XML-based resource command	93
Figure 9.43:	XML-based resource reply	93
Figure 9.44:	Database Main Operating GUI	95
Figure 9.45:	Database Searching GUI	95
Figure 9.46:	Data Outputting GUI	96
Figure 9.47:	Statistical Calculation GUI	96
Figure 10.1:	Training arm module generating the XML-based control files for the robot arm movements	97
Figure 10.2:	The robot arm movements at different time step	99
Figure 10.3:	The robot transportation in a life science laboratory	101
Figure 10.4:	The robot executing the arm manipulation at the destination (position ⑥)	102
Figure 10.5:	All running robots monitored by the RRC 2D map	102
Figure 10.6:	The robot recharging in a life science laboratory	103
Figure A.1:	DrRobot H20 mobile robot	121
Figure A.2:	Cutaway view of Microsoft Kinect sensor	122
Figure A.3:	StarGazer module	123
Figure A.4:	Ceiling landmark for the StarGazer module	123

Figure B.1:	Output of StarGazer module	124
Figure C.1:	Experimental environment with strong lighting	128
Figure C.2:	Errors of direction measurement	128
Figure C.3:	Errors of position X measurement	128
Figure C.4:	Errors of position Y measurement	128
Figure C.5:	Experimental environment with strong lighting	129
Figure C.6:	Errors of direction measurement	129
Figure C.7:	Errors of position X measurement	129
Figure C.8:	Errors of position Y measurement	129
Figure C.9:	Experimental results of Kinect-based local localization	130



## List of Tables

Table 3.1: Network settings of five H20 robots in this application	21
Table 4.1: Specification of StarGazer Module	28
Table 6.1: Initial distance matrix of Floyd algorithm	44
Table 6.2: Final distance matrix of Floyd algorithm	44
Table 6.3: Final path matrix of Floyd algorithm	44
Table 7.1: Algorithm parameters of AACA training	54
Table 8.1: Algorithm parameters of AIA training	63
Table 9.1: Function API of motion distance control	72
Table 9.2: Data format of robot status measurement	82
Table 9.3: Data format of robot waypoint measurement	82
Table 9.4: Format of path command	82
Table 9.5: Function API of Floyd path planning	84
Table 9.6: Function API of parsing the transportation resource in the PMS command	86
Table 9.7: Function API of checking the robot positions for the transportation	86
Table A.1: Parameters of the DrRobot H20 mobile robot	121
Table A.2: Specifications of the Microsoft Kinect sensor	122
Table A.3: Specifications of the StarGazer module	123
Table B.1: Data format of the measured positions from the StarGazer module	124
Table B.2: Data format of the sensing data between the RBC and the RRC	125
Table B.3: Data format of the control commands between the RBC and the RRC	125
Table B.4: Data format of the control commands between the RBC and the RAC	126
Table B.5: Data format of the H20Robot database	126

## List of Algorithms

Algorithm 5.1: Global collision avoidance adopted in this study	41
Algorithm 6.1: An iterative loop of the Floyd algorithm	43

## List of Abbreviations

<b>API/APIs</b>	Application Program Interface/ Application Program Interfaces
<b>APF</b>	Artificial Potential Field
<b>AACA</b>	Artificial Ant Colony Algorithm
<b>Celisca</b>	Center for Life Science Automation
<b>COMS</b>	Complementary Metal Oxide Semiconductor
<b>DSP</b>	Digital Signal Processor
<b>DMC</b>	Digital Magnetic Compass
<b>DM</b>	Distance Matrix
<b>GUI</b>	Graphical User Interface
<b>GPRS</b>	General Packet Radio Service
<b>GSM</b>	Global System of Mobile communication
<b>GA</b>	Genetic Algorithm
<b>IP</b>	Internet Protocol
<b>JAUS</b>	Joint Architecture for Unmanned System
<b>MKR</b>	Muratec Keio Robot
<b>PMS</b>	Process Management System
<b>PTMCS</b>	Process Transportation Module Control System
<b>P2P</b>	Peer to peer
<b>PS</b>	Path Set
<b>PID</b>	Proportional Integral Derivative
<b>PC</b>	Personal Computer
<b>PM</b>	Path Matrix
<b>RRC</b>	Robot Remote Center/ Robot Remote Control Center
<b>RBC</b>	Robot On-board Center/ Robot On-board Control Center
<b>RAC</b>	Robot Arm Component/ Robot Arm Control Component
<b>RIN</b>	Robot Internal Network

<b>REN</b>	Robot External Network
<b>RFID</b>	Radio Frequency Identification
<b>RF</b>	Radio Frequency
<b>SLAM</b>	Simultaneous Localization and Mapping
<b>SDK</b>	Software Development Kit
<b>TCP</b>	Transmission Control Protocol
<b>US</b>	Unvisited Set
<b>Wi-Fi</b>	Wireless Fidelity
<b>XML</b>	Extensible Markup Language

# Chapter 1 Introduction

## 1.1 Background of this Dissertation

In recent years, the rapid development of automation has been considerably changing the working process of traditional laboratories. Advanced laboratory automation, including optimized robotic systems [1]–[6], automated analytical & measurement systems [7]–[10] and laboratory information management systems [11]–[13] has been employed widely and made an important role in kinds of life science, chemistry and biology laboratory environments. To build a fully-automated future laboratory, a system for the whole laboratory process control named Process Management System (PMS) has been developing by our research group at Center for Life Science Automation (Celisca) & Institute of Automation Technology (IAT), University of Rostock, as demonstrated in Fig. 1.1.

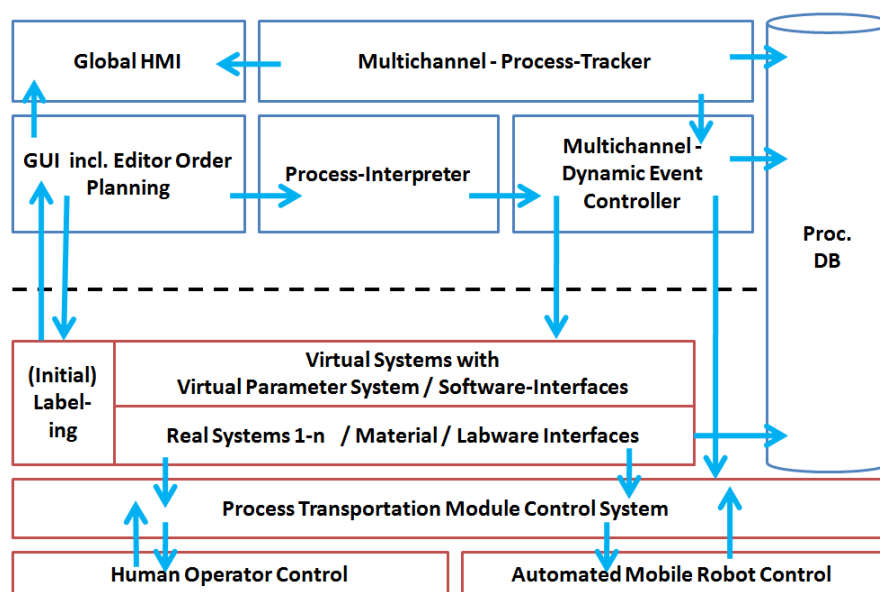


Figure 1.1: PMS (Future Lab) at Celisca, Germany

As shown in Fig. 1.1, there is a special Process Transportation Module Control System (PTMCS) at Celisca which is in charge of the transportation tasks based on the whole automated process of laboratory environments. As shown in Fig. 1.2, in the PTMCS two kinds of control options are expected: the human operation mode and the automated mobile robot mode. The former will notice the laboratory staff to do transportation tasks by sending messages to their holding mobile phones, and the

latter will use the mobile robots instead of human beings for the transportation. This dissertation is based on the development of an intelligent system for mobile robot transportation in laboratory environments. This dissertation will not differentiate the transportation task commands from the PMS and the PTMCS. Both of them are higher in laboratory automation than the mobile robot-based transportation sub-system. In the following chapters of this dissertation, the description of the PMS will be adopted to replace that of the PTMCS.

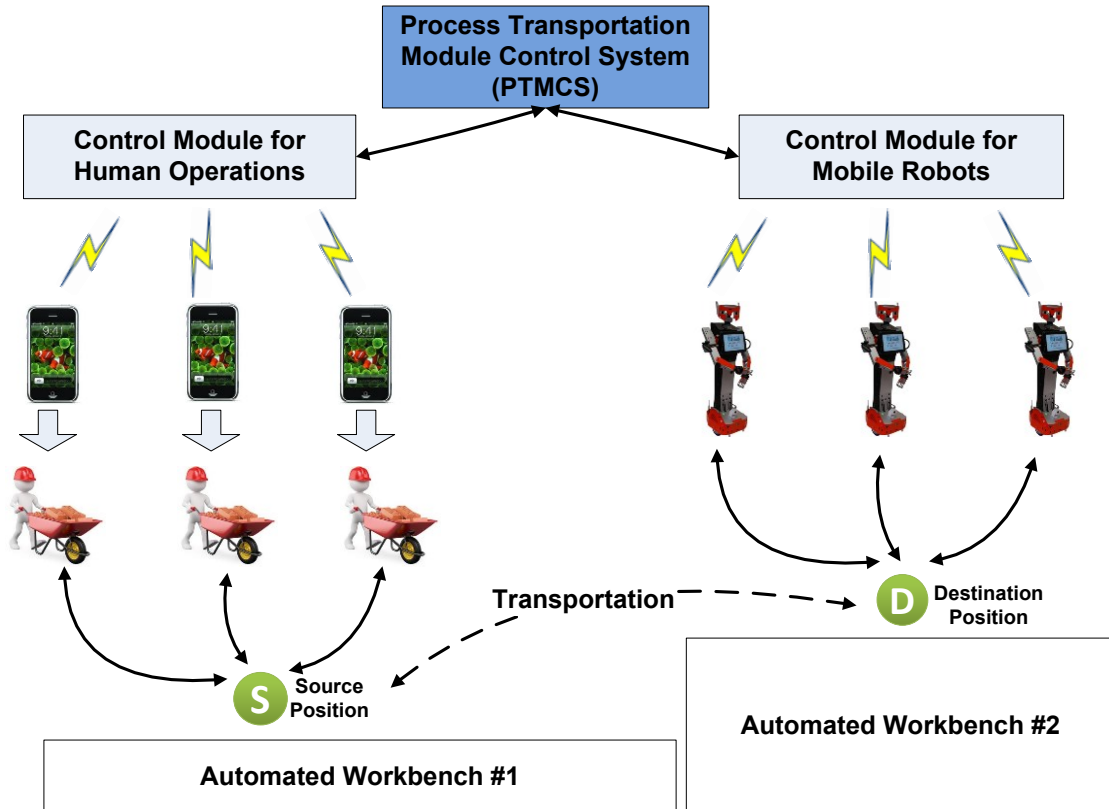


Figure 1.2: Laboratory transportation

## 1.2 Literature Review of Robot Indoor Transportation

This section provides a short review on literatures on the mobile robot based indoor transportation.

### 1.2.1 Supermarket/Shopping transportation

*N. Matshuira et al.* presented a shopping support system using kinds of environmental cameras and mobile robots as shown in Fig. 1.3 [14]. In this robotic transportation system, there were three comprised sub-systems designed: guidance robotic system, cart robot system and environmental image monitoring system. When the customers went shopping in supermarkets installed this shopping support system,

they used a membership card to drive any available guidance robot, then the guidance mobile robot leded the customers to different positions for shopping and the cart mobile robot followed the customers to carry heavy productions.

**T. Tomizawa *et al.*** developed a remote shopping robotic system for supermarkets as displayed in Fig. 1.4 [15]. The customers can stay at home to remotely communicate with this system to buy fresh foods including fruits and vegetables in some special supermarkets with different textures, shapes and weights. In the system, a series of technologies for the robot grasping and transporting were considered and solved: autonomous navigation, environment modeling & identifying, object manipulation and remote communication.

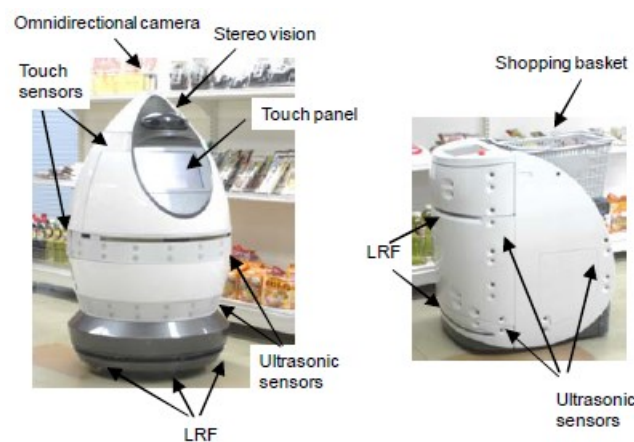


Figure 1.3: Supermarket guiding and carting robots [14]

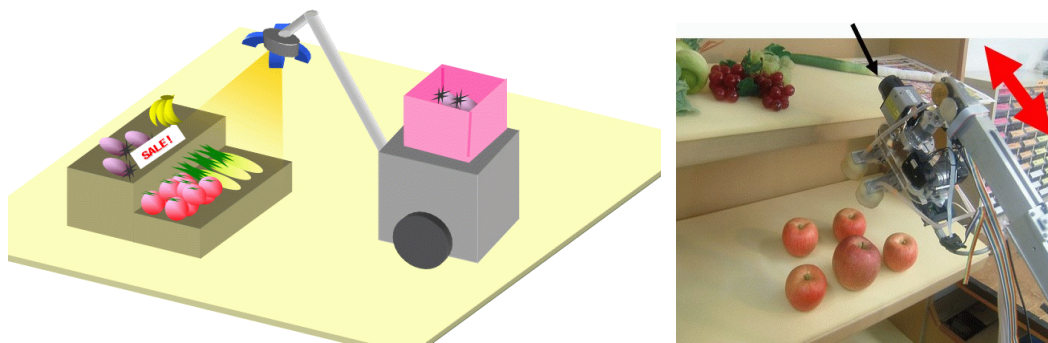


Figure 1.4: Remote shopping robotic application [15]

**S. Nishimura *et al.*** studied the robotizing daily items for an autonomous carrying system [16]. The authors in this paper designed a person following shopping cart robot, which was comprised of a shopping cart, a group of driving motion module, monitoring cameras and an on-board controller. Based on the computer vision using the installed cameras around the shopping malls, this kind of mobile robots accurately

followed the customers to realize the carrying transportation.

More examples of mobile robot transportation in supermarkets and shopping malls are demonstrated in references [17]–[19].

### 1.2.2 Hospital transportation

*A.G. Özkil et al.* discussed a service robot transportation system for hospitals [20]. In that project, they used the robot system for patients' food distribution, waste collection, etc. Since the system would be applied for hospitals, some important aspects for the hospital environments (such as patient avoidance) had been considered in their paper.

*M. Takahashi et al.* proposed a mobile robot for transport applications in hospital areas using a safe human detection algorithm [21]. As given in Fig. 1.5, in the application the authors developed a new autonomous mobile robot MKR, which used a wagon truck to transfer luggage, specimens and medical materials.

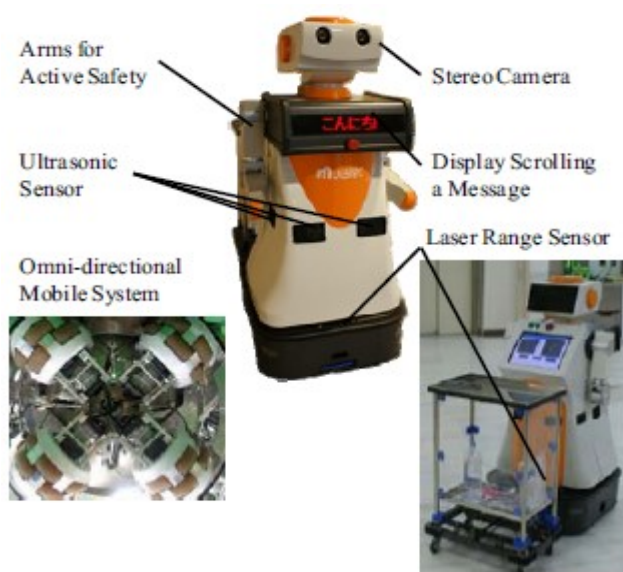


Figure 1.5: MKR robots for hospital transportation tasks [21]

*W.F. Fung et al.* developed a mobile robot for routine delivery tasks in hospitals [22]. In their application, the fluorescent lamps on the ceiling at the corridors were used as natural landmarks to localize the transportation positions and orientations. The function of collision avoidance was also included in the robot. Experimental results made at a hospital in Hong Kong showed the system was effective and suitable for hospital environments.

More applications of mobile robot transportation in hospitals can be found in



references [23]–[26].

### 1.2.3 Factory transportation

**B. Horan et al.** proposed a transportation system using OzTug mobile robots for manufacturing environments as given in Fig. 1.6 [27]. The OzTug robots used in the application could move loads ranging up to 2000 kg. In their system, a vision based controller had been designed to track the working paths and three kinds of robot-load configurations had been presented for multi-robot transportation for large loads.

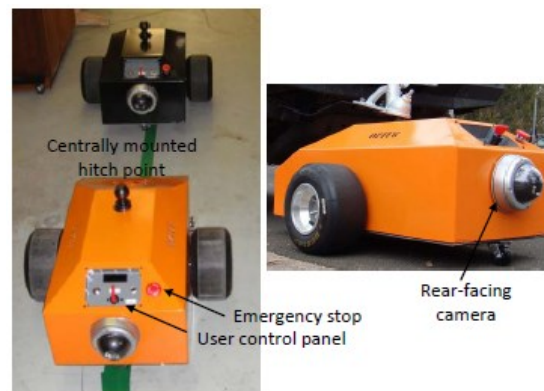


Figure 1.6: OzTug robots for manufacturing environments [27]

**J.W. Kang et al.** designed a series of assistive mobile robots to help the disabled in a factory environment [28]. Three different versions of mobile robots were provided. From the second version a wheel-based Omni-directional drive mechanism was utilized to provide the transportation in small spaces. As demonstrated in Fig. 1.7, based on this kind of mobile robots a simple transportation task made by the disabled in a factory was possible.



Figure 1.7: Assistive mobile robots for factory transportation [28]

*M. Endo et al.* presented a car transportation system named iCART using multi mobile robots in manufacturing factories [29]. In their paper, the detailed control architecture and several experiments were provided to illustrate the validity of the system. In their system there were two mobile robots adopted to do transportation. A trajectory of the car transportation was sent to one of the mobile robots, and the other robot tracked the trajectory by calculating the interactive forces between them. This creative method could cope with any size of cars and most kinds of car transportation tasks (including parking, two away service and ferry).

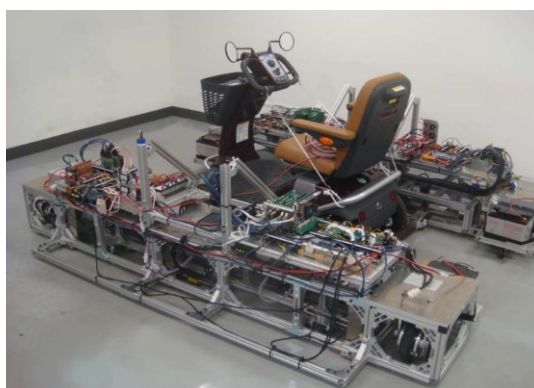


Figure 1.8: iCART transportation system [29]

More cases of mobile robot transportation in factories can be seen in references [30]–[32].

#### 1.2.4 Laboratory transportation/Delivery

*M. Wojtczyk et al.* studied the computer vision based human robot interface for robotic walkthroughs in a biotechnology laboratory as given in Fig. 1.9 [33]. They employed a mobile robot to move kinds of biotechnology facilities in an experimental platform. As described in the paper, the authors would not like to develop a robot system for the distributed laboratory transportation but focused on the image identification of the laboratory facilities.

*P. Najmabadi et al.* designed a scalable robotic-based laboratory automation system for medium-sized biotechnology laboratories [2]. In the system, a new configuration named “Tower-based configuration” was presented to control the instrumentation for the sub transportation processes including shaking and incubation. The configuration worked through two arms mounted on a common cylindrical based. An experiment provided in the paper showed that the proposed method and system for laboratory desk local transportation were effective.

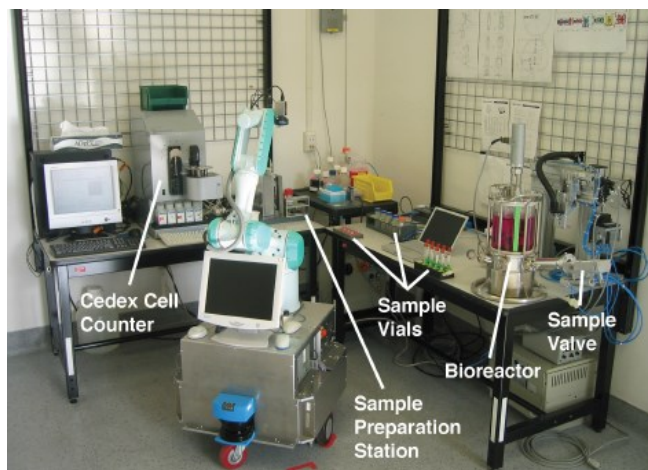


Figure 1.9: Laboratory local transportation system [33]

*W. Gecks et al.* proposed a robotics-based QCX/RoboLab system for laboratory automation [6]. This system was employed to bring the samples from nine automatic sampling points to the distributed analytical equipment. Also kinds of manually operated laboratory devices were connected to this robotic system based on a standard network.

*L.A. Marks et al.* developed a microprocessor-based robotic system for control of fluid connections in the cardiac catheterization laboratory [34]. As explained in their paper, the system had various functions of planning catheter flushing, configuring the users, calibrating the pressure transducers, etc. A verifying case provided in their paper showed that the developed system decreased the laboratory technicians' working intensity obviously.

*K. Thurow et al.* presented a robotic parallel liquid handling system for a life science laboratory [35]. In their study, a fast method to determine the liquid levels in micro plates was proposed, an optical measurement of the liquid level based on CCD cameras using special computational algorithms was found and an automated robotic system to handle with kinds of sample transportation was integrated. A series of experiments given in their paper showed that the system was effective.

More applications of robot transportation (including mobile robot transportation and desk robot delivery) in laboratories can be found in references [36]–[40].

### 1.2.5 Discussions

From the literatures demonstrated in Sections 1.2.1-1.2.4, it can be seen that:

(a) In recent decades, the topic of robotic indoor transportation has come into attention and many interesting studies have been presented. Based on those successful applications, we can find that using the mobile robots to do transportation tasks in indoor distributed environments is a right way. Especially in current more and more advanced mobile robots with low costs have been provided in the markets.

(b) In those developed mobile robot indoor transportation systems, there is a series of technical issues needing to be considered for an expected working environment, which includes transportation strategy, multi-robot communication, robot indoor localization, robot collision avoidance, transportation path planning and recharging management, system API integration with other existent automated systems, etc.

(c) Most of the mobile robot transportation systems in the upper literatures are relatively independent. They present the tasks of mobile transportation themselves and do not need to be integrated with other higher indoor process systems which control the whole automation in the working environments. In this dissertation, the expected transportation system integrates with the laboratory PMS to cooperate with other automated systems. The transportation system will control mobile robots to move experimental facilities among different automated ‘islands’.

(d) Compared to robot transportation in normal environments (supermarkets/factories), the transportation in laboratories are more complicated. The reasons can be explained as follows: in laboratories there are always high-precision experimental facilities and complex experimental crafts, the proposed mobile robot transportation is expected to cooperate seamlessly with the existent systems. It requires that the transportation system should not affect the working process of the existent facilities (systems) and can communicate with them to finish a complex task. For example, the robot transportation system should know what time is most suitable for an exhausted mobile robot to go recharging by considering the whole laboratory automated schedule.

### **1.3 Aims of this Dissertation**

The aim of this study is to present an intelligent mobile robot transportation system in laboratory environments integrating with the other laboratory automated process. The works required to develop this system have been divided to two theses: laboratory transportation (including motion and arm movements) and arm high-precision manipulation. The first one is to develop a transparent system including the server and the client sides to control all adopted mobile robots in laboratories for kinds of flexible transportation tasks. The second one is to design a module of robot arm manipulation for transportation of high-precision requirements, which will be

integrated in the transportation system as an embedded component. The detailed relationship between those two theses can be found in Chapter 9.

This dissertation is toward to the first topic considering all necessary APIs (Application Program Interfaces) for multi-robot transportation in laboratory environments. The specific aims of this dissertation are presented as follows:

- **Designing a common system framework for organizing the laboratory mobile robot transportation.** This framework is required to suit for any size of laboratories and can connect any kind of mobile robots after simple updating. The final purpose of the framework is to be transparent. It means a new hardware module or a new mobile robot could be integrated into the transportation system conveniently as a special ‘sensor’. In the framework, a series of questions will be answered: i.e. what is the dataflow inside the hardware and software components of the adopted mobile robots to organize a laboratory transportation task? How to guarantee the robots run normally even when they lose the connections to a remote controller? How to control and coordinate kinds of robots from different manufacturers in the same distributed environments?
- **Presenting a robot indoor localization method for a huge laboratory environment.** In this method, at least the following several points have to be solved. Firstly, since the proposed transportation system is expected for a huge laboratory environment, the method should be low-cost and can be extended fast. Secondly, in this study the working areas of the mobile robots are possible to be changed frequently, so the method should allow the users to redefine the indoor positioning maps conveniently. Thirdly, the method should not affect the performance of other laboratory automation modules. Fourthly, the method should include the function to calculate the relative positions of different robots. Since mobile robots may have different positioning mechanisms inside, it is important to map all of their private coordinates to the sole global origins of the laboratories.
- **Proposing a communication network for the data transmission.** Since in the future more and more mobile robots will be integrated into those transportation works, a stable, expandable and long-distance network for connecting the robots is desired. For this purpose, several strategies are under consideration: (a) proposing a good network to guarantee the stable and expandable performance of the data transmission; (b) strengthening the signal capacity of a network to connect/cover the long-distance running mobile

robots; and (c) reducing the unnecessary transmitted data between the control server side and the robot on-board sides to decrease the possibilities of data transmission jam.

- **Providing a function of path planning which can handle various kinds of mobile robot transportation.** The path planning is always crucial in the robot indoor transportation. Actually there are two kinds of robot path planning styles, which could be executed in laboratory environments. One is to define every adopted mobile robot several fixed paths. When a transportation task between two defined positions is required in the laboratory automation process, the mobile robot, which can recognize this path, will be asked to finish it. If no mobile robot understands the path, the transportation cannot be carried out. The other one is a smart mode, which needs the function of flexible and intelligent path planning computation. In this case, all robots are defined the corresponding working maps with a number of recognized positions. The real executed path for a given transportation task is dynamic based on the real-time path planning calculation. Compared to the transportation without path planning, the one with path planning is much more effective, because every robot is allowed to execute a presented transportation task.
- **Realizing charging management for the mobile robot transportation.** In this application, kinds of mobile robots need on-board batteries for their transportation tasks, so the issue of the robot recharging is important. In the charging management, at least three works are required: (a) *a system (module) to measure the real-time powers status from robots*: this module will monitor all running mobile robots' power status then control them to go recharging if their batteries are close to be exhausted. This module will be integrated together with other important modules in both of the server/client software (e.g., motion control module); (b) *a fully automated charging station*: This station should have the functions of auto-docking and auto-recharging. When a mobile robot reaches an expected position where the charging station is installed, the robot can do re-charging automatically; and (c) *there is also an optimized question*: how to configure the charging stations scientifically so the robots can go charging most smoothly and effectively.
- **Last but not least, developing operating software platform and amount of embedded APIs of the transportation system.** The operating platform of the software should be flexible and friendly. The work of the software programming will occupy a big part of this dissertation, because there are lots of works on data measurement, system integration and algorithm development

needing to be executed in the operation platform. The highest requirement of this software is to let laboratory staffs operate the presented mobile robot system comfortably and to let the system be integrated with other laboratory automation systems seamlessly.

## Chapter 2 Transportation Strategy

An effective control strategy is crucial in any kind of system development. Generally, it is presented not only based on the system functions and the adopted software & hardware platforms but also by the consideration of some special requirements (such as, the working environments). As given in Chapter 1, the developed transportation system in this dissertation is expected to be run in laboratories, so there are some special requirements from the laboratory environments which the system has to meet as follows: (a) convenient expandability; (b) fast integration; (c) robust performance; and (d) economic cost. Based on considering those four aspects, a new control strategy to design the mobile robot based transportation system is proposed in this chapter.

This Chapter is organized as follows: Section 2.1 introduces one kind of mobile robots named H20 robots adopted in this study; and Section 2.2 proposes the control strategy of the transportation system.

### 2.1 H20 Mobile Robots

One kind of mobile robots named H20 from Canadian DrRobot company has been adopted in this study to demonstrate the efficiency of the proposed transportation system, as shown in Fig. 2.1. From Fig. 2.1, it can be seen that the H20 robots meet the hardware requirements of this application, which provide all necessary robotic modules, including on-board control PCs, arm servo modules, motors, localization modules, batteries, cameras, ultrasonic sensors, etc. The specifications of the H20 robots can be found in Appendix A.1.

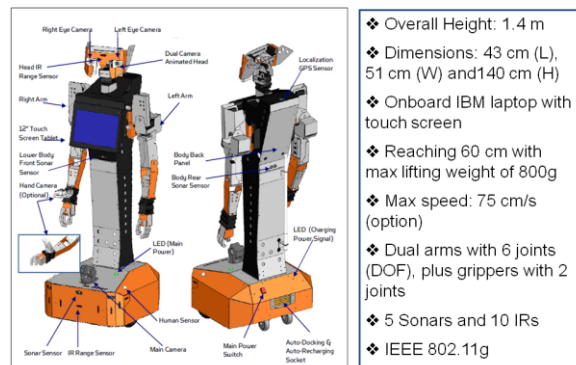


Figure 2.1: H20 mobile robots [41]



## 2.2 Control Strategy

Based on the aims of this study and the hardware properties of the adopted H20 robots, the control strategy of the proposed system is presented in Fig. 2.2.

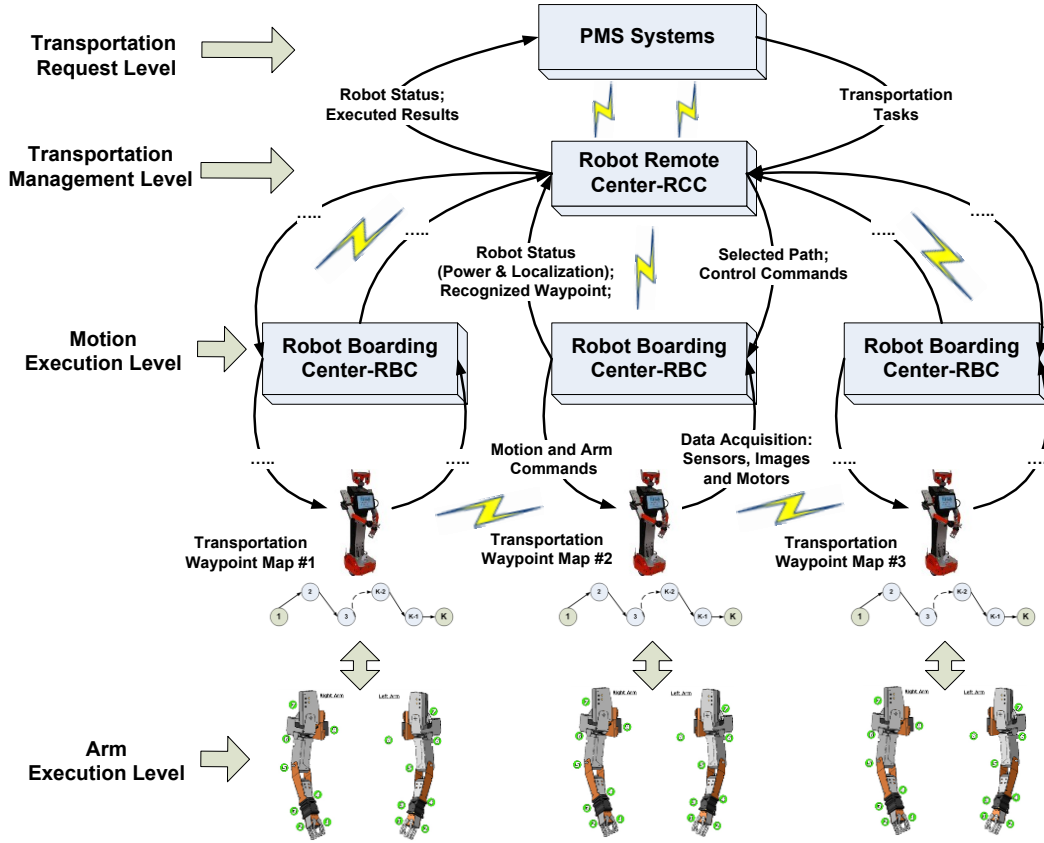


Figure 2.2: System Strategy

As shown in Fig. 2.2, there are four control levels in this system control strategy: transportation request level, transportation level, motion execution level and arm execution level. Accordingly, there are four control centers in the corresponding levels: PMS in the first level, Robot Remote Center (RCC) in the second level, Robot On-board Centers (RBC) in the third level and Robot Arm Component (RAC) in the fourth level. Their relationships can be described as follows:

- (a) The level of transportation request (in PMS) is the highest in the system, which is in charge of proposing kinds of transportation tasks when they are needed in the automated laboratory process. In life science laboratories, it considers the whole automated process based on the crafts of life science engineering. In this strategy, it does not need to care about which robot will be selected to do the presented transportation task and also how to execute it. The PMS will propose a task using an explicit data format as: *Move "Object A"*

from “S-Lab, Device A, Position 1” to “D-Lab, Device B, Position 2” (see Fig. 2.3).

- (b) The level of transportation management (in RRC) is designed to remotely control all the connected mobile robots. When a transportation task is coming from the PMS, the RRC will handle with it using the following steps: parsing the received PMS command, selecting the best mobile robot for the task, doing path planning for the task based on the source-destination condition of the task, sending the chosen robot the best transportation path and monitoring the status of the transportation executing process.

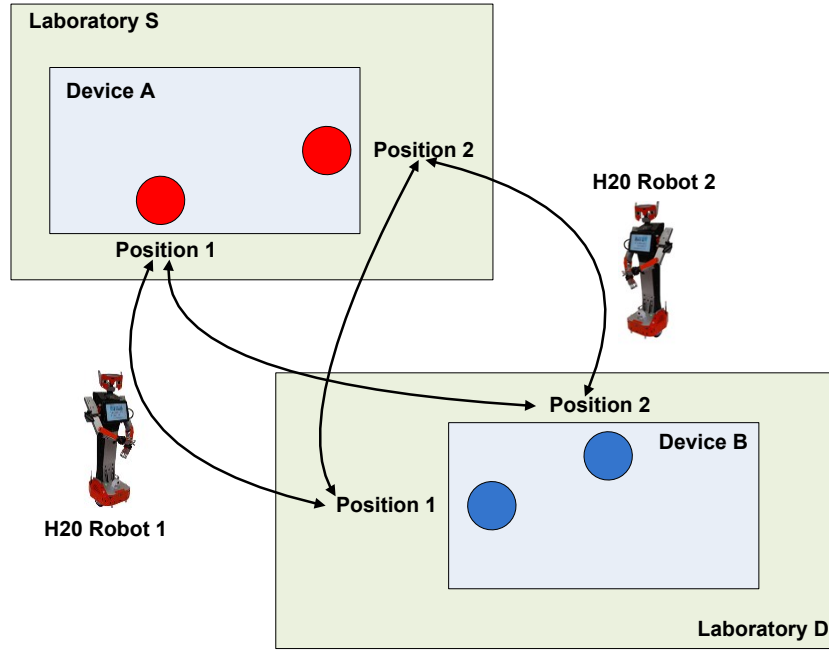


Figure 2.3: PMS presented tasks using explicit naming rule

- (c) The level of transportation motion execution (in RBC) is developed to actually implement the motion parts of the transportation tasks. Although the robot inside hardware modules could be controlled directly by a remote server-RRC, in this study we prefer to adopt the RRC-RBC distributed architecture to organize a transportation task. The advantage of defining the RBCs can be explained as follows: The RBC can reduce the transmitted data between the robots and the RRC to avoid the phenomenon of data transmission jam, only the key data (robot positions and powers) will be sent from the robots to the RRC for path planning computation. On the other hand, without those RBCs, the RRC has to communicate with the hardware modules inside robots directly. When the number of running robots is huge, the control process will become complicated and the communication jam will be serious.

Additionally, the RBC can continue an executing transportation task even sometimes the wireless communication between the RRC and the RBC is broken, because the RBC is installed in the robot controlling all the robotic hardware modules independently.

- (d) The level of transportation arm execution (in RAC) is designed to cope with the arm movements of the transportation tasks. The RAC will be embedded in the RBC platforms as a component. The working relationships between the RBC and the RAC can be described as follows: when a robot reaches the defined starting position of a transportation task, the RBC will establish a TCP/IP socket to its relevant RAC then send it a driving command to move up the arm hardware. Similarly, when a robot arrives at the defined ending position of a transportation task, the RBC will send its connected RAC to move down the arm hardware. All of the arm moving trajectories will be described in a series of XML (Extensible Markup Language)-based control files in advance, which will be stored in both of the RBC and the RAC. So the RBC only needs to tell the RAC the kinds of expected movements and the names of selected control files. Once a RAC receives those two parameters from its connecting RBC, it will do the transportation arm execution at once.

Based on this strategy, the data flows in the system can be explained as Figures 2.4, 2.5 and 2.6. The workflows in the system can be found in Chapter 9 in details.

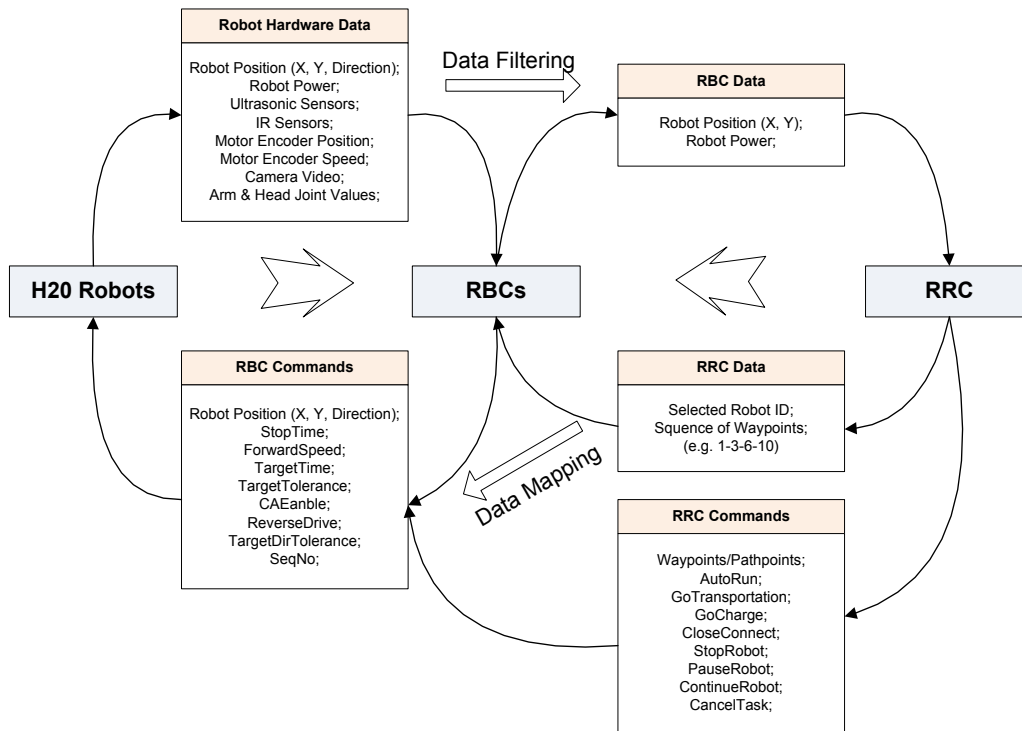


Figure 2.4: Dataflow among robots, RBCs and RRC

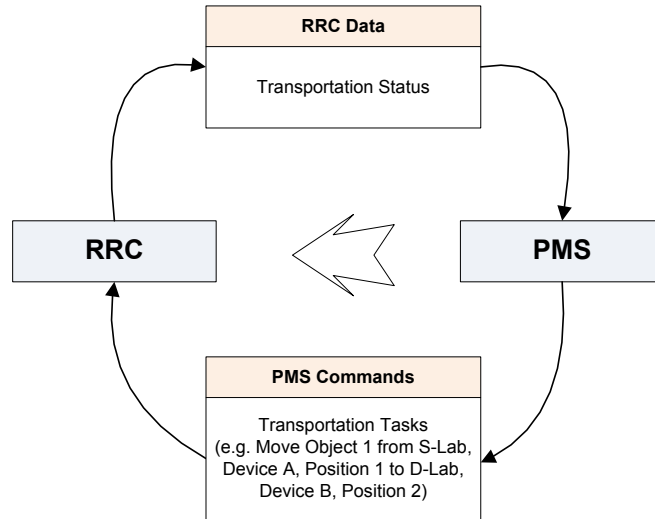


Figure 2.5: Dataflow between RRC and PMS

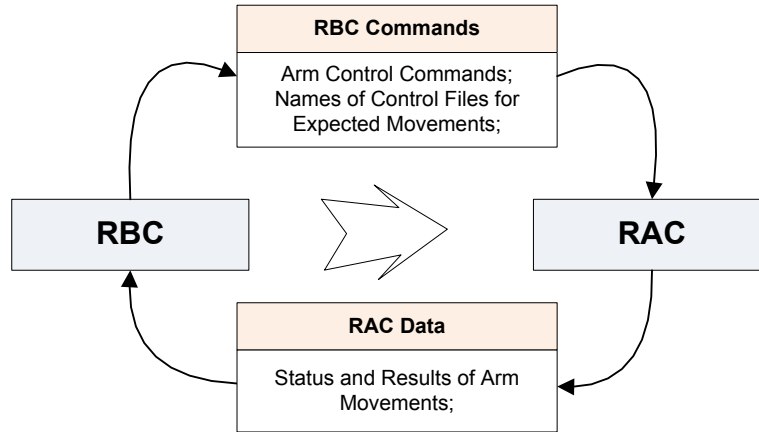


Figure 2.6: Dataflow between RBC and RAC

As shown in Fig. 2.4, the dataflow among the robots, the RBCs and the RRC can be explained as follows: (a) the hardware data of robots will be measured by their corresponding RBCs, which include the parameters of robot indoor positions, powers, ultrasonic sensors, IR sensors, motor encoder information, camera video and arm & head joint coding values (*see Fig. 2.4, Tag Robot Hardware Data*). Those data will be displayed in the sensor monitoring GUI (Graphical User Interface) in every RBC automatically if the connections between the robots and their RBCs are available; (b) the original hardware data from the robots will be filtered and sent to the RRC. The filtered data only retain the parameters of robot positions and powers (*see Fig. 2.4, Tag RBC Data*) which will be adopted to do transportation path planning and robot selections by the RRC; (c) when the RRC is distributed a transportation task by a PMS, the RRC will actively send relevant commands (*see Fig. 2.4, Tag RRC Commands → Waypoints*) to get the required filtered data from all the RBCs; (d) after necessary computational steps, the RRC will select the most suitable H2O robot and

send the RBC of the chosen robot the best path with a sequence of waypoints (*see Fig. 2.4, Tag RRC Data*); (e) when a RBC receives a path from a RRC, it means this RBC is asked to control the corresponding mobile robot to execute the transportation task. The RBC will map the received path numbers to the parameters and commands, which the robots can understand. After the fast mapping process, the RBC will send the results to the robot hardware modules for transportation movements (*see Fig. 2.4, Tag RBC Commands*); (f) when the mobile robots reach the starting and destination positions of the transportation tasks, they will drive the right arm movements (such as the actions of catching up and putting down) using the prepared XML based arm control files.

The dataflow between the RRC and the PMS in Fig. 2.5 can be specified as follows: (a) the PMS proposes a transportation task when it is required by the laboratory process. The PMS commands include the transportation parameters of laboratory rooms, starting devices/positions and ending devices/positions. As explained in Chapter 1, the PMS does not consider how the presented transportation tasks will be executed (*see Fig. 2.5, Tag PMS Commands*); and (b) when the RRC receives a distributed task, it should continue to report the latest status to the PMS during the task implementation (*see Fig. 2.5, Tag RRC Data*).

Similarly, the dataflow between the RBC and the RAC in Fig. 2.6 can be illustrated as follows: (a) the RBC distributes an arm task (*see Fig. 2.6, Tag RBC Commands*) to the RAC when it is required to finish a transportation process.; and (b) when the RAC receives a given task, it will drive the arm servo hardware and report the latest status to the RBC during the arm movement implementation (*see Fig. 2.6, Tag RAC Data*).

## Chapter 3 Communication Network

### 3.1 Introduction

Since H20 mobile robots will be controlled to work in distributed laboratories, the wireless communication is necessary although the wire method has good transmission performance. In engineering, the popular wireless methods include IEEE 802.11 networks [42]–[47], GPRS (General Packet Radio Service)/GSM (Global System of Mobile Communication) [48]–[50] and Bluetooth [51]–[56].

*A. Leelasantitham et al.* made an experiment study of performance for an automatic robot using IEEE 802.11 wireless communication [42]. Their results showed the delay time and the maximum distance between a robot installed with a normal wireless router and a remote control PC were approximately at 100ms and 40m, respectively.

