

11-16-2015

Kinematic Control of Redundant Mobile Manipulators

Mustafa Mashali

University of South Florida, mmashali@mail.usf.edu

Follow this and additional works at: <http://scholarcommons.usf.edu/etd>

 Part of the [Mechanical Engineering Commons](#), and the [Robotics Commons](#)

Scholar Commons Citation

Mashali, Mustafa, "Kinematic Control of Redundant Mobile Manipulators" (2015). *Graduate Theses and Dissertations*.
<http://scholarcommons.usf.edu/etd/5989>

This Dissertation is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact scholarcommons@usf.edu.

Kinematic Control of Redundant Mobile Manipulators

by

Mustafa Mashali

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Mechanical Engineering
College of Engineering
University of South Florida

Major Professor: Rajiv Dubey, Ph.D.
Redwan Alqasemi, Ph.D.
Sudeep Sarkar, Ph.D.
Yu Sun, Ph.D.
Kandethody Ramachandran, Ph.D.

Date of Approval:
November 12, 2015

Keywords: Dual-Trajectory Control, Trajectory Tracking, Optimization, Manipulability
Measure, Pose Estimation

Copyright © 2015, Mustafa Mashali

DEDICATION

In the name of ALLAH, Most Gracious, Most Merciful,

To my God who guided me to the right path, gave me knowledge and patience to accomplish this work and made me successful throughout this journey. So, All praise and thanks to ALLAH.

To my beloved father and mother, who sacrificed much to complete my education. I will never forget their prayers for my success. May ALLAH give them all they wish for and bless them with good health.

To my wife and children who supported and encouraged me in all of the difficult times that I faced during my study.

To my advisors, Dr. Rajiv Dubey, and Dr. Redwan Alqasemi for all of their great support and guidance. Throughout this journey, they were patient with me and always there when I needed them. Thank you all so much.

ACKNOWLEDGMENTS

First, I would like to thank my advisor Dr. Rajiv Dubey for the invaluable opportunity to work with him as a PhD student. His endless help and guidance made this work possible. I would also like to thank the member of my committee, Dr. Redwan Alqasemi for his priceless advices and support. Many thanks for the rest of my committee members Dr. Sudeep Sarkar, Yu Sun, and Kandethody Ramachandran for their important comments to this work. Special thanks to my family members and relatives for their endless support and encouragement. Thanks also to lab colleagues for their help and support throughout the years.

TABLE OF CONTENTS

LIST OF TABLES	v
LIST OF FIGURES	vi
ABSTRACT	xii
CHAPTER 1 INTRODUCTION	1
1.1 Introduction	1
1.2 Motivation	6
1.3 Problem Statement	8
1.4 Dissertation Objectives	11
1.5 Dissertation Outline	12
CHAPTER 2 BACKGROUND AND LITERATURE REVIEW	14
2.1 Introduction	14
2.2 Mobile Manipulator Classification	15
2.2.1 Mobile Manipulators Environments	15
2.2.1.1 Aerial Manipulators	16
2.2.1.2 Underwater Manipulation	16
2.2.1.3 Grounded Mobile Manipulators	17
2.2.2 Mobile Manipulator Applications	17
2.3 Mobile Manipulator Control	18
2.3.1 Coordinated Control of Mobile Manipulators	20
2.3.2 Mobile Manipulation Control	22
2.3.3 Redundant Mobile Manipulator Control	23
2.3.4 Mobile Manipulator Trajectory Tracking Control	26
2.4 Mobile Platform Pose Error Compensation	30
CHAPTER 3 MOBILE MANIPULATOR KINEMATICS MODEL	32
3.1 Introduction	32
3.2 Terminology	32
3.3 Kinematic Modeling	32
3.4 Dual-Trajectory Kinematic Representation	35
3.5 Mobile Manipulator Jacobi	37
3.5.1 Manipulator Jacobian	37
3.5.2 Mobile Platform Jacobian	39

3.6	Mobile Manipulator Combined Jacobian	47
3.7	Summary	49
CHAPTER 4 MOBILE MANIPULATOR KINEMATIC CONTROL		50
4.1	Introduction	50
4.2	Kinematic Control	50
4.2.1	Pseudo-Inverse of the Jacobian	51
4.2.2	Singularity-Robust Inverse (SR-Inverse)	52
4.2.3	Weighted Least-Norm Solution	53
4.2.3.1	General Case	54
4.2.3.2	Special Case: Joint Limit Avoidance (JLA)	56
4.2.4	Inverse Kinematics Considering the Order of Priority	57
4.2.5	Optimization Criteria	58
CHAPTER 5 NOVEL DUAL-TRAJECTORY TRACKING USING SPHERICAL CONTROL VARIABLES (D , α AND β)		62
5.1	Introduction	62
5.2	Terminology	63
5.3	Dual-Trajectory Combined Control	63
5.3.1	Spherical Control Variables (D , α and β)	64
5.3.1.1	Platform Pose Relative to End-Effector Position	66
5.3.1.2	End-Effector Position Relative to Platform Pose	67
5.3.2	Jacobian of the Spherical Control Variables	68
5.4	Dual-Trajectory Control With Order of Priority	70
5.5	Implementation Example	71
5.6	Simulation Results and Discussion	76
5.6.1	End-Effector Trajectory as First Priority	76
5.6.2	Mobile Platform Trajectory as First Priority	78
5.7	Summary	79
CHAPTER 6 DUAL-TRAJECTORY TRACKING WITH FREE PLATFORM TRANSLATION ALONG A TRACK		81
6.1	Introduction	81
6.2	Unconstrained Mobile Platform Translations Along a Track	82
6.2.1	Unconstrained Mobile Platform Translations	85
6.2.2	Maximization of the System's Manipulability Measure	86
6.3	Implementation Example	87
6.3.1	PMM Jacobian	88
6.3.2	Track Radius of Curvature (ρ)	89
6.3.3	Manipulability Measure Maximization of the PMM System	89
6.4	Simulation Results and Discussion	90
6.4.1	Predefined-Translation Case	91
6.4.2	LN Undefined-Translation Case	92
6.4.3	MM Undefined-Translation Case	93

6.4.4	Simulation Results and Discussion	93
6.5	Summary	97
CHAPTER 7	DEVELOPMENT, TESTING AND RESULTS OF POSE ESTI- MATION CORRECTION OF MOBILE PLATFORM WITH HIGH UNCERTAINTIES	99
7.1	Introduction	99
7.2	Mobile Platform Pose Estimation Methods	100
7.2.1	Encoder-Based Odometry	102
7.2.2	ICP-Based Optimized Odometry	102
7.2.2.1	ICP-Based Local Transformation Matrix	102
7.2.2.2	ICP-Based Optimized Odometry	106
7.3	Mobile Platform Motion Control Schemes	108
7.3.1	Encoder Only Motion Control	109
7.3.2	ICP-Based Updated Odometry Motion Control	109
7.4	Implementation Example	110
7.5	Experiments	111
7.5.1	Offline Data Processing	112
7.5.2	Offline Data Processing Results and Discussion	112
7.5.3	Online Implementation Results	118
7.5.4	Real-Time Mobile Platform Application	121
7.5.4.1	Real-Time Application Results and Discussion	122
CHAPTER 8	SIMULATION IMPLEMENTATION AND RESULTS FOR THE WMRA SYSTEM	125
8.1	Introduction	125
8.2	Kinematic Model for WMRA System	125
8.2.1	Wheelchair Kinematic Model	126
8.2.2	Robotic Arm Kinematic Model	129
8.2.3	Combined Wheelchair Mobility and Robotic Arm Manipula- tion	130
8.3	Implementation Using MATLAB Simulation	131
8.3.1	System Modeling Using D , α and β	131
8.3.1.1	Spherical Control Variables (D , α and β)	132
8.3.1.2	Spherical Control Variables Jacobian	133
8.3.1.3	Simulation Results and Discussion	133
8.3.2	System Modeling Using Optimized Translation Along a Track	138
8.3.2.1	Optimized Wheelchair Linear Velocity \dot{S}	140
8.3.2.2	Wheelchair Angular Velocity $\dot{\phi}$	141
8.3.2.3	Tested Cases	141
8.3.2.4	Simulation Results and Discussion	143
8.4	Summary	147

CHAPTER 9	HARDWARE IMPLEMENTATION AND RESULTS FOR THE WMRA SYSTEM	150
9.1	Introduction	150
9.2	Hardware Design of the WMRA-II System	151
9.3	Controller Hardware	153
	9.3.1 Robotic Arm Controller Hardware	153
	9.3.2 Wheelchair Controller Hardware	153
9.4	Control Algorithm Implementation	155
	9.4.1 Dual-Trajectory Tracking Using D , α and β	155
	9.4.2 Dual-Trajectory Tracking With Free Platform Translation Along a Track	156
9.5	Hardware Implementation	157
9.6	Hardware Results	159
	9.6.1 Dual-Trajectory Tracking Using D , α and β	159
	9.6.2 Dual-Trajectory Tracking With Free Platform Translation Along a Track	165
CHAPTER 10	CONCLUSIONS AND RECOMMENDATIONS	174
10.1	Overview	174
10.2	General Discussion	174
10.3	Recommendations	177
REFERENCES		179
APPENDICES		187
Appendix A	Copyright Permission	188
Appendix B	Planar Mobile Manipulator Jacobian Derivation	212
Appendix C	Hardware Implementation C++ Code	219

LIST OF TABLES

Table 8.1	The D-H parameters of the robotic arm.	131
-----------	--	-----

LIST OF FIGURES

Figure 1.1	Schematic diagram of a, “picking up an object while moving and avoiding an obstacle,” task.	2
Figure 1.2	Schematic diagram of an, “opening and going through a spring loaded door,” task.	3
Figure 1.3	Schematic diagram of a “assembly of large-scale parts by welding,” task.	3
Figure 1.4	Amazon fulfillment center [1].	4
Figure 1.5	A mobile manipulator with predefined separate trajectories for the end-effector and the mobile platform.	5
Figure 1.6	Follow end-effector trajectory by changing the mobile platform position along its track.	8
Figure 1.7	A mobile manipulator dual-trajectory (E_T and P_T) with the starting, desired, current and final end-effector and mobile platform poses ($\{E_o\}$, $\{E_d\}$, $\{E_c\}$, $\{E_F\}$, $\{P_o\}$, $\{P_d\}$, $\{P_c\}$, and $\{P_F\}$).	10
Figure 1.8	A mobile manipulator with separate trajectories for the end-effector and the mobile platform.	11
Figure 2.1	First experimental platform for aerial manipulation [2].	16
Figure 2.2	TRIDENT underwater intervention project [3].	17
Figure 2.3	“Little Helper” industrial mobile manipulator [4].	18
Figure 2.4	Main application domains of mobile manipulation [5].	18
Figure 2.5	Coordinated control for mobile manipulator composed of two sub-systems.	21
Figure 2.6	Wheeled mobile manipulator with two manipulators.	22
Figure 2.7	Block diagram of the motion control system for mobile manipulators.	23

Figure 2.8	Results of the traditional kinematic control and the kinematic control with motion distribution.	25
Figure 2.9	Wheeled mobile manipulator with desired end-effector and mobile base trajectories.	25
Figure 2.10	Desired end-effector and mobile platform trajectories and mobile manipulator animation.	28
Figure 3.1	Differentially driven mobile manipulator consisting of a differentially driven mobile platform and an n DoF robotic arm.	33
Figure 3.2	Two consecutive mobile platform frames.	40
Figure 3.3	2D map for the mobile manipulator's main coordinate frames.	42
Figure 4.1	Manipulability ellipsoid with principal axes.	60
Figure 5.1	A mobile manipulator predefined dual-trajectory with spherical control variables D and α .	64
Figure 5.2	Mobile manipulator end-effector and mobile platform frames with spherical control variables D , α and β .	65
Figure 5.3	Control variables D , and α with global distance.	69
Figure 5.4	Planar mobile manipulator with 3 DoF planar robotic arm and non-holonomic mobile platform.	72
Figure 5.5	5 DoF Planar Mobile Manipulator parameters (D_{max} , α , L_{max} and l_A).	73
Figure 5.6	Tracked dual-trajectory of PMM for the three instance of Equation (5.10) $\dot{q} = J_1^\# \dot{r}_1 + (I - J_1^\# J_1) \hat{J}_2^\# (\dot{r}_2 - J_2 J_1^\# \dot{r}_1) + (I - J_1^\# J_1) (I - \hat{J}_2^\# \hat{J}_2) H$: a) First term.	77
Figure 5.7	PMM manipulability measure for the three instances of Equation (5.10).	78
Figure 5.8	Tracked dual-trajectory of the PMM for all terms of Equation (5.10).	79
Figure 5.9	PMM manipulability measure for all terms in Equation (5.10).	79
Figure 6.1	Radius of curvature ρ of the track at point P.	83
Figure 6.2	Mobile manipulator predefined end-effector trajectory with the predefined mobile platform's track.	85

Figure 6.3	Planar mobile manipulator with a 3 DoF planar robotic arm and a nonholonomic mobile platform.	88
Figure 6.4	Tracked trajectories of the PMM for cases: (a) Predefined-translation.	94
Figure 6.5	Commanded and actual end-effector trajectories for the predefined and undefined translation cases.	95
Figure 6.6	PMM platform velocity in each time step for the predefined, MM and LN undefined translations cases.	96
Figure 6.7	PMM manipulability measure for the predefined, MM and LN undefined translation cases.	97
Figure 6.8	Arm manipulability measure for the predefined, MM and LN undefined translation cases	98
Figure 7.1	Flowchart of two estimation methods.	101
Figure 7.2	Platform and Kinect coordinates frame relative to the global coordinate frames.	103
Figure 7.3	Registration process and Kalman Filter with Kinect RGB images.	104
Figure 7.4	Flowchart of using the Extended Kalman Filter in fusing the two local transformation matrices.	107
Figure 7.5	Schematic diagram of the mobile platform encoder-based odometry motion control.	109
Figure 7.6	Schematic diagram of the mobile platform ICP-based updated odometry motion control.	110
Figure 7.7	Wheelchair Mounted Robotic Arm (WMRA-I).	111
Figure 7.8	Wheelchair encoder-based odometry (blue-dashed line) with ground truth (red-dashed line).	113
Figure 7.9	Wheelchair ICP only odometry (black-dashed line) with ground truth (red-dashed line).	114
Figure 7.10	Wheelchair ICP based-optimized odometry without updating (green-dashed line) with ground truth (red-dashed line).	115
Figure 7.11	Wheelchair ICP based-optimized odometry (blue-dashed line) with ground truth (red-dashed line).	116

Figure 7.12	Global position error for encoder-based (blue line), ICP based (green line) and ICP-based updated (red line) odometry for online implementation.	117
Figure 7.13	Global orientation angle error for encoder-based (blue line), ICP-based (green line) and ICP-based updated (red line) odometry for online implementation.	118
Figure 7.14	ICP-based updated odometry (blue-dashed line) for online implementation with wheelchair ground truth (red-dashed line).	119
Figure 7.15	Global position error for encoder-based (blue line) and ICP-based updated (black line) odometry for online implementation.	120
Figure 7.16	Global orientation angle error for encoder-based (blue line) and ICP-based updated (black line) odometry for online implementation.	120
Figure 7.17	Initial and final wheelchair poses and path planning.	122
Figure 7.18	Wheelchair pose estimation using encoder-based odometry control (blue line) with the ground truth (red-dashed line for the wheelchair and red-solid line for the obstacle), Kinect-mapped obstacle (black line) and commanded trajectory (green line) for both control schemes.	123
Figure 7.19	Wheelchair pose estimation ICP-based updated odometry control (blue line) with the ground truth (red-dashed line for the wheelchair and red-solid line for the obstacle), Kinect-mapped obstacle (black line) and commanded trajectory (green line) for both control schemes.	124
Figure 8.1	The two WMRA system prototypes.	126
Figure 8.2	WMRA system coordinate frames [6].	127
Figure 8.3	MATLAB simulated WMRA system.	131
Figure 8.4	Tracked dual-trajectory of the WMRA system for the first term of Equation (5.10) $\dot{q} = J_1^\# \dot{r}_1$.	135
Figure 8.5	Tracked dual-trajectory of the WMRA system for the first and second terms of Equation (5.10) $\dot{q} = J_1^\# \dot{r}_1 + (I - J_1^\# J_1) \hat{J}_2^\# (\dot{r}_2 - J_2 J_1^\# \dot{r}_1)$.	135
Figure 8.6	Tracked dual-trajectory of the WMRA system for all terms of Equation (5.10) $\dot{q} = J_1^\# \dot{r}_1 + (I - J_1^\# J_1) \hat{J}_2^\# (\dot{r}_2 - J_2 J_1^\# \dot{r}_1) + (I - J_1^\# J_1)(I - \hat{J}_2^\# \hat{J}_2)H$.	136
Figure 8.7	Whole WMRA system manipulability measure for the three instance of Equation (5.10) $\dot{q} = J_1^\# \dot{r}_1 + (I - J_1^\# J_1) \hat{J}_2^\# (\dot{r}_2 - J_2 J_1^\# \dot{r}_1) + (I - J_1^\# J_1)(I - \hat{J}_2^\# \hat{J}_2)H$: First term (green line).	137

Figure 8.8	Tracked dual-trajectory of the WMRA system for all terms of Equation (5.10).	138
Figure 8.9	Whole WMRA system manipulability measure for all terms in Equation (5.10).	139
Figure 8.10	The WMRA system locations sequence for both the end-effector and the wheelchair on their trajectories in the case of predefined-translations.	144
Figure 8.11	End-effector trajectory tracking error in xy and xz planes.	144
Figure 8.12	The sequence of locations for both the end-effector and the wheelchair on their trajectories in the case of the LN solution.	145
Figure 8.13	The sequence of locations for both the end-effector and the wheelchair on their trajectories, in the case of maximizing the manipulability measure.	146
Figure 8.14	The wheelchair linear velocities along its trajectory for the three cases.	147
Figure 8.15	The integration of wheelchair translations along its trajectory for three cases.	148
Figure 8.16	The manipulability measure of whole WMRA system for the three cases.	149
Figure 9.1	Two WMRA system prototypes.	151
Figure 9.2	Control board, harmonic drive and joint motor.	152
Figure 9.3	WMRA-II system with the robotic arm control box.	154
Figure 9.4	Wheelchair control box components and wheel motor encoder.	154
Figure 9.5	Control algorithms implementation flowchart.	157
Figure 9.6	End-effector's and the wheelchair's commanded and actual trajectories for the hardware implementation (high priority is given to the end-effector trajectory).	160
Figure 9.7	Commanded and actual wheelchair trajectory tracking for both MATLAB simulation and hardware implementation (high priority is given to the end-effector trajectory).	161
Figure 9.8	End-effector and wheelchair position errors (high priority is given to the end-effector trajectory).	162

Figure 9.9	Whole system manipulability measure for the MATLAB simulation and real hardware implementation (high priority is given to the end-effector trajectory).	163
Figure 9.10	Arm joints angle (high priority is given to the end-effector trajectory).	163
Figure 9.11	End-effector's and the wheelchair's commanded and actual trajectories for the MATLAB simulation and hardware implementation (high priority is given to the wheelchair trajectory).	164
Figure 9.12	End-effector trajectory tracking error in XY and XZ planes (high priority is given to the wheelchair trajectory).	164
Figure 9.13	End-effector and wheelchair position errors (high priority is given to the wheelchair trajectory).	165
Figure 9.14	Arm joint velocities (high priority is given to the wheelchair trajectory).	166
Figure 9.15	Whole system manipulability measure for the MATLAB simulation and real hardware implementation (high priority is given to the wheelchair trajectory).	166
Figure 9.16	Hardware implementation testing environment with wheelchair and end-effector commanded trajectories, manipulated object and obstacles.	167
Figure 9.17	Real hardware implementation testing environment.	167
Figure 9.18	The commanded and actual trajectories of the wheelchair and the end-effector for: (a) Predefined-translation case.	169
Figure 9.19	Wheelchair velocities for three test cases.	170
Figure 9.20	Manipulability measure of both undefined-translation cases.	171
Figure 9.21	Motion sequence of WMRA-II.	173

ABSTRACT

A mobile manipulator is a robotic arm mounted on a robotic mobile platform. In such a system, the degrees of freedom of the mobile platform are combined with that of the manipulator. As a result, the workspace of the manipulator is substantially extended. A mobile manipulator has two trajectories: the end-effector trajectory and the mobile platform trajectory. Typically, the mobile platform trajectory is not defined and is determined through inverse kinematics. But in some applications it is important to follow a specified mobile platform trajectory. The main focus of this work is to determine the inverse kinematics of a mobile manipulator to follow the specified end-effector and mobile platform trajectories, especially when both trajectories cannot be exactly followed simultaneously due to physical limitations. Two new control algorithms are developed to solve this problem.

In the first control algorithm, three joint-dependent control variables (spherical coordinates D , α and β) are introduced to define the mobile platform trajectory in relation to the end-effector trajectory and vice versa. This allows direct control of the mobile platform motion relative to the end-effector. Singularity-robust and task-priority inverse kinematics with gradient projection method is used to find best possible least-square solutions for the dual-trajectory tracking while maximizing the whole system manipulability. MATLAB Simulated Planar Mobile Manipulation is used to test and optimize the proposed control system. The results demonstrate the effectiveness of the control system in following the two trajectories as much as possible while optimizing the whole system manipulability measure.

The second new inverse kinematics algorithm is introduced when the mobile platform motion is restricted to stay on a specified virtual or physical track. The control scheme allows

the mobile manipulator to follow the desired end-effector trajectory while keeping the mobile platform on a specified track. The mobile platform is moved along a track to position the arm at a pose that facilitates the end-effector task. The translation of the redundant mobile manipulator over the mobile platform track is determined by combining the mobility of the platform and the manipulation of the redundant arm in a single control system. The mobile platform is allowed to move forward and backward with different velocities along its track to enable the end-effector in following its trajectory. MATLAB simulated 5 DoF redundant planar mobile manipulator is used to implement and test the proposed control algorithm. The results demonstrate the effectiveness of the control system in adjusting the mobile platform translations along its track to allow the arm to follow its own trajectory with high manipulability. Both control algorithms are implemented on MATLAB simulated wheelchair mounted robotic arm system (WMRA-II). These control algorithms are also implemented on real the WMRA-II hardware.

In order to facilitate mobile manipulation, a control motion scheme is proposed to detect and correct the mobile platform pose estimation error using computer vision algorithm. The Iterative Closest Point (ICP) algorithm is used to register two consecutive Microsoft Kinect camera views. Two local transformation matrices i. e., Encoder and ICP transformation matrices, are fused using Extended Kalman Filter (EKF) to filter the encoder pose estimation error. VICON motion analysis system is used to capture the ground truth of the mobile platform. Real time implementation results show significant improvement in platform pose estimation. A real time application involving obstacle avoidance is used to test the proposed updated motion control system.

CHAPTER 1

INTRODUCTION

1.1 Introduction

A mobile manipulator is a robotic system that consists of a robotic arm mounted on a mobile platform. In such a system, the degrees of freedom of a mobile platform are combined with that of a robotic manipulator. Therefore, the workspace of the manipulator is extended by the mobile platform. Mobile manipulators potentially offer tremendous opportunities to perform a wide range of tasks which are not possible with stationary manipulators. Nevertheless, tacking advantage of such a system brings about a number of challenges:

1. Combining mobility with manipulation usually creates kinematic redundancy.
2. The mobile platform is often subject to nonholonomic constraints.
3. The mobile platform usually has lower accuracy and slower dynamic response than a robot manipulator.
4. Typically, the task has to be divided into small movements carried out with the manipulator and large movements executed by the platform [7].

In general, a minimum of 6 DoF are needed to fully describe the pose of an object in space: 3 DoF are needed to specify the Cartesian position of the object and 3 DoF are needed to present the object orientation. Therefore, at least 6 DoF, or six joints, are needed in a robotic system in order to have full manipulation capability of an object in space.

Redundancy happens when the number of DoF or the number of robotic system joints exceeds the number of controlled variables. A kinematically redundant mobile manipulator

has more DoFs than required to execute its task. In such a case, the inverse kinematics problem provides an infinite number of solutions. From these redundant solutions, mobile manipulator configurations, as well as a motion trajectory, can be chosen to best satisfy the desired secondary objectives, such as avoiding joint limits, singularities and obstacles.

The mobile manipulator system consists of two subsystems: a manipulator and a mobile platform. These two subsystems can have two separate trajectories: end-effector trajectory and mobile platform trajectory. To perform complex tasks, both trajectories may have to be controlled simultaneously. Some examples of these tasks are: picking up an object while moving and avoiding an obstacle simultaneously as shown in Figure 1.1; opening and going through a spring loaded door as shown in Figure 1.2; assembling or fabricating large-scale parts as shown in Figure 1.3, and sorting items in a warehouse such as an Amazon fulfillment center as shown in Figure 1.4.

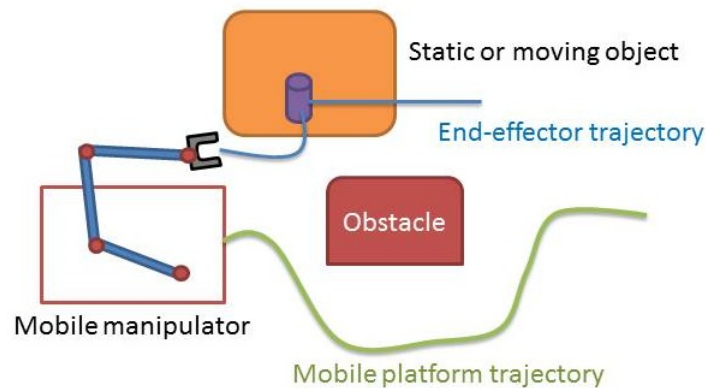


Figure 1.1: Schematic diagram of a, “picking up an object while moving and avoiding an obstacle,” task.

Completion of these tasks requires control of the end-effector and mobile platform trajectories. For example, if the task is to grasp a static or moving object while there is an obstacle in the way of the mobile platform (Figure 1.1), the end-effector will have a trajectory towards the object, and the mobile platform will have another independent trajectory to avoid the obstacle. To ensure grasping the object, and at the same time avoiding the

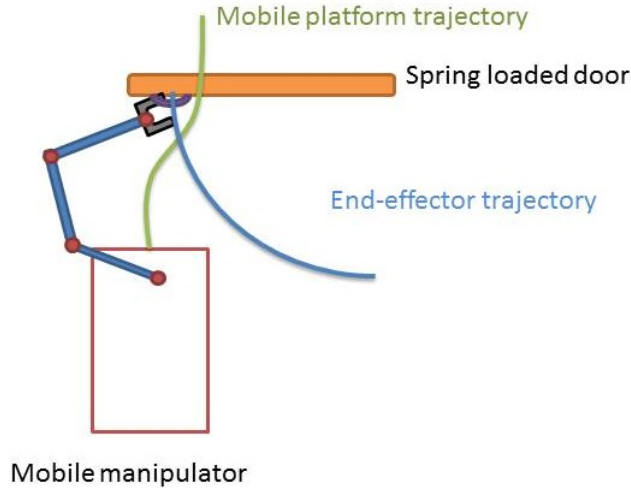


Figure 1.2: Schematic diagram of an, “opening and going through a spring loaded door,” task.

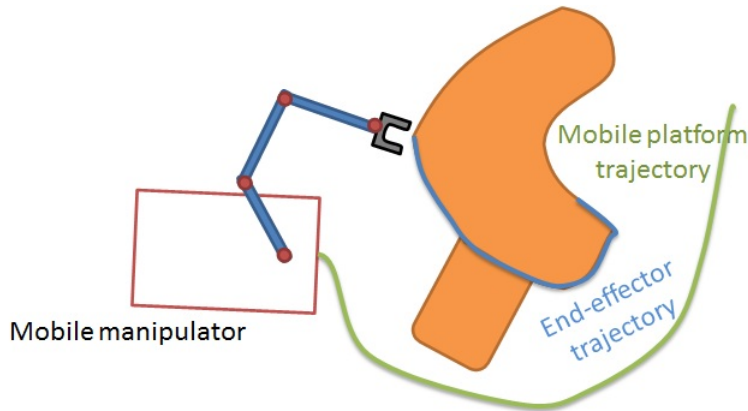


Figure 1.3: Schematic diagram of an, “assembly of large-scale parts by welding,” task.

obstacle, both trajectories should be followed simultaneously. Same scenario can happen in the task of sorting items in full automated fulfillment centers. (refer to Figure 1.4).

Both end-effector and mobile platform trajectories can be planned offline or online according to task requirements and environment conditions. Path planning is not within the scope of this work. In many instances, inverse kinematics for the manipulator and the mobile platform can be solved using well known techniques [8,9].

In this work, the term “dual-trajectory” represents both the end-effector and mobile platform trajectories. Additionally, the terms “dual-trajectory” and “mobile manipulator



Figure 1.4: Amazon fulfillment center [1].

trajectory” will be used interchangeably. Figure 1.5 shows a mobile manipulator with dual-trajectory.

This dissertation is aimed at solving the inverse kinematics problem for a mobile manipulator when both the end-effector and the mobile platform trajectories cannot be followed simultaneously in the conventional ways, and it is necessary to prioritize one over the other. Two novel algorithms are developed to address this problem. Specifically, the following new ideas are addressed:

1. New control variables are introduced to the task vector to control the mobile platform related to the end-effector.
2. Optimization of the translation of the mobile platform along a prespecified track.

The most commonly used redundancy resolution algorithms for mobile manipulators provide the ability to design the controller in two spaces: operational space and null space. In the operational space, a trajectory-following controller is designed to eliminate the error

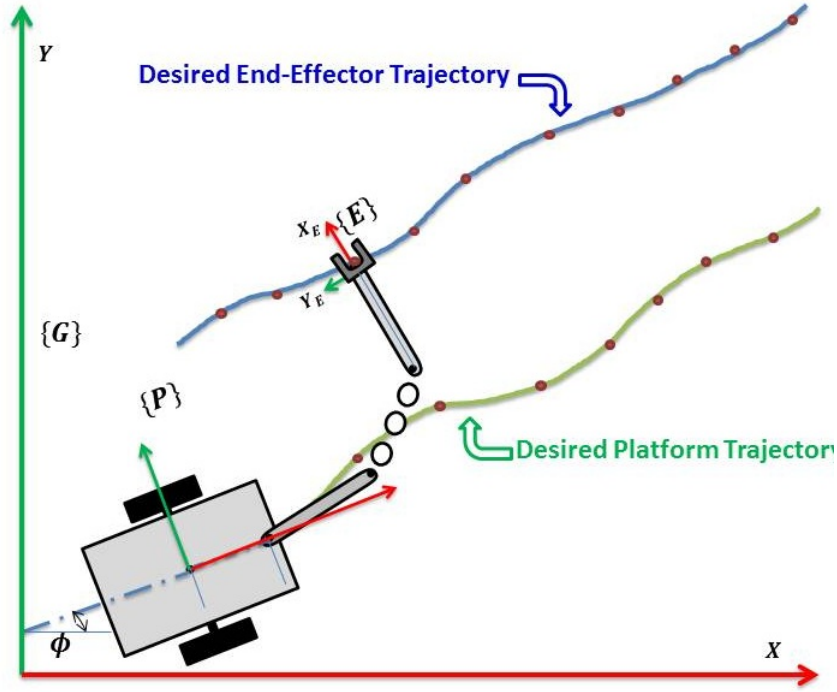


Figure 1.5: A mobile manipulator with predefined separate trajectories for the end-effector and the mobile platform.

between the end-effector's actual pose and its desired pose. In the null space, an adjustment process is designed to optimize some criteria without changing the state of the end-effector.

A typical form of these controllers is described by the following:

$$\dot{q} = \underbrace{J^\# \dot{r}}_{\text{Operational space}} + \underbrace{(I - J^\# J)H}_{\text{Null space}} \quad (1.1)$$

where \dot{q} is the velocity input vector, \dot{r} is the desired velocities of the end-effector, $J^\#$ is the pseudo-inverse of the system Jacobian J , and H is an arbitrary vector. This approach demonstrates a direct control of the end-effector pose. However, the platform pose will be indirectly controlled using some optimization criteria in the null space. Therefore, there is often a lack of a precise control for the platform.

1.2 Motivation

Mobile manipulators have been used to perform complex and dangerous tasks in some fields, such as material handling [10] and space exploration [11]. Hammer et al. [12] utilized a mobile manipulator to perform assembly tasks while the vehicle base moves on a track. Zhou et al. [13] discussed utilization of mobile manipulator systems in the aerospace manufacturing industry. Authors also listed many other possible tasks that a mobile manipulator can perform. The Southwest Research Institute [14] developed a mobile manipulator system to work accurately in a large work envelope such as aerospace manufacturing, ship building and wind turbine manufacturing.

The previously mentioned tasks need to use both the mobility and manipulation of the mobile manipulator. The mobile platform gives the robotic arm not only the ability to move towards the task workspace, but also the ability to be positioned in a place in which the arm will have maximum possible manipulation. In addition, mobile platform sometimes has to have a certain orientation. From this, it can be understood that controlling both the end-effector and the mobile platform trajectories (dual-trajectory) play a crucial role in successfully executing a desired task [15]. This guarantees that the mobile platform, while avoiding an obstacle, not only brings the robotic arm to a preferred configuration, but also orients itself to perform its tasks more effectively.

The mobile platform trajectory can be considered as a band of possible trajectories, and the mobile manipulator specific trajectory can be planned online or offline based on the status of the mobile manipulator, task requirements and optimization criteria such as, “keeping high manipulator manipulability measure,” “avoiding an obstacle” and “maintaining a certain mobile platform orientation.” However, in some cases, while the mobile platform has to avoid an obstacle, it may restrict the ability of the mobile manipulator to track the end-effector trajectory in a precise fashion. There will also be situations where a planned trajectory

of a mobile platform will have to be altered as it may not be possible to follow both the end-effector and mobile platform simultaneously due to the hardware limitations.

In this work, cases were considered where the end-effector and mobile platform planned trajectories are not trackable simultaneously by the mobile manipulator due to mechanical limitations. Innovative ways are proposed to deal with the inverse kinematics problem in such situations.

For instance, in a navigation stage, it is possible to alternate the order of priority between the end-effector and mobile platform trajectories. For example, in avoiding an obstacle that is in the way of the platform, higher priority may be given to the platform trajectory, and the end-effector trajectory can be altered until the obstacle is completely avoided. Then the original trajectory can be resumed. On the other hand, if the end-effector trajectory is more important, as when the end-effector is holding a doorknob to open a door and it has to follow a circular trajectory, the higher priority is given to the end-effector trajectory and a position error can be introduced to the mobile platform trajectory. In other words, using pseudo-inverse methods, the inverse kinematic solutions are the Least-Norm (LN) solutions when it is possible to follow both trajectories exactly. Otherwise, the solutions are the Least-Square (LS) and Least-Norm solutions that follow trajectories as close as possible to the desired trajectories. In this case, least possible tracking errors are introduced to the mobile manipulator trajectory, with specific priority given between the end-effector and mobile platform trajectories.

As shown in Figure 1.6, it is possible to change the position of the platform on its track to keep the mobile platform on its track and at the same time to follow the end-effector trajectory accurately. The mobile manipulator in Configuration (1) is not capable of following the end-effector trajectory due to mechanical limitations. However, moving the mobile manipulator to Configuration (2) along a predefined track allows the end-effector to follow its desired trajectory.

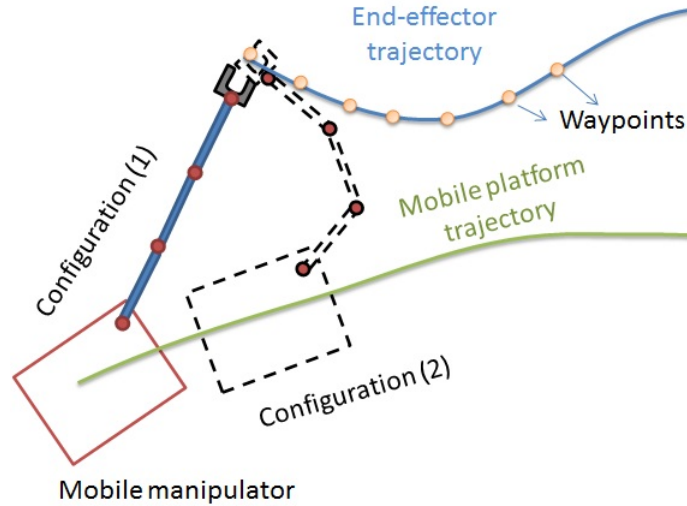


Figure 1.6: Follow end-effector trajectory by changing the mobile platform position along its track.

1.3 Problem Statement

As mentioned previously, the mobile manipulator trajectory is the combination of the end-effector and the mobile platform trajectories. Suppose that a mobile manipulator has to follow a predefined end-effector trajectory, such as the welding irregular large-scale parts or the inspection process of an airplane or a ship body (Figure 1.3). The mobile platform also has to follow its own path to avoid an obstacle. The obstacle is assumed to be fairly low so it would only block the way of the mobile platform. Figure 1.3 illustrates the situation in which the end-effector has to follow a specific trajectory (blue line), and at the same time, the mobile platform has to follow another trajectory (green line) that allows it to avoid an obstacle in its way.

So in this situation, two trajectories are planned for the mobile manipulator to follow. An intuitive way to follow these two trajectories is by waypoints. This means that the locations of the end-effector and the mobile platform are known. Then the mobile manipulator is commanded to follow each waypoint until the end. In some cases, the planned dual-trajectory is not within the mobile manipulator reachability due to physical limitations. The main

contribution of this work consists of designing and developing control algorithms that can track end-effector and mobile platform trajectories according to the following three options:

1. Follow the dual-trajectory with maximum manipulability and allow error in the mobile platform trajectory when both trajectories are not traceable.
2. Follow the dual-trajectory with maximum manipulability and allow error in the end-effector trajectory when both trajectories are not traceable.
3. Follow the dual-trajectory with maximum manipulability and compromise the mobile platform's waypoints by adjusting the mobile platform position on the mobile platform's path to satisfy both the mobile platform and the end-effector trajectories.

The aforementioned situations can be illustrated graphically as shown in Figure 1.7. In this figure, a planned dual-trajectory for a mobile manipulator is shown, where E_T and P_T are the end-effector trajectory and mobile platform trajectory, respectively. The end-effector trajectory E_T describes the full poses of the end-effector at each time instance. The end-effector pose is a six-dimensional vector (3 for the position and 3 for orientation). The mobile platform trajectory P_T describes the mobile platform poses. The mobile platform pose is a three-dimensional vector (position x_P , y_P , and orientation ϕ). Due to nonholonomic constraints of the mobile platform, the infinitesimal change in the mobile platform orientation can be calculated from the infinitesimal changes in its position coordinates.

In Figure 1.7, the curly brackets $\{*_o\}$, $\{*_d\}$, $\{*_c\}$ and $\{*_F\}$ indicate the starting, desired, current and final frames or poses of the mobile platform $\{P_*\}$ and the end-effector $\{E_*\}$. The aim is to kinematically control the mobile manipulator to follow a dual-trajectory. Therefore, at any control instance, it is necessary to define the desired and current poses of the end-effector and mobile platform ($\{E_d\}$, $\{E_c\}$, $\{P_d\}$ and $\{P_c\}$). The primary goal is as follows,

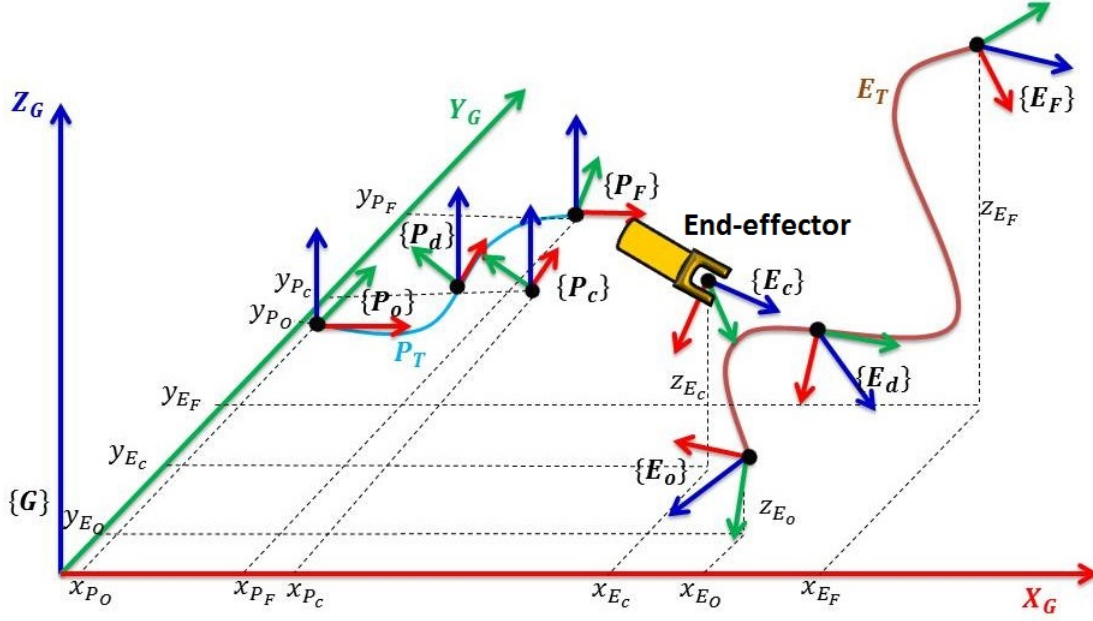


Figure 1.7: A mobile manipulator dual-trajectory (E_T and P_T) with the starting, desired, current and final end-effector and mobile platform poses ($\{E_o\}$, $\{E_d\}$, $\{E_c\}$, $\{E_f\}$, $\{P_o\}$, $\{P_d\}$, $\{P_c\}$, and $\{P_f\}$).

(refer to Figure 1.8 for variable definitions):

$$i \in \{1, 2, \dots, N\} \left(\begin{array}{c} \rho_{E_i}(E_c, E_d) \\ \rho_{P_i}(P_c, P_d) \end{array} \right) \quad (1.2)$$

where $\vec{\rho}_{E_i}(E_c, E_d) = \vec{E}_d - \vec{E}_c$ and $\vec{\rho}_{P_i}(P_c, P_d) = \vec{P}_d - \vec{P}_c$ as shown in Figure 1.8. Other secondary goals can be defined as follows:

$$\text{optimize}_{i \in \{1, 2, \dots, N\}} \text{performance measure} \left(\min \left(\begin{array}{c} \rho_{E_i}(E_c, E_d) \\ \rho_{P_i}(P_c, P_d) \end{array} \right) \right) \quad (1.3)$$

In summary, while minimizing the trajectory tracking errors for the end-effector and the mobile platform, we attempt to optimize a performance measure which can be the manipulability measure or the joint limit avoidance functions.

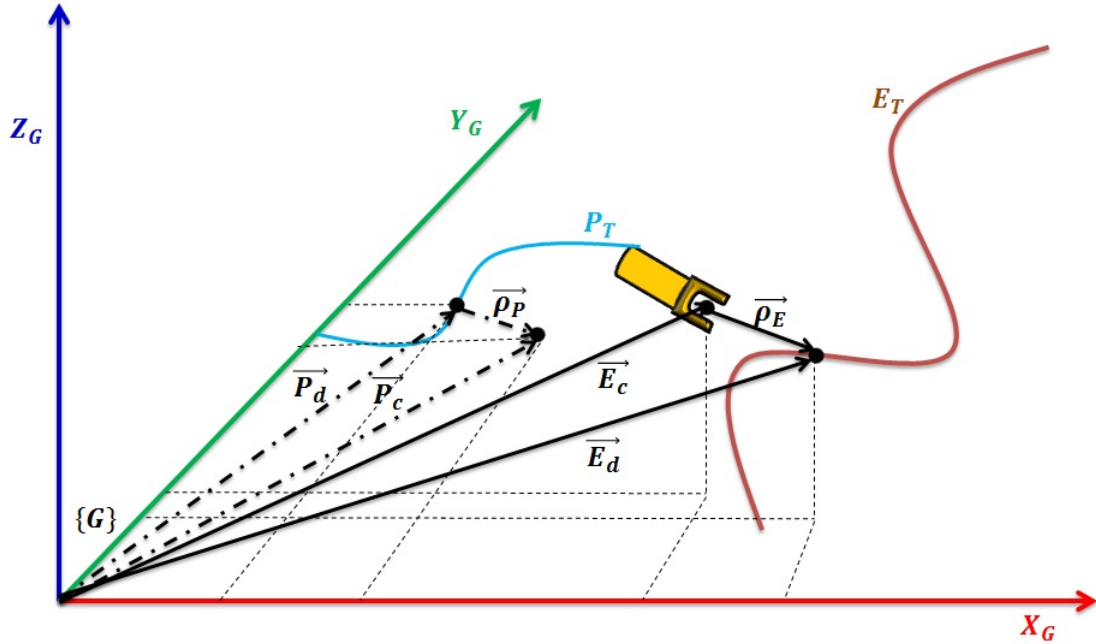


Figure 1.8: A mobile manipulator with separate trajectories for the end-effector and the mobile platform.

1.4 Dissertation Objectives

As stated previously, only the cases when the mobile manipulator trajectory can not be tracked have been considered. The goal is to design and implement a controller which is capable of tracking both predefined trajectories of the end-effector and the mobile platform, to the extent possible with given priorities and optimizing varies performance criteria. The objectives of this work can be summarized as follows:

1. Develop, optimize and test a dual-trajectory control system for redundant mobile manipulators. This control system combines the manipulation of a robotic arm and the mobility of a mobile platform in a single control system.
2. Redundancy resolution algorithms will be used to avoid singularities, obstacles, and joint limits.

3. New control variables will be introduced to the task vector allowing for presenting the mobile platform trajectory relative to the end-effector trajectory and vice versa. These control variables (spherical coordinates) allow for setting a limit the robotic arm stretch.
4. Inverse kinematics considering task priority will be used to alternate the task priority between the end-effector and the mobile platform trajectories.
5. Design and implement a new control scheme that is capable of adjusting the mobile platform locations along a prespecified track, allowing the mobile manipulator to track both trajectories.
6. A comprehensive and flexible MATLAB simulation program will be developed to test and optimize the proposed controllers.
7. Implement the proposed controllers on a real hardware. The controllers will be implemented on the WMRA system.
8. The proposed control algorithms will be evaluated for a complete “real-world” task execution.

1.5 Dissertation Outline

In this dissertation, Chapter 2 provides a background on the previous work done in the field of mobile manipulation and redundant mobile manipulators, as well as the use of computer vision for mobile robot pose estimation. Chapter 3 introduces the kinematic model of redundant mobile manipulators. In Chapter 4, mobile manipulator kinematic control theories and methods are discussed. In Chapter 5, we introduce a novel dual-trajectory tracking control algorithm using joint dependent control variables, along with its MATLAB simulation results of the implemented controller on a simulated planar mobile manipulator.

Chapter 6 introduces a new control algorithm of dual-trajectory tracking with free mobile base translation along a specified track, and its MATLAB simulation results for a simulated planar mobile manipulator. Chapter 7 discusses the mobile robot pose estimation and correction algorithms, along with its implementation results for the Physical Wheelchair Mounted Robotic Arm (WMRA) system are presented. In Chapter 8, simulation implementation and results for the WMRA system are presented and discussed for the two dual-trajectory control algorithms. In Chapter 9, hardware implementation and results for the physical WMRA system are shown. Chapter 10 concludes the dissertation with a summary, discussion and recommendations.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

2.1 Introduction

Standard robotic arms consist of several links that are connected together by joints. Traditionally, these manipulators are mounted to stationary bases, and are used to assist human workers in performing tasks that are dangerous, dirty, and repetitive such as mining, material handling and manufacturing. One of the drawbacks of these manipulators' architecture is the limited workspace due to the limitations in link dimensions. As a result, the manipulator's workspace is restricted to small and structured environments. This makes the tasks that can be executed by these manipulators significantly limited. Therefore, to use these manipulators in more applications, mobility of a base have to be combined with the manipulation abilities of a robotic arms. Gardner and Velinsky in [16] presented a method that determined the effect of mounting position of the arm on the whole system mobility. This method is called scaled manipulability ellipses and was applied on automated highway construction and maintenance tasks. Yamamoto and Yun in [17] presented a control algorithm that could compensate the effect of dynamic interaction between the arm and base while the end-effector followed a commanded path. Simulation results showed that the proposed algorithm is able to converge the tracking error to zero.

Mobility and manipulation are two abilities that are offered by robotic systems referred to as mobile manipulators. These systems consist of a robotic arm mounted on a mobile base. Compared to stationary manipulators, mobile manipulators have extended workspaces that allow them to perform tasks that need locomotion capabilities and manipulation abil-

ities. Therefore, the applications of these robotic systems are widely extended. While the manipulators are mainly deployed in factory environments, such as assembly lines in car manufacturing, mobile manipulators are used in additional applications such as space exploration, search and rescue, and in assisting people with disabilities.

Although, combining mobility and manipulation enhances the applications of manipulators, it brings about a number of challenges [7]:

1. The combination of mobility and manipulation usually creates redundancy.
2. The mobile platform is often subject to nonholonomic constraints while the manipulators are usually unconstrained to such a limitation.
3. The mobile platform typically has slower dynamic response and less accuracy than the manipulator.

2.2 Mobile Manipulator Classification

The existence of first well documented mobile manipulators can be traced back to 1966, as stated by Srinivasa et al. in [18]. In that time, the robot SHAKEY was built. Shakey was equipped with different sensors and designed to be a testbed for AI planning. Ever since, mobile manipulators have gained a lot of interest and have been used in many applications. Mobile manipulators can be grouped into categories depending on: the environment in which the mobile manipulators are deployed; the application, the way in which mobile manipulators are used, and the locomotion which dictates mobile manipulator's motion.

2.2.1 Mobile Manipulators Environments

Mobile manipulators can be used in different environments, such as on the ground, underwater, and aerial environment.

2.2.1.1 Aerial Manipulators

Using manipulation of a robotic arm mounted to manned or unmanned aerial vehicles (UAVs) is interesting because these vehicles can reach many locations that are not accessible by other types of mobile manipulators, such as wheeled mobile manipulators. Huber et al. [2] presented the first aerial manipulation consisting of a helicopter and a redundant robotics arm (Kuka arm). Figure 2.1 depicts this first experimental platform for aerial manipulation.



Figure 2.1: First experimental platform for aerial manipulation [2].

2.2.1.2 Underwater Manipulation

Underwater robots, referred to as “Autonomous Underwater Vehicles (AUVs)” have a crucial role in offshore operations such as marine search and rescue as well as oil and gas production and exploration. The need for underwater intervention tasks has led to a new concept during the ‘90s, named “Autonomous Underwater Vehicles for Intervention,” (I-AUV) [19]. Figure 2.2 shows an example of such an underwater intervention project [3]. For a summary of the most recent international underwater intervention projects, refer to [19].

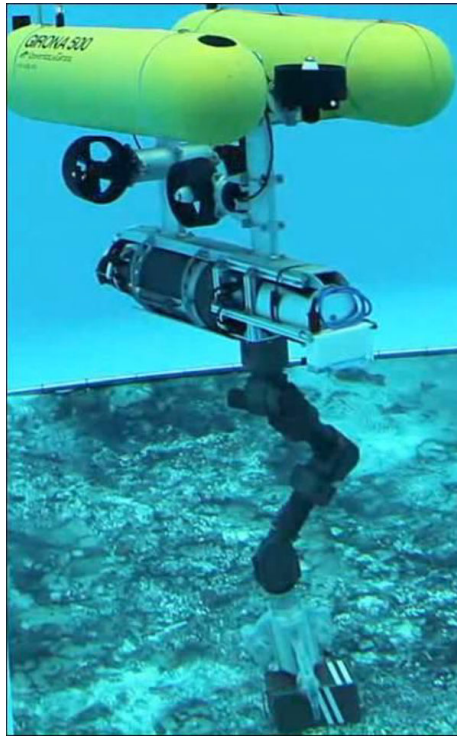


Figure 2.2: TRIDENT underwater intervention project [3].

2.2.1.3 Grounded Mobile Manipulators

Another type of mobile manipulators is the mobile manipulator that moves on the ground, such as wheeled mobile manipulators. There are many examples for this type of mobile manipulators. One recent example is the Valeri Project that was announced in 2013 [20] in which mobile manipulators for aerospace production were developed to work with humans on the production floor [13]. Figure 2.3 shows a mobile manipulator “Little Helper” that was built at Aalborg University in Denmark.

2.2.2 Mobile Manipulator Applications

Mobile manipulators have many different applications that can be arranged in four domains [5]: professional/service(home and health care), space exploration, military, and industry. Figure 2.4 shows these four domains.



Figure 2.3: “Little Helper” industrial mobile manipulator [4].



Figure 2.4: Main application domains of mobile manipulation [5].

2.3 Mobile Manipulator Control

“Robot control refers to the way in which the sensing and action of a robot are coordinated [21].” Designing an adequate controller is an important part of designing a robotic device.

Many researches have developed different ways of controlling these devices both in simulation and in real hardware.

Lin and Goldenberg in [22] proposed a feedback linearization control methodology based on a neural network (NN) for kinematically constrained mobile manipulator. The Lyapunov Theory was used to stabilize the whole system and to reject disturbances. Two NN controllers were designed to control the base and the arm separately while considering the dynamic coupling. Chitta et al. in [23] proposed a graph-based representation, optimal for efficient planning and open doors motion, to deal with a high-dimensionality problem. Mobile manipulator PR2 was used for implementation and testing of the proposed approach. Cameron et al. in [24] discussed the integration of mobility and manipulation of a mobile manipulator in a dynamic environment. The authors introduced techniques that are suitable for dynamic environments and tools that can be used for kinematic and dynamic modeling of mobile manipulators. Moreover, they integrated the kinematics and dynamics of the system with reactive control algorithm. Simulation results show the effectiveness of this approach. Ögren et al. in [25] proposed a potential field algorithm that integrated a task potential, making the arm end-point track a known path; coordination potential, causing the mobile platform to put the end-effector in the middle of its workspace, and an obstacle avoidance potential to produce motion for the mobile manipulator to avoid an obstacle. Simulation results showed the effectiveness of the proposed control algorithm. Brock et al. in [26] proposed novel approaches based on elastic strip framework. These techniques allowed for task-consistent obstacle avoidance and motion behavior. In addition, general transition approaches were presented allowing smooth transition between different motion behaviors to secure the performing of the high-priority behavior. Petersson et al. in [27] have proposed a door opening controller which relied on a hybrid dynamic system model. This model was integrated with an online scheme for estimation of the door model. The experimental results showed the robustness of the proposed technique.

Generally, there are two different approaches of mobile manipulator control algorithms found in the literature. One approach is considering the mobile manipulator system as two subsystems: mobile platform and manipulator arm. In this case, each subsystem controller is constructed separately and the coordination between the manipulator and the mobile platform controllers must be considered. The other control approach is a unified control of both subsystems [28].

2.3.1 Coordinated Control of Mobile Manipulators

Some researchers consider mobile manipulators systems as two separate systems which are the robotic arm and the mobile base. The following are some works that have been accomplished in this area.

Phan et al. in [29] proposed a decentralized motion control algorithm of welding mobile manipulators. In their work, the mobile manipulator was kinematically modeled as two separate subsystems: the mobile platform and the robotic arm. They presented two independent controllers based on the Lyapunov control function to control the two separate subsystems. The proposed controllers were tested using simulation and physical hardware. The results showed a good performance and proved the effectiveness of the proposed controllers. Similarly, Fruchard et al. in [30] proposed a framework for the feedback control of mobile manipulators. This framework focuses on motion coordination between the mobile platform and the manipulator. In this approach, the holonomic robotic arm velocity and the mobile base velocity were determined separately by minimizing a second cost function that represents the platform in the manipulator task space. Hamner et al. [31] presented an autonomous mobile manipulator for a “peg-in-hole” type assembly task. This system overcame its inherent uncertainties and exception using three control strategies: coordinated mobile platform and the arm control, combined visual and force servoing, and error detection and correction through flexible task level control. The mobile manipulator system

demonstrated experimentally high system robustness and reliability for the assembly task. A similar approach was used in [32].

Chung et al. [33] also proposed a similar controller for mobile manipulators by decomposing them into two separate subsystems: mobile base and robotic arm. They presented a redundancy resolution scheme in which the robotic arm was commanded to track the desired trajectory given in the task space and the mobile base was responsible for positioning the arm at a desired point in which the singular arm configuration was avoided. An interactive controller algorithm was developed to coordinate the two separate subsystem's motion. This control algorithm had two nonlinear controllers that were designed based on the redundancy resolution scheme as shown in Figure 2.5. The simulation results showed a good performance of the interaction controller based on their trajectory following task.

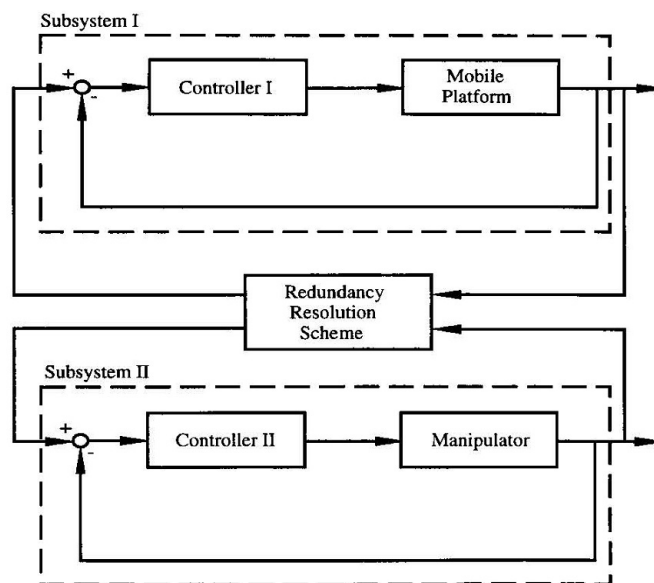


Figure 2.5: Coordinated control for mobile manipulator composed of two subsystems.

All aforementioned works consider the mobile manipulator systems as two separate subsystems: mobile base and robotic arm. The other approach considered the mobile manipulator as one system and a single controller was designed to control the motion.

2.3.2 Mobile Manipulation Control

Yamamoto et al. [34] stated that considering the mobile manipulator systems as two separate subsystems makes control and planning problems easier. However, much more effective and efficient motion control could be achieved by combining the mobile base mobility and the arm manipulation. The authors presented a unified algorithm to the task space analysis of a wheeled mobile manipulator as shown in Figure 2.6. The considered system

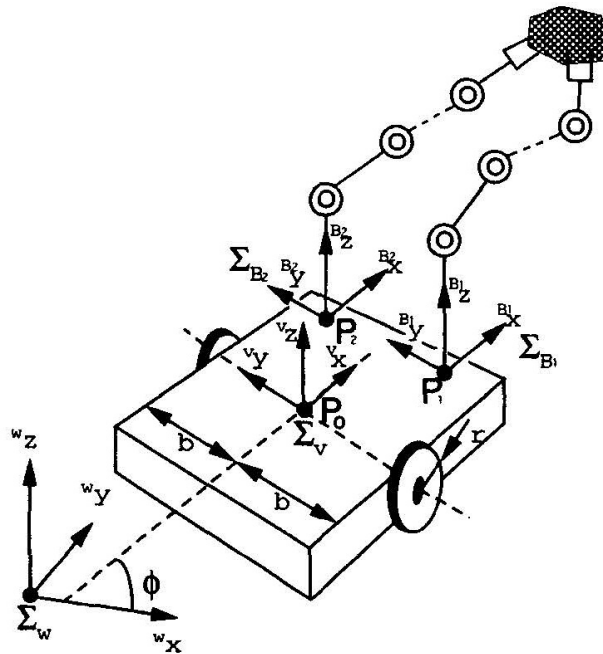


Figure 2.6: Wheeled mobile manipulator with two manipulators.

consisted of two manipulators mounted on top of a mobile platform handling a common object. They introduced the task space ellipsoid for the wheeled mobile manipulator in both kinematic and dynamic cases. The ellipsoid was taken as a measure for visualizing the contribution of the manipulator and the mobile platform to a task performance by integrating the manipulation of the arms with the mobility of the platform as one unified measure. This measure could be useful for the task space analysis of a single mobile manipulator as well as for the coordination of multiple arms, mobile robots, or mobile manipulators.

Andaluz et al. [28] presented a unified motion controller for mobile manipulators. This controller was for solving point stabilization, trajectory tracking and path following. The control problem is solved based on the mobile manipulator kinematic model with dynamic compensation as shown in Figure 2.7. The Lyapunov Method was used to prove the stability and robustness of the proposed control system. Real experiments were used to test and evaluate the proposed controller.

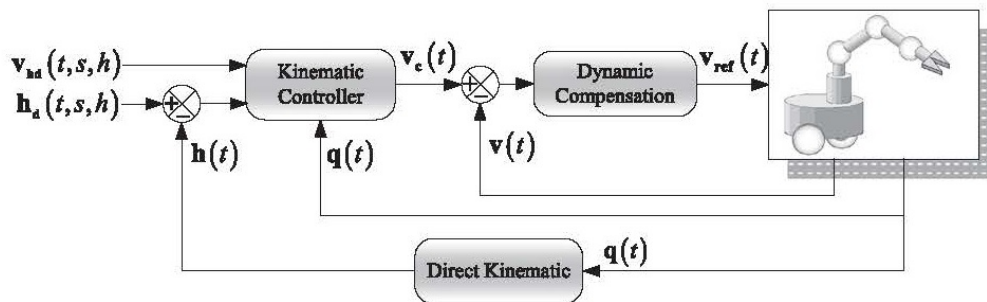


Figure 2.7: Block diagram of the motion control system for mobile manipulators.

2.3.3 Redundant Mobile Manipulator Control

In general, a minimum of 6 DoF are needed to describe the pose of an object in space: 3 DoF are needed to specify the Cartesian position of the object (x , y , and z) and 3 DoF are needed to present the object orientation. Therefore, at least 6 DoF, or six joints, are needed in a robotic arm in order to have full manipulation of an object in space. Redundancy happens when the number of DoF, or the joints of a robotic arm, exceeds the number of controlled variables. In this case, the traditional inverse kinematics for a close form solution is inadequate, and new algorithms have to be used. Furthermore, redundancy resolutions produce infinite solutions for the same task. This introduces another problem; how to choose the best solution that fulfills additional criterion. This problem is called optimization. These two problems are subjected to much research where redundancy is utilized to perform additional tasks using optimization criteria.

Mobile manipulators have gained a great deal of interest because of their applications to a wide range of complicated robotic tasks. Generally, combining the robotic arm DoF with the mobile robot DoF yields a redundant robotic system if the total DoF exceed the number of variables to be controlled in Cartesian space. Classic redundancy resolution for redundant robotic arms can be updated to be used for the redundant mobile manipulators. In [8], Nakamura proposed many approaches for optimizing some measures of performance that focused on the differential kinematics. Most of the research on controlling the redundant mobile manipulators focused on controlling the end-effector to follow a predefined trajectory while the mobile platform followed a random trajectory based on certain optimization criteria. Pin et al. in [35] presented an optimization criterion to solve redundancy based on Full Space Parameterization method (FSP). Analytical solutions were given for two constrained motion cases. Comparative trajectories that combined mobility of the base and the manipulation of the arm were used to test these solutions and demonstrate the robustness of the FSP algorithm. Chen et al. in [36] presented a genetic algorithm approach to motion planning of a mobile manipulator. The authors considered the position and configuration as two criteria to optimize the mobile robot path planning. Simulation results of two cases showed that the performance of the proposed algorithm is better than the conventional search methods.

Jia et al. [37] have proposed a new practical control method for the purpose of minimizing the end-effector trajectory tracking error of a nonholonomic mobile manipulator. In this method, an adaptive motion preference is set to coordinate the motion between the manipulator and mobile base. They used a weighted pseudo-inverse to implement a weight matrix that is a function of the measurement of manipulability index. Figure 2.8 shows the results of the traditional and proposed methods. The authors in this work only controlled the end-effector trajectory and the mobile base followed uncontrolled (random) trajectory. A similar approach was followed in [38–41].

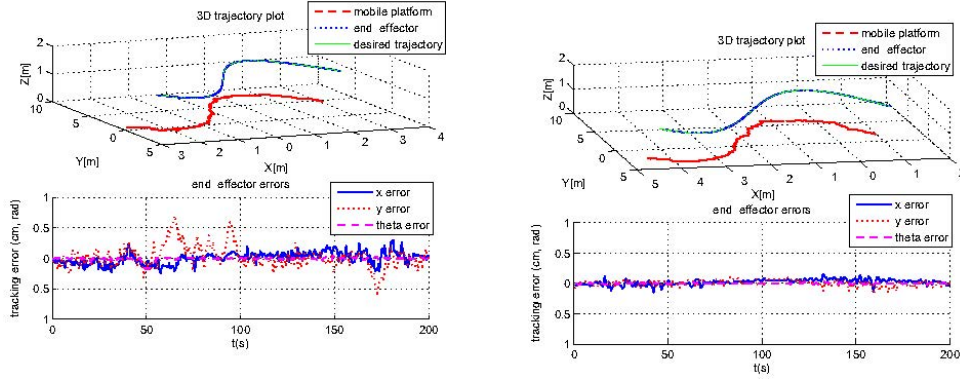


Figure 2.8: Results of the traditional kinematic control and the kinematic control with motion distribution.

White et al. [42] implemented a redundancy resolution algorithm for a nonholonomic wheeled mobile manipulator using independent controllers developed within a decoupled task space and null space. The primary end effector task control was developed to control the end effector's dynamic interaction where the surplus of actuation was used to implement a secondary null space controller. Figure 2.9 shows the setup for the wheeled mobile manipulator and the specified trajectories.

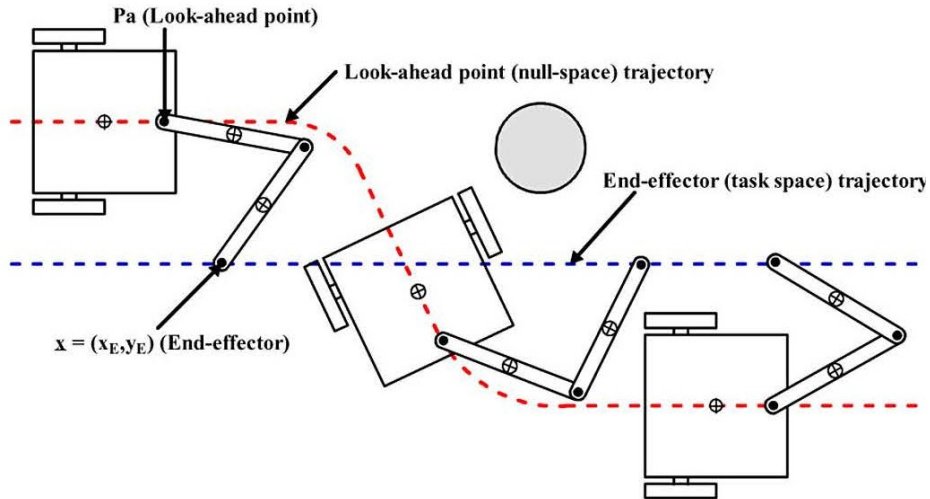


Figure 2.9: Wheeled mobile manipulator with desired end-effector and mobile base trajectories.

The authors controlled the end-effector trajectory in task space while the mobile base trajectory was controlled using the null space. This leads to setting priority to the end-effector trajectory tracking.

2.3.4 Mobile Manipulator Trajectory Tracking Control

The main role of the manipulator is to move the end-effector from a given starting pose (position and orientation) to a desired pose. By mounting a manipulator to a mobile platform, the workspace of this manipulator is extended due to the mobility of the base. This means that the end-effector has its own trajectory, and the mobile base has its own trajectory. These two trajectories will be referred to as dual-trajectory or mobile manipulator trajectory.

As previously explained, most researchers are concerned with tracking the end-effector trajectory while the mobile base trajectory was not controlled. Recently, the trend of the research in redundancy resolution of mobile manipulators is to control separate trajectories for the end effector and the platform [43]. In this section, the focus will be on the mobile manipulator trajectory tracking and the previous work done in this area. Comments will be added to highlight the differences between presented work in this dissertation and previous work.

In many applications, the mobile manipulator is commanded to move the end-effector along a predefined trajectory. Usually, other tasks are planned along with tracking the end-effector trajectory, such as obstacle avoidance and self collision avoidance. This problem has been called by [44] the, “ Motion Planning along End-effector Paths (MPEP).” Nagatani et al. [45] proposed an algorithm that planned the motion of a mobile platform in a way that the manipulability of the end-effector is kept high. Egerstedt and Hu [46] proposed and analyzed an independent control approach for a coordinated trajectory following for mobile manipulators. In their work, the mobile base trajectory was planned such that the planned

end-effector's position was within the work space of the robotic arm. Similarly, Mohri et al. in [47] proposed a mobile manipulator trajectory planning method considering the arm end-point's predefined trajectory. Order of priority was used to solve the trajectory planning method. Simulation results showed the effectiveness of the presented method. Huang et al. in [48] proposed a method in which the Zero Moment Point (ZMP) criterion was used to control the stability of the whole system. In this proposed method, a coordinated approach was followed in which the mobile platform motion was obtained according to the arm manipulability and workspace, while taking into consideration the platform stability in planning the manipulator motion. The simulation results demonstrated the effectiveness of the proposed approach. Yamamoto et al. in [49] presented a motion control system that allowed a mobile manipulator to apply force on a moving object to move it in cooperation with other robotic systems. The role of the mobile platform was to ensure that the manipulator was at the preferred operating region. This allowed for the arm's quick response to small disturbances without encountering singular configurations. The experiment was conducted on a mobile manipulator consisting of a PUMA 250 and a mobile base LABMATE. The results, according to the manipulability measure, demonstrated that the arm was kept in a good configuration. Dong in [50] studied the trajectory following and force tracking control problem for a holonomic and a nonholonomic mobile manipulator with parameter uncertainty. The author proposed adaptive controllers that ensured both the trajectory and force converged to the desired values. Simulation results showed the effectiveness of the proposed controllers. Yamamoto in [51] presented a planning and control algorithm for coordinating motion of a mobile manipulator. The design idea was to control the mobile base so that the manipulator was maintained at a configuration in which the manipulability measure of the arm was maximized. The mobile manipulator was a 2 DoF planar arm mounted on a differential driven mobile base. Dynamic equations for the mobile platform were derived while the arm was considered as a passive device whose dynamics was neglected.

