



**HAL**  
open science

# Motion Planning and Tracking of a Hyper Redundant Non-holonomic Mobile Dual-arm Manipulator

Yan Wei

► **To cite this version:**

Yan Wei. Motion Planning and Tracking of a Hyper Redundant Non-holonomic Mobile Dual-arm Manipulator. Other. Ecole Centrale de Lille, 2018. English. NNT : 2018ECLI0004 . tel-02057960

**HAL Id: tel-02057960**

**<https://theses.hal.science/tel-02057960>**

Submitted on 5 Mar 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre : 

3	5	5
---	---	---

**CENTRALE LILLE**

**THÈSE**

Présentée en vue  
d'obtenir le grade de

**DOCTEUR**

en

*Spécialité: Automatique, Génie Informatique, Traitement du Signal et Image*

par

**Yan WEI**

DOCTORAL DÉLIVRÉ PAR CENTRALE LILLE

Titre de la thèse:

**Planification et Suivi de Mouvement d'un  
Manipulateur Mobile Non-holonome à deux  
Bras**

**Motion Planning and Tracking of a Non-holonomic  
Mobile Dual-arm Manipulator**

---

Soutenue le 18 juin 2018 devant le jury d'examen :

Président	Youcef MEZOUAR	Professeur, SIGMA Clermont and Institut Pascal
Rapporteur	Daniel SIDOBRE	Maître de conférence HDR, Université Paul Sabatier
Rapporteur	Jingrui ZHANG	Professeur, Beijing Institute of Technology
Membre	Youcef MEZOUAR	Professeur, SIGMA Clermont and Institut Pascal
Membre	Pushparaj PATHAK	Ass. Prof., Indian Institute of Technology Roorkee
Directeur de thèse	Ahmed RAHMANI	Professeur, Ecole Centrale de Lille

Thèse préparée au Centre de Recherche en Informatique Signal et Automatique de Lille  
CRISTAL, UMR CNRS9189 - Centrale Lille  
Ecole Doctorale SPI 072



*Je dédie ce travail  
à tous ceux qui le méritent,*

*à mon cher mari Ang ZHANG,  
à mes chers parents,  
à toute ma famille,*

*à mon directeur bien-aimé,*

*et à mes chère(s) ami(e)s.*



Avant de s'agrandir au dehors, il  
faut s'affermir au dedans.

---

Victor Hugo

Most men appear never to have  
considered what a house is, and  
are actually though needlessly  
poor all their lives because they  
think that they must have such a  
one as their neighbors have.

---

Walden-Henry David Thoreau



**MOTION PLANNING AND TRACKING OF A NON-HOLONOMIC MOBILE DUAL-ARM MANIPULATOR****Abstract**

This thesis focuses on motion planning and tracking of a redundant Mobile Dual-Arm Manipulator System (MDAMS) for personal assistance. Firstly, the modified Denavit-Hartenberg method is used for kinematic modeling of the designed MDAMS. In order to avoid calculating system energy and energy derivatives, Kane's method is used for dynamic modeling. The physical stability is analyzed based on the constructed Kane's model and a controller is proposed using the back-stepping technique. Secondly, motion planning problem including redundancy and coordination solving between the mobile platform and upper manipulator is studied. In particular, an improved MaxiMin NSGA-II algorithm is proposed to design the optimal position-orientation of mobile platform and the optimal configuration of upper manipulator simultaneously given only the initial pose and the end-effectors' desired positions and orientations. Five objective functions are defined including two positioning accuracy criteria, the manipulability criterion and two displacement criteria in task and joint spaces respectively, to optimize the desired pose. In order to achieve the transition from the initial pose to the optimal pose, a direct-connect bidirectional RRT and gradient descent-based motion planning algorithm is proposed. Specifically, a geometric optimization method is designed to guarantee always the optimal and consistent path from the initial pose to the optimal pose. In addition, head forward movements are achieved by calculating a sequence of orientations for the mobile platform based on the planned via-positions of the mobile platform and by using the linear polynomials with parabolic blends interpolation technique, thus indicating the robot's movement intention to increase the quality of robot-human interaction. Then, in order to solve the failure problem of offline motion planning algorithm in dynamic environment, an on-line motion planning algorithm is proposed which includes on-line sensing, collision-test, on-line motion planning and tracking processes. The movements of dynamic obstacles are estimated in real time. In addition, in order to optimize via-poses, a motion planning algorithm based on end-effectors' via-points and multi-objective GA is proposed by optimizing four via-poses-based objective functions simultaneously, which are sums of joint displacements, of end effectors' directional manipulability measures, of end-effectors' positioning errors and of robot-obstacle collision measures. Finally, the motion tracking problem is studied given end-effectors' movements in the task space. The task priority and leader-follower techniques are used to solve the high redundancy and to deal with the coordination between the mobile platform and upper manipulator while tracking the end-effectors' trajectories in real time. Instead of controlling the absolute motion in the world frame, two relative motions are introduced to realize the coordination and cooperation between the mobile platform and upper manipulator. Thus, a triple-trajectory problem is formulated. Moreover, a modulated weighted least norm technique is proposed to achieve the avoidance of joint boundaries.

**Keywords:** mobile manipulator, kane's method, maximin nsga-ii, redundancy solving, motion planning, trajectory tracking.

---



# Acknowledgments

This work was carried out in “Centre de Recherche en Informatique, Signal et Automatique de Lille (CRISTAL)” at Ecole Centrale de Lille (ECLi) with the research team “Méthodes & Outils pour la Conception Intégrée de Système (MOCIS)”.

I would like to express my sincere gratitude to the people who have helped me a lot. Without their encouragement and help, I would not have reached this position and this stage of my life.

First of all, I would like to express my sincere gratitude to my PhD thesis supervisor Prof. Ahmed RAHMANI who influenced me a lot during my studies at Central Lille. He gave me the maximum freedom to search in the robot field. He taught me how to look for new areas of research, how to understand the state of the art quickly, how to write good technical papers, and how to present my ideas effectively.

Besides, I also would like to acknowledge all members of the jury: Mr. Daniel SIDOBRE, Mrs. Jingrui ZHANG, Mr. Youcef Mezouar and Mr. Pushparaj Mani Pathak, for their valuable time to review this thesis. Their invaluable comments and objective criticisms shed a light on my drawbacks and the research road in the future. Meanwhile, I want to thank Prof. Qiang ZHAN from Beihang University who helped me a lot with the research work, the thesis writing and other related discussions during his visit to France.

In addition, I would like to thank all the staff in CRISTAL-MOCIS, Mrs. Geneviève Dauphin Tanguy, Mr. Belkacem OULD BOUAMAMA, Mr. Christophe Sueur, etc. for their valuable discussions and insightful suggestions in team meetings. I also would like to thank three interns Yue CHAI, Yijing QI and Hongbo CAO who have done their research projects with me for their contribution on the virtual prototype validation via MATMAB/SIMULINK and ADAMS co-simulation. A particular acknowledgment has to be made to the visit professors Yunhe MENG and Yongguang YU, and all my PhD colleagues Jing BAI, Youwei DONG, Xing CHU, Wei JIANG, Thanh Binh Do, Wei HU, etc. And I also want to express my great thanks to the staff at Ecole Centrale de Lille, Brigitte FONCEZ, Patrick GALLAIS and Vanessa FLEURY for their kindly help.

Pursuing a P.h.D degree requires not only technical skills but also tremendous amount of stamina and courage. I would like to thank my husband ZHANG Ang for his support and accompany throughout my life abroad. And very often he gave me some valuable suggestions on the research. I wish also to thank my parents and family for sharing their unconditional love with me and giving me the necessary amount of courage required for pursuing my goals. Besides, together with my friends, Xuemei LIU, Tian TIAN, Yihan LIU, Lijie BAI, Wenjuan GU, Siyang DENG, etc. I had a very interesting time in France.

Finally, I would like to thank China Scholarship Council (CSC) for giving me the opportunity to pursue the PhD diplomat at Centrale Lille.

# Table of Contents

<b>Abstract</b>	<b>vii</b>
<b>Acknowledgments</b>	<b>ix</b>
<b>Table of Contents</b>	<b>xi</b>
<b>List of Tables</b>	<b>xv</b>
<b>List of Figures</b>	<b>xvii</b>
<b>Glossary</b>	<b>1</b>
<b>INTRODUCTION</b>	<b>3</b>
<b>1 STATE OF THE ART AND PRELIMINARIES</b>	<b>11</b>
1.1 STATE OF THE ART ON SERVICE ROBOTS . . . . .	11
1.1.1 Developments and Applications of Service Robots . . . . .	12
1.1.2 Humanoid Robots For Personal Assistance . . . . .	14
1.2 LITERATURE SURVEY . . . . .	19
1.2.1 Robotic System Modeling . . . . .	20
1.2.2 Motion Planning Methods . . . . .	21
1.2.3 Motion Tracking and Controllers . . . . .	31
1.3 PRELIMINARIES . . . . .	38
1.3.1 Forward Kinematics Background . . . . .	38
1.3.2 Kane’s Method for Dynamic Modeling . . . . .	39
1.3.3 Kinematic Control of Non-holonomic Mobile Platform . . . . .	41
1.3.4 Non-dominated Sorting Scheme . . . . .	43
1.4 Conclusions . . . . .	43
<b>2 SYSTEM DESCRIPTION AND MODELING</b>	<b>45</b>
2.1 INTRODUCTION . . . . .	45
2.2 SYSTEM MODELING FOR MDAMS . . . . .	46

2.2.1	System Description and Kinematic Modeling . . . . .	46
2.2.2	Lagrangian Formulations and Control . . . . .	56
2.2.3	Dynamic Modeling and Control using Kane's Method . . . . .	66
2.3	CONCLUSIONS . . . . .	73
<b>3</b>	<b>MOTION PLANNING IN STATIC ENVIRONMENTS</b>	<b>75</b>
3.1	INTRODUCTION . . . . .	75
3.2	MAXIMIN-BASED NSGA-II MOTION PLANNING . . . . .	79
3.2.1	Improved MaxiMin NSGA-II Algorithm for <i>Pose</i> Design . . . . .	79
3.2.2	Motion Planning in Static Environments . . . . .	85
3.3	SIMULATION RESULTS . . . . .	92
3.3.1	Task 1: Chair A . . . . .	92
3.3.2	Task 2: Chair D . . . . .	99
3.3.3	Motion Planning and Dynamic Tracking for MDAMS . . . . .	104
3.4	CONCLUSIONS . . . . .	104
<b>4</b>	<b>MOTION PLANNING IN DYNAMIC ENVIRONMENTS</b>	<b>107</b>
4.1	INTRODUCTION . . . . .	107
4.2	MOTION PLANNING AMONG DYNAMIC OBSTACLES . . . . .	109
4.2.1	On-line Motion Planning Algorithm . . . . .	110
4.2.2	Via-points and MOGA-based Motion Planning . . . . .	113
4.3	SIMULATION RESULTS . . . . .	117
4.3.1	On-line Motion Planning for Mobile Platform . . . . .	118
4.3.2	On-line Motion Planning for EE . . . . .	124
4.3.3	Motion Planning Time-consuming Statistics . . . . .	132
4.3.4	On-line Motion Planning for MDAMS . . . . .	132
4.4	CONCLUSIONS . . . . .	135
<b>5</b>	<b>MOTION TRACKING FOR MDAMS</b>	<b>137</b>
5.1	INTRODUCTION . . . . .	137
5.2	IK SOLVING AND JOINT LIMIT AVOIDANCE . . . . .	139
5.2.1	IK Solving . . . . .	139
5.2.2	Joint Limit Avoidance - mWLN . . . . .	140
5.3	DUAL-TRAJECTORY TRACKING . . . . .	141
5.3.1	Task Priority and Relative Motion-based IK Solving . . . . .	141
5.3.2	Semi-decentralized Dual-trajectory Tracking . . . . .	144
5.3.3	Simulation Results . . . . .	147
5.4	TRIPLE-TRAJECTORY TRACKING . . . . .	158
5.4.1	Task Priority and Motion Distribution-based IK Solving . . . . .	159
5.4.2	Semi-decentralized Triple-trajectory Tracking . . . . .	160
5.4.3	Simulation Results . . . . .	160

---

5.5 CONCLUSIONS . . . . .	163
<b>CONCLUSIONS, CONTRIBUTIONS AND FUTURE WORK</b>	<b>167</b>
<b>A APPENDIX</b>	<b>175</b>
A.1 Partial Velocity . . . . .	175
A.1.1 Partial Velocity Table Construction . . . . .	175
A.1.2 Vectors $\mathbb{Z}_i$ ( $i = 1, \dots, 75$ ) and $\mathbb{A}_i$ ( $i = 9, \dots, 75$ ) . . . . .	177
A.2 Matrix $D \in \mathbb{R}^{n \times n}$ . . . . .	179
A.3 Matrix $B \in \mathbb{R}^{n \times 1}$ . . . . .	180
<b>Bibliography</b>	<b>181</b>
<b>Résumé Etendu</b>	<b>201</b>



# List of Tables

- 1.1 Literature survey on robotic motion planning. . . . . 30
- 2.1 MDH Parameters of MDAMS. . . . . 48
- 2.2 Main inertial properties of MDAMS. . . . . 49
- 2.3 Link length and CoM position. . . . . 71
- 3.1 The designed optimal pose  $\theta_{op}$  [*m, deg*]. . . . . 93
- 3.2 The designed optimal pose  $\theta_{op}$  [*m, deg*]. . . . . 101
- 5.1 Control gains. . . . . 147
- 5.2 MDAMS joint limit. . . . . 156
- A.1 CoM vectors of MDAMS. . . . . 175
- A.2 Partial velocities of bodies 1-5. . . . . 175
- A.3 Partial velocities of bodies 6-19. . . . . 176
- A.4 Frame's origin vectors. . . . . 177



# List of Figures

1.1	Service robots in various fields. . . . .	12
1.2	Various types of humanoid robots. . . . .	15
2.1	The structure of the proposed robot MDAMS. . . . .	46
2.2	Modeling of the non-holonomic mobile platform. . . . .	47
2.3	Left: virtual prototype in ADAMS. Right: real robot construction. . . . .	49
2.4	MATLAB/SIMULINK and ADAMS co-simulation procedure. . . . .	58
2.5	Simplified model of robot MDAMS. . . . .	59
2.6	Case A: position tracking results. . . . .	60
2.7	Case A: velocity tracking results. . . . .	61
2.8	Case B: position tracking results. . . . .	62
2.9	Case B: velocity tracking results. . . . .	63
2.10	Comparison of position tracking errors between cases A and B. . . . .	63
2.11	Comparison of input torques between cases A and B. . . . .	64
2.12	Co-simulation shortcuts of virtual prototype in ADAMS. . . . .	65
2.13	Joint trajectory tracking results. . . . .	72
2.14	Input torques errors between Kane’s and E-L methods. . . . .	73
3.1	An example of motion planning problem for MDAMS. . . . .	78
3.2	Flowchart of the improved MaxiMin NSGA-II algorithm. . . . .	83
3.3	An example of off-line path planning. . . . .	86
3.4	MDAMS motion planning flow chart. . . . .	91
3.5	Domestic environment with one table and four chairs. . . . .	92
3.6	Motion planning results to manipulate chair A: Improved MaxiMin NSGA-II. . . . .	94
3.7	Motion planning results to manipulate chair A: combined fitness function-based MOGA. . . . .	95
3.8	Comparison between the improved MaxiMin NSGA-II and MaxiMin NSGA-II algorithms. . . . .	96
3.9	Path planning results without geometric optimization for mobile platform to approach chair A. . . . .	97

3.10	Path planning results with geometric optimization for mobile platform to approach chair A. . . . .	98
3.11	Motion planning results to manipulate chair A. Up: improved MaxiMin NSGA-II. Down: combined fitness function-based MOGA.	100
3.12	Motion planning results to manipulate chair D: Improved MaxiMin NSGA-II. . . . .	102
3.13	Motion planning results to manipulate chair D: combined fitness function-based MOGA. . . . .	103
3.14	Path planning results for mobile platform to approach chair D. . . . .	104
3.15	Motion planning results to manipulate chair D. Up: improved MaxiMin NSGA-II. Down: combined fitness function-based MOGA.	105
3.16	Dynamic simulation results: motion planning and tracking. . . . .	106
4.1	Dynamic obstacle collision space estimation. . . . .	112
4.2	Simplified structure of MDAMS for robot-obstacle collision test. . . . .	116
4.3	First task in virtual dynamic domestic environment A. . . . .	118
4.4	Off-line motion planning results for mobile base. . . . .	119
4.5	On-line motion planning results for mobile platform (a). . . . .	120
4.6	On-line motion planning results for mobile platform (b). . . . .	121
4.7	On-line motion planning results for mobile platform (c). . . . .	122
4.8	On-line motion planning results for mobile platform (d). . . . .	123
4.9	On-line motion planning results for mobile platform (e): Phase 9. . . . .	124
4.10	Second task in environment A. . . . .	125
4.11	Off-line motion planning results for EE. . . . .	126
4.12	On-line motion planning results for EE (a). . . . .	127
4.13	On-line motion planning results for EE (b). . . . .	128
4.14	On-line motion planning results for EE (c). . . . .	129
4.15	On-line motion planning results for EE (d). . . . .	130
4.16	On-line motion planning results for EE (e). . . . .	131
4.17	The time-consuming statistics of motion planning <b>Algorithm 4</b> . . . . .	132
4.18	Virtual dynamic domestic environment B. . . . .	133
4.19	Via-points design for two EEs. . . . .	134
4.20	Via-poses design for robot MDAMS (local view). . . . .	135
5.1	Semi-Decentralized dual-trajectory tracking framework. . . . .	145
5.2	Linear trajectory tracking results. . . . .	149
5.3	EEs' Attitude tracking results. . . . .	150
5.4	Sinusoidal trajectory tracking results. . . . .	150
5.5	Various tasks execution: trajectory tracking results (a). . . . .	151
5.6	Various tasks execution: trajectory tracking results (b). . . . .	152
5.7	Tracking results of EEs and mobile platform. . . . .	154

---

5.8	Object manipulation results. . . . .	155
5.9	Linear trajectory tracking under joint limit. . . . .	157
5.10	Comparison between PD and Back-stepping controllers. . . . .	158
5.11	Triple-trajectory tracking results. . . . .	161
5.12	Triple-trajectory tracking errors. . . . .	162
5.13	Triple trajectory tracking results: 3D view. . . . .	163
5.14	Dual-trajectory tracking: linear trajectory to approach the object. . . . .	164
5.15	Triple-trajectory tracking results: 2D view. . . . .	165



# Glossary

**ADAMS** Automatic dynamic analysis of mechanical systems. 6

**DoF** Degree of freedom. 4

**EE** End-effector. 46

**GA** Genetic algorithm. 26

**IK** Inverse kinematics. 20, 75

**MDAMS** Mobile dual-arm manipulator system. 4, 167

**MDH** Modified Denavit Hartenberg. 7, 9

**NSGA** Non-dominated sorting genetic algorithm. 167

**RBFNN** Radial basis function neural network. 5, 7, 45, 167

**WLN** Weighted least norm. 33



# INTRODUCTION

Since last century, many industrial manipulator systems have been used to do some repetitive tasks with high speed and high precision. However, their task policies and control inputs are normally predefined in a structured environment. In addition, they are usually fixed robotic manipulators with limited workspaces and have no interaction with human beings.

Nowadays, more and more service robots are being introduced for personal assistance, entertainment, surgery, agriculture, military, etc. Different from traditional robotic manipulators in industry, services robots are required to adapt to different environments, to interact with human beings, to make decisions independently, etc. There exist many services robots in the market, for example sweeping robots, entertainment robots, accompany robots, guidance robots, servant robots, etc. which, however, are either single-functional, or inflexible, or non-programmable, or non-interactive. Very often, they are too specialized or not intelligently autonomous, which is not practical for non-expert users. In addition, according to the world population report, aging population is increasing continuously and has become a global phenomenon, and the percentage of people with disabilities increases with people's age. As a result, the demand of the personal assistive devices and health caring is rising. Thus, the study of the multi-functional, completely autonomous and sustainable service robots is still a hot research field.

In order to meet various service demands, personal assistant robots are required to have dexterous manipulability abilities. Humanoid mobile manipulator systems are of great interests due to their human-like elegant appearance, dexterous moving and manipulation abilities over the traditional mobile vehicles and fixed manipulator robotic systems. Therefore, we designed a hyper redun-

duant humanoid non-holonomic mobile dual-arm manipulator system (MDAMS, see Fig. 2.1) and constructed a virtual prototype for virtual validation in ADAMS (see Fig. 2.3). The designed robot MDAMS has up to 56 degrees of freedom (DoFs) with one mobile platform, one moving body and two arm-hand subsystems. It has a wide workspace due to the mobile platform and can execute various tasks due to the moving waist, two redundant arms and two dexterous hands. The designed humanoid is expected to be semi or fully autonomous and can imitate humans with cognitive shock-free human-like behaviors. Specifically, it is required to navigate in standard domestic environments which consist of known and unknown, static and dynamic obstacles including human beings. It should be able to provide assistance tasks like cleaning, washing, serving, sit-down and stand up helping, and to interact with other entities like human beings.

However, implementing such a complex humanoid mobile manipulator system is very challenging. The first challenge comes from the construction of kinematic and dynamic models of the high DoFs system. It requires considering the non-holonomic mobile platform, the moving waist and two arm-hand subsystems at the same time. The second challenge comes from the task policies design and motion planning meeting human expectation. In practical applications, one given task is often composed of multiple subtasks and each subtasks requires perfect motion planning. The motion planning can happen in joint space or in task space. However, due to the high redundancy, the position-orientation of mobile platform and the configuration of upper body manipulator of the designed MDAMS can be quite different for the same task which is very challenging. In fact, human beings always know where to stand and how to use their two arm-hand subsystems to accomplish tasks under multiple constraints. Therefore, the robot is expected to have similar autonomous abilities to conduct human-like activities and complete personal assistant tasks in dynamic and complex environments. The third challenge comes from motion tracking and robot control. There are uncertainties, such as modeling uncertainties, the disturbances and uncertainties from environments and hardware systems to be dealt with for real implementation. Other problems like the coupling between mobile platform and upper body manipulator system, redundancy and singular-

ity problem, collision avoidance, on-line motion planning and control, etc. are all of great difficulty.

This thesis is a part of the collaboration between Prof. A. Rahmani (Centrale Lille) and Prof. Q. Zhan (BUAA) in the LIA 2MCSI project. The humanoid mobile manipulator can be used to complete various home services and for elderly and disabled caring. Meanwhile, multi-robot systems have potential applications in tasks which require cooperation of multiple robots. Under this background, our research team is currently focusing on investigating humanoid mobile manipulator systems and is committed to the control of a group of mobile robots to achieve greater flexibility, robustness and adaptability to accomplish tasks such as the handling of very heavy items, searching and rescuing, and large space security inspections.

## PROBLEM STATEMENT

This thesis is the continuation of [1]. In [1], the author finished the applications and functionalities designs of personal assistant robots, and accomplished system modeling of the designed MDAMS based on Lagrangian formulations. However, the calculations of energy and its partial derivatives become complex as the DoFs increase. Besides, the motion in task space for the robot is assumed to be known in advance. The author proposed a RRT-connect-based inverse kinematic solving algorithm using the pseudo-inverse of Jacobian but without considering the redundancy, singularity, joint limit or dual-arm cooperation problems, not to mention human-like movements. Besides, there are static obstacles like walls and furniture, dynamic obstacles like other robots and human beings, and removable obstacles like doors, chairs and small objects in standard domestic environment. The motion in task space is difficult to know for such a complex robotic system. And very often even though the motion in task space is known, it is still very difficult to design the optimal trajectory in joint space due to the high non-linear property. In addition, the author proposed an adaptive RBFNN controller to cope with the uncertainties and disturbances and achieved object grasping using dexterous hands through force control. However, the cooperative control between two arm-hand subsystems and the coupling problem

between the mobile platform and upper body manipulator system have not been studied. Moreover, the effectiveness of the proposed adaptive RBFNN controller remains to be validated.

To sum up, there is still a long way to go before constructing a long-term intelligent personal assistant robotic system for real applications. Therefore, the main objectives of this thesis are listed in the following.

- Kinematic modeling of the designed robot MDAMS.
- Virtual prototype validation via MATLAB/SIMULINK and ADAMS co-simulation of the adaptive RBF neural network controller.
- Dynamic modeling for the designed robot MDAMS using Kane's method.
- Human-like optimal position-orientation design for the mobile platform and optimal configuration design for the upper body manipulator at the same time under multiple constraints.
- Off-line motion planning of the designed robot MDAMS in static environments with known obstacles.
- On-line motion planning of the designed robot MDAMS in dynamic environments with unknown obstacles including human beings.
- Decoupling and Redundancy solving for the designed robot MDAMS based on task priority notion and least norm solution.
- Trajectory tracking and control of the designed robot MDAMS with joint limit and singularity avoidance.
- Validation of the proposed motion planning and tracking strategies on the virtual prototype in ADAMS via MATLAB/SIMULINK & ADAMS co-simulation.

## STRUCTURE

This thesis is organized into five Chapters. The state of the art on robotic manipulator systems and service robots especially humanoid robotic systems, and a brief introduction to the technical context are given in Chapter 1.

In Chapter 2, the kinematic model is firstly established for the designed MDAMS. The Modified Denavit Hartenberg (MDH) method is used to construct the kinematic model with only four parameters. Compact expressions of the motions of mobile platform and end-effectors are formulated with the unit quaternion being used to represent the orientations of end-effectors. Then, the Lagrangian model and the adaptive RBFNN controller are recalled. Virtual prototype validation via MATLAB/SIMULINK and ADAMS co-simulation is realized to validate the effectiveness of the adaptive RBF NN controller. In order to avoid calculating energy and partial derivatives, the dynamic equations using Kane's method are formulated for MDAMS. The physical tip-over stability is studied based on the constructed Kane's model. A controller is proposed and the control stability is analyzed using the back-stepping technique.

In Chapter 3, the motion planning problem for the designed MDAMS in static environments is investigated. Firstly, an improved MaxiMin-based non-dominated sorting genetic algorithm (NSGA-II) is proposed to design the optimal position-orientation of mobile platform and the optimal configuration of upper manipulator system (pose) for MDAMS at the same time. It speeds up the evolutionary algorithm MaxiMin NSGA-II by adding a MaxiMin sorting process. Five objective functions are defined, i.e. two reaching accuracy criteria, manipulability ability, joint displacement in joint space and end-effectors' displacements in task space. The proposed method searches for the optimal pose by optimizing five objective functions at the same time. In particular, a normalized combined-fitness function is defined and a post decision maker is introduced to settle the preferred solution among the candidates in the Pareto-optimal set. Then, in order to realize the collision-free transition from the initial pose to the designed optimal pose, an off-line motion planning algorithm is designed. In detail, a direct-connect BiRRT and gradient descent-based sample process is proposed which speeds up greatly the sampling phase. And a geometric optimization

method is designed which always guarantees the shortest and consistent path between the initial and optimal goal poses. Besides, the hybrid linear polynomials with blends and fifth-order polynomials interpolations are used for generating the time-specified trajectory. In addition, forward movements are realized by assigning reasonable orientations to the mobile platform based on its planned positions, thus indicating robot's movement intentions to increase the quality of human-robot interaction. Finally, several motion planning tasks are realized to validate the proposed method.

In Chapter 4, in order to solve the failure problem of the off-line algorithm, an on-line collision-free motion planning method is introduced for the designed MDAMS in dynamic environments by extending the off-line motion planning method proposed in Chapter 3. Firstly, the desired motion is designed off-line. Then, on-line sensing, collision test, on-line motion planning and control processes are enabled to cope with the motion planning problem in unknown dynamic environments. Environmental information is sensed on-line and the future movements of unknown obstacles are predicted in real time according to their historical motions. If there is a possible collision predicted, the on-line re-planning process will be activated and the desired motion will be updated immediately. In order to take the via-poses into account, an end-effectors' via-points and multi-objective genetic algorithm (MOGA)-based motion planning algorithm is proposed. Four via-poses-based objective functions are defined to optimize MDAMS's via-poses, i.e. (i) sum of joint displacements, (ii) directional manipulability bypassing each via-point, (iii) via-points' positioning accuracy, and (iv) robot-obstacle collision evaluation. In other words, firstly, end-effectors' via-points are designed; secondly, the proposed end-effectors' via-points and MOGA-based algorithm is used to design the corresponding via-poses; finally, the point-to-point motion planning method is used to link the optimized via-poses. At the end of this Chapter, two virtual domestic environments are constructed, and a number of motion planning simulations are realized to validate the effectiveness of the proposed method.

In Chapter 5, the trajectory tracking problem is investigated based on the task-priority notion, the least norm solving method and the leader-follower technique. Firstly, the task priority and least norm solving-based inverse kinematics solv-

ing technique is introduced. And the modulated weighted least-norm solution method for joint limit avoidance is presented. Secondly, a semi-decentralized control law is proposed with kinematic control for the non-holonomic mobile platform. Instead of controlling the absolute motion of two end-effectors, a relative motion between two end-effectors is defined using lead-follower technique. A compact relative Jacobian is formulated in terms of the individual Jacobians of two arms using the MDH method. Besides, the tracking error matrix in terms of unit quaternion is used to represent the orientation error of end-effector. The object grasping is studied by introducing the contact matrices of two end-effectors. Thirdly, inspired by human behaviors, a new relative motion - position of the right end-effector with respect to the mobile platform of MDAMS - is defined to distribute the movements among the mobile platform and two arms. The corresponding relative Jacobian of the defined relative motion is formulated in terms of the known Jacobians applying the forward kinematics. Then, a motion distribution-based triple-trajectory tracking method is proposed. Finally, several numerical simulations are presented to validate the effectiveness of the proposed tracking methods. The comparison between the PD and back-stepping-based controllers is also presented which proves that the back-stepping-based controller has better performances.

Finally, conclusions, the work under consideration and some ideas ahead are given at the end of the thesis.



# STATE OF THE ART AND PRELIMINARIES

This Chapter firstly gives the state of the art on service robotic systems in Section 1.1. Then, the literature survey on robotic system modeling, motion planning and tracking controllers is presented in Section 1.2. Finally, some preliminaries are given in Section 1.3.

## 1.1 STATE OF THE ART ON SERVICE ROBOTS

In recent decades, service robots are widely used in maintenance, repair, transportation, cleaning, security, rescue, custody and other work. According to the classification of International Federation of Robotics (IFR), service robots are mainly divided into two categories: professional service robots and personal/domestic robots. In particular, the professional service robots include: field robots, professional cleaning robots, medical robots, logistics purposes robots, inspection and maintenance robots, construction robots, underwater robots, defense, rescue and safety applications robots and so on. The personal/domestic robots mainly include assistant robots, entertainment robots, auxiliary robots for the disabled, safety and surveillance robots. With the increasing aging population and percentage of people with disabilities, the demand on personal assistant devices services and health caring increases continuously.



Figure 1.1 – Service robots in various fields.

### 1.1.1 Developments and Applications of Service Robots

Since the very first industrial robot *Unimate* presented in the 50s, robots have been widely used in manufacturing [2]. Recently, more and more service robots are introduced. For example, the logistic robots, robots on building sites, harvest robots, domestic assistant robots, medical robots, entertainment toys, underwater robots, military robots, space manipulators [3–5], aerial vehicles, rescue robots [6] and so on.

Fig. 1.1 shows some services robots that are in use or under study. Multi-mobile robots system in Fig. 1.1 (a) is used for logistics [7] by *Amazon* with high efficiency. *HRP-2* [8] in Fig. 1.1 (d) is a Japanese humanoid robotic system

which can be used as a personal assistant. *Intuitive Surgical* designed a robot system *Da Vinci* in Fig. 1.1 (e) controlled by a surgeon from a console for prostatectomies, cardiac valve repair and gynecologic surgical procedures. The bi-manual underwater humanoid robot *OceanOne* [9] in Fig. 1.1 (g) allows human to explore the depths of the oceans with an unprecedented ability. The tracked military robot *Talon* in Fig. 1.1 (h) proposed by *Foster-miller* is designed to protect war fighters and first respond against explosive threats. *Robonaut* [4, 5] in Fig. 1.1 (i) is a humanoid robotic system designed by *NASA* to study how to replace human beings to work and explore in outer space.

In fact, if we enter "service robots" in *Google Images*, a lot of service robots can be found. For instance, servant robots in restaurant, public guidance robots, vacuum-cleaning robot *Roomba* and military robots of *iRobot*, *Turtlebot* of *Willow Garage*, *DARwIn-OP* of *Robotics*. There are also robotic systems like 3D printed robots, multi-legged robotic dogs and ghost legged unmanned ground vehicles of *Boston Dynamics* and so forth.

Many scientific researchers are dedicated to service robotics. Korpela [10] introduced a framework for valve turning using an aerial vehicle endowed with dual multi-degrees of freedom (DoFs) manipulators. Srinivasa [11] designed an autonomous multi-robotic system to play the role of a robotic busboy. Zhang [12] studied a dual-arm robotic system for bolt and nut mating using the 3D dynamic simulation tool ADAMS. Doumanoglou [13] introduced a dual-manipulator system to autonomously recognize and unfold articles of clothing. Bertacchini [14] designed a shopping assistant robotic system to perform social-like interactions. The robot communicates with the customer based on retail's database and affects the customers' buying decisions by recognizing their emotions. Khatib [9] presented a robotic avatar for oceanic discovery. In addition, many mechanical systems have been widely used by people experiencing mobility disability or upper limb impairment (ULI), such as the electric wheelchairs and prosthetic devices. Lee [15] introduced a robot programming method by developing a passive exoskeleton as a master device for a master-slave tele-operation system to program dual-arm industrial robots. Miyoshi [16] designed a tele-operation dual-arm robot manipulator system for wheelchair *R-SaDaCo*. It can be operated using users' natural gestures. A coalescence/separation mechanism and a remote

control mechanism are designed and implemented.

### 1.1.2 Humanoid Robots For Personal Assistance

Among various robotic manipulator systems in manufacturing and service market, the humanoid mobile manipulator systems have attracted increasing attention due to their human-like elegant appearance, dexterous moving and manipulation abilities. They can be classified into two categories: legged humanoid robot and humanoid robot with a mobile platform, according to the type of their lower body. The former would mainly play its role in non-flat situations like stairs, while the latter proves its conveniences in flat situations. The humanoid mobile manipulator systems can operate in an extended space due to the mobility of their lower body. They can mimic human beings with human accepted motions due to the upper human-like arm-hand manipulator system with dexterous end-effectors like grippers or multi-fingered hands. In the past decades, many humanoid robots have been introduced (see Fig. 1.2).

The two-legged (biped) humanoid robot *AMIO* (2006) [17] in Fig. 1.2 (a) is created by *KAIST A.I. & Media Lab*<sup>1</sup>, Korea, for personal accompany. It has one head, two arms, two 3-fingered-hands and two legs.

The humanoid robot *Atlas* (2013) [18] in Fig. 1.2 (b) designed by *Boston Dynamics* company<sup>2</sup>, USA, is also a biped robotic system with two arms and a torso to achieve whole-body mobile manipulation, which greatly expands its workspace.

The 130cm tall biped humanoid robotic system *ASIMO* (2010) in Fig. 1.2 (c) is designed by *Honda*<sup>3</sup>, Japan. It has the ability to recognize moving objects, postures, gestures, its surrounding environment, sounds and faces, which enables it to interact with humans. However, its manipulation abilities have not been well-developed.

The annual international robotics competition *RoboCup* [19] which is founded to promote robotics and AI research attracts many researchers each year. A group of biped humanoid robots are designed to form a soccer team and cooperate

---

<sup>1</sup><http://mind.kaist.ac.kr/robotics.php>.

<sup>2</sup><https://www.bostondynamics.com/atlas>.

<sup>3</sup><http://asimo.honda.com/>.

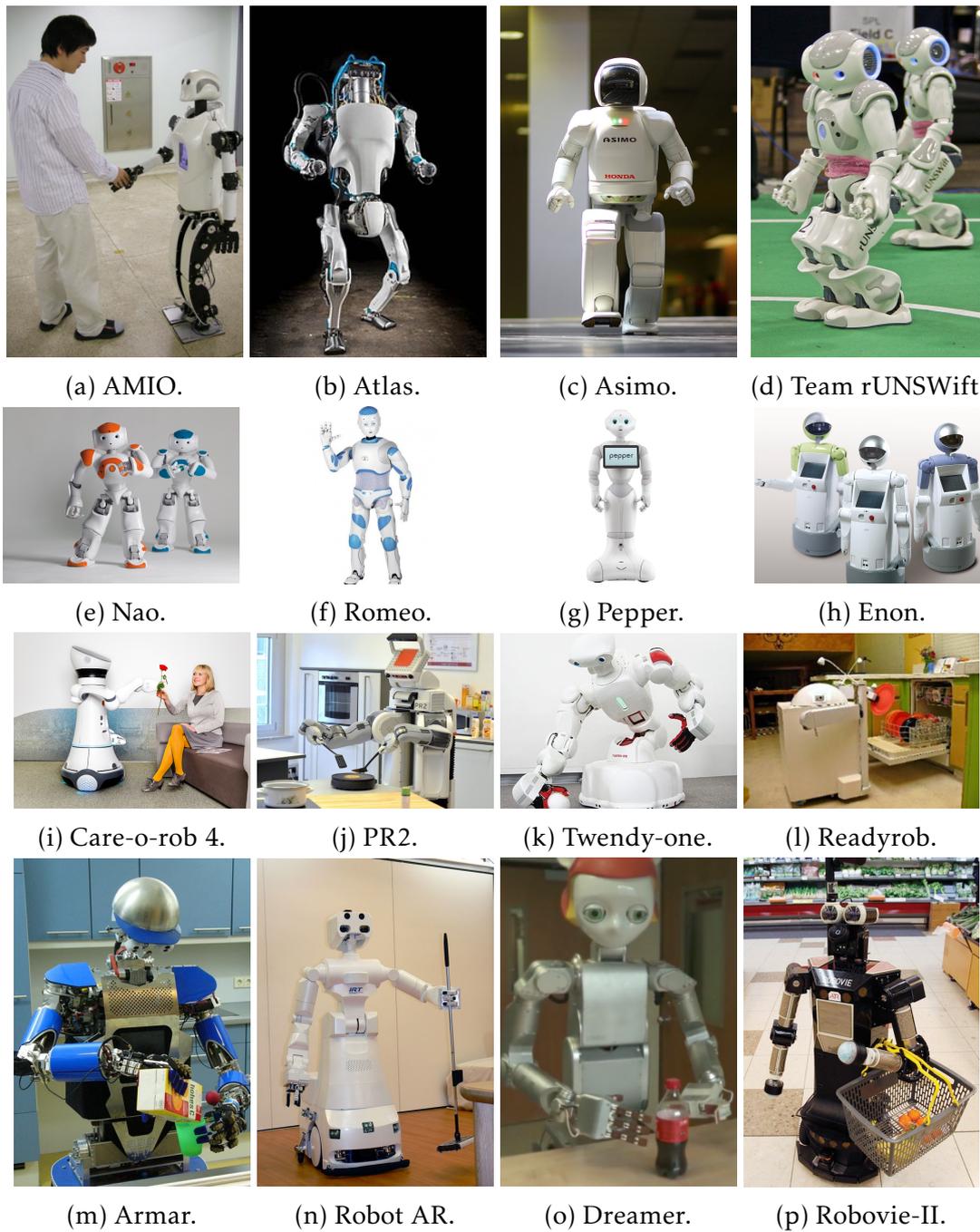


Figure 1.2 – Various types of humanoid robots.

together to play with another robotic team (e.g. team *rUNSWift* in *RoboCup* 2010 in Fig. 1.2 (d)).

*SoftBank Robotics (formerly Aldebaran Robotics)*, Japan, designed three humanoid robots: *NAO* (2006) [20], *Romeo* (2012) [21] and *Pepper* (2014) (see Figs. 1.2 (e), (f) and (g))<sup>4</sup>. In particular, *NAO* is a 58cm tall biped humanoid robotic system with 14-25-DoFs. However, it is designed for entertainment accompany, which can only offer simple assistance. The 140cm tall biped humanoid robotic system *Romeo* is a research platform for assisting elderly people and people who experience mobility disabilities. However, it is only a mechanical base for research purpose and its manipulability ability remains to be explored.

One of the challenges for designing and controlling such a biped robotic system comes from the physical instability and redundancy introduced by two legs. Thus, the humanoid robot with one mobile platform attracts people's attention. The 120cm tall humanoid robotic system *Pepper* is such a robot which is designed for day-to-day accompany. It has one mobile platform, two arms and one communication screen, and is capable of identifying user's emotion. However, it is only able to entertain and accompany people with embedded multimedia systems and to execute tasks with limited manipulability abilities.

The humanoid robotic system *Enon* (1999-2011)<sup>5</sup> in Fig. 1.2 (h) developed by *Fujitsu Frontech Limited* and *Fujitsu Laboratories Ltd.*, Japan, has one mobile platform, two arms and a screen. Its applications include providing guidance, transporting small objects, escorting guests and security patrolling. But it is mainly for navigation applications with very limited manipulation abilities.

In order to improve the manipulation abilities, many humanoid robots with high DoFs upper body have been constructed. The 33-DoFs robotic system *AMI* (2001) [22] which is another robot created by *A.I. & Media Lab*, Korea, consists of one mobile platform, one head, two arms and two three-fingered hands. It is designed for house cleaning, pick-place tasks, communication, etc.

The humanoid *Care-O-bot 3* (2011) [23] created by *Fraunhofer*<sup>6</sup>, is designed to assist people at home. It has one multi-directional mobile platform and two dexterous 7-DoFs arms. It can be used for entertainment, communication and emergency support. *Care-O-bot 4* (2015) in Fig. 1.2 (i) is a developed version

---

<sup>4</sup><https://www.softbankrobotics.com/emea/en/robots>.

<sup>5</sup><http://www.fujitsu.com/global/about/resources/news/press-releases/>.

<sup>6</sup><https://www.care-o-bot.de/en/care-o-bot-3.html>.

of *Care-O-bot 3* with 31-DoFs for communication, surveillance and personal assistance like hand over. If the user needs to do some things out of home, the newer version can be commanded in distance. In addition, the new version has different modes of interaction and is more agile than the previous one.

The humanoid robot *PR2* (2010) in Fig. 1.2 (j) [24] is a robotic research and development platform designed by *Willow Garage*<sup>7</sup>, USA, which consists of one omni-directional mobile platform, one 3-DoFs wrist, two 4-DoFs arms and two 1-DoF grippers. It is an open-platform which is designed to make it easy to develop and test robotics applications and technologies. For instance, Scholz [25] studied the application of PR2 for cart pushing in a typical office environment.

The human-symbiotic-robot *Twendy-one* (2009) [26] in Fig. 1.2 (k) designed by Sugano Laboratory<sup>8</sup>, WASEDA University, Japan, is an accurate robot with two 7-DoFs arms, two 13-DoFs fingered-hands and one 1-DoF trunk. It offers personal assistance like sit-up support, caretaker-transferring, refrigerator-using, tray-handling and so on.

*Readybot* (2008) in Fig. 1.2 (l) designed by *Readybot Robot Challenge* is a research toy which would be for cleaning in the kitchen. It can move slowly and can adjust its height according to the task. It can currently perform 30-40% of daily chores.

*Armar-III* (2006) [27, 28] in Fig. 1.2 (m) is a 43-DoFs humanoid robot which is designed by *Karlsruhe Institute of Technology*<sup>9</sup> for domestic helping like dish washing. It is a mobile platform with two dexterous arm-hand subsystems. Its grasping and helping abilities remain to be explored.

The 32-DoFs humanoid robotic system *AR* (2008) in Fig. 1.2 (n) designed by *IRT-Tokyo University*, Japan, is able to recognize the environment, to create 3D motions, to evaluate and assess itself after having finished a task. Unfortunately, it can only do some simple household chores.

There exist also other humanoid robots like *Dreamer* in Fig. 1.2 (o) proposed by *UT Austin*, robot *Robovie-II* in Fig. 1.2 (p) in supermarket, etc. Apart from the robotic systems introduced above, many research groups around the world

<sup>7</sup><http://www.willowgarage.com/pages/pr2/overview>.

<sup>8</sup>[http://twendyone.com/demo\\_e.html](http://twendyone.com/demo_e.html).

<sup>9</sup><http://h2t.anthropomatik.kit.edu/english/397.php>.

are working on personal assistant robots.

For instance, in order to face the aging society in Japan, *Jouhou System Kougaku Lab. (JSKL)*<sup>10</sup> of *Tokyo University* designs many robotic platforms, such as humanoid robots, daily life assistance systems and personal mobility systems, to forward Information and Robot Technology Research Initiative (IRT) innovations and image scenarios[29].

The French research group *LAAS*<sup>11</sup>, in collaboration with *VERIMAG*, studies the robust and reliable architecture of autonomous robots through by combining the BIP component-based design and the modular architecture development methods. Sidobre [30] gives a global view on human–robot interaction. Zhao and Sidobre [31] work on the motion planning problem for service manipulator robots.

*German Aerospace Center (DLR) - Institute of Robotics and Mechatronics*<sup>12</sup> proposes five humanoid robotic platforms: *C-Runner*, *David*, *Rollin' Justin*, *Space-Justin* and *TORO*, for research topics of robust locomotion on two legs, mimicking human two arms, autonomous dexterous mobile manipulation in human environments, tele-manipulation, dealing with biped movement and dynamics, etc. Butterfass [32] introduced 2nd generation of multisensory hand design at DLR. Ott [33] designed a humanoid dual-arm system based on the modular *DLR-III* and the *DLR-Hand-II* for studying dexterous two handed manipulation. In particular, two arms and two hands are combined with a 3-DoFs movable torso and a visual system to form a complete humanoid upper body.

In addition to *OceanOne*, *Stanford Robotics Lab* also investigates the modeling and control of humanoid robotic system to move like human beings<sup>13</sup>. *Stanford Computational Vision and Geometry Lab (CVGL)* has developed a robot prototype *Jackrabbot* to study the autonomous movements among humans.

*Chinese Institute of Mechatronic Engineering of Tsinghua University* has developed many platforms for humanoid and bionic robots research<sup>14</sup>. The biped humanoid robot *THBIP-I* is capable of accomplishing forward, backward and

---

<sup>10</sup><http://www.jsk.t.u-tokyo.ac.jp/research/irt/index.html>.

<sup>11</sup><https://www.laas.fr/public/>.

<sup>12</sup><http://www.dlr.de/rm/en/desktopdefault.aspx/tabid-11370/>.

<sup>13</sup><https://cs.stanford.edu/group/manips/projects.html>.

<sup>14</sup><http://ime.me.tsinghua.edu.cn/researchView.asp?id=22>.

lateral movements on the flat floor without cables. It can continuously walk up and down stairs up to 75mm in 0.35m steps. Its upper body is able to serve water, exercise tai chi, nod and shake hands. The micro biped humanoid *THBIP-II* is 18 cm high and weighs 18 kg with 18-DoFs. It is capable of moving forward and backward, turning around, sidewalks, walking up and down stairs, kicking and shooting, waving, grasping heavy objects, etc. Many efforts have been also done on biped running robots, bionic muscles drive biped robots, single and quadruped bionic robots, soccer robots, anti-strong external disturbance biped robots, robotic powered thigh prostheses, passive robotic dancer, bionic human-robot interactive arm band, etc.

We can see that the service robotic systems have attracted continuous attention and good results have been obtained. However, only a few simple personal assistant robots, such as the vacuum cleaning robots, guidance robots and some restaurant robot servants, can be found in real commercial settings. Most robotic systems introduced for personal assistance are still in the research stage. The reason is that they still lack the autonomous ability of adapting to complex and unknown dynamic environments including human-robot interaction. Thus, the development and investigation of autonomous, multi-functional health care and personal assistant robots for busy, elderly, disabled or impaired people are still an active area. Under this background, a non-holonomic redundant Mobile Dual-Arm Manipulator System (MDAMS) is designed by our group for personal assistance. However, the introduction of a non-holonomic mobile platform and a redundant upper manipulator system increases the system modeling, motion planning and control challenge.

## 1.2 LITERATURE SURVEY

In this Section, a brief literature survey on system modeling, motion planning and control of robotic systems is given.

## 1.2.1 Robotic System Modeling

A robotic system is a mechanical system which consist of a group of rigid bodies connected by certain joints. A concise and precise mathematical model (kinematic and dynamic) is fundamental [34–36] to study a robotic system.

### 1.2.1.1 Kinematic Modeling

The kinematics of a robotic system describes the relationship between the motion in joint space and the resulting motion of the rigid bodies in task space that form the robotic system. Forward kinematics relate the motion from joint space to task space, while the inverse kinematics (IK) relate the motion from task space to joint space. There exist various representations of the position and orientation of a rigid body in space. In particular, many methods like Z-Y-X Euler angles, X-Y-Z fixed angles, angle axis, unit quaternions, etc., have been introduced to represent the orientation of mechanic systems [37, 38]. The advantages and disadvantages of different parameterization methods become evident as the number of DoFs of robotic systems increases. The geometric Denavit-Hartenberg-based representation is widely used in robotics with only four parameters and simplifies greatly the kinematic modeling.

### 1.2.1.2 Dynamic Modeling

The dynamics of a robot manipulator system describes the relationship between the resulting motion and the applied forces and torques. There exist many dynamic modeling methods for mechanical robotic systems, such as Newton-Euler's equations, Lagrangian formulations, Kane's method, and so forth. Among these methods, the most classical one is the model based on *Lagrangian formulations* [39, 40]. For instance, Yamamoto [39] used Lagrangian formulations to model a wheeled mobile manipulator system. Korayem [40] applied Lagrange equations to model a non-holonomic mobile manipulator which has one spatial arm and one mobile platform with two driving wheels independently driven by two actuators. As a result, 7-DoFs dynamic equations are formulated. Baigzadehnoe [41] used the universal approximation properties of fuzzy logic systems to estimate unknown system dynamics based on Lagrangian model. However, due to the

second order differential item in the Lagrange model, the calculation amount increases rapidly as the number of DoFs increases.

Kane's [42] method was first introduced for modeling the astronomical systems. It has attracted constant attention because of its advantages over Lagrangian formulations. Kane [43] applied this method to improve formulation and solution computational efficiency of robotic system's motion equations. Huston [44] extended Kane's equations to multi-body mechanical systems. Purushotham [45] used Kane's method to dynamically model a manipulator arm. Ke [46] presented a detailed dynamic modeling of a manipulator that mounted on a remotely operated underwater vehicle. Isenberg [47] employed Kane's equations to dynamically model system that subjects to Pfaffian constraints (rolling without sliding).

In addition, there are other methods like *Lie group*-based method and *Bond graph* methodology. Park [48, 49] presented two formulations of robot dynamics based on *Lie Group*. Compared with Lagrange formulations, it has less calculation amount, more unified form of models and definite physical meaning. Li [50] applied Lie group theory for the space manipulator system's modeling through forward and backward recursive process. *Bond graph methodology* [51] is a graphical modeling language for modeling multi-physical systems using a unique representation. The multi-physical systems can be random combination of electrical, mechanical, magnetic, fluid, chemical and thermodynamic systems. Many works have been done using Bond graph [52, 53]. It defeats the traditional modeling methods especially in multi-physical systems.

### 1.2.2 Motion Planning Methods

The environments for traditional industrial manipulator systems are normally constructed to the needs of robots. And their trajectories are usually designed in advance and fixed. On the contrary, the service robots are required to work in non artificially constructed surroundings. Thus, they should be able to plan motion strategies and take their own decisions. Motion planning consists of geometric path design and time-specified trajectory generation (**Definition 1.1**) under multiple physical and environmental constraints.

**Definition 1.1 (Path and trajectory [54])** *A path is the locus of points in joint or task space. A path is a pure geometric description of motion. A trajectory is the path on which a time law is specified.*

Motion planning in robotics can happen in 2-dimensional (2D)  $(x, y, \phi)$ /3D  $(x, y, z, \alpha, \beta, \gamma)$  space (mobile vehicles or free-flying robots), or for multi-DoFs articulated robotic systems  $(\theta_1, \dots, \theta_n)$ , fixed or mobile manipulators). Obstacle avoidance is the major aspect of motion planning problem. In the past few decades, many motion planning methods have been introduced [55–68]. For example, visibility graphs linking obstacles' vertices ( $O(n^2 \log n)$ ), voronoi diagrams with maximum clearance ( $O(n \log n)$ ), cell decomposition ( $O(n \log n)$ ) which produce a roadmap with vertex and edges [57, 58], A\* [59] and D\* [60], potential fields methods [61], sampling-based methods like probabilistic roadmaps (PRM) [62] and Rapidly-exploring Random Trees (RRTs) [63, 69], heuristic method like genetic algorithms (GAs) [64] and particle swarm optimization (PSO) [65–68], etc.

### 1.2.2.1 Redundant Manipulators

As the number of DoFs increases, obstacle avoidance is no longer the only issue. The challenge of redundancy solving under multiple constraints occurs. Apart from the required DoFs to realize target task, the additional DoFs of redundant manipulator system can be used to achieve objectives like obstacle, joint limits and singularity avoidance, human-like movements, and so on. In addition, many systems with manipulators mounted on a mobile platform have been introduced to expand the workspace and manipulation ability. As a result, the coordination between the mobile platform and the upper mounted manipulator system becomes another challenge for motion planning.

**Direct Searching Methods** Sampling-based planners represented by RRTs are the leading family of algorithms for higher dimensional motion planning. Tsai [70] proposed a *Bi-direction RRTs in Configuration Time space (CT-RRTs)* planner for a dual-arm mobile manipulator in joint space. Saut [71] designed a RRT-based

planning framework to plan motion for a dual-arm-hand robot during a pick-and-place task which requires object exchange between the hands. Vahrenkamp [28] used RRT-based algorithms to build a tree of reachable and collision-free configurations for humanoid robot ARMAR-III. Kim [72] presented an occlusion-free and collision-free path planning algorithm for a 7DoFs manipulator with RRTs. However, the random planners are inconsistent and non-optimal, though RRT\* is asymptotically optimal.

Rodriguez [73] proposed a motion planner PRM with Obstacles (PRMwO) which searches for dual-arm system's path and determines, if necessary, the obstacles to be removed. It uses a precedence graph to look for a proper sequence of actions and for the distribution of tasks among the robot arms (can be extended to multi-arms). Wang [67] employed the direct kinematic equations in conjunction with PSO technique for trajectory planning of a redundant manipulator which is mounted on a free-floating space robot.

**IK-based Motion Planning** Though many good results exist, they mainly focus on collision-free motion planning while tracking the given path in task space of end-effector. And the resulting motion of robot links in task space and other constraints are not considered. Many practical tasks impose constraints on the motion of robotic end-effectors such as carrying a cup of coffee. Therefore, IK-based motion planning methods have attracted people's attention. Many methods have been developed to solve the IK problem. The difficulty is that the mapping from task space to joint space is non-linear for redundant robots. Hence, many motion planning methods solve the redundancy problem by introducing additional optimization objectives.

For example, the gradient-based method in [74] minimizes a defined scalar function in terms of obstacle and manipulability measure and projects it onto the Jacobian null-space. The resolved motion planning method in [75] solves multiple robotic tasks with priorities for redundant robots using pseudo-inverse Jacobian. Lewis [76] formulated an IK solving-based trajectory planning algorithm for two robots cooperating to perform an assembly task. It treats two robots as a single redundant system and derives two Jacobians to relate two end-effectors. Secondary goals such as collision and joint limit avoidance are

defined to be projected onto the Jacobian null-space. Mohri [77] proposed a motion planning method for two cooperative manipulators to hold a common object. Potential functions are defined to avoid collisions among robotic links and obstacles. Korayem [40] presented an advanced methodology for trajectory planning of wheeled mobile manipulators in obstructed environments by defining potential functions. It models the obstacles and mobile manipulator as ellipsoids. Berenson [78] proposed a Constrained BiDirectional RRT (CBiRRT2) motion planner for pose-constrained manipulation planning. A Task Space Regions (TSRs) representation is introduced. It is suitable for redundant robots and takes several constraints into account, including constraints for the end-effector. In addition, a formal proof for the probabilistic completeness of RRT-based algorithms is presented [79]. Cohen [24] designed a heuristic search and graph-based motion planning algorithm for dual-arm object manipulation using robot *PR2*. The upper arm roll joint is chosen as the redundant DoF. Then, it computes the collision-free paths for moving the object from a start pose to a goal pose while maintaining the initial roll and pitch of the object with two arms. Huang [65] introduced a PSO method for IK solving of a 7-DoFs robotic manipulator. Jamisola [80] introduced a method to maximize the utilization of the redundant DoFs of a given robot, within the framework of task prioritization. Tsai [81] introduced an IK-based planner which combines PSO algorithm and real coded genetic algorithm (RGA) to determine the configuration of a 7-DoFs manipulator.

It can be seen that the IK-based methods are designed to satisfy the primary tracking tasks, as well as secondary goals such as collision, obstacle and joint limit avoidance and world frame orientation constraints. However, they require inverse Jacobian calculation and can only solve the optimization problems expressed by continuous objective functions.

**Mobile Platform and Upper Manipulator Coordination** On the other hand, only mobile platform's mobility or manipulator manipulability has been considered in the above mentioned works. Tchon [82] designed an extended Jacobian pseudo-inverse method for mobile manipulators by introducing new output functions. Yamamoto [39] investigated coordination of mobile manipulator

systems by treating the mobile vehicle and the manipulator as two different dynamic entities. It has only one manipulator installed atop the mobile platform. Korayem [83] presented an efficient IK algorithm based on the two point boundary value problem (TPBVP) solution. It determines the maximum load-carrying capacity and the corresponding joint optimal path for kinematically redundant manipulators. Besides, it supposes that the path of end-effector is given and redundant (R) and nonredundant (NR) joint variables are assigned manually in advance. It uses the NR joints to achieve path tracking and the R joints to avoid obstacles. Harada [84] planned a sequence of base positions for the mobile dual-arm manipulator system which performs pick-and-place tasks. It formulates a quadratic programming (QP) problem to solve the IK problem. The robot uses selectively either right or left hand to pick up an object to minimize the base's movement. Ren [85] designed an algorithm to optimize mobile platform's position of a mobile manipulator to meet the requirements of local painting tasks. The globally near-optimal mobile platform's position is selected by sorting all feasible positions according to the evaluation criteria. The manipulator maneuvers to finish the painting task independently after the mobile platform arrives at a given position.

**Reasoning Methods** In addition, many analytical reasoning methods have been also presented for motion planning of robotic systems. Yahya [86] solve the motion planning problem for a planar hyper-redundant manipulator system through a geometric approach. It sets angles between the adjacent links be the same. But it is difficult to geometrically solve the planning problem for complex robots consisting of not only planar joints. Yan [87] proposed an analytical IK solving method based on the dual-arm-angle parameterization. It uses two orthogonal vectors to define two absolute reference planes and derives an absolute reference elbow attitude matrix. Lamperti [88] introduced swivel elbow angle using muscular activation. Kim [89] formulated a mathematical representation for characterizing human arm motions. It characterizes robot motion by elbow elevation angle which is determined using the position and orientation of human hands with the approximation tool-Response Surface Method (RSM).

**Heuristic Methods** It can be seen that many good works have been done on robotic motion planning. However, sometimes it is difficult to formulate proper objective functions, hence they are no longer competent to solve motion planning problem. In order to cope with the shortcomings of the above-mentioned direct, indirect and reasoning motion planning methods, many evolutionary meta-heuristic methods have been introduced. Genetic algorithm (GA) is one of the most used heuristic methods.

Inspired by the process of natural selection and natural genetics, Goldberg [64] firstly introduced the GA method. Different from conventional techniques, GA is blind by applying notions of gene, individuals, parent, offspring, child, population and chromosomes. It consists of bio-inspired operators such as mutation, crossover and selection. A standard GA has a genetic representation of the solution domain and a set of objective functions (or cost function) to evaluate the candidate solutions.

GA has the following attractive features: no constraints on the cost function's continuity, no need of inverse Jacobian computation, capable of optimizing (maximizing and minimizing at the same time) multiple objectives, etc. Many GA-based motion planning methods have been proposed for redundant robotic manipulator systems [90–95]. Parker and Goldberg [90] used GA to plan motion for a redundant robotic manipulator system. The robot's end-effector is leaded from an initial position to a final position by optimizing a combined fitness function. The combined fitness function is in terms of two objective functions: positioning error of end-effector and joint displacement. Zhao [91] solved the path planning problem for a mobile manipulator system based on GA. A combined fitness function is defined in terms of displacements of mobile platform and joint angles. The combined fitness function in [93] is in terms of joint displacement and end-effector's reaching error. Bakdi [95] designed an optimal GA-PCHIP-based off-line path planning method for a mobile manipulator by optimizing also a combined fitness function. The combined fitness function is in terms of navigation length, path-obstacles intersection and total deviation of mobile platform. However, the reproduction processes require specific consideration of various intersection cases with the obstacles.

As the increasing demand of the complicated tasks, single optimization

method seems incompetent. Therefore, hybrid optimization methods attract researchers' attention by compensating the inconveniences and combining the advantages of various optimization methods [96–98].

Most of the above-mentioned GAs search for the solution in one direction by optimizing a combined fitness function. In order to search for the solution in multiple directions, Deb [99] introduced the non-dominated and crowding distance sorting schemes to GAs. Pires [92] replaced the crowding distance sorting scheme with MaxiMin sorting scheme. The main idea behind the MaxiMin sorting scheme is to select the solutions in order to decrease the large gap areas existing in the already selected population.

Many of them have been used for high DoFs systems in complex environments under multiple constraints. But very few of them solve on-line motion planning problem and take human-like motion planning into account. In fact, the additional DoFs can be used to make the robot move more human-like, thus increasing the quality of human-robot interaction.

### 1.2.2.2 On-line Motion Planning

Autonomous navigation and manipulator abilities of service robots are increasingly demanding. But most of the above mentioned methods mainly focus on the motion planning in static environments. The off-line planned motion will be no longer realizable and applicable when the environments are time-varying. Therefore, many works have been done on motion planning [57, 58, 100–123] in real time for robotic systems.

Primatesta [100] presented a trajectory planning algorithm based on Informed-RRT\* for mobile robot navigation in crowded environments with moving people. Instead of computing the entire path in real time, it only evaluates the validity of the current path and partially repairs the path, thus often resulting in non-optimal paths. A RRT\*-based motion planning algorithm consisting of *reconnect* and *regrow* processes was introduced in [101] for mobile robots in dynamic environments. However, the RRT\* method which needs to sample throughout the task space between the initial and goal positions is time-consuming for long distance planning. Another disadvantage is that the random sampling process

will generate different paths for the same task. It treats the mobile robots as a point. Meng [102] introduced a BiRRT algorithm with pruning and re-planning processes of the global sampling tree to design paths for the unmanned aerial vehicle (UAV). Choi [104] proposed a two-level obstacle avoidance algorithm by introducing a multi-phase problem for an UAV system to avoid multiple obstacles with the minimal effort. The motion planning method proposed by Greiff [105] needs to find two feasible paths around an inflated obstacle by finding the projection points around the obstacles. And the process will become complex in a crowded and dynamic environment. Mercy [106] presented an optimization-based motion planning method based on separating hyper plane theorem and spline-based technique to generate motions for the non-holonomic vehicles. Mcleod [107] used Bezier curve to connect two via-segments of generated path in real time for mobile vehicles. However, for non-holonomic mobile platform there will be too many orientation changes. On the other hand, in a crowded environment, spline-based trajectory generation may cause unnecessary obstacle collision.