*G. Enriquez. et al.* presented a Wi-Fi (Wireless Fidelity) sensor based network for a human-aware navigation robot [44], [45]. The robot used a vision-based tracking system for human awareness. To realize a stable communication, an IEEE 802.15.4 wireless module (MICA2 MPR2400) was installed inside the robot.

*M.H. Li et al.* found a practical approach to design a guard robot alarm system using GPRS [48]. A wireless GPRS module was embedded in the robot to communicate with other robots and remote monitoring hosts. Once robot sensors detected the dangerous conditions, the GPRS module would send kinds of alarm signals to the remote control centers and the other robots.

*S. Sagioglu et al.* proposed a monitoring mobile robot vehicle system using Bluetooth communication adapter [51]. The mobile robot system was used do real-time image monitoring. The Bluetooth adapter was only employed to send higher commands, and the monitoring cameras were working independently. This system can realize monitoring, remote control, parameter adjusting and reprogramming through internet exclusively with a standard web browser without the need of any additional software.

After analyzing the upper mentioned representative works in wireless communication, we can conclude that:

- The GPRS/GSM method is one kind of global communication networks. It can be used in global inter-city communication, and its transmission data can be

conveniently sent to a standard mobile phone. This is the main reason why it is popular in remote automated applications. However it has to pay the transmission fees and its transmission bandwidth is not enough for possibly huge data cases (e.g. real-time image monitoring);

- The Bluetooth method is one of the standard wireless personal area networks having lots of advantages, such as no data transmission cost, low energy consumption and good expandability. But due to the similar problems of limited data width ( $< 1\text{Mbit/s}$ ) and short coving distance ( $< 10$  meters), it is hard to be applied in real-time monitoring & control systems. At the same time the Bluetooth way is easy to be affected by other existing Bluetooth devices in laboratories due to its open 2.4GHz frequency;
- Compared to the GPRS/GSM and the Bluetooth, the wireless IEEE 802.11 network is more suitable for laboratory robots. It has width data transmission channel ( $> 72.2\text{Mbit/s}$ ), and it can be easily expanded to unlimited numbers of PCs and devices with necessary network revisions. In this study the IEEE 802.11g is adopted because it uses the same OFDM technology as employed in 802.11a, plus backward compatibility with 802.11b [57], [58].

### 3.2 Designed Network (Internal & External Architectures)

As the proposed strategy described in Section 2.2, the transportation system adopts the standard S/C architecture which is comprised of a server RRC and a number of on-board client RBCs. For this structure a communication network is proposed as displayed in Fig. 3.1.

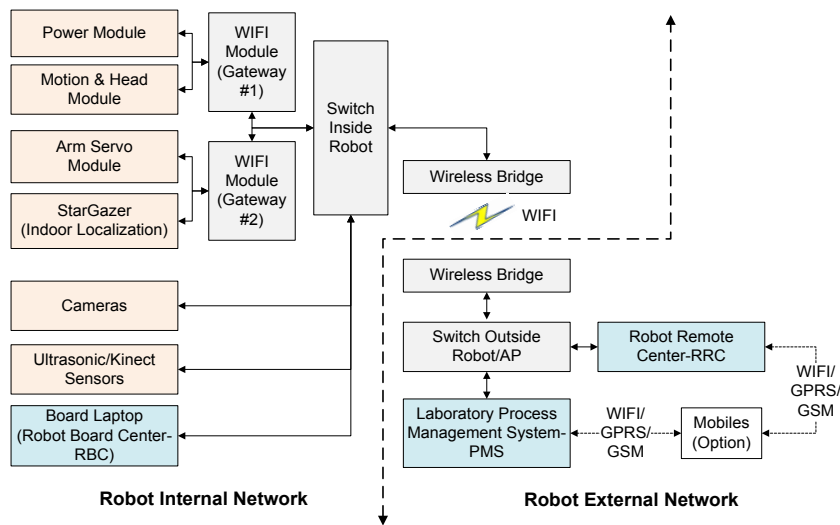


Figure 3.1: Communication network

As displayed in Fig. 3.1, the presented network is divided into a Robot Internal Network (RIN) and a Robot External Network (REN). The RIN is employed to connect all hardware modules inside robots, including power, motion & head, arm servo, indoor localization, cameras and ultrasonic/Kinect sensors. The RBC runs in the RIN to measure all hardware data. The REN is designed for the remote communication among the PMS, the RRC and the RBCs. In the RIN the cable connection is adopted to guarantee the fast transmission performance, and in the REN the Wi-Fi connection is utilized to cover the distributed RINs. Based on this presented network, the system is flexible. A new mobile robot or control module can be integrated fast after defining an IP address. All the data communication among different modules in the network is based on TCP/IP protocol. As an option, the system in this network can also be remotely controlled by a mobile phone through GPRS/GSM communication.

### 3.3 Wireless Bridge Integration

Although the performance of the normal IEEE 802.11 network is always satisfactory in engineering [57], [59], [60], however when the function of real-time image monitoring is activated in systems it will also possibly meet the phenomenon of transmission delay and congestion [61]–[63].

In the study, the function of real-time monitoring is an option. As shown in Fig. 2.1, a H20 mobile robot has two eye cameras and a main camera. So when a big number of H20 mobile robots are running and all eye cameras of them are being activated at the same time, the image transmission data will be considerable. Additionally, when a transportation task is being executed among separate buildings, a long-distance stable communication is required. To meet those cases (huge-data and long-distance) in laboratory environments, one kind of 2.4 GHz wireless marine bridges is supplied in this study between the movable RBCs (Robot Internal Network) and the RRC (Robot External Network) to strengthen the signal intensity of the IEEE 802.11g Wi-Fi section as shown in Fig. 3.2. The detailed parameters of this kind of bridges can be found in reference [64]. A case of IP configuration for this presented network is demonstrated in Table. 3.1.



(a)



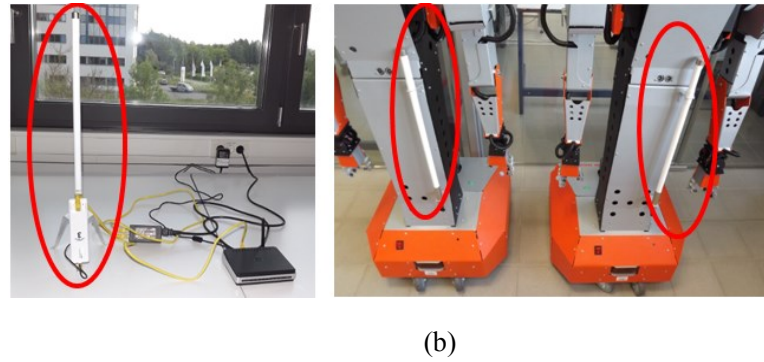


Figure 3.2: Wireless bridge communication: (a) Connection structure of wireless bridges; and (b) Real cases of installing wireless bridges in H20 robots

Table 3.1: Network settings of five H20 robots in this application

Modules	Communication IP addresses and Ports				
	Robot One	Robot Two	Robot Three	Robot Four	Robot Five
Wireless bridge on robot	192.168.0.20	192.168.0.22	192.168.0.24	192.168.0.26	192.168.0.28
Wireless bridge off robot	192.168.0.21	192.168.0.23	192.168.0.25	192.168.0.27	192.168.0.29
Power Module	192.168.0.95: 1001	192.168.0.85: 1001	192.168.0.75: 1001	192.168.0.65: 1001	192.168.0.55: 1001
Motion & Head Module	192.168.0.95: 1002	192.168.0.85: 1002	192.168.0.75: 1002	192.168.0.65: 1002	192.168.0.55: 1002
Arm Servo Module	192.168.0.94: 1001	192.168.0.84: 1001	192.168.0.74: 1001	192.168.0.64: 1001	192.168.0.54: 1001
Indoor GPS Module	192.168.0.94: 1002	192.168.0.84: 1002	192.168.0.74: 1002	192.168.0.64: 1002	192.168.0.54: 1002
Robot on-board laptop	192.168.0.109: 7030	192.168.0.108: 7030	192.168.0.107: 7030	192.168.0.106: 7030	192.168.0.105: 7030
Remote Server			192.168.0.100: 7040		
Main Camera	192.168.0.99: 8081	192.168.0.89: 8081	192.168.0.79: 8081	192.168.0.69: 8081	192.168.0.59: 8081
Right Eye Camera	192.168.0.98: 8082	192.168.0.88: 8082	192.168.0.78: 8082	192.168.0.68: 8082	192.168.0.58: 8082
Left Eye Camera	192.168.0.97: 8083	192.168.0.87: 8083	192.168.0.77: 8083	192.168.0.67: 8083	192.168.0.57: 8083

## Chapter 4 Laboratory Indoor Localization

### 4.1 Introduction

Robot indoor navigation is one of the most important works in mobile robotics, which is to plan a route a mobile robot can move from one location to another location without getting lost or colliding with other humans or objects in indoor environments [65]. In the issue of robot indoor navigation there are two main works: indoor localization and indoor path planning. The first one means a robot can know its orientations in an indoor environment [66], and the second one is to present an available path for transportation tasks. This chapter focuses on the first issue, and the second one will be discussed in Chapter 5.

In recent years, kinds of applications on robot indoor navigation have been proposed, which can be classified into three aspects: sensor-based methods, model map-based methods and landmark-based methods.

#### 4.1.1 Sensor-based Indoor Localization

This classification works using kinds of measuring sensors, such as radio frequency (RF) [67]–[73], ultrasonic & infrared sensors [74]–[79], cameras [66], [80]–[83].

*S.H. Park et al.* [67] proposed a novel Radio Frequency Identification (RFID)-based method for robot indoor navigation as shown in Fig. 4.1. In their study, a RFID module was installed in a mobile robot named UBIRO and a number of read-time passive IC tags were set in the floor to identify the RFID signals from the robot to estimate its positions. Their experimental results showed this RFID method was able to estimate the robot's location and orientation accurately without using external sensors. More cases about the RFID-based robot indoor localization can be found in references [68]–[73].

*A. Yazici et al.* [74] developed an ultrasonic-based positioning system called SESKON for indoor robotic applications as demonstrated in Fig. 4.2. Different to other indoor positioning systems using multi kinds of positioning sensors, the SESKON only adopted ultrasonic sensors. From their experiments at Eskişehir Osmangazi University based on P3-DX mobile robots described in their paper, it can be seen that the error of SESKON system was less than 30mm, which can be used for

high-precision applications. More cases about the ultrasonic sensor-based robot indoor localization can be found in [75]–[77], [79], [84].



Figure 4.1: UBIRO robot and its floor RFID-based localization method [67]



Figure 4.2: Ultrasonic-based SESKON positioning system [74]

*S.Y. Hwang et al.* [80] presented an indoor vision-based robot localization based on using the indoor features of corners, lamps and doors as given in Fig. 4.3. They considered that: although a monocular camera looking up toward the ceiling can provide a low-cost solution for robot localization, it can't achieve dependable identification due to a lack of reliable visual features on the ceiling. So they used the features of corners, lamps and doors simultaneously to replace the features on the ceiling. Their results showed that the computer vision method based on multi ceiling image features can solve the problem that line features can be easily misidentified due to nearby edges. An experiment was provided to prove the efficiency of the proposed method. More cases about computer version-based robot indoor localization can be found in [66], [81]–[83].

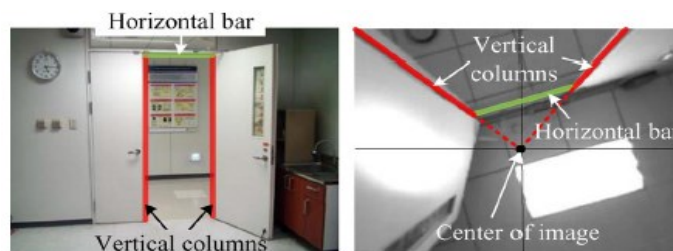


Figure 4.3: Computer vision-based robot localization using the indoor features [80]

#### 4.1.2 Map-based Indoor Localization

Apart from the classification based on the kinds of sensors, there is another classification using modeling maps as follows:

- **Map-Based Localization:** It depends on user-created geometric models or topological maps of the expected environments. Map-Based information consists of providing a robot with CAD models. The most representative work is usually mentioned to the “occupancy map” as introduced in [85]. More cases about CAD and graph maps can be found in [86]–[88].
- **Map-Building-Based Localization:** It includes two aspects: firstly, kind of recognizing sensors are used to construct the geometric or topological models of the navigation environments; and secondly the built models are utilized to do the indoor localization. In this method different kinds of landmarks are always installed in the indoor environments to assist the positioning process. Once some marked landmarks are identified, the robots will use their embedded maps to identify their positions by matching the observations against the expectations. The first case of the map-building navigation is a Stanford Cart as given in [89], [90]. More cases about robot using robot sensors to finish inside maps can be found in [80], [91]–[94].
- **Mapless Localization:** It utilizes no explicit properties about the environments, but prefers to recognize some objects in the environment or to track those objects by visual observations. Actually this kind of methods can be regarded that the robot is to build a real-time map based on recognizing the features such as desks, doorways, etc. Some examples about this kind can be found in [95], [96].

#### 4.1.3 Landmark-based Localization

This classification means the robots directly adopt the defined landmarks to do indoor positioning. A good landmark should meet the following requirements: **(a) Detection aspect:** the robots are able to determine a landmark in their views quickly; **(b) Recognition aspect:** the robots can differentiate the landmarks; **(c) Localization aspect:** once a landmark is identified, the robot can localize itself relative to the environment if the landmark position is known; **(d) Quality and identification aspect:** the number of landmarks should be enough and every landmark has a unique ID.

The progress of the indoor localizing methods using landmarks can be explained as follows: (a) In the first phase, the natural landmarks are welcomed because of their easiness. The natural landmark is generally selected in the natural scene considering their geometrical or photometrical features [97]–[99] [100]. The method of natural

landmark sometimes is not reliable enough in complex or dark environments [93]; (b) In the second phase, to improve the performance of the landmark localization, some scientists started to use artificial landmarks instead of natural landmarks. **G. Yang et al.** [101] designed a color artificial landmark with symmetric rectangles and seven-part numbers. **J. Park et al.** [102] proposed an efficient artificial visual landmark for an automated guided robot vehicle to locate into the parking position. **G.H. Li et al.** [103] presented a new artificial landmark based on image segmentation technology and projective method with directional features for mobile robot localization; (c) In recent years due to the fast development of the image capture technology, some researchers begin to use natural ceiling lighting landmarks to guide their mobile robots. They considered that using the natural ceiling lighting landmarks (e.g. ceiling lighting lamps) are the most direct and expandable way for the indoor navigation. The ceiling lighting landmarks can be possibly detected by an advanced up camera of robot through distinguishing light differences between lamps and ceiling surface. Also the ceiling lights do not need any special installations [80], [92], [104], [105]. **S.Y. Hwang et al.** [80] used the lighting lamps as ceiling landmarks to increase the identification ability of a monocular vision-based Simultaneous Localization and Mapping (SLAM) system to observe the features in various environments. **T. Kim et al.** [106] proposed a vision based artificial landmark for indoor navigation of a skid-steering mobile robot.

#### 4.1.4 Discussions

Based on comparison of different methods of indoor navigation mentioned in Section 4.1.3, it can be seen that: (a) the indoor navigation in small areas can have good performance, but it is still difficult for a large area; (b) every method both has advantages and disadvantages. Taking the method of image vision for example, it can provide a robot with high resolution information about its environments, but in some complicated cases its time performance is not satisfactory, and it is also hard to work rightly in dark environments [66]. The methods of ultrasound & infrared sensors are stable, but they are hard to be used in the multi-robot and large place navigation because multiple sensors will affect each other. It is also not easy to update their configurations once the system development is ready [107]–[109]. The Radio Frequency (RF) method, it can work well in a longer distance compared to another methods (e.g. infrared method), but it is easily affected by another communication devices using the similar frequencies and the cost of hardware configuration is always expensive [68], [70], [72].

Based on the current applications of landmark methods, it can be seen that: (a) landmark method is good to scale and maintain after installation whereas many

existing techniques fail to provide this important feature; (b) compared natural landmarks with artificial landmarks, the artificial landmarks can work with a large number of independent identification signals (e.g. ID numbers), which is precious for large laboratory automation; (c) compared active landmarks with passive landmarks, the passive landmarks do not need electrical powers and additional wiring systems; and (d) comparing passive artificial landmarks with lighting landmarks, the former are better. The passive artificial landmarks do not need electrical powers and any electrical installation while the lighting landmarks are easily affected by strong lights.

## 4.2 Hybrid Solution

In this study, the method of robot localization is not only required to provide accurate indoor positions and directions, but also is needed for large laboratory environments. In addition, the extension and developing cost of the adopted method are considered. Based on the discussions given in Section 4.1.4, plus considering the requirements in laboratory environments, a hybrid solution is proposed. The hybrid localization method includes two modes:

- ❖ ***Ceiling landmark based global localization:*** This mode is proposed to guide the mobile robots to reach a series of expected positions in laboratory environments based on the prepared ceiling landmarks. It emphasizes on the extendable performance of the localization.
- ❖ ***Floor landmark based local localization:*** This mode is presented to correct the mobile robots in some high-precision required positions in laboratories. It focuses on the accurate performance of the indoor localization.

### 4.2.1 Ceiling Landmark Based Global Localization

#### 4.2.1.1 StarGazer Module

For the laboratory global localization, an IR module named StarGazer and a number of ceiling passive artificial landmarks both from Korea Hagisonic company are adopted (*see Appendix A.3*). As shown in Fig. 4.4, the StarGazer module consists of a Complementary Metal Oxide Semiconductor (CMOS) Infrared Radio (IR) camera, an infrared LED array and a Digital Signal Processor (DSP), and it can analyze IR image which is reflected from a passive landmark with a unique ID. The passive artificial landmark is made of a retro coated film material which can strongly reflect the infrared rays coming from the StarGazer module [110], [111].

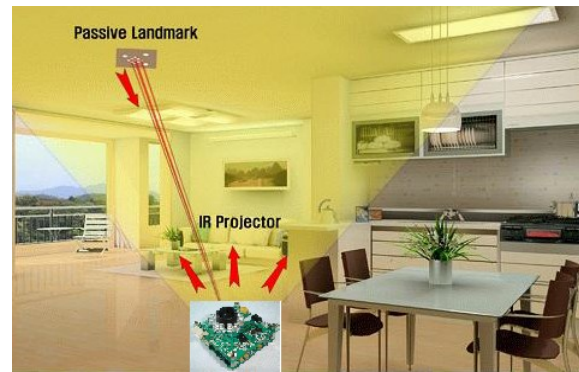


Figure 4.4: Working mechanism of StarGazer module [111]

As displayed in Fig. 4.5, the maximum ID of a  $4 \times 4$  landmark is 4095. The distance between two ceiling landmarks in real applications can be longer than 2m, so the  $4 \times 4$  landmark is normally enough for a huge laboratory. If in some cases the maximum ID of a  $4 \times 4$  landmark is not enough, it is also possible to extend the maximum ID by adding more lines or rows of points. An installing case of this kind of landmarks at our center is demonstrated in Fig. 4.6.

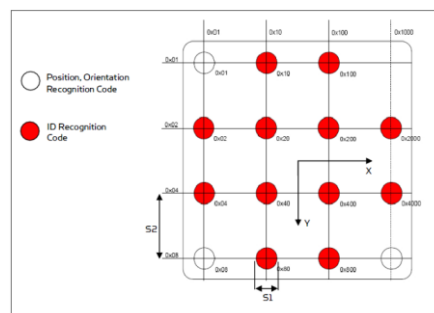


Figure 4.5: Coding format of ceiling landmarks [111]

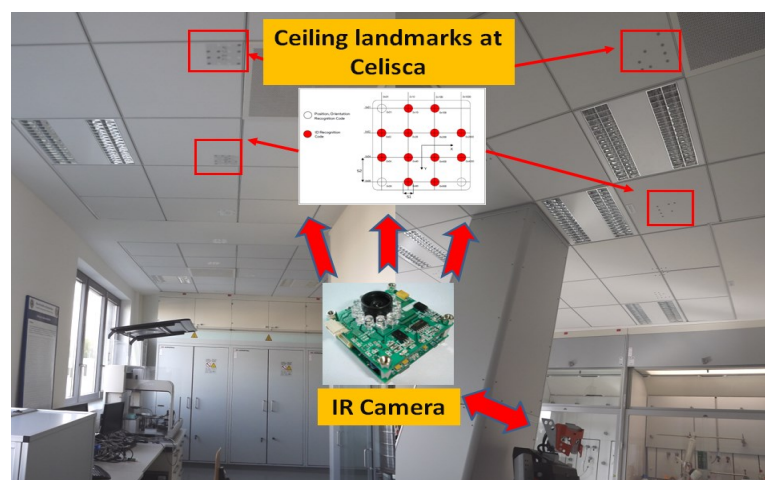


Figure 4.6: A case of installing ceiling landmarks at Celisca, Germany

The StarGazer module provides a series of commands for engineers to control the measuring process of the indoor positioning data. The key parameters of the StarGazer module are abstracted in Table 4.1. The final outputting data format of the StarGazer module include X position, Y position and Direction (*see Appendix B.1*).

Table 4.1: Specification of StarGazer Module [111]

API Interface	Communication Protocol	Sampling Time	Repetitive Precision	Heading Angle Resolution
UART (TTL 3.3V)	ASCII Code	20 Times/Sec	2 cm	1.0 degree

#### 4.2.1.2 Precision Discussions of StarGazer

There is a published paper [1] focusing on the precision of the StarGazer in laboratories. In their works, an experimental mobile robot using the StarGazer module was provided to estimate the positioning precision. From the results given in their paper, we can see that the performance of the StarGazer module was satisfactory (the maximum error of the positioning was less than 2cm). However, the reference [1] did not provide any information about the ceiling conditions of their experimental laboratory. In our opinions, the StarGazer module could be possibly affected by reflective ceilings and strong lighting conditions because it adopts the IR mapping mechanism. So it is necessary to verify the performance of the StarGazer module under strong/dark lighting conditions or reflective ceiling materials.

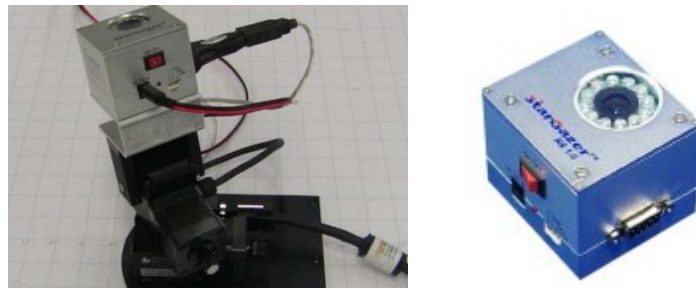


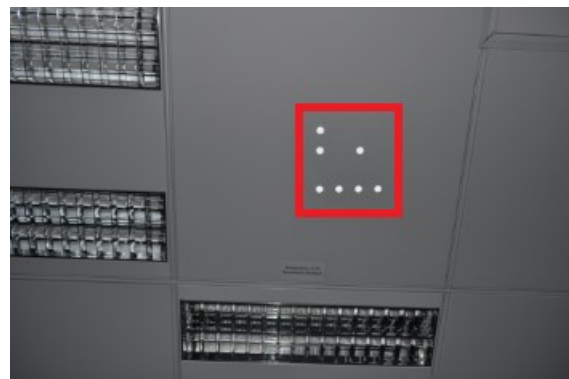
Figure 4.7: StarGazer precision experiment in reference [1]

To investigate the performance of the StarGazer module in unexpected lighting environments (e.g. strong-lighting, dark-lighting and reflective ceiling condition), a group of experiments has been done in our laboratory (Celisca, Germany). In those experiments, two kinds of issues have been focused: How is the performance of the StarGazer module in laboratory environments with dark or strongly bright lighting conditions and how is its performance in the indoor environments with reflective

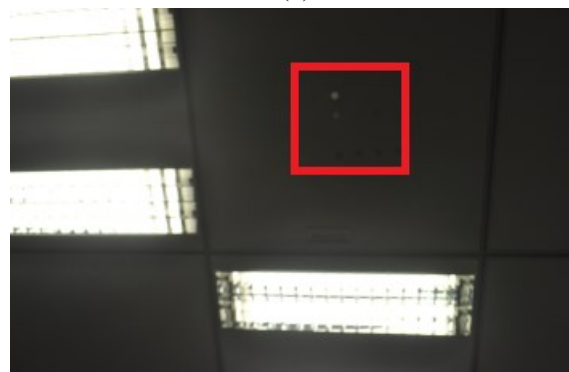


material ceilings? The measuring steps of those experiments are executed as follows: (a) a third-part IR sensor is installed in a H20 mobile robot; (b) put the chosen H20 mobile robot at a position under a special lighting condition and use the RRC of the robot to build a simple transportation path which will guides the robot to arrive at the prepared position repeatedly; (c) paste a grid paper on the prepared position and calibrate the robot's IR sensor to match the central point of the grid paper; and (d) use the RRC to control the robot to repeat the predefined experimental path and record the position error every time when the robot reaches the special position. In the experiment of light affecting measurement, several landmarks were installed in an indoor area inside a life science laboratory with strong ceiling lights as shown in Fig. 4.8. As displayed in Fig. 4.8, those installed landmarks were located closely to the bright lights. In this experiment, the adopted robot was controlled to reach a prepared position where was directly below those bright lights. During the robot movements the lights were switched several times.

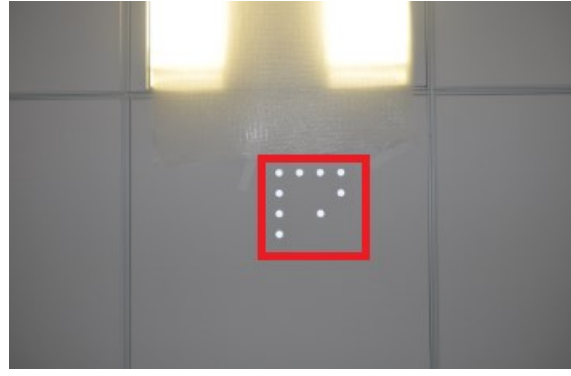
The experimental results are demonstrated in Figures 4.9, 4.10 and 4.11. From Figures 4.9, 4.10 and 4.11, it can be found that: (a) every time when the strong lights were turned on the localization performance of the StarGazer module become worse; (b) this phenomenon displayed when the strong lights were covered can prove that the performance of the StarGazer module is affected by the strong lighting conditions.



(a)



(b)



(c)

Figure 4.8: Experimental environment with strong lighting: (a) a ceiling landmark can be recognized rightly without strong lighting condition; (b) the ceiling landmark can not be recognized with strong lighting condition; and (c) the ceiling landmark can be recognized rightly again when the strong lighting condition has been covered by a paper.

Additionally two experiments using the reflective ceilings and the dark lighting condition are also provided in this study. The results showed that the performance of the StarGazer module under the dark environment was satisfactory but that under the reflective ceilings was unsatisfactory. The detailed results can be found in Appendixes C.1 and C.2.

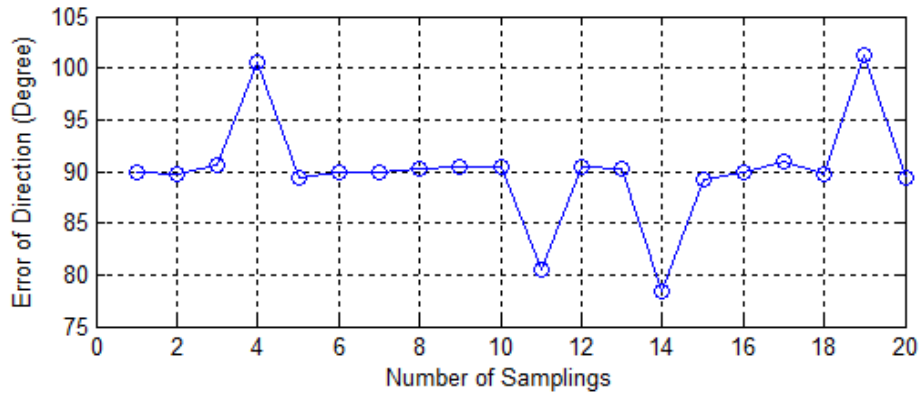


Figure 4.9: Measured errors of direction

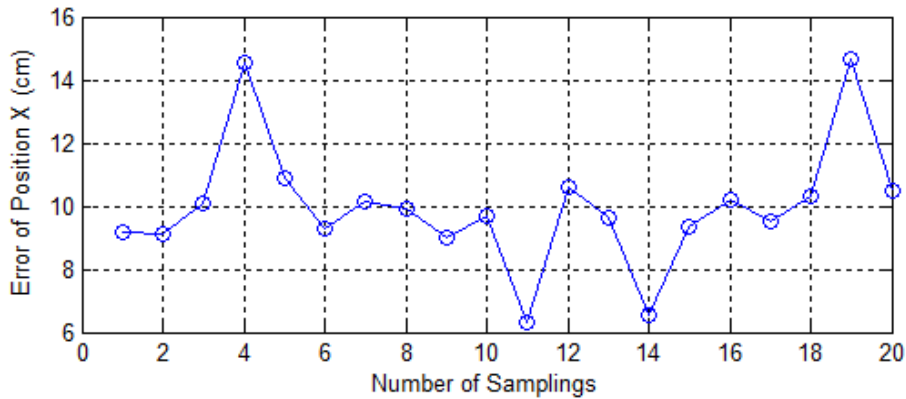


Figure 4.10: Measured errors of position X

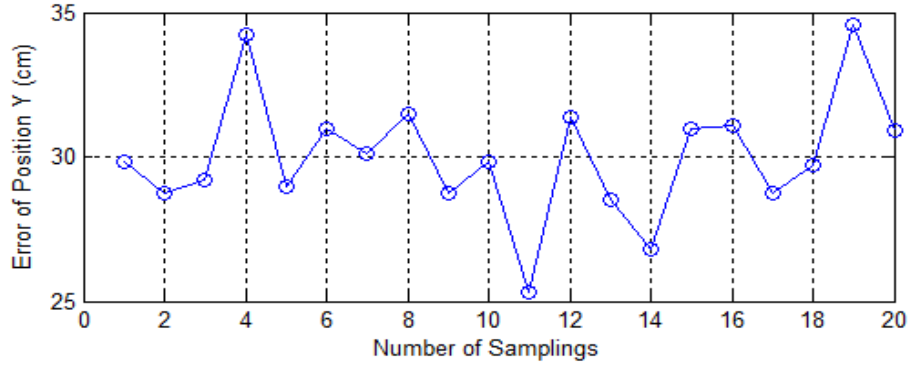


Figure 4.11: Measured errors of position Y

#### 4.2.2 Floor Landmark Based Local Localization

As demonstrated in Section 4.2.1, the performance of the StarGazer localization is unsatisfactory when it is implemented in the strong-lighting and reflective ceiling environments. But on the other hand the StarGazer module also has many impressive advantages. For instance, it is easy to be extended for large laboratories after installing additional landmarks and the cost of landmark installing is economic.

To improve the performance of the StarGazer module, in this study a hybrid method is proposed as demonstrated in Fig. 4.12. In this hybrid architecture, the works of the robot indoor localization have been classified into two kinds and in each one an independent solution is presented: (a) the StarGazer module working with the ceiling landmarks is retained to do the global indoor localization for the whole laboratory environments; (b) and a new method using a robot on-board Kinect depth sensor and a series of floor landmarks is presented to do the local localization for some high-precision required conditions (positions).

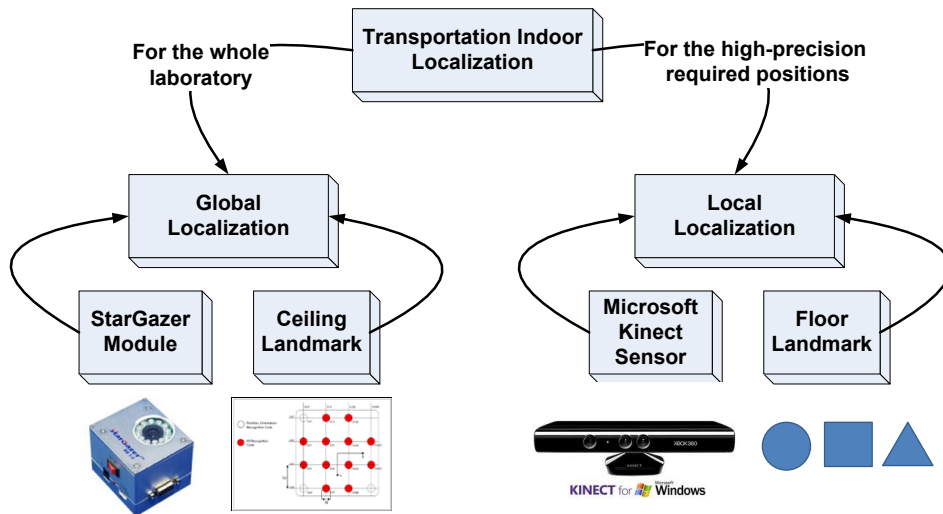


Figure 4.12: Hybrid method of indoor localization for laboratory transportation

#### 4.2.2.1 Microsoft Kinect Sensor

In the local localization, one kind of Microsoft Kinect sensors is installed in the H20 mobile robots to detect the color floor landmarks for the function of robot posture correction. Kinect sensor is a new production of Microsoft company [112] which was originally designed for the game playing. However it has been validated that it is qualified for engineering applications, especially for mobile robotics [113]–[115]. The specifications of the Microsoft Kinect sensor are given in Appendix A.2.

*L. Cheng et al.* developed a human-robot interactive demonstration system using the Microsoft Kinect sensor [113]. To let the non-expert users control the humanoid robots freely, they used the Kinect sensor to recognize the different body gestures of the operating human to control a demonstrated robot named Aldebaran humanoid robot Nao.

*D.S.O. Correa et al.* presented a system for surveillance mobile robot indoor navigation [114]. In their works, two main components were provided. The first one was to use the depth sensor of a Microsoft Kinect to detect the obstacles. The second one was to adopt an artificial neural network (ANN) to identify different obstacle configurations for the indoor navigation. The ANN was trained by the data measured from the Kinect sensor. A Pioneer P3-AT robot equipped with a Kinect sensor was given to estimate the performance of their proposed system.

*N. Moriyama et al.* discussed the developed of a navigation component for JAUS (Joint Architecture for Unmanned System) compliant mobile robots using the Kinect sensor [115]. In their paper, a laser range finder and a Kinect sensor were both employed to measure three-dimensional indoor environments, and an experiment was implemented to confirm the validity of this proposed Kinect-based navigation algorithm and system. More robotic applications using the Microsoft Kinect sensor can be seen in references [116]–[121].

Based on the upper mentioned literatures on Kinect robotic applications, the advantages of the Microsoft Kinect can be summaries as follows:

- Compared to normal 2D cameras, the Kinect sensor can be regarded as a 3D camera, which generates both of a 2D real-time image and a group of grid distances between the pixels in the 2D image and the recognized objects. This function of depth measurement is useful for the robot image-based localization and collision avoidance.
- The development API of the Kinect sensor has abundant functions and

properties, which allow the designers to develop their applications in a short time. The producer of the Kinect sensor-Microsoft company opens the latest development kits for the engineers.

- As mentioned in Chapter 2, every H20 mobile robot has an on-board laptop on which the RBC will run. Since the Kinect sensor has a USB port, it can be integrated conveniently by the RBC of the H20 robots without any hardware revision. This is also one of the reasons why we select the Kinect sensor instead of other possible solutions in the local localization.

#### 4.2.2.2 Strategy of Local Localization

The working mechanism of the proposed local localization is demonstrated in Fig. 4.13. It is explained as follows: In a transportation executing process, the selected H20 mobile robot will be given a path with a series of waypoints by the RRC. In this sequence, all of waypoints are measured and recorded automatically by the corresponding RBC. For the normal waypoints, the RBC will only use the Stargazer module and the ceiling landmarks to navigate the robots. If one of the waypoints needs high-precision localization, a floor landmark (red color circle) can be set in those positions. When the robot goes to those special positions, the robot on-board Kinect sensor will be activated automatically by the RBC to capture the floor landmark then the correcting parameters can be calculated by the RBC. The calculated results will be sent directly to the RBC motion module for the robot posture correction. A H20 robot carrying a Kinect sensor is shown in Fig. 4.14.

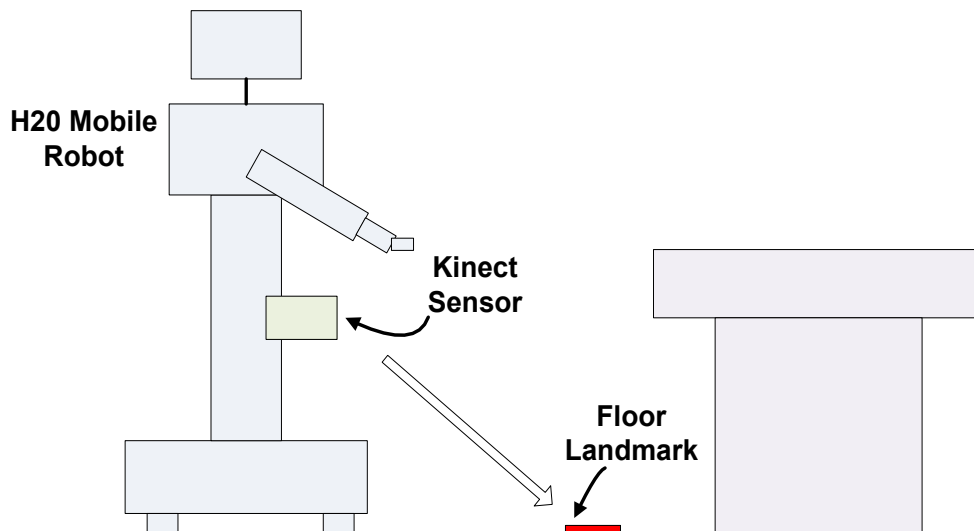


Figure 4.13: Graphical chart of local localization

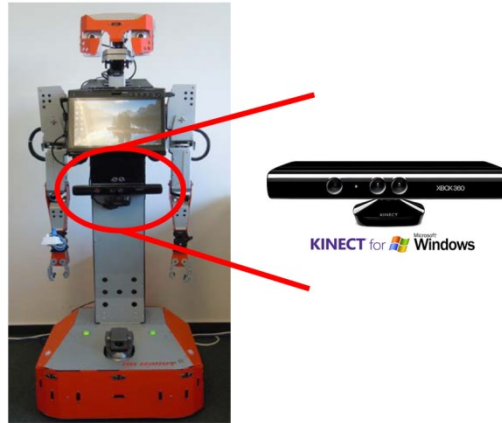


Figure 4.14: Kinect application in laboratory indoor local localization

Based on the upper proposed strategy, a software component is developed based on the Microsoft C# and the Microsoft Kinect Software Development Kit (SDK) as shown in Fig. 4.15. From Fig. 4.15, it can be seen that the developed GUI not only measures the distance between the moving robot and the floor landmark but also calculates the real-time errors (Coordinates X and Y) of those high-precision required positions. The correcting errors will be loaded to the H20 motion modules for the position correction. The workflow of the Kinect correcting function in this GUI is illustrated in Fig. 4.16. An experiment of this floor-based localization is provided in Appendix C.3.

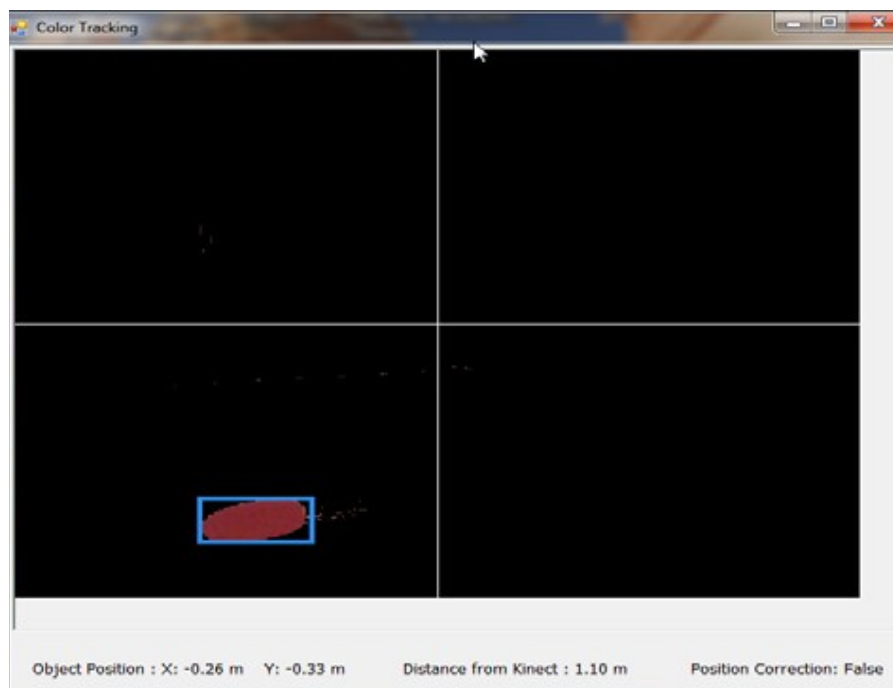


Figure 4.15: Main GUI of Kinect local localization function

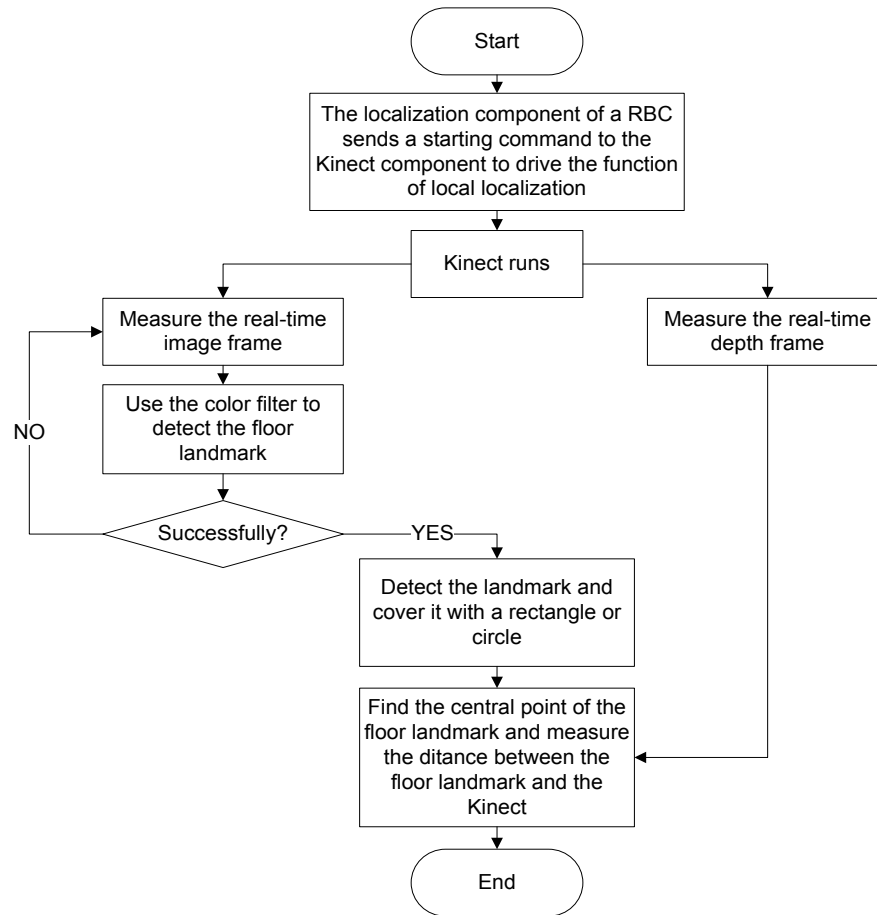


Figure 4.16: Flowchart of robot Kinect componnet

As shown in Fig. 4.16, the detailed steps of the Kinect correcting computation can be explained as follows:

- The RBC of a mobile robot sends a starting command to the GUI of the Robot Kinect Component (RKC) to drive the function of local localization;
- The RKC in this RBC measures the image and depth frame from the on-board Kinect sensor (the RGB and depth values for every pixel in the view);
- A color filter is executed on the sampled image frame;
- Build a Cartesian coordinate in the image frame;
- Detect the expected RBC object of the floor landmark and draw a rectangle or circle coving the landmark;
- Calculate the coordinates of the central pixel of the coving rectangle or circle in the built Cartesian coordinate and read the depth value of the central pixel.
- The RBC controls the robot moving to the floor landmark using the calculated distance and correcting coordinates;

## Chapter 5 Collision Avoidance

### 5.1 Introduction

Collision avoidance means the moving mobile robots recognize the front dynamic obstacles and select an alternative avoidance way automatically. The issue of robotic collision avoidance includes two main kinds: arm collision avoidance and motion collision avoidance. The first one happens when robots catch some objects through their arms, which always relates with the computation of robot arm kinematics. It tries to find a safe trajectory avoiding collide any other objects for every arm joint. The second one is for the mobile robots to move smoothly in an indoor environment without colliding any persons or obstacles.

In this study, the arm collision avoidance copes with the static obstacles (e.g. labwares on laboratory workbenches) and the motion collision avoidance handles with the dynamic objects (e.g. humans). The chapter emphasizes on the second one-motion collision avoidance. In recent years, there are some proposed methods on this topic, which are reviewed as follows:

**C.Y. Lee et al.** compared the performance between multi-ultrasonic sensors with fixed beam-width and that with different beam-widths then presented a robot indoor collision detecting method based on the comparing results [122]. In their proposed method, two different kinds of ultrasonic sensors have been used: small-beam width sensors were asked to detect environments with high resolution and large-beam width sensors were activated to measure possible obstacles in robot motion. An experiment showed that the method was effective for the robot collision measurement in a complex room.

