Mobile Robot-Assisted Material Handling in Prefabricated Housing Factories

Tingsong Ou

Advisor: Daniel Cardoso Llach Ph.D., Associate Professor

August 4, 2023

Abstract

The prefabricated housing industry in the U.S. aims to relieve the shortage of affordable housing by manufacturing housing components off-site and assembling them on-site to increase project time efficiency and reduce the total cost. However, the industry still heavily depends on human work, and lacks of automation due to its cost-conscious nature. To find a proper way to introduce automation to the current labor-intensive manufacturing process, I first visited two prefabricated housing factories in the northeast U.S. to take a deeper look into their manufacturing process and material flow to find a proper task that can be automated. Based on the site visits, adding financial and practical considerations, the thesis proposes a system that uses mobile robots to assist with the material handling tasks. The solution includes robot hardware design and material transport strategy, algorithms of pathfinding/line-following, and collision avoidance methods. Due to the complexity of the project and the nature of this as a research thesis, the solution does not aim to deliver a complete, market-ready solution, but a proof of concept of a mobile robot-assisted material handling system able to handle materials of various sizes, plan the shortest path between stops, and agilely navigate in the narrow factory space without colliding with obstacles and pedestrians. The outcomes include a physical robot that passes the maneuverability test and a virtual simulation that puts all components together to prove the viability of the system. Finally, the thesis discusses the limitations of the project and opens up discussion on the possible future directions for refining the system.[1]

[1] The design files and source code are available at https://github.com/terrence-ou/MSCD-Thesis-Mobile-Robot-Material-Handling

Acknowledgements

I would like to express my deepest gratitude to the following individuals who have played a significant role in the completion of this master's thesis.

First and foremost, I am immensely grateful to my thesis advisor, Professor Daniel Cardoso Llach. Your unwavering guidance, expertise, and encouragement throughout this journey have been invaluable. Your insights and constructive feedback have shaped the direction of this research and have truly enhanced the quality of my work. I am thankful for your patience, dedication, and belief in my abilities.

I am also express my gratitude to the members of the team Rethinking Automation in Construction (ReAC), Emek Erdolu, Yuning Wu, Jiaying Wei, (and also the broken Fetch robot), for their insightful comments and valuable suggestions in the days and nights when we're exploring the future of automation in construction together.

Finally, I want to extend my heartfelt appreciation to my wife, Shaoxian Yu, who have stood by my side throughout this journey. Your words of encouragement, listening ears, and uplifting presence have been a source of strength during both my highs and lows. I am privileged to have you in my life.

Contents

1 Introduction **1**
- 1.1 Prologue . . . 1
- 1.2 Statement of Problem . . . 2
- 1.3 Outline of Chapters . . . 2
- 1.4 Significance . . . 2

2 Background **4**
- 2.1 Prefabricated Housing Industry . . . 4
  - 2.1.1 Prefabricated houses . . . 4
  - 2.1.2 The overview of the industry . . . 8
  - 2.1.3 The industry in the U.S. . . . 11
- 2.2 Site Visit and Manufacturing Process Study . . . 13
  - 2.2.1 Site visits . . . 13
  - 2.2.2 The manufacturing process and material flow . . . 18
- 2.3 Mobile Robotics in Construction and Material Handling . . . 24
  - 2.3.1 Mobile robot on construction sites . . . 24
  - 2.3.2 Mobile robot in other industries . . . 26
- 2.4 Summary . . . 29

3 Hypothesis **30**
- 3.1 Project Positioning . . . 30
- 3.2 Proposed Scenario . . . 31

4 Methods **34**
- 4.1 Robot Design . . . 35
  - 4.1.1 System architecture . . . 36
  - 4.1.2 Circuit design . . . 37
  - 4.1.3 PID control . . . 39
  - 4.1.4 Hardware test . . . 40
- 4.2 Robot-cart Connection . . . 45
  - 4.2.1 Hypothesis setup . . . 45
  - 4.2.2 Connection design . . . 47
  - 4.2.3 Potential for multi-robot collaboration . . . 51
- 4.3 Path Planning . . . 51
  - 4.3.1 Path planning algorithm . . . 52
  - 4.3.2 Graph-based path-planning in the factory . . . 54
- 4.4 Line-following . . . 57
  - 4.4.1 The line-following mechanism . . . 58
  - 4.4.2 Line-following experiment . . . 60

4.4.3 The limitation of the current setup . . . 62
4.5 Collision Avoidance . . . 63
4.5.1 Strategy . . . 63
4.5.2 Hardware setup and virtual simulation . . . 64
4.5.3 The limitation of the current method . . . 68

5 Experiment **69**
5.1 Virtual Simulation . . . 69
5.1.1 Environment and task setup . . . 69
5.1.2 Simulation . . . 70
5.1.3 The problems in the simulation . . . 77
5.2 Moving Base Development . . . 77
5.2.1 Challenges . . . 78
5.2.2 Updated robot design . . . 80
5.3 Maneuverability test . . . 84

6 Conclusion **91**
6.1 Conclusion . . . 91
6.2 Contribution . . . 91
6.3 Limitations . . . 92
6.4 Future Works . . . 93

A Videos **95**
A.1 Chapter 3 Hypothesis . . . 95
A.2 Chapter 4 Methods . . . 95
A.3 Chapter 5 Experiments . . . 95

Bibliography **96**

List of Figures

2.1 The “palette” of industrialized building system. Source: Richard 2017 5
2.2 caption for h67 5
2.3 Statistical Data of Habitat ‘67. Source: Safdie 1967 6
2.4 Nakagin Capsule Tower. Source: Lin 2011 7
2.5 Star Apartments, Los Angeles, U.S.A. Source: Iwan Baan 7
2.6 Star caption 8
2.7 Toyota caption 8
2.8 Seksui caption 9
2.9 Randek wall system 9
2.10 Randek sheathing system 10
2.11 LHC NH3 project 11
2.12 LHC NH3 construction 11
2.13 Median Housing Price in the U.S. (2013-2022). Data source: United States Census Bureau. Diagram by the author 12
2.14 Number of registered manufactured housing factories by state. Data source: Institute for Building Technology and Safety. Diagram by the author 13
2.15 Preferred Building Systems - factory location. Source: Satellite image from Google Earth, diagram by the author 15
2.16 Preferred Building Systems - factory layout. Source: Satellite image from Google Earth, diagram by the author 15
2.17 Moments in the wood factory. Source: photos by the author 16
2.18 Full Stack Modular - factory location. Source: Satellite image from Google Earth, diagram by the author 17
2.19 Full Stack Modular - factory layout. Source: Satellite image from Google Earth, diagram by the author 17
2.20 Moments in the steel factory. Source: photos by the author 18
2.21 “Moments” and material flow in the factory. Source: Satellite image from Google Earth, diagram by the author 19
2.22 Bulk material storage. Source: photo by the author 19
2.23 General storage space. Source: photo by the author 20
2.24 Deck framing station. Source: diagram by the author 21
2.25 Wall framing station. Source: diagram by the author 21
2.26 Module assembling station. Source: diagram by the author 22
2.27 Wiring and insulation installation station. Source: diagram by the author 23
2.28 Furnishing and wrapping station. Source: diagram by the author 23
2.29 Conceptual framework of the BIM-integrated mobile robot on the construction site. Source: Follini et al. 2020 24
2.30 Hardware setup for BIM-integrated site-monitoring robot. Source: Follini et al. 2020 . 25
2.31 Dusty 25

2.32 Dusty . . . 26
2.33 Main differences between AMRs, AGVs, and AGCs. Source: Costa Barros and Nascimento 2021 . . . 26
2.34 Path-following solutions for AGVs. Source: Ullrich et al. 2015 . . . 27
2.35 Diagrams of typical AGV chassis. Source: Ullrich et al. 2015 . . . 27
2.36 AGV help transporting e-motors. . . . 28
2.37 AGVs are integrated into the assembling process. . . . 28
2.38 A fleet of small robots transport light-weighted carts. . . . 29

3.1 Target materials for the automated material handling. Source: diagram by the author 31
3.2 Project test bed - a compact version of the factory in NH. Source: diagram by the author 32
3.3 The robot picks up the cart. Source: diagrams by the author. Video: Appendix A.1 . 32
3.4 The robot transports the cart to the destination. Source: diagrams by the author. Video: Appendix A.1 . . . 33
3.5 The robot leaves the cart. Source: diagrams by the author. Video: Appendix A.1 . . . 33

4.1 The structure of this chapter. Source: diagram by the author . . . 35
4.2 System architecture. Source: diagram by the author . . . 36
4.3 Circuit design. Source: diagram by the author . . . 38
4.4 Arduino . . . 39
4.5 PID control loop. Source: diagram by the author . . . 39
4.6 Testing circuit design on a single motor. Source: photo by the author . . . 41
4.7 Direction feedback. Source: diagrams by the author . . . 41
4.8 Speed feedback. Source: diagrams by the author. Video: Appendix A.2 . . . 42
4.9 PID test with $P = 1.0, I = 0.0, D = 0.0$. Source: diagram by the author . . . 42
4.10 PID test with $P = 5.0, I = 0.0, D = 0.0$. Source: diagram by the author . . . 43
4.11 PID test with $P = 5.0, I = 0.025, D = 0.0$. Source: diagram by the author . . . 43
4.12 The motor follows the sinusoid signal from the controller. Source: diagram by the author 44
4.13 4WD setup. Source: photo by the author . . . 44
4.14 Preliminary test on 4WD setup. Source: photo by the author. Video: Appendix A.2 . 45
4.15 Diagram of the bulk storage space in the factory. Source: diagram by the author . . . 46
4.16 Hypothetic target object. Source: diagram by the author . . . 47
4.17 Method 1 - robot lifts the cart. Source: diagram by the author . . . 48
4.18 Method 2 - connect with hinges. Source: diagram by the author . . . 49
4.19 Method 3 - using friction pads. Source: diagram by the author . . . 50
4.20 Proposed multi-robot collaboration in transporting heavy material. Source: diagram by the author . . . 51
4.21 A* algorithm flow chart. Source: diagram by the author . . . 52
4.22 Checkpoints in the factory. Blue dots are waypoints, and yellow pins are loading stations. Source: diagram by the author . . . 55
4.23 Graph checking program helps to check the connectivity of each point. Source: diagram by the author. Video: Appendix A.2 . . . 55
4.24 The path selection interface and the visualized point-to-point shortest path. Source: diagram by the author . . . 56
4.25 Multi-step path-finding results. Source: diagrams by the author. Video: Appendix A.2 57
4.26 Line-following system flow chart. Source: diagram by the author . . . 59
4.27 The line-following system helps the robot correct deviations. Source: diagram by the author . . . 60
4.28 The line-following system helps the robot handle corners. Source: diagram by the author 60
4.29 Deviation check. Source: photos by the author. Video: Appendix A.2 . . . 61
4.30 Simple line-following test. Source: photos by the author. Video: Appendix A.2 . . . 62
4.31 Slametic . . . 65

4.32 The 2D LiDAR installed on top of the robot. Source: diagram by the author 65
4.33 A person shows up near the path but beyond the robot's danger radius. Source: diagram by the author 66
4.34 When LiDAR detects an obstacle that is closer than the safety threshold, the robot stops until the path is clear again. Source: diagram by the author 67
4.35 A person shows up close to the robot but opposite to the robot's current moving direction; this situation is considered safe, and the robot keeps moving. Source: diagram by the author 67

5.1 Environment for the virtual simulation. Source: diagram by the author 70
5.2 The control center finds the shortest path from the robot station via the storage to the destination. Source: diagram by the author 71
5.3 The robot follows the first path, moves toward the storage space. Source: diagram by the author 72
5.4 The robot turn around to face the path when arriving at the storage space. Source: diagram by the author 73
5.5 The robot slides beneath the cart and connect with it. Source: diagram by the author 73
5.6 The robot carrying the material slides back to the path. Source: diagram by the author 74
5.7 The robot carries the material and follows the second part of the path. Source: diagram by the author 75
5.8 The robot stopped when detecting a possible collision. Source: diagram by the author 75
5.9 The robot resumes line-following when the path is clear again. Source: diagram by the author 76
5.10 The robot drops off the cart and returns to the path. Source: diagram by the author . 77
5.11 The robot powered by batteries. Source: photo by the author 78
5.12 Indoor short-distance straight-line test. Source: diagram by the author 79
5.13 Long-distance straight-line test 1. Source: diagram by the author 79
5.14 Long-distance straight-line test 2. Source: diagram by the author 80
5.15 Testing motor coordination. Source: photo by the author. Video: Appendix A.3 . . . 81
5.16 Testing motor coordination. Source: diagram by the author. Video: Appendix A.3 . . 82
5.17 Finished moving base, front. Source: photo by the author 83
5.18 Finished moving base, back. Source: photo by the author 83
5.19 Finished moving base, motors and battery. Source: photo by the author 84
5.20 Finished moving base, wiring and electronics layout. Source: photo by the author . . . 84
5.21 Wheels control for forward/backward moving. Source: diagram by the author 85
5.22 Forward moving test. Source: diagram by the author 85
5.23 Wheels control for sideways moving. Source: diagram by the author 86
5.24 Sideway moving test. Source: diagram by the author 86
5.25 Wheels control for diagonal moving. Source: diagram by the author 87
5.26 Diagonal moving test. Source: diagram by the author 87
5.27 Wheels control for turning around. Source: diagram by the author. Source: diagram by the author 88
5.28 Turning around test. Source: diagram by the author. Source: diagram by the author . 88
5.29 Wheels control for cornering. Source: diagram by the author 89
5.30 Cornering test. Source: diagram by the author 89
5.31 Wheels control for turn of axis. Source: diagram by the author 90
5.32 Turn of axis test. Source: diagram by the author 90

6.1 The possible future works. Source: diagram by the author 93

List of Algorithms

1 PID control loop 40
2 A* Path-planning 53
3 Graph-building 54

Chapter 1

Introduction

This chapter begins with a brief introduction to the background and the problems of automation in the prefabricated housing industry in the U.S. and proposes a research question for this thesis about finding a viable method to introduce automation to the industry. The second section outlines the topic and content of each chapter, and the final section discusses the significance of this thesis.

1.1 Prologue

Prefabricated building is a notion that existed as early as the late 19 century that moved the construction activity off-site; the idea was originally for military purposes that can deploy camps quickly on battlefields (Lidelöw 2017). After the mid-20s', there were more and more practices of prefabricated housing around the world for purposes like reducing construction time and cost, material and energy efficiency, fast population growth due to urbanization, or pure architectural experiments. Nowadays, the prefabricated housing industry has become an important solution to housing challenges, and the industry in different countries/regions grows independently based on the industry level and specific housing problems each place is facing. The industry in the United States aims to address the shortage of affordable housing (Razkenari et al. 2020) but has a weak connection to the manufacturing industries due to limited profits. Therefore there's almost no automation in U.S. prefabricated housing factories.

The construction labor shortage (U.S. Bureau of Labor Statistics 2023) may bring negative effects on the prefabricated housing industry as there might be insufficient workers in factories and reducing the construction speed and the number of homes produced in the same unit time; this might increase the price of modular houses and thus making the houses getting expensive and less affordable. To address this foreseeable challenge, this thesis aims to propose a plan to automate tasks in prefabricated housing factories in the U.S. However, due to the price of the manufactured houses, the industry is budget sensitive that may not be able to change the current factory facilities and automate the manufacturing process on a large scale, so instead of proposing an ambitious automation plan to upgrade the factory or to replace the laborers with robots, making a case for the potential of supporting the off-site construction work is the primary concern of this thesis (ReAC 2022).

The automobile manufacturing industry and automated warehouses provide successful examples of using mobile robots to help with material handling in large spaces, and introducing mobile robots for such purposes does not require a significant change in the current layout and management. Therefore, the thesis proposes a mobile robot system that can adapt to the current factory setup and automate the material handling task, making it a starting point for future automation in the prefabricated housing industry.

1.2 Statement of Problem

The research question of this thesis is: how to use mobile robots to assist material handling in the prefabricated housing factory? The problem actually includes two aspects:

- Project positioning: The thesis should figure out what types of materials can be handled by mobile robots. If an object is already transported by existing machinery or humans efficiently, using the robot to handle it becomes a redundant and less meaningful choice. To solve this problem, it is necessary to look into the factory's current manufacturing process and material flow, finding which step is less efficient that can be improved by the mobile robot.
- Applicability: The proposal is based on the assumption that it does not require a significant investment in robot development, and the solution should not be too revolutionary and requires a large-scale upgrade of the current factory setup.

1.3 Outline of Chapters

- Chapter 2 first explores the examples of modular housing projects and the features of the prefabricated housing industry in different countries. Then the chapter shows the result of site visits to two factories in the northeast U.S. and a detailed manufacturing process and material flow study on one of them. The third part of this chapter includes examples of the use of mobile robots in both the construction site and the automobile manufacturing industry.
- Chapter 3 presents the hypothesis of the thesis, including the project positioning, the virtual factory space for simulations, and a proposed scenario of using a Mecanum-wheeled robot to pick up and deliver material in the factory.
- Chapter 4 provides methods to achieve the hypothetic scenario, including robot hardware design and test, path planning algorithms, line-following mechanism, and collision avoidance strategy. These methods follow the idea of proof of concept to prove the viability of the system rather than providing highly practicable real-world solutions.
- Chapter 5 shows the virtual simulation of the mobile robot-assisted material handling system in a virtual prefabricated housing factory space and presents the challenges faced in the hardware development ad their solutions.
- Chapter 6 makes a conclusion of the thesis, analyzes the contributions and limitations of the project, and drafts the possible future works.

1.4 Significance

The hypothetic significance of this thesis consists of two aspects: one is to propose a possible plan of automation in a prefabricated housing factory, and the other is to open up an opportunity to attract more people from other fields, like robotics, to explore more possibilities in automation in this industry.

In terms of the plan of automation, the thesis aims to provide an applicable plan to introduce mobile robot-assisted automated material handling into the factory that can adapt to the factory's current operation and layout without significant investment and does not conflict with the current operation.

A robotics project typically requires a collaboration of various disciplines, a feat less likely to be accomplished by a single person. Therefore, this thesis does not aim to propose a complete solution for automated material handling. Instead, it seeks to build a framework primarily based on knowledge of architecture, engineering, and manufacturing management. The hope is that this framework can attract more specialists with interdisciplinary visions to further exploit the potential of this topic or

explore more opportunities in the automation of modular housing manufacturing. In doing so, it could bring about a revolution in this field

Chapter 2

Background

This chapter consists of background research on the prefabricated housing industry and mobile robotics. The prefabricated housing industry research starts with case studies of modular housing projects and the industry in different countries, then visits two factories in the northeast U.S., and picks one of them to study the manufacturing process and material flow in the factory further. The mobile robotics part includes examples of the use of mobile robots in both construction scenarios and manufacturing industries and technical literature relating to the mobile robot system and hardware setups.

2.1 Prefabricated Housing Industry

Prefabricated housing was more like an experimental practice in the past century and is still not considered mainstream in nowadays' housing market due to on-site construction dominating construction methods in most of the world. The prefabricated housing industry is more like manufacturing than construction, so it is influenced by each country's industrialization level and shows different features in different regions.

2.1.1 Prefabricated houses

Richard 2017 conducted research on industrialized buildings around the world and proposed that the major difference between the traditional approach and the industrialized one is that the latter streamlines the construction process by utilizing standardized details that can be reused for multiple buildings. The research also created a "palette" that categorized industrialized buildings into 10 types (Figure 2.1).

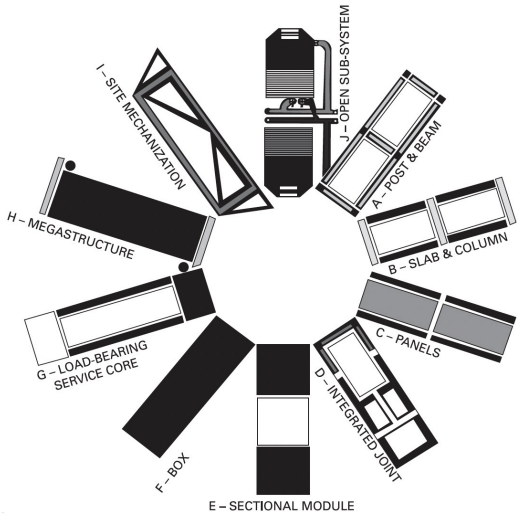


Figure 2.1: The "palette" of industrialized building system. Source: Richard 2017

Prefabricated housing can be considered as an industrialized building whose most units are substantially completed in the factory, leaving a small amount of work to be completed on-site (Gibb 1999).

The prefabricated house is not a new concept; an early example of fully modularized housing can date back to 1967, when Habitat '67 was built as a part of World Expo 1967 in Montreal, Canada, and the building is still functioning well now. The modules are made of concretes that are precast in steel molds, and furnishings are done in an assembly-line method. The finished modules are transported to the site and lifted by crane into position (Safdie 1967)(Figure 2.2). The project also provides statistical data on the list of components and the amount of material used (Figure 2.3); it is obvious from the list that the components a represented as "units" that the project was treated as an industrialized product rather than a regular building.



(a) Habitat '67 birdview



(b) Habitat '67 construction

Figure 2.2: Habitat '67 in Montreal, Canada. Source: Safdie Architects n.d.

Discussion of this paper is invited. Please forward your discussion to PCI Headquarters before May 1 to permit publication in the August 1967 issue of the PCI JOURNAL.

HABITAT '67—STATISTICAL DATA

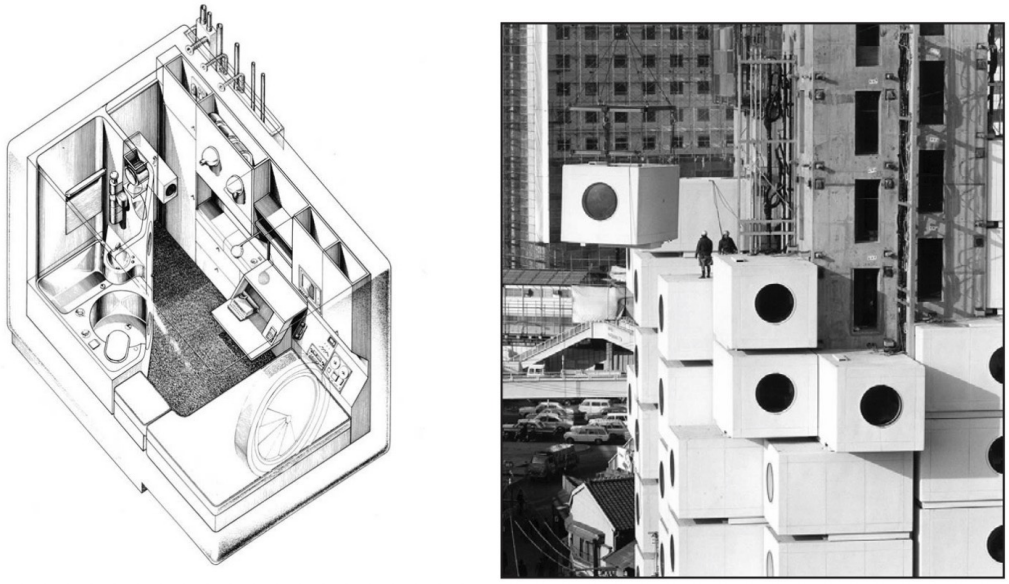
Number of house units	354
Number of street girders	18
Number of cantilever girders	6
Number of stair shafts	7
Number of elevator shafts	6
Number of precast columns	24
Number of walkway bridges	4
Number of expanded base piles	1120
Total weight of reinforcing steel	4,000 tons
Total volume of concrete	26,000 cu. yd.
Total length of prestressing steel (bars and cables)	145,600 lin. ft.
Average concrete strength	6,400 psi
Dry-packed concrete, 7-day strength	9,800 psi

Associated Architects: Moshe Safdie and David, Barott, Boulva
Structural Engineers: Dr. August E. Komendant and Monti, Lavoie, Nadon
Mechanical Engineer: Huza & Thibault
Electrical Engineer: Nicholas Fodor & Associates
General Contractor: Anglin-Norcross Quebec, Ltd.
Precast Concrete: Francon (1966) Ltd.

66 PCI Journal

Figure 2.3: Statistical Data of Habitat '67. Source: Safdie 1967

Another well-known case is the Nakagin Capsule Tower in Tokyo, Japan, designed by Kiso Kurokawa and finished in 1972. The building was designed with the idea of Metabolism, which treats the city and building as "organic" objects that are not static and are themselves processes. The building consists of on-site casted concrete cores and off-site manufactured living units, namely "capsules" (Lin 2011). The capsules are even further standardized than the Habitat '67 in that all units have the same size of 2.3 m × 3.8 m × 2.1 m. The interior is designed using industrial technology that integrates the furniture and appliances into the capsule. The living units are manufactured in the factory, transported to the site, and assembled to the concrete core with the help of cranes (Figure 2.4).



(a) Axonometric of the Capsule (b) The construction of Nakagin Capsule Tower

Figure 2.4: Nakagin Capsule Tower. Source: Lin 2011

One of the recent examples of prefabricated housing is the Star Apartments in Los Angeles, U.S., designed by Michael Maltzan, and completed in 2013. The Star Apartments project is an affordable housing project that provides 102 apartments for the homeless (Michael Maltzan Architecture 2015). The apartment consists of 102 prefabricated 300-square-foot wood-frame modules with completed furnishing and appliance installed. It took seven weeks to manufacture the modules and another six weeks to transport them to the site and assemble them on the precast foundation (Amelar 2015) (Figure 2.5, 2.6).



Figure 2.5: Star Apartments, Los Angeles, U.S.A. Source: Iwan Baan



Figure 2.6: The construction of Star Apartments. Source Skid Row Housing Trust 2013

2.1.2 The overview of the industry

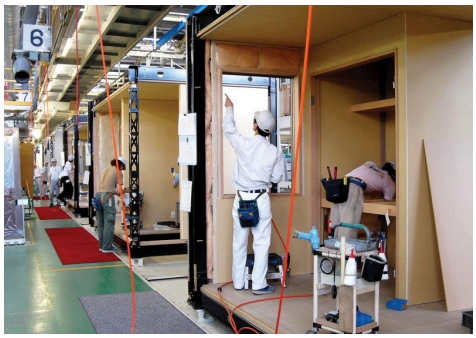
As mentioned earlier in this chapter, the prefabricated industry differs in regions and countries. Such distinctive may be due to several reasons like the use of labor, cultural difference, land price, or the level of industrialization (Gibb 1999).

Japan

Japan started putting prefabricated housing into mass production in the late '20s (Bock 2007). A unique phenomenon of the industry in Japan is that modular housing is sometimes a department of major industry leaders, including automobile giants like Mitsubishi[1] and Toyota[2] (Figure 2.7), consumer electronic giant Panasonic[3], and chemical giant Sekisui[4](Figure 2.8). The advantage is that these companies have solid technology foundations and plenty of resources that can transfer to manufacturing housing units with a high automation level.



(a) Toyota prefabricated home product.



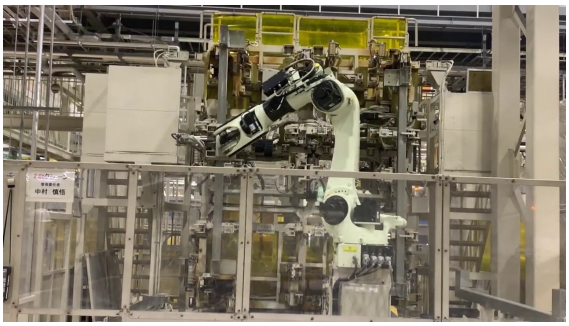
(b) Toyota Home factory.

Figure 2.7: Toyota prefabricated homes. Source: (left) Toyota Home, (right) Yuri Kageyama

[1]Mitsubishi Home website: https://www.mitsubishi-home.com/smartorder/smartorder-fit/
[2]Toyota Home website: https://www.toyotahome.co.jp/
[3]Panasonic Home website: https://homes.panasonic.com/english/
[4]Sekisui website: https://www.sekisuichemical.com/products/search/1364241_38638.html



(a) Sekisui prefabricated home product



(b) Automation in Sekisui factory

Figure 2.8: Sekisui prefabricated homes. Source: (left) Sekisui Chemical, (right) Kawasaki Robotics

The factors that drive Japanese companies to invest in prefabricated housing are limited land, lack of labor, and frequent natural disasters like typhoons and earthquakes in the country; prefabricated housing helps increase the construction speed without heavily relying on human labor and can quickly deploy comfortable homes to where the natural disaster happened (Buntrock 2017).

Sweden

Sweden started exploring prefabricated homes in 1840, and the architect Fredrik Blom made the first attempt for military purposes (Lidelöw et al. 2017). Unlike Japan, Sweden has a smaller population, around 10.43 million by the end of 2021, about one-tenth of Japan's population but larger land areas. Therefore, the driving force behind developing prefabricated housing in the market is not land price or overcrowded urban areas.

The steady process of industrialization over the span of 250 years, government support, and social awareness of green energy work together to build a solid foundation for off-site construction in that country (Lawrence and Spybey 2018, Lidelöw 2017). Sweden's renowned off-site construction company includes Skanska[5], Lindbäcks Bygg[6], Randek[7] (Figure 2.9, 2.10), etc.

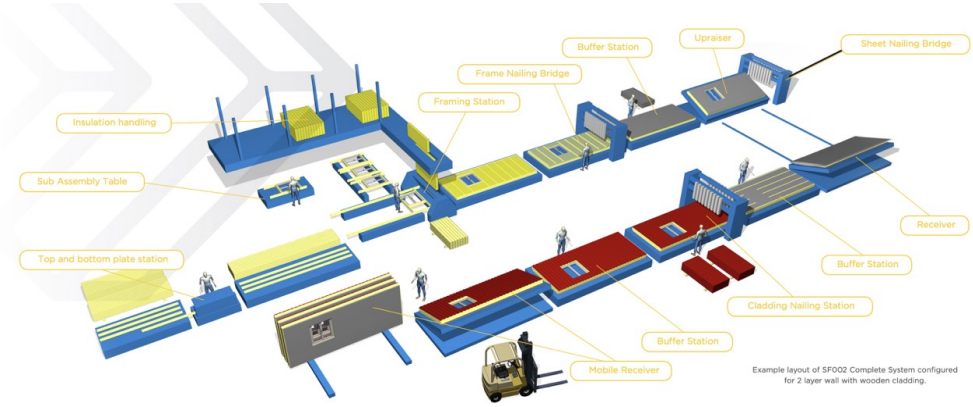


Figure 2.9: Wall system of Randek. Source: Randek

[5]Skanska website: https://group.skanska.com/projects/homes/
[6]Lindbacks website https://lindbacks.se/
[7]Randek website: https://www.randek.com/en/

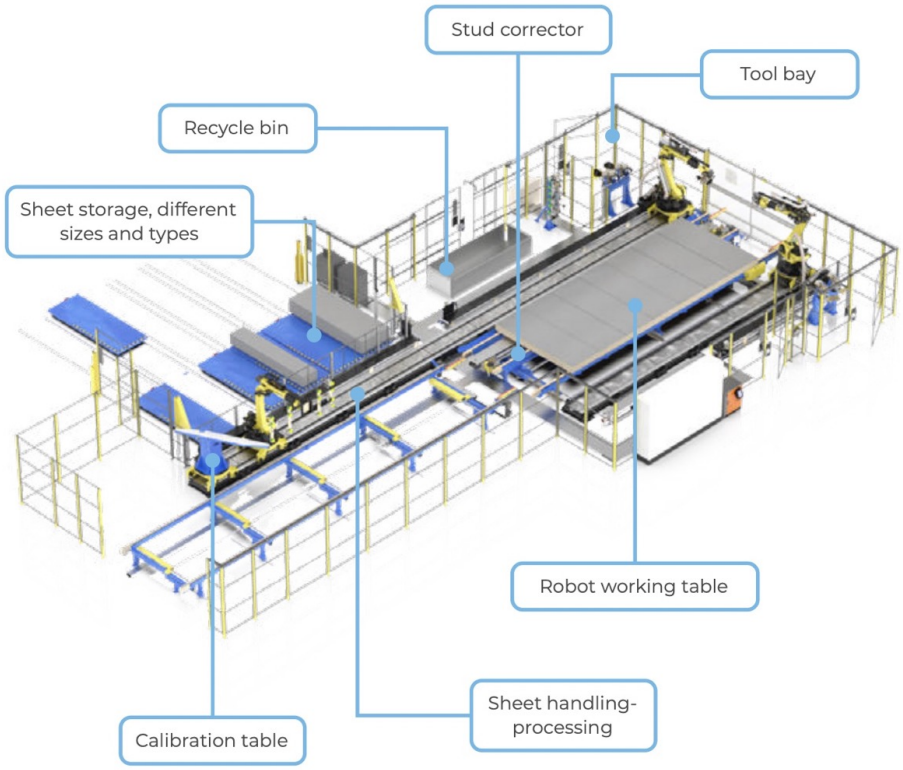


Figure 2.10: ZeroLabor Robotic Sheathing System of Randek. Source: Randek

The United Kingdom

Opposite to Japan and Sweden, the prefabricated housing industry in the UK does not driven by the industry and population but is primarily based on based on a concern with environmental context and social accessibility.

The rainy weather in the UK makes the construction in a sheltered environment, and the good silviculture practice makes the timber resource plenty in that region and lowers the cost of building houses with the local timber. The offsite construction practice in that region aims to reduce the carbon dioxide emission during the lifespan of the building to reduce its environmental impact and lower the cost of construction cost to ensure the modular houses' social accessibility (Hairstans and Sanna 2017).

The practice of off-site construction in the UK was mainly conducted by the collaboration among the architect's office, the manufacturer, and the building system companies (Hairstans and Sanna 2017); now, companies like Connect Modular[8] can manage projects and deliver products independently (Figure 2.11, 2.12).

[8]Connect Modular website: http://connectmodular.co.uk/index.htm



Figure 2.11: LHC NH3 Framework by Connect Modular. Source: Scottish Construction Now 2023



Figure 2.12: Construction of LHC NH3 Framework project. Source: Scottish Construction Now 2023

2.1.3 The industry in the U.S.

The industry in the US is similar to the industry in the UK, which does not have a close connection to the manufacturing industry and aims to solve the shortage of affordable housing shortage by reducing the cost of construction and achieving a faster completion (Razkenari et al. 2020). The US Department of Housing and Urban Development named the prefabricated houses as Manufactured homes and defined them as "homes built with a permanent chassis, which ensures initial and continued

transportability, with an area of at least 320 square feet" (U.S. Department of Housing and Urban Development 2023).

The US Census Bureau published a census record of the median housing price in the US between 2013 and 2022 and categorized the housing into built and manufactured housing (Figure 2.13). In the chart, the blue line shows the trend of (on-site) built housing price trend, and the orange one is that of manufactured housing. The prices of manufactured houses are around 40% of the built ones; the difference in price can be seen as proof that the industry in the US is affordable-oriented, and the industry has started growing since 2020.

