Design and Control of a Personal Assistant Robot

Yang Qian

N° d'ordre : 219

ECOLE CENTRALE DE LILLE

THESE

Présentée en vue
d'obtenir le grade de

DOCTEUR

En
Automatique, Génie Informatique, Traitement du Signal et Image

Par

QIAN Yang

DOCTORAT DELIVRE PAR L'ECOLE CENTRALE DE LILLE

Titre de la thèse :

Conception et Commande d'un Robot d'Assistance à la Personne

Design and Control of a Personal Assistant Robot

Soutenue le 04 Juillet 2013 devant le jury d'examen :

Président	Saïd MAMMAR	Professeur, Université d'Evry Val d'Essonne, Evry
Rapporteur	Nacer K.M'SIRDI	Professeur, Polytech Marseille, Marseille
Rapporteur	Daniel SIDOBRE	Maître de Conférences, HDR, Université Paul Sabatier, Toulouse
Rapporteur	Jianming YANG	Professeur, Meijo University, Japon
Membre	Belkacem Ould Boumama	Professeur, Polytech Lille, Lille
Membre	Qiang ZHAN	Professeur, Beihang University, Chine
Directeur de thèse	Ahmed RAHMANI	Maître de Conférences, HDR, Ecole Centrale de Lille, Lille

Thèse préparée dans le Laboratoire d'Automatique, Génie Informatique et Signal
L.A.G.I.S. - CNRS UMR 8219 - École Centrale de Lille
Ecole Doctorale Sciences pour l'ingénieur ED 072
PRES Université Lille Nord-de-France

À mes parents,

à toute ma famille,

à mes professeurs,

et à mes chèr(e)s ami(e)s.

Acknowledgements

This work was carried out at “Laboratoire d’Automatique, Génie Informatique et Signal” (LAGIS) in Ecole Centrale de Lille with the research team “Méthodes & Outils pour la Conception Intégrée de Système” (MOCIS).

I would like to express my gratitude to the people who have helped me in the successful completion of my doctoral study. Without their encouragement and help, I would not have arrived at this position and stage of my life.

First of all, I would like to express my sincere gratitude to my PhD thesis supervisor Professor Ahmed RAHMANI who influenced me most during my years at Ecole Central de Lille. Professor Ahmed taught me how to look for new areas of research, how to understand the state of the art quickly, how to write good technical papers, and how to present my ideas effectively. I would also like to thank Professor Qiang ZHAN from Beihang University. He not only provided help with robot design, but also made many design related discussions with me during his visit to France.

I would like to thank my reading committee members: Mr. Nacer K.M’SIRDI, Mr. Daniel SIDOBRE, Mr. Jianming YANG, Mr. Belkacem Ould Boumama and Mr. Saïd MAMMAR for their time, interest, and helpful comments.

I wish to thank all staff at the team, Mrs. Geneviève Dauphin-Tanguy, Mr. Christophe Sueur…, for their valuable discussions and insightful suggestions in team meetings.

A particular acknowledgement has to be made to all my PhD colleagues. I wish to thank Zhaoxia Peng, Guoguang Wen, Dapeng Yang, Jin Bai, Youwei Dong, Eugene Tarasov, Zhi Li, Yue Yu, Tian Zheng, Yongliang Huang, Baisi Liu, etc. And I also wish to thank the staff at Ecole, Chistine Yvoz, Brigitte Foncez, Patrick Gallais, Bernard Szukala and Vanessa Fleury for their kindly help during the past three years.

Pursuing a PhD requires not only technical skill but also tremendous amount of stamina and courage. I would like to thank my parents for sharing their unconditional love with me and giving me the necessary amount of courage required for pursuing my goals at Ecole.

Finally, I wish to thank the China Scholarship Council (CSC), which gave me the opportunity to pursue my PhD study in Ecole Centrale de Lille.

Table of Contents

List of Tables

List of Figures

Abbreviations

2D-Two Dimension

3D-Three Dimension

ADAMS-Automatic Dynamic Analysis of Mechanical Systems

CAD-Computer Aided Design

CAN-Controller Area Network

CFD-Computational Fluid Dynamics

CPLD-Complex Programmable Logic Device

DOF-Degree of Freedom

DSP- Digital Signal Processing

EJS-Eulerian Junction Structure

FEA-Finite Element Analysis

GFO-Grasping Force Optimization

GSTIFF -Gear Stiff Integrator

GUI-Graphical User Interface

IK-Inverse Kinematics

IKBiRRT-Inverse Kinematics-Based Bdirectional RRT

LMI-Linear Matrix Inequalities

MBS-Multi-Body Dynamic Simulation

MDH-Modified D-H

NN-Neural Network

PD-Proportional Derivative

PRM-Probabilistic Roadmap Method

RBF-Radial Basis Function

RBFNN-Radial Basis Function Neural Network

RRT-Rapidly-Exploring Random Tree

Introduction

Over the past decades, there has been increasingly interest in the challenging field of service robots. Most of the elderly and disabled people want to live in their own houses and hope that some intelligent robots can help them with tasks such as stand up or seat down assistance, preparing or warming food, serving and clearing table, fetching and carrying difficult and heavy objects, etc. Our project “Intelligent Control of Autonomous Manipulator Mobile Robots Group Formation” is associated with the LIA2MCIS (Associated International Laboratory for Material, Mechanics, Control and Information Science). By collaborating with Beihang University in this project, we focus on the design, research and virtual prototyping of a personal assitant robot which will work in domestic environments for assistance tasks.

Problem Statement

Research on personal assistant robot is still in its very early stages. There are many problems in creating such a complex robotic system. First, primary applications should be designed. The requirements of hardware and software which support those applications are needed to be considered. Secondly, structure design plays a very important role in defining the robot efficiency and flexibility. Ideally designed and optimized structure is able to enhance robot operating efficiency. Most of the assistant robots are based on a robot platform. This platform is usually equipped with a wheeled drive system and some castor wheels. And the assistant robots are all equipped with one or two compact robotic arms. This type of robot is called a mobile manipulator. Such systems combine the advantages of mobile platform and robotic arms and reduce their drawbacks. For instance, the mobile platform extends the workspace, while the arms offer many operational functions (e.g. simply opening a door for the robot). This structure seems particularly suited for personal assistant robots. In general, mobile manipulators can be categorized into three classes: (a) mobile manipulator with one arm; (b) mobile manipulator with dual arms; and (c) multiple mobile manipulators. The studies of the second one have become important in designing personal assistant robots for their human-like structures. However, as is always the case, there is a cost to pay for the advantages: more difficulties in system modeling, control and synthesis.

Simulation is an important and useful technique that can help users to better model and understand real life systems. Virtual prototyping is now a very common technique to simulate

mechanical systems. In many cases, virtual prototyping using computer has been able to significantly reduce the amount of physical testing required, the time to market and hence the cost. Once built, the models can be run to give realistic results. Using software tools, one can design and model systems by simulating separate parts of these systems and visually investigating their behaviors under conditions that are close to real situations. In this thesis, we illustrate the application of virtual prototyping to the design and development of the assistant robot. Kinematic and dynamic simulations of the complete robotic system are performed. With the virtual prototype we are able to validate the functionality of the robot in the conceptual design stage, improve the design based on the feedback from the simulation results to yield an optimized final design and demonstrate the robot's capability.

Modeling of robotic systems is usually treated as a preprocessing. It is clear that it provides a basis on the study of stability analysis, feedback control synthesis, computer simulation, implementation, and so on. The more arms mounted on the mobile platform, the more laborious and complicated procedures required in the complete modeling and the controller synthesis. Especially, the robots have dexterous hands. One of the objectives of this thesis is to completely set up the accurate models of kinematics and dynamics for the mobile manipulator system which is composed of a mobile platform, a torso, and a bimanual robotic system equipped with two arms and two dexterous hands.

In recent years, there are a number of researchers studying motion planning and control for mobile manipulators. The objective of this thesis is also focused on these two points. Given a task, we should first determine the motion in joint space for the robot. Motion planning in joint space is usually proposed by using optimization techniques to determine the joint motions so that the end-effector of a manipulator reaches some desired goal states in task space by general point-to-point motion. There exists an intrinsic problem for most mobile manipulator systems: the total number of DOFs is generally greater than six DOFs. How to deal with the redundant DOFs attracts much attention in the robotics community recently.

Motion control of mobile manipulators addresses the problem of trajectory tracking in joint space which allowing the joints to follow a desired trajectory specified as joint angles under uncertainties and disturbances. However, manipulators and mobile platforms not only have different dynamic characteristics but also have strong coupling. Since, in most cases, the wheeled mobile platforms move on the ground subjected to nonholonomic constraints and the

robotic manipulators are holonomic mechanical systems, how to control the manipulators and the mobile platforms is also important.

Due to the high mobility of wheeled platforms and the manipulability of manipulators, wheeled mobile manipulators are capable of performing dexterous manipulation tasks using some control laws. Furthermore, the study on control of more powerful robots with dexterous hands has attracted significant attention over the years. Although researches on mobile robots, robotic arms, manipulation by multi-fingered hands and even on mobile manipulators have been studied in various research institutes and universities, there are few works related to a redundant system considering mobile platform, arms and hands as a single system (as ARMAR-III [1] and Meka [2]). Most researches on robots have been developed in their own independent and separated lines without considering any coordination of each other. Object manipulation with a multi-fingered mobile manipulator is a challenging task, especially in personal assistant robot applications.

The work presented aims at designing a personal assistant robot using virtual prototyping technology, providing a unified model for this multi-fingered mobile manipulator and designing some control laws to achieve coordinated motion and object manipulation. The main objectives of the research are the following:

- Robot application design and requirement discussion.
- Mechanical structure design and virtual prototype development.
- Mathematical and bond graph modeling of a dual-arm/hand mobile manipulator.
- Development of a collision-free path planning algorithm for a redundant manipulator with the given desired pose.
- Development of a robust RBF network-based control scheme for a mobile manipulator that identifies the completely unknown mobile platform and manipulator dynamics in unknown and changing environments with the adaptation ability.
- Development of a robust RBF network-based control scheme for coordinated force/motion control of a mobile manipulator suffering both holonomic and nonholonomic constraints in the presence of uncertainties and disturbances.
- Design of a control scheme to ensure firm grip, avoid slippage and well track a given motion imposed to the object considering the dynamics of the dual-arm/hand mobile manipulator which manipulates an object with known shape by rolling contacts.

Organization

The thesis is divided into four parts. Chapter 1 lists some personal assistant robots, and then introduces the previous work related to the research topics. A short introduction to the technical background employed in this thesis is also offered in Chapter 1.

In Chapter 2, first, the robot applications are discussed. This robot's ultimate goal is the execution of complex manipulation tasks in domestic environments. And the requirements for realizing those applications are given. In the mechanical design stage, a 56 DOFs robot with repeated several robotic chains for the trunk, arms and dexterous hands is constructed. Its 3D mechanical structure designed in Solidworks is exported to ADAMS software in order to perform motion simulations. The virtual prototype in ADAMS provides a tool to explore many issues involved in robotics.

For control creation in mobile manipulation tasks, in Chapter 3, the kinematic and dynamic models of the robot are established. The kinematic model of the robot based on the MDH method is developed, and the dynamic model is derived by the standard Euler-Lagrange formulation. As the mathematical model is frequently used for several different purposes there is a need for finding ways of facilitating connection of submodels to extend these models. In Chapter 3, a novel approach, bond graph technique, to model the robot is introduced. It is a modeling tool which can be used to describe several different disciplines or energy domains using the same basic system elements. Thus electrical systems affecting mechanical systems can easily be modeled and connected. The overall bond graph model of the robot is build step by step following the Newton-Euler formalism which has been widely used for modeling this kind of system.

In Chapter 4, some algorithms for motion planning and coordinated control are presented. First, an algorithm for solving the motion planning problem is proposed. Then a robust NN-based control scheme for a mobile manipulator is presented. This scheme can identify the completely unknown dynamics of the robot in unknown and changing environments with the adaptation ability. A similar control scheme but for coordinated motion and force control of a mobile manipulator suffering both holonomic and nonholonomic constraints is proposed. In addition Chapter 4 illustrates a control scheme for multi-finger grasp synthesis to ensure firm grip, avoid slippage and well track a given motion imposed to the object. This control method is not only applicable to the dexterous hand but also to the complete mobile manipulator. In the latter case, we take into account the complete kinematics and dynamics of the dual-arm/hand mobile manipulator system.

CHAPTER1 STATE OF THE ART AND PRELIMINARIES

1.1 State of the Art in Personal Assistant Robotics

Personal assistant robots are semi or fully autonomous machines that can mimic humans with human-like motions. They can assist human beings, typically by performing a job that is dirty, dull, distant, dangerous or repetitive, including household chores such as cutting the lawn, cleaning, housekeeping, and more. Meanwhile, the development of robots for health care and assistance to elderly, disabled, or impaired people is an active area of research. Today, there are some robots that also can sing and dance. There are some personal assistant robots in production, and some examples are shown in Figure 1.1.

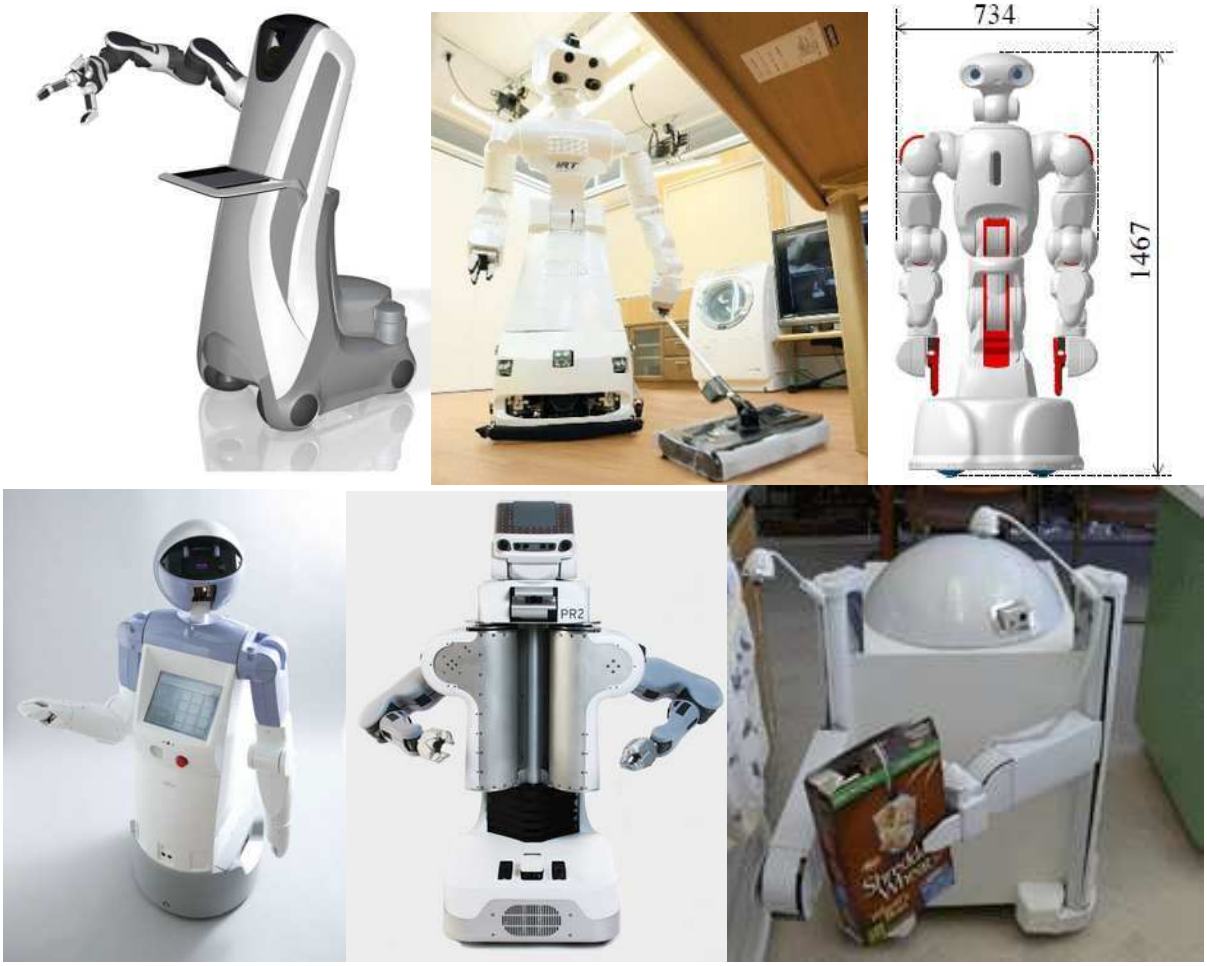


Figure 1.1: Examples of personal assistant robots: Care-O-bot 3 (Fraunhofer Institute for Manufacturing Engineering and Automation IPA); Robot AR (Maid Robot from Japan); TWENDY-ONE (WASEDA University Sugano Laboratory TWENDY team); Enon (Fujitsu Laboratories Ltd); Personal Robot PR2 (Willow Garage); Readybot (Silicon Valley)

- Care-O-bot 3

Stereo-vision color cameras, laser scanners and a 3D range camera enable Care-O-bot 3 to register its surroundings in 3D in real time. It can move in any direction which is made possible by an omnidirectional platform with four separately steered and driven wheels. The new Care-O-bot has a highly flexible arm with seven DOFs and a hand with three fingers. A tray is mounted at the front of the robot, on which Care-O-Bot can carry items such as the requested cup of coffee. Integrated in the tray is a touch screen via which the assistant can be controlled. But the robot can also be directed by spoken commands. Unlike its predecessors, it can even recognize and respond to gestures. Numerous household articles are stored in the robot's databases. It can also learn to recognize new objects.

- Robot AR

The upper body of AR consists of arms (7 DOFs), a head (3 DOFs), and a waist (1 DOF). End-effector equips 3 fingers and each finger is composed of 2 joints. On the other hand, lower body is wheeled mobile platform. This robot mounts a stereo camera on the head, and a LRF (Laser Range Finder) on the base. Force sensors are equipped on the wrist and the shoulder of the both arms. The design focus on following 3 series of tasks which are very popular in daily life: (1) pick up a tray from a table, convey it, and put it onto a kitchen, (2) gather clothes and put it in a washer, and (3) sweep a floor by using a broom. One of the characteristics of this robot is to have failure detection and recovery abilities.

The robot has two types of recognition functions which are applied to solid objects (other than clothes) or soft objects (clothes): (1) object poses recognition based on geometrical model and (2) cloth recognition based on wrinkle feature.

- TWENDY-ONE

TWENDY-ONE is a sophisticated human symbiotic robot which equips the functions such as safety, dependability and dexterity. The special feature of TWENDY-ONE is the combination function of dexterity with passivity and high-power output. TWENDY-ONE equips high output actuators with the simple mechanical passive impedance mechanism. When TWENDY-ONE manipulates an object with various shapes, it is easy for TWENDY-ONE to adapt to the object by passivity to absorb external force generated by the positioning deviation. In the same way, TWENDY-ONE can adapt to human motion and hold a human.

As a result, TWENDY-ONE can manipulate an object dexterously as well as support a human.

- Enon

Enon is an acronym of the phrase “exciting nova on network”. The phrase conveys the robot’s ability to autonomously support customers’ tasks while being linked to a network.

As a fully developed practical-use service robot, Enon features enhancements such as lighter weight, smaller size, and more safety features than the prototype that Fujitsu developed before. Enon is an advanced service robot capable of accomplishing multiple tasks such as providing guidance, escorting people, transporting objects, and security patrolling, thereby differentiating Enon from other service robots on the market that are designed specifically for only a single task such as transporting, cleaning, or surveillance.

- PR2

PR2 is an open platform. Researchers can change the system to meet their needs at any level. It combines the mobility to navigate human environments and the dexterity to grasp and manipulate objects in those environments. It is roughly the size of a person and consists of a mobile base sitting upon four steered and driven casters, torso, head, two arms, an array of sensors and sixteen cores for computation. The torso moves up and down relative to the base, to give PR2 a height ranging from 1.33m to 1.645m. The arms attached to the torso have seven DOFs and the forearm contains a 640x480 color camera that can always see the grippers. The grippers consist of a central palm with four articulating fingers. Each gripper also contains a 3 DOFs accelerometer and an LED, which can be turned on and off to assist in calibration or finding the location of the gripper in camera images. The pan-tilt head contains wide and narrow stereo cameras, a 5-megapixel camera, a texture projector and a Hokuyo laser range-finder.

- Readybot

The Readybot prototype is basically a white enamel box with wheels, two human-sized arms and retro chrome styling. The unit fits neatly in the same counter space as a dishwasher and after activation, rolls out, deploys several antenna-like cameras, and raises itself up to human height to begin work. Slowly but steadily, it picks up cups, bowls, and plates, dumps food,

loads the dishwasher, scrapes and scrubs the countertop. When needed, it grabs one of several custom tools to scrub, sponge, or maneuver.

1.2 Literature Survey

In this subchapter, a survey of the previous work related to the research topics mentioned in the introduction part is provided.

1.2.1 Review of Multi-body Computer Codes

CAD packages have increased in popularity to support the design process. Computer analysis packages provide numerous advantages over traditional analysis methods such as analytical hand calculations and prototype /experiment design schemes. Computer analysis techniques such as MBS, FEA, and CFD are the sister products of the CAD revolution.

Table 1.1: A comparison of multi-body dynamic codes

Program	Inverse dynamics	Static	Elastic analysis	Closed-loop analysis	Symbolic equation	Relative coordinates	Joint constraint	Constraint forces	Active elements	User elements	Track model	Tire model	Contact geometry	CAD interface	FEM interface	PC version	Documentation
3D-MCADA	√	√		√			√	√		√		√		√		√	
ADAMS	√	√	√	√		√	√	√	√	√		√	√	√	√	√	√
A'GEM		√		√		√	√	√		√	√	√	√	√		√	√
ALASKA	√	√		√		√	√	√		√						√	√
AUTODYN/ ROBOTRAN	√	√		√	√	√	√	√		√	√		√			√	
AUTOSIM	√	√		√	√	√	√	√		√						√	√
CMSP	√	√		√		√	√	√		√						√	
COMPAMM	√			√		√	√	√		√				√			√
DADS	√	√	√	√		√	√	√	√	√	√	√		√	√	√	√
LMS	√			√		√	√	√		√				√		√	
MADYMO	√		√			√	√	√		√				√		√	
MECANO	√	√	√	√			√	√	√	√		√			√	√	√
MEDYNA		√	√	√			√	√	√	√	√		√		√	√	√
NEWEUL	√	√		√	√	√	√		√	√				√		√	√
NUCARS		√				√	√				√		√			√	√
NUSTAR		√	√	√		√	√	√		√		√		√		√	√
SIMPACK	√	√	√	√		√	√	√	√	√		√				√	√
WORKING MODEL		√		√		√	√	√	√	√				√	√	√	√

The goal of this research is to develop a virtual prototype for a mobile manipulator which is a complex, highly nonlinear system and requires the capabilities of advanced packages to be fully analyzed. Among the various computer analysis packages, MBS packages are capable of

creating valid dynamic simulations for robots. Once the virtual prototype has been developed and validated, a real complex mobile manipulator will be created. Table 1.1 [3] outlines the various MBS packages on the market and their capabilities. By comparison, among these packages, ADAMS and DADS are the most comprehensive MBS packages available.

1.2.2 Robot Structure

In the analysis of all assistant robots, different characterizations can be argued such as biped locomotion, dexterous hand manipulation, dual-arm manipulation, and so on. It is possible to classify the assistant robots developed so far in two main groups according to the tasks that are addressed. The first one includes mobile robotic platforms (as Sony QRIO [4] and Honda ASIMO [5]). All of these robots have brought about significant advances in locomotion, nevertheless they cannot be considered as effective human assistants because they are unable to perform complex manipulation tasks. The latter group is bimanual robots consist of a torso, two arms and dexterous hands. These robots consist of incomplete platforms but provided with dexterous hands. Examples from this category include Dexter (UMass Amherst) [6] and Domo (MIT) [7]. Manipulation is an addressed feature and although they seem not complete, they are fundamental experimental tools in developing artificial intelligences with cognitive abilities. Ideally, the advances in bimanual manipulation and locomotion could be combined to provide a competent experimental platform for autonomous mobile manipulation.

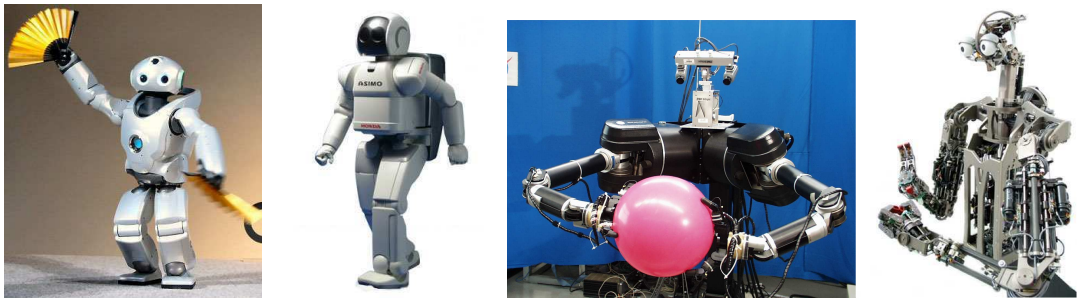


Figure 1.2: Asimo (Honda); Qrio (Sony); Dexter (UMass Amherst); Domo (MIT)

1.2.3 System Modeling

Modeling is an essential analysis for understanding the dynamical behavior of robot systems. An appropriately constructed dynamic model can greatly facilitate and guide the control design. A mobile manipulator includes dynamic interaction between the mobile base and the manipulators. And usually the wheeled platform is subjected to the well-known nonholonomic constraints. Thus the dynamics of the combined system is much more complicated. Mathematically, the dynamic model is naturally expressed by the equations of motion, which define an instantaneous relationship between the external actions (actuator

forces, disturbances), the generalized coordinates and their time-derivatives (velocities and accelerations). Generally speaking, the dynamics of mobile manipulators has been derived based on four approaches: Newton-Euler method, Lagrange's equation, forward recursive formulation, and Kane's approach. Some literatures have concentrated on the kinematic models excluding the dynamic effects [8]. In reference [9] the model is based on Lagrange's approach and in reference [10] the model is based on Kane's equation. These two papers have paid attention to derive the general form of dynamic equations and dynamic couplings between the mobile base and the manipulators, but the computation complexities are large and the detailed descriptions are omitted. Sheng [11] utilizes a neural-fuzzy approach to approximate the robot dynamics, which leads to low speed moving or long-time learning.

Mathematical models are frequently composed of thousands of equations. These models are difficult to create and even more difficult to maintain. Graphical modeling is generally more suitable for the creation of models of complex systems than equation-based modeling. Graphical approach consists of the following methods: block diagram, signal flow graph, and bond graph from which one can get the mathematical models of systems. Among these methods, block diagram and signal flow graph do not give much insight into the system topology while bond graph represents a generally usable approach to model physical systems of arbitrary types. The concepts of energy and power flows define a suitable semantic framework for bond graphs of all physical systems. Many researchers have worked on bond graph modeling of mechanical systems. Bond graph theory and notation were well developed and described in [12]-[14]. Bos [15] and Fahrenthold [16] developed bond graph techniques for hydraulic, mechatronic and thermodynamic systems. Karnopp [17] studied a comprehensive research on bond graph techniques whose results were applicable to robotics and spatially moving objects.

We give a brief summary of mathematical and bond graph modeling methods.

Classical approach for modeling physical system [18]

- Step1: Develop an engineering model
- Step2: Write differential equations
- Step3: Determine solution
- Step4: Write program

Bond graph modeling approach [18]

- Step1: Develop a schematic model
- Step2: Draw a bond graph
- Step3: Obtain computer graph to generated differential equations
- Step4: Use PASCAL or Matlab

1.2.4 Motion Planning of Mobile Manipulator

Motion planning for mobile manipulators is to find a proper collision-free path under certain evaluation criteria from the start state (including position and direction) to the target state in the work environment. We have known that the alliance of mobile platform and manipulators has an advantage because it permits to enlarge the workspace of the robotic arms, but it causes inconvenience in path planning; combining manipulators with mobile platform results in a system with redundant kinematics. And it is a persistent problem when the end-effectors' states of the manipulators are imposed.

The motion of a mobile manipulator regarding one target object can be considered as consisting of two cases: (1) moving while carrying the target and placing the target at a destination, and (2) moving to pick up the target. In the first case some planning methods were proposed. Some of them did not give full play to the abilities of locomotion and manipulation and some did not consider the dynamic effects. Yamamoto [19] presented a planning and control algorithm for the platform so that the manipulator was always positioned at the preferred operation region measured by its manipulability. Seraji [20] presented a simple and computationally efficient approach which allowed the user to assign weighting factors to individual degrees of mobility and degrees of manipulation, as well as to each task specification. Carriker [21] presented a heuristic method for searching a tree of starting points for a standard numerical algorithm to find a global minimum for the cost function. Foulon [22] reported strategies for realizing operational point to point tasks with a nonholonomic mobile manipulator. But it neglected the dynamic properties. Tchon [23] concentrated on local methods, and presented three Jacobian motion planning algorithms effective at regular configurations of a mobile manipulator. Tanner [24] presented an approach which was based on a discontinuous feedback law under the influence of a special potential field.

Mobile manipulator is a very complex multiple-input multiple-output nonlinear system and has time-varying characteristics and strong coupling. Therefore, one must take into account the dynamic interaction between the mobile base and the manipulators. Akira [25] derived the dynamics of a mobile manipulator considering it as a combined system and presented a trajectory planning method using the concept of the order of priority. Tan [26] developed a unified dynamic model for an integrated mobile platform and on-board manipulator. The kinematic redundancy and dynamic properties of the platform and the robotic arm have been considered in the unified model. Huang [27] proposed a method for vehicle motion planning considering manipulator task constraints, and manipulator motion planning considering platform stability. Haddad [28] proposed a flexible stochastic scheme for point-to-point trajectory planning of nonholonomic mobile manipulators subjected to move in a structured workspace.

In the second case, when the mobile base arrives at a grasping position, the manipulator begins to grasp the target. When the mobile platform is nonholonomic and moves in a collision-free environment, most of the planning methods which are applied in nonholonomic systems [29], [30] can be used. The problem is known to be complex if obstacles are present. The path planning problems of mobile robot and redundant multi-joint manipulator in dynamic environment with obstacles are two very important issues for achieving this task. Researchers proposed a variety of planning algorithms, such as PRM [31], [32], RRT [33], [34], artificial potential field method [35], [36], cell decomposition [37] and so on. Artificial potential field algorithm is simple and easy to implement. It is often applied to local path planning. Cell decomposition method is mainly used for low-dimensional path planning. The search algorithms based on sampling techniques (PRM, RRT) can solve high-dimensional search problem. PRM algorithm is mainly used for the search problem in static environment. It searches for a feasible path through a probabilistic roadmap which is pre-built. In configuration space RRT randomly samples some points to expand the tree until this search tree is connected to the goal point. Because of its ability to solve very complex, high dimensional planning problems and its relative ease of implementation, RRT has been used for high-dimensional manipulation planning which is our current focus.

1.2.5 Coordinated Control of Mobile Manipulator

In recent years, many researchers have investigated trajectory tracking control problem for mobile manipulators. The control schemes can be mainly classified into two categories. One

is decentralized control law, in which controller for mobile platform and that for manipulators are constructed separately and then interaction between platform and manipulators is considered or neglected. The other one is unified control law, in which controller is designed for overall mobile manipulator system. The following are some issues considered in the various control schemes: (1) dynamic coupling, (2) uncertainties and disturbances, (3) stability, (4) moving form of platform and manipulators, and (5) control of applied end-effector forces. We present a short review of the available current control methods.

Wiens [38] presented a single lever arm links with a fixed platform and discussed the influence of dynamics of the platform. But it did not consider the control of the platform [39]. Yamamoto [19] presumed that the manipulator had been well taught, and through controlling the platform the system can make the manipulator be in the preferred region and keep the best configuration. Based on [19] and combining force control, Yamamoto [40] proposed that a manipulator tracked the surface of a moving object. Lee [41] chose a safety region for the platform. Computed torque control [42] is very effective, but it requires the precise mathematical model of the system. However, the mobile manipulator is a very complicated nonlinear system. It is difficult to obtain the accurate dynamic model.

Considering uncertainties such as friction, disturbances and many other factors in practice, many researchers did much work. Through torque control, Evangelos [43] accomplished the tracking control of a manipulator under disturbances. Liu [44] proposed a decentralized control method. The dynamic coupling and the uncertainties were all regarded as the disturbances. On the premise that the system did not need know the dynamics and the uncertainties, Sheng [45] presented a robust damping control algorithm and respectively designed a neutral network controller for the two subsystems [11]. The neutral network estimated the dynamic coupling, parameters, and uncertainties online. Chung [46] solved the kinematic redundancy by using a nonlinear interaction control algorithm. A robust adaptive controller was designed for the manipulator and an I/O linear controller was designed for the platform. The coordination of the two controllers was achieved.

In the workspace, there are always many obstacles. Yamamoto [47] presumed that there were some obstacles in the desired trajectory. Super-quadratic potential functions instead of those obstacles in order to avoid them. Ogren [48] assumed that there were some obstacles in the trajectory of the mobile platform. According to the distances between the obstacles, the platform and the end-effector, a variable structure controller was designed.

The decentralized control methods described above are relatively simple. However because of the coupling between the subsystems, the performance of the controllers will be severely affected. Thus many scholars considered mobile platform and manipulators as a whole system and studied on coordinated control. Seraji [20] established a unified kinematic model for mobile manipulators and proposed configuration control. It used the redundancy to complete a series of user-defined additional tasks. Tan [26], [49] established a unified dynamic model for mobile manipulators. The model was linearized and decoupled and it completed the coordination of mobile manipulators through event-based control. The kinematic redundancy was used to control the force and position simultaneously in order to ensure the stability when there were some obstacles [26]. Besides, force and position control were decoupled and kept in the same direction by using kinematic redundancy [49].