Papadopoulos et al. [52] presented a planning and control algorithm for mobile manipulator systems in order to allow them to follow a desired end-effector and mobile base trajectories. A model-based controller was designed to control trajectory tracking errors. This control algorithm was tested on two simulated mobile manipulator systems consisting of 2 DoF planar robotic arm mounted on a differential-drive platform and a car-like platform. The authors considered both the trajectory of the end-effector and the arm base and they named it “front point”. The front point trajectory was either arbitrary or predefined with a condition that the distance between the end-effector and front point was within the reach of the robotic arm. Figure 2.10 shows the desired end-effector and mobile base trajectories along with simulated trajectories tracking. The authors used a non-redundant robotic system

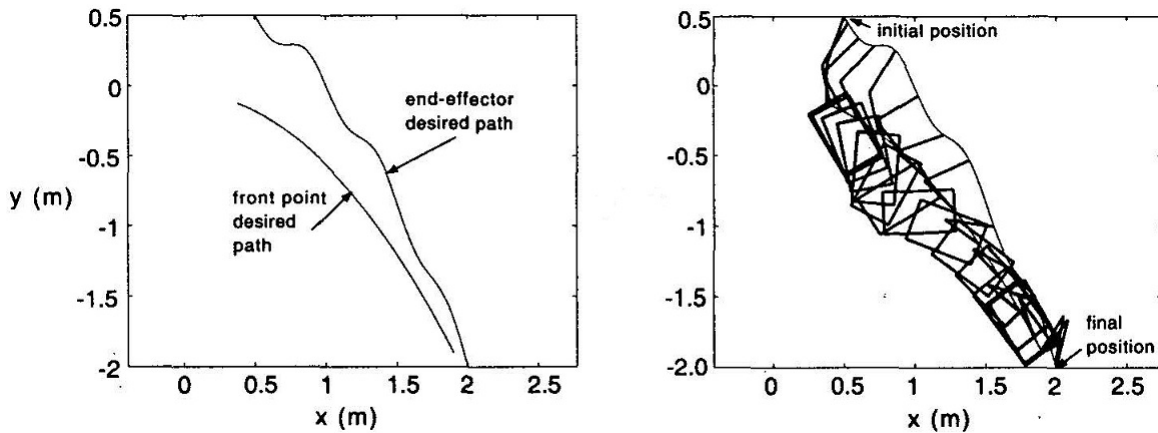


Figure 2.10: Desired end-effector and mobile platform trajectories and mobile manipulator animation.

where the planar Cartesian velocities for the end-effector and the front point $[\dot{x}_E \ \dot{y}_E \ \dot{x}_F \ \dot{y}_F]$ were inputs and two joints angle velocities $[\dot{\theta}_1 \ \dot{\theta}_2]$ for the planar arm and the left and right mobile base wheels velocities $[\dot{v}_l \ \dot{v}_r]$. In addition, they determined or predefined the front point trajectory according to a condition as the distance between the end-effector and the front point was within the reach of the arm. This was to avoid arm singularities.

The work in this dissertation is more general in the sense that the mobile manipulator trajectory is unconstrained and the system is redundant. Kinematic algorithms are imple-

mented to avoid the arm singularities in the case when the mobile manipulator can not track the trajectory due to its limitations. In addition, two control algorithms are proposed to solve this problem. This is the main contribution of this dissertation work.

In [43], Farello et al. controlled separate platform and end-effector trajectories. While the end effector followed a predefined trajectory, the platform had to follow a limited approximation of a secondary trajectory as part of the redundancy resolution and optimization algorithm. The authors used a criterion function for weighted optimization to set weights for the mobile platform motion. The mobile platform motion was executed in three stages, which were rotation, translation and orientation. Also, the weight matrix was used to alternate the mobile base motion in the three stages.

In their work, the authors used null space to control the mobile base. In this work, the mobile base motion is controlled in the task space where the mobile base trajectory variables are included in the task space.

Baerlocher et al. [53] analyzed two formulations for the kinematic control of redundant manipulators according to task prioritization. They addressed some problems associated with the two formulations, and they suggested solutions and improvements. Kanoun et al. [54] proposed a hierarchical task regulation framework based on quadratic programs to handle inequality constraints.

A majority of the aforementioned works focused on controlling a platform trajectory via certain optimization criteria. However, in this work, two novel control schemes that are capable of controlling separate trajectories of the end-effector and the mobile platform are proposed:

1. Introducing new variables in the task space to control the mobile platform motion.

This will give direct control of the mobile platform's trajectory.

2. Adjusting the translations of the mobile base along a specified track to give more flexibility to the mobile manipulator to follow the desired trajectories that cannot be followed with other approaches.

2.4 Mobile Platform Pose Error Compensation

Localization is a key problem in mobile robot navigation. This problem has gained much interest in recent years. Generally, the localization problem was solved by relative or absolute techniques [55]. The absolute positioning methods use features from the environment such as navigation beacons, landmarks, and GPS to determine the mobile platform location. The relative positioning methods use measurements from sensors that do not use any environment cues such as wheel encoders, accelerometers, and gyroscopes [56]. Relative positioning is simple, inexpensive and easy to achieve in realtime. However, it suffers from accumulating errors, without bound, over time and/or distance. These errors are due to navigation on irregular ground or smooth floor which causes the wheels to slip or slide. Localization using encoder readings can cause 20% to 25% error in pose estimation [57]. However, in the case of absolute position estimation, the error accumulation rate can be eliminated when the measurements are available due to the fact that the pose is externally determined. As a result, the error is not accumulated while the robot travels [57]. One example of the absolute position estimation is visual odometry.

Visual odometry, sometimes in literature referred to as ego-motion estimation, is a method in which the pose of a mobile robot is determined by using image information. In this method, computer vision algorithms [58–60] are used to estimate a 6 DoF pose of a moving camera frame by analyzing a sequence of video frames. It is primarily tracking visual features from one video frame to another and instantaneously determining the camera pose. By projecting the camera pose to the robot’s coordinate frame, the pose of the robot based in a global coordinate frame can be estimated. One of these vision algorithms

is Iterative Closest Point (ICP), which was introduced by Besl et al. [61, 62] in the 1990s. The ICP algorithm is a well-known algorithm for point set registration [63, 64]. It works by matching points between two overlapped range data images to estimate the sensor position change. Many variants of the ICP algorithm have been proposed that affect all aspects of the algorithm as stated by Rusinkiewicz et al. in [63]. They presented an optimized ICP algorithm that is able to align two range images in milliseconds if there is a good initial guess [64]. To improve the ICP outcomes, Hervier et al, in [65] proposed to fuse the ICP with measurements from other motion sensors by using the Kalman filter [66].

A recent trend exists whereby the relative and absolute localization procedures are combined to exploit the strengths of both techniques. One of the most widely used approaches for sensor fusion is the Kalman filter. In [67], Chen presented a review of contributions of Kalman filtering in solving mobile robot problems such as localization, mapping and navigation. The main focus of the survey was the role of Kalman filtering in robot vision. In literature, many authors have fused motion sensor measurements with vision sensor data for mobile robot localization using the Kalman filter [65, 68–70]. In these works, often different types of motion sensors (e.g. encoders, accelerometers, and gyroscopes) are combined with vision sensors.

CHAPTER 3

MOBILE MANIPULATOR KINEMATICS MODEL

3.1 Introduction

Manipulators or robotic arms can be mounted on various types of mobile platforms that differ by the driving mechanism. In this work, the mobile platform is a differentially driven type (nonholonomic mobile platform). The robotic arm is assumed to have n DoF and the mobile platform has 2 DoF (as will be explained later).

3.2 Terminology

Kinematics, as it is defined by Craig in [71], is the motion science that deals with the movement without considering the forces that caused it. The DoF of a robotic arm are simply the number of joints in a robotic arm. Throughout this dissertation, the subscript or superscript of the letters G, P, A and E refer to Ground (Global), mobile Platform, Arm base (or arm interchangeably) and End-effector coordinate frames, respectively, as shown in Figure 3.1.

3.3 Kinematic Modeling

In this section, the kinematic model for the n DoF robotic arm mounted on a differential driven mobile platform is presented. The mobile manipulator will have $(n + 2)$ DoF. Figure 3.1 shows a general representation of the redundant mobile manipulators with the coordinate frames of the end-effector “ E ”, the arm base frame “ A ”, and the mobile platform coordinate frame “ P ”.

The configuration of this mobile manipulator can be fully described by the following generalized coordinates:

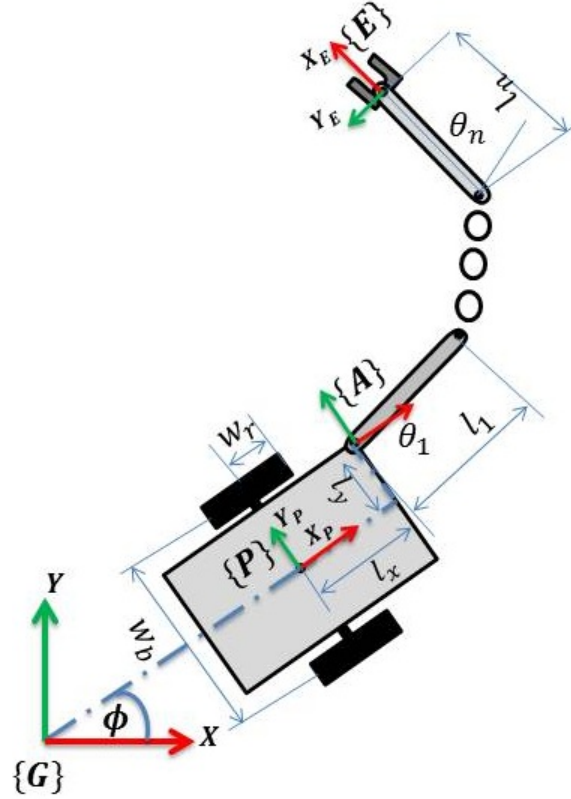


Figure 3.1: Differentially driven mobile manipulator consisting of a differentially driven mobile platform and an n DoF robotic arm.

$$q = [q_A \quad q_P]^T = [\theta_1 \quad \dots \quad \theta_n \quad x_P \quad y_P \quad \phi]^T \quad (3.1)$$

where $q_A = [\theta_1 \quad \dots \quad \theta_n]^T$ describes the configuration of the robotic arm and $q_P = [x_P \quad y_P \quad \phi]^T$ describes the configuration of the mobile platform, where x_P and y_P are the Cartesian position coordinates of the mobile platform along the global X and Y axis respectively, ϕ is the orientation angle of the mobile platform relative to the global frame G , and $\theta_1, \dots, \theta_n$ are the joint angles of the robotic arm. Due to nonholonomic constraints, the

kinematic model of the mobile platform can be defined as:

$$\begin{bmatrix} \dot{x}_P \\ \dot{y}_P \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{S} \\ \dot{\phi} \end{bmatrix} = M_{CP} \dot{q}_p \quad (3.2)$$

where \dot{S} and $\dot{\phi}$ are the linear and angular velocities of the mobile platform in polar coordinates, respectively, and M_{cp} is the Jacobian that relates Cartesian velocities to Polar velocities of the wheelchair motion. Therefore, the mobile platform velocity variables can be defined as $\dot{q}_p = [\dot{S} \ \dot{\phi}]^T$. Thus, the mobile manipulator velocity variables can be rewritten as follows:

$$\dot{q} = [\dot{q}_A \ \dot{q}_P]^T = [\dot{\theta}_1 \ \dots \ \dot{\theta}_n \ \dot{S} \ \dot{\phi}]^T \quad (3.3)$$

The Jacobian matrix that relates end-effector task vector \dot{r}_{GE} to the mobile manipulator joint velocity vector \dot{q} , can be represented as follows:

$$\dot{r}_{GE} = \begin{bmatrix} J_{GE_A} & J_{GE_P} \end{bmatrix} \begin{bmatrix} \dot{q}_A \\ \dot{q}_P \end{bmatrix} = J_{GE}(q) \dot{q} \quad (3.4)$$

where $\dot{r}_{GE} = [v_{GE} \ \omega_{GE}]^T \Rightarrow v_{GE} = [\dot{x}_{GE} \ \dot{y}_{GE} \ \dot{z}_{GE} \ \omega_{x_{GE}} \ \omega_{y_{GE}} \ \omega_{z_{GE}}]^T \in R^m$ represents the desired Cartesian velocity vector of the end-effector, $\dot{q} \in R^{n+2}$ is the joint velocity output vector, and $J_{GE}(q) \in R^{m \times (n+2)}$ is the Jacobian that relates them. The J_{GE_A} is the Jacobian matrix that relates the \dot{r}_{GE} to the robotic arm joint velocity vector \dot{q}_A and the J_{GE_P} is the Jacobian matrix that relates the \dot{r}_{GE} to the mobile platform joint velocity vector \dot{q}_P . In this work, only the kinematically redundant mobile manipulator will be considered. Kinematic redundancy occurs when the DoF of the mobile manipulator (dimension of velocity inputs vector) exceeds the dimension of the task space vector, i.e., $n + 2 > m$.

The most commonly used redundancy resolution algorithms for mobile manipulators provide the ability to design the controller in two spaces: operational space and null space. In the operational space, a tracking controller is designed to eliminate the error between the end-effector's actual pose and the end-effector's desired trajectory pose. In the null space, an adjustment process is designed to optimize some criteria without changing the state of the end-effector. This approach demonstrates a direct control of the end-effector pose. However, the platform pose will be indirectly controlled using some optimization criteria in the null space. Tracking the end-effector trajectory can be taken care of using Equation (3.4).

3.4 Dual-Trajectory Kinematic Representation

As stated by Luca in [39], the Jacobian $J(q)$ can be extended by adding additional constraints to the task vector \dot{r} . To force the mobile platform to follow a prespecified track, additional constraints should be added to the kinematic model. The mobile platform trajectory can be simply added to the task vector by adding the 2 DoF of the mobile platform, which are the platform's translation and rotation velocities (\dot{S} and $\dot{\phi}$), to vector \dot{r} . The velocity relation of the additional constraints can be represented as follows:

$$\dot{r}_{GP} = \begin{bmatrix} J_{GP_A} & J_{GP_P} \end{bmatrix} \begin{bmatrix} \dot{q}_A \\ \dot{q}_P \end{bmatrix} = J_{GP}(q_P) \dot{q} \quad (3.5)$$

where $\dot{r}_{GP} \in R^2$ is the desired Cartesian velocity vector of the platform, $\dot{q} \in R^{n+2}$ is the velocity output vector, and $J_{GP}(q) \in R^{2 \times (n+2)}$ is the Jacobian that relates them. J_{GP_A} is the Jacobian that relates the arm joint velocities \dot{q}_A to the mobile platform desired task vector velocity \dot{r}_{GP} . The arm joint velocities do not affect the mobile platform velocities. Therefore, the $J_{GP_A} = [0]^{2 \times n}$. Using Extended Jacobian [39], Equation (3.4) can be modified as follows:

$$\dot{r}_{EP} = \begin{bmatrix} \dot{r}_{GE} \\ \dots \\ \dot{r}_{GP} \end{bmatrix} = \begin{bmatrix} J_{GE} \\ \dots \\ J_{GP} \end{bmatrix} \begin{bmatrix} \dot{q}_A \\ \dot{q}_P \end{bmatrix} = \begin{bmatrix} J_{GE_A} & \vdots & J_{GE_P} \\ \dots & \dots & \dots \\ [0]^{2 \times n} & \vdots & J_{GP_P} \end{bmatrix} \begin{bmatrix} \dot{q}_A \\ \dot{q}_P \end{bmatrix} = J_{EP}(q) \dot{q} \quad (3.6)$$

where $\dot{r}_{EP} \in R^{m+2}$ is the desired Cartesian velocity vector of the mobile manipulator (end-effector and Platform) and $J_{EP}(q) \in R^{(m+2) \times (n+2)}$ is the mobile manipulator Jacobian matrix for the dual-trajectory tracking control. For simplicity, Equation (3.6) is rewritten as follows:

$$\dot{r}_{sys} = J_{sys}(q) \dot{q}_{sys} \quad (3.7)$$

where the abbreviation *sys* refers to the mobile manipulator system.

In a baseline case of tracking the mobile platform trajectory, $\dot{r}_{GP} = \dot{q}_P = [\dot{S} \ \dot{\phi}]^T$. The arm joint angle velocities \dot{q}_A have no effect on the mobile platform velocities \dot{r}_{GP} . And J_{GP_P} relates the mobile platform velocities along the prespecified track \dot{r}_{GP} to the mobile platform linear and angular velocities (\dot{q}_P). Therefore, J_{GP_P} is an identity matrix $[I]$ of a dimension 2×2 . Thus, Equation (3.6) can be rewritten as follows:

$$\dot{r}_{EP} = \begin{bmatrix} \dot{r}_{GE} \\ \dots \\ \dot{S} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} J_{GE} \\ \dots \\ J_{GP} \end{bmatrix} \begin{bmatrix} \dot{q}_A \\ \dot{q}_P \end{bmatrix} = J_{EP}(q) \dot{q} \quad (3.8)$$

where

$$J_{GP} = \begin{bmatrix} [0]^{2 \times n} & [I]^{2 \times 2} \end{bmatrix} = \begin{bmatrix} 0 & \dots & 0_n & 1 & 0 \\ 0 & \dots & 0_n & 0 & 1 \end{bmatrix} \quad (3.9)$$

It can be noticed that this Jacobian is a one to one mapping of the mobile platform joint velocity. In order to use Equation (3.6), the mobile manipulator trajectory has to be fully predefined. This means that both the end-effector and mobile platform locations are known

as waypoints. From these waypoints, the Cartesian velocities of the end-effector and polar velocities (\dot{S} and $\dot{\phi}$) of the mobile platform can be determined. This case will be referred to throughout this dissertation as the baseline case of tracking mobile manipulator trajectory. Using the same notation as Equation (3.7), the baseline case general Equation (3.8) can be rewritten as follows:

$$\dot{r}_{sys} = \begin{bmatrix} \dot{r}_{GE} \\ \cdots \\ \dot{S} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} J_{GE} \\ \cdots \\ J_{GP} \end{bmatrix} \begin{bmatrix} \dot{q}_A \\ \dot{q}_P \end{bmatrix} = J_{sys}(q) \dot{q}_{sys} \quad (3.10)$$

3.5 Mobile Manipulator Jacobi

In this section, a general introduction of how the Jacobian of the mobile manipulator is constructed for both the manipulator and the mobile platform will be presented. Standard convention used by Craig [71] will be used.

3.5.1 Manipulator Jacobian

There are many approaches that can be used to determine the Jacobian matrix. Differentiation approach is one of them [71]. As stated before, the robotic arm has n joints that are usually revolute, prismatic or a combination of both. In forward kinematics, the end-effector's Cartesian position and orientation can be computed when the joint angles are known. The end-effector task vector relative to arm base frame "A" can be presented as follows:

$$r_{AE} = f(\theta_1, \theta_2, \cdots, \theta_n) \quad (3.11)$$

where r_{AE} is a 6×1 vector representing the end-effector position and end-effector orientation with respect to the arm base frame A . These relations can be found using forward kinematics.

By differentiating Equation (3.11) with respect to arm joint angles, the following is obtained:

$$\begin{aligned}
\delta x_{AE} &= \frac{\partial f_1}{\partial \theta_1} \delta \theta_1 + \frac{\partial f_1}{\partial \theta_2} \delta \theta_2 + \cdots + \frac{\partial f_1}{\partial \theta_n} \delta \theta_n, \\
\delta y_{AE} &= \frac{\partial f_2}{\partial \theta_1} \delta \theta_1 + \frac{\partial f_2}{\partial \theta_2} \delta \theta_2 + \cdots + \frac{\partial f_2}{\partial \theta_n} \delta \theta_n, \\
\delta z_{AE} &= \frac{\partial f_3}{\partial \theta_1} \delta \theta_1 + \frac{\partial f_3}{\partial \theta_2} \delta \theta_2 + \cdots + \frac{\partial f_3}{\partial \theta_n} \delta \theta_n, \\
\delta \omega_{X_{AE}} &= \frac{\partial f_4}{\partial \theta_1} \delta \theta_1 + \frac{\partial f_4}{\partial \theta_2} \delta \theta_2 + \cdots + \frac{\partial f_4}{\partial \theta_n} \delta \theta_n, \\
\delta \omega_{Y_{AE}} &= \frac{\partial f_5}{\partial \theta_1} \delta \theta_1 + \frac{\partial f_5}{\partial \theta_2} \delta \theta_2 + \cdots + \frac{\partial f_5}{\partial \theta_n} \delta \theta_n, \\
\delta \omega_{Z_{AE}} &= \frac{\partial f_6}{\partial \theta_1} \delta \theta_1 + \frac{\partial f_6}{\partial \theta_2} \delta \theta_2 + \cdots + \frac{\partial f_6}{\partial \theta_n} \delta \theta_n
\end{aligned} \tag{3.12}$$

These relations can be rewritten as follows:

$$\delta r_{AE} = \frac{\partial f}{\partial \theta} \delta \theta \tag{3.13}$$

Velocities can be computed by dividing both sides of Equation (3.13) by time increment δt .

That gives:

$$\frac{\delta r_{AE}}{\delta t} = \frac{\partial f}{\partial \theta} * \frac{\delta \theta}{\delta t} \Rightarrow \dot{r}_{AE} = \frac{\partial f}{\partial \theta} * \dot{\theta} = J_{AE}(\theta) \dot{\theta} \tag{3.14}$$

where $J_{AE}(\theta)$ is the Jacobian matrix that relates the end-effector Cartesian velocities ($[\dot{x}_{AE} \ \dot{y}_{AE} \ \dot{z}_{AE} \ \omega_{x_{AE}} \ \omega_{y_{AE}} \ \omega_{z_{AE}}]^T$) to the joint angular velocity vector $\dot{\theta}$ with respect to the arm base frame A . It is worth mentioning that Equation (3.14) is applicable when the mobile platform is stationary and only the arm motion is considered. Equation (3.14) can be rewritten relative to global frame as follows:

$$\dot{r}_{GE_A} = J_{GE_A} * \dot{q}_A \tag{3.15}$$

where J_{GE_A} is the Jacobian that relates the end-effector task velocity vector relative to global frame “ G ” to the arm joint angles when the platform is stationary. J_{GE_A} can be calculated

as follows:

$$J_{GE_A}(\theta) = \left[\begin{array}{c|c} \begin{matrix} {}^G_A R \\ 0 \end{matrix} & \begin{matrix} 0 \\ {}^G_A R \end{matrix} \\ \hline & \end{array} \right]_{6 \times 6} J_{AE}(\theta) \quad (3.16)$$

where ${}^G_A R$ is the (3×3) rotation matrix of the arm base frame “A” relative to the global frame “G”.

3.5.2 Mobile Platform Jacobian

The global pose of mobile platform $([x_{GP_i}, y_{GP_i}, \phi_{GP_i}]^T)$ can be found by measuring the angular displacement of the right wheel θ_{r_i} and the left wheel θ_{l_i} , where x_{GP_i} and y_{GP_i} are the X and Y global coordinates of the mobile platform respectively, and ϕ_{GP_i} is the mobile platform orientation angle. Throughout this section, the subscript i means the i^{th} instance in the mobile platform motion. These two angular displacements are computed using the encoders’ readings from both wheels. The distances traveled by the left and right wheels are:

$$L_i = w_r \theta_{r_i} \quad (3.17)$$

$$R_i = w_r \theta_{l_i} \quad (3.18)$$

respectively, where w_r is the wheel radius in meters. The pose of the mobile platform relative to a global frame $([x_{GP_i}, y_{GP_i}, \phi_{GP_i}]^T)$, as shown in Figure 3.2, can be computed using Equation (3.19)

$$\begin{bmatrix} x_{GP_i} \\ y_{GP_i} \\ \phi_{GP_i} \end{bmatrix} = \begin{bmatrix} x_{GP_{i-1}} \\ y_{GP_{i-1}} \\ \phi_{GP_{i-1}} \end{bmatrix} + \begin{bmatrix} r_i \left[\sin \phi_{GP_{i-1}} - \sin \left(\phi_{GP_{i-1}} + \frac{R_i - L_i}{w_b} \right) \right] \\ r_i \left[\cos \left(\phi_{GP_{i-1}} + \frac{R_i - L_i}{w_b} \right) - \cos \phi_{GP_{i-1}} \right] \\ \frac{R_i - L_i}{w_b} \end{bmatrix} \quad (3.19)$$

the symbol r_i represents the instantaneous radius of rotation, where $r_i = \frac{w_b}{2} \left(\frac{L_i + R_i}{L_i - R_i} \right)$ and w_b is the wheel base. This model is similar to the model represented in [72]. This repre-

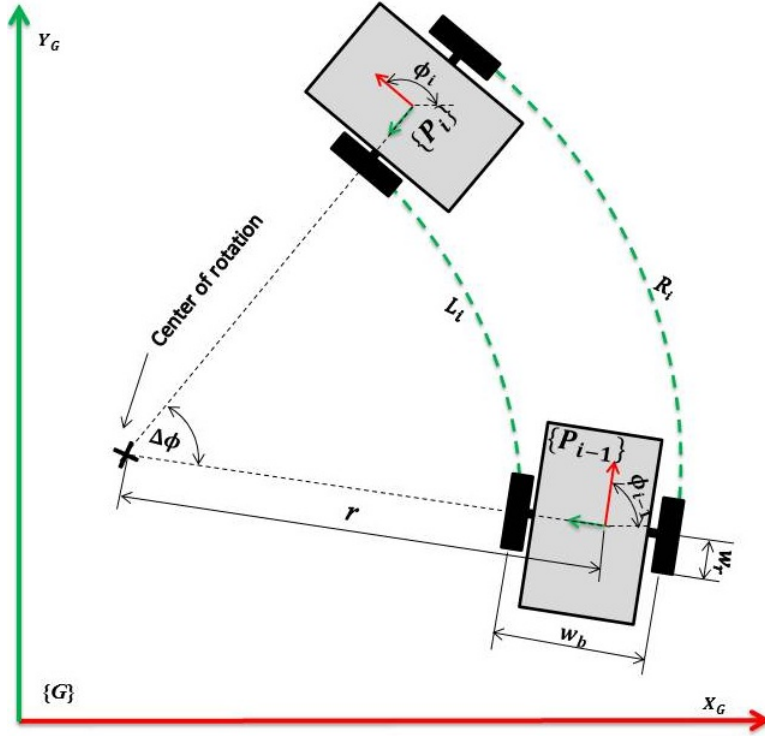


Figure 3.2: Two consecutive mobile platform frames.

sents the forward kinematics for the mobile platform. The following is a derivation of the Jacobian matrix of the mobile platform using the following mobile base dimensions for this configuration (refer to Figure 3.1):

1. Wheel base w_b is the distance between the center of the two driving wheels along the driving axle.
2. Wheel radius w_r is the wheel radius of the mobile platform's driving wheel.
3. Distance l_x is the offset distance of the arm base frame A to the mobile platform frame P along the x axis of the mobile platform.
4. Distance l_y is the offset distance of the arm base frame A to the mobile platform frame P along the y axis of the mobile platform.

5. Distance l_z is the offset distance of the arm base frame A to the mobile platform frame P along the z axis of the mobile platform.

As previously mentioned, the state variables of the mobile platform are the linear and angular motions (S and ϕ). The linear translation of the mobile platform can be computed as follows (refer to Figure 3.2):

$$S = \frac{R + L}{2} = \frac{w_r \theta_r + w_l \theta_l}{2} = \frac{w_r}{2} (\theta_r + \theta_l) \quad (3.20)$$

similarly, the angular motion of the mobile platform can be computed as follows:

$$\phi = \frac{R - L}{w_b} = \frac{w_r \theta_r - w_l \theta_l}{w_b} = \frac{w_r}{w_b} (\theta_r - \theta_l) \quad (3.21)$$

By differentiating Equations (3.20) and (3.21) with respect to time, the resultant equations:

$$\dot{S} = \frac{w_r}{2} (\dot{\theta}_r + \dot{\theta}_l) \quad (3.22)$$

$$\dot{\phi} = \frac{w_r}{w_b} (\dot{\theta}_r - \dot{\theta}_l) \quad (3.23)$$

Combining Equations (3.22) and (3.23) gives the equation:

$$\begin{bmatrix} \dot{S} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{w_r}{2} & \frac{w_r}{2} \\ \frac{w_r}{w_b} & -\frac{w_r}{w_b} \end{bmatrix} \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} \quad (3.24)$$

Equation (3.24) relates the wheels angular velocities to the mobile platform linear and angular velocities. The relation of the mobile platform Cartesian velocities to the mobile platform's linear and angular volatilities is presented in Equation (3.2). Equation (3.2) can be modified using the same notation as in Figure 3.3 as follows:

$$\begin{bmatrix} \dot{x}_P \\ \dot{y}_P \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} {}^G_P \dot{X} \\ {}^G_P \dot{Y} \\ \dot{\phi}_P \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{S} \\ \dot{\phi} \end{bmatrix} \Rightarrow V_P = J_{GP} * \dot{q}_P \quad (3.2 \text{ revisited})$$

where J_{GP} is the Jacobian that relates the mobile platform's Cartesian velocities and the

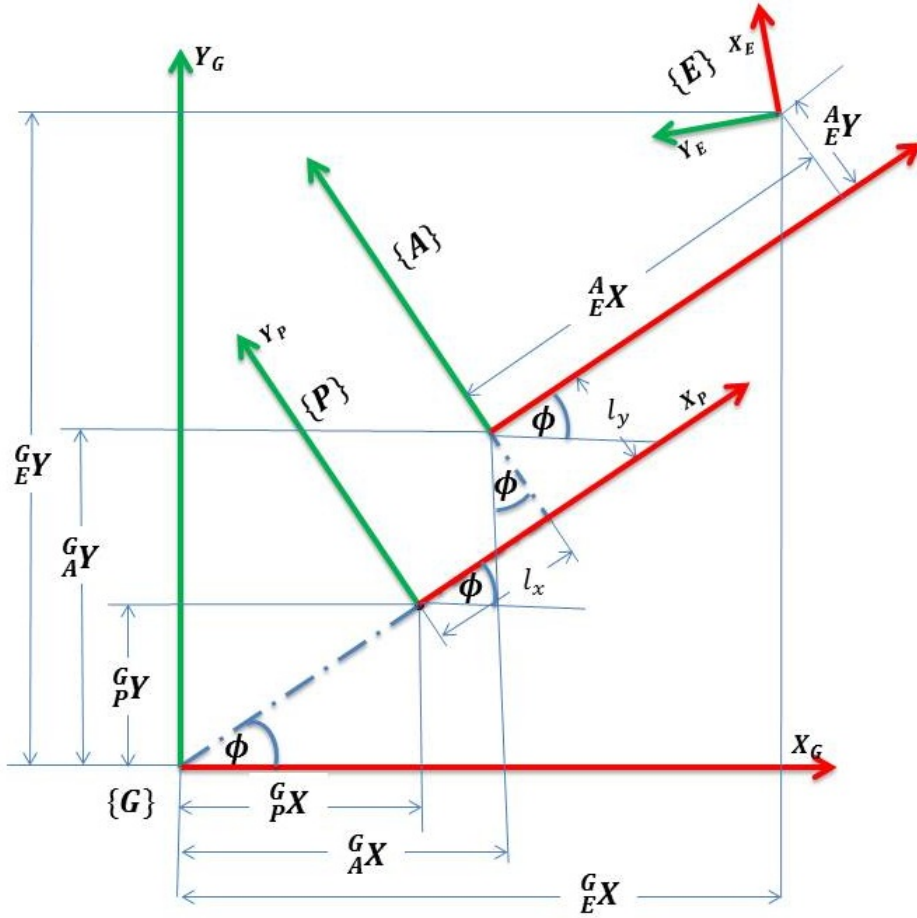


Figure 3.3: 2D map for the mobile manipulator's main coordinate frames.

mobile platform's linear and angular velocities. Using Equations (3.24) and (3.2 revisited), the relation between the mobile platform's Cartesian velocities and the mobile platform wheels' angular velocities can be expressed as follows:

$$\begin{aligned}
\begin{bmatrix} \dot{x}_P \\ \dot{y}_P \\ \dot{\phi} \end{bmatrix} &= \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{w_r}{2} & \frac{w_r}{2} \\ \frac{w_r}{w_b} & -\frac{w_r}{w_b} \end{bmatrix} \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} \\
&= \begin{bmatrix} \frac{w_r}{2} \cos\phi & \frac{w_r}{2} \cos\phi \\ \frac{w_r}{2} \sin\phi & \frac{w_r}{2} \sin\phi \\ \frac{w_r}{w_b} & -\frac{w_r}{w_b} \end{bmatrix} \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} \Rightarrow V_P = J_{GW_v} * W_v
\end{aligned} \tag{3.25}$$

where W_v is the mobile platform wheel velocities vector. The relation between the velocities of the mobile platform frame P and the velocities of the arm base frame A can be determined by finding the position of frame A relative to frame G . For simplicity, it can be assumed that the arm base frame A has the same orientation as the mobile platform frame P , as shown in Figure 3.3. Therefore, the transformation matrix of the arm base frame A relative to the mobile platform frame P can be represented as follows:

$${}^P_A T = \begin{bmatrix} 1 & 0 & 0 & l_x \\ 0 & 1 & 0 & l_y \\ 0 & 0 & 1 & l_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.26}$$

The position of the arm base frame A relative to the global frame G can be expressed as follows:

$$\begin{aligned}
{}^G_A X &= {}^G_P X + l_x \cos\phi_P - l_y \sin\phi_P \\
{}^G_A Y &= {}^G_P Y + l_x \sin\phi_P + l_y \cos\phi_P \\
\phi_A &= \phi_P
\end{aligned} \tag{3.27}$$

where ${}^G_P X$ and ${}^G_P Y$ are the X-Y coordinates of the origin of the mobile platform frame P relative to the global frame G , and ϕ_P is the orientation angle of the mobile platform. Differentiating Equation (3.27), with respect to time, yields velocities of frame A relative to frame G . These relations can be demonstrated as follows:

$$\begin{aligned}
{}^G_A\dot{X} &= {}^G_P\dot{X} - l_x \sin\phi_P \dot{\phi}_P - l_y \cos\phi_P \dot{\phi}_P \\
{}^G_A\dot{Y} &= {}^G_P\dot{Y} + l_x \cos\phi_P \dot{\phi}_P - l_y \sin\phi_P \dot{\phi}_P \\
\dot{\phi}_A &= \dot{\phi}_P
\end{aligned} \tag{3.28}$$

Rewriting Equation (3.28) into a matrix form yields:

$$\begin{bmatrix} {}^G_A\dot{X} \\ {}^G_A\dot{Y} \\ \dot{\phi}_A \end{bmatrix} = \begin{bmatrix} \dot{X}_A \\ \dot{Y}_A \\ \dot{\phi}_A \end{bmatrix} = \begin{bmatrix} 1 & 0 & -(l_x \sin\phi_P + l_y \cos\phi_P) \\ 0 & 1 & l_x \cos\phi_P - l_y \sin\phi_P \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^G_P\dot{X} \\ {}^G_P\dot{Y} \\ \dot{\phi}_P \end{bmatrix} \Rightarrow V_A = J_{PA} * V_P \tag{3.29}$$

where J_{PA} is the Jacobian that relates the arm base Cartesian velocities to the mobile platform Cartesian velocities and $\phi_P = \phi$. To find the general Jacobian that relates the arm base Cartesian velocities to the mobile platform linear and angular velocities, Equations (3.2 revisited) and (3.29) can be combined as follows:

$$V_A = J_{PA} * J_{GP} * \dot{q}_P = J_{GA} * \dot{q}_P \tag{3.30}$$

Rewriting Equation (3.30) in a matrix form yields:

$$\begin{aligned}
\begin{bmatrix} \dot{X}_A \\ \dot{Y}_A \\ \dot{\phi}_A \end{bmatrix} &= \begin{bmatrix} 1 & 0 & -(l_x \sin\phi + l_y \cos\phi) \\ 0 & 1 & l_x \cos\phi - l_y \sin\phi \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{S} \\ \dot{\phi} \end{bmatrix} \\
&= \begin{bmatrix} \cos\phi & -(l_x \sin\phi + l_y \cos\phi) \\ \sin\phi & l_x \cos\phi - l_y \sin\phi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{S} \\ \dot{\phi} \end{bmatrix}
\end{aligned} \tag{3.31}$$

The velocity relation represented in Equation (3.31) gives the Jacobian of the non-holonomic mobile platform that relates the mobile platform's linear and angular velocities to the three

Cartesian velocities of the arm base frame “A”. This relation should be modified to accommodate the six Cartesian velocities of the end-effector $[\dot{x}_{GA} \ \dot{y}_{GA} \ \dot{z}_{GA} \ \omega_{x_{GA}} \ \omega_{y_{GA}} \ \omega_{z_{GA}}]$.

The velocities relation represented in Equation (3.31) can be modified to include all the six Cartesian velocities as follows:

$$\dot{r}_{GA} = J_C * J_{GA} * \dot{q}_P = J_{GA} * \dot{q}_P \quad (3.32)$$

Equation (3.32) can be rewritten as follows:

$$\begin{aligned} \dot{r}_{GA} &= \begin{bmatrix} \dot{x}_{GA} \\ \dot{y}_{GA} \\ \dot{z}_{GA} \\ \omega_{x_{GA}} \\ \omega_{y_{GA}} \\ \omega_{z_{GA}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\phi & -(l_x \sin\phi + l_y \cos\phi) \\ \sin\phi & l_x \cos\phi - l_y \sin\phi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{S} \\ \dot{\phi} \end{bmatrix} \\ &= \begin{bmatrix} \cos\phi & -(l_x \sin\phi + l_y \cos\phi) \\ \sin\phi & l_x \cos\phi - l_y \sin\phi \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{S} \\ \dot{\phi} \end{bmatrix} \end{aligned} \quad (3.33)$$

The effect of the mobile platform’s motion at frame A on the end-effector’s frame E (without arm joints’ motion) can be determined by defining the velocity task vector of frame E relative to frame G . As presenting in Figure 3.3, the velocity task vector of frame E relative to frame G , when only platform motion occurs without arm joints’ motion, can be expressed

as follows:

$$\begin{aligned}
{}^G\dot{X}_E &= {}^G\dot{X}_A - {}^A\dot{X}_E \sin\phi \dot{\phi} - {}^A\dot{Y}_E \cos\phi \dot{\phi} \\
{}^G\dot{Y}_E &= {}^G\dot{Y}_A + {}^A\dot{X}_E \cos\phi \dot{\phi} - {}^A\dot{Y}_E \sin\phi \dot{\phi} \\
{}^G\dot{Z}_E &= {}^G\dot{Z}_A \\
\omega_{x_{GE}} &= \omega_{x_{GA}} \\
\omega_{y_{GE}} &= \omega_{y_{GA}} \\
\omega_{z_{GE}} &= \omega_{z_{GA}}
\end{aligned} \tag{3.34}$$

where The ${}^A\dot{X}_E$ and ${}^A\dot{Y}_E$ are the X and Y coordinates of the origin of the end-effector frame E relative to the arm base frame A , respectively. The ${}^G\dot{X}_A$, ${}^G\dot{Y}_A$ and ${}^G\dot{Z}_A$ are the Cartesian velocity of the end-effector frame E relative to the arm base frame A and the ${}^A\dot{X}_E$, ${}^A\dot{Y}_E$ and ${}^A\dot{Z}_E$ are the Cartesian velocity of the end-effector frame E relative to the global frame G . The $\omega_{x_{GE}}$, $\omega_{y_{GE}}$ and $\omega_{z_{GE}}$ are the rotation angles of the end-effector frame E relative to X, Y and Z axes of the global frame G , respectively. Moving on a horizontal plane, the angular velocities $\omega_{x_{GA}}$ and $\omega_{y_{GA}}$ equal to zero degree. The $\omega_{z_{GA}}$ is the same as the angular velocity of mobile platform $\dot{\phi}$. That is because there is no arm joint motion. Writing in matrix form, Equation 3.34 can be represented as follows:

$$\begin{aligned}
\begin{bmatrix} {}^G\dot{X}_E \\ {}^G\dot{Y}_E \\ {}^G\dot{Z}_E \\ \omega_{x_{GE}} \\ \omega_{y_{GE}} \\ \omega_{z_{GE}} \end{bmatrix} &= \begin{bmatrix} \dot{X}_E \\ \dot{Y}_E \\ \dot{Z}_E \\ \omega_{x_{GE}} \\ \omega_{y_{GE}} \\ \omega_{z_{GE}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -({}^A\dot{X}_E \sin\phi + {}^A\dot{Y}_E \cos\phi) \\ 0 & 1 & 0 & 0 & 0 & {}^A\dot{X}_E \cos\phi - {}^A\dot{Y}_E \sin\phi \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^G\dot{X}_A \\ {}^G\dot{Y}_A \\ {}^G\dot{Z}_A \\ \omega_{x_{GA}} \\ \omega_{y_{GA}} \\ \omega_{z_{GA}} \end{bmatrix} \\
\Rightarrow \dot{r}_{GE_P} &= J_{GE_P} * \dot{r}_{GA}
\end{aligned} \tag{3.35}$$

where J_{AE_P} is the Jacobian that relates the Cartesian velocities of the end-effector frame E to the Cartesian velocities of the arm base frame A , due to the mobile platform motion only. By substituting Equation (3.33) into Equation (3.35) we get:

$$\begin{aligned} \dot{r}_{GE_P} &= J_{GE_P} * J_{GA} * \dot{q}_P \\ &= \begin{bmatrix} \cos\phi & -(l_x \sin\phi + l_y \cos\phi) - ({}^A_E X \sin\phi + {}^A_E Y \cos\phi) \\ \sin\phi & l_x \cos\phi - l_y \sin\phi + {}^A_E X \cos\phi - {}^A_E Y \sin\phi \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{S} \\ \dot{\phi} \end{bmatrix} \end{aligned} \quad (3.36)$$

As a summary, the end-effector Cartesian velocities relative to global frame “ G ” are determining using the two cases: Case 1 when the mobile platform is stationary and only the arm motion is considered, (the subscript “ A ” was used as in Equation (3.15)). Case 2 when the arm is stationary and the mobile platform motion equation was considered, (the subscript “ P ” was used as in Equation (3.35)).

3.6 Mobile Manipulator Combined Jacobian

A single, combined, Jacobian can be derived for the robotic arm and mobile base. Substituting the Jacobian in Equations (3.15, and 3.36) into Equation (3.4) gives:

$$\begin{aligned} \dot{r}_{GE} &= \dot{r}_{GE_A} + \dot{r}_{GE_P} \\ &= J_{GE_A} * \dot{q}_A + J_{GE_P} * \dot{q}_P \\ &= J_{GE_A} * \dot{q}_A + J_{GE_P} * J_{GA} * \dot{q}_P \end{aligned} \quad (3.37)$$

Rewriting Equation (3.37) into a matrix form yields:

$$\begin{aligned}\dot{r}_{GE} &= \begin{bmatrix} J_{GE_A} & J_{GE_P} * J_{GA} \end{bmatrix} \begin{bmatrix} \dot{q}_A \\ \dot{q}_P \end{bmatrix} \\ &= J_{GE} * \dot{q}\end{aligned}\tag{3.38}$$

where J_{GE} is the combined Jacobian that combines the mobility of the mobile platform with the manipulation of the robotic arm and relates the six Cartesian velocities of the end-effector to the joint angles of the arm and the mobile platform.

The mobile platform has 2 DoF in polar coordinates, which are the translation S and the rotation ϕ . The Jacobian, in Equation (3.38), relates joint angles rates to Cartesian velocities of the end-effector when both the arm and mobile platform are active. For more stability of the control solution, the linear velocity of the mobile platform \dot{S} should be converted to angular velocity as follows [6]:

$$\dot{\theta}_S = \frac{\dot{S}}{w_r}\tag{3.39}$$

The final step is to get the dual-trajectory final kinematic model for the baseline case which is represented by Equations (3.6), (3.8), and (3.10). This case will be used as foundation for developing the dual-trajectory controllers. Revisiting Equation (3.10):

$$\dot{r}_{sys} = \begin{bmatrix} \dot{r}_{GE} \\ \dots \\ \dot{r}_{GP} \end{bmatrix} = \begin{bmatrix} \dot{r}_{GE} \\ \dots \\ \dot{S} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} J_{GE} \\ \dots \\ J_{GP} \end{bmatrix} \begin{bmatrix} \dot{q}_A \\ \dot{q}_P \end{bmatrix} = J_{sys}(q) \dot{q}_{sys}\tag{3.10 revisited}$$

where $J_{GE} = \begin{bmatrix} J_{GE_A} & J_{GE_P} \end{bmatrix}$ is the Jacobian that relates the end-effector task velocity vector relative to the global frame “G” to the mobile manipulator joint angle output vector $\dot{q}_{sys} = \begin{bmatrix} \dot{q}_A & \dot{q}_P \end{bmatrix}^T$. J_{GE_A} is the Jacobian that relates the end-effector task velocity vector relative to the global frame “G” to the arm joint angle velocity vector $q_A = \begin{bmatrix} \theta_1 & \dots & \theta_n \end{bmatrix}^T$ when only the arm motion is considered. J_{GE_P} is the Jacobian that relates the end-effector

task velocity vector relative to the global frame “ G ” to the mobile platform joint angle velocity vector $\dot{q}_P = [\dot{S} \ \dot{\phi}]^T$ when only the mobile platform motion is considered.

In this work, the most modification will be done on the part of the mobile platform trajectory $\dot{r}_{GP} = J_{GP} [\dot{q}_A \ \dot{q}_P]^T$. For the baseline Jacobian, the mobile platform trajectory

can be represented as:
$$\begin{bmatrix} \dot{S} \\ \dot{\phi} \end{bmatrix} = J_{GP} \begin{bmatrix} \dot{S} \\ \dot{\phi} \end{bmatrix}.$$
 Therefore, $J_{GP} = \begin{bmatrix} [0]^{2 \times n} & [I]^{2 \times 2} \end{bmatrix} =$

$$\begin{bmatrix} 0 & \cdots & 0_n & 1 & 0 \\ 0 & \cdots & 0_n & 0 & 1 \end{bmatrix}$$

3.7 Summary

In this chapter, a kinematic model of a mobile manipulator was presented. The mobile manipulator has $(n + 2)$ DoF in which the robotic arm has n DoF and the mobile platform has 2 DoF. All the needed Jacobi were determined to combined the two subsystems (robotic arm and mobile platform) into one Jacobian. This Jacobian relates the end-effector’s six Cartesian velocities to the system’s $n + 2$ joint angles. A baseline case for following the end-effector and mobile platform trajectories simultaneously was presented.

CHAPTER 4

MOBILE MANIPULATOR KINEMATIC CONTROL

4.1 Introduction

In the previous chapter, the general kinematic model of a mobile manipulator with a $n + 2$ DoF system was presented. Equation (3.4), repeated here, relates the m components of the end-effector Cartesian space velocities vector \dot{r}_{GE} to $n + 2$ joint velocities vector \dot{q} as follows:

$$\dot{r}_{GE} = J_{GE} \dot{q} \quad (4.1)$$

where J_{GE} is a $m \times (n + 2)$ dimensional Jacobian matrix. In this Chapter, the inverse kinematics problem will be addressed to find solutions to Equation (4.1). In addition, several methods of optimization will be presented to meet certain control objectives, such as joint limit avoidance and singularity avoidance.

4.2 Kinematic Control

Usually, combining the mobility of a mobile base with the manipulation of a robotic arm results in a redundant robot systems if the total number of joints exceeds six joints. Redundancy means that the number of columns is more than the number of rows in the Jacobian matrix. For instance, in Equation (4.1), the Jacobian matrix J_{GE} has a dimensions of $(m \times (n + 2))$ where m is the dimension of the mobile manipulator space task vector, and $n + 2$ is the dimension of the joint angle vector (DoF). The degree of redundancy (DoR)

is equal to $(n + 2) - m$. For this dissertation only the kinematically redundant mobile manipulator will be considered; $n + 2 > m$.

The most commonly used redundancy resolution algorithms for mobile manipulators provides the ability to design a controller in two spaces: operational space and null space. In the operational space, a tracking controller is designed to eliminate the error between the end-effector's actual pose and the end-effector's desired trajectory. In the null space, an adjustment process is designed to optimize some criteria without changing the state of the end-effector.

Inverse kinematic is used to find joint angles or joint velocities that corresponded to a given end-effector Cartesian positions and orientations or Cartesian velocities. In order to read the joint angle while the mobile manipulator (arm and mobile base) is moving and send it to the motion controller as a joint feedback, encoders are installed for each joint motor.

4.2.1 Pseudo-Inverse of the Jacobian

When the Jacobian matrix is a square matrix (the dimension of the end-effector space task vector is equal to the dimension of the joint angle vector) and has full rank, the joint velocities required to achieve the desired end-effector motion will be unique and can be determined as follows:

$$\dot{q} = J_{GE}^{-1} \cdot \dot{r}_{GE} \quad (4.2)$$

As stated previously, only the redundant mobile manipulator is considered and thus, the Jacobian matrix is not square. This means that the Jacobian matrix cannot be inverted with the traditional methods. One of the most used methods to invert a none-square matrix used in redundant manipulator controls is the pseudo-inverse, which can be presented as follows [9]:

$$J^\# = J^T \cdot (J \cdot J^T)^{-1} \quad (4.3)$$

where J^T is the transpose of the Jacobian matrix, and $J^\#$ is the pseudo-inverse of the J matrix. Using Equation (4.3) to solve Equation (4.1) will result in:

$$\dot{q} = J_{GE}^\# \cdot \dot{r}_{GE} \quad (4.4)$$

where $J_{GE}^\#$ is the pseudo-inverse of the Jacobian matrix J_{GE} that can be calculated as follows:

$$J_{GE}^\# = J_{GE}^T \cdot (J_{GE} \cdot J_{GE}^T)^{-1} \quad (4.5)$$

The joint velocity vector \dot{q} , in Equation (4.4), is the LN solution of Equation (4.1) [73], which provides \dot{q}_{sys} with minimum Euclidean norm ($min\|\dot{q}\|$). When an exact solution does not exist (the Jacobian matrix is not full row rank), Equation (4.4) produces a Least-Square (LS) solution that minimizes the Euclidean norm of errors while keeping the joint velocity vector norm to a minimum values. This can be represented as follows:

$$min\|\dot{r}_{GE} - J_{GE} \cdot \dot{q}\| \text{ while keeping } min\|\dot{q}\| \quad (4.6)$$

4.2.2 Singularity-Robust Inverse (SR-Inverse)

Singular configurations result in high joint velocities that cause instability of the system while trying to follow a desired trajectory. Therefore, inverting the Jacobian while avoiding singular configurations is likely to produce more stable results. The Singularity-Robust Inverse (SR-Inverse) method [8] is used to alleviate the kinematic singularity problem. This method has been known for reducing the joint velocities at or near a singular configurations by allowing more errors in the task vector. In addition, the SR-Inverse method enables the use of redundancy resolution approaches to execute different subtasks, during which, singularities can be controlled at the Jacobian inversion level. Manipulability measure [9] is used as an indicator of how far the current mobile manipulator configuration is from a

singularity. This measure is defined as follows:

$$w = \sqrt{(\det(JJ^T))} \quad (4.7)$$

For further details, see Section (4.2.5). This SR-Inverse method [8] is defined as:

$$J^\# = J^T (JJ^T + k_{SR} I_m)^{-1} \quad (4.8)$$

where I_s is an $s \times s$ identity matrix, and k_{SR} is a scale factor. Choosing the scale factor k_{SR} is critical to minimize the errors. Because the purpose of using this factor is to compromise the solution near singularities, this scale factor is updated according to the whole system manipulability measure at every time step as follows:

$$k_{SR} = \begin{cases} k_0 \left(1 - \frac{w}{w_0}\right)^2 & \text{for } w < w_0 \\ 0 & \text{for } w \geq w_0 \end{cases} \quad (4.9)$$

where w_0 is the manipulability measure threshold value at which the singular boundary is starting, and k_0 is the scale factor at singularity. Implementation of this method can be used to solve Equation (4.1) as follows:

$$\dot{q} = J_{GE}^T \cdot (J_{GE} \cdot J_{GE}^T + k_{SR} I_m)^{-1} \cdot \dot{r}_{GE} \quad (4.10)$$

4.2.3 Weighted Least-Norm Solution

In general, according to the type of joints, or depending on actuators used in a mobile manipulator, each joint in the joint velocity vector \dot{q} may have a different motion range or a different desired motion preference in comparison with the other joints in the same vector. For instance, in the case of avoiding an obstacle during a navigation phase, more importance will be given to the mobile platform motion relative to the arm motion. Similarly, the desired

velocity vector of the mobile manipulator \dot{r}_{GE} will have components with different units and priorities according to the task at hand. For instance, in a welding task, the position of the end-effector will be more important than the orientation of the hand relative to the welding torch axis [74]. The following section will review a general case for situations when weighted matrices are used; one for the joint velocity vector \dot{q} , named as W_q and the second for the desired end-effector Cartesian velocity vector \dot{r}_{GE} , named as W_r .

4.2.3.1 General Case

In this case, two weighted matrices are used, (W_q and W_r). These two weighting matrices are symmetric and positive definite. For simplicity, these weighted matrices are taken as diagonal matrices [74]. The norm of the two vectors with the weighted matrices can be defined as follows:

$$\begin{aligned} |\dot{r}_{GE}|_{W_r} &= \sqrt{\dot{r}_{GE}^T \cdot W_r \cdot \dot{r}_{GE}} \\ |\dot{q}|_{W_q} &= \sqrt{\dot{q}^T \cdot W_q \cdot \dot{q}} \end{aligned} \quad (4.11)$$

For the purpose of analysis, we define the following transformation:

$$\begin{aligned} J_W &= W_r^{\frac{1}{2}} \cdot J_{GE} \cdot W_q^{-\frac{1}{2}} \\ \dot{r}_W &= W_r^{\frac{1}{2}} \cdot \dot{r}_{GE} \\ \dot{q}_W &= W_q^{\frac{1}{2}} \cdot \dot{q} \end{aligned} \quad (4.12)$$

Using these new defined transformations, Equation (4.1) can be rewritten as follows:

$$\dot{r}_W = J_W \cdot \dot{q}_W \quad (4.13)$$

and Equations (4.11) as:

$$\begin{aligned} |\dot{r}|_{W_r} &= \sqrt{\dot{r}_W^T \cdot \dot{r}_W} \\ |\dot{q}|_{W_q} &= \sqrt{\dot{q}_W^T \cdot \dot{q}_W} \end{aligned} \quad (4.14)$$

Therefore, Equation (4.4) can be rewritten as follows:

$$\dot{q}_W = J_W^\# \cdot \dot{r}_W \quad (4.15)$$

In general, for the redundant mobile manipulator, an infinite number of transformed joint velocity vectors \dot{q}_W lead to a desired transformed mobile manipulator velocity vector \dot{r}_W . Using Equation (4.3), Equation (4.15) can be rewritten as follows:

$$\dot{q}_W = J_W^T \cdot (J_W \cdot J_W^T)^{-1} \cdot \dot{r}_W \quad (4.16)$$

Substituting Equation (4.12) into Equation (4.16) gives:

$$W_q^{\frac{1}{2}} \cdot \dot{q} = \left(W_r^{\frac{1}{2}} \cdot J_{GE} \cdot W_q^{-\frac{1}{2}} \right)^T \cdot \left[\left(W_r^{\frac{1}{2}} \cdot J_{GE} \cdot W_q^{-\frac{1}{2}} \right) \cdot \left(W_r^{\frac{1}{2}} \cdot J_{GE} \cdot W_q^{-\frac{1}{2}} \right)^T \right]^{-1} \cdot W_r^{\frac{1}{2}} \cdot \dot{r}_{GE} \quad (4.17)$$

Equation (4.17) can be modified as follows:

$$W_q^{\frac{1}{2}} \cdot \dot{q} = W_q^{-\frac{1}{2}} \cdot J_{GE}^T \cdot W_r^{\frac{1}{2}} \cdot \left[W_r^{\frac{1}{2}} \cdot J_{GE} \cdot W_q^{-\frac{1}{2}} \cdot W_q^{-\frac{1}{2}} \cdot J_{GE}^T \cdot W_r^{\frac{1}{2}} \right]^{-1} \cdot W_r^{\frac{1}{2}} \cdot \dot{r}_{GE} \quad (4.18)$$

Collecting similar terms yield to:

$$\dot{q} = W_q^{-1} \cdot J_{GE}^T \cdot W_r^{\frac{1}{2}} \cdot \left[W_r^{\frac{1}{2}} \cdot J_{GE} \cdot W_q^{-1} \cdot J_{GE}^T \cdot W_r^{\frac{1}{2}} \right]^{-1} \cdot W_r^{\frac{1}{2}} \cdot \dot{r}_{GE} \quad (4.19)$$

Equation (4.19) gives weighted LN solution for weighted joint angles and weighted Cartesian velocities of a mobile manipulator. In the following section, a case study will be presented where only a weighted joint angle matrix W_q is used and W_r is assumed to be an identity matrix.

4.2.3.2 Special Case: Joint Limit Avoidance (JLA)

Joint Limit Avoidance, (JLA), is a method in which all joint angles for the manipulator are kept within their permissible ranges to prevent any physical damage. In this special case from the general case (4.19), only the weighted matrix W_q is used. Taking this into account, Equation (4.19) can be rewritten as follows:

$$\dot{q} = W_q^{-1} \cdot J_{GE}^T \cdot \cdot \left[J_{GE} \cdot W_q^{-1} \cdot J_{GE}^T \right]^{-1} \cdot \dot{r}_{GE} \quad (4.20)$$

Using redundancy to avoid joint limits, a performance criterion that relates the joint angles to their physical joint limits may be specified and optimized. Chan and Dubey in [73] used the following performance criterion to avoid joint limits:

$$H(q) = \sum_{i=1}^n \frac{1}{4} \frac{(q_{imax} - q_{imin})^2}{(q_{imax} - q_i)(q_i - q_{imin})} \quad (4.21)$$

where q_i is the i^{th} joint angle, q_{imin} and q_{imax} are the lower and upper joint limits for joint i , respectively. The value of this criterion function H becomes smaller as the current joint angles approaches the middle of their range ($q_i = \frac{q_{imax} + q_{imin}}{2}$) and becomes higher as the joints get closer to their limits. The optimization function (4.21) can be incorporated into the weighted LN solution (4.20) by defining the weighted matrix W_q as a diagonal matrix as follows:

$$W_q = \begin{bmatrix} w_1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & w_2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & w_n & 0 & 0 \\ 0 & 0 & \cdots & 0 & w_{n+1} & 0 \\ 0 & 0 & \cdots & 0 & 0 & w_{n+2} \end{bmatrix} \quad (4.22)$$

where w_i is the i^{th} diagonal elements of W_q that can be defined as follows:

$$w_i = 1 + \left| \frac{\partial H(q)}{\partial q_i} \right| = 1 + \frac{(q_{imax} - q_{imin})^2 \cdot (2 \cdot q_i - q_{imax} - q_{imin})}{4 \cdot (q_{imax} - q_i)^2 \cdot (q_i - q_{imin})^2} \quad (4.23)$$

It may be observed that $\frac{\partial H(q)}{\partial q_i}$ is equal to zero when the current joint angle q_i is at the middle of the range of its joint limit, and goes to infinity when the current joint angle is at one of the limits.

The diagonal elements of the weight matrix W_q have a range from 1 to infinity. The values of the weight matrix varies according to how far a joint is from its limits. Mathematically, when the i_{th} diagonal element on the weight matrix W_q approaches infinity, its corresponding element in the inverse matrix W_q^{-1} approaches zero. Therefore, by using Equation (4.20), the corresponding joint velocity \dot{q}_i approaches zero. This means that this joint approaches its limit and it is not allowed to move further toward its limit. However, this joint should be allowed to move away from the joint limit toward the middle. Hence, the following conditions are added:

$$w_i = \begin{cases} 1 + \left| \frac{\partial H(q)}{\partial q_i} \right| & \text{if } \Delta \left| \frac{\partial H(q)}{\partial q_i} \right| \geq 0 \\ 1 & \text{if } \Delta \left| \frac{\partial H(q)}{\partial q_i} \right| < 0 \end{cases} \quad (4.24)$$

4.2.4 Inverse Kinematics Considering the Order of Priority

Any complicated task that is assigned to a mobile manipulator can be decomposed to simpler subtasks with orders of priority. The subtask with higher priority is performed first and if any degree of redundancy is left, it is utilized to perform the subtask with a lower priority. For instance, in this work, the goal is to control a redundant mobile manipulator to follow a dual-trajectory. This task can be broken down into two subtasks which track the end-effector's trajectory and the mobile platform's trajectory. If the exact solution that fulfills the dual-trajectory tracking does not exist, the mobile manipulator controller should follow the trajectory of the higher priority and allow a pose error in the trajectory of the

lower priority. The priority can be alternated between the two subtasks. If, for example, the mobile platform is at a stage of avoiding an obstacle, the priority is given to the platform trajectory. The end-effector trajectory can be interrupted until the obstacle is avoided, and then the original trajectory may be resumed if possible. The same scenario would occur if the end-effector trajectory has the first priority.

As stated by Nakamura in [8], the inverse kinematics solution considering task priority can be achieved using the following equation:

$$\dot{q} = J_1^\# \dot{r}_1 + (I - J_1^\# J_1) \hat{J}_2^\# (\dot{r}_2 - J_2 J_1^\# \dot{r}_1) + (I - J_1^\# J_1)(I - \hat{J}_2^\# \hat{J}_2)H \quad (4.25)$$

where J_i ($i = 1, 2, \dots$) is the Jacobian matrix of the i^{th} subtask, I is an identity matrix, $J_i^\# = J_i^T (J_i J_i^T)^{-1} J_i$ is the pseudo-inverse of J_i , \dot{r}_i ($i = 1, 2, \dots$) is the desired velocities of the i^{th} subtask variables, and $\hat{J}_2 = J_2(I - J_1^\# J_1)$. The first right hand side term of Equation (4.25) is the LN solution [73] which provides \dot{r}_1 with minimum Euclidian norm. The second term is the desired value for the modified second-subtask variables considering the effect of the first term. The third term is the orthogonal projection of vector H onto the remaining subspace.

4.2.5 Optimization Criteria

One of the most beneficial advantages of redundancy in mobile manipulators is the fact that its motion can be optimized in many different ways. We can use the additional DoF to optimize a secondary task or to set motion preference weight on the joint domain while controlling the first task. One of the optimization methods that will be used and tested is manipulability measure optimization [9].

Manipulability is the ability of the robotic system in positioning and orienting the end-effector at the tip of the manipulator. This ability can be described through two definitions. First, manipulability is the ability to reach a set of poses in the manipulator work space, and second, the ability to change the pose at a given configuration [75]. In [9], Yoshikawa has

proposed a manipulability measure that is applicable for both redundant and non-redundant manipulators. The following is the summary of the mathematical presentation of this measure.

Recall Equation (3.15), for the robotic arm:

$$\dot{r}_{GE_A} = J_{GE_A} \cdot \dot{q}_A \quad (4.26)$$

Consider the set of end-effectors Cartesian velocities $\dot{r}_{GE_A} \in \mathbb{R}^m$ that are achievable by the set of the joint velocities $\dot{q}_A \in \mathbb{R}^n$ such that the Euclidean norm satisfied:

$$\|\dot{q}_A\| = \sqrt{\dot{q}_{A1}^2 + \cdots + \dot{q}_{An}^2} \leq 1 \quad (4.27)$$

This set is an ellipsoid in the m -dimensional Euclidean space. The direction and the magnitude of the end-effector velocities are dependent on the directions and the magnitudes of the axes of the ellipsoid. While the high velocity can be achieved in the direction of the major axis, the low speed is along the minor axis. By using the singular-value decomposition (SVD) of J , the principal axes of the manipulability ellipsoid can be found. The SVD of J can be represented as follows:

$$J_{AE} = U \Sigma V^T \quad (4.28)$$

where U and V are $m \times m$ and $n \times n$ orthogonal matrices respectively, and Σ is an $m \times n$ matrix defined as:

$$\Sigma = \left[\begin{array}{ccc|c} \sigma_1 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \sigma_m & 0 \end{array} \right], \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \geq 0 \quad (4.29)$$

The scalars $\sigma_1, \sigma_2, \cdots, \sigma_m$ are called singular values. These are the larger values of the square root of eigenvalues of the matrix $J_{AE}^T J_{AE}$. The principal axes of the manipulability

ellipsoid are $\sigma_1 u_1, \sigma_2 u_2, \dots, \sigma_m u_m$ where u_i is the i^{th} column vector of U . Figure 4.1 shows the manipulability ellipsoid with the principal axes.

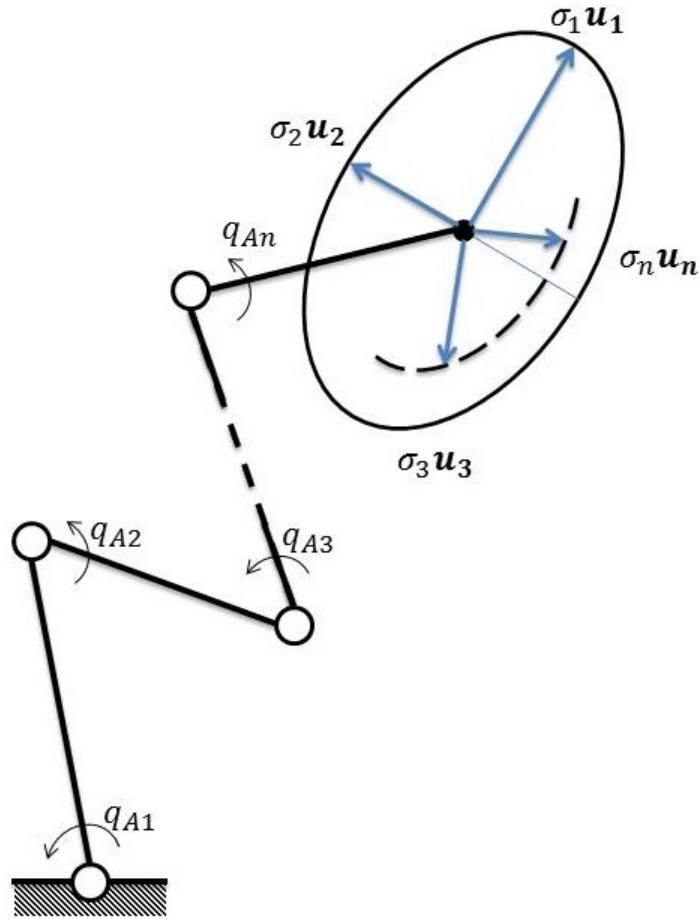


Figure 4.1: Manipulability ellipsoid with principal axes.

A representative measure of the ability of manipulation in certain configurations in 3-dimensional space is the scaled volume of the ellipsoid at that particular configuration ($w = \sigma_1 \sigma_2 \dots \sigma_m$). This can be represented as a scalar value as follows:

$$w = \sqrt{\det(J(q) \cdot J(q)^T)} \quad (4.30)$$

This measure indicates how far this particular configuration is from a singularity. The larger the w , the farther away it is from singularities and vice versa. When $w = 0$, a manipulator is at a singular configuration. For a non-redundant manipulator ($m = n$),

$w = |\det J(q)|$. This manipulability measure for robotic arms has been extended to the case of mobile manipulators [38].

CHAPTER 5
NOVEL DUAL-TRAJECTORY TRACKING USING SPHERICAL
CONTROL VARIABLES (D , α AND β)

5.1 Introduction

Mobile manipulators have two independent trajectories: end-effector trajectory and mobile platform trajectory. In complex tasks, these trajectories have to be tracked simultaneously. There are cases when the planned trajectories are not trackable by a mobile manipulator due to hardware limitations. In these cases, higher priority can be given to the trajectory that is more important to follow precisely and allow error in the other trajectory. Task priority with new control variables is presented in this chapter.

This chapter¹ presents a novel control system for redundant mobile manipulators to track separate mobile platform and end-effector trajectories (dual-trajectory) simultaneously. In this control scheme, the trajectory with higher priority is fully defined. However, the other trajectory is defined according to the mobile manipulator hardware capabilities. Three spherical control variables are introduced to the task vector. These three control variables are D , α and β , which define the mobile platform trajectory in relation to the end-effector trajectory and vice versa. Furthermore, they relate the mobile platform motion to the robotic arm joint angles. This allows direct control of the mobile platform motion. In addition, spherical control variable D can be considered as a robotic arm stretching measure. This measure can set maximum and minimum limits that determine how far/closer the end-

¹A conference paper was published related to the material that is presented in this chapter [76]. Permission is included in Appendix A.

effector can be from the mobile platform frame. A singularity-robust and task-priority inverse with gradient projection method is used to find solutions for the dual-trajectory tracking while maximizing the arm manipulability. MATLAB simulated Planar Mobile Manipulation (PMM) is used to test and optimize the proposed control system. The results demonstrate the effectiveness of the control system in tracking the two trajectories and optimizing the PMM manipulability measure.

5.2 Terminology

The dual-trajectory of a mobile manipulator consists of an end-effector and a mobile platform trajectories. Dual-trajectory will also be refer to as mobile manipulator trajectory.

5.3 Dual-Trajectory Combined Control

As mentioned previously, it is assumed that the trajectories of both the end-effector and mobile platform are defined according to the priority order and the capabilities of the mobile manipulator. This means that in each and every time instance, the pose (position and orientation) of both the end-effector and the mobile platform are known as it is shown in Figure 5.1. In regions where the dual-trajectory is not trackable, the via points of the trajectory with lower priority will be compromised to allow following the trajectory of higher priority. The aim is to design and optimize a control scheme that combines the manipulation of the robotic arm and the mobility of the platform in a single algorithm. This control scheme should guarantee that the end-effector and the mobile platform will follow the priority order of their specified independent trajectories, and at the same time satisfy other tasks or optimize a performance criteria such as maximize the manipulability measure of the PMM.

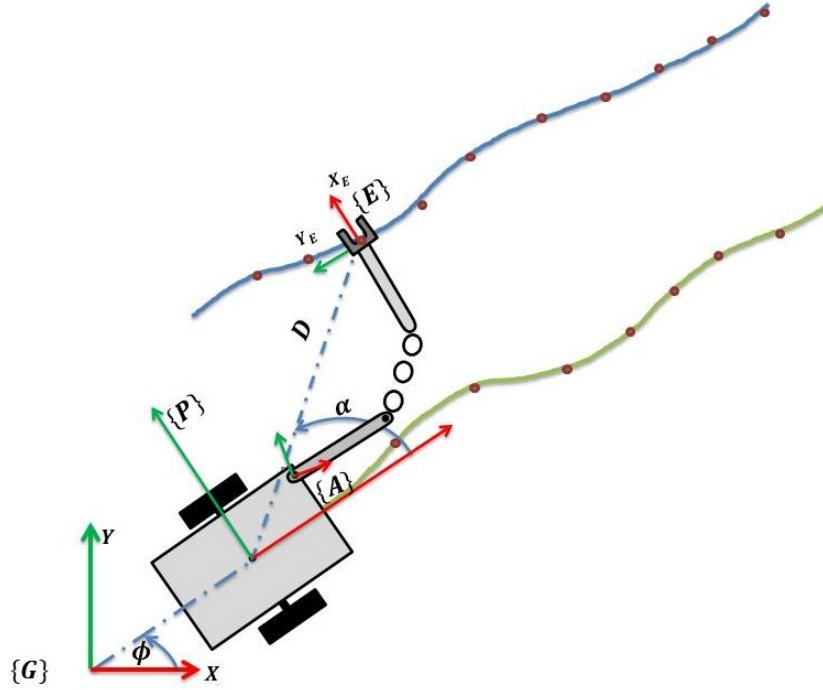


Figure 5.1: A mobile manipulator predefined dual-trajectory with spherical control variables D and α .

5.3.1 Spherical Control Variable (D , α and β)

Using Equation (3.4), the end-effector can be controlled to follow the predefined end-effector trajectory, and the mobile platform will follow the end-effector using the null space optimization criterion. This does not provide a full control of the mobile platform trajectory. Introducing control variables that represented the mobile platform trajectory to space task enables direct control over the platform trajectory. Using Equation (3.8), the mobile platform linear and angular velocities are mapped one to one from the task velocity vector to the joint angle vector by J_{GP} . As it can be noticed, there is no correlation between the arm's joint angles and the mobile platform's motion.

To correlate the mobile platform and the end-effector motion, two control variables are introduced to define the platform trajectory relative to the end-effector trajectory. As shown in Figure 5.1, these two control variables are D and α , where D is the Euclidean distance

between the end-effectors and the mobile platforms frames origins, and α is the angle on the XY plane (Azimuth angle) of the end-effector position relative to the mobile platform frame. In the case of finding the end-effector trajectory relative to the mobile platform trajectory, a third variable β is introduced to find the vertical angle which will allow finding the Z-coordinate of the end-effector as shown in Figure 5.2.