We can see that all the above techniques are for collision-free motion planning for mobile vehicles or free-flying robots. Although some of them can be applied to robots with high DoFs, their effectiveness will decrease, and they are incompetent to solve multiple constraints problems. In order to solve this problem for robots, Lee [57] established an algorithm that provides an on-line trajectory generation for a dual-arm robotic system by introducing a virtual roadmap. Berg [58] proposed an algorithm to solve on-line motion planning for any robot type in configuration space with any dimension based on a roadmap. The roadmap is constructed for the static part of the environment, then only the dynamic part is dealt with on-line. A two-level search is performed to design the shortest path. On the local level, trajectories on single edges of the roadmap are found using a depth-first search on an implicit grid in state-time space. On the global level, these local trajectories are coordinated using an A\*-search to find a global trajectory to the goal configuration. Pajak [108] proposed a motion planning method for a mobile surgery assistant. Only the knowledge of the end-effector's path is needed, and the reference trajectory of the mobile platform is not needed. It is based on a penalty function approach and a redundancy

resolution at the acceleration level. The motion of the mobile manipulator is planned in order to maximize the manipulability measure, thus to avoid manipulator singularities. Akli [109] proposed a random profile approach (RPA)-based motion planning strategy for a mobile manipulator RobuTER/ULM to approach the target point by implementing the manipulability measure. Yang [110] introduced a configuration re-planning or adaptation method for a fixed manipulator in a dynamic environment. However, it requires Jacobian computation between the relative distance space and the joint space and it demands that the desired configuration must be known in real time.

In order to avoid dynamic obstacles during manipulation, Chen [111] designed an on-line joint velocity correction term to avoid obstacles by adjusting the manipulator's joint trajectory. Xin [112] proposed a path planning algorithm for redundant robot arms in dynamic environments by introducing an escape velocity and projecting it onto the null space of Jacobian. An obstacle avoidance gain which depends on one minimum distance is defined to reduce computation burden in such a way to modify the IK homogeneous solution. The on-line obstacle avoidance method in [113] used the distance calculation and discrete detection for robot arms, but it is difficult to be applied to humanoid mobile manipulators due to the demand of the obstacle identification and the dependence on robot's structural complexity.

Nguyen [114] presented a heuristic planning method for a multi-DoFs humanoid robot's arm by employing the sampling-based RRT\* algorithm directly in task space in hierarchical fashion. However, it only considers the end-effector and elbow as two hierarchical control points. In fact, as the number of control points increases, the hierarchical motion planning method will become much more complicated. Vannoy [115] introduced an on-line adaptive motion planning algorithm by generating a population of trajectories based on the fitness evaluation for the mobile manipulators.

Very important works have been done by Trautman [116, 117]. They study the safe navigation of a mobile robot through crowds of dynamic agents with uncertain trajectories. In particular, It applies the Interacting Gaussian Processes (IGP) to estimate the behavior of a crowd with the environment being structured. There are also motion planning methods for robots in dynamic environments

Table 1.1 – Literature survey on robotic motion planning.

	Mobile ve- hicles	Fixed manipulator	Mobile manipula- tor	Coordination
-2014	[55, 56, 58, 62, 63, 89]	[57, 65, 71, 73, 77, 81, 86, 87, 90, 108, 111, 124]	[1, 28, 39, 40, 83, 115, 123, 125]	[40], [115]
2015		[110]	[67, 84, 109]	[67, 84], [109]
2016	[107, 119, 126]	[96, 97, 100, 112, 114, 127]	[128–130]	[128], [130]
2017	[59, 103–106, 118, 120]		[131]	
2018		[113]		
	Direct	Indirect	On-line	Reasoning
-2014	[28, 62, 63, 70, 71, 73, 90]	[1, 65, 75, 76, 80, 111, 132]	[1, 57, 58, 108, 111], [115]	[86, 87]
2015	[67]		[110], [109]	
2016	[97, 100, 114]	[112, 129]	[107, 112, 119, 126, 129], [130]	[89, 126]
2017	[59, 95, 98, 101, 102, 104, 105]	[101]	[101, 103, 118, 120]	[131]
2018			[113]	

with stochastic obstacles [118, 119] or motion planning based on probability theory [120]. Chiang [119] worked on the motion planning for mobile robots in environments with stochastic dynamics and uncertainty. The Monte Carlo simulation technique is used to predict stochastic obstacle motion and handle sensor uncertainty. In addition, the motion planning methods will fail if not moving the obstacles away, for instance in order to go from one room to another, door opening is unavoidable. Levihn [121] studied the situations where the obstacles are removable. Although they are of great interest but are out of this thesis's scope.

To summarize, good solutions for solving practical robotic motion planning

problems exist. They can be roughly divided into search-based planning algorithms represented by A\* algorithms, sampling-based planning algorithms represented by RRT, and heuristic-based planning algorithms represented by GAs. They usually design the collision-free motion from an initial configuration to a target configuration which is supposed to be known. Very often, an optimization problem of certain user-defined objective functions, e.g. path length, reaching accuracy, energy consuming, carrying capacity, obstacles avoidance, etc., is formed under multiple constraints. Table 1.1 summarizes the motion planning work mentioned above in the past few decades. To the author's knowledge, only few of them investigate the on-line motion planning, mobile platform and end-effectors coordination at the same time [109, 115]. It means that the coordination solving and motion planning to produce natural and human-like behavior for redundant mobile manipulator systems and the on-line motion planning in dynamic environments are still open problems. Therefore, this thesis focuses on the human-like motion planning for personal assistant robot MDAMS.

### 1.2.3 Motion Tracking and Controllers

The traditional PID controller still plays an important role in the control of robot systems in cooperation with the newly brought methods, such as relative motion control [133], force-position control [134], impedance control [135], task-priority method [136], etc. In addition, many advanced methods based on sliding mode [129], Fuzzy logic, Neural Network (NN), Chaos, and Expert experience have been introduced [137]. Besides, in order to take advantages of multiple control techniques, some hybrid controllers [66] have been also investigated, like adaptive fuzzy control [138], robust adaptive fuzzy control [139], and robust adaptive NN control [140], etc.

#### 1.2.3.1 Relative Motion Control

For a multi-manipulator system, many motion determination and control methods have been presented by introducing the notion of relative motion [80]. *Master-Slave* (MS) [141] and *Leader-Follower* (LF) [142] are two famous relative

motion-based approaches which are used to control multi-manipulator systems. Lewis [133] formulated two Jacobians which related the joint rates of the entire system to the relative motion of grippers. Hence, two robots are treated as a single redundant system to control two robots which cooperates to perform an assembly task. However, the mobile platforms are considered to be fixed. Choi [143] constructed a complex relative Jacobian which represents the motion of a fixed tool manipulator with respect to a fixed object manipulator using conventional propagation procedure. Chiacchio [144] introduced a relative Jacobian in terms of two single arm's Jacobians and used the pseudo-inverse Jacobian technique for two cooperative fixed 6-DoFs manipulators. Lee [145] used a similar relative Jacobian technique with the end-effector motion being defined by the relative motion between two end-effectors. However, only the relative motion between two end-effectors is controlled. Jamisola [146, 147] introduced a relative Jacobian for fixed parallel dual-arm manipulators in terms of individual manipulators' Jacobians. Besides, the author extends the dual-arm system to a triple-arm cooperating parallel system. Mashali [148] introduced a dual-trajectory tracking strategy for mobile manipulators. Two new joint-dependent control variables  $D$  and  $\alpha$  are defined to augment the task vector. However, it is for a mobile manipulator with a single arm and the calculation of augmented variables  $D$  and  $\alpha$  is complicated.

### 1.2.3.2 Task Priority-based Motion Control

As the complexity of target task increases, the concept of task-priority which regulates multiple tasks in order is introduced by Nakamura [149]. In that work, the author deals with a simple planar three links manipulator and considers the end-effector's motion be prior to the orientation using traditional kinematics. Baerlocher [150] presented two types of formulations for the kinematic control of highly redundant articulated robots, and validated a complex articulated human-like 3D figure control by defining three subtasks. Basile [151] proposed a task-oriented path planning and IK solving method for the cooperative manipulators. But it only deals with the end-effector oriented IK solving with the real motions of the mobile platforms being fixed tracks. Lee [152] introduced four subtasks

with two different priority levels. It solves the second level subtasks identically. Owen [153] introduced a method for selecting the null-space motion based on torque limits and conflicting performance criteria. The author in [154] designed two control laws at velocity and acceleration level, respectively, for fixed dual-arm robots. However, it controls only one single arm, while another arm will not move.

After the necessary tasks being accomplished, the redundant joints are often used to avoid joint limit, velocity/acceleration/torques saturation, singularity and obstacles [155, 156]. The common methods are the saturation in null space (such as the Gradient Projection Method (GPM)) and the Weighted Least Norm method (WLN). The former introduces a scalar function that is to be optimized, and the latter is based on the least-norm method by introducing a weight matrix. Flacco [157] formulated a Saturation in the Null Space (SNS) algorithm for joint limit avoidance. Joint angle and joint velocity are formulated in real time while avoiding the boundary. Chan [158] proposed a WLN method which uses the joint limit function's gradient as the parameter to modulate the weight matrix. Besides, the author introduces a configuration and end-effector based self motion and only dampened the motion in the direction of joint limit. Wang [159] studied the IK solving of a redundant 7-DoFs manipulator based on Closed-loop inverse kinematics (CLIK) algorithm via the null space projection to avoid joint limit. Jia [160] developed a kinematic control method with adaptive motion distribution with the manipulability weighted pseudo-inverse Jacobian. Antonelli [161] proposed a prioritized task based kinematic control with joint velocity commands being limited by dynamically chosen values. Huang [162] introduced a unified redundancy solving method which combined the WLN technique with clamping concept. Therefore, the joint limit is avoided by damping the direction towards limits and forcing the joints away from their limits. Furthermore, it used continuous weighted factor to improve the robustness. Hu [163] designed a framework to handle multi-tasking conflicts for dual-arm robots by constructing a relative Jacobian. However, the proposed recursive joint limit avoidance solving is a little complicated.

### 1.2.3.3 Two Level Motion Planning and Tracking

Most of the motion planning methods mentioned in Subsection 1.2.2 design the trajectory in the open loop. Many works focus on the close loop motion planning with tracking controller to improve the movement performance. Brock [132] presented a framework which combines the task-oriented redundancy solving method and the local controller in the presence of moving obstacles. In that paper, it supposes the task behavior is previously planned. Then, additional behaviors such as posture, obstacle avoidance, constraints are introduced and projected into the null space of task behavior. As a result, the previously planned motion is modified in real time to realize additional behaviors without violating the task behavior. The difficulty comes from the additional behaviors' specifications in various scenarios and their gradient calculations. In addition, it focuses mainly on the redundancy solving of high DoFs robotic systems. Yoshikawa [122] defined the concept of dynamic manipulability measure of robot arms by taking the arm dynamics into consideration. The joint driving force and the acceleration of the end-effector are used to quantitatively measure the manipulating ability in positioning and orienting the end-effectors. Mohri [123] introduced the dynamics into the trajectory planning problem of a nonholonomic mobile manipulator by forming an optimal control problem using the concept of the order of priority and the gradient function which are synthesized in hierarchical manner. Karray [130] derived a fuzzy controller to avoid obstacles in 3D space for the mobile manipulator by updating right and left wheels' velocities assuming that the desired trajectories of mobile platform and end-effectors are known in a static environment. Galicki [129] applies the Lyapunov stability theory to design a group of non-linear on-line trajectory generators for mobile manipulators. The author introduces an extended task error (then obtain the non-singular terminal sliding manifold in task space) and a projection term onto the null space of the extended Jacobian. Nicolis [164] applied the constraint-based trajectory generation and proposed a control frame for sensor-less execution of a force control task for a dual-arm redundant manipulator. Korayem [125] developed an open loop control path planning method and a close loop optimal controller to find maximum load carrying capacity of a two link planar nonholonomic

wheeled mobile manipulator (WMM) in the presence of environmental obstacles. In particular, it applies the obstacle avoidance in the objective function that leads to a two-point boundary value problem, and it solves the iterative solution of nonlinear Hamilton–Jacobi–Bellman (HJB) equation using Galerkin method to design an optimal nonlinear state feedback controller. The maximum load is limited by limiting torque and jerk on joints. On the other hand, the solution for the maximum allowable load depends on the type of user-defined constraints for redundancy solving. Korayem [127] developed a control design (path planning) for a wheeled mobile manipulator among multiple obstacles using the state-dependent Riccati equation (SDRE) considering holonomic and non-holonomic constraints. It solves only obstacle avoidance of the mobile platform and end-effectors without considering the redundancy solving and reasonable posture design.

The linguistic reasoning fuzzy logic method simplifies the motion planning and tracking process by using expert experiences. Son [126] proposed an intelligent rule-based sequence path planning algorithm by introducing a number of rules and by optimizing a fuzzy entropy for a part-bring task with mobile manipulators in 3D partially dynamic environments. Nishiguchi [131] analyzed the human-like behavior of a wheeled humanoid robot that acts as an actor in a human-robot interaction (HRI) theater. Ten rules are extracted for the humanoid robot to realize natural and human-like robot behaviors. The difficulty is to formulate the fuzzy rules to cover all the potential possibilities for high DoF systems.

#### 1.2.3.4 Force Control

In addition to the above control methods, many compliant and force feedback controllers have been introduced to study robot-environment interaction [165–168]. Bonitz [165] introduced an internal force-based impedance control scheme for cooperating manipulators. It enforces a relationship between the velocity of each manipulator and the internal force on the manipulated objects. Each manipulator is directly given the properties of an impedance by the controller, thus eliminating the gain limitation inherent in the structure of previously

proposed schemes. The controller uses the forces sensed at the end-effectors to compensate for the effects of the objects' dynamics and to compute the internal force using only kinematic relationships. Mattioli [169] introduced a method for reconstructing interaction forces and localizing the contact point for humanoids under the static hypothesis. Many hybrid force-motion control laws have been proposed [170–173] which cover the motion and force tracking at the same time.

### 1.2.3.5 Advanced Controllers

Another challenge for controlling the robotic system comes from the uncertainties, like modeling imprecision, unforeseen dynamics, measurement noise, etc. There are many ways to deal with those uncertainties, such as the robust control assuming bounded known function of the uncertainty [174], uncertainties observer, probabilistic approaches, uncertainties modeling using neural network [175] or fuzzy logic system [41].

In [176], an adaptive synchronized control algorithm is proposed for coordination of multiple manipulators in assembly tasks. Each robot is required to track its desired trajectory while maintaining a certain kinematic relationship with the other robots through motion synchronization. Boukattaya [177] presented an effective adaptive control strategies to investigate the trajectory tracking problem of mobile manipulators. The robots are subject to non-holonomic constraints, with the presence of external torque disturbances and dynamic uncertainties. Yang [178] designed an adaptive neural control law for the dual-arm robotic systems to guarantee global stability under prescribed tracking performance. Karray [130] proposed an adaptive controller for a mobile manipulator with model uncertainties and external disturbances with obstacle avoidance. Aleksandar [179] studied the motion adaption of a robot by combining human intervention and autonomous learning.

Ge [180] introduced the GL matrices and operator to facilitate the stability analyses in neural network controller. Then, Ge [181] proposed a direct adaptive neural network control for the robots which means there is no need for the inverse dynamical evaluation and no need for the time-consuming training process. Shuzhi [175] introduced an adaptive neural network control for robot

manipulators in task space. However, because of the inverse solving from the task space to the joint space the author only considered the non-redundant robot manipulators. Li [182] developed systematically an unified controller for a class of mechanical systems with both holonomic and non-holonomic constraints. Besides, the author decoupled the adaptive robust motion and force control with both parameter uncertainties and external disturbances. By extending [183], Qian [1] formulated an adaptive RBF neural network (RBFNN) controller for the mobile manipulator based on the universal approximation theorem. The result in [182] is employed to solve the contact problem.

In addition, many observer-based techniques have been introduced to estimate the uncertainties and disturbances in industrial sectors [184–192]. Extended state observer [189–192] is one of these techniques which introduces an extra state variable to estimate the uncertainties.

The universal approximation property of fuzzy inference systems (FIS) is also used to cope with the uncertainties and disturbances. Benzaoui [193] applied FIS to model the redundant manipulator system with unknown dynamics. Baigzadehnoe [41] designed an adaptive fuzzy back-stepping approach to deal with the unknown dynamical models and unknown external disturbances. All robot manipulators are assumed to be non-redundant and have equal DoFs. Rekha [194] proposed a motion planning approach for a redundant mobile manipulator based on Takagi-Sugeno fuzzy system to track the trajectory of the end-effector in coordination with the mobile platform by maximizing the manipulability criterion. However, many sources of uncertainties [195] occur when applying the fuzzy logic controller (FLC). Type-2 FLC [196–199] offers a powerful method to solve the uncertainties by introducing an extra dimension based on the type-1 FLC. In other words, the type-2 FLC fuzzifies the membership values of type-1. However, when the number of DoFs increases, the fuzzy rules explodes exponentially.

According to the literature survey, motion planning and tracking, force control and adaptive control have been widely studied in recent decades. They are of great interest and challenge to introduce a real personal assistant in the market. Qian [1] has studied the motion and force control of robot MDAMS, and this thesis is the continuation. We focus on the human-like motion planning

and tracking via coordinating the mobile platform and the upper manipulator system for the designed personal assistant MDAMS.

## 1.3 PRELIMINARIES

In this section, some preliminaries on robotics which will be used in the rest of this thesis are given.

### 1.3.1 Forward Kinematics Background

#### 1.3.1.1 Modified Denavit-Hartenberg Method

The Modified Denavit-Hartenberg (MDH) method relates the frames which are fixed on each robot link through the following transformation matrix

$$T_{i-1}^i = Rot(x, \alpha_{i-1})Trans(x, a_{i-1})Rot(z, \theta_i)Trans(z, d_i) \quad (1.1)$$

with four parameters  $\alpha_{i-1}$ ,  $a_{i-1}$ ,  $\theta_i$  and  $d_i$  by a sequence of rotations  $Rot(\cdot, \cdot)$  and translations  $Trans(\cdot, \cdot)$  along certain axes. Four parameters of the MDH method are defined as

$a_{i-1}$  : distance between  $(z_{i-1}, z_i)$  measured along axis  $x_{i-1}$ ,

$d_i$  : distance between  $(x_{i-1}, x_i)$  measured along axis  $z_i$ ,

$\alpha_{i-1}$  : angle between  $(z_{i-1}, z_i)$  measured about axis  $x_{i-1}$ ,

$\theta_i$  : angle between  $(x_{i-1}, x_i)$  measured about axis  $z_i$ .

#### 1.3.1.2 Orientation Tracking Error

If angular velocity  $\omega$  is known, the orientation in task space can be expressed as  $\dot{q} = \frac{1}{2}B(q)\omega$  using unit quaternion  $q = (\eta, \epsilon)$ . The corresponding orientation Jacobian [200] can be formulated as

$$\dot{q} = J_q \dot{\theta} \quad (1.2)$$

where  $J_q \triangleq \frac{1}{2}B(q)J_\omega(\theta)$  is the attitude Jacobian,  $B(q) \triangleq \begin{bmatrix} -\epsilon^T \\ \eta I_3 - \hat{\epsilon} \end{bmatrix} \in \mathcal{R}^{4 \times 3}$  and  $\langle \cdot \rangle$  is a skew-symmetric matrix. Then, the rotation matrix in terms of the unit quaternion is given as follows:

$$R(\eta, \epsilon) = \begin{bmatrix} 2(\eta^2 + \epsilon_x^2) - 1 & 2(\epsilon_x \epsilon_y - \eta \epsilon_z) & 2(\epsilon_x \epsilon_z + \eta \epsilon_y) \\ 2(\epsilon_x \epsilon_y + \eta \epsilon_z) & 2(\eta^2 + \epsilon_y^2) - 1 & 2(\epsilon_y \epsilon_z - \eta \epsilon_x) \\ 2(\epsilon_x \epsilon_z - \eta \epsilon_y) & 2(\epsilon_y \epsilon_z + \eta \epsilon_x) & 2(\eta^2 + \epsilon_z^2) - 1 \end{bmatrix}. \quad (1.3)$$

Assume  $\eta \geq 0$ , this corresponds to an angle  $\theta \in [-\pi, \pi]$ , thus any rotation can be described without singularity.

To quantify the difference between the current and desired orientations of end-effector, define the rotation matrix error as

$$\tilde{R} \triangleq R_d^T R \quad (1.4)$$

where  $R_d$  and  $R$  are respectively the desired and real rotation matrices. Then, the tracking error  $e_q \in \mathcal{R}^4$  expressed by unit quaternion can be extracted from  $\tilde{R}$  as follows:

$$e_q = \begin{bmatrix} e_o \\ e_v \end{bmatrix} = \begin{bmatrix} \eta \eta_d + \epsilon^T \epsilon_d \\ \eta \epsilon_d - \eta_d \epsilon - \hat{\epsilon} \epsilon_d \end{bmatrix} \quad (1.5)$$

where  $(\eta, \epsilon)$  and  $(\eta_d, \epsilon_d)$  are respectively the real and desired orientations,  $e_q^T e_q = e_o^2 + e_v^T e_v = 1$ .

### 1.3.2 Kane's Method for Dynamic Modeling

Kane's method [42] is introduced in the following.

**Definition 1.2 (Generalized speed)** *For a system  $S$  whose configuration in a reference frame  $\{A\}$  is characterized by  $n$  generalized coordinates  $(q_1, \dots, q_n)$ , it is often convenient to introduce  $n$  quantities called generalized speeds for  $S$  in  $\{A\}$ , as linear combinations of  $(q_1, \dots, q_n)$  by means of equations of the following form*

$$u_r \triangleq \sum_{s=1}^n Y_{rs} \dot{q}_s + Z_r, \quad r = 1, \dots, n \quad (1.6)$$

where  $n$  is the number of generalized speed.

**Definition 1.3 (Partial velocity)** The partial velocities of linear or angular velocities  $(\vec{\cdot})_i$  of body  $B_i$  in system  $S$  are obtained using the following formula:

$${}_r(\vec{\cdot})_i = \frac{\partial(\vec{\cdot})_i}{\partial u_r}, \quad r = 1, \dots, n, \quad i = 1, \dots, N \quad (1.7)$$

where  $N$  is the number of bodies in the mechanical system.

**Definition 1.4 (Generalized active forces)** Given a system  $S$  consisting of  $N_p$  particles  $p_1, \dots, p_{N_p}$ . Suppose  $u_r$  ( $r = 1, \dots, n$ ) are the generalized speeds. Let  ${}_r\vec{v}_i$  be the  $r$ -th partial velocity of particle  $p_i$  in frame  $\{A\}$ , and let  $\vec{R}_i$  be the resultant of all contact and body forces acting on particle  $p_i$ . Then,  $F_1, \dots, F_n$  are called generalized active forces for  $S$  in frame  $\{A\}$  which are defined as

$$F_r \triangleq \sum_{i=1}^{N_p} ({}_r\vec{v}_i \vec{R}_i), \quad r = 1, \dots, n. \quad (1.8)$$

**Definition 1.5 (Generalized inertia forces)** Given a system  $S$  consisting of  $N_p$  particles  $p_1, \dots, p_{N_p}$ . Suppose  $u_r$  ( $r = 1, \dots, n$ ) are the generalized speeds. Let  ${}_r\vec{v}_i$  be the  $r$ -th partial velocity of particle  $p_i$  in frame  $\{A\}$ , and let  $\vec{R}_i^*$  be the inertia force  $p_i$  in  $\{A\}$  which is expressed as follows

$$\vec{R}_i^* \triangleq -m_i \vec{a}_i, \quad (i = 1, \dots, N_p) \quad (1.9)$$

where  $m_i$  is the mass of particle  $p_i$  and  $\vec{a}_i$  is the acceleration of particle  $p_i$  in  $\{A\}$ .  $F_1^*, \dots, F_n^*$  are called generalized inertia forces for  $S$  in  $\{A\}$  which are defined as

$$F_r^* \triangleq \sum_{i=1}^{N_p} ({}_r\vec{v}_i \vec{R}_i^*), \quad r = 1, \dots, n. \quad (1.10)$$

Besides, the contribution  $(F_r^*)_k$  to  $F_r^*$  of all inertia forces for the particles of a rigid body  $B_k$ , ( $k = 1, \dots, N$ ) can be expressed in terms of  $\vec{R}_k^*$  and  $\vec{T}_k^*$  which are defined as

$$\vec{R}_k^* \triangleq -m_k \vec{a}_k^*, \quad (1.11)$$

$$\vec{T}_k^* \triangleq -\vec{\epsilon}_k I_k - \vec{\omega}_k \times I_k \vec{\omega}_k \quad (1.12)$$

where  $m_k$  is the mass of body  $B_k$ .  $\vec{\epsilon}_k$  and  $\vec{\omega}_k$  are respectively the angular acceleration and velocity of body  $B_k$  in  $\{A\}$ ,  $I_k$  is the central inertia dynamic of  $B_k$ . Then the generalized inertia forces can be expressed as

$$F_r^* \triangleq \sum_{k=1}^N [\vec{R}_k^*(r \vec{v}_k) + \vec{T}_k^*(r \vec{\omega}_k)], \quad r = 1, \dots, n. \quad (1.13)$$

**Definition 1.6 (Kane's Equations)** Using d'Alembert's principle for the force equilibrium and virtual work theory, Kane's Equations can be formulated as

$$F_r + F_r^* = 0, \quad r = 1, \dots, n. \quad (1.14)$$

### 1.3.3 Kinematic Control of Non-holonomic Mobile Platform

The non-holonomic mobile platform can be controlled kinematically using the result in [201]. The motion of the mobile platform can be characterized using  $(x, y, \phi)$ .  $(x, y)$  and  $\phi$  are respectively the position and orientation of the mobile platform. Introduce  $\theta_v = (\theta_r, \theta_l)$ , namely the rotational angles of the right and left wheels of mobile platform. We can choose here  $\mathbf{v} = (v, \omega) = (v_1, v_2)$  to represent the velocity of the mobile platform, where  $\mathbf{v}_r = (v_r, \omega_r)$  is the corresponding reference velocity. The complete equations of the nonholonomic mobile platform' motion for a mobile manipulator are given in the following.

$$\dot{\theta}_v = S(\theta_v) \mathbf{v}, \quad (1.15)$$

$$\bar{M}_v \dot{\mathbf{v}} + \bar{C}_v \mathbf{v} + \bar{G}_v + \bar{Q}_v + \bar{\tau}_{dv} = \tau_v \quad (1.16)$$

where  $\bar{M}_v$  is the symmetric and positive definite inertia matrix,  $\bar{C}_v$  is the centripetal and Coriolis matrix,  $\bar{G}_v$  is the friction vector,  $\bar{Q}_v$  is the coupling vector from the upper manipulator,  $\bar{\tau}_{dv}$  is the unknown bounded disturbances includ-

ing unstructured dynamics and  $\tau$  is the input vector.  $S(\theta_v) = \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix}$ .

Introduce the tracking error vector  $e_p = (e_1, e_2, e_3) = T_e(\boldsymbol{\theta}_{vr} - \boldsymbol{\theta}_v)$ :

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \phi_r - \phi \end{bmatrix} \quad (1.17)$$

where  $(x_r, y_r, \phi_r)$  is the reference trajectory of the mobile platform. And the derivative of the error is

$$\dot{e}_p = \begin{bmatrix} v_2 e_2 - v_1 + v_r \cos e_3 \\ -v_2 e_1 + v_r \sin e_3 \\ -v_2 + \omega_r \end{bmatrix}. \quad (1.18)$$

To achieve the asymptotic position tracking ability of the mobile platform, one can convert the dynamic control problem into the kinematic control problem

$$\dot{\mathbf{v}} = \mathbf{u}_v \quad (1.19)$$

$$\mathbf{u}_v = \dot{\mathbf{v}}_c + k_4(\mathbf{v}_c - \mathbf{v}) \quad (1.20)$$

where  $k_4$  is a definite positive diagonal matrix.  $\mathbf{v}_c$  is defined as

$$\mathbf{v}_c = \begin{bmatrix} v_r \cos e_3 + k_1 e_1 \\ \omega_r + k_2 v_r e_2 + k_3 v_r \sin e_3 \end{bmatrix}. \quad (1.21)$$

Its time derivative is

$$\dot{\mathbf{v}}_c = \begin{bmatrix} \dot{v}_r \cos e_3 \\ \dot{\omega}_r + k_2 \dot{v}_r e_2 + k_3 \dot{v}_r \sin e_3 \end{bmatrix} + \begin{bmatrix} k_1 & 0 & -v_r \sin e_3 \\ 0 & k_2 v_r & k_3 v_r \cos e_3 \end{bmatrix} \dot{e}_p \quad (1.22)$$

where  $k_i$  ( $i = 1, 2, 3$ ) are definite positive gains.

**Lemma 1.1** *Given a nonholonomic system (Eqs. (1.15) and (1.16)) with  $n$  generalized coordinates  $\boldsymbol{\theta}$ ,  $m$  independent constraints,  $r$  actuators, let the following assumptions hold: A.1. The number of actuators is equal to the number of DoFs (i.e.  $r = n - m$ ). A.2. The reference linear velocity is nonzero and bounded,  $v_r > 0$  for all  $t$ . The angular velocity  $\omega_r$  is bounded. A.3. A smooth auxiliary velocity control input*

$v_c$  is given by Eq. (1.21). A.4.  $[k_1 \ k_2 \ k_3]^T$  is a vector of positive constants. A.5.  $k_4$  is a sufficiently large positive constant.

Let the nonlinear feedback control  $u_v \in \mathbb{R}^{n-m}$  given by Eq. (1.20) be used and the vehicle input commands according to Eq. (1.16). Then, the origin  $e_p = 0$  is uniformly asymptotically stable, and the velocity vector of the mobile platform satisfies  $v \rightarrow v_c$  as  $t \rightarrow \infty$ .

### 1.3.4 Non-dominated Sorting Scheme

The definition of *Dominate* and *Non-dominated* for a multi-objective optimization problem is given in the following [93].

**Definition 1.7 (Dominate and Non-Dominated)** *If all objective functions are for minimization, a feasible solution  $s_1$  dominates another feasible solution  $s_2$  ( $s_1 > s_2$ ) in the Pareto-optimal sense if, and only if,  $s_1$  performs better in at least one objective and, at least, as good as  $s_2$  in the rest:  $s_1 > s_2 \Leftrightarrow \forall \beta \in \{1, \dots, n_{obj}\} : f_\beta(s_1) \leq f_\beta(s_2)$  and  $\exists \lambda \in \{1, \dots, n_{obj}\} : f_\lambda(s_1) < f_\lambda(s_2)$  where  $n_{obj}$  is the number of objective functions. If none of the two feasible solutions,  $s_1$  and  $s_2$ , dominates the other, then they are said to be non-dominated.*

**Definition 1.8 (Fast non-dominated Sorting [99])** *Algorithm 1 illustrates the fast non-dominated sorting scheme. The **for** loop (lines 1-13) assigns the number of individuals that dominate each individual and forms the first front with non-dominated individuals. The **while** loop (lines 14-25) assigns incrementally the following fronts among the dominated individuals.*

## 1.4 Conclusions

This chapter gives the state of the art in robotics and preliminaries which will be used in this thesis. In the following chapters, system modeling, motion planning and tracking problems will be investigated of the designed mobile dual-arm manipulator system.

---

**Algorithm 1** Fast non-dominated Sorting Scheme.
 

---

```

1: for each  $p \in P$  do
2:    $S_p = \emptyset, n_p = 0$ 
3:   for each  $q \in P$  do
4:     if ( $p < q$ ) then
5:        $S_p = S_p \cup \{q\}$  {if  $p$  dominates  $q$ }
6:     else if ( $q < p$ ) then
7:        $n_p = n_p + 1$  {if  $q$  dominates  $p$ }
8:     end if
9:     if  $n_p = 0$  then
10:       $p_{rank} = 1, F_1 = F_1 \cup \{p\}$  { $p$  belongs to the first front  $F_1$ }
11:    end if
12:  end for
13: end for
     $i = 1$  {initialize the front counter}
14: while  $F_i \neq \emptyset$  do
15:    $Q = \emptyset$  {initialize  $Q$  to store the next front}
16:   for each  $p \in F_i$  do
17:     for each  $q \in S_p$  do
18:        $n_q = n_q - 1$ 
19:       if  $n_q = 0$  then
20:          $q_{rank} = i + 1, Q = Q \cup \{q\}$  { $q$  belongs to the next front}
21:       end if
22:     end for
23:   end for
24:    $i = i + 1, F_i = Q$ 
25: end while

```

---

# SYSTEM DESCRIPTION AND MODELING

## 2.1 INTRODUCTION

The construction of a concise and precise mathematical model is the fundamental to study the robotic system. Many methods have been introduced in Chapter 1. In [1], system conception and dynamic model based on Lagrangian formulations have been completed. However, the calculations of energy and its partial derivatives become complex as the number of DoFs increases. Besides, an adaptive RBFNN controller has been designed in that work to cope with the uncertainties and disturbances, but its effectiveness remains to be validated.

In this Chapter, firstly the kinematic modeling of the designed MDAMS is constructed. The MDH method is used to establish the kinematic model with only four parameters in Subsection 2.2.1. Then, Lagrangian formulations and the adaptive RBFNN controller are recalled in Subsection 2.2.2. The validation of the adaptive RBFNN controller on the virtual prototype in ADAMS via MATLAB/SIMULINK and ADAMS co-simulation is realized. In order to overcome the shortcomings of Lagrangian formulations, Kane's method [42] is used to construct the dynamic model for MDAMS in Subsection 2.2.3. Finally, the physical tip-over stability is studied based on the constructed Kane's model and a Lyapunov stable controller is proposed using the back-stepping technique.

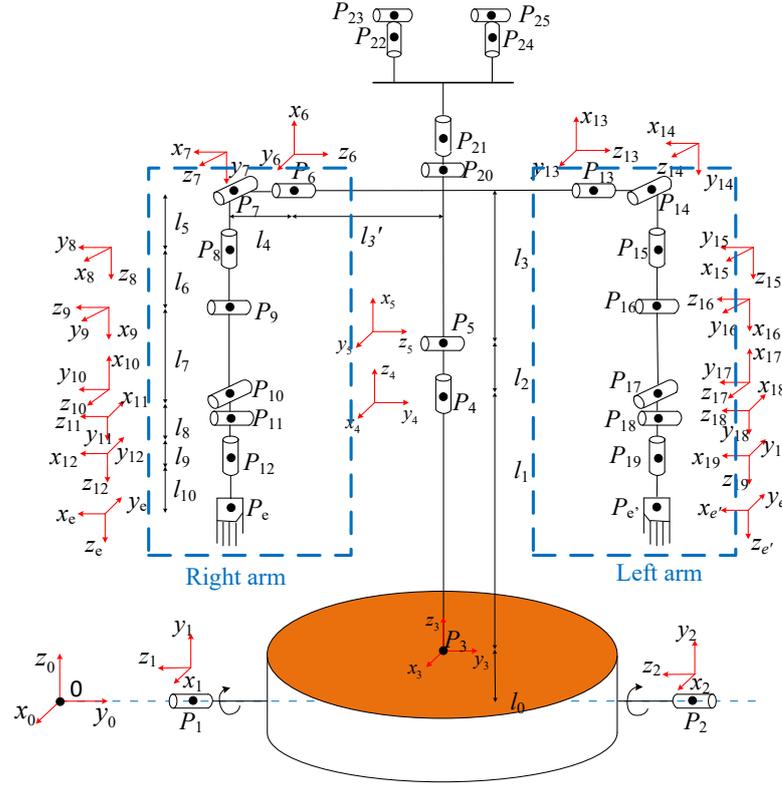


Figure 2.1 – The structure of the proposed robot MDAMS.

## 2.2 SYSTEM MODELING FOR MDAMS

### 2.2.1 System Description and Kinematic Modeling

#### 2.2.1.1 System Description

The structure of the proposed MDAMS is shown in Fig. 2.1. It consists of  $N$  bodies and has up to 22 – 56 DoFs. It has two arm-hand subsystems which share one common waist with rotation and bend-over movements and one common non-holonomic mobile platform. In total, 22 frames are defined without considering the neck, head and fingers. Specifically, frame  $\Sigma_0$  is the world frame,  $\Sigma_i$  ( $i = 1, \dots, 19$ ) is the body frame with its origin locating at  $P_i$ , and frames  $\Sigma_e$  and  $\Sigma_{e'}$  are end-effectors' (EEs) frames with a translation along axis  $z_{12}$  and  $z_{19}$ , respectively.  $P_e$  and  $P_{e'}$  are the palm centers.  $l_0$  is the translation of frame  $\Sigma_3$  along axis  $z_0$  in frame  $\Sigma_0$  and  $l_k$  ( $k = 1, \dots, 10$ ) characterizes the geometric

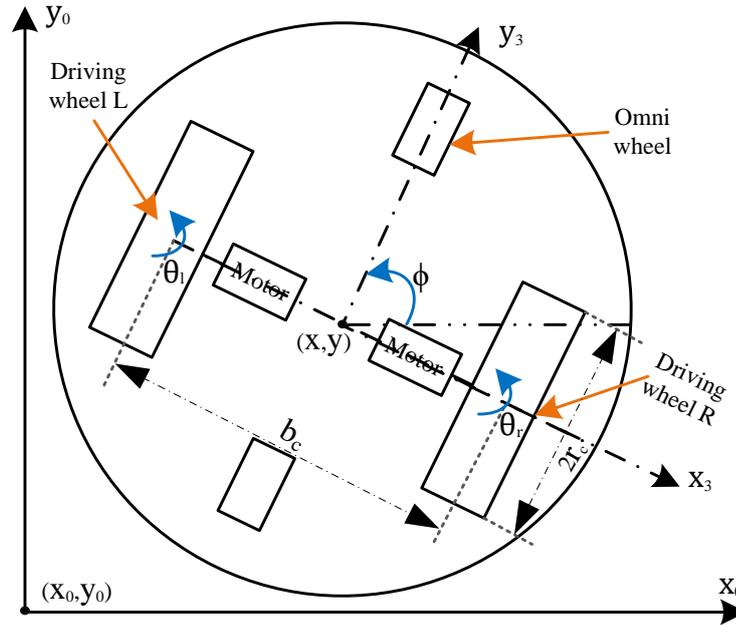


Figure 2.2 – Modeling of the non-holonomic mobile platform.

dimension of MDAMS. In particular,  $\Sigma_3$  is the mobile platform's frame which locates initially at  $(x_0, y_0)$ .  $(x, y)$  and  $\phi$  are respectively the mobile platform's position and orientation in frame  $\Sigma_0$  (see Fig. 2.2).

The MDH method (see Section 1.3.1.1) relates frame  $\Sigma_i$  with respect to frame  $\Sigma_{i-1}$  via the transformation matrix  $T_{i-1}^i$  which can be written as

$$T_{i-1}^i = \begin{bmatrix} R_{i-1}^i & P_{i-1}^i \\ 0 & 1 \end{bmatrix} \quad (2.1)$$

through a sequence of rotations and translations to characterize the mechanical system with only four parameters  $\alpha_{i-1}$ ,  $a_{i-1}$ ,  $\theta_i$  and  $d_i$ .  $R_{i-1}^i$  and  $P_{i-1}^i$  are respectively the rotation matrix and position vector of frame  $\Sigma_i$  with respect to frame  $\Sigma_{i-1}$  which have the following expressions

$$R_{i-1}^i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 \\ \cos\alpha_{i-1}\sin\theta_i & \cos\alpha_{i-1}\cos\theta_i & -\sin\alpha_{i-1} \\ \sin\alpha_{i-1}\sin\theta_i & \sin\alpha_{i-1}\cos\theta_i & \cos\alpha_{i-1} \end{bmatrix}, \quad (2.2)$$

Table 2.1 – MDH Parameters of MDAMS.

Body/Frame $\Sigma_i$	$\alpha_{i-1}$	$a_{i-1}$	$\theta_{i0}$	$d_i$
1	$90^\circ$	0	$0^\circ$	$b_v/2$
2	$90^\circ$	0	$0^\circ$	$-b_v/2$
3	-	-	-	-
4	$0^\circ$	0	$0^\circ$	$l_1 + l_2$
5	$-90^\circ$	0	$-90^\circ$	0
6	$0^\circ$	$l_3$	$0^\circ$	$-(l'_3 + l_4)$
13	$0^\circ$	$l_3$	$0^\circ$	$(l'_3 + l_4)$
7/14	$-90^\circ$	0	$90^\circ$	0
8/15	$-90^\circ$	0	$-90^\circ$	0
9/16	$-90^\circ$	0	$-90^\circ$	$l_5 + l_6$
10/17	$-90^\circ$	$l_7$	$-180^\circ$	0
11/18	$-90^\circ$	$-l_8$	$90^\circ$	0
12/19	$-90^\circ$	0	$-90^\circ$	$l_9$
e/e'	$0^\circ$	0	$0^\circ$	$l_{10}$

and

$$P_{i-1}^i = \begin{bmatrix} a_{i-1} \\ -d_i \sin \alpha_{i-1} \\ d_i \cos \alpha_{i-1} \end{bmatrix}. \quad (2.3)$$

The detailed values of  $\alpha_{i-1}$ ,  $a_{i-1}$ ,  $\theta_i$  and  $d_i$  for each frame are listed in Table 2.1, where  $r_v$  and  $b_v$  are respectively the radius of each driving wheel and the distance between the two driving wheels.

A virtual prototype is constructed in ADAMS for numerical validation via MATMAB/SIMULINK and ADAMS co-simulation and a mechanical system is under consideration for real robot construction in Fig. 2.3. The main inertial properties of robot MDAMS are listed in Table 2.2.

Table 2.2 – Main inertial properties of MDAMS.

Part	Mass [kg]	Rotation inertia around CoM $I_{ij}$ [kg.mm <sup>2</sup> ], $i = j, i, j \in \{x, y, z\}$			Products of inertia [kg.mm <sup>2</sup> ] $I_{ij} \ i \neq j$
		$I_{xx}$	$I_{yy}$	$I_{zz}$	
Head	2	$1.452 \times 10^4$	$1.196 \times 10^4$	$1.177 \times 10^4$	0
Neck	2	3180.97	3020.63	2687.458	0
L Body	50	$1.796 \times 10^7$	$1.961 \times 10^7$	$2.83 \times 10^7$	0
Waist	6	$2.112 \times 10^4$	$2.118 \times 10^4$	$1.87 \times 10^4$	0
U Body	10	$1.608 \times 10^5$	$2.419 \times 10^5$	$2.096 \times 10^5$	0
Arm1	2.477	9152.28	6142.9	6363.91	0
Arm2	1.192	4041.89	1086.44	4465.585	0
Arm4	0.898	607.69	3698.59	3851.59	0
Arm3	0.57	1436.32	1133.86	939.69	0
Arm5	0.273	103.248	163.444	218.373	0
Arm6	0.096	26.087	11.966	30.698	0
Arm7	0.3	657.051	792.638	1326.87	0

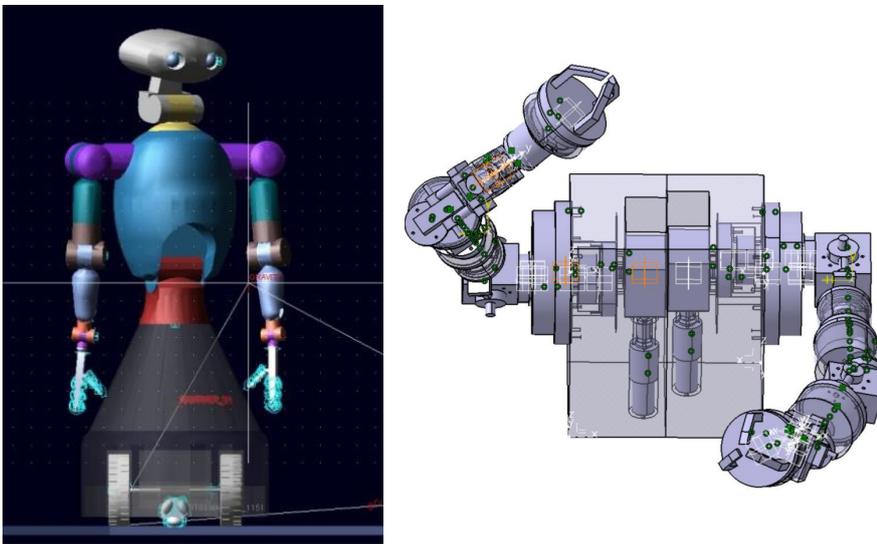


Figure 2.3 – Left: virtual prototype in ADAMS. Right: real robot construction.

### 2.2.1.2 Forward Kinematics

Define the generalized variable  $\theta$  as

$$\theta \triangleq \left[ \theta_m^T \quad \theta_w^T \quad \theta_R^T \quad \theta_L^T \right]^T \quad (2.4)$$

where  $\theta_m \in \mathbb{R}^{n_m}$  expresses the non-holonomic mobile platform's motion which can be chosen as  $(x, y, \phi)$  or  $(\theta_r, \theta_l)$ . Here,  $\theta_r$  and  $\theta_l$  are respectively the joint position of the right and left driving wheels of mobile platform.  $\theta_w \in \mathbb{R}^{n_w}$  represents the joints in the waist.  $\theta_R \in \mathbb{R}^{n_R}$  and  $\theta_L \in \mathbb{R}^{n_L}$  represent respectively the right and left arms' joint positions. In this thesis,  $n_m = 2$  (or 3),  $n_w = 2$ ,  $n_R = 7$ ,  $n_L = 7$ ,  $n = n_m + n_w + n_R + n_L$ . The generalized speed is defined as

$$u \triangleq \dot{\theta} = \left[ \dot{\theta}_m^T \quad \dot{\theta}_w^T \quad \dot{\theta}_R^T \quad \dot{\theta}_L^T \right]^T \in \mathbb{R}^n. \quad (2.5)$$

$u_r$  ( $r = 1, \dots, n$ ) and  $\dot{u}_r$  are respectively the  $r$ -th generalized speed and acceleration.

Note  $\theta_o \triangleq \left[ \theta_m^T \quad \theta_w^T \right]^T$ . For the right EE, the transformation matrix is

$$T_R = T_0^3 T_3^4 T_4^5 T_5^6 T_6^e(\theta_o, \theta_R). \quad (2.6)$$

Then, the motion of the right EE can be expressed as

$$\dot{x}_R = \begin{bmatrix} J_{vR} & 0_{n_L} \end{bmatrix} u = J_{vR} u \quad (2.7)$$

where  $x_R$  and  $\dot{x}_R \in \mathbb{R}^m$  are respectively the position and velocity of the right EE in frame  $\Sigma_0$ ,  $J_{vR} \in \mathbb{R}^{m \times (n_m + n_w + n_R)}$  and  $J_{vR} \in \mathbb{R}^{m \times n}$  are the corresponding Jacobians of the right EE. Similarly, for the left EE, the transformation matrix is

$$T_L = T_0^3 T_3^4 T_4^5 T_5^{13} T_{13}^{e'}(\theta_o, \theta_L). \quad (2.8)$$

Then, the motion of the left EE can be expressed as

$$\dot{x}_L = \begin{bmatrix} J_{vLo} & 0_{n_R} & J_{vLa} \end{bmatrix} u = J_{vL} u \quad (2.9)$$

where  $x_L$  and  $\dot{x}_L \in \mathbb{R}^m$  are respectively the position and velocity of the left EE in

frame  $\Sigma_0$ ,  $J_{vL0} \in \mathbb{R}^{m \times (n_m + n_w)}$ ,  $J_{vLa} \in \mathbb{R}^{m \times n_R}$  and  $J_{vL} \in \mathbb{R}^{m \times n}$  are the corresponding Jacobians of the left EE in frame  $\Sigma_0$ . Here after we note  $J_R \triangleq J_{vR}$  and  $J_L \triangleq J_{vL}$ .

In addition, EEs' orientation angular velocities can be represented by

$$\omega_{rl} = J_\omega u \quad (2.10)$$

where  $\omega_{rl} = \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix}$  and  $J_\omega = \begin{bmatrix} J_{\omega_R} \\ J_{\omega_L} \end{bmatrix}$ .  $J_{\omega_i}$  ( $i = R, L$ ) is the EE's orientation Jacobian.

If multi-fingered hand is installed at the end of each arm [1], the kinematics can be derived similarly by introducing a new parameter  $\theta_f \in \mathbb{R}^{n_f}$  using the MDH method, where  $n_f$  is the number of DoFs of two multi-fingered hands. Then, the generalized variable becomes  $\theta \triangleq \begin{bmatrix} \theta_m^T & \theta_b^T & \theta_f^T \end{bmatrix}^T$ , where  $\theta_b \triangleq \begin{bmatrix} \theta_w^T & \theta_R^T & \theta_L^T \end{bmatrix}^T$ .

### 2.2.1.3 Velocity and Acceleration Formulations

This part formulates linear and angular velocities and accelerations for each body  $B_i$  of MDAMS. Note  $\vec{x}_i$ ,  $\vec{y}_i$ , and  $\vec{z}_i$  as the three unit vectors of frame  $\Sigma_i$  in the world frame  $\Sigma_0$ .  $\vec{\omega}_i$  is the angular velocity of body  $B_i$  in the world frame  $\Sigma_0$ ,  $\vec{v}_i \triangleq \vec{v}_i^*$  is the linear velocity of the center of mass (CoM) of body  $B_i$  in the world frame  $\Sigma_0$ , ( $i = 1, \dots, 19$ ).  $\vec{V}_{p_i}$  is the linear velocity in the world frame  $\Sigma_0$  of point  $P_i$  which locates at the origin of frame  $\Sigma_i$ . Note  $\vec{r}_{p_i}$  and  $\vec{r}_i^*$  as the origin vector of frame  $\Sigma_i$  and the position vector of link  $i$ 's CoM which are listed respectively in Tables A.1 and A.4. We note  $\vec{r}_i \triangleq \vec{r}_i^*$  throughout the process of formulation. We also note  $\vec{e}_i$  as the angular acceleration of body  $B_i$ , and  $\vec{a}_i \triangleq \vec{a}_i^*$  as the linear acceleration of body  $B_i$ 's CoM, ( $i = 1, \dots, 19$ ),  $\vec{a}_{p_i}$  as the linear acceleration of origin  $P_i$  in the world frame  $\Sigma_0$ . Unless otherwise stated, all vectors throughout this thesis are expressed in the world frame  $\Sigma_0$ .

**Non-holonomic Mobile Platform** The mobile platform in Fig. 2.2 is a non-holonomic system. Its angular velocity is

$$\vec{\omega}_3 = \frac{r_v}{b_v} (u_1 - u_2) \vec{z}_0. \quad (2.11)$$

The linear velocity of the mobile platform's CoM is the same as that of frame  $\Sigma_3$ , i.e.  $\vec{v}_3 = \vec{v}_0$  which is expressed as follows.

$$\vec{v}_3 = \frac{r_v}{2} \cos \phi_0 (u_1 + u_2) \vec{x}_0 + \frac{r_v}{2} \sin \phi_0 (u_1 + u_2) \vec{y}_0. \quad (2.12)$$

And the angular velocities of the right and left wheels are respectively

$$\vec{\omega}_1 = \frac{r_v}{b_v} (u_1 - u_2) \vec{z}_0 + u_1 \vec{z}_1, \quad (2.13)$$

and

$$\vec{\omega}_2 = \frac{r_v}{b_v} (u_1 - u_2) \vec{z}_0 + u_2 \vec{z}_2. \quad (2.14)$$

Then, the linear velocities of the right and left wheels' CoMs are respectively

$$\vec{v}_1 = (\dot{x}_0 + \frac{r_v}{b_v} (u_1 - u_2)) b_v \vec{x}_0 + \dot{y}_0 \vec{y}_0, \quad (2.15)$$

and

$$\vec{v}_2 = (\dot{x}_0 - \frac{r_v}{b_v} (u_1 - u_2)) b_v \vec{x}_0 + \dot{y}_0 \vec{y}_0. \quad (2.16)$$

Besides, the angular acceleration of the mobile platform can be expressed as

$$\vec{\epsilon}_3 = \frac{r_v}{b_v} (\dot{u}_1 - \dot{u}_2) \vec{z}_0. \quad (2.17)$$

The angular accelerations of the right and left wheels are respectively

$$\vec{\epsilon}_1 = \frac{r_v}{b_v} (\dot{u}_1 - \dot{u}_2) \vec{z}_0 + \frac{r_v}{b_v} (u_1 - u_2) u_1 \vec{z}_0 \times \vec{z}_1 + \dot{u}_1 \vec{z}_1, \quad (2.18)$$

and

$$\vec{\epsilon}_2 = \frac{r_v}{b_v} (\dot{u}_1 - \dot{u}_2) \vec{z}_0 + \frac{r_v}{b_v} (u_1 - u_2) u_2 \vec{z}_0 \times \vec{z}_2 + \dot{u}_2 \vec{z}_2. \quad (2.19)$$

Finally, the linear acceleration of the mobile platform is  $\vec{a}_3 = \vec{a}_0$ , i.e.

$$\vec{a}_3 = \ddot{x}_0 \vec{x}_0 + \ddot{y}_0 \vec{y}_0. \quad (2.20)$$

$\vec{a}_3$  is also the linear acceleration of the mobile platform's CoM and of the frame  $\Sigma_3$ . The linear accelerations of the right and left wheels' CoMs are respectively

$$\vec{a}_1 = \vec{a}_3 - \frac{r_v}{b_v}(u_1 - u_2)b_v\vec{z}_0 \times \vec{y}_3, \quad (2.21)$$

and

$$\vec{a}_2 = \vec{a}_3 + \frac{r_v}{b_v}(u_1 - u_2)b_v\vec{z}_0 \times \vec{y}_3. \quad (2.22)$$

**Upper Body Manipulator** The angular velocities of the upper manipulator system's bodies are given as follows:

$$\vec{\omega}_i = \vec{\omega}_{i-1} + u_{i-1}\vec{z}_i, \quad i = 4, \dots, 19. \quad (2.23)$$

In particular, the angular velocity of body  $B_{13}$  is  $\vec{\omega}_{13} = \vec{\omega}_5 + u_{13}\vec{z}_{13}$ . Then, the linear velocities of upper bodies are gradually derived hereafter. The linear velocity  $\vec{v}_i$  of body  $i$ 's CoM can be expressed as

$$\vec{v}_i = \vec{V}_{p_i}^* + {}^r\vec{\omega}_i \times \vec{r}_i^*, \quad i = 1, \dots, 19 \quad (2.24)$$

where  $\vec{r}_i^*$  is body  $i$ 's CoM vector in frame  $\Sigma_i$  which is specified in Table A.1.  $\vec{V}_{p_i}^*$  is the linear velocity of origin  $P_i$  ( $i = 4, \dots, 19$ ) can be derived as

$$\vec{V}_{p_i}^* = \vec{V}_{p_{i-1}}^* + {}^r\vec{\omega}_{i-1} \times \vec{r}_{p_i}^* \quad (2.25)$$

where  $\vec{r}_{p_i}^*$  and  ${}^r\vec{\omega}_i$  are respectively the frame position vector and the relative angular velocity of frame  $\Sigma_i$  in frame  $\Sigma_{i-1}$ . Specifically, the linear velocities of point  $P_4$  and body 4's CoM are respectively

$$\vec{V}_{p_4}^* = \vec{v}_3, \quad (2.26)$$

and

$$\vec{v}_4 = \vec{v}_3. \quad (2.27)$$

Similarly, the linear velocity of origin  $P_5$  and body 5's CoM are respectively

$$\vec{V}_{p_5} = \vec{v}_4, \quad (2.28)$$

and

$$\vec{v}_5 = \vec{v}_3 + u_3 r_5^* \vec{z}_4 \times \vec{x}_5 + u_4 r_5^* \vec{y}_5. \quad (2.29)$$

And the linear velocity of origin  $P_6$  and body 6's CoM are respectively

$$\vec{V}_{p_6} = \vec{v}_3 + u_4 l_3 \vec{y}_5, \quad (2.30)$$

and

$$\vec{v}_6 = \vec{v}_3 + u_4 (l_3 \vec{y}_5 - r_6^* \vec{z}_5 \times \vec{z}_6). \quad (2.31)$$

The linear velocities of origin  $P_{13}$  and body 13's CoM are respectively

$$\vec{V}_{p_{13}} = \vec{v}_3 + u_4 l_3 \vec{y}_5, \quad (2.32)$$

and

$$\vec{v}_{13} = \vec{v}_3 + u_4 (l_3 \vec{y}_5 + r_{13}^* \vec{z}_5 \times \vec{z}_{13}). \quad (2.33)$$

Recall that the implicated velocity  $\vec{V}_{p_i}^*$  of body  $i$ 's CoM can be expressed as

$$\vec{V}_{p_i}^* = \vec{V}_{p_{i-1}} + {}^r \vec{\omega}_{i-1} \times \vec{r}_{p_i}^* \quad (2.34)$$

where  $\vec{r}_{p_i}^* = \vec{r}_{p_i} + \vec{r}_i^*$ .

The angular accelerations of the upper bodies are firstly given as follows

$$\vec{\epsilon}_i = \vec{\epsilon}_{i-1} + (\vec{\omega}_{i-1} \times {}^r \vec{\omega}_i + \dot{u}_{i-1} \vec{z}_i), \quad i = 4, \dots, 19. \quad (2.35)$$

In particular, the angular accelerations of bodies 4, 5, 6 and 13 are expressed as:

$$\vec{\epsilon}_4 = \frac{r_v}{b_v} (\dot{u}_1 - \dot{u}_2) \vec{z}_0 + \frac{r_v}{b_v} (u_1 - u_2) u_3 \vec{z}_0 \times \vec{z}_4 + \dot{u}_3 \vec{z}_4, \quad (2.36)$$

$$\vec{\epsilon}_5 = \vec{\epsilon}_4 + \frac{r_v}{b_v} (u_1 - u_2) u_4 \vec{z}_0 \times \vec{z}_5 + u_3 u_4 \vec{z}_4 \times \vec{z}_5 + \dot{u}_4 \vec{z}_5, \quad (2.37)$$

$$\vec{\epsilon}_6 = \vec{\epsilon}_5 + (\vec{\omega}_5 \times^r \vec{\omega}_6 + {}^r \vec{\epsilon}_6), \quad (2.38)$$

$$\vec{\epsilon}_{13} = \vec{\epsilon}_5 + (\vec{\omega}_5 \times^r \vec{\omega}_{13} + \dot{u}_{12} \vec{z}_{13}). \quad (2.39)$$

Since the relative velocity and acceleration of body  $i$ 's CoM with respect to frame  $\Sigma_i$  are  $\vec{v}_{ri} = \dot{r}_{xi} \vec{x}_i + \dot{r}_{yi} \vec{y}_i + \dot{r}_{zi} \vec{z}_i = \vec{0}$  and  $\vec{a}_{ri} = \ddot{r}_{xi} \vec{x}_i + \ddot{r}_{yi} \vec{y}_i + \ddot{r}_{zi} \vec{z}_i = \vec{0}$ , respectively, then the linear acceleration  $\vec{a}_i \triangleq \vec{a}_i^*$  of body  $i$ 's CoM can be expressed as

$$\vec{a}_i = \vec{a}_{p_i} + \vec{\epsilon}_i \times \vec{r}_i^* + \vec{\omega}_i \times^r \vec{v}_i, \quad i = 1, \dots, 19 \quad (2.40)$$

where  $\vec{a}_{p_i}$  is the linear acceleration of origin  $P_i$ ,  ${}^r \vec{v}_i$  is the relative linear velocity of body  $i$ 's CoM with respect to origin  $P_i$ . Then, the linear acceleration of body  $i$ 's CoM is

$$\vec{a}_i = \vec{a}_{p_i} + \vec{\epsilon}_i \times \vec{r}_i^* + \vec{\omega}_i \times^r \vec{v}_i, \quad i = 4, \dots, 19. \quad (2.41)$$

In particular, the linear accelerations of the CoMs of bodies {4} and {5} are

$$\vec{a}_4 = \vec{a}_3, \quad (2.42)$$

and

$$\vec{a}_5 = \vec{a}_{p_5} + \vec{\epsilon}_5 \times \vec{r}_5^* + \vec{\omega}_5 \times^r \vec{v}_5. \quad (2.43)$$

The linear accelerations of origins  $P_5$ ,  $P_6$  and  $P_{13}$  are respectively

$$\vec{a}_{p_5} = \vec{a}_3, \quad (2.44)$$

$$\vec{a}_{p_6} = \vec{a}_{p_5} + \vec{\epsilon}_5 \times \vec{r}_{p_6} + \vec{\omega}_5 \times^r \vec{v}_{p_6}, \quad (2.45)$$

and

$$\vec{a}_{p_{13}} = \vec{a}_{p_5} + \vec{\epsilon}_5 \times \vec{r}_{p_{13}} + \vec{\omega}_5 \times^r \vec{v}_{p_{13}}. \quad (2.46)$$

Then, the linear acceleration of origin  $P_i$  can be expressed as

$$\vec{a}_{p_i} = \vec{a}_{p_{i-1}} + \vec{\epsilon}_{i-1} \times \vec{r}_{p_{i-1}} + \vec{\omega}_{i-1} \times^r \vec{v}_{p_{i-1}}, \quad i = 6, \dots, 19 \quad (2.47)$$

where  $\vec{r}_{p_i}$  is the position vector of frame  $\Sigma_i$  with respect to frame  $\Sigma_{i-1}$ .

## 2.2.2 Lagrangian Formulations and Control

### 2.2.2.1 Dynamic Modeling using Lagrangian Formulations

In this Subsection, the dynamic model for MDAMS using Lagrange's method [1] is recalled. According to Lagrangian formulations, if the forces exerted on the system are partially potential, then the system dynamics can be expressed as follows

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right) - \frac{\partial L}{\partial \theta_i} = \tau_i \quad (2.48)$$

where  $L = K - P$ ,  $K$  and  $P$  are respectively the total kinetic energy and total potential energy of the robotic system.  $\theta_i$  is the joint variable,  $\dot{\theta}_i$  is the first time derivative of  $\theta_i$  and  $\tau_i$  is the generalized force (torque) at joint  $i$ .  $\frac{\partial(\cdot)}{\partial \theta_i}$  and  $\frac{\partial(\cdot)}{\partial \dot{\theta}_i}$  are the corresponding partial derivatives.

Then, the dynamic equations can be given as follows:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) + A^T(\theta)\lambda + \tau_d = E(\theta)\tau \quad (2.49)$$

where  $M(\theta) \in \mathcal{R}^{n \times n}$  is the symmetric and positive definite inertia matrix,  $C(\theta, \dot{\theta}) \in \mathcal{R}^{n \times n}$  is the centripetal and Coriolis matrix,  $G(\theta) \in \mathcal{R}^{n \times 1}$  is the friction and gravity vector,  $A(\theta) \in \mathcal{R}^{n \times n}$  is the constraint matrix,  $\lambda \in \mathcal{R}^{n \times 1}$  is the Lagrange multiplier vector which is the constraint forces,  $\tau_d \in \mathcal{R}^{n \times 1}$  is the unknown bounded disturbances including unstructured dynamics,  $E(\theta) \in \mathcal{R}^{n \times n}$  is the input transformation matrix and  $\tau \in \mathcal{R}^{n \times 1}$  is the torque input vector.