In general, the task is just given to the end-effector, and the platform only plays the role of positioning. But in some special occasions, it requires not only that the end-effector moves along the desired trajectory but also that the platform moves following a given trajectory. This control problem is more complex. Taking into account the dynamic coupling Dong [50]-[53] designed a robust controller based on Lyapunov stability theory, and respectively designed a robust controller when the inertial parameters were inaccurately known and there were frictions, disturbances and uncertainties. Jagannathan [54] assumed that the dynamics was unknown. A fuzzy logic controller which linearized the composite system was considered. Yamamoto [55] researched a more complicated situation—two manipulators finished a task coordinately and proposed the concept of ellipsoid in the workspace. Kang [56] presented the utilization of inertial effect in damping-based posture control of a mobile manipulator.

The dynamic coupling between manipulators and mobile platform has a great influence on the robot's performance. Tan [26] compensated the dynamic coupling by extended Jacobian matrix. But this measure required the prior knowledge of dynamics. Yamamoto [57] not only derived the dynamic model coupling but also compensated the coupling by nonlinear feedback.

Another important issue in controller design is stability. The stability of mobile manipulators is difficult to ensure because it not only needs motion stability but also needs manipulation stability. Early researches only kept the static stability through the control of the mobile platform's center of mass [58], [59]. Yoneda [60] and Dubowsk [61] discussed the dynamic stability through controlling the interaction force between the platform and the ground, but

this method cannot be achieved offline. Huang made a research in depth for this issue [62]. Stability compensation range [63] can ensure the stability by controlling the configuration of the manipulator.

1.2.6 Force Control

Several researchers have investigated trajectory tracking problem for mobile manipulators. These control techniques are only adequate when the manipulators do not interact significantly with the environment. However, for many tasks mobile manipulators come in contact with constrained surfaces such as writing on a blackboard, wiping windows, scribing, grinding, and cleaning tables, and then interaction forces develop between the end-effectors and the environment. Therefore, contact force control is at least as important as position control. In such cases, the trajectories and the constraint forces of the systems are required to asymptotically converge to the desired ones, respectively. These cases can be considered that the mobile manipulators are subject to nonholonomic and holonomic constraints simultaneously.

Considerable research on force control schemes has been reported in the last two decades. Force control schemes which focus on realizing the control of force vector, motions, and/or desired relations between force and motion, can be categorized into four classes:

- Stiffness control involving the relation between position and applied force, impedance and admittance control involving the relation between velocity and applied force.
- Explicit force control and implicit force control whose aim is to regulate the applied force.
- Parallel position/force control.
- Hybrid position/force control or hybrid impedance control which simultaneously controls position/force based on tangential and normal subspaces of the constraint surface. In these control schemes, different controls are applied in two complementary orthogonal subspaces, position-controlled subspace and force-controlled subspace, separately.

Research on impedance control [64], [65] and hybrid position/force control [66], [67] are closely related to the study of robot performance of constrained tasks. Impedance control approach proposed by Hogan [64] aimed at controlling position and force by adjusting the mechanical impedance of the end-effector to external force generated by contact with the environment. Kazerooni [65] suggested that impedance controllers were built on the basis of

the internal position control loops. A method of actively controlling the apparent stiffness of an end-effector was presented [66]. The approach allowed the programmer to specify the three translational and three rotational stiffness of a frame located arbitrarily in hand coordinates. A method for the design of hybrid position/force controllers for constrained manipulators was derived [67]. This method can be applied to all types of constraints due to contact with the environment.

However, these methods do not establish a theoretical framework which can serve as a basis for the study of robot performance of constrained tasks. Though impedance control takes care of both constrained and unconstrained motions by specifying the performance of the controlled system as the desired generalized dynamic impedance, it does not address explicitly how to keep the end-effector in contact with the surface. And impedance control deals with control during contact with an environment that deforms due to an applied force or torque. The problem addressed in this thesis is control of a mobile manipulator while the end-effector is in contact with an environment with infinite stiffness. Specifically, control during contact with rigid surface is examined. Hybrid position/force control is usually used in this case.

However, we need to find a method which is similar to the hybrid control but unnecessary to specify an intermediate constraint frame. This thesis addresses the problem of stabilization of force/motion control for a class of mobile manipulator systems with both holonomic and nonholonomic constraints with uncertainties and disturbances even if the dynamics of the robots is completely unknown. The motion/force control of mobile manipulators has received many attentions in the literature (see [68]-[74] for references). A reduced dynamic model, suitable for simultaneous independent motion and force control was developed and a robust control algorithm was derived, which guaranteed the uniform ultimate boundedness of the tracking errors [68]. A robust adaptive motion/force tracking control has been proposed for uncertain nonholonomic mechanical systems [69], [70]. Robust control strategies were presented systematically for both holonomic mechanical systems and a large class of nonholonomic mechanical systems in the presence of uncertainties and disturbances [72]. Decoupling robust motion/force control strategies have been presented for a mobile manipulator with both holonomic and nonholonomic constraints which ensured the state of the system to asymptotically converge to the desired trajectory and force [73], [74].

1.2.7 Multi-Finger Manipulation

The ability to grasp is a fundamental skill for humans and a prerequisite for performing a wide range of object manipulation tasks. Therefore, grasping is also a fundamental requirement for assistant robots, if they are to perform meaningful tasks in human environments. In recent years, a lot of research has been conducted on robotic grippers with few DOFs which may not be particularly versatile. As a result, the number of robotic hands developed with multiple fingers has been steadily increasing. It has become an important area in robotics mainly because it increases the flexibility and versatility of robotic arms, allowing the use of a single end-effector for grasping and complex manipulation of a large variety of objects.

So far there are many developed multi-fingered hands. Among the most known robotic hands one can mention in a chronological order: the Okada Hand (1979) [75], the Stanford/JPL Hand (1983) [76], the Utah/Mit Hand (1983) [77], the Barret Hand (1988) [78], [79], LMS Hand (1998) [80], the DIST Hand (1998) [81], [82], the Robonaut Hand (1999) [83], [84], the Tokyo Hand (1999) [85], the DLR-Hand II (2000) [86], the Tuat/Karlsruhe Hand (2000) [87], the Ultralight Hand (2000) [88], the Gifu Hand (2001) [89], the Shadow Hand (2002) [90], the UB Hand III (2010) [91]. These types of multi-fingered robotic hands have an advantage that the hands can be used with various types of robotic arms because the robotic hands have independent structures.

Object manipulation with multi-fingered hands is a challenging task, especially in personal assistant robot applications. Much research has been done on hand mechanism, grasp and manipulation theory [92], [93], kinematics [94], [95] and dynamics [96], [97] of dexterous hands. In this section, we present an overview of the research work in the area of dexterous manipulation.

- Operating mode

A starting question for designing a robotic hand is how it is operated. For operating, [98] gives three possibilities: (1) human demonstration guided, (2) vision sensor guided and (3) tactile sensor guided.

- Type of grasp

A human hand can grasp in many different ways. The grasp of a human hand is always adapted to the shape and the weight of the object. Similarly, a multi-fingered robotic hand can use several grasp types such as fingertip grasp, and envelope grasp with taking the advantage of DOFs. In this thesis, the contact will be assumed to be one point per fingertip, which is known as a precision grasp [99].

- Control

Another important aspect of robotic hands is the control scheme implemented. The two main types of control are position control and force control. Position control can be used in the situations where there is little force to be generated. The grasping contact points must allow the application of forces on the object that achieve the object equilibrium, or to fully restrain the object to resist external disturbances. The execution of object grasping or manipulation requires controlling also the interaction forces to ensure grasp stability. This is accomplished by satisfying form or force closure conditions. In form closure, it is required that the object cannot move, even when there is no friction at any contact point. Force closure which is less strict demands that the object cannot move when the contact has friction, when the grasp force is large enough and there is a certain limited external force or torque [100].

- Contact modeling

There are three main contact models between fingertips and object including: frictionless, hard and soft finger contact [100]. Point contact without friction can only resist a unidirectional force normal to the surface. The considered friction model of the two latter is the static Coulomb friction model [101], which is good enough for grasping. A still open problem is the selection of contact points that would selectively allow slipping at some contacts and prevent it at others [100]. Rolling contact is changing the contact point when a curved fingertip rolls on the surface of an object. However, it is very difficult to control, because the statics and dynamics of the system are completely changed.

- Force optimization

The problem of GFO for a robotic hand consists in finding the set of contact wrenches balancing the generalized external force acting on the object (including object inertia and weight). This problem involves kinematic structure of the hand, joint torque limits, friction cone constraints to avoid the slippage of the fingers on the object surface and minimizing the internal forces. All of the proposed optimization methods have in common that the optimal

variables are grasping forces. The nonlinearity of the contact friction models (point contact with friction or soft-finger contact) is one of the main reasons of the computational complexity of GFO problem. [102]-[104] presented a class of optimal algorithms. These methods are computationally intensive because of the nonlinearity of the friction cone constraints. Thus, GFO problem is usually solved by using the linearization of the friction cones [105]–[108]. Han et al. [109] casted the friction cone constraints further into LMI and solved the grasping force problem by using the interior point algorithm. Buss et al. [110] reformulated the contact force optimization problem as a semi-definite programming problem. Moreover, Liu et al. [111] gave a min-max method for automatic generation of initial points.

- Macro-micro manipulation

A lightweight manipulator has flexibility due to the structure. A much smaller manipulator can be faster and more rigid. Thus the combination of a small “micro” manipulator carried by a large “macro” manipulator creates a robot that can be dexterous over a large area [112]. A typical example of a macro-micro manipulator is the robotic arm-hand system. Placing a robotic hand on the end of a larger arm, such as Nagai and Yoshikawa [113] describe, produces a redundant macro-micro manipulator for grasping and manipulation. The hand is used to add compliance and fine motion due to its small inertia, while the arm increases the overall range of motion. Sometimes this system is mounted on a mobile platform which allows for mobility of the macro-micro manipulator. The personal assistant robot designed in our project can be treated as one pair of macro-micro manipulators mounted on a mobile platform.

1.3 Mathematics Background

1.3.1 Introduction

This subchapter offers a short introduction to the technical background employed in this thesis arranged in five sections. In the first section, some mathematical preliminaries necessary for the theoretical development in subsequent chapters are first presented. In the second section, the essence of bond graph modeling is reviewed. Then we introduce and discuss the structure of RBFNN which is used in this thesis. Next we introduce the concepts of wrench and twist in screw theory. In the last section GL matrix and its operator are presented.

1.3.2 Mathematical Preliminaries

This section presents the mathematical preliminaries that are necessary for the derivations found later in this thesis. Unless noted, all the results may be found in [114], [115].

Definition 1.1 (Compact Set):

Let $\Re^n$ denote the space of real *n*-dimensional vector. A subset $S \subset \Re^n$ is said to be open if, for every vector $x \in S$, one can find an ε-neighborhood of x: $N(x,\varepsilon) = \{z \in \Re^n \mid \| z - x \| < \varepsilon\}$. A set *S* is closed if and only if its complement $\Re^n$ is open. A set S is bounded if there is $r>0$ such that $\| x \| \leq r$ for all $x \in S$. A set S is compact if it is closed and bounded.

Definition 1.2 (Vector and Matrix Norms):

The norm of a vector $x \in \Re^n$ is defined as:

$$\| x \| = \sqrt{x^T x} \tag{1.1}$$

The induced norm of a matrix is defined as:

$$\| A \| = \sqrt{\lambda_{\max}(A^T A)} \tag{1.2}$$

where $\lambda_{\max}(\circ)$ is the maximum eigenvalue of $A^T A$.

The Frobenius norm of a matrix is defined as the root of the sum of the squares of all elements:

$$\| A \|_F^2 = tr(A^T A) \tag{1.3}$$

where the trace $tr(A)$ satisfies $tr(A) = tr(A^T)$ for any $A \in \Re^{n \times n}$. For any $B \in \Re^{m \times n}$ and $C \in \Re^{n \times m}$, $tr(BC) = tr(CB)$. Suppose that A is positive definite, then for any $B \in \Re^{m \times n}$

$$tr(BAB^T) \geq 0 \tag{1.4}$$

with equality iff B is the $m \times n$ zero matrix. And

$$d\{tr(A(t))\}/dt = tr(dA(t)/dt) \tag{1.5}$$

Theorem 1.1:

Consider non-autonomous dynamic systems of the form:

$$\dot{x} = f(x,t),\ t \geq t_0 \tag{1.6}$$

where $x \in \Re^n$. Assume that the origin is an equilibrium point. Let $L(x,t): \Re^n \times \Re \to \Re$ be a scalar time-varying function such that $L(0,t) = 0$, and S be a compact subset of $\Re^n$. Then the system (1.6) is said to be

A. Lyapunov Stable

If for system (1.6), there exists a function $L(x,t)$ with continuous partial derivatives, such that for x in a compact set $S \subset \Re^n$,

$$L(x,t) \text{ is positive definite, } L(x,t) > 0; \tag{1.7}$$

$$\dot{L}(x,t) \text{ is negative semidefinite, } \dot{L}(x,t) \leq 0, \tag{1.8}$$

then the equilibrium point is stable in the sense of Lyapunov (SISL).

B. Asymptotically Stable

If furthermore, condition (1.8) is strengthened to

$$\dot{L}(x,t) \text{ is negative definite, } \dot{L}(x,t) < 0, \tag{1.9}$$

then the equilibrium point is asymptotic stable (AS).

C. Globally Stable

If the equilibrium point is SISL (stable in the sense of Lyapunov) or AS, if $S \subset \Re^n$, and in addition, the radial unboundedness holds:

$$L(x,t) \to \infty,\ \forall t \text{ as } \| x \| \to \infty, \tag{1.10}$$

then the stability is global.

D. Uniformly Stable

If the equilibrium point is SISL or AS, and in addition $L(x,t)$ decrescent, i.e., there exists *n* time-invariant positive definite function $L_1(x)$ such that

$$L(x,t) \le L_1(x,t), \ \forall t \ge 0, \tag{1.11}$$

then the stability is uniform (e.g., independent of t_0).

The equilibrium may be both uniformly and globally stable. For instance:

Definition 1.3 (Uniform ultimate boundedness, UUB):

Consider the dynamic system $\dot{x} = f(x,t)$, with $x \in \Re^n$. Let the initial time be t_0, and the initial condition be $x_0 \equiv x(t_0)$. The equilibrium point x_e is said to be uniformly ultimately bounded if there exists a compact set $S \subset \Re^n$ so that for all $x_0 \in S$ there exists a bound B and a time $T(B, x_0)$, such that $\| x(t) - x_e \| \le B$ for all $t \ge t_0 + T$.

In the above definition, the term uniform indicates that T does not depend on t_0. The term ultimate indicates that the boundedness property holds after a time lapse *T*. If $S = \Re^n$, the system is said to be globally UUB.

It is noted that both AS and SISL are too strong requirements for closed-loop control in the presence of unknown disturbances [116]. In practical closed-loop systems, the bound *B* depends on the disturbance magnitudes and other factors. However, *B* can be made sufficiently small if the controller is properly designed. Thus, UUB stability is sufficient for most control purposes [115].

A Lyapunov extension version of the Barbalat's lemma is presented in the following.

Theorem 1.2:

Let $L(x,t)$ be a Lyapunov function so that $L(x,t) > 0, \dot{L}(x,t) \le 0$. If $\dot{L}(x,t)$ is uniformly continuous, then

$$\dot{L}(x,t) \to 0, \text{ as } t \to \infty. \tag{1.12}$$

Recall that one may check for the boundedness of $\ddot{L}(x,t)$ which implies that $\dot{L}(x,t)$ is uniformly continuous. Barbalat's extension can be used to show that certain states of a system actually go to zero, though the standard Lyapunov analysis has revealed only that the system is SISL (i.e., states are bounded).

In practical applications, there are often unknown disturbances or modeling errors presented in the dynamic systems (1.6). i.e., systems of the form:

$$\dot{x} = f(x,t) + d(t), \tag{1.13}$$

with $d(t)$ an unknown bounded disturbance.

The next theorem shows that UUB is guaranteed if the Lyapunov derivative is negative outside some bounded region of $\Re^n$.

Theorem 1.3:

For system (1.13), if there exists a function $L(x,t)$ with continuous partial derivatives such that for x in a compact $S \subset \Re^n$,

$L(x,t)$ is positive definite, $L(x,t) > 0$;

$\dot{L}(x,t) < 0$ for $R > 0 \; \| x \| > R$.

For some $R > 0$ such that the ball of radius R is contained in S, then the system is UUB, and the norm of the state is bounded to within a neighborhood of R.

1.3.3 Review of Bond Graph Technique

For the sake of clarity, the concept of bond graph will be briefly discussed. Bond graph methodology was established by Prof. Henry Paynter in 1961. It is a graphical modeling language for modeling multi-physical systems with a unique representation. Different physical systems such as electrical, mechanical, magnetic, fluid, chemical, and thermodynamic systems can be modeled and linked together. In this method, the various elements are connected by means of power bonds representing dual variables called effort (e) and flow (f) as depicted in Figure 1.3. A bond is represented by a half arrow which at the same time means the direction of power flow. Within a bond the direction of effort variable is

always in opposite to the flow variable due to the nature, and the product of both variables signifies the power magnitude. These bonds are connected to junctions and system elements.

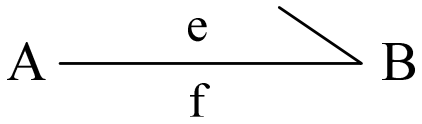


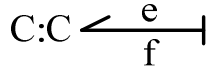
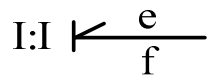
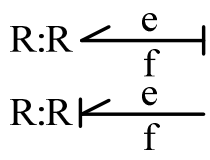
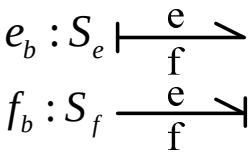
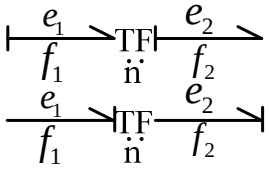
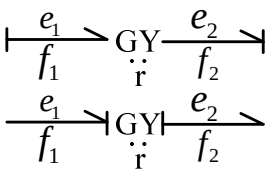
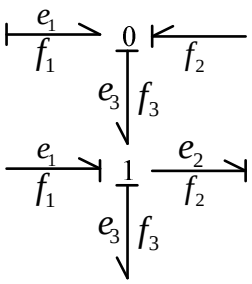
Figure 1.3: Power moves from system A to system B

Table 1.2: Domains with corresponding flow, effort, generalized displacement and generalized momentum

	f flow	e effort	$q = \int f\,dt$ generalized displacement	$p = \int e\,dt$ generalized momentum
Electromagnetic	i current	u voltage	$q = \int i\,dt$ charge	$\lambda = \int u\,dt$ magnetic flux linkage
Mechanical Translation	v velocity	f force	$x = \int v\,dt$ displacement	$p = \int f\,dt$ momentum
Mechanical Rotation	ω angular velocity	T torque	$\theta = \int \omega\,dt$ angular displacement	$b = \int T\,dt$ angular momentum
Hydraulic / Pneumatic	φ volume flow	P pressure	$V = \int \varphi\,dt$ volume	$\tau = \int P\,dt$ momentum of a flow tube
Thermal	T temperature	F_S entropy flow	$S = \int f_S\,dt$ entropy	
Chemical	μ chemical potential	F_N molar flow	$N = \int f_N\,dt$ number of moles	

A bond graphs model is composed of one or a combination of the following basic elements. Energy dissipation is modeled by resistive element **R**, energy storage devices such as springs and inertia are modeled by elements **C** and **I**, sources of efforts and flows are represented by **Se** and **Sf**, and power transformations are modeled by transformers **TF** or gyrators **GY**. Power bonds join at **0** junctions summing flows to zero with equal efforts or at **1** junctions summing efforts to zero with equal flows, which implements Newton's 2nd law. **R**, **C** and **I** are 1-port elements. **TF** and **GY** are 2-port elements. **1** and **0** junctions are multi-port elements. With these elements various systems can be built.

Table 1.3: Common bond graph elements

Symbol	Bond-graph Element	Explanation	Examples
C		Storage element for a q-type variable	Capacitor (stores charges), Spring (stores displacement)
I		Storage element for a p-type variable	Inductor (stores flux linkage), mass (stores momentum)
R		Resistor dissipating free energy	Electric resistor, Mechanical friction
Se, Sf		Effort sources and Flow sources	Electric mains (voltage source), Gravity (force source), Pump (flow source)
TF		Transformer	Electric transformer, Toothed wheels, Lever
GY		Gyrator	Electromotor, Centrifugal Pump
0,1		0 and 1 Junctions	Ideal connection between two sub models

1.3.4 Radial Basis Function Neural Network

It is assumed that a nonlinear function is completely unknown to the control designer, and that there exists a set of NN weights such that the output of a RBFNN [117] approximates the function $f(x)$. The most important feature that distinguishes the RBF network from earlier RBF-based models is its adaptive nature which generally allows it to utilize a relatively smaller number of locally-tuned units. The following is a description of the basic RBF network architecture.

$f(x): \Re^a \to \Re^b$ is assumed to be linearly parameterized by RBFNN over a sufficiently large compact region of interest $D_x \subset \Re^a$ in the state space such that:

$$f(x) = y(x) + \varepsilon(x) \tag{1.14}$$

where $y(x) = W^T \varphi(x)$.

For all $x \in D_x$ and $\varepsilon(x)$ is the function approximation error which is bounded as:

$$\|\varepsilon(x)\| \le \varepsilon_b \tag{1.15}$$

where ε_b is a positive number.

Figure 1.4 depicts a RBFNN. The design of a RBFNN in its most basic form consists of three separate layers. The input layer is the set of source nodes. The second layer is a hidden layer of high dimension. The output layer gives the response of the network to the activation patterns applied to the input layer. The transformation from the input space to the hidden-unit space is nonlinear. On the other hand, the transformation from the hidden space to the output space is linear. The output is given by:

$$y(x) = \sum_{i=1}^{n} w_i \varphi(\|x - c_i\|) \tag{1.16}$$

where

x is the input vector,
$\varphi(\cdot)$ is an arbitrary nonlinear radial basis function,
$\|\cdot\|$ denotes the norm that is usually assumed to be Euclidean,
c_i is the center of the radial basis function,
w_i is the weight.

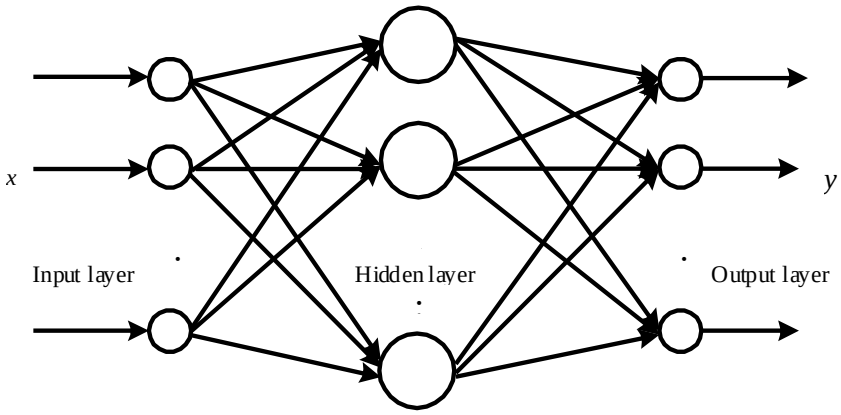


Figure 1.4: RBFNN basic structure

Radial functions are a special class of functions. Their characteristic feature is that their response decreases (or increases) monotonically with the distance from a central point and they are radially symmetric. Center, distance scale and precise shape of the radial function are the parameters of the model. There are a variety of radial functions available in literature. The most commonly used one is the Gaussian function.

$$\varphi(x) = \exp(-\frac{\|x-c\|^2}{b^2}) \tag{1.17}$$

Its parameters are its center c and its radius b (width). Figure 1.5 illustrates a Gaussian RBF with center $c=0$ and radius $b=1$. A Gaussian RBF monotonically decreases with the distance from the center.

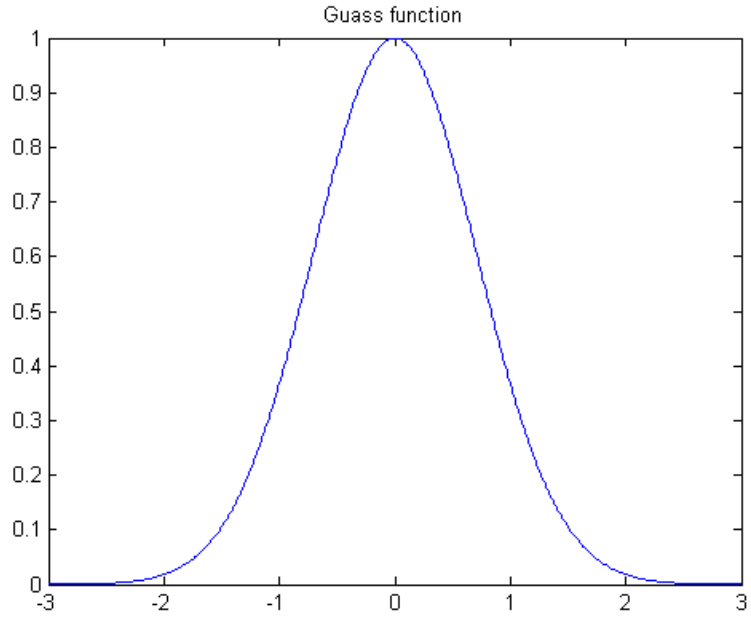


Figure 1.5: Gaussian function profile

A summary of the characteristics of RBFNN is given below:

- It is a feed-forward network.
- The hidden nodes implement a set of RBFs (e.g. Gaussian functions).
- The output nodes implement linear summation functions.
- The network training is divided into two stages: first the weights from the input to the hidden layer are determined, and then the weights from the hidden layer to the output layer are determined.
- The training/learning is very fast.

1.3.5 Twist/Wrench

Now we introduce vectors called twist and wrench. The concepts of twist and wrench were introduced by Ball [118] in his groundbreaking work. According to screw theory the velocity of a rigid body can be described by a twist represented using a vector (rotational velocity ω and translational velocity *v*). Likewise, a wrench (force f and torque τ) is a vectorial quantity that describes the resultant of a system of forces acting on a rigid body.

$$V = \begin{bmatrix} v \\ \omega \end{bmatrix} : \text{twist} \tag{1.18}$$

$$F = \begin{bmatrix} f \\ \tau \end{bmatrix} : \text{wrench} \tag{1.19}$$

We will explain how twists and wrenches transform from one coordinate frame to another. A homogeneous matrix g_{ij} represents the position of a frame *i* with respect to a frame *j* with a translation component P_{ij} and a rotation component R_{ij}:

$$g_{ij} = \begin{bmatrix} R_{ij} & P_{ij} \\ 0 & 1 \end{bmatrix} \tag{1.20}$$

which represents a change of coordinates for twists in matrix form. It is possible to see [119], [120] that a change of coordinates for twists in different frames, as applied in this thesis, can be expressed in a vector form:

$$V^i = Ad_{g_{ij}} V^j \tag{1.21}$$

with $Ad_{g_{ij}} = \begin{bmatrix} R_{ij} & P_{ij}R_{ij} \\ 0 & R_{ij} \end{bmatrix}$.

The transformation of wrenches is different from that of twists. The rule for transforming a wrench from frame *i* to frame *j* is:

$$F^j = Ad^T_{g_{ij}} F^i \tag{1.22}$$

1.3.6 GL Matrix and Multiplication Operators

When the symbol "$\{\bullet\}$" is defined as a GL matrix [147] and "$\bullet$" as its method operator, the GL vector $\{\Theta\}$ and its transpose $\{\Theta\}^T$ may be defined as the following equation:

$$\{\Theta\}=\begin{Bmatrix}\theta_{11} & \theta_{12} & \cdots & \theta_{1n}\\ \theta_{21} & \theta_{22} & \cdots & \theta_{2n}\\ \vdots & \vdots & \vdots & \vdots\\ \theta_{n1} & \theta_{n1} & \cdots & \theta_{nn}\end{Bmatrix}=\begin{Bmatrix}\{\theta_1\}\\ \{\theta_2\}\\ \vdots\\ \{\theta_n\}\end{Bmatrix} \tag{1.23}$$

$$\{\Theta\}^T=\begin{Bmatrix}\theta_{11}^T & \theta_{12}^T & \cdots & \theta_{1n}^T\\ \theta_{21}^T & \theta_{22}^T & \cdots & \theta_{2n}^T\\ \vdots & \vdots & \vdots & \vdots\\ \theta_{n1}^T & \theta_{n2}^T & \cdots & \theta_{nn}^T\end{Bmatrix} \tag{1.24}$$

For a given GL matrix $\{\Xi\}$

$$\{\Xi\}=\begin{Bmatrix}\xi_{11} & \xi_{12} & \cdots & \xi_{1n}\\ \xi_{21} & \xi_{22} & \cdots & \xi_{2n}\\ \vdots & \vdots & \vdots & \vdots\\ \xi_{n1} & \xi_{n1} & \cdots & \xi_{nn}\end{Bmatrix}=\begin{Bmatrix}\{\xi_1\}\\ \{\xi_2\}\\ \vdots\\ \{\xi_n\}\end{Bmatrix} \tag{1.25}$$

Then the GL multiplication of $\{\Theta\}^T$ and $\{\Xi\}$ may be defined as a $n\times n$ matrix, as follows:

$$\{\Theta\}^T\{\Xi\}=\begin{Bmatrix}\theta_{11}^T\xi_{11} & \theta_{12}^T\xi_{12} & \cdots & \theta_{1n}^T\xi_{1n}\\ \theta_{21}^T\xi_{21} & \theta_{22}^T\xi_{22} & \cdots & \theta_{2n}^T\xi_{2n}\\ \vdots & \vdots & \vdots & \vdots\\ \theta_{n1}^T\xi_{n1} & \theta_{n2}^T\xi_{n2} & \cdots & \theta_{nn}^T\xi_{nn}\end{Bmatrix} \tag{1.26}$$

where, it should be noted that the GL multiplication should be calculated firstly in the mixed calculation of the matrix and the GL matrix multiplication operator.

The GL product of a square matrix and a GL row vector is defined as follows:

$$\Gamma_k\bullet\{\xi_k\}=\{\Gamma_k\}\bullet\{\xi_k\}=\begin{bmatrix}\gamma_{k1}\xi_{k1} & \gamma_{k2}\xi_{k2} & \cdots & \gamma_{kn}\xi_{kn}\end{bmatrix}\in\Re^{m\times n} \tag{1.27}$$

where $\Gamma_k=\Gamma_k^T=\begin{bmatrix}\gamma_{k1} & \gamma_{k2} & \cdots & \gamma_{kn}\end{bmatrix};\ \gamma_{kj}\in\Re^{m\times n_{kj}};\ m=\sum_{j=1}^{n}n_{kj}$.

CHAPTER2 DEFINITION, DESIGN AND VIRTUAL PROTOTYPING

2.1 Robot Applications and Requirements

2.1.1 Introduction

Over decades, many academies and research groups have been studying personal assistant robots in order to realize the robots that can coexist with humans and perform a variety of tasks, especially as daily-life assistants within domestic environments. Christensen [121] named three groups of domestic applications (entertainment, assistance for elderly and handicapped, everyday tasks) for this kind of robots. And in [122], Manja Lohse proposed six categories (entertainment, personal assistant, healthcare/care giving, companionship, toy, Pet). Fong et al. also differentiated between a toy and an entertainment robot [123].

Among a large number of this type of robots that have been developed, the robots described in the state of the art in Figure 1.1 deserve a particular mention. In the functional design process we first make a comparison of these robots (Table 2.1 gives an overview of the specifics of these robots). We target our robot taking advantage of a number of these robots' design features as follows:

- Care-O-bot 3: Human-like manipulation (human-like 7DOFs arm)
- Robot AR: Advanced recognition functions
- TWENDY-ONE: Dexterous manipulation, torso flexibility (dexterous hand and flexible waist)
- Enon: Simple user interface (touch screen)
- Personal Robot PR2: Gripper observation (each forearm contains a camera); advanced vision capability (pan-tilt head contains wide and narrow stereo cameras)
- Readybot: Remote operation (linkable to network)

2.1.2 Applications

Analyzing the different classification methods and combining the advantages listed above, we design the following applications.

Healthcare /Care giving:

- Communication and social integration
  - Voice as natural and intuitive command interface,
  - Multimedia touch-screen as an additional interface,
  - Providing emergency calls,
  - Management of personal contacts,
  - Communication with doctors,
  - Management of media devices (video telephone, TV, CD, radio, etc.).
- Technical house management (Infrastructure)
  - Control of home infrastructure devices such as heating, air conditioning, lights, windows, front door, security / alarm systems etc.
- Fetch and carry tasks
  - Providing medicine and other care products,
  - Providing laundry and delivering books etc,
  - Grasping, lifting and holding objects and devices in cooperation with a human.
- Mobility support
  - Seating and stand-up assistance,
  - Giving a helping hand on the way to the bathroom.
- Personal security
  - Monitoring of personal safety,
  - Monitoring of physiological parameters (measurements of respiration rate, blood pressure, temperature, pulse rate, etc.),
  - Monitoring of mental behavior,
  - Providing health information and medical advice,
  - Detecting and distinguishing the abnormal behaviors of elder's daily activities,
  - Control of safety threats (smoke, water, gas, burglary),
  - Emergency alarms (gas leakage, fire, etc.).

Personal Assistant:

This category includes robots for cleaning and other household chores.