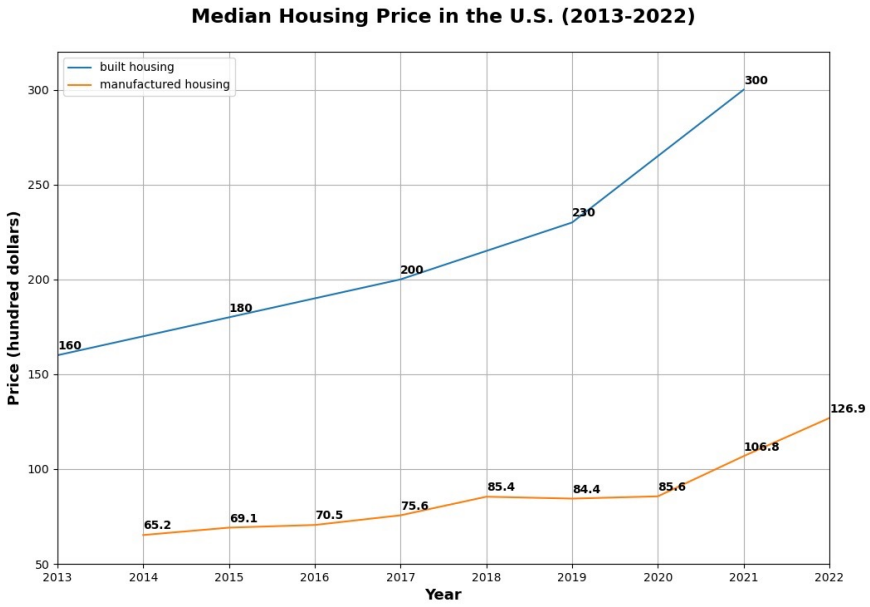


Figure 2.13: Median Housing Price in the U.S. (2013-2022). Data source: United States Census Bureau. Diagram by the author

The institute for Building Technology and Safety published the 2022 manufactured home production facilities showing that there are 143 certified prefabricated housing factories in the US (Figure 2.14). The factory distribution on the diagram indicates that factories concentrate in places with plenty of natural timber resources, which helps reduce the cost of raw material transportation, or places where natural disasters frequently happen and have high needs for post-disaster resettlements. However, the diagram does not represent the accurate factory distribution; the actual number of factories that manufacture modular housing is more than that shown in the data; many factories haven't registered in the association as it is not mandatory to do so before selling their products.

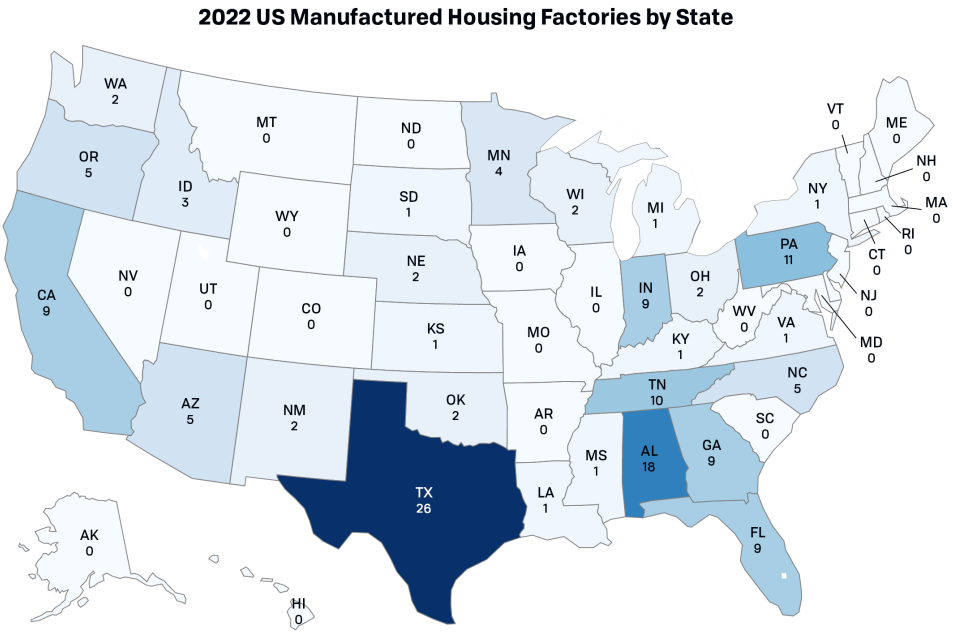


Figure 2.14: Number of registered manufactured housing factories by state. Data source: Institute for Building Technology and Safety. Diagram by the author

Some commonly seen types of prefabricated housing in the US are Wood Framed Chassis, Steel Frame Chassis, and Cross Laminated Timber (CLT). The former two types use timber/steel to construct load-bearing outer frames and fill building functionalities within the frame. The CTL, however, are panels such that prefabricated houses using this material do not follow the form of "modules"; CLT can be considered as prefabricated panels that serve as structural walls, envelop, and interior separations. In this thesis, we only focus on chassis-based prefabricated houses, as penalized houses differ fundamentally from this type. In the next section, we will look into two selected factories that produce chassis-based modules with different materials, then analyze the manufacturing process and material flow for one of them.

2.2 Site Visit and Manufacturing Process Study

Based on the literature review and the data presented in the previous section, several site visits to the current prefabricated housing factory in the northeast U.S. was conducted to gain first-hand materials of the manufacturing and the products in the region. Then the thesis picks one of the factories to study the factory operation further to understand better the manufacturing process and the material flow in the factory to figure out the potential task in the process that can be automated.

2.2.1 Site visits

Visiting the current prefabricated housing factories, observing the manufacturing activity, and interviewing the workers are the most straightforward and productive ways to study how the modular houses were produced, figure out the problem in the current factory operation mode, and excavate the potential of automating the task in the whole manufacturing process. However, hundreds of such factories in the U.S. produce types of modular houses with different materials; a good strategy of choosing sites helps to gain more information with fewer visits.

Three factors can help narrow down the factory selections: location, material, and modular type.

- **Location**: Fewer travel helps to save time and budget for site visits. Thus the geological scope of factory selection is narrowed in the northeast U.S. With the factory information provided by Triumph Modular[9] and Technical.ly[10], the geological scope can further narrow down to New York and New Hampshire.
- **Material**: Only two types of material are used in producing prefabricated houses in the United States - timer and steel. Although there are subcategories of timber modules like wood frame and cross-laminated timber, and subcategories of steel house products like structural steel or steel chassis, focusing on the major category helps reduce the complexity of formulating the research problem. Therefore, visiting one wood housing factory and one steel housing factory could be sufficient to understand the factory layout of such an industry.
- **Modular types**: A prefabricated house can be designed as a stacked modular chassis or assembled modular blocks. The former method is a combination of fully-functional house units; architects design house modules as well as the method of stacking them up to be a complete house. The latter is inverse to the chassis house - the architects design the whole house at first and then strategically divide their design into modules. In this context, visiting one chassis factory and modularized blocks factory helps efficiently study the manufacturing of different types of modules.

Therefore, the site selections become clear: selecting two factories in New York and New Hampshire, one manufactures wood modules, and the other produces steel products; One of the factory's products are chassis-based modules, while the other concentrates on modularized blocks. Preferred Building Systems and Full Stack Modular Ilc are the two factories that satisfy these presets.

Preferred Building Systems, New Hampshire

The Preferred Building Systems manufactures wood chassis modules. The factory locates on the border of New Hampshire and Vermont, taking advantage of approximating plenty of wood resources in this region (Figure 2.15). The factory consists of an office space, a showroom, and a manufacturing space that takes up the majority area of the factory. The raw material enters the factory from the loading dock at the northwest of the plant, and the finished product comes out to the yard to the southeast of the building (Figure 2.16).

[9] Triumph Modular website: https://www.triumphmodular.com/
[10] Technical.ly website: https://technical.ly/

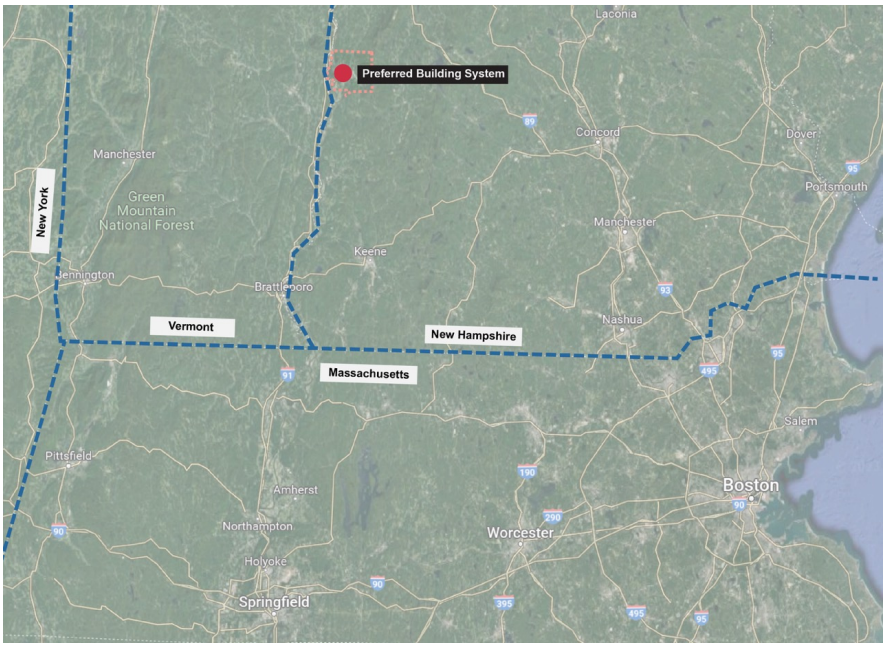


Figure 2.15: Preferred Building Systems - factory location. Source: Satellite image from Google Earth, diagram by the author

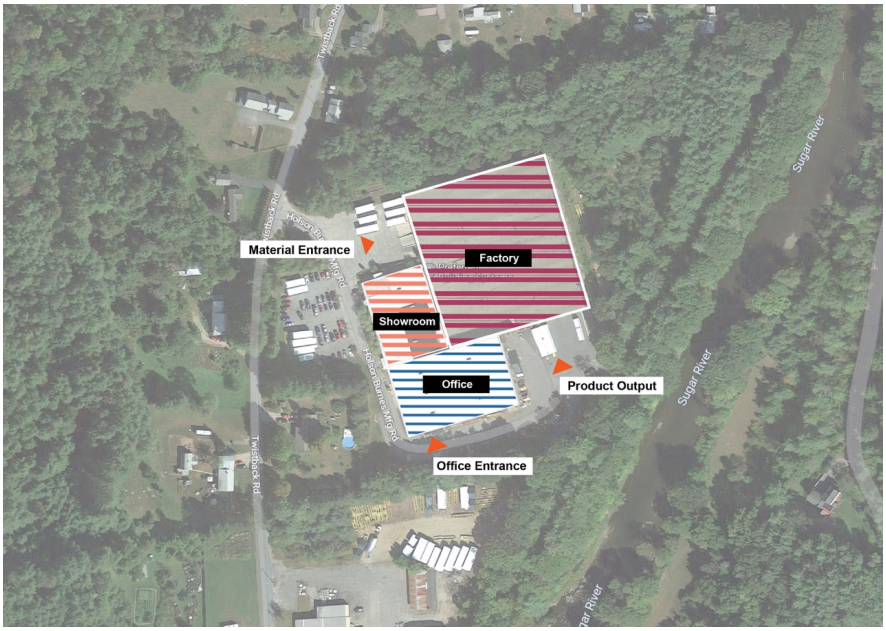


Figure 2.16: Preferred Building Systems - factory layout. Source: Satellite image from Google Earth, diagram by the author

The manufacturing line in the factory consists of several stations like material pre-processing, framing, assembling, and furnishing. There're three primary methods of transporting materials and products between stations: The overhead bridges crane lifts and moves heavy wood frames and panels, the

worker transport small-to-medium-sized material with carts with various loading capabilities, and loader uses wheels with hydraulic lifters to load the finished module to the truck's trailer. A clean and wide path connects different stations, allowing laborers, vehicles, and carts to move efficiently (Figure 2.17).

As of the date of the visit, there is no sign of automation in the manufacturing of this factory. On the contrary, the human is primarily responsible for moving and installing the components; coordinating different processes mainly relies on person-to-person communications.



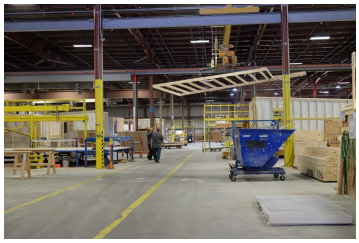
(a) Bulk material storage



(b) Building component storage



(c) Framing station



(d) Crane transporting a panel



(e) Module assembling station



(f) Furnishing station

Figure 2.17: Moments in the wood factory. Source: photos by the author

Full Stack Modular Ilc., New York

Full Stack Modular Ilc manufactures steel modularized building components, which have a smaller size than the chassis ones. The factory locates in the Navy Yard in Brooklyn, New York (Figure 2.18). Due to the narrow shape of the site, the factory's area and layout are not as spread as the previous one; the plan is a regular long rectangle along a deck with the office located at one end near the parking lot. There's no dedicated showroom in the factory; the modules in the whole process are open to visitors if it is not under work; there's a finished module near the office side that displays the finished product ready to deliver to the customer. There is only one entrance on the factory side, where materials and products come in and out through it (Figure 2.19).



Figure 2.18: Full Stack Modular - factory location. Source: Satellite image from Google Earth, diagram by the author

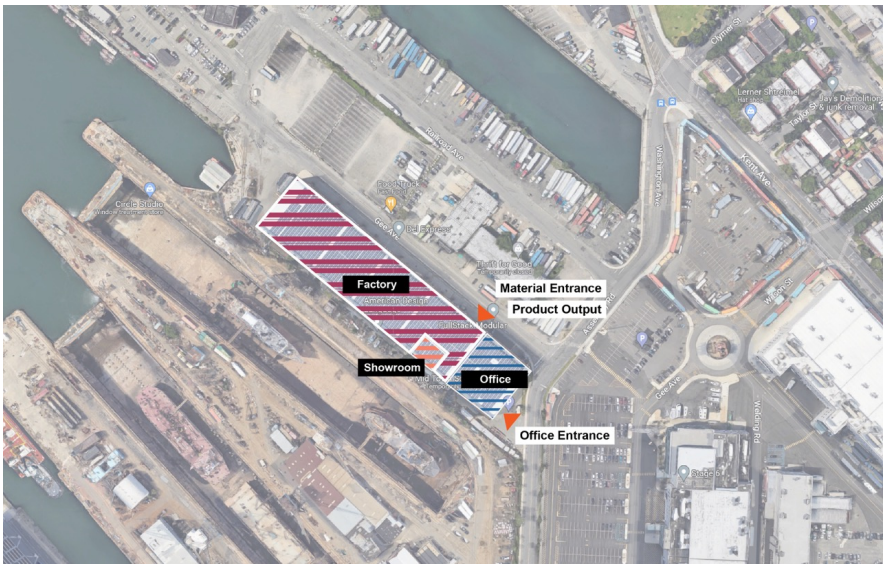


Figure 2.19: Full Stack Modular - factory layout. Source: Satellite image from Google Earth, diagram by the author

The factory does not have clear boundaries between processes after the framing deck; many small modules can be completed from scratch in the same place. Material like pipes, wires, steel poses, etc., are stored on the shelf placed along the wall and column array along the long-axis. Like the wood factory, transporting materials and products relies on humans, carts, and heavy machinery. The difference is that, as the steel modules are smaller, the wall panel and frames can be transported

with carts without the help of heavy machinery. A wheeled crane in the plant moved modules, either finished or unfinished, around inside the factory (Figure 2.20).

As of the visit date, this factory also has no sign of automation throughout the whole production process. Material handling and manufacturing coordination are in humans' hands.



(a) Framing deck



(b) Steel storage



(c) Carts



(d) Module in-progress



(e) Steel frame



(f) Wheeled crane

Figure 2.20: Moments in the steel factory. Source: photos by the author

2.2.2 The manufacturing process and material flow

A closer look at the manufacturing process helps identify in which part the automation can help make production more efficient in terms of time and human resources. The material flow is one of the most straightforward representations of the manufacturing process because each step takes the product from the previous step, plus additional resources needed, and outputs the intermediate product closer to the final one. Therefore, keeping an eye on each step's input and output materials is beneficial for a better understanding of how the module has been produced in the process.

The factories introduced in the previous section provide plenty of first-hand data for learning the process. However, choosing one with more complexity in the factory layout and product assembly could help to find more potential in introducing automation into the current operation. With this concept in mind, the wood factory would be a better choice due to the relatively complex layout, clear boundary between steps, and manufacturing process that requires more effort to coordinate with each other.

The overview of the manufacturing process

This factory has eight *"stations"* - three storage spaces and five manufacturing moments. The layout generally adapts to the factory's structure (Figure 2.21). The pre-processing station takes in raw materials, like oak wood and gypsum boards, machines the timbers into wood studs and trusses, cuts the gypsum boards into defined sizes, then stores the processed materials in the bulk storage space (Figure 2.22). The deck/wall framing station worker takes studs from the bulk storage and assembles them into wood frames. After that, frames were transported to the module assembling station and built as raw modules. Next, workers process and furnish the modules and turn them into a finished product in the wiring, insulation, and furnishing stations.

The general storage spaces are where most interactions take place: they constantly provide gears, supplies, and consumable materials to each station; workers blend paints and primers or process small components here (Figure 2.23).

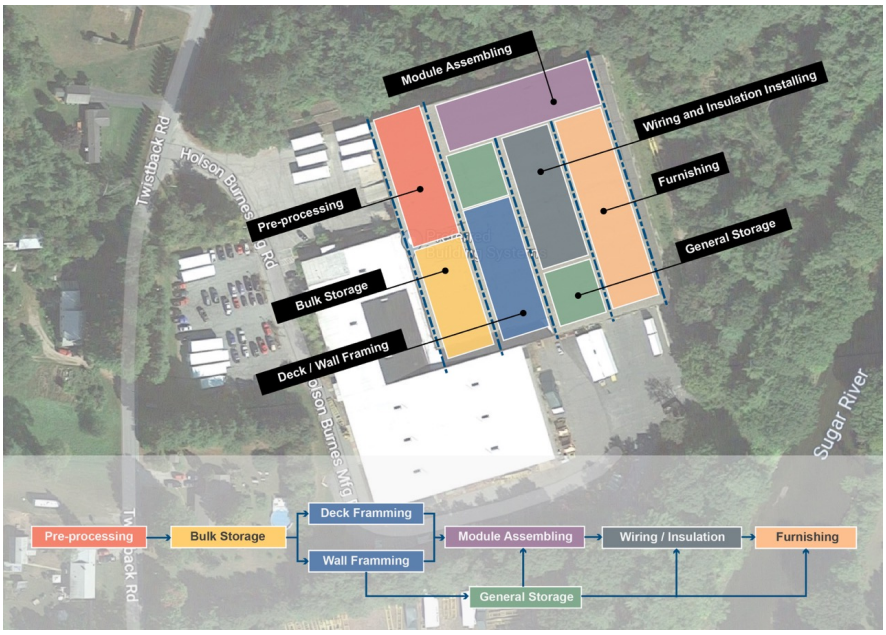


Figure 2.21: "Moments" and material flow in the factory. Source: Satellite image from Google Earth, diagram by the author



Figure 2.22: Bulk material storage. Source: photo by the author



Figure 2.23: General storage space. Source: photo by the author

The "moments"

After having a general impression of the factory layout and the overall manufacturing process, it's time to take a closer look at each " moment" to analyze the works and the input/output materials at each step. We don't consider the material preprocessing as part of the module-manufacturing process as the process is quite detached from the others - it processes raw material following requirements without interacting with other manufacturing moments. Therefore, the main process can be separated into five moments: deck framing, wall framing, module assembling, wiring and insulation installing, and furnishing and wrapping.

- **Deck framing**

  The deck framing station (Figure 2.24) consists of a framing bed and closets storing supplements like nails, glues, and gears served explicitly for this process. The overhead bridge crane starts from this place, helping to transport the finished decks to the next station. This station predeceases load-bearing timber frames like floor decks, ceilings, and structural wall frames.

  Input - thick structural-purposed timber studs, gypsum boards

  Output - load-bearing timber frames

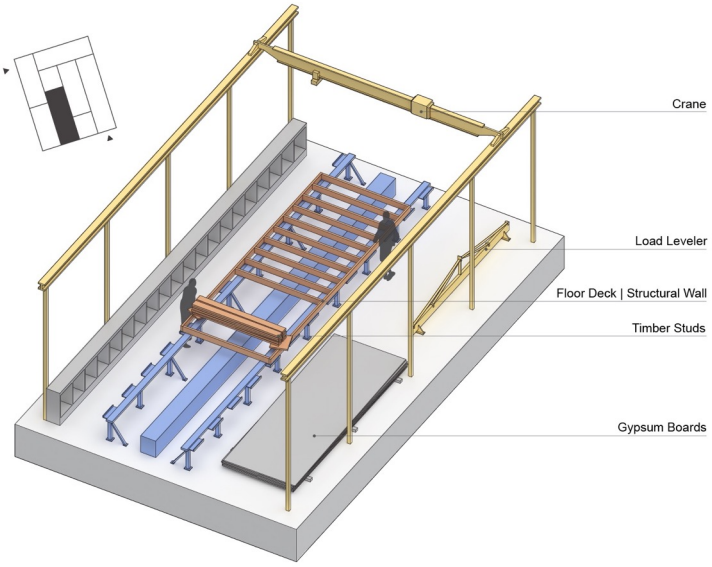


Figure 2.24: Deck framing station. Source: diagram by the author

- **Wall framing**

  There's a wide framing table in the wall framing station (Figure 2.25) where workers can assemble timber studs to build non-load-bearing wall or window frames on top of it. As the dimensions of these types of wall frames were standardized in the chassis-based modular system, this station usually assembles more standard frames than needed in the current workflow and stores the extras in the shelve in the same place or bulk storage for future purposes.

  Input - light-weighted timber studs

  Output - non-load-bearing frames and panels

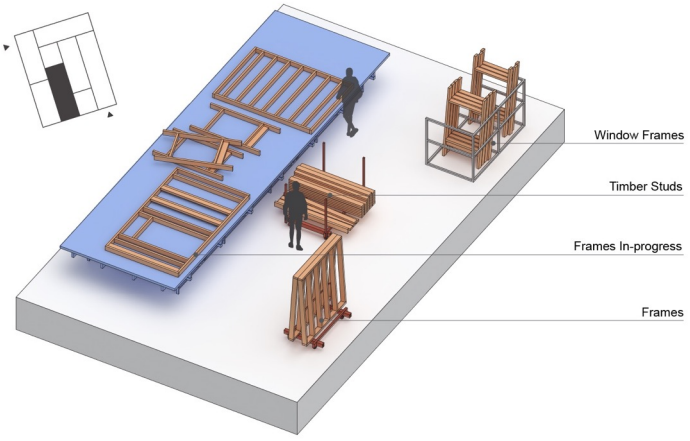


Figure 2.25: Wall framing station. Source: diagram by the author

- **Module assembling**

This station assembles decks, wall frames, and ceilings as raw modules (Figure 2.26). The crane moves heavy structural parts from the framing station, and workers transport wood panels and timber frames here with carts. The lifter in this place helps workers reach a higher position to gain a better view and working position to put the ceiling on the top of the module precisely. In addition to the wood components, gears, glues, and several supplementary materials are needed; workers grab these small items from the general storage and transport them here with tool carts.

Input - Structural and non-structural panels, wood panels, gears, and supplies

Output - Raw modules and empty supply packages and containers

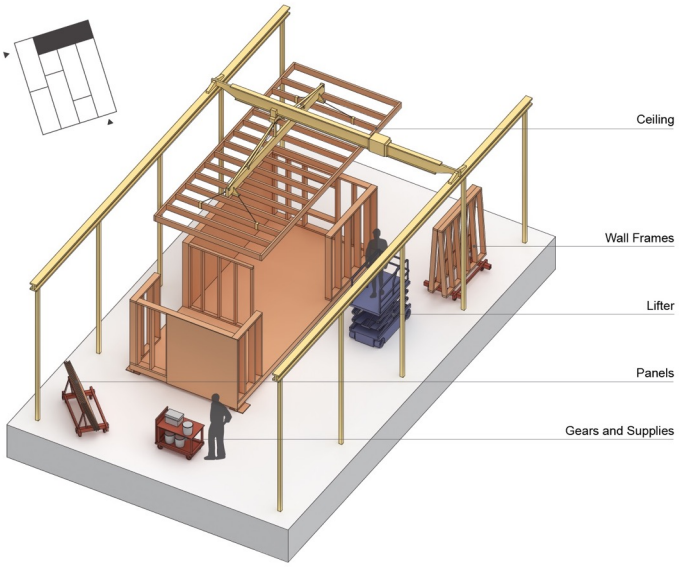


Figure 2.26: Module assembling station. Source: diagram by the author

- **Wiring and insulation installation**

  The wheels with hydraulic lifter mounted on the four corners at the bottom of the module help to move the raw module to this wiring and insulation installation station (Figure 2.27). The station has fixed scaffoldings on both sides of modules; the moveable scaffoldings on both ends of the module help provide more working spaces when the module is fixed in position. Here, workers put insulation material on the module and assemble and organize types of wirings inside it. The heavy materials no longer appear here; instead, the station needs quite a lot of transporting work to supply a large amount of work-related materials.

  Input - Wires, insulation materials, gypsum boards, wood panels, gears, and supplies

  Output - Insulated and wired modules, wastes, empty packages, and buckets

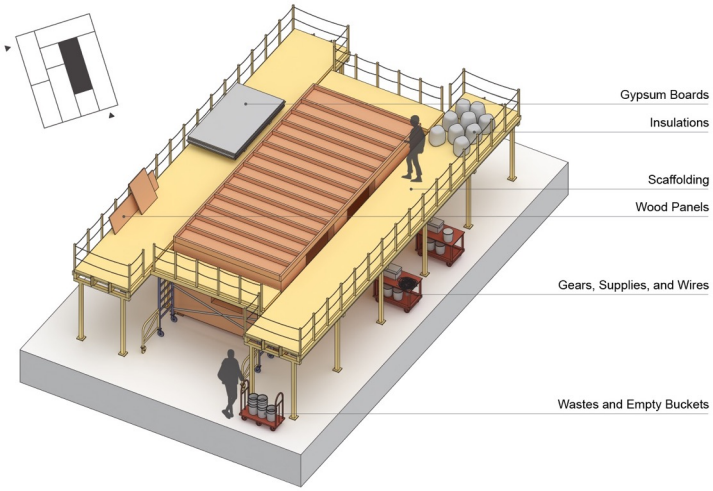


Figure 2.27: Wiring and insulation installation station. Source: diagram by the author

- **Furnishing and wrapping**

  The furnishing and wrapping is the last step of finishing the manufacturing (Figure 2.28). Like the previous step, this station does not need heavy materials but requires many transportations to move furnishing materials and furniture from the storage space to here. Workers place and fix furniture inside the module following the drawings from the design department. When all the furnishing is done and all the furniture is in place, the module will be wrapped with a layer of water-resistant plastic wrap and sent to the yard right outside the station, waiting for pickup.

  Input - insulated and wired modules, furniture, furnishing materials, windows, tubings, gears, and supplies

  Output - Finished modules, wastes, and empty containers

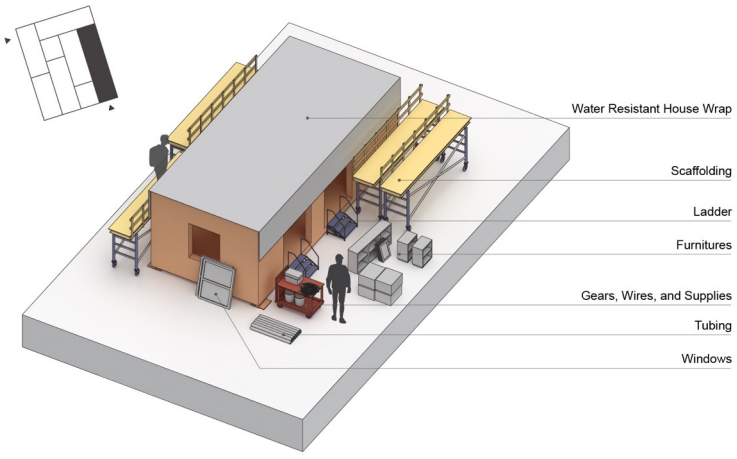


Figure 2.28: Furnishing and wrapping station. Source: diagram by the author

2.3 Mobile Robotics in Construction and Material Handling

In construction, mobile robotics are being used to automate tasks such as bricklaying, painting, and welding, and there are many examples of loading a robotic arm on the moving base to provide automated fabrication tasks with mobility, allowing the robot to work on the same job in different places on a site. However, in the prefabricated housing factory, the same task typically happens in the same place, as presented in the previous section; the automated mobile fabrication platform does not seem an proper choice for introducing mobile robots into the factory.

The prefabricated housing factories bring construction activities into factory setups, so when considering using mobile in such indoor construction scenarios, we can also look at the precedents in both manufacturing factories and construction sites.

In manufacturing, it is common to use mobile robots for material handling purposes, and they can be further categorized as Automated Guided Vehicles (AGV), Automated Guided Carts (AGC), and Autonomous Mobile Robots (AMR); they are equipped with vision or spatial sensors to help transport materials and products in large factory spaces.

2.3.1 Mobile robot on construction sites

In recent years, there has been a trend of integrating Building Information Modeling (BIM) with mobile robots to help monitor the construction progress and site maintenance. The BIM-integrated mobile robot provides some inspiration for this thesis as it takes advantage of the robots' mobility to monitor the construction progress, and in several cases, it uses the robot's localization feature to help mark or place building components on the site based on the information stored in BIM.

Follini et al. 2020 proposed a framework for using BIM-integrated mobile robots to help monitor the construction progress, support maintenance during building operations, and help with logistics during construction (Figure 2.29). The project mainly focuses on updating the BIM data by installing LiDARs and a camera on the robot (Figure 2.30), making it map the up-to-date site, detecting temporary obstacles, and providing feedback on the updated site data to the BIM when it is running on the site.

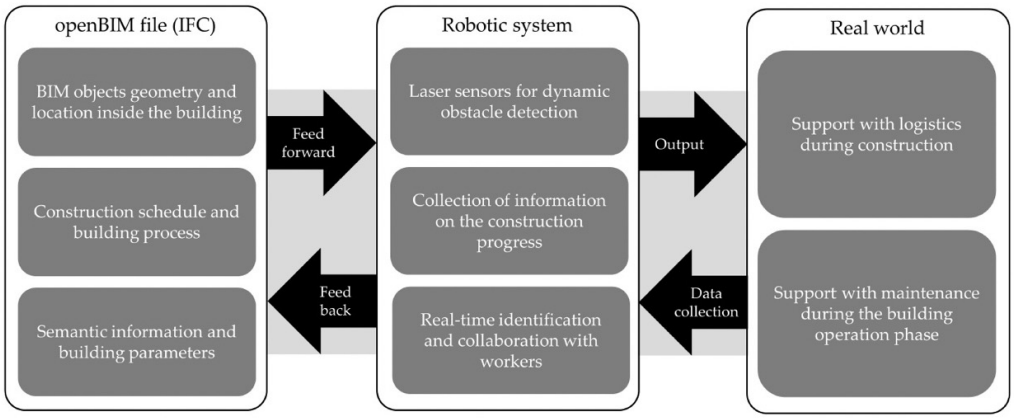


Figure 2.29: Conceptual framework of the BIM-integrated mobile robot on the construction site. Source: Follini et al. 2020

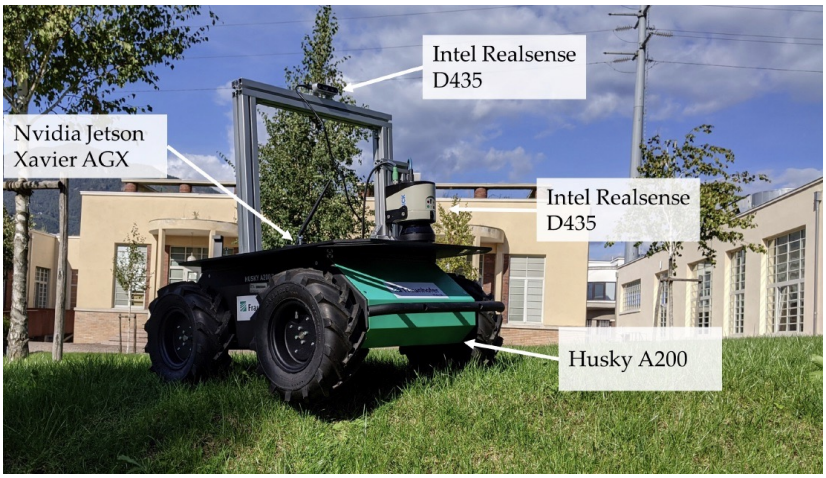


Figure 2.30: Hardware setup for BIM-integrated site-monitoring robot. Source: Follini et al. 2020

Dusty Robotics [11] developed a mobile robot, FieldPrinter (Figure 2.31), for the automation of construction site layout. The robot reads the site layout data from BIM, taking advantage of the robot's precise mobility and localization to print BIM information directly on the floors to help reduce the error from manual measurements and marks, making the task 10 times faster than done by humans (Lau 2020). HP[12] also released a similar robot, SitePrint (Figure 2.32), in 2022 for the same construction site layout task. The robot features a light-weighted design that is flexible on a variety of site surfaces like concrete, plywood, pavement, epoxy, etc.