The non-holonomic constraints indicate that the robot can only move in a direction perpendicular to the axis of mobile platform's drive wheel. In other words, the mobile platform satisfies both pure rolling and non slipping conditions:  $\dot{y}\cos\phi - \dot{x}\sin\phi = 0$ . As derived in [1], Eq. (2.49) can be rewritten as:

$$\bar{M}(\theta)\ddot{\theta} + \bar{C}(\theta, \dot{\theta}) + \bar{G}(\theta) + \bar{\tau}_d = \bar{\tau} \quad (2.50)$$

where  $\bar{\tau} = S^T E \tau$ ,  $\bar{M}(\theta) = S^T M(\theta) S$ ,  $\bar{C}(\theta, \dot{\theta}) = S^T M(\theta) \dot{S} + S^T C(\theta, \dot{\theta}) S$ ,  $\bar{G}(\theta) = S^T G(\theta)$ ,  $\bar{\tau}_d = S^T \tau_d$ . Matrix  $S = \begin{bmatrix} S_v & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$  spans the null space of  $A(\theta) = \begin{bmatrix} A_v & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$ ,

$$\text{i.e. } S^T A^T(\theta) = 0, \text{ with } S_v = \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix} \text{ and } A_v = \begin{bmatrix} -\sin\phi & \cos\phi & 0 \end{bmatrix}.$$

If the multi-fingered dexterous hands are mounted at the ends of robotic arms directly, then the following dynamic equation is given.

$$\bar{M}'\dot{\xi} + \bar{C}'\xi + \bar{G}'(\theta) + \bar{\tau}'_d = E(\theta)'\bar{\tau}' \quad (2.51)$$

with the dynamic matrices being expressed as follows.  $\xi = \begin{bmatrix} \dot{\theta}_v \\ \dot{\theta}_b \\ \dot{\theta}_f \end{bmatrix}$ ,

$$\bar{M}' = \begin{bmatrix} S_v^T M_v S_v & S_v^T M_{vb} & S_v^T M_{vf} \\ M_{bv} & M_b & M_{bf} \\ M_{fv} & M_{fv} & M_f \end{bmatrix}, \bar{C}' = \begin{bmatrix} S_v^T M_v \dot{S}_v + S_v^T C_v S_v & S_v^T C_{vb} & S_v^T C_{vf} \\ M_{bv} \dot{S}_v + C_{bv} S_v & C_b & C_{bf} \\ M_{fv} \dot{S}_v + C_{fv} S_v & C_{fb} & C_f \end{bmatrix},$$

$$\bar{G}' = \begin{bmatrix} S_v^T G_c \\ G_b \\ G_f \end{bmatrix}, \bar{\tau}'_d = \begin{bmatrix} S_v^T \tau_{dv} \\ \tau_{db} \\ \tau_{df} \end{bmatrix}, \bar{E}'\bar{\tau}' = \begin{bmatrix} S_v^T E_c \tau_v \\ \tau_b \\ \tau_f \end{bmatrix} \text{ and } E_c(\theta_c) = \frac{1}{r_v} \begin{bmatrix} \cos\phi & \cos\phi \\ \sin\phi & \sin\phi \\ \frac{b_v}{2} & \frac{-b_v}{2} \end{bmatrix}.$$

It can be seen that the Lagrangian formulations become complex and time-consuming as the DoFs of robotic system increase because of the requirements of energy and its partial derivative calculations.

### 2.2.2.2 Adaptive RBF Neural Network Controller

Recall the adaptive radial basis function neural network (RBFNN) controller proposed in [1]

$$S^T E \tau = \hat{M} \dot{\eta}_r + \hat{C} \eta_r + \hat{F} + K r + U_s \quad (2.52)$$

where  $\hat{M}$ ,  $\hat{C}$  and  $\hat{F}$  represent the estimates of the true parameter matrices and vector.  $K$  is definite positive and  $Kr$  gives the PD control. The robust controller is proposed as  $U_s = \hat{\psi} \text{sgn}(r)$  for compensating the approximation errors and uncertainties and  $\hat{\psi}$  is the estimated value of the uncertain term  $\psi$  bound.  $\eta = \begin{bmatrix} v & \omega & u_3 & \dots & u_n \end{bmatrix}^T$ ,  $\eta_r$  is the reference joint velocity and  $r \triangleq \eta_r - \eta$  is the tracking error.

The adaptive laws are proposed as follows using the universal approximation

theory and the GL matrix.

$$\begin{aligned}
\dot{\hat{\theta}}_k &= \Gamma_k \{ \xi_k(q) \} \dot{\eta}_r r_k, \\
\dot{\hat{a}}_k &= Q_k \{ \varsigma_k(z) \} \eta_r r_k, \\
\dot{\hat{\beta}}_k &= N_k \{ \sigma_k(q) \} r_k, \\
\dot{\hat{\psi}}_k &= T_k r_k \operatorname{sgn}(r) k, \\
\dot{\hat{c}}_{\theta_k} &= E_{\alpha_k} \{ \xi_{kc} \} \hat{\theta}_k \dot{\eta}_r r_k, \quad \dot{\hat{c}}_{\alpha_k} = E_{\theta_k} \{ \varsigma_{kc} \} \hat{a}_k \eta_r r_k, \quad \dot{\hat{c}}_{\beta_k} = E_{\beta_k} \{ \sigma_{kc} \} \hat{\beta}_k r_k, \\
\dot{\hat{b}}_{\theta_k} &= I_{\alpha_k} \{ \xi_{kb} \} \hat{\theta}_k \dot{\eta}_r r_k, \quad \dot{\hat{b}}_{\alpha_k} = I_{\theta_k} \{ \varsigma_{kb} \} \hat{a}_k \eta_r r_k, \quad \dot{\hat{b}}_{\beta_k} = I_{\beta_k} \{ \sigma_{kb} \} \hat{\beta}_k r_k
\end{aligned} \tag{2.53}$$

where  $\Gamma_k$ ,  $Q_k$ ,  $N_k$ ,  $T_k$ ,  $E_{\alpha_k}$ ,  $E_{\theta_k}$ ,  $E_{\beta_k}$ ,  $I_{\alpha_k}$ ,  $I_{\theta_k}$  and  $I_{\beta_k}$  are definite positive.

### 2.2.2.3 Virtual Prototype Validation via Co-simulation

In order to validate the effectiveness of the adaptive robust RBFNN control law, MATLAB/SIMULINK and ADAMS co-simulation on the virtual prototype constructed in ADAMS (see Fig. 2.3 (a)) is realized. In other words, MATLAB/SIMULINK is used as the control module to generate the RBFNN control law to dynamically control the virtual prototype in ADAMS. The MATLAB/SIMULINK and ADAMS co-simulation procedure is illustrated in Fig. 2.4.

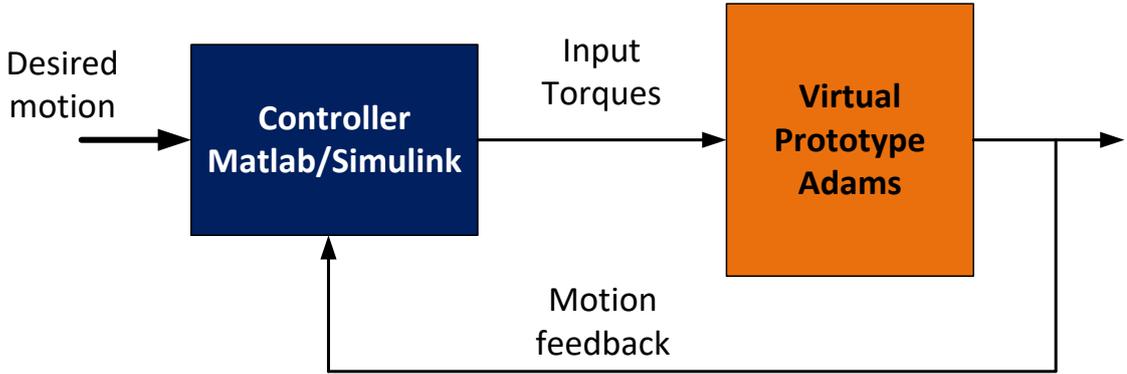


Figure 2.4 – MATLAB/SIMULINK and ADAMS co-simulation procedure.

Suppose that two driving wheels of the mobile platform and two joints of the waist are time-varying and the other joints are constant. Then, robot MDAMS

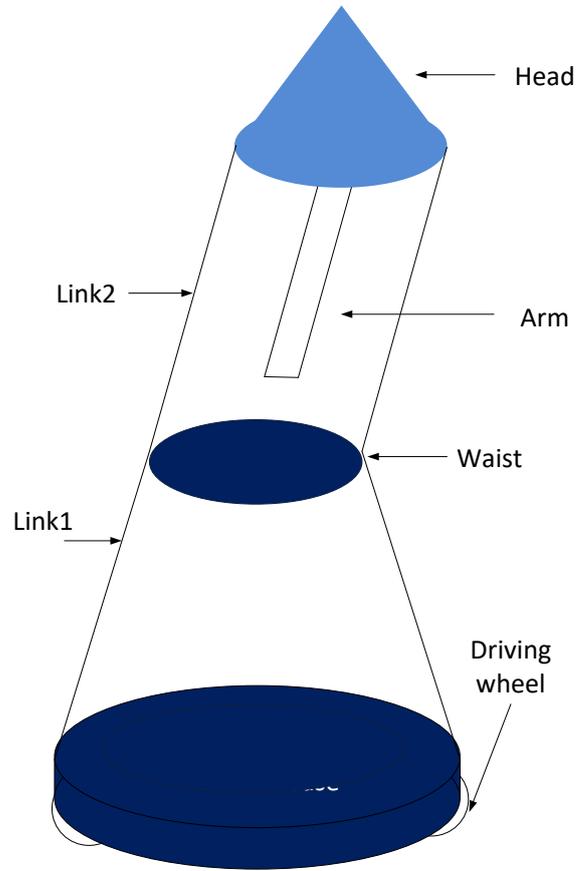


Figure 2.5 – Simplified model of robot MDAMS.

can be simplified as in Fig. 2.5. Assume that we have no knowledge about the robotic system and there is no payload. For each element of dynamic matrices  $\bar{M}$ ,  $\bar{F}$  and  $\bar{C}$ , a 200-node neural network is used. The gains for the controller are chosen as:  $\Gamma_k = \text{diag}[20]$ ,  $Q_k = \text{diag}[20]$ ,  $N_k = \text{diag}[20]$ ,  $T_k = \text{diag}[20]$ ,  $E_{\alpha_k} = E_{\theta_k} = E_{\beta_k} = \text{diag}[20]$ ,  $I_{\alpha_k} = I_{\theta_k} = I_{\beta_k} = \text{diag}[20]$  and  $K = 30$ .

Two motion tracking simulations (cases A and B) without and with consideration of disturbances are realized respectively. The simulation time is  $t = 15\text{s}$ . The trajectory for the mobile platform is designed as

$$[v_d \ \omega_d]^T = [0.1\pi\cos(0.1\pi t) \ 0.1\pi\cos(0.1\pi t)]^T,$$

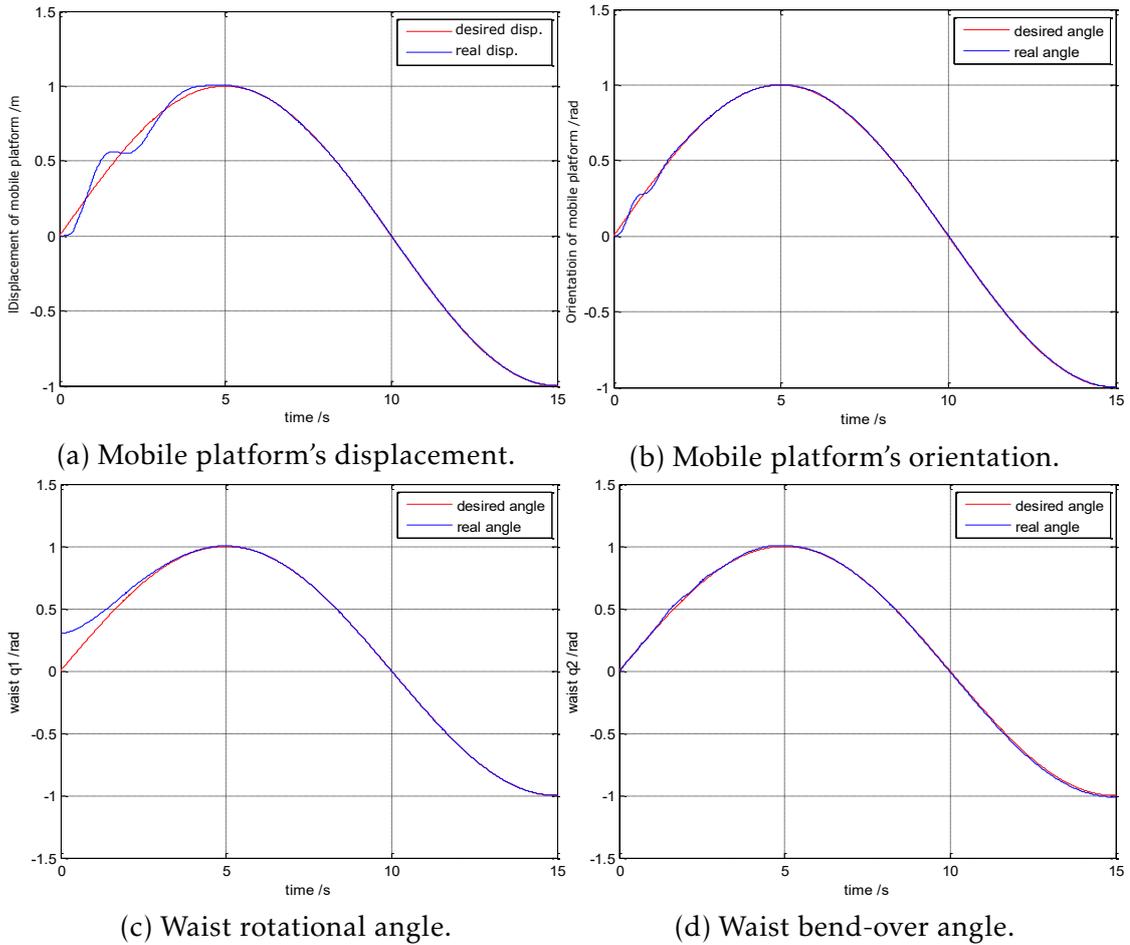


Figure 2.6 – Case A: position tracking results.

and the joint trajectory for the waist is designed as

$$[\theta_{\omega_{1d}} \ \theta_{\omega_{2d}}]^T = [\sin(0.1\pi t) \ \sin(0.1\pi t)]^T.$$

**Motion Tracking Without Disturbances** In case A, assume that there are no disturbances. The simulation results are shown in Figs. 2.6 and 2.7. In detail, Figs. 2.6 (a) and (b) show respectively the displacement and orientation tracking results of the mobile platform, Figs. 2.6 (c) and (d) show respectively the rotational and bend-over joint tracking results of the waist. Figs. 2.7 (a) and (b) show respectively the linear and angular velocity tracking results of the mobile platform, Figs. 2.7 (c) and (d) show respectively the two rotational and bend-over

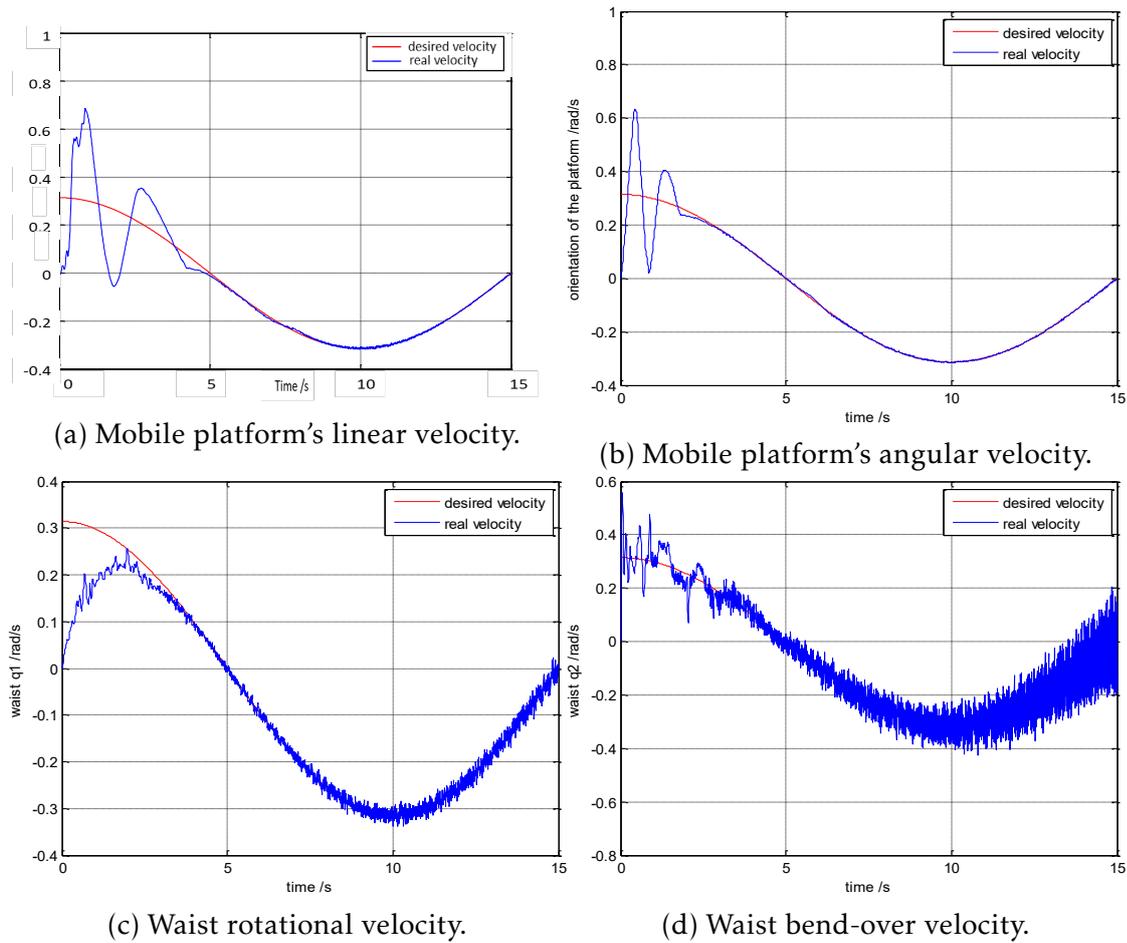


Figure 2.7 – Case A: velocity tracking results.

angular velocity tracking results of the waist. We can notice that robot MDAMS tracks perfectly the desired trajectories.

**Motion Tracking With Disturbances** In case B, a continuous disturbance is added to each joint. When  $t \geq 9$ ,  $\tau_d = [5 \ 5 \ 5 \ 5]^T$ , otherwise  $\tau_d = \mathbf{0}$ . The simulation results are shown in Figs. 2.8 and 2.9. In detail, Figs. 2.8 (a) and (b) show respectively the position and orientation tracking results of the mobile platform, Figs. 2.8 (c) and (d) show respectively the rotational and bend-over joint tracking results of the waist. Figs. 2.9 (a) and (b) show respectively the linear and angular velocity tracking results of the mobile platform, Figs. 2.9 (c) and (d) show respectively the two rotational and bend-over angular velocity tracking

results of the waist. We can see that robot MDAMS tracks perfectly the desired trajectories except for the perturbations at  $t = 9s$ .

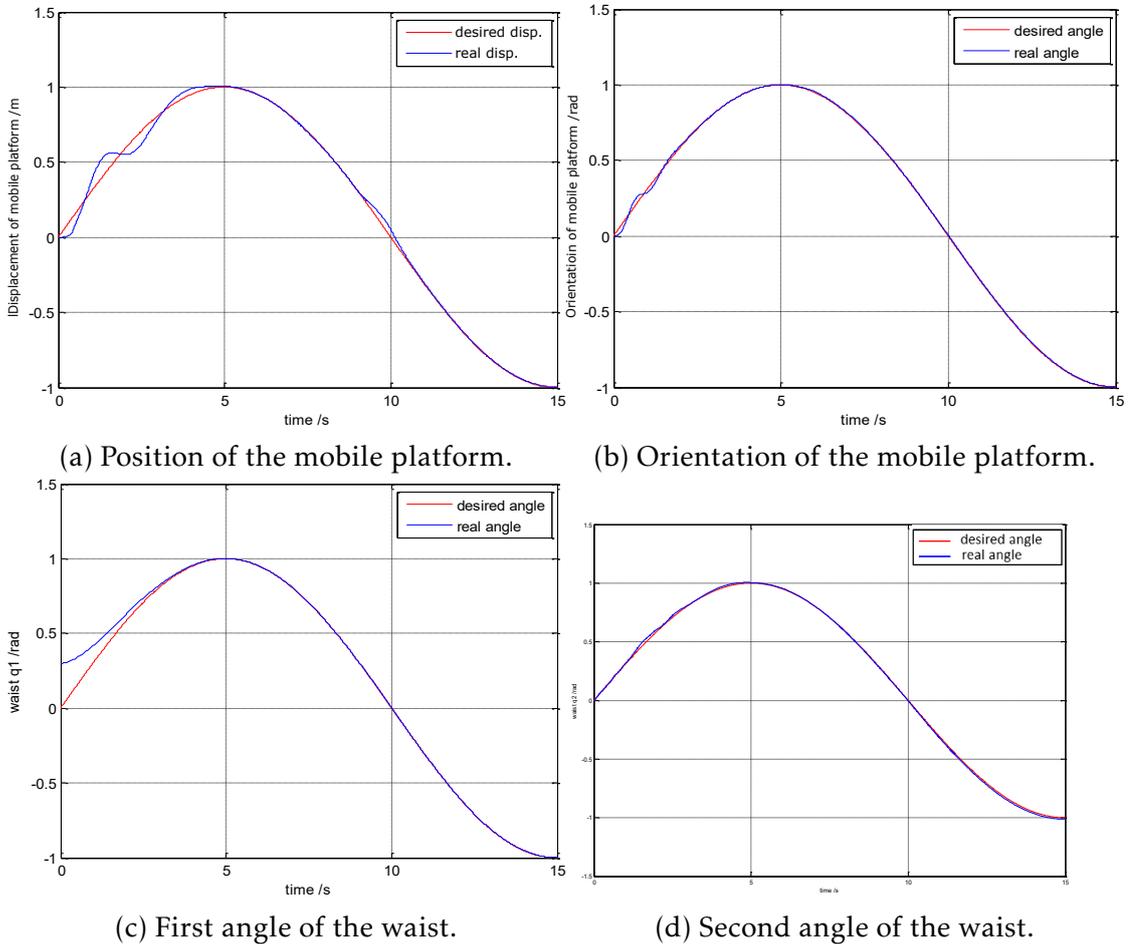


Figure 2.8 – Case B: position tracking results.

In order to compare the tracking performances of case A and case B, position tracking errors in two cases are shown in Fig. 2.10. We can see that the virtual prototype tracks perfectly the desired trajectories in both cases. The difference is that at  $t = 9s$  there are small perturbations. And after 2 seconds the tracking errors converge to zero and stable again.

In addition, the input torques in two cases are shown in Fig. 2.11. Fig. 2.11 (a) shows the input torques for the mobile platform and the waist rotational joint, and Fig. 2.11 (b) shows the waist bend-over input torque without disturbances.

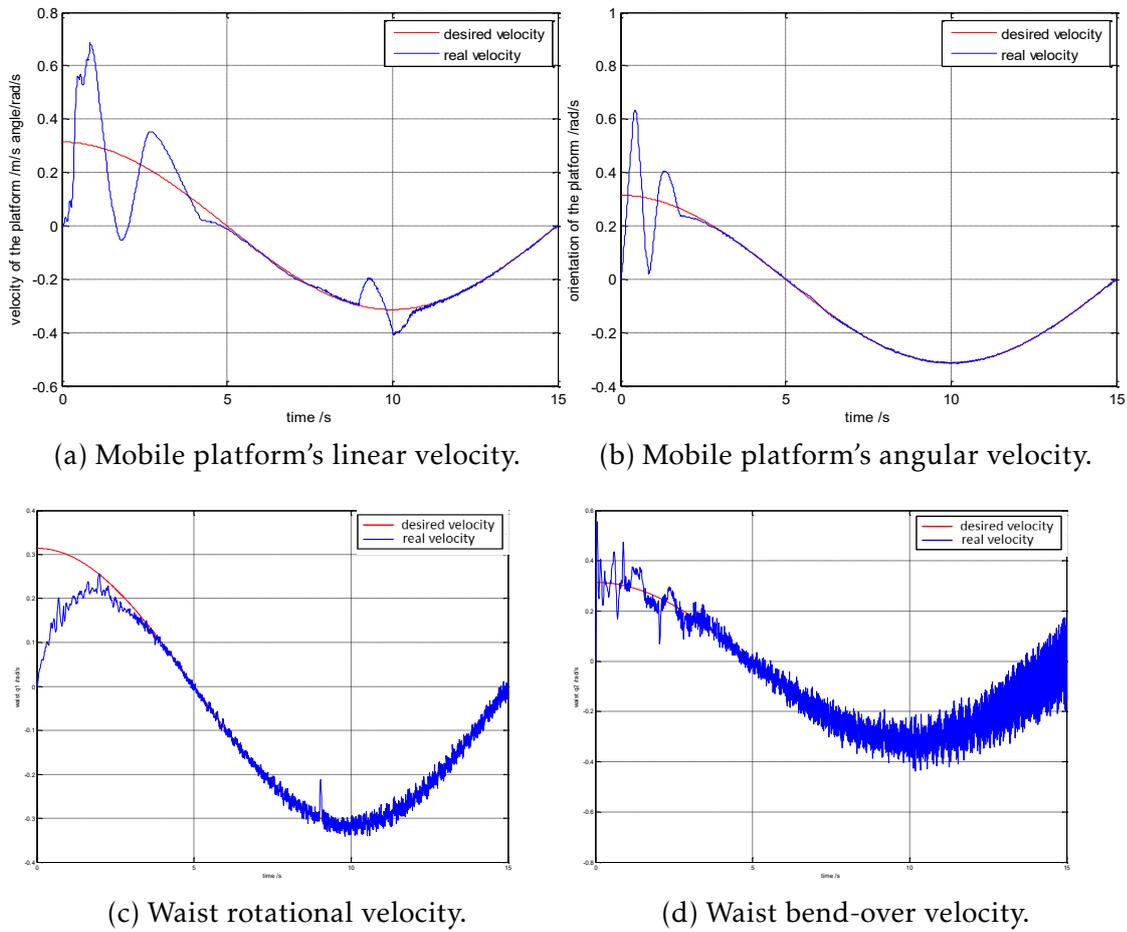


Figure 2.9 – Case B: velocity tracking results.

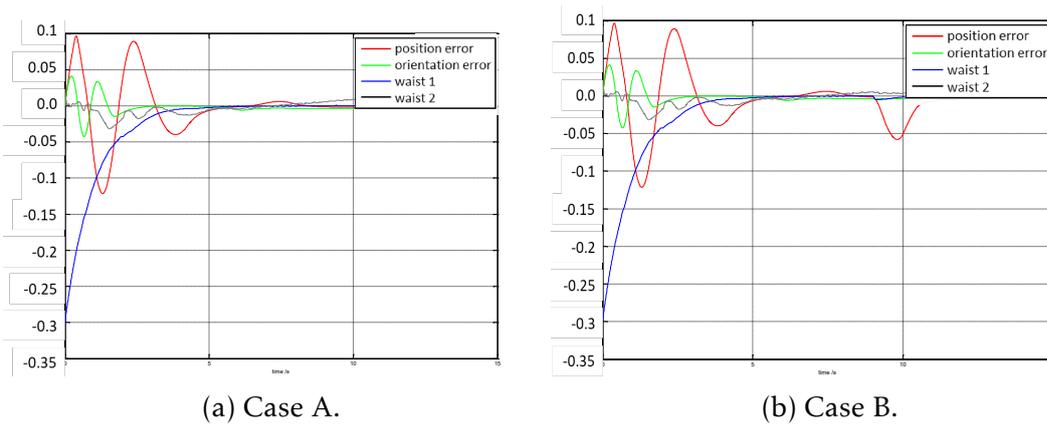
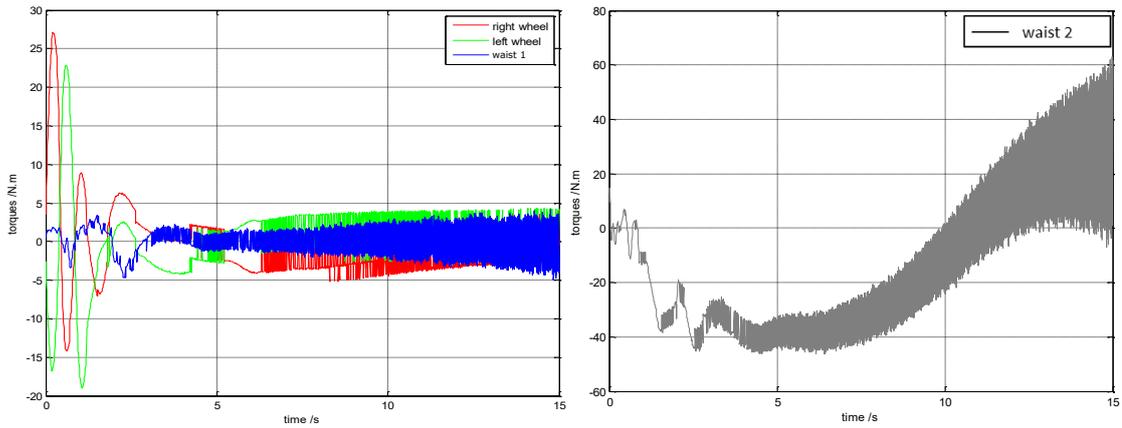
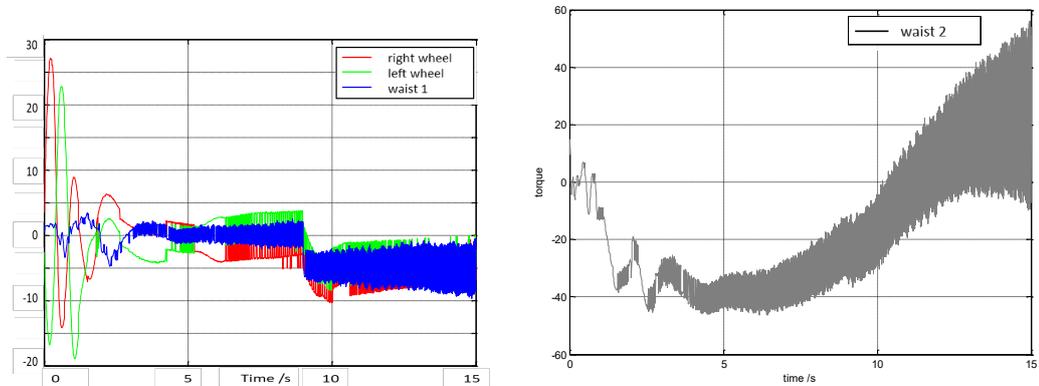


Figure 2.10 – Comparison of position tracking errors between cases A and B.



(a) Case A: mobile platform and waist rotational joint.

(b) Case A: waist bend-over.



(c) Case B: mobile platform and waist rotational joint.

(d) Case B: waist bend-over.

Figure 2.11 – Comparison of input torques between cases A and B.

Fig. 2.11 (c) shows the input torques for the mobile platform and the waist rotational joint, and Fig. 2.11 (d) shows the waist bend-over input torque under disturbances. We can see that the input torques in case B are the same as that in case A, except for a fault with an amplitude responding to the disturbances at  $t = 9s$ , which proves that the proposed controller manages well the disturbances. Several co-simulation shortcuts of the movements of the virtual prototype in ADAMS are illustrated in Fig. 2.12.

It can be clearly observed from the above simulation results that although the dynamic parameters are not completely known, the proposed RBFNN controller attenuates effectively the effects of uncertainties of the robot dynamics and

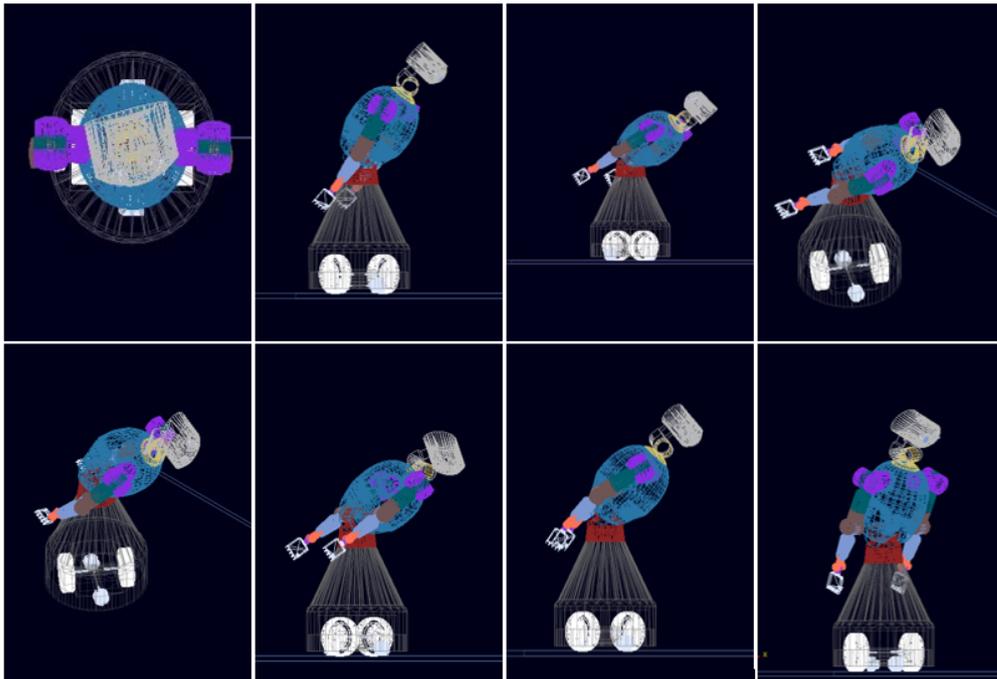


Figure 2.12 – Co-simulation shortcuts of virtual prototype in ADAMS.

disturbances.

#### 2.2.2.4 Conclusions

In this subsection, Lagrangian formulations are reformulated and a robust adaptive RBFNN controller is recalled to control the proposed robot MDAMS. The virtual prototype validation of the adaptive RBFNN controller is realized via MATLAB/SIMULATION and ADAMS co-simulation. The simulation results validate the effectiveness of the controller. However, as the number of DoFs increases, Lagrangian formulations will become complex and time-consuming. Moreover, the desired trajectory of the robot MDAMS is assumed to be known in advance. On the other hand, the motion planning for such a complex MDAMS is not easy both in task and joint spaces which will be investigated in Chapters 3 and 4.

### 2.2.3 Dynamic Modeling and Control using Kane's Method

In order to avoid calculating system energy and the partial derivatives and to facilitate the systematic assembly of additional dynamic systems like end-effectors, Kane's method [42] which was first introduced to model astronomical systems is used to model the proposed robot MDAMS.

#### 2.2.3.1 Dynamic Modeling using Kane's Method

Recall *Kane's Equations* (**Definition 1.6**)

$$F_r + F_r^* = 0, \quad r = 1, \dots, n \quad (2.54)$$

where  $F_r$  and  $F_r^*$  are the generalized active forces and the generalized inertia forces (**Definitions 1.4 and 1.5**), respectively.

Suppose that two wheels are pure rolling which means that the reaction forces of the ground to two wheels of mobile platform does not work. The generalized active forces can be expressed as

$$F_r \triangleq \sum_{i=1}^N [m_i g \vec{z}_0({}_r \vec{v}_i) + T_i \vec{z}_i({}_r \vec{\omega}_i)] \quad (2.55)$$

where  $T_3 = 0$ ,  $m_i$  is the link mass and  $T_i$  is the joint input torque.  $({}_r \vec{v}_i)$  and  $({}_r \vec{\omega}_i)$  are respectively the partial velocities (**Definition 1.3**) of linear and angular velocities  $\vec{v}_i$  and  $\vec{\omega}_i$  with respect to the  $r$ -th generalized velocity  $u_r$  according to Eq. (1.7). The partial velocity tables are constructed in Tables A.2 and A.3 in Appendix A.1.

In addition, the generalized active forces can be also written as follows:

$$F = D\tau + B \quad (2.56)$$

where  $F = [F_1 \quad \dots \quad F_n]^T \in \mathbb{R}^{n \times 1}$  is the generalized active force vector and

$$\tau = [T_1 \quad T_2 \quad T_4 \quad \dots \quad T_N]^T \in \mathbb{R}^{n \times 1} \quad (2.57)$$

is the input torque vector.  $D$  is a  $n \times n$  matrix which can be expressed as  $D(:, 1 : 2) = D_a(:, 1 : 2)$  and  $D(:, 3 : n) = D_a(:, 4 : N)$ .  $D_a \in \mathbb{R}^{n \times N}$  is defined as follows:

$$\{D_a\}_{ri} = \vec{z}_i({}_r\vec{\omega}_i), \quad r = 1, \dots, n, \quad i = 1, \dots, N. \quad (2.58)$$

Matrix  $B = \begin{bmatrix} B_1 & \dots & B_n \end{bmatrix}^T \in \mathbb{R}^{n \times 1}$ , with

$$B_r = \sum_{i=1}^N m_i g \vec{z}_0({}_r\vec{v}_i), \quad r = 1, \dots, n \quad (2.59)$$

where  $m_i$  is the mass of body  $B_i$ ,  $i = 1, \dots, N$ . Appendixes A.2 and A.3 list the detailed expressions of matrices  $D$  and  $B$ .

The generalized inertia forces  $F_r^*$  can be obtained using the following equation

$$F_r^* \triangleq \sum_{i=1}^N [-m_i \vec{a}_i^*({}_r\vec{v}_i) + (-I_i \vec{\epsilon}_i - \vec{\omega}_i \times I_i \vec{\omega}_i)({}_r\vec{\omega}_i)], \quad r = 1, \dots, n \quad (2.60)$$

where  $\vec{a}_i^*$ ,  $I_i$  and  $\vec{\epsilon}_i$  are respectively the acceleration of body  $i$ 's CoM, moment of inertia and angular velocity of body  $i$  in frame  $\Sigma_0$ . Then, the dynamic model is given as follows:

$$D\tau + B = -F^* \quad (2.61)$$

where  $F^* = \begin{bmatrix} F_1^* & \dots & F_n^* \end{bmatrix}^T \in \mathbb{R}^{n \times 1}$ . There are  $n$  unknowns and  $n$  equations, hence if the joint trajectories are known, the input torques for all joints can be numerically calculated. Besides,  $-F^*$  can be written as  $-F^* = M\dot{u} + B^*$ . Note  $C = B^* - B$ , the dynamic model can also be given as

$$M\dot{u} + C(u, \theta) = D\tau. \quad (2.62)$$

### 2.2.3.2 Controller Design using Back-stepping Technique

The back-stepping technique is used to design the control law. Rewrite Eq. (2.62) in the following form:

$$\begin{cases} \dot{\theta} = u \\ \dot{u} = M^{-1}(D\tau - C(u, \theta)) \end{cases} \quad (2.63)$$

Define the joint position tracking error as

$$z_1 = \theta - \theta_d \quad (2.64)$$

where  $\theta_d$  is the desired joint position. Differentiate  $z_1$  as

$$\dot{z}_1 = \dot{\theta} - \dot{\theta}_d = z_2 + \alpha - u_d \quad (2.65)$$

where  $z_2 = u - \alpha$  is the defined velocity tracking error,  $\alpha$  is the virtual controller to be determined. Choose the first Lyapunov function as:  $V_1 = \frac{1}{2}z_1^2$ , its differentiation is

$$\begin{aligned} \dot{V}_1 &= z_1 \dot{z}_1 \\ &= z_1(z_2 + \alpha - u_d) \end{aligned} \quad (2.66)$$

If choose

$$\alpha = -c_1 z_1 + u_d \quad (2.67)$$

where  $c_1 > 0$  is an adjustable parameter, inequality  $\dot{V}_1 = -c_1 z_1^2 + z_1 z_2 \leq 0$  is obtained. Then,

$$\dot{z}_2 = \dot{u} + c_1 \dot{z}_1 - \dot{u}_d. \quad (2.68)$$

Choose the second Lyapunov function as

$$V_2 = V_1 + \frac{1}{2}z_2^2 \quad (2.69)$$

and differentiate it as

$$\dot{V}_2 = -c_1 z_1^2 + z_1 z_2 + z_2 \dot{z}_2. \quad (2.70)$$

We expect that

$$\dot{z}_2 = -c_2 z_2 - z_1 \quad (2.71)$$

where  $c_2 > 0$  is an adjustable parameter, to obtain the following inequality

$$\dot{V}_2 = -c_1 z_1^2 - c_2 z_2^2 \leq 0. \quad (2.72)$$

Replace the left side of Eq. 2.68 with Eq. 2.71 as follows:

$$-c_2 z_2 - z_1 = \dot{u} + c_1 \dot{z}_1 - \dot{u}_d. \quad (2.73)$$

The joint acceleration can be expressed as

$$\dot{u} = \dot{u}_d - c_1 \dot{z}_1 - c_2 z_2 - z_1. \quad (2.74)$$

Then, substitute Eq. (2.74) into Eq. (2.62), the controller is given as follows:

$$\tau = D^{-1}(M(\dot{u}_d - c_2 z_2 - c_1 \dot{z}_1 - z_1) + C(u, \theta)). \quad (2.75)$$

From the above demonstration, the system will be stable in the sense of Lyapunov if the input control torque is chosen as Eq. (2.75).

### 2.2.3.3 Object Dynamics and Control

The object to be manipulated is assumed to be rigid and of known mass and inertia. Formulate the object dynamics [1] in a compact form:

$$M_o(X_o)\ddot{X}_o + C_o(X_o, \dot{X}_o) + G_o(X_o, \dot{X}_o) = F_o \quad (2.76)$$

where  $M_o$  is the inertia matrix of object,  $C_o$  is the Coriolis and centrifugal matrix,  $G_o$  is the vector of friction and gravity,  $F_o$  is the force and moment vector which exerts on the object's CoM. Suppose  $F_{oR}$  and  $F_{oL}$  are the force and moment vectors of two EEs which are applied to the object's CoM. Then, the force and moment vector on the object's CoM is:

$$F_o = F_{oR} + F_{oL} = J_{oR}^T F_{cR} + J_{oL}^T F_{cL} \quad (2.77)$$

where  $F_{cR}$  and  $F_{cL}$  are respectively the contact force and moment vectors of the right and left EEs.  $J_{oR}$  and  $J_{oL}$  are the contact matrices with  $J_{oi}^T = \begin{bmatrix} R_{ci} & 0_3 \\ \hat{r}_{ci}R_{ci} & R_{ci} \end{bmatrix}$  ( $i = R, L$ ).  $\hat{r}_{ci}$  is a skew-symmetric matrix.  $(r_{ci}, R_{ci})$  is the configuration of the contact frame  $\Sigma_i$  with respect to object frame.  $r_{ci} \in \mathbb{R}^{3 \times 1}$  is a distance vector from the center of object seen from the contact point of the  $i$ -th EE.

Therefore, the loading efforts on the EEs can be expressed as forces  $\vec{R}_e = F_{cR}$ ,  $\vec{R}_{e'} = F_{cL}$  and torques  $\vec{T}_e = 0$ ,  $\vec{T}_{e'} = 0$ . The contact force is given as follows:

$$F_{ci} = (-J_{oi}^T)^{-1} W_i (M_o \ddot{X}_{o_{re}} + C_o + G_o) \quad (2.78)$$

where  $W_i$  ( $i = R, L$ ) is the weighted matrix to distribute the resultant on the object by the two EEs.  $\ddot{X}_{o_{re}} = \ddot{X}_{od} + k_c(\dot{X}_{od} - \dot{X}_o) + k_p(X_{od} - X_o)$ , where  $X_{od}$  is the object desired motion,  $X_o$  is the object motion,  $k_c$  and  $k_p$  are the control gains. There are no inertia contributions, thus only the generalized active forces are modified. The contribution to the generalized active forces on the EEs can be derived as

$$F_{e_r} = F_{cR}(r\vec{v}_e) + F_{cL}(r\vec{v}_{e'}) \quad (2.79)$$

where  $r\vec{v}_e$  and  $r\vec{v}_{e'}$  are the partial velocities (**Definition 1.3**) of the linear velocities  $\vec{v}_e$  and  $\vec{v}_{e'}$  of two EEs with respect to the  $r$ -th generalized velocity  $u_r$  according to Eq. (1.7). As a result, Eq. (2.61) can be modified as

$$M\dot{u} + C_e(u, \theta) = D\tau \quad (2.80)$$

where  $C_e = C - B_e$  and  $B_e = \begin{bmatrix} F_{e_1} & \dots & F_{e_n} \end{bmatrix}^T \in \mathbb{R}^{n \times 1}$ . Then, the control law is proposed as follows based on Eq. (2.74):

$$\tau = D^{-1} \{M[\dot{u}_d - c_2 z_2 - c_1 \dot{z}_1 - z_1] + C_e(u, \theta)\}. \quad (2.81)$$

#### 2.2.3.4 Physical Stability Analysis

Different from many humanoid robots with a mobile platform, robot MDAMS introduced in this thesis consists of one moving waist with bend-over and rotational movements. When the robot's waist bends too much, there will be a

great possibility to tip over. In order to avoid tip-over, the following equality should be maintained

$$\left( \sum_{i=1}^n \tau_i + \Gamma + F_{cR} \times \vec{r}_e + F_{cL} \times \vec{r}_{e'} + \Gamma_g \right) \cdot \vec{e}_i = 0 \quad (2.82)$$

where  $\vec{e}_i = \vec{x}_0$  (or  $\vec{e}_i = \vec{y}_0$ ) and  $\Gamma = \sum_{i=1}^N ((-m_i \vec{a}_i^* + m_i g \vec{z}_0) \times \vec{r}_i^*)$ .  $\Gamma_g$  is the reaction torque of the ground which can only be known in real application. In fact, the ground can only passively respond to the movements of MDAMS. It is required to compensate the resulting tip-over active torque.

### 2.2.3.5 Simulation Results

Numerical simulations are realized to validate the constructed Kane's model. The virtual prototype of robot MDAMS is constructed in ADAMS (see Fig. 2.3). The main inertial properties of prototype MDAMS are listed in Table 2.2. The detailed parameters of the virtual prototype are shown in Fig. 2.1 and Table 2.3. In addition,  $r_1^* = 0cm$ ,  $r_2^* = 0cm$ ,  $r_3^* = 0cm$ ,  $r_6^* = r_{13}^*$ ,  $r_7^* = r_{14}^*$ ,  $r_8^* = r_{15}^*$ ,  $r_9^* = r_{16}^*$ ,

Table 2.3 – Link length and CoM position.

Length	$l_0$	$l_1$	$l_2$	$l_3$	$l_4$	$l_5$	$l_6$	$l_7$	$l_8$
[cm]	26.2	45	15	30	15	20	5	20	5
$l_9$	$l_{10}$	$r_4^*$	$r_5^*$	$r_6^*$	$r_7^*$	$r_8^*$	$r_9^*$	$r_{10}^*$	$r_{11}^*$
5	5	12.125	11.554	13.41	2.975	6.04	4.3	3.2	0.775

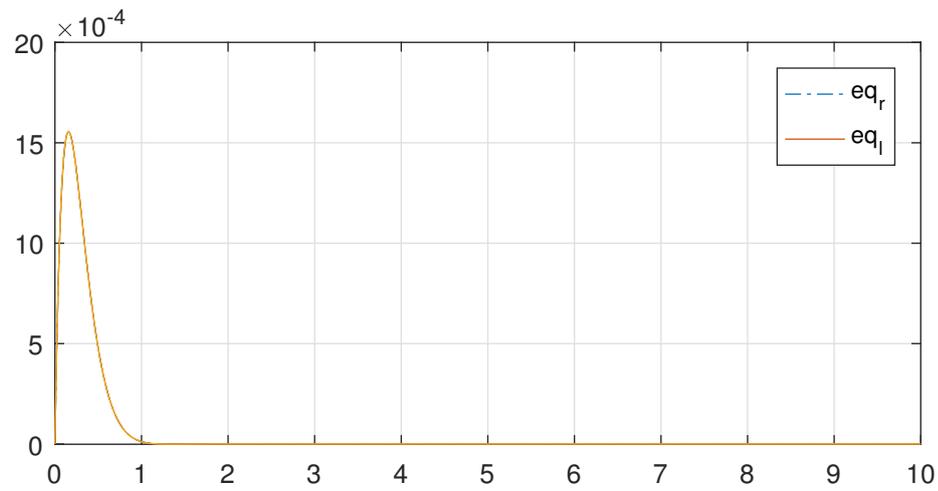
$$r_{10}^* = r_{17}^*, r_{11}^* = r_{18}^*, r_{12}^* = r_{19}^*.$$

The simulation time is  $t = 10s$ , the trajectory of the mobile platform is designed as

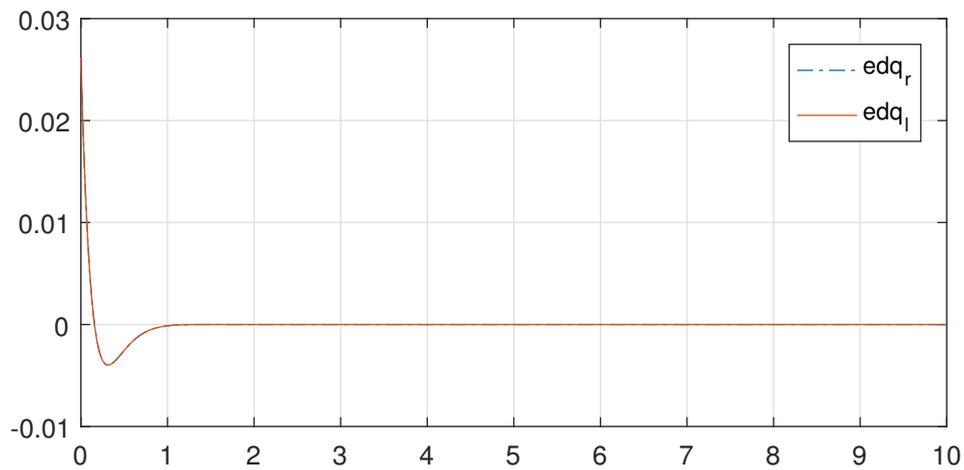
$$[q_r \ q_l]^T = [0.1 \sin(\pi/12t) \ 0.1 \sin(\pi/12t)]^T$$

and the trajectories for other joints are constant.

The controller in Eq. (2.75) is used for the trajectory tracking. The tracking results are shown in Fig. 2.13. We can see that the mobile platform tracks well the desired trajectory with the joint position tracking error converging to zero and stable in Fig. 2.13 (a) and the velocity tracking errors converging to zero and stable in Fig. 2.13 (b).



(a) Joint angle tracking error.



(b) Joint velocity tracking error.

Figure 2.13 – Joint trajectory tracking results.

In order to validate the constructed Kane's model, the same trajectory tracking simulation is realized using Lagrangian model as a comparison. Fig. 2.14 shows the input torque error between two models. It can be seen that the differences between the input torques for two wheels converge to zero stable.

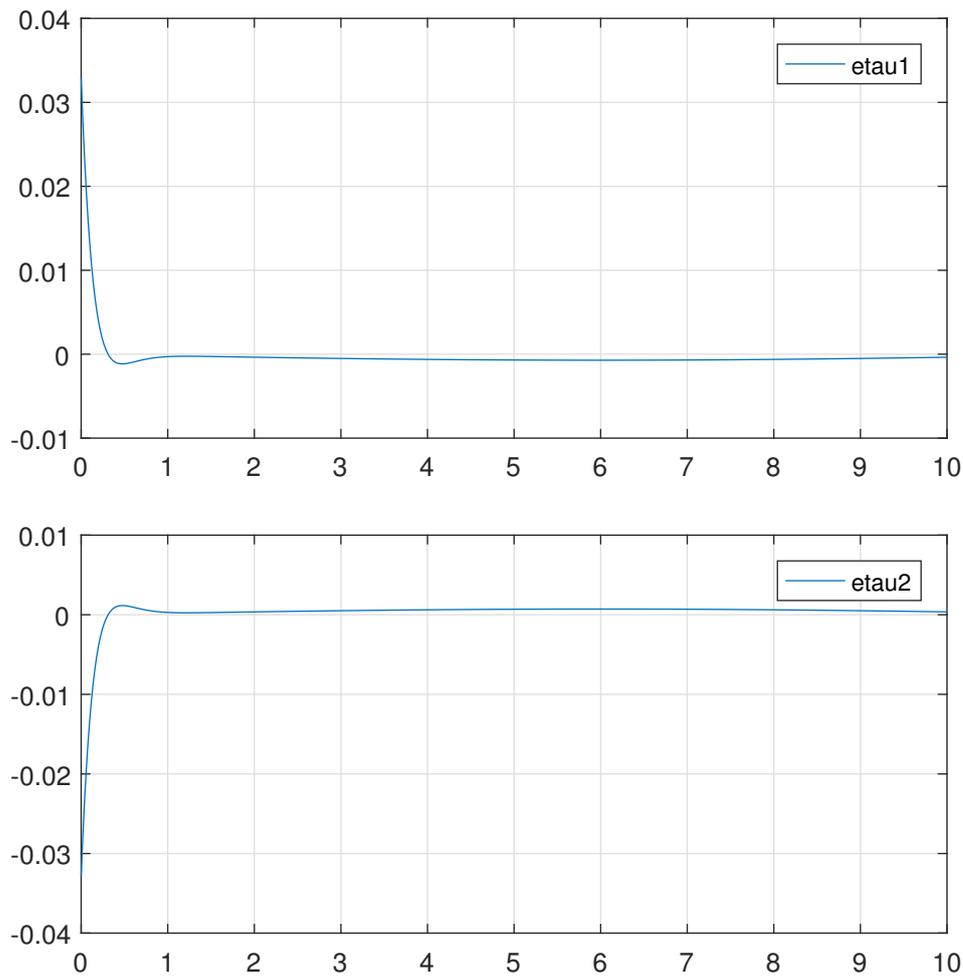


Figure 2.14 – Input torques errors between Kane’s and E-L methods.

## 2.3 CONCLUSIONS

In this Chapter, system description and forward kinematics are firstly presented. Then, Lagrangian formulations and an adaptive robust RBFNN controller for robot MDAMS are recalled. The virtual prototype validation via MATLAB/SIMULINK and ADAMS co-simulation is realized to validate the effectiveness of the RBFNN controller. However, Lagrangian formulations require calculating energy and the partial derivatives which will lose the efficiency when applied to complex high DoFs robots. In order to overcome this problem, Kane’s

method is used to dynamically model robot MDAMS. Kane's method greatly simplifies and improves the dynamic modeling process. It can be extended systematically to study object grasping and environmental interaction. In addition, physical stability based on the constructed Kane's model is analyzed which will be of great value to avoid tip-over in real application. Finally, a controller which is stable in the sense of Lyapunov is proposed using the back-stepping technique.

# MOTION PLANNING IN STATIC ENVIRONMENTS

## 3.1 INTRODUCTION

Motion planning, especially the coordination between mobile platform and upper manipulator system, is one of the main challenges in studying the proposed robot MDAMS. As reviewed in Chapter 1, many motion planning methods have been introduced [59–62]. Though they can be extended to multi-degrees of freedom (DoFs) systems, they are incapable to solve motion planning problem under multiple constraints like redundancy and singularity problems.

In order to overcome the shortcomings, many direct and inverse kinematics (IK)-based indirect motion planning methods with good results have been introduced for multi-DoFs manipulators [40, 70–72, 74, 76, 79, 84, 125, 129]. However, the EEs' movements generated by direct planning methods are difficult to predict. Moreover, the IK-based indirect methods solve the redundancy by optimizing a predefined objective function like potential-based mobile platform obstacle avoidance given task path in [84], scalar function minimization in [129] and maximum carrying capacity given task path in [125]. They usually require inverse Jacobian calculation and can only solve the optimization problems expressed by continuous objective functions. In addition, many reasoning methods, like analytical methods [86–88] and fuzzy logic methods [126, 131, 202], have

been presented to achieve human-like movements. However, specific consideration and rules for different scenarios are required which are challenging for high DoFs systems.

In order to deal with those shortcomings, many heuristic methods represented by genetic algorithms (GAs) have been proposed [90, 91, 93–95]. GA [64] has no constraints on the cost function's continuity, has no need of inverse Jacobian computation, is capable of optimizing (maximizing and minimizing at the same time) multiple objectives, etc. By defining a combined fitness function, multi-objective GAs (MOGAs) take various constraints into consideration at the same time. For instance, positioning error of EE and joint displacement in [90], displacements of mobile platform and joint in [91], joint displacement and EE's reaching error in [93], navigation length, path-obstacles intersection and accumulated change of mobile platform orientation in [95], etc. But, none of them take EE's orientation, or relative motion, or human-like pose (mobile platform's position-orientation and upper manipulator's configuration at the same time) into account for a humanoid mobile manipulator.

In addition, the combined fitness function-based MOGAs search for the solutions only in one certain direction. In order to preserve diversity, Srinivas [203] proposed a Non-dominated Sorting-based GA (NSGA) which searches for the optimal individual in multiple directions. It generates a number of optimal solutions to form a Pareto-optimal front. In order to overcome the limitation of a predefined sharing parameter of the NSGA approach, Deb [99] proposed an improved version NSGA-II by introducing a fast-non-dominated-sort algorithm and a crowding distance assignment method. However, the selected solutions at each generation are not well distributed. Pires [92] introduced the MaxiMin sorting scheme into the NSGA-II algorithm to improve the distribution in the Pareto-optimal front to preserve the diversity. However, only the last selected set is sorted according to MaxiMin sorting scheme, while the previous sets are not. Therefore, we propose an improved MaxiMin NSGA-II algorithm. At each generation, not only the last selected set but all the sets in the selected population are sorted using MaxiMin sorting scheme. In this way, the probability of inheriting good genes is improved as far as possible at each generation.

See from the literature, the existing methods usually design the collision-

free motion from an initial position to a known target position. Very often, an optimization problem of certain user-defined objective functions, e.g. path length, reaching accuracy, energy consuming, carrying capacity and obstacles avoidance, is formed. In this Chapter, only the EE's goal position of robot MDAMS is needed, the goal pose and the trajectories in task space or joint space are unknown. The objective is to make robot MDAMS be as much as possible "human-like" in both their appearance and movements to make humans feeling comfortable when working with it.

Two problems: human-like optimal *mobile platform position-orientation and the upper manipulator system configuration* (i.e. *pose*) design and approaching motion planning with obstacles avoidance in static environments, are solved given the desired positions and orientations of two EEs. Fig. 3.1 is an illustration of the motion planning problem for robot MDAMS. The position and orientation of the object to be manipulated are supposed to be known. The initial pose (black skeleton) of MDAMS represented by the black skeleton is known. The final pose (blue skeleton) to manipulate the red object on the table and the corresponding approaching trajectories  $Traj_R$ ,  $Traj_L$  and  $Traj_{MB}$  are to be designed.

Section 3.2 describes the proposed motion planning method. Firstly, an improved MaxiMin NSGA-II **Algorithm 2** is designed in Section 3.2.1 for planning the optimal pose of the proposed robot MDAMS given only the goal positions of EEs. Five objective functions are defined to be optimized simultaneously to achieve human-like behaviors. The position-orientation of mobile platform and the configuration of upper manipulator are taken into account at the same time. The reaching accuracy of EE, the displacement of EE in task space and the joint displacement are also taken into account. Apart from the traditional absolute displacement in task space, a new measure - displacement of EE in the robotic body frame - is defined. In addition, an objective function which characterizes the manipulability of EE is defined to avoid configuration singularity and to increase the potential of executing subsequent tasks. A normalized combined-fitness function is defined and a post decision maker is introduced to settle the preferred *pose* among the candidates in the Pareto-optimal front. Then, an off-line motion planning **Algorithm 3** is presented in Subsection 3.2.2. In order to improve the performance of traditional RRT methods, a direct-connect

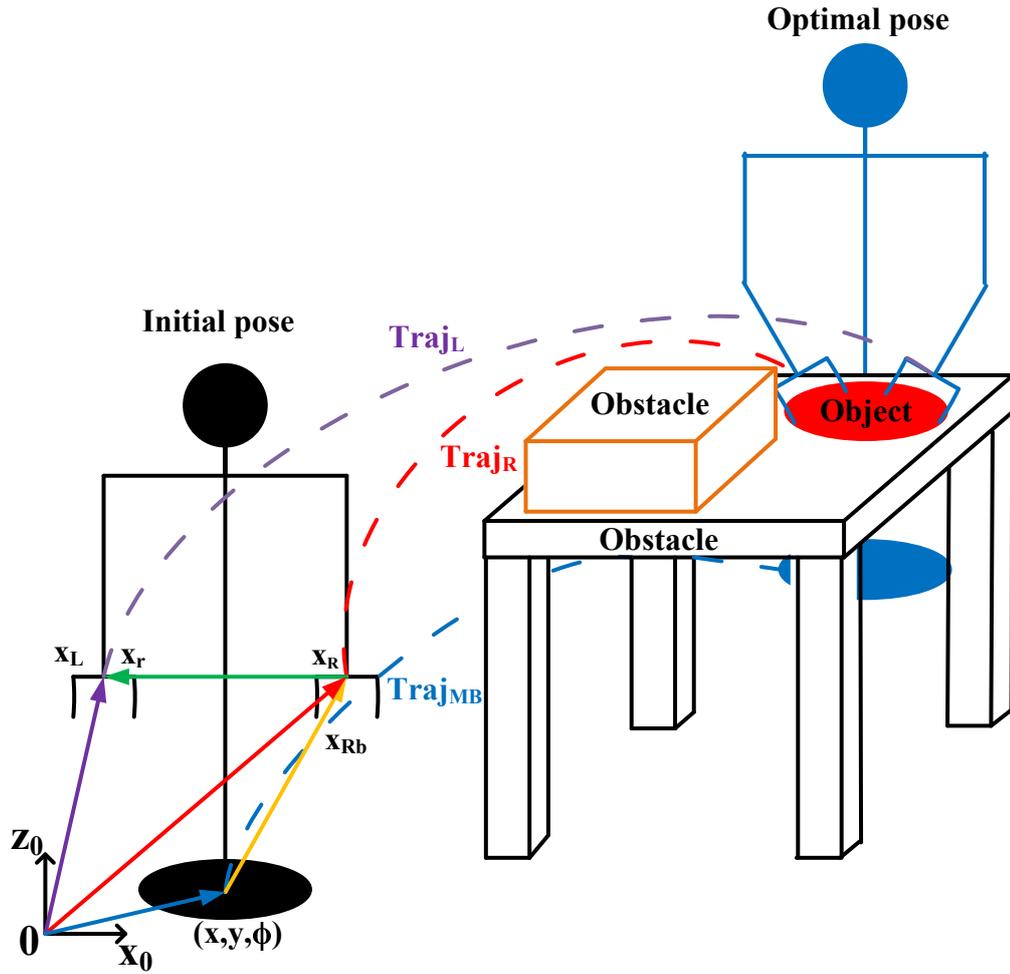


Figure 3.1 – An example of motion planning problem for MDAMS.

Bidirectional RRT and Gradient descent sampling process is designed. Besides, a geometric optimization method pruning the sampled path via node rejection and adjustment is designed to guarantee always the shortest consistent path for repeated tasks. Compared to [56] and [95], the proposed geometric method is simpler and more intuitive. And the methods in [55] and [56] need to calculate the complicated medial axis which is not practical for high DoFs systems. Moreover, different from [105] which uses a projection process to locate the collision-free points around obstacles, the proposed method in this Chapter can easily locate them. Besides, the human-like head forward movements are realized by assign a sequence of reasonable orientations which are calculated

based on the planned via-points of the mobile platform. Linear polynomials with parabolic blends interpolation is used to generate time-specified trajectory. Numerical simulations are presented in Section 3.3 to validate the effectiveness of the proposed planning method. Conclusions are given in Section 3.4.

## 3.2 MAXIMIN-BASED NSGA-II MOTION PLANNING

In this Chapter, a motion planning method consists of optimal *pose* design and approaching motion planning for MDAMS is proposed under the following assumption.

**Definition 3.1 (Pose and Optimal pose)** *Throughout this thesis, we note pose as the position-orientation of mobile base and the configuration of upper manipulator (i.e.  $\theta$ ) for the designed MDAMS.  $\theta_{op}$  denotes the optimal pose which reaches EEs' desired positions-orientations under multiple constraints.*

**Assumption 3.1** *The initial pose of robot MDAMS is known, and the desired positions and orientations of the two EEs are known or can be obtained according to the object to be manipulated.*

### 3.2.1 Improved MaxiMin NSGA-II Algorithm for Pose Design

GAs solves the optimization problems based on the natural genetics and selection mechanics [64, 204]. A GA allows a population composed of many individuals (a set of parameters are used to define each individual which is called the chromosome) to evolve under specified selection rules to a state that maximizes the fitness function. The GA modifies repeatedly the population of individuals (possible solutions). At each generation, a set of individuals are selected, then, the crossover and mutation operators are used to reproduce new individuals from existing ones. The crossover operator combines two individuals (parents) to generate new individuals (offspring) for the next generation. The mutation operator introduces random changes and reintroduces genetic diversity to the population. Over successive generations, the population evolves towards an optimal solution.