- Meal preparation and serving (e.g. micro wave, stove)
- Pouring out drinks
- Setting tables
- Loading and unloading dishwashers
- Simple cleaning and tidying tasks (sweeping a floor by using a bloom, gathering clothes and putting it in a washer etc)
- Vacuum cleaning

Companionship and Entertainment:

- Tracking and following
- Facial expressions
- Playing music
- Providing information (e.g. weather)

In addition to these domestic functions, our robot also can work in public places as a **public assistant**:

- Entertainer: it approaches the visitors and welcomes them
- Keeping tracking of people
- Guide: it gives guided tours
- Transport of objects
- Security patrolling

Table 2.1: Robot comparison

Function / Name			1 Manipulation (DOF/Number)	2 Self-navigation	3 Vision	4 Omni-directional motion	5 Fingers	6 Touch screen	7 Recognizing and responding to gestures	8 Spoken commands (has a speaker and speech recognition)
1	Care-O-bot3 (1.45 m)	Domestic robot	7×1	Yes	Stereo-vision , laser scanners, a 3D range camera	Yes (with four separately steered and driven wheels)	3 (7 DOFs)	Yes	Yes	Yes
2	Robot AR (1.5m)	Domestic robot	7×2	Yes	4 stereo cameras, a laser sensor	No (wheeled mobile platform)	3 (6 DOFs)	No	No	No
3	Twendy-one (1.5m)	Domestic robot	7×2	Yes	Two CCD cameras surrounded by LEDs	Yes (with four wheels whose circumference has 15 rotating members)	4 (13DOFs)	No	No	Yes
4	Enon	Guide & Office Robot	5×2	Yes	6 cameras	No (2 DOFs)	2	Yes	No	Yes
5	PR2	Personal robot (it is an open platform, you can change the system to meet your needs at any level)	7×2	Yes	Wide and narrow stereo cameras, a 5-megapixel camera, a texture projector and a Hokuyo laser range-finder	Yes (4 casters steered and driven)	2	No	Adding	Adding
6	Readybot	Domestic robot(house-cleaning robot)	5×2	No	Not mentioned, but it has no 3D Stereo-vision	Yes (industrial omni-wheels)	2	No	No	Yes

Function / Name		9	10	11	12	13	14	15	16	17	18	19	20
		Player	Learning to recognize new objects	With a tray	Transport of objects	Grasping	Environment Perception	Cleaning the floor	Waist	Picking up a tray and convey	Gathering clothes and putting it in washer	Failure detection and recovery	Holding a human
1	Care-O-bot3	No	Yes	Yes	Yes (by tray)	Yes	Yes	No	No	No	No	No	No
2	Robot AR	Ipod nano	No	No	Yes	Yes	Yes	Yes (by using a bloom)	Yes (1DOFs)	Yes	Yes (by finding wrinkles)	Yes	No
3	Twendy-one	No	Yes	No	Yes	Yes	Yes	Yes	Yes (4DOFs)	Yes	No	No	Yes
4	Enon	No	No	No	Yes (by using its specially designed carriage)	Yes	Yes	No	No	No	No	No	No
5	PR2	No	Yes	No	Yes	Yes	Yes	Adding	Yes (2DOFs)	Yes	Yes	No	Adding
6	Readybot	No	No	No	Yes	Yes	Yes	Yes (scraping small items into plastic bins by carpet rake)	No	Yes	No	No	No

Function / Name		21 Making breakfast	22 Feeling human touch	23 Linkable to networks	24 Providing the information	25 Displaying its operational state	26 Greeting and escorting guests	27 Security patrolling	28 Observing the grippers
1	Care-O-bot3	No	No	No	No	No	No	No	No
2	Robot AR	No	No	No	No	No	No	No	No
3	Twendy-one	Yes	Yes	No	No	Yes (LED)	No	No	No
4	Enon	No	No	Yes	Yes (by voice or image)	Yes (LED)	Yes	Yes	No
5	PR2	Adding	No	No	No	No	No	No	Yes(the forearm contains a camera)
6	Readybot	No	No	Yes	No	No	No	No	Yes

Function / Name		29 Changing the height	30 Vacuuming the carpet	31 Dish washing	32 Tidying up the rooms (including dumping, wiping…)	33 Swimming pool cleaning	34 Giving an alarm when detecting the smoke	35 Clearing snow	36 Autonomous charging
1	Care-O-bot3	No	No	No	No	No	No	No	No
2	Robot AR	No	No	No	No	No	No	No	No
3	Twendy-one	No	No	Yes	No	No	No	No	No
4	Enon	No	No	No	No	No	No	No	No
5	PR2	Yes (telescoping spine)	No	Adding	Adding	No	No	No	No
6	Readybot	Yes	Yes (by releasing a off-the-shelf cleaning robots)	Yes (putting dirty dishes into the dish washer)	Yes (performing 30-40% of common daily kitchen and general house cleaning tasks.)	No	No	No	No

2.1.3 Requirements

For realizing these robot applications, some specific requirements have to be fulfilled to maximize the usefulness of the personal assistant robot. These include the following:

- Mobile Base

A personal assistant robot imposes high requirement on the employed mobility concept. Usually the robot performs its tasks in environments where humans can walk around without any problems. Most of those environments have in common that wheeled platforms can access them. Different wheel-based mobile platforms have been developed. Differential wheeled robots use a pair of regular standard wheels; omnidirectional mobile robots employ a wide variety of wheel types such as rollers, Mecanum wheels and spherical wheels.

- Manipulation System

According to the anthropomorphic model two dexterous hands mounted on a pair of humanoid robotic arms should be used for the manipulation system in order to manage some sophisticated tasks, e.g., opening bottles. To manipulate arbitrarily positioned and oriented objects the manipulation system needs at least six DOFs. For the design of arms and hands several conditions should be considered: human-like ranges of motion and size, lightweight and low-cost. In addition, a bendable body is necessary to help the robot pick up objects from the ground.

- Sensor System

Sensors are used to give feedback of both interior and exterior states of the robotic system and to supervise the current interaction of its actuators with the environment. Similar to humans the robot's sensor system can be subdivided into exteroceptors and proprioceptors. Exteroceptors detect stimuli outside the body (e.g. eye, ears, etc.) whereas proprioceptors relate to detect stimuli inside the body (e.g. tendons, muscles and joints).

- Information Processing and Robot Control

A complex robotic system depends on fast and reliable sensor data processing as a basis for intelligent robot control. Usually a control board is used to capture image, process image, make decision, and control motions for the robot. A hierarchically structured multi-processor system seems to be the best solution. Different types of processors are to be chosen for every

hierarchical level. On the highest hierarchical level an operator should be able to enter tasks and supervise the whole system. On the second hierarchical level operator commands are transferred to sensor-based actions of the robot. The lowest hierarchical level is formed by micro controllers that control actuators and pre-process sensor data.

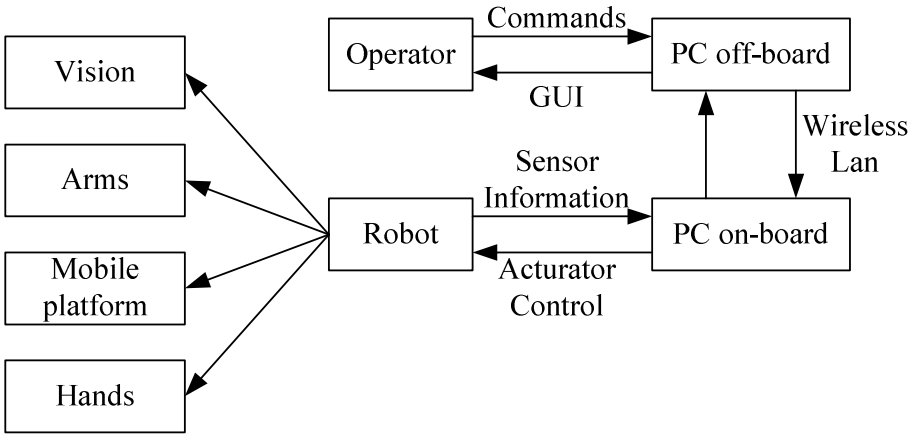


Figure 2.1: Architecture of robotic system

- Man-Machine Interface

Research in this area focuses on natural human channels of communication, such as graphical user interface, touch user interface, natural language and gestures, to provide easy and friendly multimodal interfaces between humans and the robot.

- Safety Precautions

For establishing a system which can provide the robot with abilities to achieve some tasks, failure detection and recovery abilities are necessary. Failure indicates the condition that the robot cannot plan its motion or cannot verify the success of executed motion in the middle of or after the task. The robot should have emergency switches that disconnect all drives from the on-board power supply in case of danger.

On the other hand, the robot working in close proximity to humans, even sharing their operating envelopes, is exposed to a special danger. Safety can be enhanced by placing tactile sensors on the robot's surface to ensure that the robot could not harm humans. We believe that software has a major role to play in overall system safety.

- Actuators

Actuator provides the driving force and motion for robot joint. Given the wide variety of existing actuators, the selection is based on matching the performance characteristics of the actuator, such as force and displacement, to the requirements of the given tasks.

- Power Supply

A power supply can put out a reasonable amount of current for servos and other small motors as well as run processors and sensors. For a completely autonomous robot, a sensor sensing the battery charge should be added. If the charge goes below a certain limit, the robot should stop the task and look for a charging station without any human intervention.

- Cost

Two competing factors are considered when designing the robot: cost and functionality. The assistant robot needs enough functionality to perform various tasks. While at the same time, it must be simple enough to keep the cost low. As a design priority, acceptable cost is essential for development and must be reconciled with any hi-tech solution. Low cost can be achieved by (a) using readily available components, (b) designing a reasonable control system to reduce high computational power which results in high cost of the equipment and even higher cost for developing the software for the control applications and (c) reducing the complexity of mechanical design.

2.2 Mechanical Design and 3D Modeling

2.2.1 Introduction

The aim of this subchapter is designing the mechanical structure of a personal assistant robot and constructing a virtual prototype similar to the real one using virtual simulation technology.

2.2.1.1 Mechanical Structure

Researchers worldwide are trying to develop robots that look and move more like humans. If the environment is real (stairs, not flat floor, cluttered and without modification) the biped humanoid is the best choice for its human-like structure and adability to any floor. For instance, ASIMO, QRIO. However biped humanoid robots have a very complex mechanical structure and are difficult to control. In contrast if the environment is “empty” and the floor is flat, the wheeled humanoid is the best solution duo to its easy locomotion, easy design, higher speed and easy control. Usually personal assistant robots operate in offices, hospitals, houses, etc, where the surfaces are flat. As a result wheels are more efficient. Some examples of this type of robots are illustrated in Figure 1.1. We provide a brief description of the structures of these robots.

- Care-O-bot 3: an omnidirectional platform, with four steered and driven wheels; a 7 DOFs arm; a three-finger hand.
- Robot AR: two 7 DOFs arms; a 3 DOFs head; a 1 DOF waist; each end-effector equips 3 fingers and each finger is composed of 2 joints; lower body is wheeled mobile platform.
- TWENDY-ONE: omni directional mobility; two 7 DOFs arms; two 13 DOFs hands; 3 DOFs neck; 4 DOFs body.
- Enon: two active wheels, 2 DOFs neck; two 5 DOFs arms; two grippers.
- PR2: omnidirectional mobile base; a spine; two 7 DOFs arms; two grippers.

On the basis of the analysis of different robots' structures listed above, in the mechanical design stage, we positively consider the features of architecture and the movable joint ranges. In particular, we seriously endeavor to match the shape of the joints and links with the design concept.

2.2.1.2 Virtual Simulation Technology

The purpose of virtual simulation is to create the model of the real system for the optimization purposes. In this way, we can optimize parameters of the system and test it for various conditions even before the production of physical prototype. Thus, the virtual prototype modeling and simulation improve the design efficiency. On the market there are many multi-body software solutions. For example, Simpack, SimMechanics, MEDYNA, DADS, ADAMS and so on. We chose ADAMS, because it is widely used in industry and for its easy co-simulation with Matlab/Simulink. Via Parasolid interface in ADAMS, the virtual prototype can easily be constructed through the importation of a solid model built by 3D CAD software, such as Solidworks.

Through the approach described above, in the 3D modeling stage, a 3D solid model of the mobile manipulator is first developed in Solidworks, the model is then imported into the dynamic simulation software ADAMS for kinematic and dynamic simulations. Finally, through the interface modular part (ADAMS/Controls) of Matlab and ADAMS, a mobile manipulator control system is established with Simulink and to successfully implement a coordinated simulation. This is very important, because with this tool we will be able to know approximately which will be the behavior of the robot constructing only a CAD model without the need to build a physical prototype.

2.2.2 Mechanical Design

2.2.2.1 Mechanical Configuration

The mechanical design of the complete robot requires careful and complex tradeoffs between form, function, power, weight, cost and manufacturability. According to the functionalities and the requirements discussed in the previous subchapter, we target our personal assistant robot consisting of a mobile platform equipped with dual-arm/hand system with adequate payload and dexterity characteristics. For example, the mobile base should be able to transit doors and navigate in human-sized environments. The fingertips of the dexterous hands should reach a reasonable fraction of the workspace reachable by a standing person and possibly be able to pick up objects on the ground.

For realizing the capabilities mentioned above, our human-like mobile manipulator prototype roughly stands 140 cm tall and weighs 80 kg. Figure 2.2 shows its dimensions. The high number of DOFs allows the robot to realize complex motions. Figure 2.3 describes the structure of the mobile platform. Figure 2.4 shows the structure of the robotic hand.

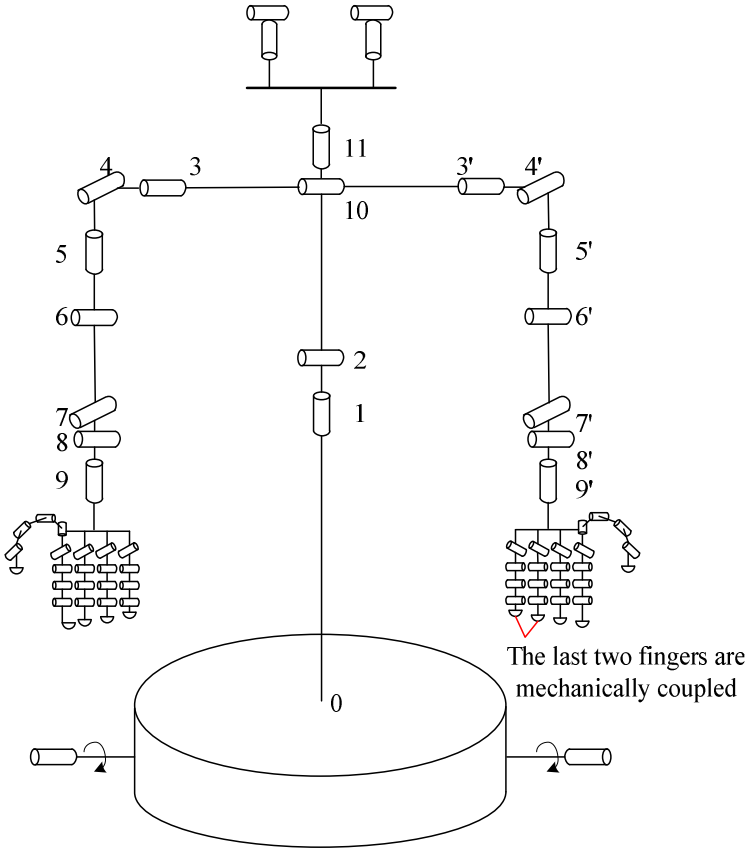


Figure 2.2: Dimensions of robot model

Part	DOF
Eyes	2 (pan-tilt) ×2
Neck	2
Waist	2
Arms	7×2
Hands	16×2
Mobile base	2
Total	56

Table 2.2: DOFs configuration

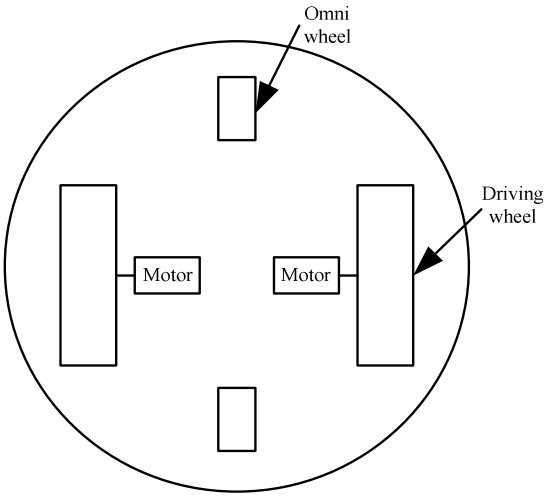


Figure 2.3: Structure of mobile platform

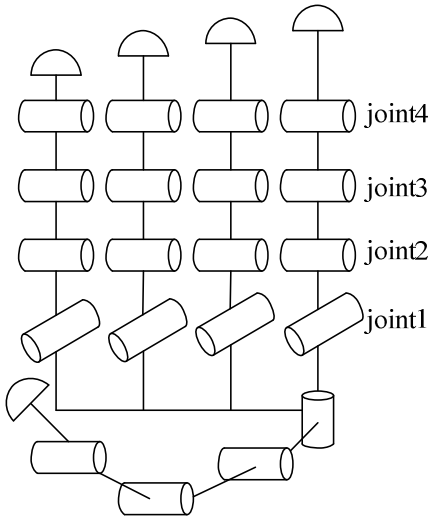


Figure 2.4: Structure of dexterous robotic hand

From the description of the dimensions, it can be seen that the robot's head can not only rotate left and right, but also rotate from up and down. It also can independently move the eyeballs because the robot has 2 DOFs for each eye (for a camera pan and tilt) which allows a wide field of vision change. As a consequence, the robot's head can obtain all needed object information. Our robot is equipped with two 7 DOFs arms (3 DOFs shoulder, 1 DOFs elbow and 3 DOFs wrist) like human and with two five-fingered hands. The shoulder gives pitch, yaw and roll, the elbow allows for pitch, and the wrist allows for pitch, yaw and roll. By design the arms can move up and down (heaving); move left and right (swaying); move forward and backward (surging); tilt up and down (pitching); turn left and right (yawing); tilt side to side (rolling). The upper body also consists of a 2 DOFs waist. This bendable waist allows the robot to pick up objects on the ground. The robotic dexterous hand has a similar configuration with the human hand. It has five fingers and each finger is composed of 4 joints. The thumb and the other four fingers (index, middle, ring and little fingers) have similar structure in terms of kinematics and dynamics. The last two fingers (the ring and the little fingers) are mechanically coupled so that they keep the same motion, therefore together they just have 4 DOFs, and the little finger only plays a supportive role. From this description the dexterous hand has 16 DOFs in total. This design can achieve the dexterity of human hand such that it can be used to perform various operations.

Most commercial manipulators are heavy, large, and power hungry. Highly mobile robots, such as legged mobility platforms, cannot easily accommodate dexterous manipulators. To overcome this difficulty, a differential drive mobile platform with two conventional wheels

and two supporting wheels has been chosen. The wheeled drive is chosen over leg drive because of safety and stability during manipulation.

2.2.2.2 Joint Angle Range

In order to ensure the robotic model can move similarly to a human, the angle ranges for the robot are defined according to the following two considerations:

- The specific angle ranges of human being

In a human body, each joint has its own reachable angle limit. Referring to the material mentioned in [124], the angle ranges of this humanoid model are defined.

- The maximum rotated angle

The other point to be concerned about is the maximum rotation angle of servo motors which are used to build up the real model is only 180 degree.

Table 2.3 and Table 2.4 make a comparison between the joint angle ranges of the human and the humanoid robot. These movable ranges of the robot are similar or over the human's ones.

Table 2.3: Comparison between human and humanoid robot angle ranges

Joint	Human (deg)				Robot (deg)			
	number	yaw	pitch	roll	number	yaw	pitch	roll
Neck	3(yaw, pitch, roll)	-80-80	-45-45	-45-45	2(yaw, pitch)	-90-90	-45-45	
Waist	3(yaw, pitch, roll)	-30-30	-30-90	-30-30	2(yaw, pitch)	-90-90	-10-120	
Right/left shoulder	3(yaw, pitch, roll)	-90-50	-90-40	-135-40	3(yaw, pitch, roll)	-90-90	-180-90	-135-10
Right/left elbow	1 (pitch)		-10-150		1(pitch)		-10-150	
Right/left wrist	3(yaw, pitch, roll)	-30-40	-70-70	-90-90	3(yaw, pitch, roll)	-90-90	-90-90	-90-90

Table 2.4: Angle ranges of finger joints

Joint	Human (deg)	Robotic finger (deg)
Joint1	-15-15	-90-90
Joint2	0-90	-30-120
Joint3	0-110	-30-120
Joint4	0-90	-30-120

Based on the above mechanical configuration, compared to the traditional industrial mobile manipulators, this robot is capable of performing various tasks in unstructured or semi-structured environments, thus it could address greater challenges in autonomous localization and navigation, object identification, control and coordination.

2.2.3 Hardware and Control Architecture Design

The hardware of the assistant robot includes sensor systems and actuators as well as necessary connections to the embedded computers. The specifications for the developed robot are shown in Table 2.5.

Table 2.5: Specifications of robot

Specifications		
Max work weight		About 20kg per arm
Weight/Height		80kg/140cm
Actuators		DC servo motors
Sensors	Motor	Motor encoders
	Head	Two CCD cameras and two microphones
	Mobile platform	A laser range finder and a number of ultrasonic distance sensors
	Arm	A force/torque sensor on each wrist; a camera on each forearm
	Hand	Multi-axis force/torque sensors and tactile sensors
Operation section		Laptop with wireless LAN, keyboard and mouse
Battery		NIMH 48V, 13Ah battery pack

Normally, a DC servo motor drives each joint with a rotary encoder. The mobile platform includes a battery pack, a laser scanner, a number of ultrasonic distance sensors and a processor system. The robot is equipped with two force/torque sensors on the wrists and two cameras on the forearms. Two microphones and two color CCD cameras are installed on the robot head. Each dexterous robotic hand needs as a minimum a set of multi-axis force/torque sensors in all fingertips and position sensors at all joints to enable control schemes like force/torque control and position control in autonomous operation. Additionally, tactile sensors are distributed on the surface of the fingers and palm.

In the actuator system, robotic hands usually can be divided into two types: built-in actuator type and external actuator type. Built-in actuator type of robotic hands generates the motion of fingers by using motors installed inside the fingers or palm. This type of robotic hands drives each joint by direct drive or gear drive. In such a configuration, the weight of the dexterous robotic hand is quite heavy because of using numerous gear parts and motors. On the other hand, external actuator type makes the structures of its fingers simple and light by using wire or belt driving mechanism. In our project, we choose the latter one. The fingers which are to be actuated independently through motors placed at distance locations.

The control system has the hierarchical control architecture. This architecture consists of four levels shown in Figure 2.5. It shows a conceptual scheme for the configurations of boards and the overall communication system.

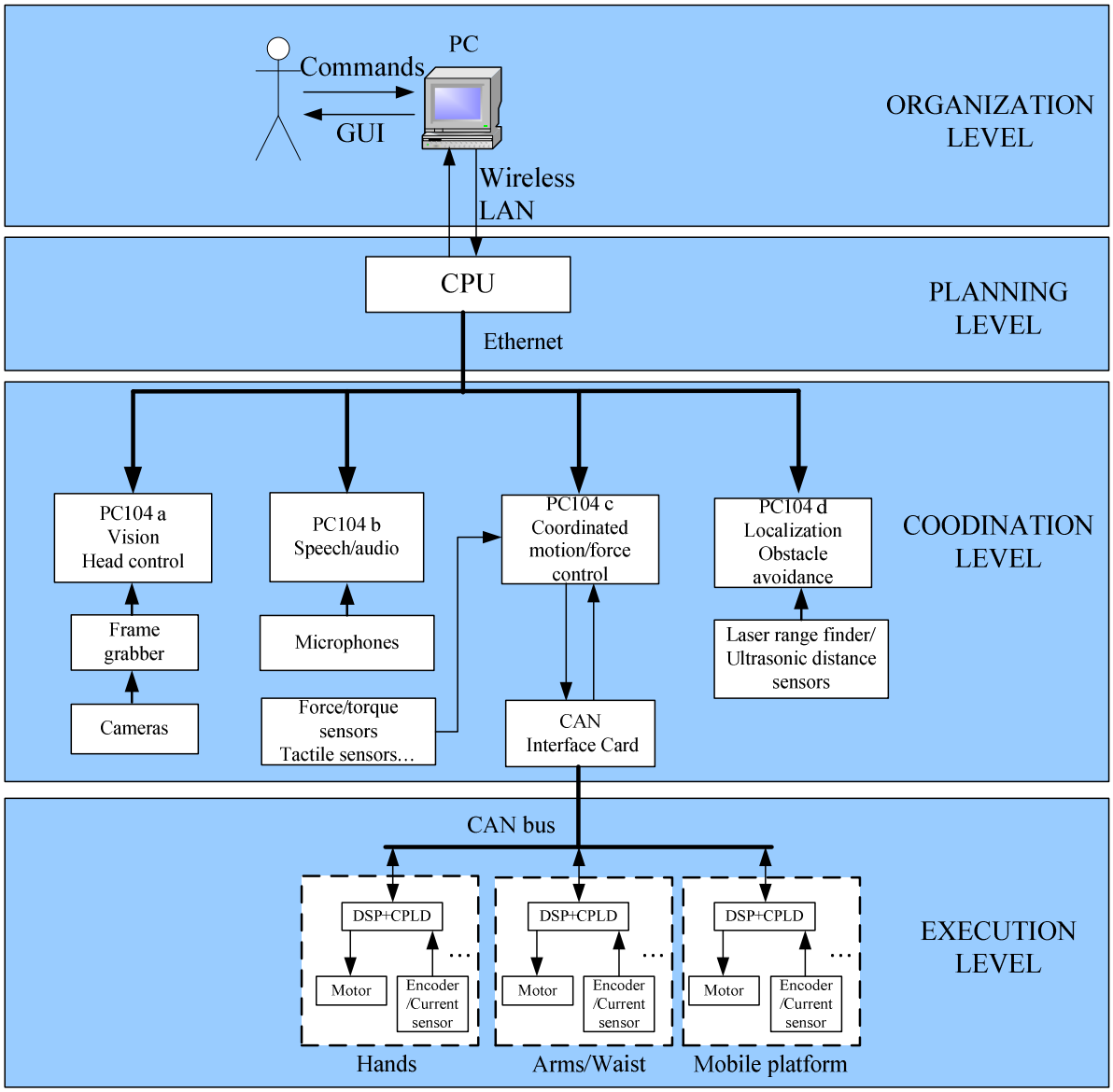


Figure 2.5: Control architecture

The highest hierarchical level connects the human operators and the robot. The operation interface is composed of a GUI, a keyboard and a mouse. The GUI contains a command input box. The operators are able to enter tasks into this box using the keyboard. A link connecting the PC in the laboratory and the host computer mounted on the robot will be provided by wireless LAN.

The planning level specifies the subtasks for the different subsystems of the robot such as hands, arms, mobile platform and head according to the task given by the operator. This can be achieved by an on-board industrial PC (host computer). Ethernet is used to interface directly between the host computer on the planning level and the PCs on the coordination level. Therefore this structure can avoid communication collision.

The coordination level employs four subsystems (a visual information processing part, a speech processing part, a motion control part, and a localization and obstacle avoidance part) by embedded PC/104 systems. The use of the multi-PC architecture speeds up the most run-time consuming parts. These PCs are connected to each other via the Ethernet. Sensors such as microphones, cameras, laser ranger finder, ultrasonic distance sensors and force/torque sensors are connected to the PCs on the coordination level directly. PC104a captures the visual images from the CCD cameras and then caluculates the target's position, the brightness of the target, etc. PC104b obtains and analyzes the speech signals captured by the microphones. PC104c generates all the motions of the different subsystems for the execution level. PC104d deals with simultaneous localization and mapping (SLAM), obstacle detection and collision avoidance by the fusion of ultrasonic sensors, laser ranger finder and the data from PC104a.

On the execution level, for each servo motor, the control and the sensor value acquisition are done with a DSP connected to a CPLD. Both the angular velocity of the motor and the angle of the driven axis will be measured. Further the motor current will be supervised and can be used for implementing torque control. CAN-bus is used to connect the components on this level and receive the requests from the coordination level.

2.2.4 Virtual Prototyping

2.2.4.1 Solid Model Creation

There are many excellent 3D-CAD softwares such as CATIA, Pro/Engineer and Solidworks. Solidworks has a standard, easy-to-use interface and compatibility with engineers' existing systems. Thus, in this thesis, Solidworks is employed to build a solid model for the assistant robot. After modeling, we save this 3D model as *.x_t document so that it can be imported into ADAMS software directly.

The model drawn in Solidworks is shown in Figure 2.6. Each part's size and shape of this robot are created as those of the actual component. To reduce the complexity of the model in ADAMS, only the mechanical parts are included in the Solidworks drawing. This structure is sufficient to reflect the physical properties of the mobile manipulator, which will be examined later.

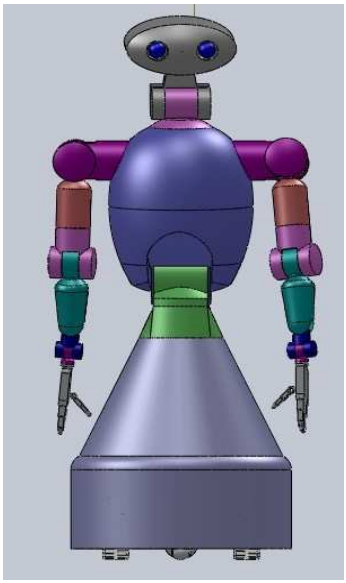


Figure 2.6: Solidworks drawing of mobile manipulator

2.2.4.2 Virtual Prototype Development

MSC.ADAMS is one of many computer programs for modeling and simulating multibody systems. In ADAMS a virtual prototype of a mechanism can be built and then be simulated. And it also allows users to import part geometry from CAD programs. The most popular file format to import into ADAMS is Parasolid (*.x_t) which increases the chances of receiving solids when importing geometry. Using Parasolid is convenient because one file contains all the geometry and ADAMS/Exchange creates a separate part for each solid. After the parts of the mechanical system are developed with Solidworks, we import this solid model which has already been saved as *.x_t document. All the constructional and technological features specified at the designing stage have found reflection in the mechanical virtual model. Figure 2.7 shows the virtual prototyping model in ADAMS.

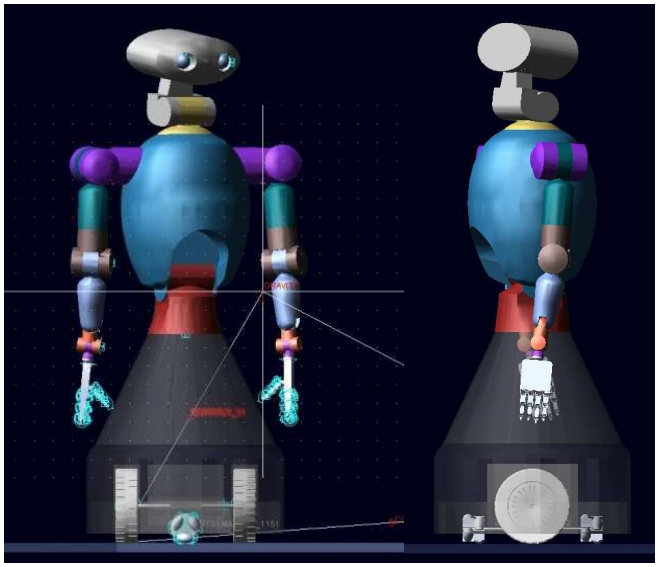


Figure 2.7: Virtual prototyping model in MSC.ADAMS

The elaborated ADAMS model takes several aspects into account, such as joint constraints, mass, inertial properties, contact constraints, friction, actuation forces and reference markers, which enable good approximation of the real robot behavior. First, each joint of the mechanism is defined with a particular motion. For example, the driving joint is specified as a revolute joint so the rotation associated with this joint specifies the rolling motion of the wheel. Secondly, according to the design parameters of the robot, material, mass and moment of inertia tensor should be defined in ADAMS. ADAMS calculates the mass and inertia of each rigid body based on its solid geometry and its material type. The default properties for the part can be modified by entering the properties we need. For each part a body-fixed reference frame with its origin at the center of mass is defined in ADAMS as a marker. Additionally other reference markers could be created if needed. Table 2.6 lists the main dynamic characteristics of the robot components for the proposed ADAMS model (excluding the hands). The symbols *Ixx*, *Iyy* and *Izz* are frequently used to express the moments of inertia of each link about its three axis passing through the link's center of mass. The products of inertia are given by *Ixy*, *Iyz* and *Izx*. Contact constraints and friction will be discussed in the next subsection.

Dexterous hand is a complex mechanism with multiple DOFs. It is an exciting field with still many unsolved problems in both mechanical design and control. It is necessary to create a separate 3D model of the robotic hand for research. Table 2.7 lists the main dynamic characteristics of the robotic fingers.

Table 2.6: Main inertial properties of robot

Part	Mass(kg)	Rotation inertia around center-of-mass ($kg \cdot mm^2$)			Products of inertia around center-of-mass ($kg \cdot mm^2$)		
		Ixx	*Iyy*	*Izz*	*Ixy*	*Iyz*	*Izx*
Head	2	1.452×10^4	1.196×10^4	1.177×10^4	0	0	0
Neck	2	3180.97	3020.63	2687.458	0	0	0
Upper body	10	1.608×10^5	2.419×10^5	2.096×10^5	0	0	0
Waist	6	2.112×10^4	2.118×10^4	1.87×10^4	0	0	0
Lower body	50	1.796×10^7	1.961×10^7	2.83×10^7	0	0	0
Arm1	2.477	9152.28	6142.9	6363.91	0	0	0
Arm2	1.192	4041.89	1086.44	4465.585	0	0	0
Arm3	0.57	1436.32	1133.86	939.69	0	0	0
Arm4	0.898	607.69	3698.59	3851.59	0	0	0
Wrist1	0.273	103.248	163.444	218.373	0	0	0
Wrist2	0.096	26.087	11.966	30.698	0	0	0
Wrist3(no fingers)	0.3	657.051	792.638	1326.87	0	0	0

Table 2.7: Summary of mass and inertia characteristics of finger

Part	Mass (kg)	Rotation inertia around center-of-mass (kg·mm^2)			Products of inertia around center-of-mass (kg·mm^2)		
		Ixx	*Iyy*	*Izz*	*Ixy*	*Iyz*	*Izx*
1	0.0024	0.127	0.149	0.134	0	0	0
2	0.005	0.301	0.646	0.698	0	0	0
3	0.005	0.301	0.646	0.698	0	0	0
4	0.01339	0.5466	0.681	0.717	0	0	0

Additionally, a numerical integrator is used and should be chosen for the simulation. The integrator is an algorithm that solves the differential equations of a dynamic problem over an interval of time during a simulation. The selected integrator for simulation is the GSTIFF. The GSTIFF integrator is the default in ADAMS environment since it provides good solutions for simulation of stiff mechanical models.

One has to specify simulation frequency, internal frequency and step size. Simulation frequency is the frequency of updating the graphic display, whereas internal frequency is closely related to the system under study. It represents the speed at which component states are changed. Internal frequency is a very important parameter in simulation setting and other solver settings must be set in accordance with the internal frequency.

Once the virtual prototype has been established, ADAMS checks the model and then runs simultaneous equations for kinematic, static, quasi-static, and dynamic simulations. The results are viewable as graph, data plots, reports, or colorful animations.