**H.D. Kim et al.** studies the SLAM of mobile robots in indoor environments with Digital Magnetic Compass (DMC) and ultrasonic sensors [123]. Those two kinds of environment sensors were utilized to let mobile robots do autonomous path planning and collision avoidance. Also as they said, some computation intelligence models were presented to infuse the gotten distance data to perceive the robots' indoor positions.

**K. Izumi et al.** proposed an obstacle avoidance method for a quadruped walking robot using both of camera images and ultrasonic sensors [124]. In their application, a



single camera was used to sample the rough information of three-dimensional obstacles while additional ultrasonic sensors were installed to obtain the accurate information of those obstacles. The proposed fusing measure using more signal resources was proved through several experiments provided in that paper.

*S. Cui et al.* proposed a fuzzy based method for mobile robot obstacle avoidance [125]. In the method the environment surrounding parameters were measured by robot installing ultrasonic sensors then the sampled data were loaded to a dual-component fuzzy controller. There were two fuzzy models proposed: one was for the obstacle avoidance, and the other one was designed to guide the robot move to the targets.

*H.W. Je et al.* presented a collision observer to detect the collisions between service robots and unknown obstacles [126]. In the method, they did not need any extra data from outside sensors such as the force, tactile or visual information. Instead of the normal sensors, the collision in that paper was measured by an observer considering two parameters: current value and control inputting value. This method was validated by a real experiment successfully.

*P. Vadakkepat et al.* combined Artificial Potential Field (APF) and genetic algorithm (GA) for real-time robot path planning considering indoor obstacles [127]. In the paper, kinds of potential field functions for obstacles and target positions were defined, the APF was employed to identify the optimal potential field functions, and a new algorithm named escape-force was presented to avoid the phenomenon of local optimization by the APF. Several simulated results showed that the method was efficient and robust for robot avoidance path planning. More similar cases on the topic of robot indoor collision avoidance can be found in references [116], [128]–[131].

From the upper literature reviewing, it can be analyzed that: (a) in most applications of indoor robots, the collision avoidance is desired; (b) to detect the robot surrounding obstacles, the cameras and ultrasonic sensors are generally adopted; (c) among the collision path planning, the APF method is one of the most successful ones. This Chapter is organized as follows: Section 5.2 demonstrates the existent functions of the collision avoidance in the H20 mobile robots; Section 5.3 presents two new methods to solve the existent collision avoidance problems of the H20 robots; and Section 5.4 concludes the results of this Chapter.

## 5.2 Original Collision Avoidance of H20 Robots

In H20 mobile robots, the issue of collision avoidance has been considered and included. A series of environment measuring sensors including the ultrasonic sensors

and the IR sensors are used to detect the surrounding obstacles and a method named APF based algorithm is utilized for the robot collision local path planning. This section will discuss the collision avoidance performance of the H20 robots and analyze the existent problems.

### 5.2.1 Measuring Sensors

Fig. 5.1 shows the configuration of sensor modules in the H20 robots, which include 5 ultrasonic sensors (Sonar sensors), 10 IR range sensors and 2 motor encoders. The ultrasonic sensors and IR sensors are installed in the robot basis and body for the obstacle measurement, and the motor encoders are included for the robot motion control.

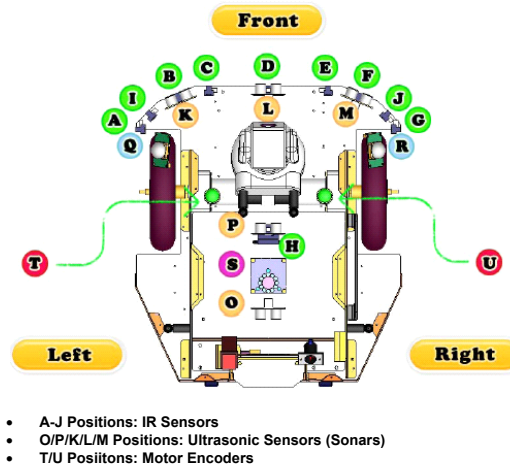


Figure 5.1: Sensing modules in H20 mobile robots for the collision avoidance [41]

### 5.2.2 Avoidance Path Planning

The H20 mobile robots adopt the APF for the collision local path planning. The APF is firstly proposed by Professor O. Khatib at Stanford University in 1995 [132]. In the APF, the robot is regarded as a particle moving immersed in a potential field, which is generated both by the target and by the obstacles. Every target makes an attractive potential while at the same time every obstacle generates a repulsive potential. The robot in the potential environment is subject to the action of a force driving it to the target while keeping it away from all obstacles. As demonstrated in Fig. 5.2, the robot actually will move by the action of the integration force ***F<sub>int</sub>*** of the attractive force ***F<sub>att</sub>*** and the repulsive force ***F<sub>rep</sub>***. Due to the simple computation and effective performance of the APF, it has been regarded as one of the most important algorithms for the robot collision avoidance and generally applied in robotic applications [133], [134]. The detailed computational steps of the APF can be found in references [127], [132], [135].

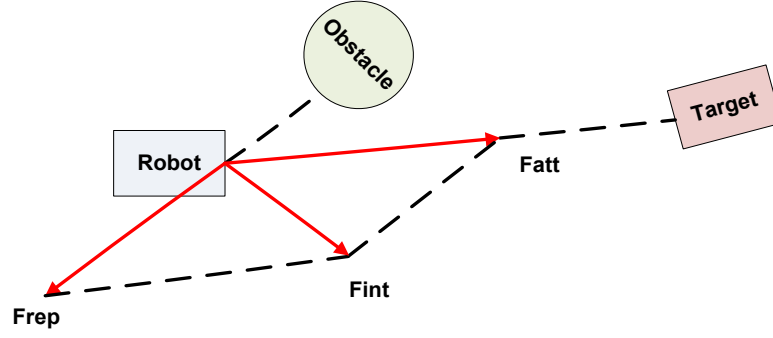


Figure 5.2: Mechanism of APF

### 5.2.3 Existent Problems

As explained in Section 5.2.2, the APF controls the robots moving to the targets based on kinds of integrated forces. Based on our previous experiments, we find that in most cases the APF can work effectively. But when the angle between the **Fatt** and the **Frep** is big, the integrated force would be insufficient to drive the robots out of the repulsive potential, and then the robots will easily go into a vibrating phenomenon as shown in Fig. 5.3. In this study the robots will run different sizes of laboratory rooms including some narrow corridors, so this disadvantage of the APF must be solved.

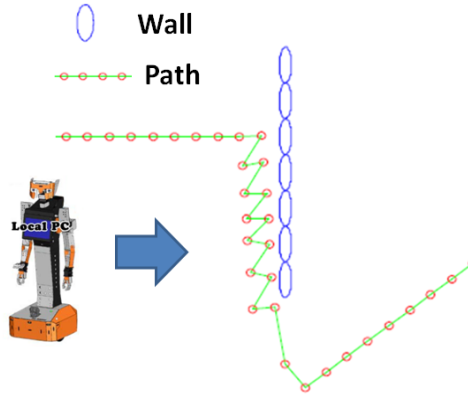


Figure 5.3: Vibrating phenomenon of APF

## 5.3 Improved Methods

To solve the vibrating phenomenon of the APF, two improved methods are provided in this application.

- One is to measure a longer distance between the robot and the obstacle. As explained in Section 5.2.3, if an obstacle could be found earlier and further, it is more possible to avoid the dead vibrating phenomenon. The robot on-board Kinect Sensor adopted in Chapter 4 for the robot indoor localization is also

utilized for the obstacle measurements in this chapter. Based on the Kinect sensor, not only the distance between the robot and the obstacle can be measured, but also the shape of the obstacle can be recognized. Another reason of using the Kinect sensor in the collision avoidance is explained as follows: in laboratories, sometimes the mobile robots need to find and avoid the obstacles in the other sides of some glass doors or walls. In those cases the Kinect sensors can be used instead of the traditional ultrasonic sensors. Two examples of Kinect obstacle recognition in the study are demonstrated in Figures 5.4 and 5.5.

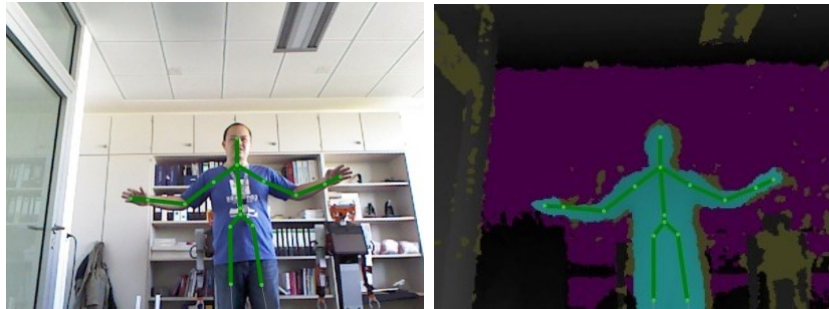


Figure 5.4: Human obstacle shape & skeleton recognition

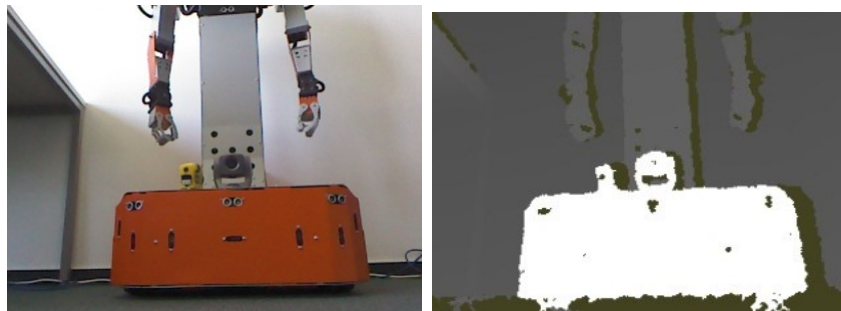


Figure 5.5: Obstacle depth recognition

- The second one is to path the collision avoidance in a global view. Since the RRC reads all positions of the running mobile robots, it is good to do a global path planning in the RRC side to avoid the collisions. Once the RRC finds more than two mobile robots are going to the same waypoint (position), it can control them to reach the position at different time by controlling the robot running speeds. The control priority could be decided based on considering both of their power status and important levels of the given transportation tasks. The global method of robot collision avoidance is demonstrated in Algorithm 5.1. The detailed computational steps of the adopted methods for the global path planning can be found in Chapters 6 and 7.

Algorithm 5.1: Global collision avoidance adopted in this study

<p><b>Input:</b> A series of distances between the robots and the obstacles.</p> <pre> 1:  d1 = the real-time sonar distance. 3:  d2 = the APF activation distance. 4:  d3 = the APF termination distance. 5:  while read the real-time distance do 6:    if d1 &lt; d2 do 7:      Start a collision avoidance program. 10:   if d1 &gt; d3 do 11:     Do a collision avoidance using the APF local navigation method. 12:   else 13:     Do a collision avoidance using the Floyd/GA path planning method 11:   end if 15: else 16:   Stop the collision avoidance program. 17: end if 18: <b>return</b> a collision avoidance path </pre>
---

## 5.4 Discussions

This chapter discusses the disadvantages of the function of collision avoidance in the H20 mobile robots. The analyzing results show that the collision avoidance of the H20 robots almost cannot be used in narrow laboratory environments. To improve the performance of this existent function, a new method is presented as follows: (a) in every H20 robot, an additional Kinect depth sensor is installed to measure farer obstacles and some obstacles in the other sides of laboratory glass doors. Compared to the existent ultrasonic sensors adopted in the H20 robots, the Kinect sensors not only can measure the distances between the obstacles and the robots but also can recognize the shapes of obstacles. The capacity of shape recognition is good for the following path planning work; and (b) a global avoidance path planning idea is proposed. As explained in Chapters 6 and 7, the transportation path planning is based on kinds of maps with a series of defined waypoints. So it is possible to control the robots to avoid each other by adjusting their running speeds. It means the issue of avoidance path planning will be considered together with the topic of transportation path planning.

## Chapter 6 Map Theory Based P2P Path Planning

### 6.1 Introduction

In this study the H20 mobile robots are always asked to do peer-to-peer (P2P) transportation tasks. So the work of path planning for the H20 robots is crucial. As explained in Chapter 2, the distributed RRC-RBC system architecture is adopted to organize the transportation. In the RBC of every H20 mobile robot, an independent transportation map will be defined. When the RBC is connected to a RRC, the RRC will do path planning for the corresponding robot and wait for a coming PMS transportation task. To do effective path planning for the RBCs, in the RRC side two kinds of works are indispensable: to sample all defined positions from the connected RBCs and to choose a right algorithm for path planning computation. The first topic will be given in details in Chapter 9 and the second one will be discussed in this Chapter.

This Chapter is organized as follows: Section 6.2 proposes a hybrid strategy using two algorithms from the map theory for the robot P2P path planning; Section 6.3 analyzes the performance of the two map theory-based planning algorithms; and Section 6.4 concludes the results of this chapter.

### 6.2 Hybrid Strategy

To have effective P2P path planning, a hybrid method based on Floyd algorithm and Dijkstra algorithm is proposed in this Chapter, which includes two computational modes:

- **Off-line mode:** Once the RRC receives all the prepared waypoints from the RBCs, the Floyd algorithm will be run at once to calculate all the shortest paths for any two waypoints. Once the RRC receives a transportation task from a higher PMS system, it will select the best mobile robot for the task and give it a path (the shortest path by default) automatically by searching for the corresponding stored path matrix. This process is a kind of off-line path planning mode.
- **On-line mode:** Once the RRC finds the next waypoint of the current running path is not available for a RBC (e.g. being occupied by a robot or a person), the Dijkstra algorithm will be immediately started to determine an alternative path between the robot current position and the target position for the RBC. If it is impossible to find an alternative path for this RBC (robot), the RRC will choose another mobile robot to go on the transportation process. This process is a kind of on-line path planning mode. Actually this mode is also strongly related with the issue of collision avoidance discussed in Chapter 5.

### 6.2.1 Floyd Algorithm

Floyd algorithm is one of the most popular graph algorithms for finding the all-pairs shortest paths in a positive or negative weighted graph, proposed by a computer scientist Robert Floyd [136]. Due to its robust performance and simple computation, it is popular in the issue of robot shortest path planning [137]–[142].

The main computational steps of the Floyd algorithm are explained as follows: (a) define an initial Distance Matrix  $DM(i,j)$  to record all adjacent edges for a graph; (b) establish an initial Path Matrix  $PM(i,j)$  to save all the numbers of the intermediate vertices in a shortest path between two vertices  $i$  and  $j$ ; (c) do an iterative process to find all the potential intermediate vertices among the data set of vertices. A searching algorithm is demonstrated in Algorithm 6.1; (d) after an iterative searching process finishes, the initial  $DM$  will be become as a final  $DM$  to store all the shortest distance for any two vertices, and the initial  $PM$  will save the corresponding numbers of passing intermediate vertices.

Algorithm 6.1: An iterative loop of the Floyd algorithm

```

1: Initialize  $DM(i,j)$  and  $PM(i,j)$ 
2: Loop  $k=1:n$ 
3:   Loop  $i=1:n$ 
4:     Loop  $j=1:n$ 
5:       If  $DM(i,j) > DM(i,k) + DM(k,j)$ 
6:          $DM(i,j) = DM(i,k) + DM(k,j)$ ;
7:          $PM(i,j) = PM(i,k)$ ;
8:       End If
9:     End Loop
10:   End Loop
11: End Loop

```

A transportation map including five waypoints is presented to demonstrate the working mechanism of the Floyd algorithm as displayed in Fig. 6.1. From the coordinate values of those waypoints in the map, a Floyd initial Distance Matrix can be calculated in Table 6.1. Based on the Floyd searching process described in Algorithm 6.1, the final Distance Matrix ( $DM$ ) and Path Matrix ( $PM$ ) can be demonstrated in Tables 6.2 and 6.3, respectively. From the Tables 6.2 and 6.3, the shortest path and its related distance for any two waypoints can be abstracted easily. For instance, the shortest path between the waypoint ① and the waypoint ④ can be output as a sequence of ①-②-③-④.

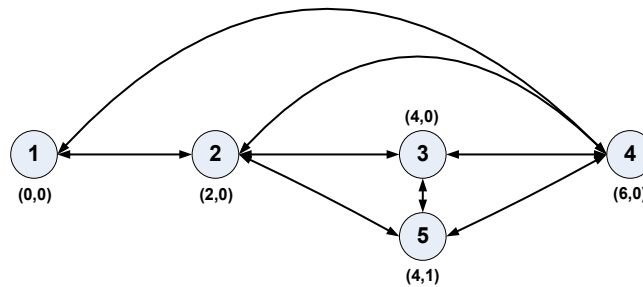


Figure 6.1: A demonstrated map with five working waypoints

Table 6.1 Initial distance matrix of Floyd algorithm

Distance	1	2	3	4	5
1	0	2	$\infty$	8	$\infty$
2	2	0	2	8	2.236
3	$\infty$	2	0	2	1
4	8	8	2	0	2.236
5	$\infty$	2.236	1	2.236	0

Table 6.2 Final distance matrix of Floyd algorithm

Distance	1	2	3	4	5
1	0	2	4	6	4.236
2	2	0	2	4	2.236
3	4	2	0	2	1
4	6	4	2	0	2.236
5	4.236	2.236	1	2.236	0

Table 6.3 Final path matrix of Floyd algorithm

Numbers of intermediate waypoints	1	2	3	4	5
1	0	2	2,3	2,3,4	2,5
2	1	0	3	3,4	5
3	2,1	2	0	4	5
4	3,2,1	3,2	3	0	5
5	2,1	2	3	4	0

### 6.2.2 Dijkstra Algorithm

In the robot path planning, there is an important issue that needs to be considered. When a robot is implementing an attributed Floyd path but the next waypoint in his path is just being occupied by another robot or a laboratory person, how to find an alternative on-line path quickly?

For this problem there are two simple solves: (a) one is to stop the moving robot to wait until the next waypoint becomes available again; and (b) the other is to re-run the finished Floyd program to update the existent shortest path matrix. Obviously the first one is not effective. As for the second one, it is also not preferred. Because once the Floyd algorithm is re-started, it will update all the shortest paths for any two waypoints including those passed ones because of the computational mechanism of the Floyd algorithm. Suppose in a huge waypoint case, the re-running process will cost a considerable time.

In this study Dijkstra algorithm is employed to solve this kind of emergencies. For example, in Fig. 6.1 when a robot is moving on this path and the waypoint ③ is just being occupied by another robot, an alternative path using a sequence of ②-⑤-④ can be presented easily by



the Dijkstra searching. Similar to the Floyd algorithm, the Dijkstra algorithm is also a famous algorithm in map theory, which is proposed to determine the single-source shortest path in a positive weighted graph. *H.I. Kang et al.* developed a path planning algorithm based on Dijkstra algorithm and particle swarm optimization [143]. *Y. Chao et al.* proposed an improved Dijkstra algorithm for the two-station shortest path planning in a big city [144].

The searching process of the Dijkstra algorithm can be demonstrated as follows: (a) Build a Path Set (*PS*) to record the intermediate waypoints for the shortest path. At the beginning the *PS* only has the starting waypoint (named ‘A’ point); (b) Initialize an Unvisited Set (*US*) to include all the waypoints except the starting waypoint; (c) Calculate all the distances between the ‘A’ point and its linked waypoints in the *US*, choose the shortest path and move its connected waypoint (named ‘B’ point) to *PS*; (d) Recalculate all the distances between the ‘B’ point and its linked waypoints in the *US*, then pick up the waypoint from the *US* whose path to the ‘A’ point through the ‘B’ point is the shortest; (f) Do an iterative searching process until the ending waypoint has been moved into the *PS* successfully, then the shortest path can be found in the *PS* based on the starting and ending points. The detailed calculation steps of the Dijkstra algorithm can be found in references [132-133].

### 6.3 Discussions

In this chapter a hybrid map theory-based path planning method is proposed for the P2P transportation tasks. It considers both of the simplicity and the flexibility of the path planning, which is comprised of two working modes (off-line/on-line). Based on the mode of off-line path planning, all available transportation paths can be calculated by the Floyd algorithm once the RRC receives all parameters of the predefined waypoints from the connected RBCs. Based on the mode of on-line path planning, a fast alternative path can be found by the Dijkstra algorithm if there is an emergency in the transportation execution. This hybrid method can be programmed conveniently using any mainstream developing languages. The developed software and the experiments using this method are demonstrated in Chapters 9 and 10, respectively.

## Chapter 7 Artificial Intelligence Based Patrol Path Planning

### 7.1 Introduction

In Chapter 6, a map theory based method for the robot P2P path planning in laboratories is presented. As described in Chapter 6, the P2P path planning works directly for a series of coming transportation tasks. Once the RRC is distributed a PMS transportation, it will build a P2P available path (always the shortest path) for the chosen mobile robot. However in some cases when the mobile robots are asked to go back to the original positions after executing a P2P path, the issue of the P2P path planning will become as a topic of patrol path planning. Also in some laboratories the mobile robots are controlled to do patrol transportation working together with other automated lines. For example, robots are asked to distribute experimental facilities to different laboratory workbenches or execute a patrol security task.

To do patrol path planning, there are two options. One kind is to repeat the process of the P2P planning a number of times based on the requirements of the expected transportation. The other kind is to plan the patrol path in a global view. Since the first one can be executed using the proposed method in Chapter 6, this chapter will focus on the second choice-global patrol path planning using Genetic Algorithm (GA) and Artificial Ant Colony Algorithm (AACA).

This Chapter is organized as follows: Section 7.2 presents a robot patrol path planning using the GA algorithm; Section 7.3 presents a robot patrol path planning using the AACA algorithm; Section 7.4 provides a comparison of the performance of the GA and AACA in the application; and Section 7.5 concludes the results of this chapter.

### 7.2 Genetic Algorithm

GA is a global optimal algorithm, which imitates the principles of biological evolution. It has been regarded as one of the best methods of global intelligent optimization. *Y. Hu et al.* proposed a knowledge based GA algorithm for the path planning of a mobile robot in unstructured environments [145]; *H.Y. Wang et al.* designed a PID (Proportional Integral Derivative) for the control of positions and velocities of mobile robots using the GA [146]; *J.S. Kong et al.* presented an optimal gait generation for a

humanoid robot using the GA [147]; *N. Sadati et al.* used a hybrid method based on Hopfield neural networks and genetic algorithm for solving the issue of robot motion planning in crisp environments [148].

The main calculation steps of the GA include making codes on chromosomes, generating initial populations, defining the fitness function and applying the evolutionary process, including selection, crossover, and mutation operations. A completed workflow of the GA is demonstrated in Fig. 7.1.

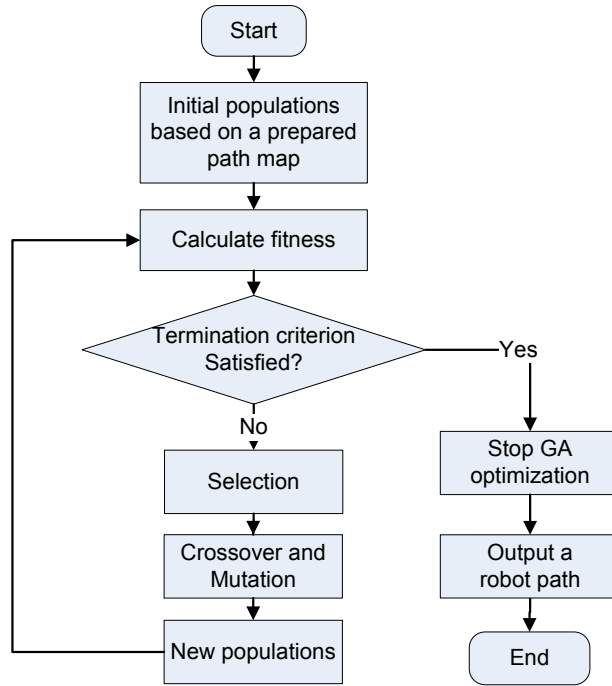


Figure 7.1: Computational steps of the GA

### 7.2.2 Chromosome coding

The traditional binary coding [149] and real-number coding [150] in GA theory do not suit for the case, so a new sequence coding is proposed. The new chromosome encoding from *Node 1* (Starting point) to *Node K* (Destination point) is demonstrated in Fig. 7.2. A chromosome will be recorded as a list of nodes.

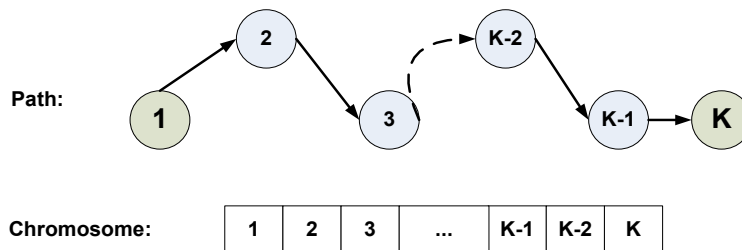


Figure 7.2: A robot path and its encoding format

### 7.2.3 Generation of initial populations

There are two ways to generate the initial populations: random initialization and heuristic initialization. Although the first one can guarantee the diversity of the populations, it is easy to cause a searching failure and needs more training time. To avoid obtaining a dead-loop or broken path in the graph map searching, a heuristic rule is proposed as follows: (a) only the connected waypoints can be included in the initial populations; (b) any waypoint is not allowed to be selected and included in the initial populations more than twenty times except for the starting point and the destination point; and (c) all the initial populations are generated in random by selected the connected middle points between the starting point and the destination point. As for the size of populations, it will be decided considering both of the model performance and the path searching time.

### 7.2.4 Fitness function

To simplify the GA searching process, a fitness function is defined as follows:

$$F_i = \frac{1}{\sum_{j=1}^{L_i-1} D_{G_i(j), G(j+1)}} \quad (\text{Equation 7.1})$$

Where  $F$  is the fitness value of a chromosome  $i$ ,  $L$  is the length of the chromosome  $i$ ,  $G_i(j)$  represents the gene (waypoint) of the path  $j$  in the chromosome  $i$ , and  $D$  is the distance between points.

### 7.2.5 Selection operation

The purpose of the selection operation is to improve the average quality of the population by giving the high-quality chromosomes a better chance to get copied into the next generation. Here the roulette wheel algorithm is adopted to do the selection operation. It selects the chromosomes based on their fitness results relative to the average fitness of the other chromosomes in the population.

### 7.2.6 Crossover operation

Crossover operation examines the current chromosomes to find better ones, which will guarantee the diversity of the solution space. To avoid the waypoint off-chain phenomenon of the parents caused by this crossover operation, two chromosomes selected for the crossover operation should have at least one common point except for the starting and destination points.

The detailed steps of the crossover operation are proposed as: (a) find the common points between two parent chromosomes except the starting and destination points;

Select any point randomly among those found common points as a crossover position;  
(c) execute a crossover operation. An example of the new crossover operation is demonstrated in Fig. 7.3.

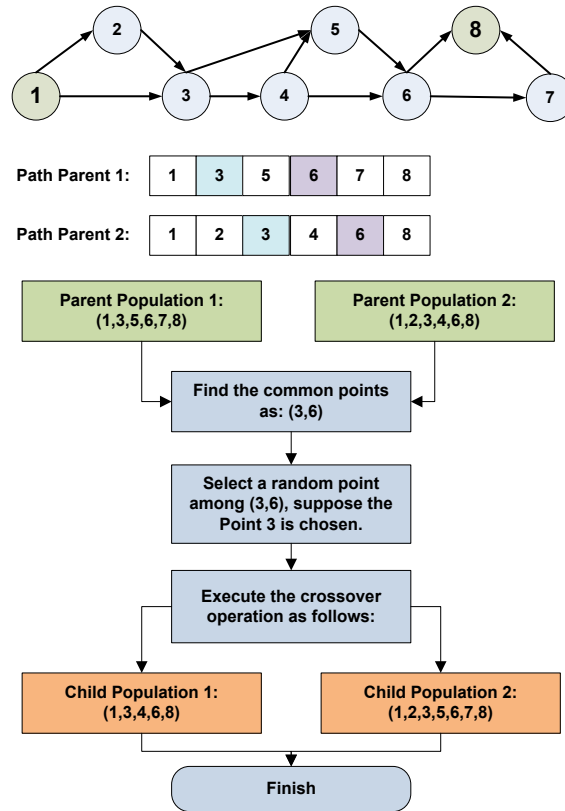


Figure 7.3: An example of the crossover operation

### 7.2.7 Mutation operation

To increase the diversity of the populations and prevent the premature convergence phenomenon during GA iterative searching process, a new rule of the mutation operation is proposed as follows: (a) select a random gene (waypoint) from the parent chromosome as the selection position; (b) find all the waypoints be connected to the selection waypoint as the potential mutation points; (c) choose a point from the found potential mutation points randomly as the official mutation points; (d) based on the calculated results by Floyd algorithm, find two shortest paths between the starting point and the mutation point, the mutation point and the destination point, respectively; (e) check if the two new sub paths have a common waypoint (the purpose of the check is to avoid a possible dead-loop problem): if no, merge them as the new child chromosome to replace the parent one; if yes, reselect an official mutation point.

A case of using the new mutation operation for a supposed map (*see Fig. 7.4*) is demonstrated in Fig. 7.5.

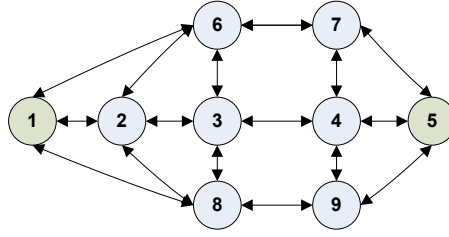


Figure 7.4: A topology graph with nine waypoints

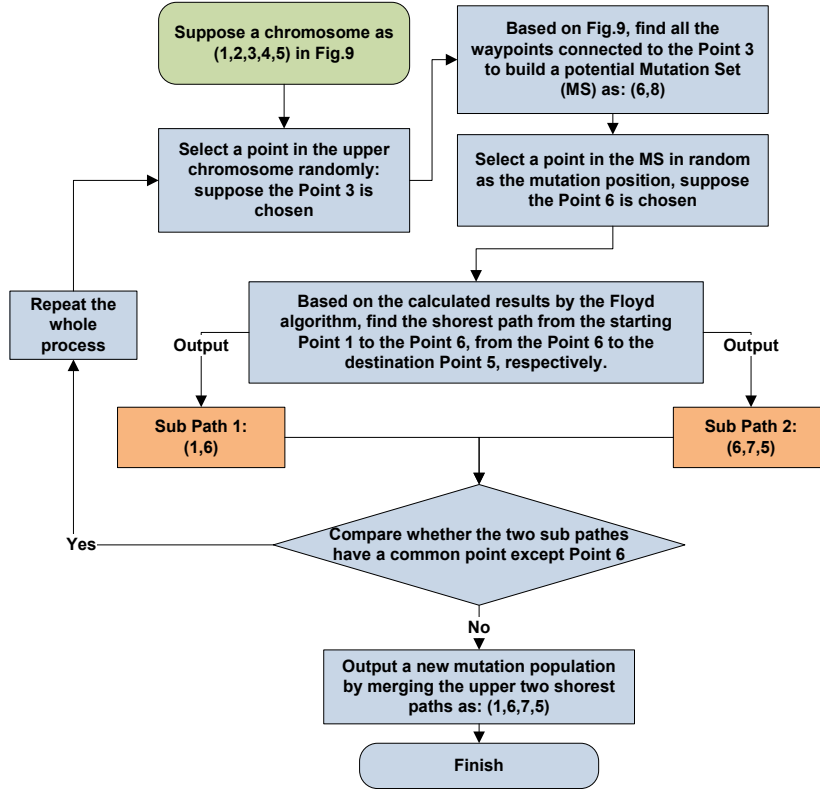


Figure 7.5: An example of the mutation operation

### 7.3 Artificial Ant Colony Algorithm

AACA is a member in artificial swarm intelligence methods, which is initially proposed by Marco Dorigo in 1992 in his doctoral thesis. It simulates the group activities of ants in finding foods [151]. As shown in Fig. 7.6, a group of ants are asked to find a food where is away to their nest. At the beginning (*see Fig. 7.6.A*), ants will go and back between the nest and the food position normally. When there is an obstacle suddenly showing on their path to the food (*see Fig. 7.6.B*), they will avoid the obstacle by passing through the side paths randomly (*see Fig. 7.6.C*). But during their avoiding movements, they will leave some smelling information (pheromone) on their walking paths. The next ants will change their paths based on the previous ants' pheromone, which show the distances of the avoiding paths. After a short time, all

ants will find the shorter path to avoid the obstacle (see Fig. 7.6.D). The AACA is original from this phenomenon. In this chapter, the AACA is adopted to find the shortest paths for kinds of robot patrol transportation tasks.

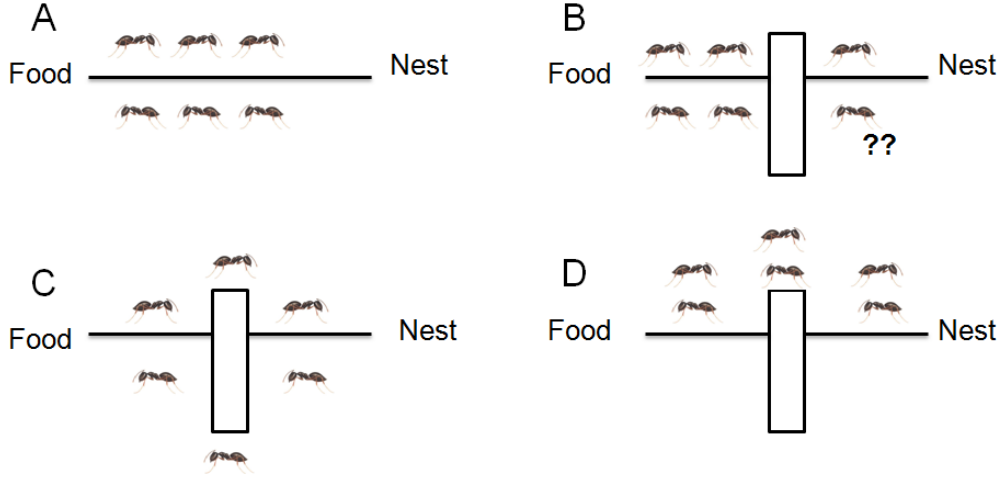


Figure 7.6: Working mechanism of AACA

In the AACA, the computational steps can be demonstrated in Fig 7.7, which includes initialization of model parameters, establishment of solution space, update of ant pheromone and determination of model termination.

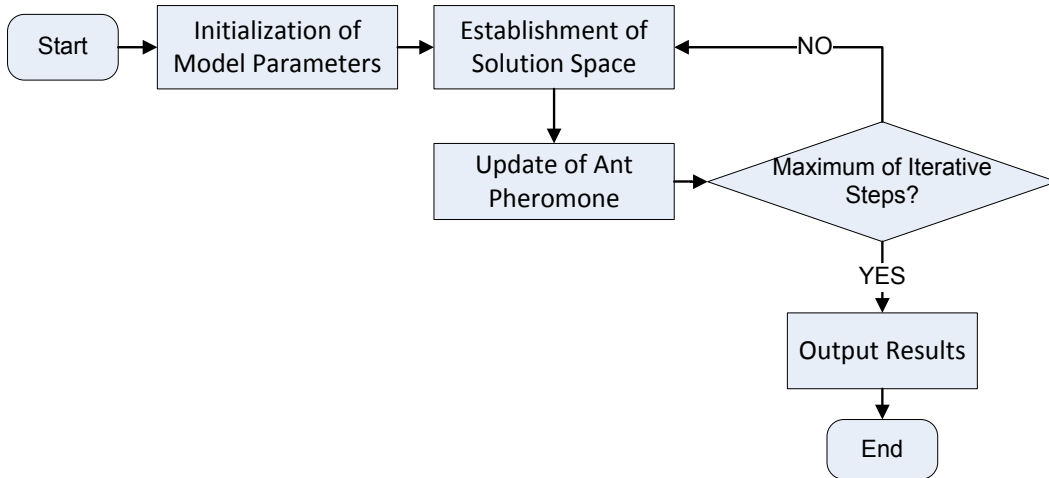


Figure 7.7: Computational steps of the AACA

In the AACA, the key parameters include number of ants, pheromone concentration, pheromone factor, heuristic function factor, pheromone quantity and number of iterative steps. Here a series of equations are adopted to explain those parameters.

$$P_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^{\alpha} [\eta_{ij}]^{\beta}}{\sum_{s \in A} [\tau_{is}(t)]^{\alpha} [\eta_{is}]^{\beta}}, & s \in A \\ 0, & s \notin A \end{cases} \quad (\text{Equation 7.2})$$

Equation 7.2 calculates the transfer probability  $P_{ij}$  for an ant  $K$  between the position  $i$  and the position  $j$ . In the Equation 7.2,  $\tau_{ij}(t)$  is the pheromone concentration at time  $t$  between the position  $i$  and the position  $j$ ,  $\eta_{ij}(t)$  is the heuristic function at time  $t$  between the position  $i$  and the position  $j$ ,  $\alpha$  is the pheromone factor,  $\beta$  is the heuristic function factor and  $A$  is the allowed assemble which records all no passed positions. In the AACA, this step of selecting the next passing position for an ant is one of the most important computational steps, because it guides the ants to choose the shortest paths. The positions with more pheromone concentration will have higher probability to be selected by ants.

On the other hand, to avoid the phenomenon of algorithm premature convergence, the AACA updates the pheromone in every iterative step. It will add the new pheromone and also volatilize the existent pheromone by using the following equations:

$$P_{ij}^k = \begin{cases} \tau_{ij}(t+1) = (1-\rho) \tau_{ij}(t) + \Delta \tau_{ij} \\ \Delta \tau_{ij} = \sum_{k=1}^n \Delta \tau_{ij}^k \end{cases}, \quad 0 < \rho < 1 \quad (\text{Equation 7.3})$$

Where  $\Delta \tau_{ij}^k$  is the added concentrate between the position  $i$  and the position  $j$  by the ant  $k$  in an iterative process,  $\Delta \tau_{ij}$  is the total added pheromone concentrate caused by all working ants between the position  $i$  and the position  $j$  in an iterative process, and  $\rho$  is the pheromone volatilization factor.

For calculating the  $\Delta \tau_{ij}^k$ , three different equations are presented as:

$$\Delta \tau_{ij}^k = \frac{Q}{L_k} \quad (\text{Equation 7.4})$$

Where  $Q$  is an ant's whole pheromone quantity in an iterative step, and  $L_k$  is the whole distance including all passed positions by an ant in an iterative step.

$$\Delta \tau_{ij}^k = \frac{Q}{D_{ij}} \quad (\text{Equation 7.5})$$

Where  $Q$  is an ant's whole pheromone quantity in an iterative step, and  $D_{ij}$  is the distance between two adjacent positions by an ant in an iterative step.

$$\Delta \tau_{ij}^k = Q \quad (\text{Equation 7.6})$$

Where  $Q$  is an ant's whole pheromone quantity in an iterative step.



Based on the different computational equations to update the new pheromone, three computational systems of the AACA are defined: ant cycle system, ant quantity system and ant density system. In this study, the ant quantity system is used.

#### 7.4 Comparison of GA and AACA

An algorithm experiment is provided to compare the performance of the GA and the AACA in this study. In this experiment, a group of 100 positions is adopted as the testing data. The GA and the AACA are asked to find the shortest path based on those 100 positions, respectively. This experiment can simulate that a mobile robot is controlled to patrol a series of 100 interested positions in laboratories.

The experimental results of the GA are given in Figures 7.8, 7.9 and 7.10. Fig. 7.8 displays the result without the GA computation. Fig. 7.9 shows the results with the GA computation. As given in Fig. 7.10, the GA needs around 1200 iterative steps to get the final optimization.

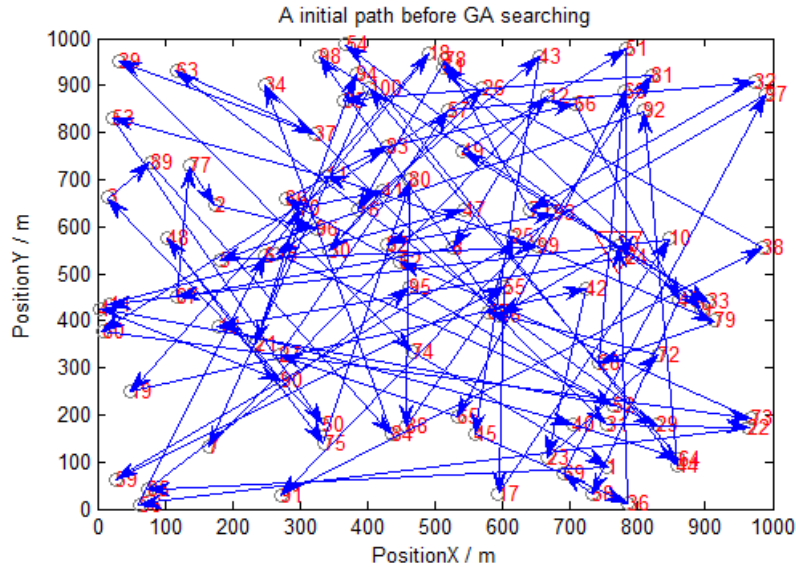


Figure 7.8: The random planning result without GA computation

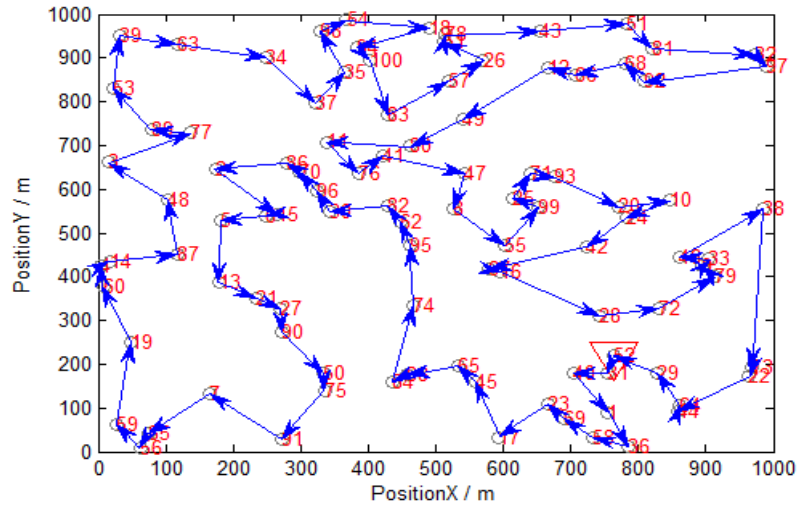


Figure 7.9: The planning result with GA computation

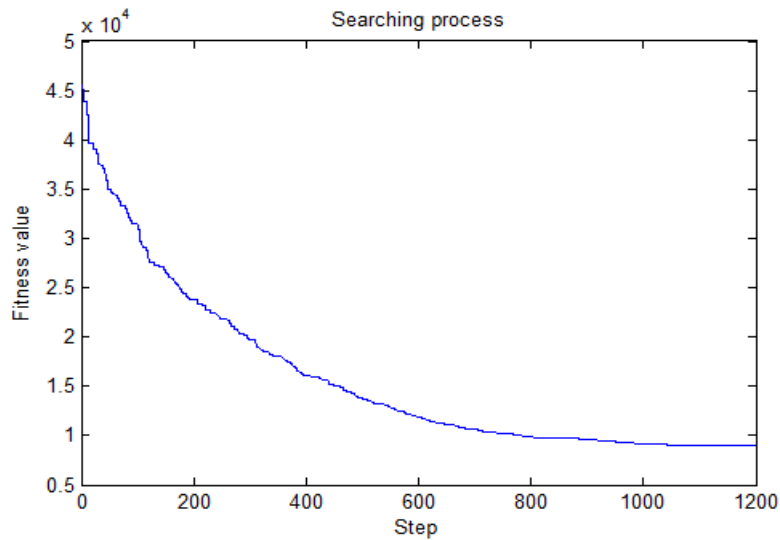


Figure 7.10: The iterative process of the GA computation

The experimental results of the AACA are given in Figures 7.11 and 7.12. Finally a path with a length of 8190m is selected as shown in Fig. 7.11. As shown in Fig. 7.12, in the AACA searching process only after 55 iterative steps an ant finds a satisfactory path, after 150 steps all ants also find this path. The parameters of the AACA used in the comparison are listed in Table 7.1.