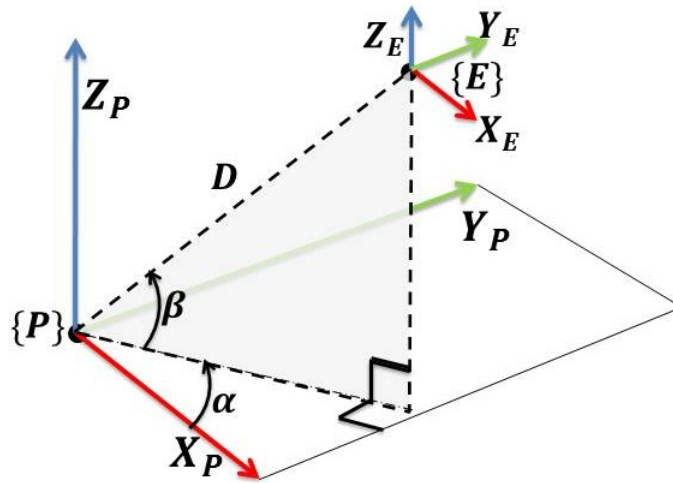


Figure 5.2: Mobile manipulator end-effector and mobile platform frames with spherical control variables D , α and β .

The control variables D , α and β were introduced for the following reasons. First, to define the mobile platform trajectory in relation to the end-effector trajectory and vice versa. Second, to relate the mobile platform motion to the robotic arm joint angles. By changing the values of these variables, there can be control of the mobile platform pose using the end-effector pose and the desired mobile platform trajectory that will be utilized for optimization and task prioritization. This is particularly useful when it is not physically possible to follow both the end-effector and platform trajectories simultaneously. The main advantage of using spherical control variables over the method represented in Equation (3.8) is that with the control variable D , a limit can be introduced as to how much the end-effector can stretch away or toward the mobile platform. More details will be presented on the “arm stretching

measure” in Section 5.5. This is useful to prevent singularity that occurs when the arm is fully stretched and to limit the robotic arm workspace when the mobile manipulator is navigating in cluttered environments.

5.3.1.1 Platform Pose Relative to End-Effector Position

As previously mentioned in Section 3.3, the mobile platform has 2 DoF. Therefore, two variables are needed to fully control the mobile platform motion. These two variables, D , and α , are functions of the arm and mobile platform joint angles. Using the same approach that is used to derive Equation (3.8), new additional constraints, that describe the mobile platform position relative to the end-effector position, can be defined as follows:

$$r_{D\alpha} = \begin{bmatrix} D \\ \alpha \end{bmatrix} = \begin{bmatrix} y_1(q_A, q_P) \\ y_2(q_A, q_P) \end{bmatrix} \quad (5.1)$$

For the new constraint, Equation (3.8) can be modified as follows:

$$\dot{r}_{EP} = \begin{bmatrix} \dot{r}_{GE} \\ \dots \\ \dot{r}_{D\alpha} \end{bmatrix} = \begin{bmatrix} \dot{r}_{GE} \\ \dots \\ \dot{D} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} J_{GE} \\ \dots \\ J_{D\alpha} \end{bmatrix} \begin{bmatrix} \dot{q}_A \\ \dots \\ \dot{q}_P \end{bmatrix} \quad (5.2)$$

where $J_{D\alpha} \in R^{2 \times (n+2)}$. The task vector for this case will be named as \dot{r}_{E2P} to distinguish from the following case. The subscript $E2P$ means the mobile platform poses relative to the end-effector poses. So, Equation (5.2) can be rewritten as follows:

$$\dot{r}_{E2P} = \begin{bmatrix} J_{GE} \\ J_{D\alpha} \end{bmatrix} \begin{bmatrix} \dot{q}_A \\ \dot{q}_P \end{bmatrix} \quad (5.3)$$

The task velocity vector for the following case will be named as \dot{r}_{P2E} which means the end-effector poses relative to mobile platform poses. This case is presented in the following section.

5.3.1.2 End-Effector Position Relative to Platform Pose

Because each end-effector trajectory waypoint has six variables (three for position and three for orientation), more variables are needed for describing the end-effector trajectory relative to the mobile platform trajectory. Using three spherical variables, only the end-effector position can be described relative to the mobile platform pose. If the end-effector orientation is needed, the corresponding orientation Jacobian J_ω from J_{GE} in Equation (3.8) is used. Since the mobile platform location is independent from the end-effector position, the same technique used in deriving Equation (3.8) is applied. Therefore, Equation (5.1) can be modified as follows:

$$r_{D\alpha\beta} = \begin{bmatrix} D \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} y_1(q_A, q_P) \\ y_2(q_A, q_P) \\ y_3(q_A, q_P) \end{bmatrix} \quad (5.4)$$

The kinematic model that relates the end-effector and mobile platform poses to the joint space poses as follows:

$$\dot{r}_{P2E} = \begin{bmatrix} \dot{D} \\ \dot{\alpha} \\ \dot{\beta} \\ \omega_x \\ \omega_y \\ \omega_z \\ \dots \\ \dot{r}_{GP} \end{bmatrix} = \begin{bmatrix} J_{D\alpha\beta} \\ \\ \\ J_\omega \\ \dots \\ J_{GP} \end{bmatrix} \begin{bmatrix} \dot{q}_A \\ \\ \\ \dots \\ \dot{q}_P \end{bmatrix} \quad (5.5)$$

where J_{GP} is similar to that in Equation (3.8). J_ω relates the end-effector orientation to the mobile manipulator joint angles. ω_x , ω_y and ω_z are the Cartesian rotation angle of the end-effector. In Equation (5.5), the control variables D , α and β represent the position of the end-effector relative to the mobile platform pose while in Equation (5.2) the control variables D , and α represent the mobile platform pose relative to end-effector pose.

5.3.2 Jacobian of the Spherical Control Variables

Before the spherical control variables Jacobian is determined, let us define these variables. Refer to Figure 5.3 for variables definitions. Control variable D can be defined as follows:

$$D = \sqrt{{}^G X_{PE}^2 + {}^G Y_{PE}^2 + {}^G Z_{PE}^2} \quad (5.6)$$

where ${}^G X_{PE} = {}^G_E X - {}^G_P X$, ${}^G Y_{PE} = {}^G_E Y - {}^G_P Y$ and ${}^G Z_{PE} = {}^G_E Z - {}^G_P Z$ are the distances in global X, Y, and Z directions respectively. These distances are from the mobile platform frame P origin to the end-effector frame E origin. Variable α is defined as follows:

$$\alpha = \tan^{-1} \left(\frac{{}^G Y_{PE}}{{}^G X_{PE}} \right) - \phi \quad (5.7)$$

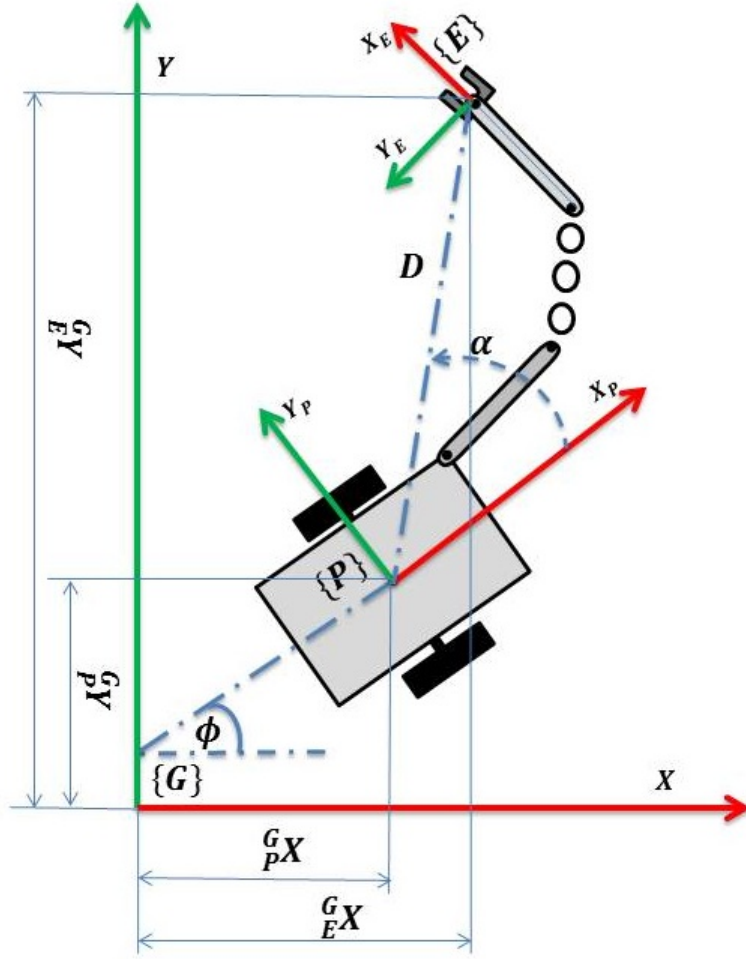


Figure 5.3: Control variables D , and α with global distance.

where ϕ is the orientation angle of the mobile platform. Variable β is defined as follows:

$$\beta = \tan^{-1} \left(\frac{G Z_{PE}}{\sqrt{G X_{PE}^2 + G Y_{PE}^2}} \right) \quad (5.8)$$

The Jacobian of the control variables D , α and β is defined as follows:

$$J_{D\alpha\beta} = \begin{bmatrix} \nabla D(\theta_1, \dots, \theta_n, S, \phi) \\ \nabla \alpha(\theta_1, \dots, \theta_n, S, \phi) \\ \nabla \beta(\theta_1, \dots, \theta_n, S, \phi) \end{bmatrix} = \begin{bmatrix} \frac{\partial y_1}{\partial q_A} + \frac{\partial y_1}{\partial q_P} \\ \frac{\partial y_2}{\partial q_A} + \frac{\partial y_2}{\partial q_P} \\ \frac{\partial y_3}{\partial q_A} + \frac{\partial y_3}{\partial q_P} \end{bmatrix} = \begin{bmatrix} \frac{\partial D}{\partial \theta_1} & \dots & \frac{\partial D}{\partial \theta_n} & \frac{\partial D}{\partial S} & \frac{\partial D}{\partial \phi} \\ \frac{\partial \alpha}{\partial \theta_1} & \dots & \frac{\partial \alpha}{\partial \theta_n} & \frac{\partial \alpha}{\partial S} & \frac{\partial \alpha}{\partial \phi} \\ \frac{\partial \beta}{\partial \theta_1} & \dots & \frac{\partial \beta}{\partial \theta_n} & \frac{\partial \beta}{\partial S} & \frac{\partial \beta}{\partial \phi} \end{bmatrix} \quad (5.9)$$

where $J_{D\alpha\beta} \in R^{3 \times (n+2)}$. As shown in Equations (5.2) and (5.5), the Jacobian of the control variables relates the velocity of these control variables \dot{D} , $\dot{\alpha}$ and $\dot{\beta}$ to the joint angle velocity vector.

5.4 Dual-Trajectory Control With Order of Priority

The main goal here is to control a redundant mobile manipulator to follow a dual-trajectory. In a case when the exact solution that fulfills the dual-trajectory tracking does not exist, the mobile manipulator should follow the high priority trajectory and allow a pose error in the low priority trajectory. In addition, if more redundancy is left for the arm, another optimization criterion can be satisfied. In this case, the controller should keep a high manipulability measure for the robotic arm.

In this case, the task of the redundant mobile manipulator is to follow a dual-trajectory. This task can be divided into two subtasks, which are: tracking the end-effector's trajectory and tracking the mobile platform's trajectory. If, for example, the mobile platform is at a stage of avoiding an obstacle, the priority is for the platform's trajectory. The end-effector trajectory can be interrupted until the obstacle is avoided, then resume the original trajectory if possible. The same scenario will occur if the end-effector's trajectory has the first priority. In this work, the techniques of alternating the order of priority between the two subtasks are not addressed, rather, the order of priority is predefined.

As it is stated by Nakamura in [8], the inverse kinematics solution considering task priority can be achieved using the following equation:

$$\dot{q} = J_1^\# \dot{r}_1 + \left(I - J_1^\# J_1 \right) \hat{J}_2^\# \left(\dot{r}_2 - J_2 J_1^\# \dot{r}_1 \right) + \left(I - J_1^\# J_1 \right) \left(I - \hat{J}_2^\# \hat{J}_2 \right) H \quad (5.10)$$

where J_i ($i = 1, 2$) is the Jacobian matrix of the i^{th} subtask. In this case, J_1 and J_2 are the Jacobian of the high and low priority subtasks, respectively. I is the identity matrix,

$J_i^\# = J_i^T (J_i J_i^T)^{-1}$ is the pseudo-inverse of J_i , \dot{r}_i ($i = 1, 2$) is the desired velocities of the i^{th} subtask variables, and $\hat{J}_2 \equiv J_2(I - J_1^\# J_1)$.

The first right hand side term of Equation (5.10) is the LN solution [73] which provides \dot{r}_1 with minimum Euclidean norm. The second term is the desired value for the modified second-subtask variables considering the effect of the first term. The third term is the orthogonal projection of vector H onto the remaining subspace. Hence, solutions of Equation (5.10) should satisfy the tracking of the mobile manipulator's dual-trajectory. No solution will exist if the mobile manipulator cannot keep track of both trajectories due to physical limitations of the system, where the system is considered to be in a singular configuration. To alleviate the kinematic singularity problems, the SR-Inverse method [8] is used (for more details see Section 4.2.2).

5.5 Implementation Example

A PMM will be used as an example to explain the proposed controller. However, this controller can be applied for any mobile manipulator. The PMM consists of a planar robotic arm with three revolute joints mounted on top of a nonholonomic mobile platform as shown in Figure 5.4. The base of the robotic arm is placed along the X-axis of the mobile platform at a distance l_A from the mobile platform frame P , (frame P is at the middle of the wheel's axis). The following describes how the robotic arm stretching measure can be calculated. Figure 5.5 illustrates the PMM parameters (D_{max} , α , L_{max} , and l_A). L_{max} is the arm full stretch length. D_{max} can be determined as follows:

$$D_{max} = L_{max} \cos\alpha_1 + l_A \cos\alpha \quad (5.11)$$

For this case, sine law can be written as follows:

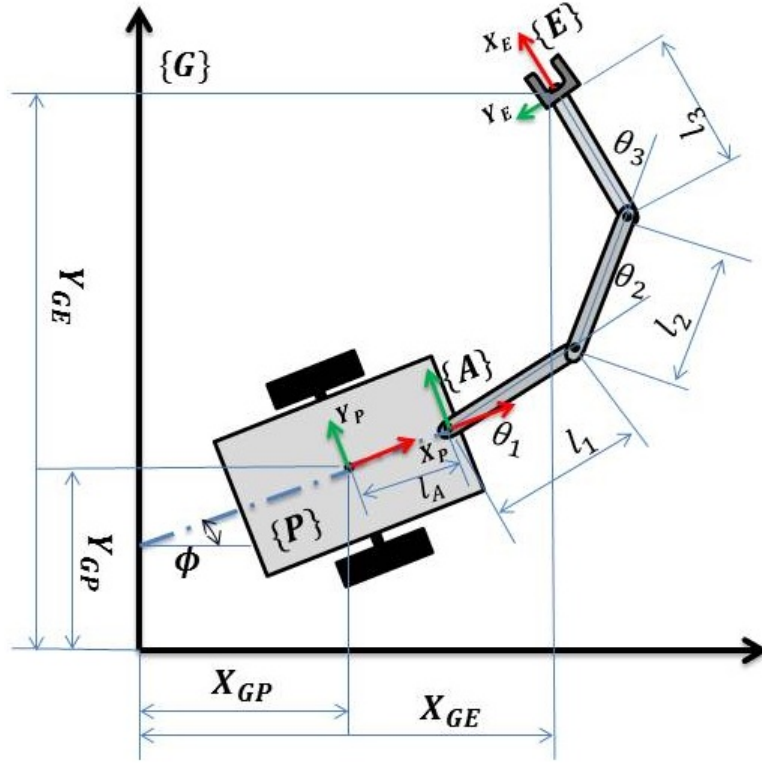


Figure 5.4: Planar mobile manipulator with 3 DoF planar robotic arm and nonholonomic mobile platform.

$$\frac{\sin\alpha}{L_{max}} = \frac{\sin\alpha_1}{l_A} = \frac{\sin\alpha_2}{D_{max}} \quad (5.12)$$

From Equation (5.12), the angle α_1 can be calculated as follows:

$$\alpha_1 = \sin^{-1}\left(\frac{l_A \sin\alpha}{L_{max}}\right) \quad (5.13)$$

Substituting Equation (5.13) into Equation (5.11) yields:

$$D_{max} = L_{max} \cos\left[\sin^{-1}\left(\frac{l_A \sin\alpha}{L_{max}}\right)\right] + l_A \cos\alpha \quad (5.14)$$

From the trigonometric relation that evaluates cosine of the arcsine:

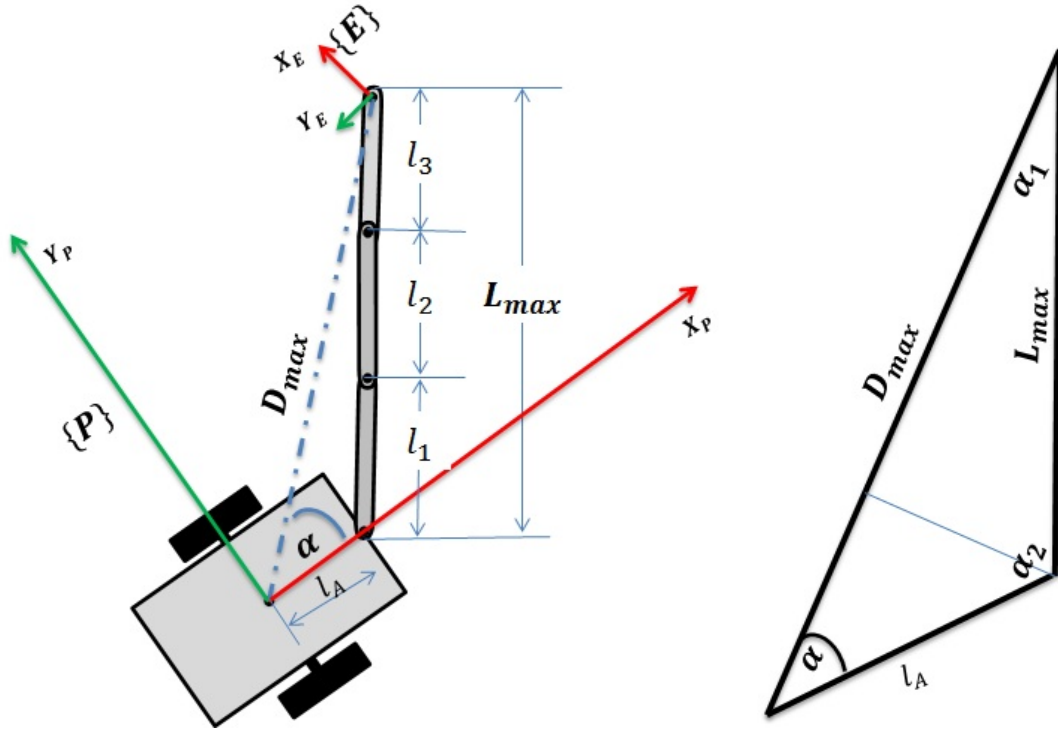


Figure 5.5: 5 DoF Planar Mobile Manipulator parameters (D_{max} , α , L_{max} and l_A).

$$\cos(\sin^{-1}x) = \sqrt{1-x^2} \quad (5.15)$$

Using Equation (5.15), Equation (5.14) can be rewritten as follows:

$$D_{max} = L_{max} \sqrt{1 - \frac{l_A^2 \sin^2 \alpha}{L_{max}^2}} + l_A \cos \alpha \quad (5.16)$$

Equation (5.16) can be modified as follows:

$$D_{max} = \sqrt{L_{max}^2 - l_A^2 \sin^2 \alpha} + l_A \cos \alpha \quad (5.17)$$

Equation (5.17) represents the relation of the maximum distance between the mobile platform P and end-effector E frame origins with mobile manipulator parameters, such as L_{max} , l_A ,

and α . As stated previously, this measure can be used to set minimum and maximum limits to the end-effector to constrain the arm workspace.

The configuration of the PMM is $q = [q_A \ q_P]^T \in \mathbb{R}^5$, with $q_A = [\dot{\theta}_1 \ \dot{\theta}_2 \ \dot{\theta}_3]^T \in \mathbb{R}^3$, and $q_P = [\dot{S} \ \dot{\phi}]^T \in \mathbb{R}^2$. The complete state variables of the PMM are:

$$\dot{r}_{GE} = \begin{bmatrix} \dot{x}_{GE} \\ \dot{y}_{GE} \\ \dot{D} \\ \dot{\alpha} \end{bmatrix} = J_{PMM} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{S} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} J_{GE_{A2 \times 3}} & J_{GE_{P2 \times 2}} \\ \hline {}^G J_{D\alpha_{2 \times 5}} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{S} \\ \dot{\phi} \end{bmatrix} \quad (5.18)$$

where the superscript G indicates that it is relative to the global coordinate frame while E is the end-effector frame, and P is the mobile platform frame. This PMM has 1 DoR. Refer also to Figure 5.4 for the variable definitions. For the end-effector's trajectory, the E frame position relative to the global frame is as follows:

$$\begin{bmatrix} x_{GE} \\ y_{GE} \end{bmatrix} = \begin{bmatrix} X_{GP} + l_A C_\phi + l_1 C_{\phi\theta_1} + l_2 C_{\phi\theta_1\theta_2} + l_3 C_{\phi\theta_1\theta_2\theta_3} \\ Y_{GP} + l_A S_\phi + l_1 S_{\phi\theta_1} + l_2 S_{\phi\theta_1\theta_2} + l_3 S_{\phi\theta_1\theta_2\theta_3} \end{bmatrix} \quad (5.19)$$

where $C_{lmn} = \cos(l + m + n)$ and $S_{lmn} = \sin(l + m + n)$. The Jacobian J_{GE_A} and J_{GE_P} are:

$$J_{GE_A} = \begin{bmatrix} -l_1 S_{\phi\theta_1} - l_2 S_{\phi\theta_1\theta_2} - l_3 S_{\phi\theta_1\theta_2\theta_3} & -l_2 S_{\phi\theta_1\theta_2} - l_3 S_{\phi\theta_1\theta_2\theta_3} & -l_3 S_{\phi\theta_1\theta_2\theta_3} \\ l_1 C_{\phi\theta_1} + l_2 C_{\phi\theta_1\theta_2} + l_3 C_{\phi\theta_1\theta_2\theta_3} & l_2 C_{\phi\theta_1\theta_2} + l_3 C_{\phi\theta_1\theta_2\theta_3} & l_3 C_{\phi\theta_1\theta_2\theta_3} \end{bmatrix} \quad (5.20)$$

$$J_{GE_P} = \begin{bmatrix} C_\phi & -l_A S_\phi - l_1 S_{\phi\theta_1} - l_2 S_{\phi\theta_1\theta_2} - l_3 S_{\phi\theta_1\theta_2\theta_3} \\ S_\phi & l_A C_\phi + l_1 C_{\phi\theta_1} + l_2 C_{\phi\theta_1\theta_2} + l_3 C_{\phi\theta_1\theta_2\theta_3} \end{bmatrix} \quad (5.21)$$

The Jacobian of the new control variables D and α can be determined as follow. The Cartesian distance between the mobile platform frame and the end-effector frame origins

relative to global frame can be described as:

$$\begin{bmatrix} {}^G X_{PE} \\ {}^G Y_{PE} \end{bmatrix} = \begin{bmatrix} l_A C_\phi + l_1 C_{\phi\theta_1} + l_2 C_{\phi\theta_1\theta_2} + l_3 C_{\phi\theta_1\theta_2\theta_3} \\ -l_A S_\phi - l_1 S_{\phi\theta_1} - l_2 S_{\phi\theta_1\theta_2} - l_3 S_{\phi\theta_1\theta_2\theta_3} \end{bmatrix} \quad (5.22)$$

The control variables can be defined as:

$$\begin{aligned} D &= \sqrt{{}^G X_{PE}^2 + {}^G Y_{PE}^2} \\ \alpha &= \tan^{-1} \left(\frac{{}^G Y_{PE}}{{}^G X_{PE}} \right) - \phi \end{aligned} \quad (5.23)$$

The Jacobian can be evaluated using the following equation:

$${}^G J_{D\alpha} = \begin{bmatrix} \nabla D(\theta_1, \theta_2, \theta_3, S, \phi) \\ \nabla \alpha(\theta_1, \theta_2, \theta_3, S, \phi) \end{bmatrix} = \begin{bmatrix} \frac{\partial D}{\partial \theta_1} & \frac{\partial D}{\partial \theta_2} & \frac{\partial D}{\partial \theta_3} & \frac{\partial D}{\partial S} & \frac{\partial D}{\partial \phi} \\ \frac{\partial \alpha}{\partial \theta_1} & \frac{\partial \alpha}{\partial \theta_2} & \frac{\partial \alpha}{\partial \theta_3} & \frac{\partial \alpha}{\partial S} & \frac{\partial \alpha}{\partial \phi} \end{bmatrix} \quad (5.24)$$

where

$$\begin{aligned} \frac{\partial D}{\partial \theta_1} &= \frac{1}{\sqrt{{}^G X_{PE}^2 + {}^G Y_{PE}^2}} \left(\frac{\partial {}^G X_{PE}}{\partial \theta_1} + \frac{\partial {}^G Y_{PE}}{\partial \theta_1} \right) = \frac{1}{D} ({}^G J_{PE}(1, 1) + {}^G J_{PE}(2, 1)) \\ \frac{\partial D}{\partial \theta_2} &= \frac{1}{\sqrt{{}^G X_{PE}^2 + {}^G Y_{PE}^2}} \left(\frac{\partial {}^G X_{PE}}{\partial \theta_2} + \frac{\partial {}^G Y_{PE}}{\partial \theta_2} \right) = \frac{1}{D} ({}^G J_{PE}(1, 2) + {}^G J_{PE}(2, 2)) \\ \frac{\partial D}{\partial \theta_3} &= \frac{1}{\sqrt{{}^G X_{PE}^2 + {}^G Y_{PE}^2}} \left(\frac{\partial {}^G X_{PE}}{\partial \theta_3} + \frac{\partial {}^G Y_{PE}}{\partial \theta_3} \right) = \frac{1}{D} ({}^G J_{PE}(1, 3) + {}^G J_{PE}(2, 3)) \\ \frac{\partial D}{\partial S} &= 0 \\ \frac{\partial D}{\partial \phi} &= \frac{1}{\sqrt{{}^G X_{PE}^2 + {}^G Y_{PE}^2}} \left(\frac{\partial {}^G X_{PE}}{\partial \phi} + \frac{\partial {}^G Y_{PE}}{\partial \phi} \right) = \frac{1}{D} (J_{GP}(1, 2) + J_{GP}(2, 2)) \\ \frac{\partial \alpha}{\partial \theta_1} &= \frac{{}^G X_{PE}^2}{{}^G X_{PE}^2 + {}^G Y_{PE}^2} \left(\frac{\partial}{\partial \theta_1} \left(\frac{{}^G Y_{PE}}{{}^G X_{PE}} \right) \right) = \frac{1}{D^2} ({}^G X_{PE} \cdot {}^G J_{PE}(2, 1) - {}^G Y_{PE} \cdot {}^G J_{PE}(1, 1)) \\ \frac{\partial \alpha}{\partial \theta_2} &= \frac{{}^G X_{PE}^2}{{}^G X_{PE}^2 + {}^G Y_{PE}^2} \left(\frac{\partial}{\partial \theta_2} \left(\frac{{}^G Y_{PE}}{{}^G X_{PE}} \right) \right) = \frac{1}{D^2} ({}^G X_{PE} \cdot {}^G J_{PE}(2, 2) - {}^G Y_{PE} \cdot {}^G J_{PE}(1, 2)) \\ \frac{\partial \alpha}{\partial \theta_3} &= \frac{{}^G X_{PE}^2}{{}^G X_{PE}^2 + {}^G Y_{PE}^2} \left(\frac{\partial}{\partial \theta_3} \left(\frac{{}^G Y_{PE}}{{}^G X_{PE}} \right) \right) = \frac{1}{D^2} ({}^G X_{PE} \cdot {}^G J_{PE}(2, 3) - {}^G Y_{PE} \cdot {}^G J_{PE}(1, 3)) \\ \frac{\partial \alpha}{\partial S} &= 0 \\ \frac{\partial \alpha}{\partial \phi} &= \frac{{}^G X_{PE}^2}{{}^G X_{PE}^2 + {}^G Y_{PE}^2} \left(\frac{\partial}{\partial \phi} \left(\frac{{}^G Y_{PE}}{{}^G X_{PE}} \right) \right) - 1 = \frac{1}{D^2} ({}^G X_{PE} \cdot {}^G J_{GP}(2, 2) - {}^G Y_{PE} \cdot {}^G J_{GP}(1, 2)) - 1 \end{aligned}$$

For more details on PMM Jacobian derivation, refer to Appendix B.

5.6 Simulation Results and Discussion

A MATLAB simulation was developed for the PMM to test the proposed controller in tracking dual-trajectories. In this simulation, different trajectories were used to evaluate the performance and the effectiveness of the controller. The PMM geometric data, was (all dimensions are in *mm*) $l_1 = l_2 = l_3 = 600$, $l_A = 400$, wheel base $w_b = 800$ and wheel radius $w_r = 200$. The initial mobile platform configuration is $[x_P \ y_P \ \phi_P] = [0 \ 0 \ -35^\circ]$. The robotic arm had an initial configuration joint angle of the robotic arm in degrees $[\theta_1 \ \theta_2 \ \theta_3] = [0 \ 60 \ 100]^\circ$. The sinusoidal trajectory $y = d + a\sin(fx)$ was assigned to both trajectories with different amplitudes and frequencies, where a and f are the amplitude and the frequency of the sine wave respectively, and d is the initial position of the end-effector or the mobile platform. For the end-effector trajectory, $a = -900$ and $f = 3000$, and for the platform trajectory, $a = -2000$ and $f = 9000$. Variable D was set to $2000(mm)$

5.6.1 End-Effector Trajectory as First Priority

In Equation (5.10), the task priority is alternated by using the Jacobian of the high priority in the first term. Therefore, in this case $J_1 = [{}^G J_{PE_{2 \times 3}} \ {}^G J_{GP_{2 \times 2}}]$, $J_2 = [{}^G J_{D_{\alpha_{2 \times 5}}}]$, $\dot{r}_1 = \begin{bmatrix} \dot{x}_{GE} \\ \dot{y}_{GE} \end{bmatrix}$ and $\dot{r}_2 = \begin{bmatrix} \dot{D} \\ \dot{\alpha} \end{bmatrix}$, refer to Equation (5.10). Figure 5.6 demonstrates the results of the three instances of the Equation (5.10).

Figure 5.6-a shows the results of using the first term of Equation (5.10). The first term is the LN solution which satisfies tracking the end-effectors trajectory. The mobile platform does not have any constraints over its trajectory. As it is seen, the mobile platform followed a random trajectory. The manipulability measure of PMM drops dramatically because the arm has to fully stretch then the mobile platform will follow as shown in Figure 5.7-a. Figure 5.6-b shows the result of the first and second terms of Equation (5.10). The second term

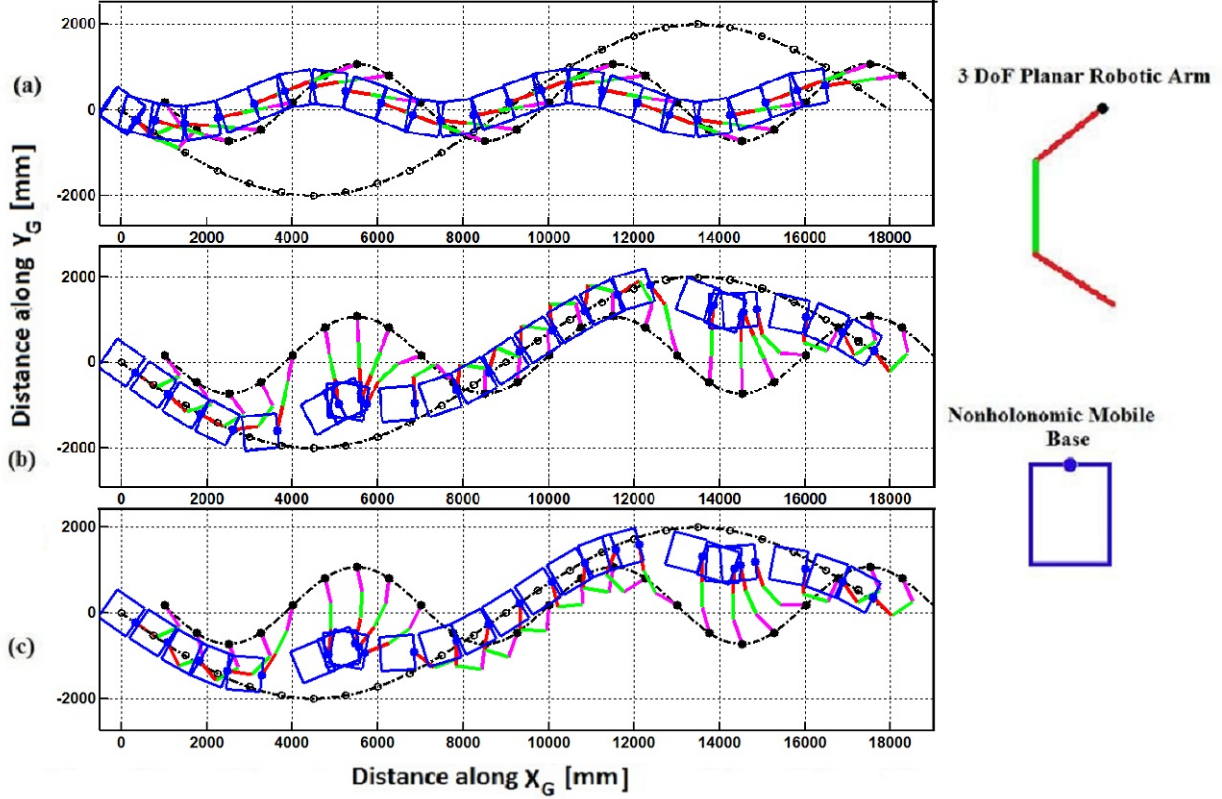


Figure 5.6: Tracked dual-trajectory of PMM for the three instance of Equation (5.10) $\dot{q} = J_1^\# \dot{r}_1 + (I - J_1^\# J_1) \hat{J}_2^\# (\dot{r}_2 - J_2 J_1^\# \dot{r}_1) + (I - J_1^\# J_1) (I - \hat{J}_2^\# \hat{J}_2) H$: a) First term. b) First and second terms. c) All terms. The first priority is given for the end-effector.

is responsible for satisfying the second priority task which is following the mobile platform trajectory. Now the mobile platform followed its predefined trajectory as long as possible. In the situation where the mobile manipulator cannot satisfy both trajectories, it followed the primary trajectory, the end-effector trajectory and then allowed pose error in the secondary trajectory, which is the mobile platform trajectory. This can be seen in Figure 5.6-b from distance 3000 to 7000 mm in X_G . The opposite scenario happened when the mobile platform trajectory was set as the first priority trajectory.

Figure 5.6-c shows the result of using all terms in Equation (5.10). In this instance, the manipulability measure of the PMM is maximized according to the redundancy left over after satisfying the first and second priorities. Figure 5.7 shows the Whole PMM

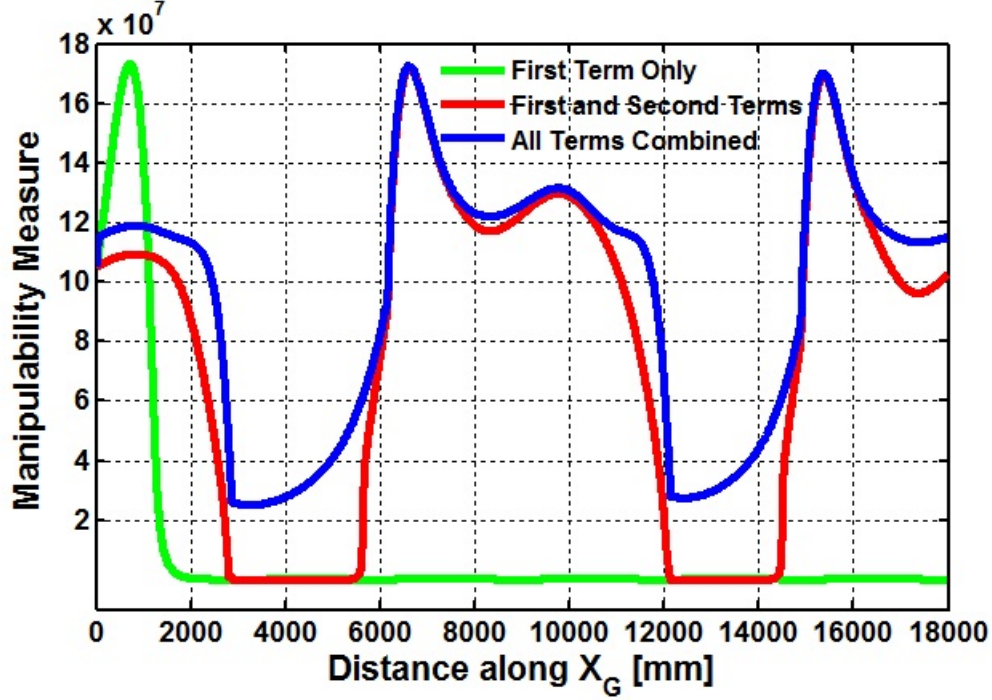


Figure 5.7: PMM manipulability measure for the three instances of Equation (5.10). The first priority is given for the end-effector trajectory.

system manipulability measure for each instance. As seen from Figure 5.7, the manipulability measure in instance *c* is the highest manipulability measure comparing with the other two instances.

5.6.2 Mobile Platform Trajectory as First Priority

In this case, $J_1 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$, $J_2 = [{}^G J_{D\alpha_{2 \times 5}}]$, $\dot{r}_1 = \begin{bmatrix} \dot{S} \\ \dot{\phi} \end{bmatrix}$ and $\dot{r}_2 = \begin{bmatrix} \dot{D} \\ \dot{\alpha} \end{bmatrix}$ in Equation (5.10). Figure 5.8 shows that the trajectory of the mobile platform was always followed while the end-effector trajectory was followed as much as possible. Also, as shown in Figure 5.9, the manipulability measure of PMM was maximized as much as possible.

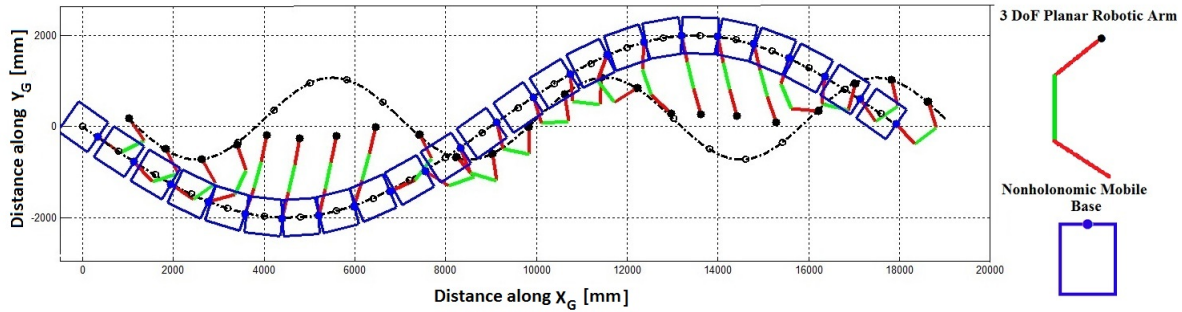


Figure 5.8: Tracked dual-trajectory of the PMM for all terms of Equation (5.10). The higher priority is given to follow the mobile platform trajectory.

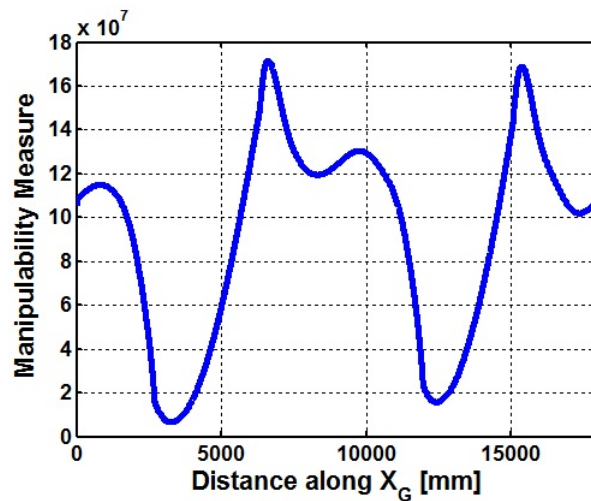


Figure 5.9: PMM manipulability measure for all terms in Equation (5.10). The first priority is set to following the mobile platform trajectory.

5.7 Summary

A dual-trajectory control system was designed and implemented for a nonholonomic redundant mobile manipulator. Three spherical control variables were introduced to the task vector. These three control variables were D , α and β . The main purpose of these variables was to relate the mobile platform motion to the robotic arm joint angles. By changing the values of these variables, the mobile platform pose could be controlled based on the end-effector pose and vice versa. The task priority redundancy resolution scheme was used to solve using the resolved rate solution for following dual-trajectories for different order

of priority between the end-effector and the mobile platform trajectories. This scheme was used with SR-Inverse to stabilize the system, and the Gradient Projection Method, (GPM) to maximize the manipulability of the robotic arm. This control scheme was implemented and tested using the MATLAB simulated PMM system. The PMM system is a planar arm with 3 DoF mounted on a differentially driven mobile platform. The results demonstrated the ability to successfully follow dual-trajectory with a different order of priority between the two trajectories while maximizing PMM manipulability measure.

CHAPTER 6
DUAL-TRAJECTORY TRACKING WITH FREE PLATFORM
TRANSLATION ALONG A TRACK

6.1 Introduction

In this chapter, a novel control scheme is introduced for tracking the trajectory of redundant mobile manipulators when the mobile platform is allowed to move freely along a specified virtual or physical track. The control scheme is capable of controlling the mobile manipulator to follow end-effector trajectory and the mobile platform track by adjusting the magnitudes and the directions of the mobile platform translations along the specified track to put the arm in a position that assists the arm to perform the task at hand. The translation of the redundant mobile manipulator over the platform track is determined by combining the mobility of the mobile platform and manipulation of the redundant arm in a single control system. The mobile platform is allowed to move forward and backward with different velocities along its track to support the end-effector in following its trajectory. The MATLAB simulated redundant Planar Mobile Manipulator (PMM) is used to implement and test the proposed control system. The results demonstrate the effectiveness of the control system in adjusting the mobile platform translations along its track to allow the arm to follow its own trajectory. As a result, this control system allows the mobile manipulator to follow both trajectories when other methods fail.

Controlling the mobile platform simultaneously with the robotic arm makes the mobile manipulator capable of performing complex tasks such as, “Open the Door,” and, “Pick Up a Moving Object while the Platform is Avoiding a Moving Obstacle.” This is because

the mobile platform gives the robotic arm not only the ability to move towards the task workspace, but also the ability to be positioned in a place in which the arm will have maximum possible manipulation. Therefore, controlling both the end-effector trajectory and mobile platform position on its track plays a crucial role in successfully executing the task at hand. This guarantees that the mobile platform brings the robotic arm to a preferred configuration that supports the arm to perform the task successfully.

6.2 Unconstrained Mobile Platform Translations Along a Track

Manipulators or robotic arms can be mounted on various types of mobile platforms that differ by the driving mechanisms. In this work, the mobile platform is a differentially driven type, (nonholonomic mobile platform). The robotic arm is assumed to have n DoF and the mobile platform has 2 DoF. Therefore, the mobile manipulator will have $(n + 2)$ DoF when combined. For detailed kinematic modeling of this mobile manipulator, refer to Chapter 3.

It is worth pointing out that the mobile platform track itself relates the mobile platforms translation (S) and orientation (ϕ) due to the nonholonomic constraint. Since the mobile platform can be controlled to follow the desired track using two consecutive movements, (rotation and translation), the mobile platform orientation angle (ϕ) can be determined by relating the mobile platform translation S to the track curvature ρ as shown in Figure 6.1. This correlation can be defined as follows:

$$\phi = \frac{S}{\rho} \quad (6.1)$$

The radius of curvature ρ of the track at point P is the radius of a circle that fits the track better than any other circle (see Figure 6.1) [77]. The radius of curvature for $y = f(x)$ can be determined as follows:

$$\rho(x) = \frac{\left|1 + \left(\frac{dy}{dx}\right)^2\right|^{\frac{3}{2}}}{\left|\frac{d^2y}{dx^2}\right|} \quad (6.2)$$

This gives an ability to find the mobile platform orientation ϕ outside the kinematic model.

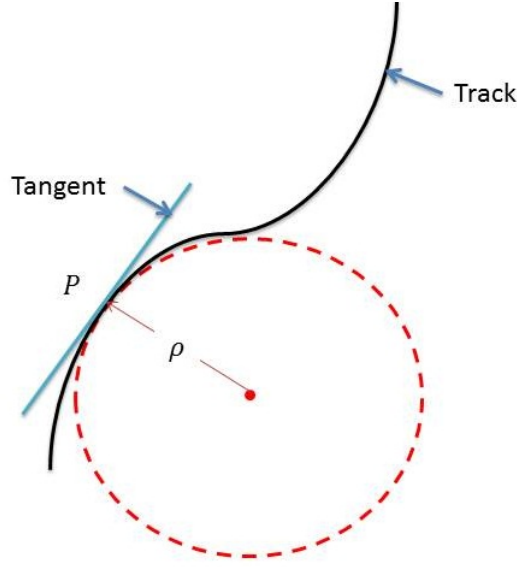


Figure 6.1: Radius of curvature ρ of the track at point P.

This leads to one extra DoR.

The following explains how to modify the Jacobian matrix J_{EP} to find S and how to use Equation (6.1) to find the related ϕ . For more convenience, Equation (3.6) is presented again here:

$$\dot{r}_{EP} = \begin{bmatrix} \dot{r}_{GE} \\ \dots \\ \dot{S} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} J_{GEA} & \vdots & J_{GEP} \\ \dots & \dots & \dots \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix}^{2 \times n} & \vdots & [J_{PP}]^{2 \times 2} \end{bmatrix} \begin{bmatrix} \dot{q}_A \\ \dots \\ \dot{S} \\ \dot{\phi} \end{bmatrix} = J_{EP}(q) \dot{q} \quad (6.3)$$

From Equation (3.4), $\dot{r}_{GE} = \frac{\partial f}{\partial q_A} \dot{q}_A + \frac{\partial f}{\partial q_P} \dot{q}_P = \begin{bmatrix} J_{GEA} & J_{GEP} \end{bmatrix} \begin{bmatrix} \dot{q}_A \\ \dot{q}_P \end{bmatrix}$

For the mobile platform $q_p = \begin{bmatrix} S \\ \phi \end{bmatrix}$

Then, $\frac{\partial f}{\partial q_P} \dot{q}_P = J_{EP}(q_P) = \begin{bmatrix} \frac{\partial f}{\partial S} & \frac{\partial f}{\partial \phi} \end{bmatrix} \begin{bmatrix} \dot{S} \\ \dot{\phi} \end{bmatrix}$

From (6.1), $\phi = \frac{1}{\rho}S$; and $\dot{\phi} = \frac{1}{\rho}\dot{S}$

$$\therefore J_{EP} \cdot \dot{q}_P = \begin{bmatrix} \frac{\partial f}{\partial S} & \frac{\partial f}{\partial \phi} \end{bmatrix} \begin{bmatrix} \dot{S} \\ \frac{1}{\rho}\dot{S} \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial S} + \frac{1}{\rho} \cdot \frac{\partial f}{\partial \phi} \end{bmatrix} \begin{bmatrix} \dot{S} \end{bmatrix} = J_{ES} [\dot{S}]$$

Therefore, in this case, $q_P = [S]$ and $J_{ES} = [\frac{\partial f}{\partial S} + \frac{1}{\rho} \cdot \frac{\partial f}{\partial \phi}]$. J_P , in Equation (3.8), is modified to $J_S = [[0]^{1 \times n} \quad 1]$. Therefore, Equation (6.3) can be rewritten as follows:

$$\dot{r}_{ES} = \begin{bmatrix} \dot{r}_{GE} \\ \dots \\ \dot{S} \end{bmatrix} = \begin{bmatrix} J_{GE_A} & \vdots & J_{ES} \\ \dots & \dots & \dots \\ & & J_S \end{bmatrix} \begin{bmatrix} \dot{q}_A \\ \dots \\ \dot{S} \end{bmatrix} = J_{AS}(q_{AS}) \dot{q}_{AS} \quad (6.4)$$

where $\dot{r}_{ES} \in \mathbb{R}^{m+1}$ and $J_{AS}(q_{AS}) \in \mathbb{R}^{(m+1) \times (n+1)}$.

Once the mobile platform translation along the track is determined, the mobile platform orientation can be determined using Equation (6.1). In this predefined-translation case, the mobile manipulator trajectory has to be fully defined. The only difference between the baseline case of tracking mobile manipulator trajectory (refer to Section 3.4) and the predefined-translation case is in the mobile base's orientation. For the baseline case, the mobile base's orientation angle is determined in the kinematics model of the mobile manipulator, while in the predefined-translation case, the mobile base's orientation angle is calculated using Equation (6.1). Moreover, the Jacobian matrix dimension is reduced by one, which translates into an extra DoR. The predefined-translation case is taken as a reference case against which two other cases will be compared.

The role of the mobile platform is not only to extend the manipulator's workspace but also to position the manipulator in a way that supports the execution of the manipulation

task. This is one of the main contributions of this work. The following section demonstrates how to allow the mobile platform to change the magnitude and direction of its translations along the track to support the manipulation task.

6.2.1 Unconstrained Mobile Platform Translations

In order to allow the mobile platform to translate freely along a track, only the track without waypoints should be specified as shown in Figure 6.2. The translation magnitude of the mobile platform is unknown, and it is left to the controller to find it out according to the robotic arm configurations. The only predefined velocity input is the end-effectors

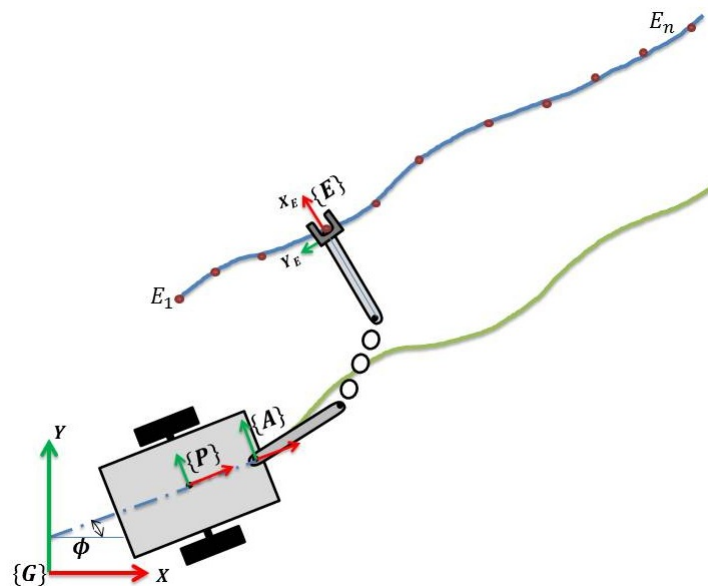


Figure 6.2: Mobile manipulator predefined end-effector trajectory with the predefined mobile platform's track.

Cartesian velocities. As a result, Equation (6.4) can be rewritten as follows:

$$\dot{r}_{GE} = \begin{bmatrix} J_{GEA} & J_{ES} \end{bmatrix} \begin{bmatrix} \dot{q}_A \\ \dot{S} \end{bmatrix} = J(q_{AS}) \dot{q}_{AS} \quad (6.5)$$

where $J(q_{AS})$ relates the arm joint velocities \dot{q}_A and the mobile platform velocity \dot{S} along the track to the Cartesian velocities of the end-effector, $J(q_{AS}) \in \mathbb{R}^{m(n+1)}$.

Once the translation magnitude of the mobile platform along the track is determined, the mobile platform orientation angle can be calculated using Equation (6.1). Therefore, to use Equation (6.5), the trajectory of the end-effector has to be completely predefined. This means that in each and every time instance, the pose (position and orientation) of the end-effector is known. However, the mobile platform trajectory (track) is partially predefined. This means that the mobile platform trail is specified (without the exact mobile platform position on it) at any specific time instance as shown in Figure 6.2.

The main role of the proposed control scheme is to follow the mobile manipulator trajectory. Since the mobile manipulator is redundant, the degrees of redundancy can be used to study the effect of the optimization criterion on the control performance. The system's manipulability measure maximization is implemented for this purpose.

6.2.2 Maximization of the System's Manipulability Measure

The inverse kinematic solution of Equation (6.5) can be determined using the GPM [78]:

$$\dot{q}_{AS} = J^\# \dot{r}_{GE} + (I - J^\# J) H \quad (6.6)$$

where $J^\# = J^T (J J^T)^{-1}$ is the pseudo-inverse of J , and \dot{r}_{GE} is the desired velocity vector of the task variables. The first right hand side term of Equation (6.6) is the LN solution [73], which provides \dot{r}_{GE} with minimum Euclidean norm. The second term is the orthogonal projection of the arbitrary vector H onto the remaining subspace.

Manipulability measure [9] of a robotic arm can be defined as follows:

$$w = \sqrt{(\det (J J^T))} \quad (6.7)$$

Vector H in Equation (6.6) is used as the gradient of the manipulability measure of the system that can be defined as:

$$H = k \nabla w(\theta_1, \dots, \theta_n, S, \phi) = k \left[\frac{\partial w}{\partial \theta_1} \quad \dots \quad \frac{\partial w}{\partial \theta_n} \quad \frac{\partial w}{\partial S} \quad \frac{\partial w}{\partial \phi} \right]^T \quad (6.8)$$

where k is a positive constant.

The other two tested cases other than the predefined-translation case are related to Equation (6.6). These two cases are:

1. The LN solution undefined-translation case: This case is abbreviated as LN undefined-translation case. In this case, free translations of the mobile platform along its predefined track is computed using the first right hand side term of Equation (6.6). This can be accomplished by letting $H = 0$ by making $k = 0$. This means that Equation (6.6) is modified to:

$$\dot{q}_{AS} = J^\# \dot{r}_{GE} \quad (6.9)$$

2. The Manipulability Measure (MM) optimized undefined-translation case: This case is abbreviated as MM undefined-translation case. In this case, the whole Equation (6.6) is used to determine free translations of the mobile platform along its predefined track. This will be explained in more detail in Section 6.4.

6.3 Implementation Example

For the purpose of demonstrating the effectiveness of the proposed method in a simple redundant mobile manipulator, a redundant PMM is simulated and used to test and evaluate the proposed controller. However, this controller can be applied to any nonholonomic wheeled mobile manipulator of n DoFs. The PMM consists of a planar robotic arm with three revolute joints mounted on top of a nonholonomic mobile platform as shown in Figure 6.3. The base

of the robotic arm is placed along the X-axis of the mobile platform at a distance l_A from the mobile platform frame P , (frame P is at the middle of the wheel axle).

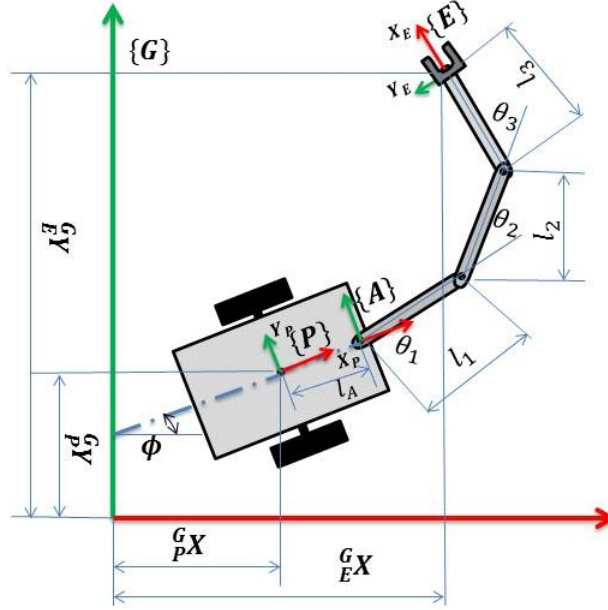


Figure 6.3: Planar mobile manipulator with a 3 DoF planar robotic arm and a nonholonomic mobile platform.

6.3.1 PMM Jacobian

Referring to Equation (6.5), the configuration of the PMM is $q_{AS} = [q_A \ S]^T \in R^2$, with $q_A = [\theta_1 \ \theta_2 \ \theta_3]^T \in R^3$. The complete state variables of the PMM are:

$$\dot{r}_{GE} = \begin{bmatrix} G_E \dot{x} \\ G_E \dot{y} \end{bmatrix} = J \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{S} \end{bmatrix} = \begin{bmatrix} J_{GEA} & \vdots & J_{ES} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{S} \end{bmatrix} \quad (6.10)$$

Refer to Figure 6.3 for variables' definition. For the end-effector's trajectory, the position of frame E relative to frame G is as follows:

$$\begin{bmatrix} \frac{G}{E}x \\ \frac{G}{E}y \end{bmatrix} = \begin{bmatrix} \frac{G}{P}X + l_A C_\phi + l_1 C_{\phi\theta_1} + l_2 C_{\phi\theta_1\theta_2} + l_3 C_{\phi\theta_1\theta_2\theta_3} \\ \frac{G}{P}Y + l_A S_\phi + l_1 S_{\phi\theta_1} + l_2 S_{\phi\theta_1\theta_2} + l_3 S_{\phi\theta_1\theta_2\theta_3} \end{bmatrix} \quad (6.11)$$

where $C_{lmn} = \cos(l + m + n)$ and $S_{lmn} = \sin(l + m + n)$. The arm Jacobian becomes:

$$J_{GEA} = \begin{bmatrix} -l_1 S_{\phi\theta_1} - l_2 S_{\phi\theta_1\theta_2} - l_3 S_{\phi\theta_1\theta_2\theta_3} & -l_2 S_{\phi\theta_1\theta_2} - l_3 S_{\phi\theta_1\theta_2\theta_3} & -l_3 S_{\phi\theta_1\theta_2\theta_3} \\ l_1 C_{\phi\theta_1} + l_2 C_{\phi\theta_1\theta_2} + l_3 C_{\phi\theta_1\theta_2\theta_3} & l_2 C_{\phi\theta_1\theta_2} + l_3 C_{\phi\theta_1\theta_2\theta_3} & l_3 C_{\phi\theta_1\theta_2\theta_3} \end{bmatrix} \quad (6.12)$$

The mobile platform Jacobian is:

$$J_{GEP} = \begin{bmatrix} C_\phi & -l_A S_\phi - l_1 S_{\phi\theta_1} - l_2 S_{\phi\theta_1\theta_2} - l_3 S_{\phi\theta_1\theta_2\theta_3} \\ S_\phi & l_A C_\phi + l_1 C_{\phi\theta_1} + l_2 C_{\phi\theta_1\theta_2} + l_3 C_{\phi\theta_1\theta_2\theta_3} \end{bmatrix} \quad (6.13)$$

Therefore,

$$J_{ES} = \begin{bmatrix} C_\phi + \frac{1}{\rho} (-l_A S_\phi - l_1 S_{\phi\theta_1} - l_2 S_{\phi\theta_1\theta_2} - l_3 S_{\phi\theta_1\theta_2\theta_3}) \\ S_\phi + \frac{1}{\rho} (l_A C_\phi + l_1 C_{\phi\theta_1} + l_2 C_{\phi\theta_1\theta_2} + l_3 C_{\phi\theta_1\theta_2\theta_3}) \end{bmatrix} \quad (6.14)$$

6.3.2 Track Radius of Curvature (ρ)

The mobile platform track is a sinusoidal trajectory in a form of $y = d + a \sin(\pi x / f)$.

The radius of curvature ρ of the track is calculated using Equation (6.2).

6.3.3 Manipulability Measure Maximization of the PMM System

Equation (6.6) represents the inverse kinematic solution with the GPM. For the PMM, the equation can be modified as follows:

$$\dot{q}_{AS} = J_{AS}^\# \dot{r}_{GE} + \left(I - J_{AS}^\# J_{AS} \right) H \quad (6.15)$$

where H is as follows:

$$H = k\nabla w = k \left[\frac{\partial w}{\partial \theta_1} \quad \frac{\partial w}{\partial \theta_2} \quad \frac{\partial w}{\partial \theta_3} \quad \frac{\partial w}{\partial S} \right]^T$$

and w :

$$w = \sqrt{\det(J_{AS} J_{AS}^T)}$$

6.4 Simulation Results and Discussion

A MATLAB simulation was developed for the PMM to test the proposed controller. In this simulation, different trajectories were used to evaluate the performance and the effectiveness of this controller.

The chosen PMM geometric data are (in mm): $l_1 = l_2 = l_3 = 600$, $l_A = 400$, wheel base $b = 800$ and wheel radius $r = 200$. The initial mobile platform configuration is $[x_P \ y_P \ \phi_P]^T = [0 \ 0 \ 40^0]^T$. The robotic arm has an initial configuration (joint angles in degrees): $[\theta_1 \ \theta_2 \ \theta_3] = [-90 \ 45 \ 30]^0$. The sinusoidal end-effector trajectory and mobile platform track are in the form of $y = d + a \sin(\pi x/f)$ where a is the amplitude, f is the frequency and d is the initial position of the end-effector of the mobile platform. For the end-effector trajectory: $a = -2300$ and $f = 2500$. In the case of the mobile platform track: $a = 650$ and $f = 2400$.

Three simulations for three cases were performed to demonstrate the effectiveness and the performance of the proposed controller. Refer to Section 6.2 for the following cases:

1. The predefined-translation case.
2. The LN undefined-translation case. That can be done by using the first right hand side term in Equation (6.6) where ($H = 0$). This was accomplished by setting $k = 0$ in Equation (6.8).
3. The MM undefined-translation case. This was accomplished using all terms in Equation (6.6) and $k = 1$.

All three cases are described in the following three subsections.

6.4.1 Predefined-Translation Case

As previously mentioned for this case, the end-effectors poses and mobile platforms poses are both predefined as trajectory waypoints. The PMM has to follow both trajectories waypoints. Therefore, the translation and rotation of the mobile platform are known. To force the mobile platform to follow the predefined mobile platforms track via the waypoints, the mobile platform translations were mapped one to one. Then the mobile platform orientation angle can be determined using Equation (6.1). For this case, the state variables vector of the PMM is as follows:

$$\begin{bmatrix} {}^G_E \dot{x} \\ {}^G_E \dot{y} \\ {}^G_P \dot{S} \end{bmatrix} = J_{AS} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{S} \end{bmatrix} \quad (6.16)$$

where

$$J_{AS} = \begin{bmatrix} -l_1 S_{\phi\theta_1} - l_2 S_{\phi\theta_1\theta_2} - l_3 S_{\phi\theta_1\theta_2\theta_3} & -l_2 S_{\phi\theta_1\theta_2} - l_3 S_{\phi\theta_1\theta_2\theta_3} & -l_3 S_{\phi\theta_1\theta_2\theta_3} & J_{AS1} \\ l_1 C_{\phi\theta_1} + l_2 C_{\phi\theta_1\theta_2} + l_3 C_{\phi\theta_1\theta_2\theta_3} & l_2 C_{\phi\theta_1\theta_2} + l_3 C_{\phi\theta_1\theta_2\theta_3} & l_3 C_{\phi\theta_1\theta_2\theta_3} & J_{AS2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$J_{AS1} = \left[C_{\phi} + \frac{1}{\rho} (-l_A S_{\phi} - l_1 S_{\phi\theta_1} - l_2 S_{\phi\theta_1\theta_2} - l_3 S_{\phi\theta_1\theta_2\theta_3}) \right]$$

$$J_{AS2} = \left[S_{\phi} + \frac{1}{\rho} (l_A C_{\phi} + l_1 C_{\phi\theta_1} + l_2 C_{\phi\theta_1\theta_2} + l_3 C_{\phi\theta_1\theta_2\theta_3}) \right]$$

refer to Equation (6.14).

The orientation of the PMM can be determined using Equation (6.1). The maximization of the manipulability measure optimization criterion was used in this case. The inverse kinematic solution can be expressed as follows:

$$\dot{q}_{AS} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{S} \end{bmatrix} = J_{AS}^{\#} \begin{bmatrix} {}^G_E \dot{x} \\ {}^G_E \dot{y} \\ {}^G_P \dot{S} \end{bmatrix} + \left(I - J_{AS}^{\#} J_{AS} \right) H \quad (6.17)$$

6.4.2 LN Undefined-Translation Case

For the LN undefined-translation case, in contrast to the previous case, the end-effector poses over its trajectory were fully predefined and only the orientation of the mobile base along the track was known. Therefore, to determine the full poses of the mobile platform, the locations of the mobile platform on the track had to be determined. The locations of the mobile platform over the track was determined by finding the translation magnitude and direction of the mobile platform at each time step. The translation magnitude of the mobile platform along the mobile platform track varies according to the LN solution which provides the joint angles q_{AS} with minimum Euclidean norm ($\min \| q_{AS} \|$). By modifying Equation (6.16), this solution can be expressed as follows:

$$\dot{r}_{GE} = \begin{bmatrix} {}^G_E \dot{x} \\ {}^G_E \dot{y} \end{bmatrix} = J_{AS} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{S} \end{bmatrix} \quad (6.18)$$

where J_{AS} in this case is as follows:

$$J_{AS} = \begin{bmatrix} -l_1 S_{\phi\theta_1} - l_2 S_{\phi\theta_1\theta_2} - l_3 S_{\phi\theta_1\theta_2\theta_3} & -l_2 S_{\phi\theta_1\theta_2} - l_3 S_{\phi\theta_1\theta_2\theta_3} & -l_3 S_{\phi\theta_1\theta_2\theta_3} & J_{AS1} \\ l_1 C_{\phi\theta_1} + l_2 C_{\phi\theta_1\theta_2} + l_3 C_{\phi\theta_1\theta_2\theta_3} & l_2 C_{\phi\theta_1\theta_2} + l_3 C_{\phi\theta_1\theta_2\theta_3} & l_3 C_{\phi\theta_1\theta_2\theta_3} & J_{AS2} \end{bmatrix}$$

$$J_{AS1} = \left[C_{\phi} + \frac{1}{\rho} (-l_A S_{\phi} - l_1 S_{\phi\theta_1} - l_2 S_{\phi\theta_1\theta_2} - l_3 S_{\phi\theta_1\theta_2\theta_3}) \right]$$

$$J_{AS2} = \left[S_{\phi} + \frac{1}{\rho} (l_A C_{\phi} + l_1 C_{\phi\theta_1} + l_2 C_{\phi\theta_1\theta_2} + l_3 C_{\phi\theta_1\theta_2\theta_3}) \right]$$

In this case, the first right hand term from Equation (6.6) is used. Therefore, the solution is as follows:

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{S} \end{bmatrix} = J_{AS}^{\#} \begin{bmatrix} {}^G_E \dot{x} \\ {}^G_E \dot{y} \end{bmatrix} \quad (6.19)$$

6.4.3 MM Undefined-Translation Case

For the MM undefined-translation case, all terms in Equation (6.6) were used. The solution was determined according to the following equation:

$$\dot{q}_{AS} = J_{AS}^{\#} \begin{bmatrix} {}^G_E \dot{x} \\ {}^G_E \dot{y} \end{bmatrix} + \left(I - J_{AS}^{\#} J_{AS} \right) H \quad (6.20)$$

6.4.4 Simulation Results and Discussion

The PMM task was to follow the end-effector and the mobile platform trajectories. Figure 6.4 shows the sequence of the PMM locations for the end-effector and the mobile platform along the end-effector trajectory and mobile platform track for the three cases. For more clarity, only the commanded end-effector trajectory and the actual end-effector trajectories for the three cases were plotted in Figure 6.4. In the reference case, which is the predefined-

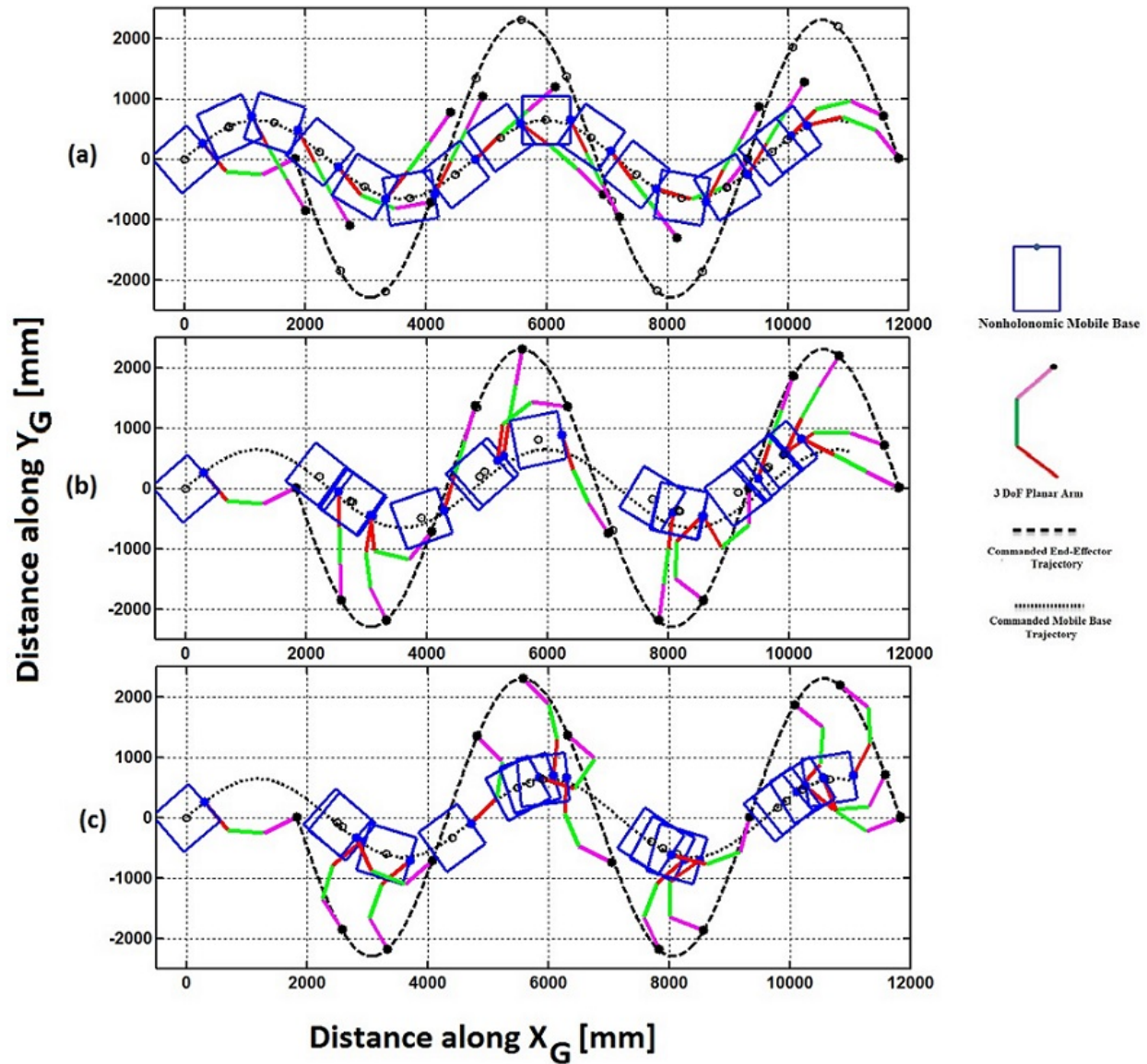


Figure 6.4: Tracked trajectories of the PMM for cases: (a) Predefined-translation. (b) LN undefined-translation. (c) MM undefined-translation.

translation case (Figure 6.4-a), the PMM failed to track both trajectories in four regions as shown in Figure (6.5); whereas in the MM undefined-translation case (Figure 6.4-c) and the LN undefined-translation case (Figure 6.4-b), the PMM was able to follow the end-effector trajectory while keeping the mobile platform on the track. This demonstrates the effectiveness and the performance of the proposed control system. As a result of unconstraining the translations of the mobile platform along its track, the mobile manipulator was able to suc-

cessfully execute the tracking task that could not be done with the conventional trajectory tracking techniques shown in the reference case.