### 3.2.1.1 Objective Functions Definitions

In order to reach the desired positions and orientations of two EEs, there exist numerous joints combinations. According to the task requirements, there is always a preference among those joints combinations. In our work, EE's tracking accuracy in task space, manipulability ability and displacements in task and joint spaces are considered. Therefore, five corresponding objective functions are defined in the following.

**EE's Position Tracking Accuracy** The objective function characterizing EE's position tracking accuracy is firstly defined as follows:

$$f_1 = \|x_R - x_{Rd}\| + \|x_L - x_{Ld}\| \quad (3.1)$$

where  $x_{Rd}$ ,  $x_{Ld}$  and  $x_R$ ,  $x_L$  are respectively the desired and real positions of the right and left EEs.

**EE's Orientation Tracking Accuracy** Recall the orientation tracking error  $e_q \in \mathbb{R}^4$  (see Eq. (1.5)), the second objective function is defined as:

$$f_2 = \|e_{q_R}\| + \|e_{q_L}\| \quad (3.2)$$

where  $e_{q_R}$  and  $e_{q_L}$  are respectively the orientation tracking errors of right and left EEs.

**EE's Manipulability** Apart from the reaching accuracy in task space, additional criteria are required to obtain satisfactory behavior of MDAMS. The manipulability ability describes the distance away from the singular configuration of mechanical systems. Hence, it can be used to optimize the potential ability of robot for subsequent tasks.

The often used manipulability criterion is defined as  $\Omega = \sqrt{\det(JJ^T)}$  or  $\Omega = \frac{\sigma_{\max}(J)}{\sigma_{\min}(J)}$ , where  $J$  is the Jacobian of EE and  $\sigma$  is the singular value of  $J$ . Besides, if there is a preference in one certain direction  $\mathbf{d}$ , the directional manipulability

[205] (e.g. Eq. 4.8) can be used. Then, the third objective function is given as

$$f_3 = -(\Omega_R + \Omega_L) \quad (3.3)$$

where  $\Omega_R$  and  $\Omega_L$  are respectively the manipulability measure of the right and left EEs.

**Joint Displacement** In order to save energy, the least joint displacement is required. Therefore, a mass-based joint displacement objective function is proposed

$$f_4 = \frac{1}{n_\kappa} \sum_{i=1}^{n_\kappa} (W_i \frac{\theta_i - \theta_{imin}}{\theta_{imax} - \theta_{imin}}) \quad (3.4)$$

which means that the most heavy part has the most difficulty to move.  $n_\kappa$  is the number of DoFs of the upper manipulator system,  $\theta_i$  is joint position, and  $\theta_{imax}$  and  $\theta_{imin}$  are joint boundaries.  $W_i = \frac{M_i}{\sum_{j=1}^{n_\kappa} M_j}$  is mass-based weight, where  $M_i = \sum_{k=i}^{n_e} m_k$ ,  $m_k$  is the mass of the  $k$ -th link, and  $n_e$  is the end joint index.

**EE's Displacement** The optimization of the above-defined objective functions is not enough for the proposed humanoid robot MDAMS. Thus, the fifth objective function related to EE's displacement is designed as follows.

*EE's displacement in world frame  $\Sigma_0$*  is defined as

$$f_{5a} = \|x_{aRf} - x_{aRi}\| + \|x_{aLf} - x_{aLi}\| \quad (3.5)$$

where  $x_{aRi}$  and  $x_{aRf}$  ( $x_{aLi}$  and  $x_{aLf}$ ) are respectively the initial and final positions of right (left) EE in frame  $\Sigma_0$ .

*EE's displacement in robotic body frame  $\Sigma_3$*  is defined as

$$f_{5b} = \|x_{bRf} - x_{bRi}\| + \|x_{bLf} - x_{bLi}\| \quad (3.6)$$

where  $x_{bRi}$  and  $x_{bRf}$  ( $x_{bLi}$  and  $x_{bLf}$ ) are respectively the initial and final positions of right (left) EE in frame  $\Sigma_3$ .

Therefore, the fifth objective function is given as

$$f_5 = f_{5a} + f_{5b}. \quad (3.7)$$

For now, all the objective functions ( $f_1, \dots, f_{n_{obj}}, n_{obj} = 5$ ) for robot MDAMS have been defined. In order to fix the solution in the Pareto-optimal front, the following fitness function is defined.

$$\min_{p_i \in P_{t+1}} z_i = \sum_{j=1}^{n_{obj}} w_{ij} z_{ij}, \quad i = 1, \dots, N_{pop} \quad (3.8)$$

where  $z_i$  is the fitness function of each individual  $p_i \in P_{t+1}$ ,  $w_{ij}$  is the weighting coefficient satisfying  $\sum_{j=1}^{n_{obj}} w_{ij} = 1$  and  $z_{ij}$  is the normalized  $j$ -th objective function of individual  $p_i$ .

### 3.2.1.2 Improved MaxiMin NSGA-II Algorithm

Based on **Assumption** 3.1, the objective here is to plan the optimal *pose* for robot MDAMS given only the desired positions of EEs. The searching intervals for the upper manipulator system are listed in Table 5.2. The chromosome is chosen as  $G = (x, y, \phi, \theta_\omega, \theta_R, \theta_L)$  and its searching interval locates in the collision-free configuration space  $C_{free}$ .

The whole process is illustrated in Fig. 3.2 and **Algorithm** 2. At the beginning, a parent population  $P_0$  of size  $N_{pop}$  is randomly created (line 1). Sort all the individuals  $\{G_i, i = 1, \dots, N_{pop}\}$  in  $P_0$  according to their fitness values (line 2, see Eq. (3.8)). The technique *BasicGA*, which employs the usual binary tournament selection, recombination and mutation operators, is used to create the first offspring population  $Q_0$  of size  $N_{pop}$  (line 3). The generation numeration is initialized  $\iota = 0$ .

Then, a **while** loop is activated to evolve the Pareto-optimal solutions (lines 4 through 30). At the beginning of each generation  $\iota$ , a combined population  $R_\iota = P_\iota \cup Q_\iota$  of size  $2N_{pop}$  is formed (line 5). In  $R_\iota$ , get the individuals which have the minimum value for each objective function (lines 6 through 9) and initialize the next parent population  $P_{\iota+1}$  (line 10). The rest individuals in

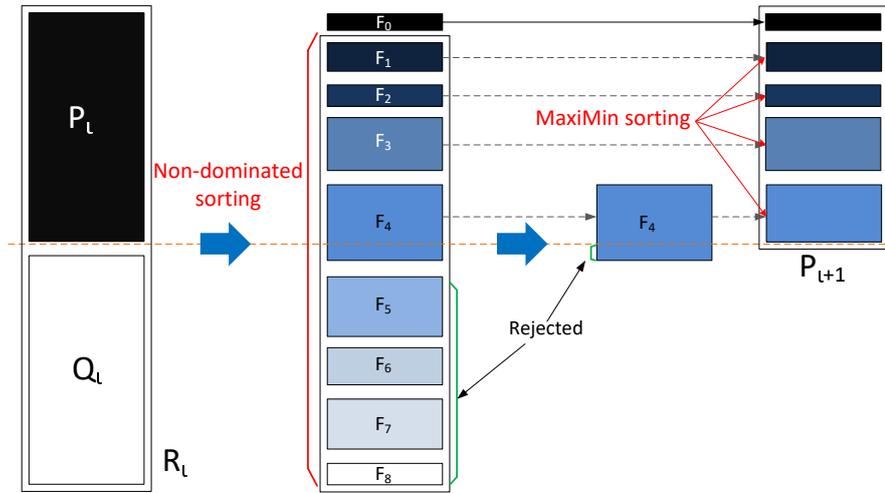


Figure 3.2 – Flowchart of the improved MaxiMin NSGA-II algorithm.

$R_t$  are sorted using the fast non-dominated sorting scheme according to their non-domination<sup>24,28</sup>.

Afterwards, select the best non-dominated individuals set by set, and sort each set using the MaxiMin sorting scheme. In particular, if the number of individuals in the selected population  $P_{t+1}$  and in the following set  $F_i$  is smaller than  $N_{pop}$  ( $\#P_{t+1} + \#F_i \leq N_{pop}$ ), continue the selection (lines 13 through 17); if not ( $\#P_{t+1} + \#F_i > N_{pop}$ ), select the individuals one by one using the MaxiMin sorting scheme until obtaining  $N_{pop}$  selected individuals (loops **for** and **while**, lines 18 through 20 and lines 21 through 27). Finally, the next offspring population is produced while keeping the order of the selected population  $P_{t+1}$  and go to the next generation ( $t = t + 1$ , line 29).

The MaxiMin sorting scheme is described here.<sup>29</sup> Firstly, the distance between each non-dominated individual  $p_j \in F_i$  ( $j = 1, \dots, \#F_i$ ) and the individuals already selected  $p_k \in P_{t+1}$  ( $k = 1, \dots, \#P_{t+1}$ ) is evaluated and save the minimum distance  $c_{p_j}$  for each individual  $p_j \in F_i$  (loop **for**, lines 18 through 20). Then, move the individual  $p_j \in F_i$  whose minimum fitness distance  $c_{p_j}$  in  $F_i$  is maximum to  $P_{t+1}$  (lines 22 through 23). Every time an individual  $p$  in  $F_i$  enters in  $P_{t+1}$ , the values  $\{c_{p_j}, j = 1, \dots, \#F_i\}$  in  $F_i$  are reevaluated (lines 24 through 26).

The main improvement of **Algorithm 2** is that the individuals in each non-dominated set  $F_i$  are sorted using the MaxiMin sorting scheme (line 14) com-

**Algorithm 2** Improved MaxiMin NSGA-II

---

```

1: Initialization : rand( $P_0$ ) {initialize the population randomly}
2: sort( $P_0$ ) {sort  $P_0$  according to Eq. (3.8)}
3:  $Q_0 = \text{BasicGA}(P_0)$  {usual GA reproduction}
    $\iota = 0$  {initialize the generation numeration}
4: while  $\iota \leq N_G$  do
5:    $R_\iota = P_\iota \cup Q_\iota$  {merge population to size  $2N_{pop}$ }
6:    $F_0 = \phi$  {initialize selected set  $F_0$ }
7:   for  $i = 1$  to  $N_{obj}$  do
8:      $F_0 = F_0 \cup \text{GetMin}(R_\iota, i)$  {move the individual with the minimum objec-
       tive function  $f_i$  from  $R_\iota$  to  $F_0$ }
9:   end for
10:   $P_{\iota+1} = F_0$  {initialize the parent population  $P_{\iota+1}$  for the next generation}
11:  fast-non-dominated-sort( $R_\iota$ ) {non-dominated sorting}
12:   $i = 1$  {get the first non-dominated set  $F_i$  in the sorted population  $R_\iota$ }
13:  while  $\#P_{\iota+1} + \#F_i \leq N_{pop}$  do
14:     $F_i = \text{MaxiMin-sort}(F_i)$  {sort  $F_i$  using the MaxiMin sorting scheme}
15:     $P_{\iota+1} = P_{\iota+1} \cup F_i$  {add  $F_i$  to the population  $P_{\iota+1}$ }
16:     $i = i + 1$  {increment of the non-dominated set counter}
17:  end while
18:  for  $j = 1$  to  $\#F_i$  do
19:     $c_{p_j} = \min_{\forall p_k \in P_{\iota+1}} \{\|f_{p_j} - f_{p_k}\|\}$  {calculate the minimum fitness distance  $c_{p_j}$ 
      and assign it to each individual in  $F_i$ }
20:  end for
21:  while  $\#P_{\iota+1} < N_{pop}$  do
22:     $p = \text{getMaxCi}(F_i)$  {get the individual with the largest minimum fitness
      distance in  $F_i$ }
23:     $P_{\iota+1} = P_{\iota+1} \cup p$  {add  $p$  to the selected population  $P_{\iota+1}$ }
24:    for  $l = 1$  to  $\#F_i$  do
25:       $c_{p_l} = \min_{\forall p_k \in P_{\iota+1}} \{\|f_{p_l} - f_{p_k}\|, c_{p_l}\}$  {update the minimum fitness distance
        of each individual in  $F_i$ }
26:    end for
27:  end while{Complete  $P_{\iota+1}$  selection using the MaxiMin sorting scheme}
28:   $Q_{\iota+1} = \text{make-new-pop}(P_{\iota+1})$  {keep the individuals' orders in the selected
    population  $P_{\iota+1}$  and use crossover and mutation operators to produce the
    next offspring population}
29:   $\iota = \iota + 1$  {increment of the generation counter}
30: end while

```

---

pared with the MaxiMin NSGA-II algorithm in Pires 05. Therefore, good genes are inherited as much as possible by increasing the crossover possibility of good individuals. As a result, the converging speed of optimal solution is largely increased.

What is more, **Algorithm 2** generates a group of solutions hence forming a *Pareto-optimal front* [93] at each generation. The individuals in the *Pareto-optimal front* conflict with each other. This thesis uses a posterior method to select the optimal solution in the *Pareto-optimal front*, in other words, a decision maker based on Eq. (3.8).

### 3.2.2 Motion Planning in Static Environments

In this Subsection, the off-line point-to-point collision-free motion planning **Algorithm 3** is proposed to lead robot MDAMS from the initial *pose* to the designed optimal *pose* in static environments.

#### 3.2.2.1 Overview

Throughout the proposed **Algorithm 3**, the path planner maintains two trees  $Tr_s$  and  $Tr_e$  which root respectively at the initial node  $q_{init}$  and final node  $q_{goal}$  (see Fig. 3.3).

A **while** loop is activated up till tree  $Tr_s$  and tree  $Tr_e$  meet each other, i.e.  $DisTree \leq DisMax$  ( $DisTree$  is the minimum distance between  $Tr_s$  and  $Tr_e$ ,  $DisMax$  is a predefined value) or there is a direct collision-free connection between  $Tr_s$  and  $Tr_e$ .

During each **while** loop, the bidirectional RRT (BiRRT) or Gradient descent extending process in  $C_{free}$  is selected (lines 3 through 10) based on the comparison between a randomly generated value  $p_{rb}$  (line 4) and a predefined value  $p_{rbM}$  ( $0 \leq p_{rbM} \leq 1$ ) which is proportional to the clutter of the environment. *BiRRT-Extend* (line 6) is the BiRRT extending process, and *GradDecExtend* (line 8) is the gradient descent extending process. The orange dash line and nodes  $\{q_{init}, q_{rand}, q_{new}\}$  in Fig. 3.3 illustrate the conventional RRT extending process [206]. *GradDecExtend* extends two trees directly along the linking line between two trees. A direct-connect method is proposed to stop the sampling phase even if

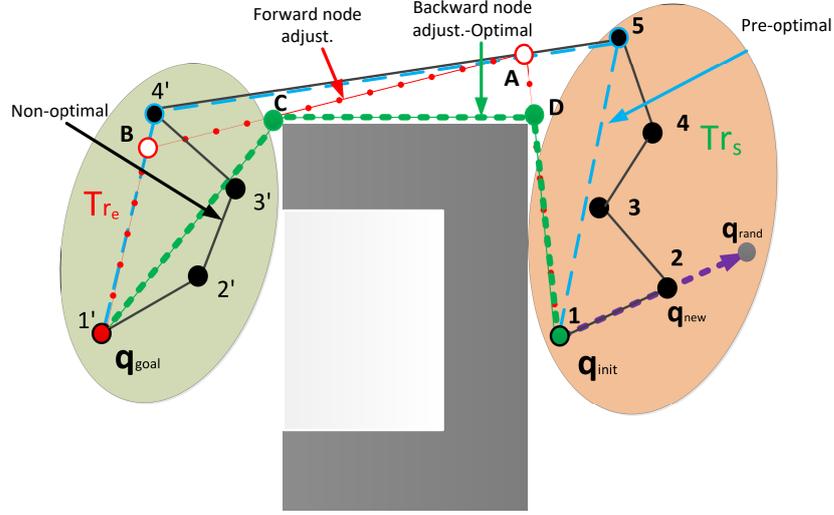


Figure 3.3 – An example of off-line path planning.

the two trees  $Tr_s$  and  $Tr_e$  are still very far away from each other. It links  $Tr_s$  and  $Tr_e$  directly (segment 5-4') once there is a collision-free connection between two arbitrary nodes in  $Tr_s$  and  $Tr_e$  respectively, thus improving the path planning process by greatly reducing the sampled nodes.

Process *PathRRTGrad* (line 11) finds the path in  $Tr_s$  and  $Tr_e$  from  $q_{init}$  to  $q_{goal}$  by linking two trees as one single tree which roots at  $q_{init}$ .

Then, process *GeomOptim* (line 12) realizes the geometric optimization of the designed path by applying the *node rejection* and *node adjustment* techniques which will be presented in Subsection 3.2.2.2.

Finally, the process *InterPolyBlend* (line 13) which will be introduced in Subsection 3.2.2.3 designs a time law for the optimal path obtained by *GeomOptim* within a defined time interval  $[t_{start}, t_{end}]$ .

**Remark 3.1** As shown in Fig. 3.3, tree  $Tr_s$  and tree  $Tr_e$  are connected directly even though the distance between them is still very large (segment 5-4'), thus accelerating the sampling phase greatly especially in large scale environments compared with conventional sampling methods. In addition, the direct-connect BiRRT and Gradient descent extending processes guarantee the fast path planning in non-crowded

**Algorithm 3** Direct-connect BiRRT and Gradient Descent Motion Planning

---

```

1: Initialization :
2:  $Tr_s \leftarrow$  start-node,  $Tr_e \leftarrow$  end-node
   Loop Process:
3: while  $DisTree > DisMax$ , or there is no direct collision-free connection do
4:    $p_{rb} \leftarrow rand()$ 
5:   if  $p_{rb} < p_{rbM}$  then
6:      $[Tr_s, Tr_e] = BiRRT-Extend(Tr_s, Tr_e)$  {bidirectional RRT extending in
        $C_{free}$ }
7:   else
8:      $[Tr_s, Tr_e] = GradDecExtend(Tr_s, Tr_e)$  {gradient descent extending in
        $C_{free}$ }
9:   end if
10: end while
11:  $Path \leftarrow PathRRTGrad(Tr_s, Tr_e)$  {non-optimal path}
12:  $OptimPath \leftarrow GeomOptim(Path)$  {geometric optimization}
13:  $Motion \leftarrow InterPolyBlend(OptimPath, t_{start}, t_{end})$ 
14: end

```

---

*environments, and at the same time avoids obstacles and local optimums in crowded environments.*

### 3.2.2.2 Geometric Path Optimization: *GeomOptim*

An example of the direct-connect BiRRT and Gradient descent-based path planning result is shown in Fig. 3.3 (the black line linking nodes  $\{1(q_{init}), \dots, 5, 4', \dots, 1'(q_{goal})\}$ ). However, the designed path is usually non-optimal due to the random node sampling process, thus a geometric path optimization method is proposed. It consists of the *node rejection* and *node adjustment* processes which are detailed in the following. Note that the path obtained after *node rejection* is the *pre-optimal path*, and the path after *node adjustment* is the *optimal path*. The blue, red and green dash lines in Fig. 3.3 illustrate the method.

**Node Rejection** The *node rejection* process consists of the following steps:

- i) Initialize the *pre-optimal path* with  $q_{init}$ ;

- ii) Test all the nodes along the designed non-optimal path i.e., black line in sequence, and select the last node **5** which path does not collide with obstacles into the *pre-optimal path*;
- iii) Repeat step ii) until reaching  $\mathbf{q}_{goal}$ .

The blue dash line which consists of nodes  $\{\mathbf{q}_{init}, \mathbf{5}, \mathbf{4}', \mathbf{q}_{goal}\}$  in Fig. 3.3 represents the *node rejection* result, i.e. the *pre-optimal path*.

**Node Adjustment** The obtained *pre-optimal path* after *node rejection* is usually non-optimal. In order to further optimize the path, a forward-backward *node adjustment* process on the *pre-optimal path* is proposed. In detail, the *node adjustment* process contains the following steps:

- i) Introduce an *auxiliary path*, and initialize the *auxiliary path* with  $\mathbf{q}_{init}$ ;
- ii) Test in sequence all the segments which link node  $\mathbf{q}_{init}$  and one of the points on the second segment  $\{\mathbf{5}-\mathbf{4}'\}$  (i.e. the point moves from **5** to **4'** with a fixed step on the segment  $\{\mathbf{5}-\mathbf{4}'\}$ ), and reserve the last node **A** on segment  $\mathbf{5}-\mathbf{4}'$  which does not collide with obstacles into the *auxiliary path* instead of the second node **5**;
- iii) Repeat step ii) until reaching  $\mathbf{q}_{goal}$ , then a path which consists of nodes  $\{\mathbf{q}_{init}, \mathbf{A}, \mathbf{B}, \mathbf{q}_{goal}\}$  is obtained;
- iv) Flip the obtained path and repeat steps i)-iii);
- v) Flip the obtained path again and obtain the *optimal path* containing nodes  $\{\mathbf{q}_{init}, \mathbf{D}, \mathbf{C}, \mathbf{q}_{goal}\}$ .

The red dash line in Fig. 3.3 is the forward *node adjustment* result. The green dash line is the backward *node adjustment* result which is also the geometrically optimized path containing a sequence of *via-points*  $\{\mathbf{q}_{init}, \mathbf{D}, \mathbf{C}, \mathbf{q}_{goal}\}$ .

**Remark 3.2** Geraerts [56] pointed out that many motion planning techniques generate low quality paths, the author presented a number of techniques to improve the quality of paths. A redundant nodes-remove technique is designed with a retraction

process to the medial axis and a branches-remove method. However, it needs complicated geometric calculation of medial axis and the implementation to the articulated robots remains to be studied, and the similar problem with [55]. Based on the proposed motion planning method, the planned via-points can be adjusted within their neighborhoods to increase the clearance avoiding calculating the complicated medial axis.

### 3.2.2.3 Interpolating Linear Polynomials With Parabolic Blends

In fact, the *OptmPath* obtained by running lines 1 through 12 in **Algorithm 3** is a geometric path, so a time law remains to be applied. Besides, the non-holonomic robot MDAMS is expected behave like a human being. In other words, robot MDAMS should always head forward along each segment. Therefore, process *InterPolyBlend* which employs the technique of linear polynomials with parabolic blends interpolation [54] is used (line 13) to design the trajectory within time interval  $[t_{start}, t_{end}]$  which is described in the following.

Consider the case when it is required to interpolate  $n_p$  via-points  $\{\mathbf{q}_1, \dots, \mathbf{q}_{n_p}\}$  at time instants  $t_k$  ( $k = 1, \dots, n_p$ ,  $t_1 = t_{start}$ ,  $t_{n_p} = t_{end}$ ), respectively. To avoid discontinuity problems on the first derivative at the time instants  $t_k$ , the function  $\mathbf{q}(t)$  shall have a parabolic profile (blend) around  $t_k$ . As a consequence, the entire trajectory is composed by a sequence of linear and quadratic polynomials, which in turn implies that a discontinuity on  $\ddot{\mathbf{q}}(t)$  is tolerated. The time scope between two adjacent via-points is  $\Delta t_k \propto dis(\mathbf{q}_k, \mathbf{q}_{k+1})$

$$\Delta t_k = \frac{t_{end} - t_{start}}{\sum_{k=1}^{n_p-1} dis(\mathbf{q}_k, \mathbf{q}_{k+1})} dis(\mathbf{q}_k, \mathbf{q}_{k+1}) \quad (3.9)$$

where  $k = 1, \dots, n_p - 1$ . The velocity  $v_{k,k+1}$  along each line polynomial segment and the acceleration  $a_k$  along each parabolic blend are

$$\mathbf{V}_{k,k+1} = \frac{\mathbf{q}_{k+1} - \mathbf{q}_k}{\Delta t_k}, \quad (3.10)$$

and

$$\mathbf{a}_k = \frac{\mathbf{V}_{k,k+1} - \mathbf{V}_{k-1,k}}{\Delta t'_k} \quad (3.11)$$

respectively, where  $\Delta t'_k$  is the  $k$ -th parabolic blend duration. The first and last segments deserve special care with the parabolic blend duration being  $\Delta t'_1/2$  and  $\Delta t'_{n_p}/2$  respectively.

**Trajectory Planning for Mobile Platform** Suppose that the obtained path by **Algorithm 3** for the mobile platform contains  $n_p$  via-points  $(x_j, y_j, \phi_j)$  ( $j = 1, 2, \dots, n_p$ ) of which  $(x_1, y_1, \phi_1) = (x_i, y_i, \phi_i)$  and  $(x_{n_p}, y_{n_p}, \phi_{n_p}) = (x_f, y_f, \phi_f)$  are respectively the initial and final states and others are the intermediate ones. The mobile platform's initial orientation is  $\phi_i$  and designed final orientation is  $\phi_f$ . The heading orientation  $\phi_k$  along each path segment is expressed as

$$\phi_k = \langle (x_k, y_k), (x_{k+1}, y_{k+1}) \rangle, k = 1, \dots, n_p - 1 \quad (3.12)$$

to always lead the robot MDAMS head forward.  $\langle \cdot, \cdot \rangle$  is the orientation angle. As a result, the orientation list  $(\phi_i, \phi_1, \dots, \phi_{n_p-1}, \phi_f)$  is obtained.

The objective is to design the trajectory which connects the position  $(x_j, y_j)$  using the linear polynomials with parabolic blends interpolation. At each path point  $(x_j, y_j)$ ,  $j = 1, 2, \dots, (n_p - 1)$ , firstly it rotates to the heading orientation  $\phi_k$ ,  $k = 2, \dots, n_p - 1$  while translating along the  $k$ -th parabolic blend in  $\Delta t_k$ , then moves to the next point  $(x_{k+1}, y_{k+1})$ . The mobile platform will rotate to the final desired orientation  $\phi_f$  after reaching the desired position.

#### 3.2.2.4 Upper Manipulator Trajectory Planning

In this Subsection, the trajectory planning for the upper manipulator system is investigated. Suppose that the initial and optimal joint positions are respectively  $\theta_{ji}$  and  $\theta_{jf}$ ,  $j = (3, 4, \dots, n_k)$ , the total navigation and manipulation execution time  $T$ . By employing the fifth-order polynomials interpolation technique, the joint trajectory can be expressed as

$$\theta_j = \begin{cases} \theta_{ji}, t \in [0, t_\theta] \\ \theta_{jf} + (\theta_{jf} - \theta_{ji})S(\tau), t \in [t_\theta, T] \end{cases} \quad (3.13)$$

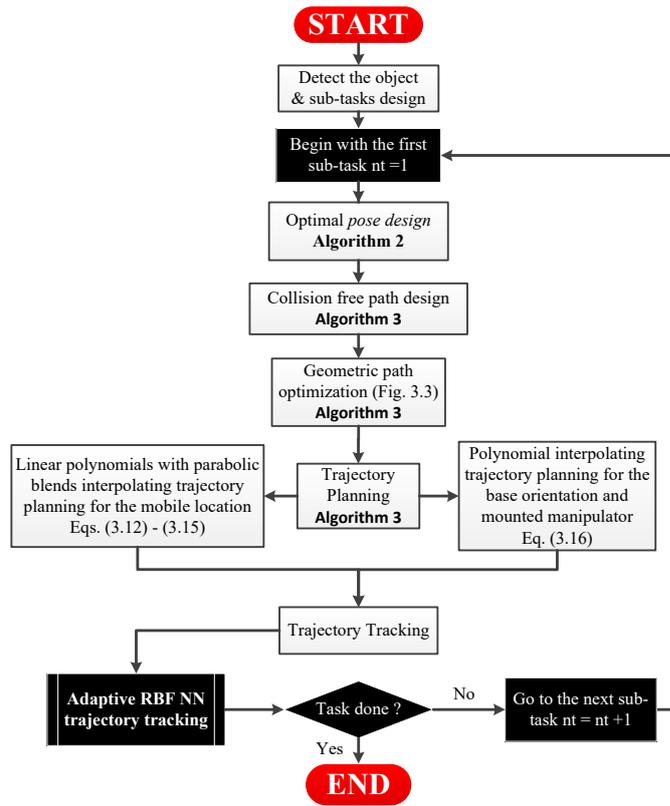


Figure 3.4 – MDAMS motion planning flow chart.

where  $S(\tau)$  ( $0 \leq S(\tau) \leq 1$ ) is the interpolation function. Choose  $S(\tau)$  as a fifth-order polynomial  $S(\tau) = 6\tau^5 - 15\tau^4 + 10\tau^3$ , and define

$$\tau = \frac{t - t_\theta}{T_\theta} \quad (3.14)$$

where  $T_\theta = T - t_\theta$ ,  $t$  is the actual time and  $t_\theta$  is the moment when the upper manipulator begins to execute.

The whole strategy can be summarized in Fig. 3.4. It contains four processes: i) object detection and subtask design, ii) optimal *pose* design for each sub-task, iii) path and trajectory planning for each sub-task, and (iv) trajectory tracking.

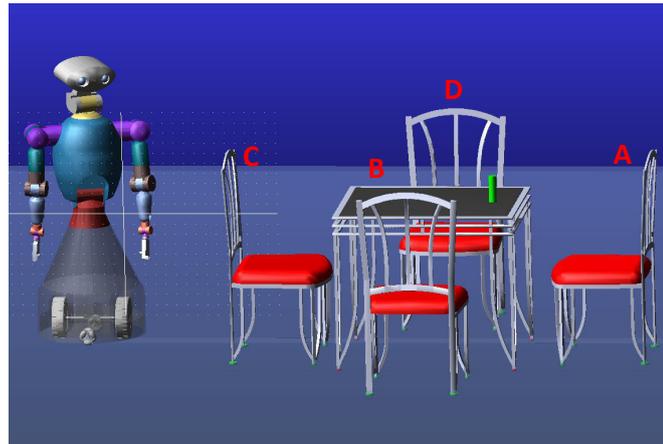


Figure 3.5 – Domestic environment with one table and four chairs.

### 3.3 SIMULATION RESULTS

Let us consider a virtual domestic environment with one table being surrounded by four chairs (A, B, C and D) in Fig. 3.5. The robot locates initially near to chair C with the initial pose. Suppose that the desired positions of two EEs are known, the desired position-orientation of the mobile platform and the desired configuration of the upper manipulator are unknown. The objective is to design the optimal pose to manipulate one of the chairs using **Algorithm 2**, to plan the collision-free motion to reach the optimal pose using **Algorithm 3** and to track the designed motion following the flow chart in Fig. 3.4.

In the following, two motion planning tasks 1 and 2 are realized for the proposed robot MDAMS to manipulate respectively chairs A and D using MATLAB R2016b in a PC (Intel(R) Core(TM) i5-3317U CPU @ 1.70GHz 1.70 GHz, 4,00Go, x64). In addition, one tracking task is realized via MATLAB/SIMULINK and ADAMS co-simulation on the prototype in ADAMS using the adaptive robust RBFNN controller proposed in Chapter 2.

#### 3.3.1 Task 1: Chair A

In order to validate the effectiveness of the proposed **Algorithms 2** and **3**, the first task is the motion planning for MDAMS to manipulate chair A.

### 3.3.1.1 Optimal Pose Design

The proposed robot MDAMS locates initially at  $x_M = [-0.2, 0.4]$  and the initial positions of two EEs are respectively  $x_R = [-0.2, 0.25, -0.219]$  and  $x_L = [-0.2, 0.55, -0.219]$ . Suppose that the desired positions of two EEs' are respectively  $x_{Rd} = [2.7, 0.3, 0]$  and  $x_{Ld} = [2.7, -0.3, 0]$  around chair A. Based on the robot's dimension, the expected position and orientation of mobile base are calculated as  $(x_{me}, y_{me}) = (3, 0)$  and  $\phi_{de} = 180^\circ$ , respectively. Set the initial pose of MDAMS as  $\theta_{init} = (-0.2, 0.4, 0, \mathbf{0}_{16})$ . The objective is to design the optimal pose for robot MDAMS while reaching the desired positions  $x_{Rd}$  and  $x_{Ld}$  of two EEs. The table and chairs are treated as cuboid obstacles.

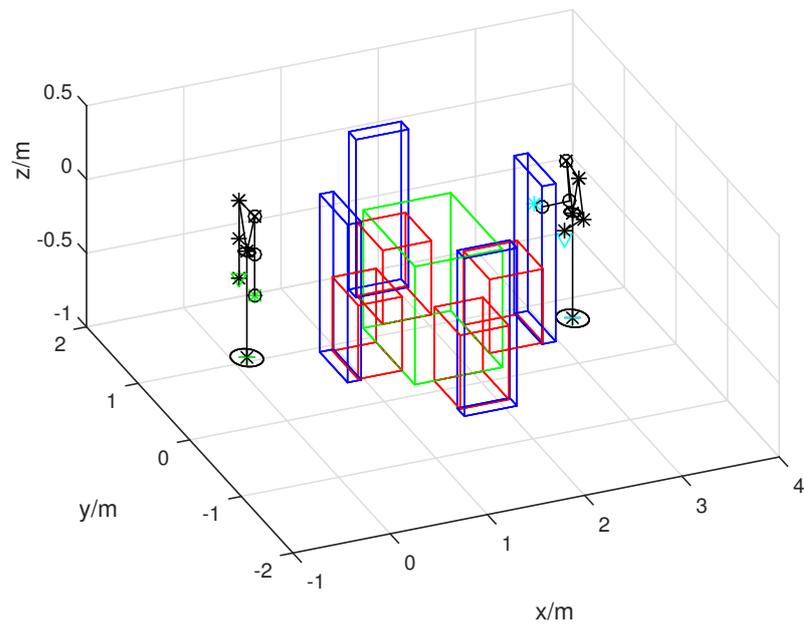
Set the genetic crossover and mutation probabilities as  $p_c = 0.7$  and  $p_m = 0.3$ , respectively. The population size is chosen as  $N_{pop} = 100$ , the generation number is  $N_G = 120$ , and the decision maker coefficients in Eq. (3.8) are set as  $w = (0.54, 0.02, 0.02, 0.02, 0.4)$ . After  $N_G$  generations, the designed optimal pose is shown in Table 3.1 and Fig. 3.6 (a).

Table 3.1 – The designed optimal pose  $\theta_{op}$  [ $m, deg$ ].

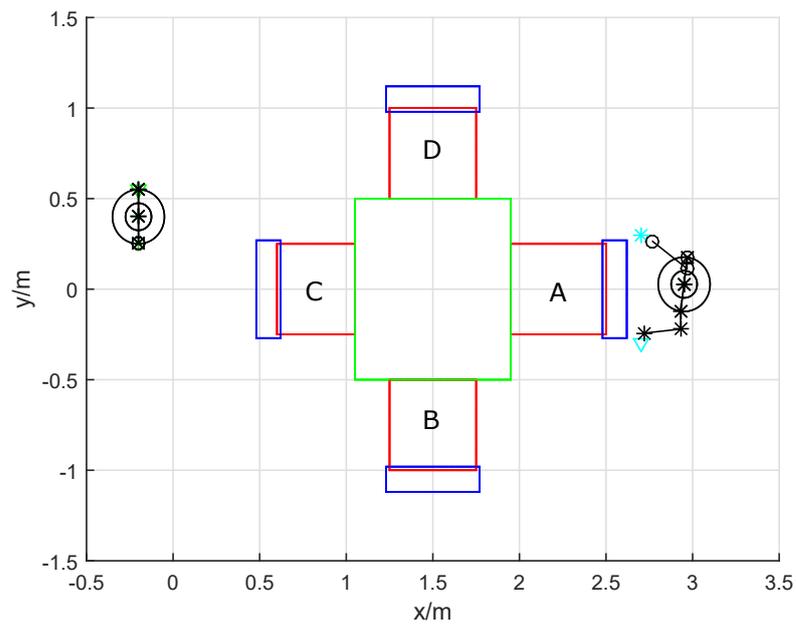
$\theta_m$ & $\theta_w$	2.95	0.027	166.77	5.90	0.24	-	-
$\theta_R$	1.11	13.73	33.19	74.63	3.85	49.38	4.81
$\theta_L$	2.77	22.07	8.19	64.46	67.65	67.63	23.91

It can be seen that the designed optimal position of mobile platform is  $(x_{md}, y_{md}) = [2.9512, 0.0269]$  which is around the expected position  $(x_{me}, y_{me})$ . The designed optimal orientation of mobile platform is  $\phi_d = 166.77^\circ$  which is around the expected orientation  $\phi_{de}$ .

Fig. 3.6 shows the motion planning results. The green rectangle represents the table, the red and blue rectangles represent four chairs. The green plus, star and triangle points represent the initial positions of the mobile platform, the right and left EEs, respectively. The cyan plus, star and triangle points represent the designed optimal positions of the mobile platform, the right and left EEs, respectively. The black skeletons represent the initial and optimal poses of robot MDAMS. It can be seen that the designed optimal pose is reasonable and with good reaching accuracy.

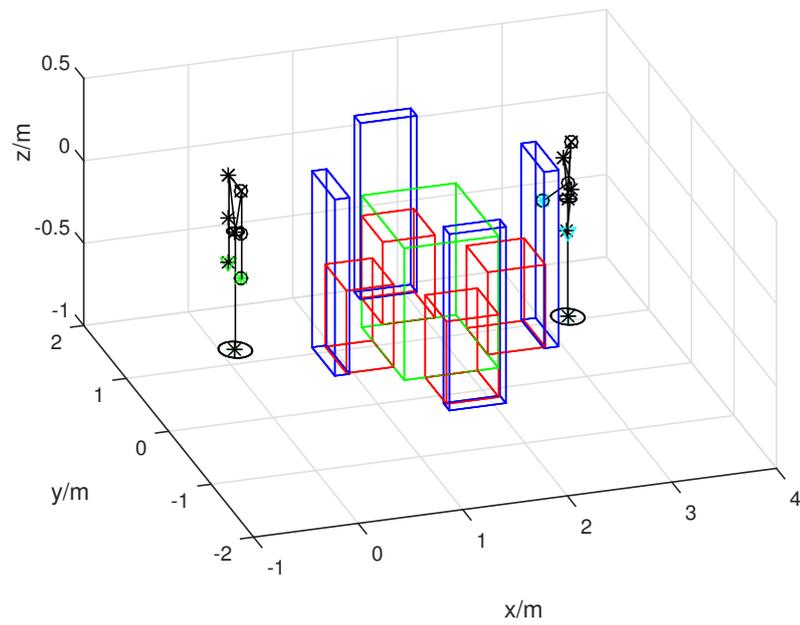


(a) 3D view.

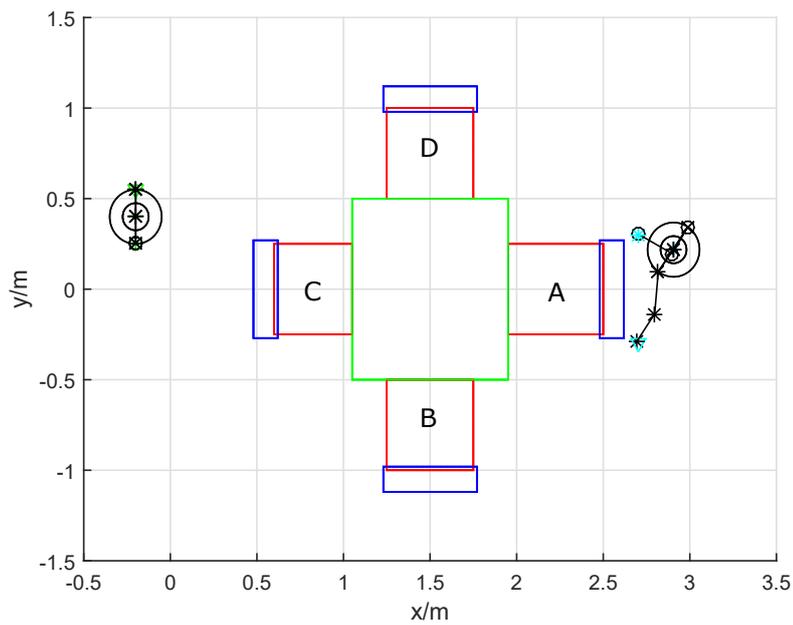


(b) 2D view.

Figure 3.6 – Motion planning results to manipulate chair A: Improved MaxiMin NSGA-II.



(a) 3D view.



(b) 2D view.

Figure 3.7 – Motion planning results to manipulate chair A: combined fitness function-based MOGA.

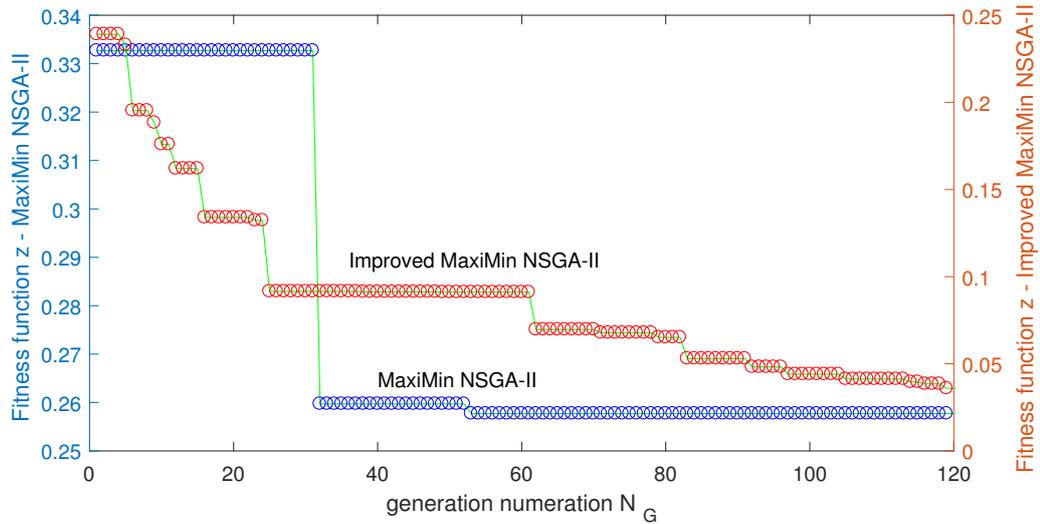


Figure 3.8 – Comparison between the improved MaxiMin NSGA-II and MaxiMin NSGA-II algorithms.

In addition, the motion planning results using the combined fitness function-based MOGA are shown in Fig. 3.7 as a comparison with the fitness function being chosen as in Eq. (3.8). It can be seen that even though the reaching accuracy is better than the improved MaxiMin NSGA-II algorithm, the position-orientation of the mobile platform and the configuration of the upper manipulator are far away from expectations. The reason is that different from the combined fitness function-based MOGA which searches for the optimal solution in only one direction, the improved MaxiMin NSGA-II algorithm optimizes multiple objective functions at the same time to preserve the diversity.

In order to validate the effectiveness of the proposed improved MaxiMin NSGA-II algorithm. Two simulations are realized for optimal pose design using respectively the MaxiMin NSGA-II and improved MaxiMin NSGA-II algorithms. It can be seen in Fig. 3.8 that the optimal solution converges more quickly and continuously with better performance using the **Algorithm 2**.

### 3.3.1.2 Path Planning for Mobile Platform

In order to validate the efficiency of the proposed path planning method in **Algorithm 3**, path design for the mobile platform is realized in the following.

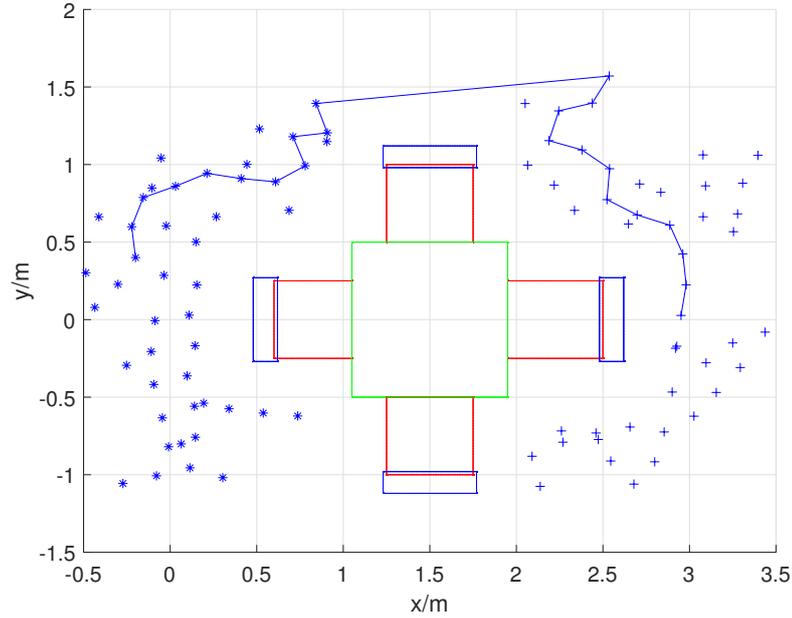


Figure 3.9 – Path planning results without geometric optimization for mobile platform to approach chair A.

Suppose that the MDAMS locates initially at  $x_M = [-0.2, 0.4]$  with the initial orientation  $\phi_0 = 0$ . The desired positions of two EEs are respectively  $x_{Rd} = [2.7, 0.3, 0]$  and  $x_{Ld} = [2.7, -0.3, 0]$ . The mobile platform's optimal position is designed as  $x_{Md} = [2.9512, 0.0269]$  using **Algorithm 2**.

Denote the mobile platform has a circle geometry. To guarantee the collision-free navigation for the mobile platform from the initial position to the designed optimal position, we enlarge the obstacles by a security hull  $\delta_{sm} = 0.3m$  including the dimension of mobile platform.

Fig. 3.9 shows the designed path without geometric optimization. The green cuboid represents the table, the red and blue cuboids represent four chairs. The blue star points represent the sampling nodes of the start exploring tree, the blue plus points represent the sampling nodes of the end exploring tree, and the blue line is the designed path. It can be seen that two exploring trees are connected directly once there is a collision-free connection between them, even if they are still very far away from each other. It can be seen that a collision-free

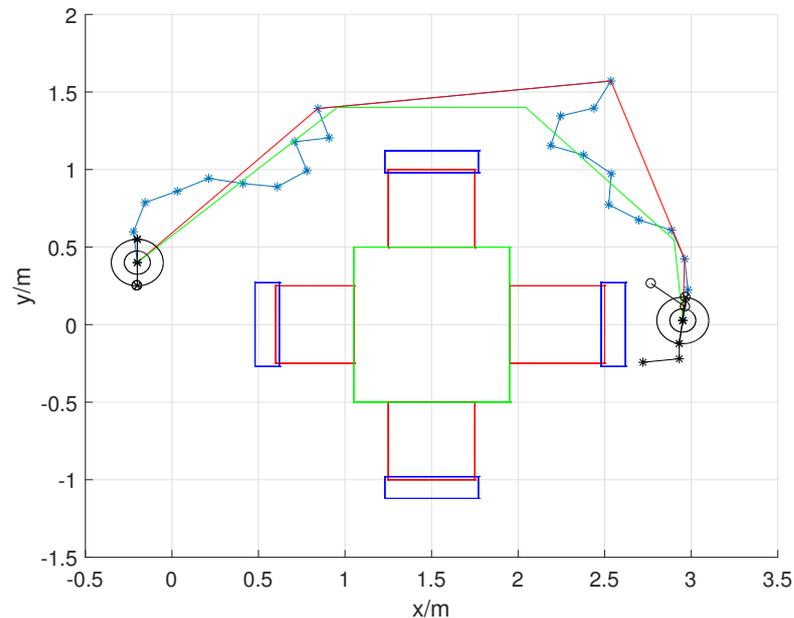


Figure 3.10 – Path planning results with geometric optimization for mobile platform to approach chair A.

path is designed from the initial position to the designed optimal position for the mobile platform. But, it is obvious not the optimal path.

Fig. 3.10 shows the designed path using geometric optimization method. The black circles representing the mobile platform, the green rectangle represents the table, the red and blue rectangles represent four chairs. The blue line which links the blue star points represents the designed path before geometric optimization. Apply the proposed geometric optimization method, the red line represents the *pre-optimal path* after *node rejection* and the green line represents the *optimal path* after *node adjustment* which is also the designed optimal path. There are in total  $n_p = 5$  nodes which characterize the designed optimal path, including the initial and designed optimal nodes. There found two via-points around chair D and one via-point around chair A to avoid collisions. The results validate the effectiveness of the proposed direct connect BiRRT and Gradient decent and geometric optimization methods.

### 3.3.1.3 Trajectory Planning for MDAMS

In the following, the time-specified trajectory which bypasses the planned via-points and links the initial pose to the designed optimal posed is designed.

Recall again the task to approach and hold chair A. The desired optimal pose is designed using **Algorithm 2** and the via-points are generated using **Algorithm 3**. Suppose that the simulation time is  $T = 20s$ , the navigation time scope and the start time of manipulation is  $t_\theta = 3/4T$ , then the time scope of manipulation is  $T_\theta = T/4$ . The navigation time scope along each segment for the mobile platform is calculated using Eq. (3.9), set the parabolic blend duration at each via-point as  $\Delta t'_k = 1/20T$ .

The designed trajectory for robot MDAMS is shown in Fig. 3.11. In detail, Figs. 3.11 (a) and (b) show the designed motion to approach and hold chair A using the improved MaxiMin NSGA-II algorithm. The black skeletons represent respectively the initial and final poses of the designed robot MDAMS. The magenta lines represent the designed trajectories of the mobile platform and two EEs, respectively. There are six orientations which leads the robot to head forward along each segment. The mobile platform changes to an orientation at each via-point, and the orientation will be kept still along each segment between two adjacent via-points. In addition, from  $t_\theta = 16s$ , the mobile platform stops at the designed optimal position with the optimal orientation and the upper manipulator moves to reach the optimal configuration.

As a comparison, the motion planning results using the combined fitness function-based MOGA are shown in Figs. 3.11 (c) and (d). It can be seen that the trajectories are perfectly designed to lead the robot from its initial pose to the optimal pose in two cases, while the later has self-collision because of the designed unreasonable pose.

### 3.3.2 Task 2: Chair D

In order to enforce the effectiveness of the proposed **Algorithms 2** and **3**, a similar motion planning task is realized in the following for robot MDAMS to manipulate chair D.

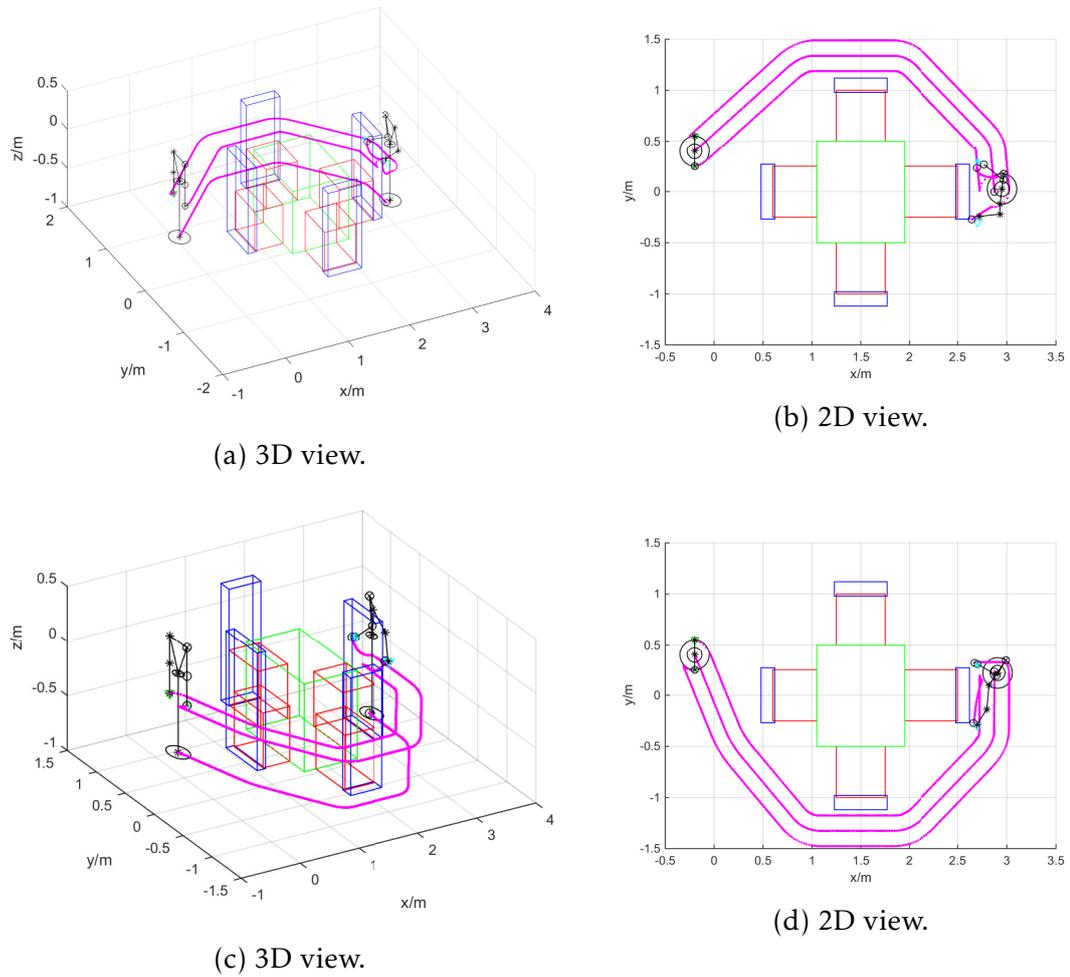


Figure 3.11 – Motion planning results to manipulate chair A. Up: improved MaxiMin NSGA-II. Down: combined fitness function-based MOGA.

### 3.3.2.1 Optimal Pose Design

The MDAMS locates initially at  $x_M = [-0.2, 0.4]$  and the initial positions of two EEs are respectively  $x_R = [-0.2, 0.25, -0.219]$  and  $x_L = [-0.2, 0.55, -0.219]$ . Suppose that the target positions of two EEs' are respectively  $x_{Rd} = [1.2, 1.12, 0.2]$  and  $x_{Ld} = [1.8, 1.12, 0.2]$  around chair D. Based on the robot's dimension, the expected position and orientation of mobile base are calculated as  $(x_{me}, y_{me}) = (1.6, 1.35)$  and  $\phi_{de} = -90^\circ$ , respectively. The designed optimal *pose* is shown in Table 3.2 and Fig. 3.12 (a).

Table 3.2 – The designed optimal pose  $\theta_{op}$  [ $m, deg$ ].

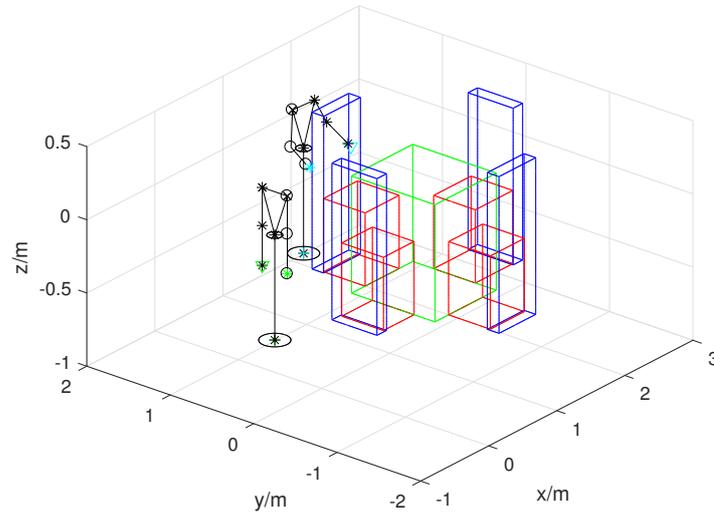
$\theta_m$ & $\theta_w$	1.57	1.33	-102.62	0.12	-0.19	-	-
$\theta_R$	-9.94	2.73	4.58	69.94	-3.31	1.36	-2.32
$\theta_L$	-2.32	2.46	8.60	72.28	-1.66	-0.48	16.93

It can be seen that the designed optimal position of the mobile platform is  $(x_{md}, y_{md}) = [1.5689, 1.33]$  which is around the expected position  $(x_{me}, y_{me})$ . The designed optimal orientation of mobile platform is  $\phi_d = -102.62^\circ$  which is around the expected orientation  $\phi_{de}$ .

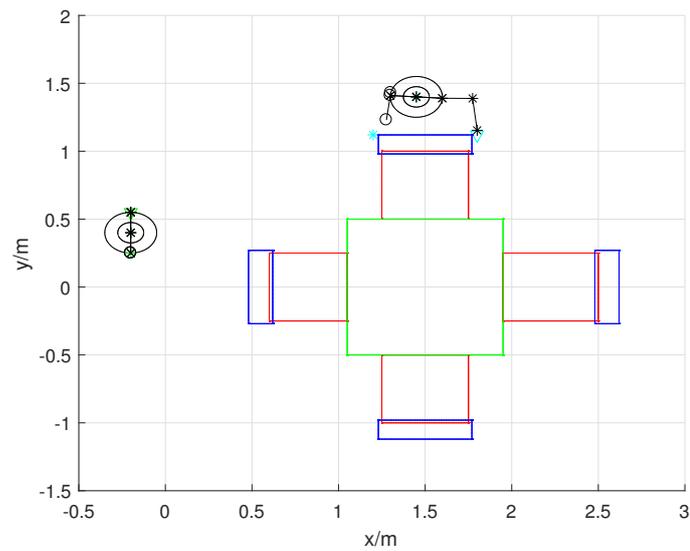
Fig. 3.12 shows the motion planning results. It can be seen that the designed optimal pose is reasonable and with good reaching accuracy. In addition, the motion planning results using the combined fitness function-based MOGA are shown in Fig. 3.13 as a comparison. It can be seen that even though the reaching accuracy is better than the improved MaxiMin NSGA-II algorithm, the position-orientation of the mobile platform and the configuration of the upper manipulator system are far away from expectations. The results enforce the conclusion in Task 1 that optimal pose can be perfectly designed with good reaching accuracy using **Algorithm 2**.

### 3.3.2.2 Trajectory Planning for MDAMS

In the following, the time-specified trajectory which bypasses the planned via-points and links the initial pose to the designed optimal posed is designed. Fig. 3.14 shows the path planning results for the mobile platform. The designed trajectory for robot MDAMS is shown in Fig. 3.15. In detail, Figs. 3.15 (a) and (b) show the planned motion using the proposed improved MaxiMin NSGA-II algorithm. There are four orientations, in detail, the orientation changes to a value which leads the robot to head forward along each segment at each via-point. The motion planning results using the combined fitness function-based MOGA are shown in Figs. 3.15 (c) and (d) as a comparison. It can be seen that the trajectories are perfectly designed in two cases, while the later has self-collision because of the designed unreasonable pose.

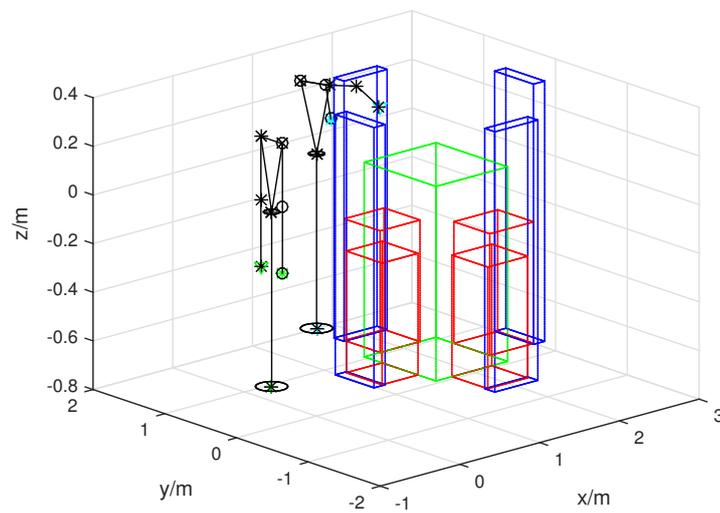


(a) 3D view.

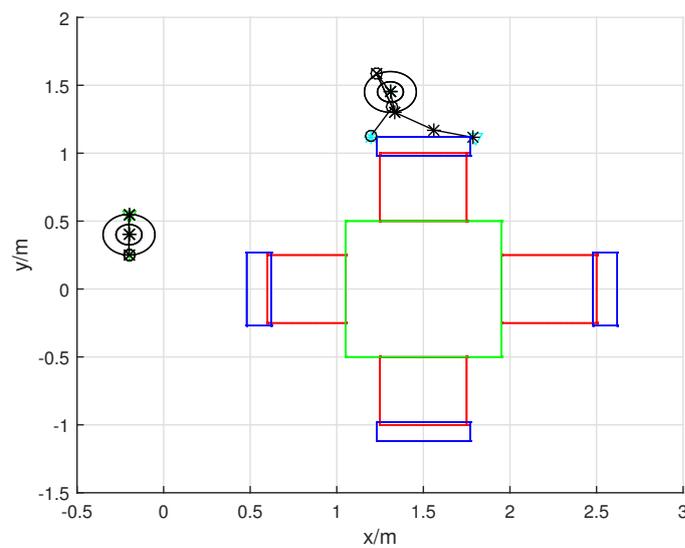


(b) 2D view.

Figure 3.12 – Motion planning results to manipulate chair D: Improved MaxiMin NSGA-II.



(a) 3D view.



(b) 2D view.

Figure 3.13 – Motion planning results to manipulate chair D: combined fitness function-based MOGA.

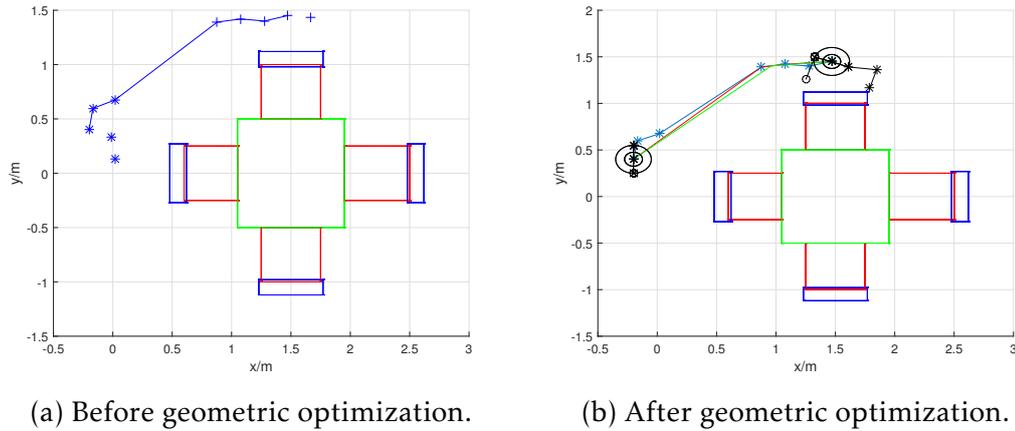


Figure 3.14 – Path planning results for mobile platform to approach chair D.

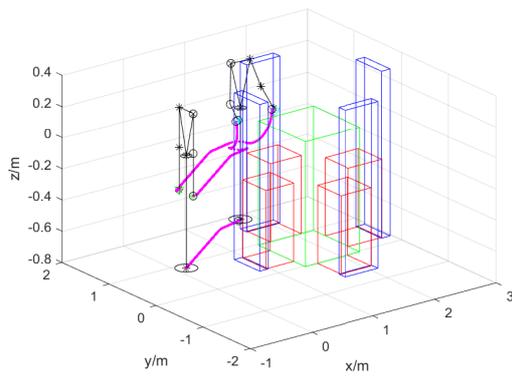
### 3.3.3 Motion Planning and Dynamic Tracking for MDAMS

The main result of this Chapter and the adaptive RBFNN controller proposed in Eq. (2.52) are used to achieve motion planning and tracking for robot MDAMS to approach chair D.