Table 7.1: Algorithm parameters of AACA training

Parameter	Value
Number of Ants	50
Pheromone Factor	10
Heuristic Function Factor	100
Pheromone Quantity	0.5
Pheromone Volatilization Factor	0.4
Number of Iterative Steps	0.95

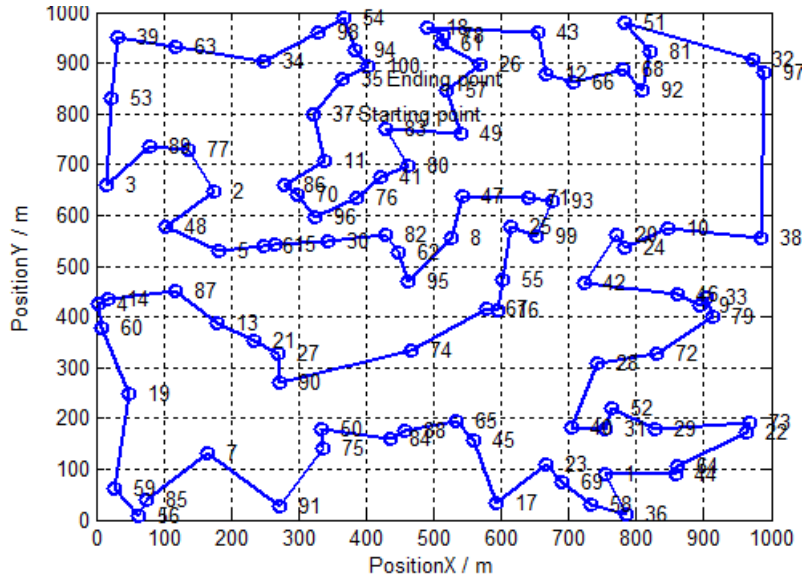


Figure 7.11: The planning result with AACA computation

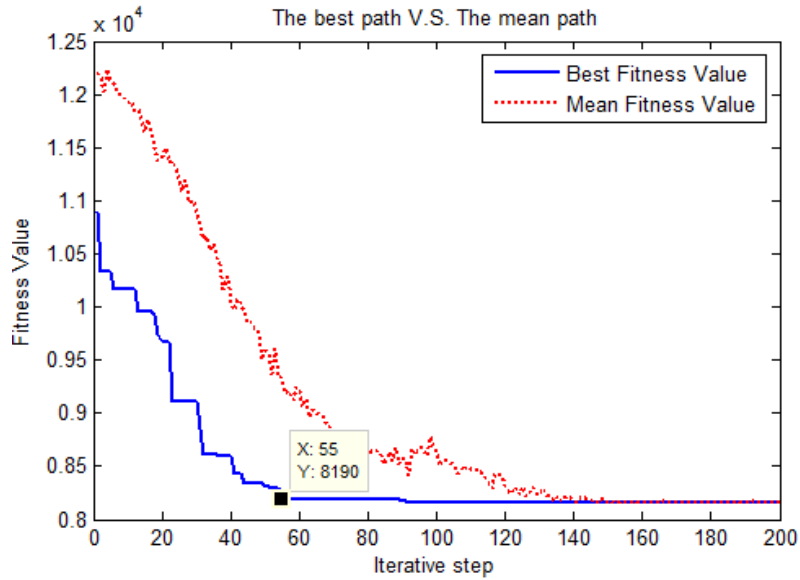


Figure 7.12: The iterative process of the AACA computation

Comparing Fig. 7.9 and Fig. 7.12, it can be found that the performance of the ACCA is much better than that of the GA. The GA uses 1200 steps to find a path with a length of 8497m, but the AACA only costs around 55 steps to get a path with a length of 8190m. In our opinions, the reason to the results can be explained as follows: although the GA has the possibility to find a global result if it runs rightly, it is still easy to drop into a local optimized area because it does not have feedback signals from its GA operating process. All of the GA operations are executed using the random principle. As for the AACA, although it also adopts the random principle in the computational process, but at the same time it utilizes the signal feedback rule. For

example, although at the beginning of the iterations all of ants are distributed randomly in every position, every ant will leave the pheromone information on their paths to guide the latter coming ants. This is one kind of positive feedback.

## **7.5 Discussions**

In the chapter two intelligent algorithms (the GA and the AACCA) are adopted to do the patrol path planning for the mobile robot transportation in this study. A performance comparison between the GA and the AACCA is provided based on the same experimental position data. The comparing results show that: (a) both of the GA and the AACCA are effective in the path planning with huge numbers of data; and (b) due to the AACCA has the capacity to receive the feedback information from the algorithm iterations, its performance is better than that of the GA.

## **Chapter 8    Multi-Robot Charging Management**

### **8.1 Introduction**

In laboratory mobile robot transportation, an important issue named charging management has to be considered. When the mobile robots are asked to do long-distance transportation, they definitely need to carry rechargeable batteries. When a big number of mobile robots are working together, their effective re-charging management becomes complicated. In our opinions, to solve this technical issue, three aspects are essential: (a) a system (or module) to measure power voltages from all working mobile robots. This module will control the mobile robots to go charging automatically if their batteries are exhausted soon. This module can be integrated with other measurement modules in the Server/Client centers (e.g., motion module, arm module and indoor localization); (b) a fully automated charging station is desired. This station should at least have the functions of auto-docking and auto-recharging. When a mobile robot reaches an expected position where the charging station is installed, it can do re-charging automatically. On the other hand, when a mobile robot finishes charging, it also can leave the station automatically; and (c) there is an optimized question: how to configure the charging stations most scientifically so mobile robots can work in laboratories most effectively.

In this chapter the methods to solve the upper three issues are presented, which are organized as follows: Section 8.2 proposes a power measurement system for the laboratory mobile robot based transportation; Section 8.3 discusses a fully-automated charging station for laboratory environments; and Section 8.4 studies an intelligent method named Artificial Immune Algorithm (AIA) for the optimization of station installing positions.

### **8.2 Power Measurement System**

This power measurement system is a part of the whole transportation control system. Since the transportation system adopts the Client/Server system architecture, the power measurement system is designed to be embedded in both of the RRC and the RBC as shown in Fig. 8.1. The real-time power voltages of all working H20 mobile robots will be measured by their RBC sides and sent to the RRC side. Since the RBCs are installed in the robot on-board laptops, besides the power voltages of the robot

batteries, the power percentage of the robot laptops are also measured. One of the power management GUIs in the RBCs is demonstrated in Fig. 8.2.

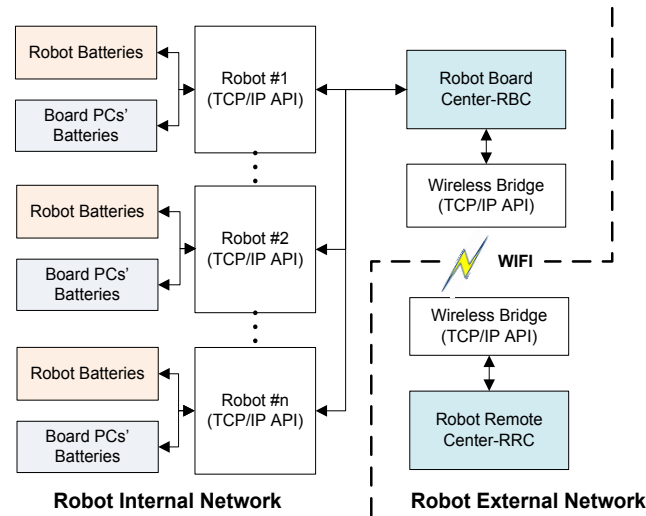


Figure 8.1: Architecture of robot charging management

Power Status				
	Voltage	Temperature	Status	PC Battery
Battery-I:	13.06	2009	Charging	98.00000190
Battery-II:	13.29	2017	Charging	Detect TH: 20
DCIN:	16.50		Using	<input checked="" type="checkbox"/> PC Battery

Figure 8.2: GUI of charging control in the RBC

### 8.3 Automated Charging Station

One kind of automated charging stations from Canada DrRobot company is utilized in the study (*see Fig.8.3 and Fig.8.4*). Once the mobile robots touch the plugs of those stations, they will go re-charging automatically.

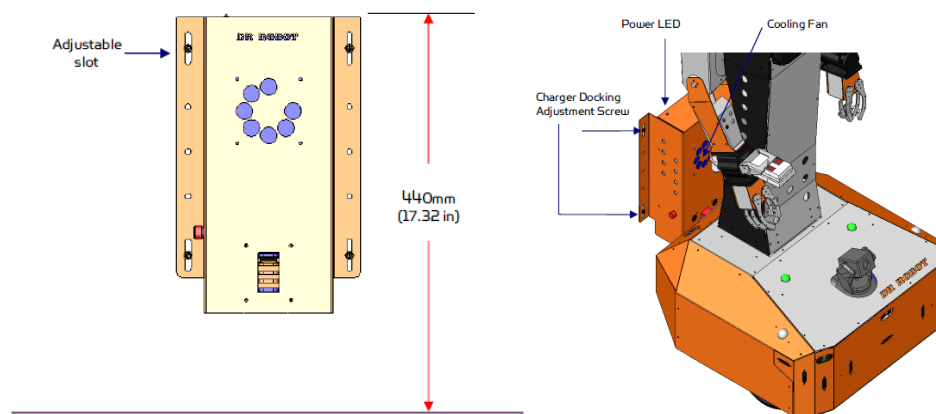


Figure 8.3: Automated charging station [41]



Figure 8.4: Charging station installed at Celisca, Germany

## 8.4 Optimization of Charging Station Installation

### 8.4.1 Artificial Immune Algorithm

In the study the AIA is presented to optimize the installation of the adopted fully-automated charging stations in laboratories, as demonstrated in Fig.8.5. The installing positions of the charging stations will be determined by two parameters: (a) the distances between all expected transportation positions and the charging stations. It means the charging stations should be installed in the area where more working positions are included; and (b) the task distribution of all transportation positions. It considers the charging stations should be closer to those positions where the mobile robots will pass through most frequently.

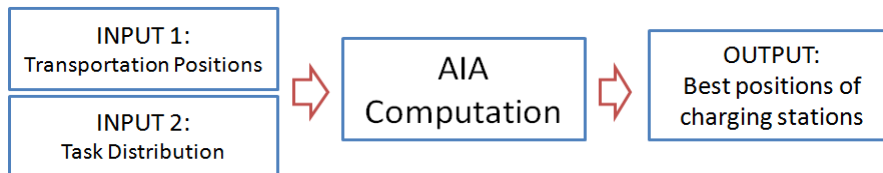


Figure 8.5: AIA optimization

### 8.4.2 Standard Computational Process

The standard computational steps of the AIA are demonstrated in Fig.8.6, which includes five steps: antigen identification, generation of initial antibodies, fitness estimation, selection of memory cells and AIA operations (selection, crossover and mutation). As shown in Fig.8.6, the working mechanism of the AIA is similar to that of the Genetic Algorithm (GA). Both of them run an iterative computational process including the operations of selection, crossover and mutation to find the best global optimized results. However, there is also an obvious difference between them. The AIA has a key step named selection of memory cells beside the standard selection

operation together with crossover and mutation. In the AIA iterative step, the best antibodies will be selected as memory cells without AIA operations, and they will be included in the next new generations. This step is important in an AIA process, because it will avoid a negative phenomenon called premature convergence in the GA. At the same time to guarantee the diversity of searching populations, the AIA also employs the operations of selection, crossover and mutation, which the GA has.

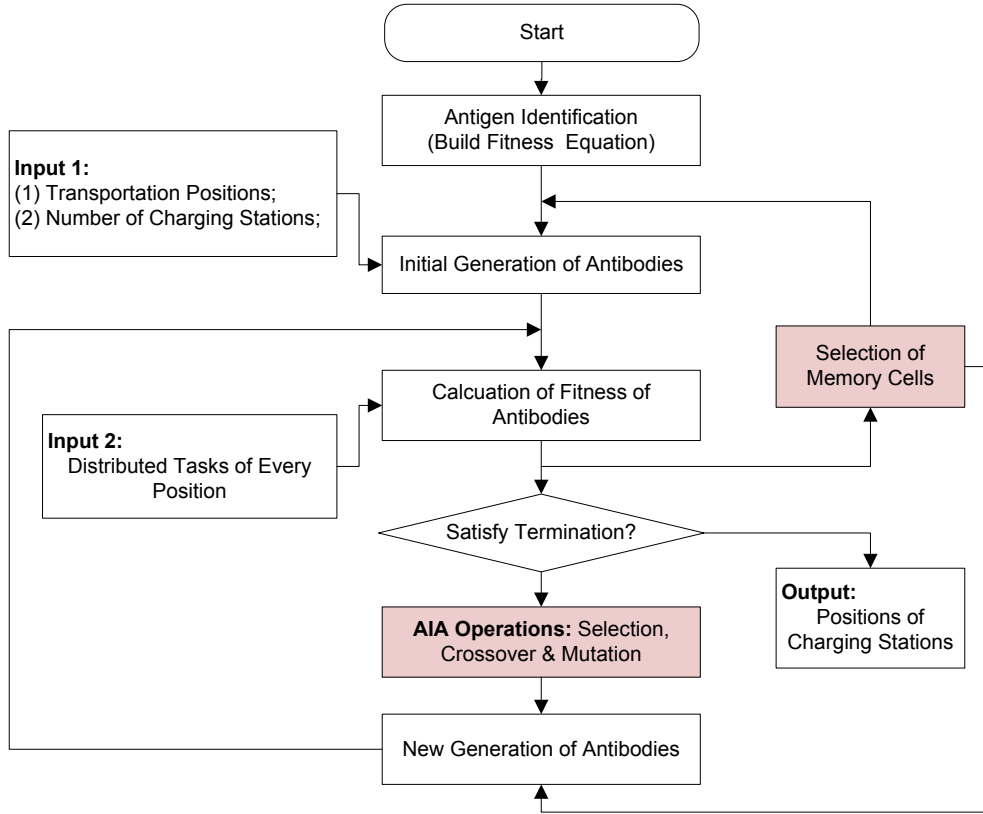


Figure 8.6: Standard AIA computational process

The equation of the fitness  $A_v$  of an antibody  $V$  is adopted as:

$$A_v = \sum_{i \in N} \sum_{j \in M} T_i \times D_{ij} \quad (\text{Equation 8.1})$$

Where  $N$  is the number of inputting positions,  $M$  is the number of charging station positions,  $T_i$  is the number of distributed tasks of the position  $i$ , and  $D_{ij}$  is the distance between the position  $i$  and the position  $j$ .

### 8.4.3 Improved Computational Steps

In this Chapter two new steps are added in the AIA computation to improve the performance of the standard AIA method as shown in Fig. 8.7.



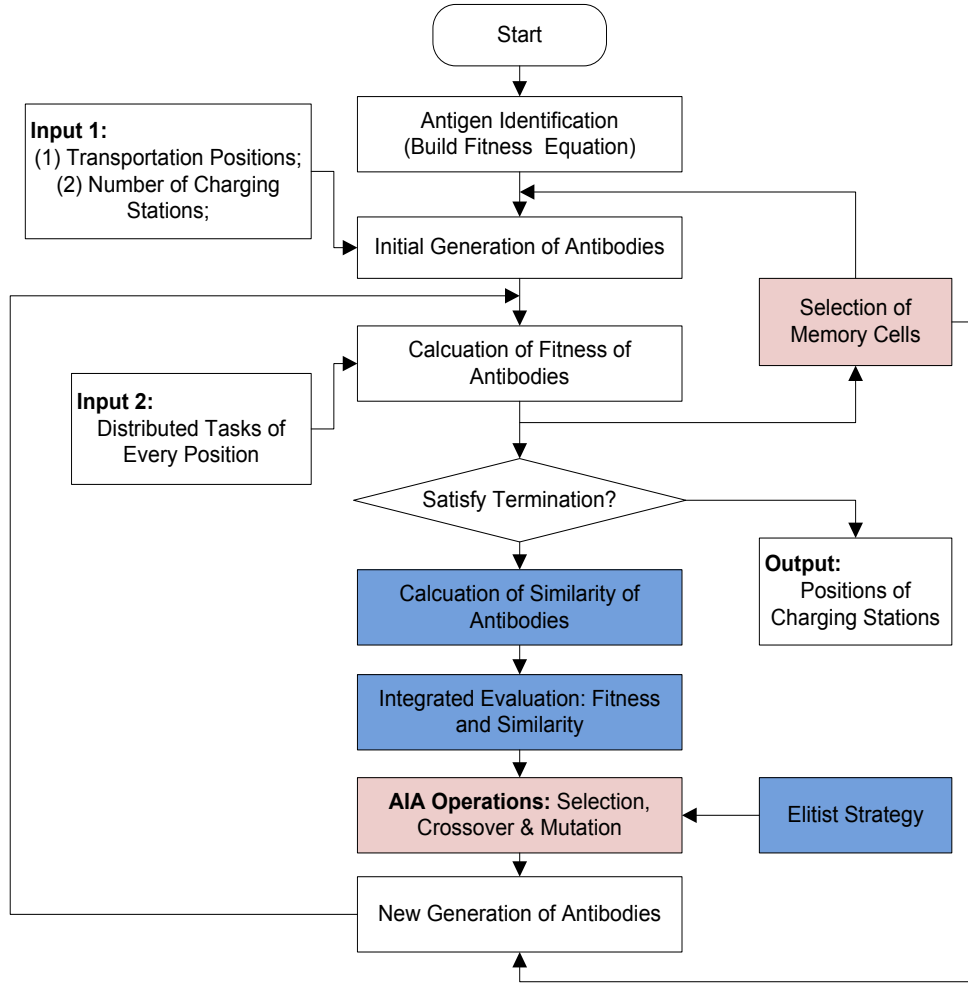


Figure 8.7: New AIA computational process

### (a) New estimation index considering both of fitness and similarity

The first one is to present a new estimation index to replace the single fitness index to select the best antibodies for the AIA operations. This new weighted parameter not only considers the fitness of an antibody in a generation but also focus on the similarity of the antibody in the generation.

$$P = a \times \frac{A_v}{\sum A_v} + (1-a) \times \frac{C_v}{\sum C_v} \quad (\text{Equation 8.2})$$

Where  $A_v$  is the fitness index of an antibody  $V$ ,  $C_v$  is the similarity index of the antibody  $V$ , and  $a$  is the scale factor which balances the fitness and the similarity for AIA operations.

### (b) Elitist strategy for AIA operation

From Equation 8.2 it can be found that in this study the best antibodies will be selected by estimating both of their fitness and similarity. The general idea of the new estimation index is to promote those antibodies with high fitness and restrain those antibodies with high similarity in a generation. However, in some cases those antibodies with high fitness are also possible to have high similarity, which could be abandoned in the next iteration. To avoid this kind of mistakes, an elitist strategy is provided in the selection step of the AIA operations. It means those antibodies with obviously excellent fitness can be kept before the selection.

#### 8.4.4 Results and Analysis

To demonstrate the effective performance of the proposed AIA method, an experiment is provided. A group of position data (*see Fig.8.8*) and task distribution data (*see Fig.8.9*) are loaded into the AIA model. The results are given in Figures 8.10 and 8.11. The model parameters in this experiment are demonstrated in Table 8.1.

Position_Coordinate: 2x31 double =							
Columns 1 through 8							
1304	3639	4177	3712	3488	3326	3238	4196
2312	1315	2244	1399	1535	1556	1229	1044
Columns 9 through 16							
4312	4386	3007	2562	2788	2381	1332	3715
790	570	1970	1756	1491	1676	695	1678
Columns 17 through 24							
3918	4061	3780	3676	4029	4263	3429	3507
2179	2370	2212	2578	2838	2931	1908	2376
Columns 25 through 31							
3394	3439	2935	3140	2545	2778	2370	
2643	3201	3240	3550	2357	2826	2975	

Figure 8.8: Experimental position inputs

Task_Distribution: 1x31 double =																
Columns 1 through 17																
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
Columns 18 through 31																
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20

Figure 8.9: Experimental task distribution inputs

Table 8.1: Algorithm parameters of AIA training

Parameter	Value
Size of Population	50
Size of Memory Cells	10
Number of Iterative Steps	100
Probability of Crossover	0.5
Probability of Mutation	0.4
Probability of Similarity	0.95
Number of Charging Stations	3

From Fig. 8.10 and Fig. 8.11, it can be seen that only after six iterative steps an optimized result has been found.

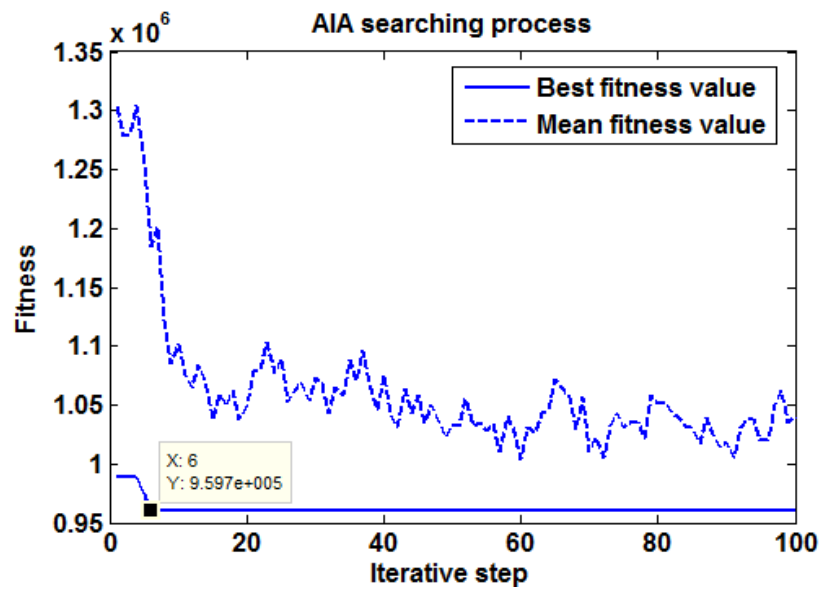


Figure 8.10: Iterative steps in Case One

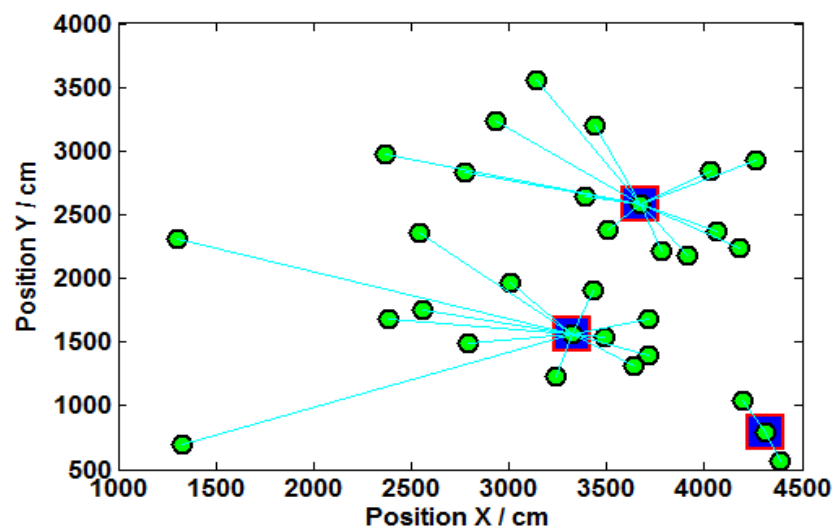


Figure 8.11: Calculated results in Case One

Another group of testing task data with different distributed is reloaded into the same AIA model as displayed in Fig. 8.12. The results are given in Figures 8.13 and 8.14. From Fig. 8.13, it can be seen that an optimized result has been found after seven iterative steps. As shown in Figures 8.12 and 8.14, the distributed tasks of a position (X: 1332, Y: 695) are changed purposely from 20 to 10000 to check the difference AIA result. From Fig. 8.14, it can be found that the best installing positions are totally changed which are moved closer to the redefined position (X: 1332, Y: 695).

Task_Distribution: 1x31 double =							
Columns 1 through 8							
20	20	20	20	20	20	20	20
Columns 9 through 16							
20	20	20	20	20	20	10000	20
Columns 17 through 24							
20	20	20	20	20	20	20	20
Columns 25 through 31							
20	20	20	20	20	20	20	

Figure 8.12: Experimental task distribution inputs

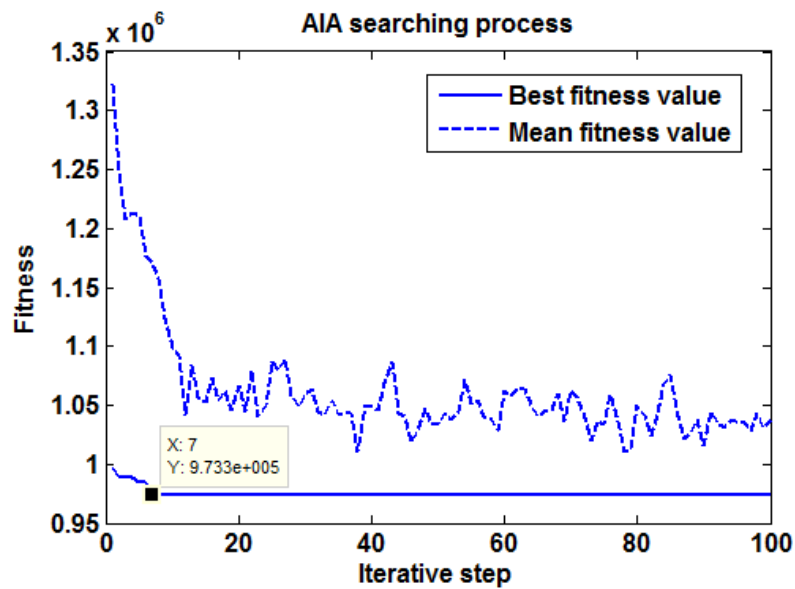


Figure 8.13: Iterative steps in Case Two

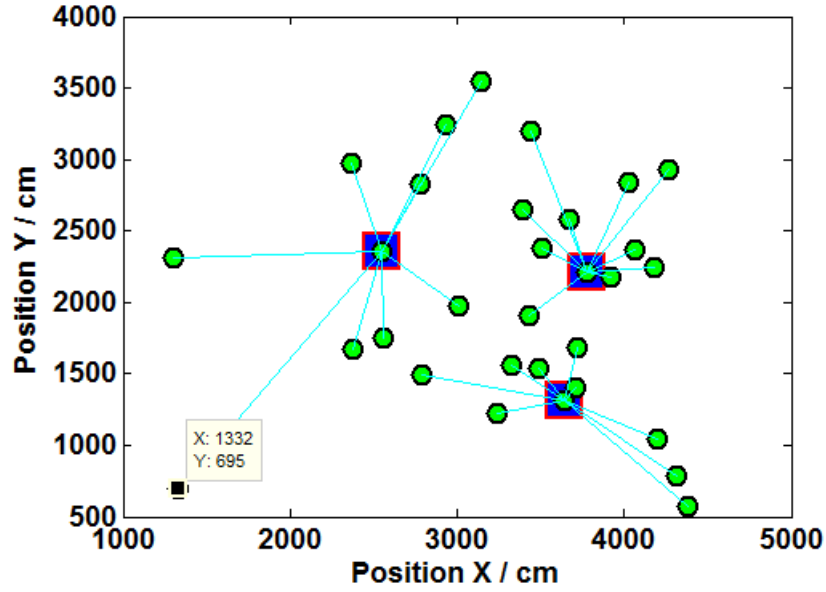


Figure 8.14: Calculated results in Case Two

## 8.1 Discussions

In this chapter a new method to study the issue of multi-robot charging management in laboratory environments is presented. This method consists of a developed robot power measurement & control component, a fully automated charging station and an ACA-based optimized algorithm. The measurement & control power component is executed in both of the RRC and the RBCs of the whole transportation system, a series of automated charging stations is also been installed in our laboratories, which installing positions are selected by the proposed ACA-based computational algorithm. It is good to say here that there are two improvements proposed in the ACA as follows: (a) a new estimation index considering both of the fitness and the similarity of an antibody is presented to execute the selection operation; and (b) an elitist strategy is provided to generate a new group of antibodies.

## Chapter 9 Software Development

### 9.1 Development Environments

As mentioned in Chapter1, there are three control centers for the transportation system: RBC, RRC and RAC. All of them are developed based on Microsoft C# language. The C# is a new programming language designed for developing a wide range of enterprise applications running on .NET framework. Compared to the previous developing languages created by Microsoft company, the C# is faster, safer and more object oriented. A developing platform named Microsoft Visual Studio is adopted in this study to do the C# programming for the transportation system.

### 9.2 Software Architecture

The architecture of the developed centers is demonstrated in Fig. 9.1. From Fig. 9.1, it can be seen that: (a) there are five control modules in the RRC, eight control modules in the RBC and three control modules in the RAC; (b) Comparatively, the RBC and the RAC are both a hardware control level and the RRC is a software control level. All hardware modules (i.e., kinds of sensors, cameras and servo components) inside the H20 mobile robots are directly connected to their relevant RBCs, all joint motors of the H20 robot arms are also directly controlled by their corresponding RACs, while all robot higher control functions are controlled by the RRC; (c) the RRC communicates with the other higher systems (e.g. PMS) to let the mobile robot based transportation system be integrated in the whole automated process of laboratory environments; (d) the RBCs will execute the distributed transportation tasks from the RRC by driving the right H20 mobile robots. When the RBCs find the robots reach the starting and destination positions of the transportation, they will activate the corresponding RACs to execute the arm moving activities; (e) all of important flow data (including the robot key data and the transportation status) are stored in a MySQL based remote database.

The workflow of the developed system (software) is demonstrated in Fig. 9.2, which can be explained as follows: (a) Configure a path with a series of waypoints for every mobile robot by using the RBC. Those paths record all the expected positions in laboratories for all available robots; (b) Start the RRC to connect all mobile robots' RRCs to establish TCP/IP communication channels. Once the connections between

the RRC and the RBCs are available, all of the robot hardware data and the prepared maps will be sent to the RRC from every RBC automatically; (c) Once the RRC receives all required data, it will do the path planning calculation for every robot based on its built waypoint maps; (d) At the same time the RRC will listen for any task from the PMS. Once the RRC receives a transportation task from the PMS, it will choose the best mobile robot by checking their real-time indoor positions and powers then give it the best path (the shortest path) to finish the transportation task. The criterions of selecting robots in the RRC include two kinds: the distance and the power; (e) When any running RBC receives a distributed path with a series of waypoint numbers, it will control the corresponding mobile robot's motion module to move; and (f) When a mobile robot reaches an expected position controlled by the selected RBC, the RBC will send a command to the arm control module inside the robot to catch up or put down an experimental facility based on a XML-based control file. Currently, the XML-based control files are generated by using a training arm from Canada DrRobot company without the complex computation of robot arm kinematics. In the future, for some high-precision required arm manipulation in the laboratory transportation, the functions (components) of arm kinematic calculation and image processing based identification will be included.

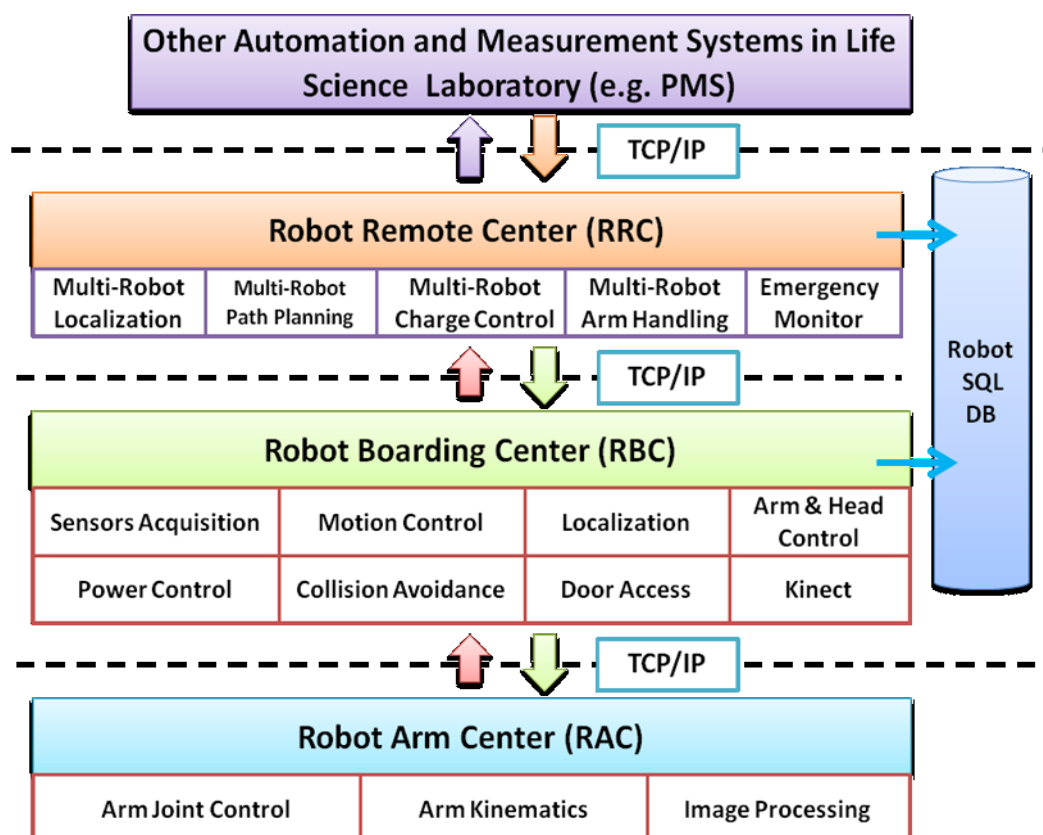


Figure 9.1: Software structure

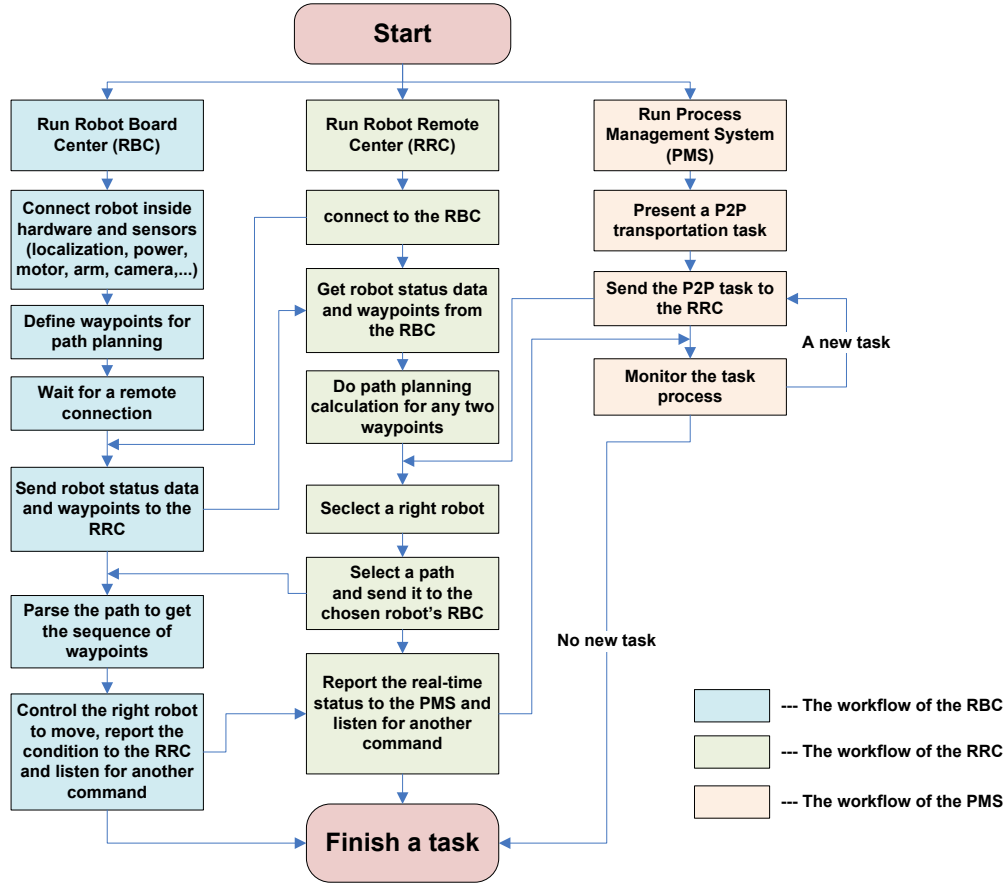


Figure 9.2: Software workflow

### 9.2.1 Robot On-board Center (RBC)

The RBC controls all hardware modules inside the mobile robots. As described in Chapter 2, the RBCs listen for commands from the RRC and drive the corresponding mobile robots to execute the received commands. The hardware modules in the RBC include indoor localization, motion control, head & arm, monitoring cameras, obstacle measurement sensors, etc. The tasks of the RBCs can be concluded as follows:

- **Communication:** In the RBC level an internal network will be built for the data acquisition between robotic on-board PCs and inside hardware modules.
- **Acquisition:** The RBC will measure the data of the robot hardware, including robot indoor positions, current powers, real-time images, ultrasonic data, etc.
- **Control:** Once a RBC receives a transportation command and a selected transportation path, it will control the corresponding mobile robot to execute them. Additional the RBC also needs to open the doors in laboratories.

As given in Section 2.1, in this study the H20 mobile robots are adopted to demonstrate the performance of this transportation application. In the programming process to develop the RBCs for the H20 robots, there are several supporting



components and libraries from the DrRobot company and robot hardware producers which have been used in the software development of the RBCs as follows:

- **VitaminControl.dll:** This is the control component for the main camera in the H20 robots. It will be used to realize the camera operations (i.e., connect, read, save, etc).
- **AXIS Media Control Library Set:** This is the control component for the eye and arm cameras in the H20 robots. The functions are similar to those of the Vitamin cameras.
- **DRROBOTSentinelCONTROL.OCX:** This is the control component for the hardware sensors (including motors, ultrasonic sensors, infra radio sensors and head & arm servo sensors) which are integrated in the H20 robots.

### 9.2.1.1 Communication

In a RBC, the issue of communication includes two classifications: one is to use the TCP/IP socket to build a data channel between the RRC and the RBC, the other one is to measure the robot inside hardware data for the RBCs. There are three TCP/IP sockets in every RBC as shown in Fig. 9.3. Our designed methods for those three sockets are given in Fig. 9.4. Among the three sockets, the robot and path sockets work as a socket server and the arm socket serves as a socket client. The relationship among those three communication sockets is demonstrated in Fig. 9.5.

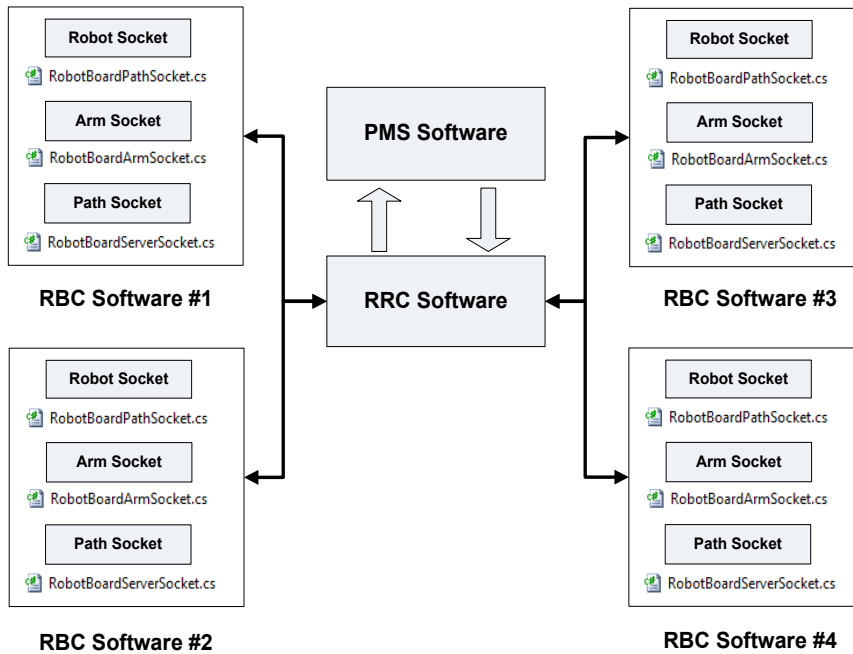


Figure 9.3: Architecture of TCP/IP sockets in the RRC

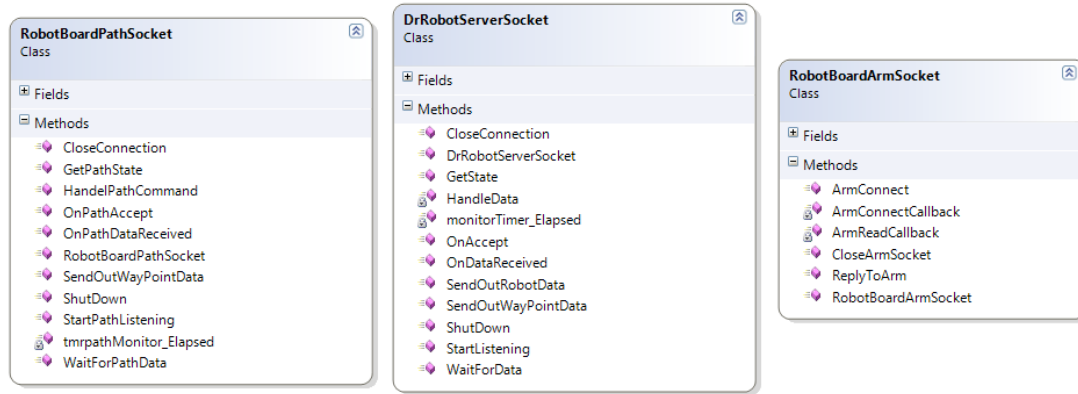


Figure 9.4: Designed methods for socket communication classes in the RBC

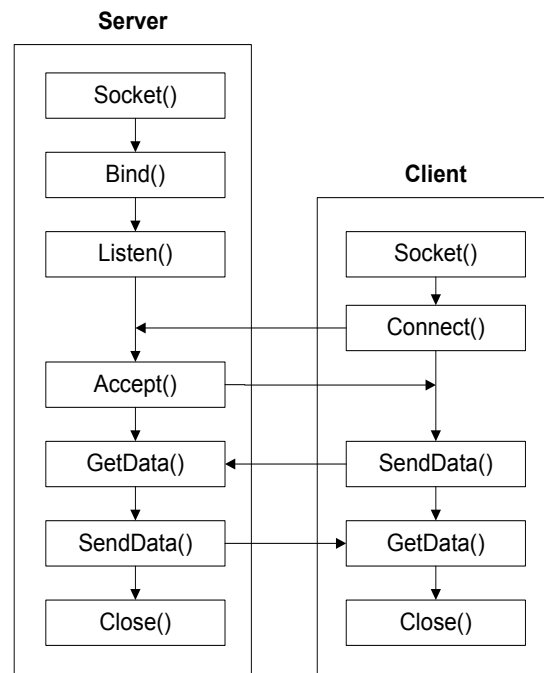


Figure 9.5: Workflow of sockets in the RBC

The architecture of the communication between the robot on-board PCs and the robot inside hardware modules is given in Fig. 9.6. From Fig. 9.6, it can be found that besides some necessary COM (Component Object Model) transmission among the low hardware (e.g. motor controllers) all other communication is based on the TCP/IP protocol. This kind of architecture will be convenient to add a new component (e.g., adding a Microsoft Kinect sensor for the obstacle measurement).

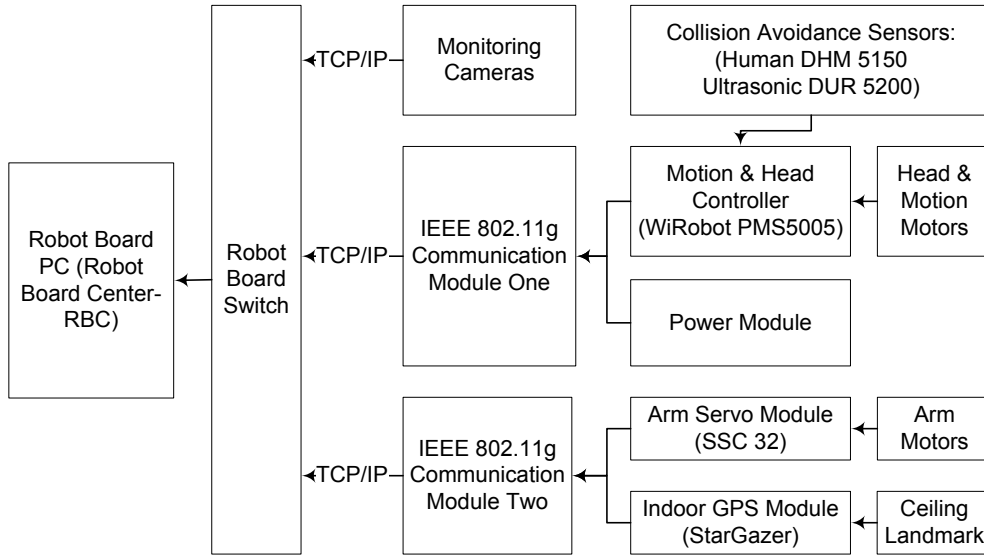


Figure 9.6: Architecture of robot hardware communication in the RRC

### 9.2.1.2 Data Acquisition

As shown in Fig. 9.6, a series of robot hardware data are measured by the RBC. Those data include hardware control commands and sensing data. The former one is mainly for the controls of motion modules (*see Fig. 9.7*). The latter one is for the real-time samplings of kinds of sensors (*see Fig. 9.8*).

```
private void motionControl_MotorSensorEvent(object sender, EventArgs e)
{
    //here reset motion control watchdog count
    leftMotor.encoderPos = motionControl.GetEncoderPulse1();
    leftMotor.encoderSpeed = (motionControl.GetEncoderSpeed1());
    leftMotor.current = 0;
    leftMotor.pwmValue = Math.Abs (motionControl.GetMotorPWMValue1() - 16384);

    rightMotor.encoderPos = motionControl.GetEncoderPulse2();
    rightMotor.encoderSpeed = (motionControl.GetEncoderSpeed2());
    rightMotor.current = 0;
    rightMotor.pwmValue = Math.Abs (motionControl.GetMotorPWMValue2() - 16384);
}
```

Figure 9.7: Acquisition function of motion modules

```
private void motionControl_StandardSensorEvent(object sender, EventArgs e)
{
    sensorData.UsDis[0] = (double)motionControl.GetSensorSonar1()/100;
    sensorData.UsDis[1] = (double)motionControl.GetSensorSonar2()/100;
    sensorData.UsDis[2] = (double)motionControl.GetSensorSonar3()/100;
    sensorData.UsDis[3] = (double)motionControl.GetSensorSonar4() / 100;
    sensorData.UsDis[4] = (double)motionControl.GetSensorSonar5() / 100;
    sensorData.UsDis[5] = (double)motionControl.GetSensorSonar6() / 100;
}
```

Figure 9.8: Acquisition function of ultrasonic sensors

### 9.2.1.3 Motion Control

In the H20 mobile robots, there are four options to control robot inside motors, the Pulse Width Modulation (PWM) control, the velocity control, the distance control and the rotate control. In this study all of them are used in the RBC development. Actually they work together with the classes of robot indoor positioning (localization). Here the function API of the motion distance control is shown in Table 9.1.