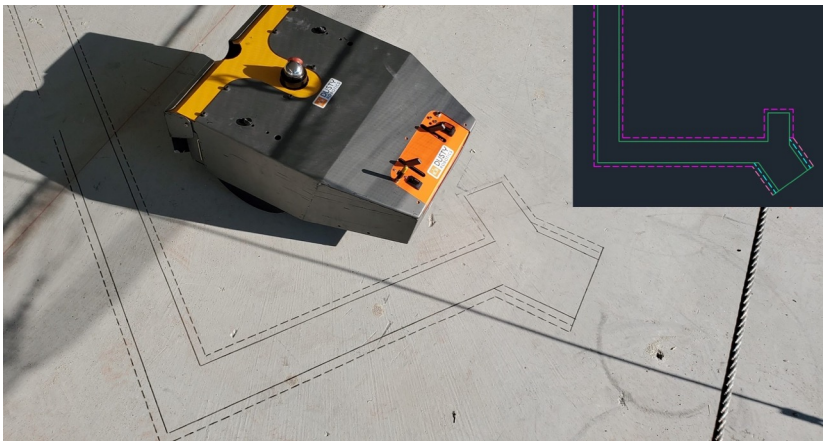


Figure 2.31: FieldPrinter by Dusty Robotics. Source: Dusty Robotics

[11] Dusty Robotics website: https://www.dustyrobotics.com/
[12] HP SitePrint website: https://www.hp.com/us-en/printers/site-print/layout-robot.html

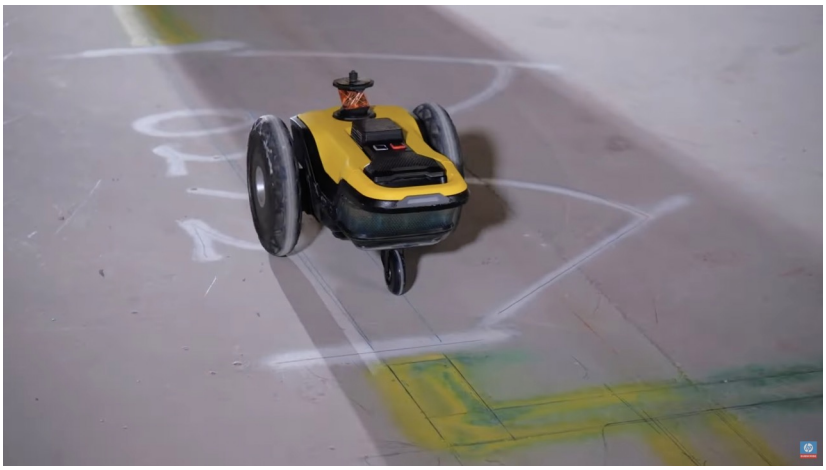


Figure 2.32: SitePrint by HP. Source: HP

2.3.2 Mobile robot in other industries

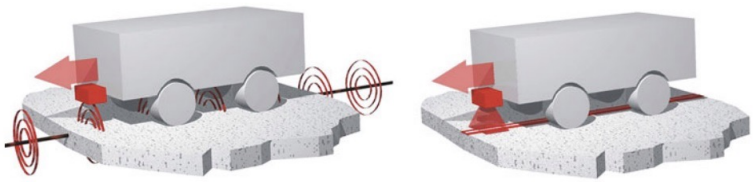
Using mobile robots in factories and warehouses for material handling purposes has a longer history than using them in construction scenarios and therefore has already established a matured system of using specific kinds of robots in warehouses/factories with different sizes for various workloads. There are three major types of material-handling purposed mobile robots in the industry: Automated Guided Vehicles (AGV), Automated Guided Carts (AGC), and Autonomous Mobile Robots (AMR), and Figure 2.33 (Costa Barros and Nascimento 2021) shows the main difference between them. In all three types, the AGV is widely used in manufacturing industries as it is capable of handling heavy materials of different sizes.

Main differences between AMRs, AGVs and AGCs.

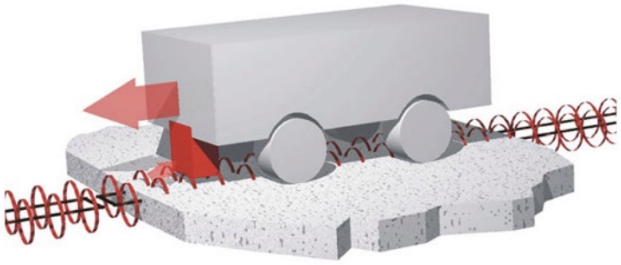
	Automated Guided Vehicle (AGV)	Automated Guided Cart (AGC)	Autonomous Mobile Robot (AMR)
Rigid preset routes	O	O	
Fixed infrastructure	O	O	
Infrastructure requirements	O	O	
Product transport	O	O	O
Picking process assistance	O	O	O
Sortation assistance			O
Redeployable			O
Intelligent			O
Modular deployment			O

Figure 2.33: Main differences between AMRs, AGVs, and AGCs. Source: Costa Barros and Nascimento 2021

Ullrich et al. 2015 provided comprehensive guidance on the technical standards and examples of applications of AGVs in the book Automated Guided Vehicle Systems; the book is very helpful when drafting an AVG-assisted material handling scenario and finding solutions for each step in operation. For instance, the book lists options and details of the physical track guidance when finding a solution for line-following, like active-inductive track, magnetic guidance track, and optical guidance track (Figure 2.34). It also provides guidance on types of chassis setups and their possible traveling motions (Figure 2.35).



(a) Magnetic guidance track (left) and optical guidance track (right)



(b) Active-inductive track

Figure 2.34: Path-following solutions for AGVs. Source: Ullrich et al. 2015

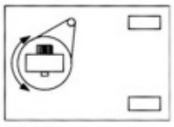
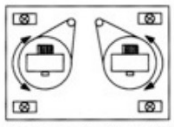
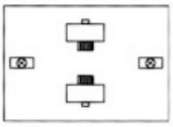
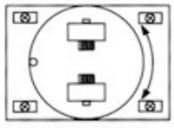
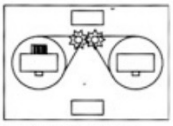
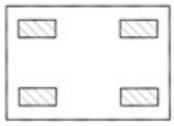
Undercarriage	Possible travelling motion	Undercarriage	Possible travelling motion
 Three-wheel	• two degrees of freedom • straight forward motion and turning about rear axle • Preferred direction of motion is forward, reverse travel is possible.	 several independent travelling/ steering units	• three degrees of freedom
 Differential	• two degrees of freedom • straight forward and reverse • Turning about centre axle is possible.	 differential with swivel axle	• three degrees of freedom
 Coupled steering drive	• two degrees of freedom • straight forward and reverse • Turning about centre axle is possible.	 Mecanum drive	• three degrees of freedom *Source: Miag Fahrzeugbau GmbH, Braunschweig*
Symbol:  fraction drive; supporting wheel; steering drive; rotatable supporting wheel (spring-loaded, if required)			

Figure 2.35: Diagrams of typical AGV chassis. Source: Ullrich et al. 2015

The Automotive Industry could probably be one of the industries that make the best use of AGVs in manufacturing; various types of mobile robots are almost everywhere in the factory to help transport materials and parts, and some of them have been integrated into the car assembling process. Looking

into an automated automobile factory can provide a lot of inspiration when thinking about using mobile robots in the prefabricated housing factory.

A car consists of components like an engine, cluster, body parts, etc. Each component is produced in a specific station in the factory, and medium-sized mobile robots manage to pick up parts from each station and transport them to the main assembly line (Figure 2.36). The main assembly line in some factories has replaced the moving floor or railed cranes with AGVs; each mobile robot carries a car-in-process, moving along the assembly line with a pre-set speed, and the robot is capable of changing its height to make chassis accessible when workers are assembling lower parts like the suspension system (Figure 2.37). There is also a fleet of smaller-sized mobile robots that are flexible in transporting light-weighted materials, supplies, or wastes between different stations when receiving requests (Figure 2.38).

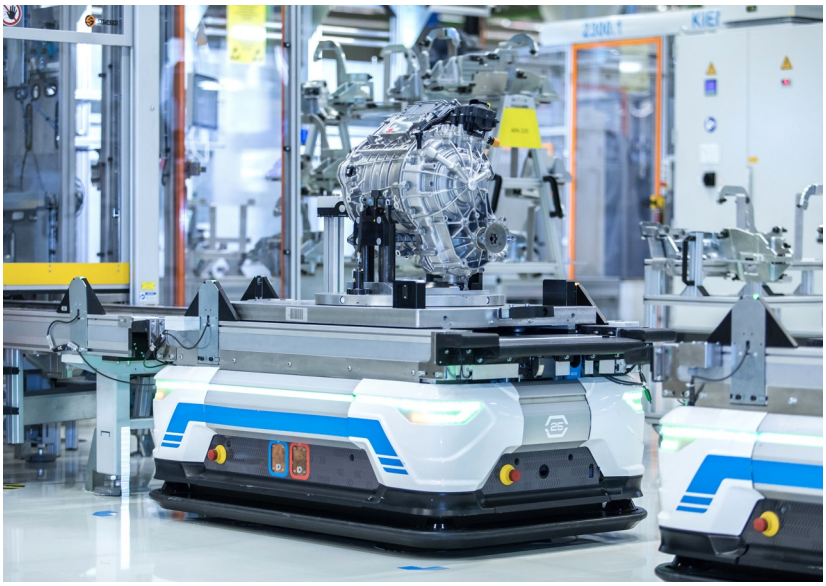


Figure 2.36: AGV help transporting e-motors. Source: Assembly 2022



Figure 2.37: AGVs are integrated into the assembling process. Source: Sergeev 2020



Figure 2.38: A fleet of small robots transport light-weighted carts. Source: Grenzebach 2019

2.4 Summary

The prefabricated housing industry is more akin to manufacturing than traditional construction and differs significantly between regions. The industry in the U.S. is interested in relieving the insufficiency of affordable housing, so factories in this country do not enthusiastic about automating the manufacturing process for budget concerns. Based on the analysis of the selected factories, we can see the manufacturing process is divided into separate stations, each of which produces different components; the connection between stations is weak.

Regarding automation in construction, the mobile robot is advantageous in integrating with the BIM, such that it is aware of the site's current state and can update the database when collecting real-time data by navigating on the site. Mobile robots in automobile factories help to connect different assembly stations by automatically transporting products between streamlines or supplying materials for stations; they increase the material flow efficiency and help inventory management in such factories.

These findings provide a starting point for drafting out a scenario of using mobile robots to support labor-intensive house manufacturing processes in prefabricated housing factories.

Chapter 3

Hypothesis

As mentioned in the previous chapter, the prefabricated housing industry in the U.S. is affordability-oriented, which makes factory operators cautious about budget control. Therefore, proposing an ambitious renovation plan to automate the factory or even part of the manufacturing streamline requires a significant investment and may lead to a temporary shutdown for factory upgrades. Such a scenario is theoretically possible but practically infeasible due to the unpredictable returns on the investment and delays in the delivery of ongoing orders. This thesis aims to find a way to introduce automation to the factory without disruptions to factories' current layout and manufacturing, with financially acceptable investment. Therefore, the thesis's objective is to propose an automation plan that supports the existing factory workflow and workers with minimum intervention to the current manufacturing process.

3.1 Project Positioning

It's not hard to conclude that prefabricated housing manufacturing is the off-site version of construction; almost all works rely on the labor force and heavy machinery, and there is a lot of space for automation. But the story is not as simple as it looks - arbitrarily proposing a project for automating several construction tasks might cause many problems, like the automation conflicts with current workflow making manufacturing coordination even harder, the automation might cause the fear of losing jobs among workers, or there is an additional cost for training workers collaborate with the robot, etc.

Therefore, here are several criteria help to select the task to be automated and project positioning:

1. The automated task is not position-binding or skill-binding, which means the automation does not interrupt the current workflow;
2. It is preferable that automation can relieve workers from heavy, repetitive, and tedious tasks, which do not generate positive financial / health value but waste a lot of time, therefore the project is comparatively worker-friendly;
3. The automation can better integrate into the current factory layout with minimum upgrades required.

Scrutinizing the manufacturing process with these criteria, the material handling task perfectly meets all these requirements. Transporting materials, including raw materials, intermediate products, supplies, and furniture, takes time and effort, and there's no need to assign a role solely managing moving materials around. Most of the time, the worker from the station walked to the storage, loaded the

needed materials onto the cart, and dragged the cart back to the station to continue the task. Automating the material handling can adapt to the current factory layout without a lot of renovation, relieve the worker from repetitive material transporting works, and not interfere with the current workflow.

In addition to humans, heavy machinery like bridge cranes is also managing to transport material between stations in the factory; such materials, like floor decks, are weighty and oversized and not suitable for ground transportation, so the large materials are not included in the scope to the automated material handling. The other extreme is the light-weighted gears - choosing which tool to use depends on the worker's experience and type of job; workers can make better decisions in selecting the handiest equipment, so the light-weighted gears are also not included in the scope of automation.

As a result, the automated material handling will manage medium-sized materials, like tool carts, types of furniture, furnishing supplies, and non-load-bearing building components (Figure 3.1). The crane still manages heavy and bulky items, and the humans decide which light-weighted tools to use and when and where to pick them up.

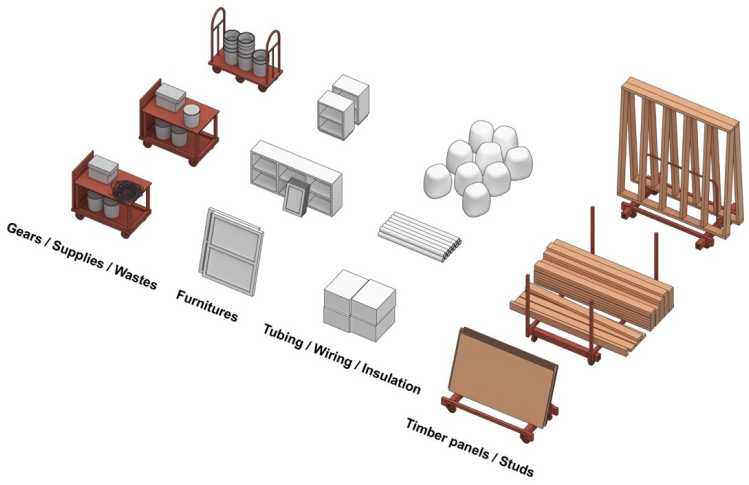


Figure 3.1: Target materials for the automated material handling. Source: diagram by the author

3.2 Proposed Scenario

Using mobile robots to handle materials in factories and warehouses has been proven to be the most efficient way of automated material handling (Oleari et al. 2014). Thus this thesis chooses the mobile robot to help manage and transport materials in the prefabricated building factory.

Different from highly-industrialized factories and warehouses, the space of the prefabricated housing factory selected in this thesis is far from generous; spaces between stations and modules are narrow, limiting the robot's operation footprint and turning radius.

Most commercial robots cannot fit in this limit as they were designed to adapt to various tasks that make their functions redundant and their sizes too large for the setting of this thesis. Therefore, creating a robot that has a compact size, good agility, and better integration into the carts becomes the key to the success of automating the material handling in the prefabricated housing factory. In the thesis, we're using a compact version of the wood factory as the test bed, which has a layout similar to the factory in New Hampshire, with all stations included (Figure 3.2).

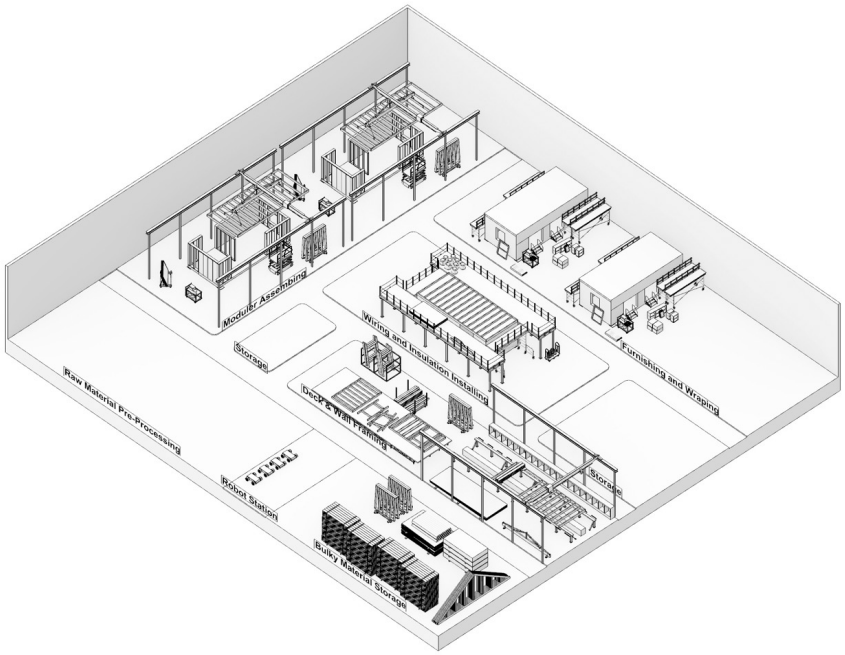


Figure 3.2: Project test bed - a compact version of the factory in NH. Source: diagram by the author

The proposed scenario is as follows:

When getting the command to transport a cart to a station, the robot first locates and moves to the cart. The robot's compact size allows it to drive below the cart and move with it to minimize its operational footprint (Figure 3.3). The robot then carries the cart toward the destination (Figure 3.4). With the help of mecanum wheels, the robot can move in any direction in the two-dimensional space (Gfrerrer 2008) maximizes maneuverability, and basically eliminates the turning radius. After arriving at the destination, the robot leaves the cart and navigates back to the robot station waiting for the next command (Figure 3.5).

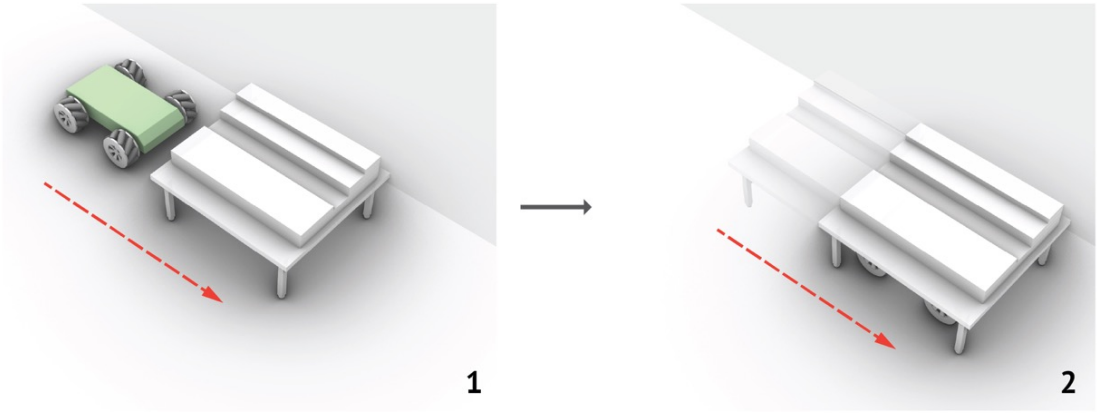


Figure 3.3: The robot picks up the cart. Source: diagrams by the author. Video: Appendix A.1

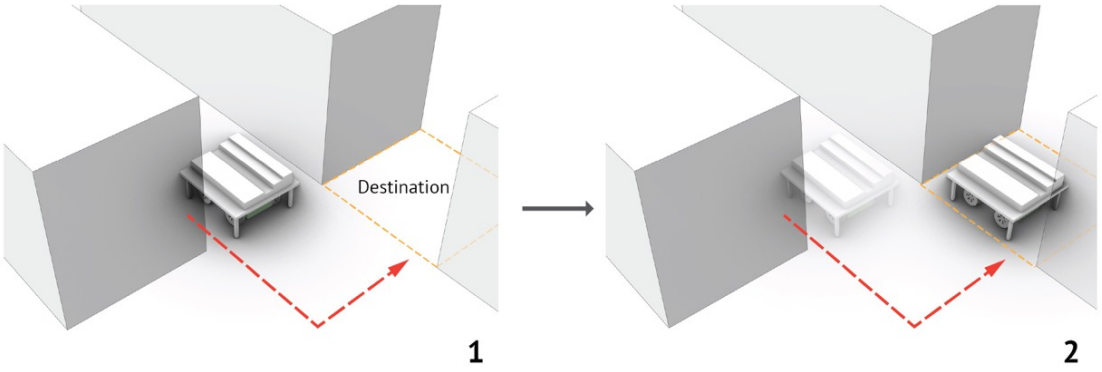


Figure 3.4: The robot transports the cart to the destination. Source: diagrams by the author. Video: Appendix A.1

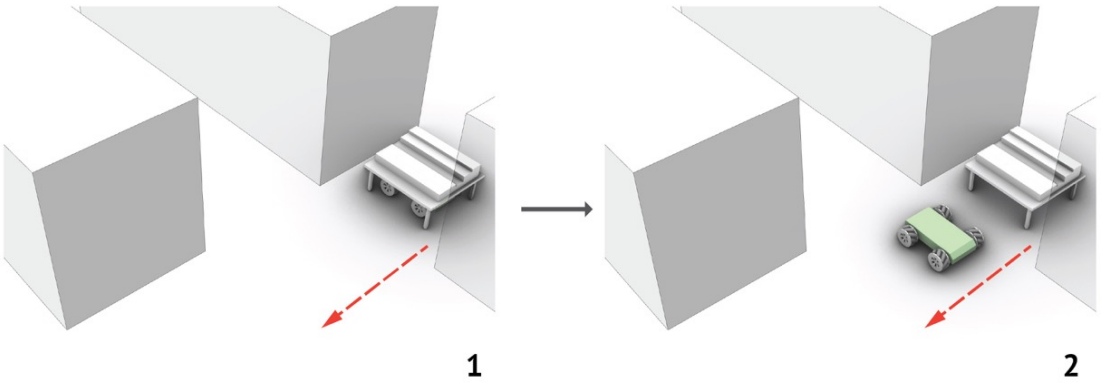


Figure 3.5: The robot leaves the cart. Source: diagrams by the author. Video: Appendix A.1

Chapter 4

Methods

The Methods chapter adheres to the Research by Design (RbD) framework, which regards design as a method of inquiry. The development of the project becomes a process of exploring potential solutions to the research question through prototype building, testing, solution scanning, and physical or virtual simulations (Roggema 2016). In other words, instead of developing a fully functional robot, the thesis aims to ask, inquire, explore, or confirm the hypothesis that introducing the mobile robot to handling materials helps support the current manufacturing process. The project's outcome, therefore, is a proof of concept.

The chapter followings a step-by-step problem-solving process (Figure 4.1):

1. This chapter starts by documenting the design and evaluation of a moving base, which is the most basic part of the robot. The robot design section includes proposing a system architecture specifying the components required and the relationships between parts in high-level, and the on-ground circuit design and hardware tests.
2. With the moving base ready, the next step is to find a way to connect the robot and cart. This section proposes several connection methods, verifies each of them by analyzing their physical property, and finds the optimal solution for the question.
3. The following question is, how does the functional robot know where the task's destination is? Therefore, the third section solves this problem by proposing a path-planning method that can help the robot know each stop of a material-delivering task and the shortest path for connecting all stops.
4. Although the robot knows where the destinations are and the shortest path to go, it still doesn't know how to follow those paths. The fourth section engaged in proposing a solution that allows the robot to follow the path with the line-following method and test the method with a smaller robot in a minimal environment setup.
5. One remaining question that needs to be addressed is, what if the robot encounters an obstacle when it is moving along paths? Due to the complication of this question, the final section only proposes a simple collision avoidance strategy without comprehensively considering all possibilities but just one scenario in the ideal setup. The end of the section also discusses the limitation of the current setup and future works for an effective collision avoidance strategy.

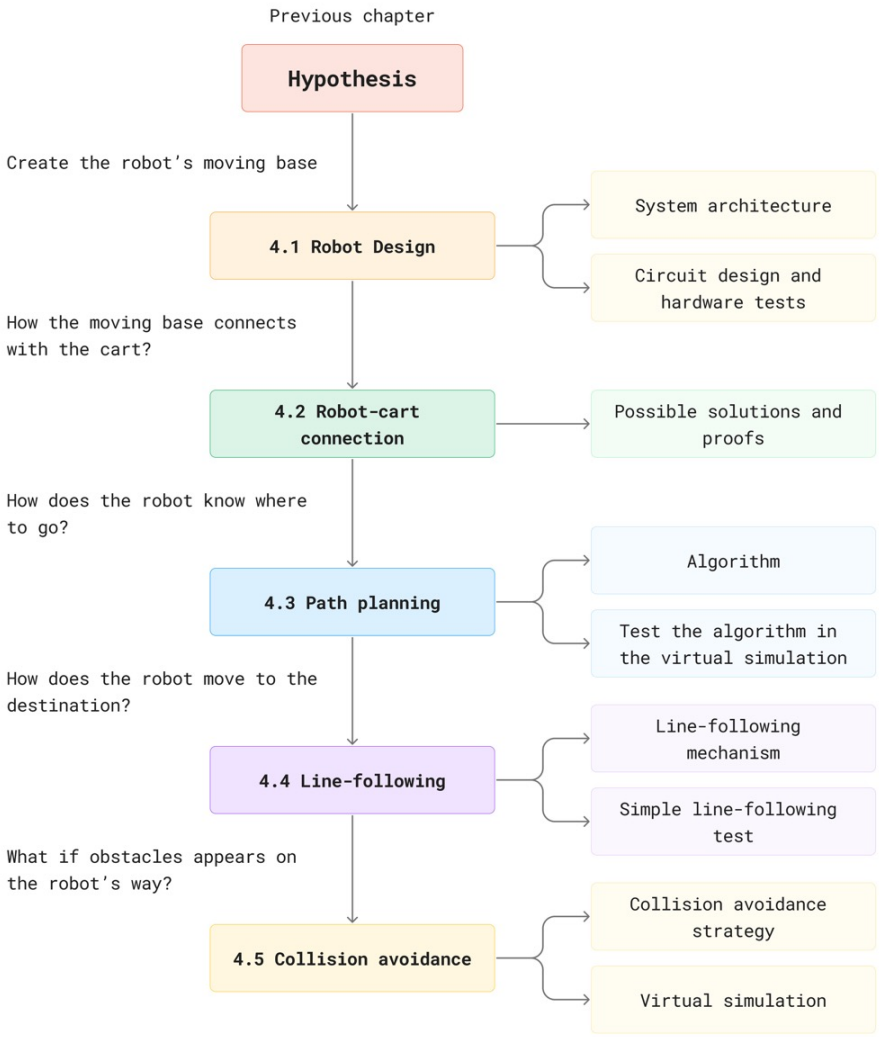


Figure 4.1: The structure of this chapter. Source: diagram by the author

4.1 Robot Design

Designing and developing a mobile robot can easily be long-lasting work because the robot is an integration of the communication layers and component-based frameworks (Joyeux and Albiez 2011). Every minor issue, like controller disorder or signal interference, can take months to address. An efficient way of developing a robot is using the bottom-up method - starting from the most fundamental and practical part, adding complexity layer by layer according to the project objectives, and keeping the system concise and clean to avoid redundancy.

4.1.1 System architecture

The robot system generally consists of two parts: hardware and software. The hardware includes robot chassis, types of sensors, and computers; the system consists of functional blocks, cooperations, and planning horizon. This thesis follows this guideline provided in (Everett 2017) when designing the robot system. The approach outlined in the guidelines is comprehensive, enumerating all the distance/vision sensors, robot systems, packages, and related algorithms that facilitate socially acceptable navigation for a mobile robot. However, this project does not aim to achieve navigation with such a high level of precision. Therefore, the system architecture was simplified and reorganized based on the hardware setup, the data flow, and the functionality of each component. This reorganization was designed to meet the minimum requirements for safe navigation.

In the system architecture diagram (Figure 4.2), the color blocks indicate the hardware groups, the red arrows mark the power, or electric, flow, in the system, the blue arrows present the data flow between hardware, and the gray box shows the specific knowledge needed to make the block work. The following part explains the system diagram by the hardware group to help clarify how each part works and how that part communicates with others.

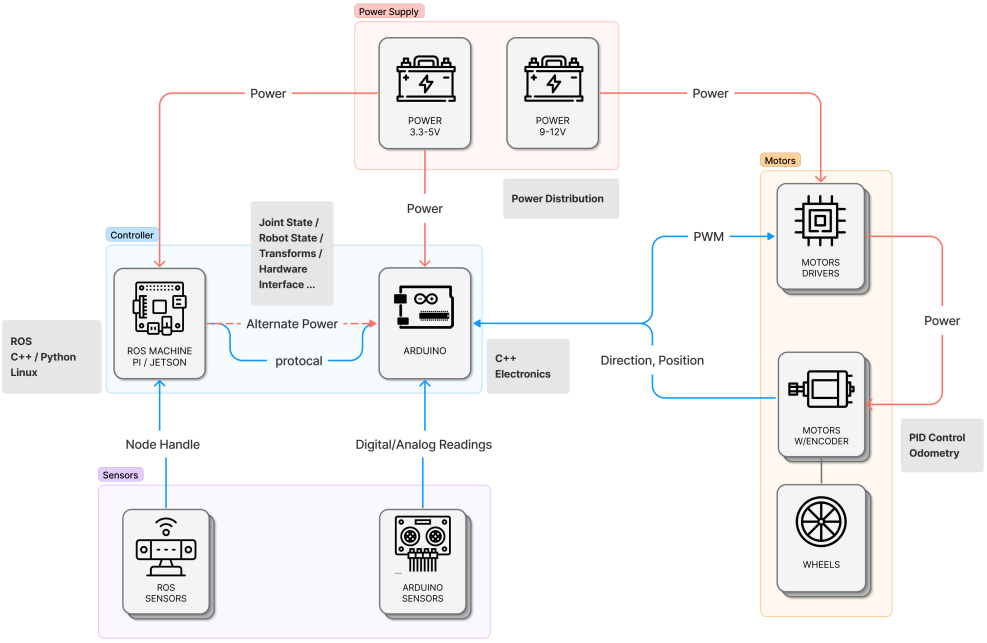


Figure 4.2: System architecture. Source: diagram by the author

Power supply

The power supply group consists of two independent batteries: one smaller battery with a voltage between 3.3 to 5 volts to provide power for the controllers, and another larger battery with a voltage between 9 to 12 volts, or an even higher voltage like 24 volts, to supply power to the motors. Using the two-set battery configuration is due to the different power requirements between the controllers and the motors; motors typically need a higher power to make individual motors respond to the speed signal precisely, but such power is too high for microcontrollers to burn them out. Another advantage

of giving the controllers an independent power source is that, in the development cycle, most of the work happens on programming the controllers where the motors can always stay disconnected to make the development process saver. Power distribution knowledge is needed to help connect wires to the electronic components correctly.

Controller

The controller is the most complicated part of the system: it is the robot's brain that handles hardware abstractions, manages the data flow, and coordinates hardware. Typically there are two controllers in this block: a higher-level controller with a powerful CPU that controls the system and solves navigation and sensing algorithms, and a lower-level controller with a minimum computational capability that receives higher-level messages, like speed and angle, converts them to lower-level signals, like voltage and rpm, and send them to electronic components. Two controllers communicate with a USB protocol.

Comprehensive software and programming knowledge are required to make the controllers work. A common higher-level controller can be Raspberry Pi or Nvidia JetsonNano, running a Linux system and using the Robot Operating System (ROS) to control the robot. Data is stored and transferred as "topics" and "states" that the system subscribes to and publishes.

Motor

The motor group consists of motors, motor drivers, and wheels. Motors themselves cannot directly receive signals from the controller; the data needed to be further decoded and translated to current and voltage to make the magnetics and coils work. Thus, the motor driver communicates with the controller and the motors. The motor driver receives rotation and PWM signals from the controller and then converts them to voltage and electrode to control the motor's spinning direction and speed. The motor should come with a pre-installed encode to read the pulses to record the number and direction of spins and send them back to the controller, where the signal will be further decoded to speed as feedback in the closed-loop control.

Sensor

The sensor group is a flexible part of the system; developers can install specific types of sensors for different tasks and remove the sensor when they no longer need to avoid hardware redundancy. For example, sensors with complicated readings, like camera or lidar, need more computation resources such that they should be connected to the higher-level controller; sensors with simple readings, like IR sensors that generate just binary data, can be directly installed on the lower-level controller to reduce the complexity on the controllers' side.

4.1.2 Circuit design

The circuit design can temporarily put the sensor part aside but focuses on the robot's moving base - with only power, controller, and motors - as sensors are flexible part of the systems; the circuit will be over complicated to debug if including sensors in the circuit design at an early stage.

The following table is the list of electronic pieces that the moving base needs:

Component List	
Item Name	Quantity
12V 70:1 DC Motor with Encoder	4
L298N Motor Driver	2
L298N Motor Driver	4
5V 2000mAh Lipo Battery	1
5V 2000mAh Lipo Battery	1
Power distribution board	1
Arduino Mega	1

Table 4.1: Component list for the moving base. Source: table by the author

A 4-wheel-drive system is a prerequisite for maximizing the agility and maneuverability of the mecanum wheel setup. Each wheel should be able to spin independently, and different combinations of rotations produce various actions like moving diagonally, shifting left and right, or doing centric-specific turnings. As every motor driver can manage two motors, thus there are two motor drivers needed to control all four motors - with each drive controlling two motors on the same side of the robot (Figure 4.3).

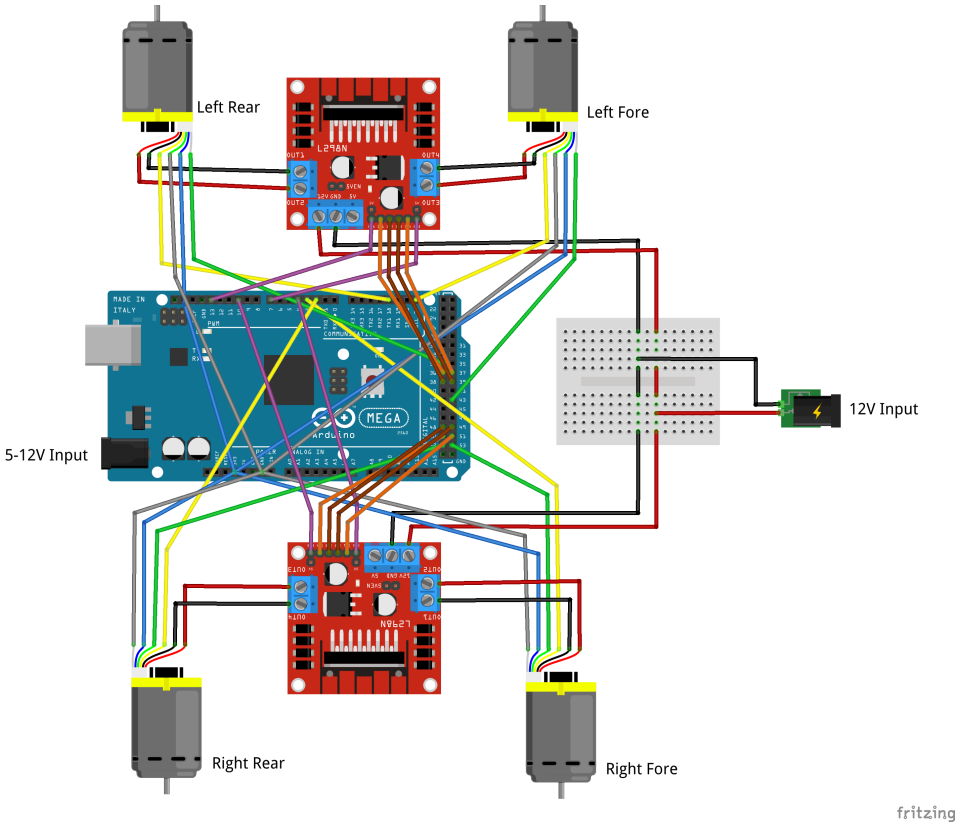


Figure 4.3: Circuit design. Source: diagram by the author

The reason for choosing Arduino Mega as the lower-level controller is that the "OUT +" pin needs to connect with the "interrupt" pin on the controller's board to help the controller get the reading of the motor's number of spins and spinning direction. The robot in this thesis uses the 4WD driving mode

such that the board should have at least 4 "interrupt" pins to communicate with all four motors. According to the board specs from Arduino (Figure 4.4), the most affordable board with enough "interrupt" pins is Arduino Mega.