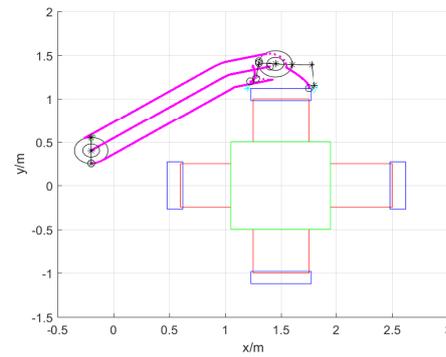
The proposed robot MDAMS locates initially at  $x_M = [-0.2, 0.4]$ . Suppose that the desired positions of two EEs are known to manipulate chair D. The simulation time is  $T = 20s$ . The simulation results are shown in Fig. 3.16, where the circle trajectories represent the real trajectories of the mobile platform and two EEs. The yellow cross trajectory represents the desired trajectory of the mobile platform. It can be seen that the mobile platform's and EEs trajectories are well-designed and the robot tracks well the designed motion.

## 3.4 CONCLUSIONS

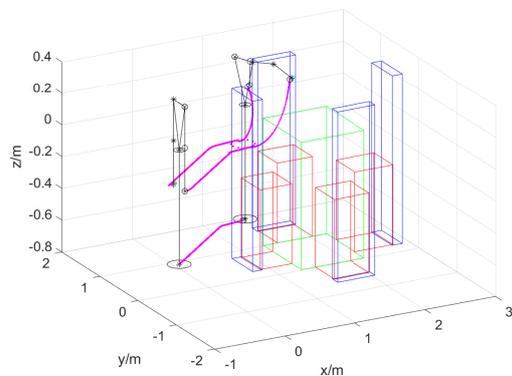
In this Chapter, an improved MaxiMin NSGA-II algorithm is proposed for optimal *pose* design, and an off-line motion planning algorithm is designed to plan motion for robot MDAMS. The mobile platform and the manipulator are treated as a whole system. It offers an efficient method for designing the optimal *pose* with only the desired positions of EEs being required and does not require the IK solving. Five objective functions are defined to meet multiple constraints at



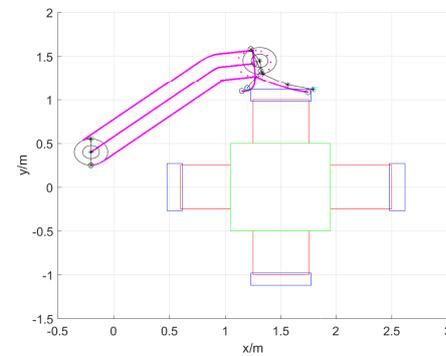
(a) 3D view.



(b) 2D view.



(c) 3D view.



(d) 2D view.

Figure 3.15 – Motion planning results to manipulate chair D. Up: improved MaxiMin NSGA-II. Down: combined fitness function-based MOGA.

the same time. Other objective functions can be introduced into the proposed algorithm, e.g. to achieve maximum carrying capacity or self-collision free movements. Besides, a direct-connect bidirectional sampling technique which speeds up greatly the path sampling phase is proposed. And a geometric optimization method which always guarantees the shortest navigation path is designed. In addition, the designed path remains the same for repeated tasks. What's more, robot MDAMS always heads forward due to the calculated orientations based on the planned via-points of the mobile platform which is very important to increase the quality of human-robot interaction.

However, the method proposed in this Chapter considers only the collision

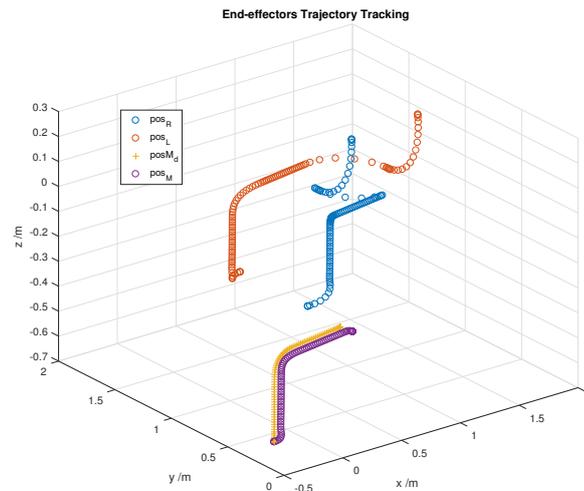


Figure 3.16 – Dynamic simulation results: motion planning and tracking.

avoidance via mobile platform's navigation, the collision of the upper manipulator system has not been studied. Besides, in standard domestic environments with unknown obstacles including human beings, the proposed off-line motion planning algorithm will lose its effectiveness and may cause collision catastrophe. In the next Chapter, motion planning problem in dynamic environments will be studied.

# MOTION PLANNING IN DYNAMIC ENVIRONMENTS

## 4.1 INTRODUCTION

In Chapter 3, the motion planning problem for robot MDAMS in static environments has been studied. The human-like optimal pose is designed by considering the mobile platform and upper manipulator as a whole system. And a point-to-point motion planning method is proposed. However, the designed motion may fail by colliding with unforeseen obstacles including human beings in dynamic environments. Therefore, on-line human-like motion planning for robot MDAMS is studied in this Chapter.

**Mobile vehicles** As reviewed in Chapter 1, many on-line motion planning methods have been introduced, especially the RRTs-based algorithms. However, they are usually random, non-optimal and inconsistent. Although the Informed-RRT\* algorithm in [100] is asymptotic optimal, it needs to evaluate the validity of the entire current path and partially repairs it, while it is not necessary to always evaluate the entire path. The RRT\* method in [101] consists of *reconnect* and *regrow* processes and can only guarantee local-optimal path. Besides, they mainly focus on motion planning for mobile or free-flying robots and consider not the non-holonomic constraints. Even though there exist studies on the non-

holonomic mobile vehicles [106], they do not look through the heading angle planning which is very important for the non-holonomic humanoid mobile robots.

**Mobile Manipulators** Although the above-mentioned methods can be applied to high DoFs robots, their efficiency will diminish as the manipulator's DoFs increase and with the combination of the mobile vehicle and upper manipulator system.

Therefore, many researchers studied the on-line motion planning problem for many DoFs manipulator systems. Lee [57] proposed a virtual roadmap-based on-line motion planning method for a dual-arm robot. Singularity avoidance of a mobile surgery assistant is studied in [108] based on penalty function and EE's path is needed. Yang [110] introduced a configuration re-planning method for a fixed manipulator in a dynamic environment. However, it requires calculating inverse Jacobian and the desired configuration must be known. Chen [111] designed a joint velocity correction term to manipulator's joint trajectory to avoid obstacles. Xin [112] introduced an escape velocity and projecting it onto Jacobian null space for redundant robot arms. Han [113] used distance calculation and discrete detection for robot arms. Vannoy [115] generated a population of trajectories based on the fitness evaluation for the mobile manipulators in dynamic environments with moving obstacles of unknown trajectories. Nguyen [114] presented a heuristic method for a multi-DoFs humanoid robot arm using RRT\* algorithm directly in operational space. But, it considers the EE and elbow as two hierarchical control points. In fact, as the number of control points increases, the hierarchical motion planning method will become complicated. Vannoy [115] introduced an on-line adaptive motion planning algorithm by generating a population of trajectories based on the fitness evaluation for the mobile manipulators in dynamic environments with moving obstacles of unknown trajectories.

However, they do not study multiple constraints problems or whether the resulting poses are reasonable. The heuristic planning method proposed by Nguyen in [114] solves the human-like motion planning through a hierarchical manner. However, it needs to generate the motion of the elbow according to the

end-effector (EE) which is not applicable to a humanoid mobile manipulator with elbows, shoulders, waist and mobile platform. In addition, many reasoning methods can also be found in [86–89, 126, 130, 131] for mobile manipulator systems. However, they require specific considerations of different scenarios.

In this Chapter, the off-line motion planning **Algorithm 3** is extended to an on-line motion planning **Algorithm 4** in Section 4.2. Subsection 4.2.1 outlines the proposed on-line motion planning algorithm. On-line sensing, collision-test and control cycles are defined. Instead of checking the entire path, **Algorithm 4** only needs to evaluate the obstacle collision in a short time  $\Delta t_{col}$ . The static obstacles and dynamic obstacles are treated differently to save time. A completely new motion will be designed if there are predicted collisions. In order to take the via-poses into consideration, a via-points and MOGA-based motion planning method is introduced in Subsection 4.2.2. A sequence of via-poses are designed to link the planned via-points of EEs. In particular, four via-poses-based objective functions are defined to evaluate the candidate via-poses. Instead of using the traditional manipulability criteria, directional manipulability [205] is employed to optimize the motion between adjacent via-points of EEs. The displacements in world and robotic body frames are considered at the same time. A criterion is defined to evaluate the robot's intersection with the environment. Section 4.3 presents three numerical motion planning tasks for robot MDAMS to validate the proposed method. Conclusions are given in Section 4.4.

## 4.2 MOTION PLANNING AMONG DYNAMIC OBSTACLES

An on-line motion planning **Algorithm 4** is presented in this Section for robot MDAMS in dynamic environments under the following assumptions.

**Assumption 4.1** *The initial position-orientation of mobile platform and the initial positions of two EEs are known.*

**Assumption 4.2** *The desired pose of robot MDAMS is known, but the approaching motion is not specified.*

**Assumption 4.3** *The global information about the dynamic environment is known in real time, i.e. the instantaneous positions and geometry characteristics of all obstacles are known, but the obstacles' future movements stay unknown.*

## 4.2.1 On-line Motion Planning Algorithm

### 4.2.1.1 Overview

Inspired by [107], three cycles - on-line sensing, collision-test and control cycles - are introduced to achieve on-line motion planning for the proposed robot MDAMS. It conducts the motion planning and tracking simultaneously in real time. In each sensing cycle, changes in the environment are captured and updated. The planning process re-plans the desired motion according to the collision test outputs. In each control cycle, the robot will switch to the newly designed motion if there are predicted collisions.

The proposed on-line motion planning method is summarized in **Algorithm 4**. It starts by the initialization: environmental information including positions of all obstacles is captured (*EnvInit*), an off-line motion is firstly designed using **Algorithm 3**, the *sensing*, *control* and *collision-test* cycles ( $\Delta t_s$ ,  $\Delta t_c$  and  $\Delta t_{col}$ ,  $\Delta t_s \leq \Delta t_c \leq \Delta t_{col}$ ) and the corresponding cycle numerations ( $iS$  and  $iC$ ) are initialized. Then, the robot starts to track the designed motion and the on-line motion planning is activated (**while** loop, lines 3 through 27).

In each *sensing* cycle  $[t_{iS}, t_{iS+1}]$ , the environment information - positions of all dynamic obstacles - is updated (line 6, *EnvUpdate*) while conserving their historical positions.

At the start of each *control* cycle  $t_{iC}$ , dynamic obstacles' future positions are predicted (*ObsEstim*, line 11, which will be presented in Eqs. (4.1) and (4.2)) and the obstacle-collision check is conducted for the current motion (line 12, *CollisionCheck*). If there are obstacle-collisions predicted in the *collision-test* cycle  $[t_{iC+1}, t_{iC+1} + \Delta t_{col}]$ , the on-line motion re-planning process *On-linePlanning* will be called (line 14).

It is worth mentioning that the start-node of on-line re-planning process is the start state  $\mathbf{X}(t_{iC+1})$  of the next *control* cycle  $[t_{iC+1}, t_{iC+2}]$  and the end-node always remains the goal state. The start time  $t_{start}$  of the on-line re-planning

**Algorithm 4** On-line Motion Planning

---

```

1: Initialization :
   PosObsInit = EnvInit() {initialize the environment}
   motion ← Algorithm 3 {off-line motion planning}
   sensing cycle  $\Delta t_s$ 
   control cycle  $\Delta t_c$ 
   collision-test cycle  $\Delta t_{col}$ 
2:  $iS \leftarrow 1, iC \leftarrow 1$ 
3: while Goal not reached do
4:   on-line sensing, re-planning, tracking
   sensing:
5:   if start of the  $iS$  sensing cycle then
6:     PosObsNew = EnvUpdate() {update the environment}
7:   else if end of the  $iS$  sensing cycle then
8:      $iS \leftarrow iS+1$ 
9:   end if
   planning:
10:  if start of the  $iC$  control cycle then
11:    VelObsNew = ObsEstim(PosObsNew) {evaluate obstacle's movement}
12:    CollisionCheck(PosObsNew, VelObsNew) {collision prediction}
13:    if collision is predicted then
14:      motion ← On-linePlanning(Algorithm 3) {on-line motion planning}
15:      if a sudden collision appears then
16:        Immediate Stop
17:        motion ← UpdateMotion() {update the desired motion}
18:      end if
19:    end if
20:  else if end of the  $iC$  control cycle then
21:     $iC \leftarrow iC+1$ 
22:    if collision is predicted then
23:      motion ← UpdateMotion() {update the desired motion}
24:    end if
25:  end if
   tracking:
26:  MoveOn(motion) {motion tracking}
27: end while

```

---

process will be the start time  $t_{iC+1}$  of the next *control cycle*  $[t_{iC+1}, t_{iC+2}]$ . For the record, the robot will stop (line 16) immediately and update the designed motion

(line 17) if a sudden collision appears (line 15). Then, the start-node of the on-line re-planning process will be the current state  $\mathbf{X}(t)$ , and the re-planning start time  $t_{start}$  will be the current time  $t$ .

At the end of each *control* cycle  $t_{iC+1}$ , the desired motion will be updated (line 23) if collisions are predicted (line 12). As time goes on, the designed motion is tracked continuously (line 26, *MoveOn*).

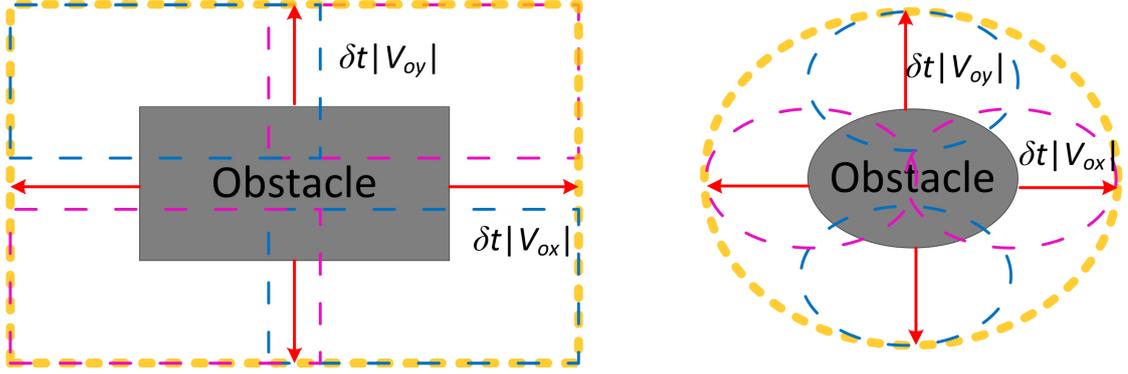


Figure 4.1 – Dynamic obstacle collision space estimation.

#### 4.2.1.2 Obstacle's Motion Prediction

In **Assumption 4.3**, we suppose that the positions of the dynamic obstacles are known in real time. Then, the future positions of the obstacles are estimated using their historical positions (line 11, *ObsEstim* in **Algorithm 4**) as follows. The velocity of obstacle  $i$  is estimated as

$$\mathbf{V}_{oi} = \max\left\{\left\|\frac{\mathbf{X}_{oi}(t_k) - \mathbf{X}_{oi}(t_{k-1})}{\Delta t_s}\right\|, k = iS - m_o, \dots, iS\right\} \quad (4.1)$$

where  $m_o$  is the numeration of historical positions of obstacle  $i$ ,  $\Delta t_s$  is the *sensing* cycle and  $iS$  is the *sensing* cycle numeration. Then, the position of obstacle  $i$  at time  $t = t_{iC} + \delta t$  can be estimated to guarantee the collision-free motion (line 12, *CollisionCheck*) as

$$\mathbf{X}_{oi}(t_{iC} + \delta t) = \mathbf{X}_{oi}(t_{iC}) + \delta t \mathbf{V}_{oi} \quad (4.2)$$

where  $iC$  is the *control* cycle numeration,  $\delta t \in (0, \Delta t_{col}]$ , and  $\Delta t_{col}$  is the *collision-test* cycle. Fig. 4.1 shows two types of obstacles and the orange blocks represent

the possible collision spaces.

**Remark 4.1** *The motion of the mobile platform and two EEs can be designed in task space using the proposed **Algorithm 4**. Theoretically, this algorithm can be applied to the motion planning of any DoF system in dynamic environments consisting of obstacles including human beings.*

**Local environment information sensing discussion** *In **Assumption 4.3**, the global environment information is assumed to be known in real time. However, the global environment information is sometimes difficult to obtain, then our proposed on-line motion planning **Algorithm 4** will lose its efficiency. In order to cope with this problem, the following released assumption is presented.*

**Assumption 4.4** *Only the local environment information is known, i.e. the environment information out of sensing ranges is unknown.*

*Then, **Algorithm 4** is modified as follows. We treat the environment out of the sensing range as a collision-free space, then a sequence of via-points  $\{X_i, i = 1, \dots, n_p\}$  which bypass the sensed obstacles can be found by running lines 1 through 12 in **Algorithm 3** (e.g., four via-points  $\{q_{init}, D, C, q_{goal}\}$  on the green dash line in Fig. 3.3). At the beginning, the robot tries to move along the first segment  $X_1-X_2$ . Then, the environment will be updated as time goes on. This process will be repeated until reaching the target.*

## 4.2.2 Via-points and MOGA-based Motion Planning

Though the proposed **Algorithm 4** can be employed to design the motion for any DoFs mechanical systems in real time, the collision-free test in joint space for a high DoFs robot will be time-consuming and complex. What is worse, since the EEs' paths are not specified, the direct motion planning in joint space will lead to unforeseen motions of EEs.

Therefore, a via-points and MOGA-based motion planning method is proposed. It avoids redundant IK solving and four objective functions characterizing the candidate via-poses are defined to be optimized at the same time (see Subsection 4.2.2.2). Here, the following assumption is proposed to release **Assumption 4.2**.

**Assumption 4.5** *The desired pose of robot MDAMS is unknown, but the approaching motion is not specified.*

#### 4.2.2.1 Via-points and MOGA-based Motion Planning Algorithm

The motion of robot MDAMS can be represented by the trajectory of mobile platform  $(x, y, \phi)$  and the joint trajectory of the upper manipulator  $(\theta_\omega, \theta_R, \theta_L)$  which have been defined in Eq. (2.4). By using **Algorithm 4**, a sequence of collision-free via-points  $\{\mathbf{X}_i = (\mathbf{X}_{Ri}^T, \mathbf{X}_{Li}^T)^T, i = 1, \dots, n_p\}$  of the right and left EEs can be planned on-line, where  $n_p$  is the number of via-points. The objective is to design a sequence of via-poses which expressed by

$$\Theta_i = (x_i, y_i, \phi_i, \theta_{\omega_i}, \theta_{R_i}, \theta_{L_i}), i = 1, \dots, n_p. \quad (4.3)$$

The chromosome of via-points and MOGA-based motion planning algorithm can be chosen as  $\{\Theta_1, \dots, \Theta_{n_p}\}$ . The searching interval for via-pose  $\Theta_i$  is  $\{\Theta_{ij} \in [\theta_{j_{min}}, \theta_{j_{max}}], j = 1, \dots, n\}$ , where  $n$  is the number of MDAMS's generalized variables, and  $\theta_{j_{max}}$  and  $\theta_{j_{min}}$  are respectively the upper and lower boundaries of  $j$ -th variable of  $\theta$  which is defined in Eq. (2.4). Specifically, the via-position  $(x_i, y_i)$  of mobile platform is searched within the reaching region of the corresponding via-point  $\{\mathbf{X}_i\}$ .

At the beginning, a parent population is randomly created. Then, the population is reproduced and the optimal solution is derived by applying multiple objective GAs (MOGAs). For instance, combined fitness function-based MOGAs and the proposed improved MaxiMin NSGA-II algorithm (in Chapter 3) are two possible choices. The fitness function is defined as

$$f = c_1 f_a + c_2 f_b + c_3 f_c + c_4 f_d \quad (4.4)$$

where  $c_i$  ( $i = 1, \dots, 4, \sum c_i = 1, 0 < c_i < 1$ ) is the weight gain, and  $f_k$  ( $k = a, \dots, d$ ) is the objective function which will be defined in the following Subsection.

### 4.2.2.2 Via-poses-based Objective Functions

Four objective functions are defined to evaluate the via-poses, i.e. EEs' approaching accuracy, joint displacement, Robot-obstacle intersection and directional manipulability.

**EEs' reaching accuracy** The designed via-poses  $\{\Theta_i\}$  are required to bypass all the planned via-points  $\{X_1, \dots, X_{n_p}\}$ . Hence, an objective function corresponding to the EEs' reaching accuracy is firstly defined as

$$f_a = \sum_{i=1}^{n_p} \left\| \frac{X_i - X_i(\Theta_i)}{n_p} \right\| \quad (4.5)$$

where  $n_p$  is the number of planned via-points and  $X_i(\Theta_i)$  is the EEs' state which corresponds to the MOGA-reproduced via-pose  $\Theta_i$  using the forward kinematics expressed in Eqs. (2.6) and (2.8).

**Joint displacement** In order to minimize energy consumption, the least joint displacement is required. Then, the second objective function is defined as

$$f_b = \sum_{i=1}^{n_p} \sum_{j=1}^n \left\| \frac{\Theta_{ij} - \Theta_{(i-1)j}}{\theta_{jmax} - \theta_{jmin}} \right\| \quad (4.6)$$

where  $n$  is the number of MDAMS's generalized variables.  $\Theta_{ij}$  is the  $j$ -th variable of via-pose  $\Theta_i$ . The searching intervals for mobile platform depend on the planned via-points of EEs, i.e. the candidate  $(x_i, y_i)$  must stay in the reaching region of the corresponding via-point  $\{X_i\}$ .

**Robot-obstacle intersection** Robot-obstacle intersection is taken into consideration. As described in Section 2.2, robot MDAMS has one mobile platform, one moving waist, two 7-DoFs arms and two hands. Simplify robot MDAMS as a skeleton in Fig. 4.2, then eight control points  $(1, \dots, 8)$  and eight links  $(a, \dots, h)$  are defined for collision-test of the via-pose candidate  $\{\Theta_i\}$ , i.e. the CoMs of the mobile platform, the waist, two shoulders, two elbows and two wrists, and the

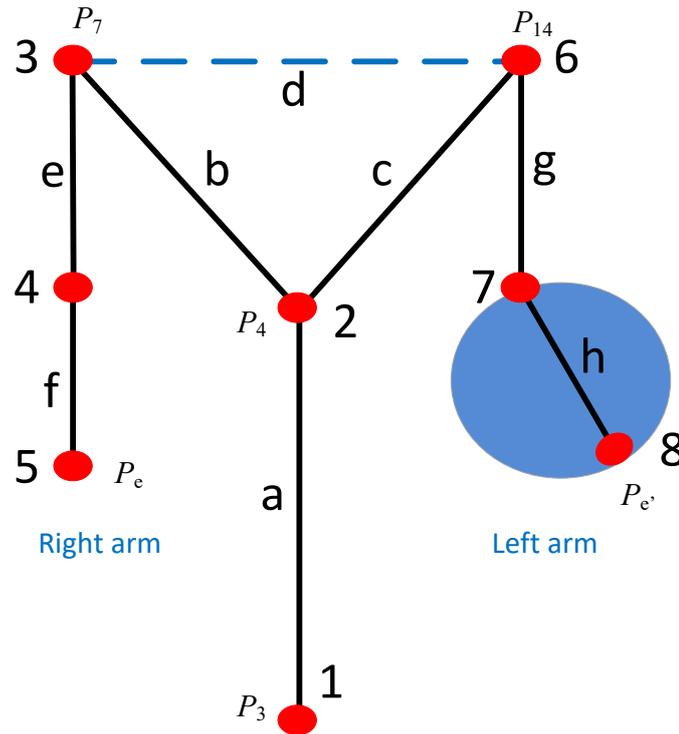


Figure 4.2 – Simplified structure of MDAMS for robot-obstacle collision test.

eight connection links among them.

Then, the third objective function characterizing the rate of MDAMS-obstacles intersection is introduced

$$f_c = \sum_{i=1}^{n_p} \sum_{i=1}^{n_c} \frac{L_{i_{col}}}{L_{col}} \quad (4.7)$$

where  $n_c$  is the number of links on robot MDAMS,  $L_{i_{col}}$  is the robot-obstacle intersection length of link  $i$ , and  $L_{col}$  is the maximum robot-obstacle intersection on robot MDAMS. Take the situation in Fig. 4.2 as an example. The left forearm of MDAMS intersects an obstacle, then the third objective function  $f_c$  can be expressed as  $f_c = \frac{h}{a+b+\dots+h}$ .

**Directional manipulability** To facilitate the motion bypassing all planned via-points  $\{X_1, \dots, X_{n_p}\}$ , there is a manipulability preference along the direction  $d_i$  which links two adjacent via-points  $X_i$  and  $X_{i+1}$  in task space. Then, the

directional manipulability  $\Omega_{Bdir}(\mathbf{d}_i)$  is expressed as follows:

$$\Omega_{Bdir}(\mathbf{d}_i) = \sum_{k=1}^m \|(\mathbf{d}_i^T \cdot \mathbf{u}_k)\sigma_k\| \quad (4.8)$$

where  $(\cdot)$  is the dot product, and  $\mathbf{d}_i = \frac{\mathbf{X}_{i+1} - \mathbf{X}_i}{\|\mathbf{X}_{i+1} - \mathbf{X}_i\|} \in \mathbb{R}^{m \times 1}$  ( $i = 1, \dots, n_p - 1$ ) is the unit vector along each via-segment. In order to obtain  $\mathbf{u}_k$  and  $\sigma_k$ , decompose the Jacobian  $J \in \mathbb{R}^{m \times n}$  using the SVD (Singular Value Decomposition) technique  $J = U\Sigma V$ . Then,  $\mathbf{u}_k$  is the  $k$ -th column vector of matrix  $U \in \mathbb{R}^{m \times m}$ ,  $\sigma_k$  is the  $k$ -th singular value of matrix  $\Sigma \in \mathbb{R}^{m \times n}$ , and matrix  $V \in \mathbb{R}^{n \times n}$ . Then, the fourth objective function is defined as follows:

$$f_d = - \sum_{i=1}^{n_p-1} \Omega_{Bdir}(\mathbf{d}_i) \quad (4.9)$$

which represents the sum of directional manipulability value via each via-points. Recall that  $n_p$  is the number of designed via-points.

### 4.3 SIMULATION RESULTS

Two  $10m \times 10m$  virtual domestic environments (A and B) with static and dynamic obstacles are constructed for numerical simulation. Suppose that each dynamic obstacle moves in the environment with a random bounded velocity. Denote the mobile platform has a circle geometry. To guarantee the collision-free navigation for mobile platform, set the security hull for mobile platform as  $\delta_{sm} = 0.2m$  (including mobile platform's dimension and the security distance) and for EEs as  $\delta_{se} = 0.05m$  on the  $x - y$  plane. Set the security distance along the  $z$ -axis as  $\delta_{sz} = 0.02m$ . The *sensing*, *control* and *collision-test* cycles are initialized respectively as  $\Delta t_s = 0.5s$ ,  $\Delta t_c = 1s$  and  $\Delta t_{col} = 2s$ . In the following, three numerical motion planning tasks are realized in real time using Matlab R2016b in a PC (Intel(R) Core(TM) i5-3317U CPU @ 1.70GHz 1.70 GHz, 4,00Go, x64).

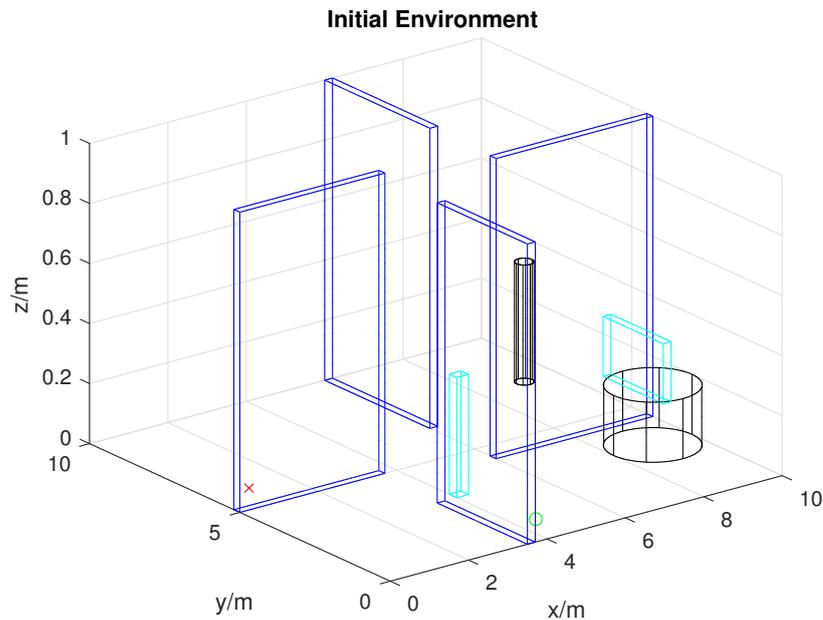


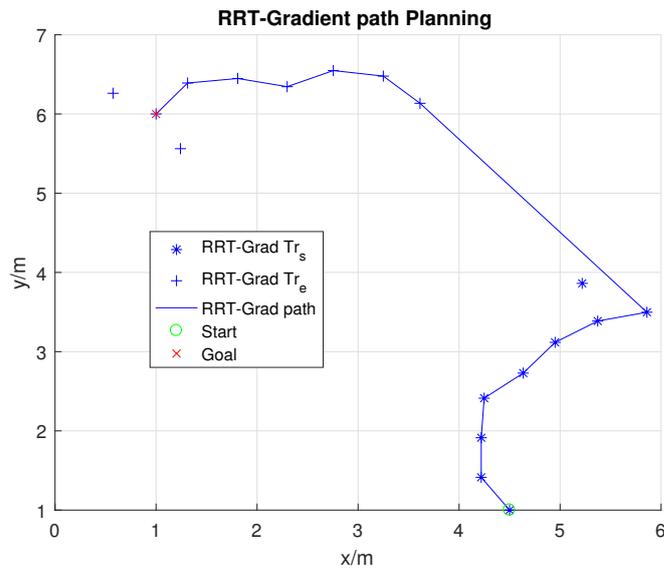
Figure 4.3 – First task in virtual dynamic domestic environment A.

### 4.3.1 On-line Motion Planning for Mobile Platform

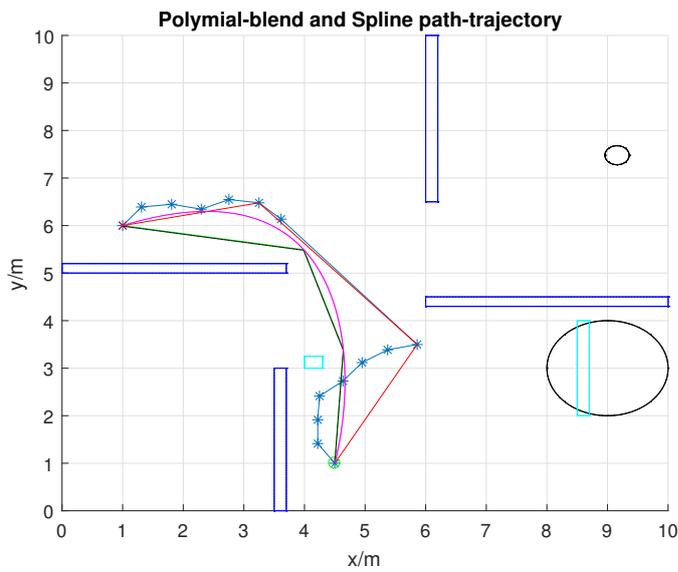
Let us consider the first  $10m \times 10m$  virtual environment A (see Fig. 4.3). A  $0.2m$  tall round table locates at  $(9, 3)$  (black cylinder) with a radius of  $1m$ . There are four static obstacles (blue cuboids) which represent walls, two dynamic obstacles (black cuboid and cyan cylinder) which move randomly on the floor and one dynamic obstacle (cyan cuboid) which locates on the table and moves at a very small random velocity.

The first task is to design the navigation motion for the mobile platform to reach one target position among the obstacles in real time. The mobile platform locates initially at  $(4.5, 1)$  (green circle in Fig. 4.3) and its target position is  $(1, 6)$  (red cross in Fig. 4.3).

At the beginning, the off-line motion is designed. Fig. 4.4 (a) shows the direct-connect BiRRT-gradient sampled path, where the star and plus points represent respectively the start and end trees. Fig. 4.4 (b) shows the geometrically optimized path and the designed trajectories. Where the red line represents the pre-optimal path after node rejection and the green line represents the optimal



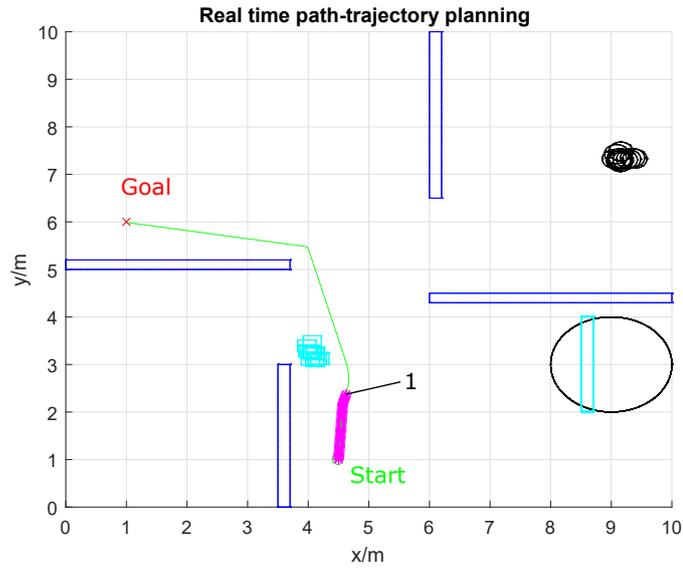
(a)



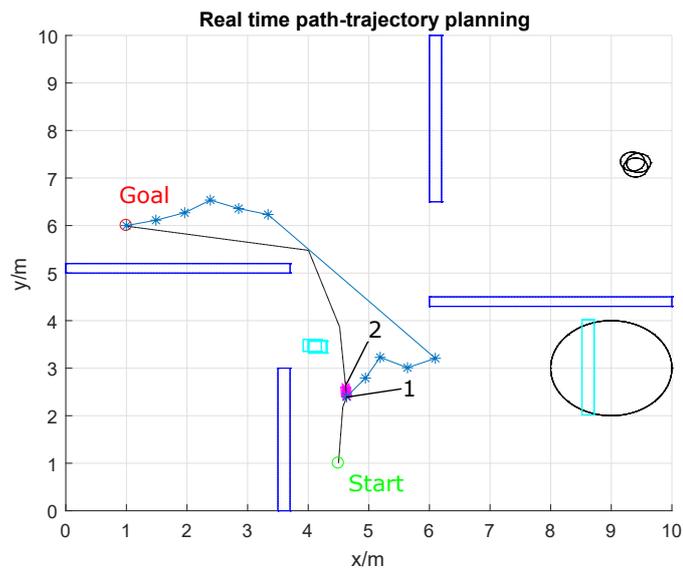
(b)

Figure 4.4 – Off-line motion planning results for mobile base.

path after node adjustment. The black dot line represents the mobile platform's trajectory using the linear polynomials with parabolic blends interpolation technique and the magenta line represents the planned trajectory using B-spline. The initial positions of dynamic obstacles are shown in Fig. 4.4.



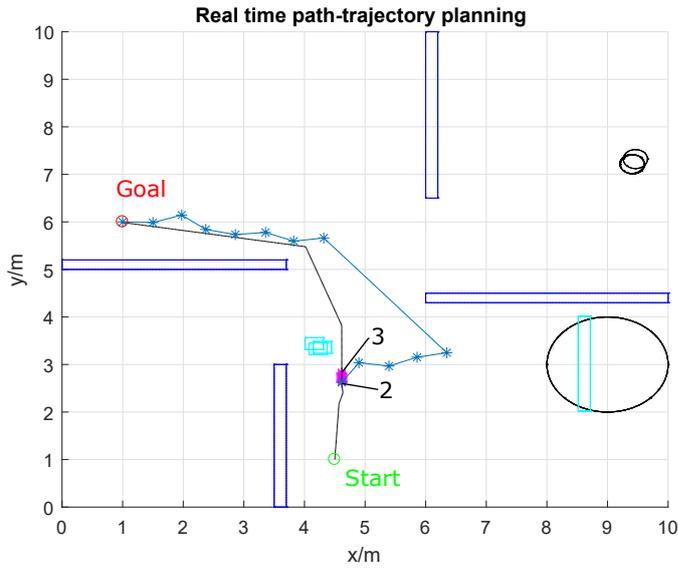
(a) Phase 1.



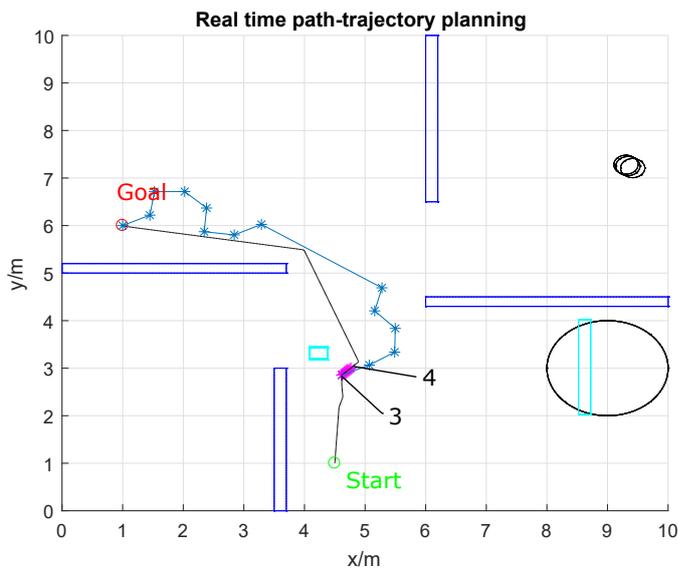
(b) Phase 2.

Figure 4.5 – On-line motion planning results for mobile platform (a).

Then, the on-line motion planning is realized running the **while** loop of **Algorithm 4**. Figs. 4.5 - 4.9 show the on-line motion planning results. The star magenta points represent the mobile platform's positions in real time. The black line behind the robot represents robot's historical trajectory and the black line



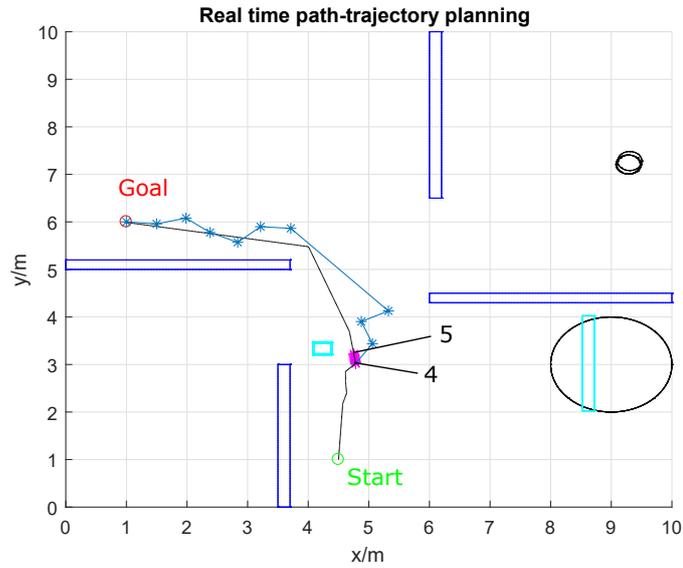
(a) Phase 3.



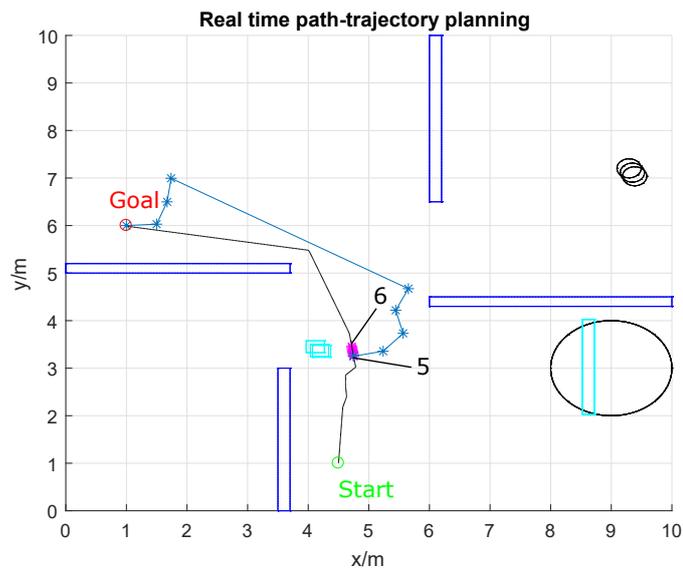
(b) Phase 4.

Figure 4.6 – On-line motion planning results for mobile platform (b).

ahead the robot represents the designed motion in real time. It can be seen that each time there are collisions predicted, the planner redesigns the motion for the mobile platform. Throughout the simulation, there are in total 8 re-planning calls to update the motion in order to avoid the cyan cuboid dynamic obstacle



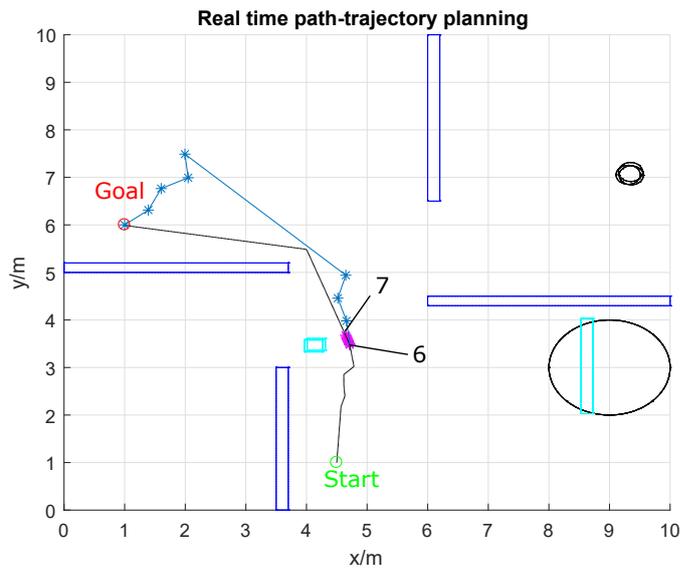
(a) Phase 5.



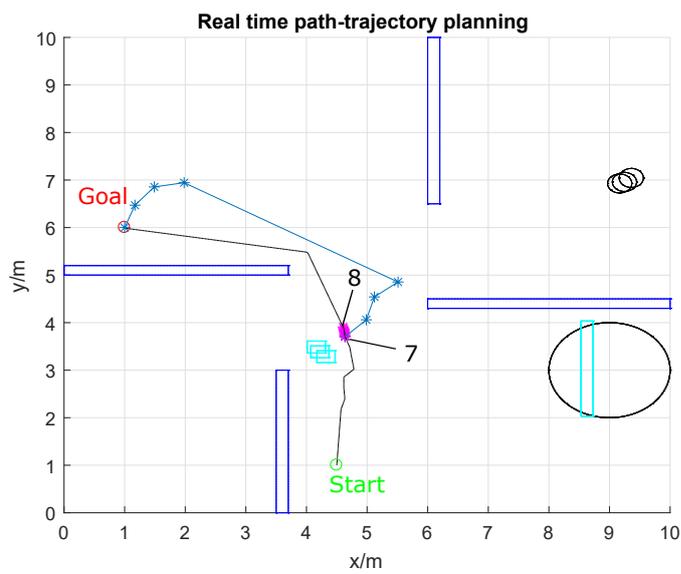
(b) Phase 6.

Figure 4.7 – On-line motion planning results for mobile platform (c).

on the way to the target position. Then, the motion remains the same after bypassing the cyan cuboid (phase 9). In particular, the black dot line in Fig. 4.9 represents the mobile platform's real tracking trajectory. The historical positions of static and dynamic obstacles are also shown in Figs. 4.5 - 4.9 which are all



(a) Phase 7.



(b) Phase 8.

Figure 4.8 – On-line motion planning results for mobile platform (d).

avoided with success.



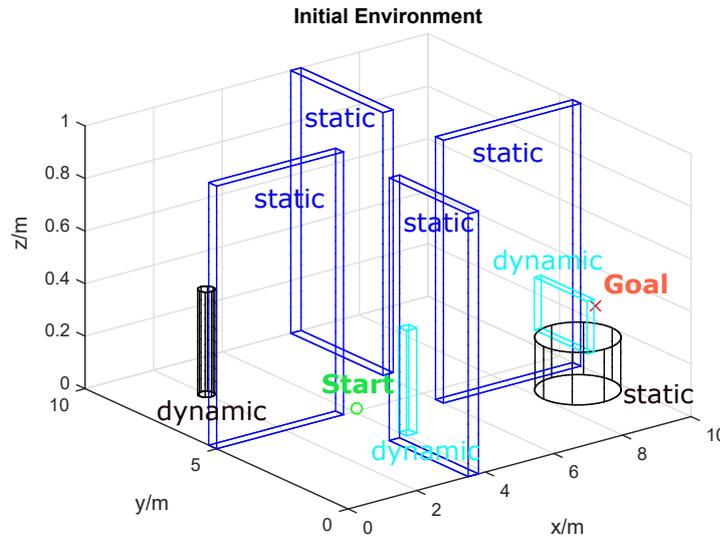
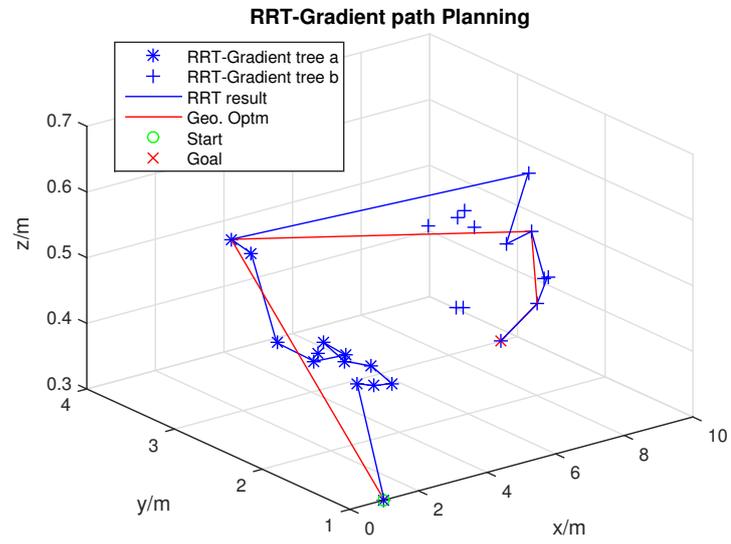


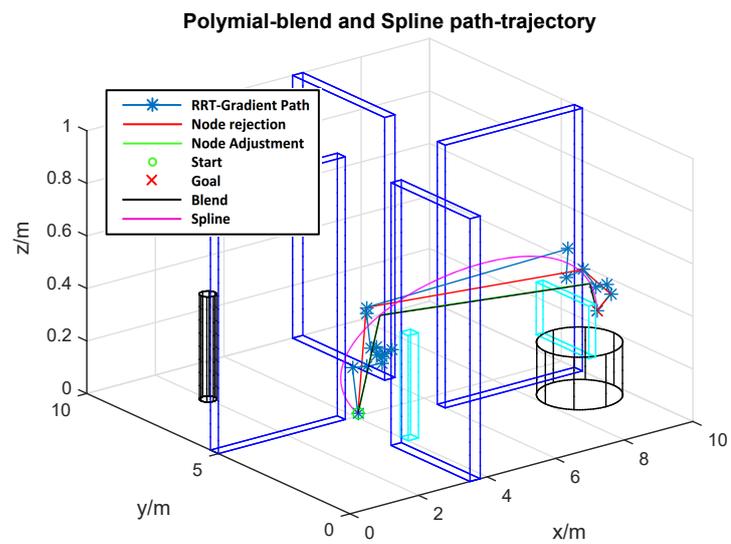
Figure 4.10 – Second task in environment A.

positions in real time. The black line behind the robot represents robot's historical trajectory and the black line ahead the robot represents the designed motion in real time. It can be seen that each time there is a predicted collision, the planner updates the designed motion for mobile platform.

During this on-line numerical simulation, there are in total 4 re-planning calls updating the designed motion to avoid the dynamic cyan cylinder obstacle on the floor, and 5 re-planning calls updating the designed motion to avoid the dynamic cyan cuboid obstacle on the round table. Each time, the designed motion remains the same after bypassing each dynamic obstacles on the way to the target position (phase 5 and phase 10). In particular, the black line in Fig. 4.16 represents the EE's real trajectory. The historical positions of static and dynamic obstacles are also shown in Figs. 4.12 - 4.16 which are all avoided with success.

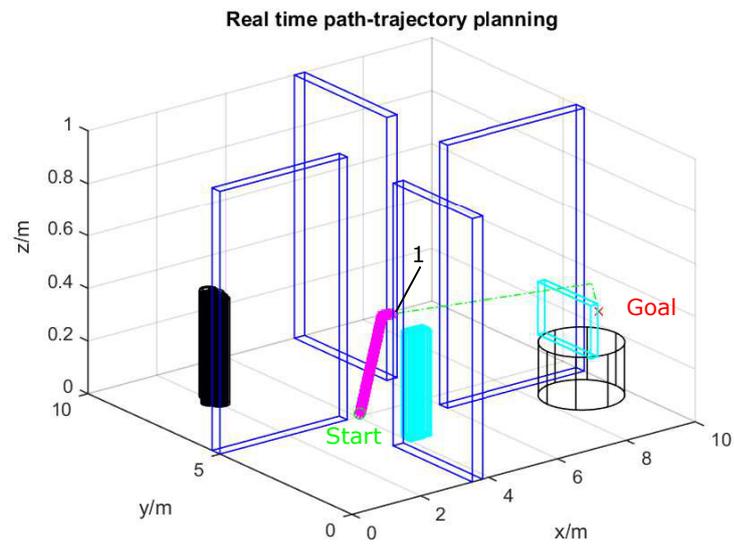


(a)

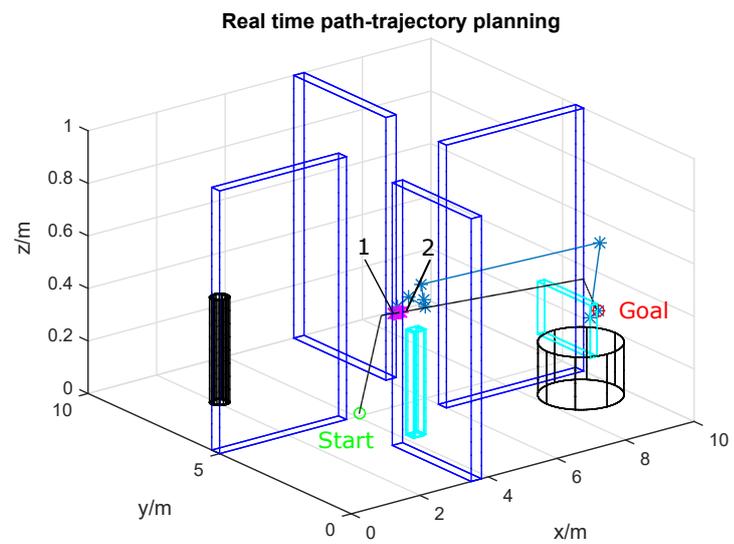


(b)

Figure 4.11 – Off-line motion planning results for EE.

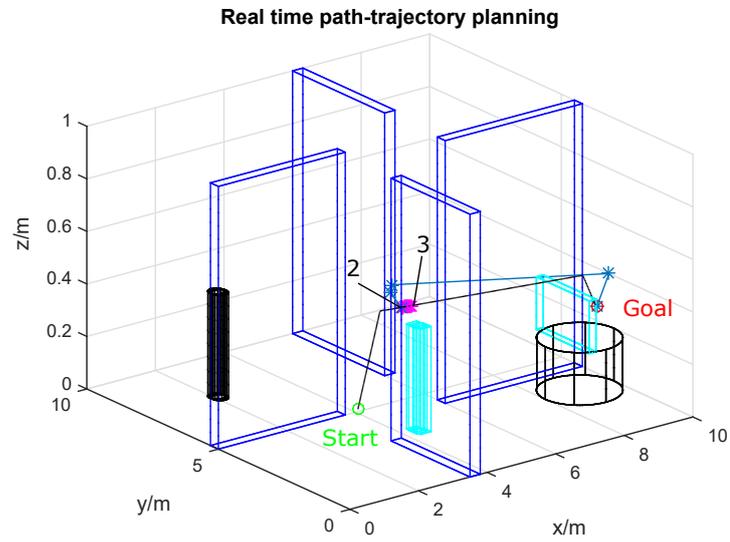


(a) Phase 1.

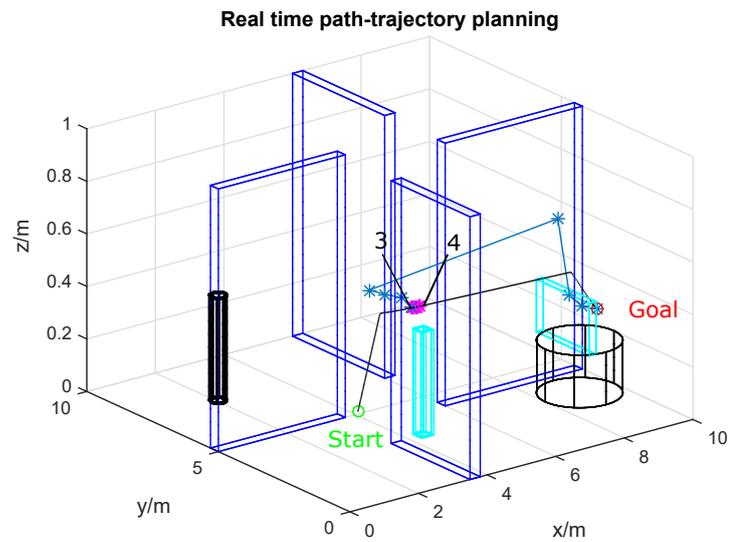


(b) Phase 2.

Figure 4.12 – On-line motion planning results for EE (a).



(a) Phase 3.



(b) Phase 4.

Figure 4.13 – On-line motion planning results for EE (b).

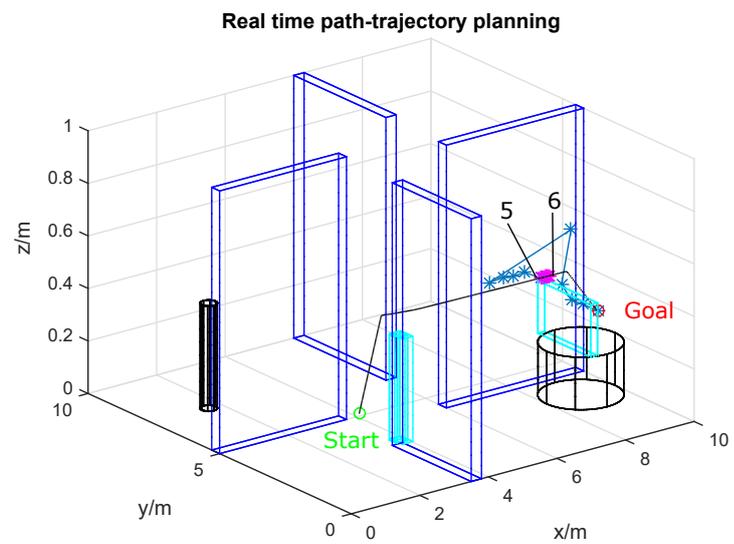
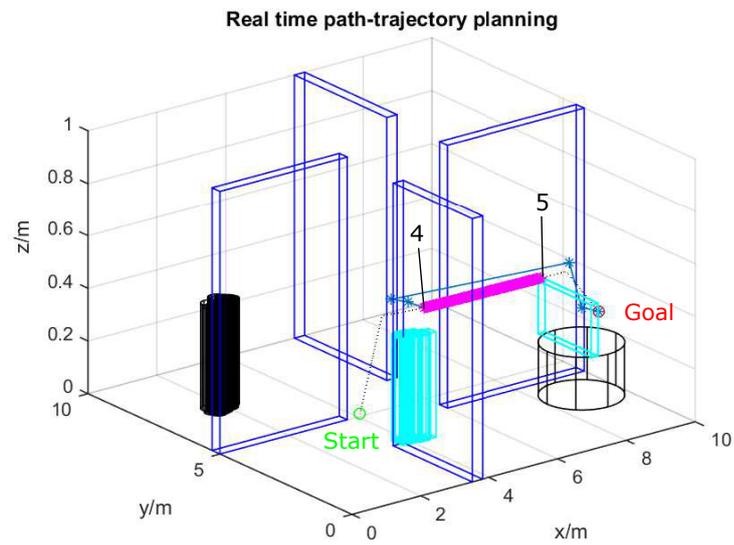
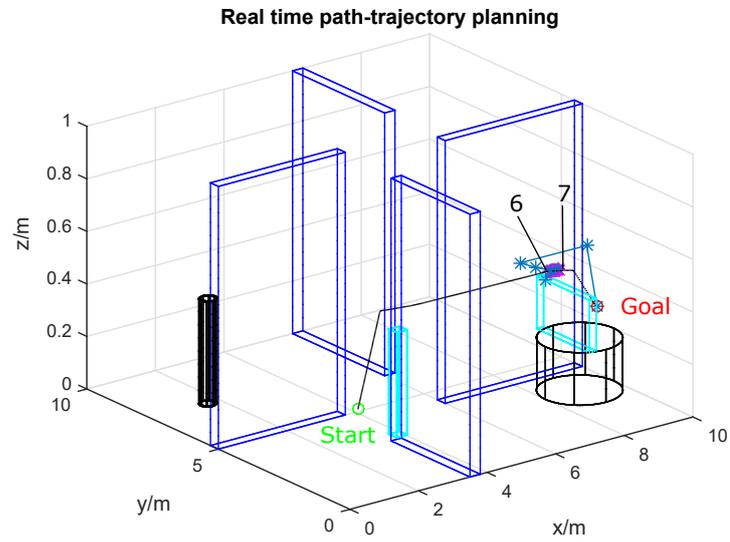
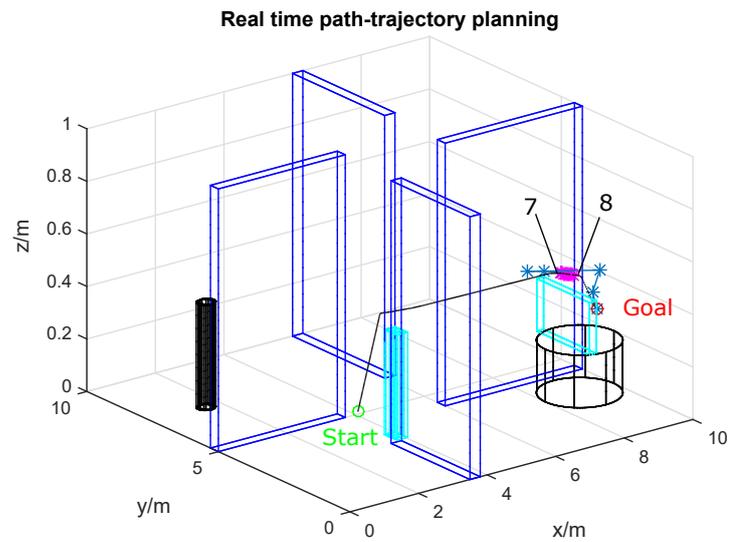


Figure 4.14 – On-line motion planning results for EE (c).

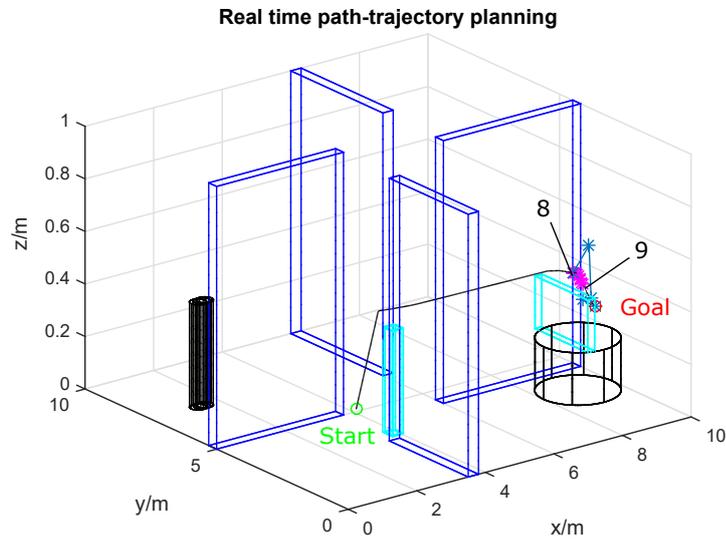


(a) Phase 7

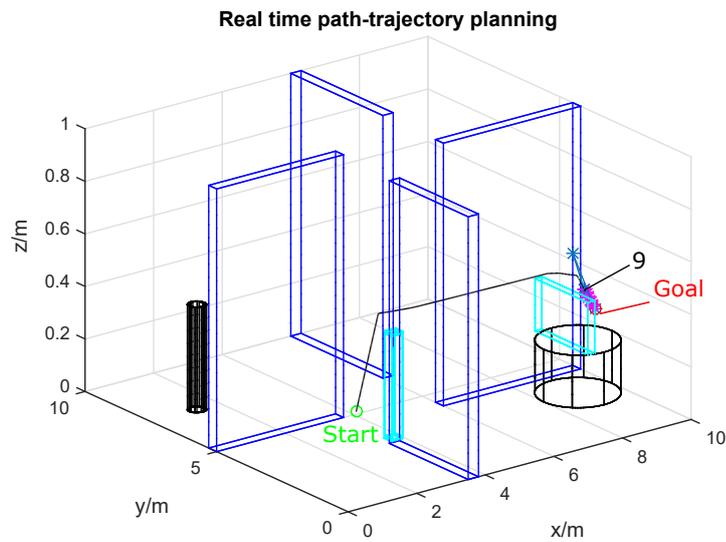


(b) Phase 8

Figure 4.15 – On-line motion planning results for EE (d).



(a) Phase 9



(b) Phase 10

Figure 4.16 – On-line motion planning results for EE (e).

### 4.3.3 Motion Planning Time-consuming Statistics

In order to validate the efficiency of the proposed on-line motion planning method. Up to 100 numerical simulations have been run for the same motion planning task in real time from an initial position  $(9, 1, 0)$  to a target position  $(1, 6, 0)$  in environment A. The off-line motion planning, on-line collision-test and on-line motion re-planning time-consuming statistics are shown in Fig. 4.17. The results show that the average off-line motion planning time is  $\sim 1 \pm 1s$ , the average on-line collision test time is  $\sim 10^{-4}s$ , and the average on-line motion re-planning time is  $\sim 0.7s$ .