Table 9.1 Function API of motion distance control

<p><b>Function:</b>  Void DcMotorPositionTimeCtrALL(short cmd1, short cmd2, short cmd3, short cmd4, short cmd5, short cmd6, short timePeriod)</p> <p><b>Description:</b>  This function sends the position (distance) commands to all 6 DC motor control channels on the sensing and motion controller (PMS 5005) at the same time. The commands includes the target positions and the time period to execute the command. The current trajectory planning method for time control is linear.</p> <p><b>Parameters:</b>  short cmd1: Target position for motor channel#1  short cmd2: Target position for motor channel#2  short cmd3: Target position for motor channel#3  short cmd4: Target position for motor channel#4  short cmd5: Target position for motor channel#5  short cmd6: Target position for motor channel#6  short timePeriod: executing time in milliseconds</p> <p><b>Return value:</b> void</p>
--

### 9.2.1.4 Power Control

#### (1) Robot batteries

In Chapter 8, the topic of the transportation charging management has already been explained. Here the APIs of the robot charging control will be focuses. In every H20 mobile robot, there are two groups of on-board batteries. As shown in Fig. 9.9, there are two APIs in the power class of the RBC for the robot charging control, which includes the one for battery voltage control and the one for battery temperature measurement.

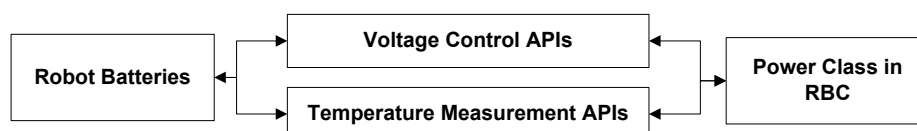


Figure 9.9: Architecture of robot charging measurement

## (2) Robot PCs

The RBCs are expected to run in the robot on-board laptops. So besides the robot battery status, the batteries of the PCs are also need to be measured (*see Fig. 9.10*). As described in Chapter 8, the power status of the PCs is included in the management of the robot charging.

```
private void tmrPCBatteryCheck_Tick(object sender, EventArgs e)
{
    pcBatteryStatus = SystemInformation.PowerStatus;
    pcBatteryPercent = pcBatteryStatus.BatteryLifePercent;
    lblBatteryPercent.Text = (pcBatteryPercent * 100).ToString() + "%";
}
```

Figure 9.10: Acquisition function of laptop batteries

### 9.2.1.5 Developed GUIs of RBC

The GUIs of the RBC are mainly comprised of an IP configuration GUI, a sensor monitoring GUI, a waypoint definition GUI, a waypoint edition GUI and a socket monitoring GUI.

The GUI of the IP configuration is given in Fig. 9.11. As explained in Chapter 2, in this application all H20 mobile robots and their hardware components have an independent IP address. This GUI is designed to build the connection between the robot on-board PCs and the robot inside hardware modules. As shown in Fig. 9.11, the RBC of a mobile robot named “1A” is being activated. Once the ID (name) of a mobile robot or the IP addresses of its inside two routers are changed, this GUI will save the new configuration into a XML recoding file automatically (*see Fig. 9.12*).

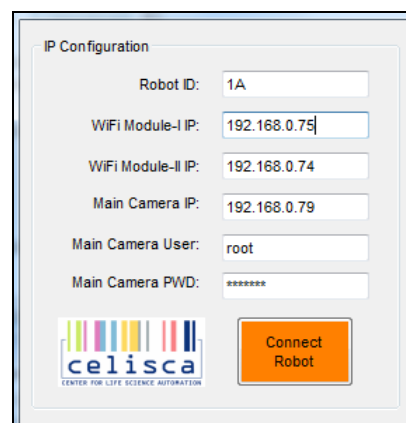


Figure 9.11: IP configuration GUI

```
<?xml version="1.0" standalone="yes"?>
<RobotConfig xmlns="http://tempuri.org/RobotConfig.xsd">
  <RobotConfigTable>
    <RobotID>DrRobot</RobotID>
    <RobotIP>192.168.0.75</RobotIP>
    <Port1>10001</Port1>
    <Port2>10002</Port2>
    <GPSIP>192.168.0.74</GPSIP>
    <GPSPort>10002</GPSPort>
    <CameraIP>192.168.0.79</CameraIP>
    <CameraPort>8081</CameraPort>
    <CameraPWD>drrobot</CameraPWD>
    <CameraUser>root</CameraUser>
  </RobotConfigTable>
</RobotConfig>
```

Figure 9.12: XML for IP configuration

The GUI of the robot sensor monitoring is shown in Fig. 9.13. From Fig. 9.13, it can be seen that all robot inside hardware modules are being measured and displayed in this GUI. Those displaying data include: (a) real-time camera images; (b) ultrasonic and infra radio sensors; (c) power status (including robot on-board batteries and PCs); (d) indoor localization (including X, Y and direction); and (e) motor sensors (including encoder position, speed and PWM value). Actually in a real transportation task those RBCs work in a fully-automation way without any external operations. This GUI is only designed for the users to monitor the robot current performance.

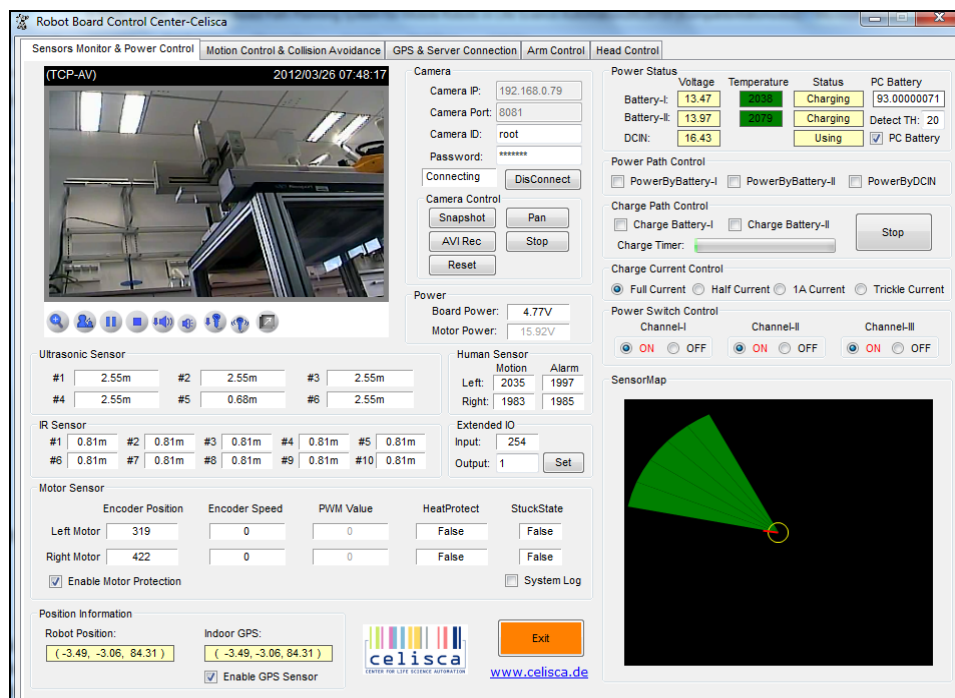


Figure 9.13: Sensor monitoring GUI

The GUI of the robot waypoint definition is shown in Fig. 9.14. From Fig. 9.14, it can be found that: (a) a real transportation path is organized using a series of waypoints. In the current GUI, there are five expected waypoints identified for a transportation

map. In every waypoint, all necessary parameters to execute a transportation task are defined in advance, including the moving velocities between every two positions, the suspending time of every waypoint, the moving mode (forward or backward) between every two positions, etc. It means the RRC only needs to tell a RBC which numbers of waypoints should be executed but it is unnecessary to send it the detailed control parameters. For the RBC side, once it receives a sequence of waypoint numbers, it will know the control parameters of all waypoints by reading the related defined file; (b) there is a programmed 2D map in the GUI for monitoring the real-time movements of the selected robots.

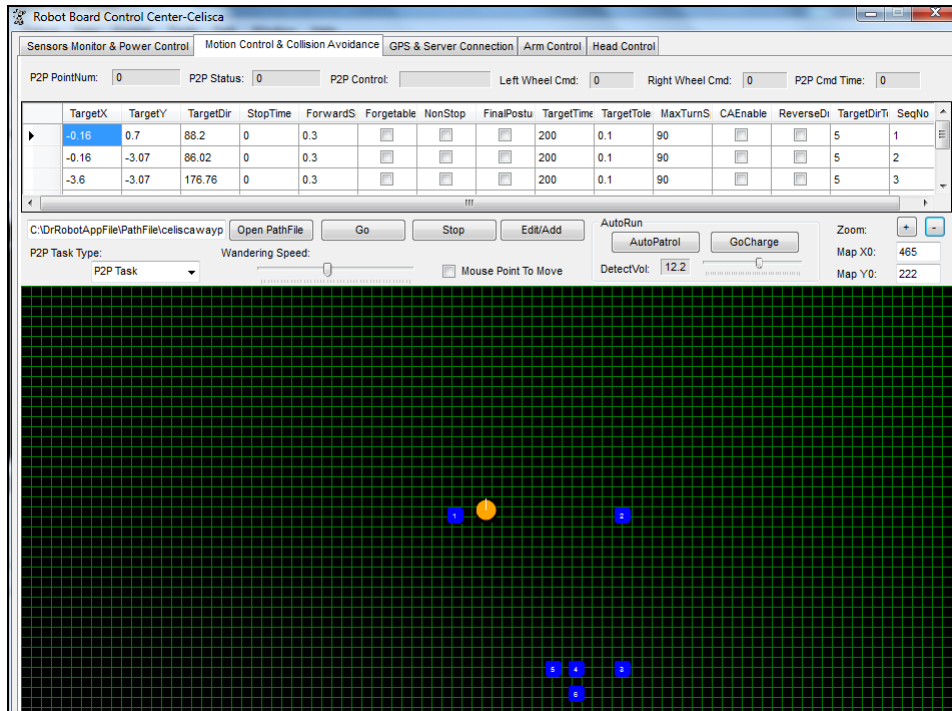


Figure 9.14: Waypoint definition GUI

The GUI of the robot waypoint edition is shown in Fig. 9.15. As shown in Fig. 9.15, a waypoint can be added, deleted and edited based on the real-time localization data from the robot on-board StarGazer module. The users do not have to input the position information manually, they only need to move the robots to those potential waypoint positions and operate this GUI, then all information will be recorded into a XML file automatically as shown in Fig. 9.16. As demonstrated in Fig. 9.16, actually not only the parameters of the robot localization but also the parameters of the motion control are written into this configuration file automatically. As explained in Section 9.2, the localization data will be used for the RRC to do path planning computation and the motion data will be utilized to execute a P2P movement.

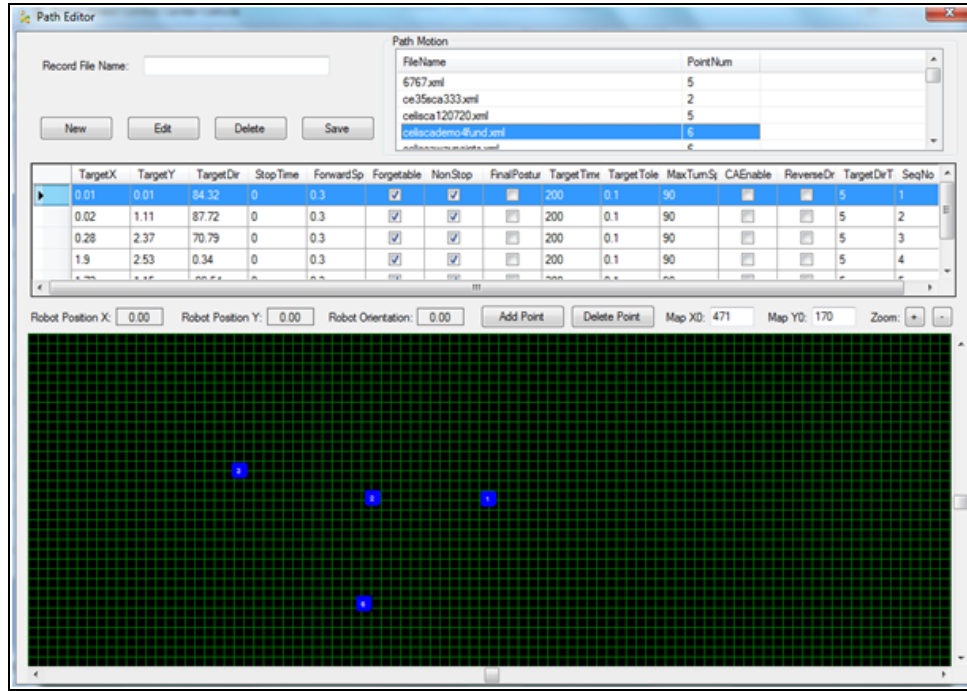


Figure 9.15: Waypoint edition GUI

```

<PointConfigTable>
  <TargetX>-0.16</TargetX>
  <TargetY>0.7</TargetY>
  <TargetDir>88.2</TargetDir>
  <StopTime>0</StopTime>
  <ForwardSpeed>0.3</ForwardSpeed>
  <Forgetable>>false</Forgetable>
  <NonStop>>false</NonStop>
  <FinalPosture>>false</FinalPosture>
  <TargetTime>200</TargetTime>
  <TargetTolerance>0.1</TargetTolerance>
  <MaxTurnSpeed>90</MaxTurnSpeed>
  <CAEnable>>false</CAEnable>
  <ReverseDrive>>false</ReverseDrive>
  <TargetDirTolerance>5</TargetDirTolerance>
  <SeqNo>1</SeqNo>
</PointConfigTable>

```

Figure 9.16: XML for waypoint definition

The GUIs of the TCP/IP communication sockets in the RRC and the RBC are given in Figures 9.17 and 9.18, respectively. From Figures 9.17 and 9.18, we can see that: (a) the socket in the RRC side services as a TCP client and the one in the RBC side works as a TCP server. As shown in those two GUIs, they recognize and connect each other rightly; (b) the connected IP address and ports in both sides are detected and shown in this GUI. This function is useful for the users to know the network status of the system; and (c) after the path planning computation made by the RRC, the selected transportation path and its whole distance will be received and displayed in the RBC of a chosen mobile robot.



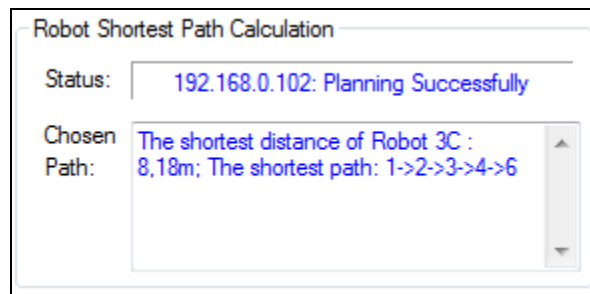


Figure 9.17: Socket monitoring GUI in the RRC to receive the robot sensor data and send the selected transportation path

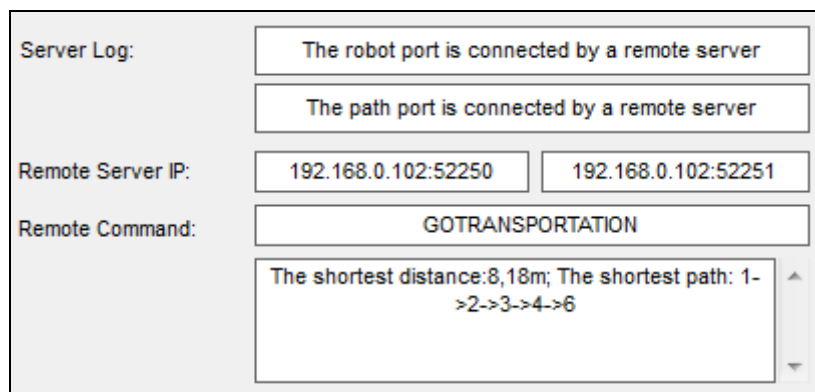


Figure 9.18: Socket monitoring GUI in the RBC to send the robot sensor data and receive the given transportation path

The GUIs of the TCP/IP communication sockets in the RBC and the RAC are given in Figures 9.19 and 9.20, respectively. From Figures 9.19 and 9.20, we can see that: (a) those two sockets recognize and connect each other rightly; (b) the connected IP address and ports are also detected and shown in this GUI; and (c) the function of arm control can be activated or closed by the RRC automatically or by the users manually.

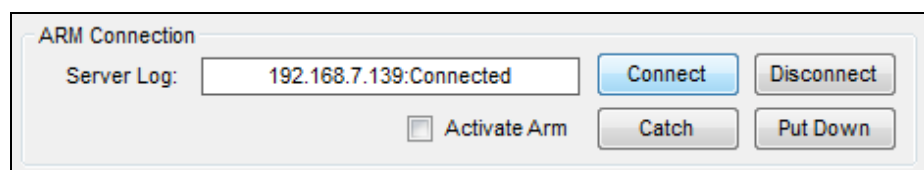


Figure 9.19: Socket monitoring GUI in the RRC to send the chosen arm control commands

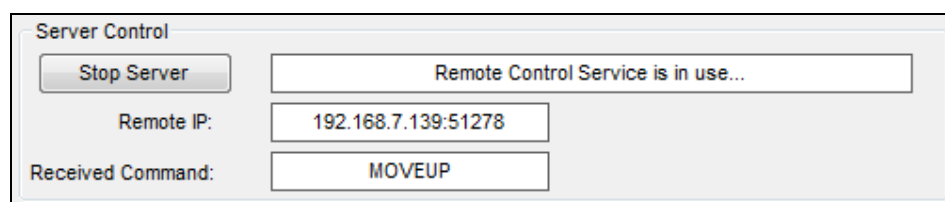


Figure 9.20: Socket monitoring GUI in the RAC to receive the given arm control commands

### 9.2.2 Robot Arm Component (RAC)

The RAC is a component of the RBC to manage the robot arm control. The functions of the RAC can be presented as follows:

- **Communication:** There is a TCP/IP communication socket between the RBC and the RAC, which will be utilized to transfer kinds of arm control commands to the robot arm servo hardware. When the RBC controls the corresponding mobile robot arriving at the expected positions (including the starting position and the destination position) rightly, the RBC will send a command to its related RAC to drive a corresponding arm movement. Once the RAC finishes a given arm command, it will notice the RBC to move to the next position.
- **Mapping:** To avoid transmitting the huge data between the RBC and the corresponding RAC, a mapping method is adopted in the RAC. The working steps of this mapping method for the arm control are explained as follows: (a) using the training arm to generate a XML-based control file which records all joint values of a movement. This training way can avoid calculating the complicated kinematic equations of the robot arms; (b) When a RBC wants to drive an arm movement, it only needs to tell its connecting RAC the XML file name of the expected arm movement and the kind of movements (catching up or putting down); and (c) Once the RAC receives a arm file name, it will start to search for it from the prepared library then move the robot arms using the arm joint values in the searched file.

#### 9.2.2.1 Communication

In a RAC there are two kinds of TCP/IP sockets, which are proposed to communicate with the robot arm servo module and the RBC. The relationship between them is demonstrated in Fig. 9.21.

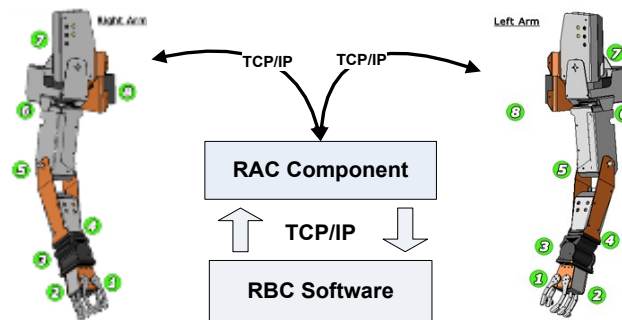


Figure 9.21: Architecture of TCP/IP sockets in the RAC

### 9.2.2.2 Arm Joint Mapping

The process of the arm joint mapping includes three steps: (a) generating a XML-based control file for an expected arm movement; (b) selecting the control file based on the received file name; and (c) loading the joint values of the selected control file to the right motors of the robot arms. The GUI of the RAC is given in Fig. 9.22. From Fig. 9.22, it can be seen that: (a) the GUI of the RAC is controlling the arm hardware to execute a predefined movement based on the received commands from the RBC; (b) all the real-time moving status is displayed in this GUI, including the dynamic values of the arm joints, the connection between the RBC and the RAC and the identified XML files. Additional based on this GUI, the users can pause or cancel an arm movement manually any time if they want.

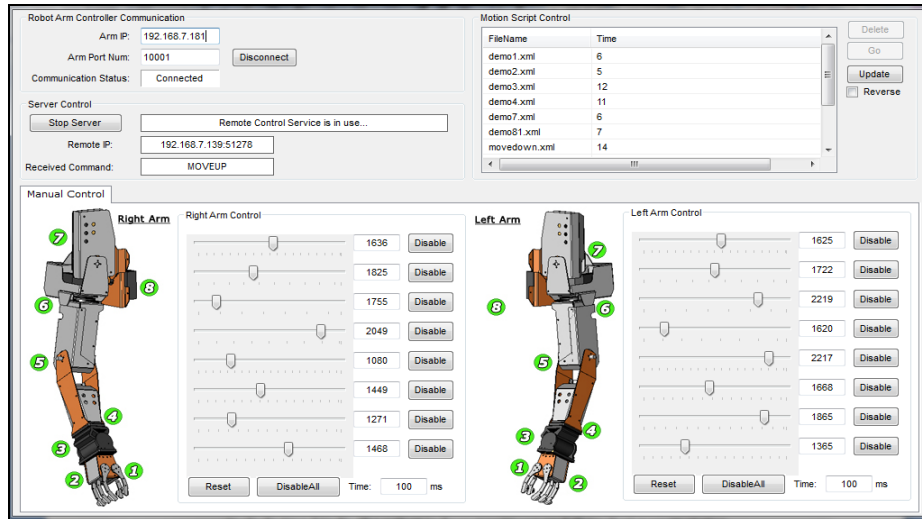


Figure 9.22: GUI of the RAC

### 9.2.3 Robot Remote Center (RRC)

The RRC does higher control of all utilized mobile robots. The ideas to design a RRC are demonstrated as follows:

- **Communication:** The RRC will communicate with both of the PMS and the RBCs. The TCP/IP protocol is adopted to realize this function. To the PMS, the RRC is a TCP/IP Server which listens for possible PMS connections. To the RBCs, the RRC is a TCP/IP Client which connects the RBCs actively if necessary. The RRC can add any kinds of mobile robots into the existent system after building a TCP/IP socket connection.
- **Acquisition:** The RRC will sample all necessary key robot parameters from all running RBCs for the multi-robot remote control. The key robot parameters include two kinds, the robot indoor positions and the robot power status. The

data format of those filtered data from the RBCs to the RRC is explained in Appendix B.2.

- **Mapping:** In the future different kinds of mobile robots will be supplied to execute the laboratory transportation. Those mobile robots could have various working mechanism. For example, some of them do indoor localization by the ceiling landmarks and others may use the computer vision to calculate their indoor positions. So the RRC should map the PSM transportation commands to the suitable commands, which could be recognized by this kind of mobile robots.
- **Path:** To do effective transportation tasks, the path planning is crucial. When an amount of mobile robots are running at the same time, their transportation paths and cooperation should be prepared in advance in the RRC remote side. As described in Chapters 6 and 7, in this study a hybrid path planning framework based on the map theory and the artificial intelligence is proposed. Besides that, in this chapter some practice planning strategies will also be explained.

### 9.2.3.1 Communication

The TCP/IP protocol is adopted to do communication among the PMS, the RRC and the RBCs. In the RRC, the architecture of the designed TCP/IP sockets in the RRC side is demonstrated in Fig. 9.23. As shown in Fig. 9.23, it can be seen that the RRC has three independent TCP/IP sockets for system communication, the PMS socket, the path socket and the robot socket. Apparently, the PMS socket is used to communicate with the higher PMS systems, the path socket is defined for transmitting the path planning data between the RRC and the RBCs, and the robot socket is presented for transmitting the robot data between the RRC and the RBCs. All of those sockets are programmed by Microsoft C# language.

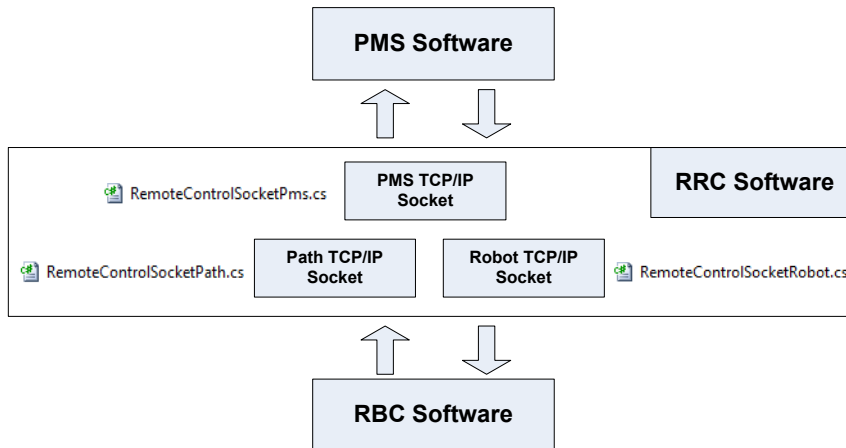


Figure 9.23: Architecture of TCP/IP sockets in the RRC

The proposed methods in those three socket classes in the RRC are given in Fig. 9.24. Taking the path socket for example, its workflow is demonstrated in Fig. 9.25.

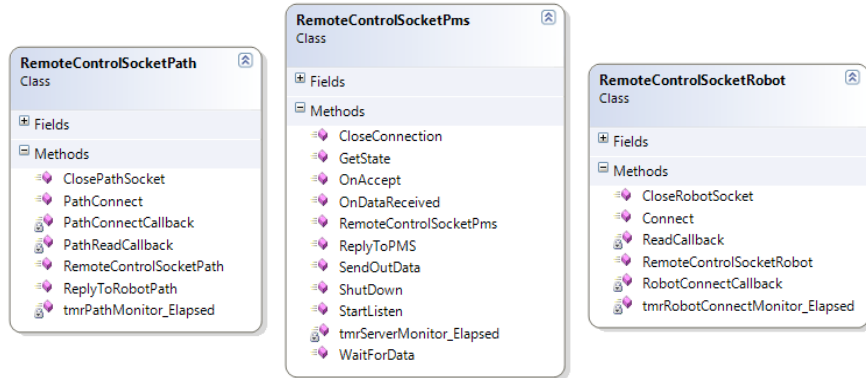


Figure 9.24: Programmed methods for socket communication classes in the RRC

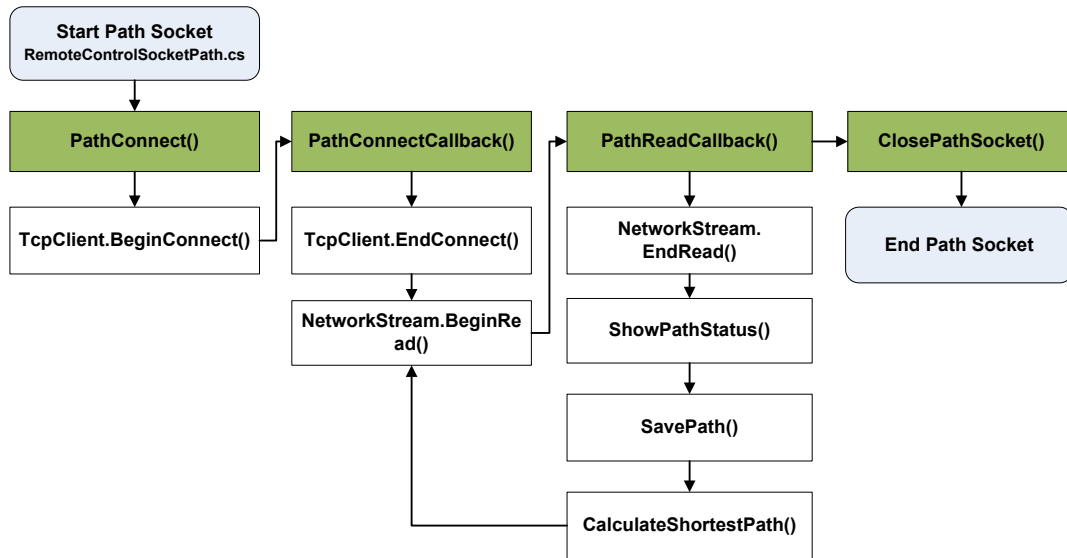


Figure 9.25: Workflow of the path socket in the RRC

### 9.2.3.2 Data Acquisition

In the RRC, it needs two kinds of data from the RBC side, the robot status data and the defined waypoints. Their data formats are demonstrated in Tables 9.2 and 9.3, respectively. As shown in Table 9.2, the robot status data are comprise of robot indoor positions, robot battery voltages & status (Charging or Using), robot on-board ultrasonic sensors and transportation executive status (P2P Going or P2P Stopping).

As described in Section 9.2, the RBC defines the transportation waypoints including their positions and control parameters. But it is unnecessary to send all defined data of the waypoints from the RBCs to the RRC, because only the position information will

be utilized to do transportation path planning. So every RBC will extract the position data and send them to the RRC (*see Table 9.3*).

Table 9.2 Data format of robot status measurement

```
public class RobotData
{
    public string robotID = null;
    public string robotStatus = null;
    public string robotPositionX = "0";
    public string robotPositionY = "0";
    public string robotPositionDir = "0";
    public string batteryOneVol = "0";
    public string batteryTwoVol = "0";
    public string dcVol = "0";
    public string robotBodyBasisSonar = "0";
    public string robotBodyFrontSonar = "0";
    public string robotBodyBackSonar = "0";
    public string robotLeftHumanSensor = null;
    public string robotRightHumanSensor = null;
    public string batteryOneStatus = null;
    public string batteryTwoStatus = null;
    public string robotWaypoint = null;
    public string robotP2PStatus = null;
}
```

Table 9.3 Data format of robot waypoint measurement

```
<PointConfigTable>
  <TargetX>0.88</TargetX>
  <TargetY>0.46</TargetY>
  <TargetDir>-94.31</TargetDir>
  <SeqNo>1</SeqNo>
</PointConfigTable>
<PointConfigTable>
  <TargetX>1.1</TargetX>
  <TargetY>1.71</TargetY>
  <TargetDir>83.55</TargetDir>
  <SeqNo>2</SeqNo>
</PointConfigTable>
```

Besides the real-time robot and the path data will be transmitted from the RBC to the RRC, there are also a series of control commands between them for the transportation organization. The control commands for the path planning are shown in Table 9.4.

Table 9.4 Format of path command

```
private class PathRemoteCommand
{
    public static string MONITOR = "MONITOR";
    public static string AUTORUN = "AUTORUN";
    public static string GOCHARGE = "GOCHARGE";
    public static string STOPROBOT = "STOPROBOT";
    public static string CLOSECONNECT = "CLOSECONNECT";
    public static string PAUSEROBOT = "PAUSEROBOT";
    public static string CONTINUEROBOT = "CONTINUEROBOT";
    public static string PATHPOINTS = "PATHPOINTS";
    public static string GOTRANSPORTATION = "GOTRANSPORTATION";
}
```

### 9.2.3.3 Path Planning Computation

The function of path planning is one of the most important works in the RRC development. Once the RRC connects a new mobile robot's RBC, it will send a command (such as the command *PATHPOINTS* demonstrated in Table 9.4) to get all defined waypoints from the RBC to do path planning. If the path planning for a mobile robot is successful, the RRC will save the results (the shortest paths for any two-waypoint) before receiving a new group of waypoints. The mobile robots with successful path planning steps will be allowed to do transportation tasks if their powers are also available. The detailed computational steps of the proposed hybrid path planning have been given in Chapters 6 and 7. This section focuses on the programmed workflow of the path planning.

The hybrid framework of the path planning can be organized in Fig. 9.26 as follows: (a) When the RRC connects a RBC, it will send relevant path commands (*see Table 9.4*) to get all defined waypoints by the RBC; (b) The RRC will do path planning computation for this RBC's mobile robot. Before calculating, it will make a check whether the number of waypoints is huge: if yes, the computational programming will execute the artificial intelligence based path planning framework, otherwise the map theory based one will be chosen; (c) the final results will be sent to the RBC from the RRC; and (d) there is also a computational mode switch which is designed to change the adopted methods if necessary.

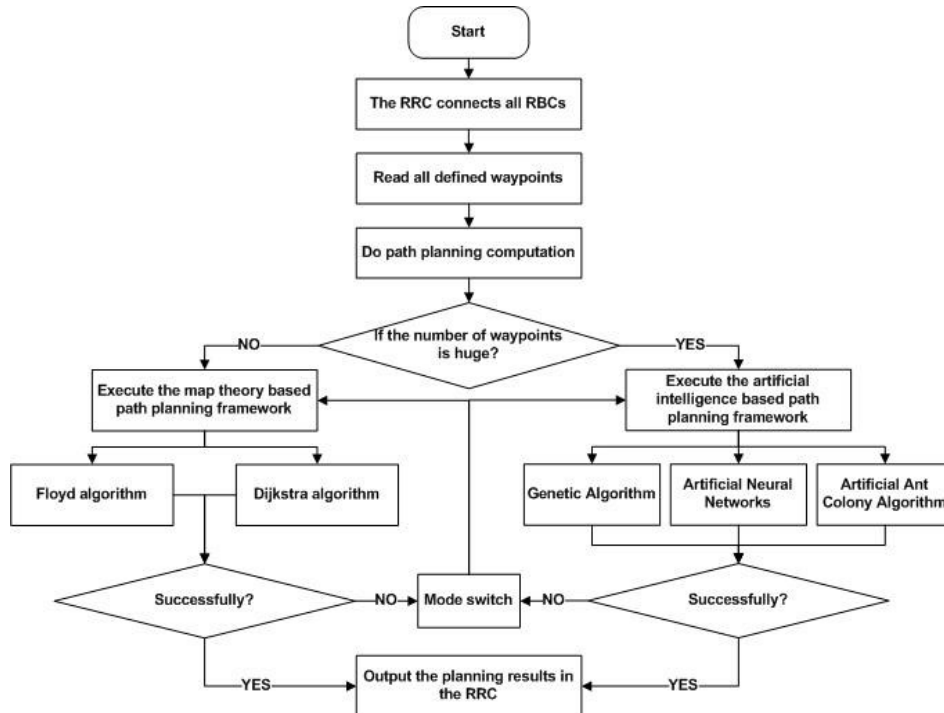


Figure 9.26: Framework of hybrid path planning

The reasons of using this hybrid framework can be further explained as: (a) as written in Chapter 6, the Floyd algorithm is expected to do off-line computation and the Dijkstra algorithm is presented to do on-line computation when the Floyd provided paths are not available. However, both of the Floyd and the Dijkstra algorithms are belonged to the map theory, which is not effective to handle with the maps with huge waypoints. In the huge cases, the map theory will cost considerable time to traverse all points; and (b) However in this application sometimes we needs the robot transportation tasks with huge waypoints in a short computational time. To let our developed software cope with the possible huge cases effectively, here the artificial intelligence based methods are supplied. The artificial intelligence based methods always do optimized calculation by using iterative steps. They can output the available computational results, which maybe are not the best. The detailed modeling process of the adopted intelligent algorithms can be referred to Chapters 6 and 7.

A case of the Floyd path planning API programmed in the RRC is demonstrated in Table. 9.5.

Table 9.5 Function API of Floyd path planning

<p><b>Function:</b>  <code>public bool RobotFloydCalculation(string waypoints)</code></p> <p><b>Description:</b>  This function returns the computational status of the Floyd shortest path planning for an expected serirs of waypointss.</p> <p><b>Parameters:</b>  The waypoints defined by the RBCs</p> <p><b>Return value:</b> bool  Return data interpretation:</p> <p>If the return is true, it means the Floyd calculation for those waypoints is successful and the results can be used for robot transportation; Otherwise the Floyd calculation is falied.</p>
---

### 9.2.3.4 PMS Command Mapping

In the RRC, there are lots of programming works on the robot status parsing. For example, the parsing work to map the higher PMS transportation commands to the sequence of robot executable waypoints, the parsing work to understand the XML-based robot control commands between the RRC and the RBCs, etc. Here those two representative parsing works are explained.

#### (1) Position matching

The RRC needs to match the received PMS commands into a series of pairs of origins and destinations for all running mobile robots. The reasons of this work are given as:



(a) all positions in the PMS commands are named using the real names of laboratories and facilities (this way is easy for laboratory staff), but in the RRC all waypoints have to be recorded in a series of real numbers because they will be used in the path planning calculation; and (b) it is impossible to build the same navigation map for various kinds of mobile robots because of their different working mechanism. It means the same PMS position could be recognized disparately by different mobile robots (*see Fig. 9.27*).

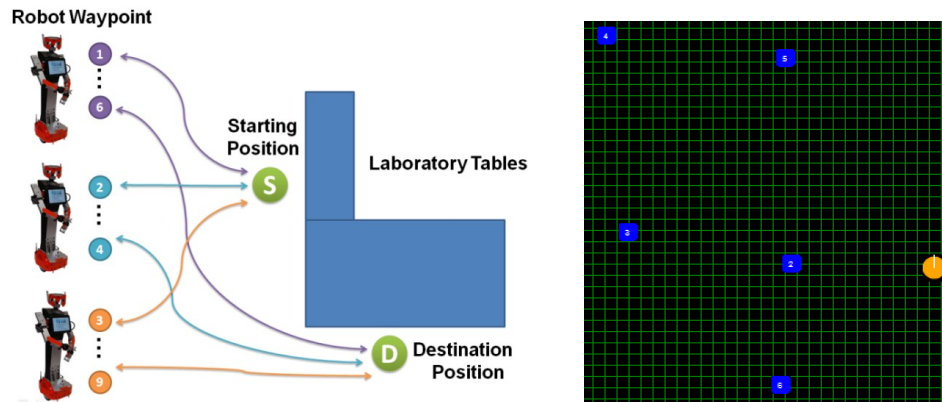


Figure 9.27: PMS position parsing: (a) matching process; and (b) a demo of built robot path with a series of waypoints

In this section an example is proposed to demonstrate the PMS parsing process in the RRC as follows:

(a) Suppose there is a PMS transportation command coming as shown in Fig. 9.28. This command is to ask the RRC to execute a transportation task in laboratories from the starting position (*S-Lab, Device A, Pos3*) to the destination position (*S-Lab, Device B, Pos2*);

Robots			
Name	Last Contact	Status	Version
1A	12.11.2012 14:07:28	Idle	1.0.0 Build 1
H20 Client for Test	12.11.2012 14:07:28	Idle	1.0.0 Build 1

Humans			
Name	Last Contact	Status	Version

Source Position		Destination Position	
Lab	S-Lab	Lab	S-Lab
Device	Device A	Device	Device B
Position	Pos3	Position	Pos2

Rack		Tick
Rack	R1	281

Server IP		Buttons	
Server IP	192.168.7.139:7030	Connect	Stop Continue Abort OK

Figure 9.28: GUI of the PMS

(b) Once the RRC receives this PMS transportation information, it will parse it to get the starting and ending positions then map the positions to different running robots' maps to understand which robots can recognize the starting and destination positions of this task. A case of parsing functions programmed in this application is given in Table 9.6.

Table 9.6 Function API of parsing the transportation resource in the PMS command

<p><b>Function:</b></p> <pre>private string MatchResourcePosition2Waypoint(string robotname, string pmsResourcePosition)</pre> <p><b>Description:</b> This function returns the waypoint ID which matches with the PMS resource position coordinates.</p> <p><b>Parameters:</b></p> <ul style="list-style-type: none"> <li>❖ <b>string</b> robotname: The ID of the mobile robot to be identified. In the programming, a loop will be provided to traverse all connected mobile robots;</li> <li>❖ <b>string</b> pmsResourcePosition: The RRC received PMS resource position;</li> </ul> <p><b>Return value:</b> string</p>
---

(c) Once the RRC finds all the robot candidates, it will select the best robot among them based on their power status and their distances to the starting transportation position. Then It will do the path planning for the best mobile robot and send it an executing path. A case of checking robot positions is demonstrated in Table 9.7.

Table 9.7 Function API of checking the robot positions for the transportation

<pre>private string[] CheckRobotPosition4PMSTrans (string[] robots, string pmsResourcePosition, string pmsDestinationPosition) {     int NUMOFROBOT = robots.Length;     int robotindex = 0;     string[] MatchRobot = new string[5];     //PMS command parsing     for (int i = 0; i &lt; NUMOFROBOT; i++)     {         if (robots[i] != null)         {             //check the starting point             string matchRobotStart = MatchResourcePosition2                 Waypoint(robots[i], pmsResourcePosition);             if (matchRobotStart != null)             {                 //check the ending point                 string matchRobotEnd = MatchDestinationPosition                     2Waypoint(robots[i], pmsDestinationPosition);                 if (matchRobotEnd != null)                 {                     MatchRobot[robotindex] = robots[i];                     robotindex = robotindex + 1;                 }             }         }     } }</pre>
---

```

    }
}
return MatchRobot;
}

```

## (2) PMS command parsing

Since the XML based commands are adopted between the PMS and the RRC, there is inevitably also a command parsing work in the RRC side similar to the process explained in the RBC side.

### 9.2.3.5 Developed GUIs of RRC

The GUIs of the RRC are comprised of a robot connection GUI, a sensor monitoring GUI, a path planning GUI and a PMS handling GUI. The GUI of the robot connection is shown in Fig. 9.29. As displayed in Fig. 9.29, the management strategy of the mobile robots in this study is based on IP address. A new mobile robot can be added into this platform after distributing an available IP address. To connect a mobile robot, the users only need to choose the right robot ID of this mobile robot and click the ‘connect’ button. If the two sockets (the robot socket and the path socket) of the mobile robot work rightly, the ‘connection status’ of the operating mobile robot in the GUI will show ‘green’.

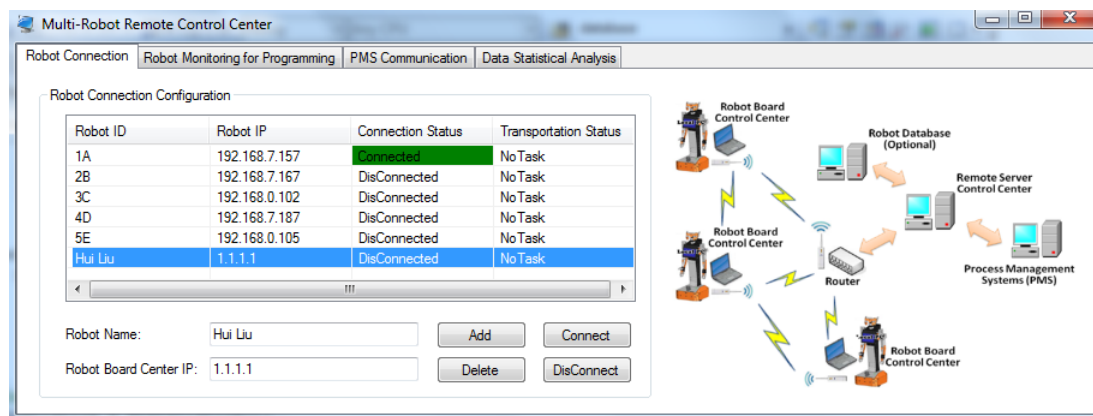


Figure 9.29: Robot connection GUI

The GUI of the sensor monitoring is shown in Fig. 9.30. If the users want to see the detailed real-time data of a connected mobile robot, they only need to double click its right robot ID to open the sensor monitoring GUI. The monitoring GUI will read the real-time key sensor data of the mobile robot by using its robot ID. Normally the key sensor data shown in this GUI will include robot position, robot battery power status and robot on-board sonar. The first and second kinds of sensing data are used to organize a transportation activity and the third group of data is for the function of

robotic collision avoidance. If the selected robot is carrying out a transportation task, the status of the transportation will be also displayed in this GUI, including passing waypoint number and robot motor working status (*'P2POver'/'P2PGO'*). Based on the monitoring GUI, the users can know the real-time performance of all connected mobile robots.

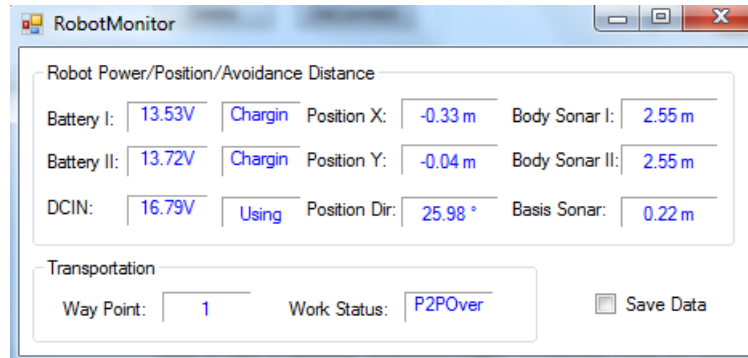


Figure 9.30: Sensor monitoring GUI

The GUI of the path planning is shown in Fig. 9.31. As shown in Fig. 9.31, this GUI has three main functions: (a) all of the received data from the connected mobile robots' RBCs are shown in the GUI. The users can save those received RBC data into a remote database or the local PC. As described in Chapter 1, the data in the robot database will be used to do statistical calculation for improving the performance of the mobile transportation; (b) the RRC works full-automatically without any human operations. To meet the possible requirements of manual operations, the GUI also has the manual buttons for the users; (c) the function of the path planning is the most important in this GUI. The completed computational process of the path planning adopted here can be further explained as: once the RRC connects a mobile robot's RBC, it will send a command to the RBC to sample all the defined waypoints. If the waypoint data of a mobile robot are gotten by the RRC, the RRC will execute the path planning function for this mobile robot at once. As described in Chapters 6 and 7, a hybrid computational strategy based on both of the map theory and the artificial intelligence is proposed for the robot complex path planning. The path planning results will also be displayed in this GUI.