BOARD	DIGITAL PINS USABLE FOR INTERRUPTS
Uno, Nano, Mini, other 328-based	2, 3
Uno WiFi Rev.2, Nano Every	all digital pins
Mega, Mega2560, MegaADK	2, 3, 18, 19, 20, 21 (**pins 20 & 21** are not available to use for interrupts while they are used for I2C communication)
Micro, Leonardo, other 32u4-based	0, 1, 2, 3, 7
Zero	all digital pins, except 4
MKR Family boards	0, 1, 4, 5, 6, 7, 8, 9, A1, A2
Nano 33 IoT	2, 3, 9, 10, 11, 13, A1, A5, A7
Nano 33 BLE, Nano 33 BLE Sense	all pins
Due	all digital pins
101	all digital pins (Only pins 2, 5, 7, 8, 10, 11, 12, 13 work with **CHANGE**)

Figure 4.4: Arduino board specs. Source: Arduino[1]

4.1.3 PID control

One feature of the robot that differs from the toy cars is that it keeps tracking the distance the robot moved and the speed command it sent to ensure correct odometry data storing and updating in the system. Many methods could help achieve precise control for the moving joints, and one of the most widely used ways in industries is PID control.

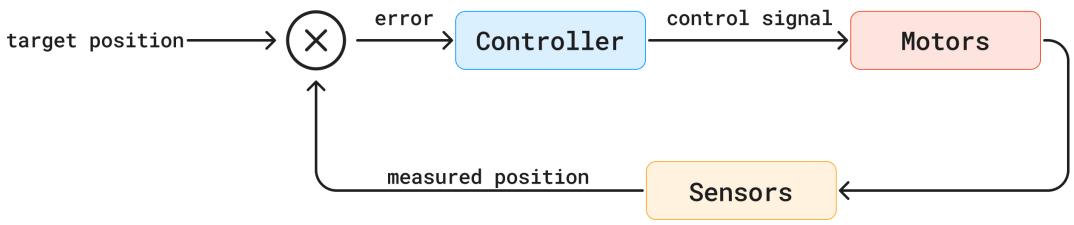


Figure 4.5: PID control loop. Source: diagram by the author

PID control is a three-term control where P for the proportional term, I for the integral term, and D for the derivative term; it is a feedback control loop (Figure 4.5) that calculates an error signal by comparing the desired reference value with the actual value the system gets to (Crowe et al. 2005). The general mathematical equation of PID control is:

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de}{dt} \tag{4.1}$$

- $u(t)$: the output of the PID controller, used as a signal for correcting the control error;

[1] Arduino website: https://www.arduino.cc/

- $e(t)$: the error value at the time t;
- K_p: the proportional gain that manages the signal strength of the error;
- K_i: the integral gain, which decides the accumulated error over time;
- K_d: - K_d: the derivative gain decides the effect of the change rate of the error on the system.

In practice, for convenience of computing, the integral and the derivative can be approximated as:

$$\int e(t)dt \approx e_{int} + e\Delta t \; ; \quad \frac{de}{dt} \approx \frac{e - e_{prev}}{\Delta t} \tag{4.2}$$

Therefore, the pseudo-code of PID control in the robot is as follows:

Algorithm 1 PID control loop

Require: $V_{target}, K_p, K_i, K_d \in \mathbb{R}$
$T_{prev} \leftarrow 0$
$e_{prev} \leftarrow 0$
$e_{int} \leftarrow 0$
repeat
  $T_{curr} \leftarrow time$ ▷ get current system time
  $\Delta_t \leftarrow T_{curr} - T_{prev}$
  $T_{prev} \leftarrow T_{curr}$ ▷ update previous time
  $V_{actual} \leftarrow actual\_value$ ▷ read actual value from sensors
  $e \leftarrow V_{target} - V_{actual}$
  $\frac{de}{dt} \leftarrow (e - e_{prev})/\Delta_t$
  $e_{int} \leftarrow e_{int} + e \times \Delta_t$
  $u = K_p \times e + K_i \times e_{int} + K_d \times \frac{de}{dt}$
  $Controller \leftarrow u$ ▷ send the signal to the controller
until program terminated

Finding a proper value combination for PID control is an iterative process that uses trial and error to find the best values that fit the system; the values and value combination could differ significantly with tasks and hardware. The next section will discuss the process of finding the optimal PID value combination for the robot's motor control.

4.1.4 Hardware test

Since the complexity of the robot system proposed at the beginning of this chapter and the time needed to invest in the hardware debugging and testing, the work of physical part of the project could not aim to produce a fully-functional mobile robot for material handling in the prefabricated building factory. However, scoping down the hardware work helps verify the project's key features. It also enables enough attention to exploring and discussing higher-level aspects, like control strategy, virtual simulation and operation, and the prospects of this project.

Therefore, the hardware part of this thesis mainly focuses on creating a moving base that agile enough to navigate in the prefabricated house factory.

The hardware test consists of three parts: the single motor test, the PID control test, and the 4WD setup test.

Single-motor setup test

This test helps verify the selection of the electronics and the wiring proposed in the circuit design and is the basis for all future tests. The first step is to choose a single set of driving units and wire them up according to the circuit diagram in 4.1.1 (Figure 4.6).

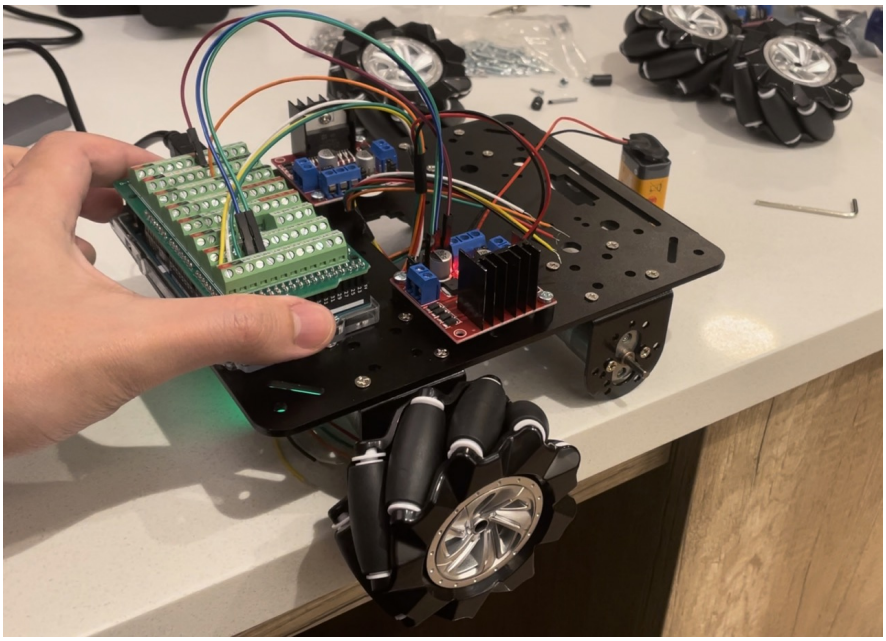
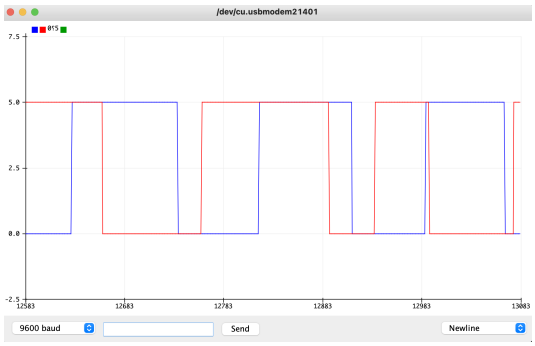


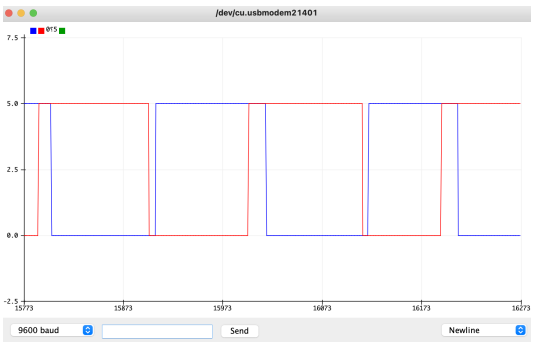
Figure 4.6: Testing circuit design on a single motor. Source: photo by the author

However, the success of this test does not merely rely on whether the motor can work but also depends on whether the motor can communicate with the controller and send/read signals. So the test plots out the readings from the interrupt pins to show if the motor feedbacks the direction and rotation counts information.

In the direction feedback plots (Figure 4.7), the blue signals appear ahead of the red ones in clockwise rotation, while in the counter-clockwise, the red signals appear ahead of the blue ones.



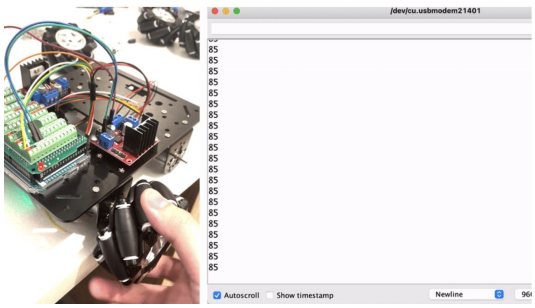
(a) clockwise rotation



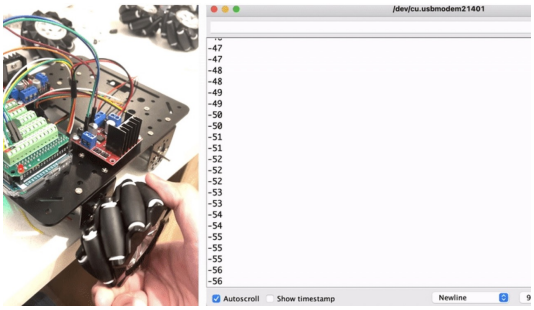
(b) counter-clockwise rotation

Figure 4.7: Direction feedback. Source: diagrams by the author

In the speed feedback plots (Figure 4.8), the encoder returns incremental positive readings when rotating the motor clockwise, or forward, and returns decremental negative readings when rotating the motor counter-clockwise, or backward.



(a) rotate the wheel clockwise



(b) rotate the wheel counter-clockwise

Figure 4.8: Speed feedback. Source: diagrams by the author. Video: Appendix A.2

PID control algorithm test

In the previous test, the motor responded to the signal from the controller and returned the pseudo distance, which is the number of coil rotations. The PID control uses the pseudo distance information to compare the error between the target and actual distance to help find an effective combination of PID values.

The test starts with a default setting with $P = 1.0, I = 0.0, D = 0.0$; this setting helps determine the actual error the motor produced (Figure 4.9). In the plot, the blue line shows the target distance, and the red curve represents the actual distance the motor traveled.

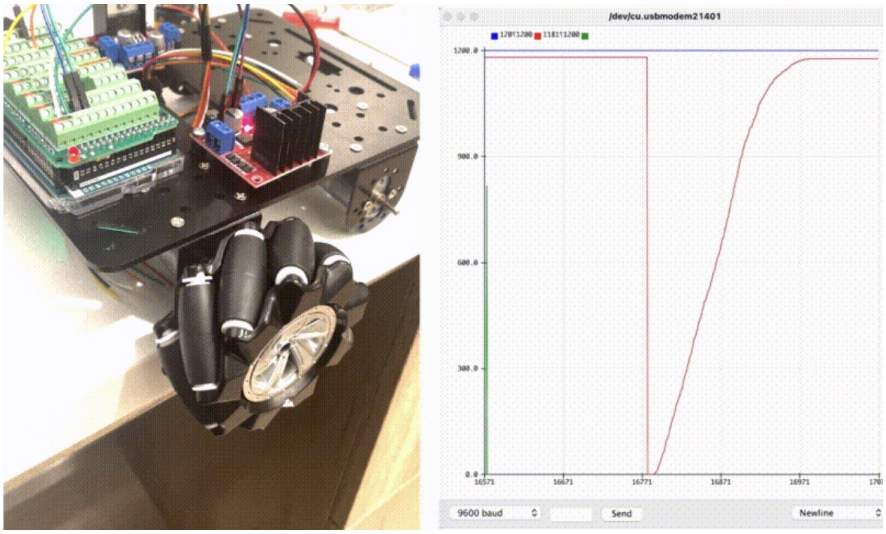


Figure 4.9: PID test with $P = 1.0, I = 0.0, D = 0.0$. Source: diagram by the author

From the plot, we can figure out a gap between the actual distance and the target in the default setting. Then the first thing to try is to increase the error signal to ask the motor to have slightly more rotations to fix the gap; thus, the P value be increased to 5.0, with the other two values remaining unchanged.

The result of a higher error value (Figure 4.10) shows that the amplified error signal does close the gap between the target and the motor's actual travel distance but with a bit of overshoot. However, such a slight overshot could be smoothed out with the integral value, so the I value was increased to 0.025 for the next round of tests. The result shows that the motor's actual travel distance perfectly meets the target we provided (Figure 4.11).

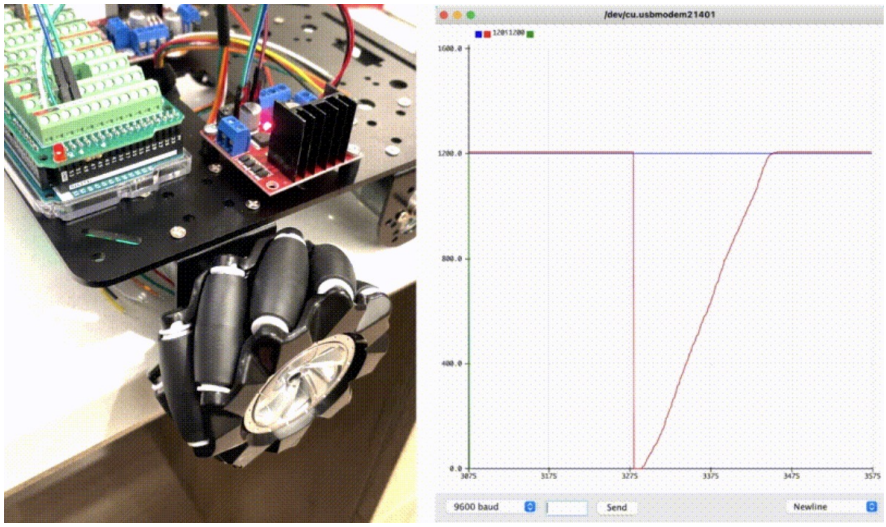


Figure 4.10: PID test with $P = 5.0, I = 0.0, D = 0.0$. Source: diagram by the author

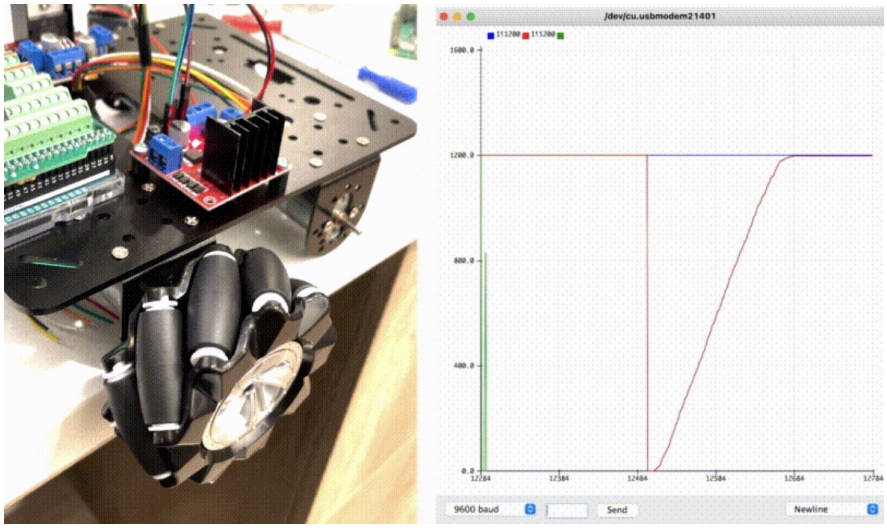


Figure 4.11: PID test with $P = 5.0, I = 0.025, D = 0.0$. Source: diagram by the author

To further verify the PID value combinations, the controller sends a sinusoid signal to the motor and checks whether the motor can follow the signal pattern under the current setting. For example, in Figure 4.12, the red sinusoid curve is the periodical target the controller sent to the motor, and the yellow curve is the motor's actual behavior; there's a slight delay for the motor in responding to the target signal because of the hardware latency, but generally following the pattern well. Thus, the optimal PID value combination is $(P = 5.0, I = 0.025, D = 0.0)$ under the current setup.

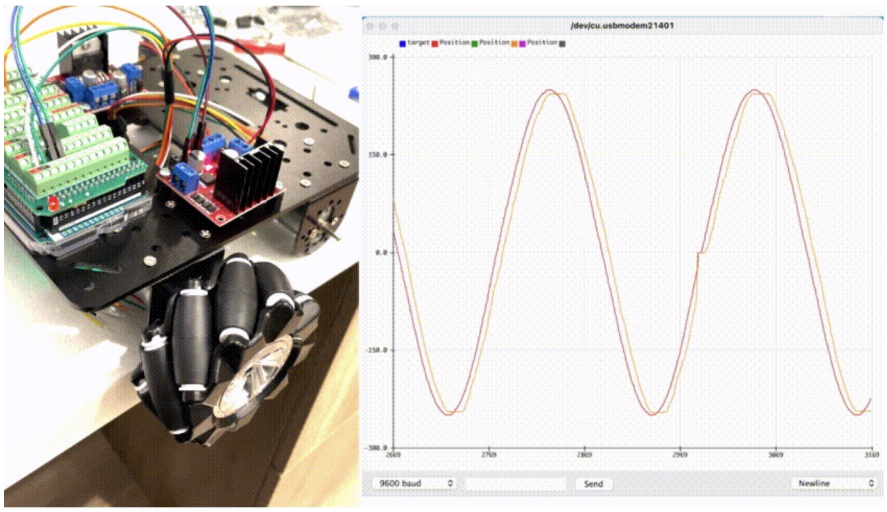


Figure 4.12: The motor follows the sinusoid signal from the controller. Source: diagram by the author

4WD setup test

With the previous two steps passed flawlessly on a single wheel, we can now move on to preliminary assemble the robot and evaluate whether the circuit design and PID control work with a 4WD setup (Figure 4.13). This helps better check how the motors coordinate with each other and whether they can adequately respond to the target signals correctly. Furthermore, the evaluation process will also help to identify any potential issues or areas for further development to continue refining and improving the design.

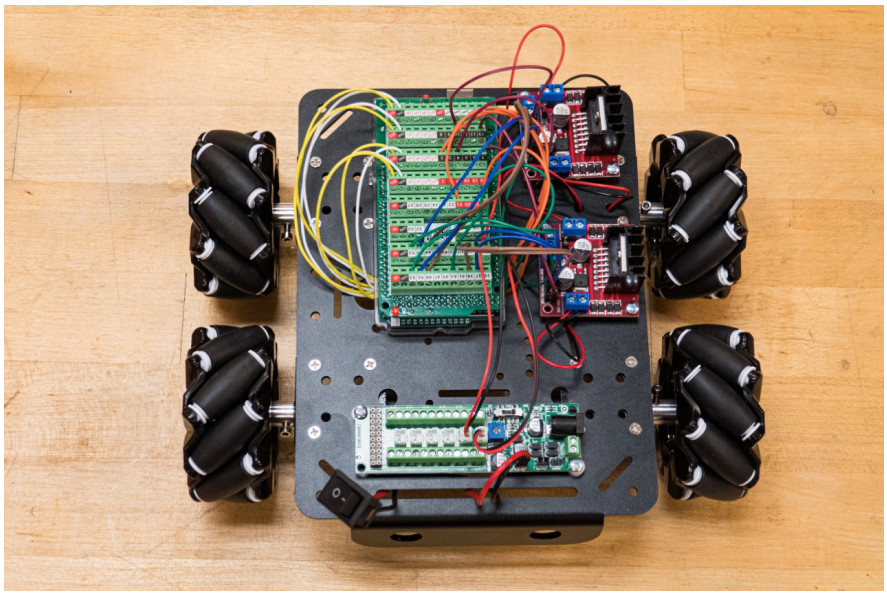


Figure 4.13: 4WD setup. Source: photo by the author

The test is overall in order - power distributed from the source to both motor drivers thus goes to all four motors without difficulty, the signals from the controllers actuate the motors, and the

motors respond to the signal with correct motions (Figure 4.14). However, four-motor coordination is slightly away from expected. On several occasions, the robot slightly deviates from the straight line, sometimes going right but sometimes left. The problem is more apparent when asking the robot to travel a longer distance, but the details and the solution to this problem will be further elaborated in the next chapter.

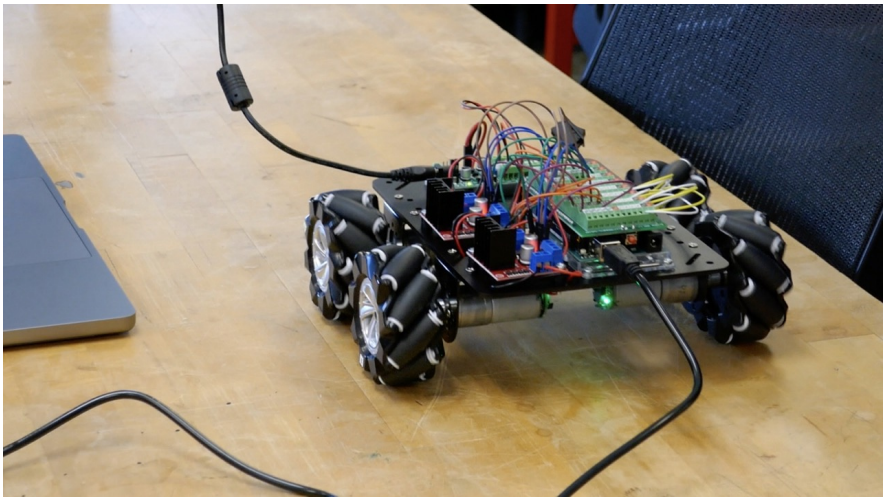


Figure 4.14: Preliminary test on 4WD setup. Source: photo by the author. Video: Appendix A.2

4.2 Robot-cart Connection

According to the proposed scenario, the robot moves beneath the cart, connects with it, and delivers it to the destination. The previous section provides a viable moving base to navigate the site; this section will discuss the method to connect the cart and the robot. Of course, this connection problem is not as simple as "connecting two objects." Still, it should consider more practical factors, like the precision the connection requires, how to ensure the connection's strength when transporting the items, how the robot can generate enough friction to move the whole system, and by what method the link makes the delivering task more energy-efficient.

The section starts with setting up a hypothesis, which is a proposed target for the material handling task, then developing a solution by discussing and analyzing several potential methods practically and physically. At the end of this section, we will discuss how the solution also adapts to the multi-robot collaboration scenario.

4.2.1 Hypothesis setup

The project positioning in Chapter 3 defines the target of the material handling task as medium-sized materials like tool carts, furniture, or non-load-bearing building components. The objective of this section is to propose a suitable object for the robot to move while ensuring that the object is the heaviest and the bulkiest thing in the category; the reason for choosing such an object is that if the robot can handle the most demanding materials in the category, it then has no difficulty dealing with all other items.

The object that perfectly meets the criteria is a cart with a pile of wood studs: the wall framing station consumes the wood studs at a fast rate; thus, workers in that station should frequently replenish the material. To avoid always leaving their work to get more materials, workers attempt to load as many

studs as possible each time; this makes the cart carrying studs the heaviest in the category. Hence we set up the hypothesis of asking the robot to move a cart containing a pile of oak wood.

Before diving into more details, defining the measurement system used in this thesis is important. Because many measurements and material property defined in the International System of Units is based on the metric system, physical calculations in this thesis are also based on the metric system to avoid confusion.

Going back to the hypothesis, the commonly seen oak wood stud in the bulky storage space (Figure 4.15) has a size of around $0.15 \times 0.06 \times 2.2 = 0.0198$ m^3 for each item, and the density of oak wood is between 600 and 900 kg/m^3 [2]; we arbitrarily pick the in-between 700 kg/m^3 for simplicity. Generally, a single cart can load $7 \times 8 = 56$ units of oak wood studs (Figure 4.16), then the total weight of the object is:

$$\text{Volume} \times \text{Density} \times \text{Count} = 0.0198 \times 700 \times 56 = 776.16 \text{ kg} \tag{4.3}$$

Compared to the target object's weight, the robot's weight, which is around 50 kg in hypothesis, is less than one-tenth of the object's load and thus can be neglected in the following calculations.

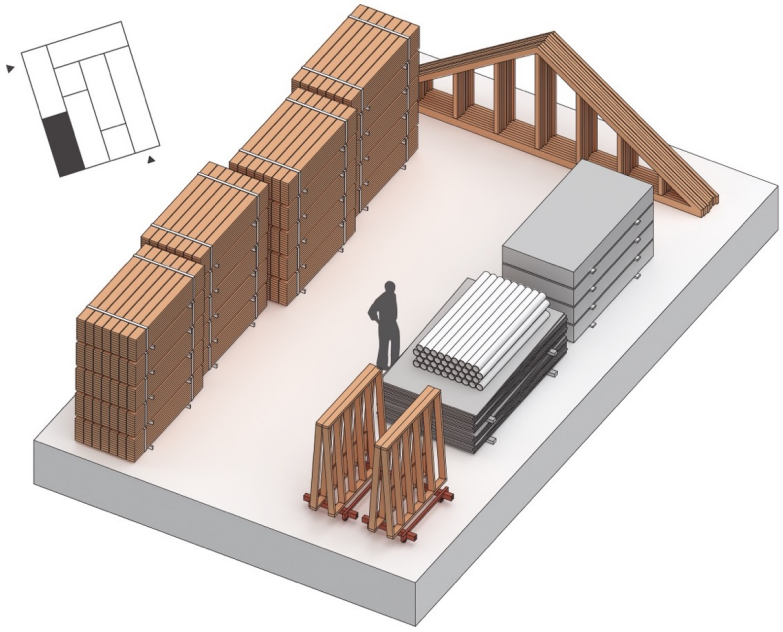


Figure 4.15: Diagram of the bulk storage space in the factory. Source: diagram by the author

To move this cart on the ground, the friction coefficient needed to be considered. Rubber pads are currently attached to the cart's feet (the wheels are also made of rubbers), so the rubber contacts the concrete ground; the coefficient between the rubber and the concrete is 0.6[3]. With this additional information, we can get the static friction between the cart and the ground as:

[2] Details about the density of oak: https://kg-m3.com/material/oak

[3] Friction coefficients between different materials and concrete: https://mae.ufl.edu/designlab/ClassProjects/BackgroundInformation/Frictioncoefficients.htm

$$\begin{aligned} \text{Static Friction}(F_s) &= \text{Coefficient of Friction } (\mu_s) \times \text{Normal Force } (N) \\ &= 0.6 \times 776.16 = 465.696 \text{ kg} \end{aligned} \tag{4.4}$$

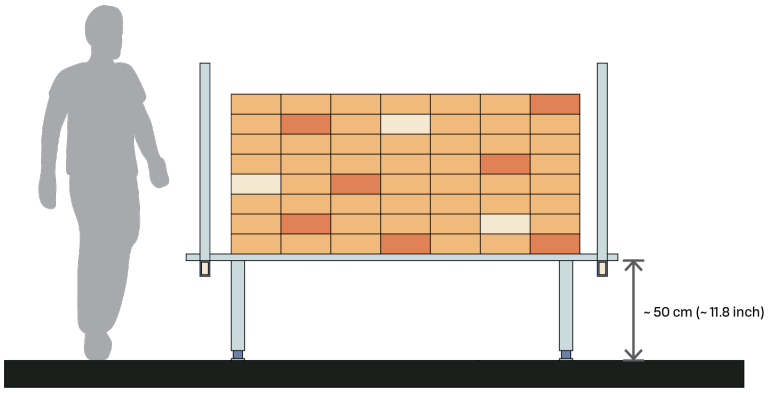


Figure 4.16: Hypothetic target object. Source: diagram by the author

Therefore, the robot's hypothetic task is to transport a cart that weighs 776 kg, with the minimum force of 465.696 kg required to make the cart move.

4.2.2 Connection design

As the proposed scenario in 3.1.4, the robot drives beneath the cart and connects with it, so the connection interface of this design is between the bottom of the cart and the top of the robot. A recent connection method on such an interface lets the robot lift the cart, with a lifter installed on the robot, like the Kiva Robot from Amazon Robotics (Costa Barros and Nascimento 2021). Another method, a more traditional way, is using hooks or hinges to connect both objects; this method requires some mobility on the cart, such that the cart should have wheels installed on its legs. This section analyzes the pros and cons of these two methods through physical property calculations and aims to find a connection design that works in the proposed scenario.

Unifying mathematical notations before starting the calculation help make the calculation less confusing. We define F_1 as the load on each of the robot's wheels, F_2 as the force on the robot-cart connections, F_3 as the static friction of the whole system, which is 465.696 kg, and F_4 as the load on the cart's legs. The calculations in this section don't aim to yield precise results but give an overall sense of the benefits and limits of different methods.

Method 1: robot lifts the cart

The first method is asking the robot to lift the cart entirely. This method needs to install lifters on the robot and does not require the cart's mobility (Figure 4.17).

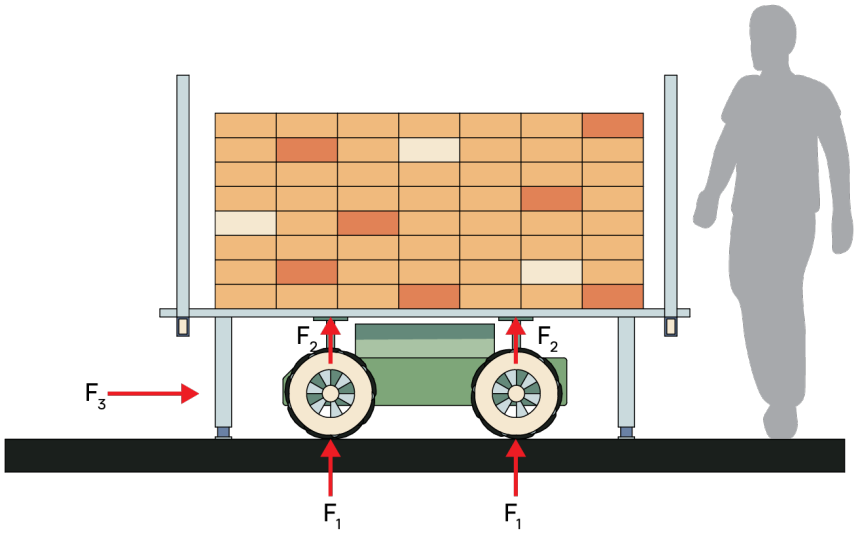


Figure 4.17: Method 1 - robot lifts the cart. Source: diagram by the author

In this method, the lifters bear all the loads and transfer them to the wheels, so we have the following:

$$F_1 = F_2 \approx \frac{\text{total weight}}{\text{number of wheels}} = \frac{776.16}{4} \text{ kg} = 194.04 \text{ kg} \tag{4.5}$$

That means each wheel should be capable of bearing 194.4 kg of load in this method; it's a heavy burden for the commercial robot's wheels. However, the real challenge happens on the shaft, where the wheel and the motor connect. As the shaft is thin and mainly functions as a connector, it doesn't have a solid capability to survive heavy loads - it has a high chance of getting damaged in this method.

On the other hand, the advantage of this method is also apparent. First of all, it does not need precise control over the robot's position when it is beneath the cart. This helps save time and computation in practice. Second, the load on the wheels, assuming that shafts are strong enough to bear such a massive load, guarantees that the wheels can generate enough friction to overcome F_3 to make the system move.

Method 2: connect with hinges

Compared to the previous method, this one goes to another extreme: the cart and the robot connect through hinges such that the robot bears no loads at all, and it asks the cart to have similar mobility as the robot has (Figure 4.18).

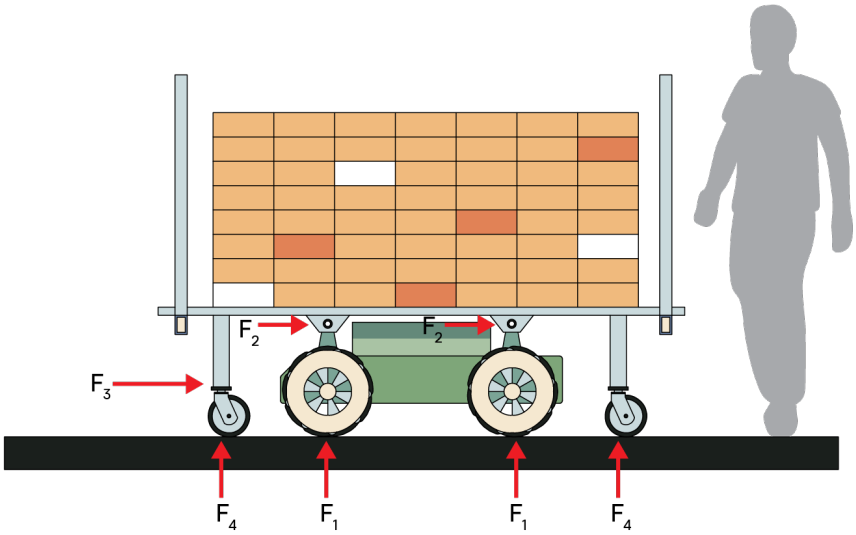


Figure 4.18: Method 2 - connect with hinges. Source: diagram by the author

In this method, fixed poses are installed on the robot instead of lifters, and hooks also need to be installed at the bottom of the cart. Then, when the robot drives beneath the cart, align its pose with the cart's hooks. The connection can be made automatically, which requires a higher level of control, or made with the help of humans. In terms of force analysis, assume we use four hinges and the cart has successfully been moved at a very low speed (almost 0 but actually moving), then have the following:

$$F_1 \approx 0, \quad F_2 = \frac{F_3}{4} = \frac{465.696}{4}\text{kg} = 116.424 \text{ kg} \tag{4.6}$$

$$F_4 = \frac{\text{total weights}}{\text{number of legs}} = \frac{776.16}{4} \text{ kg} = 194.04 \text{ kg} \tag{4.7}$$

The method seems fine from the surface as it is not hard to manufacture hinges that can bear 116.424 kg shear force and make the cart's leg bear 194.04 kg loads each. However, there's a hidden problem: the robot cannot actually move because of insufficient pressure toward the ground.