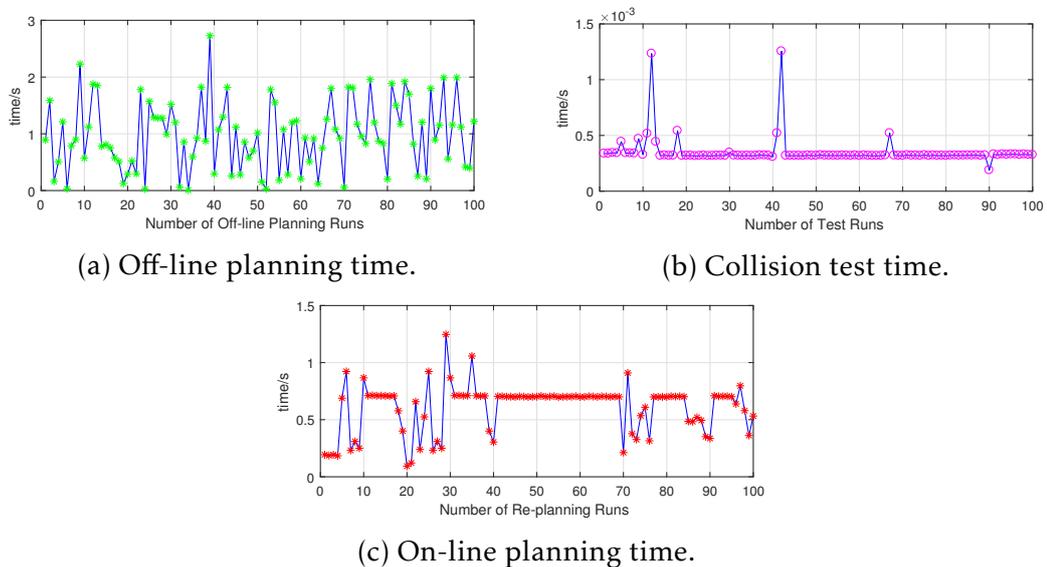
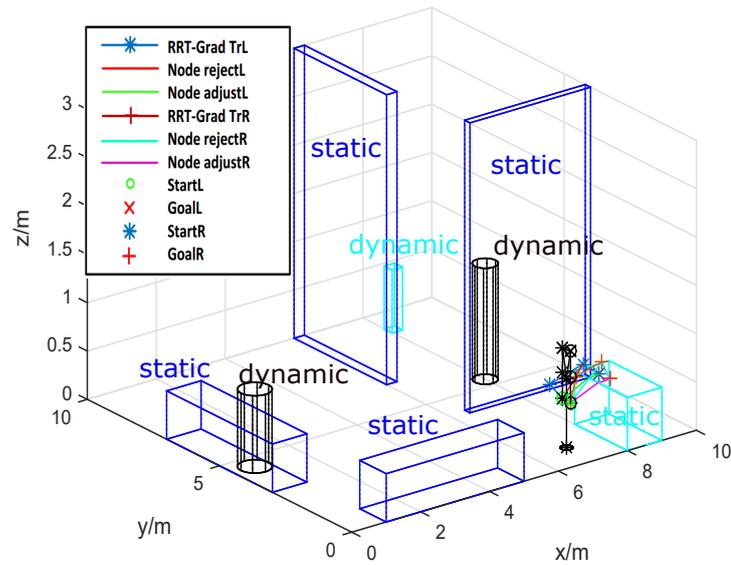


Figure 4.17 – The time-consuming statistics of motion planning **Algorithm 4**.

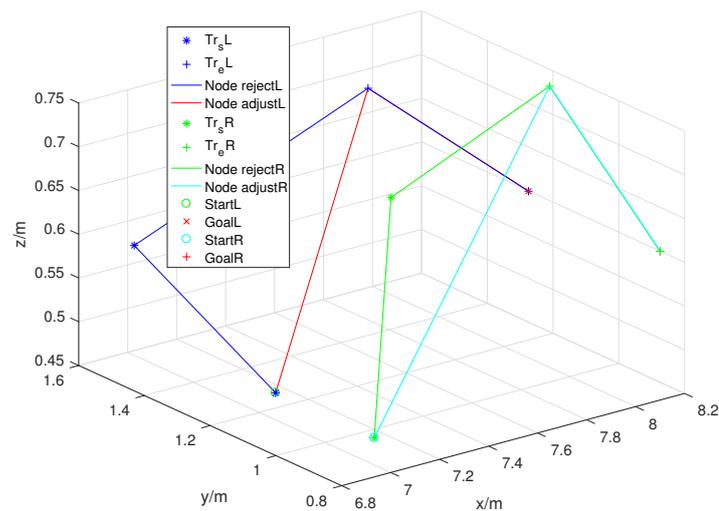
### 4.3.4 On-line Motion Planning for MDAMS

Let us consider the second environment B (see Fig. 4.18). There are 5 static obstacles which represent walls and furniture (blue cuboids) and a  $1m \times 2m \times 0.6m$  square table (big cyan cuboid) which locates at  $(8, 0, 0)$ . And 3 dynamic obstacles move randomly on the floor (black cylinder and small cyan cuboid). The third task is the motion design for robot MDAMS to reach respectively two target positions over the table among the obstacles in real time.





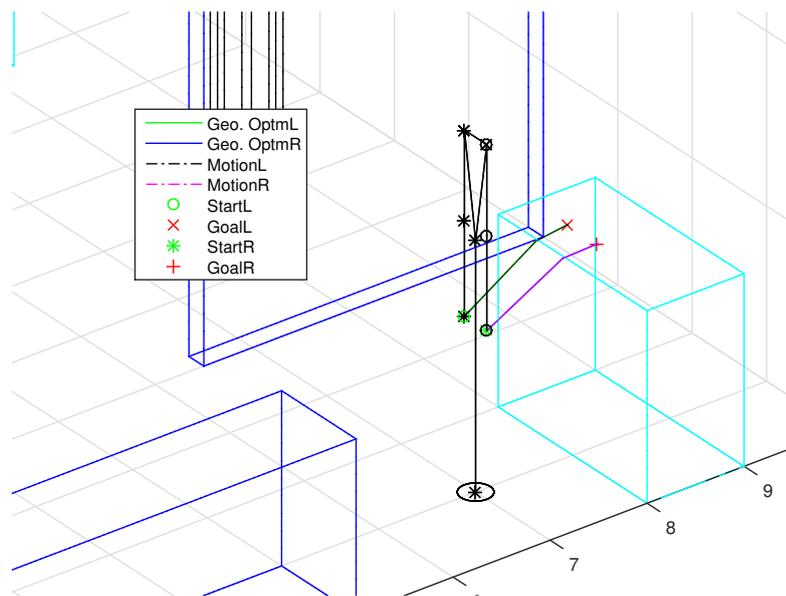
(a) Path planning (global view) for two EEs.



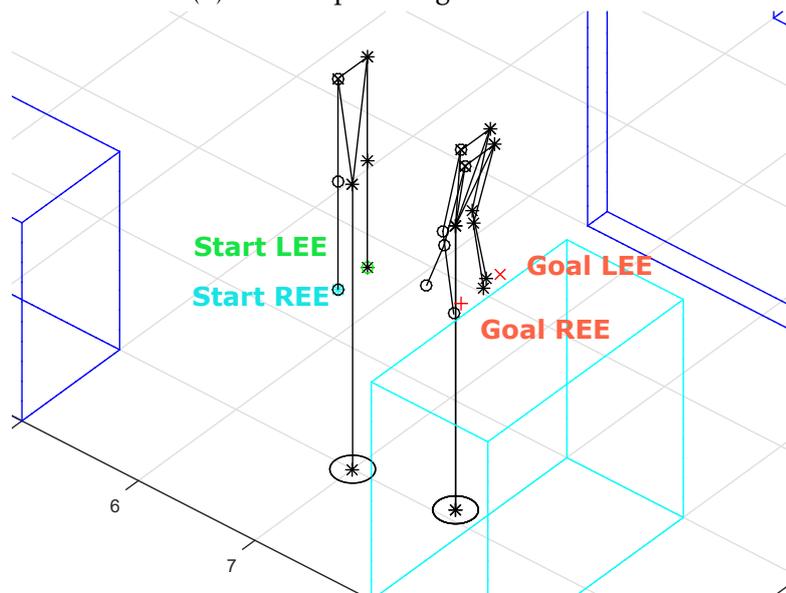
(b) Via-points (local view) for two EEs.

Figure 4.19 – Via-points design for two EEs.

points in Fig. 4.19 (b)). It can be seen that the planned via-poses avoid obstacles and bypass the via-points with reasonable poses, i.e. human-like collision-free mobile platform's positions-orientations and upper manipulator configurations.



(a) Motion planning for two EEs.



(b) Via-poses design for robot MDAMS.

Figure 4.20 – Via-poses design for robot MDAMS (local view).

## 4.4 CONCLUSIONS

An on-line motion planning **Algorithm 4** for MDAMS is introduced in this Chapter in standard domestic environments with unknown dynamic obstacles

including human beings. The environmental information is updated in real time and obstacles' future positions are predicted with success according to their historical positions. Besides, a via-points and MOGA-based motion planning method is proposed with mobile platform's motion being designed with respect to EEs' via-positions and the corresponding via-poses being planned. Four objective functions, (i) minimum joint length, (ii) maximum manipulability via each via-point, (iii) minimum reaching error, and (iv) collision avoidance, are optimized simultaneously to lead robot MDAMS to bypass all the designed via-points of two EEs. Two virtual domestic environments are constructed to validate the proposed algorithm and method. Though they are not very realistic environments, the simulation results show the capabilities of our methods. Other objectives exist in the literature can be applied to the proposed method. For now, the off-line and on-line motion planning problems for robot MDAMS in static and dynamic environments have been investigated. The following Chapter will study the motion tracking problem of the proposed robot MDAMS.

# MOTION TRACKING FOR MDAMS

## 5.1 INTRODUCTION

In the previous Chapters, motion planning problem for robot MDAMS has been studied. In this Chapter, we assume that motion of MDAMS in task space is known and the objective is to track the desired motion. The difficulty is to solve the IK problem and to control the motion of two EEs and the mobile platform at the same time.

Apart from tracking methods for absolute motion [81, 83], good results (like MS [141] and LF [142] techniques) have been done to track relative motion for multiple manipulators and object manipulation. Lewis [133] treated two fixed manipulator systems as a single redundant robot. Choi [143] defined a relative motion between two fixed tool and object manipulators using conventional propagation procedure. Jamisola [146, 147] studied the relative motion between multi-manipulators. However, they mainly focus on fixed manipulators. Many works have been done to control mobile manipulatorys by adjusting the mobile platform given EE's path [83]. Mashali [148] defined two new joint-dependent control variables  $D$  and  $\alpha$  to take the mobile platform and the EE into account at the same time. An augmented task vector is defined for a robotic system consisting of one mobile platform and a mounted manipulator arm. But, the formulations of  $D$  and  $\alpha$  are very complex.

Fuzzy logic theory [194] is used to control the mobile platform and the EEs

at the same time. However, many sources of uncertainties [195] occur when applying the fuzzy logic controller (FLC). For example uncertainties in inputs (antecedents) to the FLC; uncertainties in control outputs (consequents) of the FLC; linguistic uncertainties (rules) of the FLC; uncertainties associated with the change in the operation conditions of the controller; uncertainties associated with the use of noisy training data that could be used to learn, tune or optimize the FLC. All of those uncertainties are translated into uncertainties of the fuzzy set membership functions. Type-2 FLC [196–199] offers a powerful method to solve the uncertainties by introducing an extra dimension based on the type-1 FLC. However, when the number of DoFs increases, the fuzzy rules explodes exponentially.

In addition, most of the above-mentioned controllers are centralized for the whole mobile manipulator system. Fierro [201] constructed a kinematic controller for the nonholonomic mobile platform and a torque controller for the nonholonomic mobile robots given the linear velocity of the mobile platform. It controls the nonholonomic mobile platform with high accuracy. Qian [1] has studied the motion and force control of robot MDAMS. An adaptive robust RBFNN controller is proposed to deal with the uncertainties. However, the redundancy solving, the cooperation between two EEs and the coordination between the mobile platform and the EEs are not studied.

Dual-arm manipulation is especially essential for tasks involving objects which are harder to grasp and manipulate with a single arm. This Chapter defines two relative movements for managing the motion among the mobile platform and two EEs, and uses the task-priority notion and kinematic controller in [201] to track the trajectories. Only the movement of the right EE and two relative movements are controlled. As a result, a task priority-based trajectory tracking strategy is presented in this paper for the proposed robot MDAMS to realize the dual-arm cooperation and the non-holonomic mobile platform navigation at the same time.

In Section 5.2, the IK solving and modulated weighted least-norm (mWLN) joint limit avoidance methods are firstly presented. Then, a compact relative Jacobian is formulated relating the joint space with the task space in Section 5.3. A task priority-based redundancy solving method is proposed by defining

three ordered variables, i.e. the relative position of the left EE with respect to the right EE, the absolute position of the right EE and the absolute orientations of two EEs. A semi-decentralized dual-trajectory tracking controller is proposed with a kinematic control law for the non-holonomic mobile platform. In order to coordinate the motion among the mobile platform and two EEs, a new relative movement - motion of the right EE in robotic body frame - is defined in Section 5.4. Then, a semi-decentralized triple-trajectory tracking controller is presented with a kinematic control law for the non-holonomic mobile platform. The mathematical model constructed in Chapter 2 and several trajectory tracking and object transporting tasks are realized in Subsections 5.3.3 and 5.4.3 to validate the proposed methods.

## 5.2 IK SOLVING AND JOINT LIMIT AVOIDANCE

### 5.2.1 IK Solving

The movements of EEs can be easily calculated based on the forward kinematic model constructed in Chapter 2 if the trajectories in joint space are known. However, in real application, normally only the desired motion of EE in task space is specified. The joint position and velocity remain to be determined based on the forward kinematics which is known as IK solving.

Recall that the forward kinematics give

$$\dot{x} = J(\theta)u \quad (5.1)$$

where  $x \in \mathbb{R}^m$  and  $\dot{x} \in \mathbb{R}^m$  are respectively EE's position and velocity,  $\theta \in \mathbb{R}^n$  and  $u \in \mathbb{R}^n$  are respectively joint position and velocity (Eqs. (2.4) and (2.5)), and  $J(\theta) \in \mathbb{R}^{m \times n}$  is the Jacobian. Then, the IK solution can be expressed as

$$u = J^+(\theta)\dot{x} \quad (5.2)$$

where  $J^+(\theta)$  is the pseudo-inverse of  $J(\theta)$ . However, normally the exact solution does not exist. Apply the least-squares solution to minimize  $\|\dot{x} - J(\theta)u\|$ . Then,

the IK solution is expressed as

$$u = J(\theta)^+ \dot{x} + (I - J(\theta)^+ J(\theta)) z_0 \quad (5.3)$$

where  $z_0 \in \mathbb{R}^n$  is an arbitrary vector in a homogeneous part of the solution which can be used to solve the redundancy problem by optimizing user-defined objectives. In other words,  $z_0$  can be chosen as  $z_0 = \alpha_0 \nabla H(\theta)$ , where  $H$  is a scalar objective function.  $\alpha_0$  is taken to be positive if  $H$  is to be maximized, and negative if  $H$  is to be minimized.

### 5.2.2 Joint Limit Avoidance - mWLN

In order to dampen the motion of some joints over the others, the modulated Weighted Least-Norm (mWLN) technique is presented. Introduce a weighted norm  $W$  as follows:

$$J_W = J W^{-1}, \quad (5.4)$$

$$u_W = W^1 u \quad (5.5)$$

where  $W \in \mathbb{R}^{n \times n}$  is a symmetric and positive definite weighting matrix. Apply the transformations to Eq. (2.7) and Eq. (2.10), the following formulations are obtained.

$$\dot{x}_R = J_{R_W} u_W, \quad (5.6)$$

$$\omega_{rl} = J_{\omega_W} u_W \quad (5.7)$$

where  $J_{R_W} = J_R W^{-1}$  and  $J_{\omega_W} = J_\omega W^{-1}$ . The weighting matrix  $W$  can be chosen as a diagonal matrix  $diag(w_i)$ , with  $w_i$  being defined as

$$w_i = \begin{cases} (1 + \|\frac{\partial H(\theta)}{\partial \theta_i}\|)^{1/2^\alpha}, & \text{if } \Delta \|\frac{\partial H(\theta)}{\partial \theta_i}\| \geq 0 \\ 1, & \text{if } \Delta \|\frac{\partial H(\theta)}{\partial \theta_i}\| < 0 \end{cases} \quad (5.8)$$

where  $H$  is a scalar function and  $\alpha$  is a positive integer which modulates the weight of the joint dampen in case the weighting matrix is too big. The definition of  $w_i$  has a valuable meaning. In other words, when the joint angle changes away from the boundary, i.e.  $\Delta \|\frac{\partial H(\theta)}{\partial \theta_i}\| < 0$ , we set  $w_i = 1$  to not dampen the joint.

When the joint angle changes towards the boundary, i.e.  $\Delta \|\frac{\partial H(\theta)}{\partial \theta_i}\| > 0$ , we set  $w_i = 1 + \|\frac{\partial H(\theta)}{\partial \theta_i}\|$  to dampen the joint. The scalar function  $H$  is given as

$$H(\theta) = \sum_{i=1}^n \frac{1}{4} \frac{(\theta_{i,max} - \theta_{i,min})^2}{(\theta_{i,max} - \theta_i)(\theta_i - \theta_{i,min})} \quad (5.9)$$

where  $\theta_{i,max}$  and  $\theta_{i,min}$  are respectively the upper and lower boundaries of joint  $\theta_i$ . The corresponding partial derivative  $\nabla H_i$  can be derived as

$$\nabla H_i = \frac{\partial H(\theta)}{\partial \theta_i} = \frac{(\theta_{i,max} - \theta_{i,min})^2 (2\theta_i - \theta_{i,max} - \theta_{i,min})}{4(\theta_{i,max} - \theta_i)^2 (\theta_i - \theta_{i,min})^2}. \quad (5.10)$$

## 5.3 DUAL-TRAJECTORY TRACKING

This section studies the dual-trajectory tracking problem of two EEs using task-priority technique and relative Jacobian. Three ordered variables are defined. Object manipulation with two EEs is considered by introducing the contact matrix and the object's dynamics.

### 5.3.1 Task Priority and Relative Motion-based IK Solving

Instead of controlling two EEs respectively, a relative motion is introduced to characterize the motion of the left EE with respect to the right EE:  $x_r \triangleq x_R - x_L$ . Recall Eqs. (2.7) and (2.9), the following formulation can be formulated.

$$\dot{x}_r \triangleq \dot{x}_R - \dot{x}_L = J_r u \quad (5.11)$$

where  $J_r \triangleq J_R - J_L$  is the relative Jacobian in terms of two individual Jacobians of two EEs.

#### 5.3.1.1 IK Redundancy Solving

We find that if the relative motion  $x_r$  is chosen as the only control variable, the right EE will not move. And if the relative motion  $x_r$  is chosen as the first

variable and the right EE's motion  $x_R$  as the second variable, two EEs' motions can be perfectly tracked. EEs' orientations are also taken into account. In total, three ordered variables  $\dot{x}_r$ ,  $\dot{x}_R$  and  $\omega_{rl}$  (Eqs. (2.7), (2.10) and (5.11)) are defined to solve the IK redundancy.

The general solution of Eq. (5.11) is obtained as follows using the method introduced in Subsection 5.2.1 (see Eq. (5.3)):

$$u = J_r^+ \dot{x}_r + (I - J_r^+ J_r)z \quad (5.12)$$

by minimizing  $\|\dot{x}_r - J_r u\|$ , where  $J_r^+$  is the pseudo-inverse of  $J_r$ , and  $z$  is an  $n$ -dimensional arbitrary joint velocity vector. Now, substitute Eq. (5.12) into Eq. (2.7), we get the following equation:

$$\dot{x}_R = J_R [J_r^+ \dot{x}_r + (I - J_r^+ J_r)z]. \quad (5.13)$$

Rewrite Eq. (5.13) as:

$$J_R (I - J_r^+ J_r)z = \dot{x}_R - J_R J_r^+ \dot{x}_r. \quad (5.14)$$

Similarly, by minimizing  $\|J_R (I - J_r^+ J_r)z - (\dot{x}_R - J_R J_r^+ \dot{x}_r)\|$ , we get

$$z = \tilde{J}_R^+ (\dot{x}_R - J_R J_r^+ \dot{x}_r) + (I - \tilde{J}_R^+ \tilde{J}_R)z^* \quad (5.15)$$

where  $\tilde{J}_R \triangleq J_R (I - J_r^+ J_r)$ ,  $\tilde{J}_R^+$  is the pseudo-inverse of  $\tilde{J}_R$ , and  $z^*$  is also an  $n$ -dimensional arbitrary joint velocity vector. Then, the solution of  $u$  obtained from Eqs. (5.12) and (5.15) is

$$u = J_r^+ \dot{x}_r + (I - J_r^+ J_r) \tilde{J}_R^+ (\dot{x}_R - J_R J_r^+ \dot{x}_r) + (I - J_r^+ J_r) (I - \tilde{J}_R^+ \tilde{J}_R) z^*. \quad (5.16)$$

Substitute Eq. (5.16) into Eq. (2.10)

$$\omega_{rl} = J_\omega [J_r^+ \dot{x}_r + (I - J_r^+ J_r) \tilde{J}_R^+ (\dot{x}_R - J_R J_r^+ \dot{x}_r) + (I - J_r^+ J_r) (I - \tilde{J}_R^+ \tilde{J}_R) z^*]. \quad (5.17)$$

We get  $z^* = \tilde{J}_\omega^+ \{\omega_{rl} - J_\omega [J_r^+ \dot{x}_r + (I - J_r^+ J_r) \tilde{J}_R^+ (\dot{x}_R - J_R J_r^+ \dot{x}_r)]\}$ , where  $\tilde{J}_\omega \triangleq J_\omega (I - J_r^+ J_r) (I - \tilde{J}_R^+ \tilde{J}_R)$ , and  $\tilde{J}_\omega^+$  is the pseudo-inverse of  $\tilde{J}_\omega$ . Then, the joint velocity can be formu-

lated

$$\begin{aligned}
u = & J_r^+ \dot{x}_r + (I - J_r^+ J_r) \tilde{J}_R^+ (\dot{x}_R - J_R J_r^+ \dot{x}_r) \\
& + (I - J_r^+ J_r) (I - \tilde{J}_R^+ \tilde{J}_R) \tilde{J}_\omega^+ \{\omega_{rl} - J_\omega [J_r^+ \dot{x}_r + (I - J_r^+ J_r) \tilde{J}_R^+ (\dot{x}_R - J_R J_r^+ \dot{x}_r)]\} \\
& + (I - J_r^+ J_r) (I - \tilde{J}_R^+ \tilde{J}_R) (I - \tilde{J}_\omega^+ \tilde{J}_\omega) z^{**}
\end{aligned} \quad (5.18)$$

where  $z^{**} \in \mathbb{R}^n$  is also an  $n$ -dimensional arbitrary joint velocity vector. It can be used to avoid obstacles or to achieve other objectives. For example, the following joint velocity correction term [111]

$$z^{**} = \begin{cases} \dot{\theta}_a \sin(2\pi t/T_a), & t_a < t < t_a + T_a \\ 0, & \text{else} \end{cases} \quad (5.19)$$

can be employed to avoid obstacles during manipulation. Where  $t_a$  is the moment when the obstacle avoidance begins,  $T_a$  is the total time of the obstacle avoidance,  $\dot{\theta}_a$  is the avoiding speed,  $T_a$  and  $\dot{\theta}_a$  are both determined by path planning information and the position and size of the obstacle.

### 5.3.1.2 Joint Limit Avoidance

The mWLN method introduced in Subsection 5.2.2 is used to avoid joint limit. Apply the transformations (see Eqs. (5.4) and (5.5)) to Eq. (5.11)

$$\dot{x}_r = J_{r_W} u_W \quad (5.20)$$

where  $J_{r_W} = J_r W^{-1}$  and  $u_W = W^1 u$ . Using Eqs. (5.6) and (5.7), Eq. (5.18) can be rewritten as follows:

$$\begin{aligned}
u = & W^{-1} \{ J_{r_W}^+ \dot{x}_r + (I - J_{r_W}^+ J_{r_W}) \tilde{J}_{R_W}^+ (\dot{x}_R - J_{R_W} J_{r_W}^+ \dot{x}_r) + (I - J_{r_W}^+ J_{r_W}) (I - \tilde{J}_{R_W}^+ \tilde{J}_{R_W}) \\
& \tilde{J}_{\omega_W}^+ [\omega_{rl} - J_{\omega_W} J_{r_W}^+ \dot{x}_r - J_{\omega_W} (I - J_{r_W}^+ J_{r_W}) \tilde{J}_{R_W}^+ (\dot{x}_R - J_{R_W} J_{r_W}^+ \dot{x}_r)] \\
& + (I - J_{r_W}^+ J_{r_W}) (I - \tilde{J}_{R_W}^+ \tilde{J}_{R_W}) (I - \tilde{J}_{\omega_W}^+ \tilde{J}_{\omega_W}) z^{**} \}
\end{aligned} \quad (5.21)$$

### 5.3.2 Semi-decentralized Dual-trajectory Tracking

Based on the constructed dynamic model in Eq. (2.50), the task-oriented PD control law in task space can be formulated as

$$\tau = \bar{M}(\theta)U + \bar{C}(\theta, u) + \bar{G}(\theta) + \bar{\tau}_d \quad (5.22)$$

where  $\bar{M}(\theta)$ ,  $\bar{C}(\theta, u)$  and  $\bar{G}(\theta)$  are the dynamic matrices,  $\bar{\tau}_d$  is the disturbance vector, and  $U = \dot{u}_{re} + K_v(u_{re} - u) + K_p(\theta_{re} - \theta)$  is the reference acceleration vector.  $K_v$  and  $K_p$  are the definite positive gains. Based on Eq. (5.18), the reference joint velocity  $u_{re}$  is proposed as

$$\begin{aligned} u_{re} = & J_r^+ \dot{x}_{r_{re}} + (I - J_r^+ J_r) \tilde{J}_R^+ (\dot{x}_{R_{re}} - J_R J_r^+ \dot{x}_{r_{re}}) \\ & + (I - J_r^+ J_r) (I - \tilde{J}_R^+ \tilde{J}_R) \tilde{J}_\omega^+ \{ \omega_{rl_{re}} - J_\omega [J_r^+ \dot{x}_{r_{re}} + (I - J_r^+ J_r) \tilde{J}_R^+ (\dot{x}_{R_{re}} - J_R J_r^+ \dot{x}_{r_{re}})] \} \\ & + (I - J_r^+ J_r) (I - \tilde{J}_R^+ \tilde{J}_R) (I - \tilde{J}_\omega^+ \tilde{J}_\omega) z^{**} \end{aligned} \quad (5.23)$$

In addition, the reference joint velocity  $u_{re}$  can be also given as follows to avoid joint limit based on Eq. (5.21)

$$\begin{aligned} u_{re} = & W^{-1} \{ J_{r_W}^+ \dot{x}_{r_{re}} + (I - J_{r_W}^+ J_{r_W}) \tilde{J}_{R_W}^+ (\dot{x}_{R_{re}} - J_{R_W} J_{r_W}^+ \dot{x}_{r_{re}}) + (I - J_{r_W}^+ J_{r_W}) (I - \tilde{J}_{R_W}^+ \tilde{J}_{R_W}) \\ & \tilde{J}_{\omega_W}^+ [ \omega_{rl_{re}} - J_{\omega_W} J_{r_W}^+ \dot{x}_{r_{re}} - J_{\omega_W} (I - J_{r_W}^+ J_{r_W}) \tilde{J}_{R_W}^+ (\dot{x}_{R_{re}} - J_{R_W} J_{r_W}^+ \dot{x}_{r_{re}})] \\ & + (I - J_{r_W}^+ J_{r_W}) (I - \tilde{J}_{R_W}^+ \tilde{J}_{R_W}) (I - \tilde{J}_{\omega_W}^+ \tilde{J}_{\omega_W}) z^{**} \} \end{aligned} \quad (5.24)$$

where  $\dot{x}_{R_{re}}$  and  $\dot{x}_{r_{re}}$  are respectively the reference trajectories of the right EE and the relative motion which are chosen as

$$\dot{x}_{R_{re}} = \dot{x}_{Rd} + K_1(x_{Rd} - x_R) \quad (5.25)$$

$$\dot{x}_{r_{re}} = \dot{x}_{rd} + K_2(x_{rd} - x_r) \quad (5.26)$$

where  $x_{Rd}$  and  $x_{rd}$  are the desired motions of the right EE and of the left EE with respect to the right EE.  $\dot{u}_{re}$  is the reference acceleration obtained by numerical differentiation and  $\theta_{re}$  is the reference joint position. Recall the unit quaternion tracking error  $e_0 \in \mathbb{R}^4$  defined in Eq. (1.5), the reference angular velocity  $\omega_{rl_{re}}$  is



3. Motion control for the right EE and  $x_r$  which is represented by the black blocks;
4. Object control which is represented by the red blocks.

### 5.3.2.1 Non-holonomic Mobile Platform Control

The robot MDAMS consists of one non-holonomic mobile platform and one upper manipulator system. To achieve the asymptotic position tracking of the non-holonomic mobile platform, recall the control law for the mobile platform

$$\tau_v = \bar{M}_v U_v + \bar{C}_v(\theta_v, \dot{\theta}_v) + \bar{G}_v(\theta) + \bar{Q}_v + \bar{\tau}_{dv}. \quad (5.28)$$

where  $\bar{M}_v$  is the symmetric and positive definite inertia matrix,  $\bar{C}_v(\theta, \dot{\theta})$  is the centripetal and Coriolis matrix,  $\bar{G}_v$  is the friction vector,  $\bar{Q}_v$  is the coupling vector from the upper manipulator,  $\bar{\tau}_{dv}$  is the unknown bounded disturbances including unstructured dynamics and  $\tau_v$  is the input vector. One can convert the dynamic control problem into a kinematic control problem [201], with the nonlinear feedback acceleration control input for the mobile platform being expressed as in Eq. (1.20). The reference nonlinear feedback acceleration input  $U_v$  for the mobile platform in joint space can be obtained according to Eqs. (1.15) and (1.19).

### 5.3.2.2 Object PD Control

Based on the system dynamics in Eq. (2.50) and the object dynamics in Subsection 2.2.3.3, the control law for object manipulation is proposed as follows:

$$\tau = \bar{M}(\theta)U + \bar{C}(\theta, u) + \bar{G}(\theta) + \tau_o + \bar{\tau}_d \quad (5.29)$$

where  $\bar{M}(\theta)$  is the symmetric and positive definite inertia matrix,  $\bar{C}(\theta, u)$  is the centripetal and Coriolis matrix,  $\bar{G}(\theta)$  is the friction vector,  $\bar{\tau}_d$  is the unknown bounded disturbances including unstructured dynamics,  $\tau$  is the input vector,

and  $\tau_o = \begin{bmatrix} \tau_{oR}^T & \tau_{oL}^T \end{bmatrix}^T$  is chosen as:

$$\tau_{oi} = J_i^T (-J_{oi}^T)^{-1} W_i (M_o \ddot{X}_{ore} + C_o + G_o) \quad (5.30)$$

where  $M_o$ ,  $C_o$  and  $G_o$  are the object dynamic matrices defined in Eq. (2.76).  $W_i$  is the weighted matrix to distribute the action of two EEs on object,  $i = (R, L)$ ,  $\ddot{X}_{ore} = \ddot{X}_{od} + k_v(\dot{X}_{od} - \dot{X}_o) + k_p(X_{od} - X_o)$ .  $X_{od}$  is the object's desired motion,  $X_o$  is the object's motion in real time,  $k_v$  and  $k_p$  are the definite positive control gains.

### 5.3.3 Simulation Results

Six trajectory tracking simulations are realized, including linear-sinusoidal and symmetric-asymmetric trajectories tracking, object approaching and manipulation, joint limit avoidance, centralized and semi-decentralized tracking comparison.

#### 5.3.3.1 Linear Trajectory Tracking

We start by tracking the linear trajectories of two EEs. Suppose that the initial pose of the proposed robot MDAMS is known, i.e.  $x_{R_0} = [0.25; 0.24; 0.70]$  and  $x_{L_0} = [0.25; -0.36; 0.70]$ . The simulation time is  $t = 20s$ . When  $t < 10s$ , the desired trajectories of the right and left EEs are respectively  $x_{Rd} = x_{R_0} + [0.01t; 0.01t; 0.05t]$  and  $x_{Ld} = x_{L_0} + [-0.01t; -0.01t; 0.03t]$ . When  $t > 10s$ ,  $x_{Rd} = x_{R_0} + [0.1; 0.1; 0.5]$  and  $x_{Ld} = x_{L_0} + [-0.1; -0.1; 0.3]$ . When  $10s < t < 15s$ , the desired angular velocity of the right and left EEs is  $[\omega_R^T \ \omega_L^T]^T = \frac{\pi}{10} \text{ones}(6, 1)$ , otherwise  $[\omega_R^T \ \omega_L^T]^T = \text{zeros}(6, 1)$ . The control gains are chosen as in Table 5.1.

The tracking results are shown in Fig. 5.2. In detail, Fig. 5.2 (a) shows the desired trajectories and tracking results of the right EE. Fig. 5.2 (b) shows the

Table 5.1 – Control gains.

$K_p$	$K_v$	$K_1$	$K_2$	$K_3$	$K_0$
$10I_3$	$25I_3$	$2.25I_3$	$2.25I_3$	$0.1I_3$	$I_6$
$k_1$	$k_2$	$k_3$	$k_4$	$K_4$	$\alpha$
10	100	100	$\text{diag}(25, 2)$	$0.1I_3$	4

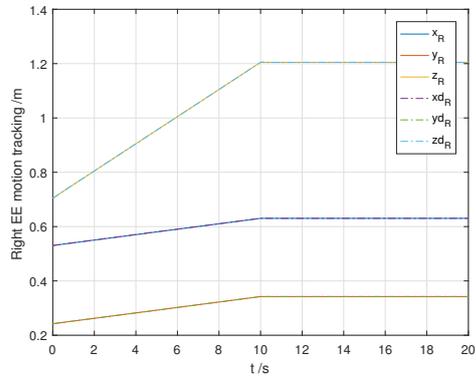
generated reference trajectories and tracking results of the mobile platform. Fig. 5.2 (c) shows the tracking errors of the right EE and the relative motion  $x_r$ . It can be seen that the right EE tracks well its desired trajectory with small tracking errors ( $10^{-4}m$ ) which converge to zero stable. Fig. 5.2 (d) shows the tracking errors of the mobile platform. Though the motion of the mobile platform is not coordinately generated with respect to the EEs, it is well-tracked with the tracking errors converging to zero stable. Fig. 5.2 (e) and Fig. 5.2 (f) show respectively the trajectory tracking results of the mobile platform and two EEs and their projections on the plane  $x$ - $y$ . The attitude tracking results of two EEs are shown in Fig. 5.3, where the green lines represent the desired trajectories. It can be seen that the orientations of EEs are also well tracked.

### 5.3.3.2 Sinusoidal Trajectory Tracking

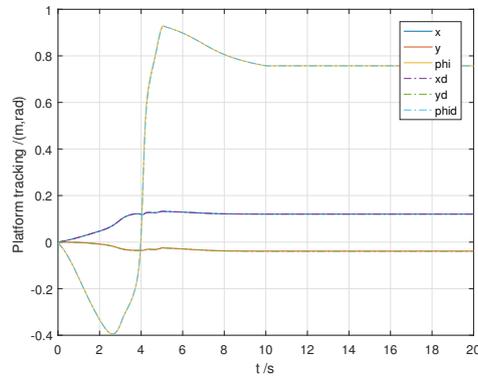
The sinusoidal trajectory tracking of two EEs is realized in this Subsection. Suppose that the initial pose of robot MDAMS is known, i.e. the initial positions of two EEs ( $x_{R_0}$  and  $x_{L_0}$ ) are known. The simulation time is  $t = 20s$ . When  $t < 10s$ ,  $x_{Rd} = x_{R_0} + [0.2\sin(\pi/10t); 0.2\sin(\pi/10t); 0.05]$ ,  $x_{Ld} = x_{L_0} + [0.1\sin(\pi/10t); 0.1\sin(\pi/10t); 0]$  and the desired relative motion is  $x_{rd} = x_{Rd} - x_{Ld}$ , otherwise the velocities are zero. Choose the control gains as in Table 5.1.

The trajectories tracking results are shown in Fig. 5.4. In detail, Fig. 5.4 (a) shows the desired trajectories and tracking results of the right EE. Fig. 5.4 (b) shows the generated reference trajectories and tracking results of the mobile platform. Fig. 5.4 (c) shows that the tracking errors of the right EE and the relative motion  $x_r$  converge to zero stable. Fig. 5.4 (d) shows that the tracking errors of the mobile platform converge to zero stable.

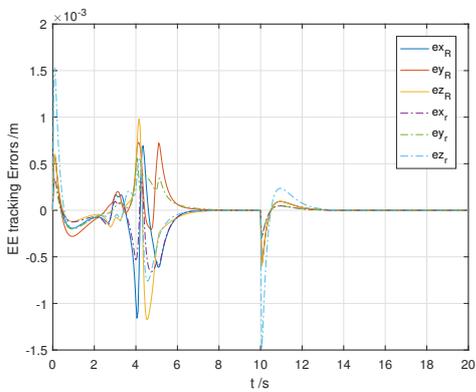
Another four trajectory tracking tasks are realized to validate the effectiveness of the proposed method. In Fig. 5.5 (a), the desired trajectories of two EEs are linear symmetric. The start positions of the desired trajectories are different from the initial positions of the EEs. In Fig. 5.5 (b) and Fig. 5.6 (a), the desired trajectories of two EEs are respectively linear and sinusoidal. In Fig. 5.6 (b), the left EE does not move and the right EE moves along a straight line. The results prove that various tasks can be accomplished with good tracking accuracy.



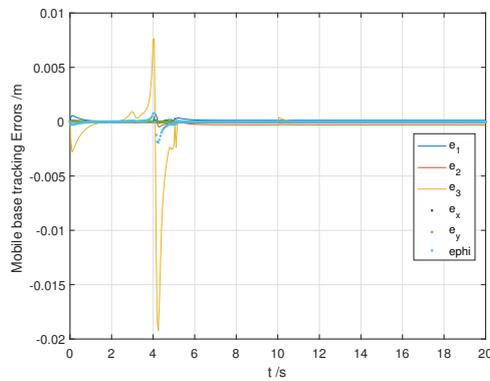
(a) Right EE motion tracking.



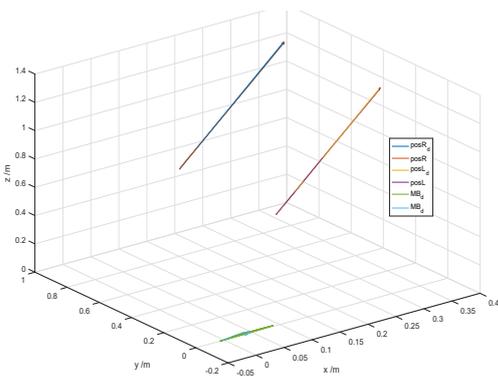
(b) Mobile platform tracking.



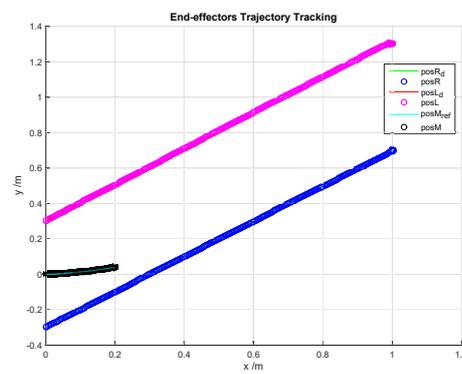
(c) EEs motion tracking errors.



(d) Mobile platform tracking errors.



(e) Trajectories tracking: 3D.



(f) Trajectories tracking: 2D.

Figure 5.2 – Linear trajectory tracking results.

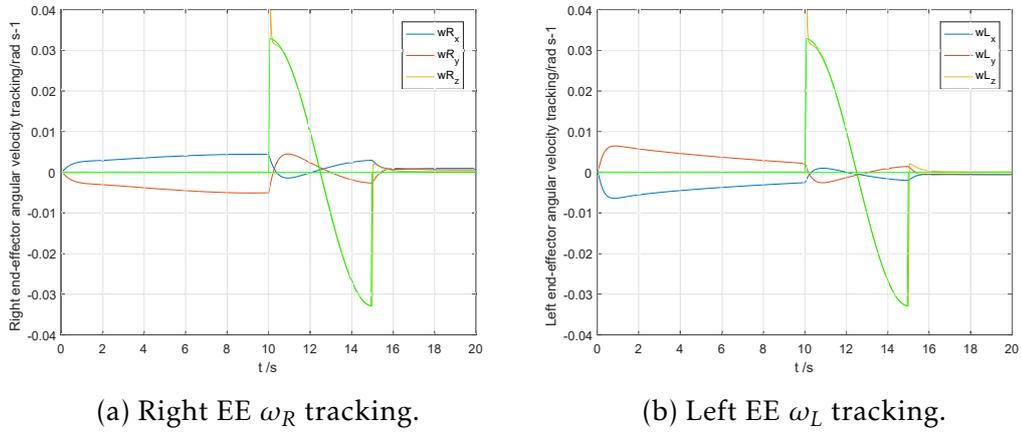


Figure 5.3 – EEs' Attitude tracking results.

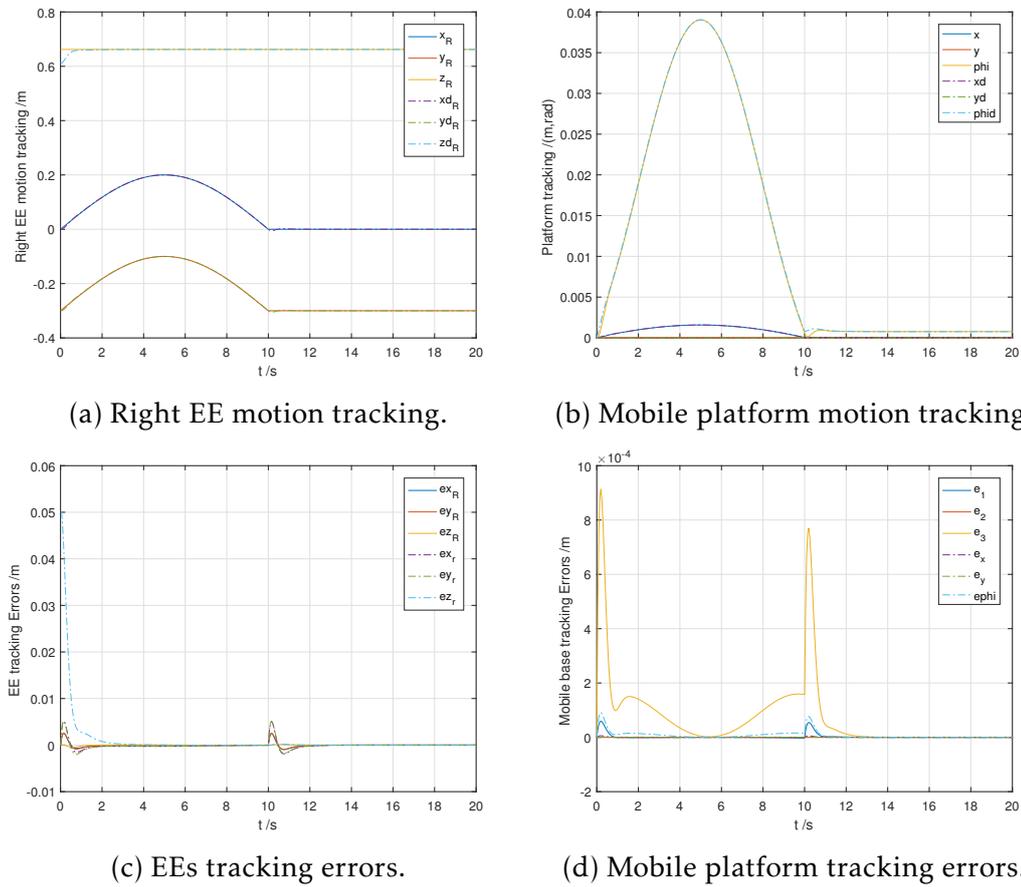
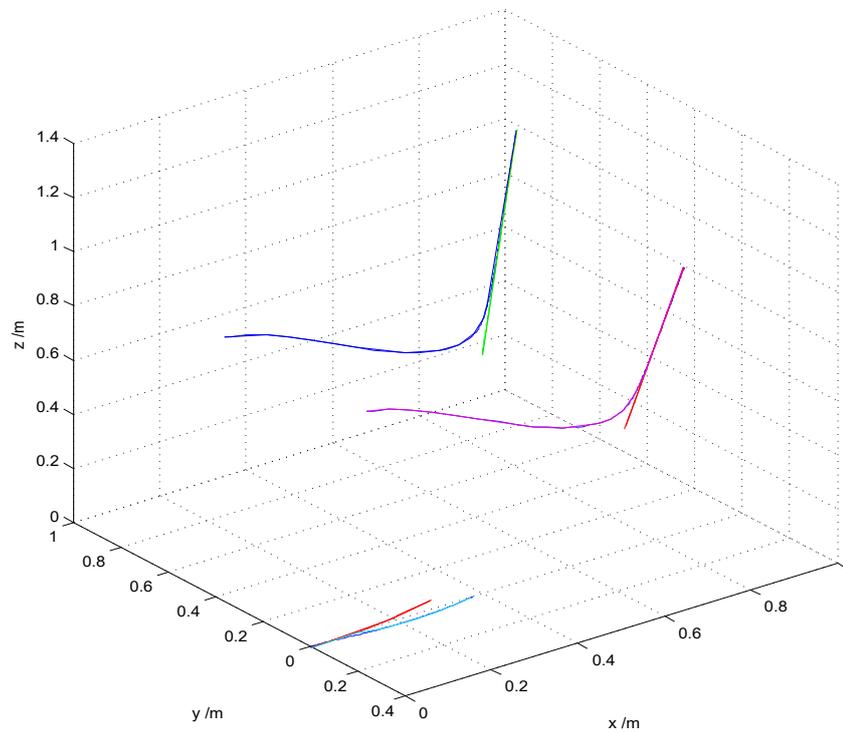
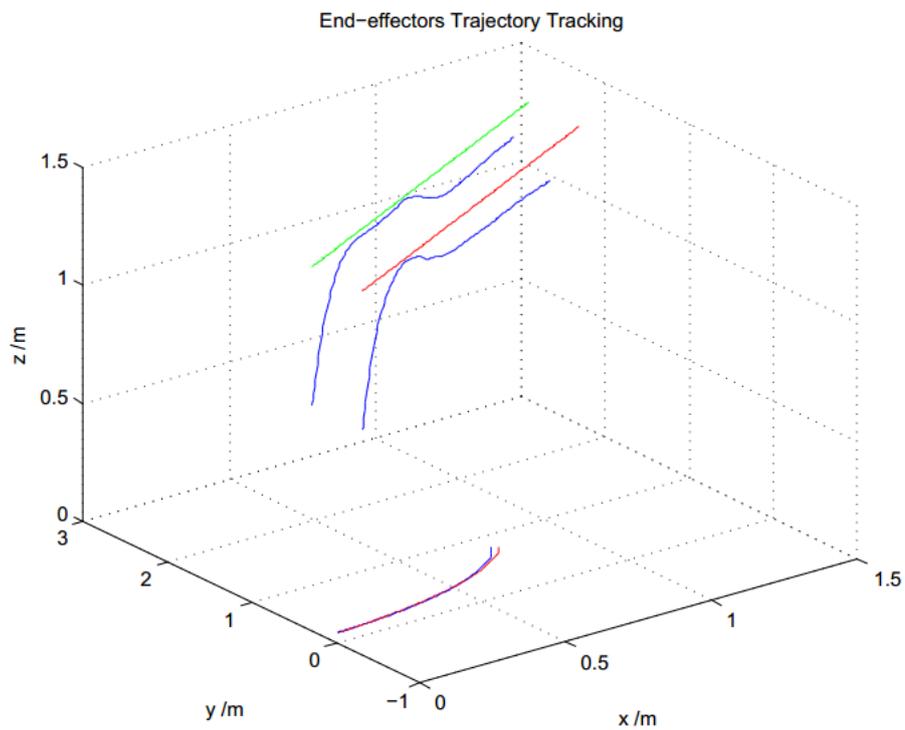


Figure 5.4 – Sinusoidal trajectory tracking results.

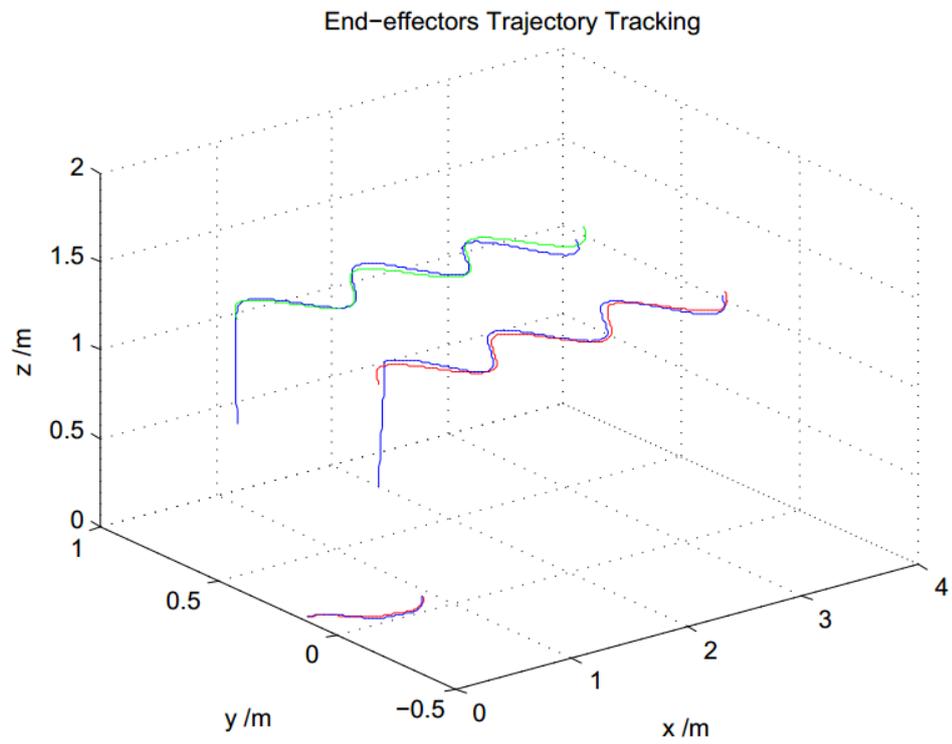


(a) Symmetric dual-linear trajectories.

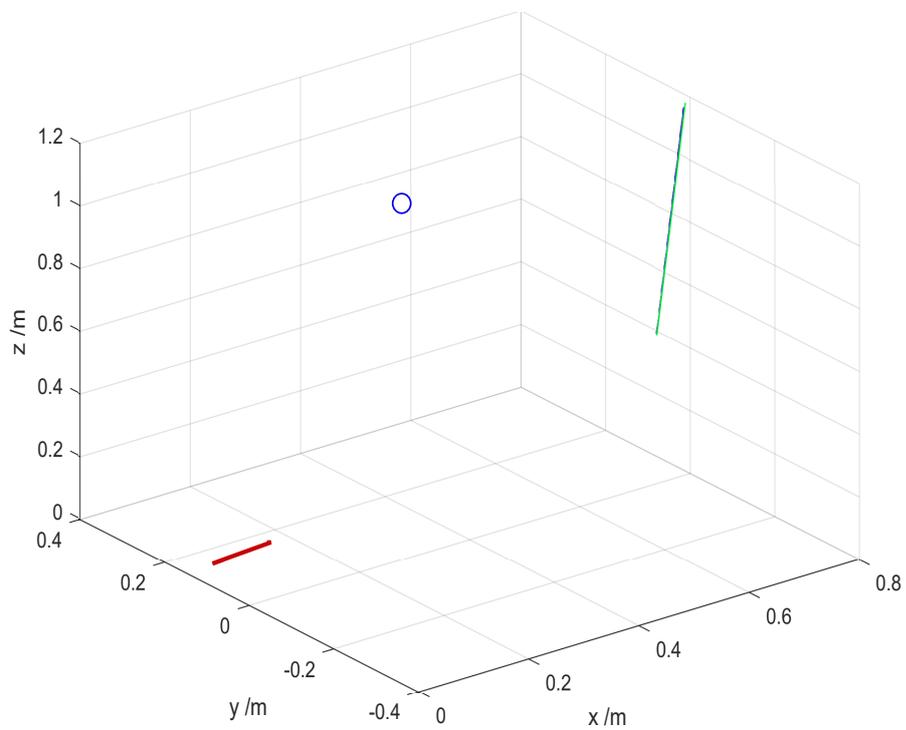


(b) Symmetric dual-linear trajectories.

Figure 5.5 – Various tasks execution: trajectory tracking results (a).



(a) Symmetric dual-sinusoidal trajectories.



(b) Asymmetric trajectories.

Figure 5.6 – Various tasks execution: trajectory tracking results (b).

### 5.3.3.3 Centralized and Semi-decentralized Tracking Comparison

In order to validate the effectiveness of the semi-decentralized control law, the centralized control law is applied for trajectory tracking as a comparison.

Fig. 5.7 (a) and (b) show respectively the trajectory tracking results of the centralized and semi-decentralized control laws. The blue, magenta and black circle lines represent respectively the tracking result of the right EE, the left EE and the mobile platform, the lines represent the corresponding reference trajectories. It proves that the mobile platform's reference motion can be tracked much more precisely using the semi-decentralized control law.

### 5.3.3.4 Object Approaching and Transport

Suppose that the task is to approach a solid object, solidly grasp it and transport it along a desired trajectory. The simulation time is 20s. When  $t < 10s$ ,  $x_{oi} = [0.35; 0.50; 1.312]$ , when  $t \geq 10s$ ,  $x_o = x_{oi} + [0.02(t - 10); 0.02(t - 10); 0]$ . When  $t < 10s$ ,  $\omega_o = [0; 0; 0]$ , when  $10s \leq t \leq 15s$ ,  $\omega_o = [0; 0; \frac{\pi}{60} \sin(\frac{\pi}{5}(t - 10))]$ , when  $t > 15s$ ,  $\omega_o = \text{zeros}(3, 1)$ . Choose the control gains as in Table 5.1.

The simulation results are shown in Fig. 5.8. The trajectory tracking results are shown in Fig. 5.8 (a). The blue dash line represents the desired trajectory of the left EE. The light blue, dark blue and dark purple lines represent the desired trajectories of the right EE, the mobile platform and the object, respectively. The wine red line represents the real trajectory of the left EE. The green, wine red and orange circles represent the real trajectories of the right EE, the mobile platform and the object, respectively. The results show that the mobile platform and two EEs track well the desired trajectories. The tracking errors of the object are shown in Fig. 5.8 (b) which converge to zero stable ( $10^{-3}$ ).

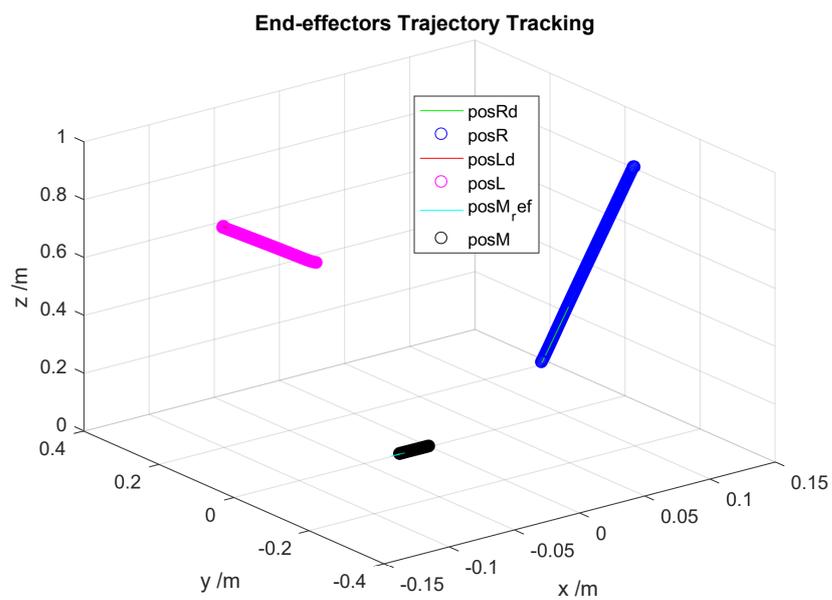
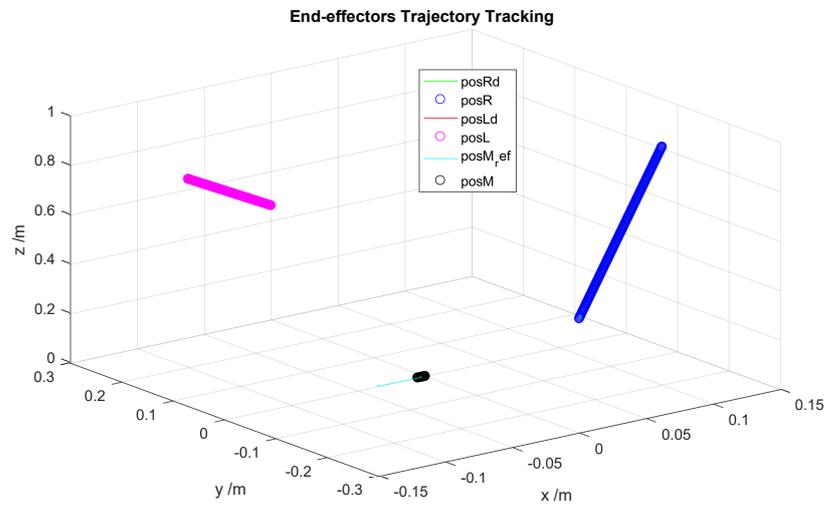
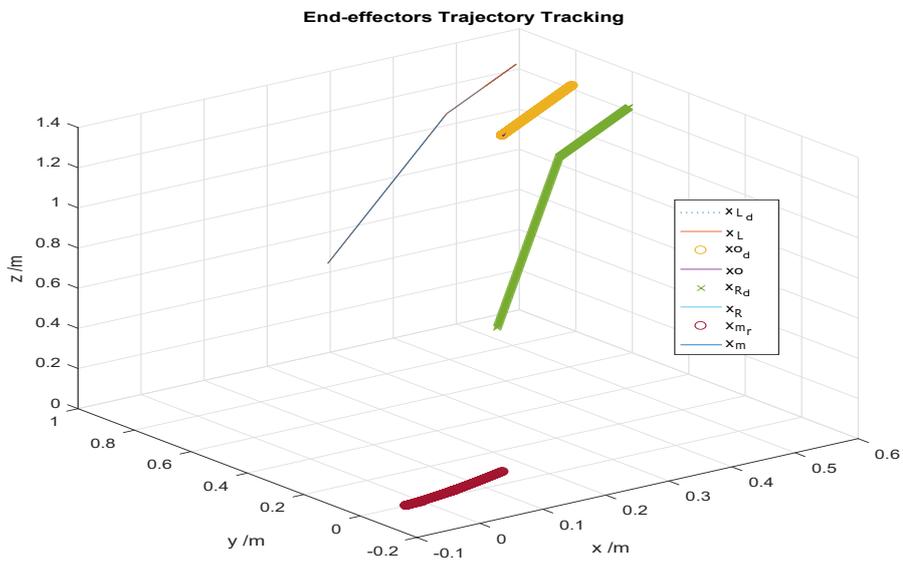
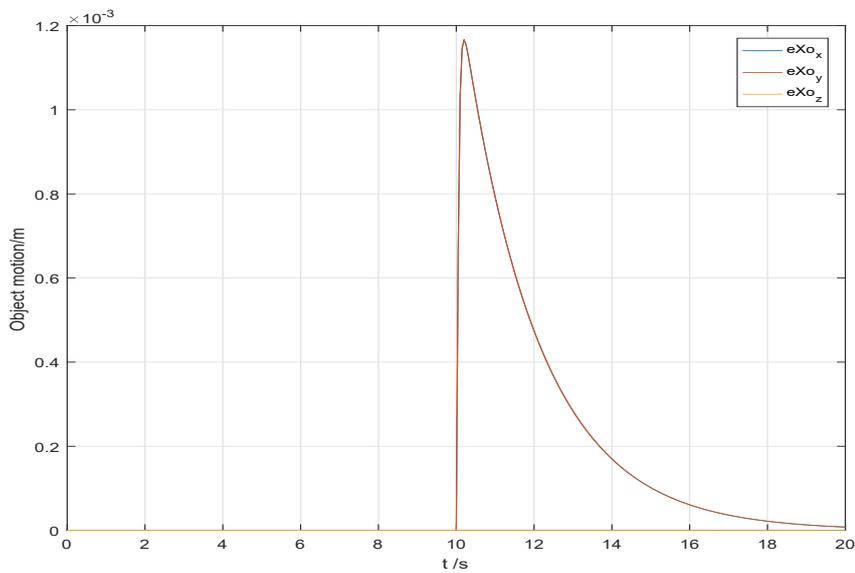


Figure 5.7 – Tracking results of EEs and mobile platform.



(a) Trajectories tracking results.



(b) Trajectories tracking errors.

Figure 5.8 – Object manipulation results.

Table 5.2 – MDAMS joint limit.

Joint	Robot			
	DoFs	a	b	c
Waist	2	$[-90^\circ, 90^\circ]$	$[-10^\circ, 120^\circ]$	-
R/L shoulder	$3 \times 2$	$[-90^\circ, 50^\circ]$	$[-90^\circ, 40^\circ]$	$[-135^\circ, 40^\circ]$
R/L elbow	$1 \times 2$	-	$[-10^\circ, 90^\circ]$	-
R/L wrist	$3 \times 2$	$[-30^\circ, 40^\circ]$	$[-70^\circ, 70^\circ]$	$[-90^\circ, 90^\circ]$

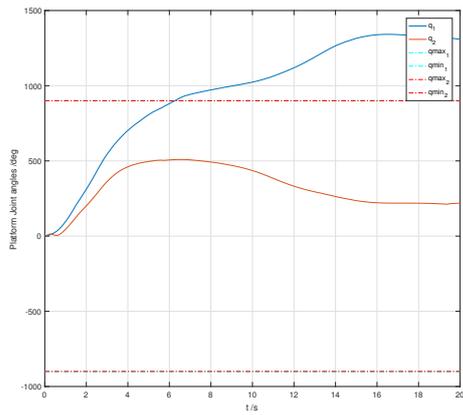
### 5.3.3.5 Trajectory Tracking With Joint Limit Avoidance

In order to dampen the joint positions, we simulate the system by applying the control law Eq. (5.24). Suppose that the desired trajectories of two EEs are known. Table 5.2 lists the joint limits of robot MDAMS. The simulation time is  $t = 20s$ . Choose the control gains as in Table 5.1.

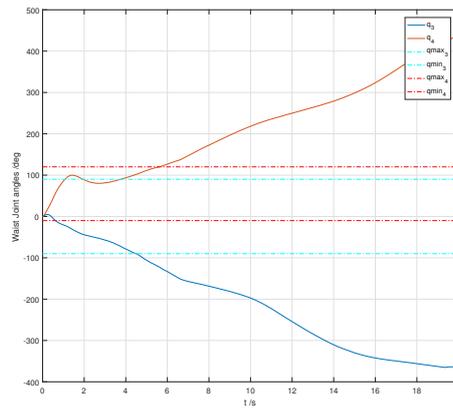
The simulation results are shown in Fig. 5.9, where the horizontal lines represent respectively the upper and lower boundaries of the corresponding joint, the range of two mobile platform wheels is  $[-5\pi, 5\pi]$ . It can be seen that all joint angles are perfectly dampened within their boundaries. However, when the desired EEs position are out of reach, the robot will break. In that case, the range of the mobile platform should be expanded.