As a test to verify the reasonableness of the design, the right and left dexterous hands are modeled respectively. The simulation results in ADAMS show the effectiveness of the developed mechanism in grasping objects of various shape and size. Figure 2.8 demonstrates the virtual prototyping model of right hand in ADAMS. Figure 2.8(a) shows the initial configuration of the hand-object system. Figure 2.8(b) shows the system configuration when the hand is grasping the object.

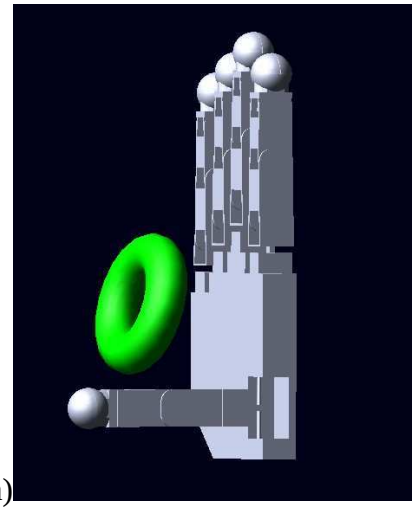
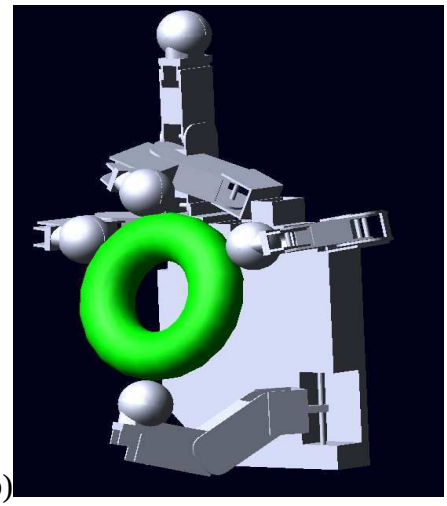
(a)  (b) 

Figure 2.8: Virtual prototyping model of dexterous hand in MSC.ADAMS (a) Initial configuration of hand-object system (b) Final configuration

2.2.4.3 Contact Modeling between Bodies in ADAMS

In this subsection it will be presented how the contacts are modeled in ADAMS. The connection between each wheel and the floor is modeled using a contact with impact function. The parameters of hardness of impact force, force exponent, damping, penetration depth and friction, etc., must be set. If the contact parameters are incorrectly specified, we will create a system which requires a lot of computational time to solve. For example, if we choose high stiffness and small damping, the system would oscillate on high frequency and would be difficult to solve. With the right parameters, the model behaves like a soft tire.

The constraints between the fingers and the object are also contacts with impact functions. Then the contact constraints need to be created between each finger and the object, and the parameters of hardness of impact force, force exponent, damping, penetration depth and friction, etc., also must be set.

The impact function is written out:

$$F_n = k\delta^e + step(\delta, 0, 0, d, C)\dot{\delta} \tag{2.1}$$

where F_n is the contact force, k denotes the stiffness, e represents the force exponent, δ is the distance between the contacting bodies, C is the damping coefficient and d denotes the penetration depth.

Equation (2.1) shows that the contact force defined in ADAMS is composed of two parts. An elastic component $k\delta^e$ acts like a nonlinear spring. The other one $step(\delta, 0, 0, d, C)\dot{\delta}$ is the damping force which is a function of the contact-collision velocity.

The coulomb friction, F_f, is calculated by multiplying a friction coefficient, μ_f, with the contact force, F_n.

$$F_f = \mu_f F_n \tag{2.2}$$

The friction coefficients between the wheels and the ground are 0.95 for the static condition and 0.8 for the dynamic condition; the friction coefficients between the fingertips and the object are 0.7 for the static condition and 0.6 for the dynamic condition.

2.2.4.4 Co-simulation Connection between ADAMS and Matlab/Simulink

After the mechanical part has been built, a co-simulation platform is set up to develop a control system for the complete robotic system. ADAMS has toolbox ADAMS/Controls that will prepare our model for connection with other software designed for control. One can choose between Matlab/Simulink and MSC.Easy5. In this thesis co-simulation with Matlab/Simulink is used. The detailed procedure is shown in Figure 2.9 as a flowchart. First, a trajectory planning system is built by Simulink. The feasibility of the motion generated in the first step is then verified through the collaborative simulation between Simulink and ADAMS. The feasible results will be used to the physical prototype.

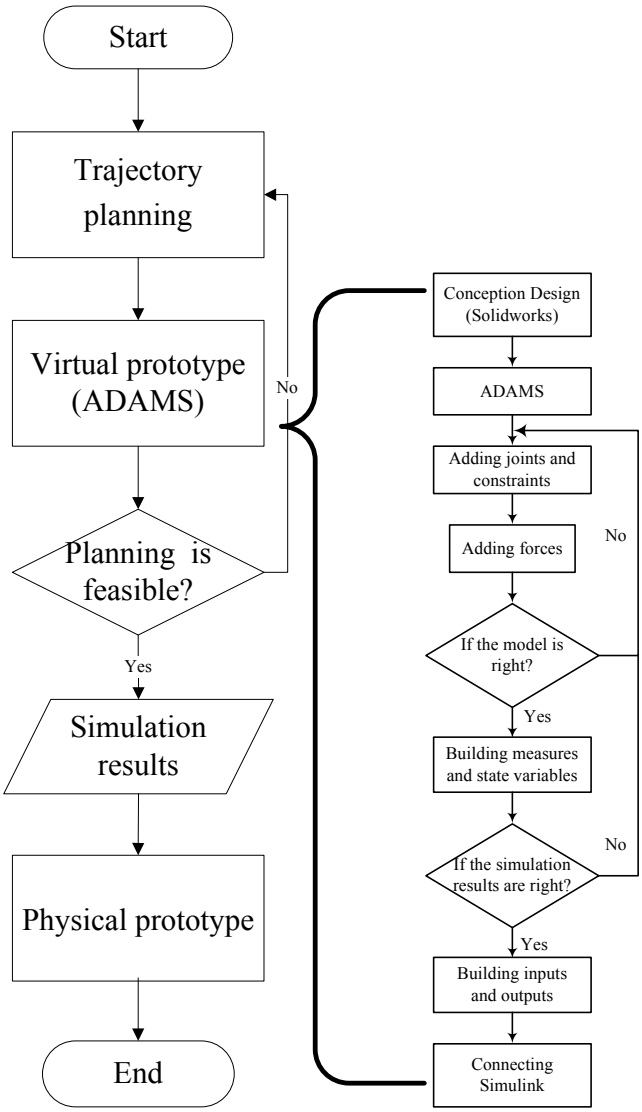


Figure 2.9: Flowchart of collaborative simulation

Below is a more detailed explanation of the procedure. First, the model built in Solidworks is imported into ADAMS. Then the joint constrains and the properties are added. Next, the measures and the state variables should be built. Simulations are created in ADAMS to verify the virtual prototype. The correctness of the ADAMS simulation results can be verified by comparison with the theoretical results. After defining the input and output variables, it can form a closed information loop between ADAMS and Matlab. Matlab reads the output information of the plant and calculates the control output while the ADAMS solver solves the system equations for the plant. ADAMS/Controls and Matlab communicate by passing the state variables back and forth (Figure 2.10). Our main goal is to control the joints to make the robot perform some human tasks. To achieve this goal, the joint torques are defined as the input variables; the joint angles and joint angular velocities are defined as the output state variables.

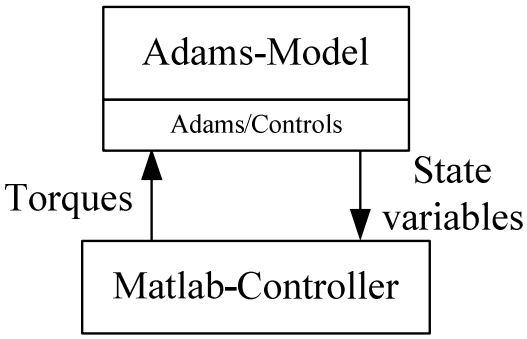


Figure 2.10: Information flowchart of collaborative simulation

The various steps involved in setting up a coordinated simulation between ADAMS and Simulink are:

- Loading ADAMS /Controls
- Defining Input and Output Variables
- Referencing Input Variables in the ADAMS Model
- Exporting the ADAMS Block
- Connecting the ADAMS Block and the CES Block in Simulink
- Running the Co-simulation

Note that the output step size will determine the communication interval between Simulink and ADAMS and must be the same.

Through the co-simulation we can validate the virtual prototype of the personal assistant robot. Then based on the virtual prototype, the efficiency of the complex motion planning algorithms and control schemes proposed in this thesis can be verified.

2.2.5 Workspace Computation

Exact computation of the workspace of robot manipulator is very important for its optimum design and application. The workspace of robot manipulator is defined as the set of points that can be reached by its end-effector and dependent on the joint angle ranges. The joint angle ranges of our robot have been defined in the mechanical design process so that the workspace can be computed. Figure 2.11(a) to Figure 2.11(d) show the workspace of a single arm including the motion of the waist when the mobile platform remains stationary from different views. Figure 2.11(e) and Figure 2.11(f) show the robot workspace of the dual-arm/waist system. The shape and volume of the workspace, which we most concern, are illustrated. This cartesian workspace forms a kind of capability map for the redundant arms of our assistant

robot. For example, the information can be used to search for valid grasps in regions where the hand can easily be placed by the robotic arm.

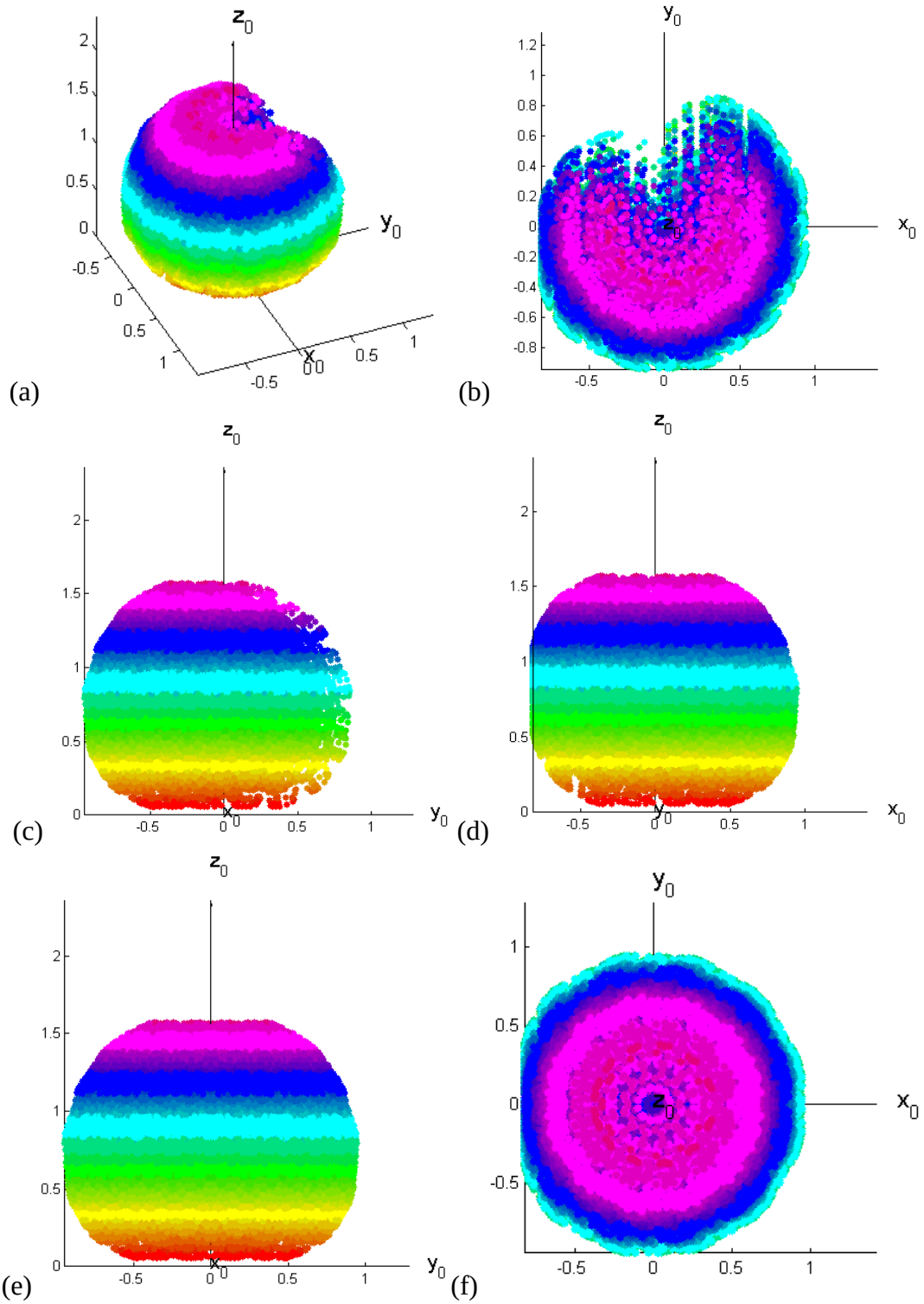


Figure 2.11: 3D workspace computed by Matlab (a) 3D view (b) *x-y* view (c) *y-z* view (d) *x-z* view (e) Shared workspace of two arms in *y-z* view (f) Shared workspace of two arms in *x-y* view

CHAPTER3 MODELING OF DUAL-HAND MOBILE MANIPULATOR

3.1 Mathematical Modeling

3.1.1 Introduction

For control creation in mobile manipulation tasks, the kinematic and dynamic models of the personal assistant robot are developed in this subchapter.

The kinematics involves determining the possible movements of the robot, without considering the forces and torques acting on the robot. Getting the kinematic model is the prerequisite needed to establish a dynamic model. The Denavit-Hartenberg representation [125], also called the D-H convention, describes a systematic way to develop the forward kinematics for rigid robots. This convention to develop reference frames is commonly used in robotic applications. There are two slightly different approaches to the convention: the so called standard D-H notation described in for example [126] and the Modified D-H form found in [127].

The response to the forces and torques is the subject of the dynamics. There are several fundamental methods for the formulation of equations of motion, such as Newton-Euler formulation [128], Euler-Lagrange principle [129], Kane's method [130], and Screw theory [131]. All the above mentioned approaches have their own advantages and disadvantages when applied to complex robots like mobile manipulators.

- Newton-Euler approach is one of the classical methods for dynamic formulation. Mathematically, the Newton-Euler equations lead to three translational equations of motion, and three equations determining the rotational motion of the rigid body. The Newton-Euler equations of any two free bodies are related through the constraint forces (which include both forces and torques) acting at their interface. So Newton-Euler is comprehensive in that a complete solution for all the forces and kinematic variables are obtained. But applying this method requires that force and torque balances are applied for each body taking in consideration every interactive and constraint force. As a result, it is inefficient due to redundant equations with more equations than unknowns.

- Euler-Lagrange formulation is another classical approach which is widely used for dynamic modeling. The Euler-Lagrange formulation selects generalized coordinates to eliminate the explicit use of constraint equations from the formulation (i.e., the constraint forces do not appear in the equations of motion). Typically the number of

generalized coordinates equals DOFs of a system. The major disadvantage of this method is the need to differentiate scalar energy functions (kinetic and potential energy). This is not much of a problem for small multibody systems, but becomes an efficiency problem for large multibody systems.

- Kane's formulation has also been used by many researchers for the development of equations of motion. Kane's method is sometimes referred to as Lagrange's form of D'Alembert's Principle. But Kane's method also uses generalized speeds, which allows motion constraints to be embedded. The big advantage in comparison to the Lagrange method is that there is no differentiation of energy functions which makes the other approach inefficient when it comes to large multibody systems.

- Screw theory is a general mathematical technique of representing the motion of rigid bodies in space. The method of dynamic analysis based on screw theory builds up dynamic equations by using the mathematical tools of twist-product-of-exponential formula.

In this subchapter, MDH and the Euler-Lagrange methodology have been chosen to lead our study. Two cases will be discussed separately: (1) without fingers and (2) with fingers.

3.1.2 Kinematic Modeling

In this section, the overall kinematic equations of the robot are discussed briefly. As described in the previous chapter, the personal assistant robot is built from a set of links, joined with revolute joints that can rotate the robot around a given axis. In the first case which does not take into account the dexterous hands, the robot's links and joints can be divided into five main parts: mobile platform, waist, left arm, right arm and head. As shown in Figure 3.1(a), three kinematic loops are formulated:

- Loop *a*: it contains platform, waist, right shoulder and right arm.
- Loop *b*: it contains platform, waist, left shoulder and left arm.
- Loop *c*: it contains platform, waist, neck and head.

Substituting a pair of dexterous hands for the end-effectors, in the second case, the complete robot can be divided into nine kinematic loops Loop R_1, Loop R_2, Loop R_3, Loop R_4, Loop L_1, Loop L_2, Loop L_3, Loop L_4 and Loop *c* as denoted in Figure 3.1(b). Figure 3.1 (b) shows that each independent loop contains a finger.

Note that the last two fingers (the ring and the little fingers) are mechanically coupled, in the following these two fingers are considered as one finger.

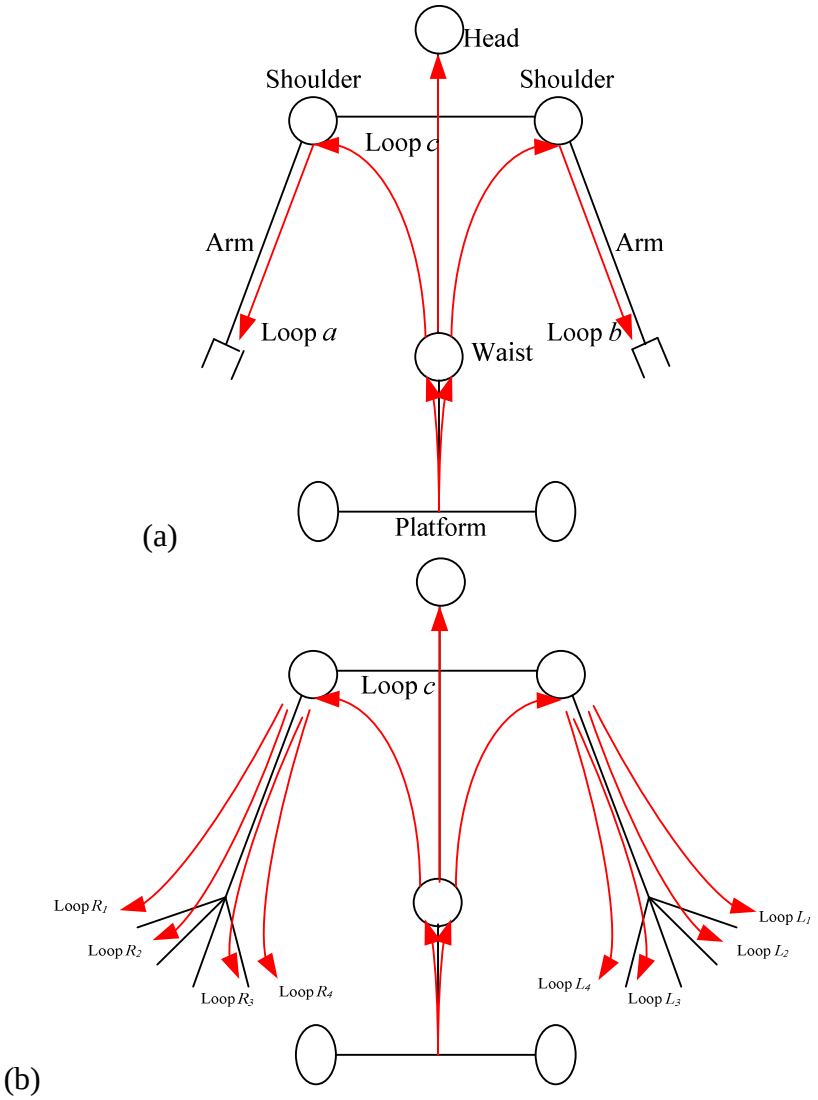


Figure 3.1: Kinematic loops (a) The simplified kinematic loops of the robot without fingers (b) The simplified kinematic loops of the robot with fingers

As can be seen, the personal assistant robot in this thesis has a tree-like structure with multi-DOF, and each finger can be treated as a branch on this tree. The complete kinematic model of the multi-fingered mobile manipulator can be obtained by modeling these separate loops. Due to the independence the kinematic models of all the loops can be derived by the same approach. Note that, in the following, Loop *c* will be neglected and not be simulated because it just supports the vision system. Using the virtual prototype, all the information needed to use can be obtained by creating the corresponding measures. So the vision system is not necessary for the simulation.

In this thesis, q denotes the vector of generalized coordinates of Loop *a* and Loop *b* and may be separated into four sets $q = \begin{bmatrix} q_v^T & q_w^T & q_r^T & q_l^T \end{bmatrix}^T \in \Re^p$. The subscripts *v*, *w*, *r* and *l*

represent the mobile platform, waist, right arm and left arm respectively. $q_v \in \Re^m$ describes the vector of generalized coordinates of the mobile platform, $q_w \in \Re^n$ is the vector of generalized coordinates of the waist, and q_r, $q_l \in \Re^a$ are the vectors of generalized coordinates of the right and left arms respectively. The total number of the generalized coordinates p is $m+n+2a$. The vector of generalized coordinates q' of Loop R_1, R_2, R_3, R_4, L_1, L_2, L_3 and L_4 may be divided into three sets $q' = \begin{bmatrix} q_v^T & q_b^T & q_f^T \end{bmatrix} \in \Re^{p'}$. The subscripts *v*, *b*, *f* denote the mobile platform, robot body (including waist and dual arms) and fingers respectively. $q_b = \begin{bmatrix} q_w^T & q_r^T & q_l^T \end{bmatrix}^T \in \Re^{n+2a}$ describes the vector of generalized coordinates of the waist and two arms, $q_f = \begin{bmatrix} q_{rf}^T & q_{lf}^T \end{bmatrix}^T$, q_{rf}, $q_{lf} \in \Re^b$, are the vectors of generalized coordinates of the right and left hands and the total number of the generalized coordinates $p' = m+n+2a+2b$.

In this section, first, the mobile platform, the manipulator parts and the dexterous hands are discussed separately. Then the kinematics of Loop R_1, R_2, R_3, R_4, L_1, L_2, L_3 and L_4 are derived using a similar method. Finally, the kinematics of a rigid body is discussed.

3.1.2.1 Mathematical Model of Differential Wheeled Platform

As described in the mechanical design process, the platform has two driving wheels (conventional wheels) and two passive supporting wheels (omni wheels). The two driving wheels are independently driven by the DC motors. The schematic of this differential wheeled robot is shown in Figure 3.2.

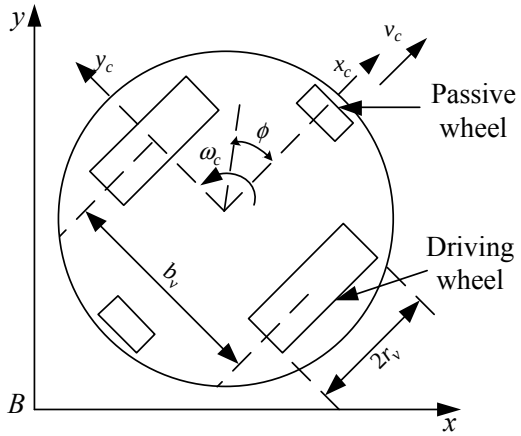


Figure 3.2: Schematic of differential wheeled robot

$[x_c \quad y_c]^T$ are the coordinates of the center of mass in the world coordinate system, ϕ is the heading angle of the platform measured from the *x*-axis of the world coordinate system, ω_c is the angular velocity of the mobile platform, b_v is the distance between the driving wheels and r_v is the radius of each driving wheel.

The nonholonomic constraint states that the robot can only move in the direction normal to the axis of the driving wheels, i.e., the mobile base satisfies the conditions of pure rolling and non slipping [19]:

$$\dot{y}_c \cos\phi - \dot{x}_c \sin\phi = 0 \tag{3.1}$$

Let the Lagrange coordinates of the mobile platform be $q_v = [x_c \quad y_c \quad \phi]^T$ which represent the position of the robot in the world coordinate system, the constraint can be written in the form:

$$A_v(q_v)\dot{q}_v = 0 \tag{3.2}$$

which is according to (3.1), where $A_v(q_v) = [-\sin\phi \quad \cos\phi \quad 0]$.

3.1.2.2 Forward Kinematics of Manipulator

As mentioned above, without the dexterous hands, the robot can be divided into three independent loops. For the body (excluding mobile platform), each loop is considered as a linkage, which is defined to be a set of attached rigid bodies. Let us take Loop *a* as an example. Each rigid body is referred to as a link, denoted by D. Let D_1, D_2, . . ., and D_{n+a} denote a set of $n+a$ links. The kinematic analysis mainly includes two sides, one is the forward kinematic analysis, and the other is the IK analysis. The forward kinematic analysis means that the location and pose of the linkage end point in a given reference coordinate system can be worked out with the given geometry parameters of the links and the variables of the joints. Let $Q = \begin{bmatrix} q_w^T & q_r^T \end{bmatrix}^T = [q_1 \quad q_2 \quad \ldots \quad q_{n+a}]^T$ denote the set of angles of rotational joints, then Q yields an acceptable position and orientation of the links in the chain by using the homogeneous transformation matrix. Given the matrix $g_{(i-1)i}$ expressing the difference between the coordinate frame of D_{i-1} and the coordinate frame of D_i, the application of

$g_{(i-1)i}$ transforms any point in D_{i-1} to the body frame of D_i. Repeating the above procedure, the end-effector location $\begin{bmatrix} x' & y' & z' \end{bmatrix}^T$ in the frame $\sum_0$ attached to the mobile base is determined by multiplying the transformation matrices and the end-effector location $\begin{bmatrix} x & y & z \end{bmatrix}^T \in D_{n+a}$:

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix}^T = g_{01}g_{12}...g_{(n+a-1)(n+a)} \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T = g_{0(n+a)} \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T \qquad (3.3)$$

where

$$g_{0(n+a)} = \begin{bmatrix} R_{0(n+a)} & P_{0(n+a)} \\ 0...0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \in \Re^{4\times4} \quad , \quad R_{0(n+a)} = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \in \Re^{3\times3} \quad \text{and}$$

$P_{0(n+a)} = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^T \in \Re^3$ give the rotation axis and the position in the frame $\sum_0$, respectively.

(3.3) represents the forward kinematics for positioning a linkage by a specific joint configuration. The MDH method is used to analyze the kinematics. This method is commonly used dealing with manipulator kinematics. Figure 3.3 shows the Modified form of Denavit-Hartenberg. Frame *i* has its origin along the axis of joint *i*.

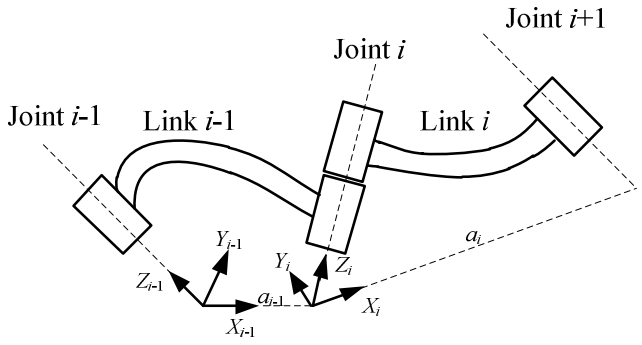


Figure 3.3: Modified form of Denavit-Hartenberg

According to the Modified form of Denavit-Hartenberg, the transformation matrix could be expressed by the following equation:

$$g_{(i-1)i} = Rot(x,\alpha_{i-1}) \times Trans(x,a_{i-1}) \times Rot(z,q_i) \times Trans(z,d_i)$$

$$g_{(i-1)i} = \begin{bmatrix} \cos q_i & -\sin q_i & 0 & a_{i-1} \\ \sin q_i \cos \alpha_{i-1} & \cos q_i \cos \alpha_{i-1} & -\sin \alpha_{i-1} & -d_i \sin \alpha_{i-1} \\ \sin q_i \sin \alpha_{i-1} & \cos q_i \sin \alpha_{i-1} & \cos \alpha_{i-1} & d_i \cos \alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

where

α_{i-1} = the angle between Z_{i-1} and Z_i measured about X_{i-1},
a_{i-1} = the distance from Z_{i-1} to Z_i measured along X_{i-1},
q_i = the angle between X_{i-1} and X_i measured about Z_i,
d_i = the distance from X_{i-1} to X_i measured along Z_i.

Figure 3.4 illustrates the idea of attaching frames of the mobile manipulator according to MDH. The frame $x_0 y_0 z_0$ is attached to the mobile base. The other frames are attached to each link.

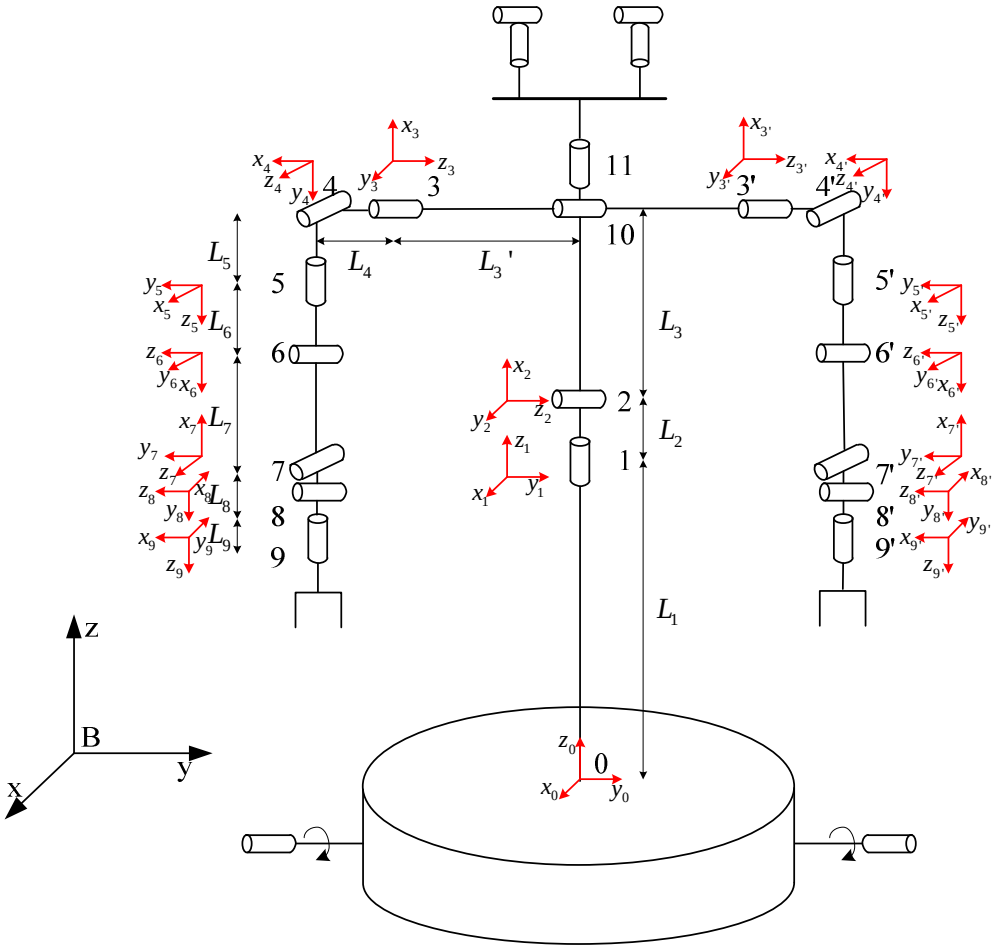


Figure 3.4: Coordinate frames attached to mobile manipulator

MDH parameters of all joints are depicted in Table 3.1.

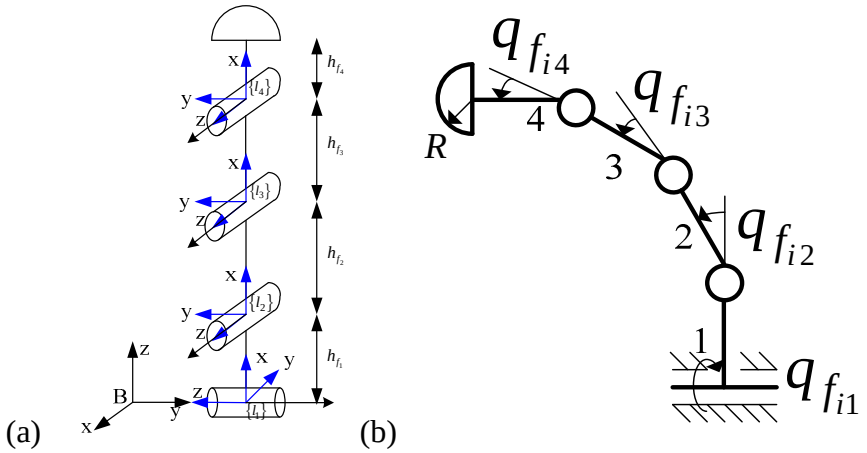
Table 3.1: MDH parameters of mobile base and manipulators

Link	α	a	q	d
0-1	0°	0	0°	$L_1 + L_2$
1-2	-90°	0	-90°	0
2-3	0°	L_3	0°	$-(L_3' + L_4)$
2-3’	0°	L_3	0°	$(L_3' + L_4)$
3-4/3’-4’	-90°	0	90°	0
4-5/4’-5’	-90°	0	-90°	$L_5 + L_6$
5-6/5’-6’	-90°	0	-90°	0
6-7/6’-7’	-90°	L_7	180°	0
7-8/7’-8’	-90°	$-L_8$	90°	0
8-9/8’-9’	-90°	0	-90°	0

Tasks to be performed by a manipulator are in task space, whereas actuators work in joint space. Task space includes orientation matrix and position vector. However, joint space is represented by joint angles. The conversion from the position and orientation of the end-effector in task space to the joint angles in joint space is called as IK problem. The inverse formula of (3.3) forms IK problem in which a set of joint angles need to be calculated corresponding to a given spatial constraint of the end-effector. In this thesis the numbers of joint DOFs of all loops are more than the number of spatial constraints, the system is redundant and there are multiple solutions.

3.1.2.3 Kinematics Analysis of Five-Fingered Hand

This part sets out the framework for the mathematical modeling of kinematic equations for the multi-fingered hand. Forward kinematics is used to obtain the fingertip position and orientation according to the finger joint angles.