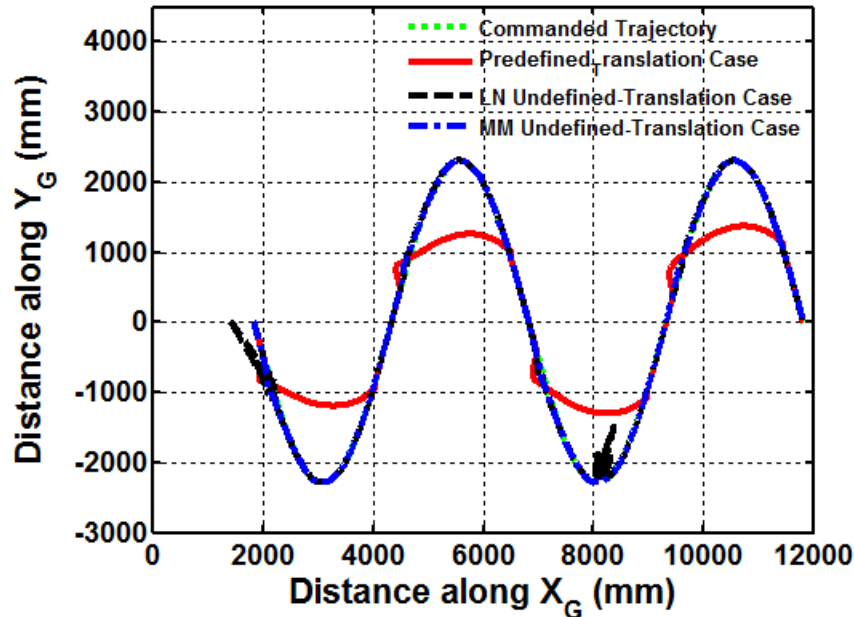


Figure 6.5: Commanded and actual end-effector trajectories for the predefined and undefined translation cases.

In the LN undefined-translation case (Figure 6.4-b), the PMM, most of the time, was able to follow the end-effector trajectory except in two occasions in which the PMM was unstable to comply due to singularity. This presented the effectiveness of using PMM system manipulability measure maximization for allowing the system to be away as much as possible from singular configurations.

The mobile platform velocities along its track in the three cases were demonstrated in Figure 6.6. As it can be seen from Figure 6.6, the mobile platform velocities for undefined-translation cases were not constant compared to the predefined-translation case. Mobile platform velocities in LN undefined-translation case were generally lower compared to the MM undefined-translation case. This was expected, since the LN solutions tend to minimize the joint angle velocity. On the other hand, in the MM undefined-translation case, the

solutions were pushing for maximizing the PMM manipulability measure. Figure 6.6 shows

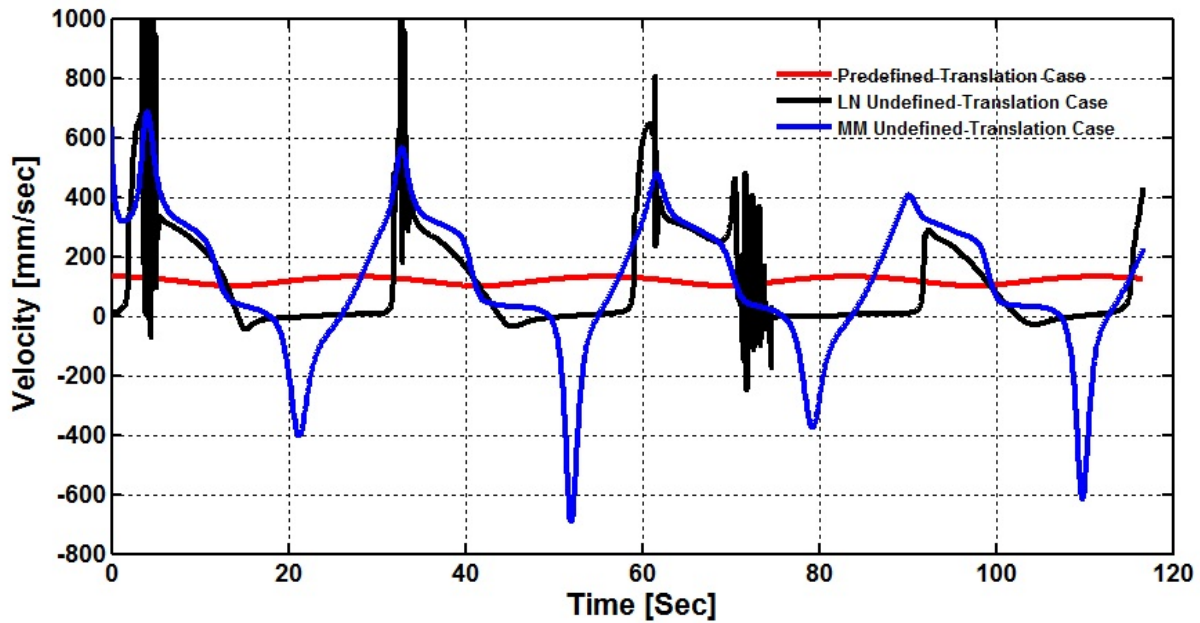


Figure 6.6: PMM platform velocity in each time step for the predefined, MM and LN undefined translations cases.

the ability that the proposed control gives to the mobile platform. The mobile platform was able to move backwards and forward with different velocities that allowed placing the mobile platform at positions in which the manipulator was able to follow its trajectory.

The PMM manipulability measures for the two cases (LN undefined-translation and MM undefined-translation) are shown in Figure 6.7. The third case (predefined-translation) manipulability measure is at different scale due to the difference on the Jacobian matrix dimensions. In this figure, the manipulability measure of the PMM system in MM undefined-translation case was maximized. Using this method, the PMM not only had maximum possible manipulability measure at all times, but also prevented the system from encountering a singular configuration. This allowed the MM undefined-translation case to follow both the trajectory and the track where the other cases failed. To compare the three cases manipulability measure, the manipulability measure of the robotic arm is used. Figure 6.8 shows robotic arm manipulability measure for the three cases. The arm manipulability measure is

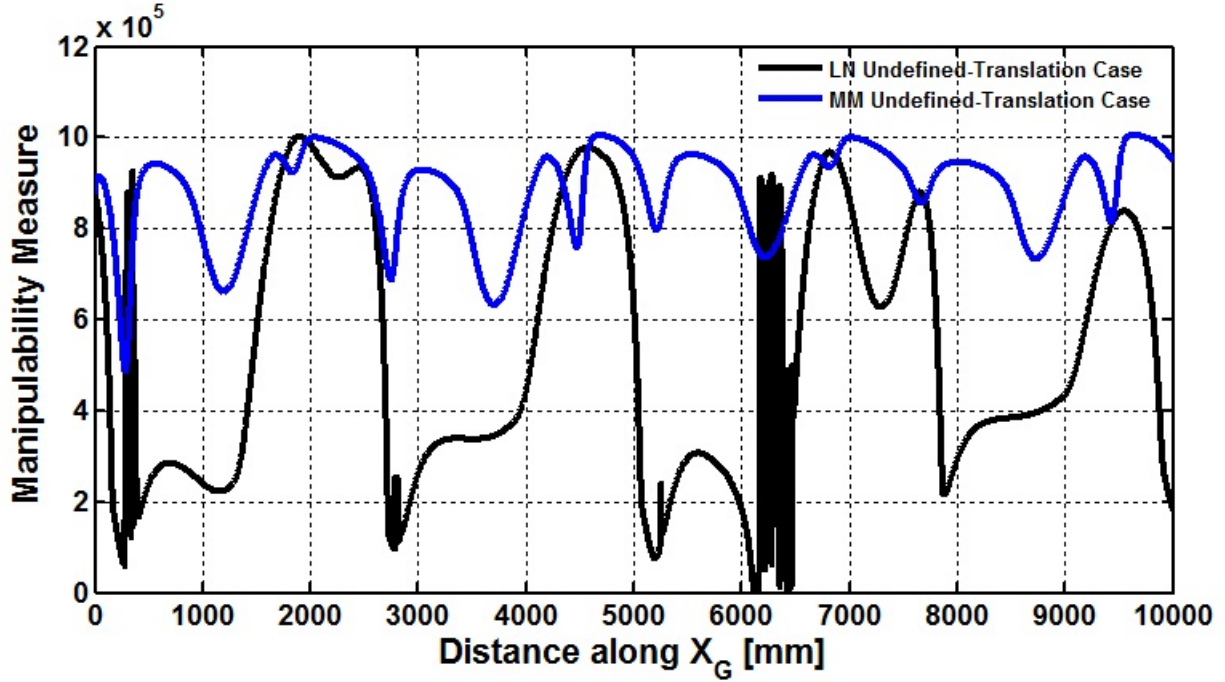


Figure 6.7: PMM manipulability measure for the predefined, MM and LN undefined translation cases.

used as an indicator of the whole system manipulability measure since the Jacobian matrix of the robotic arm has same dimensions for all the cases. The figure shows that the arm manipulability measure of the MM undefined-translation case has the highest manipulability measure possible most of the time.

In Equation (6.6), the LN solutions are obtained when k in Equation (6.8) equals to zero, while in the other undefined-translation case $k = 3 \times 10^{-6}$. Therefore, a trade off between the manipulability measure and the mobile platform velocities can be achieved by changing the value of k .

6.5 Summary

In this chapter, a new control scheme is introduced for mobile manipulator trajectory tracking. This control scheme is capable of tracking the end-effector trajectory and mobile platform track. The main contribution of this work was in adjusting the mobile platform

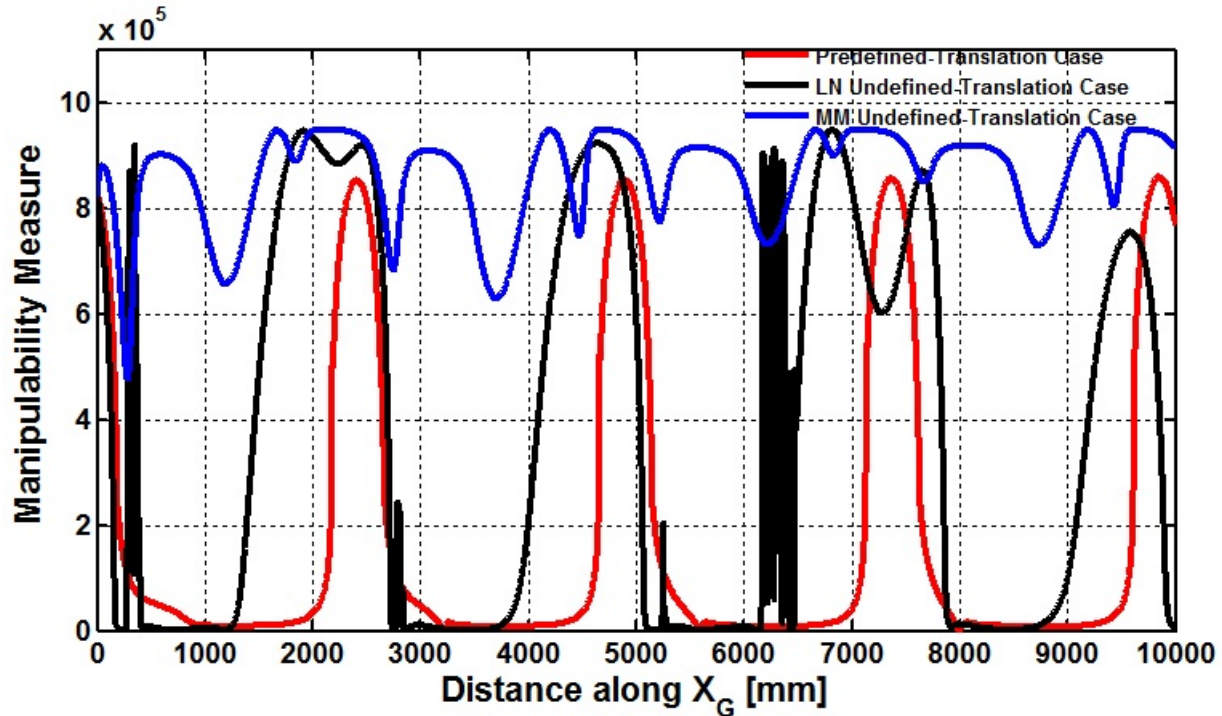


Figure 6.8: Arm manipulability measure for the predefined, MM and LN undefined translation cases

translations along its specified track to support the manipulator task. The task was to keep the mobile platform on its prespecified track and, at the same time, keep the end-effector on its independent trajectory. Using the proposed control technique, the mobile platform was given the ability to move forward and backward with different velocities along its track to allow the mobile manipulator to successfully execute its task. The results showed that this controller was able to successfully track both the end-effector trajectory and the mobile platform track when other methods failed. Reference case of trajectory tracking was compared with the LN undefined-translation and MM undefined-translation cases. Although, this work was implemented in simulation, this algorithm will be implemented onto hardware, specifically, a Wheelchair Mounted Robotic Arm System (WMRA).

CHAPTER 7

DEVELOPMENT, TESTING AND RESULTS OF POSE ESTIMATION CORRECTION OF MOBILE PLATFORM WITH HIGH UNCERTAINTIES

7.1 Introduction

Localization, i.e. estimating the position and orientation (pose) of a mobile robot from sensory data, is an active problem in autonomous mobile robots. A mobile robot has to accurately localize itself relative to its surrounding environment at all times in order to navigate safely and efficiently. Without an accurate localization, autonomous navigation and obstacle avoidance will be impossible [79, 80]. In literature, there are varieties of sensors, techniques and models that have been employed to handle this problem¹.

The dead reckoning method, commonly referred to as odometry, is the common practice for localizing mobile robots. In this method, the current robot pose is computed incrementally by knowing the previous pose and a measure of the movement that is carried out by the robot. It is well known that odometry is subject to many sources of measurement errors which make it impossible to maintain an accurate estimate of robot pose over long paths. Therefore, process of measuring and correcting systems' inaccuracies and sensors' errors is crucial for increasing the precision of the localization procedure.

Recently, the assistive navigation systems for individuals with disabilities became a new area of research in mobile robotics [82]. Many studies have been undertaken to design smart wheelchairs with different levels of autonomy to assist people with disabilities in performing their activity of daily living (ADL).

¹A conference paper was published related to the material presented in this Chapter [81]. Permission is included in Appendix A.

Power wheelchairs are designed for manual operation which depends on human control and perception [83]. These wheelchairs lack precise motion hardware, such as built-in encoders, and precise motion controllers. As stated by Simpson et al, in[84], most smart wheelchairs that have been developed from power wheelchairs need major modifications to operate properly. These modifications involve adding sensors and by-passing the wheelchair’s controller to directly control the wheelchair’s motors. As a result, a wheelchair without any major modification is determined to be a mobile platform that is highly inaccurate for precise motion or autonomous operation.

In the literature, several works have been carried out to evaluate the accuracy of the power wheelchairs. For example, in [82], Horn et al. carried out an experiment to estimate the odometric error for a smart wheelchair, (VAHM). They set position estimation margins, (location and orientation) in order for the wheelchair to follow the programmed trajectories properly. They had to use two more sensors (sonar and camera sensors) to achieve the required margins. Wheelchair localization was the core process for enhancing the performance in relation to autonomy and mobility [85].

7.2 Mobile Platform Pose Estimation Methods

Two methods are used to estimate the mobile platform location and heading: encoder-based odometry and ICP-based odometry. These methods are implemented using two inexpensive sensors which are wheel encoders and vision sensor. The wheel encoders are ball bearing optical shaft encoders H5 from US Digital [86]. The vision sensor is a Microsoft XBOX 360 Kinect camera [87]. This camera is able to capture 30 frames per second with a resolution of 640×480 pixels. For each pixel, the Kinect measures the associated depth information by projecting a pattern of infrared lights and then use stereo triangulation to calculate the depth. Figure 7.1 shows a flowchart that illustrates the steps of the two pose estimation methods. In the encoder-based odometry, measurements of optical encoders

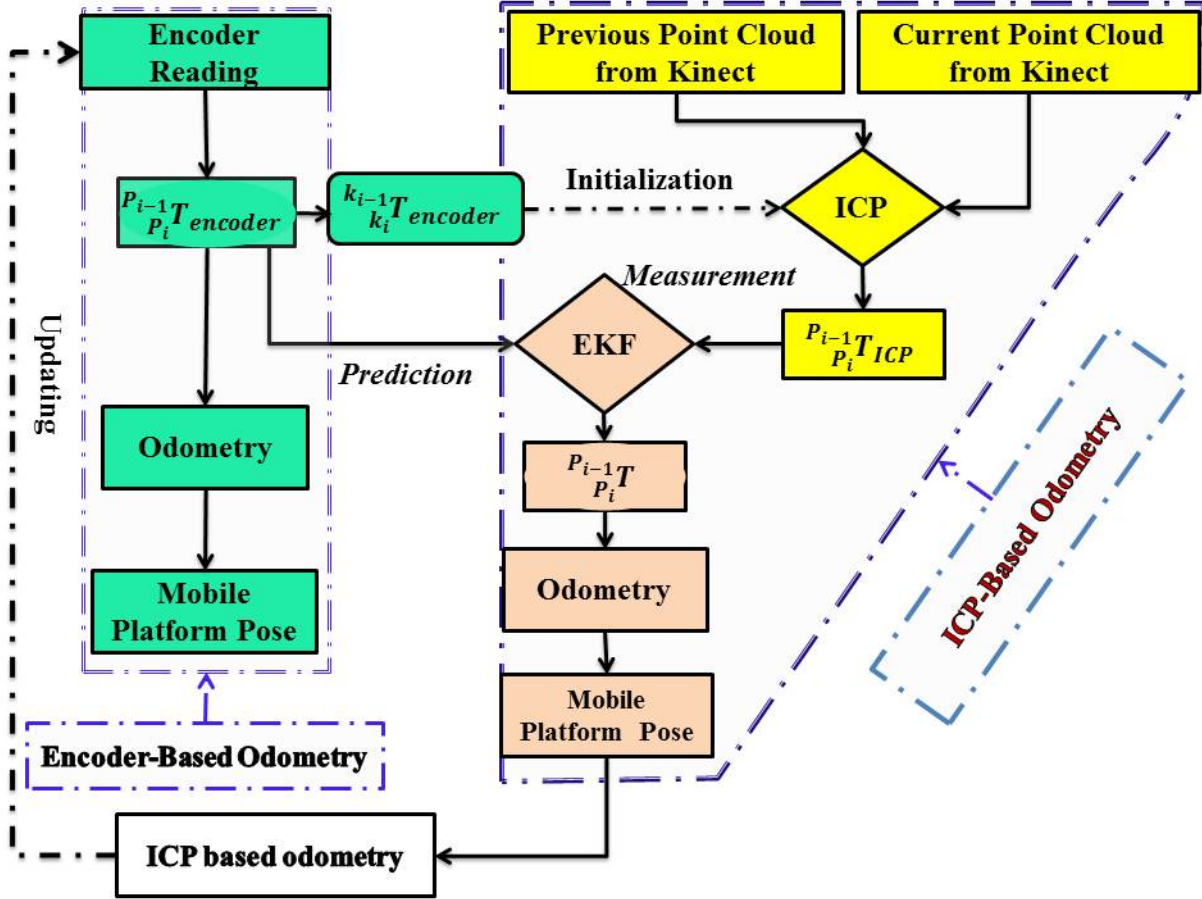


Figure 7.1: Flowchart of two estimation methods.

mounted on each driving wheel were used for calculating the transformation matrix between two consecutive platform frames $P_{i-1}T_{P_i}$ (refer to Figure 7.2). We refer to this transformation matrix as the local transformation matrix. The same transformation matrix was determined by applying the ICP algorithm on two consecutive overlapped Kinect point clouds. The Extended Kalman Filter (EKF) was used to fuse the two local transformation matrices to get the optimized local transformation matrix which was used to calculate the ICP-based odometry. Then the encoder-based odometry was updated with ICP-based odometry. The following is a detailed explanation of these two methods.

7.2.1 Encoder-Based Odometry

Encoders are used to track the mobile platform global pose $[X_i, Y_i, \phi_i]^T$ by measuring the angular displacement of the right wheel θ_{r_i} and the left wheel θ_{l_i} , where X_i and Y_i are the X and Y global coordinates of the mobile platform, respectively, and ϕ_i is the mobile platform orientation angle. Throughout this chapter, the subscript i means the i^{th} instance in the mobile platform motion. These two angular displacements are computed using the encoders' readings from both wheels. The distance traveled by the left and right wheels are $L_i = w_r \theta_{r_i}$ and $R_i = w_r \theta_{l_i}$ respectively, where w_r is the wheel radius in meters. The pose of the mobile platform $[X_i, Y_i, \phi_i]^T$ relative to a global frame G can be computed by using Equation (3.19). For more details, refer to Section 3.5.2.

$$\begin{bmatrix} X_i \\ Y_i \\ \phi_i \end{bmatrix} = \begin{bmatrix} X_{i-1} \\ Y_{i-1} \\ \phi_{i-1} \end{bmatrix} + \begin{bmatrix} r_i \left[\sin \phi_{i-1} - \sin \left(\phi_{i-1} + \frac{R_i - L_i}{w_b} \right) \right] \\ r_i \left[\cos \left(\phi_{i-1} + \frac{R_i - L_i}{w_b} \right) - \cos \phi_{i-1} \right] \\ \frac{R_i - L_i}{w_b} \end{bmatrix} \quad (7.1)$$

The symbol r_i represents the instantaneous radius of curvature, where $r_i = \frac{w_b}{2} \left(\frac{L_i + R_i}{L_i - R_i} \right)$ and w_b is the wheel base.

7.2.2 ICP-Based Optimized Odometry

As it is illustrated in Figure 7.1, the ICP-based optimized odometry was obtained in two steps as follows:

7.2.2.1 ICP-Based Local Transformation Matrix

The alignment of two point clouds, also referred to as registration, means finding the transformation matrix $\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$ (rotation matrix \mathbf{R} and translation vector \mathbf{t}) that will transform one data set to the other. Thus, given two data sets, one is a target data set denoted

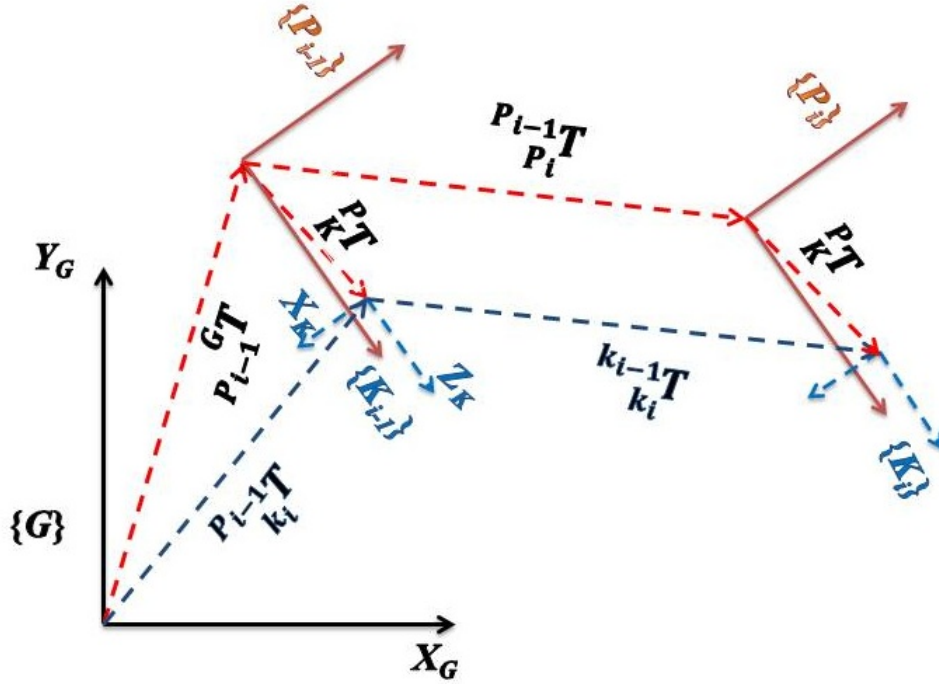


Figure 7.2: Platform and Kinect coordinate frames relative to the global coordinate frames.

as $M \triangleq \{\vec{m}_i\}_{i=1}^{N_m}$ with N_m points, and the other is a source data set denoted as $P \triangleq \{\vec{p}_i\}_{i=1}^{N_p}$ with N_p points. The goal is to find the transformation parameters between the two point sets in which the error between the transformed source data and the closest points in the target data will be minimum. This problem statement is presented in [88] using the following equation:

$$R, \vec{t}, j \in \{1, 2, \dots, N_m\} \left(\sum_{i=1}^{N_p} \left\| \mathbf{R}\vec{p}_i + \vec{t} - \vec{m}_j \right\|_2^2 \right) \quad (7.2)$$

Knowing the transformation matrix between each two consecutive frames (see Figure 7.2) by applying a registration process, the pose of a mobile platform can be tracked. Using the ICP algorithm, the registration process is usually composed of two stages: coarse and fine alignments. The coarse alignment is implemented for roughly aligning the two frames by using, for example, feature matching or encoder measurements. This makes the ICP algorithm faster for the fine alignment and avoids local minima.

In this work, registration using the ICP algorithm was applied to the two consecutive point clouds captured by the Kinect. The following is to illustrate how the local transformation matrix was determined based on the ICP algorithm. Figure 7.3 illustrates the steps of the registration process with Kinect RGB images. Also refer to Figure 7.2 for the equations' variables.

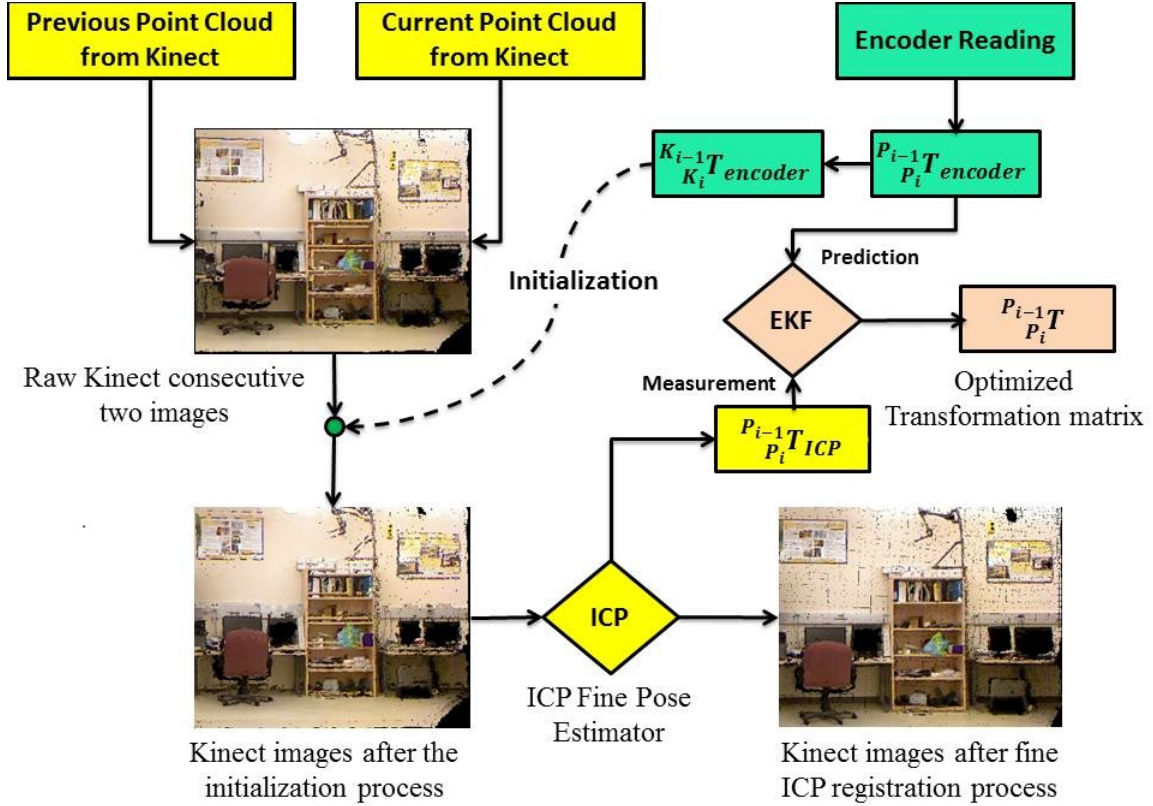


Figure 7.3: Registration process and Kalman Filter with Kinect RGB images.

First, point clouds associated with Kinect coordinate frames K_{i-1} and K_i were captured. They were initially aligned using the local transformation matrix of the current Kinect frames relative to the previous Kinect frames ${}^{k_{i-1}}T_{k_i} encoder$ that can be determined using Equation (7.3):

$${}^{k_{i-1}}T_{k_i} encoder = {}^P T^{-1} {}^{P_{i-1}} T_{P_i} encoder {}^P T \quad (7.3)$$

where transformation matrix ${}^{P_i}T_{encoder}^{P_{i-1}}$ can be calculated from Equation (3.19) by letting $X_{i-1} = Y_{i-1} = \phi_{i-1} = 0$, where P_kT is the transformation matrix of the Kinect frame relative to the mobile platform frame P . The perfection of the initial alignment depends on how accurate the wheel encoders are as shown in Figure 7.3. As can be noticed from Kinect images, after the initialization process in Figure 7.3 there is a mis-alignment between them that indicates an error on the encoder based transformation matrix between the two mobile platform consecutive frames. The initial misalignment was obviously noticed during rotational motion more than the translational motion. This indicates that the encoder-based odometry is more accurate during translation compared with rotation. Next, the ICP algorithm was applied on the encoder-aligned Kinect point clouds to get a 6 DoF fine-alignment transformation matrix, T_{ICP} . This transformation matrix is a compensation for any error in the local transformation matrix determined from encoder's measurements. The overall transformation matrix between the previous and current Kinect point cloud is:

$${}^{k_{i-1}}T_{ICP}^{k_i} = T_{ICP} {}^{k_{i-1}}T_{encoder}^{k_i} \quad (7.4)$$

The local transformation matrix between two consecutive mobile platform frames P_{i-1} and P_i can then be determined as follows (refer to Figure 7.2):

$${}^{P_{i-1}}T_{ICP}^{P_i} = {}^P_kT {}^{k_{i-1}}T_{ICP}^{k_i} {}^P_kT^{-1} \quad (7.5)$$

The global mobile platform pose can be calculated by using the following:

$${}^G_{P_i}T = {}^G_{P_{i-1}}T \cdot {}^{P_{i-1}}T_{ICP}^{P_i} \quad (7.6)$$

The previous global mobile platform pose is named as ICP based odometry. The accuracy of the local transformation depends on how accurate the registration process is. The ICP

algorithm sometimes fails to align Kinect point cloud, which leads to inaccurate estimation of the local transformation matrix. This can be due to sensor noise and the nature of this algorithm. To smooth the mobile platform pose estimation, a sensor fusion algorithm is used. The Kalman filter is used to fuse the measurements from wheel encoders and ICP algorithm.

7.2.2.2 ICP-Based Optimized Odometry

As shows in Figure 7.1, the Extended Kalman Filter (EKF) is applied to fuse the two measurements of the local transformation matrices determined from encoder-based odometry, ${}^{P_i}T_{encoder}^{P_{i-1}}$, and the ICP algorithm, ${}^{P_i}T_{ICP}^{P_{i-1}}$. Figure 7.4 shows a flowchart of using the EKF to fuse the two local transformation matrices.

The local transformation matrices (${}^{P_i}T_{encoder}^{P_{i-1}}$, and ${}^{P_i}T_{ICP}^{P_{i-1}}$) are relative to the mobile platform local frame. This means that the previous state vector has no effect on the current state vector. In this work, the EKF filter deals with the following model:

$$\begin{cases} S_i = f(0, u_i, w_i) \\ y_i = h(S_i) + v_i \end{cases} \quad (7.7)$$

where $S_i = [\Delta X_i \ \Delta Y_i \ \Delta Z_i \ \Delta \alpha_i \ \Delta \beta_i \ \Delta \gamma_i]^T$ is the state vector at instance i . w_i and v_i are supposed to be zero-mean Gaussian noises for the system and measurements respectively. $f(\cdot)$ and $h(\cdot)$ are the models of the system and the measurements respectively. y_i is the vector of measurements returned by the sensor. For each iteration, the EKF calculates the best estimate of the state vector in two stages:

1. Prediction stage:

$$\begin{cases} \hat{S}_i = f(0, u_i, 0) \\ \hat{P}_i = (\nabla_{s_i} \mathbf{F}) P_i (\nabla_{s_i} \mathbf{F}^T) + W_i \end{cases} \quad (7.8)$$

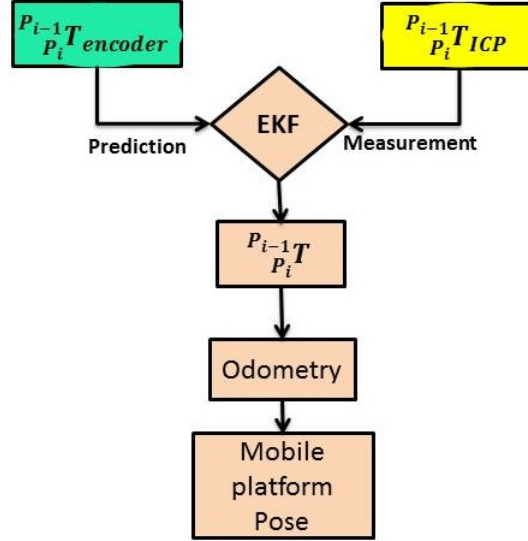


Figure 7.4: Flowchart of using the Extended Kalman Filter in fusing the two local transformation matrices.

where $\mathbf{F}(\cdot)$ is the Jacobian matrix that linearizes the system model $f(\cdot)$, \hat{P}_i is the covariance matrix for predicted state vector \hat{S}_i and W_i is the covariance matrix of the system noise w_i . The predicted state vector is calculated from the system model, (Equation (3.19)) by eliminating the previous state vector variable which will yield Equation (7.9). From Equation (7.9), we can calculate the Jacobian $\nabla_{s_i} \mathbf{F} = \frac{\partial f(0, u_i, 0)}{\partial s}$. In the case when $\Delta\phi_i = 0$, which means that the mobile platform has translation with no rotation, $\Delta X_i = \frac{(R_i + L_i)}{2}$, $\Delta Y_i = \Delta\phi_i = 0$.

$$\hat{S}_i = \begin{bmatrix} \Delta X_i \\ \Delta Y_i \\ \Delta Z_i \\ \Delta\alpha_i \\ \Delta\beta_i \\ \Delta\phi_i \end{bmatrix}_{encoder} = \begin{bmatrix} -r_i \sin\left(\frac{R_i - L_i}{w_b}\right) \\ r_i \left(\cos\left(\frac{R_i - L_i}{w_b}\right) - 1\right) \\ 0 \\ 0 \\ 0 \\ \frac{R_i - L_i}{w_b} \end{bmatrix} \quad (7.9)$$

2. Update stage:

$$\begin{cases} K_i = \hat{P}_i \mathbf{H} \left(\mathbf{H} \hat{P}_i \mathbf{H}^T + V_i \right)^{-1} \\ S_i = \hat{S}_i + K_i (y_i - h(\hat{s}_i)) \\ P_i = (I_{n \times n} - K_i \mathbf{H}) \hat{P}_i \end{cases} \quad (7.10)$$

where K_i is the Kalman gain at instance i , $\mathbf{H}(\cdot)$ is the Jacobian matrix that linearizes the measurement model $h(\cdot)$, $y_i = [\Delta X_i \ \Delta Y_i \ \Delta Z_i \ \Delta \alpha_i \ \Delta \beta_i \ \Delta \phi_i]_{ICP}^T$ and V_i is the covariance matrix of the measurement noise v_i . The error covariance matrices were determined by calculating the error between the ground truth and the estimated mobile platform pose in both the encoder-based and visual odometry cases as will be explained in Section 7.5.2.

The global mobile platform pose can be calculated by using the following:

$${}^G_{P_i}T = {}^G_{P_{i-1}}T \cdot {}^{P_{i-1}}_{P_i}T \quad (7.11)$$

This is named the ICP based-optimized odometry.

7.3 Mobile Platform Motion Control Schemes

The main aim of this chapter is to design and implement a control motion scheme for a mobile platform that has high uncertainty. This is a step towards making the mobile platform capable of performing high precision tasks such as "obstacle avoidance" and "go through doorway". The existing motion control depends solely on wheel encoders which makes the system unreliable. However, encoder-based odometry has a higher frequency rate than the ICP-based odometry. That is because the latter uses a computer vision algorithm which needs more computational power. Therefore, the idea is to use the encoder-based odometry to control the mobile platform, and then update it once the ICP-based odometry

output is available. The update rate should be fast enough to guarantee the accuracy of position estimation. The following two types of motion control are used.

7.3.1 Encoder Only Motion Control

The mobile platform pose is estimated according to the encoder-based odometry method explained in Section 7.2.1. Then the mobile platform pose is compared with the motion reference input to calculate the motion error. Based on the error, a signal for rotation and/or translation is sent to the platform controller to minimize this error. Figure 7.5 shows a schematic diagram of this controller.

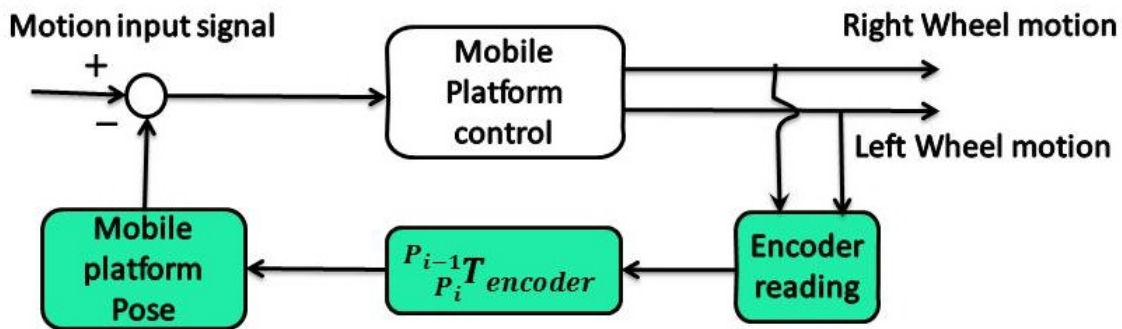


Figure 7.5: Schematic diagram of the mobile platform encoder-based odometry motion control.

7.3.2 ICP-Based Updated Odometry Motion Control

In this motion control, the two methods of mobile platform pose estimation are running at the same time. Because the encoder-based odometry is faster than the ICP-based odometry, it is used to control the mobile platform similar to the previous control scheme. The difference is that we use the ICP-based odometry here to correct the encoder-based odometry, (red updating link in Figure 7.6). As shown in Figure 7.6, the EKF is used to fuse the local transformation matrices computed by the ICP algorithm and encoders to obtain the optimized pose estimation. Then the encoder-based odometry is updated with the

ICP-based odometry. The resulting pose estimation is called ICP-based updated odometry.

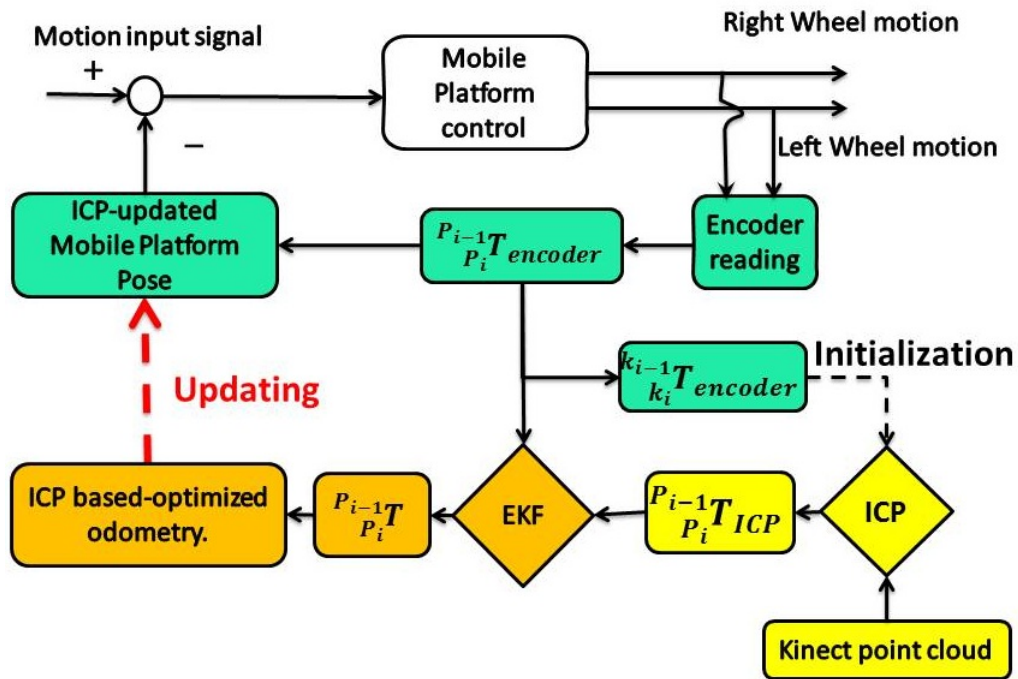


Figure 7.6: Schematic diagram of the mobile platform ICP-based updated odometry motion control.

7.4 Implementation Example

As stated previously, the assistive navigation systems for individuals with disabilities became a new area of research. One of the assistive devices is the WMRA system that has been developed at the Center for Assistive, Rehabilitation and Robotics Technologies (CARRT) at the University of South Florida [89,90]. Two prototypes of the WMRA system have been designed and built. Figure 7.7 shows the first prototype, WMRA I.

The WMRA system is used as an implementation example to test and evaluate the proposed motion controller. This device is a 7 DoF robotic arm attached to a power wheelchair. In the remainder of this chapter, mobile platform and wheelchair are used interchangeably.

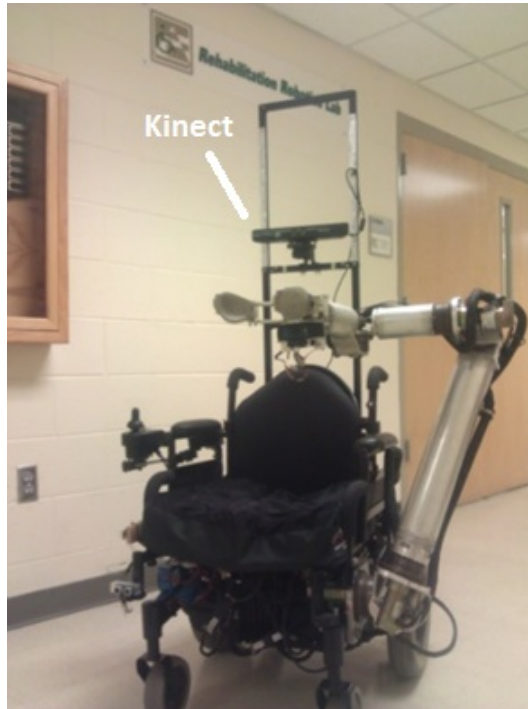


Figure 7.7: Wheelchair Mounted Robotic Arm (WMRA-I).

This work aims to design and implement a control motion scheme for the power wheelchair that has high uncertainty. This is a step towards making the WMRA system capable of performing high precision tasks such as "go and pick up" and "open the door". The existing motion control depends solely on wheel encoders which makes the system unreliable.

7.5 Experiments

In this work, the experiments were performed in two stages: offline data processing, and online wheelchair application. In both stages, the wheelchair ground truth was captured using the state-of-the-art VICON[®] (Oxford, UK) system with eight motion capture cameras. The cameras use infrared lights to detect passive reflective markers attached to the WMRA system and the Kinect. The markers' locations were captured at a frequency of 120 Hz. A Matlab[®] program was used for post motion data processing to compute the wheelchair and

Kinect frame poses. An accurate transformation matrix of the Kinect coordinate frame with respect to the wheelchair coordinate frame was determined from that data.

7.5.1 Offline Data Processing

The purposes of this stage were: to refine the ICP parameters and to determine the covariance matrices of the Kalman filter. In this stage, the wheelchair was commanded to move in a square motion for five loops. It was controlled using the encoder-only motion control scheme (Section 7.3.1). Kinect point clouds associated with the encoders reading and time stamp were stored using an onboard laptop. At the same time, the ground truth of the wheelchair was captured. The synchronization process between the ground truth wheelchair pose and ICP-based updated wheelchair pose is accomplished by using time stamp saved in both programs. The open source Point Cloud Library (PCL) [91] was used to process the Kinect point clouds and to implement the ICP algorithm. For the Kalman filter covariance matrices, four covariance matrices were estimated according to the case of the wheelchair motion, (translation or rotation): one translation error and one rotation error covariance matrices for encoder-based odometry, and one translation error and one rotation error covariance matrices for ICP algorithm.

7.5.2 Offline Data Processing Results and Discussion

To obtain the covariance matrices of the EKF, the encoder-based odometry and ICP-based odometry were calculated for the five loops. By calculating the error in the local transformation matrix estimated by these methods, we were able to determine the covariance matrices of the encoder-based odometry and the ICP algorithm. This was accomplished by calculating the error in the local transformation matrix estimated by the encoder-based odometry and the ground truth. A similar calculation was performed for the ICP registration process.

As stated before, the offline data processing is for evaluating the performance of our approach in terms of error detection and correction. Figure 7.8 shows the wheelchair pose estimated by the encoder-based odometry (blue-dashed line) with the corresponding wheelchair ground truth motion (red-dashed line). Only one square loop is shown for the clarity of the figure. Figure 7.8 demonstrates the high angle drift between the wheelchair pose that was estimated by the encoder-based odometry and the ground truth. The localization error was

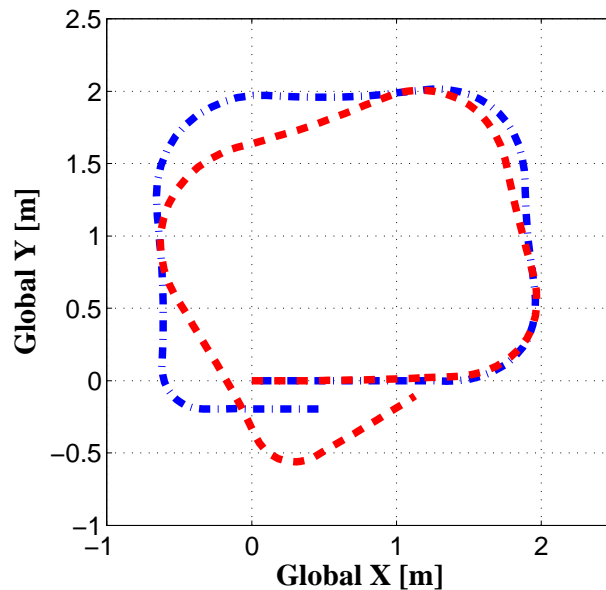


Figure 7.8: Wheelchair encoder-based odometry (blue-dashed line) with ground truth (red-dashed line).

mainly due to high orientation error which made the position error accumulate faster as the wheelchair moved. This justifies the assumption made earlier about this mobile platform high uncertainty. It is worth mentioning that the encoders and odometry equations were not calibrated with any odometry calibration procedure, such as UMBmark [92]. This is because it is necessary to determine how effective the approach is in detecting and compensating the localization error in the case of platforms with high inaccuracies.

Figure 7.9 shows the ICP only estimated wheelchair position (black-dashed line) and the ground truth (red-dashed line). The position error of the ICP is resulted from the fact that the ICP determined the transformation matrix between the two consecutive frames. If there

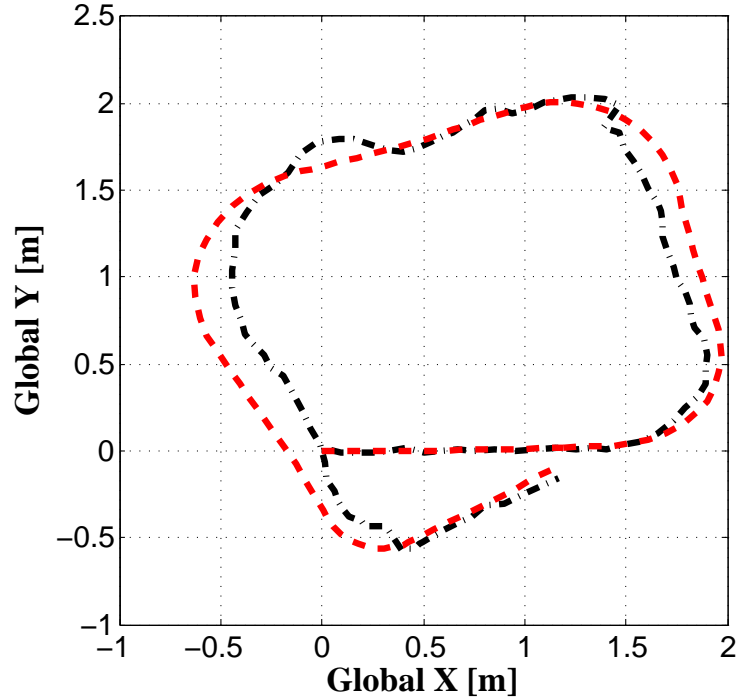


Figure 7.9: Wheelchair ICP only odometry (black-dashed line) with ground truth (red-dashed line).

is any alignment error, this error will propagate in all of the coming frames. This makes the ICP-based wheelchair position estimation inaccurate. However, the ICP algorithm could detect, with a good accuracy, the wheelchair orientation angle.

Figure 7.10 shows the ICP based-optimized odometry (green-dashed line) relative to the ground truth (red-dashed line). This demonstrates the ICP based-optimized odometry without updating it to the wheelchair encoder based odometry. This can be visualized, in Figure 7.6 by take out the updating link (red dashed arrow). It shows the effect of the updating process on the accuracy of the ICP algorithm registration process and on the pose estimation process. The improvement can be noticed by comparing Figure 7.10 with Figure 7.11.

The important step in Kalman filter implementation is the system and measurement error modeling. We assume that these errors are time invariant. Having a precise ground truth allows us to have accurate error covariance matrices. The error modeling and testing were

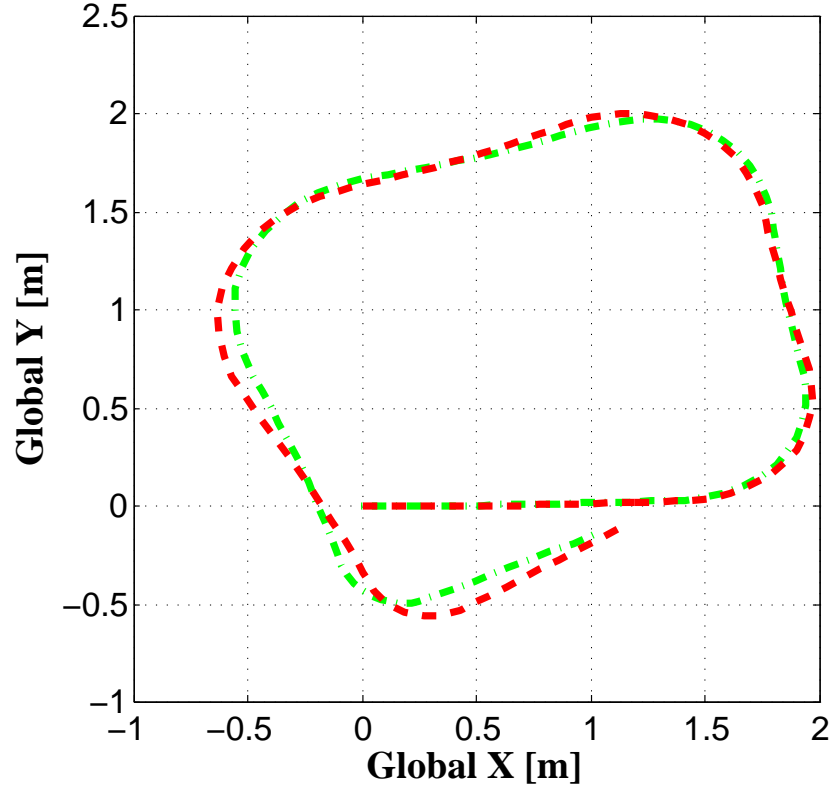


Figure 7.10: Wheelchair ICP based-optimized odometry without updating (green-dashed line) with ground truth (red-dashed line).

done in two different data sets. Four covariance matrices were determined: encoder rotation, encoder translation, ICP rotation and ICP translation. As it is illustrated in Figure 7.10, the optimized odometry has the best wheelchair position tracking compared to the two previous methods. Also, applying the Kalman filter gives another advantage. If the ICP algorithm gives any unexpected transformation matrix between two wheelchair locations, the Kalman filter will optimize the current observation according to the previous observations. This makes the optimized odometry smoother compared to the ICP algorithm.

For real time wheelchair motion control, the encoder based odometry is updated with the ICP based- optimized odometry. This is because the optimized odometry is too slow to control the wheelchair motion. For now, this process is done offline. Figure 7.11 shows the result of offline correction of the encoder based odometry with the optimized one. The

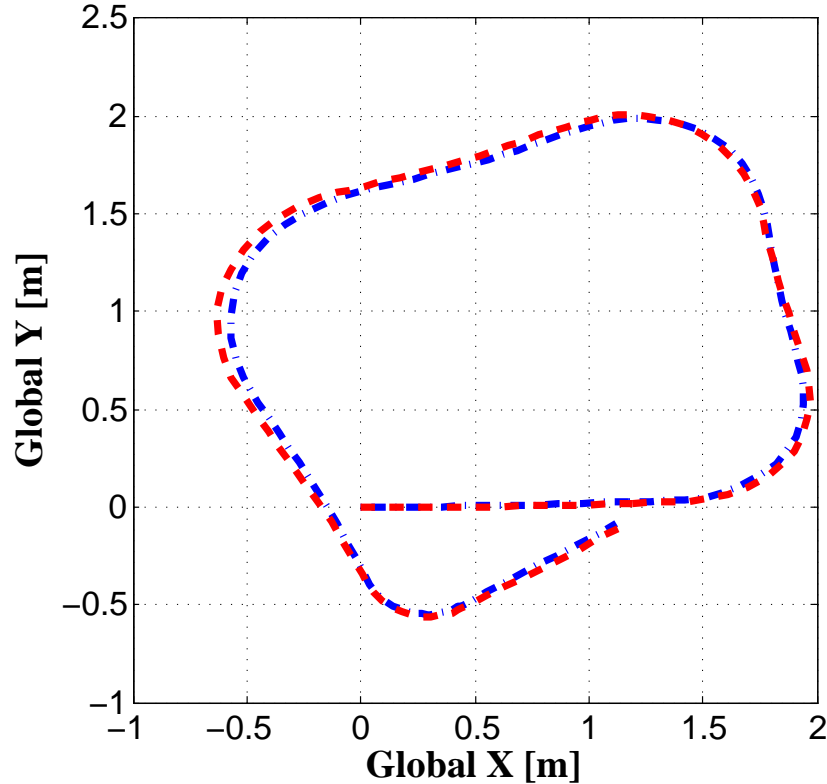


Figure 7.11: Wheelchair ICP based-optimized odometry (blue-dashed line) with ground truth (red-dashed line).

correction process was accomplished in two steps: a) the difference between the global poses of the wheelchair estimated by encoder based odometry and ICP based-optimized odometry is calculated. The difference is added to the encoder based odometry global pose; b) The encoder counts for left and right wheels equivalent to the difference is computed. These encoder counts are added to the encoder readings. For online implementation, the ICP algorithm parameters have to be relaxed to get enough correcting rate and enough overlap between the scenes of the Kinect camera. This makes the online implementation results different from the offline results. Improving the online implementation will be the future plan of this work.

For a complete comparison, Figure 7.12 and Figure 7.13 show the global position and orientation error for the encoder-based odometry (refer to Section 7.2.1), ICP based odometry

(refer to Section 7.2.2.1) and ICP based-optimized odometry (refer to Section 7.2.2.2).

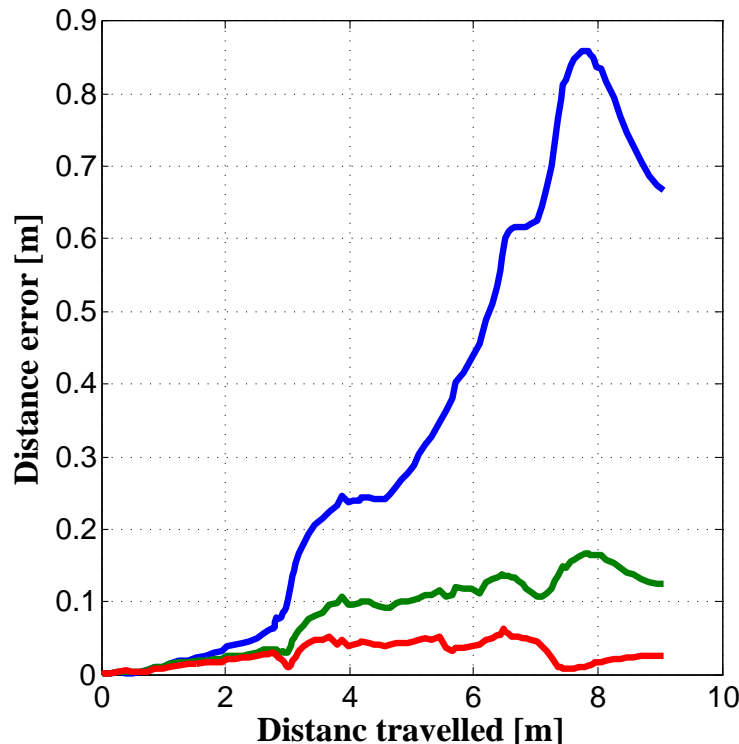


Figure 7.12: Global position error for encoder-based (blue line), ICP based (green line) and ICP-based updated (red line) odometry for online implementation.

These errors are the difference between the wheelchair pose estimation methods and the ground truth. The encoder-based odometry (blue line) has larger error compared with the two other methods. After the wheelchair moved approximately eight meters, the encoder-based odometry has almost a 0.8m position error, which is 10% of the traveled distance, while ICP based-updated odometry has less than 0.05m error, which is 0.6% of the traveled distance. This is an improvement of about fifteen times. The same outcome is observed with the orientation angle; after the wheelchair rotated for approximately 360° , the encoder-based odometry had almost 26 degrees of error, which is 7.2% of the rotated angle, while the ICP based-updated odometry had less than one degrees of error, which is 0.27% of the rotated angle. This is an improvement of about twenty-six times. This shows how much

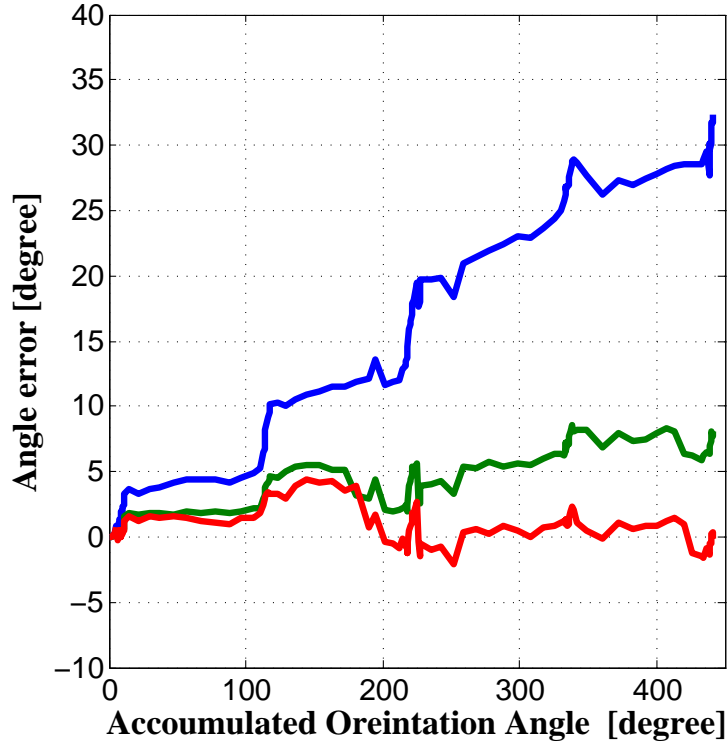


Figure 7.13: Global orientation angle error for encoder-based (blue line), ICP-based (green line) and ICP-based updated (red line) odometry for online implementation.

the encoder-based odometry improved by using ICP based-updated odometry in offline data processing.

7.5.3 Online Implementation Results

As it was previously stated in online implementation, the rate of updating the encoder-based odometry with the ICP-based odometry should be fast enough to guarantee the Kinect’s scenes have sufficient overlap, which is crucial for a successful registration process. As a result, the wheelchair pose error is corrected regularly. This was achieved by relaxing the ICP’s parameters, which made the online results differ from the offline results. Figure 7.14 shows online implementation results with the wheelchair ground truth of the proposed algorithm for one square motion. The figure demonstrates four times the improvement in wheelchair pose estimation.

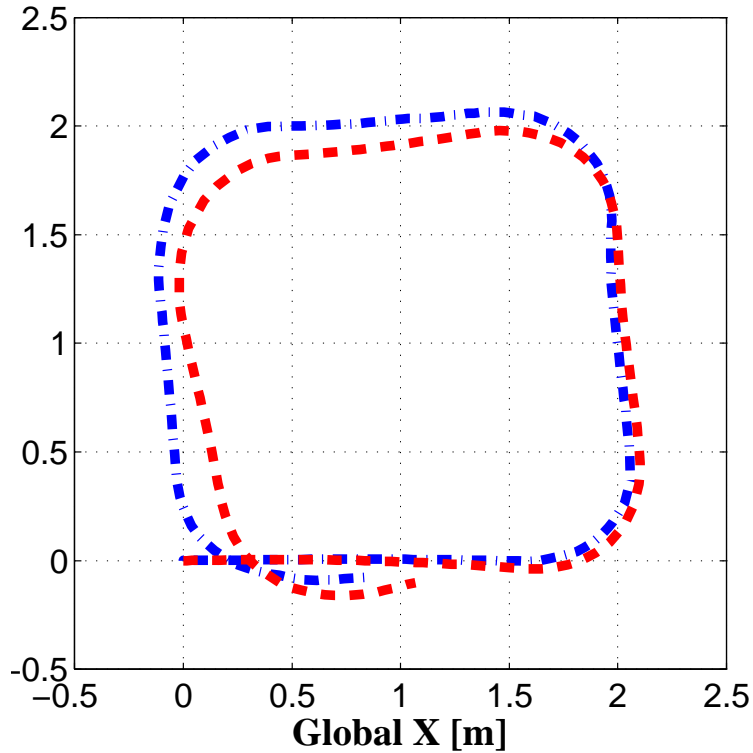


Figure 7.14: ICP-based updated odometry (blue-dashed line) for online implementation with wheelchair ground truth (red-dashed line).

For a global position and orientation errors, Figure 7.15 and Figure 7.16 show the global position and orientation error for the encoder-based (blue lines) and ICP-based updated odometry (black lines) in online implementation. These errors are the difference between the wheelchair pose estimation using these methods and the ground truth for a one square loop. The encoder-based odometry (blue line) has larger error in both cases (position and orientation) compared with the other method. After the wheelchair moved approximately 8 meters, the encoder-based odometry has almost 0.8m position error, which is 10% of the traveled distance, while in the ICP-based updated odometry, the position error was 0.2m which is 2.5% of the traveled distance. This is an improvement of four times. More improvement can be achieved by using high computational power. The same outcome is observed with the orientation angle; after the wheelchair rotated for approximately 360° , the encoder-based odometry had almost 27 degrees of error, which is 7.5% of the rotated angle, while

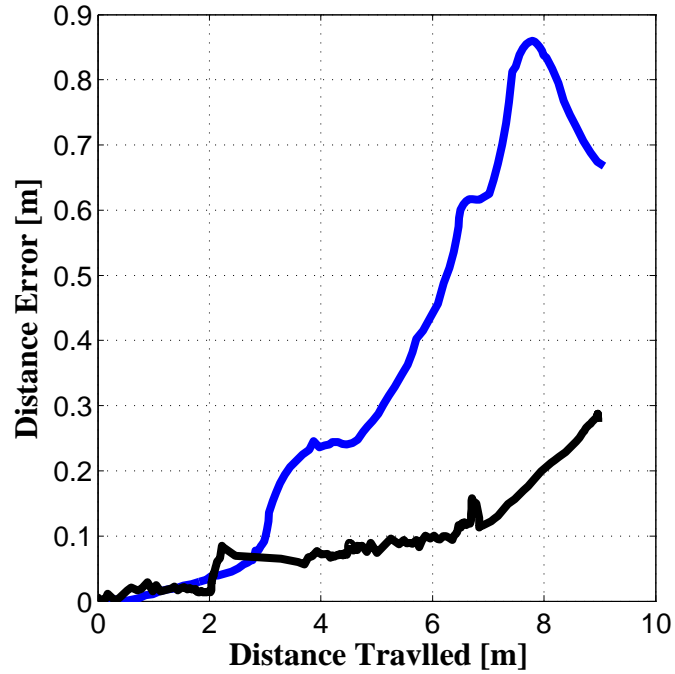


Figure 7.15: Global position error for encoder-based (blue line) and ICP-based updated (black line) odometry for online implementation.

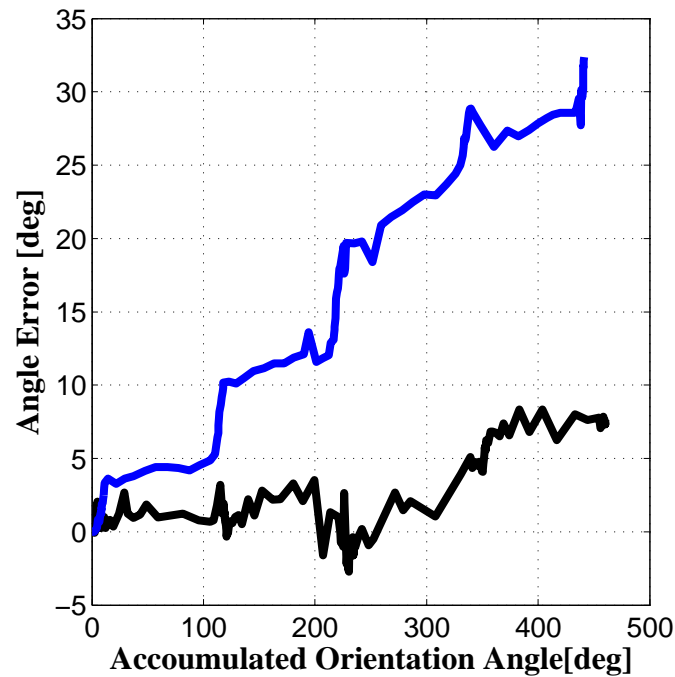


Figure 7.16: Global orientation angle error for encoder-based (blue line) and ICP-based updated (black line) odometry for online implementation.

the ICP-based updated odometry had around 7 degrees of error, which is around 2% of the rotated angle. This is an improvement of about four times. Even though the online application improvement is less than the offline data processing, The online implementation could be improved by using more computational power, such as using GPU, which is considered as a future work.

Although Kinect cannot operate in an outdoor environment, this technique can be applied outdoors by using other sensors that suit these environments, such as a 3D laser scanner. The important issue here is that the registration process needs 3D static features within the depth range of the sensor to get good alignment results. Therefore, this technique is not applicable in outdoor or indoor environments that lack these features. Using the probabilistic model to fuse the measurements can cope with some failures of the alignment process. In a worst case scenario, in which the registration always fails, the wheelchair localization will be solely dependent on the wheel encoders.

7.5.4 Real-Time Mobile Platform Application

This experiment stage was designed to test and evaluate the proposed motion control scheme on an obstacle avoidance problem. This application involves many crucial tasks which need faster, more accurate motion control. These tasks are: detecting an obstacle, mapping the environment, path planning, and path execution. Another Microsoft Kinect camera was used to detect the obstacle and to create a 2D-map of the environment. The A-star algorithm [93] combined with a path smoother algorithm was used for path planning [94].

At the beginning of this experiment, the wheelchair was at a zero-orientation angle relative to the global coordinate frame, and the obstacle was at an angle of approximately $+135^\circ$ relative to the global coordinate frame as illustrated in Figure 7.17. The wheelchair was commanded to move from a point $(0, 0)$ to a point $(-2.5, 2.5)$ with respect to the

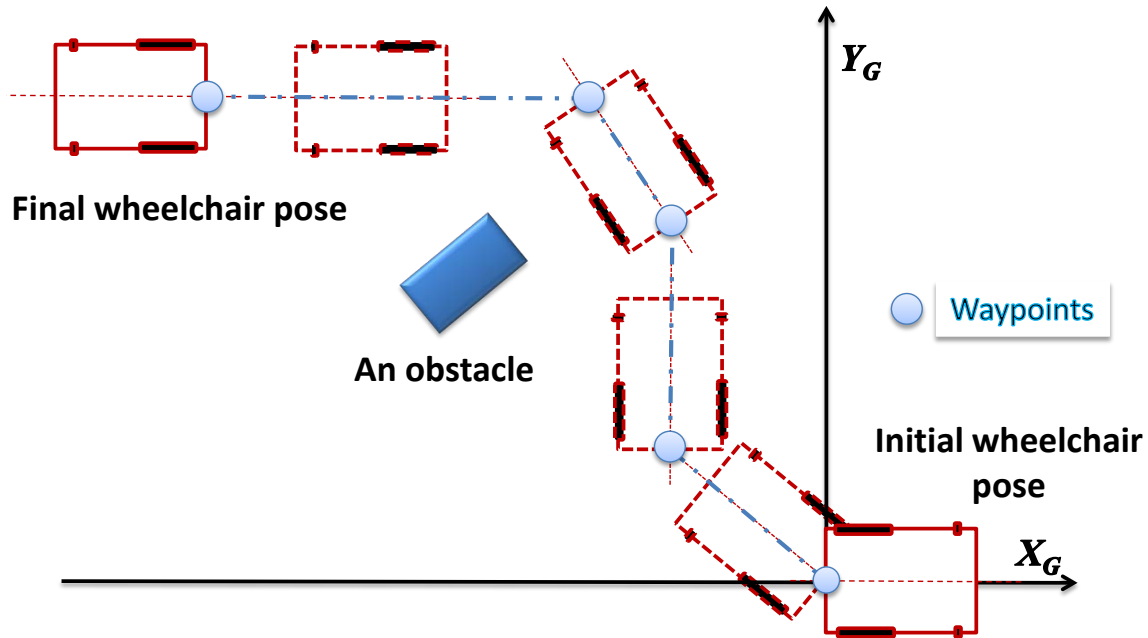


Figure 7.17: Initial and final wheelchair poses and path planning.

global coordinate frame. Therefore, the wheelchair had to rotate 135° first, and then the wheelchair faced the obstacle. Once the wheelchair detected the obstacle, the path planning process started. The wheelchair stopped moving and the Kalman and the ICP Algorithms were paused until the waypoints of the path were determined. The wheelchair environment was then mapped by the second Microsoft Kinect camera. The environment's 2D map was uploaded to the planning algorithm. The output of the path planning algorithm was the waypoints of the wheelchair trajectory. The wheelchair was then commanded to move from one waypoint to the other in two consecutive movements, rotation and translation until it reached the final waypoint.