### 5.3.3.6 Comparison of Back-stepping and PD Tracking Methods

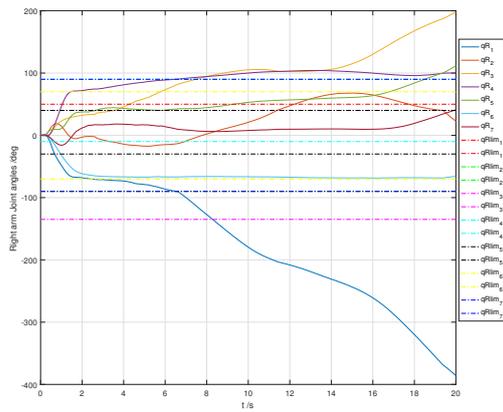
The back-stepping-based controller proposed in Eq. (2.75) is implemented as a comparison. The trajectory tracking of two EEs is realized using controllers Eq. (2.75) and Eq. (5.22) with Eq. (5.24). The tracking results in Fig. 5.10 show that controller Eq. (2.75) responds more quickly. Besides, it is more stable and the tracking errors are smaller than the PD controller Eq. (5.22).



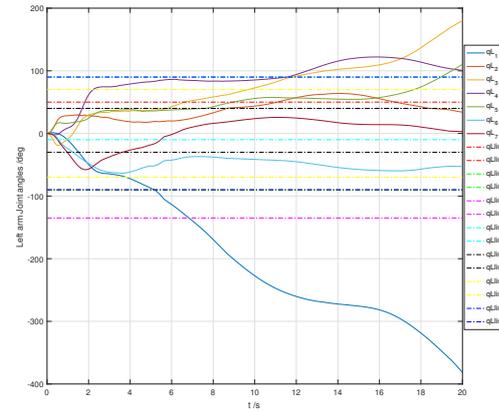
(a) Two wheels' rotation.



(b) Waist rotation.

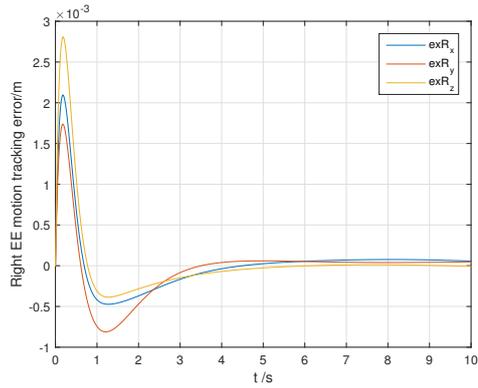


(c) Right arm's joint rotation.

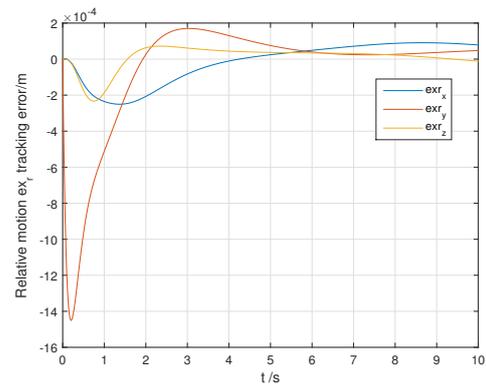
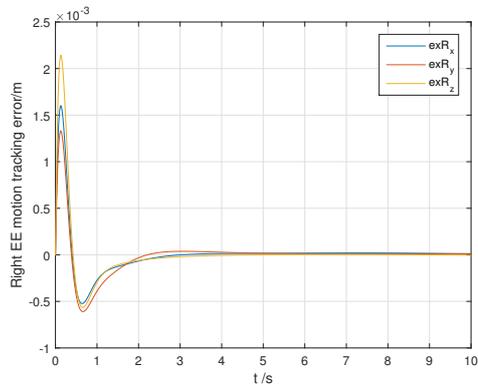


(d) Left arm's joint rotation.

Figure 5.9 – Linear trajectory tracking under joint limit.



(a) PD: right EE tracking error.

(b) PD: relative motion  $x_r$  tracking error.

(c) Back-stepping: right EE tracking error.

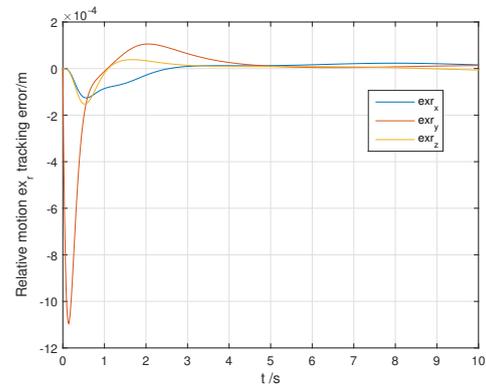
(d) Back-stepping:  $x_r$  tracking error.

Figure 5.10 – Comparison between PD and Back-stepping controllers.

## 5.4 TRIPLE-TRAJECTORY TRACKING

By analyzing the simulation results in Section 5.3, it can be concluded that the trajectories of two EEs are well-tracked. Though the mobile platform tracks well the generated reference trajectory, it can not coordinate its motion cooperatively with respect to EEs. Together with the three subtasks studied in Section 5.3, a new subtask  $x_{R_b}$  is introduced in this section to solve the motion coordination between the mobile platform and the upper manipulator system.

### 5.4.1 Task Priority and Motion Distribution-based IK Solving

Given one target, human beings can quickly figure out the optimal movement strategy in the world frame and with in the body frame simultaneously. Inspired by this phenomenon, a new movement is defined, i.e. the motion of the right EE with respect to the mobile platform (see Fig. 3.1).

Similar to Eq. (2.7), the right EE's motion expressed in robotic body frame {3} can be expressed as  $T_{x_{R_b}} = T_3^5 T_5^e(\theta_w, \theta_R)$  and the corresponding velocity (4-th variable) is obtained as follows:

$$\dot{x}_{R_b} = J_{R_b} \begin{bmatrix} \dot{\theta}_w \\ \dot{\theta}_R \end{bmatrix} = \begin{bmatrix} 0_{n_v} & J_{R_b} & 0_{n_L} \end{bmatrix} \begin{bmatrix} \dot{\theta}_o \\ \dot{\theta}_R \\ \dot{\theta}_L \end{bmatrix} = J_{R_0} u \quad (5.31)$$

where  $J_{R_b}$  is the Jacobian of the right EE in frame {3}. Rewrite Eq. (5.18) as  $u = \tilde{u} + \tilde{I}z^{**}$ , where  $\tilde{I} \triangleq (I - J_r^+ J_r)(I - \tilde{J}_R^+ \tilde{J}_R)(I - \tilde{J}_\omega^+ \tilde{J}_\omega)$ . Define  $J_{rR\omega} \triangleq J_{R_0} \tilde{I}$  and project the fourth variable  $\dot{x}_{R_b}$  into the null space in Eq. (5.18) using the task-priority method, the joint velocity is given as

$$u = \tilde{u} + \tilde{I}J_{rR\omega}^+(\dot{x}_{R_b} - J_{R_0}\tilde{u}) + (I - J_{rR\omega}^+ J_{rR\omega})z^{***}. \quad (5.32)$$

where  $J_{rR\omega}^+$  is the pseudo-inverse of  $J_{rR\omega}$ , again  $z^{***}$  is an  $n$ -dimensional arbitrary joint velocity vector. Transform Eq. (5.31) as

$$\dot{x}_{R_b} = J_{R_0W} u_W \quad (5.33)$$

where  $J_{R_0W} = J_{R_0} W^{-1}$ . Then, Eq. (5.32) becomes

$$\begin{aligned} u &= W^{-1} u_W \\ &= W^{-1} [J_{rW}^+ \dot{x}_r + (I - J_{rW}^+ J_{rW}) \tilde{J}_{RW}^+ (\dot{x}_R - J_{RW} J_{rW}^+ \dot{x}_r) + (I - J_{rW}^+ J_{rW})(I - \tilde{J}_{RW}^+ \tilde{J}_{RW}) \\ &\quad \tilde{J}_{\omega W}^+ (\omega_{rI} - J_{\omega_{rI}W} J_{rW}^+ \dot{x}_r - J_{\omega_{rI}W} (I - J_{rW}^+ J_{rW}) \tilde{J}_{RW}^+ (\dot{x}_R - J_{RW} J_{rW}^+ \dot{x}_r))] \\ &\quad + \tilde{I}W J_{rR\omega W}^+ (\dot{x}_{R_b} - J_{R_0W} \tilde{u}_W) + (I - J_{rR\omega W}^+ J_{rR\omega W}) z^{***} \end{aligned} \quad (5.34)$$

where

$$\begin{aligned} \tilde{u}_W = & J_{r_W}^+ \dot{x}_r + (I - J_{r_W}^+ J_{r_W}) \tilde{J}_{R_W}^+ (\dot{x}_R - J_{R_W} J_{r_W}^+ \dot{x}_r) + (I - J_{r_W}^+ J_{r_W}) (I - \tilde{J}_{R_W}^+ \tilde{J}_{R_W}) \tilde{J}_{\omega_W}^+ \\ & \{\omega_{rl} - J_{\omega_W} [J_{r_W}^+ \dot{x}_r + (I - J_{r_W}^+ J_{r_W}) \tilde{J}_{R_W}^+ (\dot{x}_R - J_{R_W} J_{r_W}^+ \dot{x}_r)]\}. \end{aligned} \quad (5.35)$$

### 5.4.2 Semi-decentralized Triple-trajectory Tracking

Based on Eq. (5.21) and Eq. (2.50), the task oriented PD control law is proposed as follows

$$\bar{\tau} = \bar{M}(\theta) U_t + \bar{C}(\theta, u) + \bar{G}(\theta) + \bar{\tau}_d \quad (5.36)$$

with

$$U_t = \dot{u}_{re} + K_v(u_{re} - u) + K_p(\theta_{re} - \theta) \quad (5.37)$$

where  $\dot{u}_{re}$  is the reference joint acceleration which can be obtained by numerical differentiation of the reference joint velocity  $u_{re}$

$$\begin{aligned} u_{re} = & W^{-1} \{J_{r_W}^+ \dot{x}_{re} + (I - J_{r_W}^+ J_{r_W}) \tilde{J}_{R_W}^+ (\dot{x}_{R_{re}} - J_{R_W} J_{r_W}^+ \dot{x}_{re}) + (I - J_{r_W}^+ J_{r_W}) (I - \tilde{J}_{R_W}^+ \tilde{J}_{R_W}) \\ & \tilde{J}_{\omega_W}^+ [\omega_{rl_{re}} - J_{\omega_{rl_W}} J_{r_W}^+ \dot{x}_{re} - J_{\omega_{rl_W}} (I - J_{r_W}^+ J_{r_W}) \tilde{J}_{R_W}^+ (\dot{x}_{R_{re}} - J_{R_W} J_{r_W}^+ \dot{x}_{re})]\} \\ & + \tilde{I}_W J_{r_{R_W} \omega}^+ (\dot{x}_{R_{b_{re}}} - J_{R_{0W}} \tilde{u}_{re_W}) + (I - J_{r_{R_W} \omega}^+ J_{r_{R_W} \omega}) z^{***} \end{aligned} \quad (5.38)$$

where  $\tilde{u}_{re_W}$  is the corresponding weighed reference velocity vector,  $\dot{x}_{R_{re}}$ ,  $\dot{x}_{r_{re}}$  and  $\omega_{rl_{re}}$  are defined the same as in Eqs. (5.25), (5.26) and (5.27).

$$\dot{x}_{R_{b_{re}}} = \dot{x}_{R_{bd}} + K_3(x_{R_{bd}} - x_{R_b}) \quad (5.39)$$

where  $\dot{x}_{R_{bd}}$  is the corresponding desired trajectory of the subtask.  $K_3$  is a positive definitive gain matrix.

### 5.4.3 Simulation Results

A triple-trajectory tracking task is realized. Suppose that the initial position of the right EE is  $x_{Ri} = (0, -0.3, 0.462)$ , the object to be manipulated locates initially at  $(1, 0, 0.602)$ . The final position of the right EE is  $x_{Rf} = (1, -0.2, 0.602)$ , the initial and final relative positions of the left EE with respect to the right EE

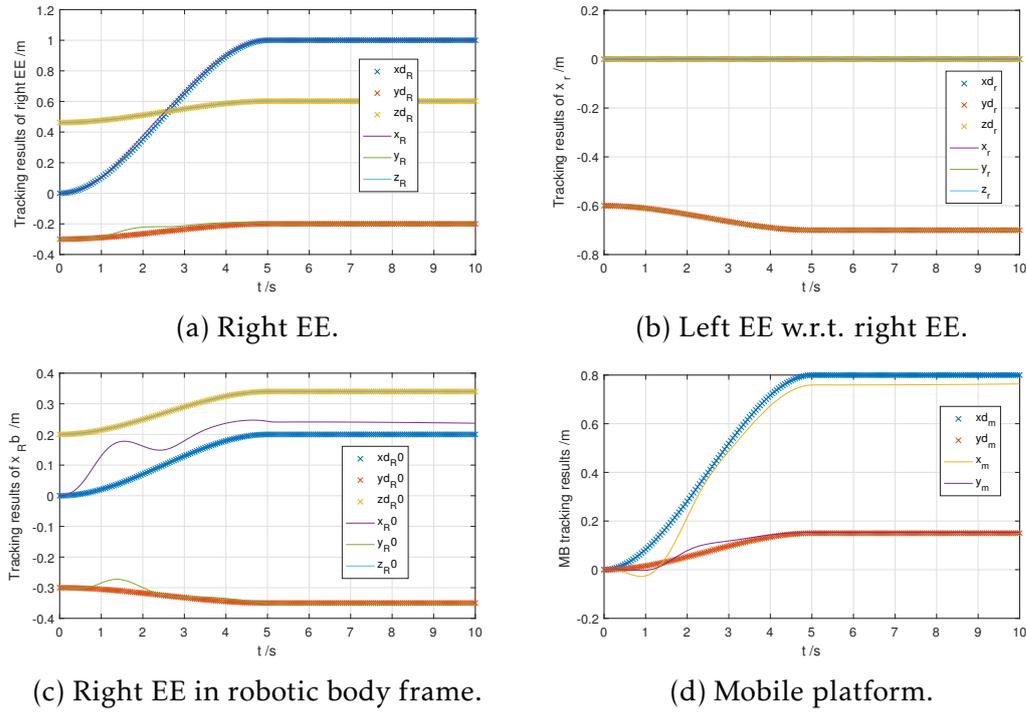


Figure 5.11 – Triple-trajectory tracking results.

are respectively  $x_{r_i} = (0, -0.6, 0)$  and  $x_{r_f} = (0, -0.7, 0)$ . And the initial and final relative positions of the right EE with respect to the mobile platform are respectively  $x_{R_{b_i}} = (0, -0.3, 0.2)$  and  $x_{R_{b_f}} = (0.2, -0.35, 0.34)$ . Then, three continuous trajectories ( $x_{R_d}(t)$ ,  $x_{r_d}(t)$ ,  $x_{R_{b_d}}(t)$ ) are designed using 3-order polynomials (see Fig. 5.11). The objective is to track the three trajectories simultaneously.

Use the proposed control strategy Eq. (5.37), the simulation results are shown in Figs. 5.11 and 5.12. The desired and real trajectories of the right EE in frames  $\{0\}$  and  $\{3\}$ . The desired and real trajectories of the relative motion and the mobile platform are shown in Fig. 5.11. The tracking errors are shown in Fig. 5.12.

It can be seen that the tracking errors of the right and left EEs converge to zero stable. The mobile platform tracks the generated reference trajectory with good accuracy. Since the positions of the right EE in robotic body frame and world frame are determined respectively by  $(\theta_w, \theta_R)$  and  $(\theta_v, \theta_w, \theta_R)$ , there is an unavoidable trade-off between them. Therefore, the tracking accuracy of the

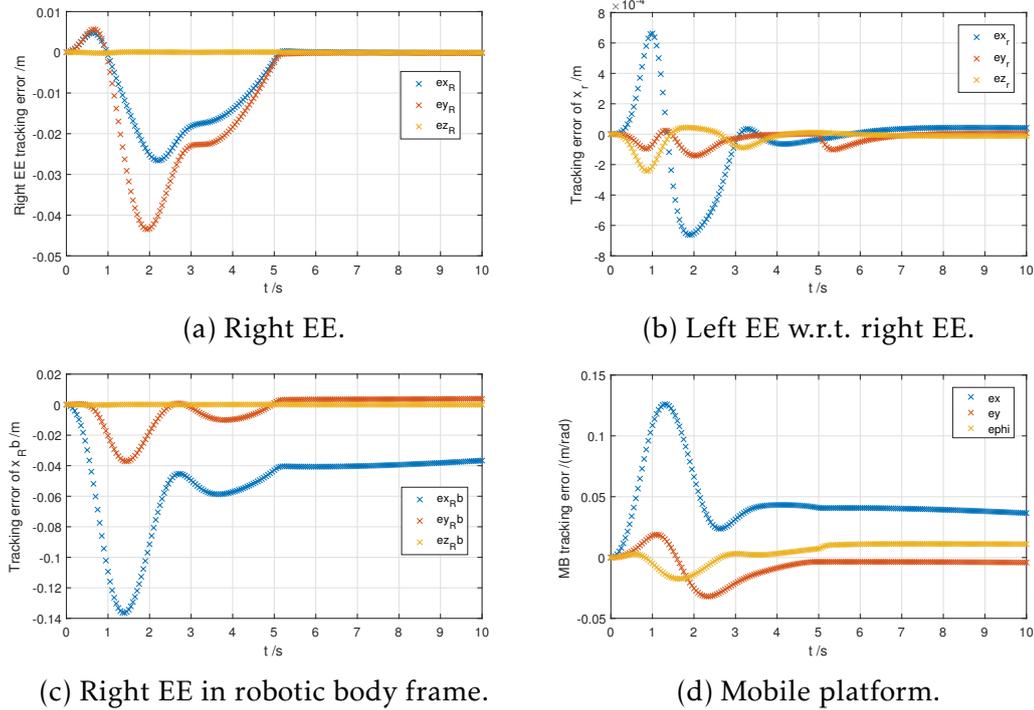


Figure 5.12 – Triple-trajectory tracking errors.

right EE's motion with respect to the robotic body frame  $\{3\}$  is not as good as the motion in the world frame  $\{0\}$  in Fig. 5.11.

The 3D tracking results are shown in Fig. 5.13. The solid lines represent the desired trajectories and the circle lines represent the real trajectories. In detail, the black circle line represents the motion of the mobile platform, the blue circle line represents the motion of the right EE and the rose circle line represents the motion of the left EE. It can be seen that robot MDAMS tracks the trajectories well and the mobile platform navigates cooperatively with respect to two EEs.

Recall the dual-trajectory tracking results presented in Fig. 5.2 as a comparison. The 3D view of dual-trajectory tracking results are shown in Fig. 5.14. It can be seen that two EEs track well the desired trajectories, but the mobile platform does not move cooperatively. Fig. 5.15 shows the results of triple-trajectory tracking projected on the ground. Compared to Fig. 5.3 (d), the mobile platform behaves much better validating that the motion between the mobile platform and two EEs can be controlled coordinately. It can be concluded that

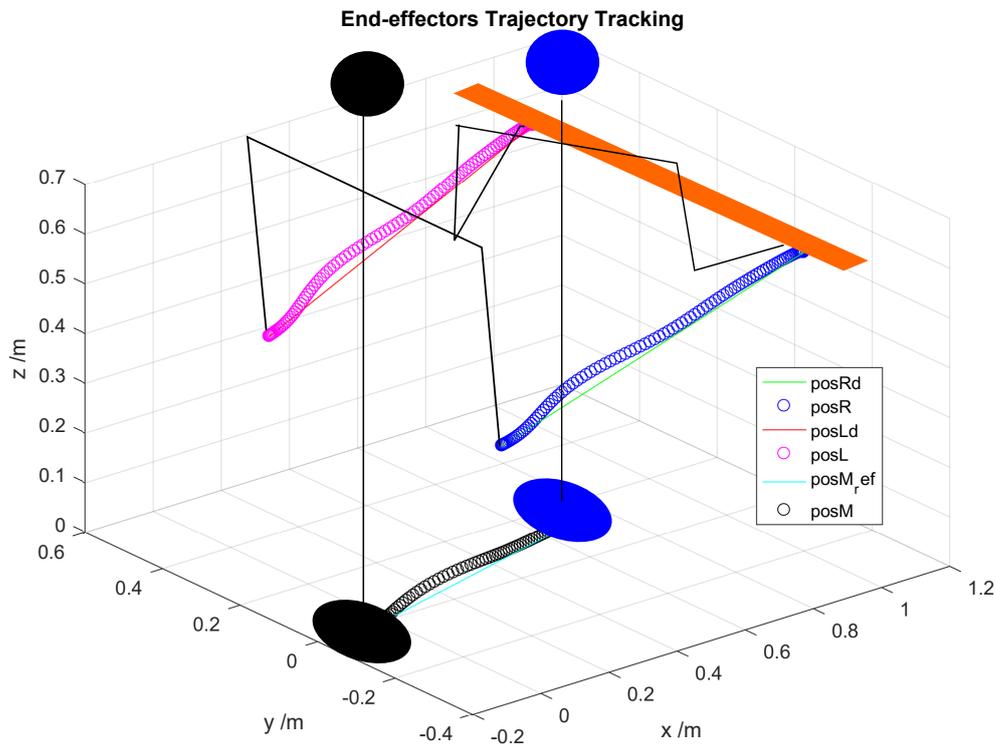


Figure 5.13 – Triple trajectory tracking results: 3D view.

the triple-trajectory tracking achieves motion distribution among the mobile platform and two EEs in such a way that preventing catastrophes.

## 5.5 CONCLUSIONS

A task-priority and relative motion-based trajectory tracking method is proposed in this Chapter for robot MDAMS. A relative motion is introduced to characterize the motion of the left EE with respect to the right EE in the world frame. Another relative motion which characterizes the motion of the right EE in robotic body frame is also introduced. In total, two relative variables are defined and two relative Jacobians are formulated to characterize the coordination and cooperation movements among two arms and the mobile platform. Therefore, the EEs of two arms can maintain a relative configuration with respect to each

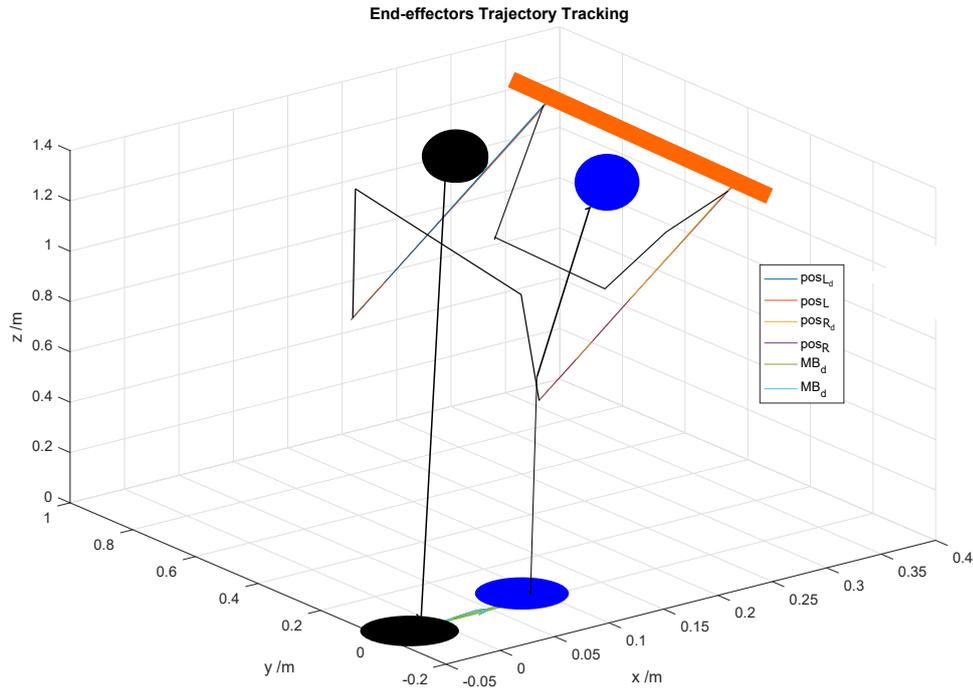


Figure 5.14 – Dual-trajectory tracking: linear trajectory to approach the object.

other and with respect to the robotic body frame. It can be seen that the relative Jacobian can be obtained more conveniently based on the Jacobian of each arm using the MDH method. A semi-decentralized triple-trajectory tracking strategy is proposed with a kinematic controller for the non-holonomic mobile platform. Compared to the dual-trajectory tracking strategy, the triple-trajectory tracking method solves effectively the coordination between the mobile platform and upper manipulator system. Joint limit avoidance is achieved using the mWLN technique. Moreover, object manipulation is realized using relative motion tracking strategy.

This Chapter proves that if only considers the relative motion, the right EE will not move. Besides, under semi-decentralized control strategy, the performance of the mobile platform is much better than the centralized control strategy. Traditional PD controller is used to validate the effectiveness of the proposed redundancy solving strategy. We have proved the effectiveness of the

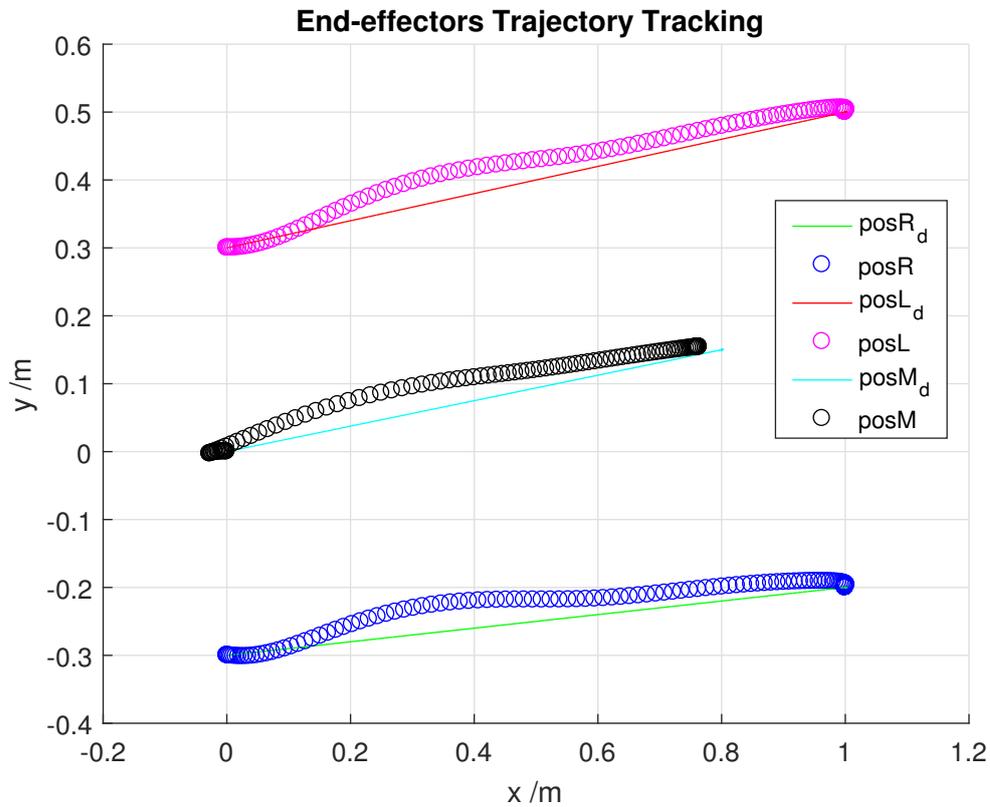


Figure 5.15 – Triple-trajectory tracking results: 2D view.

adaptive robust RBFNN controller in Chapter 2. Therefore, it is straightforward to implement the results in this Chapter into RBFNN for intelligently controlling the proposed robot MDAMS. On the other hand, movements of the waist and shoulder have not been considered. Introduction of relative motion in terms of the waist and shoulder's motions might be a choice. In addition, the hierarchical Fuzzy Logic-based control [202, 207] retrieves the inference structure by introducing an extra reasoning level to reduce the fuzzy base greatly. Therefore, hierarchical type-2 FLC will be another choice for further studies to control robot MDAMS.



# CONCLUSIONS, CONTRIBUTIONS AND FUTURE WORK

## CONCLUSIONS

This thesis studies the kinematic and dynamic modeling, on-line and off-line human-like motion planning and trajectory tracking problems for humanoid robot MDAMS which is designed for personal assistance.

In this thesis, the kinematic model is firstly established using the MDH method with only four parameters. The compact Jacobians are formulated in terms of individual Jacobians. Then, the dynamic model using Lagrangian formulations and the adaptive RBFNN controller are recalled. And the virtual prototype validation via MATLAB/SIMULINK and ADAMS co-simulation is realized to validate the effectiveness of the proposed RBF neural network control law. In order to overcome the shortcomings of Lagrangian formulations requiring the calculation of energy and partial derivatives, Kane's method is used. Physical stability is analyzed based on Kane's model and a control law is proposed using the back-stepping technique. It can be seen that if new end-effectors (like dexterous multi-fingered hands) are installed at the end of two arms it is easier to extend the dynamic model using Kane's method. And it facilitates the investigation of human-robot interaction compared to Lagrange's method. The simulation results validate the equivalence between the two dynamic models.

We have proposed an improved MaxiMin non-dominated sorting genetic algorithm (NSGA-II) to design the human-like optimal pose of MDAMS with only the desired positions of end-effectors being known. Five objective functions

are defined to solve the coupling, joint limit constraint, redundancy and singularity problems at the same time. Compared to the existing MaxiMin NSGA-II algorithm, the proposed algorithm speeds up the evolutionary process by introducing a MaxiMin sorting process for each selected set at each generation. As a result, a Pareto-optimal set is formed consisting of a number of solutions. A normalized combined fitness-based function is defined and a post decision maker is used to select the preferred solution in the Pareto-optimal set. Different from the combined fitness function-based MOGAs which search for the optimal solution in one direction, the proposed algorithm searches for the optimal pose in multiple directions. The mobile platform and the upper manipulator system are treated as a whole system and five objective functions are optimized at the same time. The ability to execute the subsequent tasks is optimized by maximizing the end-effector's manipulability, which is very significant for real applications. Several simulation results are presented to validate the proposed method.

An off-line collision-free motion planning strategy is designed for robot MDAMS to link the initial and designed optimal poses. A direct-connect bidirectional RRT and gradient-descent sampling process is designed. It speeds up the sampling phase by greatly reducing the number of sampled nodes. And a geometric optimization method is proposed which always guarantees the shortest and consistent path. The planned via-poses can be adjusted within their neighborhoods to increase the clearance. Besides, a sequence of orientations are calculated based on the planned via-positions of mobile platform. The linear polynomials with parabolic blends interpolation is used to drive robot MDAMS to head forward between each adjacent via-poses. The effectiveness is validated through numerical simulations.

In addition, an on-line motion planning method is proposed for robot MDAMS. The on-line sensing, collision test and control cycles are defined to achieve collision-free movements in dynamic environments among unforeseen obstacles including human beings. The environmental information is updated and the movements of dynamic obstacles are predicted based on their historical positions in real time. If there are collisions predicted, the on-line re-planning process will be called and the trajectory will be updated immediately. On the

other hand, in order to take the via-poses into consideration, a via-points and MOGA-based motion planning method is proposed. Four objective functions are defined to characterize the candidate via-poses, i.e. (i) joint displacement, (ii) directional manipulability bypassing each via-point, (iii) reaching accuracy to each via-point, and (iv) robot-obstacle intersection. Firstly, the proposed an improved MaxiMin NSGA-II and off-line motion planning methods are used to design the via-points of end-effectors. Secondly, the via-points and MOGA-based method is used to design the corresponding via-poses to the designed via-points. Then, the designed via-poses are linked using the point-to-point motion planning method. Two dynamic virtual environments are established and three motion planning tasks are realized to prove the effectiveness and efficiency of the proposed method.

We have also proposed two task-priority-based semi-decentralized trajectory tracking control laws with a kinematic controller for the non-holonomic mobile platform. Firstly, a dual-trajectory tracking law is presented to track the trajectories of two end-effectors successfully and transport the object with high accuracy. Three ordered variables are introduced. Instead of controlling the absolute motions of end-effectors in task space, a relative motion between two end-effectors is defined. However, in some cases the robot will act strange since the motion distribution among the mobile platform and the upper manipulator is not considered. Therefore, a second relative motion, i.e. the right end-effector's motion with respect to the mobile platform, is defined. As a result, two relative variables are defined to control the coordination and cooperation motions among the mobile platform and two end-effectors, thus forming a triple-trajectory tracking problem. The relative Jacobians are constructed much more conveniently in terms of the known Jacobian of each arm using the MDH method. It is found that if only the relative motion is controlled, the right end-effector will not move, and that the most priori variable has the best tracking accuracy. The proposed semi-decentralized control strategy achieves better performance of the mobile platform than the centralized controller. Besides, coordination of the arms with respect to the mobile platform is realized by tracking the trajectory in robotic body frame.

## CONTRIBUTIONS

In summary, this thesis has the following contributions:

- Kinematic modeling of the designed hyper redundant non-holonomic mobile dual-arm manipulator system (MDAMS) using the Modified Denavit Hartenberg method (Chapter 2).
- Validation of the adaptive RBF neural network controller on the virtual prototype in ADAMS via MATLAB/SIMULINK and ADAMS co-simulation (Chapter 2).
- Simplification of the dynamic modeling process for the designed MDAMS using Kane's method. A Lyapunov stable controller is designed using back-stepping technique and physical stability analysis is presented based on constructed Kane's model (Chapter 2).
- The introduction of an improved MaxiMin NSGA-II algorithm for solving the multi-objective optimization problem. The design of the optimal position-orientation of the mobile platform and the optimal configuration of the upper manipulator by treating them as a whole system. Five objective functions are defined to be optimized at the same time (Chapter 3).
- A direct-connect bidirectional sampling method and a geometric optimization technique are designed. And a sequence of mobile platform's orientations are calculated and the linear polynomials with parabolic blends interpolation is used to lead robot MDAMS always head forward (Chapter 3).
- An on-line motion planning method is presented for the designed robot MDAMS in dynamic environments. On-line sensing, collision test and control cycles are defined to realize on-line motion planning with motion prediction of unknown obstacles in real time (Chapter 4).
- A via-points and MOGA-based motion planning method is proposed for robot MDAMS based on a sequence of planned via-points of end-effectors.

Four objective functions are defined to characterize the corresponding via-poses including the directional manipulability criterion (Chapter 4).

- A task-priority and relative motion-based semi-decentralized trajectory tracking and object manipulation strategy is designed (Chapter 5).
- A motion distribution-based semi-decentralized triple-trajectory tracking strategy is proposed by defining two relative motions including the motion in the robotic body frame (Chapter 5).
- The work in this thesis allows human beings to be more comfortable with the movements of robot MDAMS.

**Validated Tasks** The designed robot MDAMS for personal assistance now can accomplish the following tasks.

- Mobile platform navigation, e.g., path planning and tracking on the ground in static environments;
- Guidance in dynamic environments with unknown moving obstacles including human beings;
- Determination of the optimal position-orientation of mobile platform and the optimal configuration of upper manipulator given a task;
- Dual-arm cooperation with automatic coordination of the mobile platform, e.g., arbitrary motion tracking of two end-effectors, arbitrary relative motion realization between two end-effectors with reasonable mobile platform's coordination;
- Object manipulation, e.g., a solid object reaching, grasping rotation and transport.

## FUTURE WORK

In order to implement the designed redundant non-holonomic robot MDAMS in real construction and applications, there is still a lot of work to do.

**Current and short term work** The motion planning and trajectory tracking methods proposed in this thesis have been validated through numerical simulations. The effectiveness remains to be validated on real robot. However, it takes time to construct a real robot. People can make use of this period to validate the effectiveness of the proposed methods with the aide of simulator. Different environments can be constructed in the simulator. Besides, more finance support will be obtained if visible tasks can be realized. Therefore, virtual prototype validation via MATLAB/SIMULINK and ADAMS co-simulation is under consideration to save time and money.

We are now working on the interval type-2 fuzzy logic (IT2FL) theory to deal with the complexities of system modeling, motion planning and control, and the uncertainties caused by disturbances, sensor-based inputs and actuators. A hierarchical interval type-2 fuzzy logic control law is under consideration to control the mobile platform's navigation, the waist rotation and bend-over movements, end-effectors manipulation and obstacle avoidance at the same time. Different from the proposed open loop motion planning methods in this thesis, the fuzzy theory-based controller will achieve human-like behaviors for MDAMS such as bend-down, turn-back, and other zero-shock movements in a close loop manner. A sequence of human-like behavior rules have been formulated using expert experiences. Besides, intelligent controller combining the fuzzy logic and universal approximation theories will be studied. It is expected to solve the system modeling and human-like motion tracking problems at the same time.

In addition, the adaptive robust RBFNN controller deals with the uncertainties based on Lagrangian formulations, which is somehow model-based. However, the model-based controller may fail in presence of the uncertainties in robot dynamics. Benzaoui [193] proposed an adaptive model-free controller to solve the uncertainties of robotic system using fuzzy logic and universal approximation theories, which could be one of the research topics in the future. To summarize, the following problems are under study or will be studied in the short term.

- Validation of the proposed methods on the virtual prototype constructed in ADAMS via MATLAB/SIMULINK and ADAMS co-simulation.

- Human-like motion planning and tracking in a close loop manner to avoid cognitive shock using the hierarchical type-2 fuzzy logic theory.
- Robot control and uncertainties management by combining the hierarchical type-2 fuzzy logic theory and the universal approximation technique.
- Probabilistic completeness study and complete task policies design, motion planning and accomplishment of complex daily life helping tasks;
- Study of concrete problems such as time delay, state's inaccessibility (observers), joint failure, in real applications;
- Rapid response study to the interference, i.e. fast reconfiguration when there is an interaction;
- More complex and intelligent controller design for robot MDAMS, like extended RBF NN control law, type-2 fuzzy control law, etc.;
- Lie group based system modeling, motion planning and control of robot MDAMS.

### **In the long term**

- Dexterous multi-fingered hands and complex grasping investigation, e.g. study of soft contact at the finger tip;
- Investigation of human-robot interaction, e.g. covering the whole robot with soft skin;
- Real robot construction and validation of the proposed methods in this thesis;
- Long-term service is another topics in personal robots [208], the realization of long-term assistance will be studied.



# Appendix A

## APPENDIX

### A.1 Partial Velocity

#### A.1.1 Partial Velocity Table Construction

Table A.1 lists the origin position vector  $\vec{r}_{p_i}$  of each frame  $\Sigma_i$ . Construct the

Table A.1 – CoM vectors of MDAMS.

Body i	4	6	7	8	9	10	11	12
$\vec{r}_i^*$	$r_4^* \vec{z}_4$	$-r_6^* \vec{z}_6$	$r_7^* \vec{y}_7$	$r_8^* \vec{z}_8$	$r_9^* \vec{x}_9$	$-r_{10}^* \vec{x}_{10}$	$r_{11}^* \vec{y}_{11}$	$r_{12}^* \vec{z}_{12}$
Body i	5	13	14	15	16	17	18	19
$\vec{r}_i^*$	$r_5^* \vec{x}_5$	$r_{13}^* \vec{z}_{13}$	$r_{14}^* \vec{y}_{14}$	$r_{15}^* \vec{z}_{15}$	$r_{16}^* \vec{x}_{16}$	$-r_{17}^* \vec{x}_{17}$	$r_{18}^* \vec{y}_{18}$	$r_{19}^* \vec{z}_{19}$

partial velocity tables in Tables A.2 and A.3, where vectors  $\mathbb{Z}_i$  ( $i = 1, \dots, 75$ ) and  $\mathbb{A}_i$  ( $i = 9, \dots, 75$ ) are defined vectors in Appendix A.1.2.

Table A.2 – Partial velocities of bodies 1-5.

Generalized speeds ( $u_r$ )	$r\vec{v}_1$ $r\vec{\omega}_1$	$r\vec{v}_2$ $r\vec{\omega}_2$	$r\vec{v}_3$ $r\vec{\omega}_3$	$r\vec{v}_4$ $r\vec{\omega}_4$	$r\vec{v}_5$ $r\vec{\omega}_5$
r=1	$\mathbb{Z}_1$ $\mathbb{Z}_3 + \vec{z}_1$	$\mathbb{Z}_2$ $\mathbb{Z}_3 \vec{z}_0$	$\mathbb{Z}_4$ $\mathbb{Z}_3$	$\mathbb{Z}_4$ $1\vec{\omega}_3$	$\mathbb{Z}_5$ $1\vec{\omega}_3$
r=2	$\mathbb{Z}_2$ $-\mathbb{Z}_3 \vec{z}_0$	$\mathbb{Z}_1$ $-\mathbb{Z}_3 + \vec{z}_2$	$\mathbb{Z}_4$ $-\mathbb{Z}_3$	$\mathbb{Z}_4$ $2\vec{\omega}_3$	$\mathbb{Z}_6$ $2\vec{\omega}_3$
r=3	0   0	0   0	0   0	0 $\vec{z}_4$	$\mathbb{Z}_7$ $3\vec{\omega}_4$
r=4	0   0	0   0	0   0	0   0	$\mathbb{Z}_8$ $\vec{z}_5$
r=5-18	0   0	0   0	0   0	0   0	0   0

Table A.3 – Partial velocities of bodies 6-19.

Generalized speeds ( $u_r$ )	$r\vec{v}_6$	$r\vec{\omega}_6$	$r\vec{v}_7$	$r\vec{\omega}_7$	$r\vec{v}_8$	$r\vec{\omega}_8$	$r\vec{v}_9$	$r\vec{\omega}_9$	$r\vec{v}_{10}$	$r\vec{\omega}_{10}$	$r\vec{v}_{11}$	$r\vec{\omega}_{11}$	$r\vec{v}_{12}$	$r\vec{\omega}_{12}$	$r\vec{v}_e$	$r\vec{\omega}_e$
r=1	$\mathbb{Z}_9$	$1\vec{\omega}_5$	$\mathbb{Z}_{14}$	$1\vec{\omega}_6$	$\mathbb{Z}_{20}$	$1\vec{\omega}_7$	$\mathbb{Z}_{27}$	$1\vec{\omega}_8$	$\mathbb{Z}_{35}$	$1\vec{\omega}_9$	$\mathbb{Z}_{44}$	$1\vec{\omega}_{10}$	$\mathbb{Z}_{54}$	$1\vec{\omega}_{11}$	$\mathbb{Z}_{65}$	$1\vec{\omega}_{12}$
r=2	$\mathbb{Z}_{10}$	$2\vec{\omega}_5$	$\mathbb{Z}_{15}$	$2\vec{\omega}_6$	$\mathbb{Z}_{21}$	$2\vec{\omega}_7$	$\mathbb{Z}_{28}$	$2\vec{\omega}_8$	$\mathbb{Z}_{36}$	$2\vec{\omega}_9$	$\mathbb{Z}_{45}$	$2\vec{\omega}_{10}$	$\mathbb{Z}_{55}$	$2\vec{\omega}_{11}$	$\mathbb{Z}_{66}$	$2\vec{\omega}_{12}$
r=3	$\mathbb{Z}_{11}$	$3\vec{\omega}_5$	$\mathbb{Z}_{16}$	$3\vec{\omega}_6$	$\mathbb{Z}_{22}$	$3\vec{\omega}_7$	$\mathbb{Z}_{29}$	$3\vec{\omega}_8$	$\mathbb{Z}_{37}$	$3\vec{\omega}_9$	$\mathbb{Z}_{46}$	$3\vec{\omega}_{10}$	$\mathbb{Z}_{56}$	$3\vec{\omega}_{11}$	$\mathbb{Z}_{67}$	$3\vec{\omega}_{12}$
r=4	$\mathbb{Z}_{12}$	$4\vec{\omega}_5$	$\mathbb{Z}_{17}$	$4\vec{\omega}_6$	$\mathbb{Z}_{23}$	$4\vec{\omega}_7$	$\mathbb{Z}_{30}$	$4\vec{\omega}_8$	$\mathbb{Z}_{38}$	$4\vec{\omega}_9$	$\mathbb{Z}_{47}$	$4\vec{\omega}_{10}$	$\mathbb{Z}_{57}$	$4\vec{\omega}_{11}$	$\mathbb{Z}_{68}$	$4\vec{\omega}_{12}$
r=5	$\mathbb{Z}_{13}$	$\vec{z}_6$	$\mathbb{Z}_{18}$	$5\vec{\omega}_6$	$\mathbb{Z}_{24}$	$5\vec{\omega}_7$	$\mathbb{Z}_{31}$	$5\vec{\omega}_8$	$\mathbb{Z}_{39}$	$5\vec{\omega}_9$	$\mathbb{Z}_{48}$	$5\vec{\omega}_{10}$	$\mathbb{Z}_{58}$	$5\vec{\omega}_{11}$	$\mathbb{Z}_{69}$	$5\vec{\omega}_{12}$
r=6	0	0	$\mathbb{Z}_{19}$	$\vec{z}_7$	$\mathbb{Z}_{25}$	$6\vec{\omega}_7$	$\mathbb{Z}_{32}$	$6\vec{\omega}_8$	$\mathbb{Z}_{40}$	$6\vec{\omega}_9$	$\mathbb{Z}_{49}$	$6\vec{\omega}_{10}$	$\mathbb{Z}_{59}$	$6\vec{\omega}_{11}$	$\mathbb{Z}_{70}$	$6\vec{\omega}_{12}$
r=7	0	0	0	0	$\mathbb{Z}_{26}$	$\vec{z}_8$	$\mathbb{Z}_{33}$	$7\vec{\omega}_8$	$\mathbb{Z}_{41}$	$7\vec{\omega}_9$	$\mathbb{Z}_{50}$	$7\vec{\omega}_{10}$	$\mathbb{Z}_{60}$	$7\vec{\omega}_{11}$	$\mathbb{Z}_{71}$	$7\vec{\omega}_{12}$
r=8	0	0	0	0	0	0	$\mathbb{Z}_{34}$	$\vec{z}_9$	$\mathbb{Z}_{42}$	$8\vec{\omega}_9$	$\mathbb{Z}_{51}$	$8\vec{\omega}_{10}$	$\mathbb{Z}_{61}$	$8\vec{\omega}_{11}$	$\mathbb{Z}_{72}$	$8\vec{\omega}_{12}$
r=9	0	0	0	0	0	0	0	0	$\mathbb{Z}_{43}$	$\vec{z}_{10}$	$\mathbb{Z}_{52}$	$9\vec{\omega}_{10}$	$\mathbb{Z}_{62}$	$9\vec{\omega}_{11}$	$\mathbb{Z}_{73}$	$9\vec{\omega}_{12}$
r=10	0	0	0	0	0	0	0	0	0	0	$\mathbb{Z}_{53}$	$\vec{z}_{11}$	$\mathbb{Z}_{63}$	$10\vec{\omega}_{11}$	$\mathbb{Z}_{74}$	$10\vec{\omega}_{12}$
r=11	0	0	0	0	0	0	0	0	0	0	0	0	$\mathbb{Z}_{64}$	$\vec{z}_{12}$	$\mathbb{Z}_{75}$	$11\vec{\omega}_{12}$
<b>r=12-18</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
Generalized speeds ( $u_r$ )	$r\vec{v}_{13}$	$r\vec{\omega}_{13}$	$r\vec{v}_{14}$	$r\vec{\omega}_{14}$	$r\vec{v}_{15}$	$r\vec{\omega}_{15}$	$r\vec{v}_{16}$	$r\vec{\omega}_{16}$	$r\vec{v}_{17}$	$r\vec{\omega}_{17}$	$r\vec{v}_{18}$	$r\vec{\omega}_{18}$	$r\vec{v}_{19}$	$r\vec{\omega}_{19}$	$r\vec{v}_e$	$r\vec{\omega}_e$
r=1	$\mathbb{A}_9$	$1\vec{\omega}_5$	$\mathbb{A}_{14}$	$1\vec{\omega}_{13}$	$\mathbb{A}_{20}$	$1\vec{\omega}_{14}$	$\mathbb{A}_{27}$	$1\vec{\omega}_{15}$	$\mathbb{A}_{35}$	$1\vec{\omega}_{16}$	$\mathbb{A}_{44}$	$1\vec{\omega}_{17}$	$\mathbb{A}_{54}$	$1\vec{\omega}_{18}$	$\mathbb{A}_{65}$	$1\vec{\omega}_{19}$
r=2	$\mathbb{A}_{10}$	$2\vec{\omega}_5$	$\mathbb{A}_{15}$	$2\vec{\omega}_{13}$	$\mathbb{A}_{21}$	$2\vec{\omega}_{14}$	$\mathbb{A}_{28}$	$2\vec{\omega}_{15}$	$\mathbb{A}_{36}$	$2\vec{\omega}_{16}$	$\mathbb{A}_{45}$	$2\vec{\omega}_{17}$	$\mathbb{A}_{55}$	$2\vec{\omega}_{18}$	$\mathbb{A}_{66}$	$2\vec{\omega}_{19}$
r=3	$\mathbb{A}_{11}$	$3\vec{\omega}_5$	$\mathbb{A}_{16}$	$3\vec{\omega}_{13}$	$\mathbb{A}_{22}$	$3\vec{\omega}_{14}$	$\mathbb{A}_{29}$	$3\vec{\omega}_{15}$	$\mathbb{A}_{37}$	$3\vec{\omega}_{16}$	$\mathbb{A}_{46}$	$3\vec{\omega}_{17}$	$\mathbb{A}_{56}$	$3\vec{\omega}_{18}$	$\mathbb{A}_{67}$	$3\vec{\omega}_{19}$
r=4	$\mathbb{A}_{12}$	$4\vec{\omega}_5$	$\mathbb{A}_{17}$	$4\vec{\omega}_{13}$	$\mathbb{A}_{23}$	$4\vec{\omega}_{14}$	$\mathbb{A}_{30}$	$4\vec{\omega}_{15}$	$\mathbb{A}_{38}$	$4\vec{\omega}_{16}$	$\mathbb{A}_{47}$	$4\vec{\omega}_{17}$	$\mathbb{A}_{57}$	$4\vec{\omega}_{18}$	$\mathbb{A}_{68}$	$4\vec{\omega}_{19}$
<b>r=5-11</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
r=12	$\mathbb{A}_{13}$	$\vec{z}_{13}$	$\mathbb{A}_{18}$	$12\vec{\omega}_{13}$	$\mathbb{A}_{24}$	$12\vec{\omega}_{14}$	$\mathbb{A}_{31}$	$12\vec{\omega}_{15}$	$\mathbb{A}_{39}$	$12\vec{\omega}_{16}$	$\mathbb{A}_{48}$	$12\vec{\omega}_{17}$	$\mathbb{A}_{58}$	$12\vec{\omega}_{18}$	$\mathbb{A}_{69}$	$12\vec{\omega}_{19}$
r=13	0	0	$\mathbb{A}_{19}$	$\vec{z}_{14}$	$\mathbb{A}_{25}$	$13\vec{\omega}_{14}$	$\mathbb{A}_{32}$	$13\vec{\omega}_{15}$	$\mathbb{A}_{40}$	$13\vec{\omega}_{16}$	$\mathbb{A}_{49}$	$13\vec{\omega}_{17}$	$\mathbb{A}_{59}$	$13\vec{\omega}_{18}$	$\mathbb{A}_{70}$	$13\vec{\omega}_{19}$
r=14	0	0	0	0	$\mathbb{A}_{26}$	$\vec{z}_{15}$	$\mathbb{A}_{33}$	$14\vec{\omega}_{15}$	$\mathbb{A}_{41}$	$14\vec{\omega}_{16}$	$\mathbb{A}_{50}$	$14\vec{\omega}_{17}$	$\mathbb{A}_{60}$	$14\vec{\omega}_{18}$	$\mathbb{A}_{71}$	$14\vec{\omega}_{19}$
r=15	0	0	0	0	0	0	$\mathbb{A}_{34}$	$\vec{z}_{16}$	$\mathbb{A}_{42}$	$15\vec{\omega}_{16}$	$\mathbb{A}_{51}$	$15\vec{\omega}_{17}$	$\mathbb{A}_{61}$	$15\vec{\omega}_{18}$	$\mathbb{A}_{72}$	$15\vec{\omega}_{19}$
r=16	0	0	0	0	0	0	0	0	$\mathbb{A}_{43}$	$\vec{z}_{17}$	$\mathbb{A}_{52}$	$16\vec{\omega}_{17}$	$\mathbb{A}_{62}$	$16\vec{\omega}_{18}$	$\mathbb{A}_{73}$	$16\vec{\omega}_{19}$
r=17	0	0	0	0	0	0	0	0	0	0	$\mathbb{A}_{53}$	$\vec{z}_{18}$	$\mathbb{A}_{63}$	$17\vec{\omega}_{18}$	$\mathbb{A}_{74}$	$17\vec{\omega}_{19}$
r=18	0	0	0	0	0	0	0	0	0	0	0	0	$\mathbb{A}_{64}$	$\vec{z}_{19}$	$\mathbb{A}_{75}$	$18\vec{\omega}_{19}$

Table A.4 lists the position vector  $\vec{r}_i^*$  of link  $i$ 's CoM.

Table A.4 – Frame's origin vectors.

Frame $i$	1	5	6	7	8
$\vec{r}_{p_i}$	$-b_v \vec{y}_3 - l_0 \vec{z}_3$	$l_2 \vec{z}_4$	$l_3 \vec{x}_5 - l'_3 \vec{z}_5$	$-l_4 \vec{z}_6$	$l_5 \vec{y}_7$
Frame $i$	2	–	13	14	15
$\vec{r}_{p_i}$	$b_v \vec{y}_3 - l_0 \vec{z}_3$	–	$l_3 \vec{x}_5 + l'_3 \vec{z}_5$	$l_4 \vec{z}_{13}$	$l_5 \vec{y}_{14}$
Frame $i$	9	10	11	12	e
$\vec{r}_{p_i}$	$l_6 \vec{z}_8$	$l_7 \vec{x}_9$	$-l_8 \vec{x}_{10}$	$l_9 \vec{y}_{11}$	$l_{10} \vec{z}_{12}$
Frame $i$	16	17	18	19	e'
$\vec{r}_{p_i}$	$l_6 \vec{z}_{15}$	$l_7 \vec{x}_{16}$	$-l_8 \vec{x}_{17}$	$l_9 \vec{y}_{18}$	$l_{10} \vec{z}_{19}$

### A.1.2 Vectors $\mathbb{Z}_i$ ( $i = 1, \dots, 75$ ) and $\mathbb{A}_i$ ( $i = 9, \dots, 75$ )

#### A.1.2.1 $\mathbb{Z}_i$ ( $i = 1, \dots, 75$ )

$$\begin{aligned} \mathbb{Z}_1 &= r \vec{x}_3, \mathbb{Z}_2 = \vec{0}, \mathbb{Z}_3 = \frac{r_v}{b_v} \vec{z}_3, \mathbb{Z}_4 = \frac{r_v}{2} \vec{x}_3, \mathbb{Z}_5 = \mathbb{Z}_4 + \mathbb{Z}_3 \times \vec{r}_5^*, \\ \mathbb{Z}_6 &= \mathbb{Z}_4 - \mathbb{Z}_3 \times \vec{r}_5^*, \mathbb{Z}_k = \vec{z}_{k-3} \times \vec{r}_5^* \quad (k = 7, 8), \\ \mathbb{Z}_9 &= \mathbb{Z}_4 + \mathbb{Z}_3 \times \mathbb{P}_1, \mathbb{Z}_{10} = \mathbb{Z}_4 - \mathbb{Z}_3 \times \mathbb{P}_1, \mathbb{Z}_k = \vec{z}_{k-7} \times \mathbb{P}_1 \quad (k = 11, 12), \\ \mathbb{Z}_{13} &= \vec{z}_6 \times \vec{r}_6^*, \mathbb{Z}_{14} = \mathbb{Z}_4 + \mathbb{Z}_3 \times \mathbb{P}_2, \mathbb{Z}_{15} = \mathbb{Z}_4 - \mathbb{Z}_3 \times \mathbb{P}_2, \\ \mathbb{Z}_{16} &= \vec{z}_4 \times \mathbb{P}_2, \mathbb{Z}_k = \vec{z}_{k-12} \times \mathbb{P}_2 \quad (k = 17, 18), \\ \mathbb{Z}_{19} &= \vec{z}_7 \times \vec{r}_7^*, \mathbb{Z}_{20} = \mathbb{Z}_4 + \mathbb{Z}_3 \times \mathbb{P}_4, \mathbb{Z}_{21} = \mathbb{Z}_4 - \mathbb{Z}_3 \times \mathbb{P}_4, \\ \mathbb{Z}_{22} &= \vec{z}_4 \times \mathbb{P}_4, \mathbb{Z}_k = \vec{z}_{k-18} \times \mathbb{P}_{k-19} \quad (k = 23, 24, 25), \\ \mathbb{Z}_{26} &= \vec{z}_8 \times \vec{r}_8^*, \mathbb{Z}_{27} = \mathbb{Z}_4 + \mathbb{Z}_3 \times \mathbb{P}_7, \mathbb{Z}_{28} = \mathbb{Z}_4 - \mathbb{Z}_3 \times \mathbb{P}_7, \mathbb{Z}_{29} = \vec{z}_4 \times \mathbb{P}_7, \\ \mathbb{Z}_k &= \vec{z}_{k-25} \times \mathbb{P}_{k-23} \quad (k = 30, \dots, 33), \\ \mathbb{Z}_{34} &= \vec{z}_9 \times \vec{r}_9^*, \mathbb{Z}_{35} = \mathbb{Z}_4 + \mathbb{Z}_3 \times \mathbb{P}_{11}, \mathbb{Z}_{36} = \mathbb{Z}_4 - \mathbb{Z}_3 \times \mathbb{P}_{11}, \\ \mathbb{Z}_{37} &= \vec{z}_4 \times \mathbb{P}_{11}, \mathbb{Z}_k = \vec{z}_{k-33} \times \mathbb{P}_{k-27} \quad (k = 38, \dots, 42), \\ \mathbb{Z}_{43} &= \vec{z}_{10} \times \vec{r}_{10}^*, \mathbb{Z}_{44} = \mathbb{Z}_4 + \mathbb{Z}_3 \times \mathbb{P}_{16}, \mathbb{Z}_{45} = \mathbb{Z}_4 - \mathbb{Z}_3 \times \mathbb{P}_{16}, \\ \mathbb{Z}_{46} &= \vec{z}_4 \times \mathbb{P}_{16}, \mathbb{Z}_k = \vec{z}_{k-42} \times \mathbb{P}_{k-31} \quad (k = 47, \dots, 52), \\ \mathbb{Z}_{53} &= \vec{z}_{11} \times \vec{r}_{11}^*, \mathbb{Z}_{54} = \mathbb{Z}_4 + \mathbb{Z}_3 \times \mathbb{P}_{22}, \mathbb{Z}_{55} = \mathbb{Z}_4 - \mathbb{Z}_3 \times \mathbb{P}_{22}, \mathbb{Z}_{56} = \vec{z}_4 \times \mathbb{P}_{22}, \\ \mathbb{Z}_k &= \vec{z}_{k-52} \times \mathbb{P}_{k-35} \quad (k = 57, \dots, 63), \\ \mathbb{Z}_{64} &= \vec{z}_{12} \times \vec{r}_{12}^*, \mathbb{Z}_{65} = \mathbb{Z}_4 + \mathbb{Z}_3 \times \mathbb{P}_{29}, \mathbb{Z}_{66} = \mathbb{Z}_4 - \mathbb{Z}_3 \times \mathbb{P}_{29}, \mathbb{Z}_{67} = \vec{z}_4 \times \mathbb{P}_{29}, \end{aligned}$$

$$\mathbb{Z}_k = \vec{z}_{k-63} \times \mathbb{P}_{k-39} \quad (k = 68, \dots, 74), \quad \mathbb{Z}_{75} = \vec{z}_{12} \times \vec{r}_e$$

where

$$\mathbb{P}_k = \begin{cases} \vec{r}_6^* + \vec{r}_{p_6} & (k = 1), \\ \vec{r}_7^* + \sum_{i=k+4}^7 \vec{r}_{p_i} & (k = 2, 3), \\ \vec{r}_8^* + \sum_{i=k+2}^8 \vec{r}_{p_i} & (k = 4, \dots, 6), \\ \vec{r}_9^* + \sum_{i=k-1}^9 \vec{r}_{p_i} & (k = 7, \dots, 10), \\ \vec{r}_{10}^* + \sum_{i=k-5}^{10} \vec{r}_{p_i} & (k = 11, \dots, 15), \\ \vec{r}_{11}^* + \sum_{i=k-10}^{11} \vec{r}_{p_i} & (k = 16, \dots, 21), \\ \vec{r}_{12}^* + \sum_{i=k-16}^{12} \vec{r}_{p_i} & (k = 22, \dots, 27), \\ \vec{r}_{12}^* + \sum_{i=k-23}^{12} \vec{r}_{p_i} & (k = 29, \dots, 35). \end{cases}$$

### A.1.2.2 $\mathbb{A}_i$ ( $i = 9, \dots, 64$ )

$$\begin{aligned} \mathbb{A}_9 &= \mathbb{Z}_4 + \mathbb{Z}_3 \times \mathbb{Q}_1, \quad \mathbb{A}_{10} = \mathbb{Z}_4 - \mathbb{Z}_3 \times \mathbb{Q}_1, \\ \mathbb{A}_k &= \vec{z}_{k-7} \times \mathbb{Q}_1 \quad (k = 11, 12), \\ \mathbb{A}_{13} &= \vec{z}_{13} \times \vec{r}_{13}^*, \quad \mathbb{A}_{14} = \mathbb{Z}_4 + \mathbb{Z}_3 \times \mathbb{Q}_2, \quad \mathbb{A}_{15} = \mathbb{Z}_4 - \mathbb{Z}_3 \times \mathbb{Q}_2, \\ \mathbb{A}_{16} &= \vec{z}_4 \times \mathbb{Q}_2, \quad \mathbb{A}_k = \vec{z}_{k-12} \times \mathbb{Q}_2 \quad (k = 17, 18), \quad \mathbb{A}_{19} = \vec{z}_{14} \times \vec{r}_{14}^*, \quad \mathbb{A}_{20} = \mathbb{Z}_4 + \mathbb{Z}_3 \times \mathbb{Q}_4, \\ \mathbb{A}_{21} &= \mathbb{Z}_4 - \mathbb{Z}_3 \times \mathbb{Q}_4, \quad \mathbb{A}_{22} = \vec{z}_4 \times \mathbb{Q}_4, \quad \mathbb{A}_{23} = \vec{z}_5 \times \mathbb{Q}_4, \quad \mathbb{A}_{24} = \vec{z}_{13} \times \mathbb{Q}_5, \quad \mathbb{A}_{25} = \vec{z}_{14} \times \mathbb{Q}_{13}, \\ \mathbb{A}_{26} &= \vec{z}_{15} \times \vec{r}_{15}^*, \quad \mathbb{A}_{27} = \mathbb{Z}_4 + \mathbb{Z}_3 \times \mathbb{Q}_7, \quad \mathbb{A}_{28} = \mathbb{Z}_4 - \mathbb{Z}_3 \times \mathbb{Q}_7, \quad \mathbb{A}_{29} = \vec{z}_4 \times \mathbb{Q}_7, \\ \mathbb{A}_k &= \vec{z}_{k-25} \times \mathbb{Q}_{k-23} \quad (k = 30, \dots, 33), \\ \mathbb{A}_{34} &= \vec{z}_{16} \times \vec{r}_{16}^*, \quad \mathbb{A}_{35} = \mathbb{Z}_4 + \mathbb{Z}_3 \times \mathbb{Q}_{11}, \\ \mathbb{A}_{36} &= \mathbb{Z}_4 - \mathbb{Z}_3 \times \mathbb{Q}_{11}, \quad \mathbb{A}_{37} = \vec{z}_4 \times \mathbb{Q}_{11}, \quad \mathbb{A}_{46} = \vec{z}_4 \times \mathbb{Q}_{16}, \quad \mathbb{A}_k = \vec{z}_{k-33} \times \mathbb{Q}_{k-27} \quad (k = 38, \dots, 42), \\ \mathbb{A}_{43} &= \vec{z}_{17} \times \vec{r}_{17}^*, \quad \mathbb{A}_{44} = \mathbb{Z}_4 + \mathbb{Z}_3 \times \mathbb{Q}_{16}, \quad \mathbb{A}_{45} = \mathbb{Z}_4 - \mathbb{Z}_3 \times \mathbb{Q}_{16}, \\ \mathbb{A}_k &= \vec{z}_{k-42} \times \mathbb{Q}_{k-31} \quad (k = 47, \dots, 52), \\ \mathbb{A}_{53} &= \vec{z}_{18} \times \vec{r}_{18}^*, \quad \mathbb{A}_{54} = \mathbb{Z}_4 + \mathbb{Z}_3 \times \mathbb{Q}_{22}, \quad \mathbb{A}_{55} = \mathbb{Z}_4 - \mathbb{Z}_3 \times \mathbb{Q}_{22}, \\ \mathbb{A}_{56} &= \vec{z}_4 \times \mathbb{Q}_{22}, \quad \mathbb{A}_k = \vec{z}_{k-52} \times \mathbb{Q}_{k-35} \quad (k = 57, \dots, 63), \\ \mathbb{A}_{64} &= \vec{z}_{19} \times \vec{r}_{19}^*, \quad \mathbb{A}_{65} = \mathbb{Z}_4 + \mathbb{Z}_3 \times \mathbb{Q}_{29}, \quad \mathbb{A}_{66} = \mathbb{Z}_4 - \mathbb{Z}_3 \times \mathbb{Q}_{29}, \quad \mathbb{A}_{67} = \vec{z}_4 \times \mathbb{Q}_{29}, \\ \mathbb{A}_k &= \vec{z}_{k-56} \times \mathbb{Q}_{k-39} \quad (k = 68, \dots, 74), \quad \mathbb{A}_{75} = \vec{z}_{19} \times \vec{r}_e \end{aligned}$$

where

$$Q_k = \begin{cases} \vec{r}_{13}^* + \vec{r}_{p_{13}}^* & (k = 1), \\ \vec{r}_{14}^* + \sum_{i=k+12}^{14} \vec{r}_{p_i}^* & (k = 2, 3), \\ \vec{r}_{15}^* + \sum_{i=k+9}^{15} \vec{r}_{p_i}^* & (k = 4, \dots, 6), \\ \vec{r}_{16}^* + \sum_{i=k+6}^{16} \vec{r}_{p_i}^* & (k = 7, \dots, 10), \\ \vec{r}_{17}^* + \sum_{i=k+1}^{17} \vec{r}_{p_i}^* & (k = 11, \dots, 15), \\ \vec{r}_{18}^* + \sum_{i=k-3}^{18} \vec{r}_{p_i}^* & (k = 16, \dots, 21), \\ \vec{r}_{19}^* + \sum_{i=k-9}^{19} \vec{r}_{p_i}^* & (k = 22, \dots, 27), \\ \vec{r}_{e'}^* + \sum_{i=k-16}^{19} \vec{r}_{p_i}^* & (k = 29, \dots, 35). \end{cases}$$

## A.2 Matrix $D \in \mathbb{R}^{n \times n}$

$$\begin{aligned} D_{11} &= \vec{z}_1(\mathbb{Z}_3 + \vec{z}_1), \quad D_{12} = \vec{z}_2\mathbb{Z}_3, \quad D_{1k} = \vec{z}_{(k+1)}\mathbb{Z}_3 \quad (k = 3, \dots, 18), \\ D_{21} &= \vec{z}_1(-\mathbb{Z}_3), \quad D_{22} = \vec{z}_2(-\mathbb{Z}_3 + \vec{z}_2), \quad D_{2k} = \vec{z}_{(k+1)}(-\mathbb{Z}_3) \quad (k = 3, \dots, 18), \\ D_{kk} &= 1 \quad (k = 3, \dots, 18), \\ D_{34} &= \vec{z}_5\vec{z}_4, \\ D_{3k} &= \vec{z}_{(k+1)}\vec{z}_4 \quad (k = 13, \dots, 18), \\ D_{4k} &= \vec{z}_{(k+1)}\vec{z}_5 \quad (k = 13, \dots, 18), \\ D_{5k} &= \vec{z}_{(k+1)}\vec{z}_6 \quad (k = 6, \dots, 11), \\ D_{6k} &= \vec{z}_{(k+1)}\vec{z}_7 \quad (k = 7, \dots, 11), \\ D_{7k} &= \vec{z}_{(k+1)}\vec{z}_8 \quad (k = 8, \dots, 11), \\ D_{8k} &= \vec{z}_{(k+1)}\vec{z}_9 \quad (k = 9, \dots, 11), \\ D_{9k} &= \vec{z}_{(k+1)}\vec{z}_{10} \quad (k = 10, 11), \\ D_{10k} &= \vec{z}_{(k+1)}\vec{z}_{11} \quad (k = 11), \\ D_{12k} &= \vec{z}_{(k+1)}\vec{z}_{13} \quad (k = 13, \dots, 18), \\ D_{13k} &= \vec{z}_{(k+1)}\vec{z}_{14} \quad (k = 14, \dots, 18), \\ D_{14k} &= \vec{z}_{(k+1)}\vec{z}_{15} \quad (k = 15, \dots, 18), \\ D_{15k} &= \vec{z}_{(k+1)}\vec{z}_{16} \quad (k = 16, \dots, 18), \\ D_{16k} &= \vec{z}_{(k+1)}\vec{z}_{17} \quad (k = 17, 18), \\ D_{17k} &= \vec{z}_{(k+1)}\vec{z}_{18} \quad (k = 18), \end{aligned}$$

$D_{jk} = 0$ , otherwise.