(a) (b)

Figure 3.5: Finger linage (a) Definition of reference frames (b) Geometry of a finger module

According to the mechanical structure, each dexterous hand is composed of five fingers with total 16 DOFs and all of the fingers have the same size and shape. The kinematic structure of the finger mechanism is based on a simplification of a human finger.

As indicated in Figure 3.5, the mechanism consists of four joints. The angular positions of the first and second joints of finger *i* of the right (left) hand are defined by $q_{r(l)f_{i1}}$ and $q_{r(l)f_{i2}}$. The angular positions of the third and fourth joints are defined by $q_{r(l)f_{i4}}$ and $q_{r(l)f_{i4}}$, respectively. Neglecting the superscripts *r* and *l*, $q_{f_i} = \begin{bmatrix} q_{f_{i1}} & q_{f_{i2}} & q_{f_{i3}} & q_{f_{i4}} \end{bmatrix}^T$ represents the generalized coordinates of each finger. Table 3.2 shows the dimensions of the fingers.

Table 3.2: Kinematic parameters of fingers

Link lengths (mm)		Location of center of mass (mm)	
h_{f_1}	21.00	1	5
h_{f_2}	28.53	2	20
h_{f_3}	27.16	3	20
h_{f_4}	18.50	4	7.5

Considering the fingertip as the end point, the forward kinematics of the finger is also derived by MDH method. The parameters of all joints are depicted in Table 3.3.

Table 3.3: MDH parameters of fingers

Link	α	a	q	d
0-1	90°	0	90°	0
1-2	90°	h_{f_1}	0°	0
2-3	0°	h_{f_2}	0°	0
3-4	0°	h_{f_3}	0°	0

To solve the inverse problem for each finger, the first step to take is to measure the orientation and position of the fingertip.

3.1.2.4 Robot Kinematics with Dexterous Hands

Consider a bimanual manipulation system composed by a mobile platform, a two DOFs waist and a dual-arm/hand system. The direct kinematics can be computed by introducing a frame $\sum_B$ fixed with the ground, a frame, $\sum_0$, attached at the center of the mobile base, two frames, $\sum_r$ and $\sum_l$, attached at the bases of the right and left arms, respectively, and two frames, $\sum_{rp}$ and $\sum_{lp}$, attached to the palms of the right and left hands, respectively. Moreover, assuming that each arm ends with a robotic hand composed by N fingers, it is useful to introduce a frame $\sum_{rf_i}$ ($\sum_{lf_i}$), attached to the distal phalanx of finger i ($i = 1 \ldots N$) of the right (left) hand.

Consider Loop R_i. The pose of $\sum_{rf_i}$ with respect to the base frame $\sum_B$ can be represented by the well known (4 × 4) homogeneous transformation matrix $g_{B(rf_i)}$ ($R_{B(rf_i)}$, $P_{B(rf_i)}$), where $R_{B(rf_i)}$ is the (3×3) rotation matrix expressing the orientation of $\sum_{rf_i}$ with respect to the frame $\sum_B$ and $P_{B(rf_i)}$ is the (3 × 1) position vector of the origin of $\sum_{rf_i}$ with respect to $\sum_B$.

Hence, the direct kinematics can be expressed as:

$$g_{B(rf_i)} = g_{B0} g_{0r} g_{r(rp)} g_{(rp)rf_i} \tag{3.5}$$

where g_{B0} is the matrix relating the mobile base frame to the base frame, g_{0r} is the matrix at the basis of the right arm to the mobile base frame and depends on the waist joint vector, $g_{r(rp)}$ is the matrix relating the right palm frame to the base frame of the right arm and depends on the joint vector of the right arm, and $g_{(rp)rf_i}$ is the matrix relating the frame attached to the distal phalanx of finger *i* to the palm frame of the right hand and depends on the finger joint vector q_{rf_i}, where the fingers are assumed to be identical. An equation similar to (3.5) holds for the left hand fingers (Loop L_i), with subscript *l* in place of subscript *r*.

Due to the branched structure of the mobile manipulator, the motions of both the right and left arms are independent. Therefore, the kinematics of the right and left hands can be considered separately. Hence, in the sequel, the superscripts *r* and *l* will be omitted and will be used explicitly only when it is required to distinguish between the right and the left arm.

The differential kinematic equation relating the joint velocities to the velocity of finger frame $\sum_{f_i}$ can be written as:

$$V_{f_i} = J_{f_i} \dot{q}_i' \tag{3.6}$$

where $q_i' = \begin{bmatrix} q_v^T & q_b^T & q_{f_i}^T \end{bmatrix}^T$ and J_{f_i} is the Jacobian matrix of the mobile part, the waist, the arm, ending with the i_{th} finger.

Therefore, the differential kinematic equation of the complete mobile base-waist-arm-hand system can be written in the form:

$$V_f = J_f \dot{q}' \tag{3.7}$$

where $V_f = \begin{bmatrix} V_{f_1}^T & \dots & V_{f_N}^T \end{bmatrix}^T$, $q' = \begin{bmatrix} q_v^T & q_b^T & q_{f_1}^T & \dots & q_{f_N}^T \end{bmatrix}^T$ and J_f is the Jacobian matrix of the overall mobile base-waist-arm-hand system.

3.1.2.5 Kinematic Description of Object

A rigid body that is free to move in space has six DOFs. This can be described by an open kinematic chain with six joints (three prismatic joints and three revolute joints). To describe

the posture of the body we consider ZYZ Euler rotation. Similar to the previous case it is easy to obtain the direct kinematics. The homogeneous matrix which describes the posture of the object (O) in space relative to the fixed reference frame $\sum_B$ depends on the values of the local parameterization $X_o = \begin{bmatrix} x & y & z & \alpha & \beta & \gamma \end{bmatrix}^T$. $P_{Bo} = \begin{bmatrix} x & y & z \end{bmatrix}^T \in \Re^3$ is the object position relative to $\sum_B$ and $\begin{bmatrix} \alpha & \beta & \gamma \end{bmatrix}^T$ represents the ZYZ angles. The rotation axis of the object in the frame $\sum_B$ is denoted by R_{Bo} which is expressed as following:

$$\begin{aligned} R_{Bo} &= Rot(z,\alpha)Rot(y,\beta)Rot(z,\gamma) \\ &= \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{3.8}$$

3.1.3 Dynamic Modeling

The dynamics of mobile manipulators subject to kinematic constraints can be obtained using the Lagrangian approach in the form [115]:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(q) + A^T(q)\lambda + \tau_d = E(q)\tau \tag{3.9}$$

where *r* kinematic constraints are described by:

$$A(q)\dot{q} = 0, \tag{3.10}$$

$q \in \Re^p$ denotes the p generalized coordinates, $M(q) \in \Re^{p\times p}$ is a symmetric and positive definite inertia matrix, $C(q,\dot{q}) \in \Re^{p\times p}$ is a centripetal and Coriolis matrix, $F(q,\dot{q}) \in \Re^p$ is a friction and gravity vector, $A(q) \in \Re^{r\times p}$ represents a constraint matrix, $\lambda \in \Re^r$ is the Lagrange multiplier vector which denotes the constraint forces, $\tau_d \in \Re^p$ denotes unknown bounded disturbances including unstructured dynamics, $E(q) \in \Re^{p\times(p-r)}$ is the input transformation matrix, and $\tau \in \Re^{p-r}$ is a torque input vector.

According to the standard matrix theory, there exists a full rank matrix $S(q) \in \Re^{p\times(p-r)}$ formed by $p-r$ columns that span the null space of $A(q)$ defined in (3.10). i.e..

$$S^T(q)A^T(q) = 0 \tag{3.11}$$

As mentioned above, the generalized coordinate q of Loop a and Loop b is separated into four sets $q = \begin{bmatrix} q_v^T & q_w^T & q_r^T & q_l^T \end{bmatrix}^T$. We can find an auxiliary vector $\zeta = \begin{bmatrix} v^T & \dot{q}_w^T & \dot{q}_r^T & \dot{q}_l^T \end{bmatrix}^T \in \Re^{(p-r)}$ such that,

$$\dot{q} = S(q)\zeta \quad \text{or} \quad \dot{q}_v = S_v(q)v \tag{3.12}$$

v can be defined as $\begin{bmatrix} \dot{\theta}_r & \dot{\theta}_l \end{bmatrix}^T$ or $\begin{bmatrix} v_c & \omega_c \end{bmatrix}^T$, where $\dot{\theta}_r$ and $\dot{\theta}_l$ are the angular velocities of the right and left wheels, respectively; v_c and ω_c are the linear and angular velocities of the mobile base, respectively.

The time derivate of (3.12) is:

$$\ddot{q} = S(q)\dot{\zeta}(t) + \dot{S}(q)\zeta(t) \tag{3.13}$$

Multiplying both sides of (3.9) by S^T and rewriting it as:

$$\bar{M}\dot{\zeta} + \bar{C}\zeta + \bar{F} + \bar{\tau}_d = \bar{E}\tau \tag{3.14}$$

where $\bar{M} = S^T M S$, $\bar{C} = S^T C S + S^T M \dot{S}$, $\bar{F} = S^T F$, $\bar{\tau}_d = S^T \tau_d$, and $\bar{E} = S^T E$.

Property 3.1: The inertia matrix $\bar{M}(q)$ is symmetric and positive definite.

Property 3.2: Matrix $\dot{\bar{M}}(q) - 2\bar{C}(q, \dot{q})$ is skew-symmetric.

Property 3.3: The time-varying unstructured disturbance term τ_d is bounded by $\sup_t \| \tau_d \| \leq \tau_N$.

Using the separated generalized coordinates $q = \begin{bmatrix} q_v^T & q_w^T & q_r^T & q_l^T \end{bmatrix}^T$, (3.9) can be expressed as:

$$\begin{bmatrix} M_v & M_{vw} & M_{vr} & M_{vl} \\ M_{wv} & M_w & M_{wr} & M_{wl} \\ M_{rv} & M_{rw} & M_r & 0 \\ M_{lv} & M_{lw} & 0 & M_l \end{bmatrix} \begin{bmatrix} \ddot{q}_v \\ \ddot{q}_w \\ \ddot{q}_r \\ \ddot{q}_l \end{bmatrix} + \begin{bmatrix} C_v & C_{vw} & C_{vr} & C_{vl} \\ C_{wv} & C_w & C_{wr} & C_{wl} \\ C_{rv} & C_{rw} & C_r & 0 \\ C_{lv} & C_{lw} & 0 & C_l \end{bmatrix} \begin{bmatrix} \dot{q}_v \\ \dot{q}_w \\ \dot{q}_r \\ \dot{q}_l \end{bmatrix} + \begin{bmatrix} F_v \\ F_w \\ F_r \\ F_l \end{bmatrix} + \begin{bmatrix} A_v^T(q_v)\lambda \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \tau_{dv} \\ \tau_{dw} \\ \tau_{dr} \\ \tau_{dl} \end{bmatrix} = \begin{bmatrix} E_v\tau_v \\ \tau_w \\ \tau_r \\ \tau_l \end{bmatrix} \tag{3.15}$$

Equations (3.9) and (3.15) represent the dynamic equations of the mobile manipulator subject to kinematic constraints without the dexterous hands.

The dexterous hands can be mounted on the ends of the robotic arms directly. Based on the discussion given above, the complete dynamic model with a pair of dexterous hands can be expressed as:

$$\begin{bmatrix} M_v & M_{vb} & M_{vf} \\ M_{bv} & M_b & M_{bf} \\ M_{fv} & M_{fb} & M_f \end{bmatrix} \begin{bmatrix} \ddot{q}_v \\ \ddot{q}_b \\ \ddot{q}_f \end{bmatrix} + \begin{bmatrix} C_v & C_{vb} & C_{vf} \\ C_{bv} & C_b & C_{bf} \\ C_{fv} & C_{fb} & C_f \end{bmatrix} \begin{bmatrix} \dot{q}_v \\ \dot{q}_b \\ \dot{q}_f \end{bmatrix} + \begin{bmatrix} F_v \\ F_b \\ F_f \end{bmatrix} + \begin{bmatrix} A_v^T(q_v)\lambda \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \tau_{dv} \\ \tau_{db} \\ \tau_{df} \end{bmatrix} = \begin{bmatrix} E_v\tau_v \\ \tau_b \\ \tau_f \end{bmatrix} \tag{3.16}$$

Considering the nonholonomic constraints and their time-derivatives, the dynamics can be expressed as:

$$\begin{bmatrix} S_v^T M_v S_v & S_v^T M_{vb} & S_v^T M_{vf} \\ M_{bv} S_v & M_b & M_{bf} \\ M_{fv} S_v & M_{fb} & M_f \end{bmatrix} \begin{bmatrix} \dot{v} \\ \ddot{q}_b \\ \ddot{q}_f \end{bmatrix} + \begin{bmatrix} S_v^T M_v \dot{S}_v + S_v^T C_v S_v & S_v^T C_{vb} & S_v^T C_{vf} \\ M_{bv}\dot{S}_v + C_{bv} S_v & C_b & C_{bf} \\ M_{fv}\dot{S}_v + C_{fv} S_v & C_{fb} & C_f \end{bmatrix} \begin{bmatrix} v \\ \dot{q}_b \\ \dot{q}_f \end{bmatrix} + \begin{bmatrix} S_v^T F_v \\ F_b \\ F_f \end{bmatrix} + \begin{bmatrix} S_v^T \tau_{dv} \\ \tau_{db} \\ \tau_{df} \end{bmatrix} = \begin{bmatrix} S_v^T E_v \tau_v \\ \tau_b \\ \tau_f \end{bmatrix} \tag{3.17}$$

where the subscripts *v*, *b*, *f* denote the mobile platform, the body including the waist and the arms, and the fingers respectively.

Similarly, the complete dynamic equation can also be described in a compact form:

$$\bar{M}'\dot{\xi}+\bar{C}'\xi+\bar{F}'+\bar{\tau}_d'=\bar{E}'\tau' \tag{3.18}$$

where

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

$$\bar{\tau}_d'=\begin{bmatrix} S_v^T \tau_{dv} \\ \tau_{db} \\ \tau_{df} \end{bmatrix},\ \bar{E}'\tau'=\begin{bmatrix} S_v^T E_v \tau_v \\ \tau_b \\ \tau_f \end{bmatrix},\ \xi=\begin{bmatrix} v \\ \dot{q}_b \\ \dot{q}_f \end{bmatrix}.$$

In this section, first, the dynamics of the mobile platform is discussed. Secondly, the dynamics of the manipulators mounted on the mobile platform without the hands is presented. Then the dynamics of the dexterous hand is given separately. Finally, the dynamics of a rigid body is discussed.

3.1.3.1 Dynamic Model of Mobile Platform

Let us consider the first *m*-equations of (3.15):

$$\begin{aligned} E_v\tau_v = {} & M_v\ddot{q}_v + M_{vw}\ddot{q}_w + M_{vr}\ddot{q}_r + M_{vl}\ddot{q}_l + C_v\dot{q}_v + C_{vw}\dot{q}_w \\ & + C_{vr}\dot{q}_r + C_{vl}\dot{q}_l + F_v + A_v^T(q_v)\lambda + \tau_{dv} \end{aligned} \tag{3.19}$$

Multiplying both sides of (3.19) by S_v^T to eliminate the constraint forces. We obtain:

$$\begin{aligned} S_v^T E_v\tau_v = {} & S_v^T M_v\ddot{q}_v + S_v^T M_{vw}\ddot{q}_w + S_v^T M_{vr}\ddot{q}_r + S_v^T M_{vl}\ddot{q}_l + S_v^T C_v\dot{q}_v \\ & + S_v^T C_{vw}\dot{q}_w + S_v^T C_{vr}\dot{q}_r + S_v^T C_{vl}\dot{q}_l + S_v^T F_v + S_v^T\tau_{dv} \end{aligned} \tag{3.20}$$

Substituting both (3.12) and (3.13) into (3.20) yields:

$$\begin{aligned} S_v^T E_v\tau_v = {} & S_v^T M_v S_v\dot{v} + S_v^T M_v\dot{S}_v v + S_v^T M_{vw}\ddot{q}_w + S_v^T M_{vr}\ddot{q}_r + S_v^T M_{vl}\ddot{q}_l \\ & + S_v^T C_v S_v v + S_v^T C_{vw}\dot{q}_w + S_v^T C_{vr}\dot{q}_r + S_v^T C_{vl}\dot{q}_l + S_v^T F_v + S_v^T\tau_{dv} \end{aligned} \tag{3.21}$$

Let us rewrite (3.21) in a compact form as:

$$\bar{M}_v\dot{v} + \bar{C}_v v + f_v = \bar{\tau}_v \tag{3.22}$$

where $\bar{M}_v = S_v^T M_v S_v$, $\bar{C}_v = S_v^T M_v\dot{S}_v + S_v^T C_v S_v$ and

$$\overline{\tau}_v = S_v^T E_v \tau_v \tag{3.23}$$

$$f_v = S_v^T (M_{vw}\ddot{q}_w + M_{vr}\ddot{q}_r + M_{vl}\ddot{q}_l + C_{vw}\dot{q}_w + C_{vr}\dot{q}_r + C_{vl}\dot{q}_l + F_v + \tau_{dv}) \tag{3.24}$$

f_v consists of the gravitational and friction force vector F_v, the disturbances on the mobile base τ_{dv} (e.g., terrain disturbance force), and the dynamics interaction with the mounted manipulator arms ($M_{vw}\ddot{q}_w + M_{vr}\ddot{q}_r + M_{vl}\ddot{q}_l + C_{vw}\dot{q}_w + C_{vr}\dot{q}_r + C_{vl}\dot{q}_l$) which has been shown to have significant effect on the base motion, thus it needs to be compensated for [61].

The dynamic model of the mobile platform is constructed using the Lagrange equations expressed by:

$$\begin{aligned} &\frac{d}{dt}\frac{\partial L_r}{\partial \dot{q}_{vi}} - \frac{\partial L_r}{\partial q_{vi}} + A_v^T(q_v)\lambda_n = F_i,\ i = 1 \cdots m \\ &F = \begin{bmatrix} f_x & f_y & f_\phi \end{bmatrix}^T \end{aligned} \tag{3.25}$$

where q_v are the Lagrange coordinates of the mobile platform, F is the Lagrange force vector, λ_n is the vector of the Lagrange multipliers and L_r is the function of the generalized coordinates which is defined by $L_r = K_r - P_r$.

K_r and P_r are the total kinetic energy and potential energy of the robotic system respectively, defined as:

$$K_r = \frac{1}{2}m\dot{x}_c^2 + \frac{1}{2}m\dot{y}_c^2 + \frac{1}{2}I_c\omega_c^2 + \frac{1}{2}I_{wl}\dot{\theta}_l^2 + \frac{1}{2}I_{wr}\dot{\theta}_r^2 + K,\ P_r = 0 + P \tag{3.26}$$

where $\begin{bmatrix} \dot{x}_c & \dot{y}_c \end{bmatrix}^T$ are the velocities of the center of mass along *x* and *y* axis in the world coordinate system; $m = m_c + 2m_w$; m_c is the mass of the rigid platform without the wheels; m_w is the mass of each driving wheel; I_c is the moment of inertia of the platform along *z* axis; I_{wl} is the moment of inertia of the left wheel along wheel axis; I_{wr} is the moment of inertia of the right wheel along wheel axis; K and P denote the kinetic energy and the potential energy of the robot excluding the mobile part and will be discussed in detail in the next subsection. Note that the mobile platform does not affect the potential energy of the whole robot.

If there is no mounted manipulator arm, the dynamic coupling term transmitted from the manipulator in (3.22) vanishes. If v is defined as $\begin{bmatrix}\dot{\theta}_r & \dot{\theta}_l\end{bmatrix}^T$, by ignoring friction, using MATLAB the definitions of the model parameters in (3.22) can be obtained as:

```
Mv=[(Iw + (rv^2*((mw*bv^2)/2 + Ic  ))/bv^2 +
(rv*cos((phi)/(bv))*cos(phi)*(rv)*(m))/4 +
(rv*sin((phi)/(bv))*sin(phi)*(rv)*(m))/4),((rv*cos((phi)/(bv))*cos(phi)*(rv
)*(m))/4 - (rv^2*((mw*bv^2)/2 + Ic  ))/bv^2 +
(rv*sin((phi)/(bv))*sin(phi)*(rv)*(m))/4);
    ((rv*cos((phi)/(bv))*cos(phi)*(rv)*(m))/4 - (rv^2*((mw*bv^2)/2 +
Ic ))/bv^2 + (rv*sin((phi)/(bv))*sin(phi)*(rv)*(m))/4),(Iw +
(rv^2*((mw*bv^2)/2 + Ic ))/bv^2 + (rv*cos((phi)/(bv))*cos(phi)*(rv)*(m))/4
+ (rv*sin((phi)/(bv))*sin(phi)*(rv)*(m))/4)];
Cv=[- ((rv^2*cos((phi)/(bv))*sin(phi)*omega*m)/4 -
(rv^2*sin((phi)/(bv))*cos(phi)*omega*m)/4), -
((rv^2*cos((phi)/(bv))*sin(phi)*omega*m)/4 -
(rv^2*sin((phi)/(bv))*cos(phi)*omega*m)/4);
     - ((rv^2*cos((phi)/(bv))*sin(phi)*omega*m)/4 -
(rv^2*sin((phi)/(bv))*cos(phi)*omega*m)/4), -
((rv^2*cos((phi)/(bv))*sin(phi)*omega*m)/4 -
(rv^2*sin((phi)/(bv))*cos(phi)*omega*m)/4)];
```

$$E_v(q_v)=\frac{1}{r_v}\begin{bmatrix}\cos\phi & \cos\phi\\ \sin\phi & \sin\phi\\ \frac{b_v}{2} & -\frac{b_v}{2}\end{bmatrix},\ F_v(q_v,\dot{q}_v)=0,$$

$$A_v^T(q_v)=\begin{bmatrix}-\sin\phi\\ \cos\phi\\ 0\end{bmatrix},\ S_v(q_v)=\begin{bmatrix}\frac{r_v}{2}\cos\phi & \frac{r_v}{2}\cos\phi\\ \frac{r_v}{2}\sin\phi & \frac{r_v}{2}\sin\phi\\ \frac{r_v}{b_v} & -\frac{r_v}{b_v}\end{bmatrix} \tag{3.27}$$

where omega represents ω_c and phi denotes ϕ; $I_{wl}=I_{wr}=I_w$.

The steering system (3.12) has the form of:

$$\begin{bmatrix}\dot{x}_c\\ \dot{y}_c\\ \omega_c\end{bmatrix}=\begin{bmatrix}\frac{r_v}{2}\cos\phi & \frac{r_v}{2}\cos\phi\\ \frac{r_v}{2}\sin\phi & \frac{r_v}{2}\sin\phi\\ \frac{r_v}{b_v} & -\frac{r_v}{b_v}\end{bmatrix}\begin{bmatrix}\dot{\theta}_r\\ \dot{\theta}_l\end{bmatrix} \tag{3.28}$$

3.1.3.2 Dynamic Model of Manipulators Mounted on Mobile Platform

Combined mobile platform-manipulator system is a highly nonlinear and coupled system. Let us take the waist for example. The waist is in both Loop *a* and Loop *b*. Therefore, all the links in Loop *a* and Loop *b* will affect the performance of the waist.

Consider the second *n*-equations of (3.15):

$$\begin{aligned}\tau_w = {} & M_{wv}\ddot{q}_v + M_w\ddot{q}_w + M_{wr}\ddot{q}_r + M_{wl}\ddot{q}_l + C_{wv}\dot{q}_v \\ & + C_w\dot{q}_w + C_{wr}\dot{q}_r + C_{wl}\dot{q}_l + F_w + \tau_{dw}\end{aligned} \tag{3.29}$$

Rearranging (3.29) as follows:

$$M_w\ddot{q}_w + C_w\dot{q}_w + f_w = \tau_w \tag{3.30}$$

(3.30) represents the dynamic equation of the waist:

$$f_w = M_{wv}\ddot{q}_v + M_{wr}\ddot{q}_r + M_{wl}\ddot{q}_l + C_{wv}\dot{q}_v + C_{wr}\dot{q}_r + C_{wl}\dot{q}_l + F_w + \tau_{dw} \tag{3.31}$$

f_w consists of the manipulator dynamics $M_{wr}\ddot{q}_r + M_{wl}\ddot{q}_l + C_{wr}\dot{q}_r + C_{wl}\dot{q}_l$, the dynamic interaction with the mobile base $M_{wv}\ddot{q}_v + C_{wv}\dot{q}_v$, the gravitational and friction force vector F_w, and the disturbances on the waist τ_{dw}.

In the following, the dynamic model of the manipulators without dexterous hands is established in a systematic approach.

The dynamic model of the robot body is constructed using the Lagrange equations expressed by:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i,\ i = 1 \cdots n + 2a \tag{3.32}$$

where q_i is the i_{th} joint angle, $\dot{q}_i$ indicates the i_{th} joint velocity, τ_i is the i_{th} joint torque and $L = K - P$. The total kinetic energy K and the potential energy P are defined by:

$$K = \sum_{i=1}^{n+2a} K_i,\ P = \sum_{i=1}^{n+2a} P_i \tag{3.33}$$

where K_i and P_i are the kinetic and potential energy of link i respectively.

If the augmented link i is defined as the combination of link i and motor i, then the kinetic energy is given by:

$$K_t = K + K_a = \frac{1}{2}\sum_{i=1}^{n+2a} tr\left[\sum_{j=1}^{i}\sum_{k=1}^{i}\frac{\partial g_{Bi}}{\partial q_j} I_i \frac{\partial g_{Bi}^T}{\partial q_k}\dot{q}_j\dot{q}_k\right] + \frac{1}{2}\sum_{i=1}^{n+2a} I_{ai}k_{ri}^2\dot{q}_i^2 \tag{3.34}$$

where k_{ri} is the reduction gear ratio of joint i, I_i represents pseudo-inertia matrix of link i dependent on the mass distribution of link i and is expressed in the i_{th} frame, and I_{ai} denotes the rotational inertia of motor i.

Figure 3.6 illustrates the relationship between a joint and a link.

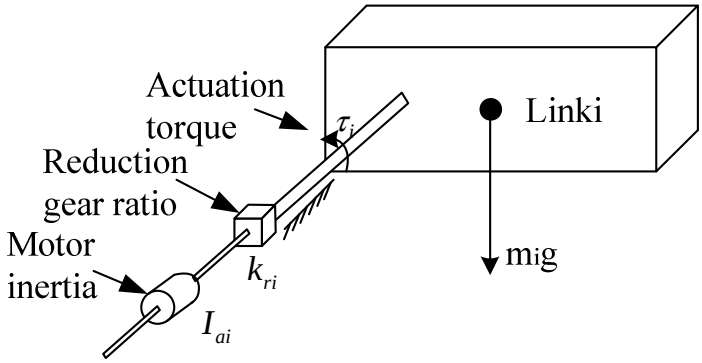


Figure 3.6: Relationship between joint and link

Let the homogenous transformation matrix from the base frame of the manipulator to the inertial frame $\sum_B$ be denoted by $g_{B0}(q_v)$, the transformation matrix, g_{Bi} from the i_{th} frame of the manipulator which is mounted on the platform to the inertial frame $\sum_B$ is given by:

$$g_{Bi} = g_{B0}g_{01}g_{12}\cdots g_{(i-1)i} \tag{3.35}$$

Potential energy of the body system is:

$$P = \sum_{i=1}^{n+2a} P_i = -\sum_{i=1}^{n+2a} m_i g(g_{Bi})\bar{r}_{i,ci}^{\,i} \tag{3.36}$$

where m_i is the mass of link i, g is the gravity row vector expressed in base frame $\sum_B$, and $\bar{r}_i^{\,i} = \begin{bmatrix}\bar{x}_i & \bar{y}_i & \bar{z}_i & 1\end{bmatrix}^T$ represents the position of the center of mass of link i and expressed in the i_{th} frame.

Then the generalized torque vector in the i_{th} joint is obtained:

$$\tau_i = \sum_{j=i}^{n+2a}\sum_{k=1}^{j} tr\left(\frac{\partial g_{Bj}}{\partial q_k} I_j \frac{\partial g_{Bj}^T}{\partial q_i}\right)\ddot{q}_k + \sum_{j=i}^{n+2a}\sum_{k=1}^{j}\sum_{m=1}^{j} tr\left(\frac{\partial^2 g_{Bj}}{\partial q_k \partial q_m} I_j \frac{\partial g_{Bj}^T}{\partial q_i}\right)\dot{q}_k\dot{q}_m - \sum_{j=i}^{n+2a} m_j g \frac{\partial g_{Bj}}{\partial q_i}\bar{r}_{j,cj}^{\,j} + I_{ai}k_{ri}^2\ddot{q}_i \tag{3.37}$$

For a rotary joint, equation (3.4) has shown:

$$g_{(i-1)i} = \begin{bmatrix} \cos q_i & -\sin q_i & 0 & a_{i-1} \\ \sin q_i \cos\alpha_{i-1} & \cos q_i \cos\alpha_{i-1} & -\sin\alpha_{i-1} & -d_i \sin\alpha_{i-1} \\ \sin q_i \sin\alpha_{i-1} & \cos q_i \sin\alpha_{i-1} & \cos\alpha_{i-1} & d_i \cos\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

And its derivative is:

$$\frac{\partial g_{(i-1)i}}{\partial q_i} = \begin{bmatrix} -\sin q_i & -\cos q_i & 0 & 0 \\ \cos q_i \cos\alpha_{i-1} & -\sin q_i \cos\alpha_{i-1} & 0 & 0 \\ \cos q_i \sin\alpha_{i-1} & -\sin q_i \sin\alpha_{i-1} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.38}$$

From (3.4) and (3.38), the derivative can be rewritten and simplified as:

$$\begin{aligned} \frac{\partial g_{(i-1)i}}{\partial q_i} &= g_{(i-1)i} Q_i \\ &= \begin{bmatrix} \cos q_i & -\sin q_i & 0 & a_{i-1} \\ \sin q_i \cos\alpha_{i-1} & \cos q_i \cos\alpha_{i-1} & -\sin\alpha_{i-1} & -d_i \sin\alpha_{i-1} \\ \sin q_i \sin\alpha_{i-1} & \cos q_i \sin\alpha_{i-1} & \cos\alpha_{i-1} & d_i \cos\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned} \tag{3.39}$$

Using (3.39), (3.37) is rearranged as follows:

$$\begin{aligned} \tau_i &= \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} \\ &= \sum_{j=i}^{n+2a}\sum_{k=1}^{j} tr(U_{jk} I_j U_{ji}^{\ T})\ddot{q}_k + \sum_{j=i}^{n+2a}\sum_{k=1}^{j}\sum_{m=1}^{j} tr\left(\frac{\partial U_{jk}}{\partial q_m} I_j U_{ji}^{\ T}\right)\dot{q}_k\dot{q}_m \\ &\quad - \sum_{j=i}^{n+2a} m_j g U_{ji}\bar{r}_{j,cj}^{\,j} + I_{ai}k_{ri}^2\ddot{q}_i \end{aligned} \tag{3.40}$$

where the effect of the motion of joint *j* on all the points on link *i* is as following:

$$U_{ij} \equiv \frac{\partial g_{0i}}{\partial q_j} = \begin{cases} g_{0j} Q_j g_{ji} & for \quad j \le i \\ 0 & for \quad j > i \end{cases} \tag{3.41}$$

The interaction effect of the motion of joint *j* and joint *k* on all the points on link *i* is given by:

$$\frac{\partial U_{ij}}{\partial q_k} \equiv U_{ijk} = \begin{cases} g_{0j} Q_j g_{jk} Q_k g_{ki} & i \ge k \ge j \\ g_{0k} Q_k g_{kj} Q_j g_{ji} & i \ge j \ge k \\ 0 & i < j \ \text{ or } \ i < k \end{cases} \tag{3.42}$$

3.1.3.3 Multi-Fingered Hand Modeling

(a) Dynamics of single finger

As discussed above, each finger can be represented by an open kinematic chain with some links, so in a manner entirely similar to the case of a manipulator its dynamic equation is as:

$$M_{f_i}(q_{f_i})\ddot{q}_{f_i} + C_{f_i}(q_{f_i}, \dot{q}_{f_i})\dot{q}_{f_i} + F_{f_i}(q_{f_i}, \dot{q}_{f_i}) + \tau_{df_i} = \tau_{f_i} \tag{3.43}$$

where

M_{f_i} : the acceleration – related inertia matrix term,
C_{f_i} : the Coriolis and Centrifugal matrix,
F_{f_i} : the friction and gravity vector,
τ_{f_i} :the torque input vector applied to each joint,
τ_{df_i} : the unknown disturbances.

(b) Dynamics of dexterous hand

Consider a hand with *N* fingers, the overall dynamic equation is given by:

$$M_f(q_f)\ddot{q}_f + C_f(q_f, \dot{q}_f)\dot{q}_f + F_f(q_f, \dot{q}_f) + \tau_{df} = \tau_f \tag{3.44}$$

where q_f represents the configurations of all joints, and the dynamic matrices are:

$$M_f = \begin{bmatrix} M_{f_1} & & 0 \\ & \ddots & \\ 0 & & M_{f_N} \end{bmatrix}, C_f = \begin{bmatrix} C_{f_1} & & 0 \\ & \ddots & \\ 0 & & C_{f_N} \end{bmatrix}, F_f = \begin{bmatrix} F_{f_1} \\ \vdots \\ F_{f_N} \end{bmatrix}, \tau_{df} = \begin{bmatrix} \tau_{df_1} \\ \vdots \\ \tau_{df_N} \end{bmatrix}, \tau_f = \begin{bmatrix} \tau_{f_1} \\ \vdots \\ \tau_{f_N} \end{bmatrix}$$

3.1.3.4 Object Dynamics

As mentioned above, a rigid body with six DOFs can be described by a kinematic chain with six open link, three revolute and three prismatic joints. Thus, the dynamic equations can be obtained in a similar manner:

$$M_o(X_o)\ddot{X}_o + C_o(X_o, \dot{X}_o)\dot{X}_o + N_o(X_o, \dot{X}_o) = F_o \tag{3.45}$$

where $X_o = [x \quad y \quad z \quad \alpha \quad \beta \quad \gamma]^T$ represents the posture of the object relative to the fixed reference, while F_o is the wrench applied to the object's center of mass; the dynamic matrices $M_o \in \Re^{6\times6}$, $C_o \in \Re^{6\times6}$ and $N_o \in \Re^{6\times1}$ are the acceleration-related inertia matrix, Coriolis and centrifugal matrix and vector of friction and gravity respectively.