7.5.4.1 Real-Time Application Results and Discussion

The wheelchair and the obstacle ground truth were captured using the motion capture system. The obstacle was also mapped using the second Microsoft Kinect camera (black line

in Figure 7.18 and Figure 7.19). Figure 7.18 and Figure 7.19 show the results of the encoder only and ICP-based updated motion control schemes, respectively, with the ground truth of the wheelchair and the obstacle. In Figure 7.18, the Kinect-obstacle coordinates did not

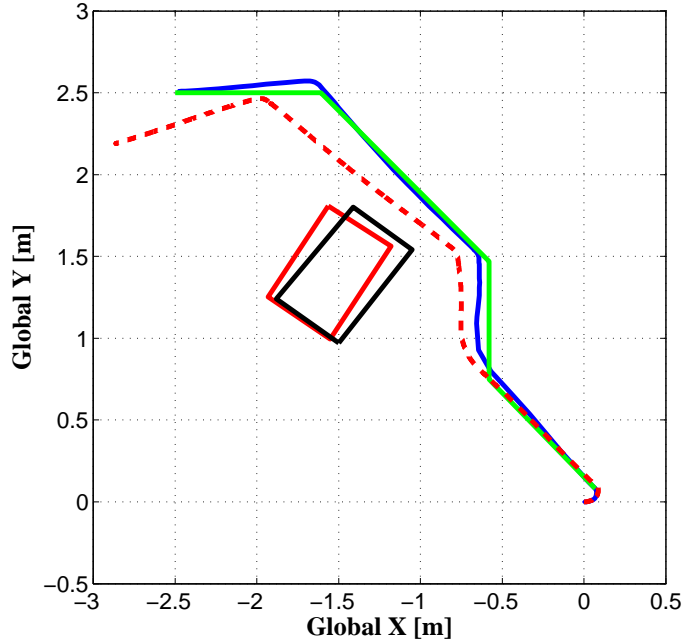


Figure 7.18: Wheelchair pose estimation using encoder-based odometry control (blue line) with the ground truth (red-dashed line for the wheelchair and red-solid line for the obstacle), Kinect-mapped obstacle (black line) and commanded trajectory (green line) for both control schemes.

match the obstacle ground truth. This was due to the encoder-based odometry orientation error.

This introduced obstacle localization error which increased the possibility of collision with the obstacle. In addition, the wheelchair pose estimated by the encoder based odometry had a considerable deviation from the wheelchair ground truth at the end of the path execution, due to the starting orientation angle error. However, in the case of ICP-based updated odometry motion control scheme (refer to Figure 7.19), the obstacle mapping was more accurate compared to the former method. This was due to the pose error correction of the ICP-based updated odometry motion control. It was also observed that the wheelchair ground truth had a near perfect match with the ICP-based updated odometry.

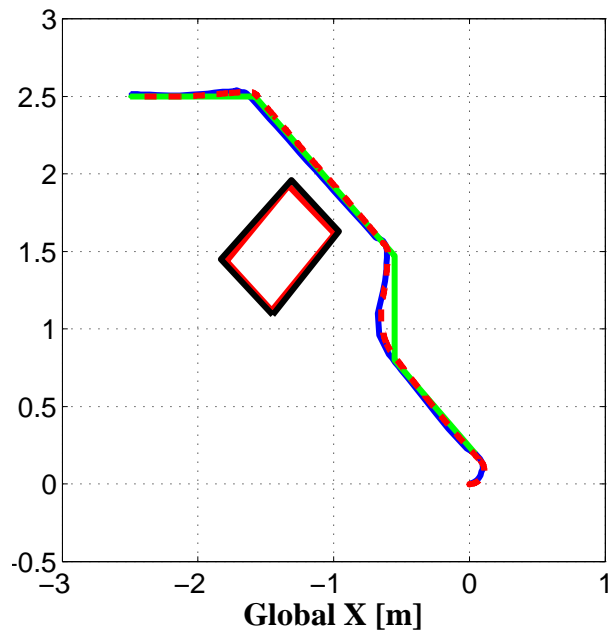


Figure 7.19: Wheelchair pose estimation ICP-based updated odometry control (blue line) with the ground truth (red-dashed line for the wheelchair and red-solid line for the obstacle), Kinect-mapped obstacle (black line) and commanded trajectory (green line) for both control schemes.

CHAPTER 8

SIMULATION IMPLEMENTATION AND RESULTS FOR THE WMRA SYSTEM

8.1 Introduction

The simulation implementation of dual-trajectory control was a crucial step to evaluate, test and adjust the proposed controller before use on the real hardware. In this chapter, the two controllers that were designed and tested in Chapters (5 and 6), were implemented and tested on MATLAB simulation of the wheelchair mounted robotic arm (WMRA) system.

The WMRA system was used as an example platform for the proposed controllers. The WMRA system is an assistive device designed and developed to enhance the capabilities of mobility-impaired individuals who use power wheelchairs and have very limited hand motions. The WMRA system consists of a standard power-wheelchair and a 7 DoF robotic arm (manipulator). As shown in Figure 8.1, two prototypes of the WMRA were designed and built in the Center for Assistive, Rehabilitation and Robotics Technologies (CARRT) at the University of South Florida [41, 95].

8.2 Kinematic Model for WMRA System

The WMRA system has 9 DoFs. The whole joint space of the WMRA system can be defined as follows:

$$q = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5 \ \theta_6 \ \theta_7 \ S \ \phi]^T \quad (8.1)$$



WMRA-I System.



WMRA-II System.

Figure 8.1: The two WMRA system prototypes.

where S is the linear translation of the wheelchair along its x-axis, and ϕ is the rotation angle of the wheelchair about its z-axis, which is named the “orientation” angle. Throughout this chapter, the subscript or superscript of the letters G, W, A and E refer to Ground (Global), Wheelchair, Arm base and End-effector coordinate frames, respectively as shown in Figure 8.2. The subscript and superscript W which refers to the wheelchair is used instead of subscript P which refers to a mobile platform. Since in this chapter, the two control algorithms will be implemented on the wheelchair, the symbol W will be used to refer to the wheelchair. For more details regarding the WMRA system refer to [6].

8.2.1 Wheelchair Kinematic Model

Two of the DoFs are provided by the nonholonomic motion of the wheelchair. The wheelchair frame W is located at the midpoint between the driving wheels. This subsystem of the WMRA system uses two input polar velocity variables $\dot{q}_W = [\dot{S} \ \dot{\phi}]^T$. The translation

and rotation of the wheelchair can be calculated from the angular rotation of the left and right wheels $\dot{\theta}_l$ and $\dot{\theta}_r$, respectively using Equation (8.2):

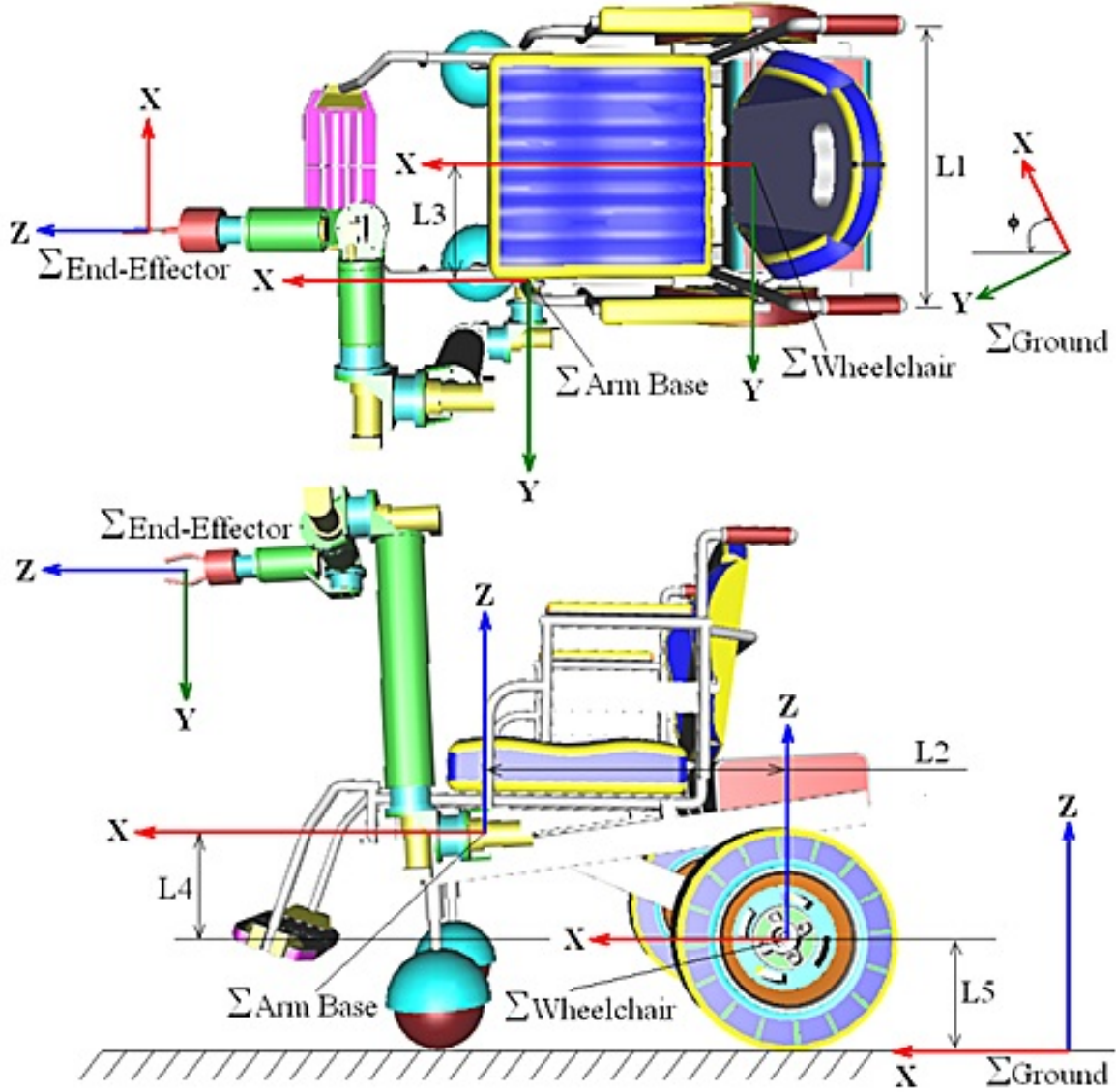


Figure 8.2: WMRA system coordinate frames [6].

$$\dot{q}_W = \begin{bmatrix} \dot{S} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{L_5}{2} & \frac{L_5}{2} \\ -\frac{L_5}{L_1} & \frac{L_5}{L_1} \end{bmatrix} \begin{bmatrix} \dot{\theta}_l \\ \dot{\theta}_r \end{bmatrix} \quad (8.2)$$

The wheelchair pose with respect to (w.r.t.) the G frame can be defined as: $r_{GW} = [x_{GW} \ y_{GW} \ \phi]^T$ where x_{GW} and y_{GW} are the x and y global coordinates, and ϕ is the

orientation angle. Due to nonholonomic constraints, the relation between the wheelchair velocities \dot{r}_{GW} and the wheelchair translation and rotation velocities can be determined using Equation (8.3):

$$\begin{aligned} \begin{bmatrix} \dot{x}_{GW} \\ \dot{y}_{GW} \\ \dot{\phi} \end{bmatrix} &= J_{GW} \begin{bmatrix} \dot{S} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{S} \\ \dot{\phi} \end{bmatrix} \\ \Rightarrow \dot{r}_{GW} &= J_{GW} * \dot{q}_W \end{aligned} \quad (8.3)$$

The Cartesian velocities of the frame A w.r.t. the frame G (\dot{r}_{GA}) can be calculated using Equation (8.4):

$$\begin{aligned} \begin{bmatrix} \dot{x}_{GA} \\ \dot{y}_{GA} \\ \dot{\phi} \end{bmatrix} &= J_{GA} \begin{bmatrix} \dot{x}_{GW} \\ \dot{y}_{GW} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -(L_2 \sin\phi + L_3 \cos\phi) \\ 0 & 1 & L_2 \cos\phi - L_3 \sin\phi \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_{GW} \\ \dot{y}_{GW} \\ \dot{\phi} \end{bmatrix} \\ \Rightarrow \dot{r}_{GA} &= J_{GA} * \dot{r}_{GW} \end{aligned} \quad (8.4)$$

The Cartesian velocities of frame E w.r.t. frame A (\dot{r}_{AE}), which is caused by the wheelchair motion only (when the manipulator is static), can be calculated using Equation (8.5):

$$\begin{aligned} \begin{bmatrix} \dot{x}_{AE} \\ \dot{y}_{AE} \\ \dot{\phi} \end{bmatrix} &= J_{AE} \begin{bmatrix} \dot{x}_{GA} \\ \dot{y}_{GA} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -(x_{AE} \sin\phi + y_{AE} \cos\phi) \\ 0 & 1 & x_{AE} \cos\phi - y_{AE} \sin\phi \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_{GA} \\ \dot{y}_{GA} \\ \dot{\phi} \end{bmatrix} \\ \Rightarrow \dot{r}_{AE} &= J_{AE} * \dot{r}_{GA} \end{aligned} \quad (8.5)$$

By combining Equation (8.3), Equation (8.4), and Equation (8.5):

$$\dot{r}_{GEW} = J_{GEW} * \dot{q}_W = J_C * J_{AE} * J_{GA} * J_{GW} * \dot{q}_W \quad (8.6)$$

where J_{GEW} is the Jacobian that relates the end-effector Cartesian velocities \dot{r}_{GEW} , which are caused by the wheelchair's motions, to wheelchair velocities \dot{q}_W and J_C is the Jacobian matrix that maps the three Cartesian coordinates of the wheelchair to six Cartesian coordinates of the end-effector as follows:

$$J_C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T$$

8.2.2 Robotic Arm Kinematic Model

The robotic arm mounted on the wheelchair provided 7 DoFs. All of the joints on the robotic arm were revolute joints. The joint angle vector of the arm was defined as follows:

$$q_A = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5 \ \theta_6 \ \theta_7]^T$$

The manipulator is mounted on the wheelchair at frame A as shown in Figure 8.2. The transformation matrix of the arm base frame A w.r.t. the wheelchair frame W is

$${}^W_A T = \begin{bmatrix} 1 & 0 & 0 & L_2 \\ 0 & 1 & 0 & L_3 \\ 0 & 0 & 1 & L_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

From the D-H parameters of the arm, the arm's 6×7 Jacobian matrix J_A can be calculated based on Craig's notation [71]. J_A is the Jacobian that relates the manipulator's joint rates \dot{q}_A to the Cartesian velocities of the end-effector w.r.t the frame A (\dot{r}_{AE}) as follows:

$$\dot{r}_{AE} = J_A \cdot \dot{q}_A \tag{8.7}$$

where $\dot{r}_{AE} = \begin{bmatrix} \dot{x}_{AE} & \dot{y}_{AE} & \dot{z}_{AE} & \omega_{x_{AE}} & \omega_{y_{AE}} & \omega_{z_{AE}} \end{bmatrix}^T$. The Cartesian velocities of the end-effector w.r.t the global frame G caused by arm motions (\dot{r}_{GE_A}) can be computed as follows:

$$\dot{r}_{GE_A} = J_{GE_A} * \dot{q}_A \quad (8.8)$$

where $\dot{r}_{GE_A} = \begin{bmatrix} \dot{x}_{GE} & \dot{y}_{GE} & \dot{z}_{GE} & \omega_{x_{GE}} & \omega_{y_{GE}} & \omega_{z_{GE}} \end{bmatrix}^T$ represents the end-effector Cartesian velocities caused by the arm motion only (the wheelchair is stationary) and J_{GE_A} is the Jacobian that relates the end-effector Cartesian velocities to the manipulator's joint velocity vector \dot{q}_A . The J_{GE_A} can be calculated from J_A by using the rotation matrix of the frame A w.r.t. the frame G (${}^G_A R$) as follows:

$$J_{GE_A} = \begin{bmatrix} {}^G_A R_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & {}^G_A R_{3 \times 3} \end{bmatrix} * J_A \quad (8.9)$$

8.2.3 Combined Wheelchair Mobility and Robotic Arm Manipulation

The forward kinematics of the WMRA system can be presented as follows:

$$\dot{r}_{GE} = \begin{bmatrix} \dot{x}_{GE} \\ \dot{y}_{GE} \\ \dot{z}_{GE} \\ \omega_{x_{GE}} \\ \omega_{y_{GE}} \\ \omega_{z_{GE}} \end{bmatrix} = J_{sys} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \\ \dot{\theta}_7 \\ \dot{S} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} J_{GE_{A_{6 \times 7}}} & \vdots & J_{GE_{W_{6 \times 2}}} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \\ \dot{\theta}_7 \\ \dot{S} \\ \dot{\phi} \end{bmatrix} \quad (8.10)$$

8.3 Implementation Using MATLAB Simulation

A MATLAB simulation was developed for the WMRA system to test the proposed controllers in tracking dual-trajectories. In this simulation, different trajectories were used to evaluate the performance and the effectiveness of the controllers. Figure 8.3 shows the MATLAB simulated WMRA System. The physical parameters of the WMRA system are: for the

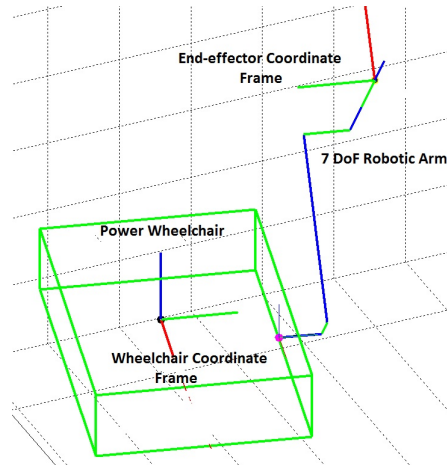


Figure 8.3: MATLAB simulated WMRA system.

wheelchair (refer to Figure 8.1) $L_1 = 560, L_2 = 440, L_3 = 230, L_4 = 182,$ and $L_5 = 168(mm)$.

The robotic arm's D-H parameters are shown in Table 8.1.

Table 8.1: The D-H parameters of the robotic arm.

i	α_{i-1} (degrees)	a_{i-1} (mm)	d_i (mm)	θ_i (degrees)
1	-90	0	110	θ_1
2	90	0	119	θ_2
3	-90	0	500	θ_3
4	90	0	121	θ_4
5	-90	0	235	θ_5
6	90	0	0	θ_6
7	-90	0	277	θ_7

8.3.1 System Modeling Using D , α and β

In this section, the spherical control variables D , α and β for WMRA system's dual-trajectory are determined. This is a special case of the general case that was explained in

Section 5.3. In this controller design, the trajectories of both the end-effector and wheelchair are completely predefined. This means that in each and every time instance, the pose (position and orientation) of both the end-effector and the wheelchair is known.

As stated previously, the aim is to test and evaluate the control scheme which was designed and tested in Chapter 5, on MATLAB simulated WMRA system. The dual-trajectory of the WMRA system can be divided into two subtasks which are: follow end-effect trajectory and follow wheelchair trajectory. The controller algorithm will be tested with task order priority (refer to Section 5.4 for more details). In this work, the techniques of alternating the order of priority between the two subtasks are not addressed, rather, the order of priority is predefined.

8.3.1.1 Spherical Control Variables (D , α and β)

This control algorithm had been tested on 2D simulated PMM (refer to Section 5.5). In this implementation example two spherical variables (D and α) were used. For the WMRA system the third control variable (β) has to be used to find the end-effector's frame height in the Z direction as illustrated in Figure 5.2. The control variables are defined as follows:

$$\begin{aligned}
 D &= \sqrt{{}^G X_{WE}^2 + {}^G Y_{WE}^2 + {}^G Z_{WE}^2} \\
 \alpha &= \tan^{-1} \left(\frac{{}^G Y_{WE}}{{}^G X_{WE}} \right) - \phi \\
 \beta &= \tan^{-1} \left(\frac{{}^G Z_{WE}}{\sqrt{{}^G X_{WE}^2 + {}^G Y_{WE}^2}} \right)
 \end{aligned} \tag{8.11}$$

where ${}^G X_{WE} = {}^G_E X - {}^G_W X$, ${}^G Y_{WE} = {}^G_E Y - {}^G_W Y$, and ${}^G Z_{WE} = {}^G_E Z - {}^G_W Z$ are the distances, in global X, Y, and Z directions respectively, from the wheelchair frame W to the end-effector frame E , and ϕ is the orientation angle of the wheelchair.

8.3.1.2 Spherical Control Variables Jacobian

The Jacobian of the control variables can be determined as follows:

$$J_{D\alpha\beta} = \nabla \begin{bmatrix} D(\theta_1, \dots, \theta_7, S, \phi) \\ \alpha(\theta_1, \dots, \theta_7, S, \phi) \\ \beta(\theta_1, \dots, \theta_7, S, \phi) \end{bmatrix} = \begin{bmatrix} \frac{\partial D}{\partial \theta_1} & \dots & \frac{\partial D}{\partial \theta_7} & \frac{\partial D}{\partial S} & \frac{\partial D}{\partial \phi} \\ \frac{\partial \alpha}{\partial \theta_1} & \dots & \frac{\partial \alpha}{\partial \theta_7} & \frac{\partial \alpha}{\partial S} & \frac{\partial \alpha}{\partial \phi} \\ \frac{\partial \beta}{\partial \theta_1} & \dots & \frac{\partial \beta}{\partial \theta_7} & \frac{\partial \beta}{\partial S} & \frac{\partial \beta}{\partial \phi} \end{bmatrix} \quad (8.12)$$

where $J_{D\alpha\beta} \in \mathbb{R}^{3 \times 9}$. It is worth mentioning here that in the case of finding the wheelchair pose relative to the end-effector pose, just $J_{D\alpha} \in \mathbb{R}^{2 \times 9}$ is used. On the other hand, $J_{D\alpha\beta} \in \mathbb{R}^{3 \times 9}$ is used if the pose of the end-effector is determined relative to the wheelchair pose.

8.3.1.3 Simulation Results and Discussion

The initial joint angles in degrees for the arm are $q_{A_0} = [45 \ 90 \ 90 \ 90 \ 0 \ 0 \ 90]$ degrees. The sinusoidal trajectories for the end-effector are: $y(x) = d_y + a_y \sin\left(\frac{\pi x}{f}\right)$ and $z(x) = d_z + a_z \sin\left(\frac{\pi x}{f}\right)$ where a and f are the amplitude and frequency of the sine wave respectively, and d is the initial position of the end-effector. For the end-effector trajectory, the total traveled distance in the X-direction is $x = 6000$ with a step of $5(mm)$, $f = 5000$, $d_y = 317.7$, $a_y = 700$, $d_z = 1150$, and $a_z = 100$. For the wheelchair, $y(x) = d_w + a_w \sin\left(\frac{\pi x}{f}\right)$ where $f = 2500$, $d_w = 0$ and $a_w = -200$. The wheelchair initial pose is $q_w = [x \ y \ \phi]^T = [0 \ 0 \ -14.1^0]^T$. The variable D is set to $1500(mm)$.

In Equation (5.10), the task priority is set by using the Jacobian of the high priority in the first term. Equation (5.10) is rewritten here for convenience:

$$\dot{q} = J_1^\# \dot{r}_1 + \left(I - J_1^\# J_1\right) \hat{J}_2^\# \left(\dot{r}_2 - J_2 J_1^\# \dot{r}_1\right) + \left(I - J_1^\# J_1\right) \left(I - \hat{J}_2^\# \hat{J}_2\right) H \quad (5.10 \text{ revisited})$$

In the case of end-effector trajectory as first priority, the complete state variables of the WMRA system can be presented as follows:

$$\dot{r}_{GE} = \begin{bmatrix} \dot{x}_{GE} \\ \dot{y}_{GE} \\ \dot{z}_{GE} \\ \omega_{x_{GE}} \\ \omega_{y_{GE}} \\ \omega_{z_{GE}} \\ \dot{D} \\ \dot{\alpha} \end{bmatrix} = J_{sys} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \\ \dot{\theta}_7 \\ \dot{S} \\ \dot{\phi} \end{bmatrix} = \left[\begin{array}{c|c} J_{GEA_{6 \times 7}} & J_{GEW_{6 \times 2}} \\ \hline & J_{D\alpha_{2 \times 9}} \end{array} \right] \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \\ \dot{\theta}_7 \\ \dot{S} \\ \dot{\phi} \end{bmatrix} \quad (8.13)$$

Therefore, in this case, $J_1 = \begin{bmatrix} J_{GEA_{6 \times 7}} & J_{GEW_{6 \times 2}} \end{bmatrix}$, $J_2 = \begin{bmatrix} J_{D\alpha_{2 \times 9}} \end{bmatrix}$, $\dot{r}_1 = [\dot{x}_{GE} \ \dot{y}_{GE} \ \dot{z}_{GE} \ \omega_{x_{GE}} \ \omega_{y_{GE}} \ \omega_{z_{GE}}]^T$ and $\dot{r}_2 = [\dot{D} \ \dot{\alpha}]^T$.

Three instances were simulated from Equation (5.10): using only the first term, using the first and second terms, and using all three terms. Figures 8.4, 8.5 and 8.6 show first term instance, first and second terms instance and all terms instance, respectively. In this simulation, the task is to follow the dual-trajectory with fixed end-effectors orientation. The first priority is given to the end-effector (position and orientation). The second priority is given to the wheelchair track. The surplus of the degrees of redundancy is for optimizing the whole WMRA system manipulability measure.

Figure 8.4 shows the result of using the first term of Equation (5.10), which is the LN solution that satisfies the tracking of the end-effector trajectory. In this case, the end-effector moves while the wheelchair is stationary until the arm is fully stretched, then the wheelchair moves. The wheelchair does not have any constraints over its trajectory. As can be seen, the wheelchair followed a random trajectory. This is the reason behind controlling the wheelchair trajectory. Therefore, by controlling the wheelchair trajectory, the wheelchair will be placed

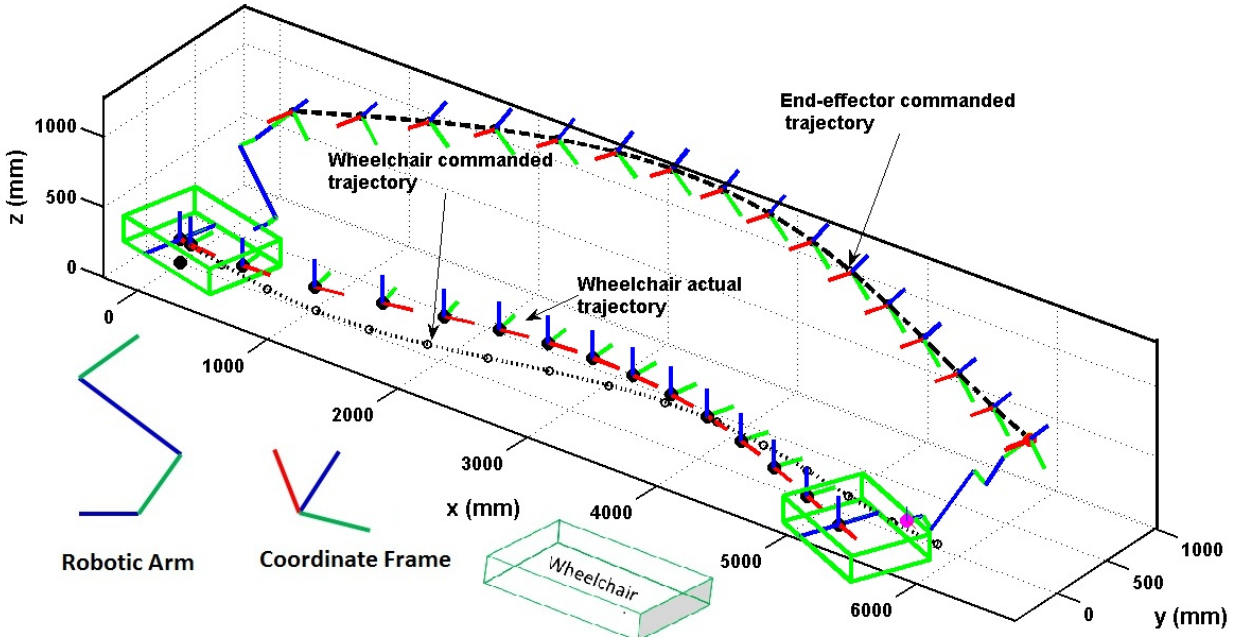


Figure 8.4: Tracked dual-trajectory of the WMRA system for the first term of Equation (5.10) $\dot{q} = J_1^\# \dot{r}_1$. The first priority is given for the end-effector. Refer to Figure 8.3 for more details about this Figure contents.

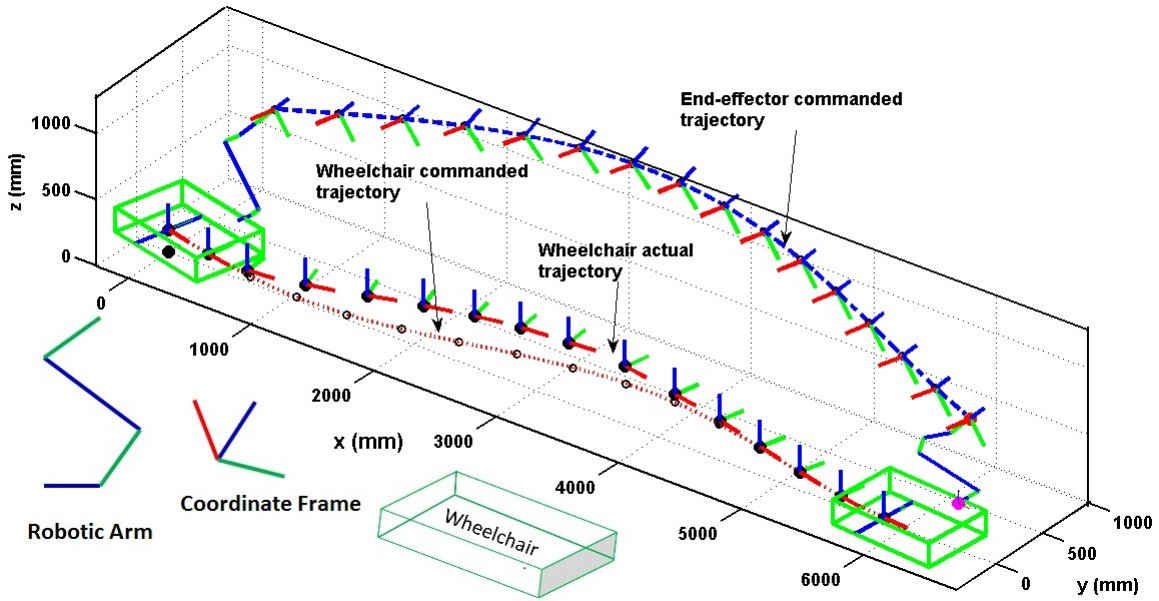


Figure 8.5: Tracked dual-trajectory of the WMRA system for the first and second terms of Equation (5.10) $\dot{q} = J_1^\# \dot{r}_1 + (I - J_1^\# J_1) \hat{J}_2^\# (\dot{r}_2 - J_2 J_1^\# \dot{r}_1)$. The first priority is given for the end-effector. Refer to Figure 8.3 for more details about this Figure contents.

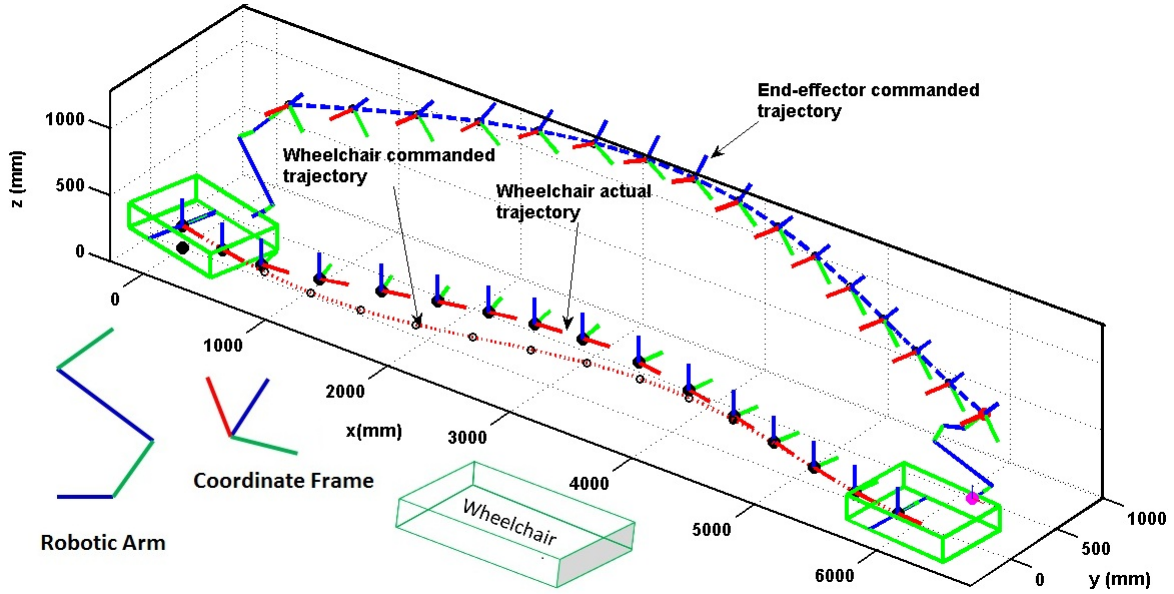


Figure 8.6: Tracked dual-trajectory of the WMRA system for all terms of Equation (5.10) $\dot{q} = J_1^\# \dot{r}_1 + (I - J_1^\# J_1) \hat{J}_2^\# (\dot{r}_2 - J_2 J_1^\# \dot{r}_1) + (I - J_1^\# J_1)(I - \hat{J}_2^\# \hat{J}_2) H$. The first priority is given for the end-effector. Refer to Figure 8.3 for more details about this Figure contents.

in a desired position that can support the arm in performing its task. The end-effector's trajectory was reached completely in position and orientation.

Figure 8.5 shows the results of the first and second terms of Equation (5.10). The second term is responsible for satisfying the second priority task, which is to follow the wheelchair trajectory. In this case, the controller will follow the end-effector's trajectory first and then track the wheelchair trajectory as much as possible. In the situation where the WMRA system cannot satisfy both trajectories, it follows the higher priority task, which is the end-effector's trajectory in this case, and allowed pose error in the lower priority task, which is the wheelchair trajectory. This can be seen in Figure 8.5 from distance 500 to 3500 mm. Figure 8.6 shows the results of using all terms in Equation (5.10). In this case, the third term is used to maximize the manipulability measure of the whole WMRA system using whatever redundancy is left after satisfying the first and second task priorities. This is clear in Figure 8.7. As expected, using the third term in Equation 5.10 allows the WMRA system to have a high manipulability measure possible at all times.

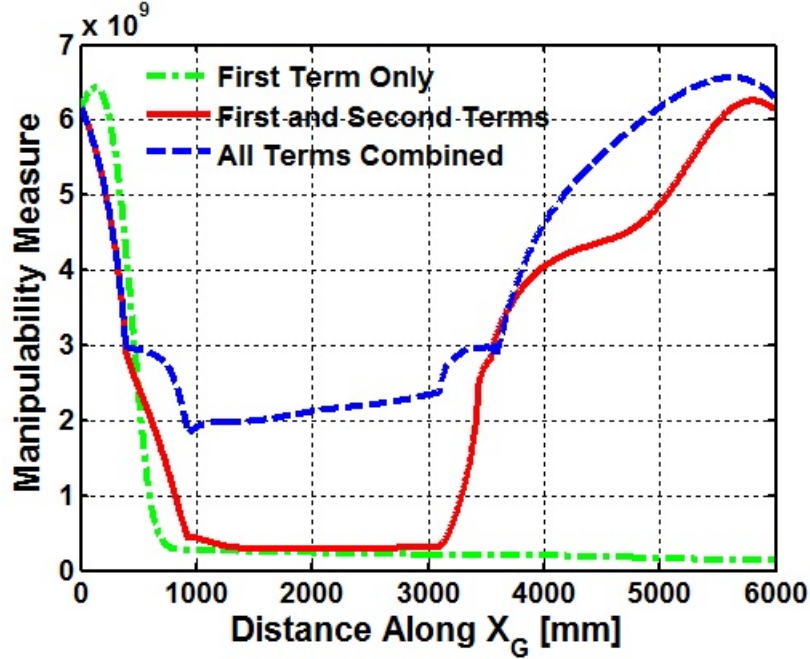


Figure 8.7: Whole WMRA system manipulability measure for the three instance of Equation (5.10) $\dot{q} = J_1^\# \dot{r}_1 + (I - J_1^\# J_1) \hat{J}_2^\# (\dot{r}_2 - J_2 J_1^\# \dot{r}_1) + (I - J_1^\# J_1)(I - \hat{J}_2^\# \hat{J}_2)H$: First term (green line). First and second terms (red line). All terms (blue line). The first priority is given for the end-effector.

In the case in which the wheelchair trajectory is the first priority, the end-effector position will be determined relative to the wheelchair trajectory using the developed three control variables D , α and β . Equation (5.5) becomes as follows:

$$\begin{aligned}
 G\dot{r}_E = \begin{bmatrix} \dot{S} \\ \dot{\phi} \\ \dot{D} \\ \dot{\alpha} \\ \dot{\beta} \\ \omega_{X_{GE}} \\ \omega_{Y_{GE}} \\ \omega_{Z_{GE}} \end{bmatrix} &= J_{sys} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \\ \dot{\theta}_7 \\ \dot{S} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} J_{S\phi_{2 \times 9}} \\ J_{D\alpha\beta_{3 \times 9}} \\ J_{\omega_{3 \times 9}} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \\ \dot{\theta}_7 \\ \dot{S} \\ \dot{\phi} \end{bmatrix} \quad (8.14)
 \end{aligned}$$

where $J_{S\phi} = \begin{bmatrix} [0]^{2 \times 7} & [I]^{2 \times 2} \end{bmatrix}$ and J_w is the last three rows of J_{sys} in Equation (8.10). To set the first priority for the wheelchair trajectory, $J_1 = J_{S\phi}$, $J_2 = \begin{bmatrix} J_{D\alpha\beta} & J_w \end{bmatrix}^T$, $\dot{r}_1 = \begin{bmatrix} \dot{S} & \dot{\phi} \end{bmatrix}^T$ and $\dot{r}_2 = \begin{bmatrix} \dot{D} & \dot{\alpha} & \dot{\beta} & \omega_{x_{GE}} & \omega_{y_{GE}} & \omega_{z_{GE}} \end{bmatrix}^T$. In this case, we only present the instance when all the terms of the Equation 5.10 are implemented. The other cases are similar to the previous cases. Figure 8.8 shows a sequence of locations of the end-effector and wheelchair which are represented by their respective coordinate frames. As can be noticed, the wheelchair always follows its trajectory while the end-effector follows its trajectory as much as possible. Manipulability measure of the whole WMRA system in this case is presented in Figure 8.9.

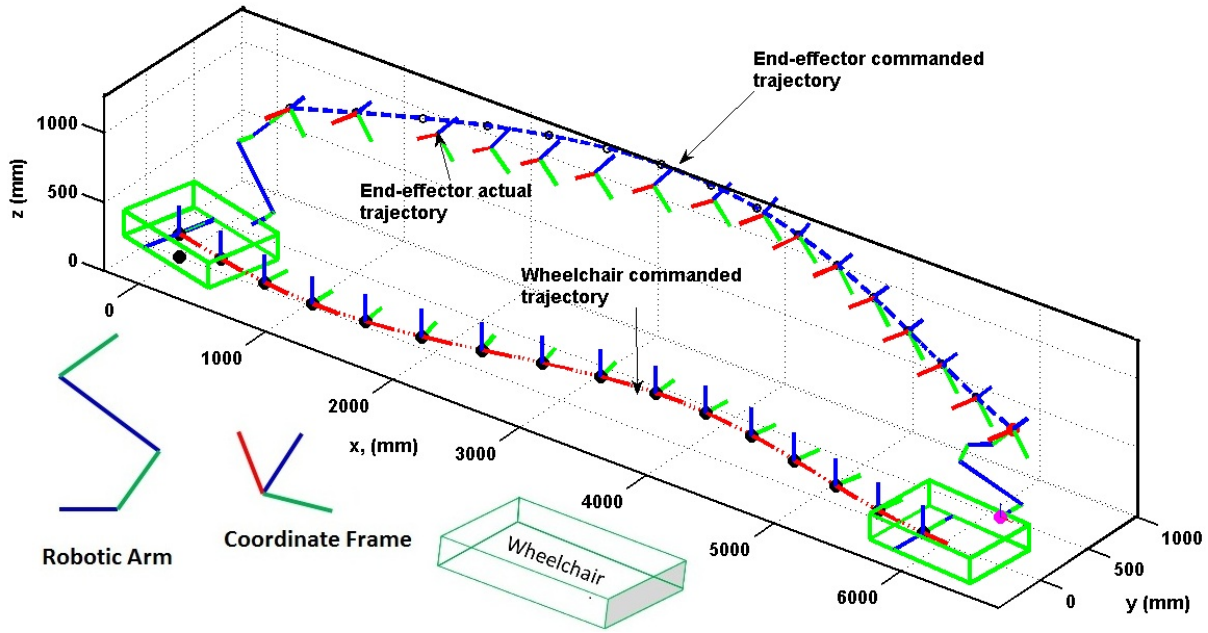


Figure 8.8: Tracked dual-trajectory of the WMRA system for all terms of Equation (5.10). The higher priority is given to follow the wheelchair’s trajectory. Refer to Figure 8.3 for more details about this figure contents.

8.3.2 System Modeling Using Optimized Translation Along a Track

In this section, the same controller algorithm explained in Section 6.2 will be implemented to MATLAB simulated WMRA system (Section 8.3) as an introduction to hardware implementation. The following is a quick revisit to the theory behind the control algorithm.

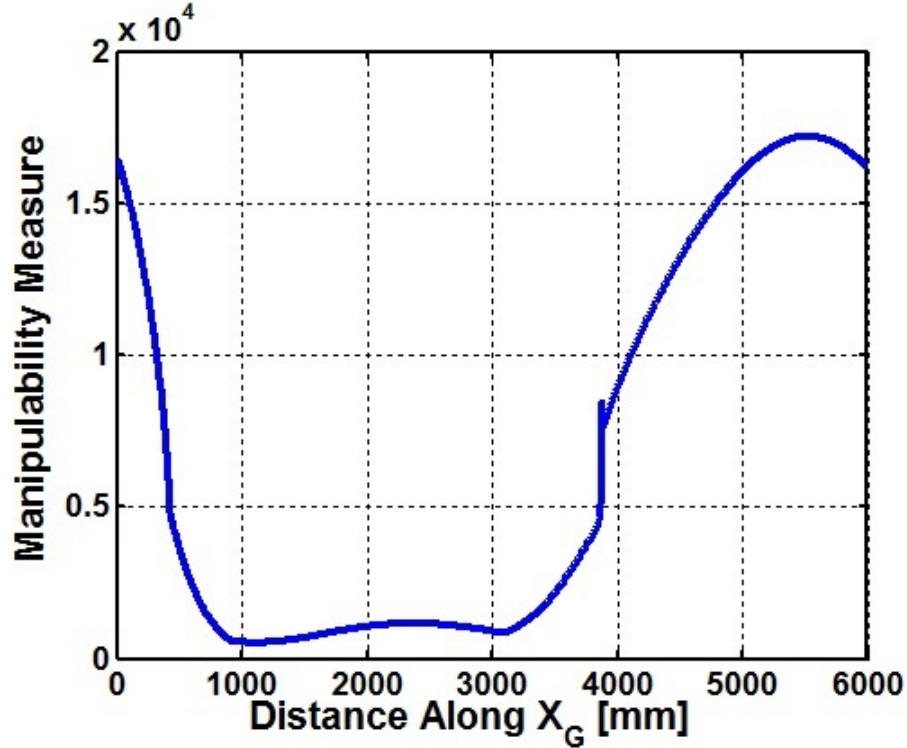


Figure 8.9: Whole WMRA system manipulability measure for all terms in Equation (5.10). The first priority is set to following the wheelchair's trajectory.

As it was previously pointed out, the mobile base track relates the mobile base's translation (S) and orientation (ϕ) due to the nonholonomic constraint. This correlation can be defined as follows:

$$\phi = \frac{S}{\rho} \quad (8.15)$$

where ρ is the radius of curvature of the track at a point P . ρ can be computed for a track, that is defined as $y = f(x)$, as follows:

$$\rho(x) = \frac{\left| 1 + \left(\frac{dy}{dx} \right)^2 \right|^{\frac{3}{2}}}{\left| \frac{d^2y}{dx^2} \right|} \quad (8.16)$$

Since the wheelchair can be controlled to follow the desired track using two consecutive movements (rotation and translation), the wheelchair orientation angle (ϕ) can be determined by relating the wheelchair translation S to the track curvature ρ . This requires

a modification to the Jacobian matrix J_{sys} in Equation (8.10) to allow determining the wheelchair orientation angle (ϕ) using Equation (8.15). These modifications are similar to those done to the PMM explained in Section 6.2.

8.3.2.1 Optimized Wheelchair Linear Velocity \dot{S}

The forward kinematic of the WMRA system can be expressed as follows:

$$\dot{r}_{GE} = \begin{bmatrix} J_{GEA_{6 \times 7}} & \vdots & J_{GEW_{6 \times 2}} \end{bmatrix} \begin{bmatrix} \dot{q}_A \\ \dot{q}_W \end{bmatrix} \quad (8.17)$$

where $\dot{q}_A = \begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_3 & \dot{\theta}_4 & \dot{\theta}_5 & \dot{\theta}_6 & \dot{\theta}_7 \end{bmatrix}^T$ and $\dot{q}_W = \begin{bmatrix} \dot{S} & \dot{\phi} \end{bmatrix}^T$. In order to allow the wheelchair to translate freely on its prespecified track, only the path without waypoints is specified as shown in Figure (6.2). The magnitudes of the translation of the wheelchair along its track are left to the controller to determine them according to the robotic arm configurations. Only the Jacobian matrix of the wheelchair J_W is modified to find the wheelchair unconstrained translations. These modifications are as follows:

From Equation (8.17), $J_{GEW_{6 \times 2}} \cdot \dot{q}_W = \begin{bmatrix} J_{WS_{6 \times 1}} & J_{W\phi_{6 \times 1}} \end{bmatrix} \begin{bmatrix} \dot{S} \\ \dot{\phi} \end{bmatrix}$, where $J_{WS_{6 \times 1}}$ is the J_W first column which relates the wheelchair translation velocity along the track \dot{S} to the robotic arm end-effector Cartesian velocities, and $J_{W\phi_{6 \times 1}}$ is the J_W second column which relates the wheelchair angular velocity $\dot{\phi}$ to the robotic arm end-effector Cartesian velocities.

From Equation (8.15), $\phi = \frac{1}{\rho}S$; and $\dot{\phi} = \frac{1}{\rho}\dot{S}$

$$J_{GEW_{6 \times 2}} \cdot \dot{q}_W = \begin{bmatrix} J_{WS} & J_{W\phi} \end{bmatrix} \begin{bmatrix} \dot{S} \\ \frac{1}{\rho}\dot{S} \end{bmatrix} = \begin{bmatrix} J_{WS} + \frac{1}{\rho} \cdot J_{W\phi} \end{bmatrix} [\dot{S}] = [J_{\dot{S}}][\dot{S}]$$

Therefore, Equation (8.17) can be modified as follows:

$$\dot{r}_{GE} = \begin{bmatrix} J_{GEA_{6 \times 7}} & \vdots & J_{\dot{S}_{6 \times 1}} \end{bmatrix} \begin{bmatrix} \dot{q}_A \\ \dot{S} \end{bmatrix} = J_{A\dot{S}} \begin{bmatrix} \dot{q}_A \\ \dot{S} \end{bmatrix} \quad (8.18)$$

Equation (8.18) will allow finding the wheelchair translations along the prespecified wheelchair track. To fully determine the wheelchair pose, the corresponding wheelchair angular velocity must be determined. This will be explained next.

8.3.2.2 Wheelchair Angular Velocity $\dot{\phi}$

Once the wheelchair linear velocity \dot{S} along the track is determined, the wheelchair angular velocity can be determined using following equation:

$$\dot{\phi} = \frac{\dot{S}}{\rho} \quad (8.19)$$

By finding the linear \dot{S} and angular $\dot{\phi}$ velocities, the wheelchair driven wheels' velocities ($\dot{\theta}_l$, and $\dot{\theta}_r$) can be determined as follows:

$$\begin{bmatrix} \dot{\theta}_l \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \frac{1}{L_5} & \frac{-L_1}{2L_5} \\ \frac{1}{L_5} & \frac{L_1}{2L_5} \end{bmatrix} \begin{bmatrix} \dot{S} \\ \dot{\phi} \end{bmatrix} \quad (8.20)$$

Using the driven wheels' velocities, the final pose of the wheelchair can be calculated using Equations (3.17), (3.18), and (3.19).

8.3.2.3 Tested Cases

Same cases as in Section 6.2 will be tested. These cases are: predefined-translation case, LN undefined-translation case, and MM undefined-translation case. The following is a quick revisit to the previously mentioned cases.

1. Predefined-Translation Case: This case has been established in Section 6.2 for a general case. As it was stated earlier in this case, both the end-effector and mobile base are completely predefined. Therefore, for WMRA system, Equation (6.4) can be modified as follow:

$$\dot{r}_{ES} = \begin{bmatrix} \dot{r}_{GE} \\ \dots \\ \dot{S} \end{bmatrix} = \begin{bmatrix} J_{GE_{A_1 \times 7}} & \vdots & J_{\dot{S}} \\ \dots & \dots & \dots \\ [0]_{1 \times 7} & \vdots & 1 \end{bmatrix} \begin{bmatrix} \dot{q}_A \\ \dot{S} \end{bmatrix} \quad (8.21)$$

2. LN Undefined-Translation Case: In this case, only LN solution of Equation (9.3) is used. This solution can be expressed as follows:

$$\begin{bmatrix} \dot{q}_A \\ \dot{S} \end{bmatrix} = J_{A\dot{S}}^{\#} \dot{r}_{GE} \quad (8.22)$$

where $J_{A\dot{S}}^{\#}$ is the pseudo-inverse of $J_{A\dot{S}}$.

3. MM Undefined-Translation Case: In this case, the effect of the maximized manipulability measure of the whole WMRA system on the performance of the controller algorithm is evaluated. The solution of this case can be determine using the following equation:

$$\begin{bmatrix} \dot{q}_A \\ \dot{S} \end{bmatrix} = J_{A\dot{S}}^{\#} \dot{r}_{GE} + \left(I - J_{A\dot{S}}^{\#} J_{A\dot{S}} \right) H \quad (8.23)$$

where,

$$H = k \nabla w(\theta_1, \dots, \theta_7, S, \phi) = k \begin{bmatrix} \frac{\partial w}{\partial \theta_1} & \dots & \frac{\partial w}{\partial \theta_7} & \frac{\partial w}{\partial S} & \frac{\partial w}{\partial \phi} \end{bmatrix}^T$$

and $w = \sqrt{\left(\det \left(J_{A\dot{S}} \ J_{A\dot{S}}^T \right) \right)}$

8.3.2.4 Simulation Results and Discussion

The same MATLAB simulated WMRA is used to implement and test the proposed controller scheme. Different sinusoidal trajectories are implemented to evaluate the performance and the effectiveness of this controller.

The initial joint angles for the arm in degrees are $q_{A_0} = [90 \ 0 \ -90 \ -90 \ 30 \ 90 \ 0]$ degrees. The sinusoidal trajectory for the end-effector and the wheelchair are of the form: $r(x) = d_r + a_r \sin\left(\frac{\pi x}{f}\right)$ where a and f are the amplitude and frequency of the sine wave, respectively, and d is the initial position. The end-effector has sinusoidal trajectory in X and Z axes as follows: $y(x) = d_y + a_y \sin\left(\frac{\pi x}{f}\right)$ and $z(x) = d_z + a_z \sin\left(\frac{\pi x}{f}\right)$. For the end-effector trajectory, the total traveled distance in the X-direction is $x = 7000$ with a step of $5(mm)$, $f = 3500$, $d_y = 842.6$, $a_y = -200$, $d_z = 610$, and $a_z = -250$. For the wheelchair trajectory, $y(x) = d_w + a_w \sin\left(\frac{\pi x}{f}\right)$ where $f = 6000$, $d_w = 0$ and $a_w = -500$. The wheelchair initial pose is $q_w = [x \ y \ \phi]^T = [0 \ 0 \ -14.7^0]^T$.

The WMRA system task is to follow the end-effector and the wheelchair trajectories. The end-effector should keep the same initial orientation all over its trajectory. The controller objective is to control the end-effector to follow its predefined trajectory while allowing the wheelchair to follow its track with no constraints on its translation along its predefined track. The following are the simulation results for the three cases:

1. Predefined-Translation Case:

In this case, both the end-effector and wheelchair trajectories are fully predefined. Therefore, the translations of both end-effector and wheelchair along their trajectories are known. Figure 8.10 shows a sequence of WMRA system locations for the end-effector and the wheelchair along their trajectories. For the clarity of the figure, only coordinate frames for both end-effector and the wheelchair are shown. It is noticeable that the coordinate frames are equally spaced. Also, the end-effector coordinate frames are not at the same orientation throughout the trajectory, and sometimes are not

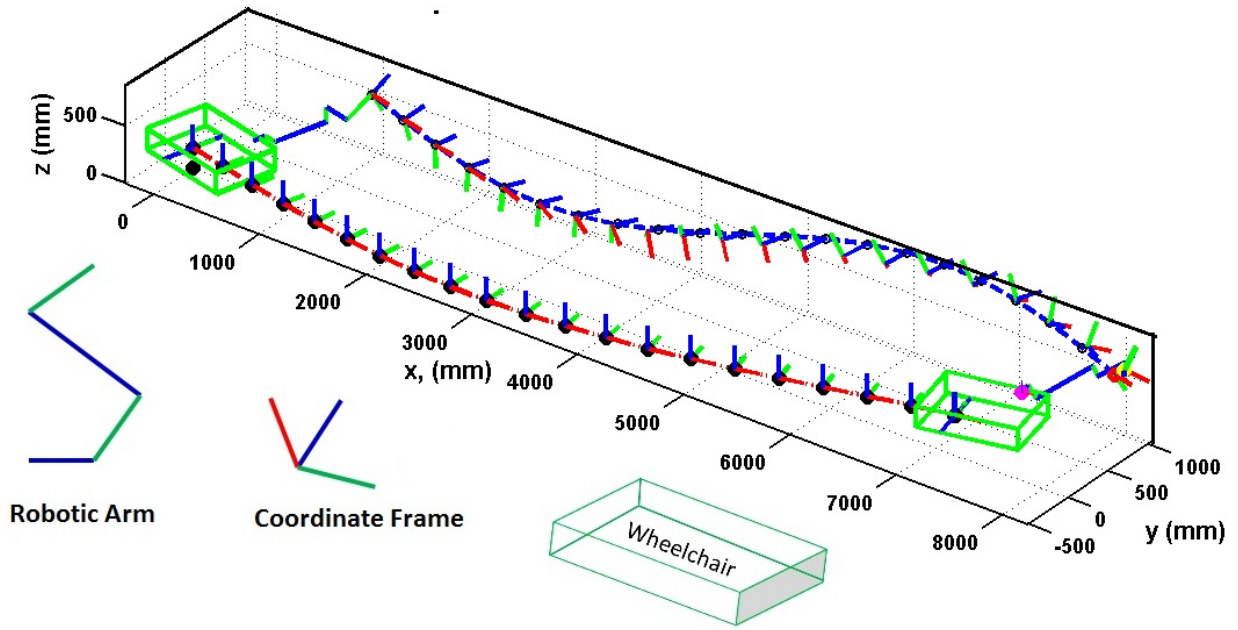


Figure 8.10: The WMRA system locations sequence for both the end-effector and the wheelchair on their trajectories in the case of predefined-translations.

laying on the trajectory. This indicates that the WMRA system cannot track both the trajectories at the same time because of its physical limitation. Figure 8.11 shows the end-effector trajectory tracking error. In this figure, the end-effector error in xy and

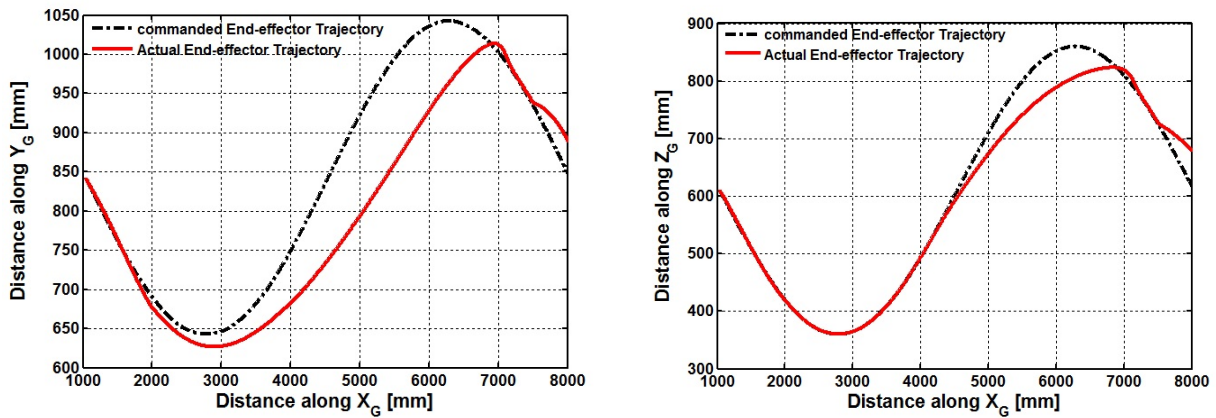


Figure 8.11: End-effector trajectory tracking error in xy and xz planes.

xz planes are presented to show the error magnitude in each plane. As shown in the

figure, the error in xy plane is more than that in the xz plane. As a conclusion, the WMRA system has failed to track both trajectories using this approach.

2. LN Undefined-Translation Case:

In this case, the least-norm solution is used to define the translation of the wheelchair along its trajectory. As shown in Figure 8.12, the end-effector coordinate frames are equally spaced because the end-effector translations along its trajectory are predefined while the coordinate frames of the wheelchair are at different spaces due to unequal translations. There are no trajectory tracking errors for the end-effector or

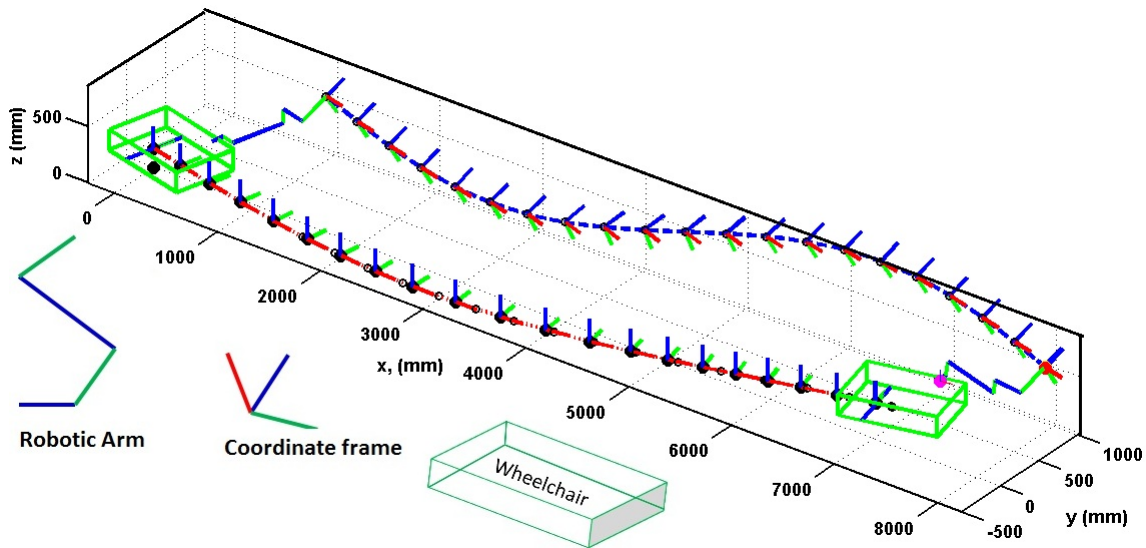


Figure 8.12: The sequence of locations for both the end-effector and the wheelchair on their trajectories in the case of the LN solution.

the wheelchair. In this case, the WMRA system can accurately track both trajectories without errors.

3. MM Undefined-Translation Case:

In this case, the effect of using an optimization criterion on the performance of the controller is evaluated. Whole WMRA system manipulability measure maximization is the performance criterion that was used in this case. Figure 8.13 shows a sequence of

the end-effector and the wheelchair locations along their trajectories. Compared to the

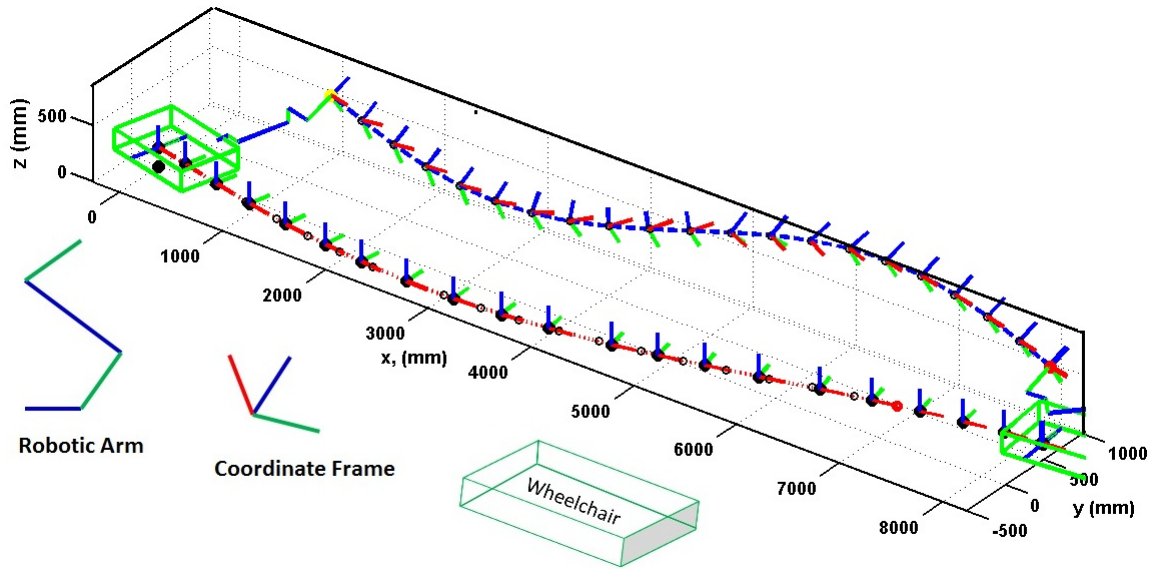


Figure 8.13: The sequence of locations for both the end-effector and the wheelchair on their trajectories, in the case of maximizing the manipulability measure.

previous figure, the WMRA system configurations are different due to the maximization of the WMRA system manipulability measure.

Figure 8.14 shows the wheelchair velocities for the three cases. As shown in the figure, the wheelchair velocity, for the predefined case, is almost constant because the wheelchair translation along its track is the same for each instance, while in the other cases, the wheelchair translations are varying according to the velocities that are commanded by the control algorithm. This explains the irregular distribution of the wheelchair coordinate frames in Figures 8.12 and 8.13.

The wheelchair velocity in the case of least-norm solution is usually less than the wheelchair velocity in the case of manipulability measure maximization. This is expected because the least-norm solution tends to minimize the joint velocities. For the manipulability measure optimization case, a high wheelchair velocity can be noticed when using high gain in the calculation of vector H in Equation (8.23). As a result of wheelchair velocity variation, the wheelchair translations also vary. Figure 8.15 shows the integration of the wheelchair

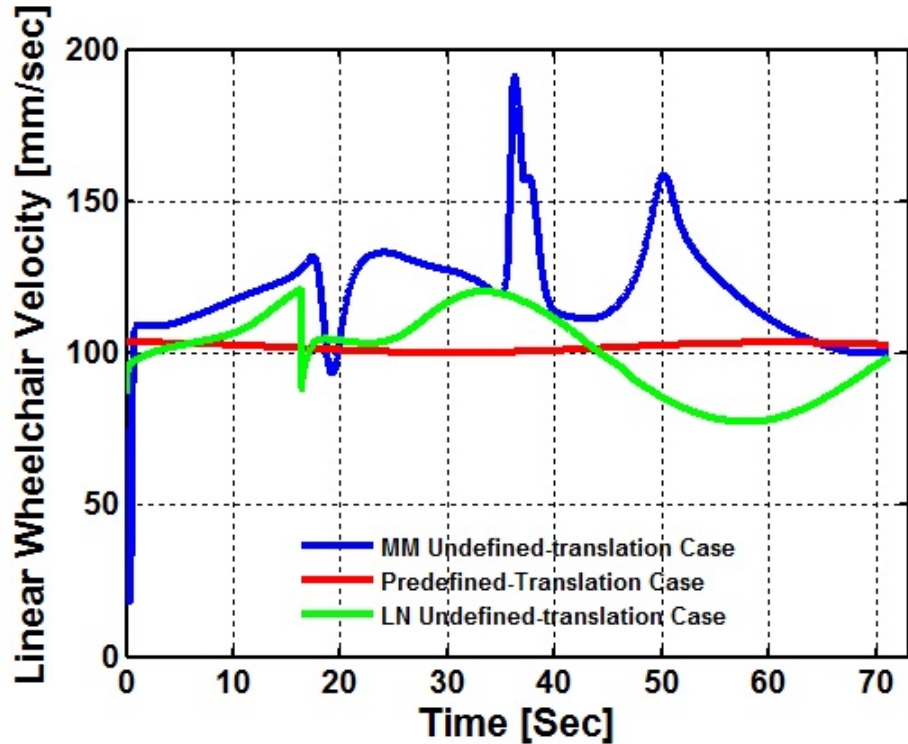


Figure 8.14: The wheelchair linear velocities along its trajectory for the three cases.

translation along its trajectory for the three cases. As expected from Figure 8.14, the maximization of the manipulability measure case has the highest wheelchair translations due to the high wheelchair velocities.

The last comparison is the manipulability measure. Figure 8.16 shows the manipulability measure of the whole WMRA system for the three cases. The MM undefined-translation case has the highest manipulability measure due to the use of the maximization of manipulability measure. Adjusting the gain in the calculation of vector H in Equation (8.23) can produce high manipulability measure, but that can increase the velocity of the wheelchair causing instability of the system.

8.4 Summary

In this chapter, the two dual-trajectory control algorithms are implemented on MATLAB simulated WMRA system. The first one is based on the spherical control variables, and the

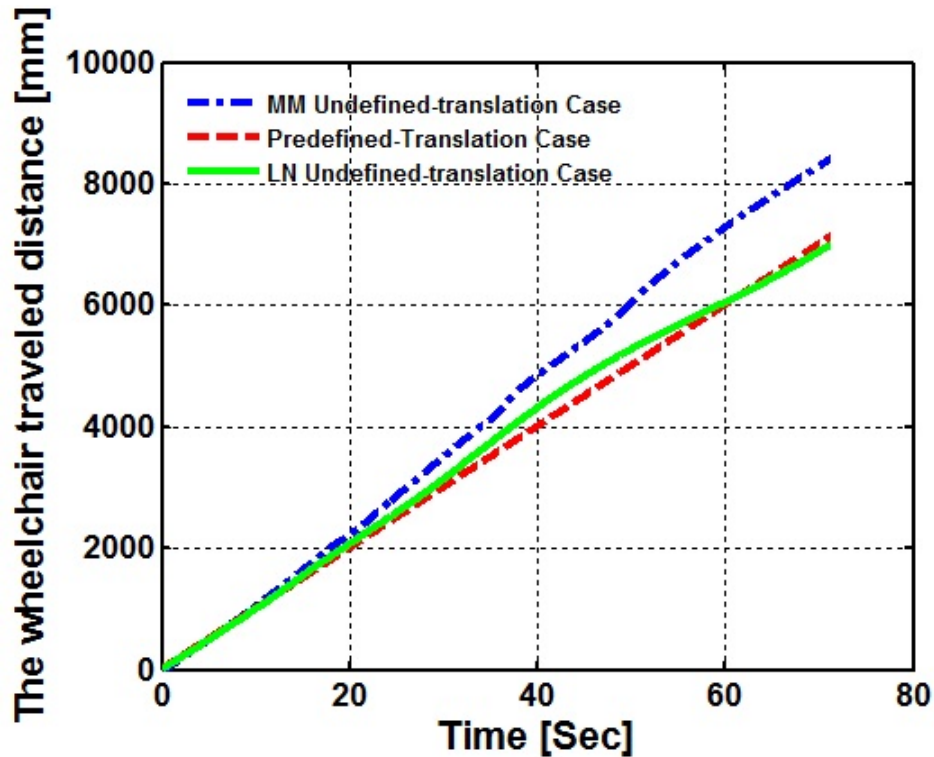


Figure 8.15: The integration of wheelchair translations along its trajectory for three cases.

second is the optimized mobile base translation along a predefined track. These two control schemes were introduced in Chapters 5 and 6.

First, three spherical control variables were introduced to the task vector. These three control variables are D , α and β . The main purpose of these variables is to relate the relative wheelchair motion to the robotic arm joint angles. By changing the values of these variables, the wheelchair pose can be controlled according to the end-effector pose and vice versa. The task priority redundancy resolution scheme was used to solve for the resolved rate solution for tracking dual-trajectory and alternate the order of priority between the end-effector and wheelchair trajectories. This scheme was used with SR-Inverse method to stabilize the system, and GPM to maximize the manipulability measure of the whole system. The results demonstrate the ability of the WMRA to follow the dual-trajectory and successfully alternate the priority between the two trajectories while maximizing the WMRA system manipulability measure.

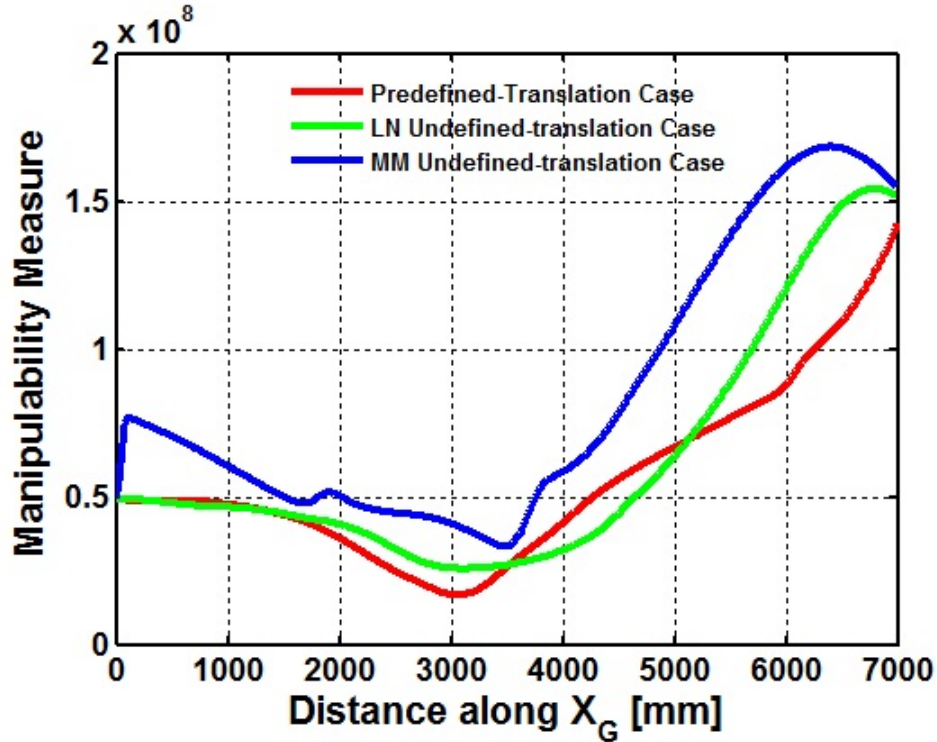


Figure 8.16: The manipulability measure of whole WMRA system for the three cases.