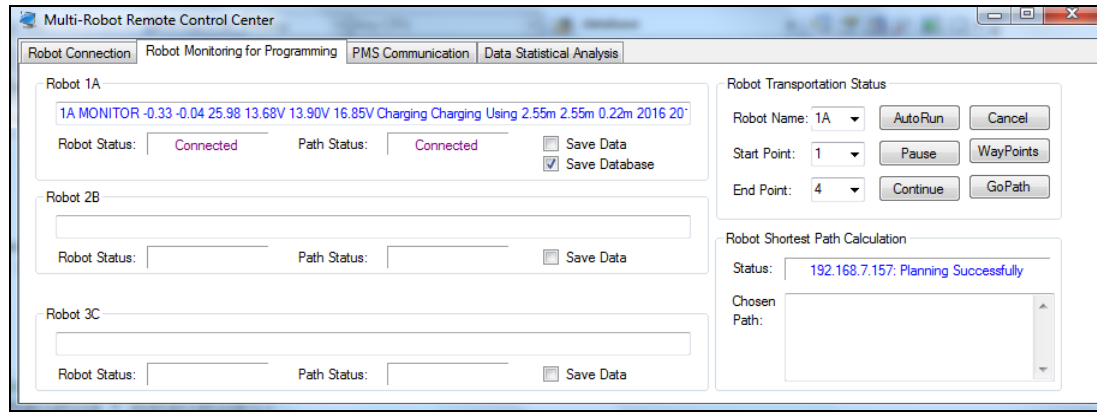


Figure 9.31: Path planning GUI

The GUI of the PMS handling is shown in Fig. 9.32. This GUI is designed to receive the control commands from the PMS. It will parse the received PMS commands for the organizations of the transportation tasks in the RRC and reply the executed results or the real-time performance of the transportation to the PMS if necessary. Also the PMS communication socket in the RRC will be monitored by this GUI. As shown in Fig. 9.32, a coming PMS command is parsed and the PMS socket is monitored. The definitions of the PMS commands will be found in details in Section 9.3.

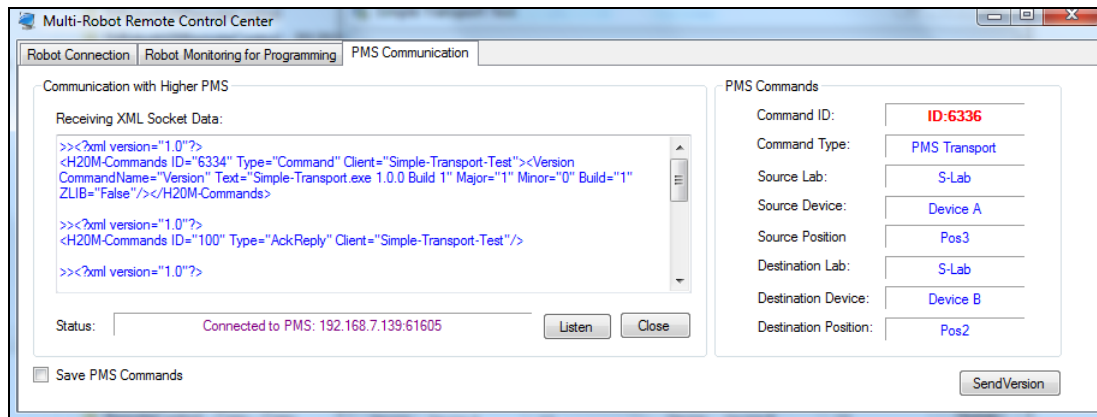


Figure 9.32: PMS handling GUI

## 9.3 Command Protocol

As mentioned in Sections 9.2 and 9.3, a XML-based protocol is proposed for the control commands among the RRC, the RBCs and the PMS. The advantages of using the XML in the system development have been summarized in references [152]–[154].

### 9.3.1 Commands between the RRC and the RBC

There is a series of designed XML-based commands for the mobile robot remote

control between the RRC and the RBCs as demonstrated in Fig. 9.33. The commands include: ‘Connect’, ‘GetWayPoint’, ‘ReplyWayPoint’, ‘RunPath’, ‘Pause’, ‘Continue’, ‘Cancel’, etc. The functions of those commands are explained in Appendix B.3.

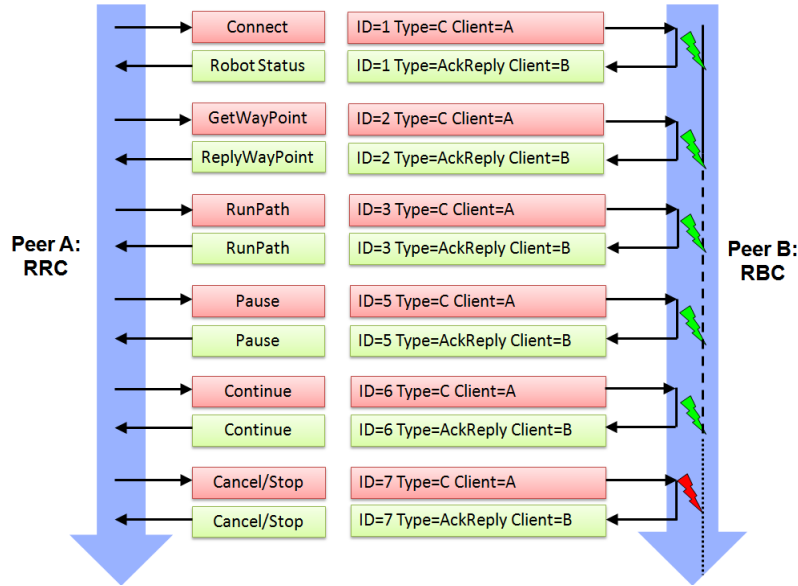


Figure 9.33: XML-based commands between the RRC and the RBC

A case of the ‘GetWayPoint’ command and its replied parameters is demonstrated in Fig. 9.34.

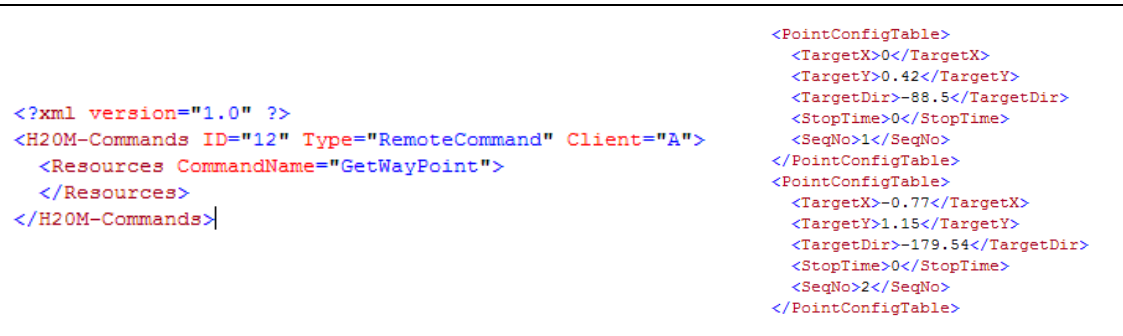


Figure 9.34: XML-based ‘GetWayPoint’ command and its replied data

Similarly, the commands for the RBCs to control/drive their corresponding RACs are given in Appendix B.4.

### 9.3.2 Commands between the PMS and the RRC

There is also a series of proposed XML-based commands for the PMS transportation communication between the RRC and the PMS as shown in Fig. 9.35. The workflow of those commands is demonstrated in Fig. 9.36.

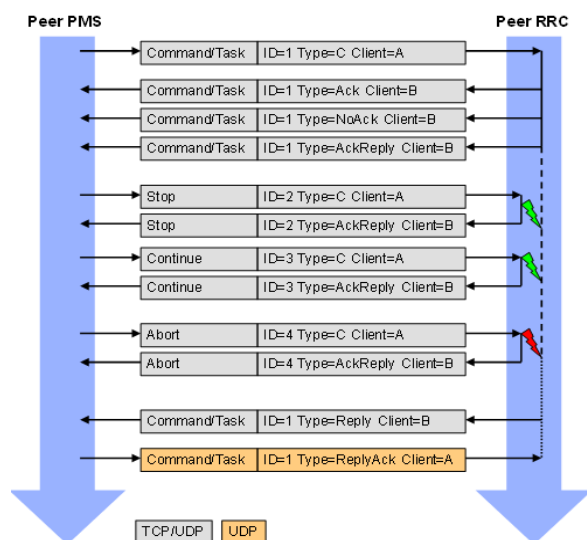


Figure 9.35: XML-based commands between the PMS and the RRC

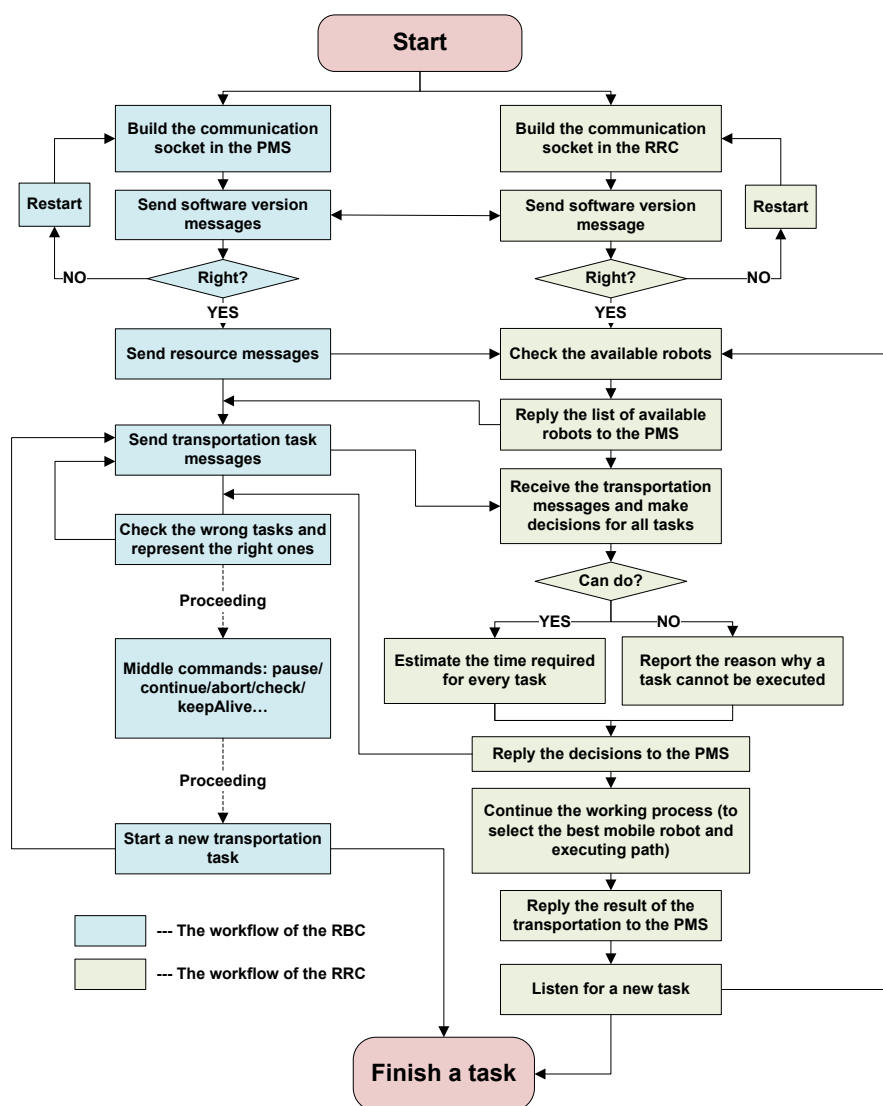


Figure 9.36: Command workflow between the PMS and the RRC

As shown in Fig. 9.36, there are two kinds of the most important control commands between the PMS and the RRC, the commands to describe the transportation contents and to request the lists of available mobile robots (transporters). Here they are further explained in details below.

### 9.3.2.1 Transportation Command

The transportation command is to transmit the specific contents of the transportation tasks using the PMS viewing. The PMS will present a transparent command including the starting position and the destination ending position, as demonstrated in Fig. 9.37. To guarantee the RRC receive the transportation command completed, the length of the command is also required to be included as given in Fig. 9.38.

```

1 <?xml version="1.0" ?>
2 <H2OM-Commands ID="88" Type="Command" Client="MessageTest">
3   <Transport CommandName="Transport">
4     <Source Lab="S-Lab" LabBC="X" Device="Device A" DeviceBC="Y" Position="Pos1" PositionBC="Z" />
5     <Destination Lab="D-Lab" LabBC="X" Device="Device B" DeviceBC="Y" Position="Pos1" PositionBC="Z" />
6     <Rack Name="Rack A" BC="RA1">
7       <PosData Pos="P1">
8         <LWData Name="Labw1" BC="BC 1" />
9         <LWData Name="LabwX-9" BC="BC 1-Tango-Emil" />
10      </PosData>
11    </Rack>
12  </Transport>
13 </H2OM-Commands>

```

Figure 9.37: XML-based transportation command

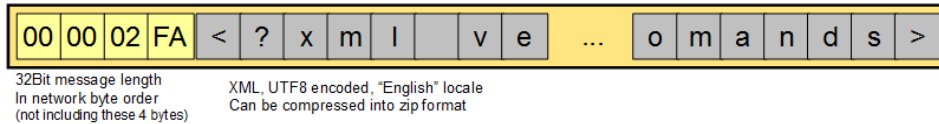


Figure 9.38: Format of transportation command

In the RRC side, when the RRC receives a coming transportation command from the PMS, it should reply the PMS whether the transportation can be further executed. The process of coping with the transportation commands can be explained as follows: (a) the positive condition is that the received command can be understood by the RRC. In that case, the RRC will go on selecting the best mobile robot among the available ones then calculate the needed time to finish this task based on the status of the chosen robot. This estimated time should be replied to the PMS timely (*see Fig. 9.39*); (b) the negative condition is that the received command cannot be recognized by any mobile robots, which are connected to the RRC. In that situation the RRC should report an error notice to the PMS (*see Fig. 9.40*). Then the PMS has the possibility to check this transportation command and represent a revised one; (c) once the distributed transportation command (task) has been executed rightly by a mobile robot, the RRC should also report a success notice to the PMS in time (*see Fig. 9.41*); and (d)



during the communicating process, the ‘*KeepAlive*’ command will be sent by the PMS frequently to the RRC. If the RRC does not reply this kind of ‘*KeepAlive*’ command more than several times (five time by default), the PMS will regard the socket communication to the RRC has been closed then it will restart it. Also if the PMS does not receive a response from the RRC about the result of a transportation task beyond the RRC’s promising time (estimated time), it will give up the transportation task and restart the procedure.

```
1 <?xml version="1.0" ?>
2 <H2OM-Commands ID="88" Type="Ack" Client="RRC">
3   <Transport CommandName="Transport" EstimateTime="1000" />
4 </H2OM-Commands>
```

Figure 9.39: XML-based transportation estimation time reply

```
1 <?xml version="1.0" ?>
2 <H2OM-Commands ID="88" Type="Ack" Client="RRC">
3   <Error Code="7">The transportation task cannot be understood<Error />
4 </H2OM-Commands>
```

Figure 9.40: XML-based transportation error reply

```
1 <?xml version="1.0" ?>
2 <H2OM-Commands ID="88" Type="Reply" Client="RRC">
3   <Transport CommandName="Transport" Success="True" ErrorText="TEXT"/>
4 </H2OM-Commands>
```

Figure 9.41: XML-based estimation time reply

### 9.3.2.2 Resource Command

The resource command is proposed for the PMS to query the information about available robots and the defined positions from the RRC. A case of the resource commands is displayed in Figures 9.42 and 9.43.

```
1 <?xml version="1.0" ?>
2 <H2OM-Commands ID="88" Type="Command" Client="RRC">
3   <Resources CommandName="List Transporter">
4   </Resources>
5 </H2OM-Commands>
```

Figure 9.42: XML-based resource command

```
<?xml version="1.0" ?>
- <H2OM-Commands ID="15725" Type="AckReply" Client="RRC">
- <Resources CommandName="List Transporter">
  <Transporter Name="1A" StatusAsText="Busy" StatusCode="2" />
  <Transporter Name="2B" StatusAsText="Busy" StatusCode="2" />
  <Transporter Name="3C" StatusAsText="Idle" StatusCode="1" />
  <Transporter Name="4D" StatusAsText="Busy" StatusCode="2" />
  <Transporter Name="5E" StatusAsText="Busy" StatusCode="2" />
</Resources>
</H2OM-Commands>
```

Figure 9.43: XML-based resource reply

## 9.4 Database Design

In the study, a network database named *H20RobotDB* based on My SQL platform is built to store important data of the transportation system. The storing parameters of the *H20RobotDB* are demonstrated in Appendix B.5. The functions of the *H20RobotDB* database are given as follows:

- **Estimate the transportation performance:** by using the stored transportation data, a kind of optimized works could be provided. For instance, the task distribution of every mobile robot and every defined waypoint can be calculated statistically for optimizing the installing positions of the automated charging stations in laboratories.
- **Provide real-time deciding assistances for the transportation:** as described in Section 9.3, when the RRC receives a transportation task from the PMS, it will estimate the necessary time to complete the task by checking the status of all connected robots. This estimated time is important in the transportation, because it can affect the whole automated process of a laboratory. Although the RRC can calculate the estimated time based on the route distance and the robot current moving velocity, the result is theoretical. Sometimes additional time may be required in the transportation if some accidents happen, such as the time needed for an activity of collision avoidance. So it would be better if the RRC can reply the PMS the estimated time based on both of the theoretical computation of the current robotic configuration and the statistical results of the stored real data.
- **Provide necessary assistances to the users:** based on the established database, the users can request the related parameters of the transportation system if they want. For example, they can generate a statistical drawing for the previous changing conditions of robot batteries.

Fig. 9.44 shows the main GUI of the database operating. The ADO.NET component is adopted to build the data connection between the RRC and the My SQL-based *H20RobotDB*. In this GUI, the users can read all saved transportation data from the *H20RobotDB* database. If the users want to see more detailed data, they can operate the data searching function as displayed in Fig. 9.45. As shown in Fig. 9.45, the data searching can be done using the request conditions of robot power range, robot name and transportation executing time. In addition, the users can use the data outputting function of this GUI to save the filtered data into an Excel-format file for a deeper analysis as demonstrated in Fig. 9.46.

Besides the upper standard functions of database searching, there is an advantageous function included in this section. The function is to do statistical calculation. As shown in Fig. 9.47, the functions of statistical calculation designed here include: (a) provide the average times of the same transportation paths for every running mobile robot; (b) provide the charging times of robot batteries; and (c) provide the costing times of robot batteries. This kind of information will be used to optimize the transportation works in the future.

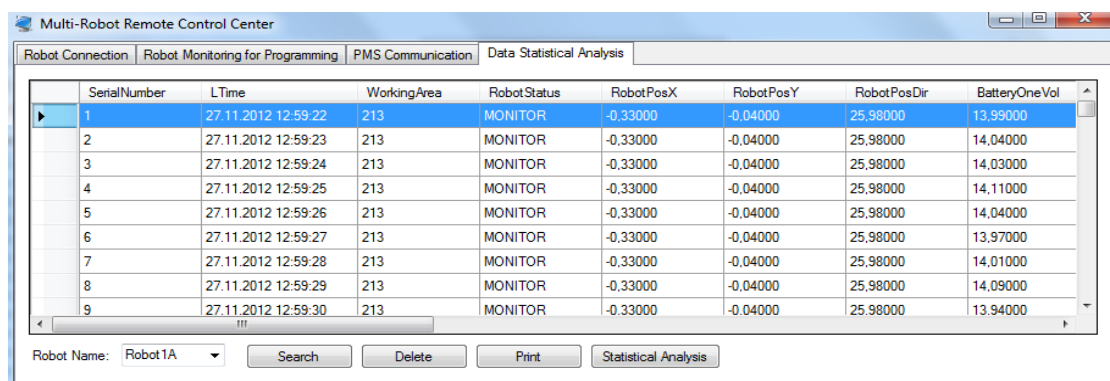


Figure 9.44: Database Main Operating GUI

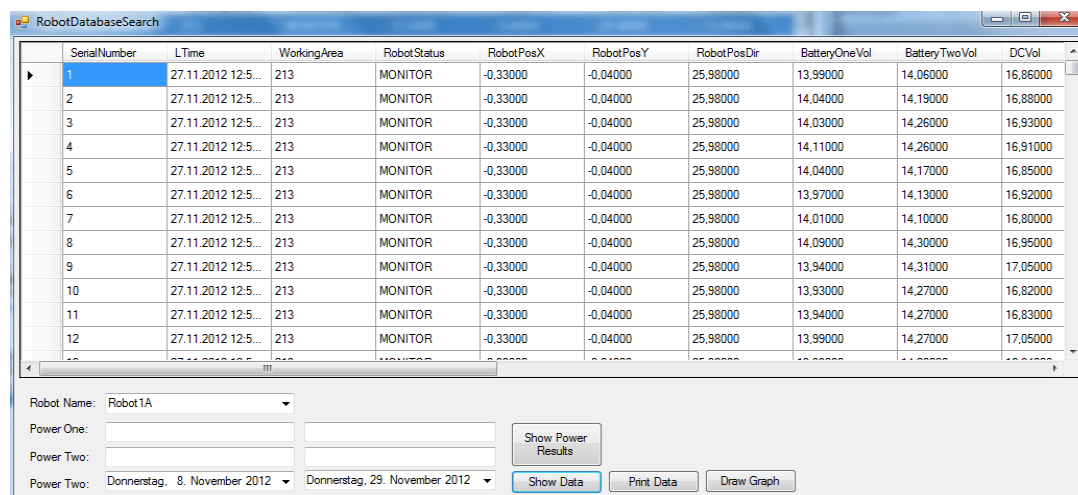


Figure 9.45: Database Searching GUI

SerialNumber	Time	WorkingArea	RobotStatus	RobotPosX	RobotPosY	RobotPosDir	BatteryOneV	BatteryTwoV	DCVol	BatteryOneS	BatteryTwoS	DCStatus	FrontSonaR	MiddleSonaR	BackSonaR
1	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	13,99	14,06	16,86	Charging	Charging	Using	2,55	2,55	0,2
2	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	14,04	14,19	16,88	Charging	Charging	Using	2,55	2,55	0,2
3	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	14,03	14,26	16,93	Charging	Charging	Using	2,55	2,55	0,2
4	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	14,11	14,26	16,91	Charging	Charging	Using	2,55	2,55	0,2
5	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	14,04	14,17	16,83	Charging	Charging	Using	2,55	2,55	0,2
6	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	13,97	14,13	16,92	Charging	Charging	Using	2,55	2,55	0,2
7	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	14,01	14,1	16,8	Charging	Charging	Using	2,55	2,55	0,2
8	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	14,09	14,3	16,95	Charging	Charging	Using	2,55	2,55	0,2
9	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	13,94	14,31	17,05	Charging	Charging	Using	2,55	2,55	0,2
10	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	13,93	14,27	16,82	Charging	Charging	Using	2,55	2,55	0,2
11	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	13,94	14,27	16,83	Charging	Charging	Using	2,55	2,55	0,2
12	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	13,99	14,27	17,05	Charging	Charging	Using	2,55	2,55	0,2
13	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	13,99	14,2	16,84	Charging	Charging	Using	2,55	2,55	0,2
14	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	13,97	14,28	17,01	Charging	Charging	Using	2,55	2,55	0,2
15	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	14,13	14,28	16,97	Charging	Charging	Using	2,55	2,55	0,2
16	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	14,13	14,11	16,91	Charging	Charging	Using	2,55	2,55	0,2
17	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	13,98	14,27	17,08	Charging	Charging	Using	2,55	2,55	0,2
18	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	14,05	14,21	16,93	Charging	Charging	Using	2,55	2,55	0,2
19	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	14,03	14,24	16,85	Charging	Charging	Using	2,55	2,55	0,2
20	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	14,05	14,27	16,98	Charging	Charging	Using	2,55	2,55	0,2
21	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	14,01	14,1	16,93	Charging	Charging	Using	2,55	2,55	0,2
22	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	13,94	14,24	17,05	Charging	Charging	Using	2,55	2,55	0,2
23	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	13,93	14,29	16,96	Charging	Charging	Using	2,55	2,55	0,2
24	27.11.2012 12:59	213	MONITOR	-0,33	-0,04	25,98	13,84	14,15	16,9	Charging	Charging	Using	2,55	2,55	0,2

Figure 9.46: Data Outputting GUI

Robot Names: Robot1A

Statistical Information

No.Of Tasks: 4

Avg. Battery One Voltage: 13,951061755

Avg. Battery One Charge Time: 1900,0000

Avg. Battery One Discharge Time: 815,0000

Avg. Battery Two Voltage: 14,130000000

Avg. Battery Two Charge Time: 3800,0000

Avg. Battery Two Discharge Time: 815,0000

Task	AvgTime
2-3-10	123
1-3-10	125
1-2-5-10	171
1-2-3	504

Figure 9.47: Statistical Calculation GUI

## 9.5 Conclusions

In this chapter the software development of the transportation system is presented in details. On the basis of the transportation control strategy proposed in Chapter 2, there are three software control centers designed in the system, the RRC, the RBCs and the RACs. In every control centers, the developing ideas are explained and the developed GUIs are demonstrated. Additional the control commands among those three centers and the established network database are also given.

## Chapter 10 Experiments

In this chapter, three different kinds of experiments are provided to demonstrate and verify the performance of the developed transportation system as follows: (a) robot arm movements; (b) robot transportation; and (c) robot charging management.

### 10.1 Robot Arm Movements

The first experiment is provided to demonstrate the performance of the robot arm movements. In this experiment a robot arm is activated to move up and put down a box using the programmed RAC software. Here a robot training arm from Canada DrRobot company is integrated in the RAC to generate the XML-based control files for the expected arm movements. The training arm will simultaneously control the real arms of H20 robots to do any kind of movements under the constraints of the arm joint freedoms. During the robot real arm moving, the trajectories of all joints in the H20 real arms will be recorded by the RAC software automatically. The working mechanism of this training arm method is demonstrated in Fig. 10.1.

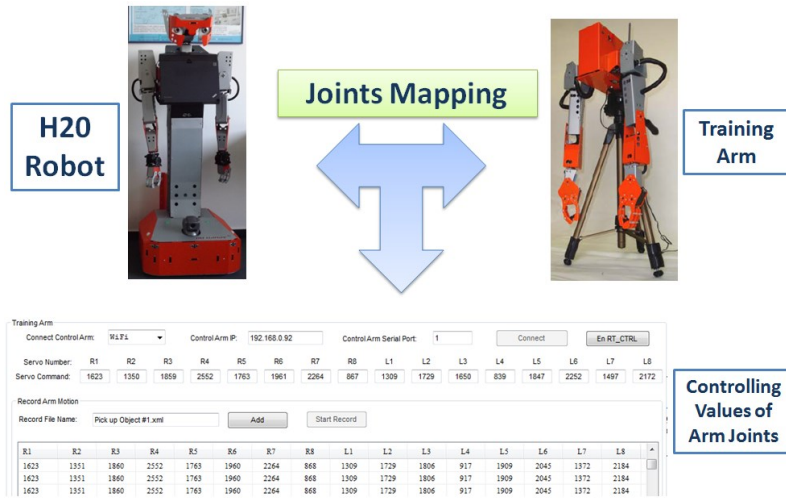
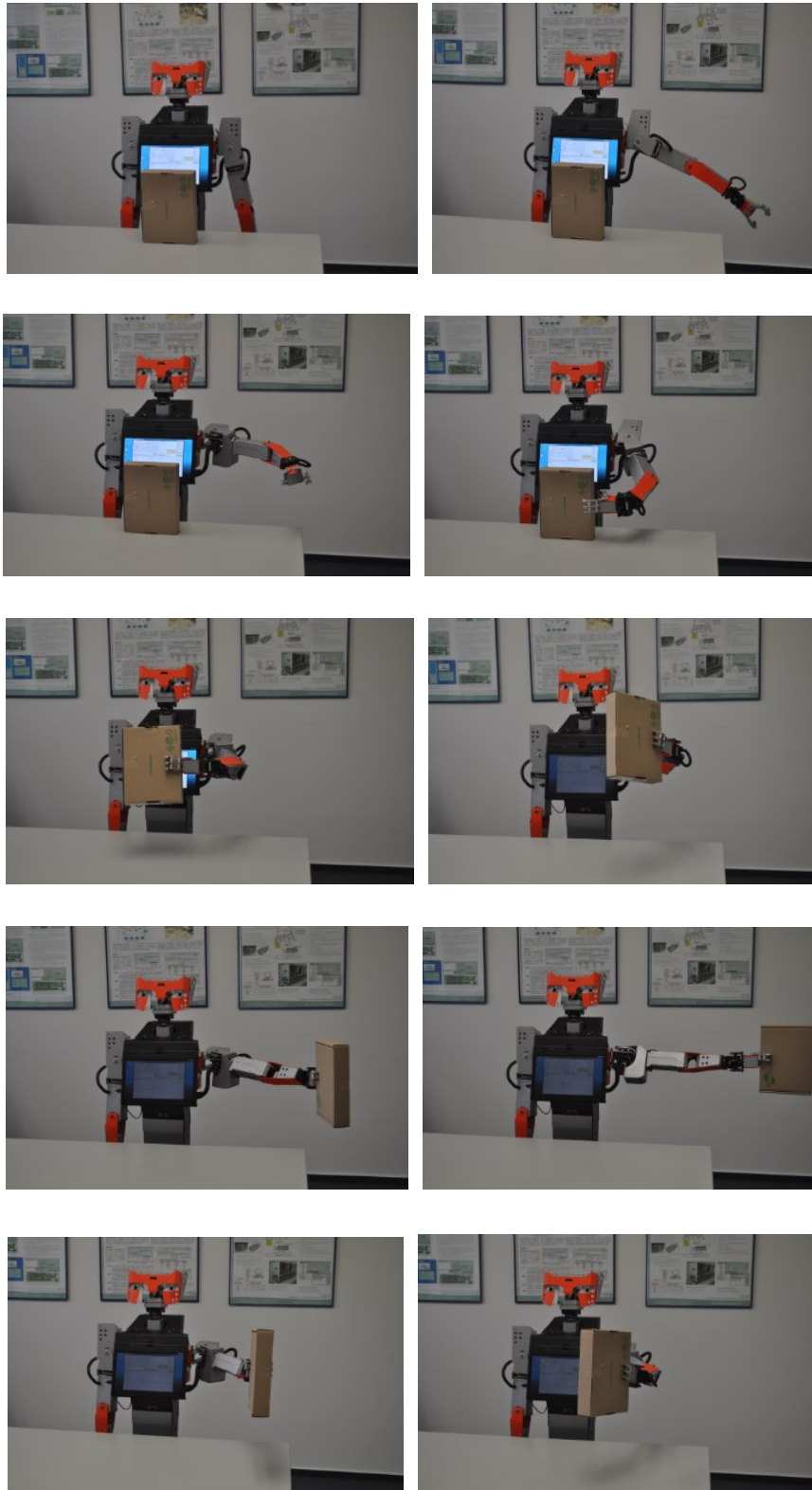


Figure 10.1: Training arm module generating the XML-based control files for the robot arm movements

This training method suits for the application in this study. The reasons are explained as follows: (a) this method can avoid the complex computation of the kinematic equations of the robot real arms for some required movements; and (b) the generated XML-based files can be recognized and reused by the servo hardware of the robot real

arms to repeat the expected actions fast; and (c) based on the programmed RAC, the preparation of the control files for the robot arms can be done in a short time. The results of this experiment are given in Fig. 10.2.



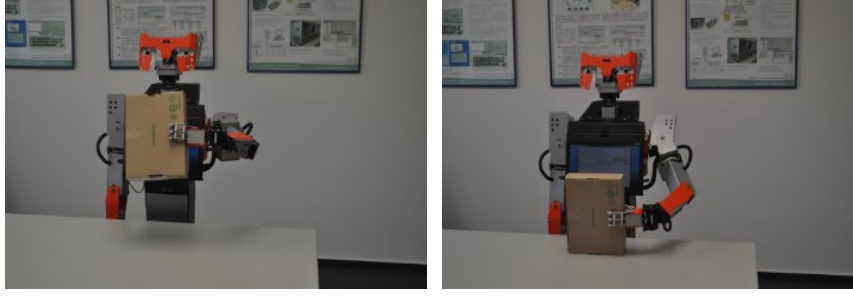
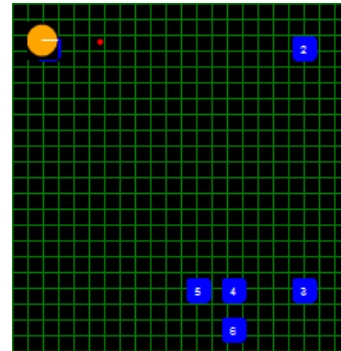
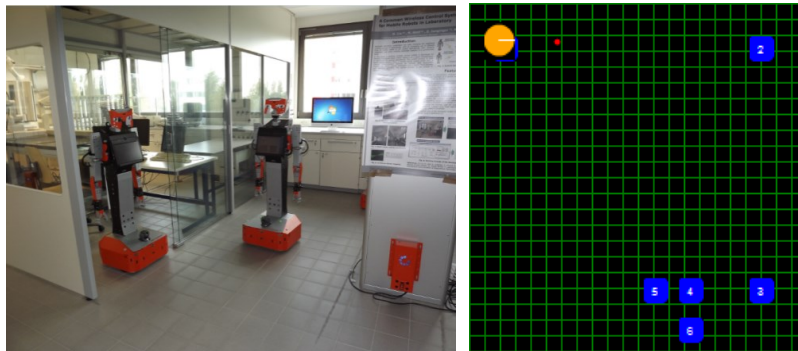


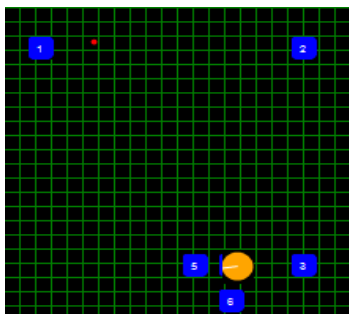
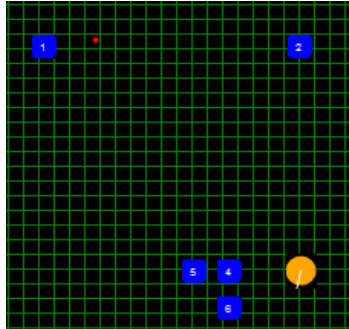
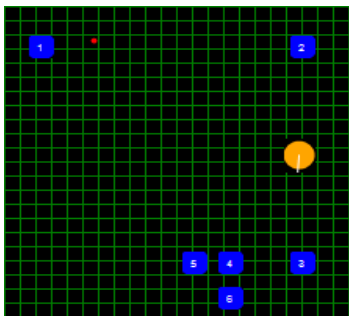
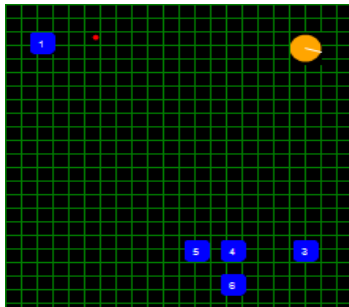
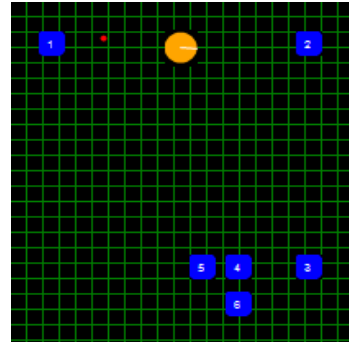
Figure 10.2: The robot arm movements at different time step

## 10.2 Robot Transportation

The second experiment is made to demonstrate the performance of the transportation. The experiment is executed as follows: (a) use the waypoint definition GUI of the RBC to define six points for a real life science laboratory (*at Celisca, Germany*) for two robots (Robot 3C and Robot 4D); (b) run the RRC to connect the two robots' RBCs and finish the path planning calculation for both of them; (c) suppose there is a P2P transportation task which requires a mobile robot to patrol a shortest path between the defined starting position (position ①) and the destination position (position ⑥) then to finish a completed arm manipulation (including pick up and put down) when the mobile robot arrives at the destination position. The RRC is defined to select the candidate robot by the standard of robot power. Finally the Robot 3C is chosen because of its higher power; (d) the RRC sends the calculated shortest path to the RBC of the Robot 3C; (e) the RBC controls the robot to finish this P2P task. From Fig. 10.3, we can see that the RBC is controlling the Robot 3C rightly following the selected path ①->②->③->④->⑥; (f) as shown in Fig. 10.4, when the Robot 3C reaches the position ⑥, it drives its on-board RAC to pick the expected box up then put it down successfully; and (g) as given in Fig. 10.5, the relevant indoor positions of all the connected robots are monitored by the RRC's global 2D map in real-time.









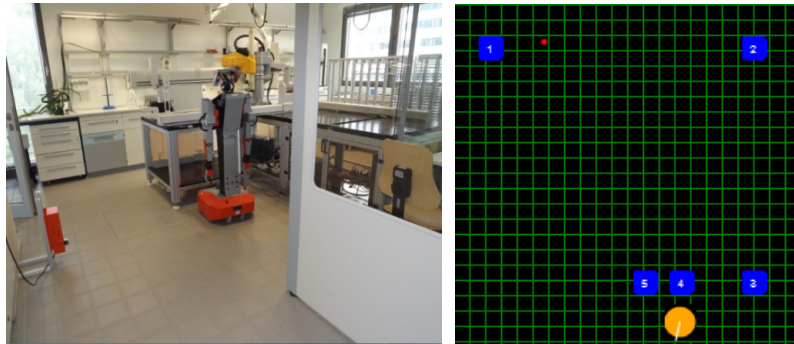
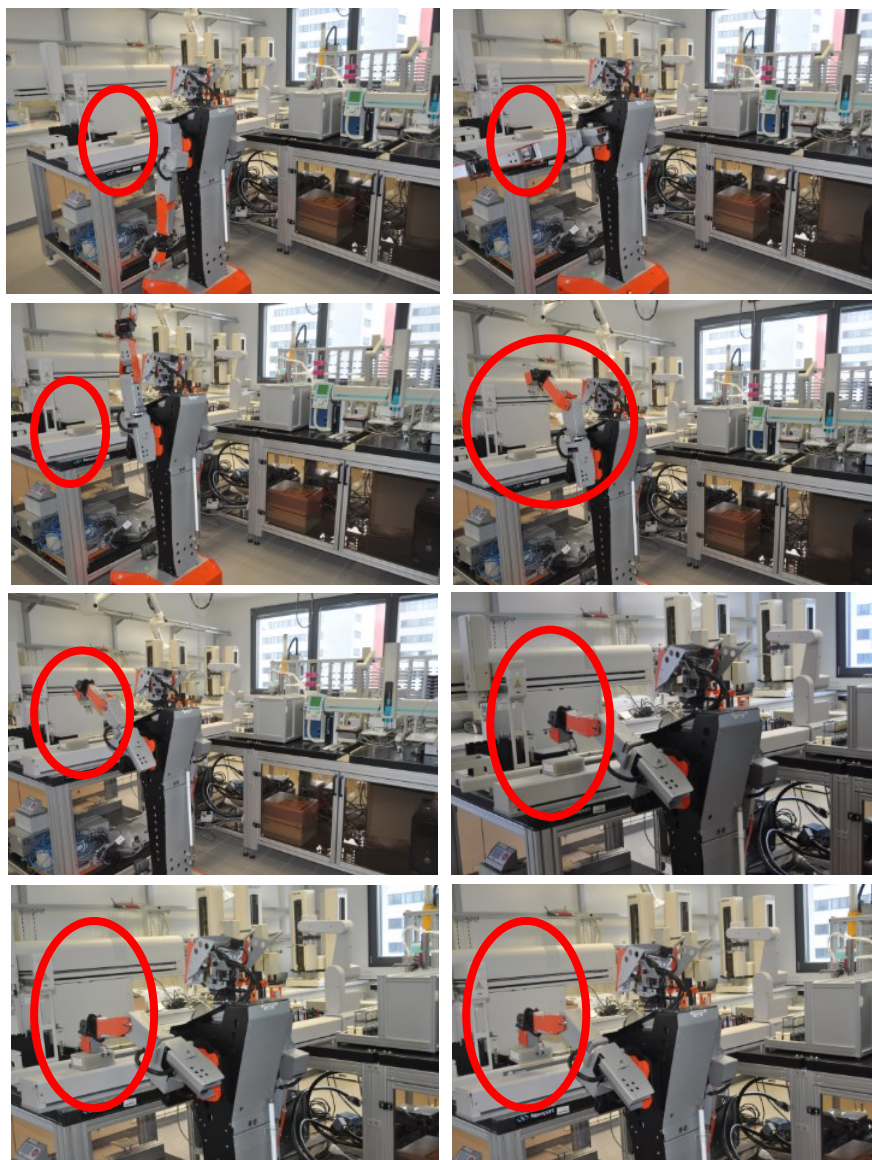


Figure 10.3: The robot transportation in a life science laboratory



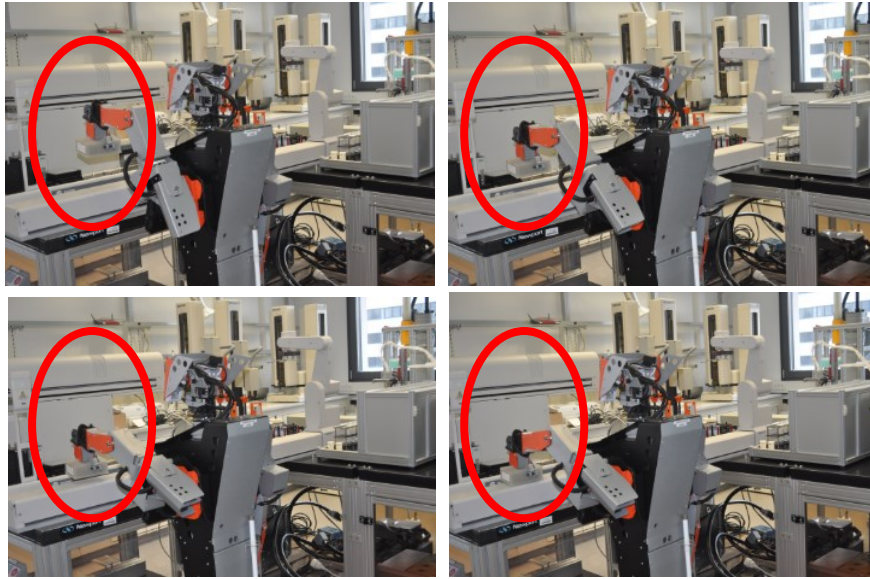


Figure 10.4: The robot executing the arm manipulation at the destination (position ⑥)



Figure 10.5: All running robots monitored by the RRC 2D map

### 10.3 Charging Management

The third experiment is given to demonstrate the performance of the robot recharging. The experiment is given as follows: (a) a transportation map is built in a H2O mobile robot by using its on-boarding RBC. This map consists of three transportation positions (see Fig. 10.5-a/b/c), a waiting position (see Fig. 10.5-d) and two recharging positions (see Fig. 10.5-e/f). The waiting position is chosen in front of the first recharging position; and (b) this robot is asked to do loop-patrol transportation in the

map. When the power of the robot is lower than a defined value (e.g., 12 V), the robot will move to the charging waiting position (*see Fig. 10.5-d*) for correcting the posture (i.e., turning around) then go backward to the charging station to do recharging. The movements in Fig. 10.5 show that the recharging management function of the system works rightly.