To calculate the maximum friction generated on the robot's wheels, we temporarily take the robot's weight back into consideration; assuming the robot's weight is 50kg, and there's no load on the robot in this method, then have the following:

$$F_1 = \text{Coefficient of Friction} \times \text{Robot Weight } = 0.6 \times 50 \text{ kg} = 30 \text{ kg} \tag{4.8}$$

As F_1 is only 6.4% of F_3, which is the minimum force needed to make the system move, thus the wheels might spin in place or burn out the motor without moving the cart.

Method 3: using friction pads - a combination of the previous methods

The third method combines the previous two - it asks the robot to partially lift the cart to generate enough friction without placing too much burden on the motor shafts and does not require precise control to connect the robot and the cart (Figure 4.19).

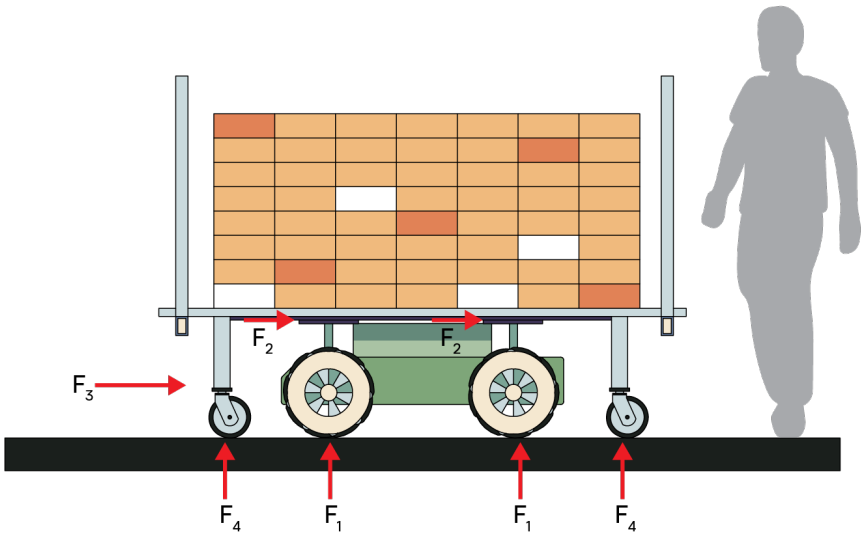


Figure 4.19: Method 3 - using friction pads. Source: diagram by the author

The keystone of this method is finding the minimum pressure the robot needs to generate enough friction to the ground, and that's positively correlated to the degree the lifters lift the cart. When decomposing the force, the F_2 is now the friction between the cart and the lifter, and the minimum sum of F_2 should at least equal to or slightly larger than F_3. Here comes another trick: if the material's coefficient of friction between the top of the lifters and the bottom of the cart is greater than 1, then the robot can use less force to move the system based on the equation:

$$\text{Normal Force } (N) = \frac{F_3}{\text{Coefficient of Friction}(\mu_s)}, \text{ where } F_2 = F_3,\ N = F_1 \tag{4.9}$$

According to [4], a viable interface with a coefficient of friction higher than 1 is a dry contact between aluminum and aluminum, with a value of 1.05. With the friction pads made of aluminum attached to the top of the lifters and the bottom of the cart, we have the force calculations as follows:

$$F_1 = \frac{N}{\text{num. wheels}} = \frac{F_3}{4\mu_s} = \frac{456.696}{4 \times 1.05}\text{kg} \approx 108.737 \text{ kg} \tag{4.10}$$

$$\begin{aligned} F_4 &= \frac{\text{total weight} - \text{loads carried by robot}}{\text{number of legs}} \\ &= \frac{776.16 - F_1 \times 4}{4} \\ &= \frac{776.16}{4} - F_1 = 194.04 - 108.737 \text{ kg} \\ &= 85.303 \text{ kg} \end{aligned} \tag{4.11}$$

Therefore, in this method, each motor shaft bears 108.737 kilograms of load, about 56% of that in method 1, which is a much more acceptable shear force for the motor's shafts. Each leg of the cart

[4] A list of different materials' coefficient of friction: https://engineeringlibrary.org/reference/coefficient-of-friction

carries 85.303 kilograms of load when working with the robot but bears total weight when not. This solution is the most viable of all three choices, so the method can be considered a possible solution to the robot-cart connection problem in this thesis.

4.2.3 Potential for multi-robot collaboration

The solution to the robot-cart connection in the previous section brings up some opportunities for multi-robot collaboration when moving heavy items.

One of the advantages of the solution is that the robot-cart connection does not need very precise control - the robot drives behind approximately the weight center of the cart to move the cart around. So, what if the cart becomes larger and contains more materials? Assume that the bottom of the cart is covered with the friction pad made of dry aluminum, then the robot can actually connect with the cart at any position with the friction pad, and that means more than one robot can be deployed to move a cart if it is large enough; in this situation, transporting more materials becomes possible because each robot can bear a portion of the weight of the materials (Figure 4.20).

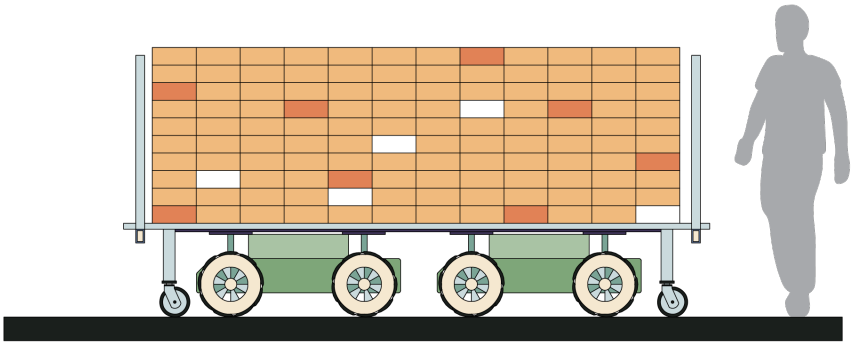


Figure 4.20: Proposed multi-robot collaboration in transporting heavy material. Source: diagram by the author

Instead of asking the robot to drive behind the weight center of the cart, a multi-robot scheme requires the average of the x and y coordinates of the robots' centers to approximate the x and y coordinates of the weight center; this helps to ensure the stability in transportation and avoids material from falling out of the cart. Another change that needs to be considered in the multi-robot scenario is the movement won't be as agile as in the single-robot system because robots move on their own axes. Therefore, better movement control in the multi-robot collaboration scenario must coordinate the robots' individual axes and the system's moving axis.

Nonetheless, achieving such a collaboration must solve way more problems than those mentioned in this chapter, and the detail of multi-robot collation is out of this thesis's scope. Hence, this section only aims to discuss rather than propose a comprehensive solution.

4.3 Path Planning

After the robot design and robot-cart connection design are done, the next task is to help the robot find a way to target materials, destinations, and a way back to the station with the least effort or the shortest distance. A proper path-planning algorithm helps to achieve the task, and there are some industry-proved algorithms for selection, like A*, Dijkstra, Rapidly-exploring Random Tree (RRT),

etc. This section aims to find a proper path-planning algorithm for the material handling task and test it in a virtual factory environment.

4.3.1 Path planning algorithm

The criteria for choosing a proper algorithm depends on the task, the environment of the factory, and the robot's hardware setup. In the material handling task, the robot finds the target, transports it to the destination, and navigates back to the robot station. With the non-complex manufacturing process, relatively static environment, and clear layout, the algorithm does not need to be complex to consider the possibilities of unexpected changes in surroundings or the environmental dynamics in finding the best path - the shortest distance and least effort are the only two factors needed to be considered.

The simplicity of the task makes the A* and Dijkstra algorithms proper candidates for both the mission and the robot. Furthermore, both algorithms help find the shortest path from one point to another without considering the environment's dynamics, thus not requiring specific hardware or sensors installed on the robot.

The two algorithms are similar, except that A* aims to find the most promising state to save computation time potentially (Karur et al. 2021). Therefore, the A* algorithm has been selected as the path-planning algorithm in this project.

A* path-planning

The A* algorithm is a map-based or graph-based search algorithm. By giving it an initial position and a target, it calculates the cost of possible actions at each state and detects the next steps until it finds reaches the target (Figure 4.21).

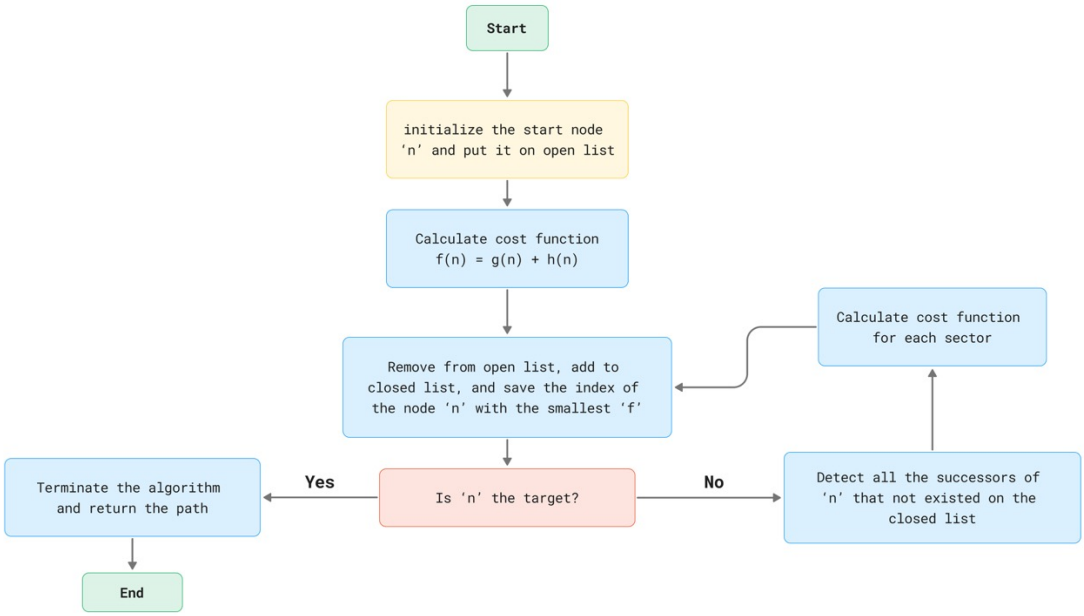


Figure 4.21: A* algorithm flow chart. Source: diagram by the author

An A* path-planning algorithm in this task includes the following components:

- Node - the potential positions with unique coordinates
- Action - the possible moves brings the robot to the next position
- Starting Node - the start point of each path-planning task
- Goal Node - the target position of each path-planning task
- Search Space - A collection of nodes, depending on how to represent the factory space as nodes
- Cost - A real number represents the effort of moving from one node to the other; in this task, distance is the cost
- g(n) - the exact cost from starting node to any node n
- h(n) - estimated cost from node n to the goal
- f(n) - the lowest cost in the neighboring node n

Algorithm 2 A* Path-planning

Require: Starting node s
OPEN $\leftarrow [s]$
CLOSED $\leftarrow []$
$g(s) \leftarrow 0$
$f(s) \leftarrow h(s)$
while OPEN **do**
    $n \leftarrow$ OPEN.$pop()$ ▷ n is a state with minimum cost
    CLOSED.$push(n)$
    if $n \in$ Goal **then**
        return $f(n)$ ▷ terminate the program when finding the shortest path
    end if
    for $m \leftarrow$ successor of n **do**
        if $m \notin$ [OPEN $\cup$ CLOSE] **then**
            OPEN.$push(m)$
            $f(m) \leftarrow g(m) + h(n)$
        else
            $f(m) \leftarrow g(m) + h(m)$
        end if
    end for
end while

Graph representation

One of the components still missing to complete the A* algorithm is the search space, which is the representation of the factory space in the search algorithm. Two common ways of converting the physical space to the search space are creating a 2D map as a raster image or storing paths and checkpoints in the graph data structure.

Building a map needs thoroughly scanning the site to get a raster map with the value of each pixel in the range of 0 to 1. The value is proportional to the reachability of the corresponding place - the lower the value, the lower the possibility of getting to the place. The advantage of this method is it records the factory layout in detail, thus making a better prediction of the shortest path. This method is widely used in path-planning tasks for mobile robots. However, the method might not fit well into the prefabricated housing factory, because the place of major obstacles - the modules - move between stations from time to time, and there's no fixed position for modules. In busy times, there

are more modules and thus more obstacles in the factory, while in non-busy seasons, there are fewer obstacles, and the space in the factory is generous. Keeping a precise record of the factory layout requires frequent scanning, say, every day.

Building a graph is less autonomous work - waypoints and essential checkpoints should define and store in the system by humans, and the format of the points is normally local coordinates. With the points ready and running the graph-building algorithm, the system generates a graph recording each point's coordinates and neighborhoods. To avoid frequently changing the waypoints and checkpoints, the factory can set up routes and loading stations that don't affect by the factory's daily operation for robots. The advantage of this method is it plans paths for the robot; thus, the robot does not need to care too much about the surrounding when executing the material handling task, and it is easier for the management work as the robot only operates within the given routes. Therefore, this project uses the graph to represent the search space for running the path-planning algorithm.

Algorithm 3 Graph-building

Require: Pre-defined coordinates POINTS, search radius R

```
Graph ← dict
for i ← 1, Points.length do
    for j ← i, Points.length do
        distance ← L2(POINTS[i], POINTS[j])        ▷ get L2 distance between two points
        if distance < R then
            Graph[POINTS[i]].add(POINTS[j])
            Graph[POINTS[j]].add(POINTS[i])
        end if
    end for
end for
 return Graph
```

4.3.2 Graph-based path-planning in the factory

With the algorithm and the factory representation ready, we can test the path-finding implementation in the simulated environment. The simulation is based on Rhino 7 and Grasshopper, which provides a good interface for modeling, programming, sequential control, and visualization. The test will also follow the Minimum Viable Product (MVP) criteria that aim to pass the preliminary test rather than include details and dynamics of the environment exhaustively.

The simplified factory representation uses the current path and arbitrarily assigns a loading stop to each station to fulfill the minimum requirements for the A* path-finding algorithm. The distance between checkpoints on paths is between 5 - 7 meters after several trails; a shorter distance is not necessary because the factory's traffic is not too complicated to increase the resolution of the graph, while a larger distance is also not preferred because it might fail to provide sufficient information of corners, turnings, and loading stops. Running the graph-building algorithm when the checkpoints are ready (Figure 4.22) helps build the graph and visualize the connectivity between checkpoints for verifying and debugging purposes (Figure 4.23).



Figure 4.22: Checkpoints in the factory. Blue dots are waypoints, and yellow pins are loading stations. Source: diagram by the author

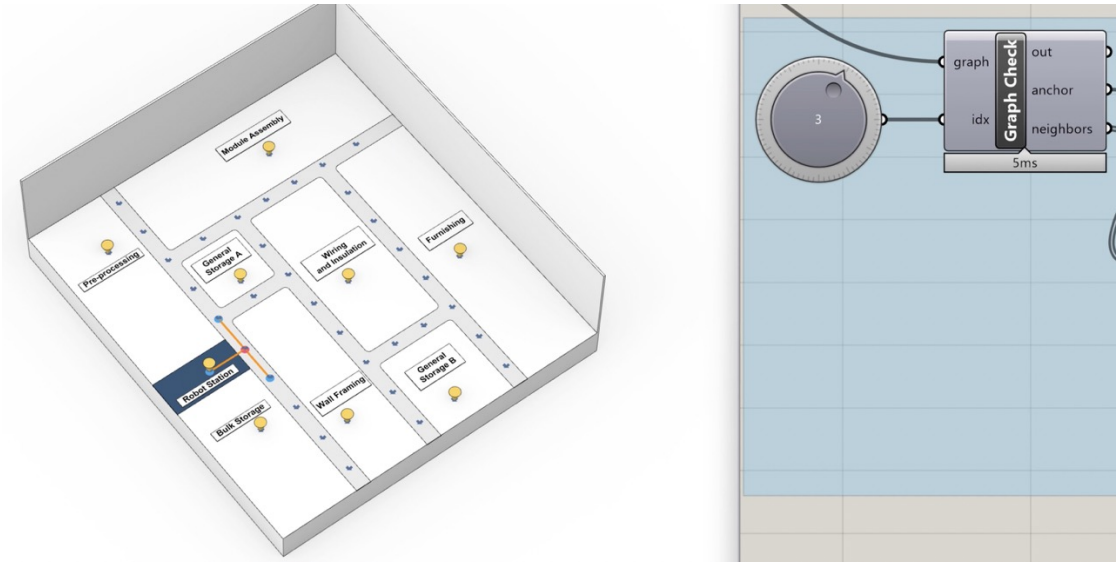


Figure 4.23: Graph checking program helps to check the connectivity of each point. Source: diagram by the author. Video: Appendix A.2

The next step is passing the built graph to the A* algorithm to test the path-finding functionality in the simplified factory setting. The test consists of two parts: the point-to-point test and the multi-step shortest path test.

The point-to-point test checks if the algorithm successfully finds the shortest path between designated start and target points. The Grasshopper's canvas provides an interface that helps to select the points interactively, and the result that appears on the Rhino's interface allows users to check the success of

the path-finding visually (Figure 4.24).

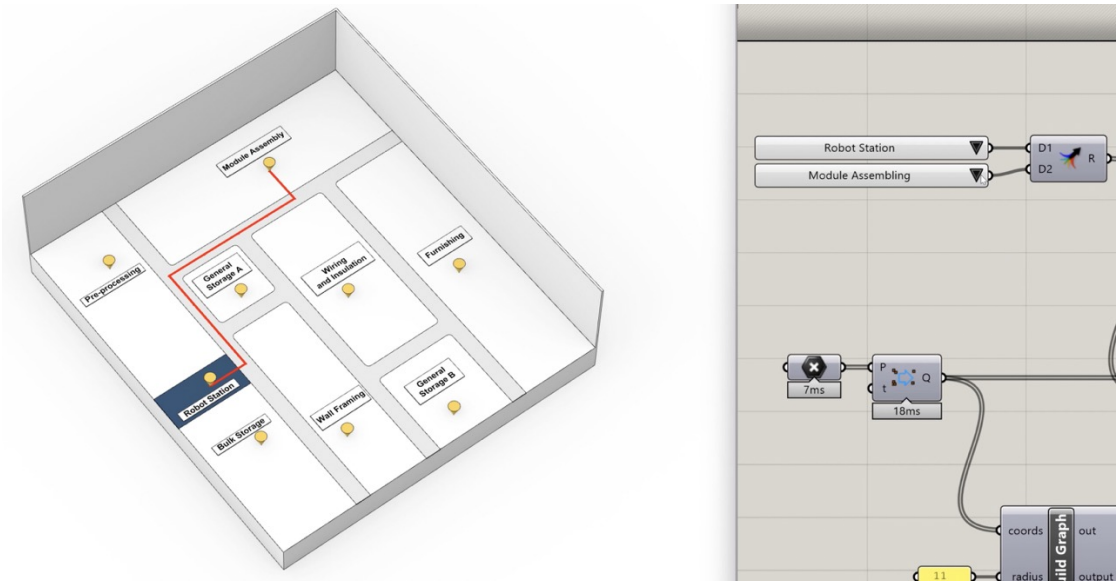


Figure 4.24: The path selection interface and the visualized point-to-point shortest path. Source: diagram by the author

After checking all possible paths of start-end point combinations, it is safe to conclude that the point-to-point path planning works well in the current setup. Now let's move to a more complicated test. The multi-step path planning test is closer to the real-world application scenario - the operator asks the robot to pick up materials from the storage and transports them to a working station. The complete path in this task is as follows: the robot starts from the robot station, moves to the storage station to pick up the item, transports the item to the target station, and finally goes back to the robot station to wait for the next command. The most important criterion for evaluating the multi-step task is the algorithm should not mix the target and the path in different steps; by assigning each path with different colors in the program, this task can be easily checked visually (Figure 4.25).



(a) path 1: from the robot station to the general storage A



(b) path 2: from the general storage A to the module assembly station



(c) path 3: from the module assemble station to the furnishing station



(d) path 4: from the furnishing station back to the robot station

Figure 4.25: Multi-step path-finding results. Source: diagrams by the author. Video: Appendix A.2

As there are an exponential number of possible combinations of paths, the test cannot test exhaustively check all possibilities, but the multi-step path planning is successful under 20 random checks. Therefore, it can temporarily conclude that the path planning algorithm works in the test scenario unless an unexpected error happens in the future.

4.4 Line-following

Using a graph-based A* path-planning algorithm does not grant the robot high degrees of freedom when moving in the factory; it requires the robot to strictly follow the path all the time before arriving at the destination. However, the algorithm's output is structured data of the shortest path, not the actual path the humans or the robot can sense. In this situation, a method should be proposed to

bridge the gap between the virtual representation of the path and the path with physical properties in the real world. The easiest way of fixing such a discrepancy is placing sensors at the exact coordinates of checkpoints, using dark lines and curves to connect all the checkpoints, and then building visible routes for all possible paths. Therefore, the robot can use the real paths as a reference to the shortest path from the A* algorithm to navigate the factory. This section introduces the principles and the algorithm of the line-following mechanism.

4.4.1 The line-following mechanism

The line-following has fixed and continuous lines on the ground to require the robot to follow. Unless the robot reads the sign of stop or turns around, it will keep moving along the pre-defined path, correcting itself from deviating from the line. The paths are usually painted in the dark to strongly contrast the light or gray ground. An array of infrared light sensors and LEDs will be installed at the bottom of the robot to keep tracking the robot's relative position to the line to calculate the necessary changes in motor behaviors.

A minimum number of IR sensors required for the line-following task is three - one in the center to check if the robot is following the line, and two in both sizes to tell the robot it deviates from the line in which direction. However, the three-sensor setup only works on the small robot because it only provides the direction of deviation but misses how much the robot deviates from the line. This is because the small robot can swing left and right frequently to find a proper pose to follow the line. Still, in a larger robot, the swing is energy-inefficient and might cause stability and safety concerns. In this case, a five-sensor, or more, setup is preferred because the extra sensors can help precisely calculate deviation and ask the motor to correct the robot's pose without blindness trails.

The proposed IR sensor setup for this project is - three sensors installed near the head of the robot to respond to gentle deviations and two more sensors installed more apart near the shaft of the front wheels to react to sharp deviations or turnings.

The line-following system works as (Figure 4.26) shows; the sensors read reflections from the ground and send the readings to the controller; the controller synthesizes the readings and compares them with a standard reference to get the deviation value. The deviation has then been decoded to the changes in speeds and directions for each motor and sent to the PID controllers to drive the robot back to the path.

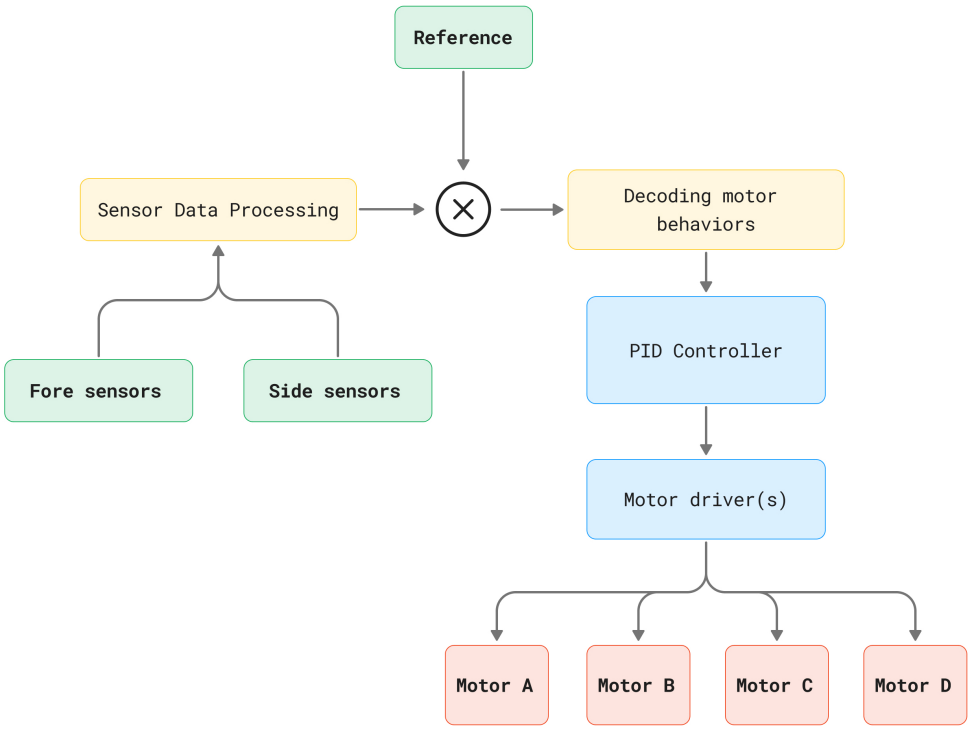


Figure 4.26: Line-following system flow chart. Source: diagram by the author

The equation for calculating the error from the five-sensor setup is as follows:

$$\varepsilon(t) = 1000 \times \frac{\sum_{i=1}^{5} i \times L_i(t)}{\sum_{i=1}^{5} L_i(t)} - 2000 \tag{4.12}$$

where $\varepsilon(t)$ is the error at the time t, $L_i(t)$ means the i-th sensor reading at the time t, and 2000 is a reference term to help normalize the error value to 0 when the robot is right on the line. In this setting, the negative error value indicates a deviation to the right, and a positive value indicates a deviation to the left. The absolute value indicates how much the robot is deviating from the line. With such error information, the controller can then send correction signals to the motor to bring the robot back to the line (Figure 4.27). This error-handling mechanism also helps the robot to make turns when encountering corners (Figure 4.28).

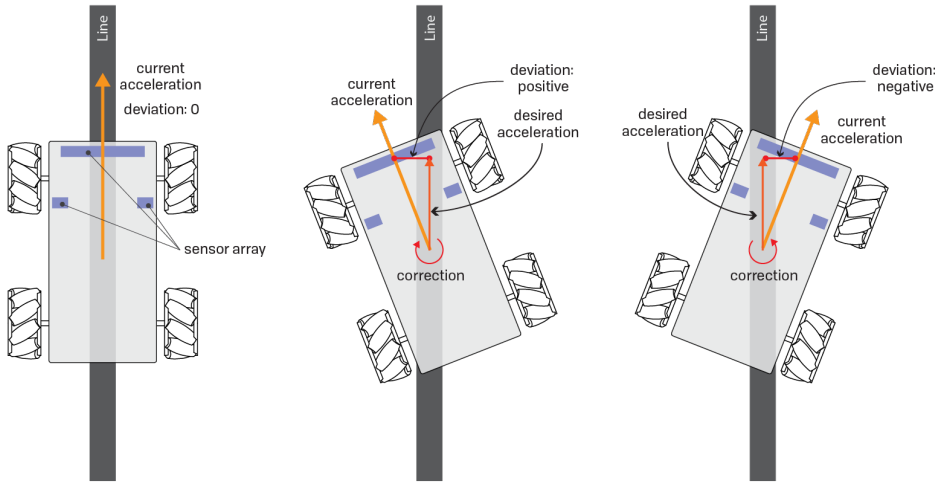


Figure 4.27: The line-following system helps the robot correct deviations. Source: diagram by the author

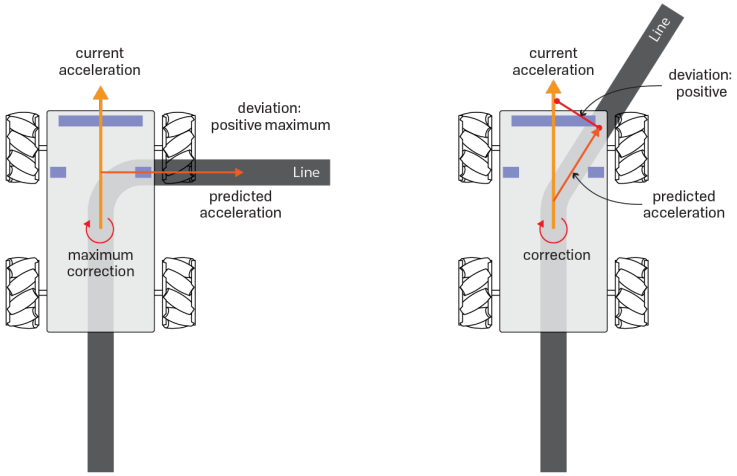


Figure 4.28: The line-following system helps the robot handle corners. Source: diagram by the author

4.4.2 Line-following experiment

The current state of the moving base is still in the experimental stage; it is not yet ready for the testing of the line-following behavior. Furthermore, the 4WD setup added another layer of difficulty when decoding the error signal to the change of motions for four wheels. Therefore, the test aims to verify the line-following setup rather than prove the full functionality of our task-oriented robot, so the experiment be conducted on a pre-configured robot with a two-wheel configuration and five IR sensors installed in this condition.

The robot for the line-following experiment is a palm-sized 3pi+ with a 30:1 motor gear ratio and

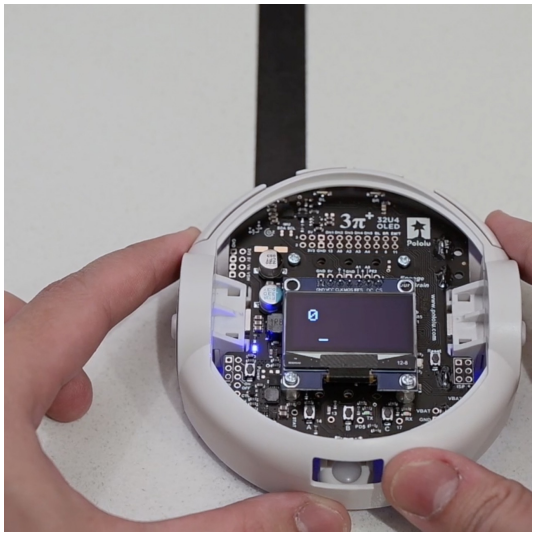
comes from Polou[5]; it is a small robot with a whole bunch of pre-installed sensors that can be used for various tests. The test uses the robot's API to read values from the IR sensor, but the whole line-following algorithm is written and implemented from scratch.

Deviation check

The test's first step is to read sensor values and use the function presented in the previous section to calculate the deviations. The error values will be printed on the screen to help monitor and debug the implementation.



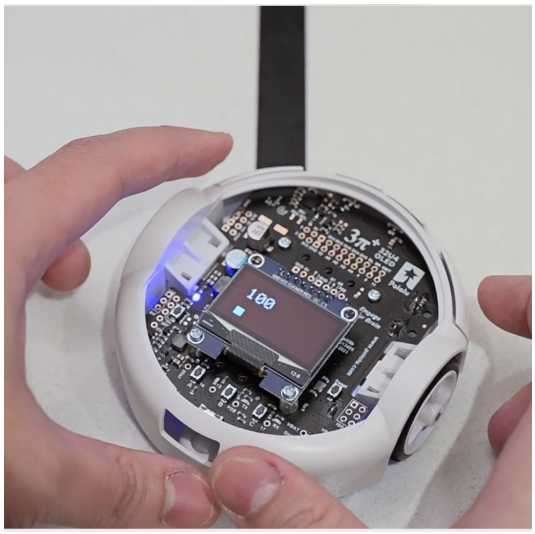
(a) The robot to the left of the line



(b) The robot to the right of the line



(c) The robot is deviating left from the line



(d) The robot is deviating right from the line

Figure 4.29: Deviation check. Source: photos by the author. Video: Appendix A.2

The values printed on the screen are raw losses before subtracting the reference value, so values are

[5]Pololu website: https://www.pololu.com/product/4975

always positive and range from 0 and 4000. In addition to the values, the amplitude of sensor readings is also printed as bars right below the error values to help check the responses from sensors (Figure 4.29).

A simple line-following test

The environment setup of the line-following test uses black tape to build a path on the light-colored background. The path consists of several line segments and 90-degree corners; the path ends with a terminal sign - a dark block on the ground that is thick enough to make all five sensors get full readings. The setting aims to test the robot's capability to follow the line and handle turnings; the test does not include decision-making aspects like making the correct turn when encountering crossroads or T-junctions (Figure 4.30).

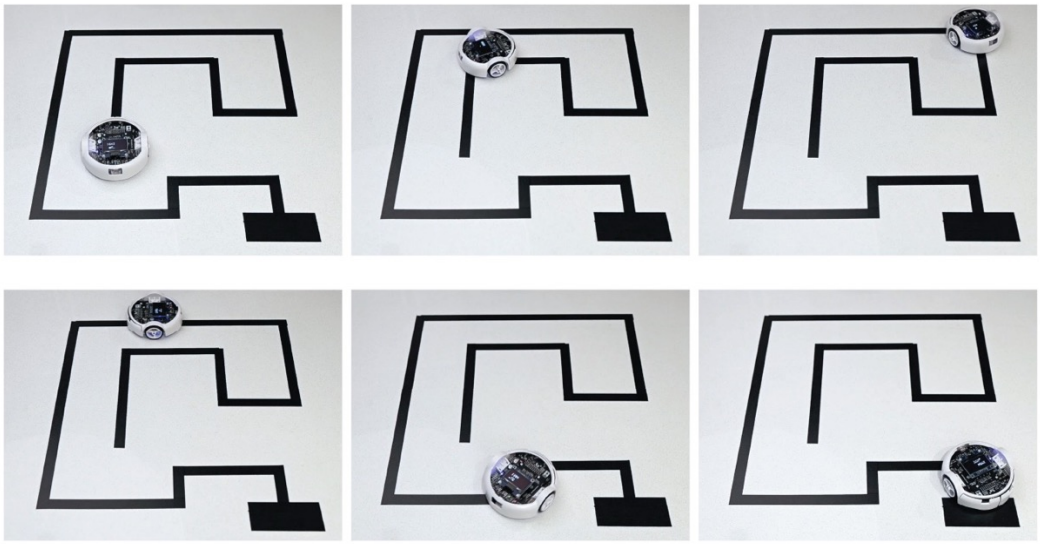


Figure 4.30: Simple line-following test. Source: photos by the author. Video: Appendix A.2

4.4.3 The limitation of the current setup

The line-following in the real-world material handling task is far more complicated than shown in the above tests. One of the most obvious limitations is that the current setup does not include progress checking and decision-making when the robot follows the line. In a real-world scenario, there are interconnected paths with intersections that require the robot to make the right decision whenever it faces multiple choices. In addition, the robot should also continuously read the checkpoint on the path to update its progress with the path-planning algorithm to ensure it is always on the right route.

Another limitation comes from the robot side. The current test is implemented on a two-wheel robot, but the robot in the proposed scenario has four wheels with 4WD, which needs a more delicate error decoding and coordination for all motors. The line-following algorithm also does not fully utilize the mobility of the Mecanum wheels but only uses them as regular wheels.

Solving the above problems requires a lot of effort and time that cannot be done within one or two years. Furthermore, developing a line-following algorithm that fully uses the Mecanum wheels is technically challenging and needs to explore in another master-level research. Therefore, these problem does not include in the scope of this thesis.