Second, the control algorithm of optimizing the wheelchair translation along prespecified track is implemented. This control scheme is capable of tracking the end-effector trajectory and wheelchair track. The task is to keep the mobile base on its prespecified track and, at the same time, keep the end-effector on its independent trajectory. Using the proposed control technique, the wheelchair is given the ability to move forward and backward with different velocities along its track to allow the WMRA system to successfully execute its task. The results show that this controller was able to successfully track both the end-effector trajectory and the wheelchair track when other methods failed. General case of trajectory tracking is compared with the LN undefined-translation and MM undefined-translation cases.

This simulation work is a step towards implementing these control schemes on the physical WMRA system hardware. The next chapter will present the implementation and testing of both control algorithms on the WMRA system hardware.

CHAPTER 9

HARDWARE IMPLEMENTATION AND RESULTS FOR THE WMRA SYSTEM

9.1 Introduction

Many people with disabilities rely on a caregiver's assistance to perform essential activities of daily living (ADLs) such as taking medications, walking, and feeding. Using an assistive device that is capable of providing independent assistance and mobility can have a positive impact on increased self-sufficiency, quality of life, and reduced dependence on caregivers.

The demand for integrated assistive systems is rapidly growing. A wheelchair mounted robotic arm (WMRA) can enhance the manipulation capabilities of people with disabilities who are using power wheelchairs. WMRA is an intelligent system that combines the mobility of the wheelchair and the manipulation of a robotic arm in an effort to improve performance, usability and control, as well as, reduce mental load of the user while maintaining cost competitiveness.

The WMRA system consists of a standard power-wheelchair and a 7 DoF robotic arm (manipulator). As shown in Figure 9.1, two prototypes of WMRA were designed and built in the Center for Assistive, Rehabilitation and Robotics Technologies (CARRT) at the University of South Florida [41,95]. The WMRA-II system was chosen for implementing, testing and evaluating the performance of the proposed dual-trajectory tracking algorithms. These algorithms are dual-trajectory tracking using spherical control variables and dual-trajectory tracking with optimized mobile base translation along a predefined track. The WMRA sys-



WMRA-I System.



WMRA-II System.

Figure 9.1: Two WMRA system prototypes.

tem is an assistive device mainly used for helping individuals with limited upper mobility to perform activities of daily living (ADL's).

The main goal of this work is to control the mobility of the power wheelchair and the manipulation of the robotic arm using an algorithm to control two independent trajectories, one for the end-effector, and the other for the mobile platform. This is a step towards implementing completely autonomous ADL tasks such as “go and open the door” and “pick up an object”.

9.2 Hardware Design of the WMRA-II System

The arm is a 7 DoF design, all the joints are revolute. Throughout the arm, all adjacent axes are perpendicular to each other. This meets two goals, which are: first, simplify the mechanical design, and second, simplify the kinematic model of the arm. Each joint has a high-reduction gearhead, spur-gear reduction, a DC motor with encoder, and an aluminum

bracket that holds the two components and connects two adjacent links. The D-H parameter can be found in the Table 8.1.

There are many components that are integrated into the WMRA-II design, such as DC servo motors, harmonic drive gear heads, control and amplifier boards, and wiring material. Figure 9.2 shows three components of the robotic arm which are the Galil motion control board, the harmonic drive gear head and the Maxon Precision DC motor.



Harmonic Drive Gear Head Assembly.



Maxon Precision DC Motor.

Figure 9.2: Control board, harmonic drive and joint motor.

The WMRA-II is mounted on a differentially driven mobile base (wheelchair) that represents a 2 DoF planar system [52]. Two of the DoFs are provided by the nonholonomic motion of the wheelchair.

9.3 Controller Hardware

9.3.1 Robotic Arm Controller Hardware

There is a harmonic drive for each joint and the harmonic drive torque handling capability varies according to the joint position. It is in range from 108 (NM) for the first joint to 8.9 (NM) for the seventh joint. All the harmonic drives have reduction ratios of 100:1.

Most of the power wheelchairs come with a set of two 12V batteries in series that give 24 volts. However, a voltage reducer was installed to provide the necessary voltage to the controlling board. To efficiently process the inputs and outputs of the control and feedback systems of the motors for this work, it was crucial to have a good control board. For this purpose, the Galil Motion DMC-2183 Control board was selected to be installed for this application. The DMC-2183 has the ability to run up to eight motors simultaneously. This board is used to control the seven joints of the arm and the gripper. Another Galil Motion Control board, the DMC-41x3, was installed to control the wheel motors.

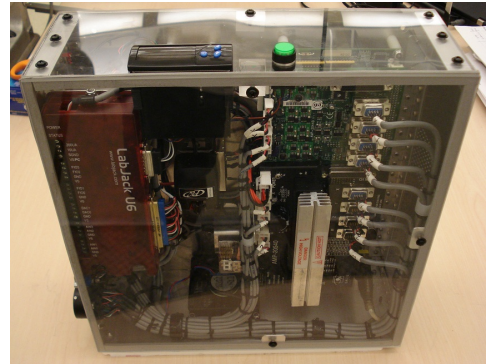
A control box was designed and built to control the WMRA-II. Figure 9.3 shows the WMRA-II mounted onto a wheelchair with the control box attached to the back of the wheelchair.

9.3.2 Wheelchair Controller Hardware

The WMRA-II system consists of a redundant robotic arm installed on a modified power wheelchair “Ability”. The wheelchair has been modified by installing incremental encoders directly onto the wheel motors as shown in Figure 9.4. This allows for the individual control of each wheel with feedback, independently. Another control box was designed and built to control the wheelchair. Figure 9.4 shows the wheelchair control box with the Galil Motion DMC-41x3 control board attached to the back of the wheelchair.

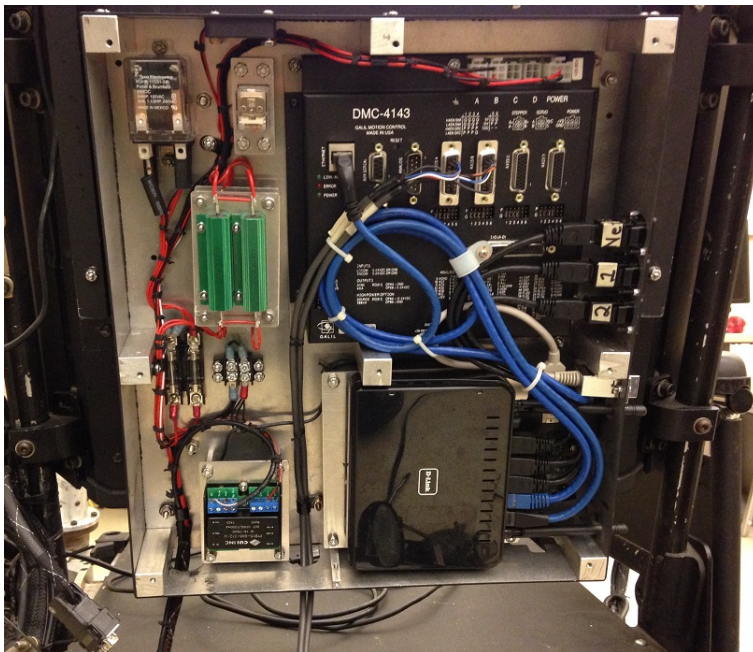


WMRA-II System.



Arm Control Box.

Figure 9.3: WMRA-II system with the robotic arm control box.



Galil Motion Control DMC41x3 Board installed in wheelchair control box



Wheel Motor Encoder

Figure 9.4: Wheelchair control box components and wheel motor encoder.

9.4 Control Algorithm Implementation

The purpose of these algorithms is to track mobile manipulator dual trajectory (end-effector and mobile platform trajectories.) The two control algorithms that are implemented are dual-trajectory tracking with task priority using three spherical control variables (D , α and β) and dual-trajectory tracking with mobile base optimized translation along a predefined track. These two algorithms can be summarized as follows.

9.4.1 Dual-Trajectory Tracking Using D , α and β

In the case of dual-trajectory tracking with task priority using spherical control variables, both the end-effector and the mobile base trajectories are fully predefined. This means that the pose of the mobile manipulator is known in each time instance. This control algorithm is implemented for two instances as follows:

1. First priority set to end-effector trajectory:

$$\dot{r}_{GE} = \begin{bmatrix} \dot{x}_{GE} \\ \dot{y}_{GE} \\ \dot{z}_{GE} \\ \omega_{x_{GE}} \\ \omega_{y_{GE}} \\ \omega_{z_{GE}} \\ \dot{D} \\ \dot{\alpha} \end{bmatrix} = J_{sys} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \\ \dot{\theta}_7 \\ \dot{S} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} J_{GEA_{6 \times 7}} & J_{GEW_{6 \times 2}} \\ \hline J_{D\alpha_{2 \times 9}} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \\ \dot{\theta}_7 \\ \dot{S} \\ \dot{\phi} \end{bmatrix} \quad (9.1)$$

2. First priority set to wheelchair trajectory:

$$\dot{r}_{GE} = \begin{bmatrix} \dot{S} \\ \dot{\phi} \\ \dot{D} \\ \dot{\alpha} \\ \dot{\beta} \\ \omega_{x_{GE}} \\ \omega_{y_{GE}} \\ \omega_{z_{GE}} \end{bmatrix} = J_{sys} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \\ \dot{\theta}_7 \\ \dot{S} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} J_{S\phi_{2 \times 9}} \\ J_{D\alpha\beta_{3 \times 9}} \\ J_{\omega_{3 \times 9}} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \\ \dot{\theta}_7 \\ \dot{S} \\ \dot{\phi} \end{bmatrix} \quad (9.2)$$

9.4.2 Dual-Trajectory Tracking With Free Platform Translation Along a Track

In the case of dual-trajectory tracking with mobile base optimized translation along a predefined track, the end-effector trajectory is fully predefined and the mobile base trajectory is partially defined. This means that for the mobile base trajectory, only the mobile base track is predefined without the mobile platform exact positions on the track. The exact mobile base locations on the track are left for the control algorithm to determine. This algorithm can be defined according to the following equation:

$$\dot{r}_{GE} = \begin{bmatrix} \dot{x}_{GE} \\ \dot{y}_{GE} \\ \dot{z}_{GE} \\ \omega_{x_{GE}} \\ \omega_{y_{GE}} \\ \omega_{z_{GE}} \end{bmatrix} = \begin{bmatrix} J_{GE_{A_6 \times 7}} & \vdots & J_{\dot{S}_{6 \times 1}} \end{bmatrix} \begin{bmatrix} \dot{q}_A \\ \dot{S} \end{bmatrix} = J_{A\dot{S}} \begin{bmatrix} \dot{q}_A \\ \dot{S} \end{bmatrix} \quad (9.3)$$

9.5 Hardware Implementation

The flowchart, shown in Figure 9.5, presents the control algorithm implementation on the WMRA hardware. As it was stated previously, the dual-trajectory of the WMRA-II

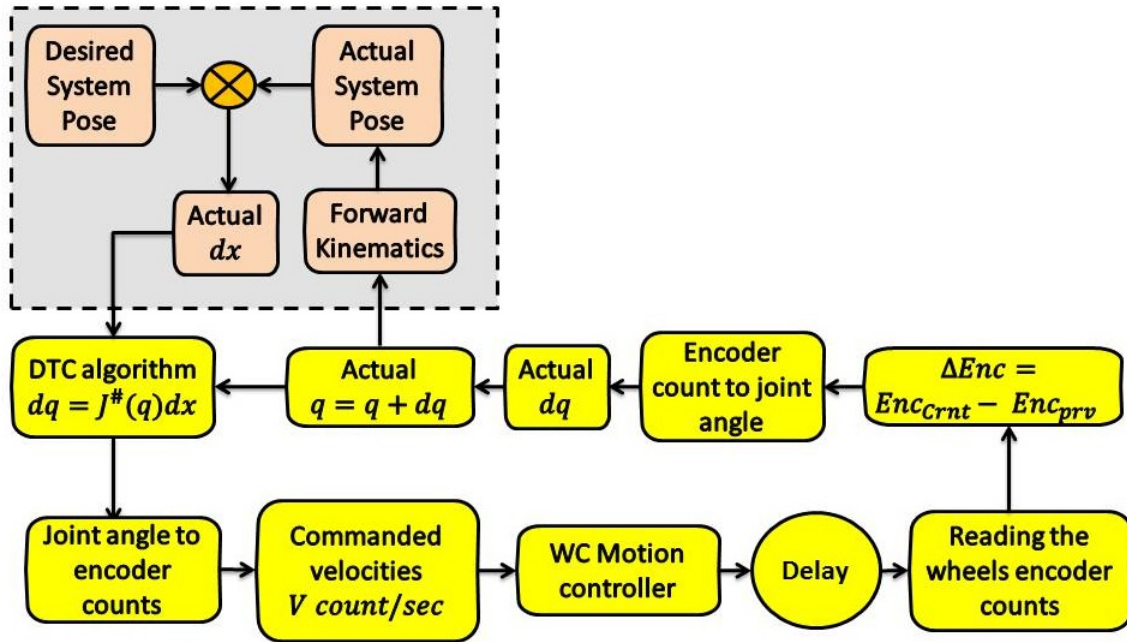


Figure 9.5: Control algorithms implementation flowchart.

system is fully or partially defined. The error between the actual and desired wheelchair poses, named dx , is determined for both the end-effector and the wheelchair according to the control algorithm implemented. The nine joint velocities for the WMRA system are determined according to the following equation:

$$dq = J^\#(q) dx \quad (9.4)$$

The joint velocities are converted from (rad/sec) to ($encoder\ counts/sec$). For the wheelchair, the rotation $\dot{\phi}$ and translation \dot{S} velocities are converted to right $\dot{\theta}_r$ and left

$\dot{\theta}_l$ wheel angular velocities as follows:

$$\begin{bmatrix} \dot{\theta}_l \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \frac{1}{w_r} & \frac{-w_b}{2w_r} \\ \frac{1}{w_r} & \frac{w_b}{2w_r} \end{bmatrix} \begin{bmatrix} \dot{S} \\ \dot{\phi} \end{bmatrix} \quad (9.5)$$

where w_r is the wheel radius and w_b is the wheel base length. And the wheel angular velocities converted to encoder count velocities as follows:

$$\begin{bmatrix} \dot{v}_l \\ \dot{v}_r \end{bmatrix} = K_v \frac{\text{encoder counts per wheel revolution}}{2\pi i} \begin{bmatrix} \dot{\theta}_l \\ \dot{\theta}_r \end{bmatrix} \quad (9.6)$$

where K_v is a gain to speedup the system response. These velocities are sent to the Galil Motion DMC-41x3 control board. A similar procedure is implemented for the 7 arm joint velocities. A command is sent to the Galil boards to read the encoder counts for the 9 DC motors. The actual executed joint angles for the current instance are determined by finding the difference in encoder counts for all of the motors as follows:

$$\Delta Enc = Encoder_{current} - Encoder_{previous} \quad (9.7)$$

A reverse process is accomplished to convert the encoder counts to joint angles. These joint angles are named “ dq_{actual} ”. The final accumulated joint angles are determined as follows:

$$q_{current} = q_{previous} + dq_{actual} \quad (9.8)$$

Using the system forward kinematics, the actual poses of the wheelchair and the end-effector are determined. In this step, any pose errors will be considered and they will be compensated for by scaling the system joint velocities according to the error dx in Equation(9.4).

9.6 Hardware Results

C++ programming language was used to implement the controller on the WMRA-II system (Refer to Appendix C). Different trajectories are implemented to evaluate the performance and the effectiveness of these controllers.

9.6.1 Dual-Trajectory Tracking Using D , α and β

The initial joint angles for the arm in degrees are $q_{A_0} = [45 \ 90 \ 90 \ 90 \ 0 \ 0 \ 90]$ degrees. The joint limits for the WMRA-II are $q_{max} = [170 \ 170 \ 170 \ 170 \ 170 \ 100 \ 200]$ and $q_{min} = [-170 \ -170 \ -170 \ -170 \ -170 \ -100 \ -200]$ degrees. The total traveled distance in the X-direction is $x = 6000$ with a step of $5(mm)$. For the end-effector trajectory, $y(x) = d_y + a_y \sin\left(\frac{\pi x}{f}\right)$ and $z(x) = d_z + a_z \sin\left(\frac{\pi x}{f}\right)$ where a and f are the amplitude and frequency of the sine wave respectively, and d is the initial position of the end-effector, $f = 5000$, $d_y = 317.7$, $a_y = 700$, $d_z = 1150$, and $a_z = 100$. For the wheelchair, $y(x) = d_w + a_w \sin\left(\frac{\pi x}{f}\right)$ where $f = 2500$, $d_w = 0$ and $a_w = -200$. The WMRA-II system task is to follow both the end-effector and the wheelchair trajectories. The end-effector should keep the same initial orientation all over its trajectory. Variable D is limited to $1500(mm)$. The hardware implementation results are compared against the MATLAB simulation results. It is worth mentioning that the end-effector and the wheelchair pose estimation in the hardware implementation are dependent on the wheel's and arm joints' encoder readings.

The priority task for this control algorithm was tested using MATLAB simulation in two cases: higher priority was given to the end-effector trajectory and higher priority was given to the wheelchair trajectory. In the hardware implementation experiment, implementation and testing was only done for the instance where all terms of the task priority Equation (5.10) were used for both priority cases. First, the case when the higher priority was given to the end-effector is discussed.

Figure 9.6 shows a 3 D plot of the end-effector's and wheelchair's commanded and actual trajectories for the real hardware implementation. As expected, when the high priority is

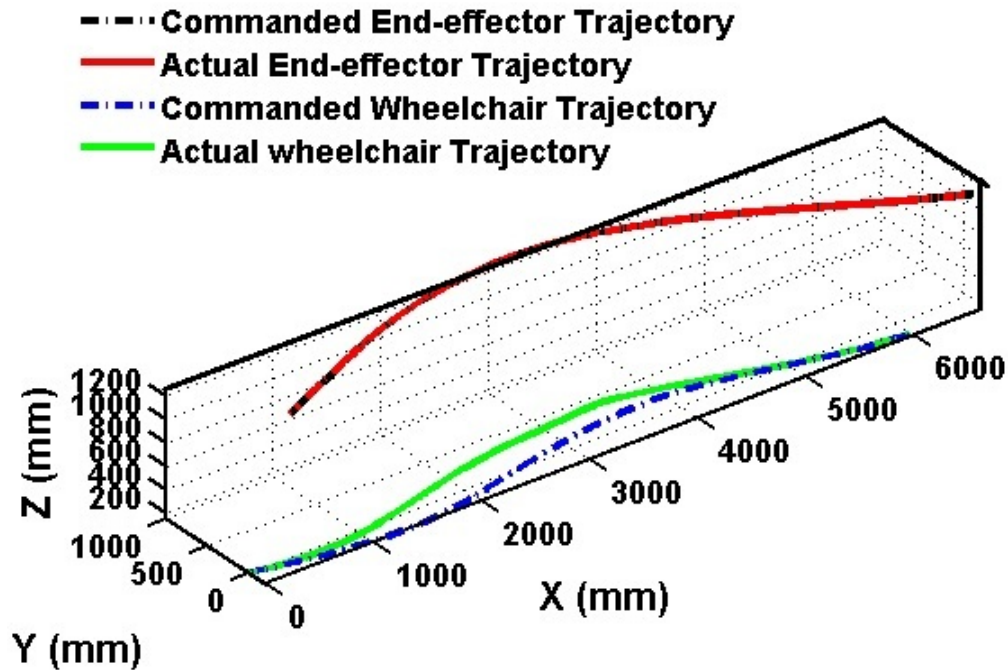


Figure 9.6: End-effector's and the wheelchair's commanded and actual trajectories for the hardware implementation (high priority is given to the end-effector trajectory).

given to the end-effector trajectory, the end-effector trajectory was always followed while the wheelchair trajectory was followed as much as possible. When it was impossible to follow the end-effector simultaneously with the wheelchair trajectory, a tracking error was introduced to the wheelchair trajectory following as in the region from 500(mm) to 5000(mm). To compare the tracking error between the real hardware and MATLAB simulation, Figure 9.7 shows XY-plane for commanded and actual wheelchair trajectory for both the WMRA-II system MATLAB simulation and the real hardware implementation. The figure shows a good agreement between the simulation and real hardware results. For a complete comparison, Figure 9.8 shows the relative position errors for wheelchair trajectory tracking for both MATLAB simulation and hardware implementation. Position error is the difference between the commanded and the actual position for both the end-effector and wheelchair. For the

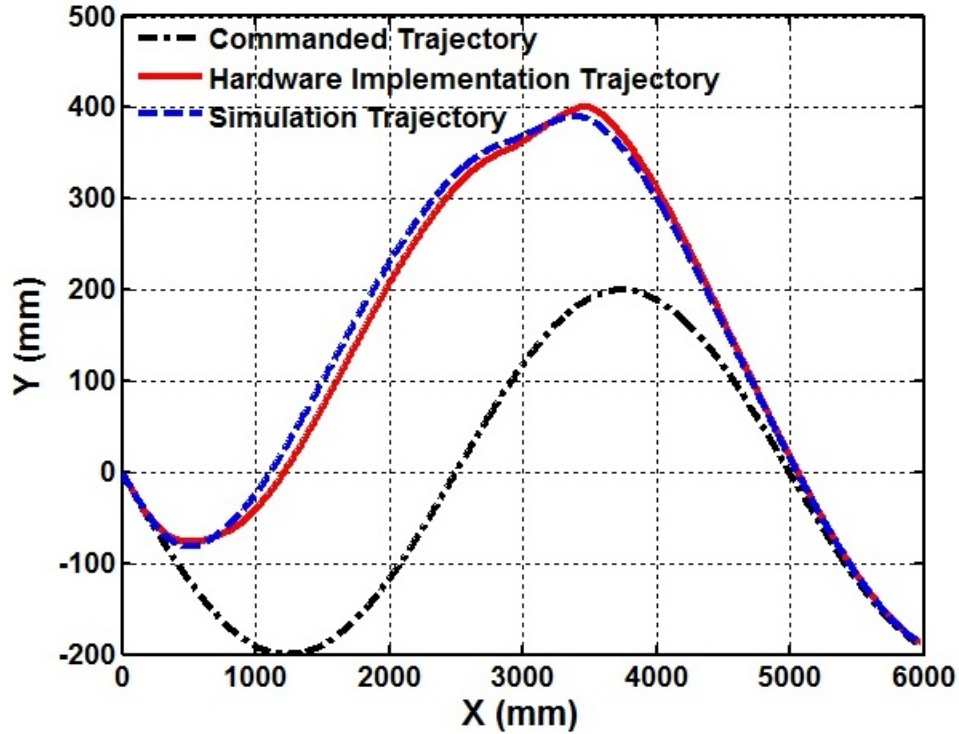


Figure 9.7: Commanded and actual wheelchair trajectory tracking for both MATLAB simulation and hardware implementation (high priority is given to the end-effector trajectory).

wheelchair results, the figure shows slightly more tracking errors compared to the previous figure. In the latter figure, the relative error is the difference between the commanded and the actual wheelchair position. This is due to the effect of the wheelchair mass on the WMRA-II system response to the changing in the commanded velocities. The end-effector position error is zero in the simulation case. The results are measured according to the wheels' encoder readings. It is theorized that the actual wheelchair and end-effector position errors are greater, if the real wheelchair and end-effector ground truth was captured. Figure 9.9 shows the whole system manipulability measure for both MATLAB simulation and the hardware implementation. The figure shows that the manipulability measure is almost the same in both cases. Figure 9.10 shows the robotic arm's seven joint angles, in degrees, for the case of hardware implementation. It shows that none of the arm joints exceeded the joint limit because of the implementation of joint limit avoidance algorithm.

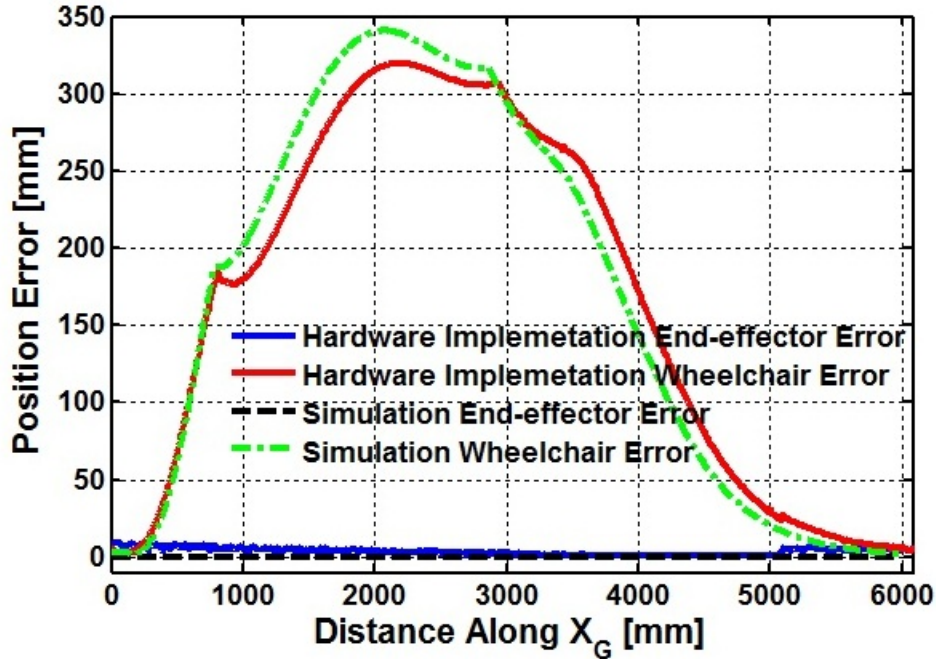


Figure 9.8: End-effector and wheelchair position errors (high priority is given to the end-effector trajectory).

The second case demonstrates when the higher priority is given to the wheelchair trajectory. All the wheelchair and end-effector initial conditions and trajectories remain the same. Contrary to the previous method, when the WMRA-II system cannot follow both trajectories, a position error is introduced to the end-effector trajectory. Figure 9.11 shows a 3 D plot of the wheelchair and the end-effector commanded and actual trajectory for real hardware implementation. As shown in the figure, the wheelchair trajectory was accurately followed while the end-effector trajectory was followed as much as possible. Therefore, when it is impossible to follow both trajectories, the WMRA-II system follows the wheelchair trajectory and introduces a position error into the end-effector trajectory. This is in line with what is expected when a higher priority is given to the wheelchair trajectory. The end-effector position error in XY and XZ planes is presented in Figure 9.12. The relative position error for the end-effector and the wheelchair for both the hardware and MATLAB simulation are presented in Figure 9.13. As shown in the figure, the position error for the wheelchair is

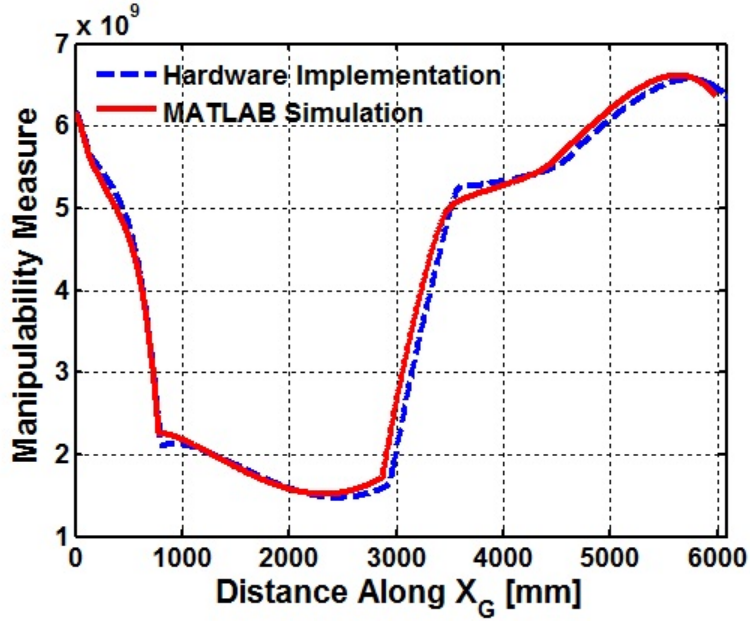


Figure 9.9: Whole system manipulability measure for the MATLAB simulation and real hardware implementation (high priority is given to the end-effector trajectory).

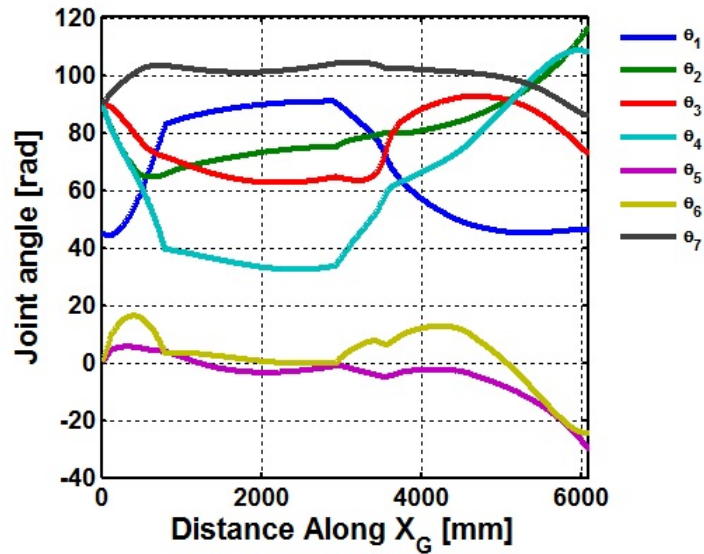


Figure 9.10: Arm joints angle (high priority is given to the end-effector trajectory).

far less than that of the end-effector. The position error for the wheelchair in MATLAB simulation is zero due to the ideal motion presented in simulation. The position error for the wheelchair (red line in Figure 9.13) in the hardware implementation is not zero and it is decreased as the wheelchair moves. Also, the wheelchair position error is propagated to

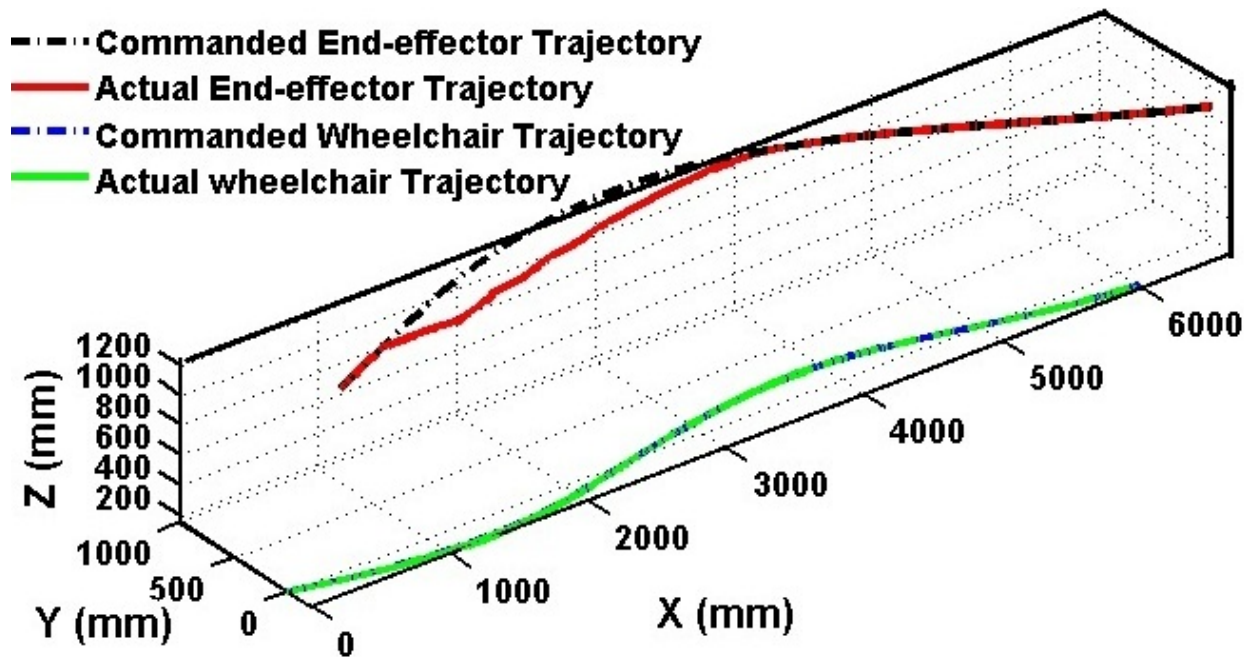


Figure 9.11: End-effector's and the wheelchair's commanded and actual trajectories for the MATLAB simulation and hardware implementation (high priority is given to the wheelchair trajectory).

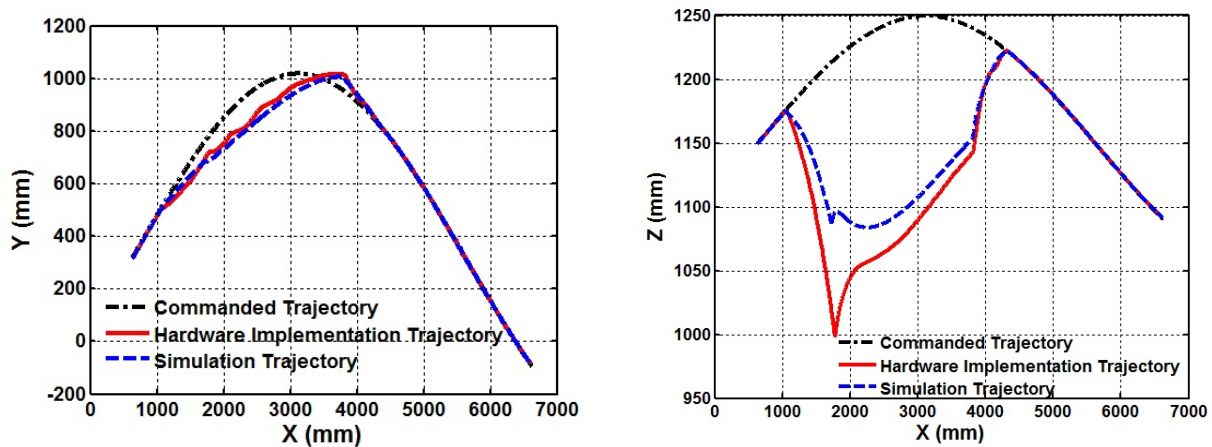


Figure 9.12: End-effector trajectory tracking error in XY and XZ planes (high priority is given to the wheelchair trajectory).

the end-effector position error. Although the arm is mounted on the wheelchair, error propagation can be prevented by compensating the wheelchair position error when calculating the end-effector position. As in the previous case with the high priority for the end-effector,

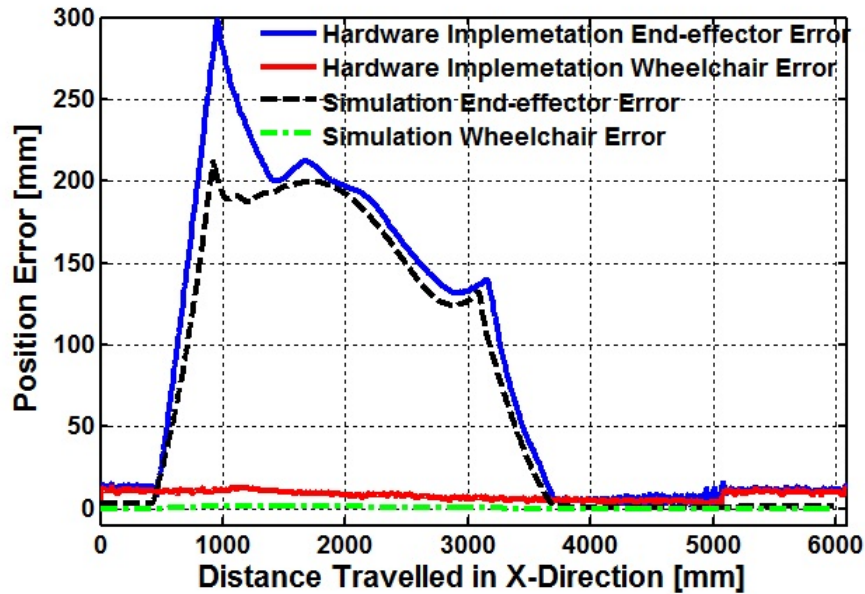


Figure 9.13: End-effector and wheelchair position errors (high priority is given to the wheelchair trajectory).

the end-effector position error in the MATLAB simulation is less than that in the hardware implementation. This is due to the slower response of the real hardware to the change in the system joint velocities. The joint angles for the arm are presented in Figure 9.14. All joint angles are within the joint limits. The last comparison is presented in Figure 9.15. The figure shows the manipulability measure for the MATLAB simulation and hardware implementations, demonstrating a good agreement between them.

9.6.2 Dual-Trajectory Tracking With Free Platform Translation Along a Track

In this algorithm implementation, the WMRA-II system task was to pick up and place an object onto a flat surface while avoiding obstacles. The task and test environment is presented in Figure 9.16. As shown in the figure, the WMRA-II system's main task is following a dual-trajectory. First, the WMRA-II system will pick up an object, and then the wheelchair will avoid an obstacle. At the same time, the end-effector will avoid another obstacle. Once the WMRA-II system finishes the avoiding process, it will place the object

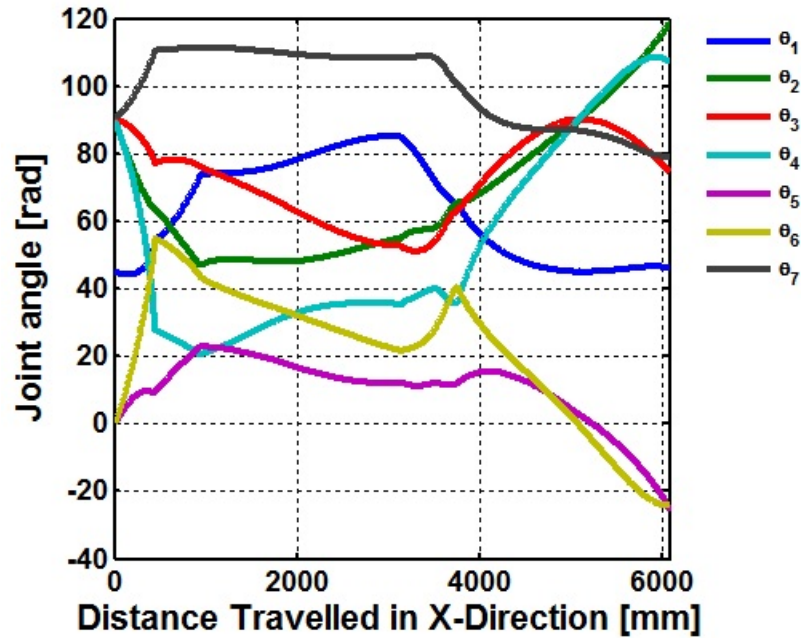


Figure 9.14: Arm joint velocities (high priority is given to the wheelchair trajectory).

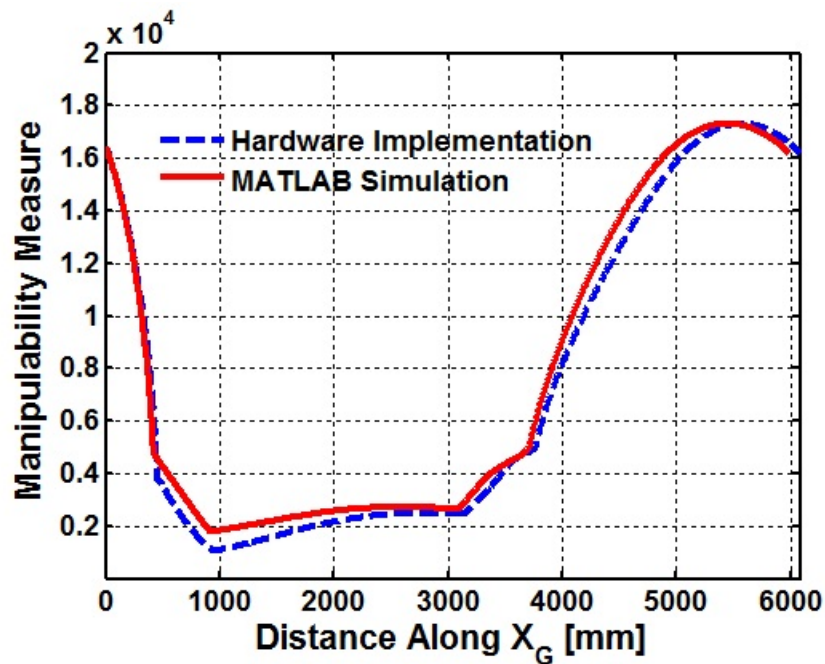


Figure 9.15: Whole system manipulability measure for the MATLAB simulation and real hardware implementation (high priority is given to the wheelchair trajectory).

and finish following the dual-trajectory. The real testing environment is shown in Figure 9.17. Three cases were tested (refer to Section 8.3.2.3 for summary of these cases). These

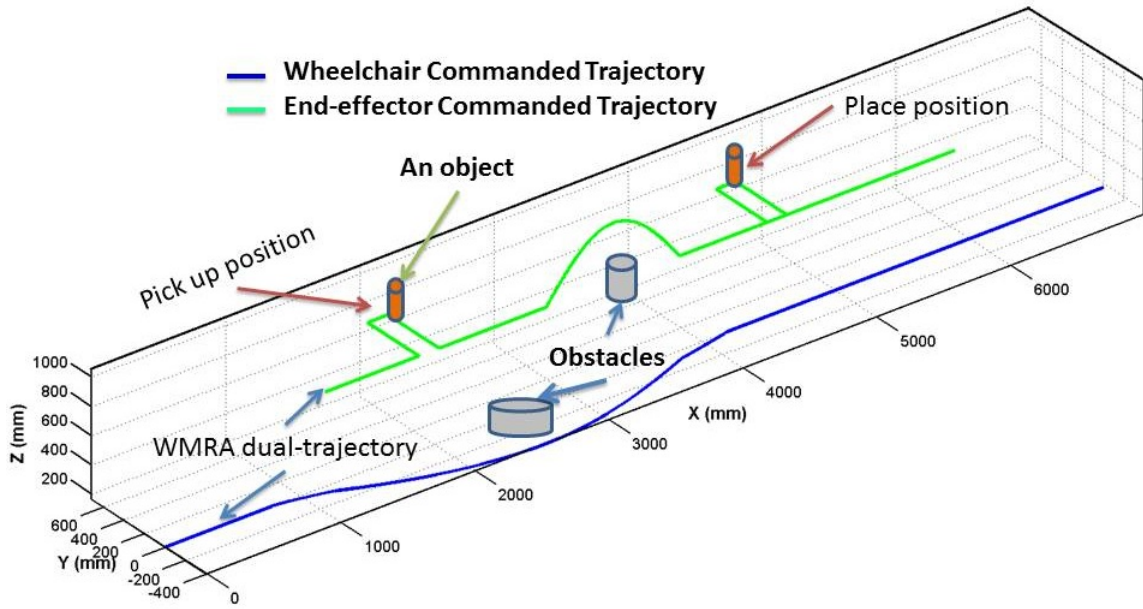


Figure 9.16: Hardware implementation testing environment with wheelchair and end-effector commanded trajectories, manipulated object and obstacles.

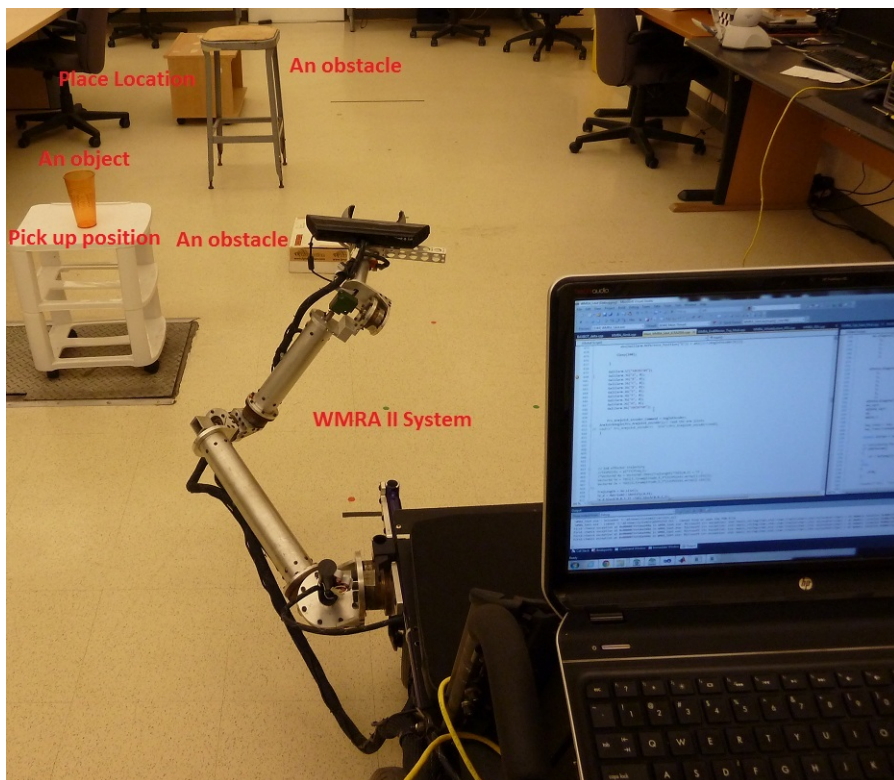


Figure 9.17: Real hardware implementation testing environment.

cases are:

1. Predefined-Translation Case.
2. LN Undefined-Translation Case.
3. MM Undefined-Translation Case.

The initial joint angles for the arm in degrees are $q_{A_0} = [165 \ 60 \ -45 \ 90 \ 150 \ 80 \ 65]$ degrees. The initial position of the end-effector is $[x_{EE} \ y_{EE} \ z_{EE}] = [1360 \ 219 \ 650.6]$ while the initial pose of the wheelchair is $[x_{WC} \ y_{WC} \ \phi_{WC}] = [0 \ 0 \ 0]$. The joint limits for the WMRA-II are $q_{max} = [270 \ 170 \ 170 \ 170 \ 170 \ 100 \ 200]$ and $q_{min} = [-170 \ -75 \ -170 \ -270 \ -170 \ -100 \ -200]$ degrees. The total traveled distance in the X-direction is approximately $x = 4600(mm)$ with a step of $2(mm)$.

The WMRA-II system's dual-trajectory consists of straight lines and curves as shown in Figure 9.15. The WMRA-II system task is to follow the preplanned end-effector trajectory and to keep the wheelchair on its preplanned track. The end-effector should keep the same initial orientation all over its trajectory. It is worth mentioning that the end-effector and the wheelchair pose estimation in the hardware implementation are dependent on the wheel's and arm joints' encoder readings.

Figure 9.18 shows the commanded and actual trajectories of the wheelchair and the end-effector for the three tested cases. As seen in the figure, the WMRA-II system was able to follow the planned dual-trajectory in both undefined cases while it failed to follow the end-effector trajectory in the predefined-translation case. In the predefined-translation case (Figure 9.18-a), the wheelchair followed its track using via points. As a result, the wheelchair velocity was constant as shown in Figure 9.19. This allowed the wheelchair was moving forward while the end-effector was moving to grasp the object. However, this made the end-effector give up its orientation in the stage of picking up the object, leading to the failure of grasping the object. Moreover, the end-effector slowly lost its position after losing

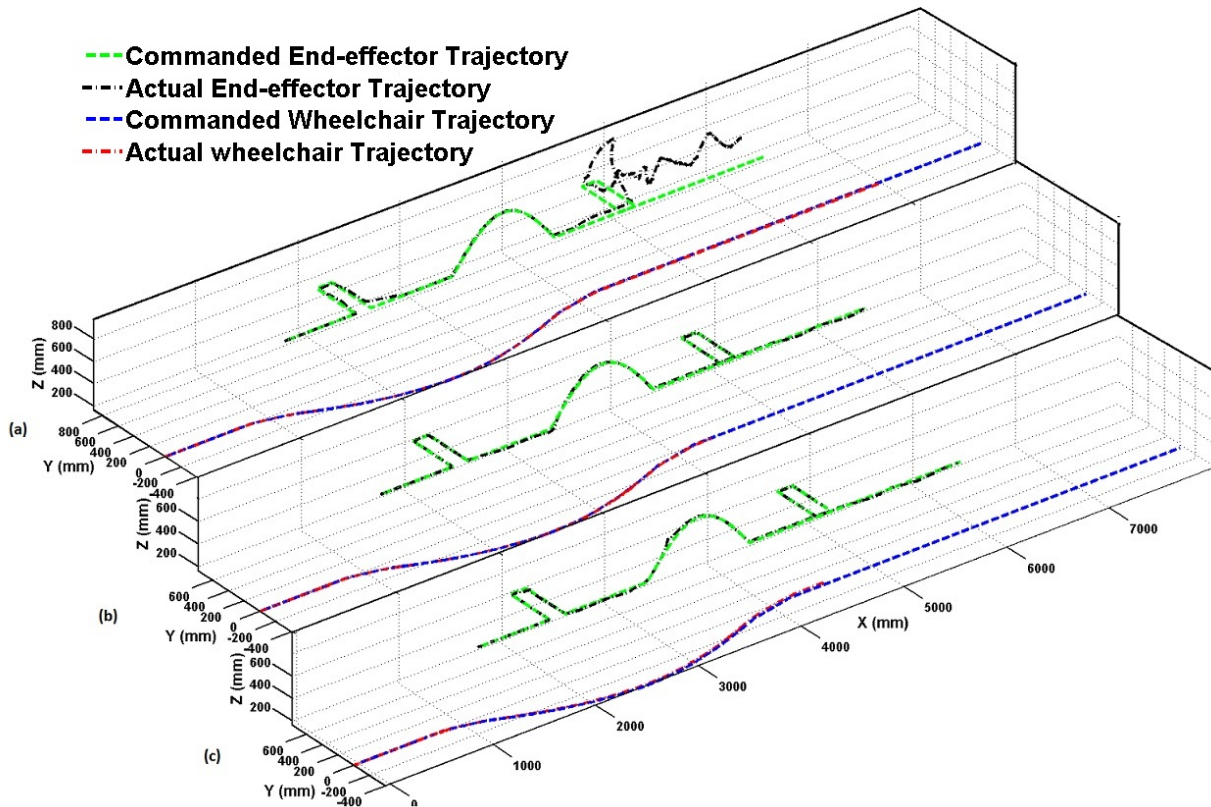


Figure 9.18: The commanded and actual trajectories of the wheelchair and the end-effector for: (a) Predefined-translation case. (b) LN undefined-translation case. (c) MM undefined-translation case.

its orientation as the task execution progressed. This became clear in the placing stage in which the end-effector completely lost its position and was pointing backwards instead of forward.

In both undefined-translation cases, the wheelchair velocities were varied according to the end-effector task, as shown in Figure 9.19. In the picking up stage, the wheelchair was slowing down, and stopping or speeding up, allowing the end-effector motion to successfully grasp the object or move back to its straight line trajectory. Another example of changing the wheelchair's velocities is in the end-effector obstacle avoiding process. When the end-effector was moving up to avoid the obstacle, the wheelchair was sped up to compensate for the end-effector's vertical motion. The same scenario can be noticed in the placing stage. In

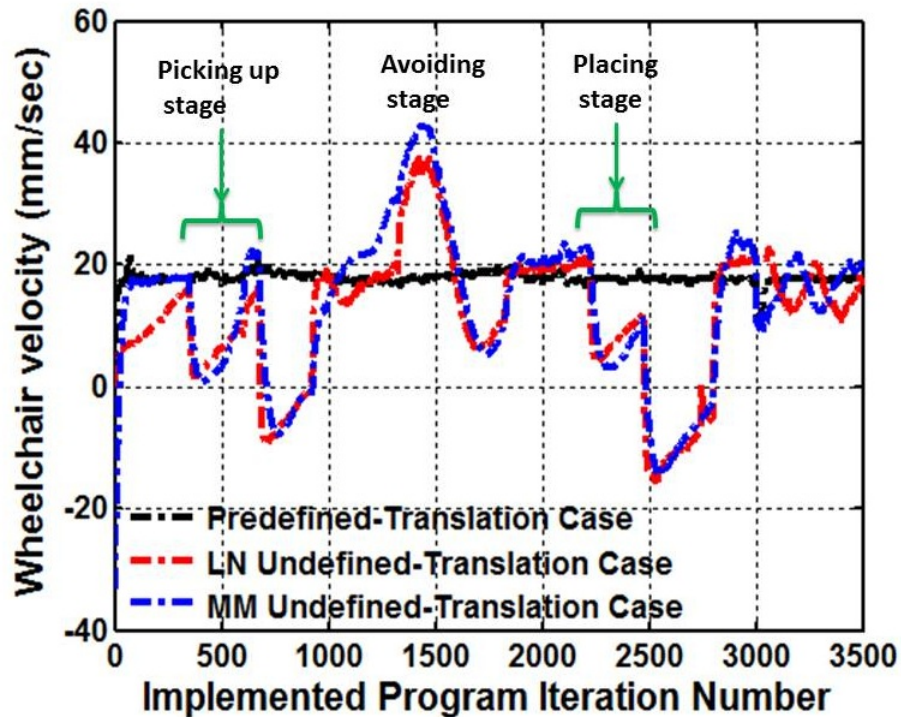


Figure 9.19: Wheelchair velocities for three test cases.

the two undefined-translation cases, the WMRA-II system successfully executed the task. The main difference between the two methods was the manipulability measure.

Figure 9.20 shows the manipulability measure for the LN and MM undefined-translation cases. As expected, the manipulability measure of the MM undefined-translation case was higher than the manipulability measure of the LN undefined-translation case. This is because of using the GPM to maximize the manipulability measure. Snapshots of the real hardware task execution are shown in Figure 9.21. These snapshots are from the implementation of LN undefined-translation case. The following is a description of the snapshots:

1. The initial configuration of the WMRA-II system.
2. The WMRA-II system moved in its straight line trajectory.
3. The end-effector moved to the left to grasp an object.
4. The end-effector grasped the object.

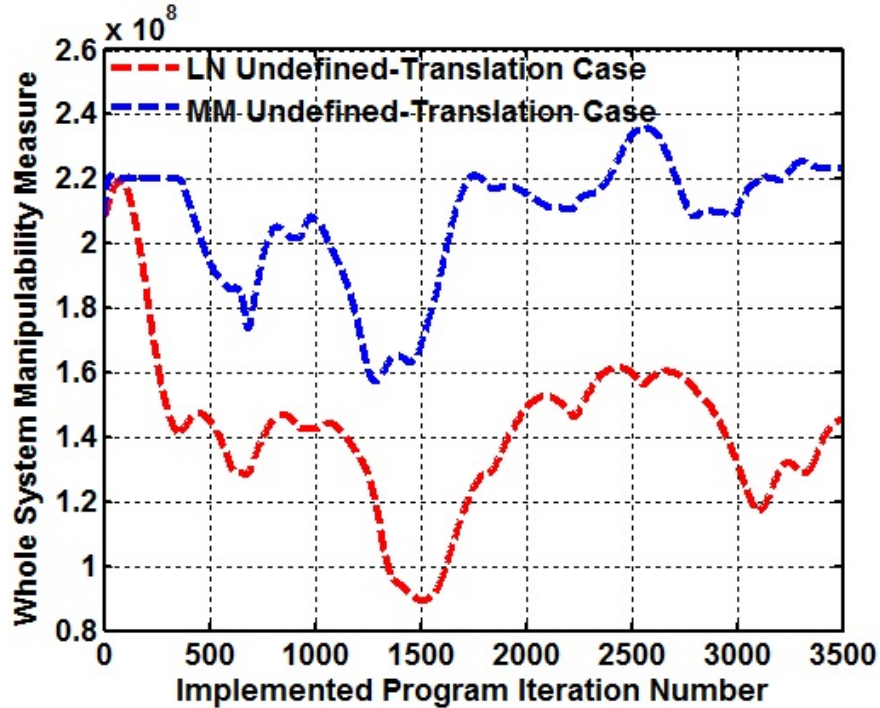


Figure 9.20: Manipulability measure of both undefined-translation cases.

5. The end-effector moved back to its straight line trajectory.
6. The end-effector moved in its straight line trajectory.
7. The end-effector moved in its straight line trajectory and the wheelchair started to avoid an obstacle.
8. The end-effector started to avoid another obstacle and the wheelchair avoided its obstacle.
9. The WMRA-II system avoided the end-effector and wheelchair obstacles.
10. The WMRA-II system completed the obstacles avoidance.
11. The end-effector moved in its straight line trajectory.
12. The end-effector moved to the left to place the object.

13. The end-effector placed the object and went back to its straight line trajectory.
14. The end-effector moved in its straight line trajectory.
15. The WMRA-II system successfully completed the task.

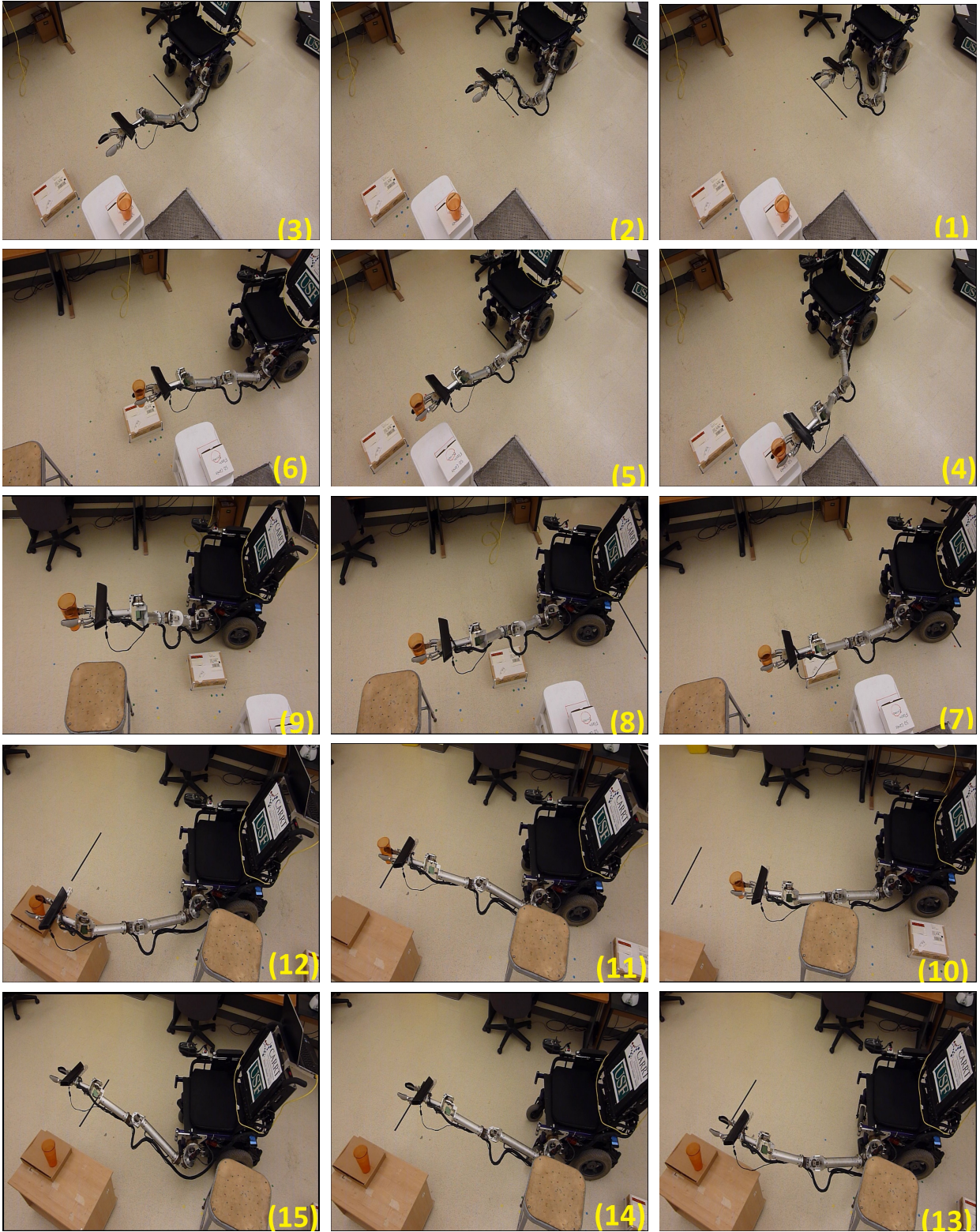


Figure 9.21: Motion sequence of WMRA-II. The wheelchair motion is from right to left.

CHAPTER 10

CONCLUSIONS AND RECOMMENDATIONS

10.1 Overview

Mobile manipulators are robotic devices that consist of a robotic arm mounted on a mobile platform. Usually combining the platform's mobility with the manipulation of the robotic arm results in a redundant system. Each subsystem of the mobile manipulator has its own trajectory (end-effector and mobile platform). In complex tasks, there is a need to control the end-effector trajectory as well as the mobile platform trajectory. This is to allow the robotic arm to be positioned at a place that supports the execution of the desired manipulation task. On the other hand, the mobile platform needs to maintain certain orientation. As a result, dual-trajectory control has a crucial role on successfully performing the desired task. Path planning of the dual-trajectory can be done online or offline. For this work, it was assumed that the trajectories of the end-effector and mobile platform were already predefined. The main focus of this work is to find a feasible solution to control the mobile manipulator when the dual-trajectory cannot be followed simultaneously due to the physical limitations of the system. Two control algorithms have been proposed and presented to solve this problem.

10.2 General Discussion

A n DoF robotic arm and a 2 DoF nonholonomic mobile platform were kinematically modeled and controlled. A combination of the two subsystems mathematical models created a $n + 2$ DoF redundant mobile manipulator. Several kinematic control methods, such as SR-Inverse, Pseudo-inverse, and Weighted Least-Norm solution, were implemented.

To cope with the problem when the dual-trajectory could not be simultaneously followed, two novel control algorithms were designed, developed, tested and implemented. These two control algorithms were able to control a mobile manipulator to follow its dual-trajectory. The control algorithms were implemented in a MATLAB simulation to test their abilities to control a redundant mobile manipulator to track its dual-trajectory.

In the first control algorithm, three new spherical variables (D , α and β) were introduced to the task vector. These variables were introduced for the following reasons: first, they define the mobile platform trajectory in relation to the end-effector trajectory and vice versa. Second, they relate the mobile platform motion to the robotic arm joint angles. This was particularly useful when it was not physically possible to follow both the end-effector and platform trajectories simultaneously. In such a situation, a constraint on the variable D was introduced. Variable D presents a measure of how far the robotic arm can be stretched. This is useful to prevent arm singularity that occurs when the arm is fully stretched. By changing the values of these variables, the mobile platform pose relative to the end-effector pose, can be determined and vice versa. A singularity-robust and task-priority inverse with gradient projection method was used to find solutions for the dual-trajectory tracking while maximizing the whole system manipulability measure. This control algorithm was implemented initially on a simple 5 DoF planar mobile manipulator. MATLAB simulated planar mobile manipulation was used to test and optimize the proposed control system. Simulation results demonstrated the effectiveness of the control system in tracking the two trajectories and optimizing the arm's manipulability measure.

In the second control algorithm, a novel control algorithm was introduced for tracking the trajectory of redundant mobile manipulators when the translation of the mobile platform was restricted to follow its specified virtual or physical track. The control scheme was capable of controlling the mobile manipulator to follow the end-effector trajectory and the mobile platform track by adjusting the magnitudes and the directions of the mobile platform

translations along its predefined track. This allowed the mobile platform to put the arm in a position that assists the arm to successfully perform the task at hand. Initially, MATLAB simulated redundant planar mobile manipulator was used to implement and test the proposed control system. The simulation results demonstrated the effectiveness of the control system in adjusting the mobile platform translations along its track to allow the arm to follow its own trajectory. As a result, this control system allowed the mobile manipulator to follow both trajectories when other methods failed.

A control motion scheme was designed and implemented for power wheelchairs that have relatively high positioning error. In this work, the control scheme was implemented on the wheelchair mounted robotic arm system (WMRA). This is a step towards making the WMRA system capable of performing high precision tasks such as, “go and open a door” and “pick up an object”. The existing motion control is dependent solely on the wheel encoders, which made the system unreliable. The idea behind this motion control was to use a vision algorithm to correct the encoder estimation of the wheelchair pose. The vision algorithm uses Iterative Closest Point (ICP). Real time obstacle avoidance was used to test the proposed motion control scheme. The results showed the effectiveness of this control motion scheme.

Intensive simulated and real experiments were conducted to proof the effectiveness and the robustness of the control schemes. First, the two controllers were implemented and tested on a MATLAB simulated physical hardware (the WMRA system). Second, C++ programming language was used to implement the controllers on the real WMRA-II system.

The following is a list of major contributions made in this dissertation:

1. Development of general inverse-kinematics scheme to combine the mobility of nonholonomic mobile platforms and n DoF robotic arms.
2. Introduction of spherical control variables (D, α and β) to present the end-effector trajectory relative to the mobile platform trajectory and vice versa.

3. The use of the spherical variables to prioritize the mobile manipulator's dual-trajectory tracking.
4. The use of the whole system's manipulability measure for optimization.
5. Optimization of the magnitude and the direction of the mobile platform translation along a predefined track.
6. Design of a computer vision integrated motion control scheme for the detection and correction of the wheel encoder pose estimation.
7. The implementation of the control algorithms on a redundant planar mobile manipulator.
8. The implementation of the control algorithms on the MATLAB simulated WMRA system.
9. The implementation of the control algorithms on the actual WMRA system.

10.3 Recommendations

It is recommended to consider the dual-trajectory as a band of trajectories for both the end-effector and mobile platform instead of just one certain path. Then the dual-trajectory can be chosen according to a certain optimization criterion. This process can be done online or offline. This will allow for optimal dual-trajectory and path planning.

Tracking a mobile manipulator's dual-trajectory using spherical control variables needs more detailed investigation. It is recommended to perform a comprehensive study to know the effect of control variable D values on the stability of the system when it tracks the dual trajectory.

The mobile manipulator dynamics model should be integrated into the control algorithm to eliminate the effect of the mobile platform's mass, especially when there are changes in the platform velocities.

For the hardware implementation, it is recommended to integrate more precise sensors to the WMRA system for mapping and localizing the system in its environment. This will allow for online path planning.

REFERENCES

- [1] Wikimedia Commons, “This is a picture of the amazon fulfillment center in kentucky.” 2006. [Online]. Available: <http://www.flickr.com/photos/61023128@N08/5568927353/>
- [2] F. Huber, K. Kondak, K. Krieger, D. Sommer, M. Schwarzbach, M. Laiacker, I. Kossyk, S. Parusel, S. Haddadin, and A. Albu-Schäffer, “First analysis and experiments in aerial manipulation using fully actuated redundant robot arm,” in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Nov 2013, pp. 3452–3457.
- [3] E. Simetti, G. Casalino, S. Torelli, A. Sperind, and A. Turetta, “Floating underwater manipulation: Developed control methodology and experimental validation within the trident project,” *Journal of Field Robotics*, vol. 31, no. 3, pp. 364–385, 2014.
- [4] Wikimedia Commons, “A picture of the industrial mobile manipulator ”little helper” from the department of production at aalborg university in denmark.” 2009. [Online]. Available: https://commons.wikimedia.org/wiki/File/Mobilrobot_02.jpg
- [5] M. Hvilshøj, S. Bøgh, O. S. Nielsen, and O. Madsen, “Autonomous industrial mobile manipulation (AIMM): past, present and future,” *Industrial Robot: An International Journal*, vol. 39, no. 2, pp. 120–135, 2012.
- [6] R. Alqasemi, “Maximizing manipulation capabilities of persons with disabilities using a smart 9-degree-of-freedom wheelchair-mounted robotic arm system,” PhD Thesis, University of South Florida, Tampa, FL, USA, 2007.
- [7] A. Hentout, B. Bouzouia, I. Akli, and R. Toumi, *Mobile Manipulation: A Case Study, Robot Manipulators New Achievements*. InTech, 2010.
- [8] Y. Nakamura, *Advanced robotics : redundancy and optimization*. Reading, Mass: Addison-Wesley Pub. Co, 1991.
- [9] T. Yoshikawa, *Foundations of robotics : analysis and control*. Cambridge, Mass: MIT Press, 1990.
- [10] J. Perret, “Service robots for nuclear safety: new developments by cybernetix,” in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 3, 1998, pp. 2106–2109 vol.3.
- [11] T. Huntsberger, G. Rodriguez, and P. Schenker, “Robotics challenges for robotic and human mars exploration,” in *Robotics 2000*, pp. 340–346.

- [12] B. Hamner, S. Koterba, J. Shi, R. Simmons, and S. Singh, “Mobile robotic dynamic tracking for assembly tasks,” in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2009, pp. 2489–2495.
- [13] Z. Kai, G. Ebenhofer, C. Eitzinger, U. Zimmermann, C. Walter, J. Saenz, L. P. Castano, M. A. Fernandez Hernandez, and J. Navarro Oriol, “Mobile manipulator is coming to aerospace manufacturing industry,” in *Robotic and Sensors Environments (ROSE), 2014 IEEE International Symposium on*, 2014, pp. 94–99.
- [14] Southwest Research Institute, “Mobile manipulation: Robotics & automation engineering.” [Online]. Available: <http://www.swri.org/4org/d10/msd/automation/mobile-manipulator.htm>
- [15] Y. Yamamoto and X. Yun, “Coordinating locomotion and manipulation of a mobile manipulator,” in *Decision and Control, 1992., Proceedings of the 31st IEEE Conference on*, 1992, pp. 2643–2648 vol.3.
- [16] J. F. Gardner and S. A. Velinsky, “Kinematics of mobile manipulators and implications for design,” *Journal of Robotic Systems*, vol. 17, no. 6, pp. 309–320, 2000.
- [17] Y. Yamamoto and X. Yun, “Effect of the dynamic interaction on coordinated control of mobile manipulators,” *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 5, pp. 816–824, 1996.
- [18] S. Srinivasa, D. Berenson, M. Cakmak, A. Collet Romea, M. Dogar, A. Dragan, R. A. Knepper, T. D. Niemueller, K. Strabala, J. M. Vandeweghe, and J. Ziegler, “Herb 2.0: Lessons learned from developing a mobile manipulator for the home,” *Proceedings of the IEEE*, vol. 100, no. 8, pp. 1–19, July 2012.
- [19] J. Pérez, J. Sales, A. Peñalver, J. Fernández, P. Sanz, J. García, J. Martí, R. Marín, and D. Fornas, “Robotic manipulation within the underwater mission planning context,” in *Motion and Operation Planning of Robotic Systems*, ser. Mechanisms and Machine Science, G. Carbone and F. Gomez-Bravo, Eds. Springer International Publishing, 2015, vol. 29, pp. 495–522.
- [20] Fraunhofer Institute for Factory Operation and Automation (IFF), “Factories of the future: Mobile manipulators for aerospace production,” Press Release, April 2013. [Online]. Available: <http://www.iff.fraunhofer.de/en/press/press-releases/2013/factories-of-the-future-mobile-manipulators-for-aerospace-production.html>
- [21] W. C. Harris, “An integrated architecture for a networked robotics laboratory using an asynchronous distance learning network tool,” Ph.D. dissertation, New York, NY, USA, 2008.
- [22] S. Lin and A. Goldenberg, “Neural-network control of mobile manipulators,” *Neural Networks, IEEE Transactions on*, vol. 12, no. 5, pp. 1121–1133, Sep 2001.