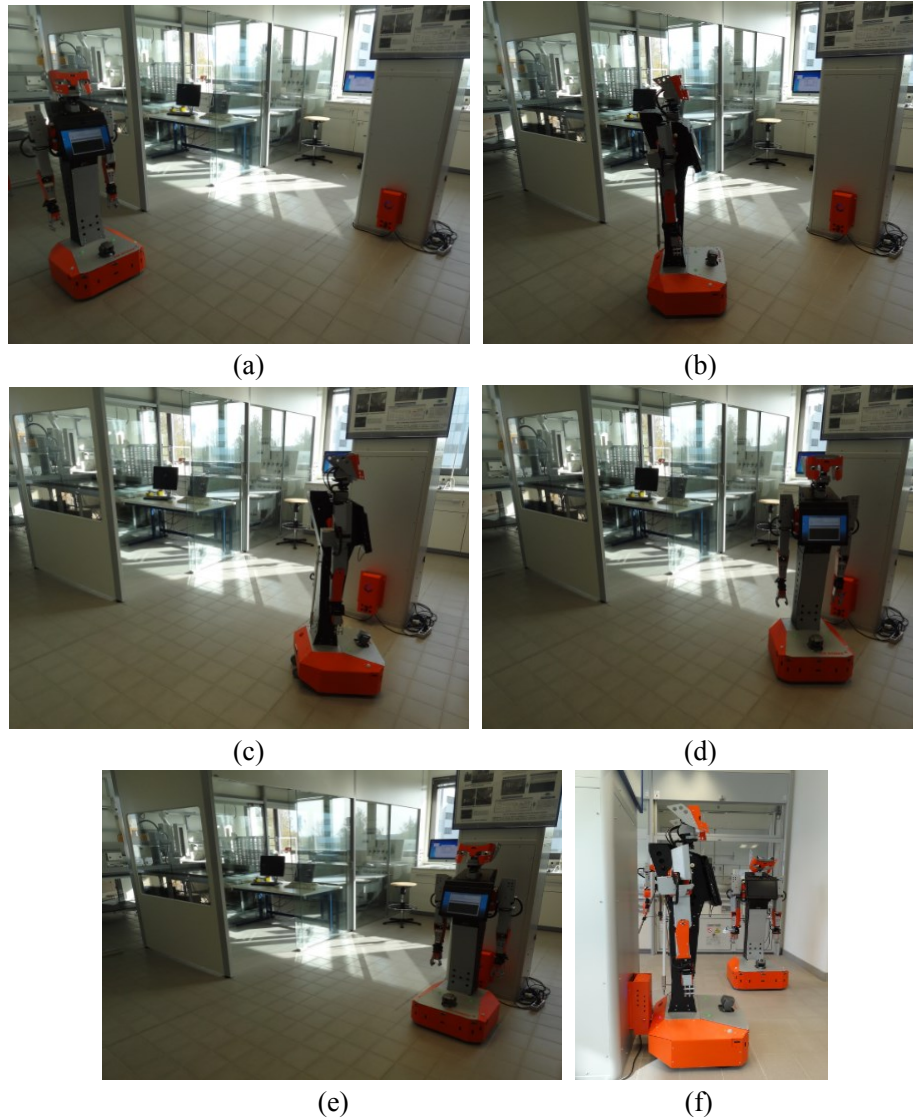


Figure 10.6: The robot recharging in a life science laboratory

## Chapter 11 Conclusions and Outlook

### 11.1 Conclusions

In this study a smart multi-mobile robot system has been presented for the transportation tasks in the distributed laboratory environments. In this system a series of technical and scientific issues have been done as follows:

(a) A two level Client/Server based control strategy is proposed considering all of the special requirements of the laboratory environments (including economy, expandability, flexibility and stability). The strategy is comprised of four levels, the PMS in the first level for the transportation task request, the RRC in the second level for the transportation management, the RBC in the third level for the transportation motion execution and the RAC in the fourth level for the transportation arm execution. Based on this control strategy the finally developed transportation system can suit for any kinds of laboratory environments and any kinds of robotic sensing modules or mobile robots could be integrated conveniently (*see Chapter 2*);

(b) A hybrid communication network is presented to do the data transmission of the transportation. This communication architecture consists of an internal component and an external component. In the internal component a standard IEEE 802.11g cable based network is adopted for the hardware data measurement between the RBCs and the robot inside hardware modules. Considering the mobility of the developed transportation system, in the external component a standard IEEE 802.11g Wi-Fi network is used for the command and data transmission among the PMS, the RRC and the RBCs. To guarantee the stable performance of the communication when the transportation system run in a super-huge laboratory environment and avoid the possibilities of data jam caused by the robot on-board multiple cameras, an improved method using the marine wireless bridges is provide to improve the communication performance between the RRC and the RBCs (*see Chapter 3*);

(c) A new method is proposed for the multi-robot indoor localization in laboratory environments. The method is composed of a ceiling landmark based strategy and a floor landmark based strategy. In the ceiling strategy, one kind of indoor positioning module named StarGazer is adopted working together with a series of ceiling landmarks. The ceiling way is designed for the robot indoor global localization of the



whole laboratory environments (buildings). The advantages of the ceiling localization are economy and expansibility. It can be extended to any sizes of laboratories after installed additional landmarks. In the floor strategy, one kind of depth camera named Microsoft Kinect is utilized and an effective floor landmark is developed. The floor measure is provided for the robot indoor local localization for some special positions where the high-precision performance is desired. The floor localization can cooperate with the previous ceiling one to attain the satisfactory performance of the robotic transportation (*see Chapter 4*).

(d) A new method is provided to improve the original performance of the H20 robots in indoor collision avoidance. The H20 robots use the ultrasonic and IR sensors to measure a front obstacle (human beings or laboratory facilities) and utilized the APF to find an alternative avoidance path. Due to the limited ranges of the used ultrasonic sensors, the H20 robots will vibrate in front of close obstacles. Aims at this condition, an improved method using the Microsoft Kinect is proposed to measure farer obstacles and recognize the shapes of the coming obstacles. Also a control strategy of the improved collision avoidance is put forward (*see Chapter 5*).

(e) A new framework is built for the transportation path planning. In the planning framework both of the map theory based algorithms and the artificial intelligence based algorithms are included. In the component of map theory based algorithms, a hybrid method based on Floyd algorithm and Dijkstra algorithm is presented. The Floyd algorithm will calculate all shortest paths for any pair of waypoints in a transportation map and the Dijkstra will determine an alternative path when a waypoint in the Floyd paths becomes unavailable. This map theory sub-framework suit for the P2P transportation tasks with a small number of transportation waypoints. In the component of artificial intelligence based algorithms, a hybrid method based on GA and AACA is proposed to do the patrol transportation path planning with a big number of waypoints. At the same time a comparison of the GA and the AACA in the path planning computation is done. The experimental results show that the AACA has better performance than the GA. Based on this new framework any sizes of transportation path planning works can be done (*see Chapters 6 and 7*).

(f) A new robot charging management method is provided in this study. In this method, an automated charging station from the Canada DrRobot company is introduced, a robot power measuring component is programmed and a new method using the intelligent AIA to determine the best installing positions of the numbers of automated charging stations is designed. Here it is necessary to mention that this AIA based computational algorithm not only considers the distances between the charging stations and the transportation waypoints but also regards the task distribution of

every transportation waypoint (*see Chapter 8*).

(g) A control software platform is developed to organize the robot transportation tasks in laboratories. In the developed platform there are three control centers, the RRC, the RBC and the RAC. All of them are programmed using the Microsoft C# language with the .Net framework. Among them, the RAC is designed for the arm movements, the RBC is presented to execute the transportation by controlling and measuring all robot inside hardware, and the RRC is proposed to manage the transportation by communicating with both of the higher PMS and the lower RBCs. The architecture of this software platform is flexible, which means other kinds of mobile robots and automated systems in laboratories can be integrated seamlessly. In addition, a XML-based protocol is designed for the command communication among those three control centers and a My SQL-based network database is developed to store the important information of the transportation system (*see Chapter 9*).

(h) A series of experiments has been done to verify the performance of the developed laboratory transportation system, including the initialization of robot arm movements, the robot transportation, the robot recharging management and the robot collision avoidance. The experimental results show that this system is effective and efficient for the laboratory transportation in the distributed laboratory environments (*see Chapter 10*).

## 11.2 Outlook

As described in Section 11.1, the presented control strategy and its developed system are ready to be integrated with the laboratory automation to execute transportation tasks. Nevertheless, there are still some technical issues which will be carried out by our group at Celisca in the near future as follows:

(a) ***Performance of arm manipulation:*** In this dissertation the RAC component is developed to work together with the DrRobot training arm tool to generate the expected arm movements of the H20 robots. This way is effective but not very accurate. So if the transportation system is asked to do high-accuracy arm manipulation, the image processing based arm correction is desired.

(b) ***Execution efficiency of transportation tasks:*** In the current transportation path planning, the execution time for a coming transportation task can be calculating based on the total distance of a selected path and the robot running velocity. However this computational time is a theoretical value. It could be different to the real execution time especially in some cases when the selected paths are changed because of some

dynamic reasons (e.g. the request of collision avoidance). The best solution to this issue is that when a transportation task is distributed from the PMS to the RRC, the RRC can reply the PMS the needed time by searching for the stored data in the database. This solution requires a statistical calculation function in the RRC. Although in this dissertation a simple statistical function has been already considered, a more powerful one is desired in the future.

(c) ***Full automation of transportation system:*** we can say the current transportation system is highly automated. No any human operation is needed in the whole working process of the transportation system. The transportation task will be presented from the PMS through the RRC to the selected RBC automatically. Even if in some special cases when the distributed tasks cannot be recognized or executed by the robots because of some unexpected errors, the whole transportation system will still work rightly and the error information will be transmitted to different control levels of this system automatically. Nevertheless aims at the full automation, this system needs more developing works. For instance, the robots should open the laboratory doors automatically during their transportation. In the next step the robotic entrance automatic module will be developed.

## References

- [1] I. Ul-Haque and E. Prassler, "Experimental Evaluation of a Low-cost Mobile Robot Localization Technique for Large Indoor Public Environments", in *41st International Symposium on Robotics (ISR) and 6th German Conference on Robotics (ROBOTIK)*, Jun. 2010, Germany: Munich, pp 1–7.
- [2] P. Najmabadi, A. A. Goldenberg, and A. Emili, "A scalable robotic-based laboratory automation system for medium-sized biotechnology laboratories", in *IEEE International Conference on Automation Science and Engineering (CASE)*, Aug. 2005, Canada: Edmonton, pp 166–171.
- [3] J. W. Brodack, M. R. Kilbourn, and M. J. Welch, "Automated production of several positron-emitting radiopharmaceuticals using a single laboratory robot", *International Journal of Radiation Applications and Instrumentation. Part A. Applied Radiation and Isotopes*, vol 39, no 7, pp 689–698, 1988.
- [4] N. Brenes, M. N. Quigley, and W. S. Reid, "Determination of soil pH using a laboratory robot", *Analytica chimica acta*, vol 310, no 2, pp 319–327, 1995.
- [5] B. J. Choi, K. Noh, J. W. Kim, S. M. Jin, J. C. Koo, S. M. Ryew, J. Kim, W. H. Son, K. Tak Ahn, W. Chung, and H. R. Choi, "Intelligent BioRobot Platform for Integrated Clinical Test", in *International Joint Conference on SICE-ICASE*, Oct. 2006, Korea: Busan, pp 5828–5832.
- [6] W. Gecks and S. T. Pedersen, "Robotics-an efficient tool for laboratory automation", *IEEE Transactions on Industry Applications*, vol 28, no 4, pp 938–944, 1992.
- [7] B. He, B. J. Burke, X. Zhang, R. Zhang, and F. E. Regnier, "A picoliter-volume mixer for microfluidic analytical systems", *Analytical chemistry*, vol 73, no 9, pp 1942–1947, 2001.
- [8] H. Andersson, W. Van Der Wijngaart, P. Nilsson, P. Enoksson, and G. Stemme, "A valve-less diffuser micropump for microfluidic analytical systems", *Sensors and Actuators B: Chemical*, vol 72, no 3, pp 259–265, 2001.
- [9] J. Wang, "Electrochemical detection for microscale analytical systems: a review", *Talanta*, vol 56, no 2, pp 223–231, 2002.
- [10] A. N. Papas, M. Y. Alpert, S. M. Marchese, J. W. Fitzgerald, and M. F. Delaney, "Evaluation of robot automated drug dissolution measurements", *Analytical chemistry*, vol 57, no 7, pp 1408–1411, 1985.
- [11] D. Kaber, N. Stoll, and K. Thurow, "Human-Automation Interaction Strategies for Life Science Applications: Implications and Future Research", in *IEEE International Conference on Automation Science and Engineering (CASE)*, Sept. 2007, USA: Scottsdale, pp nil92–nil92.
- [12] S. Y. Cho, K. S. Park, J. E. Shim, M. S. Kwon, K. H. Joo, W. S. Lee, J. Chang, H. Kim, H. C. Chung, H. O. Kim, and others, "An integrated proteome database for two-dimensional electrophoresis data analysis and laboratory information management system", *Proteomics*, vol 2, no 9, pp 1104–1113, 2002.



- [13] B. Gode, S. Holzmüller-Lae, K. Rimane, M.-Y. Chow, and N. Stoll, "Laboratory Information Management Systems - An Approach as an Integration Platform within Flexible Laboratory Automation for Application in Life Sciences", in *IEEE International Conference on Automation Science and Engineering (CASE)*, Sept. 2007, USA: Scottsdale, pp 841–845.
- [14] N. Matsuhira, F. Ozaki, S. Tokura, T. Sonoura, T. Tasaki, H. Ogawa, M. Sano, A. Numata, N. Hashimoto, and K. Komoriya, "Development of robotic transportation system - Shopping support system collaborating with environmental cameras and mobile robots", in *41st International Symposium on Robotics (ISR) and 6th German Conference on Robotics (ROBOTIK)*, Jun. 2010, Germany: Munich, pp 1–6.
- [15] T. Tomizawa, A. Ohya, and S. Yuta, "Remote Shopping Robot System, -Development of a hand mechanism for grasping fresh foods in a supermarket", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2006, China: Beijing, pp 4953–4958.
- [16] S. Nishimura, H. Takemura, and H. Mizoguchi, "Development of attachable modules for robotizing daily items -Person following shopping cart robot", in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2007, China: Sanya, pp 1506–1511.
- [17] Y. Ludewig, N. Döring, and N. Exner, "Design and evaluation of the personality trait extraversion of a shopping robot", in *IEEE International Symposium on Robots and Human Interactive Communications*, Sept. 2012, France: Paris, pp 372–379.
- [18] M.-Y. Shieh, Y.-H. Chan, Z.-X. Lin, and J.-H. Li, "Design and Implementation of A Vision-Based Shopping Assistant Robot", in *IEEE International Conference on Systems, Man and Cybernetics*, Oct. 2006, Taipei, pp 4493–4498.
- [19] H.-M. Gross, H. Boehme, C. Schroeter, S. Mueller, A. Koenig, E. Einhorn, C. Martin, M. Merten, and A. Bley, "TOOMAS: Interactive Shopping Guide robots in everyday use - final implementation and experiences from long-term field trials", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2009, USA: St. Louis, Missouri, pp 2005–2012.
- [20] A. G. Ozkil, Z. Fan, S. Dawids, H. Aanes, J. K. Kristensen, and K. H. Christensen, "Service robots for hospitals: A case study of transportation tasks in a hospital", in *IEEE International Conference on Automation and Logistics*, Aug. 2009, China: Shenyang, pp 289–294.
- [21] M. Takahashi, T. Suzuki, F. Cinquegrani, R. Sorbello, and E. Pagello, "A mobile robot for transport applications in hospital domain with safe human detection algorithm", in *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2009, China: Gulin, pp 1543–1548.
- [22] W. K. Fung, Y. Y. Leung, M. K. Chow, Y. H. Liu, Y. Xu, W. Chan, T. W. Law, S. K. Tso, and C. Y. Wang, "Development of a hospital service robot for transporting task", in *IEEE International Conference on Robotics, Intelligent Systems and Signal Processing*, Oct. 2003, China: Changsha, pp 628–633.

- [23] S. Thiel, D. Habe, and M. Block, "Co-operative robot teams in a hospital environment", in *IEEE International Conference on Intelligent Computing and Intelligent Systems*, Nov. 2009, China: Shanghai, pp 843–847.
- [24] J. Evans, B. Krishnamurthy, B. Barrows, T. Skewis, and V. Lumelsky, "Handling real-world motion planning: a hospital transport robot", *IEEE Control Systems*, vol 12, no 1, pp 15–19.
- [25] T. Skewis, J. Evans, V. Lumelsky, B. Krishnamurthy, and B. Barrows, "Motion planning for a hospital transport robot", in *IEEE International Conference on Robotics and Automation (ICRA)*, Apr. 1991, USA: Sacramento, pp 58–63.
- [26] Y. Takahashi, M. Kohda, Y. Kanbayashi, K. Yamahira, and T. Hoshi, "Tray carrying robot for hospital use", in *IEEE International Conference on Industrial Technology*, Dec. 2005, China: Hong Kong, pp 371–376.
- [27] B. Horan, Z. Najdovski, T. Black, S. Nahavandi, and P. Crothers, "OzTug mobile robot for manufacturing transportation", in *IEEE International Conference on Systems, Man, and Cybernetics*, Oct. 2011, USA: Anchorage, pp 3554–3560.
- [28] J. W. Kang, B. S. Kim, and M. J. Chung, "Development of Assistive Mobile Robots Helping the Disabled Work in a Factory Environment", in *IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, Oct. 2008, China: Beijing, pp 426–431.
- [29] M. Endo, K. Hirose, Y. Hirata, K. Kosuge, T. Kanbayashi, M. Oomoto, K. Akune, H. Arai, H. Shinoduka, and K. Suzuki, "A car transportation system by multiple mobile robots - iCART -", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept. 2008, France: Nice, pp 2795–2801.
- [30] U. Rembold, T. Lueth, and T. Ogasawara, "From autonomous assembly robots to service robots for factories", in *IEEE/RSJ/GI International Conference on Intelligent Robots and Systems (IROS)*, Sept. 1994, Germany: Munich, pp 2160–2167.
- [31] B. Lauterbach, E. Forgber, G. Urban, and W. Anheier, "Vision-based Navigation Assistance For Autonomous Factory Transport Vehicles", in *IEEE Symposium on Intelligent Vehicles*, Jul. 1993, Japan: Tokyo, pp 329–334.
- [32] M. Zaera, M. Esteve, C. E. Palau, J. C. Guerri, F. J. Martinez, and P. F. de Cordoba, "Real-time scheduling and guidance of mobile robots on factory floors using Monte Carlo methods under windows NT", in *8th IEEE International Conference on Emerging Technologies and Factory Automation*, Oct. 2001, France: Antibes, pp 67–74.
- [33] M. Wojtczyk, G. Panin, C. Lenz, T. Roder, S. Nair, E. Roth, A. Knoll, R. Heidemann, K. Joeris, C. Zhang, M. Burnett, and T. Monica, "A vision based human robot interface for robotic walkthroughs in a biotech laboratory", in *4th ACM/IEEE International Conference on Human-Robot Interaction*, Mar. 2009, USA: San Diego, pp 309–310.
- [34] L. A. Marks, K. L. Short, D. Hoffmann, and A. Lew, "Microprocessor-based robotic system for control of fluid connections in the cardiac catheterization laboratory", *IEEE Transactions on Biomedical Engineering*, vol 35, no 2, pp 161–166, 1988.

- [35] K. Thurow, N. Stoll, and K. Ritterbusch, "A Fast Optical Method for the Determination of Liquid Levels in Microplates", *Journal of Automated Methods and Management in Chemistry*, vol 2011, pp 1–6, 2011.
- [36] G. Petroni, M. Niccolini, S. Caccavaro, C. Quaglia, A. Menciassi, S. Schostek, G. Basili, O. Goletti, M. O. Schurr, and P. Dario, "A novel robotic system for single-port laparoscopic surgery: preliminary experience", *Surgical Endoscopy*, Article in press, 2013.
- [37] C. Krishnan, D. Kotsapouikis, Y. Y. Dhaher, and W. Z. Rymer, "Reducing Robotic Guidance During Robot-Assisted Gait Training Improves Gait Function: A Case Report on a Stroke Survivor", *Archives of Physical Medicine and Rehabilitation*, Article in press, 2013.
- [38] M. Khamassi, P. Enel, P. F. Dominey, and E. Procyk, *Medial prefrontal cortex and the adaptive regulation of reinforcement learning parameters*, *Progress in Brain Research*, vol 202, pp 441–464, 2013.
- [39] E. Palma and C. Bufarini, "Robot-assisted preparation of oncology drugs: The role of nurses", *International Journal of Pharmaceutics*, vol 439, no 1–2, pp 286–288, 2012.
- [40] S. Andersson, M. Norman, R. Olsson, R. Smith, G. Liu, and J. Nord, "High-precision, room temperature screening assay for inhibitors of microsomal prostaglandin e synthase-1", *Journal of Biomolecular Screening*, vol 17, no 10, pp 1372–1378, 2012.
- [41] Dr Robot Inc., "WiFi 802.11 robot, Network-based Robot, robotic, robot kit, humanoid robot, OEM solution". [Online]. Available: [http://www.drrobot.com/products\\_H20.asp](http://www.drrobot.com/products_H20.asp). [Accessed: 15-Sep-2011].
- [42] A. Leelasantitham and P. Chaiprapa, "A study of performances on an automatic IEEE 802.11g wireless-standard robot using infrared sensors", in *IEEE International Conference on Robotics and Biomimetics*, Feb. 2009, Thailand: Bangkok, pp 1556–1560.
- [43] A. Leelasantitham, "A Study of Performances on a Wireless Robot Based on Various Brands of IEEE 802.11 g Standard and Controlled by Notebook", in *23rd International Technical Conference on Circuits/Systems, Computers and Communications*, Jul. 2008, Japan: Shimonoseki, pp 6–9.
- [44] G. Enriquez, Sunhong Park, and S. Hashimoto, "Wireless sensor network and RFID sensor fusion for mobile robots navigation", in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2010, China: Tianjin, pp 1752–1756.
- [45] G. Enriquez and S. Hashimoto, "Wireless sensor network-based navigation for human-aware guidance robot", in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2009, China: Gulin, pp 2034–2039.
- [46] C. Rohrig and M. Muller, "Indoor location tracking in non-line-of-sight environments using a IEEE 802.15.4a wireless network", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2009, USA: St. Louis, Missouri, pp 552–557.
- [47] Z. G. Niu and Y. B. Wu, "Research on Wireless Remote Control for Coal Mine Detection Robot", in *International Conference on Digital Manufacturing and Automation*, Dec. 2010, China: Changsha, pp 315–318.

- [48] M. H. Li, L. N. Sun, Q. H. Huang, Z. Cai, and S.H. Piao, "GPRS Based Guard Robot Alarm System Design", in *Fourth International Conference on Internet Computing for Science and Engineering*, May 2009, Canada: Vancouver, pp 211–216.
- [49] L. N. Sun, M. H. Li, and T. Chen, "Guard Robot Alarm System Design Based on GPRS", *Applied Mechanics and Materials*, vol 43, pp 217–220, 2011.
- [50] E. Manoj, J. Hassan, and A. Abdulla, "Robot Teleoperation System based on GPRS for Militant Confidential", *International Journal of Computer Applications*, vol 1, no 18, pp 103-109, 2010.
- [51] S. Sagiroglu, N. Yilmaz, and M. Arif Wani, "Web Robot Learning Powered by Bluetooth Communication System", in *5th International Conference on Machine Learning and Applications*, Dec. 2006, USA: Orlando, pp 149–156.
- [52] Y. C. Fai, S. H. Amin, N. bt Fisal, and J. A. Bakar, "Bluetooth enabled mobile robot", in *IEEE International Conference on Industrial Technology*, Dec. 2002, Thailand: Bangkok, pp 903–908.
- [53] J. Park, S. Park, D. -H. Kim, P. -D. Cho, and K. -R. Cho, "Experiments on radio interference between wireless LAN and other radio devices on a 2.4 GHz ISM band", in *57th IEEE Semiannual on Vehicular Technology Conference*, Oct. 2003, USA: Orlando, pp 1798–1801.
- [54] B. You, D. Li, J. Xu, and D. Jia, "The Design of On-line Mobile Robot over Bluetooth Technology", in *IEEE International Conference on Information Acquisition*, Aug. 2006, China: Weihai, pp 224–228.
- [55] J. Suhonen, K. Haataja, N. Paivinen, and P. Toivanen, "The effect of interference in the operation of ZigBee and Bluetooth robot cars", in *4th IEEE/IFIP International Conference on Internet*, Sept. 2008, Uzbekistan: Tashkent, pp 1–5.
- [56] C. A. Rodriguez and F. Guerrero, "Wireless robot teleoperation via internet using IPv 6 over a bluetooth personal area network", *Universidad de Antioquia. Facultad de Ingenieria. Revista*, no 52, pp 172–184, 2010.
- [57] A. Doefexi, S. Armour, B. S. Lee, A. Nix, and D. Bull, "An evaluation of the performance of IEEE 802.11 a and 802.11 g wireless local area networks in a corporate office environment", in *IEEE International Conference on Communications*, May 2003, USA: Anchorage, pp 1196–1200.
- [58] P. Mahasukhon, M. Hempel, Song Ci, and H. Sharif, "Comparison of Throughput Performance for the IEEE 802.11a and 802.11g Networks", in *21st International Conference on Advanced Information Networking and Applications*, May 2007, Canada: Niagara Falls, pp 792–799.
- [59] Z. Xiang, Q. Wang, and H. Wen, "The Study of Multi-Robot Communication of Autonomous Soccer Robots Based on C/S Mode", in *International Conference on Multimedia Technology*, Jun. 2010, Spain: Malaga, pp 1–4.
- [60] Y. F. Liu and M. H. Du, "Speech Quality of IEEE 802.11b and 802.11g Wireless LANs", in *International Conference on Computational Intelligence and Security Workshops*, Nov.

2007, USA: Silicon Vally, pp 873–876.

[61] N. Nishiuchi, K. Kurihara, S. Sakai, and H. Takada, “A man-machine interface for camera control in remote monitoring using line-of-sight”, in *2000 IEEE International Conference on Systems, Man, and Cybernetics*, Oct. 2000, USA: Nashville, pp 882–887.

[62] M. Seelye, G. Sen Gupta, J. Seelye, and S. C. Mukhopadhyay, “Camera-in-hand robotic system for remote monitoring of plant growth in a laboratory”, in *2010 IEEE Instrumentation and Measurement Technology Conference*, May 2010, USA: Austin, pp 809–814.

[63] B. Ko, H.-J. Hwang, I.-G. Lee, and J.-Y. Nam, “Fire Surveillance System Using an Omni-directional Camera for Remote Monitoring”, in *IEEE 8th International Conference on Computer and Information Technology Workshops*, Jul. 2008, USA: Utah, pp 427–432.

[64] L-COM Inc., “2.4 GHz 8.5 dBi Omni Marine Antenna - 3ft RP-SMA Plug Connector - HG2409UM-RSP”. [Online]. Available: <http://www.l-com.com/item.aspx?id=22165>. [Accessed: 21-Sep-2011].

[65] N. Sariff and N. Buniyamin, “An Overview of Autonomous Mobile Robot Path Planning Algorithms”, in *4th Student Conference on Research and Development*, Jun. 2006, Malaysia: Shah Alam, pp 183–188.

[66] G. N. Desouza and A. C. Kak, “Vision for mobile robot navigation: a survey”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 24, no 2, pp 237–267, 2002.

[67] S. Park and S. Hashimoto, “Indoor localization for autonomous mobile robot based on passive RFID”, in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2009, Thailand: Bangkok, pp 1856–1861.

[68] S. Chumkamon, P. Tuvaphanthaphiphat, and P. Keeratiwintakorn, “A blind navigation system using RFID for indoor environments”, in *5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, May 2008, Thailand: Krabi, pp 765–768.

[69] B. S. Choi, J. W. Lee, J. J. Lee, and K. T. Park, “A Hierarchical Algorithm for Indoor Mobile Robot Localization Using RFID Sensor Fusion”, *IEEE Transactions on Industrial Electronics*, vol 58, no 6, pp 2226–2235, 2011.

[70] M. Suruz Miah and W. Gueaieb, “A stochastic approach of mobile robot navigation using customized RFID systems”, in *3rd International Conference on Signals, Circuits and Systems*, Nov. 2009, Tunisia: Djerba, pp 1–6.

[71] H. H. Lin, C. C. Tsai, S. M. Hu, and H. Y. Chang, “Automatic mapping for an indoor mobile robot assistant using RFID and laser scanner”, in *Annual Conference on SICE*, Sept. 2007, Japan: Kagawa, pp 2102–2108.

[72] G. Retscher and Q. Fu, “Continuous indoor navigation with RFID and INS”, in *Symposium on Position Location and Navigation*, May 2010, USA: California, pp 102–112.

[73] W. S. Mooi, T. C. Eng, and N. Huda bt Nik Zulkifli, “Efficient RFID tag placement framework for in building navigation system for the blind”, in *8th Asia-Pacific Symposium on Information and Telecommunication Technologies*, Jun. 2010, Malaysia: Kuching, Sarawak,

pp 1–6.

- [74] A. Yazici, U. Yayan, and H. Yucel, “An ultrasonic based indoor positioning system”, in *International Symposium on Innovations in Intelligent Systems and Applications*, Jun. 2011, Turkey: Istanbul, pp 585–589.
- [75] B. H. Kim, J. S. Choi, S. I. Ko, and M. Park, “Improved active beacon system using multi-modulation of ultrasonic sensors for indoor localization”, in *International Joint Conference on SICE-ICASE*, Oct. 2006, Korea: Busan, pp 1366–1371.
- [76] H. Kim, J. Choi, and M. Park, “Indoor localization system using multi-modulation of ultrasonic sensors and digital compass”, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept. 2008, France: Nice, pp 1359–1364.
- [77] M. Ocana, L. M. Bergasa, M. A. Sotelo, and R. Flores, “Indoor robot navigation using a POMDP based on WiFi and ultrasound observations”, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Aug. 2005, Canada: Alberta, pp 2592–2597.
- [78] B. Christian, “Local discriminant bases and optimized wavelet to classify ultrasonic echoes: application to indoor mobile robotics”, in *Proceedings of IEEE Sensors*, vol 2, pp 1654–1659, 2002.
- [79] F. J. Toledo, J. D. Luis, L. M. Tomas, M. A. Zamora, and H. Martinez, “Map building with ultrasonic sensors of indoor environments using neural networks”, in *IEEE International Conference on Systems, Man, and Cybernetics*, Oct. 2000, USA: Nashville, pp 920–925.
- [80] S.-Y. Hwang and J.-B. Song, “Monocular Vision-Based SLAM in Indoor Environment Using Corner, Lamp, and Door Features From Upward-Looking Camera”, *IEEE Transactions on Industrial Electronics*, vol 58, no 10, pp 4804–4812, 2011.
- [81] Y. Abe, M. Shikano, T. Fukuda, F. Arai, and Y. Tanaka, “Vision based navigation system for autonomous mobile robot with global matching”, in *IEEE International Conference on Robotics and Automation (ICRA)*, May 1999, USA: Michigan, pp 1299–1304.
- [82] A. Rodić and G. Mester, “Virtual WRSN–Modeling and simulation of wireless robot-sensor networked systems”, in *8th International Symposium on Intelligent Systems and Informatics*, Sept. 2010, Serbia: Subotica, pp 115–120.
- [83] Y. X. Li and S. T. Birchfield, “Image-based segmentation of indoor corridor floors for a mobile robot”, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2010, Taipei, pp 837–843.
- [84] J. Yuan, X. G. Wang, L. Dong, N. Li, F. Wang, Y. L. Huang, F. C. Sun, and Y. Wang, “ISILON-An intelligent system for indoor localization and navigation based on RFID and ultrasonic techniques”, in *8th World Congress on Intelligent Control and Automation*, Jul. 2010, China: Jinan, pp 6625–6630.
- [85] H. Moravec and A. Elfes, “High resolution maps from wide angle sonar”, in *IEEE International Conference on Robotics and Automation (ICRA)*, Mar. 1985, USA: Missouri, pp 116–121.
- [86] D. Kim and R. Nevatia, “Symbolic Navigation with a Generic Map”, *Autonomous*

*Robots*, vol 6, no 1, pp 69–88, 1999.

[87] J. Borenstein and Y. Koren, “Real-time obstacle avoidance for fast mobile robots in cluttered environments”, in *IEEE International Conference on Robotics and Automation (ICRA)*, May 1990, Japan: Tsukuba, pp 572–577.

[88] J. M. Maja, T. Takahashi, Z. D. Wang, and E. Nakano, “Real-time obstacle avoidance algorithm for visual navigation”, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2000, Japan: Tsukuba, pp 925–930.

[89] H. P. Moravec, “The Stanford cart and the CMU rover”, *Proceedings of the IEEE*, vol 71, no 7, pp 872–884, 1983.

[90] H. P. Moravec, “Obstacle avoidance and navigation in the real world by a seeing robot rover”, *tech report CMURITR8003 Robotics Institute Carnegie Mellon University doctoral dissertation Stanford University*, 1980.

[91] H.-H. Yu, H. W. Hsieh, Y. K. Tasi, Z. H. Ou, Y. S. Huang, and T. Fukuda, “Visual localization for mobile robots based on composite map”, *Journal of Robotics and Mechatronics*, vol 25, no1, pp 25–37, 2013.

[92] H. B. Wang, H. N. Yu, and L. F. Kong, “Ceiling Light Landmarks Based Localization and Motion Control for a Mobile Robot”, in *IEEE International Conference on Networking, Sensing and Control*, Apr. 2007, UK: London, pp 285–290.

[93] G. Jang, S. Lee, and I. Kweon, “Color landmark based self-localization for indoor mobile robots”, in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2002, USA: Washington DC, pp 1037–1042.

[94] H. B. Wang and L. L. Zhang, “Path Planning Based on Ceiling Light Landmarks for a Mobile Robot”, in *IEEE International Conference on Networking, Sensing and Control*, Apr. 2008, China: Hunan, pp 1593–1598.

[95] T. Nakamura and M. Asada, “Motion sketch: Acquisition of visual motion guided behaviors”, in *IEEE International Conference on Artificial Intelligence*, Aug. 1995, Canada: Quebec, pp 126–132.

[96] T. Nakamura and M. Asada, “Stereo sketch: Stereo vision-based target reaching behavior acquisition with occlusion detection and avoidance”, in *IEEE International Conference on Robotics and Automation (ICRA)*, Apr. 1996, USA: Minneapolis, pp 1314–1319.

[97] S. Thrun, “Finding landmarks for mobile robot navigation”, in *IEEE International Conference on Robotics and Automation (ICRA)*, May 1998, Belgium: Leuven, pp 958–963.

[98] F. Dellaert, W. Burgard, D. Fox, and S. Thrun, “Using the condensation algorithm for robust, vision-based mobile robot localization”, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 1999, USA: Providence, RI, pp 1–5.

[99] R. Murrieta-Cid, M. Briot, and N. Vandapel, “Landmark identification and tracking in natural environment”, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 1998, Canada: Victoria, pp 179–184.

- [100] X. Chai, F. Wen, and K. Yuan, "Fast vision-based object segmentation for natural landmark detection on Indoor Mobile Robot", in *International Conference on Mechatronics and Automation*, Aug. 2011, China: Beijing, pp 2232–2237.
- [101] Y. Guo and X. H. Xu, "Color Landmark Design for Mobile Robot Localization", in *Multiconference on Computational Engineering in Systems Applications*, Oct. 2006, China: Beijing, pp 1868–1874.
- [102] J. H. Park, Y. S. Park, and S. W. Kim, "AGV parking system using artificial visual landmark", in *International Conference on Control, Automation and Systems*, Oct. 2008, Korea: Seoul, pp 1579–1582.
- [103] Li Guanghui and Jiang Zhijian, "An Artificial Landmark Design Based on Mobile Robot Localization and Navigation", in *International Conference on Intelligent Computation Technology and Automation*, Mar. 2011, China: Shenzhen, pp 588–591.
- [104] W. Y. Jeong and K. Mu Lee, "CV-SLAM: a new ceiling vision-based SLAM technique", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Jul. 2005, Canada: Alberta, pp 3195–3200.
- [105] D. Lecking, O. Wulf, and B. Wagner, "Localization in a wide range of industrial environments using relative 3D ceiling features", in *IEEE International Conference on Emerging Technologies and Factory Automation*, Sept. 2008, Italy: Cagliari, pp 333–337.
- [106] T. Kim and J. Lyou, "Indoor navigation of skid steering mobile robot using ceiling landmarks", in *IEEE International Symposium on Industrial Electronics*, Sept. 2009, Spain: Palma de Mallorca, pp 1743–1748.
- [107] T. Tanzawa, N. Kiyohiro, S. Kotani, and H. Mori, "The ultrasonic range finder for outdoor mobile robots", in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Aug. 1995, USA: Pittsburgh, pp 368–373.
- [108] A. Amanatiadis, D. Chrysostomou, D. Koulouriotis, and A. Gasteratos, "A fuzzy multi-sensor architecture for indoor navigation", in *IEEE International Conference on Imaging Systems and Techniques*, Jul. 2010, Greek: Thessaloniki, pp 452–457.
- [109] M. B. Holder, M. M. Trivedi, and S. B. Marapane, "Mobile robot navigation by wall following using a rotating ultrasonic scanner", in *13th International Conference on Pattern Recognition*, Aug. 1996, Turkey: Istanbul, pp 298–302.
- [110] W. -Y. Lee, Y. -T. Su, and T. -C. Hung, "Indoor Service Robot with Multi-sensors Implementation on the MOM", in *2010 International Conference on Computational Aspects of Social Networks*, Sept. 2010, China: Taiyuan, pp 121–125.
- [111] HAGISONIC CO.LTD, "Localization system Star Gazer for Intelligent Robots User\_Manual\_Ver\_04\_080417". [Online]. Available: [http://www.robotshop.com/content/PDF/StarGazer\\_User\\_Manual\\_Ver\\_04\\_080417\(English\).pdf](http://www.robotshop.com/content/PDF/StarGazer_User_Manual_Ver_04_080417(English).pdf)
- [112] MICROSOFT Inc., "Kinect for Windows | Voice, Movement & Gesture Recognition Technology". [Online]. Available: <http://www.microsoft.com/en-us/kinectforwindows/>. [Accessed: 26-Nov-2012].



- [113] L. Cheng, Q. Sun, H. Su, Y. Cong, and S. Zhao, "Design and implementation of human-robot interactive demonstration system based on Kinect", in *24th Chinese Control and Decision Conference (CCDC)*, Jul. 2012, China: Hefei, pp 971–975.
- [114] D. S. O. Correa, D. F. Sciotti, M. G. Prado, D. O. Sales, D. F. Wolf, and F. S. Osorio, "Mobile Robots Navigation in Indoor Environments Using Kinect Sensor", in *Second Brazilian Conference on Critical Embedded Systems*, May 2012, Brazil: Campinas, pp 36–41.
- [115] N. Moriyama, K. Takahashi, T. Yokota, T. Cho, K. Kobayashi, K. Watanabe, and Y. Kurihara, "Development of a Kinect-sensor-based navigation component for JAUS compliant mobile robots", in *Annual Conference on SICE*, Aug. 2012, Japan: Akita, pp 824–827.
- [116] F. Flacco, T. Kroger, A. De Luca, and O. Khatib, "A depth space approach to human-robot collision avoidance", in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2012, USA: Minnesota, pp 338–345.
- [117] R. R. Igorevich, E. P. Ismoilovich, and D. Min, "Behavioral synchronization of human and humanoid robot", in *International Conference on Ubiquitous Robots and Ambient Intelligence*, Nov. 2011, Korea: Seoul, pp 655–660.
- [118] C.-S. Park, S.-W. Kim, D. Kim, and S.-R. Oh, "Comparison of plane extraction performance using laser scanner and Kinect", in *8th International Conference on Ubiquitous Robots and Ambient Intelligence*, Nov. 2011, Korea: Seoul, pp 153–155.
- [119] G. Csaba and Z. Vamossy, "Fuzzy based obstacle avoidance for mobil robots with Kinect sensor", in *4th IEEE International Symposium on Logistics and Industrial Informatics*, Sept. 2012, Slovakia: Smolenice, pp 135–144.
- [120] Y. Zou, W. Chen, X. Wu, and Z. Liu, "Indoor localization and 3D scene reconstruction for mobile robots using the Microsoft Kinect sensor", in *10th IEEE International Conference on Industrial Informatics (INDIN)*, Jul. 2012, China: Beijing, pp 1182–1187.
- [121] M. Van den Bergh, D. Carton, R. De Nijs, N. Mitsou, C. Landsiedel, K. Kuehnlenz, D. Wollherr, L. Van Gool, and M. Buss, "Real-time 3D hand gesture interaction with a robot for understanding directions from humans", in *IEEE International Conference on RO-MAN*, Jul. 2011, USA: Georgia, pp 357–362.
- [122] C.-Y. Lee, H.-G. Choi, J.-S. Park, K.-Y. Park, and S.-R. Lee, "Collision Avoidance by the Fusion of Different Beam-width Ultrasonic Sensors", in *IEEE International Conference on Sensors*, Oct. 2007, USA: Georgia, pp 985–988.
- [123] H.-D. Kim, S.-W. Seo, I. Jang, and K.-B. Sim, "SLAM of mobile robot in the indoor environment with Digital Magnetic Compass and Ultrasonic Sensors", in *International Conference on Control, Automation and Systems*, Oct. 2007, Korea: Seoul, pp 87–90.
- [124] K. Izumi, K. Watanabe, M. Shindo, and R. Sato, "Acquisition of Obstacle Avoidance Behaviors for a Quadruped Robot Using Visual and Ultrasonic Sensors", in *9th International Conference on Control, Automation, Robotics and Vision*, Dec. 2006, Singapore, pp 1–6.
- [125] S. Cui, X. Su, L. Zhao, Z. Bing, and G. Yang, "Study on ultrasonic obstacle

avoidance of mobile robot based on fuzzy controller”, in *International Conference on Computer Application and System Modeling*, Oct. 2010, China: Taiyuan, pp 233–237.

[126] H.-W. Je, J.-Y. Baek, and M.-C. Lee, “A study of the collision detection of robot manipulator without torque sensor”, in *International Conference on ICCAS-SICE*, Sept. 2009, China: Shanghai, pp 4468–4471.

[127] P. Vadakkepat, K. C. Tan, and W. Ming-Liang, “Evolutionary artificial potential fields and their application in real time robot path planning”, in *Congress on Evolutionary Computation*, Jul. 2000, USA: California, pp 256–263.

[128] R. Jiang, X. Tian, L. Xie, and Y. Chen, “A Robot Collision Avoidance Scheme Based on the Moving Obstacle Motion Prediction”, in *ISECS International Colloquium on Computing, Communication, Control, and Management*, Aug. 2008, China: Guangzhou, pp 341–345.

[129] N. Qi, B. J. Ma, X. Liu, Z. Zhang, and D. C. Ren, “A modified artificial potential field algorithm for mobile robot path planning”, in *7th World Congress on Intelligent Control and Automation*, Jun. 2008, China: Chongqing, pp 2603–2607.

[130] L. E. Zarate, M. Becker, B. D. Garrido, and H. S. Rocha, “An artificial neural network structure able to obstacle avoidance behavior used in mobile robots”, in *28th Annual Conference on Industrial Electronics*, Nov. 2002, Spain: Sevilla, pp 2457–2461.

[131] S. Ping, L. Kejie, H. Xiaobing, and Q. Guangping, “Formation and obstacle-avoidance control for mobile swarm robots based on artificial potential field”, in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2009, China: Gulin, pp 2273–2277.

[132] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots”, in *IEEE International Conference on Robotics and Automation (ICRA)*, Mar. 1985, USA: Missouri, pp 500–505.

[133] T. Tsuji, P. G. Morasso, and M. Kaneko, “Trajectory generation for manipulators based on artificial potential field approach with adjustable temporal behavior”, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 1996, Japan: Osaka, pp 438–443.

[134] T. N. Chang and E. Hou, “Synthesis of robust servo-controller using adaptive artificial potential field method”, in *4th IEEE Conference on Control Applications*, Sept. 1995, USA: New York, pp 476–481.

[135] C. W. Warren, “Multiple robot path coordination using artificial potential fields”, in *IEEE International Conference on Robotics and Automation (ICRA)*, May 1990, Japan: Tsukuba, pp 500–505.

[136] R. W. Floyd, “Algorithm 97: Shortest path”, *Commun. ACM*, vol 5, no 6, p 345, 1962.

[137] C. Rao and L. Yu, “A particular method for the shortest path finding of mobile robots”, in *2nd International Conference on Industrial Mechatronics and Automation*, May 2010, China: Wuhan, pp 221–224.

- [138] S. K. Lam and T. Srikanthan, "Accelerating the k-shortest paths computation in multimodal transportation networks", in *5th IEEE International Conference on Intelligent Transportation Systems*, Sept. 2002, Singapore, pp 491–495.
- [139] H. Dong, S. Duan, and Y. Zhao, "Delaunay Graph Based Path Planning Method for Mobile Robot", in *International Conference on Communications and Mobile Computing*, Apr. 2010, China: Shenzhen, pp 528–531.
- [140] J. Wang, Y. Sun, Z. Liu, P. Yang, and T. Lin, "Route Planning based on Floyd Algorithm for Intelligence Transportation System", in *IEEE International Conference on Integration Technology*, Mar. 2007, China: Shenzhen, pp 544–546.
- [141] M. -X. Yang, B. -T. Wang, and W. -D. Guo, "Research on the performance of dynamic routing algorithm", in *International Conference on Machine Learning and Cybernetics*, Jul. 2009, China: Baoding, pp 2647–2650.
- [142] H. Lee, S. Kim, G. Baek, and S. Kim, "Modified floyd algorithm for intelligent walking frame", in *2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence*, Nov. 2011, Korea: Seoul, pp 739–742.
- [143] H. Kang, B. Lee, and K. Kim, "Path Planning Algorithm Using the Particle Swarm Optimization and the Improved Dijkstra Algorithm", in *Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, Dec. 2008, China: Wuhan, pp 1002–1004.
- [144] C. Yin and H. Wang, "Developed Dijkstra shortest path search algorithm and simulation", in *International Conference on Computer Design and Applications*, Jun. 2010, China: Qinhuangdao, pp 116–119.
- [145] Y. Hu, S. X. Yang, L.-Z. Xu, and Q.-H. Meng, "A Knowledge Based Genetic Algorithm for Path Planning in Unstructured Mobile Robot Environments", in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Aug. 2004, China: Shenyang, pp 767–772.
- [146] W. Haiying, X. Rui, B. Mingyou, and J. Jinlin, "Optimization design of mobile robot based on genetic algorithm", in *International Conference on Mechatronics and Automation*, Aug. 2009, China: Changchun, pp 767–771.
- [147] J.-S. Kong, B.-H. Lee, and J.-G. Kim, "A study on the gait generation of a humanoid robot using genetic algorithm", in *Annual Conference on SICE*, Aug. 2004, Japan: Sapporo, pp 187–191.
- [148] N. Sadati and J. Taheri, "Genetic algorithm in robot path planning problem in crisp and fuzzified environments", in *IEEE International Conference on Industrial Technology*, Dec. 2002, Thailand: Bangkok, pp 175–180.
- [149] A. H. Wright, "Genetic algorithms for real parameter optimization", *Foundations of genetic algorithms*, vol 1, pp 205–218, 1991.
- [150] F. Herrera, M. Lozano, and J. L. Verdegay, "Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis", *Artificial Intelligence Review*, vol 12, no 4, pp 265–319, 1998.