4.5 Collision Avoidance

Although the robot only follows the designated paths in the factory, the paths are not isolated, and pedestrians or other objects may still be present near the robot and cause collisions. Therefore, adding the collision avoidance feature to the robot helps protect the robot and pedestrians when it navigates in the factory.

The most commonly seen collision avoidance method on the line-following setup uses distant-detecting sensors, which helps the robot become aware of the distance between the surrounding objects and itself and take action when some objects are getting too close. As the simplicity of our problem setting, collision avoidance does not need to be so intelligent that it requires arithmetic-intensive algorithms; it can be implemented under fuzzy logic (Almasri, Elleithy, and Alajlan 2016) that makes decisions based on imprecise observation.

4.5.1 Strategy

The real-world collision avoidance system considers many factors more than distance, like the type of obstacle, the relative speed between the obstacle and self, the obstacle's possible actions, and the probability of ways the object avoids colliding with the obstacles, etc. These factors require precise sensor readings and high-level control theories that make the workload and difficulty of this thesis overwhelming. As collision avoidance is part of the proof of concept in this project, this section aims to propose a viable, also minimal, solution to make the system work without diving too deep into this field.

We need to define three things for the collision avoidance strategy: what is the obstacle, how to detect it, and how to avoid colliding with the object.

Obstacle

There are static and dynamic obstacles in the factory. Static obstacles are permanent structures or stationary objects in the environment, like columns, shelves, heavy machinery, etc. The modules can also be considered static obstacles as they always stay in working stations and do not move around frequently. Because our robot only follows the predefined paths in the factory, and the paths are carefully set up to avoid all static obstacles, theoretically, there is no need to consider static obstacles, at least in our minimum setups.

The dynamic obstacles in the factory are mainly humans; they frequently move around and can often appear on the path, causing damage or injuries to the robot or themselves. Therefore, we define the humans in the factory as the main obstacle to avoid for the robot.

Detection

A wide range of detection methods can help the robot figure out the potential obstacles in its route, like using distance sensors to check if the robot is approaching something in the scene, using cameras to recognize the object that needs to be cared about, or using an even advanced algorithm to not only recognize object but also predict its movement to decide whether stop or proceeding. The latter two methods require machine learning and a controller with high computational capability, which is out of this project's scope; therefore, we choose the distance-detecting method to ensure the task is achievable.

The commonly seen computationally cheap distance detection sensors include ultrasonic sensors, infrared reflection detection sensors, and the much more advanced light detection and ranging sensors (LiDAR). The previous two sensors can only detect changes in the distance in a single direction with a very narrow window; they cannot feed back full information about the robot's surroundings unless we

install hundreds on them, which sounds impracticable. On the contrary, 2D LiDAR is a single device that can scan 360 degrees and return much more precise distance information with the corresponding angle. Considering the maneuverability of the Mecanum-wheeled robot, knowing the distance between the robot itself and all of the surroundings helps the robot move freely without colliding with other objects. Therefore, we choose 2D LiDAR as the distance sensor for the robot to detect potential obstacles when performing material-handling tasks in the factory.

Action

The only thing remaining for collision avoidance is: what actions should the robot take when potential obstacles are detected? If placing this question in the general case that the robot can move freely in the factory, the answer seems relatively straightforward the robot can bypass the obstacle and replan the path. However, in the setting of this project, the robot stays on the line when moving; if the robot bypasses the object, it still needs to find its way back to the line and makes the line-following somewhat a burden for collision avoidance.

A simple solution is to ask the robot to stop when an obstacle is closer than the threshold of the safety distance, which is a parameter that the operator can define based on the robot's speed and the potential obstacle's moving speed. The robot can resume line-following behavior when the path is clear again. Nevertheless, in the real world, the obstacle's states can be more unpredictable, that simply stopping the robot is not a preferable solution; this will be further discussed at the end of this section.

4.5.2 Hardware setup and virtual simulation

Hardware setup

The only way to install 2D LiDAR (Figure 4.31) on the robot is by placing the device on top of it, giving it a better view when scanning the surroundings (Figure 3.48). The LiDAR emits and receives laser beams while rotating and returns an array of distance readings corresponding to specific angles; that means we know not only how close an object is but also where it is in 2D space. The higher the LiDAR's precision, the more data it returns per rotation (Figure 4.32). The LiDAR's other specification is the detecting range, which is the closest and the farthest distance the LiDAR can detect; the measuring range is device-specific and commonly ranges from 5 to 50 meters. 10 to 20 meters of measuring range is enough for this project as the robot does not move fast.



Figure 4.31: Slamtec 2D LiDAR. Source [6]

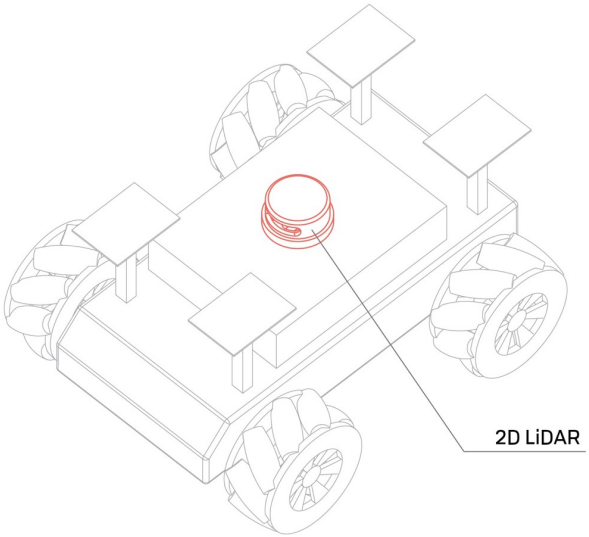


Figure 4.32: The 2D LiDAR installed on top of the robot. Source: diagram by the author

Virtual simulation

With the strategy proposed and hardware setup defined, the collision avoidance method is then tested in the virtual simulation with minimal setup. The simulation is running on Rhino 7, with tasks being automated with Grasshopper and GhPython, and the scenario is the robot following a straight line while a person presents on the path. Furthermore, for the simplicity and clarity of the simulation, the robot's danger radius has been set to 1 meter and visualized with red dots.

[6] Slamtec RPLidar A1 product page: https://www.slamtec.ai/home/rplidar_a1/

The first scenario is a person showing up near the path in front of the robot but beyond the robot's danger radius. In this condition, the robot can keep moving forward as the values returned from the LiDAR aren't lower than the preset safety threshold (Figure 4.33).

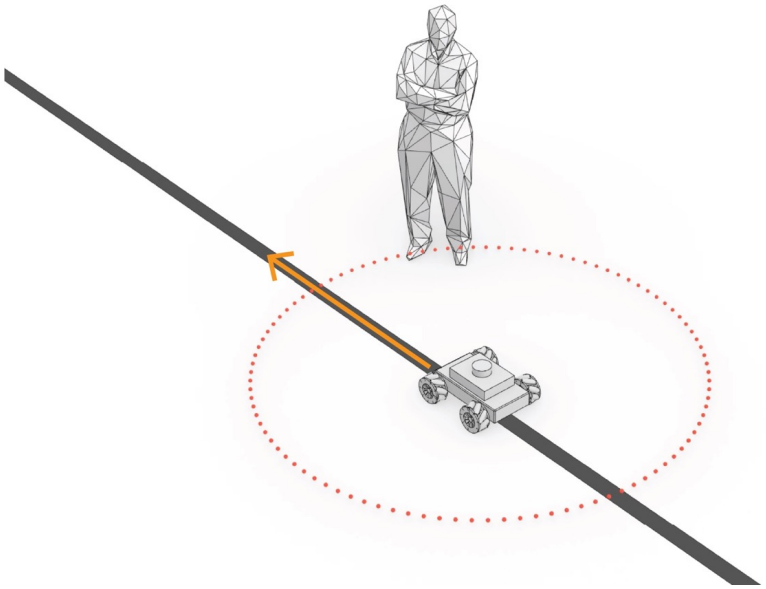


Figure 4.33: A person shows up near the path but beyond the robot's danger radius. Source: diagram by the author

The second scenario is quite similar to the first one, except that the person is within the robot's dangerous radius. As shown in Figure 3.51, some LiDAR readings from the front-right of the robot are lower than the safety threshold, that force the robot to stop until all readings are higher than that threshold again.

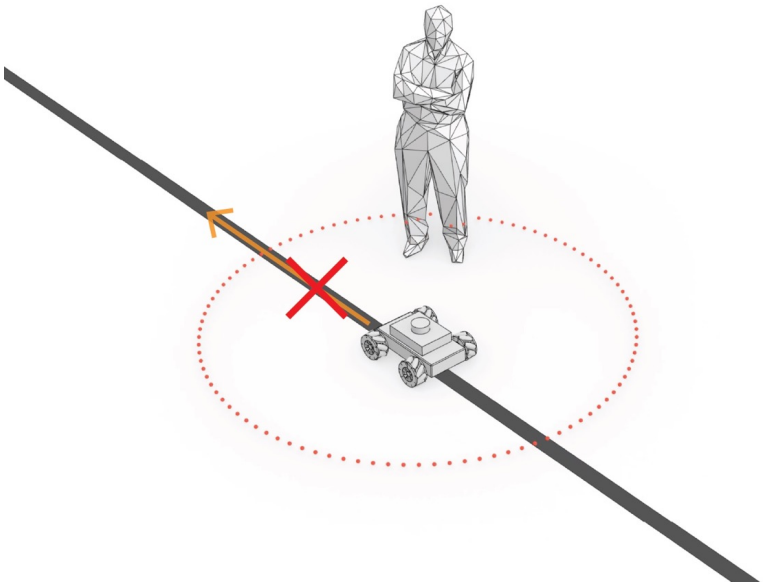


Figure 4.34: When LiDAR detects an obstacle that is closer than the safety threshold, the robot stops until the path is clear again. Source: diagram by the author

The third scenario is that a person shows up at the back of the robot at a distance smaller than the preset safety threshold. In this situation, although the LiDAR returns small distance readings, the readings come from angles that are opposite to the robot's current movement; there's almost no possibility the robot will collide with the person unless the person walks straight towards the robot, then the robot can consider such situation is safe and keep moving forward (Figure 4.35).

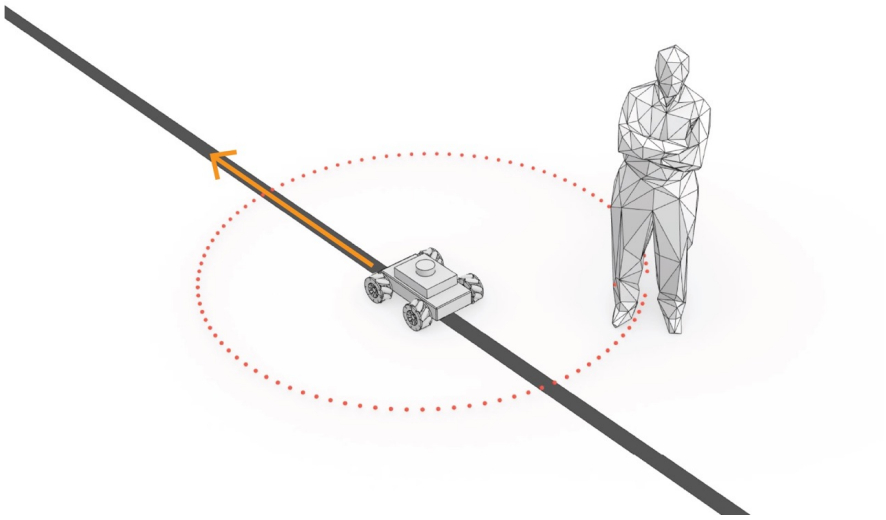


Figure 4.35: A person shows up close to the robot but opposite to the robot's current moving direction; this situation is considered safe, and the robot keeps moving. Source: diagram by the author

4.5.3 The limitation of the current method

As the chapter serves as the proof of concept for the automated material handling in the prefabricated housing factory, the collision avoidance strategy is based on an ideal setup that ignores the complexity of the real factory environment. Here we discuss several conditions where the current solution may fail or not effectively solve the problem.

- **The perfect path does not always exists:** The previous section mentioned that "the paths are carefully set up to avoid all static obstacles, theoretically" when defining the obstacle, but the fact is that the factory's space isn't so generous that it allows drawing a path away from all static obstacles; a highly possible scenario is that the path goes through narrow alleys to connect stations. In this situation, the LiDAR will return short-distance readings from many directions that make the robot stop and never go through that path. Arbitrary ignoring the sensor readings from several directions cannot fully adapt to all possible occasions. An effective way of solving this problem is using other object-detection methods or fusing readings from different types of sensors to make predictions on whether the object is an obstacle or just a fixed item.
- **The pedestrian may not be aware of the robot:** As the robot is relatively small, it will be easily ignored in the noisy and busy factory, especially when it does not carry materials. Sometimes workers are distracted by something happening in the factory and walk toward the robot without awarding it. In this situation, if the robot just stopped and waited there, the worker may run into the robot and get hurt, and the consequence will be even worse if the robot is carrying heavy or sharp materials. To solve this problem, collision avoidance should be flexible enough to respond to the dynamic obstacle's behavior and make the right decision when multiple dynamic obstacles appear nearby.
- **The obstacle stays for a relatively long period:** Another possible scenario is that the obstacle may stay on the robot's path for a relatively long period, like someone places materials on the path and forget to move them away or some equipment, say a movable lifter, broken right on the robot's way. It makes no sense for the robot to stop and wait for the path to be cleared on such occasions. A possible solution is to set a timer inside the robot; when the waiting time is longer than a threshold, the robot can either replan the path or notify the technician to check what's happening in person.

There are far more unforeseen situations that could make collision avoidance fail in the factories than discussed above, so choosing a correct collision avoidance solution should base on the fact of the application scenario and keep improving it by solving different problems in daily operations.

Chapter 5

Experiment

This chapter consists of two parts - a virtual simulation of a material handling task and hardware development of the robot's moving base. The virtual simulation chapter puts all background study and methods components together, sets up a virtual scene of a prefabricated housing factory, and simulates a complete material-delivering task. The hardware development section presents the challenges of developing a moving base that meets the robot's requirement of maneuverability and shows how the problems and difficulties have been addressed.

5.1 Virtual Simulation

Virtual simulation is essential for testing proposed concepts and methods presented so far as a whole system in the experimental setting; it helps save time and resources in verifying current solutions and troubleshooting compared to doing a real-world scenario. The virtual simulation does not aim to present a perfect material handling task or show a production-level demo; the purpose of the simulation is a proof of concept of the mobile-robot-assisted material handling with all methods and technical pieces proposed in previous chapters, and more importantly, it helped the problems and limitations of the current method.

The section starts with the environment setup, discusses the criteria for setting up a minimal virtual factory space for testing, then defines a task that can thoroughly test features of the thesis proposed so far. After running the simulation, we'll analyze the problems and limitations spotted in the process, along with possible improvements.

5.1.1 Environment and task setup

The simulation runs on Rhinoceros 7, where modelings are done manually, and the simulation is automated with Grasshopper, GhPython, and C#. The principle of the simulation is to have as few human-interference and hard coding as possible, but due to the complexity of the system, There are still some works that cannot be fully automated, like the robot picking up and dropping off the cart and the human movement.

Environment

We're using a compact version of the wood factory as the test bed, which has a similar layout as the factory in New Hampshire, and with all stations included; the difference is that each station is smaller than the actual ones and has only one or two building modules inside. The pick-up and drop-off locations in each station are arbitrarily defined for the simplicity of the simulation, and paths

are drawn manually following the path-defining criteria proposed in Chapter 4 (Figure 5.1). The environment also includes the feature of human movement for collision avoidance testing.

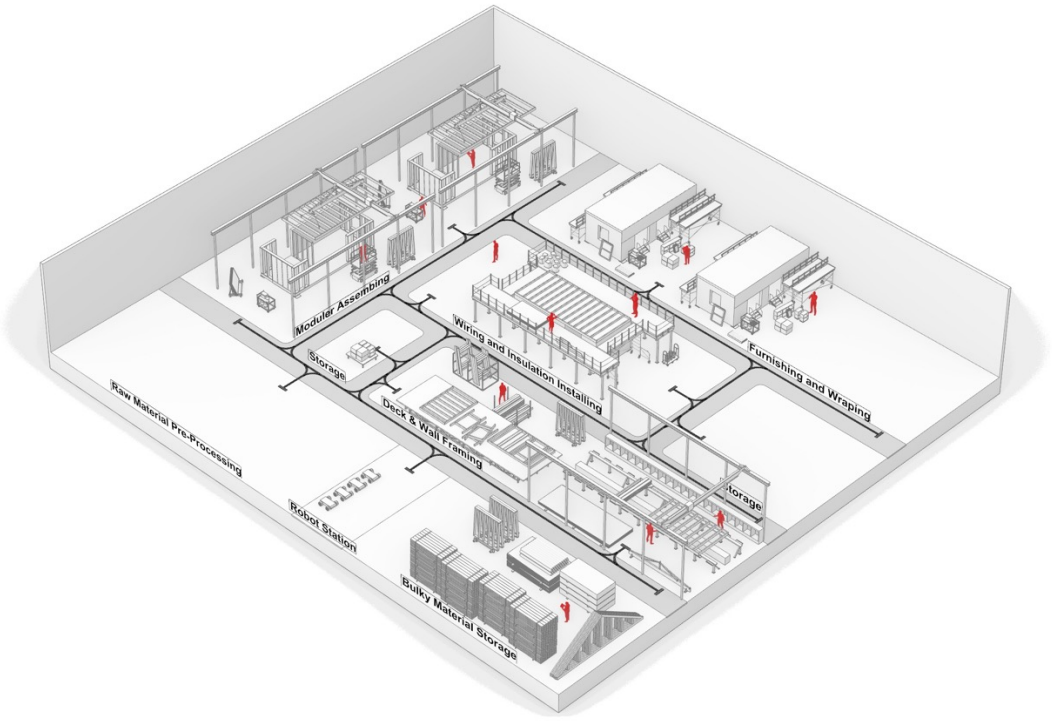


Figure 5.1: Environment for the virtual simulation. Source: diagram by the author

Task

The task of the simulation is to test one complete process of material delivery: the robot is requested to transport a cart of materials from one of the storage spaces to the furnishing and wrapping station. When receiving the request, the "control center" will first do the multi-step path planning, find the shortest path from the robot station to the general storage space, and then get the shortest path from the storage to the drop-off place of the furnishing and wrapping station. After getting the paths, one of the robots starts following the line to pick up the cart and delivery it to the destination. A person will show up on the path when the robot is on the move, and the robot should be able to detect the pedestrian and stop when the distance between the obstacle and itself is too close. The robot should also follow the shortest path back to the charging station, but this step is duplicated from the delivery steps, so we can ignore this last step in the simulation (Appendix A.3).

5.1.2 Simulation

Path planning

When getting a request from the furnishing station, the control center checks the inventory to find where the requested material is located. Then a worker in the corresponding storage space will be notified to load the material on the cart and move it to the loading position in that space. When the material preparation is completed, the control center will do the multi-step path planning, finding the shortest path from the robot station to the storage and the storage to the furnishing station (Figure 5.2).

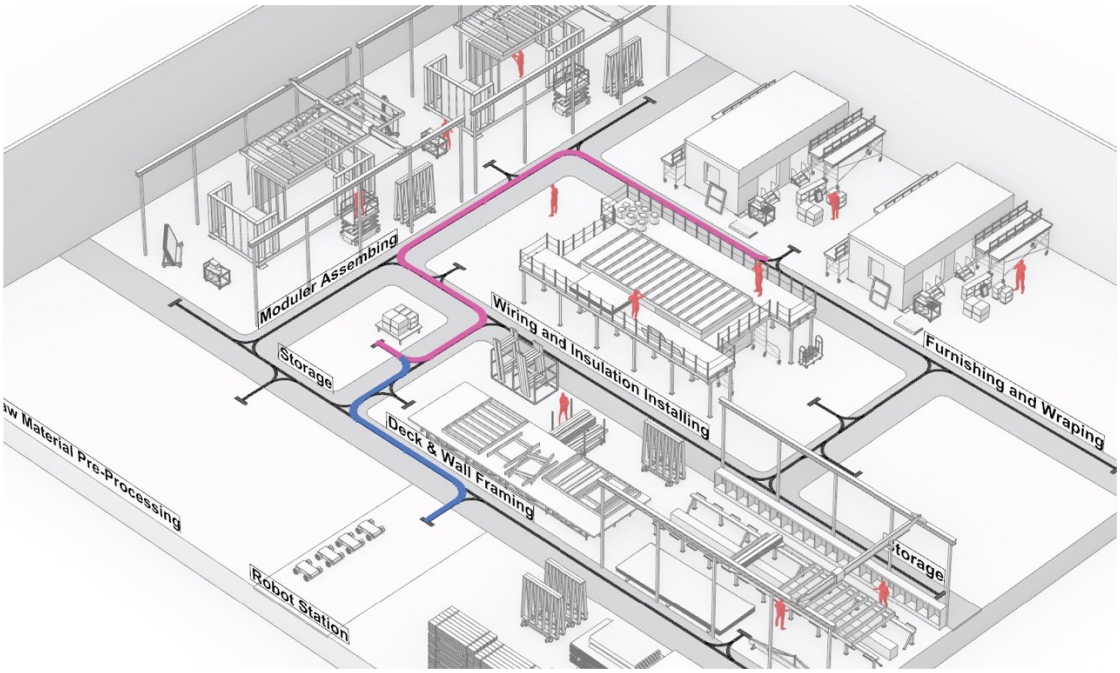


Figure 5.2: The control center finds the shortest path from the robot station via the storage to the destination. Source: diagram by the author

Moving to the storage space

With the path ready, the control center will check the stand-by robots' battery level and assign the fully charged one to do the task. The selected robot leaves the charging station and follows the first path toward the storage space to pick up the cart (Figure 5.3).

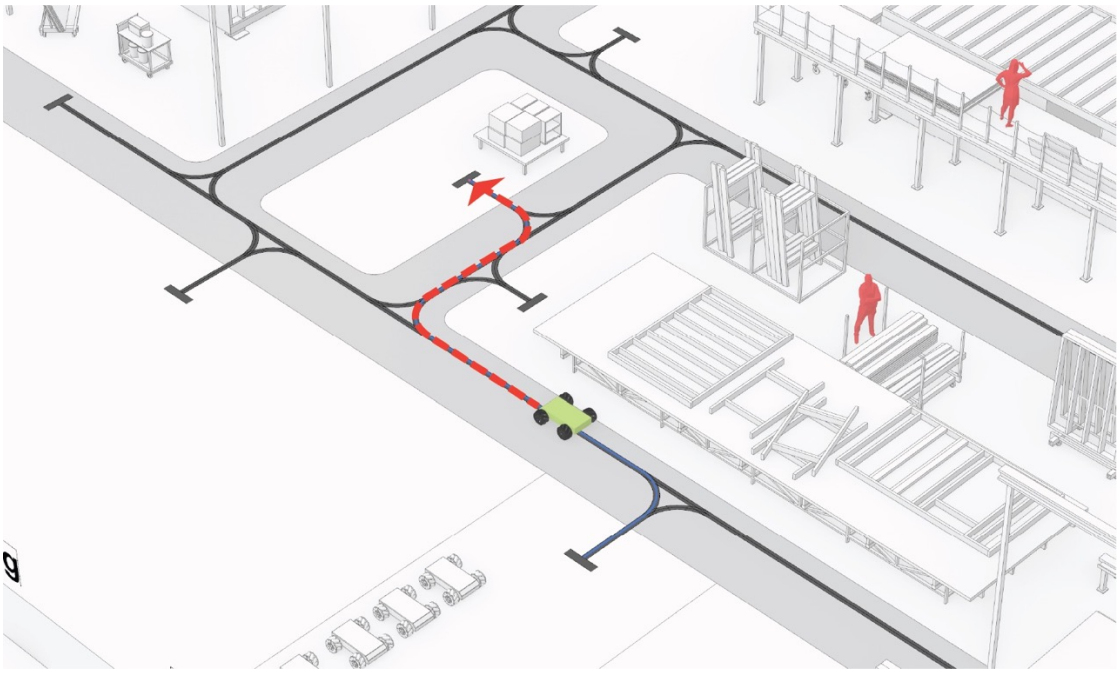


Figure 5.3: The robot follows the first path, moves toward the storage space. Source: diagram by the author

Picking up the cart

When arriving at the storage space, the robot first turns around, facing the path (Figure 5.4). This extra step before picking up the cart helps reduce the robot's workload to turn around with the heavy material, which exerts strong inertia when its orientation is changed. Then the robot slides beneath the cart, finds the cart's weight center, and raises the lifter to connect with the cart (Figure 5.5). The robot carries the material and slides back to the path when the connection is made and safe (Figure 5.6).

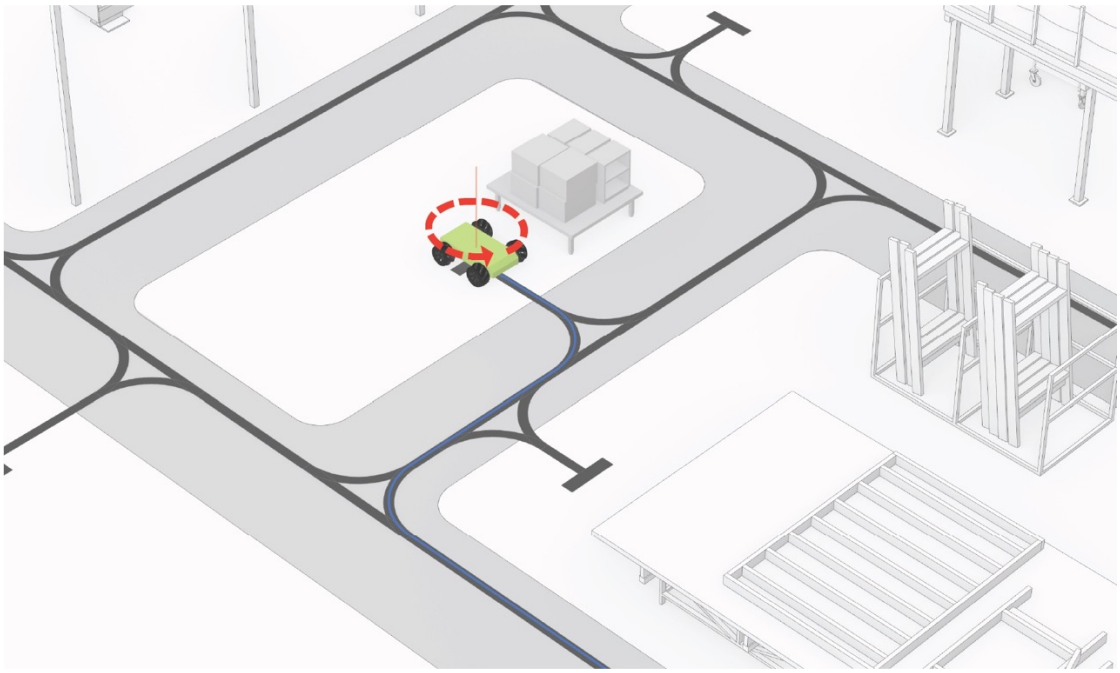


Figure 5.4: The robot turn around to face the path when arriving at the storage space. Source: diagram by the author

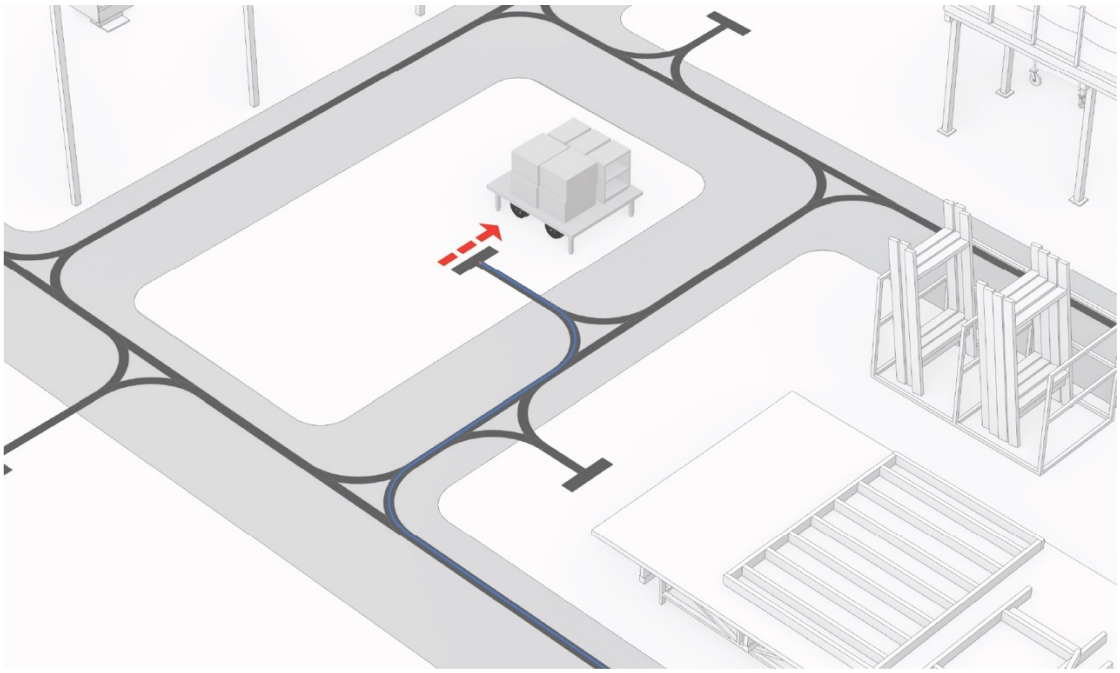


Figure 5.5: The robot slides beneath the cart and connect with it. Source: diagram by the author

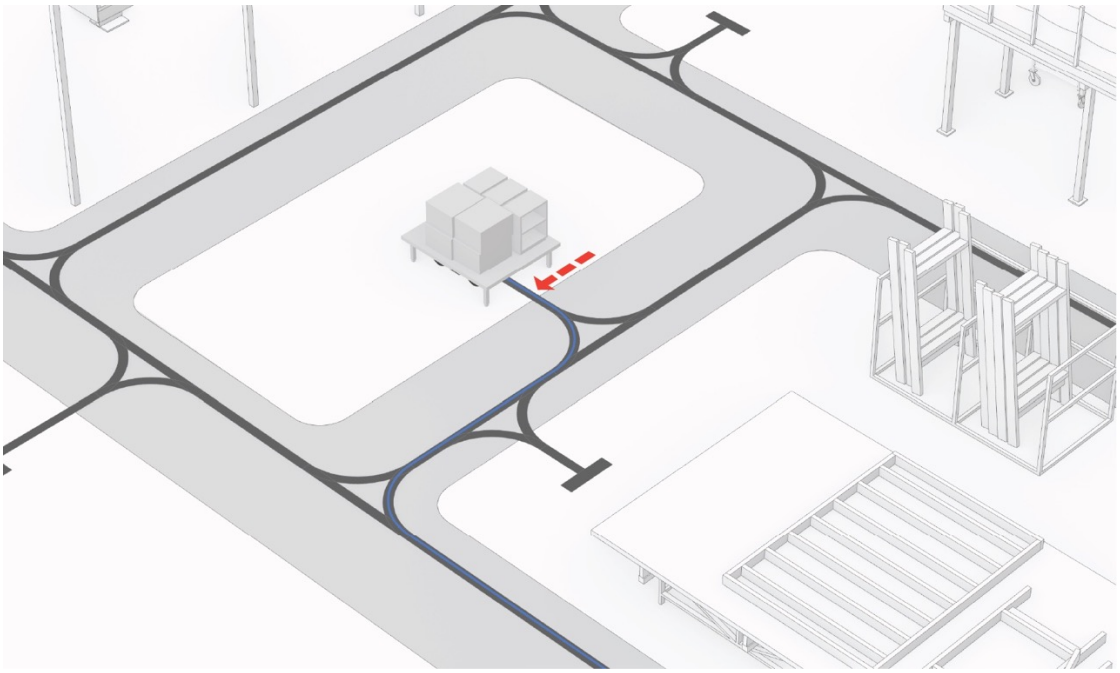


Figure 5.6: The robot carrying the material slides back to the path. Source: diagram by the author

Moving toward the destination

The robot's next step is to carry the material to the destination by following the second path (Figure 5.7). In this part of the simulation, we purposefully control a pedestrian moving across the path when the robot is approaching to test if the robot can properly stop when detecting a possible collision. The test is successful: the robot stops when the pedestrian is walking across the path and resumes line-following when the path is clear again (Figure 5.8, 5.9).