- [23] S. Chitta, B. Cohen, and M. Likhachev, “Planning for autonomous door opening with a mobile manipulator,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 1799–1806.
- [24] J. Cameron, D. MacKenzie, K. Ward, R. Arkin, and W. Book, “Reactive control for mobile manipulation,” in *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, May 1993, pp. 228–235 vol.3.
- [25] P. Ogren, M. Egerstedt, and X. Hu, “Reactive mobile manipulation using dynamic trajectory tracking,” in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 4, 2000, pp. 3473–3478 vol.4.
- [26] O. Brock, O. Khatib, and S. Viji, “Task-consistent obstacle avoidance and motion behavior for mobile manipulation,” in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 1, 2002, pp. 388–393 vol.1.
- [27] L. Peterson, D. Austin, and D. Kragic, “High-level control of a mobile manipulator for door opening,” in *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, vol. 3, 2000, pp. 2333–2338 vol.3.
- [28] V. Andaluz, F. Roberti, J. M. Toibero, and R. Carelli, “Adaptive unified motion control of mobile manipulators,” *Control Engineering Practice*, vol. 20, no. 12, pp. 1337 – 1352, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0967066112001517>
- [29] T. Phan, T. Chung, M. Ngo, H. Kim, and S. Kim, “Decentralized control design for welding mobile manipulator,” *Journal of Mechanical Science and Technology*, vol. 19, no. 3, pp. 756–767, 2005.
- [30] M. Fruchard, P. Morin, and C. Samson, “A framework for the control of nonholonomic mobile manipulators,” *The International Journal of Robotics Research*, vol. 25, no. 8, pp. 745–780, 2006. [Online]. Available: <http://ijr.sagepub.com/content/25/8/745.abstract>
- [31] B. Hamner, S. Koterba, J. Shi, R. Simmons, and S. Singh, “An autonomous mobile manipulator for assembly tasks,” *Autonomous Robots*, vol. 28, no. 1, pp. 131–149, 2010. [Online]. Available: <http://dx.doi.org/10.1007/s10514-009-9142-y>
- [32] J. Peng, J. Yu, and J. Wang, “Robust adaptive tracking control for nonholonomic mobile manipulator with uncertainties,” *{ISA} Transactions*, vol. 53, no. 4, pp. 1035 – 1043, 2014, disturbance Estimation and Mitigation.
- [33] J. H. Chung, S. A. Velinsky, and R. A. Hess, “Interaction control of a redundant mobile manipulator,” *The International Journal of Robotics Research*, vol. 17, no. 12, pp. 1302–1309, 1998.
- [34] Y. Yamamoto and X. Yun, “Unified analysis on mobility and manipulability of mobile manipulators,” in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 2, 1999, pp. 1200–1206 vol.2.

- [35] F. G. Pin, K. A. Morgansen, F. A. Tulloch, C. J. Hacker, and K. B. Gower, “Motion planning for mobile manipulators with a non-holonomic constraint using the fsp (full space parameterization) method,” *Journal of Robotic Systems*, vol. 13, no. 11, pp. 723–736, 1996. [Online]. Available: [http://dx.doi.org/10.1002/\(SICI\)1097-4563\(199611\)13:11<723::AID-ROB4j3.0.CO;2-X](http://dx.doi.org/10.1002/(SICI)1097-4563(199611)13:11<723::AID-ROB4j3.0.CO;2-X)
- [36] M. Chen, “A genetic approach to motion planning of redundant mobile manipulator systems considering safety and configuration,” *Journal of Robotic Systems*, vol. 14, no. 7, pp. 529–544, 1997.
- [37] Y. Jia, N. Xi, Y. Cheng, and S. Liang, “Coordinated motion control of a nonholonomic mobile manipulator for accurate motion tracking,” in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, Sept 2014, pp. 1635–1640.
- [38] B. Bayle, J. Y. Fourquet, and M. Renaud, “Manipulability of wheeled mobile manipulators: Application to motion generation,” *The International Journal of Robotics Research*, vol. 22, no. 7-8, pp. 565–581, 2003.
- [39] A. De Luca, G. Oriolo, and P. R. Giordano, “Kinematic modeling and redundancy resolution for nonholonomic mobile manipulators,” in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, 2006, pp. 1867–1873.
- [40] H. Seraji, “A unified approach to motion control of mobile manipulators,” *The International Journal of Robotics Research*, vol. 17, no. 2, pp. 107–118, 1998.
- [41] R. M. Alqasemi, “Maximizing manipulation capabilities of persons with disabilities using a smart 9-degree-of-freedom wheelchair-mounted robotic arm system,” 2007.
- [42] G. D. White, R. M. Bhatt, T. Chin Pei, and V. N. Krovi, “Experimental evaluation of dynamic redundancy resolution in a nonholonomic wheeled mobile manipulator,” *Mechatronics, IEEE/ASME Transactions on*, vol. 14, no. 3, pp. 349–357, 2009.
- [43] F. Farelo, R. Alqasemi, and R. Dubey, “Optimized dual-trajectory tracking control of a 9-dof wmra system for adl tasks,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 1786–1791.
- [44] G. Oriolo and C. Mongillo, “Motion planning for mobile manipulators along given end-effector paths,” in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 2005, pp. 2154–2160.
- [45] K. Nagatani, T. Hirayama, A. Gofuku, and Y. Tanaka, “Motion planning for mobile manipulator with keeping manipulability,” in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 2, 2002, pp. 1663–1668 vol.2.
- [46] M. Egerstedt and H. Xiaoming, “Coordinated trajectory following for mobile manipulation,” in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 4, 2000, pp. 3479–3484 vol.4.

- [47] A. Mohri, S. Furuno, and M. Yamamoto, “Trajectory planning of mobile manipulator with end-effector’s specified path,” in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 4, 2001, pp. 2264–2269 vol.4.
- [48] Q. Huang, K. Tanie, and S. Sugano, “Coordinated motion planning for a mobile manipulator considering stability and manipulation,” *The International Journal of Robotics Research*, vol. 19, no. 8, pp. 732–742, 2000. [Online]. Available: <http://ijr.sagepub.com/content/19/8/732.abstract>
- [49] Y. Yamamoto and X. Yun, “Control of mobile manipulators following a moving surface,” in *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, May 1993, pp. 1–6 vol.3.
- [50] W. Dong, “On trajectory and force tracking control of constrained mobile manipulators with parameter uncertainty,” *Automatica*, vol. 38, no. 9, pp. 1475 – 1484, 2002. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0005109802000602>
- [51] Y. Yamamoto and X. Yun, “Coordinating locomotion and manipulation of a mobile manipulator,” in *Decision and Control, 1992., Proceedings of the 31st IEEE Conference on*, 1992, pp. 2643–2648 vol.3.
- [52] E. Papadopoulos and J. Poulakakis, “Planning and model-based control for mobile manipulators,” in *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, vol. 3, 2000, pp. 1810–1815 vol.3.
- [53] P. Baerlocher and R. Boulic, “Task-priority formulations for the kinematic control of highly redundant articulated structures,” in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, vol. 1, 1998, pp. 323–329 vol.1.
- [54] O. Kanoun, F. Lamiroux, and P. B. Wieber, “Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task,” *Robotics, IEEE Transactions on*, vol. 27, no. 4, pp. 785–792, 2011.
- [55] G. Ippoliti, A. Manna, and S. Longhi, “Robust robot localization by sensors with different degree of accuracy,” *Journal of Intelligent and Robotic Systems*, vol. 56, no. 3, pp. 259–276, 2009.
- [56] J. Borenstein and L. Feng, “Gyrodometry: a new method for combining data from gyros and odometry in mobile robots,” in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 1, 1996, pp. 423–428 vol.1.
- [57] P. Goel, S. I. Roumeliotis, and G. S. Sukhatme, “Robust localization using relative and absolute position estimates,” in *Intelligent Robots and Systems, 1999. IROS '99. Proceedings. 1999 IEEE/RSJ International Conference on*, vol. 2, 1999, pp. 1134–1140 vol.2.

- [58] W. L. D. Lui, T. J. J. Tang, T. Drummond, and L. Wai Ho, “Robust egomotion estimation using icp in inverse depth coordinates,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012, pp. 1671–1678.
- [59] J. Martinez-Carranza and A. Calway, “Efficient visual odometry using a structure-driven temporal map,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012, pp. 5210–5215.
- [60] A. Milella, G. Reina, and R. Siegwart, *Computer Vision Methods for Improved Mobile Robot State Estimation in Challenging Terrains*, ser. 2006, 2006, vol. 1.
- [61] P. J. Besl and H. D. McKay, “A method for registration of 3-d shapes,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, no. 2, pp. 239–256, 1992.
- [62] Z. Zhang, “Iterative point matching for registration of free-form curves,” Tech. Rep., 1992.
- [63] S. Rusinkiewicz and M. Levoy, “Efficient variants of the icp algorithm,” in *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, 2001, pp. 145–152.
- [64] G. Dissanayake, H. Shoudong, W. Zhan, and R. Ranasinghe, “A review of recent developments in simultaneous localization and mapping,” in *Industrial and Information Systems (ICIIS), 2011 6th IEEE International Conference on*, 2011, pp. 477–482.
- [65] T. Hervier, S. Bonnabel, and F. Goulette, “Accurate 3d maps from depth images and motion sensors via nonlinear kalman filtering,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 5291–5297.
- [66] R. E. Kalman and R. S. Bucy, “New results in linear filtering and prediction theory,” *Journal of Basic Engineering*, vol. 83, no. 1, pp. 95–108, 1961.
- [67] S. Y. Chen, “Kalman filter for robot vision: A survey,” *Industrial Electronics, IEEE Transactions on*, vol. 59, no. 11, pp. 4409–4420, 2012.
- [68] B. Bischoff, N.-T. Duy, F. Streichert, M. Ewert, and A. Knoll, “Fusing vision and odometry for accurate indoor robot localization,” in *Control Automation Robotics & Vision (ICARCV), 2012 12th International Conference on*, 2012, pp. 347–352.
- [69] M. Polanczyk, P. Baranski, M. Strzelecki, and K. Slot, “The application of kalman filter in visual odometry for eliminating direction drift,” in *Signals and Electronic Systems (ICSES), 2010 International Conference on*, 2010, pp. 131–134.
- [70] L. Tae-jae, B. Wook, J. Byung-moon, S. Ho-Jeong, and C. Dong-il Dan, “A new localization method for mobile robot by data fusion of vision sensor data and motion sensor data,” in *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*, 2012, pp. 723–728.




- [71] J. J. Craig, *Introduction to robotics : mechanics and control*. Addison-Wesley Publishing, 2003.
- [72] K. S. Chong and L. Kleeman, “Accurate odometry and error modelling for a mobile robot,” in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 4, Apr 1997, pp. 2783–2788 vol.4.
- [73] C. Tan Fung and R. V. Dubey, “A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators,” *Robotics and Automation, IEEE Transactions on*, vol. 11, no. 2, pp. 286–292, 1995.
- [74] R. Dubey and J. Luh, “Redundant robot control for higher flexibility,” in *1987 IEEE International Conference on Robotics and Automation.*, vol. 4, 1987, pp. 1066–1072.
- [75] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., 1994.
- [76] M. Mashali, R. Alqasemi, and R. Dubey, “Task priority based dual-trajectory control for redundant mobile manipulators,” in *Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on*, Dec 2014, pp. 1457–1462.
- [77] D. G. Zill and M. R. Cullen, *Advanced engineering mathematics*, 3rd ed., ser. Prindle, Weber & Schmidt series in mathematics. Sudbury, MA: Jones and Bartlett, 2006.
- [78] A. Liegeois, “Automatic supervisory control of the configuration and behavior of multi-body mechanisms,” *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 7, no. 12, pp. 868–871, 1977.
- [79] J. S. Gutmann and D. Fox, “An experimental comparison of localization methods continued,” in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 1, 2002, pp. 454–459 vol.1.
- [80] J. S. Gutmann, W. Burgard, D. Fox, and K. Konolige, “An experimental comparison of localization methods,” in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, vol. 2, 1998, pp. 736–743 vol.2.
- [81] M. Mashali, R. Alqasemi, S. Sarkar, and R. Dubey, “Design, implementation and evaluation of a motion control scheme for mobile platforms with high uncertainties,” in *Biomedical Robotics and Biomechatronics (2014 5th IEEE RAS EMBS International Conference on*, Aug 2014, pp. 1091–1097.
- [82] O. Horn and M. Kreutner, “Smart wheelchair perception using odometry, ultrasound sensors, and camera,” *Robotica*, vol. 27, no. 02, pp. 303–310, 2009.
- [83] C. De la Cruz, T. F. Bastos, F. A. A. Cheein, and R. Carelli, “Slam-based robotic wheelchair navigation system designed for confined spaces,” in *Industrial Electronics (ISIE), 2010 IEEE International Symposium on*, 2010, pp. 2331–2336.


- [84] R. Simpson, E. LoPresti, S. Hayashi, I. Nourbakhsh, and D. Miller, “The smart wheelchair component system,” *Journal of Rehabilitation Research and Development*, vol. 41, no. 3B, pp. 429–442, 2004.
- [85] Y. Touati, H. Aoudia, and A. Ali-Cherif, “Intelligent wheelchair localization in wireless sensor network environment: A fuzzy logic approach,” in *Intelligent Systems (IS), 2010 5th IEEE International Conference*, 2010, pp. 408–413.
- [86] US DIGITAL H5 Ball Bearing Optical Shaft Encoder. [Online]. Available: <http://www.usdigital.com/products/encoders>
- [87] Microsoft Kinect for X-BOX 360, 2011. [Online]. Available: www.xbox.com/en-us/kinect
- [88] S. Du, N. Zheng, S. Ying, Q. You, and Y. Wu, “An extension of the icp algorithm considering scale factor,” in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 5, Sept 2007, pp. V – 193–V – 196.
- [89] R. Alqasemi and R. Dubey, “Combined mobility and manipulation control of a newly developed 9-dof wheelchair-mounted robotic arm system,” in *Robotics and Automation, 2007 IEEE International Conference on*, 2007, pp. 4524–4529.
- [90] K. Edwards, R. Alqasemi, and R. Dubey, “Wheelchair-mounted robotic arms: Design and development,” in *Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on*, 2006, pp. 613–618.
- [91] R. B. Rusu and S. Cousins, “3d is here: Point cloud library (pcl),” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 1–4.
- [92] J. Borenstein and L. Feng, “Umbmark: a benchmark test for measuring odometry errors in mobile robots,” vol. 2591, 1995, pp. 113–124.
- [93] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *Systems Science and Cybernetics, IEEE Transactions on*, vol. 4, no. 2, pp. 100–107, 1968.
- [94] P. Lester, “A* pathfinding for beginners,” *online*. *GameDev WebSite*. <http://www.gamedev.net/reference/articles/article2003.asp> (Acesso em 08/02/2009), 2005.
- [95] P. Schrock, F. Farelo, R. Alqasemi, and R. Dubey, “Design, simulation and testing of a new modular wheelchair mounted robotic arm to perform activities of daily living,” in *Rehabilitation Robotics, 2009. ICORR 2009. IEEE International Conference on*, 2009, pp. 518–523.

APPENDICES

Appendix A Copyright Permission

Below is permission of the use of Figure 2.1.

HomeAccount InfoHelp



Title: First analysis and experiments in aerial manipulation using fully actuated redundant robot arm

Conference Proceedings: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on

Author: Huber, F.; Kondak, K.; Krieger, K.; Sommer, D.; Schwarzbach, M.; Laiacker, M.; Kosyik, I.; Parusel, S.; Haddadin, S.; Albu-Schaffer, A.

Publisher: IEEE

Date: 3-7 Nov. 2013

Copyright © 2013, IEEE

Logged in as:
Mustafa Mashali
Account #:
3000925634

LOGOUT

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACKCLOSE WINDOW

Copyright © 2015 [Copyright Clearance Center, Inc.](#) All Rights Reserved. [Privacy statement](#). [Terms and Conditions](#).

Appendix A (Continued)

Below is permission of the use of Figure 2.2.

JOHN WILEY AND SONS LICENSE TERMS AND CONDITIONS	
	Nov 11, 2015
<hr/> <hr/>	
This Agreement between Mustafa Mashali ("You") and John Wiley and Sons ("John Wiley and Sons") consists of your license details and the terms and conditions provided by John Wiley and Sons and Copyright Clearance Center.	
License Number	3746060572382
License date	Nov 11, 2015
Licensed Content Publisher	John Wiley and Sons
Licensed Content Publication	Journal of Field Robotics
Licensed Content Title	Floating Underwater Manipulation: Developed Control Methodology and Experimental Validation within the TRIDENT Project
Licensed Content Author	Enrico Simetti, Giuseppe Casalino, Sandro Torelli, Alessandro Sperindé, Alessio Turetta
Licensed Content Date	Jan 2, 2014
Pages	22
Type of use	Dissertation/Thesis
Requestor type	University/Academic
Format	Print and electronic
Portion	Figure/table
Number of figures/tables	1
Original Wiley figure/table number(s)	Figure 1
Will you be translating?	No
Title of your thesis / dissertation	Kinematic control of redundant mobile manipulator
Expected completion date	Dec 2015
Expected size (number of pages)	200
Requestor Location	Mustafa Mashali 14221 Les Palms Circle APT # 102 TAMPA, FL 33613 United States Attn: Mustafa Mashali
Billing Type	Invoice
Billing Address	Mustafa Mashali 14221 Les Palms Circle APT # 102 TAMPA, FL 33613 United States

Appendix A (Continued)

Attn: Mustafa Mashali

Total

0.00 USD

[Terms and Conditions](#)

TERMS AND CONDITIONS

This copyrighted material is owned by or exclusively licensed to John Wiley & Sons, Inc. or one of its group companies (each a "Wiley Company") or handled on behalf of a society with which a Wiley Company has exclusive publishing rights in relation to a particular work (collectively "WILEY"). By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the billing and payment terms and conditions established by the Copyright Clearance Center Inc., ("CCC's Billing and Payment terms and conditions"), at the time that you opened your RightsLink account (these are available at any time at <http://myaccount.copyright.com>).

Terms and Conditions

- The materials you have requested permission to reproduce or reuse (the "Wiley Materials") are protected by copyright.
- You are hereby granted a personal, non-exclusive, non-sub licensable (on a stand-alone basis), non-transferable, worldwide, limited license to reproduce the Wiley Materials for the purpose specified in the licensing process. This license, **and any CONTENT (PDF or image file) purchased as part of your order**, is for a one-time use only and limited to any maximum distribution number specified in the license. The first instance of republication or reuse granted by this license must be completed within two years of the date of the grant of this license (although copies prepared before the end date may be distributed thereafter). The Wiley Materials shall not be used in any other manner or for any other purpose, beyond what is granted in the license. Permission is granted subject to an appropriate acknowledgement given to the author, title of the material/book/journal and the publisher. You shall also duplicate the copyright notice that appears in the Wiley publication in your use of the Wiley Material. Permission is also granted on the understanding that nowhere in the text is a previously published source acknowledged for all or part of this Wiley Material. Any third party content is expressly excluded from this permission.
- With respect to the Wiley Materials, all rights are reserved. Except as expressly granted by the terms of the license, no part of the Wiley Materials may be copied, modified, adapted (except for minor reformatting required by the new Publication), translated, reproduced, transferred or distributed, in any form or by any means, and no derivative works may be made based on the Wiley Materials without the prior permission of the respective copyright owner. **For STM Signatory Publishers clearing permission under the terms of the [STM Permissions Guidelines](#) only, the terms of the license are extended to include subsequent editions and for editions in other languages, provided such editions are for the work as a whole in situ and does not involve the separate exploitation of the permitted figures or extracts**, You may not alter, remove or suppress in any manner any copyright, trademark or other notices displayed by the Wiley Materials. You may not license, rent, sell, loan, lease, pledge, offer as security, transfer or assign the Wiley Materials on a stand-alone basis, or any of the rights granted to you hereunder to any other person.

Appendix A (Continued)

- The Wiley Materials and all of the intellectual property rights therein shall at all times remain the exclusive property of John Wiley & Sons Inc, the Wiley Companies, or their respective licensors, and your interest therein is only that of having possession of and the right to reproduce the Wiley Materials pursuant to Section 2 herein during the continuance of this Agreement. You agree that you own no right, title or interest in or to the Wiley Materials or any of the intellectual property rights therein. You shall have no rights hereunder other than the license as provided for above in Section 2. No right, license or interest to any trademark, trade name, service mark or other branding ("Marks") of WILEY or its licensors is granted hereunder, and you agree that you shall not assert any such right, license or interest with respect thereto
- NEITHER WILEY NOR ITS LICENSORS MAKES ANY WARRANTY OR REPRESENTATION OF ANY KIND TO YOU OR ANY THIRD PARTY, EXPRESS, IMPLIED OR STATUTORY, WITH RESPECT TO THE MATERIALS OR THE ACCURACY OF ANY INFORMATION CONTAINED IN THE MATERIALS, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF MERCHANTABILITY, ACCURACY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, USABILITY, INTEGRATION OR NON-INFRINGEMENT AND ALL SUCH WARRANTIES ARE HEREBY EXCLUDED BY WILEY AND ITS LICENSORS AND WAIVED BY YOU.
- WILEY shall have the right to terminate this Agreement immediately upon breach of this Agreement by you.
- You shall indemnify, defend and hold harmless WILEY, its Licensors and their respective directors, officers, agents and employees, from and against any actual or threatened claims, demands, causes of action or proceedings arising from any breach of this Agreement by you.
- IN NO EVENT SHALL WILEY OR ITS LICENSORS BE LIABLE TO YOU OR ANY OTHER PARTY OR ANY OTHER PERSON OR ENTITY FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, INDIRECT, EXEMPLARY OR PUNITIVE DAMAGES, HOWEVER CAUSED, ARISING OUT OF OR IN CONNECTION WITH THE DOWNLOADING, PROVISIONING, VIEWING OR USE OF THE MATERIALS REGARDLESS OF THE FORM OF ACTION, WHETHER FOR BREACH OF CONTRACT, BREACH OF WARRANTY, TORT, NEGLIGENCE, INFRINGEMENT OR OTHERWISE (INCLUDING, WITHOUT LIMITATION, DAMAGES BASED ON LOSS OF PROFITS, DATA, FILES, USE, BUSINESS OPPORTUNITY OR CLAIMS OF THIRD PARTIES), AND WHETHER OR NOT THE PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION SHALL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN.
- Should any provision of this Agreement be held by a court of competent jurisdiction to be illegal, invalid, or unenforceable, that provision shall be deemed amended to achieve as nearly as possible the same economic effect as the original provision, and the legality, validity and enforceability of the remaining provisions of this Agreement shall not be affected or impaired thereby.

Appendix A (Continued)

- The failure of either party to enforce any term or condition of this Agreement shall not constitute a waiver of either party's right to enforce each and every term and condition of this Agreement. No breach under this agreement shall be deemed waived or excused by either party unless such waiver or consent is in writing signed by the party granting such waiver or consent. The waiver by or consent of a party to a breach of any provision of this Agreement shall not operate or be construed as a waiver of or consent to any other or subsequent breach by such other party.
- This Agreement may not be assigned (including by operation of law or otherwise) by you without WILEY's prior written consent.
- Any fee required for this permission shall be non-refundable after thirty (30) days from receipt by the CCC.
- These terms and conditions together with CCC's Billing and Payment terms and conditions (which are incorporated herein) form the entire agreement between you and WILEY concerning this licensing transaction and (in the absence of fraud) supersedes all prior agreements and representations of the parties, oral or written. This Agreement may not be amended except in writing signed by both parties. This Agreement shall be binding upon and inure to the benefit of the parties' successors, legal representatives, and authorized assigns.
- In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall prevail.
- WILEY expressly reserves all rights not specifically granted in the combination of (i) the license details provided by you and accepted in the course of this licensing transaction, (ii) these terms and conditions and (iii) CCC's Billing and Payment terms and conditions.
- This Agreement will be void if the Type of Use, Format, Circulation, or Requestor Type was misrepresented during the licensing process.
- This Agreement shall be governed by and construed in accordance with the laws of the State of New York, USA, without regards to such state's conflict of law rules. Any legal action, suit or proceeding arising out of or relating to these Terms and Conditions or the breach thereof shall be instituted in a court of competent jurisdiction in New York County in the State of New York in the United States of America and each party hereby consents and submits to the personal jurisdiction of such court, waives any objection to venue in such court and consents to service of process by registered or certified mail, return receipt requested, at the last known address of such party.

WILEY OPEN ACCESS TERMS AND CONDITIONS

Wiley Publishes Open Access Articles in fully Open Access Journals and in Subscription journals offering Online Open. Although most of the fully Open Access journals publish open access articles under the terms of the Creative Commons Attribution (CC BY) License only, the subscription journals and a few of the Open Access Journals offer a choice of

Appendix A (Continued)

Creative Commons Licenses. The license type is clearly identified on the article.

The Creative Commons Attribution License

The [Creative Commons Attribution License \(CC-BY\)](#) allows users to copy, distribute and transmit an article, adapt the article and make commercial use of the article. The CC-BY license permits commercial and non-

Creative Commons Attribution Non-Commercial License

The [Creative Commons Attribution Non-Commercial \(CC-BY-NC\)License](#) permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.(see below)

Creative Commons Attribution-Non-Commercial-NoDerivs License

The [Creative Commons Attribution Non-Commercial-NoDerivs License](#) (CC-BY-NC-ND) permits use, distribution and reproduction in any medium, provided the original work is properly cited, is not used for commercial purposes and no modifications or adaptations are made. (see below)

Use by commercial "for-profit" organizations

Use of Wiley Open Access articles for commercial, promotional, or marketing purposes requires further explicit permission from Wiley and will be subject to a fee.

Further details can be found on Wiley Online Library

<http://olabout.wiley.com/WileyCDA/Section/id-410895.html>

Other Terms and Conditions:

v1.10 Last updated September 2015

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

Appendix A (Continued)

Below is permission of the use of Figures 2.4.

EMERALD GROUP PUBLISHING LIMITED LICENSE TERMS AND CONDITIONS	
Sep 23, 2015	
<hr/> <hr/>	
<p>This Agreement between Mustafa Mashali ("You") and Emerald Group Publishing Limited ("Emerald Group Publishing Limited") consists of your license details and the terms and conditions provided by Emerald Group Publishing Limited and Copyright Clearance Center.</p>	
<p>The publisher has provided special terms related to this request that can be found at the end of the Publisher's Terms and Conditions.</p>	
License Number	3714900206987
License date	Sep 22, 2015
Licensed Content Publisher	Emerald Group Publishing Limited
Licensed Content Publication	Industrial Robot The international journal of industrial and service robotics
Licensed Content Title	Autonomous industrial mobile manipulation (AIMM): past, present and future
Licensed Content Author	Mads Hvilshøj, Simon Bøgh, Oluf Skov Nielsen, et al
Licensed Content Date	03/02/2012
Licensed Content Volume Number	39
Licensed Content Issue Number	2
Type of Use	Dissertation/Thesis
Requestor type	Academic
Author of requested content	No
Portion	Figures/table/illustration
Number of figures/tables	1
Will you be translating?	No
Format	Electronic
Geographic Rights	World rights
Order Reference Number	None
Emerald VAT number	GB 665 3593 06
Requestor Location	Mustafa Mashali 14221 Les Palms Circle APT # 102 TAMPA, FL 33613 United States Attn: Mustafa Mashali
Billing Type	Invoice
Billing Address	Mustafa Mashali

Appendix A (Continued)

14221 Les Palms Circle APT # 102

TAMPA, FL 33613
United States
Attn: Mustafa Mashali

Total 0.00 USD

[Terms and Conditions](#)

TERMS AND CONDITIONS

1. The publisher for this copyrighted material is Emerald Group Publishing Limited. By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the Billing and Payment terms and conditions established by Copyright Clearance Center, Inc. ("CCC"), at the time that you opened your Rightslink account and that are available at any time at).
2. Limited License. Publisher hereby grants to you a non-exclusive license to use this material. Licenses are for one-time use only with a maximum distribution equal to the number that you identified in the licensing process; any form of republication must be completed within 12 months from the date hereof (although copies prepared before then may be distributed thereafter).
3. Geographic Rights: Scope. Licenses may be exercised only in the geographic regions you identified in the licensing process.
4. Altering/Modifying Material: Not Permitted. You may not alter or modify the licensed material in any manner. For permission to translate the material into another language please contact permissions@emeraldinsight.com.
5. Reservation of Rights. Publisher reserves all rights not specifically granted in the combination of (i) the license details provided by you and accepted in the course of this licensing transaction, (ii) these terms and conditions and (iii) CCC's Billing and Payment terms and conditions.
6. License Contingent on Payment. While you may exercise the rights licensed immediately upon issuance of the license at the end of the licensing process for the transaction, provided that you have disclosed complete and accurate details of your proposed use, no license is finally effective unless and until full payment is received from you (either by publisher or by CCC) as provided in CCC's Billing and Payment terms and conditions. If full payment is not received on a timely basis, then any license preliminarily granted shall be deemed automatically revoked and shall be void as if never granted. Further, in the event that you breach any of these terms and conditions or any of CCC's Billing and Payment terms and conditions, the license is automatically revoked and shall be void as if never granted. Use of materials as described in a revoked license, as well as any use of the materials beyond the scope of an unrevoked license, may constitute copyright infringement and publisher reserves the right to take any and all action to protect its copyright in the materials.
7. Emerald always informs its authors of requests to republish their article. In the unlikely event that any author objects to the granting of the license to republish, Emerald reserves the right to revoke said license. The licensee will be informed and the license fee reimbursed within 10 working days.
8. Copyright notice: Disclaimer: You must include the following copyright and permission notice in connection with any reproduction of the licensed material and shall ensure that every published article gives due prominence on the title page to the original author/s, the journal title, volume, issue, page numbers and the copyright designation "© Emerald Group Publishing Limited all rights reserved."
9. Warranties: None. Publisher makes no representations or warranties with respect to the

Appendix A (Continued)

licensed material and adopts on its own behalf the limitations and disclaimers established by CCC on its behalf in its Billing and Payment terms and conditions for this licensing transaction.

10. Indemnity. You hereby indemnify and agree to hold harmless publisher and CCC, and their respective officers, directors, employees and agents, from and against any and all claims arising out of your use of the licensed material other than as specifically authorized pursuant to this license.

11. No Transfer of license. This license is personal to you and may not be sublicensed, assigned, or transferred by you to any other person without publisher's written permission.

12. No Amendment Except in Writing. This license may not be amended except in a writing signed by both parties (or, in the case of publisher, by CCC on publisher's behalf).

13. Objection to Contrary Terms: Publisher hereby objects to any terms contained in any purchase order, acknowledgment, check endorsement or other writing prepared by you, which terms are inconsistent with these terms and conditions or CCC's Billing and Payment terms and conditions. These terms and conditions, together with CCC's Billing and Payment terms and conditions (which are incorporated herein), comprise the entire agreement between you and publisher (and CCC) concerning this licensing transaction. In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall control.

14. Jurisdiction: This license transaction shall be governed by and construed in accordance with the laws of the United Kingdom. You hereby agree to submit to the jurisdiction of the courts located in the United Kingdom for purposes of resolving any disputes that may arise in connection with this licensing transaction.






15. Special Terms:


v 1.3

Questions? customer@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

Appendix A (Continued)

Below is permission of the use of Figure 2.5.





Title: Interaction Control of a Redundant Mobile Manipulator:
Author: Jae H. Chung, Steven A. Velinsky, Ronald A. Hess
Publication: International Journal of Robotics Research
Publisher: SAGE Publications
Date: 12/01/1998
Copyright © 1998, © SAGE Publications

Logged in as:
Mustafa Mashali
[LOGOUT](#)

Gratis Reuse


Permission is granted at no cost for use of content in a Master's Thesis and/or Doctoral Dissertation. If you intend to distribute or sell your Master's Thesis/Doctoral Dissertation to the general public through print or website publication, please return to the previous page and select 'Republish in a Book/Journal' or 'Post on intranet/password-protected website' to complete your request.

[BACK](#) [CLOSE WINDOW](#)

Copyright © 2015 [Copyright Clearance Center, Inc.](#) All Rights Reserved. [Privacy statement](#). [Terms and Conditions](#).
Comments? We would like to hear from you. E-mail us at customer@copyright.com

Appendix A (Continued)

Below is permission of the use of Figure 2.6.

HomeCreate AccountHelp



Requesting permission to reuse content from an IEEE publication

Title: Unified analysis on mobility and manipulability of mobile manipulators

Conference Proceedings: Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on

Author: Yamamoto, Y.; Yun, X.

Publisher: IEEE

Date: 1999

Copyright © 1999, IEEE

[LOGIN](#)

If you're a [copyright.com user](#), you can login to RightsLink using your [copyright.com](#) credentials. Already a [RightsLink user](#) or want to [learn more?](#)

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.htm to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACKCLOSE WINDOW

Copyright © 2015 [Copyright Clearance Center, Inc.](#) All Rights Reserved. [Privacy statement.](#) [Terms and Conditions.](#)
Comments? We would like to hear from you. E-mail us at customercare@copyright.com

Appendix A (Continued)

Below is permission of the use of Figure 2.7.

ELSEVIER LICENSE TERMS AND CONDITIONS		Jun 05, 2015
<hr/> <hr/>		
<p>This is a License Agreement between Mustafa Mashali ("You") and Elsevier ("Elsevier") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by Elsevier, and the payment terms and conditions.</p>		
<p>All payments must be made in full to CCC. For payment instructions, please see information listed at the bottom of this form.</p>		
Supplier	Elsevier Limited The Boulevard, Langford Lane Kidlington, Oxford, OX5 1GB, UK	
Registered Company Number	1982084	
Customer name	Mustafa Mashali	
Customer address	14221 Les Palms Circle APT # 102 TAMPA, FL 33613	
License number	3642580986106	
License date	Jun 05, 2015	
Licensed content publisher	Elsevier	
Licensed content publication	Control Engineering Practice	
Licensed content title	Adaptive unified motion control of mobile manipulators	
Licensed content author	V́ctor Andaluz, Flavio Roberti, Juan Marcos Toibero, Ricardo Carelli	
Licensed content date	December 2012	
Licensed content volume number	20	
Licensed content issue number	12	
Number of pages	16	
Start Page	1337	
End Page	1352	
Type of Use	reuse in a thesis/dissertation	
Intended publisher of new work	other	
Portion	figures/tables/illustrations	
Number of figures/tables/illustrations	1	
Format	print	
Are you the author of this Elsevier article?	No	

Appendix A (Continued)

Will you be translating?	No
Original figure numbers	Figure 4
Title of your thesis/dissertation	Kinematic Control of Redundant Mobile Manipulator
Expected completion date	Jul 2015
Estimated size (number of pages)	200
Elsevier VAT number	GB 494 6272 12
Permissions price	0.00 USD
VAT/Local Sales Tax	0.00 USD / 0.00 GBP
Total	0.00 USD
Terms and Conditions	

INTRODUCTION

1. The publisher for this copyrighted material is Elsevier. By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the Billing and Payment terms and conditions established by Copyright Clearance Center, Inc. ("CCC"), at the time that you opened your Rightslink account and that are available at any time at <http://myaccount.copyright.com>).

GENERAL TERMS

2. Elsevier hereby grants you permission to reproduce the aforementioned material subject to the terms and conditions indicated.

3. Acknowledgement: If any part of the material to be used (for example, figures) has appeared in our publication with credit or acknowledgement to another source, permission must also be sought from that source. If such permission is not obtained then that material may not be included in your publication/copies. Suitable acknowledgement to the source must be made, either as a footnote or in a reference list at the end of your publication, as follows:

"Reprinted from Publication title, Vol /edition number, Author(s), Title of article / title of chapter, Pages No., Copyright (Year), with permission from Elsevier [OR APPLICABLE SOCIETY COPYRIGHT OWNER]." Also Lancet special credit - "Reprinted from The Lancet, Vol. number, Author(s), Title of article, Pages No., Copyright (Year), with permission from Elsevier."

4. Reproduction of this material is confined to the purpose and/or media for which permission is hereby given.

5. Altering/Modifying Material: Not Permitted. However figures and illustrations may be altered/adapted minimally to serve your work. Any other abbreviations, additions, deletions and/or any other alterations shall be made only with prior written authorization of Elsevier Ltd. (Please contact Elsevier at permissions@elsevier.com)

6. If the permission fee for the requested use of our material is waived in this instance,

Appendix A (Continued)

please be advised that your future requests for Elsevier materials may attract a fee.

7. Reservation of Rights: Publisher reserves all rights not specifically granted in the combination of (i) the license details provided by you and accepted in the course of this licensing transaction, (ii) these terms and conditions and (iii) CCC's Billing and Payment terms and conditions.

8. License Contingent Upon Payment: While you may exercise the rights licensed immediately upon issuance of the license at the end of the licensing process for the transaction, provided that you have disclosed complete and accurate details of your proposed use, no license is finally effective unless and until full payment is received from you (either by publisher or by CCC) as provided in CCC's Billing and Payment terms and conditions. If full payment is not received on a timely basis, then any license preliminarily granted shall be deemed automatically revoked and shall be void as if never granted. Further, in the event that you breach any of these terms and conditions or any of CCC's Billing and Payment terms and conditions, the license is automatically revoked and shall be void as if never granted. Use of materials as described in a revoked license, as well as any use of the materials beyond the scope of an unrevoked license, may constitute copyright infringement and publisher reserves the right to take any and all action to protect its copyright in the materials.

9. Warranties: Publisher makes no representations or warranties with respect to the licensed material.

10. Indemnity: You hereby indemnify and agree to hold harmless publisher and CCC, and their respective officers, directors, employees and agents, from and against any and all claims arising out of your use of the licensed material other than as specifically authorized pursuant to this license.

11. No Transfer of License: This license is personal to you and may not be sublicensed, assigned, or transferred by you to any other person without publisher's written permission.

12. No Amendment Except in Writing: This license may not be amended except in a writing signed by both parties (or, in the case of publisher, by CCC on publisher's behalf).

13. Objection to Contrary Terms: Publisher hereby objects to any terms contained in any purchase order, acknowledgment, check endorsement or other writing prepared by you, which terms are inconsistent with these terms and conditions or CCC's Billing and Payment terms and conditions. These terms and conditions, together with CCC's Billing and Payment terms and conditions (which are incorporated herein), comprise the entire agreement between you and publisher (and CCC) concerning this licensing transaction. In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall control.

14. Revocation: Elsevier or Copyright Clearance Center may deny the permissions described in this License at their sole discretion, for any reason or no reason, with a full refund payable to you. Notice of such denial will be made using the contact information provided by you. Failure to receive such notice will not alter or invalidate the denial. In no event will Elsevier or Copyright Clearance Center be responsible or liable for any costs, expenses or damage incurred by you as a result of a denial of your permission request, other than a refund of the

Appendix A (Continued)

amount(s) paid by you to Elsevier and/or Copyright Clearance Center for denied permissions.

LIMITED LICENSE

The following terms and conditions apply only to specific license types:

15. **Translation:** This permission is granted for non-exclusive world **English** rights only unless your license was granted for translation rights. If you licensed translation rights you may only translate this content into the languages you requested. A professional translator must perform all translations and reproduce the content word for word preserving the integrity of the article. If this license is to re-use 1 or 2 figures then permission is granted for non-exclusive world rights in all languages.

16. **Posting licensed content on any Website:** The following terms and conditions apply as follows: Licensing material from an Elsevier journal: All content posted to the web site must maintain the copyright information line on the bottom of each image; A hyper-text must be included to the Homepage of the journal from which you are licensing at <http://www.sciencedirect.com/science/journal/xxxx> or the Elsevier homepage for books at <http://www.elsevier.com>; Central Storage: This license does not include permission for a scanned version of the material to be stored in a central repository such as that provided by Heron/XanEdu.

Licensing material from an Elsevier book: A hyper-text link must be included to the Elsevier homepage at <http://www.elsevier.com>. All content posted to the web site must maintain the copyright information line on the bottom of each image.

Posting licensed content on Electronic reserve: In addition to the above the following clauses are applicable: The web site must be password-protected and made available only to bona fide students registered on a relevant course. This permission is granted for 1 year only. You may obtain a new license for future website posting.

17. **For journal authors:** the following clauses are applicable in addition to the above:

Preprints:

A preprint is an author's own write-up of research results and analysis, it has not been peer-reviewed, nor has it had any other value added to it by a publisher (such as formatting, copyright, technical enhancement etc.).

Authors can share their preprints anywhere at any time. Preprints should not be added to or enhanced in any way in order to appear more like, or to substitute for, the final versions of articles however authors can update their preprints on arXiv or RePEc with their Accepted Author Manuscript (see below).

If accepted for publication, we encourage authors to link from the preprint to their formal publication via its DOI. Millions of researchers have access to the formal publications on ScienceDirect, and so links will help users to find, access, cite and use the best available version. Please note that Cell Press, The Lancet and some society-owned have different preprint policies. Information on these policies is available on the journal homepage.

Appendix A (Continued)

Accepted Author Manuscripts: An accepted author manuscript is the manuscript of an article that has been accepted for publication and which typically includes author-incorporated changes suggested during submission, peer review and editor-author communications.

Authors can share their accepted author manuscript:

- immediately
 - via their non-commercial person homepage or blog
 - by updating a preprint in arXiv or RePEc with the accepted manuscript
 - via their research institute or institutional repository for internal institutional uses or as part of an invitation-only research collaboration work-group
 - directly by providing copies to their students or to research collaborators for their personal use
 - for private scholarly sharing as part of an invitation-only work group on commercial sites with which Elsevier has an agreement
- after the embargo period
 - via non-commercial hosting platforms such as their institutional repository
 - via commercial sites with which Elsevier has an agreement

In all cases accepted manuscripts should:

- link to the formal publication via its DOI
- bear a CC-BY-NC-ND license - this is easy to do
- if aggregated with other manuscripts, for example in a repository or other site, be shared in alignment with our hosting policy not be added to or enhanced in any way to appear more like, or to substitute for, the published journal article.

Published journal article (JPA): A published journal article (PJA) is the definitive final record of published research that appears or will appear in the journal and embodies all value-adding publishing activities including peer review co-ordination, copy-editing, formatting, (if relevant) pagination and online enrichment.

Policies for sharing publishing journal articles differ for subscription and gold open access articles:

Subscription Articles: If you are an author, please share a link to your article rather than the full-text. Millions of researchers have access to the formal publications on ScienceDirect, and so links will help your users to find, access, cite, and use the best available version.

Theses and dissertations which contain embedded PJAs as part of the formal submission can be posted publicly by the awarding institution with DOI links back to the formal

Appendix A (Continued)

publications on ScienceDirect.

If you are affiliated with a library that subscribes to ScienceDirect you have additional private sharing rights for others' research accessed under that agreement. This includes use for classroom teaching and internal training at the institution (including use in course packs and courseware programs), and inclusion of the article for grant funding purposes.

Gold Open Access Articles: May be shared according to the author-selected end-user license and should contain a [CrossMark logo](#), the end user license, and a DOI link to the formal publication on ScienceDirect.

Please refer to Elsevier's [posting policy](#) for further information.

18. **For book authors** the following clauses are applicable in addition to the above: Authors are permitted to place a brief summary of their work online only. You are not allowed to download and post the published electronic version of your chapter, nor may you scan the printed edition to create an electronic version. **Posting to a repository:** Authors are permitted to post a summary of their chapter only in their institution's repository.

19. **Thesis/Dissertation:** If your license is for use in a thesis/dissertation your thesis may be submitted to your institution in either print or electronic form. Should your thesis be published commercially, please reapply for permission. These requirements include permission for the Library and Archives of Canada to supply single copies, on demand, of the complete thesis and include permission for Proquest/UMI to supply single copies, on demand, of the complete thesis. Should your thesis be published commercially, please reapply for permission. Theses and dissertations which contain embedded PJAs as part of the formal submission can be posted publicly by the awarding institution with DOI links back to the formal publications on ScienceDirect.

Elsevier Open Access Terms and Conditions

You can publish open access with Elsevier in hundreds of open access journals or in nearly 2000 established subscription journals that support open access publishing. Permitted third party re-use of these open access articles is defined by the author's choice of Creative Commons user license. See our [open access license policy](#) for more information.

Terms & Conditions applicable to all Open Access articles published with Elsevier:

Any reuse of the article must not represent the author as endorsing the adaptation of the article nor should the article be modified in such a way as to damage the author's honour or reputation. If any changes have been made, such changes must be clearly indicated.

The author(s) must be appropriately credited and we ask that you include the end user license and a DOI link to the formal publication on ScienceDirect.

If any part of the material to be used (for example, figures) has appeared in our publication with credit or acknowledgement to another source it is the responsibility of the user to ensure their reuse complies with the terms and conditions determined by the rights holder.

Additional Terms & Conditions applicable to each Creative Commons user license:

Appendix A (Continued)

CC BY: The CC-BY license allows users to copy, to create extracts, abstracts and new works from the Article, to alter and revise the Article and to make commercial use of the Article (including reuse and/or resale of the Article by commercial entities), provided the user gives appropriate credit (with a link to the formal publication through the relevant DOI), provides a link to the license, indicates if changes were made and the licensor is not represented as endorsing the use made of the work. The full details of the license are available at <http://creativecommons.org/licenses/by/4.0>.

CC BY NC SA: The CC BY-NC-SA license allows users to copy, to create extracts, abstracts and new works from the Article, to alter and revise the Article, provided this is not done for commercial purposes, and that the user gives appropriate credit (with a link to the formal publication through the relevant DOI), provides a link to the license, indicates if changes were made and the licensor is not represented as endorsing the use made of the work. Further, any new works must be made available on the same conditions. The full details of the license are available at <http://creativecommons.org/licenses/by-nc-sa/4.0>.

CC BY NC ND: The CC BY-NC-ND license allows users to copy and distribute the Article, provided this is not done for commercial purposes and further does not permit distribution of the Article if it is changed or edited in any way, and provided the user gives appropriate credit (with a link to the formal publication through the relevant DOI), provides a link to the license, and that the licensor is not represented as endorsing the use made of the work. The full details of the license are available at <http://creativecommons.org/licenses/by-nc-nd/4.0>. Any commercial reuse of Open Access articles published with a CC BY NC SA or CC BY NC ND license requires permission from Elsevier and will be subject to a fee.

Commercial reuse includes:

- Associating advertising with the full text of the Article
- Charging fees for document delivery or access
- Article aggregation
- Systematic distribution via e-mail lists or share buttons

Posting or linking by commercial companies for use by customers of those companies.




20. Other Conditions:


v1.7

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

Appendix A (Continued)

Below is permission of the use of Figure 2.8.

HomeAccount InfoHelpLive Chat



Title: Coordinated motion control of a nonholonomic mobile manipulator for accurate motion tracking

Conference Proceedings: Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on

Author: Yunyi Jia; Ning Xi; Yu Cheng; Siyang Liang

Publisher: IEEE

Date: 14-18 Sept. 2014

Copyright © 2014, IEEE

Logged in as:
Mustafa Mashali
Account #:
3000925634

LOGOUT

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.




If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.


BACKCLOSE WINDOW

Copyright © 2015 [Copyright Clearance Center, Inc.](#) All Rights Reserved. [Privacy statement](#). [Terms and Conditions](#).
Comments? We would like to hear from you. E-mail us at customercare@copyright.com

Appendix A (Continued)

Below is permission of the use of Figure 2.9.

HomeAccount InfoHelpLive Chat



Title: Experimental Evaluation of Dynamic Redundancy Resolution in a Nonholonomic Wheeled Mobile Manipulator

Author: White, G.D.; Bhatt, R.M.; Chin Pei Tang; Krovi, V.N.

Publication: Mechatronics, IEEE/ASME Transactions on

Publisher: IEEE

Date: June 2009

Copyright © 2009, IEEE

Logged in as:
Mustafa Mashali
Account #:
3000925634

LOGOUT

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.



If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACKCLOSE WINDOW

Copyright © 2015 [Copyright Clearance Center, Inc.](#) All Rights Reserved. [Privacy statement.](#) [Terms and Conditions.](#)
Comments? We would like to hear from you. E-mail us at customer@copyright.com

Appendix A (Continued)

Below is permission of the use of Figure 2.10.

HomeAccount InfoHelp



Title: Planning and model-based control for mobile manipulators
Conference Proceedings: Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on
Author: Papadopoulos, E.; Poulakakis, J.
Publisher: IEEE
Date: 2000
Copyright © 2000, IEEE

Logged in as:
Mustafa Mashali
Account #:
3000925634
LOGOUT

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.htm to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACKCLOSE WINDOW

Copyright © 2015 [Copyright Clearance Center, Inc.](#) All Rights Reserved. [Privacy statement.](#) [Terms and Conditions.](#)
Comments? We would like to hear from you. E-mail us at customercare@copyright.com

Appendix A (Continued)

Below is permission for the use of material in Chapter 5.

HomeCreate AccountHelp



Title: Task priority based dual-trajectory control for redundant mobile manipulators

Conference Proceedings: Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on

Author: Mashali, M.; Alqasemi, R.; Dubey, R.

Publisher: IEEE

Date: 5-10 Dec. 2014

Copyright © 2014, IEEE

LOGIN

If you're a **copyright.com** user, you can login to RightsLink using your copyright.com credentials. Already a **RightsLink** user or want to [learn more?](#)

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACKCLOSE WINDOW

Copyright © 2015 [Copyright Clearance Center, Inc.](#) All Rights Reserved. [Privacy statement.](#) [Terms and Conditions.](#) Comments? We would like to hear from you. E-mail us at customer@copyright.com

Appendix A (Continued)

Below is permission for the use of material in Chapter 7.

HomeCreate AccountHelpLive Chat



Title: Design, implementation and evaluation of a motion control scheme for mobile platforms with high uncertainties

Conference Proceedings: Biomechanics (2014 5th IEEE RAS & EMBS International Conference on

Author: Mashali, Mustafa; Alqasemi, Redwan; Sarkar, Sudeep; Dubey, Rajiv

Publisher: IEEE

Date: 12-15 Aug. 2014

Copyright © 2014, IEEE

LOGIN

If you're a **copyright.com** user, you can login to RightsLink using your copyright.com credentials. Already a **RightsLink** user or want to [learn more?](#)

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACKCLOSE WINDOW

Copyright © 2015 [Copyright Clearance Center, Inc.](#) All Rights Reserved. [Privacy statement.](#) [Terms and Conditions.](#)
Comments? We would like to hear from you. E-mail us at customer@copyright.com

Appendix A (Continued)

Below is permission of the use of Figure 8.2.

Alqasemi, Redwan 11:06 AM (1 hour ago)

to me

Permission Granted.

Redwan.

From: Mustafa Mashali [mailto:mmashali@mail.usf.edu]
Sent: Wednesday, November 11, 2015 10:34 PM
To: Alqasemi, Redwan
Subject: Copyright Clearance

Hello Dr. Redwan

I would like to get copyright clearance to use figures from your PhD dissertation, " Maximizing manipulation capabilities of persons with disabilities using a smart 9-degree-of-freedom wheelchair-mounted robotic arm system" in my PhD dissertation, " Kinematic Control of Redundant Mobile Manipulators".

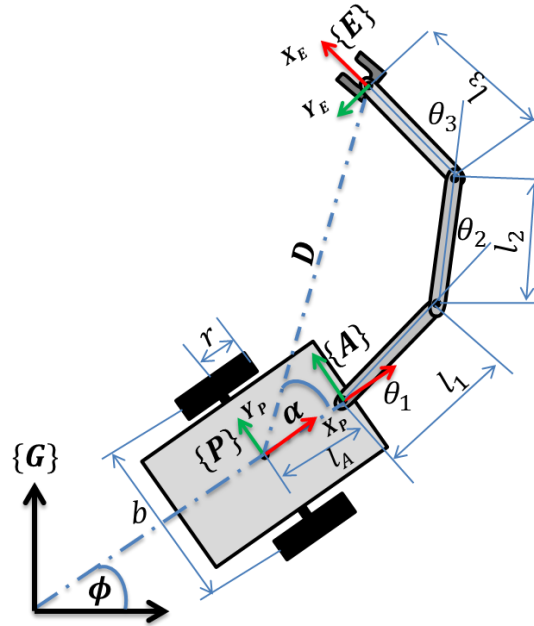
Could you please reply to this email granting permission.

Thanks,

Mustafa Mashali
--
PhD Candidate and Research Assistant ,
Department of Mechanical Engineering
University of South Florida
Ph: [\(813\)974-7367](tel:(813)974-7367)

Appendix B Planar Mobile Manipulator Jacobian Derivation

Novel method to control the dual trajectory of a mobile manipulator



Mobile manipulator kinematic model:

The kinematic model of the manipulator:

D-H parameters for the arm:

The manipulator is a planar arm with three revolute joints. The following table shows the D-H parameters for the arm

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	l_1	0	θ_2
3	0	l_2	0	θ_3
4	0	l_3	0	0

Forward kinematic for the arm:

Frame 0 is at the base and frame 4 is at end-effector. The transformation matrices for the arm:

$${}^0_1T = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^1_2T = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & l_1 \\ \sin\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Appendix B (Continued)

$${}^2_3T = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & l_2 \\ \sin\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^3_4T = \begin{bmatrix} 1 & 0 & 0 & l_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The forward kinematic for the arm is:

$${}^0_4T = {}^0_A T * {}^1_2T * {}^2_3T * {}^3_4T =$$

$$\begin{bmatrix} \cos(\theta_1 + \theta_2 + \theta_3) & -\sin(\theta_1 + \theta_2 + \theta_3) & 0 & l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ \sin(\theta_1 + \theta_2 + \theta_3) & \cos(\theta_1 + \theta_2 + \theta_3) & 0 & l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^P_E T = {}^P_A T * {}^A_E T =$$

$$\begin{bmatrix} 1 & 0 & 0 & l_A \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} *$$

$$\begin{bmatrix} \cos(\theta_1 + \theta_2 + \theta_3) & -\sin(\theta_1 + \theta_2 + \theta_3) & 0 & l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ \sin(\theta_1 + \theta_2 + \theta_3) & \cos(\theta_1 + \theta_2 + \theta_3) & 0 & l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} \cos(\theta_1 + \theta_2 + \theta_3) & -\sin(\theta_1 + \theta_2 + \theta_3) & 0 & l_A + l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ \sin(\theta_1 + \theta_2 + \theta_3) & \cos(\theta_1 + \theta_2 + \theta_3) & 0 & l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The kinematic model of the platform:

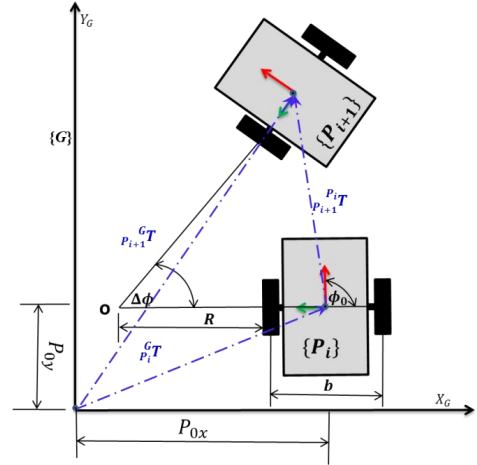
$${}^{P_i}_{P_{i+1}} T = D_y \left(R + \frac{b}{2} \right) \cdot R_z(\Delta\phi) \cdot D_y \left(-R - \frac{b}{2} \right) =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \left(R + \frac{b}{2} \right) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\Delta\phi) & -\sin(\Delta\phi) & 0 & 0 \\ \sin(\Delta\phi) & \cos(\Delta\phi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \left(-R - \frac{b}{2} \right) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} \cos(\Delta\phi) & -\sin(\Delta\phi) & 0 & \sin(\Delta\phi) \left(R + \frac{b}{2} \right) \\ \sin(\Delta\phi) & \cos(\Delta\phi) & 0 & (1 - \cos(\Delta\phi)) \cdot \left(R + \frac{b}{2} \right) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^G_{P_i} T = \begin{bmatrix} \cos(\phi_0) & -\sin(\phi_0) & 0 & P_{0x} \\ \sin(\phi_0) & \cos(\phi_0) & 0 & P_{0y} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Appendix B (Continued)

$${}_{P_{i+1}}{}^G T = \begin{bmatrix} \cos(\phi_0) & -\sin(\phi_0) & 0 & P_{0x} \\ \sin(\phi_0) & \cos(\phi_0) & 0 & P_{0y} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos(\Delta\phi) & -\sin(\Delta\phi) & 0 & \sin(\Delta\phi)\left(R + \frac{b}{2}\right) \\ \sin(\Delta\phi) & \cos(\Delta\phi) & 0 & (1 - \cos(\Delta\phi)) \cdot \left(R + \frac{b}{2}\right) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} \cos(\Delta\phi + \phi_0) & -\sin(\Delta\phi + \phi_0) & 0 & P_{0x} + \left(R + \frac{b}{2}\right)(\sin(\Delta\phi + \phi_0) - \sin(\phi_0)) \\ \sin(\Delta\phi + \phi_0) & \cos(\Delta\phi + \phi_0) & 0 & P_{0y} - \left(R + \frac{b}{2}\right)(\cos(\Delta\phi + \phi_0) - \cos(\phi_0)) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} X_{GE} \\ Y_{GE} \end{bmatrix} = \begin{bmatrix} X_{GP} \\ Y_{GP} \end{bmatrix}_G + \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \cdot \begin{bmatrix} X_{PE} \\ Y_{PE} \end{bmatrix}_P = \begin{bmatrix} P_{0x} + \left(R + \frac{b}{2}\right)(\sin(\Delta\phi + \phi_0) - \sin(\phi_0)) \\ P_{0y} - \left(R + \frac{b}{2}\right)(\cos(\Delta\phi + \phi_0) - \cos(\phi_0)) \end{bmatrix} +$$

$$\begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \cdot \begin{bmatrix} l_A + l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \end{bmatrix}$$

The aim is to control the trajectory of the platform \mathbf{P} with a reference to the trajectory of the end-effector \mathbf{E} . This will be accomplished by introducing two controlled variables \mathbf{D} and α where D is the distance between the platform and end-effector frames $D = \sqrt{X_{PE}^2 + Y_{PE}^2}$ and α is the angle of the end-effector position relative to platform frame $\alpha = \tan^{-1}\left(\frac{Y_{PE}}{X_{PE}}\right)$.

The complete state variables of the Planar Mobile Manipulator system are:

$${}_{E}^G \dot{\mathbf{r}} = \begin{bmatrix} \dot{x}_{GE} \\ \dot{y}_{GE} \\ \dot{D} \\ \dot{\alpha} \end{bmatrix} = J_m \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} J_{PE_{2 \times 3}} & \vdots & J_{GP_{2 \times 2}} \\ \cdots & \cdots & \cdots \\ & J_{D\alpha_{2 \times 5}} & \end{bmatrix}_G \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\phi} \end{bmatrix} \Rightarrow \begin{bmatrix} \dot{D} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} J_{D\alpha_{2 \times 5}} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\phi} \end{bmatrix}$$

The Mobile Manipulator Jacobian:

$$\begin{aligned} X_{GE} &= X_{GP}(X, \phi) + \cos(\phi) [l_A + l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3)] \\ &\quad - \sin(\phi) [l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3)] \\ &= X_{GP}(X, \phi) + l_A \cos(\phi) + l_1 \cos(\theta_1 + \phi) + l_2 \cos(\theta_1 + \theta_2 + \phi) \\ &\quad + l_3 \cos(\theta_1 + \theta_2 + \theta_3 + \phi) \end{aligned}$$

$$\begin{aligned} Y_{GE} &= Y_{GP}(X, \phi) + \sin(\phi) [l_A + l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3)] \\ &\quad + \cos(\phi) [l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3)] \\ &= Y_{GP}(X, \phi) + l_A \sin(\phi) + l_1 \sin(\theta_1 + \phi) + l_2 \sin(\theta_1 + \theta_2 + \phi) \\ &\quad + l_3 \sin(\theta_1 + \theta_2 + \theta_3 + \phi) \end{aligned}$$

Appendix B (Continued)

Jacobian of the manipulator with respect of the ground frame (J_{PE}):

$$J_{PE} = \nabla_{\theta_1, \theta_2, \theta_3} \begin{bmatrix} X_{GE} \\ Y_{GE} \end{bmatrix} = \begin{bmatrix} \frac{\partial X_{GE}}{\partial \theta_1} & \frac{\partial X_{GE}}{\partial \theta_2} & \frac{\partial X_{GE}}{\partial \theta_3} \\ \frac{\partial Y_{GE}}{\partial \theta_1} & \frac{\partial Y_{GE}}{\partial \theta_2} & \frac{\partial Y_{GE}}{\partial \theta_3} \end{bmatrix}$$

$$\frac{\partial X_{GE}}{\partial \theta_1} = -l_1 \sin(\theta_1 + \phi) - l_2 \sin(\theta_1 + \theta_2 + \phi) - l_3 \sin(\theta_1 + \theta_2 + \theta_3 + \phi)$$

$$\frac{\partial X_{GE}}{\partial \theta_2} = -l_2 \sin(\theta_1 + \theta_2 + \phi) - l_3 \sin(\theta_1 + \theta_2 + \theta_3 + \phi)$$

$$\frac{\partial X_{GE}}{\partial \theta_3} = -l_3 \sin(\theta_1 + \theta_2 + \theta_3 + \phi)$$

$$\frac{\partial Y_{GE}}{\partial \theta_1} = l_1 \cos(\theta_1 + \phi) + l_2 \cos(\theta_1 + \theta_2 + \phi) + l_3 \cos(\theta_1 + \theta_2 + \theta_3 + \phi)$$

$$\frac{\partial Y_{GE}}{\partial \theta_2} = l_2 \cos(\theta_1 + \theta_2 + \phi) + l_3 \cos(\theta_1 + \theta_2 + \theta_3 + \phi)$$

$$\frac{\partial Y_{GE}}{\partial \theta_3} = l_3 \cos(\theta_1 + \theta_2 + \theta_3 + \phi)$$

$$J_{PE} = \begin{bmatrix} J_{PE}(1,1) & J_{PE}(1,2) & J_{PE}(1,3) \\ J_{PE}(2,1) & J_{PE}(2,2) & J_{PE}(2,3) \end{bmatrix}$$

Where:

$$J_{PE}(1,1) = -l_1 \sin(\theta_1 + \phi) - l_2 \sin(\theta_1 + \theta_2 + \phi) - l_3 \sin(\theta_1 + \theta_2 + \theta_3 + \phi)$$

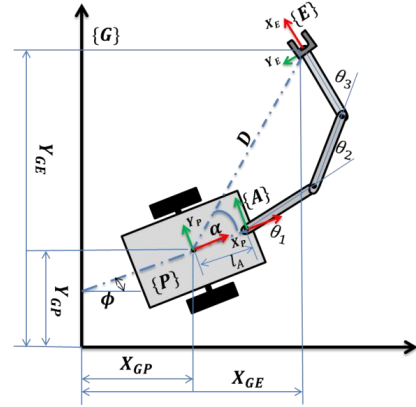
$$J_{PE}(2,1) = l_1 \cos(\theta_1 + \phi) + l_2 \cos(\theta_1 + \theta_2 + \phi) + l_3 \cos(\theta_1 + \theta_2 + \theta_3 + \phi)$$

$$J_{PE}(1,2) = -l_2 \sin(\theta_1 + \theta_2 + \phi) - l_3 \sin(\theta_1 + \theta_2 + \theta_3 + \phi)$$

$$J_{PE}(2,2) = l_2 \cos(\theta_1 + \theta_2 + \phi) + l_3 \cos(\theta_1 + \theta_2 + \theta_3 + \phi)$$

$$J_{PE}(1,3) = -l_3 \sin(\theta_1 + \theta_2 + \theta_3 + \phi)$$

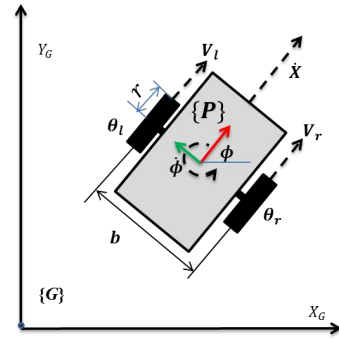
$$J_{PE}(2,3) = l_3 \cos(\theta_1 + \theta_2 + \theta_3 + \phi)$$



Jacobian of the platform with respect of the ground frame (J_{GP}):

$$\begin{bmatrix} X_{GE} \\ Y_{GE} \end{bmatrix} = \begin{bmatrix} X_{GP} \\ Y_{GP} \end{bmatrix}_G + \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \cdot \begin{bmatrix} l_A + l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \end{bmatrix}_P$$

$$J_{GP} = \nabla_{X, \phi} \begin{bmatrix} X_{GE} \\ Y_{GE} \end{bmatrix} = \begin{bmatrix} \frac{\partial X_{GE}}{\partial X} & \frac{\partial X_{GE}}{\partial \phi} \\ \frac{\partial Y_{GE}}{\partial X} & \frac{\partial Y_{GE}}{\partial \phi} \end{bmatrix} = \begin{bmatrix} \frac{\partial X_{GP}}{\partial X} + \frac{\partial X_{PE}}{\partial X} \Big|_G & \frac{\partial X_{GP}}{\partial \phi} + \frac{\partial X_{PE}}{\partial \phi} \Big|_G \\ \frac{\partial Y_{GP}}{\partial X} + \frac{\partial Y_{PE}}{\partial X} \Big|_G & \frac{\partial Y_{GP}}{\partial \phi} + \frac{\partial Y_{PE}}{\partial \phi} \Big|_G \end{bmatrix}$$



Appendix B (Continued)

$X_{GP} = X \cos(\phi)$, and $Y_{GP} = X \sin(\phi)$ where X is the platform translation and ϕ is the platform rotation.

$$\frac{\partial X_{GP}}{\partial X} = \cos(\phi) \text{ and } \frac{\partial X_{GP}}{\partial \phi} = 0 \text{ because } \phi \text{ is constant}$$

$$\frac{\partial Y_{GP}}{\partial X} = \sin(\phi) \text{ and } \frac{\partial Y_{GP}}{\partial \phi} = 0 \text{ because } \phi \text{ is constant}$$

$$X_{PE}|_G = l_A \cos(\phi) + l_1 \cos(\theta_1 + \phi) + l_2 \cos(\theta_1 + \theta_2 + \phi) + l_3 \cos(\theta_1 + \theta_2 + \theta_3 + \phi)$$

$$Y_{PE}|_G = l_A \sin(\phi) + l_1 \sin(\theta_1 + \phi) + l_2 \sin(\theta_1 + \theta_2 + \phi) + l_3 \sin(\theta_1 + \theta_2 + \theta_3 + \phi)$$

$$\left. \frac{\partial X_{PE}}{\partial X} \right|_G = \left. \frac{\partial Y_{PE}}{\partial X} \right|_G = 0$$

$$\left. \frac{\partial X_{PE}}{\partial \phi} \right|_G = -l_A \sin(\phi) - l_1 \sin(\theta_1 + \phi) - l_2 \sin(\theta_1 + \theta_2 + \phi) - l_3 \sin(\theta_1 + \theta_2 + \theta_3 + \phi)$$

$$\left. \frac{\partial Y_{PE}}{\partial \phi} \right|_G = l_A \cos(\phi) + l_1 \cos(\theta_1 + \phi) + l_2 \cos(\theta_1 + \theta_2 + \phi) + l_3 \cos(\theta_1 + \theta_2 + \theta_3 + \phi)$$

$$J_{GP} = \begin{bmatrix} \cos(\phi) & -l_A \sin(\phi) - l_1 \sin(\theta_1 + \phi) - l_2 \sin(\theta_1 + \theta_2 + \phi) - l_3 \sin(\theta_1 + \theta_2 + \theta_3 + \phi) \\ \sin(\phi) & l_A \cos(\phi) + l_1 \cos(\theta_1 + \phi) + l_2 \cos(\theta_1 + \theta_2 + \phi) + l_3 \cos(\theta_1 + \theta_2 + \theta_3 + \phi) \end{bmatrix}$$

Jacobian of the two controls variables with respect to the Ground frame ($J_{D\alpha}$):

$$\begin{bmatrix} {}^G X_{PE} \\ {}^G Y_{PE} \end{bmatrix} = \begin{bmatrix} l_A \cos(\phi) + l_1 \cos(\theta_1 + \phi) + l_2 \cos(\theta_1 + \theta_2 + \phi) + l_3 \cos(\theta_1 + \theta_2 + \theta_3 + \phi) \\ l_A \sin(\phi) + l_1 \sin(\theta_1 + \phi) + l_2 \sin(\theta_1 + \theta_2 + \phi) + l_3 \sin(\theta_1 + \theta_2 + \theta_3 + \phi) \end{bmatrix}$$

$$D = \sqrt{{}^G X_{PE}^2 + {}^G Y_{PE}^2}, \alpha = \tan^{-1} \left(\frac{{}^G Y_{PE}}{{}^G X_{PE}} \right) - \phi$$

$$\nabla_{\theta_1, \theta_2, \theta_3, X, \phi} [D] = \begin{bmatrix} \frac{\partial D}{\partial \theta_1} & \frac{\partial D}{\partial \theta_2} & \frac{\partial D}{\partial \theta_3} & \frac{\partial D}{\partial X} & \frac{\partial D}{\partial \phi} \\ \frac{\partial \alpha}{\partial \theta_1} & \frac{\partial \alpha}{\partial \theta_2} & \frac{\partial \alpha}{\partial \theta_3} & \frac{\partial \alpha}{\partial X} & \frac{\partial \alpha}{\partial \phi} \end{bmatrix}$$

$$\nabla_{\theta_1, \theta_2, \theta_3} (D) = \begin{bmatrix} \frac{\partial D}{\partial \theta_1} & \frac{\partial D}{\partial \theta_2} & \frac{\partial D}{\partial \theta_3} & \frac{\partial D}{\partial X} & \frac{\partial D}{\partial \phi} \end{bmatrix}$$

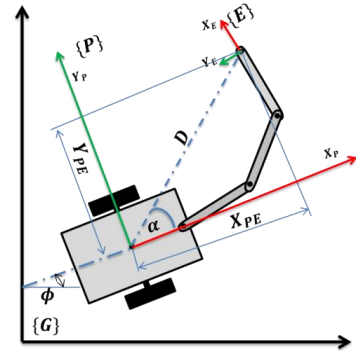
$$\frac{\partial D}{\partial \theta_1} = \frac{1}{\sqrt{{}^G X_{PE}^2 + {}^G Y_{PE}^2}} \left(\frac{\partial {}^G X_{PE}}{\partial \theta_1} + \frac{\partial {}^G Y_{PE}}{\partial \theta_1} \right)$$

$$= \frac{1}{D} (-l_1 \sin(\theta_1 + \phi) - l_2 \sin(\theta_1 + \theta_2 + \phi) - l_3 \sin(\theta_1 + \theta_2 + \theta_3 + \phi) + l_1 \cos(\theta_1 + \phi) + l_2 \cos(\theta_1 + \theta_2 + \phi) + l_3 \cos(\theta_1 + \theta_2 + \theta_3 + \phi)).$$

$$\frac{\partial D}{\partial \theta_1} = \frac{1}{D} (J_{PE}(1,1) + J_{PE}(2,1))$$

$$\frac{\partial D}{\partial \theta_2} = \frac{1}{\sqrt{{}^G X_{PE}^2 + {}^G Y_{PE}^2}} \left(\frac{\partial {}^G X_{PE}}{\partial \theta_2} + \frac{\partial {}^G Y_{PE}}{\partial \theta_2} \right) = \frac{1}{D} (-l_2 \sin(\theta_1 + \theta_2 + \phi) - l_3 \sin(\theta_1 + \theta_2 + \theta_3 + \phi) +$$

$$l_2 \cos(\theta_1 + \theta_2 + \phi) + l_3 \cos(\theta_1 + \theta_2 + \theta_3 + \phi)) = \frac{1}{D} (J_{PE}(1,2) + J_{PE}(2,2))$$