3.1.4 Robot-Object System

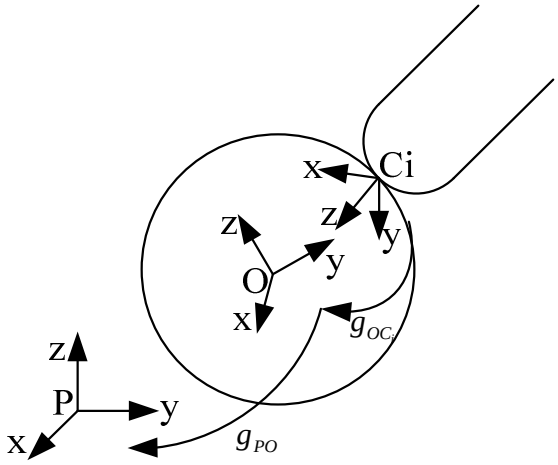


Figure 3.7: Coordinate frames and representation of a finger touching a sphere

Consider an object manipulated by a robotic hand as a closed kinematic chain. In this case the palm is fixed and a frame $\sum_P$ is attached to palm. First, we introduce the reference frames at the contact point. Figure 3.7 shows a situation where there is a contact from a finger with the coordinate frame $\sum_{C_i}$. $\sum_O$ denotes the object coordinate frame. The choice of reference point is always the center of mass of the object.

For clarity from now on we will consider two reference frames at the contact point: one is on the object ($\sum_{C_{O_i}}$) and the other one is on the fingertip ($\sum_{C_{f_i}}$). The position and orientation of the contact point on the object with respect to the reference system centered on the object ($\sum_O$) are described by the homogeneous matrix g_{OC_i}, which depends on the local coordinates of this point. The posture of the object relative to the reference frame on the palm ($\sum_P$) is expressed by the homogeneous matrix g_{PO}.

3.1.4.1 Constraint on Contact Force

(a) Wrenches between the fingers and the grasped object

A contact is basically a mapping between the wrench exerted by a finger at the contact point and the resultant wrench at a reference point on an object. There are many different contact types used to describe the wrench a finger is exerting on an object. They are frictionless point contact, point contact with friction, and soft-finger contact. In this section, we study hard-finger contact model which belongs to the second type. The hard-finger contact allows the forces to be applied within the friction cone. For the i_{th} finger, this is represented by:

$$F_{ci} = B_{ci} f_{ci},\ f_{ci} \in FC_{ci} \tag{3.46}$$

where the forces applied by a contact are modeled as a wrench F_{ci} applied at the origin of the contact frame ($\sum_{C_{O_i}}$ or $\sum_{C_{f_i}}$); f_{ci} is a vector represents the magnitude of the contact forces applied by the finger; B_{ci} is wrench basis matrix and for the hard-finger contact it is defined as: $B_{ci} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T$.

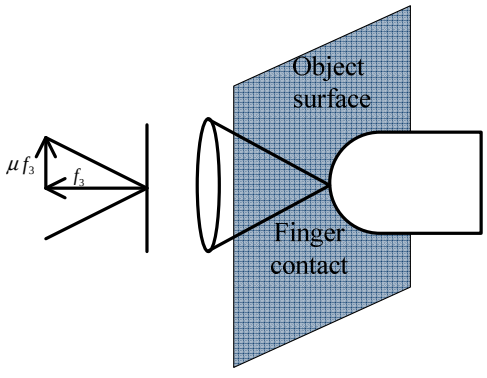


Figure 3.8: Hard-finger contact model

The set FC_{ci} represents the friction cone of contact *i*, and the hard-finger model applies:

$$FC_{c_i} = \{ f_{c_i} \in \Re^3 : \frac{1}{\mu_i} \sqrt{f_{i1}^2 + f_{i2}^2} \leq f_{i3},\ f_{i3} \geq 0 \} \tag{3.47}$$

where $f_{ci} = [\, f_{i1} \quad f_{i2} \quad f_{i3} \,]^T$, f_{i1} and f_{i2} denote the tangent components of the friction force, f_{i3} denotes the normal component of the contact force, and μ_i denotes the friction coefficient.

(b) Grasp map

Grasp map is used to determine the effect of contact wrenches on the object. To determine the effect, the wrenches must be transformed to the object coordinate frame. If $(R_{oc_{fi}}, P_{oc_{fi}})$ is the configuration of the i_{th} contact frame relative to the object frame, then the wrench exerted by this contact can be written in the object coordinate frame as:

$$F_o = Ad_{goc_{fi}}^{-T} F_{ci} = Ad_{goc_{fi}}^{-T} B_{ci} f_{ci} \tag{3.48}$$

where $Ad_{goc_{fi}}^{-T}$ is the wrench transformation matrix that maps contact wrench to object wrench.

We define the grasp matrix G_i as a linear map between the contact forces and the wrench act on the object of finger *i*:

$$G_i = Ad_{goc_{fi}}^{-T} B_{ci} \tag{3.49}$$

If we consider *N* fingers in contact, the total wrench is the combination of all wrenches on the object due to the fingers. The grasp matrix G that maps all contact forces on the object is:

$$G = [G_1 \quad ... \quad G_N] = \left[Ad_{goc_{f1}}^{-T} B_{c1} \quad ... \quad Ad_{goc_{fN}}^{-T} B_{cN} \right] \tag{3.50}$$

In this way the total wrench on the object is given by:

$$F_o = G f_c \tag{3.51}$$

where $f_c = \left[f_{c1}^T \quad f_{c2}^T \quad \cdots \quad f_{cN}^T \right]^T$.

(c) Force-closure

If a grasp can resist any applied wrench, the grasp is said to be force-closure. When grasping an object and lifting it from the table, it is necessary to apply appropriate wrenches at appropriate locations on the object so as not to allow gravity or possibly other external forces to pull the object out of the grasp. If a grasp can resist any external wrench F_e applied to the object, there exist contact forces $f_c \in FC_c$, such that:

$$Gf_c = -F_e \tag{3.52}$$

(d) Internal force

A key feature of a force-closure grasp is the existence of internal forces. Internal force is a set of contact forces which result in no net force on the object.

$$F_N \in N(G),\ GF_N = 0 \tag{3.53}$$

Internal forces can be used to insure that the contact forces satisfy the friction cone constraints.

3.1.4.2 Grasp Constraints

In this case, the constraints between the object and the fingers can be formulated by requiring that certain velocities are equal. For example, at a given contact point, the velocity of the contact point on the fingertip and that on the object must agree in the direction normal to the surface.

(a) Grasp constraints of multi-fingered hand

Recall that for the hand each finger is as an open kinematic chain, the spatial velocity of the fingertip can be written as:

$$V_{s_i f_i}^{f_i} = J_{s_i f_i}^{f_i}(q_{f_i})\dot{q}_{f_i} \tag{3.54}$$

where $J_{s_i f_i}^{f_i}(q_{f_i})$ is the distal Jacobian matrix that transforms the angular velocities ($\dot{q}_{f_i}$) of finger joints to the velocity of end of a finger ($V_{s_i f_i}^{f_i}$) in the frame attached to the finger's end;

$\sum_{s_i}$ is the base frame of the i_{th} finger attached to the palm and at the point where this finger is connected.

Assume that each finger has a fingertip with known shape. The transformation $g_{f_i c_{fi}}$ from the end of the finger to the contact point at the i_{th} fingertip depends on the local coordinates of the contact point on the surface of the fingertip.

In general, the directions in which the motions are constrained are precisely those in which the forces can be exerted. Hence, for a contact with the wrench basis B_{c_i} which has been illustrated in equation (3.46), we require that:

$$B_{c_i}^T V_{c_{oi} c_{fi}}^{c_{fi}} = 0 \tag{3.55}$$

The explicit form of the relative velocity between the fingertip and the object at the contact point is:

$$V_{c_{oi} c_{fi}}^{c_{fi}} = Ad_{g_{c_{fi} f_i}} J_{s_i f_i}^{f_i} \dot{q}_{f_i} - Ad_{g_{c_{fi} o}} V_{po}^{o} \tag{3.56}$$

where V_{po}^{o} is the velocity of the object relative to the palm in the object frame, and $g_{c_{fi} o}$ represents the homogeneous transformation matrix from the object frame $\sum_O$ to the contact frame $\sum_{C_{fi}}$ on the fingertip, which is as:

$$g_{c_{fi} o} = g_{c_{fi} c_{oi}} g_{c_{oi} o} \tag{3.57}$$

where $g_{c_{fi} c_{oi}}$ represents the homogeneous transformation from the frame $\sum_{C_{O_i}}$ to the frame $\sum_{C_{fi}}$ and $g_{c_{oi} o}$ represents the homogeneous transformation matrix from the object frame $\sum_O$ to the contact frame $\sum_{C_{O_i}}$ on the object.

Then

$$B_{c_i}^T Ad_{g_{c_{fi} f_i}} J_{s_i f_i}^{f_i} \dot{q}_{f_i} = B_{c_i}^T Ad_{g_{c_{fi} o}} V_{po}^{o} \tag{3.58}$$

Using (3.49) and (3.50), for the hand with *N* fingers, the contact kinematics has the form:

$$J_h(q_f)\dot{q}_f = G^T V_{po}^o \tag{3.59}$$

where $J_h(q_f)$ is the hand Jacobian matrix defined as:

$$J_h(q_f) = \begin{bmatrix} B_{c_1}^T Ad_{g_{c_{f_1} f_1}} J_{s_1 f_1}^{f_1} & 0 \\ 0 & B_{c_N}^T Ad_{g_{c_{f_N} f_N}} J_{s_N f_N}^{f_N} \end{bmatrix}$$

This equation describes the grasp kinematics in terms of a set of ordinary differential equations.

(b) Grasp constraints of multi-fingered mobile manipulator

In this thesis, the dexterous hands are mounted on the arms and then the velocities of the fingertips are related to those of the ends of the arms. Thus, a small change in the contact kinematics should be required. The contact kinematics is in the form:

$$J_B(q')\dot{q}_B' = G^T V_{Bo}^o \tag{3.60}$$

where $J_B(q')$ is the Jacobian matrix of the whole system defined as:

$$J_B(q') = \begin{bmatrix} B_{c_1}^T Ad_{g_{c_{f_1} f_1}} J_{Bf_1}^{f_1} & 0 \\ 0 & B_{c_N}^T Ad_{g_{c_{f_N} f_N}} J_{Bf_N}^{f_N} \end{bmatrix}$$

$q' = \begin{bmatrix} q_v^T & q_b^T & q_{f_1}^T & \cdots & q_{f_N}^T \end{bmatrix}^T$ has been defined in subsection 3.1.2.4; q_B' is the extension of vector q' and $q_B' = \begin{bmatrix} q_v^T & q_b^T & q_{f_1}^T & \cdots & q_v^T & q_b^T & q_{f_N}^T \end{bmatrix}$; V_{Bo}^o is the velocity of the object relative to the base frame in the object frame.

3.1.4.3 Contact Kinematics

In most of human and robotic hand tasks, rolling contact plays an important role in the mechanics of the manipulation. In this subsection we focus on the research of the manipulation which involves the rolling contacts by a set of fingers. An accurate modeling of the rolling phenomenon is fundamental for the research. We begin with a kinematic study of

the case of one smooth finger rolling on a smooth object and extend the grasping formulation to the moving contacts. If the fingertip and the object shapes are completely known, the contact kinematics can be described by a suitable parameterization of the contact surfaces.

(a) Surface parameterizations [132]

Given an object in $\Re^3$, we describe the surface of the object using a local coordinate chart, $C: U \subset \Re^2 \rightarrow \Re^3$, as shown in Figure 3.9.

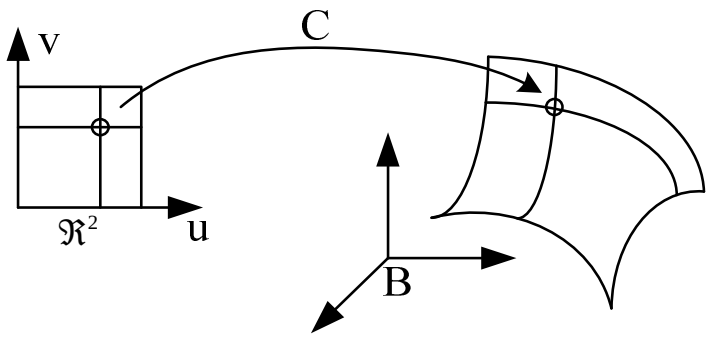


Figure 3.9: Surface chart for a two-dimensional object in $\Re^3$

Surface chart for a 2D object in $\Re^3$:

$$c(u,v) = [x(u,v),\ y(u,v),\ z(u,v)] \tag{3.61}$$

The map C takes a point $(u,\ v) \in \Re^2$ to a point $p \in \Re^3$ on the surface of the object, written in the object frame.

At any point on the object, we can define a tangent plane which consists of the space of all vectors which are tangent to the surface of the object at that point. The tangent plane is spanned by the vectors:

$$c_u = \frac{\partial c}{\partial u},\ c_v = \frac{\partial c}{\partial v} \tag{3.62}$$

That is, any vector which is tangent to the surface at a point $c(u,\ v)$ may be expressed as a linear combination of the vectors c_u and c_v, evaluated at $(u,\ v)$.

Given a parameterization, (M_p, K_p, T_p) are collectively referred to as the geometric parameters of the surface. These parameters describe the local geometry of the surface (metric tensor, curvature tensor, torsion) and play an important role in the kinematics of contact.

$$M_p = \begin{bmatrix} \| c_u \| & 0 \\ 0 & \| c_v \| \end{bmatrix},\ K_p = \begin{bmatrix} \dfrac{c_u^T n_u}{\| c_u \|^2} & \dfrac{c_u^T n_v}{\| c_u \| \| c_v \|} \\ \dfrac{c_v^T n_u}{\| c_v \| \| c_u \|} & \dfrac{c_v^T n_v}{\| c_v \|^2} \end{bmatrix},$$
$$T_p = \begin{bmatrix} \dfrac{c_v^T c_{uu}}{\| c_u \|^2 \| c_v \|} & \dfrac{c_v^T c_{uv}}{\| c_u \| \| c_v \|^2} \end{bmatrix} \tag{3.63}$$

where $n_u = \dfrac{\partial n}{\partial u},\ n_v = \dfrac{\partial n}{\partial v},\ c_u = \dfrac{\partial c}{\partial u},\ c_v = \dfrac{\partial c}{\partial v},$ while $c_{uu} = \dfrac{\partial^2 c}{\partial u^2},\ c_{uv} = \dfrac{\partial^2 c}{\partial u \partial v},\ n = N(u, v)$ is the unit normal at a point on the surface.

(b) Gauss frame

The curvature, torsion, and metric tensors can also be computed in terms of a special coordinate frame called the normalized Gauss frame. If $c(u,v)$ is an orthogonal chart, we define the normalized Gauss frame as:

$$x = \frac{c_u}{\| c_u \|},\ y = \frac{c_v}{\| c_v \|},\ z = \frac{c_u \times c_v}{\| c_u \times c_v \|} \tag{3.64}$$

The normalized Gauss frame provides an orthonormal frame at each point on the surface.

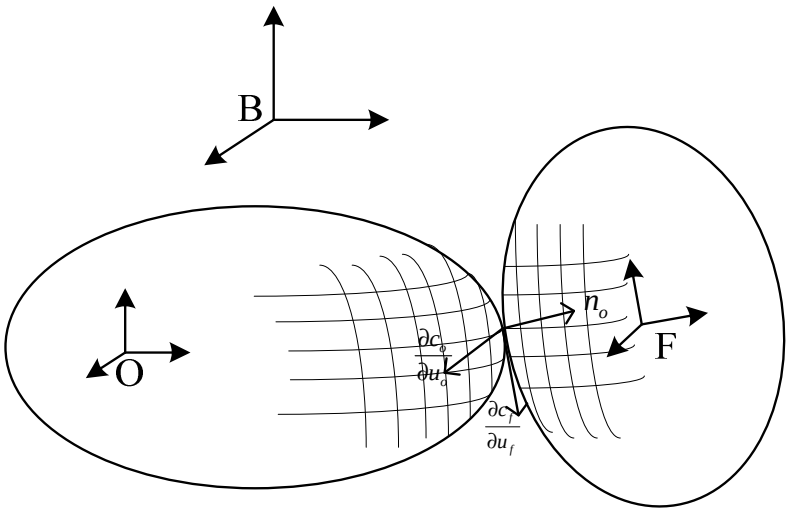


Figure 3.10: Motion of two objects in contact

Consider two objects with the surfaces S_o and S_f which are touching at a point, as shown in Figure 3.10. Let $p_o(t) \in S_o$ and $p_f(t) \in S_f$ be the positions at time t of the contact point relative to two body-fixed frames $\sum_O$ and $\sum_F$, respectively. Let (c_o, U_o) and (c_f, U_f) be

the charts for the two surfaces, and $\alpha_o = c_o^{-1}(p_o) \in U_o$ and $\alpha_f = c_f^{-1}(p_f) \in U_f$ be the local coordinates. Let ψ be the angle of contact, defined as the angle between the tangent vectors $\frac{\partial c_o}{\partial u_o}$ and $\frac{\partial c_f}{\partial u_f}$.We choose the sign of ψ so that a rotation of $\frac{\partial c_o}{\partial u_o}$ through an angle ψ around the outward normal of S_o aligns $\frac{\partial c_o}{\partial u_o}$ with $\frac{\partial c_f}{\partial u_f}$.Collecting the quantities which describe the contact, we obtain the contact coordinates $\begin{bmatrix} \alpha_f & \alpha_o & \psi \end{bmatrix}^T$ for S_f and S_o.

The contact kinematics allows the grasp matrix G and the hand Jacobian matrix J_h to be computed using the contact coordinates:

$$\kappa = \begin{bmatrix} \alpha_f & \alpha_o & \psi \end{bmatrix}^T = \begin{bmatrix} u_f & v_f & u_o & v_o & \psi \end{bmatrix}^T \tag{3.65}$$

In this work, it is assumed that the fingertips are semi-spheres and the object is a torus. Thus, the body-fixed frames $\sum_F$ and $\sum_O$ mentioned above are attached at the origin of the semi-sphere and the origin of the torus respectively. The geometric parameters of the sphere and the torus are introduced in the following section.

(c) Geometric parameters of a sphere in $\Re^3$

A coordinate chart for a sphere of radius ρ can be obtained by using spherical coordinates, as shown in Figure 3.11.

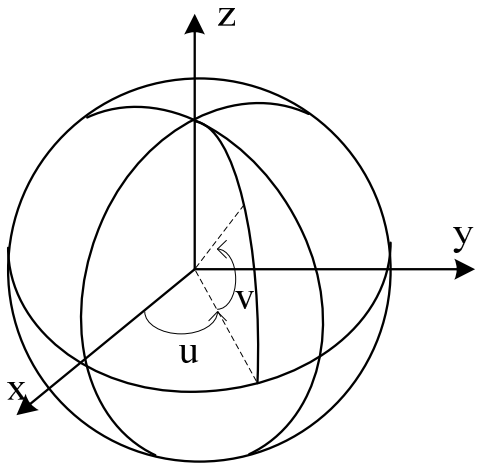


Figure 3.11: Spherical finger

For the hemisphere centered at the origin and above the *xy*-plane, we have:

$$c_f(u_f, v_f) = \begin{bmatrix} \rho \cos v_f \cos u_f \\ \rho \cos v_f \sin u_f \\ \rho \sin v_f \end{bmatrix} \tag{3.66}$$

with $U_f = \{(u_f,\ v_f)\ : 0 <\ u_f < 2\pi, 0 <\ v_f < \pi\}$. The position and orientation of the contact frame relative to the reference frame are given by:

$$x_f = \frac{c_{f,u_f}}{\| c_{f,u_f} \|} = \begin{bmatrix} -\sin u_f \\ \cos u_f \\ 0 \end{bmatrix} \tag{3.67}$$

$$y_f = \frac{c_{f,v_f}}{\| c_{f,v_f} \|} = \begin{bmatrix} -\sin v_f \cos u_f \\ -\sin v_f \sin u_f \\ \cos v_f \end{bmatrix} \tag{3.68}$$

$$z_f = \frac{c_{f,u_f} \times c_{f,v_f}}{\| c_{f,u_f} \times c_{f,v_f} \|} = \begin{bmatrix} \cos v_f \cos u_f \\ \sin u_f \cos v_f \\ \sin v_f \end{bmatrix} \tag{3.69}$$

$$R_{fc_f} = \begin{bmatrix} x_f & y_f & z_f \end{bmatrix} \tag{3.70}$$

$$p_{fc_f}(u_f, c_f) = c_f(u_f, v_f) \tag{3.71}$$

$$g_{fc_f} = \begin{bmatrix} R_{fc_f} & p_{fc_f} \\ 0_{1\times 3} & 1 \end{bmatrix} \tag{3.72}$$

The curvature, torsion, and metric tensors are:

$$M_f(v_f) = \begin{bmatrix} \rho \cos v_f & 0 \\ 0 & \rho \end{bmatrix},\ K_f = \begin{bmatrix} \frac{1}{\rho} & 0 \\ 0 & \frac{1}{\rho} \end{bmatrix},\ T_f(v_f) = \begin{bmatrix} \frac{\tan v_f}{\rho} & 0 \end{bmatrix} \tag{3.73}$$

(d) Geometric parameters of a torus in $\Re^3$

a_o is the radius of the tube and b_o is the distance from the center of the tube to the center of the torus, as shown in Figure 3.12. a_o and b_o are also known as the minor radius and major radius, respectively.

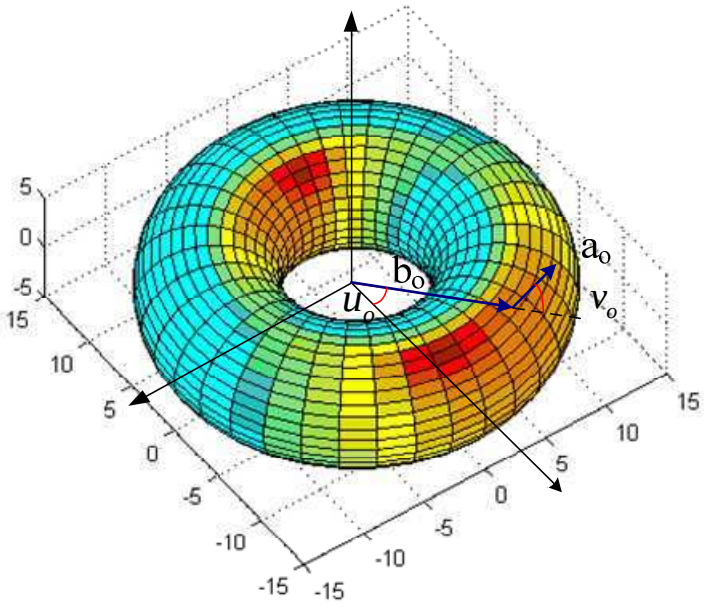


Figure 3.12: Parameterization of torus surface

Similarly, the position and orientation of the contact frame relative to the reference frame are given by:

$$c_o(u_o, v_o) = \begin{bmatrix} (b_o + a_o \cos v_o)\cos u_o \\ (b_o + a_o \cos v_o)\sin u_o \\ a_o \sin v_o \end{bmatrix} \tag{3.74}$$

$$x_o = \frac{c_{o,u_o}}{\| c_{o,u_o} \|} = \begin{bmatrix} -\sin u_o \\ \cos u_o \\ 0 \end{bmatrix} \tag{3.75}$$

$$y_o = \frac{c_{o,v_o}}{\| c_{o,v_o} \|} = \begin{bmatrix} -\sin v_o \cos u_o \\ -\sin v_o \sin u_o \\ \cos v_o \end{bmatrix} \tag{3.76}$$

$$z_o = \frac{c_{o,u_o} \times c_{o,v_o}}{\| c_{o,u_o} \times c_{o,v_o} \|} = \begin{bmatrix} \cos v_o \cos u_o \\ \sin u_o \cos v_o \\ \sin v_o \end{bmatrix} \tag{3.77}$$

$$R_{oc_o} = \begin{bmatrix} x_o & y_o & z_o \end{bmatrix} \tag{3.78}$$

$$p_{oc_o}(u_o, c_o) = c_o(u_o, v_o) \tag{3.79}$$

$$g_{oc_o} = \begin{bmatrix} R_{oc_o} & p_{oc_o} \\ 0_{1\times 3} & 1 \end{bmatrix} \tag{3.80}$$

$$M_o(v_o) = \begin{bmatrix} b_o + a_o \cos v_o & 0 \\ 0 & a_o \end{bmatrix},\ K_o(v_o) = \begin{bmatrix} \dfrac{\cos v_o}{b_o + a_o \cos v_o} & 0 \\ 0 & \dfrac{1}{a_o} \end{bmatrix},$$
$$T_o(v_o) = \begin{bmatrix} \dfrac{\sin v_o}{b_o + a_o \cos v_o} & 0 \end{bmatrix} \tag{3.81}$$

(e) Kinematic equations for rolling contact

We assume the pure rolling between the fingertips and the manipulated object. The motion of the contact coordinates, $\dot{\kappa} = \begin{bmatrix} \dot{\alpha}_f & \dot{\alpha}_o & \dot{\psi} \end{bmatrix}^T$, as a function of the relative motion is given by:

$$\begin{aligned} \dot{\alpha}_f &= M_f^{-1}(K_f + \tilde{K}_o)^{-1} \begin{bmatrix} -\omega_y \\ \omega_x \end{bmatrix} \\ \dot{\alpha}_o &= M_o^{-1} R_\psi (K_f + \tilde{K}_o)^{-1} \begin{bmatrix} -\omega_y \\ \omega_x \end{bmatrix} \\ \dot{\psi} &= T_f M_f \dot{\alpha}_f + T_o M_o \dot{\alpha}_o \end{aligned} \tag{3.82}$$

where $V_{c_o c_f}^{c_f} = \begin{bmatrix} v_x & v_y & v_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ has been defined in (3.56), $R_\psi = \begin{bmatrix} \cos\psi & -\sin\psi \\ -\sin\psi & -\cos\psi \end{bmatrix}$ and $\tilde{K}_o = R_\psi K_o R_\psi$.

Note that given the initial condition we can integrate the vector $\dot{\kappa}$ for each contact point and obtain the contact coordinates κ for each instant of time.

3.1.4.4 Constrained Dynamics of Hand-Object System

In section 3.1.3, we have analyzed the dynamics of the fingers and the object which are all assumed free, i.e. neglecting the closure constraints. However, the physics of the hand and the

object as a whole system must be considered in a grasping task. To take into account the mutual relations and the wrenches exchanged between the fingers and the manipulated object we must slightly modify the dynamic equations.

Assume no friction. With regard to the equations of the fingers, we should add a term $J_h^T f_c$ to take account of the forces exerted by the object on the fingertips [133]:

$$M_f(q_f)\ddot{q}_f + C_f(q_f,\dot{q}_f)\dot{q}_f + F_f(q_f,\dot{q}_f) = \tau_f - J_h^T f_c \tag{3.83}$$

$$f_c = G^{\dagger} F_o + F_N \tag{3.84}$$

where f_c is a vector of contact forces, $G^{\dagger}$ is the pseudoinverse of the grasp matrix G, F_o is the wrench applied to the center of the object and F_N represents the vector of internal forces at the contact points which play a fundamental role in grasp stability (i.e. slippage avoidance).

As regards the wrench applied to the object F_o, from (3.83) and (3.84), it is equivalent to:

$$F_o = GJ_h^{-T}(\tau_f - M_f\ddot{q}_f - C_f\dot{q}_f - F_f) \tag{3.85}$$

Recall the fundamental equation of the kinematic constraints:

$$J_h(q_f)\dot{q}_f = G^T V_{po}^o \tag{3.86}$$

where V_{po}^o can be calculated as a function of X_o and $\dot{X}_o$.

In the case of Euler ZYZ, we have:

$$V_{po}^o = P(X_o)^{-1}\dot{X}_o \tag{3.87}$$

where

$$P(X_o) = \begin{bmatrix} R_{bs}(X_o) & 0_{3\times3} \\ 0_{3\times3} & G_{bs}(X_o)^{-1} \end{bmatrix},$$

$$R_{bs}(X_o) = Rot(z,\alpha)Rot(y,\beta)Rot(z,\gamma),$$

$$G_{bs}(X_o) = \begin{bmatrix} -\cos\gamma\sin\beta & \sin\gamma & 0 \\ \sin\gamma\sin\beta & \cos\gamma & 0 \\ \cos\beta & 0 & 1 \end{bmatrix}.$$

$Rot(z,\alpha)$, $Rot(y,\beta)$ and $Rot(z,\gamma)$ are rotation matrices.

Therefore we have:

$$J_h(q_f)\dot{q}_f = G^T P^{-1}\dot{X}_o \tag{3.88}$$

Taking the time derivative of (3.88) gives:

$$\dot{J}_h\dot{q}_f + J_h\ddot{q}_f = \dot{G}^T P^{-1}\dot{X}_o + G^T\dot{P}^{-1}\dot{X}_o + G^T P^{-1}\ddot{X}_o \tag{3.89}$$

Assume the invertibility of J_h, equations (3.88) and (3.89) become:

$$\dot{q}_f = J_h^{-1}G^T P^{-1}\dot{X}_o \tag{3.90}$$

$$\ddot{q}_f = J_h^{-1}(\dot{G}^T P^{-1}\dot{X}_o + G^T\dot{P}^{-1}\dot{X}_o + G^T P^{-1}\ddot{X}_o - \dot{J}_h J_h^{-1}G^T P^{-1}\dot{X}_o) \tag{3.91}$$

If we replace $\dot{q}_f$ and $\ddot{q}_f$ in (3.85) with (3.90) and (3.91), we obtain the wrench applied to the object as a format only with X_o, $\dot{X}_o$ and $\ddot{X}_o$:

$$\begin{aligned} F_o = GJ_h^{-T}(\tau_f - M_f J_h^{-1}(&\dot{G}^T P^{-1}\dot{X}_o + G^T\dot{P}^{-1}\dot{X}_o + G^T P^{-1}\ddot{X}_o \\ &- \dot{J}_h J_h^{-1}G^T P^{-1}\dot{X}_o) - C_f J_h^{-1}G^T P^{-1}\dot{X}_o - F_f) \end{aligned} \tag{3.92}$$

Using the equation above, (3.45) can be written as:

$$\tilde{M}\ddot{X}_o + \tilde{C}\dot{X}_o + \tilde{N}_o = GJ_h^{-T}\tau_f \tag{3.93}$$

where

$\tilde{M} = M_o + GJ_h^{-T}M_f J_h^{-1}G^T P^{-1}$,
$\tilde{C} = C_o + GJ_h^{-T}M_f J_h^{-1}(\dot{G}^T P^{-1} + G^T\dot{P}^{-1} - \dot{J}_h J_h^{-1}G^T P^{-1}) + GJ_h^{-T}C_f J_h^{-1}G^T P^{-1}$,
$\tilde{N} = N_o + GJ_h^{-T}F_f$.

We obtain the description of the dynamics of the system composed of a dexterous hand and an object to be manipulated.

3.1.4.5 Dynamics of Dual-Hand Mobile Manipulator and Object System

The overall dynamics of a dual-hand mobile manipulator system manipulating an object is discussed in this subsection.

First, recall the kinematic constraint equation in terms of $\dot{q}'$ which is the vector of the time derivative of the generalized coordinates q' defined in subsection 3.1.2.4:

$$J_B \dot{q}_B' = G^T V_{Bo}^o \tag{3.94}$$

According to the nonholonomic constraints, for $\xi = \begin{bmatrix} v & \dot{q}_b & \dot{q}_f \end{bmatrix}^T$, there exists a full rank matrix $S'(q) \in \Re^{p' \times (p'-r)}$ that satisfies:

$$\dot{q}' = S' \xi \tag{3.95}$$

We have known that $\dot{q}_B'$ is the extension of $\dot{q}'$, thus there exists a matrix S_B' that satisfies:

$$\dot{q}_B' = S_B' \xi \tag{3.96}$$

The kinematic constraints become:

$$\begin{aligned} J_B S_B' \xi &= G^T V_{Bo}^o \\ J_B' \xi &= G^T V_{Bo}^o \end{aligned} \tag{3.97}$$

where $J_B' = J_B S_B'$.

From (3.97), the following kinematics can be derived:

$$J_B' \xi = G^T P^{-1} \dot{X}_o \tag{3.98}$$

We obtain the time derivative of (3.98):

$$\dot{J}_B' \xi + J_B' \dot{\xi} = \dot{G}^T P^{-1} \dot{X}_o + G^T \dot{P}^{-1} \dot{X}_o + G^T P^{-1} \ddot{X}_o \tag{3.99}$$

The matrix J_B' is not invertible due to the redundancy of the entire system. Equations (3.98) and (3.99) become:

$$\begin{aligned}\xi &= J_B^{'\dagger} G^T P^{-1} \dot{X}_o + N\sigma \\ \dot{\xi} &= J_B^{'\dagger}(\dot{G}^T P^{-1}\dot{X}_o + G^T \dot{P}^{-1}\dot{X}_o + G^T P^{-1}\ddot{X}_o - \dot{J}_B'(J_B^{'\dagger} G^T P^{-1}\dot{X}_o + N\sigma)) \\ &\quad + N\sigma'\end{aligned} \tag{3.100}$$

where the symbol † denotes a weighted right pseudoinverse, $N = I - J_B^{'\dagger} J_B'$ is a projector in the null space of the Jacobian matrix J_B', σ is a considered task function and σ' is another considered task function.

Similarly, we should add a term $J_B^{'T} f_c$ to the complete dynamics of the mobile manipulator to take account of the forces exerted by the object on the fingertips:

$$\bar{M}'\dot{\xi} + \bar{C}'\xi + \bar{F}' = \bar{E}'\tau' - J_B^{'T} f_c \tag{3.101}$$

According to (3.84) and (3.101), the wrench applied to the object is equivalent to:

$$F_o = G J_B^{'\dagger T}(\bar{E}'\tau' - \bar{M}'\dot{\xi} - \bar{C}'\xi - \bar{F}') + GN'\sigma'' \tag{3.102}$$

where $N' = I - J_B^{'T\dagger} J_B^{'T}$ is a projector in the null space of the Jacobian matrix $J_B^{'T}$, and σ'' means the third considered task function.