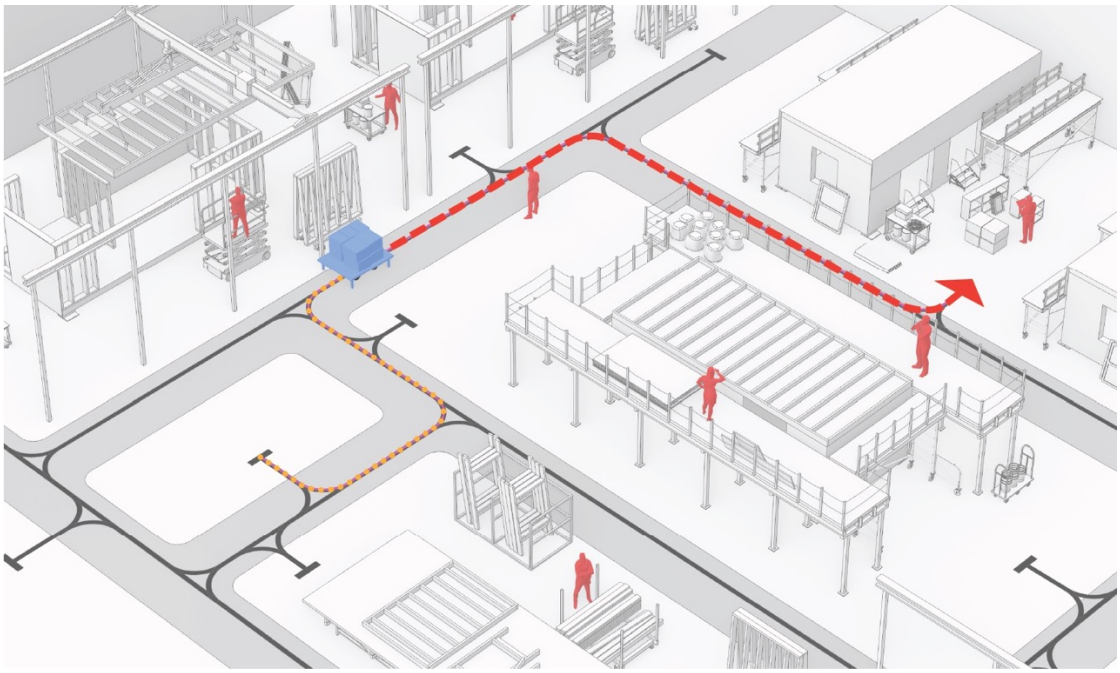


Figure 5.7: The robot carries the material and follows the second part of the path. Source: diagram by the author

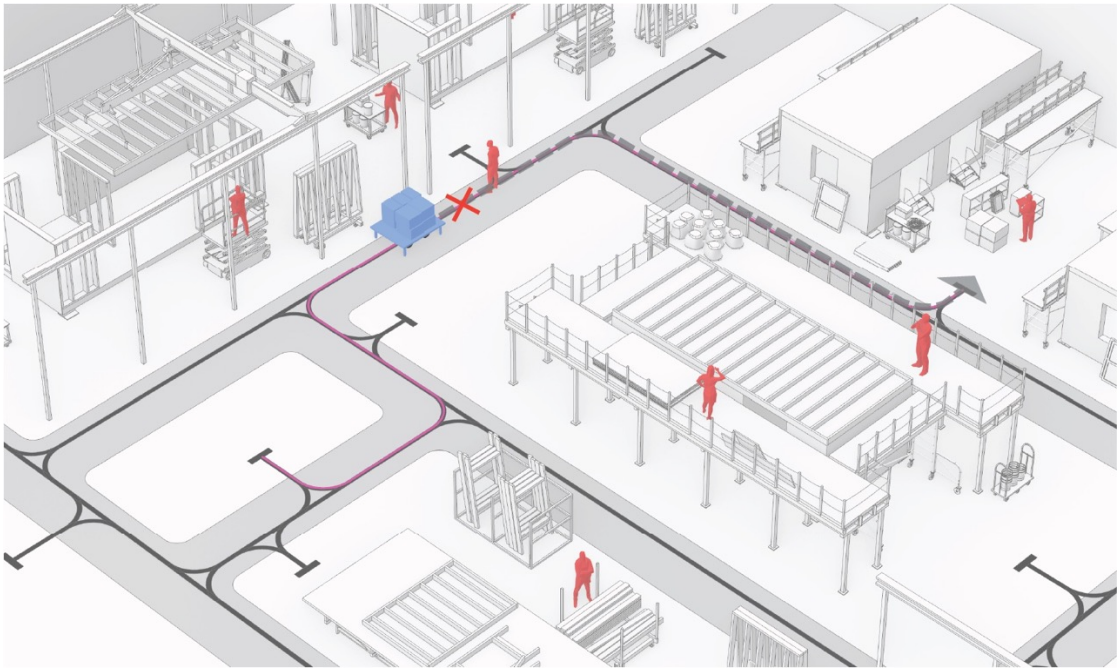


Figure 5.8: The robot stopped when detecting a possible collision. Source: diagram by the author

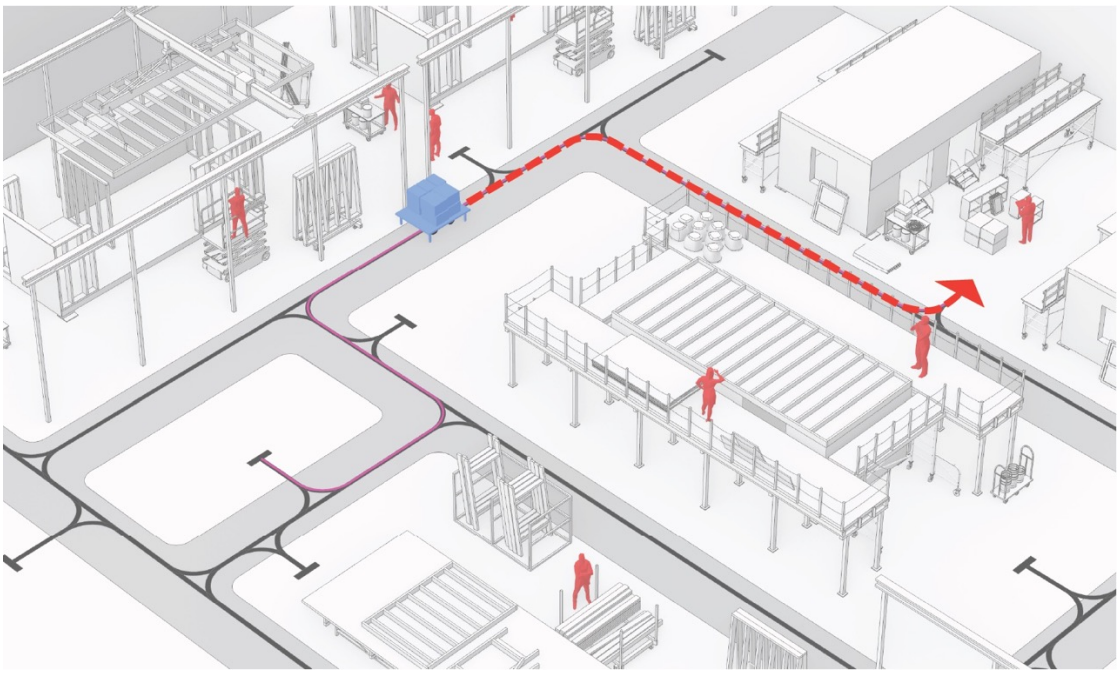


Figure 5.9: The robot resumes line-following when the path is clear again. Source: diagram by the author

Drop off the cart and return to the path

When arriving at the destination, the robot slides to the drop-off position, lower down the lifter to disconnect the cart, and slides back to the path (Figure 5.10). So far, the whole material-delivering task has been successfully completed, and the robot can head back to the charging station. Returning also consists of path-planning and line-following that is similar to the material delivery and can be considered redundant in the simulation, so we ignore it and finish up the whole virtual simulation.

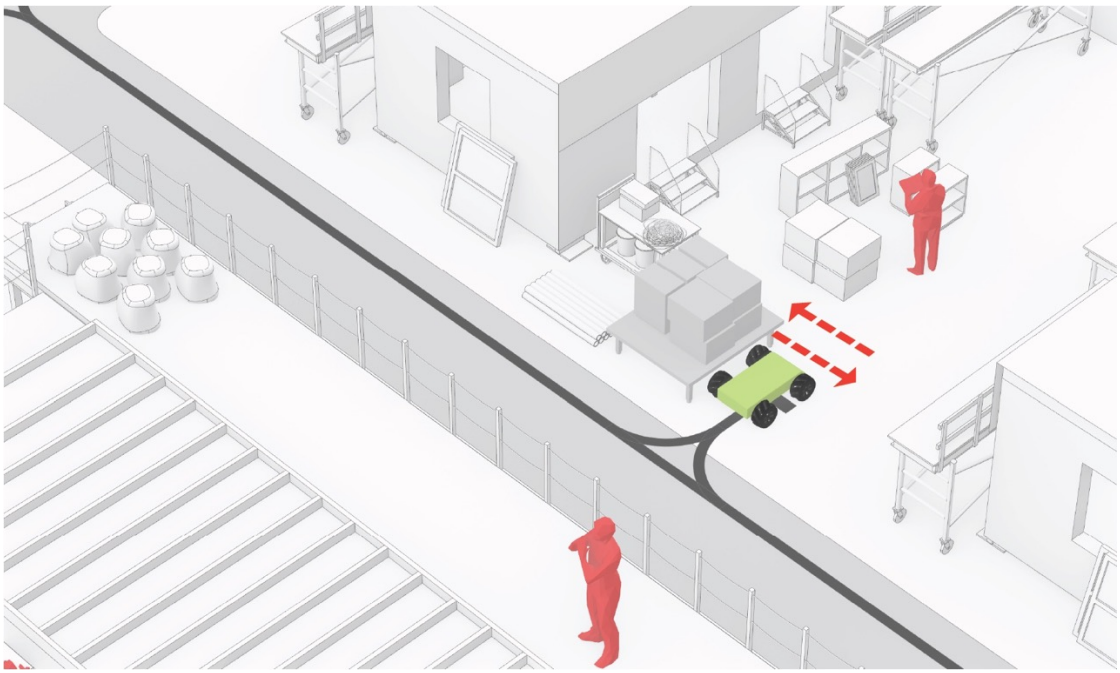


Figure 5.10: The robot drops off the cart and returns to the path. Source: diagram by the author

5.1.3 The problems in the simulation

In Chapter 4, we already discussed the possible problems and limitations of the current line-following and collision-avoidance methods; the simulation also encountered some of those problems, like failing to make the right decision when encountering intersections on the path and falsely detecting obstacles because of the narrow path, and they were only addressed by hardcoding. We don't expand discussions on those problems in this section as they were already discussed in the previous chapter.

There is another problem that wasn't discovered in the method chapter but appears in the simulation and turned out to be difficult to solve; that is how does the robot know the position of the cart in the storage and how can the robot decide where to drop off the cart at the destination. In both scenarios, the robot has to leave the path for picking up and dropping off the part, but no method proposed in the previous chapter can help the robot locate the cart or find a safe place to drop it off. The solution to this problem in the simulation is that, in the storage space, the cart was placed right next to the endpoint of the first path so that the robot could slide right beneath the cart. In the destination, there is an open space right next to the end checkpoint as well, and then the robot can slide left to drop off the cart. The general and effective solution to the problem needs more future work, so it is not in the scope of this thesis.

5.2 Moving Base Development

The hardware part of the experiment chapter aims to develop a moving base based on the scenario proposed in the hypothesis chapter and the circuit design with the selected components proposed in the methods chapter. The moving base in the context of mobile robotics is the chassis with motors and wheels that allows the robot to change its position or location in physical space; in this case, the selection of the Mecanum wheels endowed the robot with a higher maneuverability that can move in a wider direct and rotate around more axis compared to the normal wheels. Therefore, this section will

talk into details of moving base development and verify the effectiveness of circuit design, hardware selection, and the final product's maneuverability.

5.2.1 Challenges

A prototype of the moving base was finished at the end of chapter 3.2; the chassis was plugged with external power sources and went through a preliminary functional test, and all the motors, controllers, and drivers could respond to the speed and direction signals. The next thing is to replace the external power with batteries and test the robot's movement in a larger space.

Due to the different power consumption levels between the controller and the motors, we use a 9-volt single-use Alkaline battery for the Arduino Mega controller and a 3500 mAh 12-volt rechargeable LIPO battery for the motors (Figure 5.11). A voltage regulator is installed on the Arduino board to lower the input voltage from 9 to 5 to make the board flexible to the power source with a voltage range between 5 to 20 volts. With the power set up, the robot was first tested in the MMCH 503 Code Lab at Carnegie Mellon University.

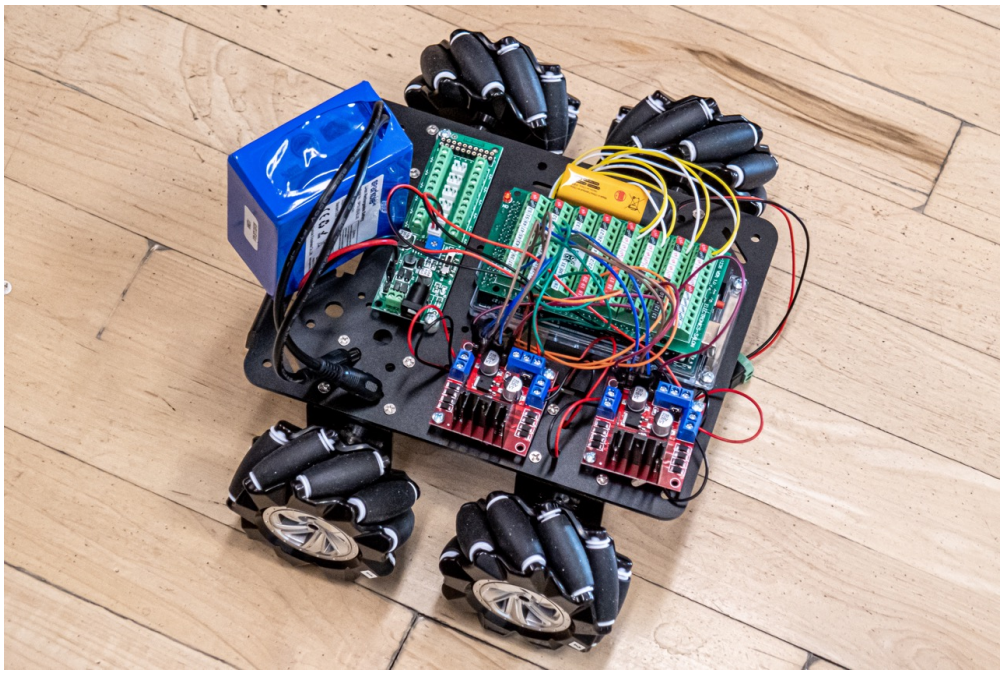


Figure 5.11: The robot powered by batteries. Source: photo by the author

As this is the first ground test, we still provide power for the controller with external power for safety; if the robot runs into any dangerous situation, we can just quickly drag the power cable to make the robot stop. The test asks the robot to move forward at a small distance to check if the four motors coordinate correctly.

The problem of the four-motor coordination appears at the first 1.5 meters of the move. In Figure 5.12, the robot's original position is on the upper left; the moved position is on the lower right. The red line with an arrow indicates the robot's actual path, while the white dashed line is the reference straight path. It is obvious from a glance that the robot is actually not going straight.

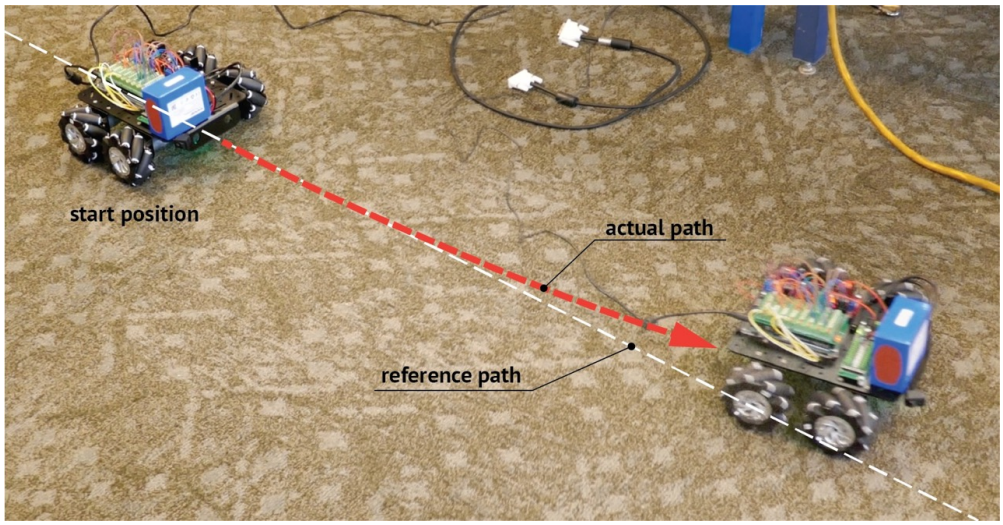


Figure 5.12: Indoor short-distance straight-line test. Source: diagram by the author

After running several identical tests, it can be confirmed that the deviation is not a coincidence; the robot never goes straight in all of the tests. Instead of trying to find and solve the problem immediately, a long-distance test was conducted in a wider space on the 4th floor in MMCH to test how worse the motor-coordination problem was. The result was not optimistic: the robot never goes straight in all tests, and the robot always deviates to the left of its moving direction with no exceptions; the difference between tests is only the degree of deviations - sometimes slight and sometimes significant (Figure 5.13, 5.14).

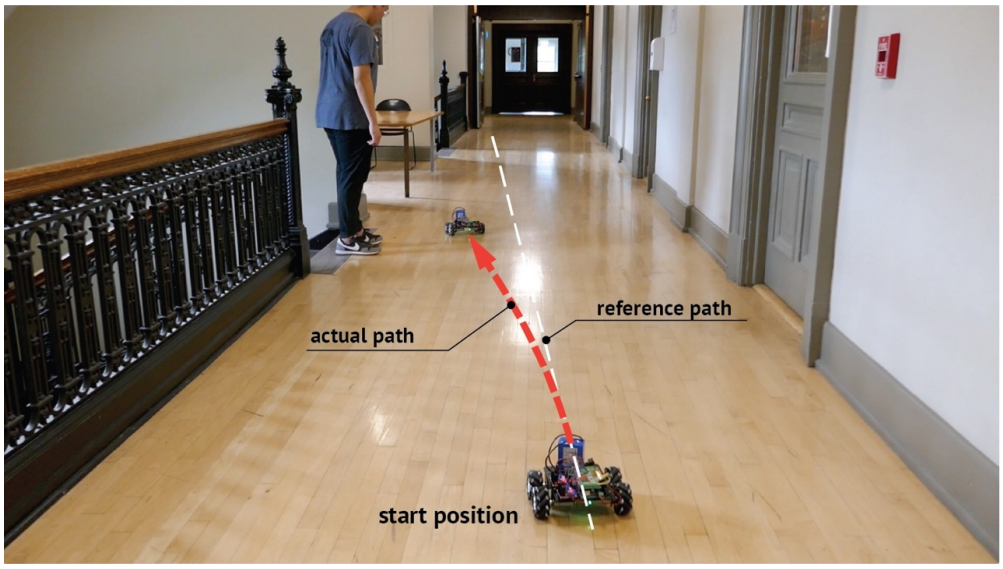


Figure 5.13: Long-distance straight-line test 1. Source: diagram by the author

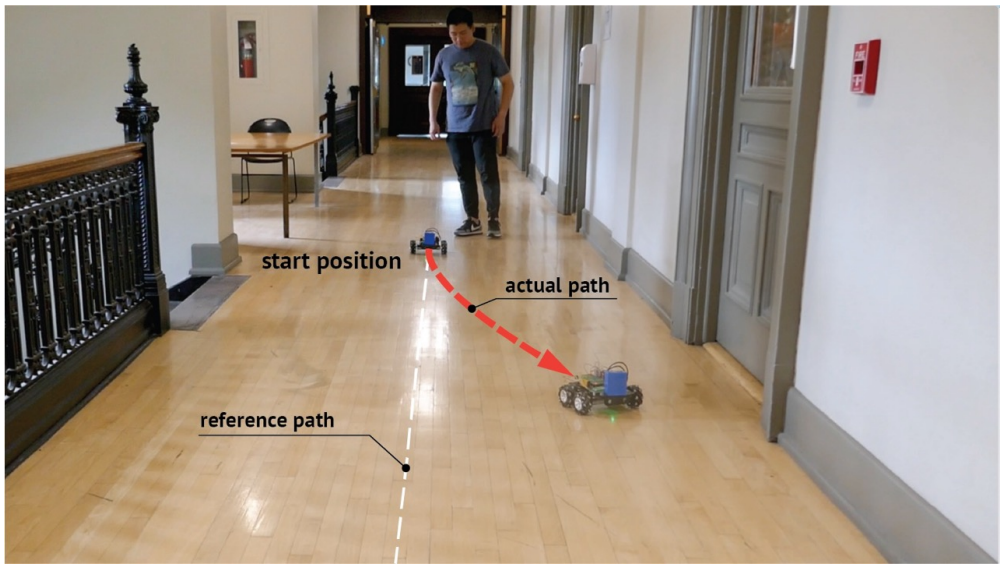


Figure 5.14: Long-distance straight-line test 2. Source: diagram by the author

What is gratifying is that these observations provide a clear hint of where the problem is - one of the motors running slower or faster than the rest, which causes the deviation rather than swinging, and the voltage on that motor is unstable, which is the reason why the degree of deviation differ between tests.

5.2.2 Updated robot design

It turns out to be the left rear motor is dysfunctional, one that runs slower than all the others. However, several encoder reading tests show that the problem is far more complicated than the observations: the voltage on all four motors is unstable, all running at slightly different speeds, with the dysfunctional runs even slower. This problem makes me wary about the quality of the motors.

Around twenty motors with the same brand and specs have been tested on the same voltage and speed and use 2000 encoder readings as a baseline check. The result is that the standard deviation of the 2000 readings is about +/- 300, which is significant that can never enable the robot to move along the provided direction with the given speed. Therefore, the solution is only using much more reliable motors with better build qualities.

Taking the opportunity to change the motors, we can also update the chassis and the layout of electronic components, as the current chassis is only for test purposes. To avoid similar motor problems on the previous setup, each motor has been strictly tested before being installed on the new chassis (Figure 5.15).

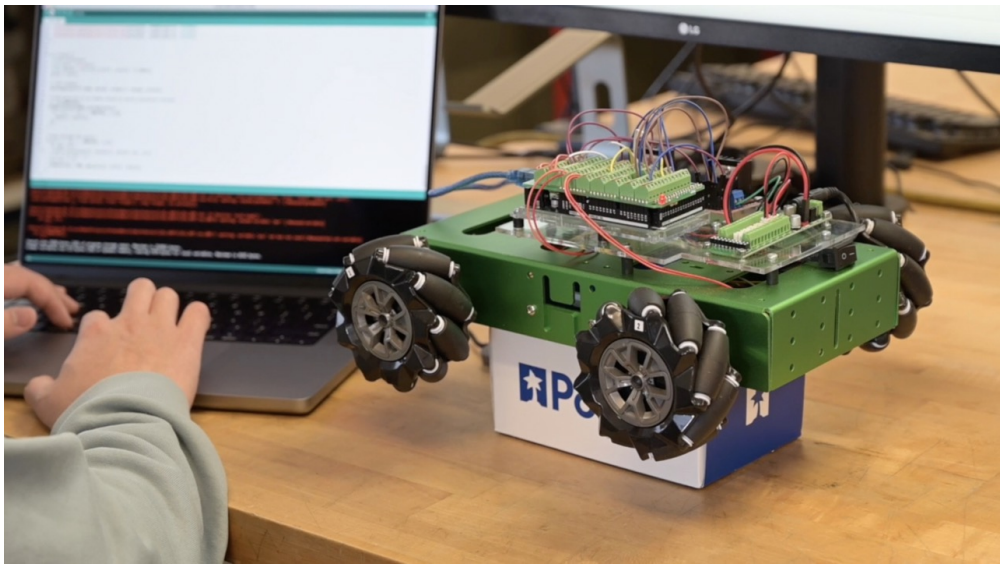


Figure 5.15: Testing motor coordination. Source: photo by the author. Video: Appendix A.3

The new robot is put to the straight-line test immediately after finishing the motor and circuit tests. At this time, the robot can go straight without difficulties; there's still a subtle deviation but that was mainly caused by the imperfect wheel alignment, but the deviation can be ignorable as we don't pursue the industrial performance in this thesis (Figure 5.16).

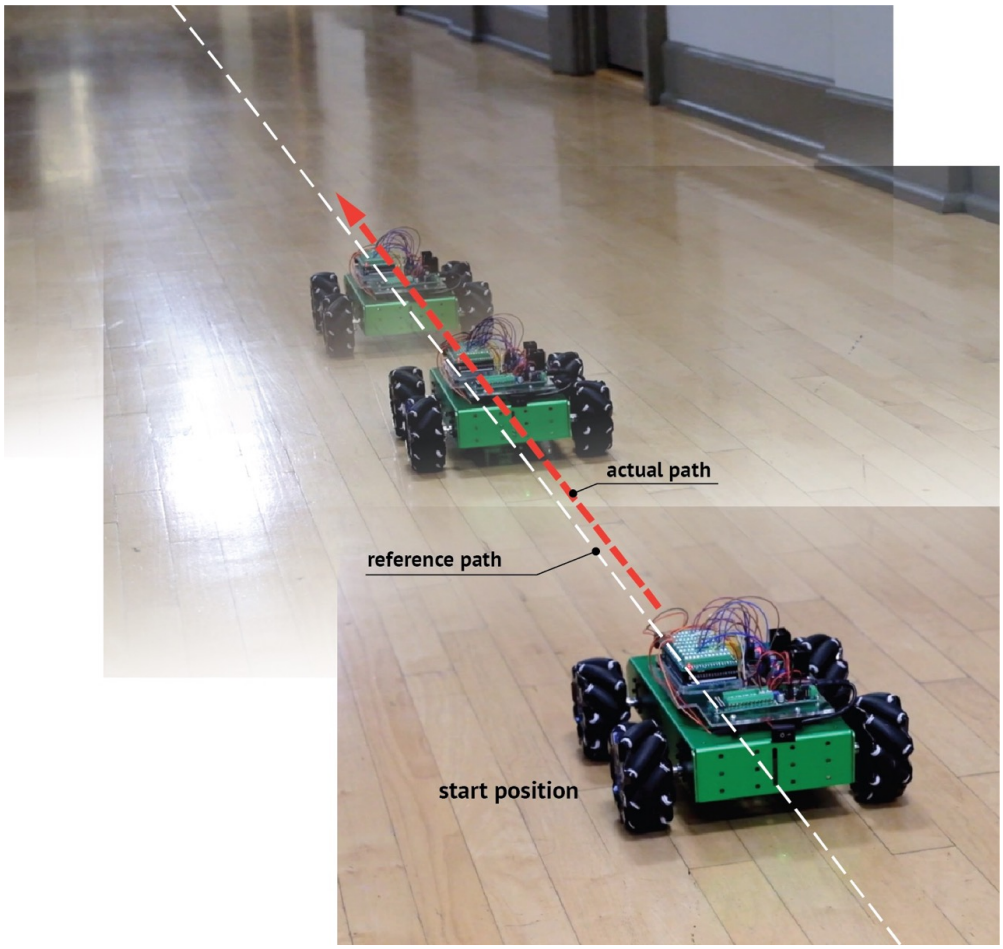


Figure 5.16: Testing motor coordination. Source: diagram by the author. Video: Appendix A.3

So far, the circuit design verification and four-motor coordination tests and basically be considered successful. The final thing is to clean up the robot and organize the wires to give it a tidy look (Figure 5.17 - 5.20).

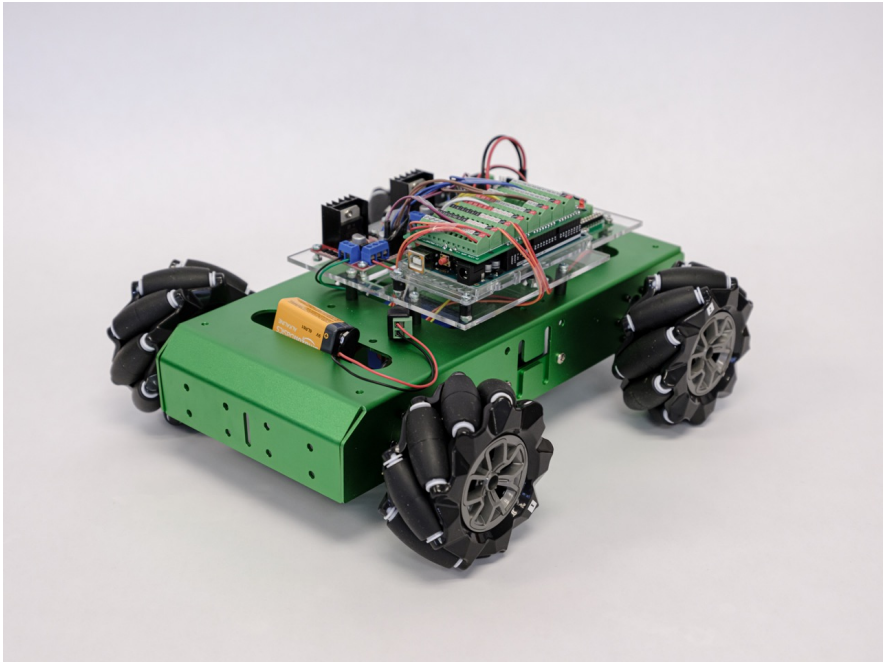


Figure 5.17: Finished moving base, front. Source: photo by the author

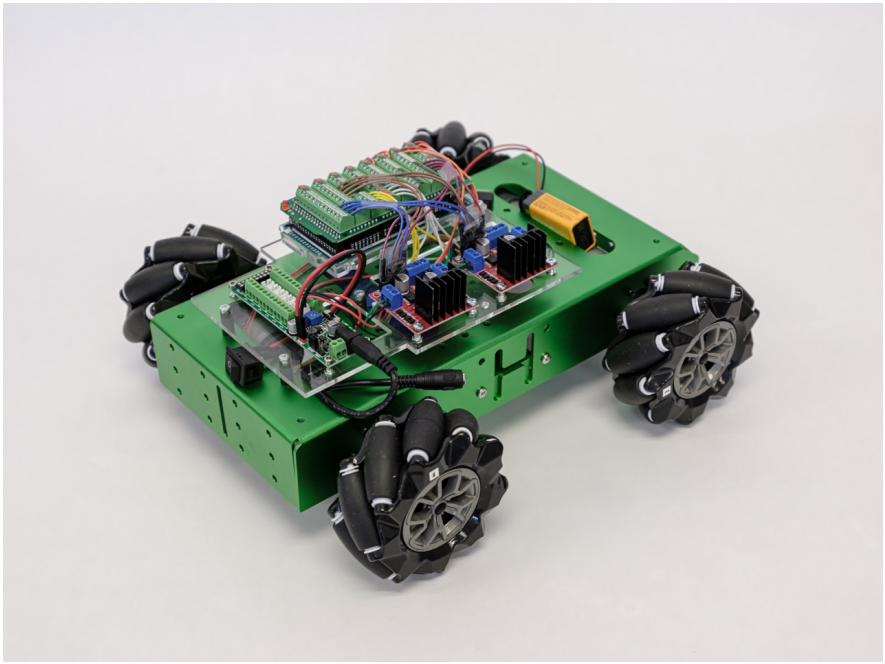


Figure 5.18: Finished moving base, back. Source: photo by the author

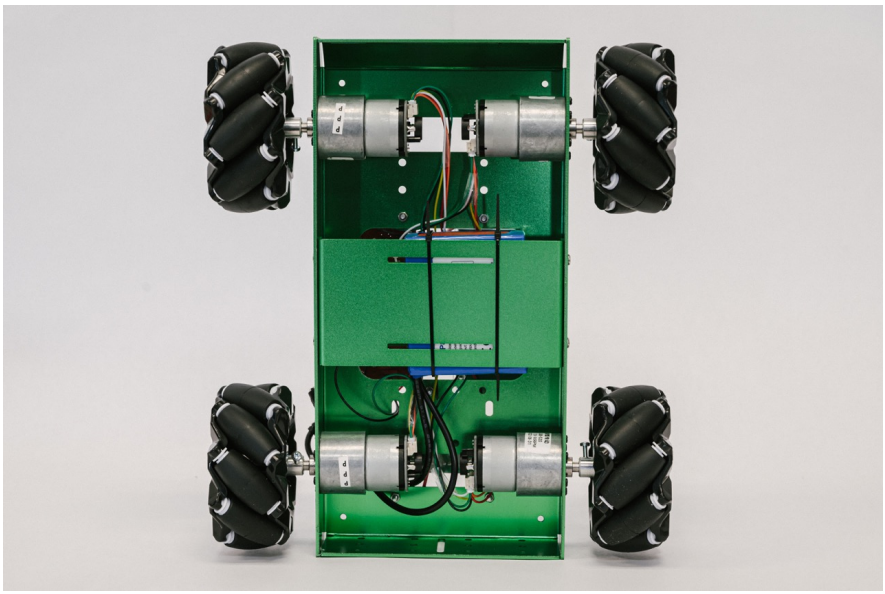


Figure 5.19: Finished moving base, motors and battery. Source: photo by the author

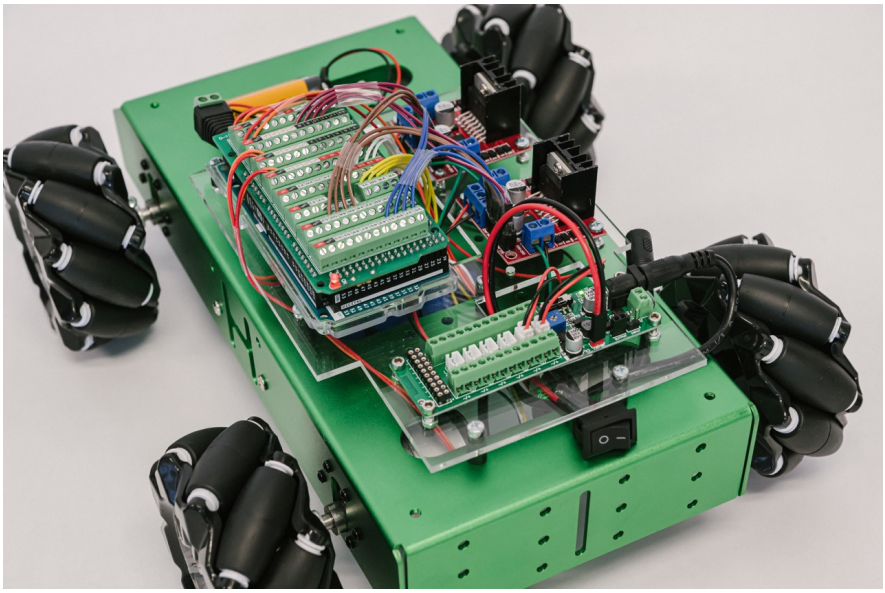


Figure 5.20: Finished moving base, wiring and electronics layout. Source: photo by the author

5.3 Maneuverability test

The reason for choosing 4 Mecanum-wheeled Drive for the robot is due to its superior maneuverability and agility compared to the regular 4WD setup. The 4WD setup can move on limited axes and requires a large turning radius when making turns or changing postures precisely; navigating narrow spaces is challenging. On the contrary, the Mecanum-wheeled robot can move effortlessly in multiple directions, including sideways and diagonally, or turn with various axes without complicated steering mechanisms or extra components like differentials.

The key to achieving such freedom in moving is that it uses special wheels: a set of 45-degree rollers mounted at a 45-degree angle to the main wheel. Installing the wheels correctly on the robot is very important, or the drive will not work. Six major types of maneuvers the Mecanum wheels can provide are forward/backward movement, sideways movement, diagonal movement, turning around, turning on the axis, and cornering (Taheri, Qiao, and Ghaeminezhad 2015); we can easily achieve these types of movements by simply defining each of wheel's directions (Appendix A.3).

Forward and backward

Moving forward and backward is the simplest type of maneuver that very wheeled robots can achieve. To make the robot move forward/backward, we just need to set all wheels spinning in the same direction as the desired moving direction (Figure 5.21, 5.22).