Appendix B (Continued)

$$\frac{\partial D}{\partial \theta_3} = \frac{1}{\sqrt{G_{X_{PE}}^2 + G_{Y_{PE}}^2}} \left(\frac{\partial G_{X_{PE}}}{\partial \theta_3} + \frac{\partial G_{Y_{PE}}}{\partial \theta_3} \right) = \frac{1}{D} (-l_3 \sin(\theta_1 + \theta_2 + \theta_3) + l_3 \cos(\theta_1 + \theta_2 + \theta_3)) =$$

$$\frac{1}{D} (J_{PE}(1,3) + J_{PE}(2,3))$$

$$\frac{\partial D}{\partial X} = 0$$

$$\frac{\partial D}{\partial \phi} = \frac{1}{\sqrt{G_{X_{PE}}^2 + G_{Y_{PE}}^2}} \left(\frac{\partial G_{X_{PE}}}{\partial \phi} + \frac{\partial G_{Y_{PE}}}{\partial \phi} \right) = \frac{1}{D} (-l_A \sin(\phi) - l_1 \sin(\theta_1 + \phi) - l_2 \sin(\theta_1 + \theta_2 + \phi) -$$

$$l_3 \sin(\theta_1 + \theta_2 + \theta_3 + \phi) + l_A \cos(\phi) + l_1 \cos(\theta_1 + \phi) + l_2 \cos(\theta_1 + \theta_2 + \phi) + l_3 \cos(\theta_1 + \theta_2 + \theta_3 + \phi)) = \frac{1}{D} (J_{GP}(1,2) + J_{GP}(2,2))$$

$$\alpha = \tan^{-1} \left(\frac{G_{Y_{PE}}}{G_{X_{PE}}} \right) - \phi$$

$$\nabla_{\theta_1, \theta_2, \theta_3, X, \phi} (\alpha) = \begin{bmatrix} \frac{\partial \alpha}{\partial \theta_1} & \frac{\partial \alpha}{\partial \theta_2} & \frac{\partial \alpha}{\partial \theta_3} & \frac{\partial \alpha}{\partial X} & \frac{\partial \alpha}{\partial \phi} \end{bmatrix}$$

$$\frac{\partial \alpha}{\partial \theta_1} = \frac{G_{X_{PE}}^2}{G_{X_{PE}}^2 + G_{Y_{PE}}^2} \left(\frac{\partial}{\partial \theta_1} \left(\frac{G_{Y_{PE}}}{G_{X_{PE}}} \right) \right) = \frac{G_{X_{PE}}^2}{G_{X_{PE}}^2 + G_{Y_{PE}}^2} \cdot \frac{G_{X_{PE}} \cdot \frac{\partial G_{Y_{PE}}}{\partial \theta_1} - G_{Y_{PE}} \cdot \frac{\partial G_{X_{PE}}}{\partial \theta_1}}{G_{X_{PE}}^2} = \frac{1}{G_{X_{PE}}^2 + G_{Y_{PE}}^2} \left(G_{X_{PE}} \cdot \frac{\partial G_{Y_{PE}}}{\partial \theta_1} -$$

$$G_{Y_{PE}} \cdot \frac{\partial G_{X_{PE}}}{\partial \theta_1} \right) = \frac{1}{D^2} (G_{X_{PE}} \cdot J_{PE}(2,1) - G_{Y_{PE}} \cdot J_{PE}(1,1))$$

$$\frac{\partial \alpha}{\partial \theta_2} = \frac{G_{X_{PE}}^2}{G_{X_{PE}}^2 + G_{Y_{PE}}^2} \left(\frac{\partial}{\partial \theta_2} \left(\frac{G_{Y_{PE}}}{G_{X_{PE}}} \right) \right) = \frac{G_{X_{PE}}^2}{G_{X_{PE}}^2 + G_{Y_{PE}}^2} \cdot \frac{G_{X_{PE}} \cdot \frac{\partial G_{Y_{PE}}}{\partial \theta_2} - G_{Y_{PE}} \cdot \frac{\partial G_{X_{PE}}}{\partial \theta_2}}{G_{X_{PE}}^2} = \frac{1}{G_{X_{PE}}^2 + G_{Y_{PE}}^2} \left(G_{X_{PE}} \cdot \frac{\partial G_{Y_{PE}}}{\partial \theta_2} - G_{Y_{PE}} \cdot$$

$$\frac{\partial G_{X_{PE}}}{\partial \theta_2} \right) = \frac{1}{D^2} (G_{X_{PE}} \cdot J_{PE}(2,2) - G_{Y_{PE}} \cdot J_{PE}(1,2))$$

$$\frac{\partial \alpha}{\partial \theta_3} = \frac{G_{X_{PE}}^2}{G_{X_{PE}}^2 + G_{Y_{PE}}^2} \left(\frac{\partial}{\partial \theta_3} \left(\frac{G_{Y_{PE}}}{G_{X_{PE}}} \right) \right) = \frac{G_{X_{PE}}^2}{G_{X_{PE}}^2 + G_{Y_{PE}}^2} \cdot \frac{G_{X_{PE}} \cdot \frac{\partial G_{Y_{PE}}}{\partial \theta_3} - G_{Y_{PE}} \cdot \frac{\partial G_{X_{PE}}}{\partial \theta_3}}{G_{X_{PE}}^2} = \frac{1}{G_{X_{PE}}^2 + G_{Y_{PE}}^2} \left(G_{X_{PE}} \cdot \frac{\partial G_{Y_{PE}}}{\partial \theta_3} -$$

$$G_{Y_{PE}} \cdot \frac{\partial G_{X_{PE}}}{\partial \theta_3} \right) = \frac{1}{D^2} (G_{X_{PE}} \cdot J_{PE}(2,3) - G_{Y_{PE}} \cdot J_{PE}(1,3))$$

$$\frac{\partial \alpha}{\partial X} = 0;$$

$$\frac{\partial \alpha}{\partial \phi} = \frac{G_{X_{PE}}^2}{G_{X_{PE}}^2 + G_{Y_{PE}}^2} \left(\frac{\partial}{\partial \phi} \left(\frac{G_{Y_{PE}}}{G_{X_{PE}}} \right) \right) - 1 = \frac{G_{X_{PE}}^2}{G_{X_{PE}}^2 + G_{Y_{PE}}^2} \cdot \frac{G_{X_{PE}} \cdot \frac{\partial G_{Y_{PE}}}{\partial \phi} - G_{Y_{PE}} \cdot \frac{\partial G_{X_{PE}}}{\partial \phi}}{G_{X_{PE}}^2} - 1 = \frac{1}{G_{X_{PE}}^2 + G_{Y_{PE}}^2} \left(G_{X_{PE}} \cdot \frac{\partial G_{Y_{PE}}}{\partial \phi} - G_{Y_{PE}} \cdot$$

$$\frac{\partial G_{X_{PE}}}{\partial \phi} \right) - 1 = \frac{1}{D^2} (G_{X_{PE}} \cdot J_{GP}(2,2) - G_{Y_{PE}} \cdot J_{GP}(1,2)) - 1$$

$$J_{D\alpha} = \begin{bmatrix} J_{D\alpha}(1,1) & J_{D\alpha}(1,2) & J_{D\alpha}(1,3) & J_{D\alpha}(1,4) & J_{D\alpha}(1,5) \\ J_{D\alpha}(2,1) & J_{D\alpha}(2,2) & J_{D\alpha}(2,3) & J_{D\alpha}(2,4) & J_{D\alpha}(2,5) \end{bmatrix}$$

Appendix B (Continued)

$$J_{D\alpha}(1,1) = \frac{1}{D}(l_1(\cos(\theta_1) - \sin(\theta_1)) + l_2(\cos(\theta_1 + \theta_2) - \sin(\theta_1 + \theta_2)) + l_3(\cos(\theta_1 + \theta_2 + \theta_3) - \sin(\theta_1 + \theta_2 + \theta_3)))$$

$$J_{D\alpha}(2,1) = \frac{D^2 - l_A X_{PE}}{D^2}$$

$$J_{D\alpha}(1,2) = \frac{1}{D}(l_2(\cos(\theta_1 + \theta_2) - \sin(\theta_1 + \theta_2)) + l_3(\cos(\theta_1 + \theta_2 + \theta_3) - \sin(\theta_1 + \theta_2 + \theta_3)))$$

$$J_{D\alpha}(2,2) = \frac{1}{D^2}(X_{PE} \cdot (l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3)) - Y_{PE} \cdot (-l_2 \sin(\theta_1 + \theta_2) - l_3 \sin(\theta_1 + \theta_2 + \theta_3)))$$

$$J_{D\alpha}(1,3) = \frac{1}{D}(-l_3 \sin(\theta_1 + \theta_2 + \theta_3) + l_3 \cos(\theta_1 + \theta_2 + \theta_3))$$

$$J_{D\alpha}(2,3) = \frac{1}{D^2}(X_{PE} \cdot (l_3 \cos(\theta_1 + \theta_2 + \theta_3)) - Y_{PE} \cdot (-l_3 \sin(\theta_1 + \theta_2 + \theta_3)))$$

$$J_{D\alpha}(1,4) = J_{D\alpha}(2,4) = 0$$

$$J_{D\alpha}(1,5) = \frac{1}{D}(J_{GP}(1,2) + J_{GP}(2,2))$$

$$J_{D\alpha}(2,5) = \frac{1}{D^2}({}^G X_{PE} \cdot J_{GP}(2,2) - {}^G Y_{PE} \cdot J_{GP}(1,2)) - 1$$

Appendix C Hardware Implementation C++ Code

```
#include "application.h"

#include <string>

#include "Galil.h" //vector string Galil
#include "dmc-class.h"

#pragma comment (lib, "libmat.lib")
#pragma comment (lib, "libmx.lib" )
#pragma comment (lib, "libmex.lib")
#pragma comment (lib, "libeng.lib")
#pragma comment (lib, "libut.lib")

using namespace std;

using namespace Eigen;

//double PI = 3.141592654;
// char *tempChar; //temporary char pointer passed to thread
// int flag = 0;
//extern bool FirstCall;

#define DEFAULT_BUFFER_LENGTH 512
Controller Galilwheel("192.168.1.40");
Controller Galilarm("192.168.1.22");
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Motor control and wheelmotor initialization
void wheel_motors_initialization(void)
{
    // initilize the controller for the wheelchair
    Galilwheel.Abort(1);
    Galilwheel.RS();
    Galilwheel.ST("AB");
    //Galilwheel.BG("AB");
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Motor control and Arm_motor initialization
void arm_motors_initialization(void)
{
    //send_disconnection_protect_joint_position();
    Galilarm.ST("ABCDEFGH");
    Galilarm.AC("A", 180000);
    Galilarm.DC("A", 180000);
    Galilarm.AC("B", 800000);
    Galilarm.DC("B", 800000);
    Galilarm.AC("C", 800000);
    Galilarm.DC("C", 800000);
    Galilarm.AC("D", 800000);
    Galilarm.DC("D", 800000);
    Galilarm.AC("E", 800000);
    Galilarm.DC("E", 800000);
    Galilarm.AC("F", 800000);
    Galilarm.DC("F", 800000);
    Galilarm.AC("G", 400000);
    Galilarm.DC("G", 400000);
    Galilarm.AC("H", 800000);
    Galilarm.DC("H", 800000);
    Galilarm.BG("ABCDEFGH");
}

}
```

Appendix C (Continued)

```
////////////////////////////////////
void set_joint_speed(VectorXd & JointSpeed)
{
    Galilarm.JG("A", JointSpeed(0));
    Galilarm.JG("B", JointSpeed(1));
    Galilarm.JG("C", JointSpeed(2));
    Galilarm.JG("D", JointSpeed(3));
    Galilarm.JG("E", JointSpeed(4));
    Galilarm.JG("F", JointSpeed(5));
    Galilarm.JG("G", JointSpeed(6));

    /*Galilarm.SP("A", JointSpeed(0));
    Galilarm.SP("B", JointSpeed(1));
    Galilarm.SP("C", JointSpeed(2));
    Galilarm.SP("D", JointSpeed(3));
    Galilarm.SP("E", JointSpeed(4));
    Galilarm.SP("F", JointSpeed(5));
    Galilarm.SP("G", JointSpeed(6));*/
}
////////////////////////////////////
void set_joint_speed_SP(VectorXd & JointSpeed)
{
    Galilarm.SP("A", JointSpeed(0));
    Galilarm.SP("B", JointSpeed(1));
    Galilarm.SP("C", JointSpeed(2));
    Galilarm.SP("D", JointSpeed(3));
    Galilarm.SP("E", JointSpeed(4));
    Galilarm.SP("F", JointSpeed(5));
    Galilarm.SP("G", JointSpeed(6));
}
////////////////////////////////////
void set_joint_position(VectorXd & AngleEncoder)
{
    Galilarm.IP("A", AngleEncoder(0));
    Galilarm.IP("B", AngleEncoder(1));
    Galilarm.IP("C", AngleEncoder(2));
    Galilarm.IP("D", AngleEncoder(3));
    Galilarm.IP("E", AngleEncoder(4));
    Galilarm.IP("F", AngleEncoder(5));
    Galilarm.IP("G", AngleEncoder(6));
}
////////////////////////////////////
void ArmJointAngle(VectorXi & joint_position)
{
    joint_position(0) = Galilarm.Reference_Position("A");
    joint_position(1) = Galilarm.Reference_Position("B");
    joint_position(2) = Galilarm.Reference_Position("C");
    joint_position(3) = Galilarm.Reference_Position("D");
    joint_position(4) = Galilarm.Reference_Position("E");
    joint_position(5) = Galilarm.Reference_Position("F");
    joint_position(6) = Galilarm.Reference_Position("G");
}
////////////////////////////////////
#define up 72
#define down 80
#define left 75
#define right 77
#define Global_speed_max 10
```

Appendix C (Continued)

```

void move_wheel(double left_wheel, double right_wheel)
{
    int temp_left=0, temp_right=0;
    temp_left =left_wheel*2000;// encodr counts/sec
    temp_right=right_wheel*2000;

    Galilwheel.JGA(temp_left);
    Galilwheel.JGB(temp_right);

}

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
VectorXd dHo;

int main()
{

    double pi = M_PI, dq_X = 0, dq_phi = 0;
    double detJ0, XGpe, YGpe, ZGpe, DeltaX, DeltaY, DeltaZ;
    double d2r =pi/180.0;
    double r2d = 180.0/pi;
    string A = "A";
    string B = "B";
    int KP_WC=0;
    int Prv_encoder_R_Command = 0;
    int Prv_encoder_L_Command = 0;
    int Actual_encoder_R = 0;
    int Actual_encoder_L = 0;
    int Error_encoder_L = 0;
    int Error_encoder_R = 0;
    int OnlyWC = 0;
    int OnlyArm = 0;
    int TrajLength = 0, encoder_R=0, encoder_L=0, LeftEncoder_FB_prv=0, RightEncoder_FB_prv=0,
    LeftEncoder_FB=0, RightEncoder_FB=0;
    double tic, toc;
    VectorXi AngleEncoder(7);
    VectorXd MPi(3), qAi(7), qPi(2),dx_6(6), qn(9), JointSpeed(7);
    VectorXd M(5), L(5), qarm_actual(7);
    Matrix4d TGPI(4,4), Tbase(4,4), TPA(4,4), TGA(4,4),TGEi(4,4), TAE(4,4), TGP(4,4),
    Te_d(4,4), TGE(4,4) ;
    Matrix4d T01(4,4), T12(4,4), T23(4,4), T34(4,4), T45(4,4), T56(4,4), T67(4,4);
    MatrixXd J_PE(6,7), J_PE_G(6,7), R_GF(6,6),J_GP(6,2),JAP_BAXBOT(6,9),
    J_MPF(2,9),J_BAXBOT(8,9), J_DAlphaBeta(3,9) ;
    vector<Eigen::Matrix4d,Eigen::aligned_allocator<Eigen::Matrix4d> > Tt_MB, Tt_EE;
    VectorXi Prv_Armjoint_encoder_Command(7), Actual_Armjoint_encoder(7),
    Prv_Armjoint_encoder(7);
    Prv_Armjoint_encoder_Command << 0,0,0,0,0,0,0;
    Actual_Armjoint_encoder = Prv_Armjoint_encoder_Command = Prv_Armjoint_encoder;
    VectorXd Test_Encoder(10);

    Test_Encoder<< 1000, -1000, 1000,1000,2000,3000,5000,5000,2000,-2000;
    //////////////////////////////////////
    std::ofstream saveTGE("EndEffectorPosition.csv", std::ostream::out);
    std::ofstream saveTGP("wheelchairPosition.csv", std::ostream::out);
    std::ofstream saveMM("ArmSystemManipulability.csv", std::ostream::out);
    std::ofstream E_Traj("EE_Trajectory.csv", std::ostream::out);
    std::ofstream WC_Traj("WCTrajectory.csv", std::ostream::out);
    std::ofstream Specs("programParametersInitilization.csv", std::ostream::out);

```

Appendix C (Continued)

```
std::ofstream DAlphaBeta("DAB.csv", std::ostream::out);
std::ofstream saveThrThl("Theta_R_L.csv", std::ostream::out);
std::ofstream saveArmTheta("A_Th.csv", std::ostream::out);
E_Traj<<"Xe"<<" , "<<"Ye"<<" , "<<"Ze"<<"\n";
WC_Traj<<"Xp"<<" , "<<"Yp"<<" , "<<"Zp"<<"\n";
DAlphaBeta<<"D"<<" , "<<"Alpha"<<" , "<<"Beta"<<"\n";
saveThrThl<<"Theta Right"<<" , "<<"Theta Left"<<"\n";
saveArmTheta<<"Theta 1"<<" , "<<"Theta 2"<<" , "<<"Theta 3"<<" , "<<"Theta 4"<<" , "<<"Theta
5"<<" , "<<"Theta 6"<<" , "<<"Theta 7"<<"\n";

/*****/
// Put the arm in ready position
// [ 90 90 0 90 90 90 0];
VectorXd ReadyAngle(7);
ReadyAngle <<90 ,90 ,0, 90, 90, 60, 0;
ReadyAngle = ReadyAngle*d2r;

/////ShellExecute (NULL,"open","C:\Users\RobotLab\Documents\Visual Studio
2010\Projects\Mustafa Projects\WMRASystem\build\WMRA_calibration.exe",NULL,NULL,
SW_SHOWDEFAULT);
//ShellExecute (NULL,"open","WMRA_calibration.exe",NULL,NULL, SW_SHOWDEFAULT);
/////system ("C:\Users\RobotLab\Documents\Visual Studio 2010\Projects\Mustafa
Projects\WMRASystem\build\WMRA_calibration.exe");
/////system ("WMRA_calibration.exe");
//system("pause");

/*cout<<" Motor A=: "<<Galilwheel.Reference_Position(A)<<endl;
cout<<" Motor B=: "<<Galilwheel.Reference_Position("B")<<endl;
cout<<" Motor C=: "<<Galilwheel.Reference_Position("C")<<endl;
cout<<" Motor D=: "<<Galilwheel.Reference_Position("D")<<endl;
cout<<" Motor E=: "<<Galilwheel.Reference_Position("E")<<endl;
cout<<" Motor F=: "<<Galilwheel.Reference_Position("F")<<endl;
cout<<" Motor G=: "<<Galilwheel.Reference_Position("G")<<endl;*/

// Reading WMRA systm,s's constant dimention, all dimention are
// converted in millimeters:
L=BAXBOT_PBD();

// Method: 0 no task priority
// Method: 2 task priority
int Method = 2;

// Priority: 1 for end-effector trajectory as higher prioieity
// Priority: 2 for the wheelchair trajectory as higher priority

int Priority = 1;
int JLA =0;
int MM = 0; // Maximize the Manipulability Measure
OnlyWC = 1; // 0 = No, 1 = yes
OnlyArm =1; // 0 = No, 1 = yes
if(Priority == 1){
    KP_WC = 4;
}else{
    KP_WC = 1.25;
}
int KP_arm = 2;
// Linear and angular velocities of the end-effector and mobile base
double v = 50; // Linear Velocity of the gripper and the Pltform in mm/s.
double vw = 0.05; // Rotation Velocity of the platform in rad/s.
double dt=0.05; // Time increment in seconds.
//LoadProgramParameters();
```

Appendix C (Continued)

```

/*****
//
// Dual-Trajectory Parameters
*****/

// Initilize the parameters
int freq_E = 0, amplitude_E_Y = 0.0, amplitude_E_Z = 0, freq_P = 0, amplitude_P = 0, SimTime=0;
double step = 0, X_sinusoidal=0;
step = 5;
double StartPoint=0;

X_sinusoidal = 3500; // whole distance
SimTime = 60; // Whole Simulation Time
int Iter_n = (int)(X_sinusoidal/step); // Number of iteration rounded up
//step = X_sinusoidal/Iter_n; // distance for each time step
//dt=(double)SimTime/Iter_n;

if(Method == 0)
{

    // End effector trajectory
    freq_E = 1500;
    amplitude_E_Y = 0.0; //-2000;%950;
    amplitude_E_Z = 0.0; //-2000;%950;

    // Mobile platform trajectory
    MPi=VectorXd::Zero(3);
    freq_P = 1500;
    amplitude_P = 0; //300;%600;
    // Starting robotic arm configuration joint angle
    qAi << 90 ,90 ,135, 90, 90, -60, 0; // ready position joint angles converted to
radians
//qAi << 45,60,90,80,0,60,100;
}
if(Method == 2)
{

    // End effector trajectory
    freq_E = 4000;
    amplitude_E_Y = 0; //-2000;%950;
    amplitude_E_Z = 0; //-2000;%950;

    // Mobile platform trajectory
    MPi=VectorXd::Zero(3);
    freq_P = 8000;
    amplitude_P = -400; //300;%600;
    // Starting robotic arm configuration joint angle
    qAi << 90 ,90 ,135, 90, 90, -30, 0; // ready position joint angles converted to
radians
//qAi << 45,90,90,90,0,0,0;
}
TrajLength = Iter_n;
TrajLength = (X_sinusoidal/step)+1;
VectorXd TT= VectorXd::LinSpaced(TrajLength,0,X_sinusoidal);
//VectorXd TT(TrajLength);
for (int i=0; i<TrajLength-1; i++)
    TT(i) = i*step;
//WMRA_PolyBlend(StartPoint, X_sinusoidal, TrajLength, TT);
// Mobile base Trajectory
VectorXd SinPoints = 2*pi*TT/freq_P;
VectorXd Xp = TT + VectorXd::Ones(TrajLength)*MPi(0);

```


Appendix C (Continued)

```
VectorXd Yp = MPi(1)+amplitude_P*SinPoints.array().sin();

// Update the mobile base orientation angle according to the trajectory
MPi(2) = atan2(Yp(1)-Yp(0),Xp(1)-Xp(0));

qAi = qAi*d2r;

// Calculating the Transformation Matrix of the initial position of the Planar mobile
manipulator's base:
TGpi=BAXBOT_pose2T(MPi(0),MPi(1),MPi(2));

// Calculating the initial Mobile base Variables
qPi<<sqrt(MPi(0)*MPi(0)+MPi(1)*MPi(1)), MPi(2) ;
//cout<<qPi<<endl;
//Calculating the number of iteration and the time increment (deltat) if
// the linear step increment of the tip is 1 mm:
Specs << "Priority" << " , " << Priority << " , "<<"JLA"<< " , " <<JLA<<"\n";
Specs << "Method" << " , " << Method << " , "<<"MM"<< " , "<<MM<< " \n";
Specs << "qAi" << " , " << qAi(0) << " , " << qAi(1) << " , " << qAi(2) << " , " << qAi(3) << " , "
<< qAi(4) << " , " << qAi(5) << " , " << qAi(6) << " \n";
Specs << "freq_E" << " , " << freq_E << " , " << "amplitude_E_Y" << " , " << amplitude_E_Y << " , "
<< "amplitude_E_Z" << " , " << amplitude_E_Z<<" \n";
Specs << "freq_P" << " , " << freq_P << " , " << "mplitude_P" << " , " << amplitude_P << " , " <<
"step" << " , " << step<<"\n";
Specs.close();
// In case of using the wheelchair
if(OnlyWC){

wheel_motors_initialization();
//Wheelchair pose to encoder count
TranslationToEncoder(qPi(0), qPi(1), encoder_R, encoder_L);

//set the speed of the wheelchair
Galilwheel.SP(A,6000);
Galilwheel.SP(B,6000);

// Left wheel Motor A
Galilwheel.PA(A, encoder_L);

// Right wheel Motor B
Galilwheel.PA(B, encoder_R);

// Begin motion
Galilwheel.BG(A);
Galilwheel.BG(B);
while(abs(Galilwheel.Reference_Position(A)) < abs(encoder_L) &&
abs(Galilwheel.Reference_Position(B))< abs(encoder_R) ){

Sleep(100);

}
// Previous commanded encoder count for each wheel
Prv_encoder_R_Command = Prv_encoder_R_Command + encoder_R;

Prv_encoder_L_Command = Prv_encoder_L_Command + encoder_L;

// Current actual Encoder counts for Right and Left wheels
LeftEncoder_FB = Galilwheel.Reference_Position(A);
RightEncoder_FB = Galilwheel.Reference_Position(B);
}
```

Appendix C (Continued)

```

// In case of using robotic arm
if(OnlyArm){
    Galilarm.ST("ABCDEFGH");
    Galilarm.AC("A", 180000);
    Galilarm.DC("A", 180000);
    Galilarm.AC("B", 800000);
    Galilarm.DC("B", 800000);
    Galilarm.AC("C", 800000);
    Galilarm.DC("C", 800000);
    Galilarm.AC("D", 800000);
    Galilarm.DC("D", 800000);
    Galilarm.AC("E", 800000);
    Galilarm.DC("E", 800000);
    Galilarm.AC("F", 800000);
    Galilarm.DC("F", 800000);
    Galilarm.AC("G", 400000);
    Galilarm.DC("G", 400000);
    Galilarm.AC("H", 800000);
    Galilarm.DC("H", 800000);
    // set he encoder value to zero
    Galilarm.DP("A",0);
    Galilarm.DP("B",0);
    Galilarm.DP("C",0);
    Galilarm.DP("D",0);
    Galilarm.DP("E",0);
    Galilarm.DP("F",0);
    Galilarm.DP("G",0);
    Galilarm.MT("F",-1);
    Galilarm.CE("F",2);

    JointSpeed << 100000, 100000, 100000, 100000, 75000, 75000, 75000;
    set_joint_speed_SP(JointSpeed);
    // Send the arm to absolute confogration relative to ready position
    VectorXd DeltaAngle (7);
    DeltaAngle = qAi - ReadyAngle;
    //cout<<" Motor B=: "<<DeltaAngle<<endl;
    WMRA_Rad2Encoder(DeltaAngle, AngleEncoder);
    //cout<<(int)AngleEncoder(0)<<endl;
    Galilarm.PA("A", (int)AngleEncoder(0));
    Galilarm.PA("B", (int)AngleEncoder(1));
    Galilarm.PA("C", (int)AngleEncoder(2));
    Galilarm.PA("D", (int)AngleEncoder(3));
    Galilarm.PA("E", (int)AngleEncoder(4));
    Galilarm.PA("F", (int)AngleEncoder(5));
    Galilarm.PA("G", (int)AngleEncoder(6));
    Galilarm.BG("ABCDEFGH");
    while(abs(Galilarm.Reference_Position("A")) < abs((int)AngleEncoder(0)) ||
          abs(Galilarm.Reference_Position("B")) < abs((int)AngleEncoder(1)) ||
          abs(Galilarm.Reference_Position("C")) < abs((int)AngleEncoder(2)) ||
          abs(Galilarm.Reference_Position("D")) < abs((int)AngleEncoder(3)) ||
          abs(Galilarm.Reference_Position("E")) < abs((int)AngleEncoder(4)) ||
          abs(Galilarm.Reference_Position("F")) < abs((int)AngleEncoder(5)) ||
          abs(Galilarm.Reference_Position("G")) < abs((int)AngleEncoder(6))){

        Sleep(100);

    }

    Galilarm.ST("ABCDEFGH");
    Galilarm.JG("A", 0);
    Galilarm.JG("B", 0);
    Galilarm.JG("C", 0);
    Galilarm.JG("D", 0);
    Galilarm.JG("E", 0);

```

Appendix C (Continued)

```

Galilarm.JG("F", 0);
Galilarm.JG("G", 0);
Galilarm.BG("ABCDEFGH");

Prv_Armjoint_encoder_Command = AngleEncoder;
ArmJointAngle(Prv_Armjoint_encoder); // read the arm joints
//cout<<" Prv_Armjoint_encoder=: \n\n"<<Prv_Armjoint_encoder<<endl;
}

// The transformation matrix arm base relative to the platform frame

TPA<< 1, 0, 0, L(1),
      0, 1, 0, L(2),
      0, 0, 1, L(3),
      0, 0, 0, 1;

//TPA = TPA * Tbase;

//Calculating the Transformation Matrix of the initial position of the PMM's base:
TGA=TGPi*TPA;

BATBOT_TransformationMatrices(1, qAi, qPi, TGPi, TGEi, TAE, TGP, T01, T12, T23, T34, T45,
T56, T67);

// End effector trajectory
SinPoints = 2*pi*TT/freq_E;
VectorXd Xe = VectorXd::Ones(TrajLength)*TGEi(0,3) + TT ;
VectorXd Ye = TGEi(1,3)+amplitude_E_Y*(SinPoints.array().sin());
VectorXd Ze = TGEi(2,3)+amplitude_E_Z*(SinPoints.array().sin());
for(int count=0; count<Xe.size();count++){
    /*E_Traj<<Xe <<" "<<Ye<<" "<<Ze<<"\n";
    WC_Traj<<Xp <<" "<<Yp<<" "<<"\n";*/
}

Te_d = Matrix4d::Identity(4,4);
Te_d.block(0,0,3,3) =TGEi.block(0,0,3,3);
Te_d.block(0,3,3,1) << Xe(TrajLength-1), Ye(TrajLength-1), Ze(TrajLength-1);

//Calculating the number of iteration and the time increment (deltat) if
// the linear step increment of the tip is 1 mm:
//double dt=0.2; // Time increment in seconds.
//double D_Trashold_Max = 4000;
double D_Trashold_Min = 0;

// Initializing the joint angles, the Transformation Matrix, and time:
VectorXd dq=VectorXd::Zero(9);
double dg=0;
VectorXd qo(9);
qo << qAi, qPi;
qn = qo;
double tt=0;
VectorXd dx =VectorXd::Zero(8);
VectorXd D =VectorXd::Zero(TrajLength);
VectorXd Alpha=VectorXd::Zero(TrajLength);
VectorXd Beta =VectorXd::Zero(TrajLength);

// Find the parametera D, Alpha and Beta

```

Appendix C (Continued)

```
BAXBOT_D_Alpha_Beta(Xp,Yp,Xe,Ye,Ze, D, Alpha, Beta, DAlphaBeta );

// Find the transformation matrices for the waypoints of the Mobile base trajectory
BAXBOT_MB_traj(TGPI, Xp, Yp, Tt_MB);

// Find the transformation matrices for the waypoints of the end-effector trajectory
BAXBOT_EE_Traj(TGEi, Te_d, Xe,Ye,Ze, Tt_EE);

TGA=TGP*TPA;
TGP=TGPI;
TGE=TGEi;
dHo=VectorXd::Zero(9);

/*=====*/
/* Matlab simulation */

/*=====*/
=====*/
/* open matlab engine */
Engine *m_pEngine;
m_pEngine = engOpen("null");

/*=====*/
=====*/

/*===== Passing Data to Matlab =====*/
double C_T[4][4];
double *ii_p,*Xtemparray, *Ytemparray, *Ztemparray, *Xptemparray, *Yptemparray;
VectorXd ii(1);
mxArray *EE_XE, *EE_YE, *EE_ZE, *EE_Xp, *EE_Yp, *M_T, *ii_A;

M_T= mxCreateNumericMatrix(4,4, mxDOUBLE_CLASS,mxREAL);

Map<Matrix<double,4,4,RowMajor> >(*C_T,4,4) = TGEi;
memcpy(mxGetPr(M_T), C_T, sizeof(C_T));
engPutVariable(m_pEngine, "TGEi", M_T);

Map<Matrix<double,4,4,RowMajor> >(*C_T,4,4) = TGP;
memcpy(mxGetPr(M_T), C_T, sizeof(C_T));
engPutVariable(m_pEngine, "TGP", M_T);

Map<Matrix<double,4,4,RowMajor> >(*C_T,4,4) = T01;
memcpy(mxGetPr(M_T), C_T, sizeof(C_T));
engPutVariable(m_pEngine, "T01", M_T);

Map<Matrix<double,4,4,RowMajor> >(*C_T,4,4) = T12;
memcpy(mxGetPr(M_T), C_T, sizeof(C_T));
engPutVariable(m_pEngine, "T12", M_T);

Map<Matrix<double,4,4,RowMajor> >(*C_T,4,4) = T23;
memcpy(mxGetPr(M_T), C_T, sizeof(C_T));
engPutVariable(m_pEngine, "T23", M_T);

Map<Matrix<double,4,4,RowMajor> >(*C_T,4,4) = T34;
memcpy(mxGetPr(M_T), C_T, sizeof(C_T));
engPutVariable(m_pEngine, "T34", M_T);

Map<Matrix<double,4,4,RowMajor> >(*C_T,4,4) = T45;
memcpy(mxGetPr(M_T), C_T, sizeof(C_T));
engPutVariable(m_pEngine, "T45", M_T);

Map<Matrix<double,4,4,RowMajor> >(*C_T,4,4) = T56;
memcpy(mxGetPr(M_T), C_T, sizeof(C_T));
```

Appendix C (Continued)

```

engPutVariable(m_pEngine, "T56", M_T);

Map<Matrix<double,4,4,RowMajor> >(*C_T,4,4) = T67;
memcpy(mxGetPr(M_T), C_T, sizeof(C_T));
engPutVariable(m_pEngine, "T67", M_T);

Map<Matrix<double,4,4,RowMajor> >(*C_T,4,4) = Te_d;
memcpy(mxGetPr(M_T), C_T, sizeof(C_T));
engPutVariable(m_pEngine, "Td", M_T);

ii<<1;

ii_A = mxCreateDoubleMatrix(1,1,mxREAL);
ii_p=mxGetPr(ii_A);
ii_p[0] = ii[0];

EE_XE = mxCreateDoubleMatrix(1,Xe.size(),mxREAL);
EE_YE = mxCreateDoubleMatrix(1,Xe.size(),mxREAL);
EE_ZE = mxCreateDoubleMatrix(1,Xe.size(),mxREAL);
EE_Xp = mxCreateDoubleMatrix(1,Xp.size(),mxREAL);
EE_Yp = mxCreateDoubleMatrix(1,Xp.size(),mxREAL);

Xtemparray = mxGetPr(EE_XE);
Ytemparray = mxGetPr(EE_YE);
Ztemparray = mxGetPr(EE_ZE);
Xptemparray = mxGetPr(EE_Xp);
Yptemparray = mxGetPr(EE_Yp);

for(unsigned i=0; i<Xe.size(); i++){
    Xtemparray[i] = Xe[i];
    Ytemparray[i] = Ye[i];
    Ztemparray[i] = Ze[i];
    Xptemparray[i] = Xp[i];
    Yptemparray[i] = Yp[i];
}

engPutVariable(m_pEngine, "Xe", EE_XE);
engPutVariable(m_pEngine, "Ye", EE_YE);
engPutVariable(m_pEngine, "Ze", EE_ZE);
engPutVariable(m_pEngine, "Xp", EE_Xp);
engPutVariable(m_pEngine, "Yp", EE_Yp);
engPutVariable(m_pEngine, "iii", ii_A);

// Printing the output of the matlab engine in C Command window
/*char matlab_out[10000];
engOutputBuffer(m_pEngine , matlab_out, 10000);
*/

// Run the matlab simulation function
engEvalString(m_pEngine, "WMRA_ML_CPP_Dual_Trajectory_Animation(1, TGEi, Td,
TGP,Xe,Ye,Ze,Xp,Yp, T01, T12, T23, T34, T45, T56, T67,1)");

// Printing the output.
//printf("%s", matlab_out);

/*=====
=====*/

//// Starting a timer:
//tic = clock()/1000.0;
VectorXd dqgL(TrajLength-2), dqgR(TrajLength-2);
int i =0;
int k = 0;

```

Appendix C (Continued)

```

vector<Eigen::Vector3d,Eigen::aligned_allocator<Eigen::Vector3d> > PGP(TrajLength-1),
PGE(TrajLength-1);
Galilwheel.ST("AB");
Galilwheel.JG("A", 0);
Galilwheel.JG("B", 0);

Galilwheel.BG("AB");

int LeftWheel_speed_prv = 0;
int RightWheel_speed_prv = 0;
saveMM<<"armMM"<<" "<<"6x9J_MM"<<" "<<"wholeSystemMM"<<"\n";
saveTGE<<"TGEx"<<" "<<"TGEy"<<" "<<"TGEz"<<" "<<"\n";
saveTGP<<"TGPx"<<" "<<"TGPy"<<" "<<"TGPz"<<" "<<"\n";
// Starting a timer:
tic = clock()/((double)CLOCKS_PER_SEC);
while ( i < TrajLength-2)
{
    PGP.at(k) = TGP.block(0,3,3,1);
    PGE.at(k) = TGE.block(0,3,3,1);
    saveTGE<<TGE(0,3)<<" "<<TGE(1,3)<<" "<<TGE(2,3)<<" "<<"\n";
    saveTGP<<TGP(0,3)<<" "<<TGP(1,3)<<" "<<TGP(2,3)<<" "<<"\n";

    // Calculating the 6X7 Jacobian of the arm in frame A:
    WMRA_J07(T01, T12, T23, T34, T45, T56, T67, J_PE, detJ0);

    double phi=atan2(TGP(1,0),TGP(0,0));
    R_GF = MatrixXd::Zero(6,6);
    R_GF.block(0,0,3,3) = BAXBOT_rotz(phi).block(0,0,3,3);
    R_GF.block(3,3,3,3) = BAXBOT_rotz(phi).block(0,0,3,3);
    J_PE_G = R_GF*J_PE;
    //cout<<" J_PE_G=: \n" <<J_PE_G<<endl;

    MatrixXd J0_T = J_PE_G;
    J0_T.transposeInPlace();
    double detJ_arm =sqrt( (J_PE_G*J0_T).determinant());

    // Calculating the 6X2 Jacobian based on the WMRA's base in the ground frame:
    // the wheelchair frame has the same orientation of the arm base frame
    BAXBOT_JGA(phi, TAE.block(0,3,2,1), J_GP);
    //cout<<" J_GP=: \n" <<J_GP<<endl;
    JAP_BAXBOT << J_PE_G ,J_GP;
    MatrixXd JAP_BAXBOT_T = JAP_BAXBOT;
    JAP_BAXBOT_T.transposeInPlace();
    double detJAP_BAXBOT =sqrt( (JAP_BAXBOT*JAP_BAXBOT_T).determinant());

    //if (D(i)>D_Trashold_Max){ D(i) = D_Trashold_Max;}
    //if (D(i)<D_Trashold_Min){ D(i) = D_Trashold_Min;}

    if ( Method == 0 ){
        //BAXBOT_D_Alpha_Beta(qn, J_DAlphaBeta);
        WMRA_JDAlphaBeta(qn, J_DAlphaBeta);
        J_BAXBOT<< J_PE_G, J_GP,
            J_DAlphaBeta.block(0,0,2,9);
        //cout<<" J_DAlphaBeta=: \n\n" <<J_DAlphaBeta<<endl;
        BAXBOT_delta(TGE,Tt_EE.at(i+1), dx_6);
        dx.head(6) = dx_6;

        // determine the current D and Alpha
        XGpe = TGE(0,3)-TGP(0,3);
        YGpe = TGE(1,3)-TGP(1,3);
        ZGpe = TGE(2,3)-TGP(2,3);

        double D_PE = sqrt(XGpe*XGpe+YGpe*YGpe+ZGpe*ZGpe);
        double Alpha_test = atan2(YGpe,XGpe);

```

Appendix C (Continued)

```

        double Alpha_PE = Alpha_test - phi;
        dx(6) = (D(i+1)-D_PE);
        dx(7) = ((Alpha(i+1)-Alpha_PE));
        //cout<<" dx=: \n" <<dx<<endl;
    }

else if ( Method == 1 ){
    J_MPF << 0, 0, 0, 0, 0, 0, 0, 0, L(4), 0,
           0, 0, 0, 0, 0, 0, 0, 0, 1;
    J_BAXBOT<<J_PE_G, J_GP,
            J_MPF;

    BAXBOT_delta(TGE,Tt_EE.at(i+1), dx_6);
    dx.head(6) = dx_6;

    DeltaX = Tt_MB.at(i+1)(0,3) - TGP(0,3);
    DeltaY = Tt_MB.at(i+1)(1,3) - TGP(1,3);
    DeltaZ = Tt_MB.at(i+1)(2,3) - TGP(2,3);
    dx(6) = sqrt(DeltaX*DeltaX+DeltaY*DeltaY+ DeltaZ*DeltaZ);
    dx(7) = atan2(DeltaY,DeltaX)-phi;}
else if( Method == 2 ){
    if(Priority ==1){

        WMRA_JDAlphaBeta(qn, J_DAlphaBeta);

        J_BAXBOT<< J_PE_G, J_GP,
                J_DAlphaBeta.block(0,0,2,9);

        BAXBOT_delta(TGE,Tt_EE.at(i+1), dx_6);
        dx.head(6) = dx_6;

        // determine the current D and Alpha
        XGpe = TGE(0,3)-TGP(0,3);
        YGpe = TGE(1,3)-TGP(1,3);
        ZGpe = TGE(2,3)-TGP(2,3);

        double D_PE = sqrt(XGpe*XGpe+YGpe*YGpe+ZGpe*ZGpe);
        double Alpha_test = atan2(YGpe,XGpe);
        double Alpha_PE = Alpha_test - phi;
        dx(6) = (D(i)-D_PE);
        dx(7) = ((Alpha(i)-Alpha_PE));
    }

else if(Priority==2){
    J_MPF<< 0, 0, 0, 0, 0, 0, 0, 0, L(4), 0,
           0, 0, 0, 0, 0, 0, 0, 0, 1;
    WMRA_JDAlphaBeta(qn, J_DAlphaBeta);
    //if (D(i)>D_Trashold_Max){ D(i) = D_Trashold_Max;}
    J_BAXBOT<< J_MPF
            J_DAlphaBeta
            JAP_BAXBOT.block(3,0,3,9);

    BAXBOT_delta(TGE,Tt_EE.at(i+1), dx_6);
    // determine the current D, alpha and Beta
    XGpe = TGE(0,3)-TGP(0,3);
    YGpe = TGE(1,3)-TGP(1,3);
    ZGpe = TGE(2,3)-TGP(2,3);
    double D_PE = sqrt(XGpe*XGpe+YGpe*YGpe+ZGpe*ZGpe);
    double Alpha_test = atan2(YGpe,XGpe);
    double Alpha_PE = Alpha_test-phi;
    double Beta_PE = atan2(ZGpe,sqrt(XGpe*XGpe+YGpe*YGpe));
}

```

Appendix C (Continued)

```

DeltaX = Tt_MB.at(i+1)(0,3) - TGP(0,3);
DeltaY = Tt_MB.at(i+1)(1,3) - TGP(1,3);
DeltaZ = Tt_MB.at(i+1)(2,3) - TGP(2,3);
dx(0) = sqrt(DeltaX*DeltaX+DeltaY*DeltaY+ DeltaZ*DeltaZ);
dx(1) = atan2(DeltaY,DeltaX)-phi;
dx(2) = (D(i)-D_PE);
dx(3) = (Alpha(i)-Alpha_PE);
dx(4) = (Beta(i)-Beta_PE);
dx(5) = dx_6(3);
dx(6) = dx_6(4);
dx(7) = dx_6(5);

}

}
J0_T = J_BAXBOT;
J0_T.transposeInPlace();
double detJ_BAXBOT =sqrt( (J_BAXBOT*J0_T).determinant());
saveMM<<detJ_arm<< " <<detJAP_BAXBOT<< " <<detJ_BAXBOT<< " << "\n";
//      std::cout <<XGpe << std::endl;

BAXBOT_Opt_PriorityOrder(MM, Method, Priority, JLA, J_BAXBOT,J_PE_G, qn, dx, dHo,dq, dt
);
//cout<< "\n dt=: " <<dt<<endl;
//cout<< " dq=: " <<dq<<endl;
if(OnlyWC){
/*****/
// find the position error between the the previous commanded position
int Actual_encoder_R = Galilwheel.Reference_Position(B);
int Actual_encoder_L = Galilwheel.Reference_Position(A);

//cout<< " Actual_encoder_R=: " <<Actual_encoder_R<<endl;
//cout<< " Actual_encoder_L=: " <<Actual_encoder_L<<endl;

// Encoder Error
Error_encoder_L = Prv_encoder_L_Command - Actual_encoder_L ;
Error_encoder_R = Prv_encoder_R_Command - Actual_encoder_R ;

//cout<< " Error_encoder_L=: " <<Error_encoder_L<<endl;
//cout<< " Error_encoder_R=: " <<Error_encoder_R<<endl;

// Sending joint angle to controller as EncoderCounts
TranslationToEncoder(dq(7),dq(8), encoder_R, encoder_L);// get the encoder counts
for the translation and rotation of the wheelchair

//cout<< " encoder_R=: " <<encoder_R<<endl;
//cout<< " encoder_L=: " <<encoder_L<<endl;

Prv_encoder_R_Command = Prv_encoder_R_Command + encoder_R;
Prv_encoder_L_Command = Prv_encoder_L_Command + encoder_L;

//cout<< " Prv_encoder_R_Command=: " <<Prv_encoder_R_Command<<endl;
//cout<< " Prv_encoder_L_Command=: " <<Prv_encoder_L_Command<<endl;

int LeftWheel_speed = (int) ((*Error_encoder_L+*/KP_WC* encoder_L));
int RightWheel_speed = (int) ((*Error_encoder_R+*/KP_WC*encoder_R));

//int LeftWheel_speed = (int) (((/*Error_encoder_L+*/ encoder_L)/0.03) +
LeftWheel_speed_prv)/2.0);
//int RightWheel_speed = (int) (((/*Error_encoder_R+*/ encoder_R)/0.03)+

```


Appendix C (Continued)

```

RightWheel_speed_prv)/2.0);
//cout<<" LeftWheel_speed=: " <<LeftWheel_speed<<endl;
//cout<<" RightWheel_speed=: " <<RightWheel_speed<<endl;
int LeftLimit_Up = 20000;
int LeftLimit_Down = 6000;
int RightLimit_Up = 20000;
int RightLimit_Down = 6000;

if (LeftWheel_speed>LeftLimit_Up){ LeftWheel_speed = LeftLimit_Up;}
//if (LeftWheel_speed<LeftLimit_Down ) { LeftWheel_speed = LeftLimit_Down;}
if (RightWheel_speed>RightLimit_Up){ RightWheel_speed = RightLimit_Up;}
//if (RightWheel_speed<RightLimit_Down ) { RightWheel_speed = RightLimit_Down;}
//cout<<" LeftWheel_speed=: " <<LeftWheel_speed<<endl;
//cout<<" RightWheel_speed=: " <<RightWheel_speed<<endl;
LeftWheel_speed_prv = LeftWheel_speed;
RightWheel_speed_prv = RightWheel_speed;
// set the speed of the wheelchair
Galilwheel.SP(A,LeftWheel_speed);
Galilwheel.SP(B,RightWheel_speed);

dqQL(i) = LeftWheel_speed;
dqQR(i) = RightWheel_speed;
saveThrThl<<RightWheel_speed<<" " <<LeftWheel_speed<<"\n";

// Add the new commanded encoder count to the previous for the wheelchair
// Left wheel Motor A
//Galilwheel.IP(A, -5000/*encoder_L*/);
//Galilwheel.JG("A", LeftWheel_speed);

//int Sample = 1024;
//Galilwheel.PV("A",encoder_L,/*KP_WC*/ encoder_L,Sample);
//// Right wheel Motor B
////Galilwheel.IP(B, Test_Encoder(i)/*encoder_R*/);
////Galilwheel.JG("B", RightWheel_speed);
//Galilwheel.PV("B",encoder_R,/*KP_WC*/ encoder_R,Sample);
//cout<<" \n\n Sample =: " <<Sample<<" encoder_L =: " <<encoder_L<<" KP_WC*
encoder_R =: " <<KP_WC* encoder_R<<endl;
////Galilwheel.BG("AB");
//
//Galilwheel.BT("A");
//Galilwheel.BT("B");
//Sleep(10); // pause to execute the cotroller command

}

if(OnlyArm){

// Sending joint angle to controller as EncoderCounts
VectorXd dq_arm(7);
dq_arm= dq.head(7);
//cout<<" dq_arm=: " <<dq_arm<<endl;

WMRA_Rad2Encoder(dq_arm, AngleEncoder);
//cout<<" AngleEncoder=: \n\n" <<AngleEncoder<<endl;

// set the speed of the arm joints according to the commanded encoder count
KP_arm = 2;
for(int ii = 0; ii<7 ; ii++)
{
    JointSpeed(ii) = (int)(KP_arm * AngleEncoder(ii));
}
}

```

Appendix C (Continued)

```

//cout<< " jointSpeed=: \n\n"<<JointSpeed<<endl;
set_joint_speed(JointSpeed);

saveArmTheta<<JointSpeed(0)<<" "<<JointSpeed(1)<<" "<<JointSpeed(2)<<" "<<JointSpee
3)<<" "<<JointSpeed(4)<<" "<<JointSpeed(5)<<" "<<JointSpeed(6)<<"\n";
//cout<<" i=: " <<i<<endl;
/*Galilarm.IP("A", (int)AngleEncoder(0));
Galilarm.IP("B", (int)AngleEncoder(1));
Galilarm.IP("C", (int)AngleEncoder(2));
Galilarm.IP("D", (int)AngleEncoder(3));
Galilarm.IP("E", (int)AngleEncoder(4));
Galilarm.IP("F", (int)-AngleEncoder(5));
Galilarm.IP("G", (int)AngleEncoder(6));*/
//Sleep(100); // pause to execute the cotroller command

//cout<<" qarm_actual=: " <<qarm_actual<<endl;
//cout<<" dqlll=: " <<dq<<endl;
}
if (OnlyWC){
// Previous Encoder counts for Right and Left wheels
LeftEncoder_FB_prv = LeftEncoder_FB;
RightEncoder_FB_prv = RightEncoder_FB;

// Current Encoder counts for Right and Left wheels
LeftEncoder_FB = Galilwheel.Reference_Position(A);// Actual_encoder_L;
RightEncoder_FB = Galilwheel.Reference_Position(B);// Actual_encoder_R;

int Delta_Leftencoder = LeftEncoder_FB - LeftEncoder_FB_prv;
int Delta_Rightencoder = RightEncoder_FB -RightEncoder_FB_prv;

WheelchairPose_Increment(Delta_Rightencoder, Delta_Leftencoder , dq(7), dq(8) );
}
if(OnlyArm){
// Previous Encoder counts for The arm joints
Prv_Armjoint_encoder_Command = Prv_Armjoint_encoder_Command + AngleEncoder;

// Read the Arm Joints
ArmJointAngle(Actual_Armjoint_encoder);

VectorXi Delta_ArmJoint(7);
Delta_ArmJoint = Actual_Armjoint_encoder - Prv_Armjoint_encoder;
/*cout<<" Actual_Armjoint_encoder=: \n\n" <<Actual_Armjoint_encoder<<endl;
cout<<" Prv_Armjoint_encoder=: \n\n" <<Prv_Armjoint_encoder<<endl;*/
// cout<<" Delta_ArmJoint=: \n\n" <<Delta_ArmJoint<<endl;
Prv_Armjoint_encoder = Actual_Armjoint_encoder;
WMRA_Encoder2Rad(qarm_actual, Delta_ArmJoint);

for(int ii=0; ii<7; ii++){
    dq(ii) = qarm_actual(ii);
}
}

qn=qo+dq;
//cout<<" qn=: \n\n" <<qn<<endl;
VectorXd dq_MB = dq.tail(2);

```

Appendix C (Continued)

```
Matrix4d TGP_prv = TGP;

/*****
BATBOT_TransformationMatrices(2, qn, dq_MB, TGP_prv, TGE, TAE, TGP, T01, T12, T23, T34,
T45, T56, T67);

TGA = TGP*TPA;
qo=qn;

/*=====*/
/* Matlab simulation */

/*===== Passing Data to Matlab
=====*/

Map<Matrix<double,4,4,RowMajor> >(*C_T,4,4) = TGEi;
memcpy(mxGetPr(M_T), C_T, sizeof(C_T));
engPutVariable(m_pEngine, "TGEi", M_T);

Map<Matrix<double,4,4,RowMajor> >(*C_T,4,4) = TGP;
memcpy(mxGetPr(M_T), C_T, sizeof(C_T));
engPutVariable(m_pEngine, "TGP", M_T);

Map<Matrix<double,4,4,RowMajor> >(*C_T,4,4) = T01;
memcpy(mxGetPr(M_T), C_T, sizeof(C_T));
engPutVariable(m_pEngine, "T01", M_T);

Map<Matrix<double,4,4,RowMajor> >(*C_T,4,4) = T12;
memcpy(mxGetPr(M_T), C_T, sizeof(C_T));
engPutVariable(m_pEngine, "T12", M_T);

Map<Matrix<double,4,4,RowMajor> >(*C_T,4,4) = T23;
memcpy(mxGetPr(M_T), C_T, sizeof(C_T));
engPutVariable(m_pEngine, "T23", M_T);

Map<Matrix<double,4,4,RowMajor> >(*C_T,4,4) = T34;
memcpy(mxGetPr(M_T), C_T, sizeof(C_T));
engPutVariable(m_pEngine, "T34", M_T);

Map<Matrix<double,4,4,RowMajor> >(*C_T,4,4) = T45;
memcpy(mxGetPr(M_T), C_T, sizeof(C_T));
engPutVariable(m_pEngine, "T45", M_T);

Map<Matrix<double,4,4,RowMajor> >(*C_T,4,4) = T56;
memcpy(mxGetPr(M_T), C_T, sizeof(C_T));
engPutVariable(m_pEngine, "T56", M_T);

Map<Matrix<double,4,4,RowMajor> >(*C_T,4,4) = T67;
memcpy(mxGetPr(M_T), C_T, sizeof(C_T));
engPutVariable(m_pEngine, "T67", M_T);

Map<Matrix<double,4,4,RowMajor> >(*C_T,4,4) = Te_d;
memcpy(mxGetPr(M_T), C_T, sizeof(C_T));
engPutVariable(m_pEngine, "Td", M_T);

ii<<k;
ii_p=mxGetPr(ii_A);
ii_p[0] = ii[0];
engPutVariable(m_pEngine, "jj", ii_A);

for(unsigned i=0; i<Xe.size(); i++){
```

Appendix C (Continued)

```

        Xtemparray[i] = Xe[i];
        Ytemparray[i] = Ye[i];
        Ztemparray[i] = Ze[i];
        Xptemparray[i] = Xp[i];
        Yptemparray[i] = Yp[i];
    }

    engPutVariable(m_pEngine, "Xe", EE_XE);
    engPutVariable(m_pEngine, "Ye", EE_YE);
    engPutVariable(m_pEngine, "Ze", EE_ZE);
    engPutVariable(m_pEngine, "Xp", EE_Xp);
    engPutVariable(m_pEngine, "Yp", EE_Yp);
    // Printing the output of the matlab engine in C Command window
    /*      char matlab_out[10000];
    engOutputBuffer(m_pEngine , matlab_out, 10000);*/

    // Run the matlab simulation function
    engEvalString(m_pEngine, "WMRA_ML_CPP_Dual_Trajectory_Animation(2, TGEi, Td,
    TGP,Xe,Ye,Ze,Xp,Yp, T01, T12, T23, T34, T45, T56, T67,jj)");
    engEvalString(m_pEngine, "drawnow");
    // Printing the output.
    /*printf("%s", matlab_out);*/

    /*=====
    =====*/

    /*=====*/

    // Delay to comply with the required speed:
    toc = (clock())/(double)CLOCKS_PER_SEC;

    tt=tt+dt;
    //cout<<" TimeStep=: " <<tt-toc<<endl;
    if (toc < tt)
    {
        //Sleep(1000*(tt-toc));
        cout<<" TimeStep=: " <<tt-toc<<endl;
    }

    i++;
    k++;
}

Galilwheel.PV("A",0,0,0);
Galilwheel.PV("B",0,0,0);

Galilwheel.ST("AB");

Galilarm.ST("ABCDEFGH");
EE_Yp = mxCreateDoubleMatrix(1,dqqL.size(),mxREAL);
EE_Xp = mxCreateDoubleMatrix(1,dqqR.size(),mxREAL);
Ytemparray = mxGetPr(EE_Yp);
Xtemparray = mxGetPr(EE_Xp);
for(unsigned i=0; i<dqqL.size(); i++){
    Xtemparray[i] = dqqR[i];
    Ytemparray[i] = dqqL[i];
}
engPutVariable(m_pEngine, "dqqL", EE_Yp);
engPutVariable(m_pEngine, "dqqR", EE_Xp);
engEvalString(m_pEngine, "figure(1)");
engEvalString(m_pEngine, "plot(dqqL)");

```

Appendix C (Continued)

```
engEvalString(m_pEngine, "figure(2)");
engEvalString(m_pEngine, "plot(dqqR)");

Galilwheel.JGA(0);
Galilwheel.JGB(0);
mxDestroyArray(EE_XE);
mxDestroyArray(EE_YE);
mxDestroyArray(EE_ZE);
mxDestroyArray(EE_Xp);
mxDestroyArray(EE_Yp);
mxDestroyArray(M_T);
mxDestroyArray(ii_A);
////////// Close all the files//////////
saveTGE.close();
saveTGP.close();
saveMM.close();
E_Traj.close();
WC_Traj.close();

saveThrThl.close();
saveArmTheta.close();

//system("pause");
return 0;
}
```

Appendix C (Continued)

```
#ifndef APPLICATION_H //these statements define SLAM.h
#define APPLICATION_H

#include <fstream>
#include <iostream>
#include <stdio.h>
#include <vector>
#include <engine.h>
#include <time.h>
#include <cstdlib>
#include <Windows.h>

// #include <mrpt\base.h>
// #include <mrpt\gui.h>
// #include <mrpt\gui\CDisplayWindowPlots.h>
// #include <mrpt\obs.h>
// #include <mrpt\bayes/CKalmanFilterCapable.h>

#include <Eigen\stdvector>
// #include <unsupported\Eigen\MatrixFunctions>
// #include <unsupported\Eigen\src\MatrixFunctions\MatrixFunction.h>
// #include <unsupported\Eigen\src\MatrixFunctions\MatrixLogarithm.h>
// #include <unsupported\Eigen\src\MatrixFunctions\MatrixPower.h>
// #include <unsupported\Eigen\src\MatrixFunctions\MatrixExponential.h>
// #include <unsupported\Eigen\src\MatrixFunctions\MatrixFunctionAtomic.h>
// #include <unsupported\Eigen\src\MatrixFunctions\MatrixSquareRoot.h>
// #include <unsupported\Eigen\src\MatrixFunctions\StemFunction.h>
#include <Eigen\Dense>
#include <Eigen\lu>
#include <matrix.h>

using namespace Eigen; // ::MatrixXd;

//double pi = M_PI;
int Sign_Function(const double &x);
VectorXd BAXBOT_PBD();
Matrix4d BAXBOT_pose2T(double &x, double &y, double&a);
MatrixXd BAXBOT_DH(VectorXd &q);
Matrix4d BAXBOT_rotx(double &t);
Matrix4d BAXBOT_roty(double &t);
Matrix4d BAXBOT_rotz(double &t);
Matrix4d BAXBOT_transl(const double& x, const double& y, const double& z);
void BATBOT_TransformationMatrices(const int& kk, VectorXd &q, VectorXd &dq, Matrix4d &TGP_prv,
Matrix4d &TGE, Matrix4d &TAE, Matrix4d &TGP, Matrix4d &T1, Matrix4d &T2, Matrix4d &T3, Matrix4d
&T4, Matrix4d &T5, Matrix4d &T6, Matrix4d &T7);
void BAXBOT_w2T(Matrix4d &TGP_prv, VectorXd &q, Matrix4d &TGP_current);
void BAXBOT_D_Alpha_Beta(VectorXd &Xp, VectorXd &Yp, VectorXd &Xe, VectorXd &Ye, VectorXd
&Ze, VectorXd &D, VectorXd &Alpha, VectorXd &Beta, std::ofstream& DAlphaBeta );
void BAXBOT_MB_traj(Matrix4d &Ti, VectorXd &Xpb, VectorXd &Ypb,
std::vector<Eigen::Matrix4d, Eigen::aligned_allocator<Eigen::Matrix4d> > &Tt);
void BAXBOT_LinearTraj(const double &qi, double &qf, int &n, VectorXd &qt);
void BAXBOT_EE_Traj(Matrix4d &Ti, Matrix4d &Td, VectorXd &Xe, VectorXd &Ye, VectorXd &Ze,
std::vector<Eigen::Matrix4d, Eigen::aligned_allocator<Eigen::Matrix4d> > &Tt);
void BAXBOT_J07(VectorXd &q, MatrixXd &J0, double &detJ0);
void BAXBOT_JGA(double &p, const Vector2d &XY, MatrixXd &J);
void BAXBOT_delta(Matrix4d &Ti, Matrix4d &Td, VectorXd &delta);
void BAXBOT_D_Alpha_Beta_J(VectorXd &q, MatrixXd &J);
void BAXBOT_Jlimit(VectorXd &qmin, VectorXd &qmax);
void BAXBOT_WeightMatrix(VectorXd &q, VectorXd &dH);
void BAXBOT_Opt_PriorityOrder(int &imm, int &Method, int &Priority, int &JLA, MatrixXd
```

Appendix C (Continued)

```
&Whole_J,MatrixXd &J_arm, VectorXd &q, VectorXd &dx, VectorXd &dHo, VectorXd &dq, double &dt );
void Map_MatrixXd2Carray(Matrix4d &T,mxArray *M_T );
int TranslationToEncoder(double &X, double &phi, int &encoder_R, int &encoder_L);
void WheelchairPose_Increment(int &EncoderRight, int &EncoderLeft, double &XX, double &Pphi );
int joint_max_speed(int joint_name);
int joint_speed_limit(int joint_name, int speed);
void set_joint_speed_SP(VectorXd & JointSpeed);
int joint_limit_avoidance(int joint_name, int encoder_count, int speed);
void read_joint_position(double SendData[32],int joint_position[8]);
void WMRA_Rad2Encoder(VectorXd &Angle_rad, VectorXi &AngleEncoder);
void WMRA_Encoder2Rad(VectorXd &Angle_rad, VectorXi &AngleEncoder);
void WMRA_J07(Matrix4d &M1, Matrix4d &M2, Matrix4d &M3, Matrix4d &M4, Matrix4d &M5, Matrix4d
&M6, Matrix4d &M7, MatrixXd &J0, double &detJ0);
void WMRA_JDAlphaBeta(VectorXd &q, MatrixXd &J);
void WMRA_PDJ(int &ii, int &kk, VectorXd &q, VectorXd &PDJ_1);
void WMRA_PolyBlend(double &qi, double &qf, int &n, VectorXd &qt);

//int LoadProgramParameters();

#endif
```

Appendix C (Continued)

```

#include "application.h"

using namespace std;
//using namespace Eigen;
using namespace Eigen;
void BAXBOT_D_Alpha_Beta(VectorXd &Xp,VectorXd &Yp,VectorXd &Xe,VectorXd &Ye,VectorXd
&Ze,VectorXd &D,VectorXd &Alpha,VectorXd &Beta, std::ofstream& DAlphaBeta)
{
    VectorXd L=BAXBOT_PBD();
    VectorXd Zp=L(4)*VectorXd::Ones(Xp.size());

    // Calculating the desired D and Alpha
    for (int i = 0; i< Xe.size()-1;i=i+1){
        double XGpei = Xe(i)-Xp(i);
        double YGpei = Ye(i)-Yp(i);
        double ZGpei = Ze(i)-Zp(i);

        double Xphi = Xp(i+1)-Xp(i);
        double Yphi = Yp(i+1)-Yp(i);
        double phi_expected = atan2(Yphi,Xphi);

        D(i) = sqrt(XGpei*XGpei+YGpei*YGpei+ZGpei*ZGpei);
        Alpha(i) = atan2(YGpei,XGpei)-phi_expected;
        Beta(i) = atan2(ZGpei,sqrt(XGpei*XGpei+YGpei*YGpei));
        DAlphaBeta << D(i) << " , " << Alpha(i) <<" , " << Beta(i)<<"\n";
    }
    DAlphaBeta.close();
}

#include "application.h"
using namespace std;
//using namespace Eigen;
using namespace Eigen;
void BAXBOT_delta(Matrix4d &Ti,Matrix4d &Td, VectorXd &delta)
{
    delta = VectorXd::Zero(6);
    Vector3d Ti_1= Ti.block(0,0,3,1);
    Vector3d Ti_2= Ti.block(0,1,3,1);
    Vector3d Ti_3= Ti.block(0,2,3,1);
    Vector3d Td_1= Td.block(0,0,3,1);
    Vector3d Td_2= Td.block(0,1,3,1);
    Vector3d Td_3= Td.block(0,2,3,1);
    Vector3d ep=Td.block(0,3,3,1)-Ti.block(0,3,3,1);

    Vector3d eo=0.5*( Ti_1.cross(Td_1) + Ti_2.cross(Td_2) + Ti_3.cross(Td_3)); // From
equation 17 on page 189 of (Robot Motion Planning and Control) Book by Micheal Brady et al.
Taken from the paper (Resolved-Acceleration Control of Mechanical Manipulators) By John Y.
S. Luh et al.

    delta.head(3) = ep;
    delta.tail(3) = eo;
}

#include "application.h"
using namespace std;
//using namespace Eigen;
using namespace Eigen;

MatrixXd BAXBOT_DH(VectorXd &q)
{
    double pi = M_PI;
    MatrixXd DH(7,4);
    DH<< -pi/2, 0, 109.72, q(0) ,

```


Appendix C (Continued)

```
    pi/2, 0, 118.66, q(1) ,
    -pi/2, 0, 499.67, q(2) ,
    pi/2, 0, 121.78, q(3) ,
    -pi/2, 0, 235.67, q(4) ,
    pi/2, 0, 0, q(5) ,
    -pi/2, 0, 276.68, q(6);

// BaxBot DH
/*DH<<    0, 0, 270, q(0) ,
    -pi/2, 69, 0, q(1),
    pi/2, 0, 362, q(2) ,
    -pi/2, 69, 0, q(3) ,
    pi/2, 0, 375, q(4) ,
    -pi/2, 10, 0, q(5) ,
    pi/2, 0, 116, q(6);*/

return DH;
}

#include "application.h"
using namespace std;
//using namespace Eigen;
using namespace Eigen;
void BAXBOT_EE_Traj(Matrix4d &Ti, Matrix4d &Td, VectorXd &Xe, VectorXd &Ye, VectorXd &Ze,
std::vector<Eigen::Matrix4d, Eigen::aligned_allocator<Eigen::Matrix4d> > &Tt)
{
    double kx, ky, kz;
    double pi = M_PI;
    Matrix3d R(3,3), R_Ti(3,3), dR(3,3), R_Ti_transpose(3,3);
    VectorXd at;
    int n = Xe.size();
    //std::vector<Eigen::Matrix4d, Eigen::aligned_allocator<Eigen::Matrix4d> > Tt(n);

    // Finding the rotation of the desired point based on the initial point:
    R_Ti = Ti.block(0,0,3,3); //
    R_Ti.transposeInPlace();
    R = R_Ti * Td.block(0,0,3,3);

    // Initial single-angle representation of the rotation:
    double a =
    atan2(sqrt((R(2,1)-R(1,2))*(R(2,1)-R(1,2))+(R(0,2)-R(2,0))*(R(0,2)-R(2,0))+(R(1,0)-R(0,1))*(R
    (1,0)-R(0,1))), (R(0,0)+R(1,1)+R(2,2)-1));
    double s=sin(a);
    double c=cos(a);
    double v=1-c;

    // Finding the single-vector components for the rotation:
    if( a<0.001){
        kx=1;
        ky=0;
        kz=0;
    }
    else if (a<pi/2+0.001){
        kx=(R(2,1)-R(1,2))/(2*s);
        ky=(R(0,2)-R(2,0))/(2*s);
        kz=(R(1,0)-R(0,1))/(2*s);
    }
    else{
        kx=Sign_Function(R(2,1)-R(1,2))*sqrt((R(0,0)-c)/v);

```

Appendix C (Continued)

```

ky=Sign_Function(R(0,2)-R(2,0))*sqrt((R(1,1)-c)/v);
kz=Sign_Function(R(1,0)-R(0,1))*sqrt((R(2,2)-c)/v);
if (kx>ky && kx>kz){
    ky=(R(1,0)+R(0,1))/(2*kx*v);
    kz=(R(0,2)+R(2,0))/(2*kx*v);}
else if (ky>kx && ky>kz){
    kx=(R(1,0)+R(0,1))/(2*ky*v);
    kz=(R(2,1)+R(1,2))/(2*ky*v);}
else{
    kx=(R(0,2)+R(2,0))/(2*kz*v);
    ky=(R(2,1)+R(1,2))/(2*kz*v);}
}
// Running the desired trajectory method: Linear function.
BAXBOT_LinearTraj(0.0,a,n, at);
std::vector<Eigen::Matrix4d,Eigen::aligned_allocator<Eigen::Matrix4d> > Tt_i(n);
Tt_i.at(0)=Ti;

for (int i=1; i<n; i++){
    // Single-angle Change:
    double da=at(i)-at(0);

    s=sin(da);
    c=cos(da);
    v=1-c;

    // Rotation and Position Change:
    dR << kx*kx*v+c    , kx*ky*v-kz*s, kx*kz*v+ky*s,
          kx*ky*v+kz*s, ky*ky*v+c    , ky*kz*v-kx*s,
          kx*kz*v-ky*s, ky*kz*v+kx*s, kz*kz*v+c;

    // Finding the trajectory points along the trajectory line:
    Tt_i.at(i)=Matrix4d::Identity(4,4);
    Tt_i.at(i).block(0,0,3,3)=Ti.block(0,0,3,3)*dR;
    Tt_i.at(i).block(0,3,3,1) << Xe(i),Ye(i),Ze(i);
}
Tt=Tt_i;
}

#include "application.h"
using namespace std;
//using namespace Eigen;
using namespace Eigen;

void BAXBOT_Jlimit(VectorXd &qmin,VectorXd &qmax)
{
    double pi = M_PI;
    qmin = VectorXd::Zero(7);
    qmax = VectorXd::Zero(7);
    // Inputting the joint limits in a vector form, dimensions are in radians:
    // Dimentions based on the actual physical arm:
    qmin<< -170,
          -170,
          -170,
          -170,
          -170,
          -100,
          -200;
    qmin = qmin*pi/180;
    qmax << 170,
          170,
          170,

```

Appendix C (Continued)

```
        170,  
        170,  
        100,  
        200;  
qmax = qmax*pi/180;  
}
```