- [151] M. Dorigo, “Swarm-Bots and Swarmanoid: Two Experiments in Embodied Swarm Intelligence”, in *IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies*, Sept. 2009, Italy: Milano, pp 2–3.
- [152] J. Park, T. H. Song, S. M. Jung, and J. W. Jeon, “XML based robot description language”, in *International Conference on Control, Automation and Systems*, Oct. 2007, Korea: Seoul, pp 2477–2482.
- [153] M. Makatchev and S. K. Tso, “Human-robot interface using agents communicating in an XML-based markup language”, in *IEEE International Workshop on Robot and Human Interactive Communication*, Sept. 2000, Japan: Osaka, pp 270–275.
- [154] Y. P. Draganov and F. Sahin, “Agent-in-the-loop simulation for testing swarm robots using an XML-based system of system framework in discrete-event simulation specification”, in *6th International Conference on System of Systems Engineering*, Jun. 2011, USA: New Mexico, pp 293–298.

# Appendixes

## Appendix A: Specifications of Hardware

### A.1 H20 Mobile Robot

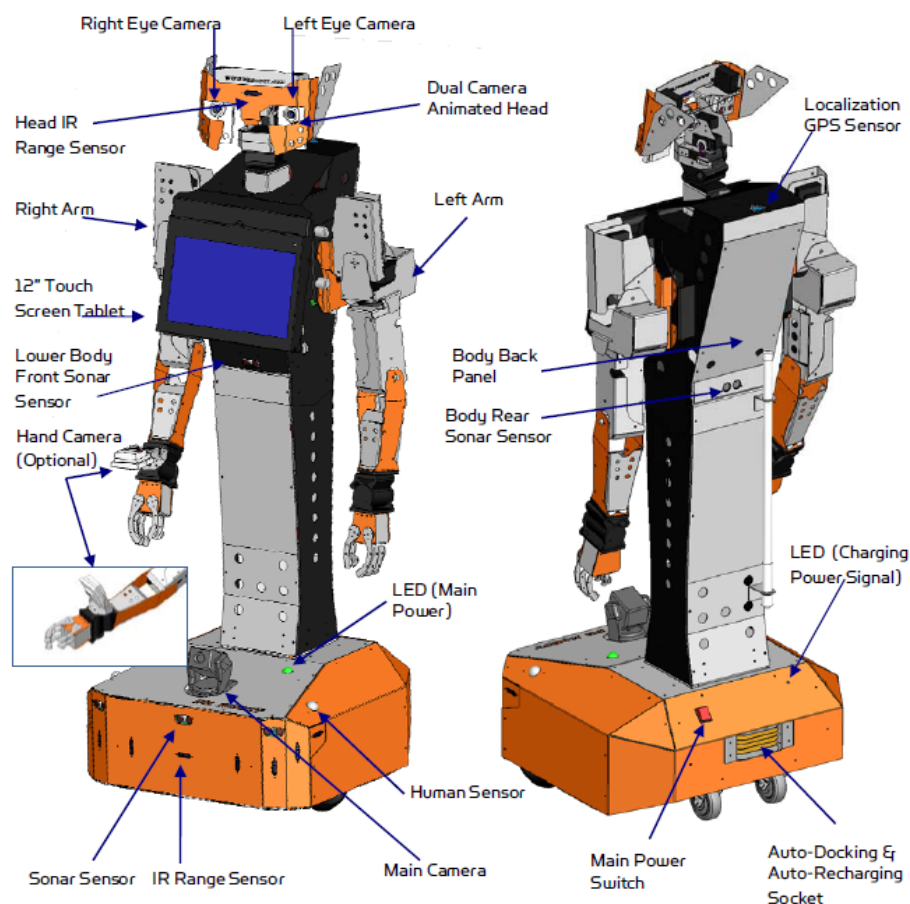


Figure A.1 DrRobot H20 mobile robot

Table A.1 Parameters of the DrRobot H20 mobile robot

Dimensions	51cm x 43cm x 140cm (Width x Length x Height)
Max Speed	75 cm/second
Max Payload	40 kg (Optional 80 kg)
Extended Operating Time	2 hours (Optional 4,8 hours)
Other features	<ul style="list-style-type: none"> <li>✧ 12" touch laptop on chest</li> <li>✧ Dual arms with 6 joins (DOF) + 2 DOF</li> </ul>

	<p>gripper, reaching 60 cm</p> <ul style="list-style-type: none"> <li>✧ 6 DOF head with dual 640 × 480 color cameras</li> <li>✧ StarGazer based localization</li> <li>✧ Automated recharging station</li> <li>✧ Wireless network 802.11g</li> <li>✧ Navigation sensors including 5 sonar and 10 (Infra Radio) IR sensors</li> </ul>
--	---

## A.2 Microsoft Kinect

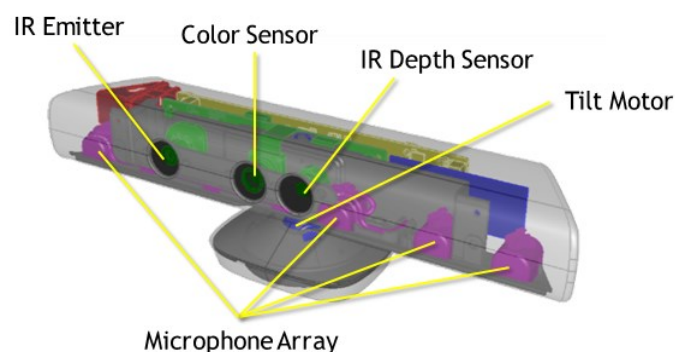


Figure A.2 Cutaway view of Microsoft Kinect sensor

Table A.2 Specifications of the Microsoft Kinect sensor

Dimensions	11" x 2.5" x 1.5" (Width x Length x Height)
Viewing angle	43° vertical by 57° horizontal field of view
Vertical tilt range	±27°
Frame rate	30 Frames Per Second (FPS)
Audio format	16-kHz, 24-bit mono pulse code modulation
Audio input characteristics	A four-microphone array with 24-bit analog-to-digital converter (ADC) and Kinect-resident signal processing including acoustic echo cancellation and noise suppression
Accelerometer characteristics	A 2G/4G/8G accelerometer configured for the 2G range, with a 1° accuracy upper limit.
Sensors	<ul style="list-style-type: none"> <li>✧ An RGB camera with 1280x960 resolution</li> <li>✧ An IR emitter and an IR depth sensor</li> </ul>

	<ul style="list-style-type: none"> <li>✧ A multi-array microphone</li> <li>✧ A 3-axis accelerometer</li> </ul>
--	--

### A.3 StarGazer Module & Ceiling landmark



Figure A.3 StarGazer module

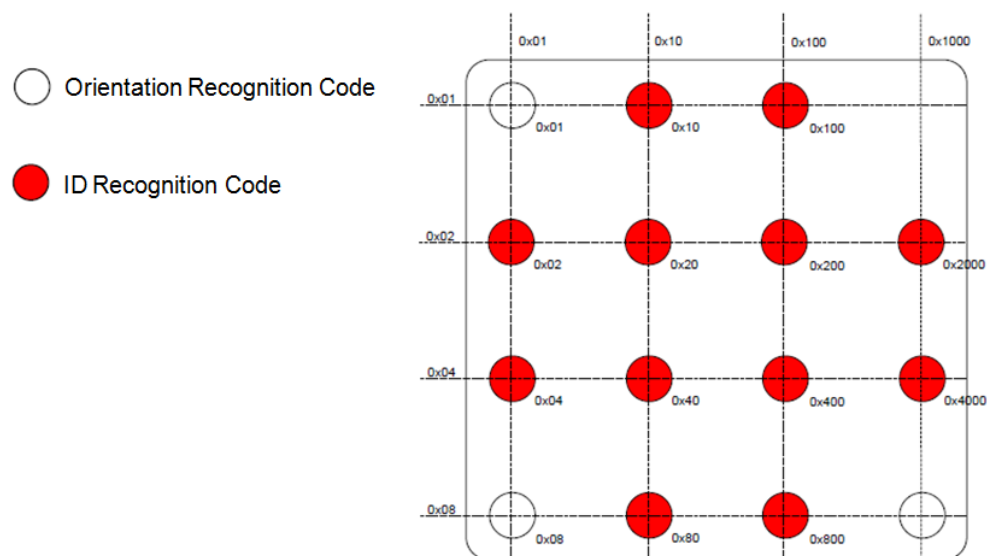


Figure A.4 Ceiling landmark for the StarGazer module

Table A.3 Specifications of the StarGazer module

Dimensions	50cm × 50cm × 28mm (Width x Length x Height)
Hardware interface	UART(TTL 3.3V), 115,200bps
Communication protocol	ASCII code
Sampling frequency	20 times/sec

Rang per landmark	2.5~5 m in diameter
Heading Angle Resolution	1.0degree
Landmark Types	✧ Type 1: $1.2 \leq \text{height} \leq 2.9$ m ✧ Type 2: $2.9 \leq \text{height} \leq 4.5$ m ✧ Type 3: $4.5 \leq \text{height} \leq 6.5$ m ✧ Type 4: $6.5 \leq \text{height} \leq 15$ m
Landmark Types	HL1: 31 ea (for a normal space) HL2: 4,095 ea (for a larger space)
Power interface	5 V: 300 mA, 12 V: 70 mA
Repetitive Precision	2 cm

## Appendix B: Data Format of Software

### B.1 StarGazer-based Indoor Localization

~	^	F/I/Z	±aaaa.aa   ±bbbb.bb   ±cccc.cc   dddd
---	---	-------	---------------------------------------

Figure B.1 Output of StarGazer module

Table B.1 Data format of the measured positions from the StarGazer module

~	Means the module is working rightly
^	Means the result data
F/I/Z	<ul style="list-style-type: none"> <li>● Indicates the Map-Building Mode</li> <li>● Indicates the Map Mode</li> <li>● Indicates the Height Calculation Mode</li> </ul>
±aaaa.aa	Value of Angle (degrees; $-180^{\circ} \sim +180^{\circ}$ )
±bbbb.bb	Position on X axis (cm)
±cccc.cc	Position on Y axis (cm)
dddd	The number of an ID



## B.2 Robot sending sensor data between RBC and RRC

Table B.2 Data format of the sensing data between the RBC and the RRC

Robot Name: 1A	Robot State: Connected/Disconnected	Robot X position: 10.01 m
Robot Y position: 10.02 m	Robot Orientation: 30°	Battery Power #1: 13.25 V
Battery Power #2: 13.25 V	DC Power: 17.25 V	Battery State #1: Charging/Using
Battery State #2: Charging/Using	DC State: Charging/Using	Robot Basis Middle Sonar: 0.55 m
Robot Body Front Sonar: 2.55 m	Robot Body Back Sonar: 2.55 m	Robot Left Hunan Sensor: true/false
Robot Right Hunan Sensor: true/false	Current Waypoint Number: 1	Current Transportation Status: P2PGo/P2PStop

## B.3 Robot control commands between RBC and RRC

Table B.3 Data format of the control commands between the RBC and the RRC

Name of Command	Description
Connect/Disconnect	This command is to connect/ disconnect a remote socket in the RRC and RBC sides.
GetWayPoint/ ReplyWayPoint	This command is for the RRC to get the predefined waypoints from the RBC.
Runpath	This command is to ask the RBC to run a distributed path by the RRC.
Pause	This command is to pause a mobile robot which is running a transportation task. It works together with other commands (Continue, Cancel/Stop,...)
Continue	This command is to continue a mobile robot which is running a transportation task.
Cancel/Stop	This command is to cancel/stop a transportation task for a mobile robot.

## B.4 Robot control commands between RBC and RAC

Table B.4 Data format of the control commands between the RBC and the RAC

Name of Command	Description
Connect/Disconnect	This command is to connect/ disconnect a remote socket in the RBC and RAC sides.
MoveUp	This command is to drive the arms of the H20 robots to catch an experimental facility.
MoveDown	This command is to drive the arms of the H20 robots to put down an experimental facility.
Camera-based Correction	This command is to drive the function of the camera-based correction in the RAC.
Pause	This command is to pause a robot arm which is running a manipulating task. It works together with other commands (Continue, Cancel/Stop...)
Continue	This command is to continue a robot arm which is running a manipulating task.
Cancel/Stop	This command is to cancel/stop a manipulating task for a robot arm.

## B.5 H20Robot Database

Table B.5 Data format of the H20Robot database

Name of Parameters	Descriptions
LTime	This parameter shows the executing time of the data record;
WorkingArea	This parameter shows the working laboratory room number of a transportation process, which is sent from the RBC of every robot;
RobotPosX	This parameter shows the X position of a mobile robot, which is original from the StarGazer navigation;
RobotPosY	This parameter shows the Y position of a mobile robot, which is original from the StarGazer navigation;
RobotPosDir	This parameter shows the direction of a mobile robot, which is original from the StarGazer navigation;

BatteryOneVol	This parameter shows the voltage of the battery #1 in a mobile robot, which is original from the Power module;
BatteryTwoVol	This parameter shows the voltage of the battery #2 in a mobile robot, which is original from the Power module;
DCVol	This parameter shows the voltage of the charging channel in a mobile robot, which is original from the Power module;
BatteryOneStatus	This parameter shows the status of the battery #1 in a mobile robot, which is original from the Power module;
BatteryTwoStatus	This parameter shows the status of the battery #2 in a mobile robot, which is original from the Power module;
DCStatus	This parameter shows the status of the charging channel in a mobile robot, which is original from the Power module;
FrontSonarSensor	This parameter shows the distance value of the installed body front sonar in a mobile robot;
MiddleSonarSensor	This parameter shows the distance value of the installed basis middle sonar in a mobile robot;
BackSonarSensor	This parameter shows the distance value of the installed body back sonar in a mobile robot;
RightHumanSensor	This parameter shows the distance value of the installed right-side human sensor in a mobile robot;
LeftHumanSensor	This parameter shows the distance value of the installed left-side human sensor in a mobile robot;
RobotP2PStatus	This parameter shows the moving status of a mobile robot executing a transportation task;
RobotShortestPath	This parameter shows the executing shortest path of a transportation task;
NextWaypoint	This parameter shows the next expected waypoint of a transportation task;

## Appendix C: Experimental Results

### C.1 StarGazer performance under dark environment

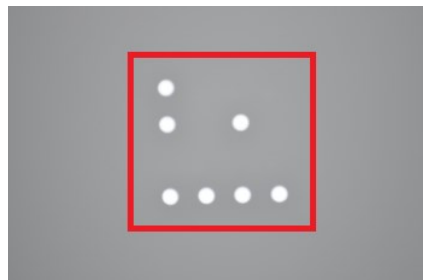


Figure C.1 Experimental environment with strong lighting

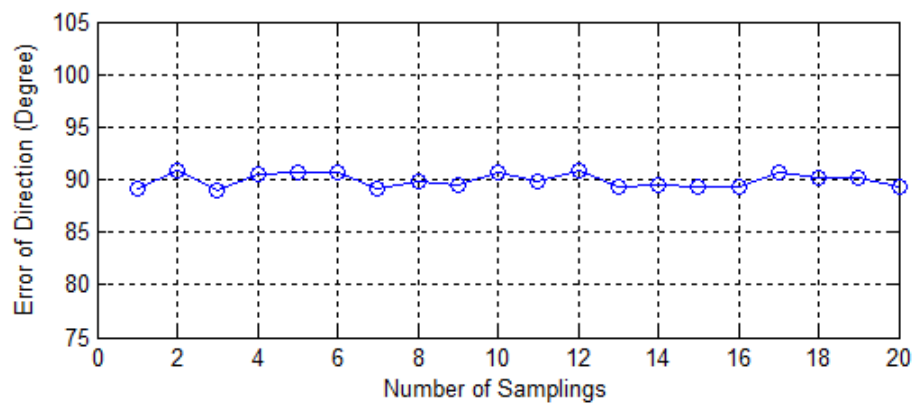


Figure C.2 Errors of direction measurement

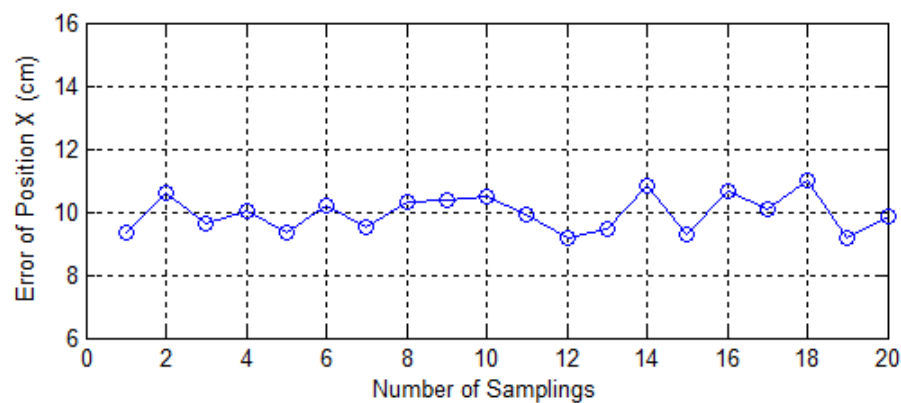


Figure C.3 Errors of position X measurement

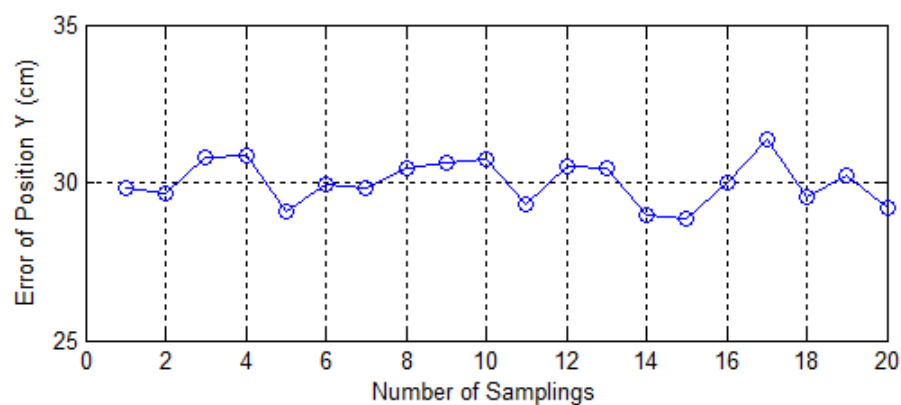


Figure C.4 Errors of position Y measurement

## C.2 StarGazer performance under reflective ceiling environment

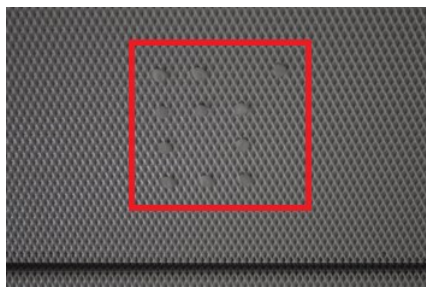


Figure C.5 Experimental environment with strong lighting

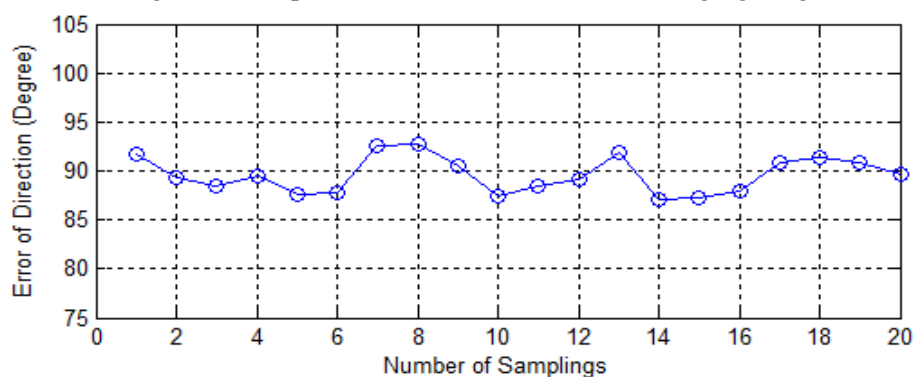


Figure C.6 Errors of direction measurement

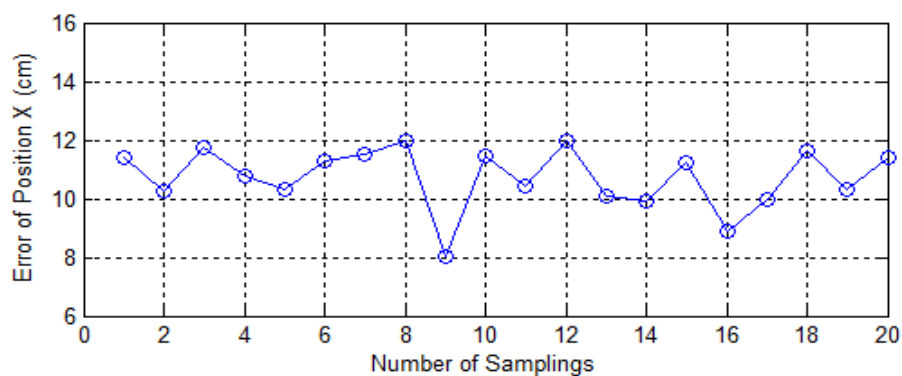


Figure C.7 Errors of position X measurement

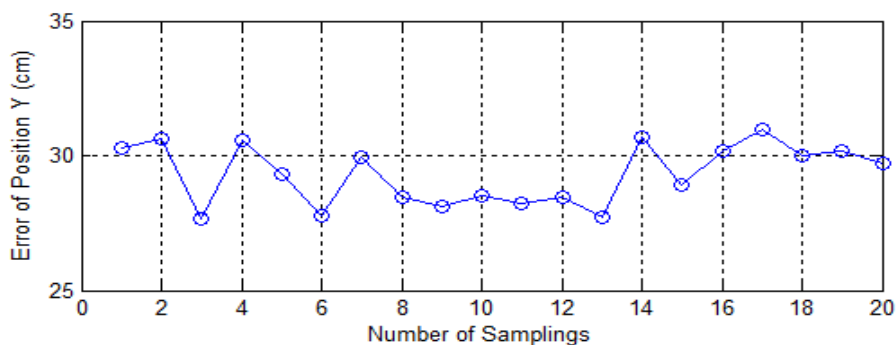


Figure C.8 Errors of position Y measurement

### C.3 Kinect-based indoor local localization

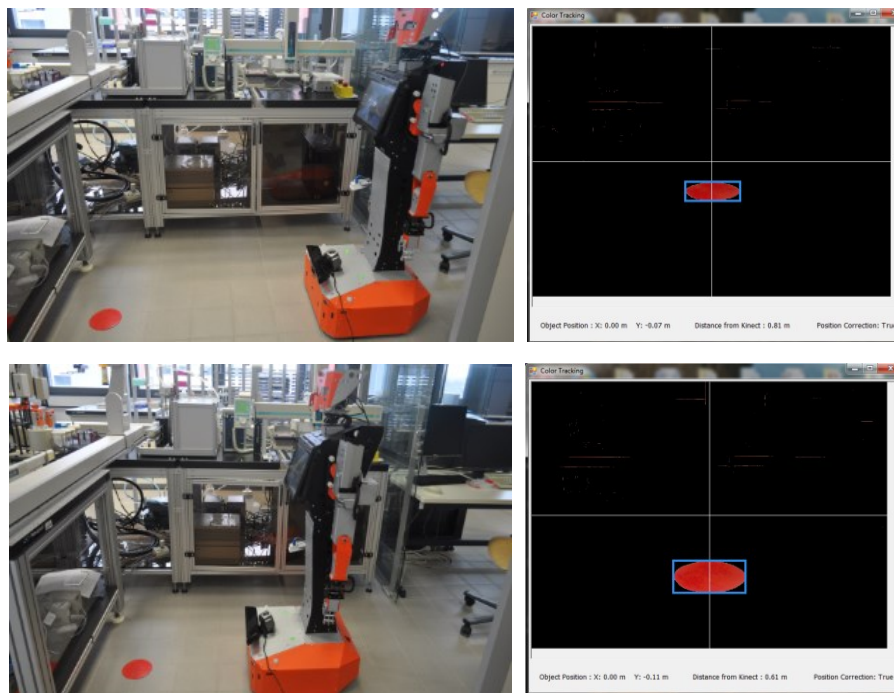


Figure C.9 Experimental results of Kinect-based local localization

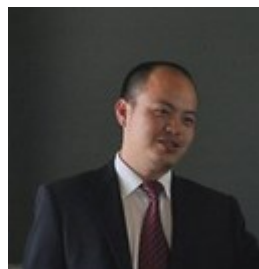
## Declaration

This dissertation ‘Mobile Robot Transportation in Laboratory Automation’ is a presentation of my original research work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions. The work of this dissertation has been done by me under the guidance of Prof. Dr.-Ing. habil. Kerstin Thürow and Prof. Dr. -Ing. Norbert Stoll, at the University of Rostock, Germany. Also the dissertation has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Hui Liu

Rostock, 22 March, 2013

# Curriculum Vitae



## 1. Personal Information:

- ✧ **Name:** Hui Liu
- ✧ **Gender:** Male
- ✧ **Nationality:** Chinese
- ✧ **Birthday:** 07-02-1983

## 2. Education/Work Qualifications:

- ✧ **2011-2013:** Doctoral Scientist (Wissenschaftler) at Center for Life Science Automation (Celisca) & Institute of Automation, University of Rostock, Germany.
- ✧ **2010-2011:** Leader of Project Group (Project No.2006BAC07B03/No.2008 YB042) at the Key Laboratory of Traffic Safety on Track of Ministry of Education of China, Central South University, China.
- ✧ **2009-2010:** Academic Visitor (estimated top 3% in Faculty) at the Laboratory for Motion Generation and Analysis, Department of Mechanical and Aerospace Engineering, Faculty of Engineering, Monash University, Australia.
- ✧ **2008-2009:** Research Assistant/Lecturer at the key laboratory of traffic safety on track of Ministry of Education of China, Central South University, China.
- ✧ **2005-2008:** Master of Engineering in Mechanical and Control Engineering (Honors I, State Medal, the top one student in all department master students), Central South University, China.
- ✧ **2004-2005:** Engineer in Bureau of Railway of Shanghai City, China.
- ✧ **2000-2004:** Bachelor of Engineering in Mechanical and Control Engineering (Honors I, University Medal, the top 5% student in all department undergraduate students), Central South University, China.

## 3. Research Interests:

- ✧ Mobile Robotics;
- ✧ Laboratory Automation;



- ✧ Wind Engineering & Wind Power System;
- ✧ Hybrid Artificial Intelligence;

#### **4. Experiences as Reviewer:**

- ✧ Journal Paper Reviewers:
  - IEEE Transaction on Power Systems (IF: 2.678)
  - IET Renewable Power Generation (IF: 1.74)
  - International Journal of Electrical Power & Energy Systems (IF: 3.432);
  - Applied Energy (IF: 5.106)
  - Neurocomputing (IF: 1.634)
- ✧ Conference Paper Reviewers:
  - IEEE CASE 2012, IEEE PECON 2012, IEEE ICOEIS 2012, etc.

#### **5. Awards:**

- ✧ International Eni Award 2013 Nomination. Nominated by Eni Award Scientific Secretariat, 2012.
- ✧ Excellent Scientist in CAE Key Project (Topic: Internet of Things & Traffic Engineering). Awarded by Chinese Academy of Engineering (CAE), 2011.
- ✧ China Mao Yisheng Outstanding Research Award for Postgraduate in Engineering. Awarded by Chinese Technology & Development Council, 2007.
- ✧ State Excellent Student Medal. Awarded by Department of Public Education of Hunan Province (State), 2008.
- ✧ State Honors I Graduate Medal. Awarded by Department of Public Education of Hunan Province (State), 2008.
- ✧ State Excellent Master Thesis Award. Thesis Title: Strong Wind Speed Signal Modeling & Forecasting Optimization Algorithms for Qinghai-Tibet Railway Safety Protecting System. Awarded by Academic Committee of Hunan Province (State), 2008.
- ✧ Mittal Excellent Engineering Student Research Medal. Awarded by England Arcelor Mittal Steel Company & Central South University, 2008.
- ✧ University Excellent Postgraduate Award. Awarded by Central South University, 2006 and 2007.

## 6. Publications:

### ● 2013:

- ✧ **Liu H.;** Tian H.Q.; Pan D.F., Li Y.F.: Forecasting models for wind speed using wavelet, wavelet packet, time series and artificial neural networks. *Applied Energy*, 2013, 107(7): 191-208. DOI: 10.1016/j.apenergy.2013.02.002. **(IF: 5.106, SCOPUS & SCI Thomson)**
- ✧ **Liu H.;** Tian H.Q.; Chen C., Li Y.F.: An experimental investigation of two Wavelet-MLP hybrid frameworks for wind speed prediction using GA and PSO optimization. *International Journal of Electrical Power and Energy Systems*, 2013, 52(11): 161-173. DOI: 10.1016/j.ijepes.2013.03.034. **(IF: 3.432, SCOPUS & SCI Thomson)**
- ✧ **Liu H.;** Stoll, N.; Junginger S., Thurow, K.: Mobile Robot for Life Science Automation. *International Journal of Advanced Robotic System*, 2013, 10(7): 1-14. DOI: 10.5772/56670. **(IF: 0.821, SCOPUS & SCI Thomson)**
- ✧ **Liu H.;** Stoll, N.; Junginger S., Thurow, K.: A Fast Method for Mobile Robot Transportation in Life Science Automation, *IEEE I2MTC-International Instrumentation and Measurement Technology Conference*, Minneapolis (USA), 6-9, May, 2013, pp. 238-242. **(SCOPUS)**
- ✧ **Liu H.;** Stoll, N.; Junginger S., Thurow, K.: An Application of Charging Management for Mobile Robot Transportation in Laboratory Environments, *IEEE I2MTC-International Instrumentation and Measurement Technology Conference*, Minneapolis (USA), 6-9, May, 2013, pp. 435-439. **(SCOPUS)**

### ● 2012:

- ✧ **Liu H.;** Tian H.Q.; Li Y.F.: Comparison of two new ARIMA-ANN and ARIMA-Kalman hybrid methods for wind speed prediction. *Applied Energy*, 2012, 98(10): 415-424. DOI: 10.1016/j.apenergy.2012.04.001. **(IF: 5.106, SCOPUS & SCI Thomson)**
- ✧ **Liu H.;** Chen C.; Tian H.Q.; Li Y.F.: A hybrid model for wind speed prediction using empirical mode decomposition and artificial neural network. *Renewable Energy*, 2012, 48(12): 545-556. DOI: 10.1016/j.renene.2012.06.012. **(IF: 2.978, SCOPUS & SCI Thomson)**
- ✧ **Liu H.;** Tian H.Q.; Chen C.; Li Y.F.: Hybrid wind speed forecasting models based on wavelet, time series, genetics and BP neural networks. *Information Sciences*, 2012. **(IF: 3.291, SCOPUS & SCI Thomson, Accepted)**

- ✧ **Liu H.;** Stoll, N.; Junginger S., Thurow, K.: A Common Wireless Remote Control System for Mobile Robots in Laboratory, Proceedings, IEEE I2MTC-International Instrumentation and Measurement Technology Conference, Graz (Austria), 13-16, May, 2012, pp. 688-693. **(SCOPUS)**
- ✧ **Liu H.;** Stoll, N.; Junginger S., Thurow, K.: A Floyd-Dijkstra Hybrid Application for Mobile Robot Path Planning in Life Science Automation, Proceedings, IEEE CASE-8th IEEE International Conference on Automation Science and Engineering, Seoul (Korea), 20-24, August, 2012, pp. 275-280. **(SCOPUS)**
- ✧ **Liu H.;** Stoll, N.; Junginger S., Thurow, K.: A Floyd-Genetic Algorithm Based Path Planning System for Mobile Robots in Laboratory Automation, Proceedings, IEEE ROBIO-IEEE International Conference on Robotics and Biomimetics, Guangzhou (China), 11-14, December, 2012, pp. 1550-1555. **(SCOPUS)**

● **2011 and before:**

- ✧ **Liu H.;** Tian H.Q.; Chen C.; Li Y.F.: A hybrid statistical method to predict wind speed and wind power. Renewable Energy, 2010, 35(8): 1857-1861. DOI: 10.1016/j.renene.2009.12.011. **(IF: 2.978, SCOPUS & SCI Thomson)**
- ✧ **Liu H.;** Tian H.Q.; Li Y.F.: Short-term forecasting optimization algorithms for wind speed along Qinghai-Tibet railway based on different intelligent modeling theories. Journal of Central South University of Technology (English Edition), 2009, 16(4): 690-696. DOI: 10.1007/s11771-009-0114-3. **(SCOPUS & SCI Thomson)**
- ✧ **Liu H.;** Tian H.Q.; Li Y.F.: Short-term forecasting optimization algorithm for unsteady wind speed signal based on wavelet analysis method and neural networks method. Journal of Central South University of Technology (Science and Technology), 2011, 42(9): 2704-2711. **(SCOPUS)**
- ✧ **Liu H.;** Tian H.Q.; Li Y.F.: Short-term forecasting optimization algorithm for wind speed from wind farms based on wavelet analysis method and rolling time series method. Journal of Central South University of Technology (Science and Technology), 2010, 41(1): 370-375. **(SCOPUS)**
- ✧ **Liu H.;** Pan D.F.; Li Y.F.: Qinghai-Tibet Railway Gale Forecasting Optimization Model and Algorithm Based on Train Running Safety. Journal of Wuhan University of Technology (Traffic & Transportation Edition), 2008, 32(6): 986-989. **(EI Compendex)**
- ✧ Pan D.F.; **Liu H.;** Li Y.F.: Optimization Algorithm of Short-term Multi-step Wind Speed Forecast. Proceedings of the CSEE, 2008, 28(26): 87-91. **(SCOPUS)**
- ✧ Pan D.F.; **Liu H.;** Li Y.F.: A Short-Term Forecast Method for Wind Speed along Golmud-Lhasa Section of Qinghai-Tibet Railway. China Railway Science, 2008, 29(5): 128-133. **(EI Compendex)**

## 7. Authorized Patents

- ✧ Title: An intelligent wind speed forecasting method for wind farms (17-Feb-2012) PRC Invention Application Publication (Source: SIPO); Publication No. CN 102609766A published on 25-Jul-2012; Application No. CN 201210036118.0 filed on 17-Feb-2012
- ✧ Title: An intelligent and hybrid method to predict wind speed along high-speed railway (17-Feb-2012) PRC Invention Application Publication (Source: SIPO); Publication No. 102609788A published on 25-Jul-2012; Application No. CN 201210036109.1 filed on 17-Feb-2012
- ✧ Title: Method for forecasting wind speed (02-Dec-2009) PRC Invention Application Publication (Source: SIPO); Publication No. CN 101592673 published on 02-Dec-2009; Application No. CN 200910009302.4 filed on 18-Feb-2009
- ✧ Title: Method and system for measuring and analyzing noise in train (17-Feb-2010) PRC Invention Application Publication (Source: SIPO); Publication No. CN 101650221 published on 17-Feb-2010; Application No. CN 200910169986.4 filed on 14-Sep-2009
- ✧ Title: Determining method and measuring system of safety retreat distance of side personnel under action of train wind (17-Feb-2010) PRC Invention Application Publication (Source: SIPO); Publication No. CN 101650255 published on 17-Feb-2010; Application No. CN 200910169985.X filed on 14-Sep-2009

Hui Liu

Rostock, 22 March 2013

## Theses

1. A mobile robot based intelligent control system is developed for the indoor transportation among distributed automated ‘islands’ in laboratory environments.
2. The developed transportation system is integrated in the whole process of the laboratory automation. It communicates with the Process Management System (PMS) to get the presented transportation tasks and finish them automatically.
3. A new controlling/organizing strategy is proposed for the system development. In this strategy there are four control levels and corresponding software centers, the PMS in the highest level for the transportation request, the RRC (Robot Remote Center) in the second level for the transportation management, the RBC (Robot On-board Center) in the third level for the transportation motion execution and the RAC (Robot Arm Center) for the arm execution.
4. A hybrid network is designed for the communication of the transportation. In this network there are two sub-networks proposed, the RIN (Robot Internal Network) for the communication between the robot RBCs and the inside robotic hardware modules and the REN (Robot External Network) for that between the RBCs and the RRC. This hybrid architecture is flexible and extendable.
5. To let the robots run a long distance and decrease the possibilities of data transmission jam caused by the real-time image monitoring, an improved method using the marine wireless bridge is adopted for the communication between the RBCs and the RRC.
6. A new method is presented for the robot indoor localization. This method consists of a ceiling landmark based global localization and a floor landmark based local localization. The StarGazer module and the Microsoft Kinect sensor are used to detect the ceiling and the floor landmarks, respectively.
7. A hybrid framework is developed for the transportation path planning. This framework is based on using both of the map theory based algorithms and the artificial intelligence based algorithms. It can handle any size of transportation maps.
8. A hybrid method is provided in the map theory based path planning. The Floyd algorithm is used to do off-line computation for the shortest paths for any pair of transportation positions and the Dijkstra algorithm is utilized to do on-line computation to find an alternative path when the Floyd paths are unavailable.
9. In the artificial intelligence based path planning, two representative algorithms are included, the Genetic Algorithm (GA) and the Artificial Ant Colony

Algorithm (AACA). At the same time a comparison of them is provided in this dissertation. The compared results show that the performance of the AACA is better than that of the GA in this case.

10. The map theory based path planning is suitable for the transportation tasks with a small number of defined waypoints. It has been applied for the P2P transportation in this study.
11. The artificial intelligence based path planning is designed for the transportation cases with a big number of defined waypoints. It has been adopted for the patrol transportation in this application.
12. An improved method is proposed for the laboratory robot indoor collision avoidance. In the method the Microsoft Kinect sensor is integrated in the system to measure a farer obstacle and recognize the shapes of the obstacles. The improved method can avoid the dead-locking phenomenon of the APF in mobile robotics.
13. To decrease the possibilities of two-robot collision in the transportation, besides the function of local collision avoidance a global strategy considering the whole transportation process is also presented for the collision avoidance.
14. A control software platform has been developed using the Microsoft C#.Net framework. It includes the RRC, the RBCs, the RACs and their GUIs & APIs. Based on this platform a laboratory transportation activity can be organized conveniently.
15. A My-SQL based network database is established to store the important information of the transportation system. Based on this database a series of statistical computation can be made.
16. A series of XML-based protocols is designed for the API controls among the different software centers.
17. A series of transportation experiments has been made in this dissertation to verify the performance of the developed system. The experimental results show that the proposed transportation strategy and its corresponding system are effective and efficient for the life science automation.

## Abstract

In modern laboratories, more and more complicated transportation is desired among distributed automatic workbenches and rooms. In this dissertation a new method is presented to control and organize the transportation in life science environments by using kinds of mobile robots. The new innovative points of this method are listed as follows:

- ✧ Firstly, a flexible system is developed for this method, including four control levels: the Process Management System (PMS) for the transportation request, and Robot Remote Center (RCC) for the transportation management, the Robot Boarding Center (RBC) for the transportation motion execution and the Robot Arm Component (RAC) for the transportation arm execution. This system architecture suits for any sizes of laboratories and any kinds of mobile robots.
- ✧ Secondly, a hybrid network is proposed for the data communication of the transportation. It consists of a Robot Internal Network (RIN) and a Robot External Network (REN). The former is for the robot boarding measurement and the latter for the communication between the robots (RBCs) and the RRC. An option by equipping one kind of wireless bridges between the RRC and the RBCs is also provided for the possible super long-distance transportation and huge data transmission.
- ✧ Thirdly, a new method is presented for the robot indoor localization in laboratory environments. It is comprised of two working components: ceiling landmark-based global one and floor landmark-based local one. The first one considers the expansibility and the economy of the localization and the second one focuses on the accuracy of the localization. They can cooperate together to obtain the satisfactory robot positioning performance for the transportation.
- ✧ Fourthly, a new computational framework is designed for the path planning of the transportation. It includes two working modes: map theory-based peer-to-peer one and artificial intelligence-based patrol one. In the first kind, a hybrid method using the Floyd algorithm and the Dijkstra algorithm is presented. In the second one, a method adopting both of the Genetic Algorithm (GA) and the Artificial Ant Colony Algorithm (AACA) is proposed. Based on this path planning frameworks, the transportation with different numbers of waypoints can be executed easily and effectively.
- ✧ Fifthly, a new method is proposed to improve the performance of collision avoidance in the adopted H20 robots. In this method one kind of depth sensors named Microsoft Kinect is adopted to detect the obstacles and a global-based strategy considering the whole path planning is proposed for the collision avoidance. This method can decrease the possibilities of transportation collisions.

- ✧ Sixthly, a completed method is developed to manage the robot recharging in the transportation. It is comprised of three aspects: an embedded software component is programmed for the robot real-time power measurement; a fully-automated charging station is discussed and a computational algorithm using the Artificial Immune Algorithm (AIA) is put forward to optimize the best installing positions of the charging stations in laboratories.
- ✧ Finally, a group of experiments is provided to demonstrate and verify the performance of the developed transportation system. The results show that the presented methods and developed centers for the mobile robot transportation in life science laboratories are effective and efficient.



## Zusammenfassung

In modernen Labors werden Transportaufgaben immer komplexer durch verteilte automatisierte Workstations, Systeme und Anlagen. In dieser Dissertationsschrift wird ein neues Verfahren vorgestellt, das Steuerung und Organisation von Transportaufgaben in Life Science Laborumgebungen und den Einsatz mobiler Robotik realisiert. Die neuen innovativen Eigenschaften dieses Verfahrens sind nachfolgend aufgeführt:

- ✧ Erstens es wurde ein flexibles System für dieses Verfahren entwickelt, das vier Steuerungsebenen beinhaltet: Das Prozess Management System (PMS) für Transportaufträge, das Robot Remote Center (RRC) für das Transportmanagement, das Robot On-board Center (RBC) für die Transportsteuerung und die Robot Arm Component (RAC) für die Steuerung der Roboterarme. Diese Systemarchitektur kann angewendet werden für jede Laborgröße und beliebige Formen von mobilen Robotern.
- ✧ Zweitens wird eine hybride Netzwerkstruktur für die Transport-Datenkommunikation vorgeschlagen. Sie besteht aus einem Robot Internal Network (RIN) für interne Komponentenkommunikation und Robot External Network (REN) für die Kommunikation zwischen Robotern (RBCs) und dem Robot Remote Center. Für gebäudeweite, unterbrechungsfreie Drahtlosnetzwerkverbindungen der Roboter wurden wireless peer-to-peer Bridges eingesetzt.
- ✧ Drittens wird ein Verfahren der Indoor Lokalisierung in Laborumgebungen präsentiert. Es besteht aus zwei Komponenten: Globale Navigation anhand von Landmarken an der Labordecke sowie lokale Navigation anhand von Fußbodenmarken. Während die erste Komponente die gebäudeweite effektive und erweiterbare Lokalisation ermöglicht, dient die zweite Komponente zur genauen Positionierung der Roboter an charakteristischen Arbeitsstationen. Beide Komponenten arbeiten kooperierend, um gemeinsam eine hohe Positionierungsperformance zu erreichen.
- ✧ Viertens wurde ein neues Softwareframework für die Pfadplanung der Transporte erarbeitet. Dieses beinhaltet zwei Arbeitsmodi: Graphentheorie basierte Punkt zu Punkt Verfahren und KI basierte Bewegungsverfahren. Erstere beinhalten Floyd- und Dijkstra Algorithmen, letztere nutzen Genetische Algorithmen (GA) und Artificial Ant Colony Algorithmen (AACA). Unter Nutzung und Kombination dieser Methoden in einem Framework können Transportprozesse mit verschiedenartiger Art und Anzahl von Wegpunkten leicht und effektiv realisiert werden.
- ✧ Fünftens wird ein Verfahren zur Verbesserung der Kollisionsvermeidung unter Nutzung der vorhandenen Roboterplattform H20 vorgestellt. Bei diesem Verfahren wird unter Nutzung der Microsoft Kinect Plattform mit Laserscanner –

Kamera Kombination die globale Pfadplanung mit der Funktionalität der Kollisionsvermeidung kombiniert. Dieses Verfahren hilft, Transportprobleme durch Kollisionen zu vermeiden.

- ✧ Sechstens wurde das automatische Energiemanagement mit eigenständiger Wiederaufladung der Akkus der Roboter an Ladestationen entwickelt. Dieses nutzt drei Aspekte: Integrierte interne Roboter Echtzeit Energiemessung, automatisierte Ladestationen und die Bestimmung der günstigsten Installationspositionen für Ladestationen unter Verwendung eines Artificial Immune Algorithmus (AIA).
- ✧ Abschließend wird eine Reihe von Experimenten dokumentiert, mit der die Performance des entwickelten Transportsystems erprobt und demonstriert wurde. Die Ergebnisse dieser Untersuchungen zeigen, dass das entwickelte System und seine Steuerumgebung für die Transportprozesse unter Verwendung mobiler Robotik in Life Science Laborumgebung effektiv und effizient sind.