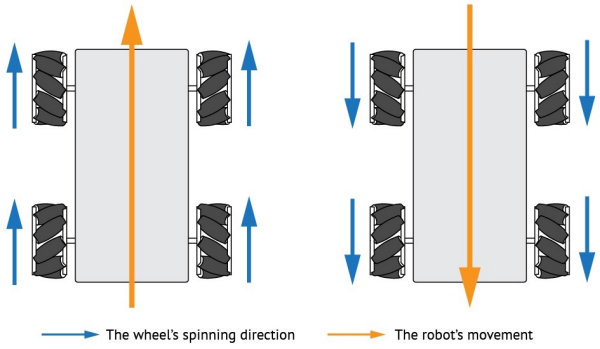


Figure 5.21: Wheels control for forward/backward moving. Source: diagram by the author

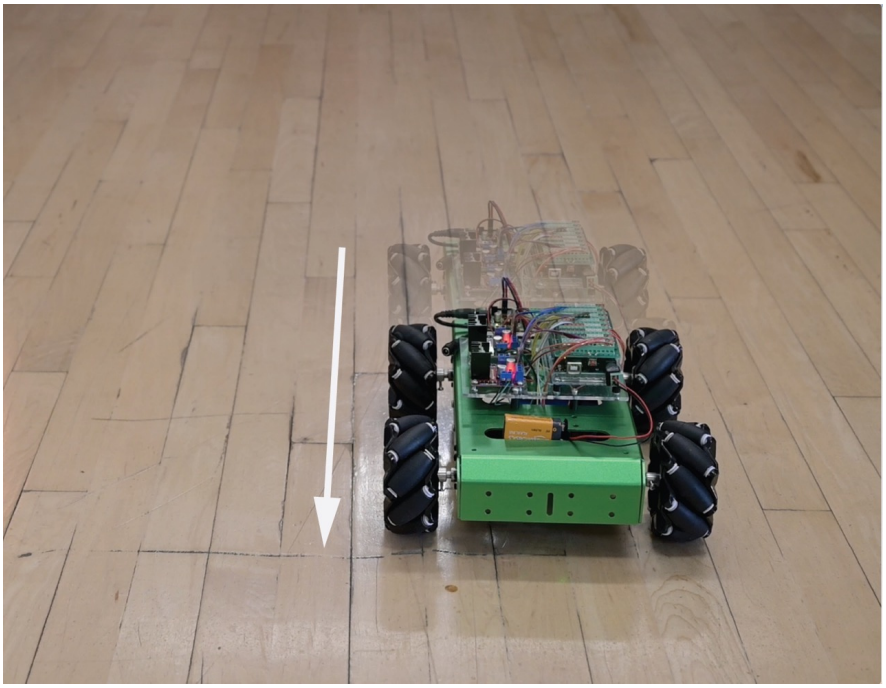


Figure 5.22: Forward moving test. Source: diagram by the author

Sideways

Making the robot move sideways needs independent control on all wheels: the wheels corresponding to the moving direction should spin inward, while the wheels opposite that direction spin outwards (Figure 5.23, 5.24).

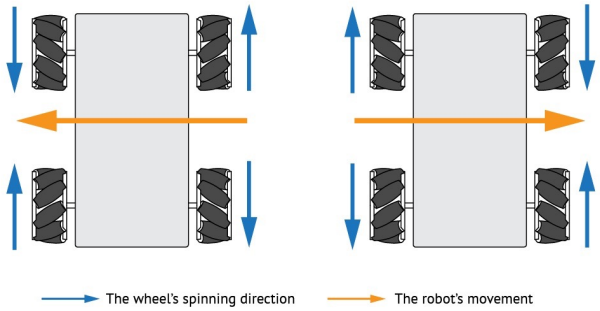


Figure 5.23: Wheels control for sideways moving. Source: diagram by the author

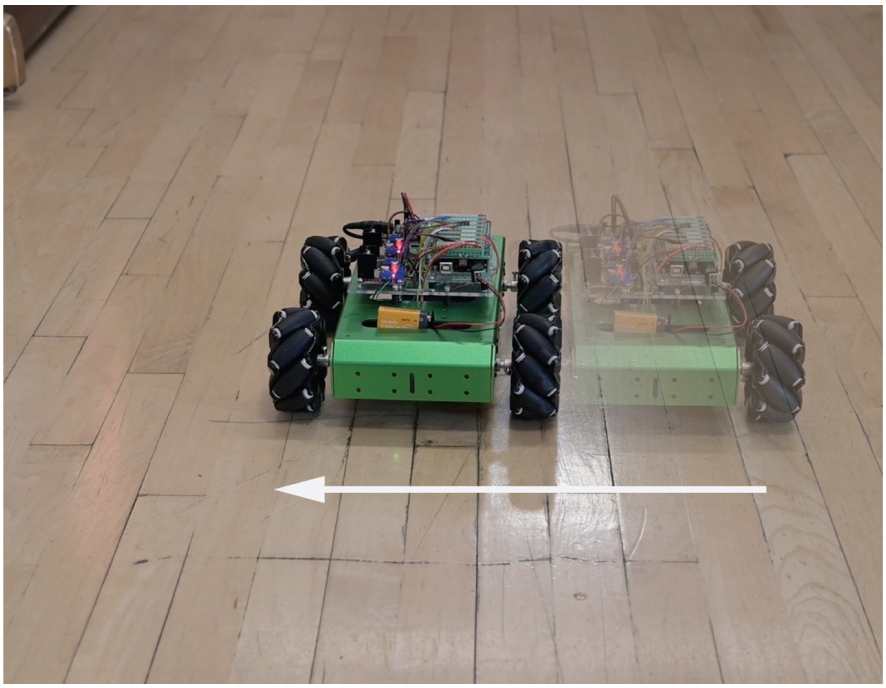


Figure 5.24: Sideway moving test. Source: diagram by the author

Diagonal

To make the robot move diagonally forward/backward need to make two wheels work while making the rest two stops: the wheels beyond the moving axis should spin forward/backward, and the wheels on the moving axis stay still (Figure 5.25, 5.26).

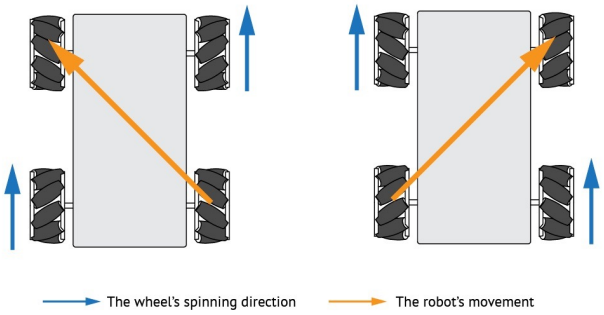


Figure 5.25: Wheels control for diagonal moving. Source: diagram by the author

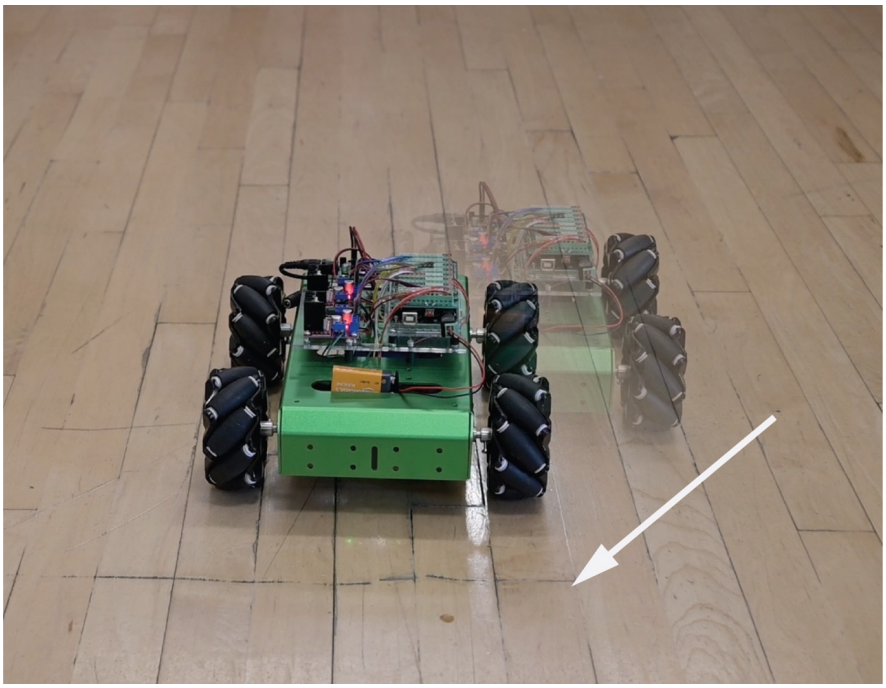


Figure 5.26: Diagonal moving test. Source: diagram by the author

Turning around

Turning around is also a common type of maneuver that almost every robot can achieve. Making the left two wheels spin forward and making the other two spins backward can turn the robot clockwise, and reversing the wheels' behaviors can make the robot turn counterclockwise (Figure 5.27, 5.28).

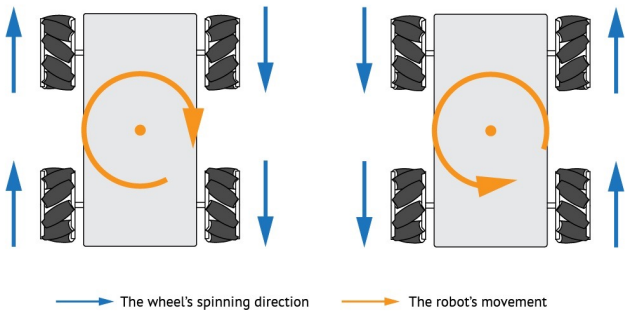


Figure 5.27: Wheels control for turning around. Source: diagram by the author. Source: diagram by the author

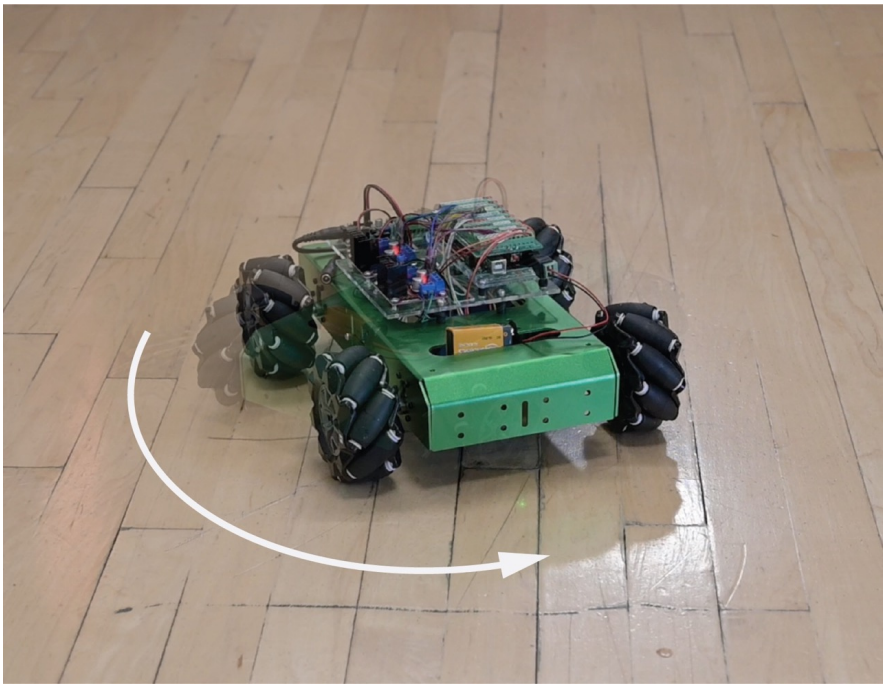


Figure 5.28: Turning around test. Source: diagram by the author. Source: diagram by the author

Cornering

Cornering needs quite more sophisticated control: the turning axis is located in the center of one of the wheels, and only two motors on the same side work with the other two to stay still. The turning center is generally on the rear wheel on the still side based on the moving direction (Figure 5.29, 5.30).

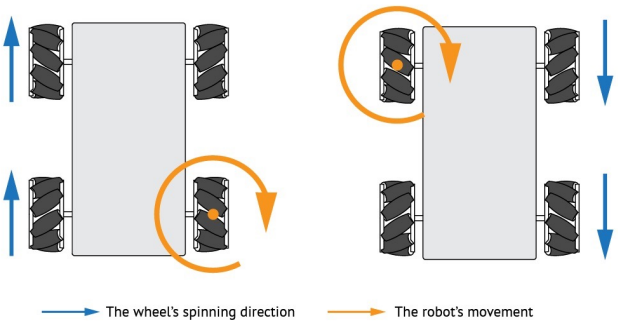


Figure 5.29: Wheels control for cornering. Source: diagram by the author

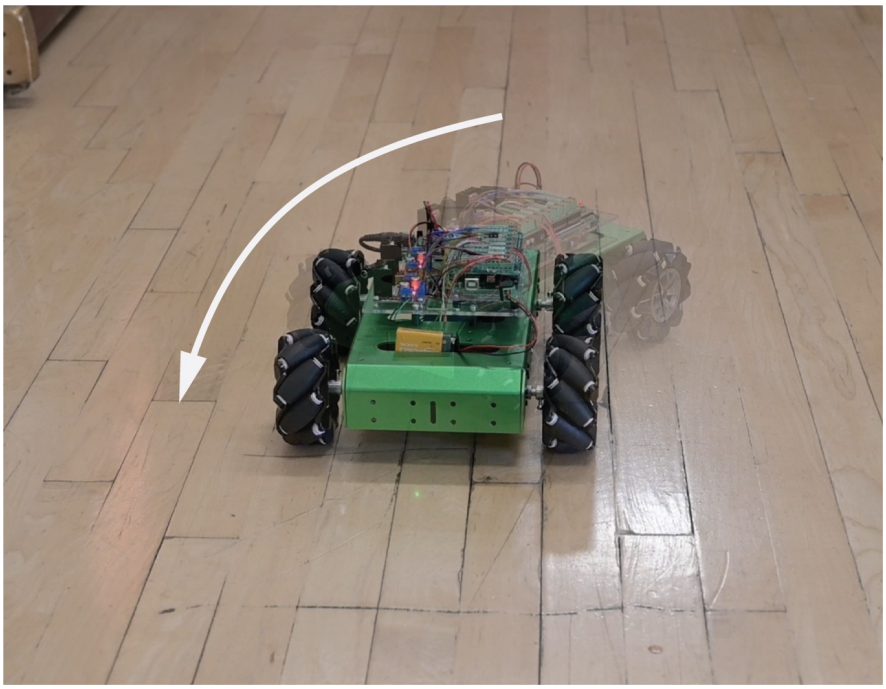


Figure 5.30: Cornering test. Source: diagram by the author

Turn of axis

This method turns based on the robot's shaft axis and can be achieved by making the wheels on one of the axes spin and the others stay still. The turning axis locates in the center of the non-working axis. The robot turns clockwise when making the left wheel spin forward, and the right one spin backward and turns counterclockwise by making the left wheel spin backward and the right one forward (Figure 5.31, 5.32).

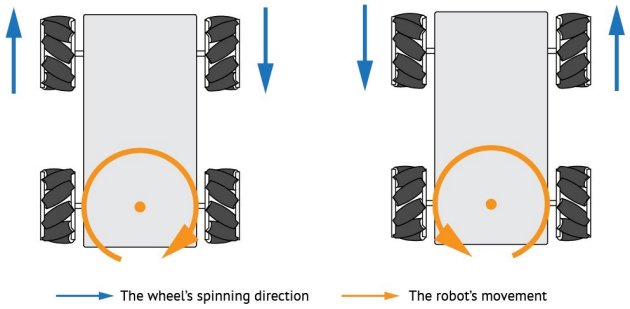


Figure 5.31: Wheels control for turn of axis. Source: diagram by the author

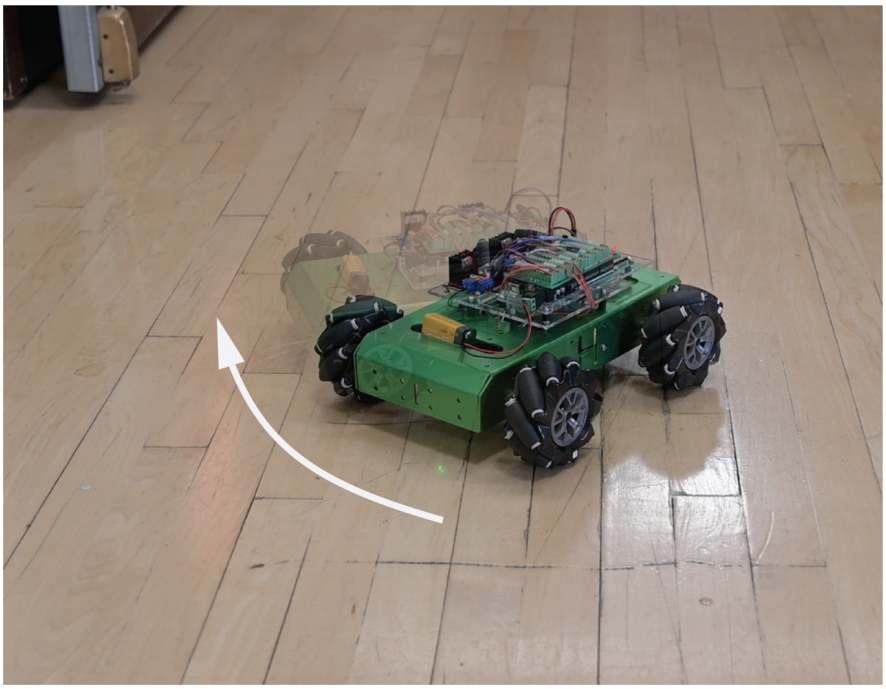


Figure 5.32: Turn of axis test. Source: diagram by the author

These are basic types of maneuvers of the Mecanum-wheeled robot, and with a combination of these methods, adding more sophisticated calculations and control on the wheel's spinning speed, the robot can achieve a high degree of freedom of moving in a narrow space like the prefabricated housing factory in our proposed scenario.

Chapter 6

Conclusion

6.1 Conclusion

The prefabricated housing industry in the United States has been rising both in the number of factories and the sales price of modular housing products since 2014. Compared to the industry in Europe and Japan, the industry in the U.S. aims to solve the shortage of affordable housing by reducing the cost of construction and achieving faster completion. However, the affordable-oriented objective makes the industry slow in automating manufacturing because of insufficient investment in the relative field. Based on this observation, this thesis proposed a practical automation solution for material handling in the prefabricated factory to introduce automation to the industry without interfering with the current factory layout and manufacturing process.

By visiting modular housing factories and analyzing the manufacturing process and material flow helps in building, the thesis defines the automation plan that to use Mecanum-wheeled robots automatically transport carts with medium-sized material between different working stations in the tightly-layout factory to free workers from distracting by the insufficient material and grabbing that material from the storage space to focusing on the module-assembling tasks.

Instead of developing a fully-functional robot for automated material handling, the thesis proposed a system including hardware, software, algorithm, and simulated application scenario for the task, such that the proposal can have the flexibility to adapt to different factory layouts and will not be bind by a specific type of robot. The methods in Chapter 3 help to prove the feasibility of the concepts, and the simulation in Chapter 5 verifies the system through the virtual simulation and the maneuverability of the robot's moving base proposed.

The whole thesis does not aim to provide a fixed answer to the automation of material handling in the prefabricated housing factory but tries to open up a discussion of the possible automation in the industry and hopes to invite more people to contribute to the automation in the prefabricated housing industry to through my efforts.

6.2 Contribution

As mentioned earlier, the project does not aim to develop a robot for the prefabricated housing factory purely but to find the opportunity to introduce automation to the industry. This thesis identifies material handling as a possible entry point to use mobile robots in the factory without disrupting the factory's current operation, hoping this minimal interference way could attract both researchers' and factory owner's interest in reimagining the future of the automated modular housing manufacturing

and taking actions to bring automation-related renovations to the prefabricated housing industry in the United States. Therefore, the contribution of this thesis includes the following:

- **Presenting the site-visit documentation on two selected factories in the northeast U.S. and a manufacturing process study on one of them.**
  The thesis studies the current modular housing manufacturing in the U.S. by analyzing two factories, one produces wood modules and another produces steel ones, in New York and New Hampshire. To find a potential task in the process that can be automated, the thesis takes a deeper look into the manufacturing process and material flow in one of the factories and presents the results with diagrams.
- **Developed an automated material handling scenario based on the factory's current layout and operation.**
  Instead of proposing an imaginative and ambitious automation plan, which will be considered too futuristic and impractical to the factory owners due to the unpredictable high cost, a practical solution may have a higher chance of being accepted by the industry and make forerunners start trying to take it into practice and therefore bring changes to the industry.
- **Introducing an open-source framework that can easily adapt to different factories.**
  The thesis proposed an open-source framework that does not depend on the specific type of robot; developers can use or change the current framework to adapt to different real-world scenarios or modify a commercial mobile based on the framework for experimenting with various tasks.
- **Attracting academic's and industry's attention to this field.** The prefabricated housing industry has a lot of room for automation. This thesis could potentially open up a discussion on automation renovation in the industry and attract attention from academia and industry. With more paper and practice released, the field will grow faster and become more promising.

6.3 Limitations

The limitations of the methods have already been discussed in the corresponding chapter and sections, so here, we'll mainly focus on the limitations from a high-level perspective. The Method and Experiment chapters proposed and verified the mobile robot-assisted material handling system; the thesis so far treats the system as a stand-alone to simplify the problem. However, the system cannot be independent of the environment and the manufacturing process, and we haven't looked into how the robot interacts with the human and integrates into the factory's operation.

- **Human-robot interactions**
  The only human-robot interaction discussed in the thesis is collision avoidance when people show up on the path. But in material handling tasks, there are way more interactions that should be taken into consideration. For example, when the robot is picking up a cart in storage, how could the robot know where the cart is and how heavy the material is, and how does the robot know the robot-cart connection is made and safe? The factory's robot control center cannot decide these but need on-site instructions from the workers. The same scenario also happens when the robot delivers the material to the destination that it does not know which people need that material and where to drop it off.
- **Integration into the current manufacturing process**
  The current setup treats the mobile robot-assisted material handling as a stand-alone system that adapts to the factory's layout; it hasn't been a part of the factory that seamlessly exchanges the information with other parts of the factory. For instance, the robot can be used to cross-validation the material inventory information with the management team by recording the amount of material it delivered and the amount of waste it collected to help the team better

manage the inventory in the daily operation, or the robot can also equip with cameras to help monitor the manufacturing process when it is transporting materials between different stations.

6.4 Future Works

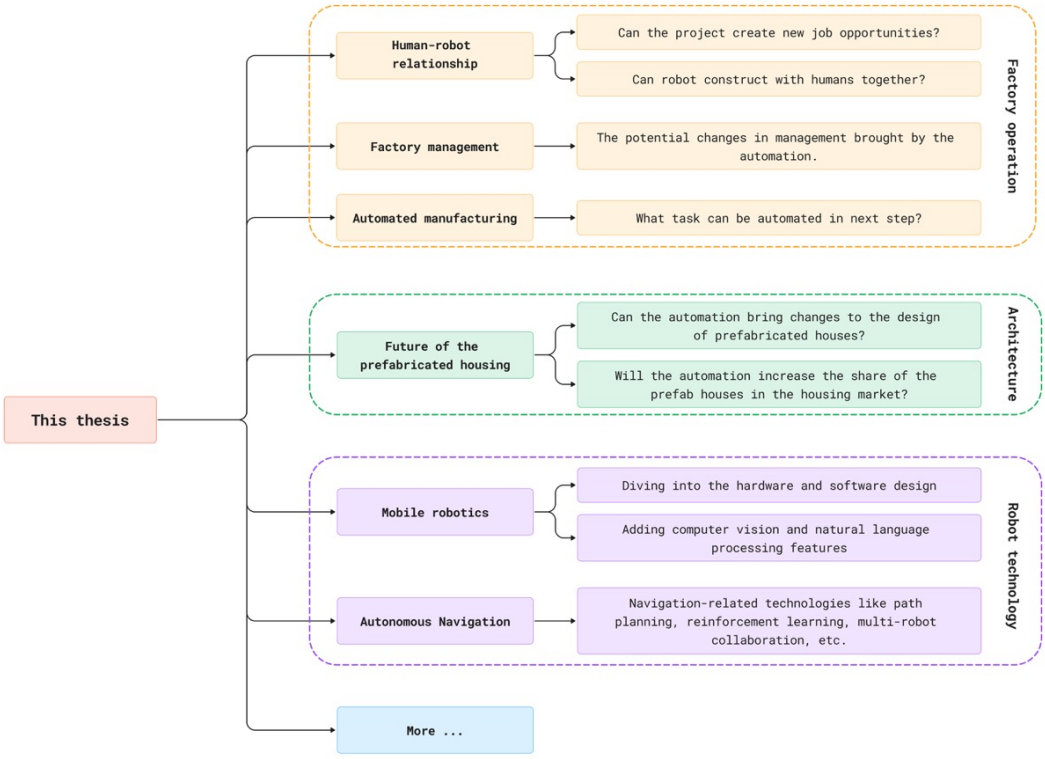


Figure 6.1: The possible future works. Source: diagram by the author

The system proposed in this thesis is a prototype that opens up many opportunities for future developments. There can be three major directions for the next step of this project, including factory operation, architecture, and robot technology (Figure 6.1).

Factory operation

The introduction of automation will definitely influence the current factory's operation. In this branch, possible future works include:

- Exploring the human-robot relationship, like can the introduction of automation create new job opportunities or bring changes to current roles? Or, is it possible that the robot can collaborate with humans in manufacturing rather than working independently?
- Automation brings challenges to the current management of the factory, so what changes does the management team make to ensure efficient and profitable manufacturing?
- Imagine the automated material handling successfully being deployed, then what could be the next task to be automated?

Future of the prefabricated housing

Automated material handling may bring changes to the material for efficient transportation or reduce the cost of manufacturing. In this branch, possible future works include:

- If the materials are further standardized for efficient manufacturing, will the architectural design of the modular housing change to adapt to the new material standard, and will the new design revolutionize the prefabricated houses' functionality and appearance?
- If the cost of prefabricated housing is lowered due to automation, will the share of the prefabricated house in the market increase, and what such change in the share will affect the housing market in the U.S.?

Robot technology

The mobile robot in the current system is still in the preliminary stage that can only work in the theory scenario; it is very far from a real-world application. Therefore, in this branch, the possible future works include:

- Keep developing the robot, making hardware and software meet industry standards. Heavy on-site experiments are also required to help spot possible problems and refine the robot design.
- Another direction is to work on autonomous navigation, which requires quite an intense development of path-planning algorithms, computer vision, reinforcement learning, or other related technologies.

There are more possibilities that can start from this thesis, and the collaboration among talents from different majors with different backgrounds is indispensable for the future development of the prefabricated housing industry.

Appendix A

Videos

A.1 Chapter 3 Hypothesis

- 3.2 Hypothesis scenario - robot picking up a cart: https://vimeo.com/user119251141/hypopickup
- 3.2 Hypothesis scenario - robot delivering the cart: https://vimeo.com/user119251141/hypodeliver

A.2 Chapter 4 Methods

- 4.1.4 Circuit design - wheel measuring: https://vimeo.com/user119251141/wheelmeasure
- 4.1.4 Circuit design - 4-wheel coordination test: https://vimeo.com/user119251141/wheelcoordtest
- 4.3.2 Path finding algorithm - graph test: https://vimeo.com/user119251141/pathfindalggraph
- 4.3.2 Path finding algorithm test: https://vimeo.com/user119251141/pathfindalg
- 4.4.2 Line following calibration: https://vimeo.com/user119251141/linecalib
- 4.4.2 Line following simulation: https://vimeo.com/user119251141/linesim
- 4.5.2 Collision avoidance simulation: https://vimeo.com/user119251141/pdt

A.3 Chapter 5 Experiments

- 5.1.2 Virtual simulation: https://vimeo.com/user119251141/virtualsim
- 5.2.2 Final design - Motors test: https://vimeo.com/user119251141/newrobmotors
- 5.2.2 Final design - straight line test: https://vimeo.com/user119251141/newrobstraight
- 5.3 Final design - maneuverability test: https://vimeo.com/user119251141/finaldemo

Bibliography

Safdie, Moshe (1967). "Habitat '67-towards the development of a building system". In: *PCI Journal* 12.1, pp. 60–66.

Gibb, Alistair GF (1999). *Off-site fabrication: prefabrication, pre-assembly and modularisation*. John Wiley & Sons.

Crowe, James et al. (2005). *PID control: new identification and design methods*. Springer.

Davies, Colin (2005). *The prefabricated home*. Reaktion books.

Miltenburg, John (2005). *Manufacturing strategy: how to formulate and implement a winning plan*. CRC Press.

Bock, Thomas (2007). "Construction robotics". In: *Autonomous Robots* 22, pp. 201–209.

Gfrerrer, Anton (2008). "Geometry and kinematics of the Mecanum wheel". In: *Computer Aided Geometric Design* 25.9, pp. 784–791.

Kang, Jung Won, Bong Sung Kim, and Myung Jin Chung (2008). "Development of omni-directional mobile robots with mecanum wheels assisting the disabled in a factory environment". In: *2008 International Conference on Control, Automation and Systems*. IEEE, pp. 2070–2075.

Smith, Ryan E (2010). *Prefab architecture: A guide to modular design and construction*. John Wiley & Sons.

Joyeux, Sylvain and Jan Albiez (2011). "Robot development: from components to systems". In: *6th National Conference on Control Architectures of Robots*, 15–p.

Lin, ZHONGJIE (2011). "Nakagin capsule tower and the Metabolist movement revisited". In: *Architectural Education* 65, pp. 13–32.

Engin, Mustafa and Dilşad Engin (2012). "Path planning of line follower robot". In: *2012 5th European DSP Education and Research Conference (EDERC)*. IEEE, pp. 1–5.

Russell, Andrew L (2012). "Modularity: An interdisciplinary history of an ordering concept". In: *Information & Culture* 47.3, pp. 257–287.

Skid Row Housing Trust (2013). *A Star Apartment Being Hoisted In Place 1 10 13*. Accessed July 31, 2023. URL: https://www.youtube.com/watch?v=EVfwvYa5iIc.

Oleari, Fabio et al. (2014). "Industrial AGVs: Toward a pervasive diffusion in modern factory warehouses". In: *2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP)*. IEEE, pp. 233–238.

Amelar, Sarah (2015). *Star Apartments – Los Angeles*. Accessed July 31, 2023. URL: https://www.architecturalrecord.com/articles/7997-star-apartments-los-angeles.

Michael Maltzan Architecture (2015). *Star Apartments*. Accessed July 31, 2023. URL: https://www.mmaltzan.com/projects/star-apartments/.

Schneier, Michael, Michael Schneier, and Roger Bostelman (2015). *Literature review of mobile robots for manufacturing*. US Department of Commerce, National Institute of Standards and Technology ...

Taheri, Hamid, Bing Qiao, and Nurallah Ghaeminezhad (2015). "Kinematic model of a four mecanum wheeled mobile robot". In: *International journal of computer applications* 113.3, pp. 6–9.

Ullrich, Günter et al. (2015). "Automated guided vehicle systems". In: *Springer-Verlag Berlin Heidelberg. doi* 10, pp. 978–3.

Almasri, Marwah M, Khaled M Elleithy, and Abrar M Alajlan (2016). "Development of efficient obstacle avoidance and line following mobile robot with the integration of fuzzy logic system in static and dynamic environments". In: *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*. IEEE, pp. 1–6.

Roggema, Rob (2016). "Research by design: Proposition for a methodological approach". In: *Urban science* 1.1, p. 2.

Ben-Ari, Mordechai and Francesco Mondada (2017). *Elements of robotics*. Springer Nature.

Buntrock, Dana (2017). "Prefabricated housing in Japan". In: *Offsite architecture. Constructing the future*.

Everett, Michael F (2017). "Robot designed for socially acceptable navigation". PhD thesis. Massachusetts Institute of Technology, Boston, USA.

Hairstans, Robert and Fausto Sanna (2017). "A Scottish perspective on timber offsite construction". In: *Offsite Archit. Constr. Futur. Taylor & Francis, Routledge*.

Lidelöw, Helena (2017). "Offsite construction in Sweden". In: *Offsite Architecture: Constructing the future*, p. 214.

Richard, Roger-Bruno (2017). "Industrialized building system categorization". In: *Off-Site Architecture: Constructing the Future, 1st ed.; Routledge: Oxfordshire, UK*, pp. 3–20.

Lawrence, Peter and Tony Spybey (2018). *Management and society in Sweden*. Routledge.

Grenzebach (2019). *An intelligent fleet for the Audi A8*. Accessed July 31, 2023. URL: https://www.grenzebach.com/en/company/news-press/detail/an-intelligent-fleet-for-the-audi-a8/.

Manley, Karen and Kristian Widén (2019). "Prefabricated housing firms in Japan and Sweden: Learning from leading countries". In: *Offsite Production and Manufacturing for Innovative Construction—People Process, and Technology*, pp. 399–418.

Fang, Zhihao (2020). "Towards multi-drone autonomous construction via deep reinforcement learning". MA thesis. Carnegie Mellon University, Pittsburgh, USA.

Follini, Camilla et al. (2020). "Bim-integrated collaborative robotics for application in building construction and maintenance". In: *Robotics* 10.1, p. 2.

Lau, Tessa (2020). "Building Dusty Robotics". In: *Technology— Architecture+ Design* 4.1, pp. 14–16.

Razkenari, Mohamad et al. (2020). "Perceptions of offsite construction in the United States: An investigation of current practices". In: *Journal of building engineering* 29, p. 101138.

Sergeev, Angel (2020). *Audi E-Tron GT Enters Production Before The Car's Official Premiere*. Accessed July 31, 2023. URL: https://www.motor1.com/news/459374/audi-e-tron-gt-enters-production/.

Costa Barros, Ítalo Renan da and Tiago Pereira Nascimento (2021). "Robotic mobile fulfillment systems: A survey on recent developments and research opportunities". In: *Robotics and Autonomous Systems* 137, p. 103729.

Karur, Karthik et al. (2021). "A survey of path planning algorithms for mobile robots". In: *Vehicles* 3.3, pp. 448–468.

Madhankumar, S et al. (2021). "Design and Modelling of Autonomous Mobile Robot for Material Handling". In: *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*. Vol. 1. IEEE, pp. 738–742.

Assembly (2022). *Audi Prepares Plan for Future EV Production*. Accessed July 31, 2023. URL: https://www.assemblymag.com/articles/97530-audi-prepares-plan-for-future-ev-production.

ReAC (2022). *Rethinking Automation in Construction*. Accessed July 30, 2023. URL: https://c0delab.github.io/ReAC/.

Scottish Construction Now (2023). *Connect Modular appointed on £1.2bn public sector affordable homes framework*. Accessed July 31, 2023. URL: https://www.scottishconstructionnow.com/

articles/connect-modular-appointed-on-ps12bn-public-sector-affordable-homes-framework.

U.S. Bureau of Labor Statistics (2023). *Industries at a Glance: Construction: NAICS 23*. Accessed July 31, 2023. URL: https://www.bls.gov/iag/tgs/iag23.htm.

U.S. Department of Housing and Urban Development (2023). *HOMEOWNER FACT SHEET*. Accessed July 31, 2023. URL: https://www.hud.gov/program_offices/housing/rmra/mhs/csp/mhcqa/factsheet.

Safdie Architects (n.d.). *Habitat '67'*. Accessed July 31, 2023. URL: https://www.safdiearchitects.com/projects/habitat-67.