If we replace ξ and $\dot{\xi}$ with (3.100), we obtain the wrench applied to the object as a format only with X_o, $\dot{X}_o$ and $\ddot{X}_o$:

$$\begin{aligned}F_o = G J_B^{'\dagger T}(&\bar{E}'\tau' - \bar{M}'(J_B^{'\dagger}(\dot{G}^T P^{-1}\dot{X}_o + G^T \dot{P}^{-1}\dot{X}_o \\ &+ G^T P^{-1}\ddot{X}_o - \dot{J}_B'(J_B^{'\dagger} G^T P^{-1}\dot{X}_o + N\sigma)) \\ &+ N\sigma') - \bar{C}'(J_B^{'\dagger} G^T P^{-1}\dot{X}_o + N\sigma) - \bar{F}') + GN'\sigma''\end{aligned} \tag{3.103}$$

Then we obtain the complete description of the dynamics of the whole robotic system composed of a mobile manipulator with multiple fingers and an object to be manipulated:

$$\tilde{M}\ddot{X}_o + \tilde{C}\dot{X}_o + \tilde{N} = G J_B^{'\dagger T}\bar{E}'\tau' \tag{3.104}$$

where

$$\tilde{M} = M_o + GJ_B^{\dagger T}\bar{M}'J_B^{\dagger}G^T P^{-1},$$
$$\tilde{C} = C_o + GJ_B^{\dagger T}\bar{M}'J_B^{\dagger}(\dot{G}^T P^{-1} + G^T\dot{P}^{-1} - \dot{J}_B'J_B^{\dagger}G^T P^{-1}) + GJ_B^{\dagger T}\bar{C}'J_B^{\dagger}G^T P^{-1},$$
$$\tilde{N} = N_o + GJ_B^{\dagger T}\bar{F}' - GJ_B^{\dagger T}\bar{M}'J_B^{\dagger}\dot{J}_B'N\sigma + GJ_B^{\dagger T}\bar{M}'N\sigma' + GJ_B^{\dagger T}\bar{C}'N\sigma - GN'\sigma".$$

3.1.5 Simulation and Results

The mathematical model has been developped in Matlab/Simulink environment and in order to validate the model accuracy and the consistency with the ADAMS model of the personal assistant robot some complex numerical simulations are performed. The main simulation purposes are: (1) validate the accuracy of kinematics; (2) validate the accuracy of dynamics.

As ADAMS is meant for 3D simulation of dynamics, it has very limited capability for realizing a controller. We create an interface between ADAMS and Matlab/ Simulink so that the controller can be implemented using Simulink. To call the generated plant in Matlab, the plant name should be entered in the “command window” so that the input and output information will appear. By executing “Adams_sys” command, the blocks containing information about the dynamics of the robot in Simulink environment are loaded (Figure 3.13). By having “Adams-sub” block in Simulink environment, the robot model is suitable for control and motion simulation as a defined system in Matlab.

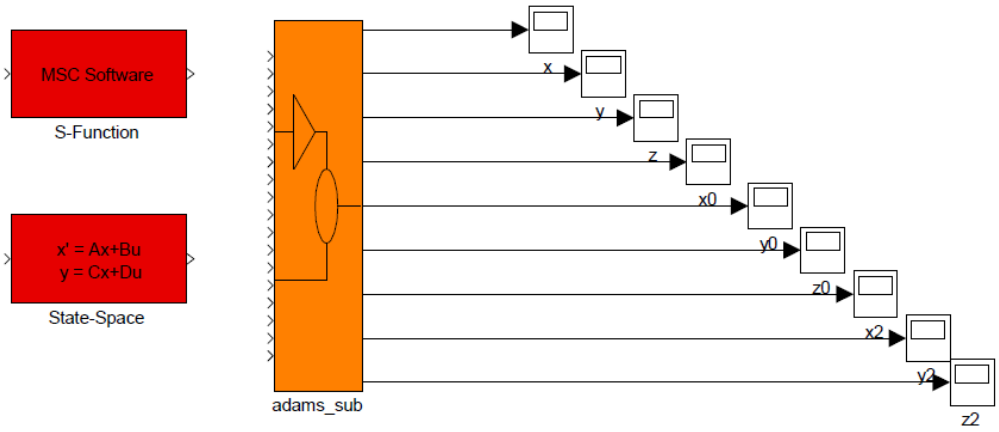


Figure 3.13: Model plant and state space blocks

3.1.5.1 Kinematics Validation

To ensure the success of dynamic simulation, first, we need to validate the kinematics of the mathematical model. The dynamic effects of the robot are neglected during the simulation and we only validate Loop *a* and Loop *b* (i.e. the fingers keep stationery). The kinematic simulation can also verify the reasonableness of the mechanical design.

A velocity level controller shown in Figure 3.14 is used as a closed-loop feedback controller and developed in Simulink. The general inverse solution of a kinematically redundant robot at the velocity level has the following form:

$$\dot{q} = J^{\dagger}(\dot{p}_d + K_p(p_d - p)) + (I - J^{\dagger}J)\dot{\varphi} \tag{3.105}$$

where q represents the vector of joint variables, K_p is a symmetric positive definite matrix, and the choice of K_p guarantees that the error uniformly converges to zero; p represents the end-effector position and p_d is the desired end-effector position; J is the Jacobian matrix of the mobile manipulator and $J^{\dagger}$ is its pseudoinverse; as for $\dot{\varphi}$, (4.24) is used.

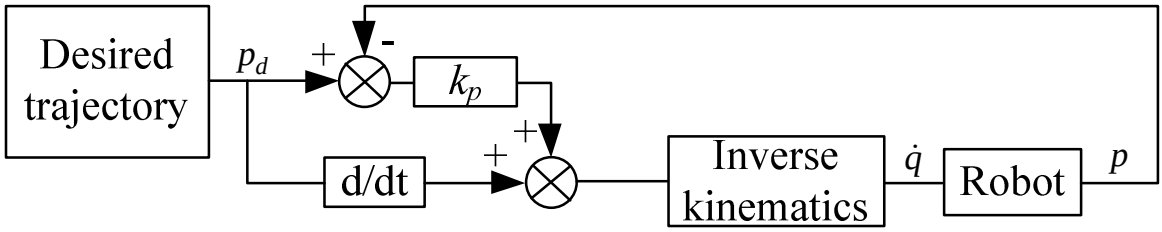


Figure 3.14: Redundancy resolution at the velocity level

Figure 3.15 demonstrates the block diagram of the control system in Simulink and through it the co-simulation between Matlab and ADAMS can be achieved.

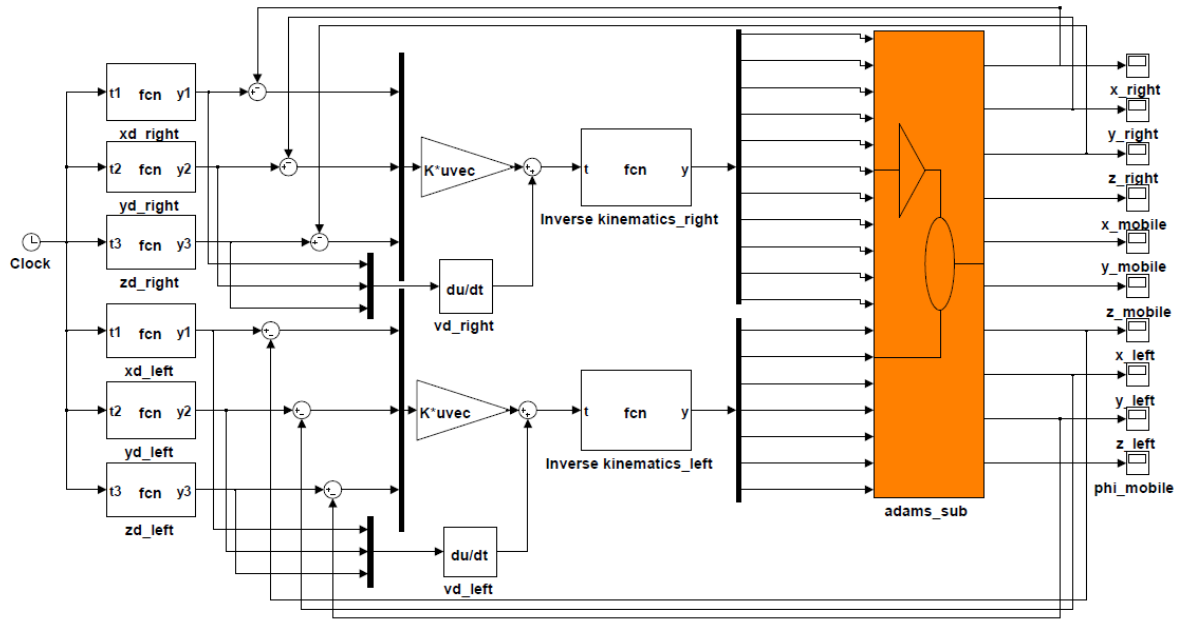


Figure 3.15: Block diagram of closed-loop control in Simulink

As shown in Figure 3.15, ADAMS outputs the end-effector positions of left and right arms respectively. The desired trajectories of the end-effectors have been given as a constraint.

According to this condition, the position errors can be acquired. And through the controller, Simulink gives each joint's angular velocity to reduce the position errors.

First, suppose that the fingers, the right arm and the mobile platform remain stationary. The left manipulator is tested with a given trajectory based on the Jacobian pseudoinverse method. Figure 3.16(a) draws the desired trajectories along *x*, *y*, and *z* axis. The reference in task space is generated by the equations: $x = x_{l0} + 0.01t,\ y = y_{l0} + 0.015t,\ z = z_{l0} + 0.02t$, where $\begin{bmatrix} x_{l0} & y_{l0} & z_{l0} \end{bmatrix}^T$ is the initial position of the left end-effector. The end-effector moves with a reasonable speed $\begin{bmatrix} 0.01 & 0.015 & 0.02 \end{bmatrix}^T$ along three orthogonal directions respectively.

Figure 3.16(b) depicts the practical motions in these three directions. Figure 3.16(c) shows the practical velocities of the end-effector and Figure 3.16(d) shows the trajectory tracking errors. Comparing Figure 3.16(b) with Figure 3.16(a), we can see that the manipulator can move following the desired trajectories. Figure 3.17 gives the robot's final configuration after execution of the task.

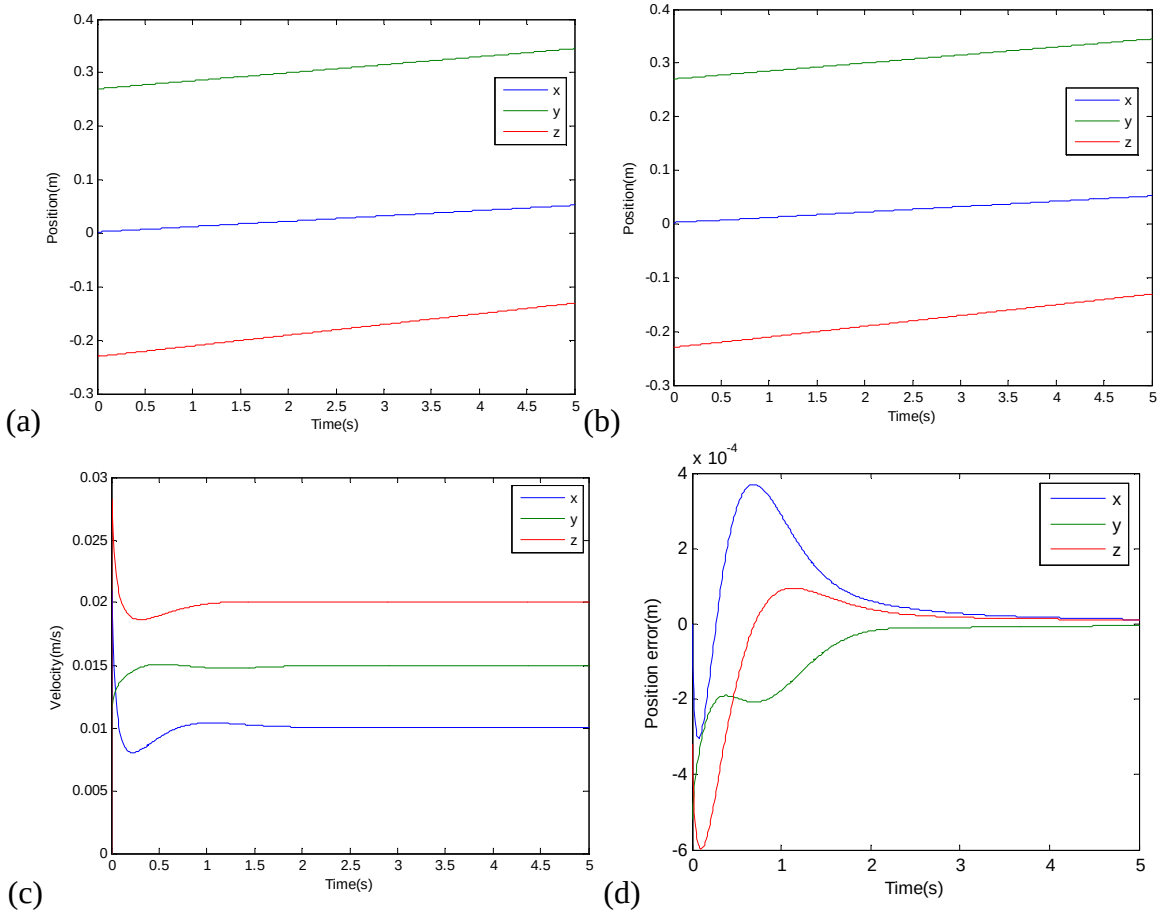


Figure 3.16: Simulation results of one redundant manipulator (a) Desired trajectories (b) Practical trajectories (c) Practical velocities (d) Trajectory tracking errors

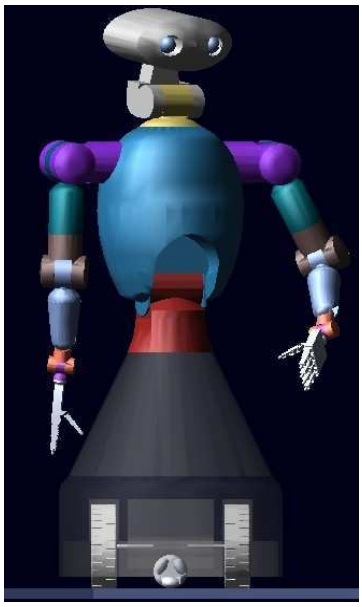
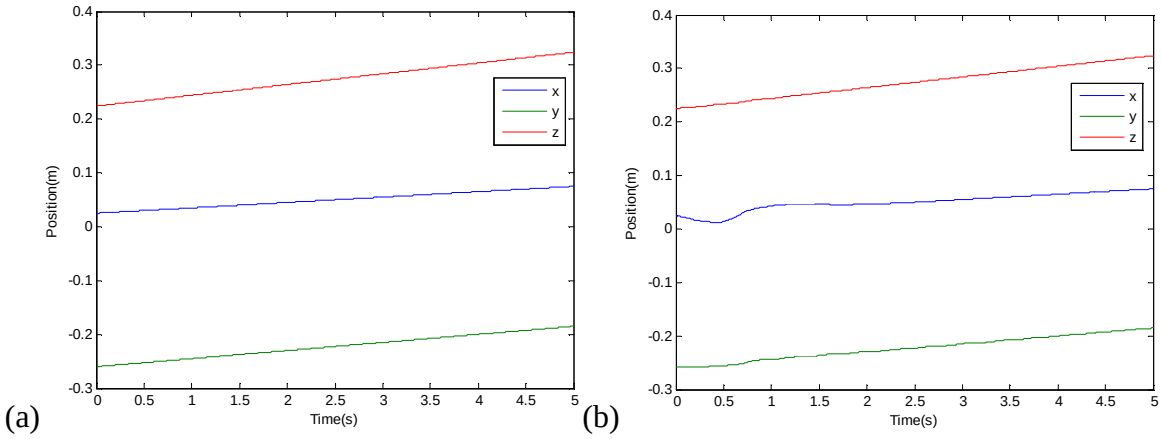


Figure 3.17: Final configuration

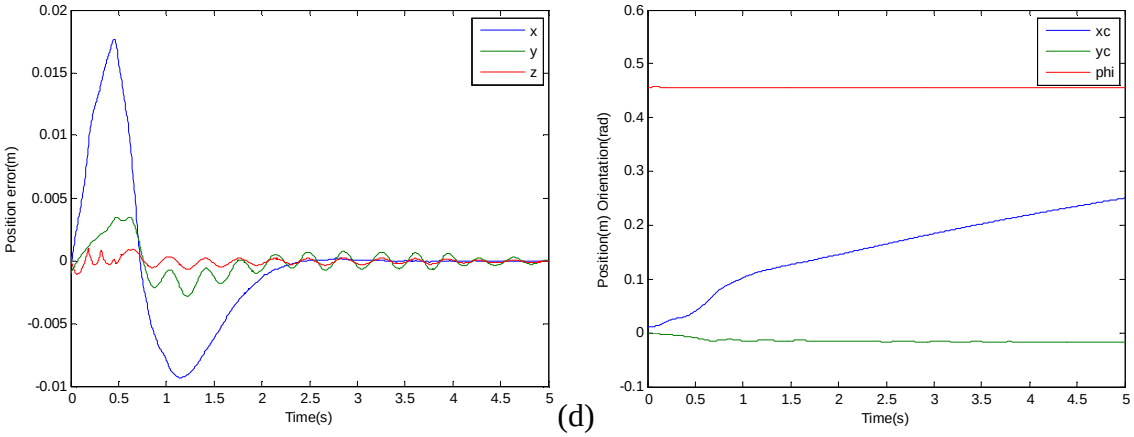
Secondly, we validate the robot's kinematics without hands (mobile base + dual manipulators). The mobile platform and the right manipulator, which compose Loop *a*, are regarded as a redundant system and the motion control method (3.105) is used. The reference trajectory in task space is given to the right end-effector. Each joint of the left arm keeps the same motion with the corresponding joint of the right arm. The reference in task space is generated by the equations:

$$x = x_{r0} + 0.01t,\ y = y_{r0} + 0.015t,\ z = z_{r0} + 0.02t$$

where $\begin{bmatrix} x_{r0} & y_{r0} & z_{r0} \end{bmatrix}^T$ is the initial position of the right end-effector. The right end-effector also moves with a speed $\begin{bmatrix} 0.01 & 0.015 & 0.02 \end{bmatrix}^T$ along three orthogonal directions respectively. The difference between this task and the previous task is the motions of the mobile base and the other arm.



(a) (b)



(c) (d)

Figure 3.18: Kinematic simulation results of complete mobile manipulator (a) Desired trajectories (b) Practical motions (c) Trajectory tracking errors (d) Motions of mobile platform

The simulation results are shown in Figure 3.18. Figure 3.18(a) and (b) depict the desired trajectories and the practical motions in the *x*, *y* and *z* directions, respectively. Figure 3.18(c) shows the trajectory tracking errors. Figure 3.18(d) gives the motions of the mobile platform along *x*, *y* directions and the heading angle. The robot's final configuration is shown in Figure 3.19. Comparing the results of two simulations we find that the curves in the *x* and *y* directions oscillate more obviously. This result indicates that adding the mobile part seriously affects the dynamic behavior. In other words the mobile manipulator has a complex dynamic model and strong coupling which are possibly caused by the mobile platform. Thus, we must take into account the dynamic characteristics when designing the control algorithm.

Figure 3.19: Final configuration

3.1.5.2 Dynamics Validation

Using ADAMS, a multi-body dynamic model has been created to represent the real robot. Connecting the mechanical model with the control system, some specific results which can be used to evaluate the robot's dynamic behavior have been obtained. This simulation process creates a closed-loop in which the control inputs from Matlab affect the ADAMS simulation, and the ADAMS outputs affect the control input level. In this case, the inputs to the robotic system are the joint torques computed by Simulink. The outputs, which will be transmitted to the controller, are the joint angles and joint angular velocities measured in ADAMS.

(a) Without hand

Assume no frictions. Here, for validating the model, a classical closed-loop control (computed torque method) based on the precise mathematical model is applied:

$$\bar{M}(\dot{\zeta}_d + k_v\dot{e} + k_p e) + \bar{C}\zeta + \bar{F} = \bar{E}\tau \tag{3.106}$$

where k_p and k_v are position and velocity gains respectively, $\dot{e} = \zeta_d - \zeta$ is the vector of velocity tracking errors, ζ_d represents a desired joint velocity vector, e is the vector of trajectory tracking errors and $\bar{M}$, $\bar{C}$ and $\bar{F}$ represent the computed values of $\bar{M}(q)$, $\bar{C}(q,\dot{q})$ and $\bar{F}(q)$ respectively.

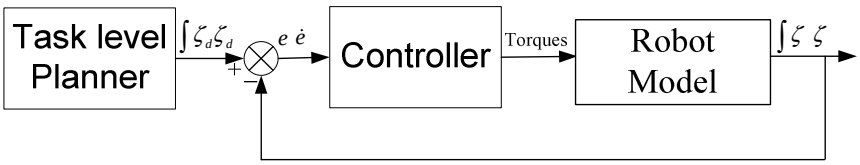


Figure 3.20: Typical computed torque control

The control system is designed to rotate the robot's joints for accomplishing various tasks. The first task is designed to move the robot's joints from the initial configuration q_0 to a desired configuration q_d. The initial value q_0 is detected by ADAMS. The goal configuration of each joint is zero degree.

As can be seen in Figure 3.21, in comparison with Figure 2.7, all of the revolute joint angles are zero based on the coordinated simulation system.

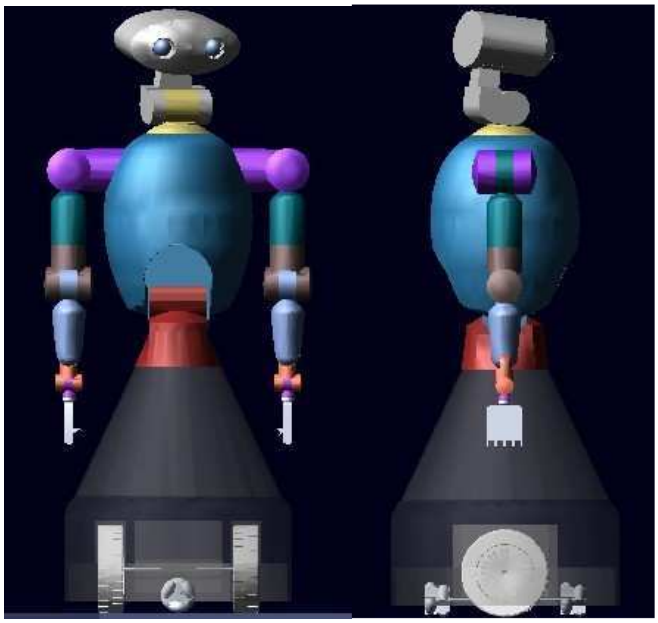


Figure 3.21: Final robot configuration

The trajectory tracking errors of all joints except the wheels are given by Figure 3.22(a), (b) and (c). Figure 3.22(d) shows the position and the orientation of the mobile platform. It can be observed that the simulation results coincide with the desired ones.

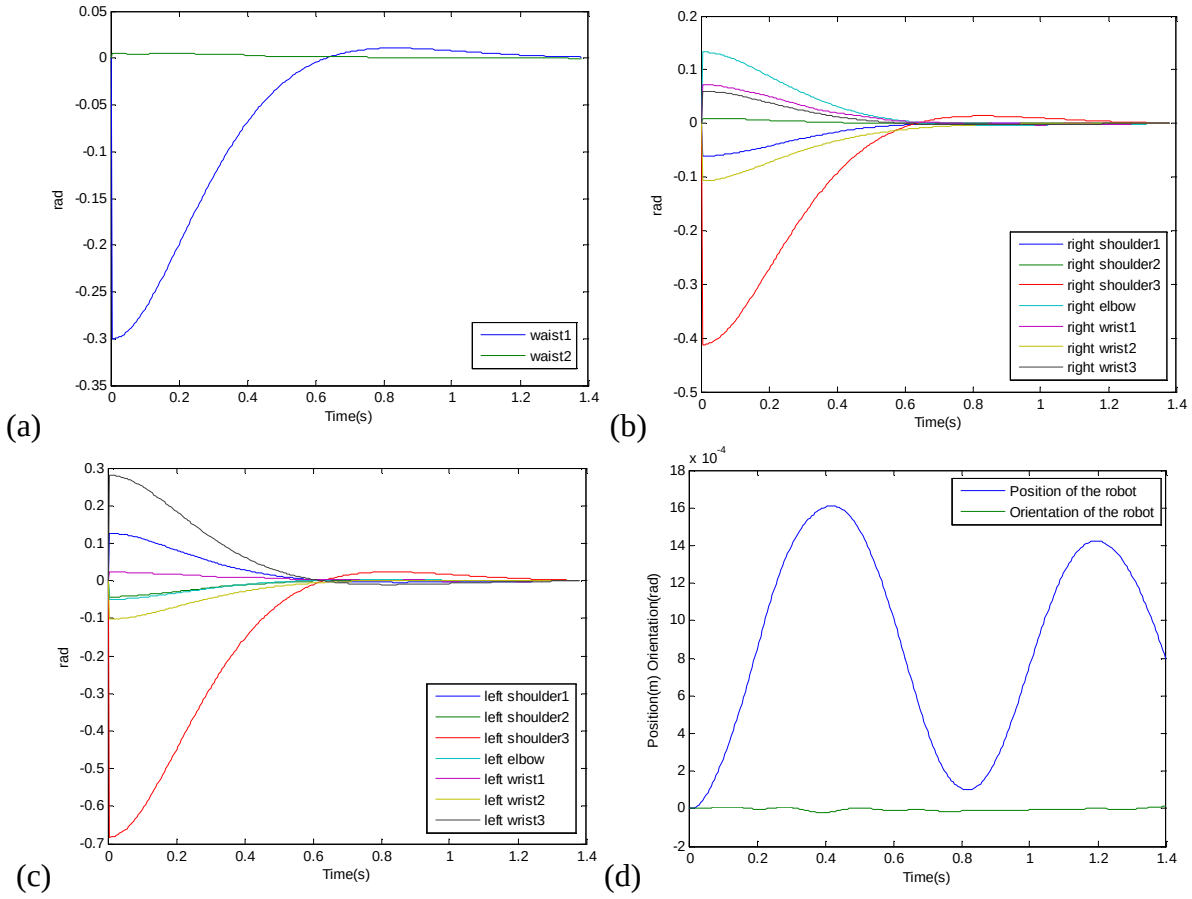


Figure 3.22: Simulation results of dynamics validation (a) Tracking errors of waist (b) Tracking errors of right arm (c) Tracking errors of left arm (d) Posture of mobile platform

When the mobile base keeps stationary, the motions of the manipulator joints which are measured by ADAMS are illustrated in Figure 3.23. As it can be seen from this figure, the motions are much smoother than those in Figure 3.22. This result again demonstrates the strong coupling between the mobile platform and the manipulators.

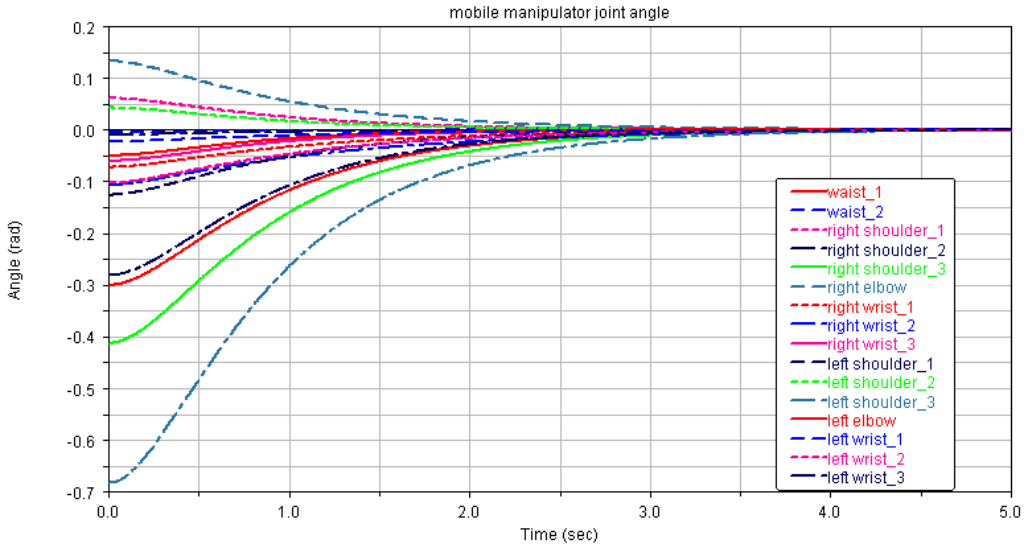


Figure 3.23: Motions of manipulator joints without mobile platform measured by ADAMS

The second simulation is to make each joint track a reference trajectory from the initial configuration.

The reference trajectories of two wheels for the mobile platform are designed as:

$$\begin{bmatrix} q_{rd} & q_{ld} \end{bmatrix}^T = \begin{bmatrix} 0.3t & 0.3t \end{bmatrix}^T + \begin{bmatrix} q_{r0d} & q_{l0d} \end{bmatrix}^T$$

Meanwhile, the reference trajectories for the onboard waist and arms are given by:

$$\begin{aligned} & q_{w1d} = q_{w2d} = q_{r1d} = q_{r2d} = q_{r3d} = q_{r4d} = q_{r5d} = q_{r6d} \\ & = q_{r7d} = q_{l1d} = q_{l2d} = q_{l3d} = q_{l4d} = q_{l5d} = q_{l6d} = q_{l7d} = 0.2t + q_{b0d}^{\ T} \end{aligned}$$

where $\begin{bmatrix} q_{r0d} & q_{l0d} \end{bmatrix}$ is the initial configuration of the wheels and q_{b0d} is the initial configuration of the waist and the arms.

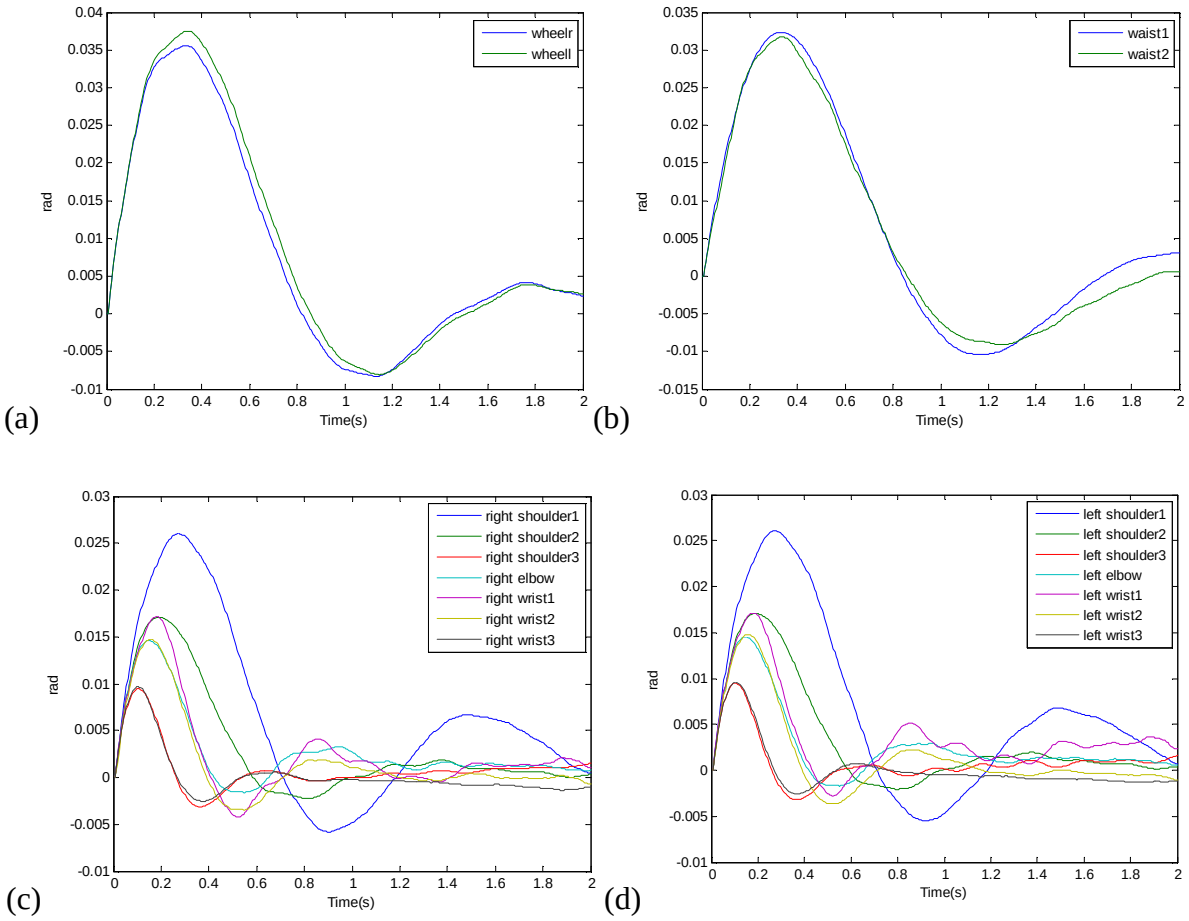


Figure 3.24: Trajectory tracking errors (a) Tracking errors of wheels (b) Tracking errors of waist (c) Tracking errors of right arm (d) Tracking errors of left arm

The simulation results are shown in Figure 3.24. It can be seen that the robot can track the desired trajectories in joint space by properly choosing the gains k_p and k_v. We have known that ADAMS prototype works in the virtual environment similar to the real world where there are uncertainties (e.g. modeling errors) and disturbances (e.g. frictions and vibrations), so that it is very difficult to obtain the precise dynamic model. However, the computed torque control is a model-base method, so the errors cannot converge to zero completely. A robust adaptive control method is necessary to compensate for the uncertainties and disturbances, and to guarantee the convergence.

Considering the above discussion, the results still demonstrate that the co-simulation between ADAMS and Matlab is effective in controlling the robot to follow the desired trajectories in joint coordinate system, and the ADAMS model is verified to represent the real robot.