### A.3 Matrix $B \in \mathbb{R}^{n \times 1}$

$$B_4 = \sum_{i=5}^{19} m_i g \vec{z}_0(4 \vec{v}_i),$$

$$B_6 = \sum_{i=7}^{12} m_i g \vec{z}_0(6 \vec{v}_i),$$

$$B_8 = \sum_{i=9}^{12} m_i g \vec{z}_0(8 \vec{v}_i),$$

$$B_9 = \sum_{i=10}^{12} m_i g \vec{z}_0(9 \vec{v}_i),$$

$$B_{10} = \sum_{i=11}^{12} m_i g \vec{z}_0(10 \vec{v}_i),$$

$$B_{13} = \sum_{i=14}^{19} m_i g \vec{z}_0(13 \vec{v}_i),$$

$$B_{15} = \sum_{i=16}^{19} m_i g \vec{z}_0(15 \vec{v}_i),$$

$$B_{16} = \sum_{i=17}^{19} m_i g \vec{z}_0(16 \vec{v}_i),$$

$$B_{17} = \sum_{i=18}^{19} m_i g \vec{z}_0(17 \vec{v}_i),$$

$B_j = 0$ , otherwise.

# Bibliography

- [1] Yang Qian. “Design and Control of a Personal Assistant Robot”. PhD thesis. France: Ecole Centrale de Lille, 2013. 214 pp.
- [2] Sungchul Kang, Munsang Kim, Chong-Won Lee, and Kyo-Il Lee. “A target approachable force-guided control for complex assembly”. *1998 IEEE Int. Conf. Robotics and Automation, 1998. Proceedings*. Vol. 1. IEEE. 1998, pp. 826–831.
- [3] Wenfu Xu, Yu She, and Yangsheng Xu. “Analytical and semi-analytical inverse kinematics of SSRMS-type manipulators with single joint locked failure”. *Acta Astronautica* 105.1 (2014), pp. 201–217.
- [4] M. A. Diftler et al. “Robonaut 2 - The first humanoid robot in space”. *2011 IEEE Int. Conf. Robotics and Automation*. 2011, pp. 2178–2183.
- [5] R. O. Ambrose et al. “Robonaut: NASA’s space humanoid”. *IEEE Intelligent Systems and their Applications* 15.4 (2000), pp. 57–63.
- [6] S. Hong, W. S. Lee, Y. S. Kang, and Y. W. Park. “Kinematic control algorithms and robust controller design for rescue robot”. *2014 14th Int. Conf. Control, Automation and Systems (ICCAS 2014)*. 2014, pp. 637–642.
- [7] Jun-tao Li and Hong-jian Liu. “Design Optimization of Amazon Robotics”. *Automation, Control and Intelligent Systems* 4.2 (2016), p. 48.
- [8] N. Kanehira et al. “Design and experiments of advanced leg module (HRP-2L) for humanoid robot (HRP-2) development”. *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. Vol. 3. 2002, 2455–2460 vol.3.
- [9] O. Khatib et al. “Ocean One: A Robotic Avatar for Oceanic Discovery”. *IEEE Robotics Automation Magazine* 23.4 (2016), pp. 20–29.
- [10] Christopher Korpela, Matko Orsag, and Paul Oh. “Towards valve turning using a dual-arm aerial manipulator”. *2014 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. IEEE, 2014, pp. 3411–3416.
- [11] Siddhartha Srinivasa et al. “The robotic busboy: Steps towards developing a mobile robotic home assistant” (2008).

- [12] Fuhai Zhang, Lei Hua, Yili Fu, and Bin Guo. “Dynamic simulation and analysis for bolt and nut mating of dual arm robot”. *Robotics and Biomimetics (ROBIO), 2012 IEEE Int. Conf. IEEE*, 2012, pp. 660–665.
- [13] Andreas Doumanoglou, Andreas Kargakos, Tae-Kyun Kim, and Sotiris Malassiotis. “Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning”. *2014 IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE, 2014, pp. 987–993.
- [14] Francesca Bertacchini, Eleonora Bilotta, and Pietro Pantano. “Shopping with a robotic companion”. *Computers in Human Behavior* (2017).
- [15] Hooman Lee, Joongbae Kim, and Taewoo Kim. “A robot teaching framework for a redundant dual arm manipulator with teleoperation from exoskeleton motion data”. *2014 IEEE-RAS Int. Conf. Humanoid Robots*. IEEE, 2014, pp. 1057–1062.
- [16] Tomoyuki Miyoshi, Hirotake Yamazoe, and Joo-Ho Lee. “Natural behavior based teleoperation for dual robot manipulators mounted on a wheelchair”. *2015 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2015, pp. 888–893.
- [17] Yong-Ho Seo, Il-Woong Jeong, and Hyun S. Yang. “Motion capture-based wearable interaction system and its application to a humanoid robot, AMIO”. *Advanced Robotics* 21.15 (2007), pp. 1725–1741.
- [18] Scott Kuindersma et al. “Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot”. *Auton Robot* 40.3 (2016), pp. 429–455.
- [19] R. Gerndt, D. Seifert, J. H. Baltes, S. Sadeghnejad, and S. Behnke. “Humanoid Robots in Soccer: Robots Versus Humans in RoboCup 2050”. *IEEE Robotics Automation Magazine* 22.3 (2015), pp. 147–154.
- [20] S. Shamsuddin et al. “Humanoid robot NAO: Review of control and motion exploration”. *2011 IEEE Int. Conf. Control System, Computing and Engineering*. 2011, pp. 511–516.
- [21] J. Lafaye, D. Gouaillier, and P. B. Wieber. “Linear model predictive control of the locomotion of Pepper, a humanoid robot with omnidirectional wheels”. *2014 IEEE-RAS Int. Conf. Humanoid Robots*. 2014, pp. 336–341.
- [22] H. S. Yang et al. “Design and Development of Biped Humanoid Robot, AMI2, for Social Interaction with Humans”. *2006 6th IEEE-RAS Int. Conf. Humanoid Robots*. 2006, pp. 352–357.

- [23] B. Graf, U. Reiser, M. Hägele, K. Mauz, and P. Klein. “Robotic home assistant Care-O-bot 3 - product vision and innovation platform”. *2009 IEEE Workshop on Advanced Robotics and its Social Impacts*. 2009, pp. 139–144.
- [24] B. Cohen, S. Chitta, and M. Likhachev. “Search-based planning for dual-arm manipulation with upright orientation constraints”. *2012 IEEE Int. Conf. Robotics and Automation (ICRA)*. 2012, pp. 3784–3790.
- [25] Jonathan Scholz, Sachin Chitta, Bhaskara Marthi, and Maxim Likhachev. “Cart pushing with a mobile manipulation system: Towards navigation with moveable objects”. *Robotics and Automation (ICRA), 2011 IEEE Int. Conf. IEEE*, 2011, pp. 6115–6120.
- [26] Hiroyasu Iwata and Shigeki Sugano. “Design of human symbiotic robot TWENDY-ONE”. *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on. IEEE*. 2009, pp. 580–586.
- [27] Tamim Asfour et al. “ARMAR-III: An integrated humanoid platform for sensory-motor control”. *Humanoid Robots, 2006 6th IEEE-RAS International Conference on. IEEE*. 2006, pp. 169–175.
- [28] N. Vahrenkamp, T. Asfour, and R. Dillmann. “Simultaneous Grasp and Motion Planning: Humanoid Robot ARMAR-III”. *IEEE Robotics Automation Magazine* 19.2 (2012), pp. 43–57.
- [29] Moju Zhao et al. “Design, Modeling, and Control of an Aerial Robot DRAGON: A Dual-Rotor-Embedded Multilink Robot With the Ability of Multi-Degree-of-Freedom Aerial Transformation”. *IEEE Robotics and Automation Letters* 3.2 (2018), pp. 1176–1183.
- [30] Daniel Sidobre et al. “Human–Robot Interaction”. *Advanced Bimanual Manipulation*. Springer Tracts in Advanced Robotics. Springer, Berlin, Heidelberg, 2012, pp. 123–172.
- [31] R. Zhao and D. Sidobre. “Trajectory smoothing using jerk bounded shortcuts for service manipulator robots”. *2015 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*. 2015, pp. 4929–4934.
- [32] J. Butterfass, M. Grebenstein, H. Liu, and G. Hirzinger. “DLR-Hand II: next generation of a dextrous robot hand”. *Proceedings 2001 ICRA. IEEE Int. Conf. Robotics and Automation (Cat. No.01CH37164)*. Vol. 1. 2001, 109–114 vol.1.
- [33] Ch Ott et al. “A humanoid two-arm system for dexterous manipulation”. *2006 6th IEEE-RAS Int. Conf. Humanoid Robots. IEEE*, 2006, pp. 276–283.

- [34] Richard M. Murray, Zexiang Li, S. Shankar Sastry, and S. Shankara Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.
- [35] John J. Craig. *Introduction to Robotics: Mechanics and Control*. 2nd. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.
- [36] Bruno Siciliano and Oussama Khatib, eds. *Springer handbook of robotics*. OCLC: 244008134. Berlin: Springer, 2008. 1611 pp.
- [37] T. Fernando, J. Chandiramani, T. Lee, and H. Gutierrez. “Robust adaptive geometric tracking controls on  $SO(3)$  with an application to the attitude dynamics of a quadrotor UAV”. *2011 50th IEEE Conference on Decision and Control and European Control Conference*. 2011, pp. 7380–7385.
- [38] Qing Gao, Jinguo Liu, Tongtong Tian, and Yangmin Li. “Free-flying dynamics and control of an astronaut assistant robot based on fuzzy sliding mode algorithm”. *Acta Astronautica*. The Fifth Int. Conf. Tethers in Space 138 (Supplement C 2017), pp. 462–474.
- [39] Yoshio Yamamoto. “Control and Coordination of Locomotion and Manipulation of a Wheeled Mobile Manipulator”. PhD Thesis. Philadelphia, PA, USA: University of Pennsylvania, 1994.
- [40] M. H. Korayem, M. Nazemizadeh, and V. Azimirad. “Optimal trajectory planning of wheeled mobile manipulators in cluttered environments using potential functions”. *Scientia Iranica* 18.5 (2011), pp. 1138–1147.
- [41] Barmak Baigzadehnoe, Zahra Rahmani, Alireza Khosravi, and Behrooz Rezaie. “On position/force tracking control problem of cooperative robot manipulators using adaptive fuzzy backstepping approach”. *ISA Transactions* 70 (Supplement C 2017), pp. 432–446.
- [42] T. R. Kane, P. W. Likins, and D. A. Levinson. *Spacecraft dynamics*. 1983.
- [43] Thomas R. Kane and David A. Levinson. “The Use of Kane’s Dynamical Equations in Robotics”. *The International Journal of Robotics Research* 2.3 (1983), pp. 3–21.
- [44] Ronald L. Huston. “Multibody Dynamics Formulations Via Kane’s Equations”. *Mechanics And Control Of Large Flexible Structures*. 0 vols. Progress in Astronautics and Aeronautics. American Institute of Aeronautics and Astronautics, 1990, pp. 71–86.
- [45] A. Purushotham, A. Purushotham, and A. Purushotham A. Purushotham. “Kane’s Method for Robotic Arm Dynamics: a Novel Approach”. *IOSR Journal of Mechanical and Civil Engineering* 6 (2017), pp. 7–13.

- [46] Y. Ke, W. Xuyang, G. Tong, and W. Chao. "A Dynamic Model of ROV with a Robotic Manipulator Using Kane's Method". *2013 Fifth Int. Conf. Measuring Technology and Mechatronics Automation*. 2013, pp. 9–12.
- [47] D. R. Isenberg. "Control of systems subject to Pfaffian constraints: motivation for Kane's formulation". *SoutheastCon 2016*. 2016, pp. 1–8.
- [48] Frank C. Park, James E. Bobrow, and Scott R. Ploen. "A Lie group formulation of robot dynamics". *The International Journal of Robotics Research* 14.6 (1995), pp. 609–618.
- [49] Frank C. Park and James E. Bobrow. "A recursive algorithm for robot dynamics using lie groups". *Robotics and Automation, 1994. Proceedings., 1994 IEEE Int. Conf. IEEE*, 1994, pp. 1535–1540.
- [50] L. Li, X. Hongwei, Z. Qinghua, D. Ya, and W. Changhong. "Modeling and Simulation of Space Manipulator Based on Lie Group". *2016 Sixth Int. Conf. Instrumentation Measurement, Computer, Communication and Control (IMCCC)*. 2016, pp. 681–686.
- [51] Jenny Montbrun-Di Filippo, Marisol Delgado, Claude Brie, and Henry M. Paynter. "A survey of bond graphs : Theory, applications and programs". *Journal of the Franklin Institute* 328.5 (1991), pp. 565–606.
- [52] Yang Qian and Ahmed Rahmani. "Bond Graph Modeling and Simulation of a Dual-Arm Mobile Manipulator". *INACoMM 2013*. Roorkee, India, 2013.
- [53] M. M. Gor, P. M. Pathak, A. K. Samantaray, J. M. Yang, and S. W. Kwak. "Bond graph Modeling and Control of Compliant Legged Quadruped Robot". *Bond Graphs for Modelling, Control and Fault Diagnosis of Engineering Systems*. Springer, Cham, 2017, pp. 497–546.
- [54] Lorenzo Sciavicco and Bruno Siciliano. *Modelling and control of robot manipulators*. 2. ed., 6. print. Advanced textbooks in control and signal processing. OCLC: 255826188. London: Springer, 2005. 378 pp.
- [55] S. A. Wilmarth, N. M. Amato, and P. F. Stiller. "MAPRM: a probabilistic roadmap planner with sampling on the medial axis of the free space". *Proceedings 1999 IEEE Int. Conf. Robotics and Automation (Cat. No.99CH36288C)*. Vol. 2. 1999, 1024–1031 vol.2.
- [56] R. Geraerts and M. H. Overmars. "Clearance based path optimization for motion planning". *2004 IEEE Int. Conf. Robotics and Automation, 2004. Proceedings. ICRA '04*. Vol. 3. 2004, 2386–2392 Vol.3.

- [57] S. Lee, H. Moradi, and Chunsik Yi. "A real-time dual-arm collision avoidance algorithm for assembly". , 1997 *IEEE International Symposium on Assembly and Task Planning, 1997. ISATP 97*. 1997, pp. 7–12.
- [58] J. P. van den Berg and M. H. Overmars. "Roadmap-based motion planning in dynamic environments". *IEEE Trans. Robot.*. 21.5 (2005), pp. 885–897.
- [59] R. Raja and A. Dutta. "Path planning in dynamic environment for a rover using A\* and potential field method". 2017 *18th Int. Conf. Advanced Robotics (ICAR)*. 2017, pp. 578–582.
- [60] Athony Stentz. "Optimal and Efficient Path Planning for Unknown and Dynamic Environments". *International Journal of Robotics Automation* 10.3 (1993), pp. 89–100.
- [61] M. H. Korayem and S. R. Nekoo. "The SDRE control of mobile base cooperative manipulators: Collision free path planning and moving obstacle avoidance". *Robotics and Autonomous Systems* 86 (2016), pp. 86–105.
- [62] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces". *IEEE Trans. Robot. and Automation* 12.4 (1996), pp. 566–580.
- [63] Steven M. Lavalle. *Rapidly-Exploring Random Trees: A New Tool for Path Planning*. 1998.
- [64] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.
- [65] H. C. Huang, C. P. Chen, and P. R. Wang. "Particle swarm optimization for solving the inverse kinematics of 7-DOF robotic manipulators". 2012 *IEEE Int. Conf. Systems, Man, and Cybernetics (SMC)*. 2012, pp. 3105–3110.
- [66] M. Vijay and Debashisha Jena. "PSO based neuro fuzzy sliding mode control for a robot manipulator". *Journal of Electrical Systems and Information Technology* 4.1 (2017), pp. 243–256.
- [67] Mingming Wang, Jianjun Luo, and Ulrich Walter. "Trajectory planning of free-floating space robot using Particle Swarm Optimization (PSO)". *Acta Astronautica* 112 (2015), pp. 77–88.
- [68] Mohammad Reza Bonyadi and Zbigniew Michalewicz. "Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review". *Evolutionary Computation* 25.1 (2016), pp. 1–54.

- [69] Léonard Jaillet, Juan Cortés, and Thierry Siméon. “Sampling-based path planning on configuration-space costmaps”. *IEEE Transactions on Robotics* 26.4 (2010), pp. 635–646.
- [70] Y. C. Tsai and H. P. Huang. “Motion planning of a dual-arm mobile robot in the configuration-time space”. *2009 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. 2009, pp. 2458–2463.
- [71] J. P. Saut, M. Gharbi, J. Cortés, D. Sidobre, and T. Siméon. “Planning pick-and-place tasks with two-hand regrasping”. *2010 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*. 2010, pp. 4528–4533.
- [72] Y. J. Kim et al. “A RRT-based collision-free and occlusion-free path planning method for a 7DOF manipulator”. *2014 IEEE Int. Conf. Mechatronics and Automation*. 2014, pp. 1017–1021.
- [73] Carlos Rodríguez, Andrés Montaña, and Raúl Suárez. “Planning manipulation movements of a dual-arm system considering obstacle removing”. *Robotics and Autonomous Systems* 62.12 (2014), pp. 1816–1826.
- [74] J. Hollerbach and Ki Suh. “Redundancy resolution of manipulators through torque optimization”. *IEEE Journal on Robotics and Automation* 3.4 (1987), pp. 308–316.
- [75] F. Flacco and A. De Luca. “A reverse priority approach to multi-task control of redundant robots”. *2014 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. 2014, pp. 2421–2427.
- [76] C. L. Lewis. “Trajectory generation for two robots cooperating to perform a task”. , *1996 IEEE Int. Conf. Robotics and Automation, 1996. Proceedings*. Vol. 2. 1996, 1626–1631 vol.2.
- [77] A. Mohri, M. Yamamoto, and G. Hirano. “Cooperative path planning for two manipulators”. , *1996 IEEE Int. Conf. Robotics and Automation, 1996. Proceedings*. Vol. 3. 1996, 2853–2858 vol.3.
- [78] Dmitry Berenson, Siddhartha Srinivasa, and James Kuffner. “Task Space Regions: A framework for pose-constrained manipulation planning”. *The International Journal of Robotics Research* 30.12 (2011), pp. 1435–1460.
- [79] D. Berenson and S. S. Srinivasaz. “Probabilistically complete planning with end-effector pose constraints”. *2010 IEEE Int. Conf. Robotics and Automation*. 2010, pp. 2724–2730.
- [80] R. S. Jamisola, P. H. Chang, and J. Lee. “Guaranteeing task prioritization for redundant robots given maximum number of tasks and singularities”. *TENCON 2012 - 2012 IEEE Region 10 Conference*. 2012, pp. 1–6.

- [81] C. C. Tsai, C. C. Hung, and C. F. Chang. "Trajectory planning and control of a 7-DOF robotic manipulator". *2014 Int. Conf. Advanced Robotics and Intelligent Systems (ARIS)*. 2014, pp. 78–84.
- [82] Krzysztof Tchoń and Janusz Jakubiak. "Extended Jacobian motion planning algorithm for mobile manipulators". 2003.
- [83] M. H. Korayem and A. Nikoobin. "Maximum payload path planning for redundant manipulator using indirect solution of optimal control problem". *Int J Adv Manuf Technol* 44.7-8 (2009), p. 725.
- [84] Kensuke Harada et al. "Base position planning for dual-arm mobile manipulators performing a sequence of pick-and-place tasks". *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th Int. Conf. IEEE*, 2015, pp. 194–201.
- [85] Shunan Ren et al. "A Method for Optimizing the Base Position of Mobile Painting Manipulators". *IEEE Trans. Autom. Sci. Eng.* (2016), pp. 1–6.
- [86] Samer Yahya, M. Moghavvemi, and Haider A. F. Mohamed. "Geometrical approach of planar hyper-redundant manipulators: Inverse kinematics, path planning and workspace". *Simulation Modelling Practice and Theory. Modeling and Performance Analysis of Networking and Collaborative Systems* 19.1 (2011), pp. 406–422.
- [87] Lei Yan, Zonggao Mu, and Wenfu Xu. "Analytical inverse kinematics of a class of redundant manipulator based on dual arm-angle parameterization". *2014 IEEE Int. Conf. Systems, Man, and Cybernetics (SMC)*. IEEE, 2014, pp. 3744–3749.
- [88] Cecilia Lamperti, Andrea Maria Zanchettin, and Paolo Rocco. "A redundancy resolution method for an anthropomorphic dual-arm manipulator based on a musculoskeletal criterion". *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ Int. Conf. IEEE*, 2015, pp. 1846–1851.
- [89] Seungsu Kim, ChangHwan Kim, and Jong Hyeon Park. "Human-like arm motion generation for humanoid robots using motion capture database". *2006 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. IEEE, 2006, pp. 3486–3491.
- [90] J. K. Parker, A. R. Khoogar, and D. E. Goldberg. "Inverse kinematics of redundant robots using genetic algorithms". *1989 Int. Conf. Robotics and Automation Proceedings*. 1989, 271–276 vol.1.
- [91] Min Zhao, N. Ansari, and E. S. H. Hou. "Mobile Manipulator Path Planning By A Genetic algorithm". *Proceedings of the IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. Vol. 1. 1992, pp. 681–688.

- [92] E. J. Solteiro Pires, P. B. de Moura Oliveira, and J. A. Tenreiro Machado. "Multi-objective MaxiMin Sorting Scheme". *Int. Conf. Evolutionary Multi-Criterion Optimization*. Springer, Berlin, Heidelberg, 2005, pp. 165–175.
- [93] Maria da Graça Marcos, J. A. Tenreiro Machado, and T. P. Azevedo-Perdicoúlis. "A multi-objective approach for the motion planning of redundant manipulators". *Applied Soft Computing* 12.2 (2012), pp. 589–599.
- [94] Thi Thoa Mac, Cosmin Copot, Duc Trung Tran, and Robin De Keyser. "Heuristic approaches in robot path planning: A survey". *Robotics and Autonomous Systems* 86 (Supplement C 2016), pp. 13–28.
- [95] Azzeddine Bakdi et al. "Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control". *Robotics and Autonomous Systems* 89 (2017), pp. 95–109.
- [96] X. Jin, J. Kang, J. Zhang, and X. Yang. "Trajectory Planning of a Six-DOF Robot Based on a Hybrid Optimization Algorithm". *2016 9th International Symposium on Computational Intelligence and Design (ISCID)*. Vol. 2. 2016, pp. 148–151.
- [97] Hongxin Cao, Shouqian Sun, Kejun Zhang, and Zhichuan Tang. "Visualized trajectory planning of flexible redundant robotic arm using a novel hybrid algorithm". *Optik - International Journal for Light and Electron Optics* 127.20 (2016), pp. 9974–9983.
- [98] Zhen Yang Lim, S. G. Ponnambalam, and Kazuhiro Izui. "Multi-objective hybrid algorithms for layout optimization in multi-robot cellular manufacturing systems". *Knowledge-Based Systems* 120 (2017), pp. 87–98.
- [99] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. "A fast and elitist multiobjective genetic algorithm: NSGA-II". *IEEE Trans. Evol. Comput.* 6.2 (2002), pp. 182–197.
- [100] "Dynamic trajectory planning for mobile robot navigation in crowded environments". *2016 IEEE 21st Int. Conf. Emerging Technologies and Factory Automation (ETFA)*.
- [101] O. Adiyatov and H. A. Varol. "A novel RRT\*-based algorithm for motion planning in Dynamic environments". *2017 IEEE Int. Conf. Mechatronics and Automation (ICMA)*. 2017, pp. 1416–1421.
- [102] Li Meng, Song Qing, and Zhao Qin Jun. "UAV path re-planning based on improved bidirectional RRT algorithm in dynamic environment". *2017 3rd Int. Conf. Control, Automation and Robotics (ICCAR)*. 2017, pp. 658–661.

- [103] Faris Janjoš, Ron Reichart, and Philipp Niermeyer. “Smooth Path-Generation Around Obstacles Using Quartic Splines and RRTs”. *IFAC-PapersOnLine* 50.1 (2017), pp. 9108–9113.
- [104] Youngjun Choi, Hernando Jimenez, and Dimitri N. Mavris. “Two-layer obstacle collision avoidance with machine learning for more energy-efficient unmanned aircraft trajectories”. *Robotics and Autonomous Systems* 98 (Supplement C 2017), pp. 158–173.
- [105] Marcus Greiff and Anders Robertsson. “Optimisation-based motion planning with obstacles and priorities”. *IFAC-PapersOnLine* 50.1 (2017), pp. 11670–11676.
- [106] T. Mercy, R. Van Parys, and G. Pipeleers. “Spline-Based Motion Planning for Autonomous Guided Vehicles in a Dynamic Environment”. *IEEE Trans. Control Syst. Technol.* PP.99 (2017), pp. 1–8.
- [107] S. McLeod and J. Xiao. “Real-time adaptive non-holonomic motion planning in unforeseen dynamic environments”. *2016 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*. 2016, pp. 4692–4699.
- [108] Grzegorz Pajak and Iwona Pajak. “Motion planning for mobile surgery assistant”. *Acta Bioeng Biomech* 16.2 (2014), pp. 11–20. pmid: 25088187.
- [109] Isma Akli, Brahim Bouzouia, and Noura Achour. “Motion analysis of a mobile manipulator executing pick-up tasks”. *Computers & Electrical Engineering* 43 (2015), pp. 257–269.
- [110] Y. Yang, V. Ivan, and S. Vijayakumar. “Real-time motion adaptation using relative distance space representation”. *2015 Int. Conf. Advanced Robotics (ICAR)*. 2015, pp. 21–27.
- [111] G. Chen, Q. Jia, P. Ye, and H. Sun. “A novel autonomous obstacle avoidance path planning method for manipulator in joint space”. *2014 9th IEEE Conference on Industrial Electronics and Applications*. 2014, pp. 1877–1882.
- [112] Gao Xin et al. “Real-time dynamic system to path tracking and collision avoidance for redundant robotic arms”. *The Journal of China Universities of Posts and Telecommunications* 23.1 (2016), pp. 73–96.
- [113] Dong Han, Hong Nie, Jinbao Chen, and Meng Chen. “Dynamic obstacle avoidance for manipulators using distance calculation and discrete detection”. *Robotics and Computer-Integrated Manufacturing* 49 (Supplement C 2018), pp. 98–104.

- [114] P. D. H. Nguyen, M. Hoffmann, U. Pattacini, and G. Metta. “A fast heuristic Cartesian space motion planning algorithm for many-DoF robotic manipulators in dynamic environments”. *2016 IEEE-RAS 16th Int. Conf. Humanoid Robots (Humanoids)*. 2016, pp. 884–891.
- [115] J. Vannoy and J. Xiao. “Real-Time Adaptive Motion Planning (RAMP) of Mobile Manipulators in Dynamic Environments With Unforeseen Changes”. *IEEE Trans. Robot.*. 24.5 (2008), pp. 1199–1212.
- [116] P. Trautman and A. Krause. “Unfreezing the robot: Navigation in dense, interacting crowds”. *2010 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. 2010, pp. 797–803.
- [117] P. Trautman, J. Ma, R. M. Murray, and A. Krause. “Robot navigation in dense human crowds: the case for cooperation”. *2013 IEEE Int. Conf. Robotics and Automation*. 2013, pp. 2153–2160.
- [118] N. Malone, H. T. Chiang, K. Lesser, M. Oishi, and L. Tapia. “Hybrid Dynamic Moving Obstacle Avoidance Using a Stochastic Reachable Set-Based Potential Field”. *IEEE Trans. Robot.*. 33.5 (2017), pp. 1124–1138.
- [119] H. T. Chiang, N. Rackley, and L. Tapia. “Runtime SES planning: Online motion planning in environments with stochastic dynamics and uncertainty”. *2016 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*. 2016, pp. 4802–4809.
- [120] Shlomi Hacohen, Shraga Shoal, and Nir Shvalb. “Applying probability navigation function in dynamic uncertain environments”. *Robotics and Autonomous Systems* 87 (Supplement C 2017), pp. 237–246.
- [121] M. Levihn, J. Scholz, and M. Stilman. “Planning with movable obstacles in continuous environments with uncertain dynamics”. *2013 IEEE Int. Conf. Robotics and Automation*. 2013, pp. 3832–3838.
- [122] T. Yoshikawa. “Dynamic manipulability of robot manipulators”. *1985 IEEE Int. Conf. Robotics and Automation Proceedings*. Vol. 2. 1985, pp. 1033–1038.
- [123] A. Mohri, S. Furuno, M. Iwamura, and M. Yamamoto. “Sub-optimal trajectory planning of mobile manipulator”. *Proceedings 2001 ICRA. IEEE Int. Conf. Robotics and Automation (Cat. No.01CH37164)*. Vol. 2. 2001, 1271–1276 vol.2.
- [124] Qiang ZHAN, Yan-hui HE, and Ming CHEN. “Collision Avoidance of Co-operative Dual Redundant Manipulators”. *Chinese Journal of Aeronautics* 16.2 (2003), pp. 117–122.

- [125] M. H. Korayem, V. Azimirad, and M. Irani Rahagi. "Maximum Allowable Load of Mobile Manipulator in the Presence of Obstacle Using Non-Linear Open and Closed Loop Optimal Control". *Arab J Sci Eng* 39.5 (2014), pp. 4103–4117.
- [126] Changman Son. "Intelligent rule-based sequence planning algorithm with fuzzy optimization for robot manipulation tasks in partially dynamic environments". *Information Sciences* 342 (Supplement C 2016), pp. 209–221.
- [127] M. H. Korayem and S. R. Nekoo. "The SDRE control of mobile base cooperative manipulators: Collision free path planning and moving obstacle avoidance". *Robotics and Autonomous Systems* 86 (2016), pp. 86–105.
- [128] S. D. Wickramaratna et al. "Field based navigation for 3D obstacle avoidance". *2016 Moratuwa Engineering Research Conference (MERCon)*. 2016, pp. 391–396.
- [129] Mirosław Galicki. "Real-time constrained trajectory generation of mobile manipulators". *Robotics and Autonomous Systems* 78 (2016), pp. 49–62.
- [130] Amal Karray, Malek Njah, Moez Feki, and Mohamed Jallouli. "Intelligent mobile manipulator navigation using hybrid adaptive-fuzzy controller". *Computers & Electrical Engineering* 56 (2016), pp. 773–783.
- [131] Shogo Nishiguchi et al. "Theatrical approach: Designing human-like behaviour in humanoid robots". *Robotics and Autonomous Systems* 89 (2017), pp. 158–166.
- [132] Oliver Brock and Oussama Khatib. "Elastic Strips: A Framework for Motion Generation in Human Environments". *The International Journal of Robotics Research* 21.12 (2002), pp. 1031–1052.
- [133] C. L. Lewis and A. A. Maciejewski. "Trajectory generation for cooperating robots". *IEEE Int. Conf. Systems Engineering, 1990*. 1990, pp. 300–303.
- [134] D. J. F. Heck, A. Saccon, N. van de Wouw, and H. Nijmeijer. "Switched position-force tracking control of a manipulator interacting with a stiff environment". *2015 American Control Conference (ACC)*. 2015, pp. 4832–4837.
- [135] Sebastian Erhart, Dominik Sieber, and Sandra Hirche. "An impedance-based control architecture for multi-robot cooperative dual-arm mobile manipulation". *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ Int. Conf. IEEE, 2013*, pp. 315–322.

- [136] E. Simetti and G. Casalino. “Whole body control of a dual arm underwater vehicle manipulator system”. *Annual Reviews in Control* 40 (2015), pp. 191–200.
- [137] W. F. Kao, C. F. Hsu, and T. T. Lee. “Intelligent control for a dynamically stable two-wheel mobile manipulator”. *2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th Int. Conf. Soft Computing and Intelligent Systems (IFSA-SCIS)*. 2017, pp. 1–5.
- [138] D. H. Zhai and Y. Xia. “Adaptive Fuzzy Control of Multilateral Asymmetric Teleoperation for Coordinated Multiple Mobile Manipulators”. *IEEE Trans. Fuzzy Syst.* 24.1 (2016), pp. 57–70.
- [139] Mauricio Begnini, Douglas Wildgrube Bertol, and Nardênio Almeida Martins. “A robust adaptive fuzzy variable structure tracking control for the wheeled mobile robot: Simulation and experimental results”. *Control Engineering Practice* 64 (Supplement C 2017), pp. 27–43.
- [140] Shuzhi Sam Ge. “Robust adaptive NN feedback linearization control of nonlinear systems”. *Proceedings of the 1996 IEEE International Symposium on Intelligent Control*. 1996, pp. 486–491.
- [141] Samad A. Hayati. “Position and force control of coordinated multiple arms”. *IEEE transactions on aerospace and electronic systems* 24.5 (1988), pp. 584–590.
- [142] Yasuhisa Hirata, Takuro Sawada, Zhi-Dong Wang, and Kazuhiro Kosuge. “Leader-follower type motion control algorithm of multiple mobile robots with dual manipulators for handling a single object in coordination”. *Intelligent Mechatronics and Automation, 2004. Proceedings. 2004 Int. Conf. IEEE, 2004*, pp. 362–367.
- [143] Jong-Dho Choi, Sungchul Kang, Munsang Kim, C. W. Lee, and Jae-Bok Song. “Two-arm cooperative assembly using force-guided control with adaptive accommodation”. *1999 IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 1999. IROS '99. Proceedings. Vol. 2*. 1999, 1253–1258 vol.2.
- [144] P. Chiacchio, S. Chiaverini, and B. Siciliano. “Task-Oriented Kinematic Control of Two Cooperative 6-DOF Manipulators”. *American Control Conference, 1993*. 1993, pp. 336–340.
- [145] Jino Lee, Pyung Hun Chang, and Rodrigo S. Jamisola. “Relative Impedance Control for Dual-Arm Robots Performing Asymmetric Bimanual Tasks”. *IEEE Trans. Ind. Electron.* 61.7 (2014), pp. 3786–3796.
- [146] Rodrigo S. Jamisola and Rodney G. Roberts. “A more compact expression of relative Jacobian based on individual manipulator Jacobians”. *Robotics and Autonomous Systems* 63 (2015), pp. 158–164.

- [147] R. S. Jamisola, P. Kormushev, D. G. Caldwell, and F. Ibikunle. “Modular relative Jacobian for dual-arms and the wrench transformation matrix”. *2015 IEEE 7th Int. Conf. Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*. 2015, pp. 181–186.
- [148] Mustafa Mashali, Rdwan Alqasemi, and Rajiv Dubey. “Mobile manipulator dual-trajectory tracking using control variables introduced to end-effector task vector”. *World Automation Congress (WAC), 2016*. TSI Enterprise Inc (TSI Press), 2016, pp. 1–6.
- [149] Yoshihiko Nakamura, Hideo Hanafusa, and Tsuneo Yoshikawa. “Task-priority based redundancy control of robot manipulators”. *The International Journal of Robotics Research* 6.2 (1987), pp. 3–15.
- [150] P. Baerlocher and R. Boulic. “Task-priority formulations for the kinematic control of highly redundant articulated structures”. , *1998 IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 1998. Proceedings*. Vol. 1. 1998, 323–329 vol.1.
- [151] F. Basile, F. Caccavale, P. Chiacchio, J. Coppola, and C. Curatella. “Task-oriented motion planning for multi-arm robotic systems”. *Robotics and Computer-Integrated Manufacturing* 28.5 (2012), pp. 569–582.
- [152] J. Lee, P. H. Chang, and R. S. Jamisola. “Relative task prioritization for dual-arm with multiple, conflicting tasks: Derivation and experiments”. *2013 IEEE Int. Conf. Robotics and Automation (ICRA)*. 2013, pp. 1928–1933.
- [153] W. S. Owen, E. A. Croft, and B. Benhabib. “On-line trajectory resolution for two-armed systems with conflicting performance criteria”. *Mechanism and Machine Theory* 44.5 (2009), pp. 949–965.
- [154] Meiling Wang, Minzhou Luo, Tao Li, and Marco Ceccarelli. “A Unified Dynamic Control Method for a Redundant Dual Arm Robot”. *Journal of Bionic Engineering* 12.3 (2015), pp. 361–371.
- [155] W. S. Owen, E. A. Croft, and B. Benhabib. “Acceleration and torque redistribution for a dual-manipulator system”. *IEEE Trans. Robot.*. 21.6 (2005), pp. 1226–1230.
- [156] Vahid Nazari and Leila Notash. “Failure recovery of manipulators under joint velocity limits using constrained optimization and partitioned Jacobian matrix”. *Mechanism and Machine Theory* 99 (2016), pp. 58–71.
- [157] Fabrizio Flacco, Alessandro De Luca, and Oussama Khatib. “Control of Redundant Robots Under Hard Joint Constraints: Saturation in the Null Space”. *IEEE Trans. Robot.*. 31.3 (2015), pp. 637–654.

- [158] Tan Fung Chan and Rajiv V. Dubey. “A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators”. *IEEE Trans. Robot. and Automation* 11.2 (1995), pp. 286–292.
- [159] Jingguo Wang, Yangmin Li, and Xinhua Zhao. “Inverse kinematics and control of a dof redundant manipulator based on the closedloop algorithm”. *International Journal of Advanced Robotic Systems* 7.4 (2010), pp. 1–9.
- [160] Yunyi Jia, Ning Xi, Yu Cheng, and Siyang Liang. “Coordinated motion control of a nonholonomic mobile manipulator for accurate motion tracking”. *2014 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. IEEE, 2014, pp. 1635–1640.
- [161] G. Antonelli, G. Indiveri, and S. Chiaverini. “Prioritized closed-loop inverse kinematic algorithms for redundant robotic systems with velocity saturations”. *2009 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. 2009, pp. 5892–5897.
- [162] S. Huang, Y. Peng, W. Wei, and J. Xiang. “Clamping weighted least-norm method for the manipulator kinematic control: Avoiding joint limits”. *Proceedings of the 33rd Chinese Control Conference*. 2014, pp. 8309–8314.
- [163] Yang Hu, Bidan Huang, and Guang-Zhong Yang. “Task-priority redundancy resolution for co-operative control under task conflicts and joint constraints”. *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ Int. Conf. IEEE*, 2015, pp. 2398–2405.
- [164] D. Nicolis, A. M. Zanchettin, and P. Rocco. “Constraint-Based and Sensorless Force Control With an Application to a Lightweight Dual-Arm Robot”. *IEEE Robotics and Automation Letters* 1.1 (2016), pp. 340–347.
- [165] Robert G. Bonitz and Tien C. Hsia. “Internal force-based impedance control for cooperating manipulators”. *IEEE Trans. Robot. Autom.* 12.1 (1996), pp. 78–89.
- [166] Pushpraj Mani Pathak, Amalendu Mukherjee, and Anirvan Dasgupta. “Impedance Control of Space Robots Using Passive Degrees of Freedom in Controller Domain”. *J. Dyn. Sys., Meas., Control* 127.4 (2005), pp. 564–578.
- [167] M. Bjerckeng, J. Schrimpf, T. Myhre, and K. Y. Pettersen. “Fast dual-arm manipulation using variable admittance control: Implementation and experimental results”. *2014 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. 2014, pp. 4728–4734.

- [168] Flacco Fabrizio and Alessandro De Luca. “Real-Time Computation of Distance to Dynamic Obstacles With Multiple Depth Sensors”. *IEEE Robotics and Automation Letters* 2.1 (2017), pp. 56–63.
- [169] Tommaso Mattioli and Marilena Vendittelli. “Interaction Force Reconstruction for Humanoid Robots”. *IEEE Robotics and Automation Letters* 2.1 (2017), pp. 282–289.
- [170] Jung-Hua Yang, Feng-Li Lian, and Li-Chen Fu. “Adaptive hybrid position/force control for robotic manipulators with compliant links”. *Proceedings of 1995 IEEE Int. Conf. Robotics and Automation*. Vol. 1. 1995, 603–608 vol.1.
- [171] W. Gueaieb, F. Karray, and S. Al-Sharhan. “A Robust Hybrid Intelligent Position/Force Control Scheme for Cooperative Manipulators”. *IEEE/ASME Transactions on Mechatronics* 12.2 (2007), pp. 109–125.
- [172] Daniel E. Whitney. “Historical Perspective and State of the Art in Robot Force Control”. *The International Journal of Robotics Research* 6.1 (1987), pp. 3–14.
- [173] T. Yoshikawa. “Dynamic hybrid position/force control of robot manipulators—Description of hand constraints and calculation of joint driving force”. *IEEE Journal on Robotics and Automation* 3.5 (1987), pp. 386–392.
- [174] P. S. Londhe, Yogesh Singh, M. Santhakumar, B. M. Patre, and L. M. Waghmare. “Robust nonlinear PID-like fuzzy logic control of a planar parallel (2PRP-PPR) manipulator”. *ISA Transactions* 63 (Supplement C 2016), pp. 218–232.
- [175] S. Ge Shuzhi, C. C. Hang, and L. C. Woon. “Adaptive neural network control of robot manipulators in task space”. *IEEE Trans. Ind. Electron.* 44.6 (1997), pp. 746–752.
- [176] Dong Sun and J.K. Mills. “Adaptive synchronized control for coordination of multirobot assembly tasks”. *IEEE Trans. Robot.. and Automation* 18.4 (2002), pp. 498–510.
- [177] Mohamed Boukattaya, Mohamed Jallouli, and Tarak Damak. “On trajectory tracking control for nonholonomic mobile manipulators with dynamic uncertainties and external torque disturbances”. *Robotics and Autonomous Systems* 60.12 (2012), pp. 1640–1647.
- [178] Chenguang Yang, Yiming Jiang, Zhijun Li, Wei He, and Chun-Yi Su. “Neural Control of Bimanual Robots with Guaranteed Global Stability and Motion Precision”. *IEEE Trans Ind. Informat.* (2016), pp. 1–1.

- [179] Aleksandar Jevtić, Adrià Colomé, Guillem Alenyà, and Carme Torras. “Robot motion adaptation through user intervention and reinforcement learning”. *Pattern Recognition Letters* (2017).
- [180] S. S. Ge, Zhan-Lin Wang, and Zong-Ji Chen. “Adaptive static neural network control of robots”. , *Proceedings of the IEEE Int. Conf. Industrial Technology, 1994*. 1994, pp. 240–244.
- [181] S. S. GE and C. C. HANG. “Direct adaptive neural network control of robots”. *International Journal of Systems Science* 27.6 (1996), pp. 533–542.
- [182] Zhijun Li, Shuzhi Sam Ge, and Aiguo Ming. “Adaptive Robust Motion/Force Control of Holonomic-Constrained Nonholonomic Mobile Manipulators”. *IEEE Trans. Syst., Man, Cybern. B, Cybern.* 37.3 (2007), pp. 607–616.
- [183] Shuzhi S. Ge, C. C. Hang, and L. C. Woon. “Adaptive neural network control of robot manipulators in task space”. *IEEE Trans. Ind. Electron.* 44.6 (1997), pp. 746–752.
- [184] Qi Zhou, Chengwei Wu, and Peng Shi. “Observer-based adaptive fuzzy tracking control of nonlinear systems with time delay and input saturation”. *Fuzzy Sets and Systems*. Theme: Control Engineering 316 (2017), pp. 49–68.
- [185] Hocine Oubabas, Said Djennoune, and Maamar Bettayeb. “Interval sliding mode observer design for linear and nonlinear systems”. *Journal of Process Control* 61 (2018), pp. 12–22.
- [186] W. H. Chen, J. Yang, L. Guo, and S. Li. “Disturbance-Observer-Based Control and Related Methods #x2014;An Overview”. *IEEE Trans. Ind. Electron.* 63.2 (2016), pp. 1083–1095.
- [187] A. Wayse and A. Digrase. “Uncertainty and disturbance estimation using inertial delay observer with integral-sliding mode control”. *2017 Int. Conf. Intelligent Computing and Control Systems*. 2017, pp. 771–775.
- [188] L. Sun and Z. Zheng. “Disturbance-Observer-Based Robust Backstepping Attitude Stabilization of Spacecraft Under Input Saturation and Measurement Uncertainty”. *IEEE Trans. Ind. Electron.* 64.10 (2017), pp. 7994–8002.
- [189] Z. L. Zhao and B. Z. Guo. “A Novel Extended State Observer for Output Tracking of MIMO Systems With Mismatched Uncertainty”. *IEEE Trans. Autom. Control* 63.1 (2018), pp. 211–218.

- [190] Ling Zhao, Haiyan Cheng, and Tao Wang. “Sliding mode control for a two-joint coupling nonlinear system based on extended state observer”. *ISA Transactions* 73 (2018), pp. 130–140.
- [191] Hongjiu Yang, Yang Yu, Yuan Yuan, and Xiaozhao Fan. “Back-stepping control of two-link flexible manipulator based on an extended state observer”. *Advances in Space Research* 56.10 (2015), pp. 2312–2322.
- [192] S. A. Krasnova and A. V. Utkin. “Sigma function in observer design for states and perturbations”. *Autom Remote Control* 77.9 (2016), pp. 1676–1688.
- [193] M. Benzaoui, H. Chekireb, M. Tadjine, and A. Boulkroune. “Trajectory tracking with obstacle avoidance of redundant manipulator based on fuzzy inference systems”. *Neurocomputing* 196 (Supplement C 2016), pp. 23–30.
- [194] R. Raja and A. Dutta. “Motion planning of a mobile manipulator using fuzzy controller to dexterity measures”. *2015 IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions (WCI)*. 2015, pp. 1–6.
- [195] H. Hagnas. “Type-2 FLCs: A New Generation of Fuzzy Controllers”. *IEEE Computational Intelligence Magazine* 2.1 (2007), pp. 30–43.
- [196] J. M. Mendel, R. I. John, and F. Liu. “Interval Type-2 Fuzzy Logic Systems Made Simple”. *IEEE Trans. Fuzzy Syst.* 14.6 (2006), pp. 808–821.
- [197] Anupam Kumar and Vijay Kumar. “Evolving an interval type-2 fuzzy PID controller for the redundant robotic manipulator”. *Expert Systems with Applications* 73 (Supplement C 2017), pp. 161–177.
- [198] Mauricio A. Sanchez, Oscar Castillo, and Juan R. Castro. “Generalized Type-2 Fuzzy Systems for controlling a mobile robot and a performance comparison with Interval Type-2 and Type-1 Fuzzy Systems”. *Expert Systems with Applications* 42.14 (2015), pp. 5904–5914.
- [199] H. A. Hagnas. “A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots”. *IEEE Trans. Fuzzy Syst.* 12.4 (2004), pp. 524–539.
- [200] B. Xian, M.S. deQueiroz, D. Dawson, and I. Walker. “Task-Space Tracking Control of Robot Manipulators via Quaternion Feedback”. *IEEE Trans. Robot.. and Automation* 20.1 (2004), pp. 160–167.

- [201] Rafael Fierro and Frank L. Lewis. “Control of a nonholonomic mobile robot: backstepping kinematics into dynamics”. *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*. Vol. 4. IEEE, 1995, pp. 3805–3810.
- [202] Mohammad M. Emamzadeh, Nasser Sadati, and William A. Gruver. “Fuzzy-based interaction prediction approach for hierarchical control of large-scale systems”. *Fuzzy Sets and Systems*. Theme: Control 329 (Supplement C 2017), pp. 127–152.
- [203] N. Srinivas and K. Deb. “Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms”. *Evolutionary Computation* 2.3 (1994), pp. 221–248.
- [204] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Google-Books-ID: cyV7nQEACAAJ. MIT Press, 1992. 211 pp.
- [205] Khiar Nait-Chabane. “Exploitation de la redondance pour la commande coordonnée d’un manipulateur mobile d’assistance aux personnes handicapées.” Université d’Evry-Val d’Essonne, 2006.
- [206] Steven M. Lavalle. “Rapidly-Exploring Random Trees: A New Tool for Path Planning” (1999).
- [207] F. Furlán, E. Rubio, H. Sossa, and V. Ponce. “Humanoid robot hierarchical navigation using Petri nets and fuzzy logic”. *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*. 2017, pp. 1521–1526.
- [208] Roberto Pinillos, Samuel Marcos, Raul Feliz, Eduardo Zalama, and Jaime Gómez-García-Bermejo. “Long-term assessment of a service robot in a hotel environment”. *Robotics and Autonomous Systems* 79 (Supplement C 2016), pp. 40–57.



# Résumé Etendu

Cette thèse est une continuation de [1]. [1] a fini les conceptions d'applications et de fonctionnalités du robot d'assistance à la personne et a complété la modélisation dynamique d'un manipulateur mobile à deux bras (MDAMS) en utilisant la méthode lagrangienne. Cependant, à mesure que les degrés de liberté du robot augmentent, les calculs de l'énergie et de ses dérivées partielles seront complexe et prendra du temps. En outre, l'auteur a proposé un algorithme de résolution de cinématique inverse basé sur RRT-connect pour concevoir le mouvement de l'espace en supposant que le mouvement du robot dans l'espace de tâche est connu, sans traitant des problèmes comme l'optimisation du chemin, la redondance et la singularité. En fait, il n'y a pas seulement des obstacles statiques tels que des murs et des meubles, mais aussi des obstacles dynamiques tels que des robots et des humains, et des obstacles mobiles tels que des portes et des chaises dans des environnements domestiques standard. Par conséquent, il est très difficile de concevoir le mouvement du robot dans l'espace de tâche pour l'humanoïde étudié dans cette thèse. Pire encore, en raison de la non-linéarité élevée entre les espaces de tâche et des joints, même si le mouvement dans l'espace de tâche est connu, il est encore difficile de déterminer le mouvement dans l'espace des joints sur lequel la commande robotique dépend directement ou indirectement. En outre, [1] a également conçu un contrôleur adaptatif des réseaux neuronaux RBF (RBFNN) pour résoudre les problèmes d'incertitude et d'interférence, et a réalisé la manipulation des objets en contrôlant la force d'interaction. Cependant, le contrôle collaboratif entre les bras et le couplage entre la plate-forme mobile et le manipulateur supérieur n'ont pas été étudiés. De plus, l'efficacité du contrôleur RBFNN reste à vérifier.

En résumé, il existe de nombreux problèmes à être résolus pour réaliser un robot d'assistance à la personne intelligente pour des applications pratiques. Cette thèse est organisée en cinq chapitres. L'état de l'art des manipulateurs robotiques et des robots de service, en particulier des robotiques humanoïdes, et une brève introduction au contexte technique sont donnés au chapitre 1.

Au chapitre 2, le modèle cinématique du robot conçu est d'abord établi. La méthode modifiée de Denavit Hartenberg est utilisée à construire le modèle

cinématique avec seulement quatre paramètres. Les expressions compactes des mouvements de la plate-forme mobile et des outils terminaux (OTs) sont formulées en utilisant l'unité quaternion à représenter les orientations des OTs. Ensuite, le modèle lagrangien et le contrôleur adaptatif RBFNN sont rappelés. La validation virtuelle via la co-simulation MATLAB/SIMULINK et Adams est réalisée pour valider l'efficacité du contrôleur RBFNN. Afin d'éviter le calcul d'énergie et de dérivées partielles, les équations dynamiques sont formulées en utilisant la méthode de Kane. La stabilité du renversement physique est étudiée en utilisant le modèle de Kane construit. Finalement, un contrôleur est proposé et la stabilité du contrôleur est analysée à l'aide de la technique de back-stepping.

Chapitre 3 s'intéresse à la planification de mouvement pour le robot conçu dans des environnements statiques. Tout d'abord, un algorithme génétique de tri non dominé (NSGA-II) basé sur MaxiMin est proposé pour concevoir simultanément la position-orientation optimale de la plate-forme mobile et la configuration optimale du manipulateur supérieur (pose optimale). Il accélère l'algorithme évolutif MaxiMin NSGA-II en ajoutant un processus de tri MaxiMin. Cinq fonctions objectives, i.e. deux critères de précision d'atteinte des OTs, la capacité de manipulabilité, le déplacement des articulations et les déplacements des OTs dans l'espace de tâche et des joints, sont définies et optimisées en même temps. En particulier, une fonction de fitness combinée normalisée est définie et un décideur a posteriori est introduit pour régler la solution préférée parmi les candidats dans l'ensemble Pareto-optimal. Ensuite, afin de réaliser la transition sans collision de la pose initiale à la pose optimale, un algorithme hors ligne de planification de mouvement à connexion directe est conçu en combinant BiRRT et la gradient-descente, ce qui accélère considérablement la phase d'échantillonnage. Et une méthode d'optimisation géométrique est proposée qui garantit toujours le chemin le plus court et le plus cohérent entre les poses initiale et optimale. En outre, les polynômes linéaires avec des polynômes et les polynômes de cinquième ordre sont utilisés pour générer la trajectoire. De plus, les mouvements en avant sont réalisés en assignant des orientations raisonnables à la plate-forme mobile en fonction de ses positions planifiées, indiquant ainsi l'intention du robot pour augmenter la qualité de l'interaction homme-robot. Enfin, plusieurs tâches de planification de mouvement sont réalisées pour valider les algorithmes proposés.

Pour résoudre le problème d'échec de l'algorithme hors ligne, chapitre 4 présente un algorithme de planification de mouvement en ligne dans des environnements dynamiques en étendant l'algorithme hors ligne proposé au chapitre 3. Tout d'abord, le mouvement souhaité est conçu hors ligne. Ensuite, les processus de détection environnementale en ligne, de test de collision, et de planification et de suivi de mouvement en ligne sont activés. L'information

environnementale est détectée et les futurs mouvements d'obstacles inconnus sont prédits en temps réel en fonction de leurs positionnements historiques. Si une collision possible est prévue, la planification en ligne sera activé et le mouvement souhaité sera mis à jour immédiatement. Afin de prendre en compte les via-poses, un algorithme de planification de mouvement basée sur les via-points des OTs et d'algorithme génétique multi-objectif (MOGA) est introduit. Quatre fonctions objectives basées sur des via-poses sont définies pour optimiser les via-poses, i.e. (i) le déplacement des joints, (ii) la manipulabilité directionnelle contournant les via-points, (iii) la précision de positionnement des via-points, et (iv) l'évaluation de collision robot-obstacle. En d'autres termes, premièrement, les via-points des OTs sont conçus; deuxièmement, l'algorithme basé sur les via-points des OTs et MOGA est utilisé pour optimiser les via-poses correspondantes. Ensuite, la méthode de planification de mouvement est utilisée pour lier les via-poses optimales. Finalement, deux environnements domestiques virtuels sont construits et des simulations de planification de mouvement sont réalisées pour valider l'efficacité des algorithmes proposés.

Chapitre 5 étudie le suivi de trajectoire du robot sous l'hypothèse que la trajectoire d'OT dans l'espace de tâche est connue. Dans un premier temps, la méthode de résolution de cinématique inverse basée sur la priorité de tâche, la solution de moindre norme est introduite brièvement, et la méthode de WLN modulé est conçue pour éviter la limite de joint. Deuxièmement, au lieu de suivre directement le mouvement absolu des OTs, le mouvement relatif entre deux OTs est introduit afin de suivre la trajectoire d'OT en utilisant l'art de leader-suiveur. En particulier, la cinématique du mouvement relatif est obtenue en utilisant les cinématiques des OTs, et l'erreur d'attitude de l'OT est résolue en utilisant la matrice d'erreur de suivi représentée par l'unité quaternion. De plus, une méthode de suivi semi-décentralisée à double trajectoire est proposée basée sur le contrôle cinématique de la plate-forme mobile non-holonome. Troisièmement, afin de réaliser le contrôle du robot pour saisir l'objet, la matrice de contact entre l'OT et l'objet est définie, et la dynamique de l'objet est introduite dans le modèle dynamique du robot. De plus, inspirés par le comportement humain, un nouveau mouvement relatif a été introduit, i.e. le mouvement de l'OT par rapport à la plate-forme mobile du robot, afin de réaliser le contrôle coordonné entre la plate-forme et le manipulateur supérieur. Jusqu'à présent, deux mouvements relatifs ont été définis, et puis la méthode de suivi semi-décentralisée à triple trajectoire basée sur le contrôle cinématique de la plate-forme mobile est proposée. Enfin, afin de vérifier l'efficacité des méthodes proposées, des simulations de suivi des mouvements sont présentées.

Enfin, les conclusions, le travail à l'étude et les perspectives sont listés à la fin de la thèse.

## Planification et Suivi de Mouvement d'un Manipulateur Mobile Non-holonome à deux Bras

**Résumé:** Cette thèse se concentre sur la planification et le suivi de mouvement d'un manipulateur mobile non-holonome à deux bras. D'abord, la méthode MDH est utilisée pour la modélisation cinématique et la co-simulation sur le prototype est réalisée pour valider l'efficacité du contrôleur RBFNN. Afin d'éviter de calculer l'énergie et ses dérivées partielles, la méthode de Kane est utilisée. Puis, un algorithme avancé MaxiMin NSGA-II est proposé à concevoir la pose optimale étant données la pose initiale et les positions-orientations souhaitées des outils terminaux (OTs). Un algorithme à connexion directe combinant BiRRT et la gradient-descente est conçu pour la transition de la pose initiale à la pose optimale, et une optimisation géométrique est conçue pour optimiser et cohérer le chemin. Afin de résoudre le problème d'échec de l'algorithme hors ligne, un algorithme en ligne est proposé. De plus, un algorithme basé sur les via-points des OTs et MOGA est proposé à optimiser les via-poses. Enfin, le problème de suivi de mouvement est étudié étant données les mouvements des OTs. Au lieu de contrôler le mouvement absolu, deux mouvements relatifs sont introduits pour réaliser la coordination et la coopération entre la plate-forme et le manipulateur supérieur en évitant les limites des joints.

**Mots-clés:** Manipulateur mobile, Méthode de Kane, MaxiMin NSGA-II, résolution de redondance, Planification de mouvement, suivi de trajectoire.

## Motion Planning and Tracking of a Non-holonomic Mobile Dual-arm Manipulator

**Abstract:** This thesis focuses on the motion planning and tracking of a dual-arm mobile manipulator. First, MDH is used for kinematic modeling and the co-simulation on prototype is realized to validate RBFNN controller's effectiveness. In order to avoid calculating energy and its derivatives, Kane's method is used for dynamic modeling with physical stability's analysis/ Secondly, an improved MaxiMin NSGA-II is proposed to design the optimal pose. A direct connect algorithm combining BiRRT and gradient-descent is designed to transform from the initial pose to optimal pose with a geometric path optimization method being proposed. In order to solve the failure problem of off-line algorithm, an on-line algorithm is proposed. In addition, an algorithm based on EEs' via-points and MOGA is proposed to optimize via-poses. Finally, the motion tracking problem is studied given EEs' motions. Instead of controlling the absolute motion, two relative motions are introduced to realize the coordination and cooperation between the mobile base and upper manipulator with joint limit avoidance.

**Keywords:** Mobile manipulator, Kane's Method, MaxiMin NSGA-II, redundancy solving, motion planning, trajectory tracking.