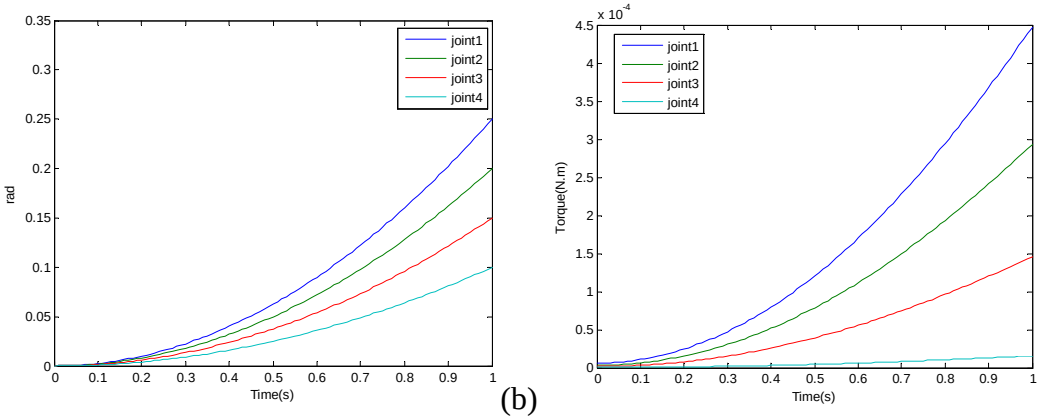
(b) Dexterous hand validation

Recall that the dynamics of each finger is in the form:

$$M_{f_i}(q_{f_i})\ddot{q}_{f_i} + C_{f_i}(q_{f_i}, \dot{q}_{f_i})\dot{q}_{f_i} + F_{f_i}(q_{f_i}, \dot{q}_{f_i}) = \tau_{f_i}$$

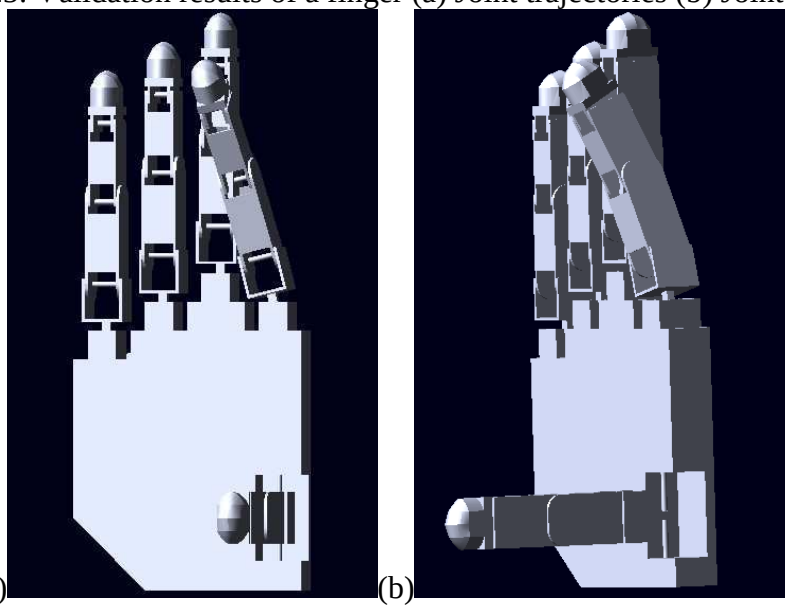
Assuming that the parameters are all known and there are no disturbances, we have obtained the precise mathematical dynamic model of each finger which has the same form as the above equation. For validating the virtual prototype of the hand, we use this precise mathematical model to construct an open-loop control system for calculating the torques of driving motors. We now apply these torques as the inputs of the model to track the reference trajectories in joint space. The desired joint trajectories for a finger are given as:

$$\begin{bmatrix} q_{f11} & q_{f12} & q_{f13} & q_{f14} \end{bmatrix}^T = \begin{bmatrix} 0.25t^2 & 0.2t^2 & 0.15t^2 & 0.1t^2 \end{bmatrix}^T$$



(a) (b)

Figure 3.25: Validation results of a finger (a) Joint trajectories (b) Joint torques



(a) (b)

Figure 3.26: Final configuration of dexterous hand

The several output results are shown. Figure 3.25(a) gives each joint's trajectory. In Figure 3.25(b), the torques obtained from the mathematical model are shown. From the above figures it could be clearly observed that the virtual prototype of the finger built in ADAMS is in

accordance with the mathematical model and the desired trajectories are well tracked using the open-loop control system. The simulation results have shown the effectiveness of the model of single finger. Using the same method, the other fingers' models are validated. Figure 3.26 describes the final configuration of the hand.

3.2 Bond Graph Modeling

3.2.1 Introduction

The model of the personal assistant robot has been well developed using Lagrange's formation. This subchapter proposes the use of bond graph technique as an alternative method.

Bond graph notation ensures encapsulation of functionality, extendibility and reusability of each element of the model. Multiphysic dynamic systems such as electrical, mechanical, magnetic, fluid, chemical, and thermodynamic systems can be modeled and linked together. As a result this method is generally more suitable for the modeling of complex robotic systems like mobile manipulators than equation-based modeling. Similar work has been done. In [134] bond graph method has been applied to model the mechanical dynamics of an excavating manipulator which is modeled as a 3 DOFs planar manipulator. This is done by applying forward recursive equations similar to those applied in Newton-Euler method only to determine the centroid velocities of the links. In [135] the dynamic behavior of the youBot (mobile manipulator robot) is decomposed into three types of components: Mecanum wheels, joints of the manipulator and links of the manipulator. The kinematic representation of mechanical elements is captured using screw theory. For the bond graph modeling of hand, a bond graph model has been developed for a prosthetic mechanism — a thumb prosthesis with capabilities to move and do a large class of manipulative tasks [136]. Contact interaction of the prosthetic mechanism with a rigid object has been modeled. The Word Bond Graph Objects (WBGO) have been used to model the dynamics of a hand prosthesis system [137]. Although significant research has been done on modeling of different robots by bond gragh, no one has established a complete bond graph model for the personal assistant robot.

In this subchapter the bond graph model of the complex dual-arm/hand mobile manipulator system is build step by step following the Newton-Euler formalism which has been widely used for modeling this kind of system. Bond graphs are used to represent the dynamics of each part including the mobile platform, the arm/hand system and the electromechanical system.

3.2.2 Modeling of Electromechanical System

An electromechanical system can be considered as a concatenation of four parts: electrical part of DC motor, mechanical part of DC motor, gear set and load part. The word bond graph model is given by Figure 3.27. Figure 3.28 describes the bond graph model of the electromechanical part. The electrical part consists of three components: resistance R_a, inductance L_a and an input effort source U which represents the voltage source of the motor. The gyrator GY describes the electromechanical conversion in the motor. The mechanical part is characterized by its inertia I_a and viscous friction coefficient B_m. The transmission elasticity of the shaft is represented by the element $C = 1/K$. TF with a reducer constant k_r models the gearbox. The load part (joint) is characterized by its inertia J and viscous friction coefficient B.

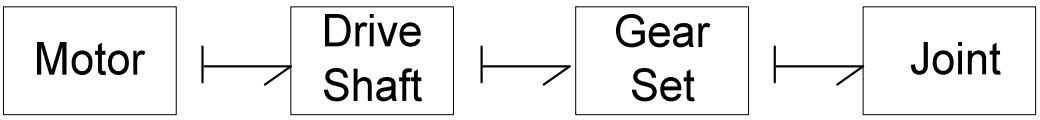


Figure 3.27: Word bond graph of electromechanical system

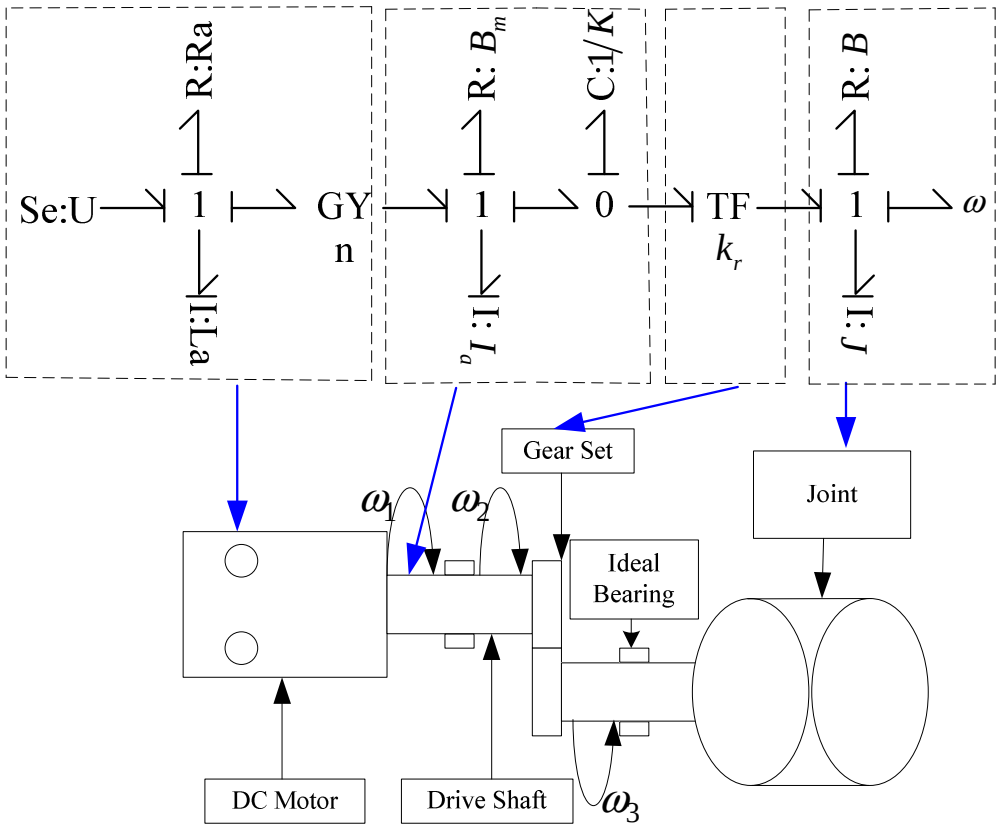


Figure 3.28: Schematic of electromechanical system

3.2.3 Modeling of Manipulator

The modeling of a manipulator which is a chain of links connected by revolute joints involves the modeling of angular and linear dynamics. In this section, bond graph modeling is done using various objects like angular dynamic object, linear dynamic object, EJS object etc. These objects are created for a single link and can be repeatedly used for the bond graph modeling of each link. Then these links are connected by the joints.

3.2.3.1 Modeling of Joint

For modeling a joint, it is decided only to model the transfer of energy in a joint. Therefore, the following behavior has explicitly been neglected: (1) friction in the joint, and (2) energy storage in the joint.

Since a joint establishes a connection between two links, it imposes a relation between the angular velocities and the torques of the two bodies. A torque applied in a joint is applied between the two connected links and, in fact, applies the same torque to both links. This specifies the relation between the torque applied by the actuator (τ^i_{acti}) and the torques applied to the connected links ($\tau^i_{joint\,i,i-1}$ and $\tau^i_{joint\,i,i}$):

$$\tau^i_{acti} = \tau^i_{joint\,i,i-1} = \tau^i_{joint\,i,i} \tag{3.107}$$

On the other hand, there is a relation between the angular velocities of ω^i_{i-1} and ω^i_i. Just as with other flow type variables the following relation holds:

$$\omega^i_i = \omega^i_{i-1} + \omega^i_{acti} \tag{3.108}$$

In this subchapter a^i_j denotes the value a of link j expressed in frame i. Thus ω^i_i and ω^i_{i-1} are respectively the angular velocity of link i expressed in frame i and the angular velocity of link $i-1$ expressed in frame i. ω^i_{acti} denotes the angular velocity generated by the motor. The above two relations can be represented by 0-junction in the bond graph language described in Figure 3.29(a) and (b).

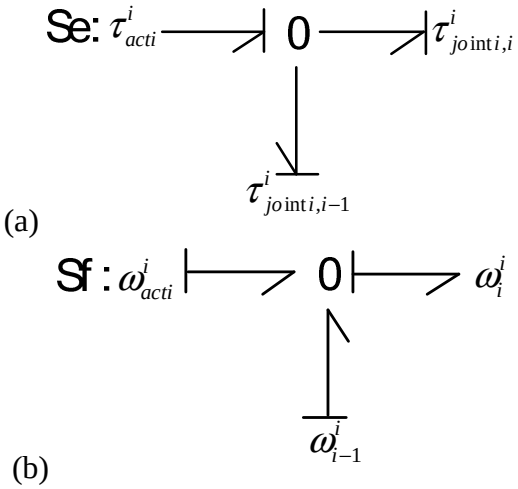


Figure 3.29: Bond graph of a joint (a) Effort source (b) Flow source

3.2.3.2 Modeling of Link

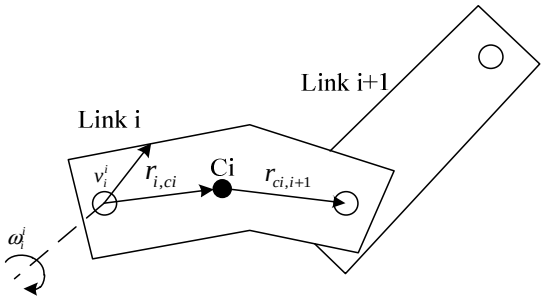


Figure 3.30: Velocity relation between links

The velocities of link *i* are completely determined by the velocities of link *i*-1 and the motion of joint *i*. Figure 3.30 describes the velocity relations between two links. These relations can be expressed by equations (3.109), (3.110) and (3.111).

Angular velocity of link *i*:

$$\omega_i^i = R_{i(i-1)}(\omega_{i-1}^{i-1} + \omega_{acti}^{i-1}) \tag{3.109}$$

Translational velocity of link *i*:

$$v_{ci}^i = R_{i(i-1)} v_i^{i-1} + \tilde{r}_{i,ci}\omega_i^i \tag{3.110}$$

$$v_{i+1}^i = v_{ci}^i + \tilde{r}_{ci,i+1}\omega_i^i \tag{3.111}$$

where c_i denotes the center of mass of link *i*; point *i* and *i*+1 denote the i_{th} and $(i+1)_{th}$ connection points between links; ω_{i-1}^{i-1} is the angular velocity of link $i-1$ expressed in frame $i-1$; v_{ci}^i, v_i^{i-1} and v_{i+1}^i represent the linear velocities of the center of mass of link *i* expressed

in frame i, frame origin of link i expressed in frame $i-1$ and frame origin of link $i+1$ relative to frame i respectively; $R_{i(i-1)}$ is the rotation matrix between frames i and $i-1$; $r_{i,ci}$ is a vector from point i to point ci; similarly, $r_{ci,i+1}$ is a vector from point ci to point i+1. The skew-symmetric matrix $\tilde{r}$ is built from r such that

$$r = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}, \ \tilde{r} = \begin{bmatrix} 0 & r_3 & -r_2 \\ -r_3 & 0 & r_1 \\ r_2 & -r_1 & 0 \end{bmatrix}$$

3.2.3.3 Newton-Euler Equations

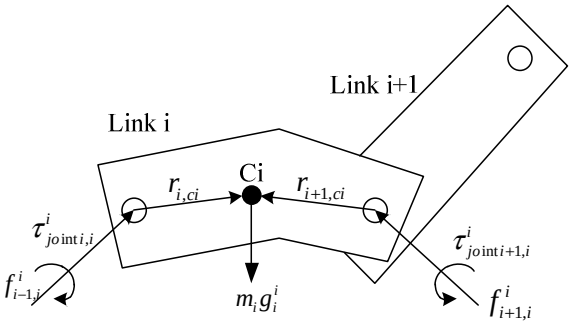


Figure 3.31: Force relation between links

Figure 3.31 shows link i together with all forces and torques acting on it. In this figure, m_i is the mass of link i and g^i_i is the gravitational acceleration expressed in frame i. By the law of action and reaction $f^i_{i-1,i}$ and $f^i_{i+1,i}$ are respectively the force acting from link (i-1) on link i and the force acting from link (i+1) on link i expressed in frame i. As shown in (3.107), $\tau^i_{joint\,i,i}$ and $\tau^i_{joint\,i+1,i}$ are the torques applied by the actuators relative to frame i. When all vectors in Figure 3.31 are expressed in frame i, the total force and torque applied to a link can be stated as

Total force applied to a link:

$$F^i_i = f^i_{i-1,i} + f^i_{i+1,i} + m_i g^i_i \tag{3.112}$$

Total torque applied to a link:

$$\tau^i_i = \tau^i_{joint\,i,i} + \tau^i_{joint\,i+1,i} + \tilde{r}_{i,ci} f^i_{i-1,i} + \tilde{r}_{i+1,ci} f^i_{i+1,i} \tag{3.113}$$

3.2.3.4 Eulerian Junction Structure in 3D Mechanical Systems

From (3.112) and (3.113), the coordinates of the forces and the torques are all written relative to a body-fixed frame attached at the center of mass instead of with respect to an inertial frame. In this way, the inertial tensor remains constant. Hence, we can formulate the d'Alembert principle in a body-fixed coordinate system without updating the inertial tensor. However, we now must calculate the relative coordinate transformations across joints. We must also take into account the gyroscopic torques and forces that result from formulating the d'Alembert principle in an accelerated frame. The nonlinear differential equations (3.114) known as Newton-Euler's equations derived by applying D'Alembert principle represent the body motion in the 3D local frame.

$$\begin{aligned}
\tau_{ix}^{i} &= J_{ixx}^{i}\dot{\omega}_{ix}^{i} + \omega_{iy}^{i}J_{izz}^{i}\omega_{iz}^{i} - \omega_{iz}^{i}J_{iyy}^{i}\omega_{iy}^{i} \\
\tau_{iy}^{i} &= J_{iyy}^{i}\dot{\omega}_{iy}^{i} + \omega_{iz}^{i}J_{ixx}^{i}\omega_{ix}^{i} - \omega_{ix}^{i}J_{izz}^{i}\omega_{iz}^{i} \\
\tau_{iz}^{i} &= J_{izz}^{i}\dot{\omega}_{iz}^{i} + \omega_{ix}^{i}J_{iyy}^{i}\omega_{iy}^{i} - \omega_{iy}^{i}J_{ixx}^{i}\omega_{ix}^{i} \\
F_{ix}^{i} &= m_i\dot{v}_{cix}^{i} + m_i\omega_{iy}^{i}v_{ciz}^{i} - m_i\omega_{iz}^{i}v_{ciy}^{i} \\
F_{iy}^{i} &= m_i\dot{v}_{ciy}^{i} + m_i\omega_{iz}^{i}v_{cix}^{i} - m_i\omega_{ix}^{i}v_{ciz}^{i} \\
F_{iz}^{i} &= m_i\dot{v}_{ciz}^{i} + m_i\omega_{ix}^{i}v_{ciy}^{i} - m_i\omega_{iy}^{i}v_{cix}^{i}
\end{aligned} \tag{3.114}$$

where $F_{ix,iy,iz}^{i}$ are the components of F_i^i, $\tau_{ix,iy,iz}^{i}$ are the components of τ_i^i and $J_{ixx,iyy,izz}^{i}$ are the components of J_i^i which is the moment of inertia of link *i* about its center of mass. All the forces $F_{ix,iy,iz}^{i}$, the torques $\tau_{ix,iy,iz}^{i}$ and the inertial tensor $J_{ixx,iyy,izz}^{i}$ are written relative to the body-fixed frame. F_i^i and τ_i^i have been derived in (3.112) and (3.113).

Euler's equations of a rigid body contain a term with an exterior product between the momentum and the angular velocity. This term is associated with the gyroscopic moments M_{gyro}. The common bond graph representation of this exterior product in Euler's equations is the EJS. For the torques it is denoted by EJS-T:

$$M_{gyro-T} = \omega_i^i \times J_i^i \omega_i^i \tag{3.115}$$

where

$$J_i^i = \begin{bmatrix} J_{ixx}^i & 0 & 0 \\ 0 & J_{iyy}^i & 0 \\ 0 & 0 & J_{izz}^i \end{bmatrix} \text{ and } \omega_i^i = \begin{bmatrix} \omega_{ix}^i & \omega_{iy}^i & \omega_{iz}^i \end{bmatrix}^T .$$

From equation (3.115), the following expression can be derived:

$$\begin{aligned} M_{gyro-T} &= \begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} 0 & J_{izz}^i \omega_{iz}^i & -J_{iyy}^i \omega_{iy}^i \\ -J_{izz}^i \omega_{iz}^i & 0 & J_{ixx}^i \omega_{ix}^i \\ J_{iyy}^i \omega_{iy}^i & -J_{ixx}^i \omega_{ix}^i & 0 \end{bmatrix} \begin{bmatrix} \omega_{ix}^i \\ \omega_{iy}^i \\ \omega_{iz}^i \end{bmatrix} \\ &= X(J_i^i \omega_i^i)\omega_i^i \end{aligned} \tag{3.116}$$

Similarly, the EJS for the forces called EJS-F is as equation (3.117):

$$M_{gyro-F} = \omega_i^i \times m_i v_{ci}^i \tag{3.117}$$

where $v_{ci}^i = \begin{bmatrix} v_{cix}^i & v_{ciy}^i & v_{ciz}^i \end{bmatrix}^T$.

In multiband notation, it has been shown that the EJS may be represented by a nonlinear three-port gyrator.

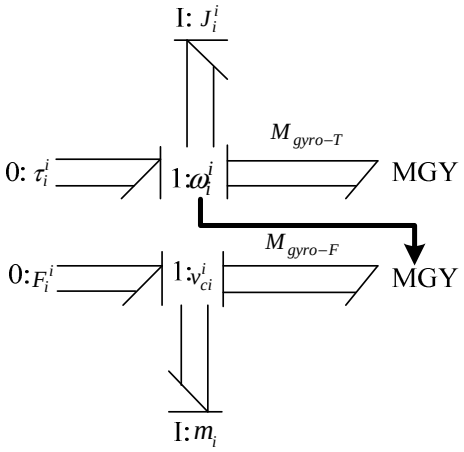


Figure 3.32: Nonlinear 3-port gyrator composition of EJS

Figure 3.33 shows the detailed bond graph representations of the Newton-Euler's equations.

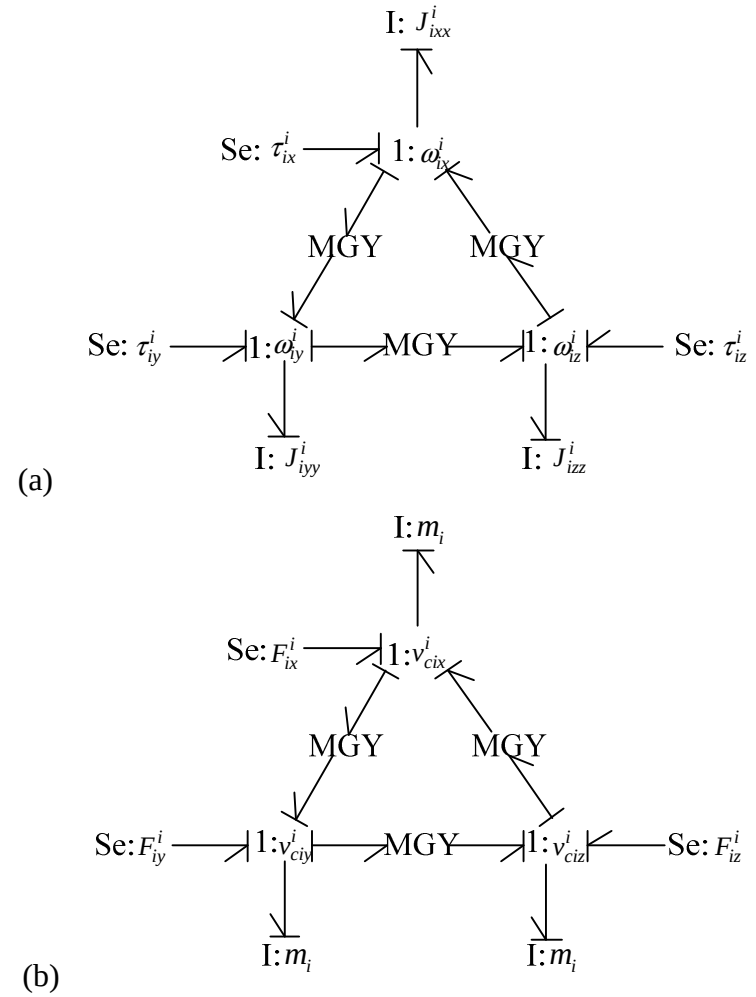


Figure 3.33: Bond graph models of the Newton-Euler's equations (a) EJS-T (b) EJS-F

3.2.3.5 Bond Graph Modeling of a Link with a Joint

The bond graph model of a link including a joint module is illustrated in Figure 3.34. This model is derived based on the exchange of power and movement between the constituents of the physical system corresponding to the Newton–Euler formulation. To represent the dynamics of a link with a joint, the bond graph of Figure 3.34 is augmented with inertia elements, and the linear velocity of the center of mass is described. First, an external velocity source Sf is supposed to act on a 0-junction as the motor angular velocity. At the left side the diagonal moment of inertia matrix J_i^i is connected as an I-element to the l-junction of the rotational velocities in the body frame. At the right side the mass matrix m_i is connected as an I-element to the l-junction of the translational velocities in the body-fixed frame, together with the gravity force source Se. The gyroscopic forces which act between these velocities are represented by the modulated gyrator MGY. Because the Euler equations represent these forces, the MGY is referred to as an EJS. The meanings of all the transformers MTF are shown in Figure 3.34.

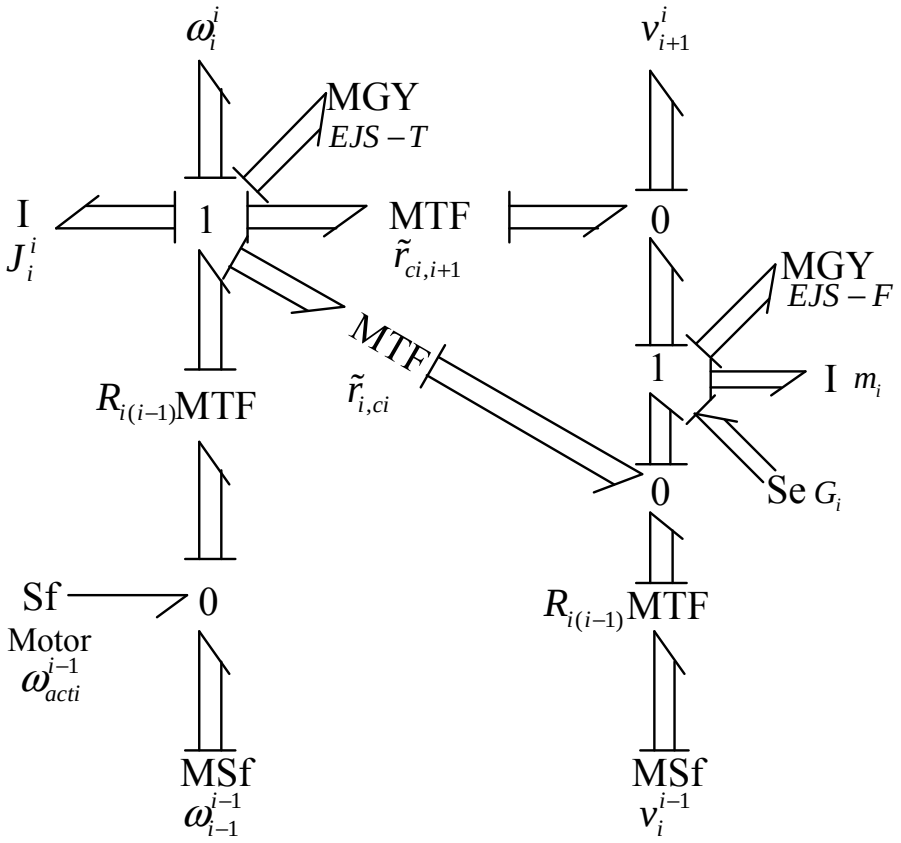


Figure 3.34: Bond graph representation of link *i* with joint *i*

3.2.4 Bond Graph Modeling of Dexterous Hand

3.2.4.1 Bond Graph Modeling of a Finger

As mentioned above, the dexterous hand is composed of five chains of rigid bodies (links) connected by joints. Thus, the bond graph model of the dexterous hand can be considered as a combination of five bond graph models of the fingers.

In this thesis, each finger consists of four joints. The simplified bond graph of a finger is shown in Figure 3.35; similar graphs are implemented on the other fingers.

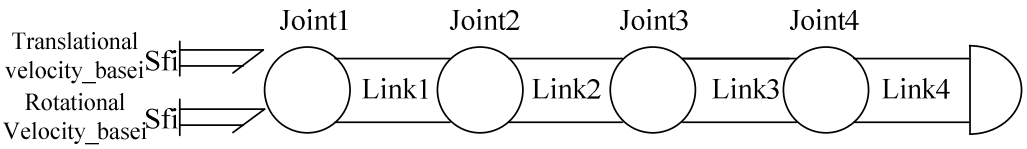


Figure 3.35: Model of finger kinematic chain

A bond graph model of this kinematic chain can be established through the connection of the links and the joints. The joints and the links as the submodels in Figure 3.35 can be replaced by Figure 3.28 and Figure 3.34 respectively. As a rigid body, the bond graph of each link also should connect with EJS which is shown in Figure 3.32 and Figure 3.33. The flow sources are

the translational velocity and the rotational velocity of the base frame Σ_{Si} of finger *i* attached to the palm. These velocities are dependent on the velocity of the palm and the position of frame Σ_{Si} in frame Σ_P.

3.2.4.2 Bond Graph Modeling of Hand

The fingers are mounted on the palm directly. Therefore, once the bond graph model of every single finger is known, the bond graph model of the hand with respect to the palm can be easily constructed on the basis of the constant homogeneous matrices corresponding to the coordinate transformations between the base frames of the five fingers and the palm. Figure 3.36 illustrates the bond graph model of the whole dexterous hand with five fingers.

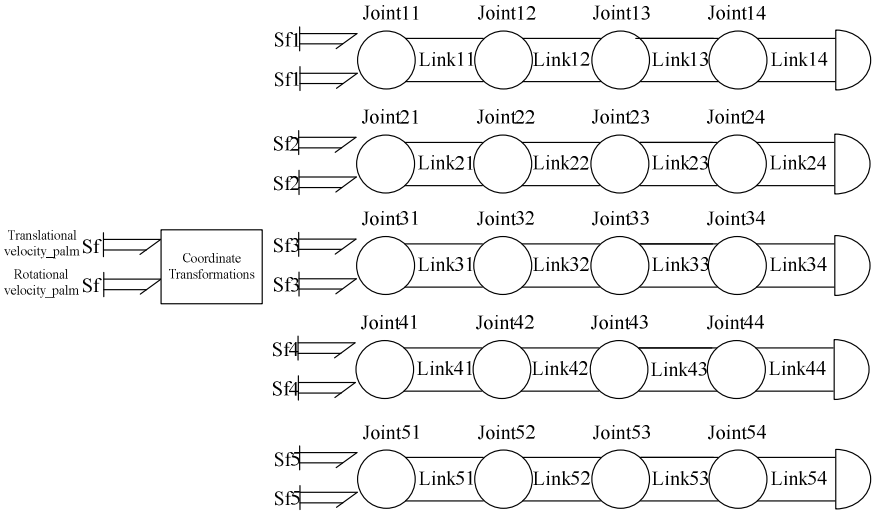


Figure 3.36: Bond graph model of dexterous hand

3.2.5 Modeling of Mobile Platform

A mobile platform consists of left and right wheel drive subsystems and a rigid chassis. The bond graph for a differential wheeled robot is shown in Figure 3.37. The 1-junctions represent the common velocity points, e.g., the rotational velocity of the left wheel $\dot{\theta}_l$, the rotational velocity of the right wheel $\dot{\theta}_r$, the forward velocity of the robot v_c, and the rotational velocity of the rigid chassis ω_c. The 0-junctions represent the common force points, e.g., the forces on the left and right sides of the robot, F_l and F_r. The wheel subsystems include the flow sources (Sf_l and Sf_r), and the inertial elements (I) that model the wheel mass m_w and the inertia I_{wl}, I_{wr}. The resistive elements (R) (with parameters R_l and R_r) model the energy dissipation (i.e., friction) in the wheels. The transformers (TF) model the transformations

between the linear and rotational velocities. The chassis subsystem includes the inertia components that model its mass m_c and rotational inertia I_c.

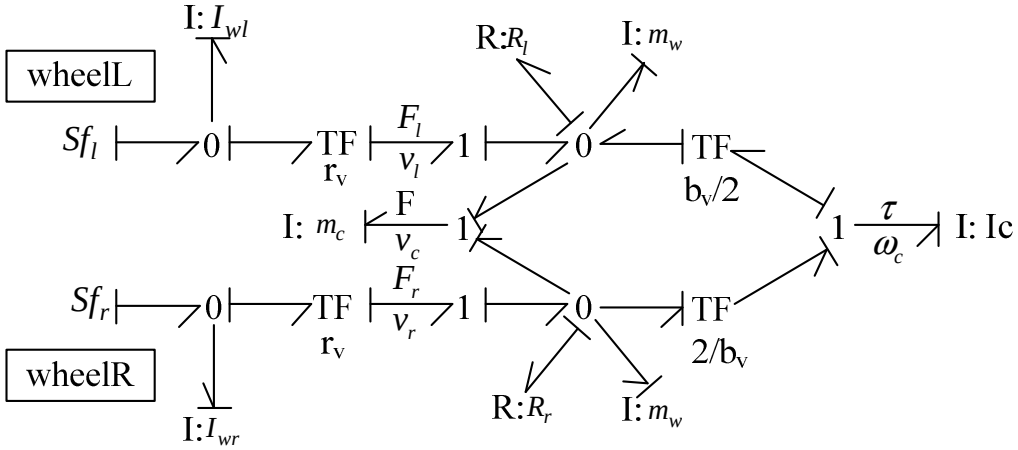


Figure 3.37: Bond graph model of differential wheeled mobile robot

3.2.6 Connection between Mobile Platform and Manipulators

Figure 3.38 shows the bond graph model which connects the manipulators with the mobile platform. The blocks "wheel *i*" and "link 1" are the submodels containing the model of a conventional wheel and a link as discussed earlier. The mobile platform consists of two wheels and a base. All the forces applied by the wheels to the base can be summed. In the bond graph language, this is represented by a 1-junction. The Sf elements (Sf_l and Sf_r) are the flow sources which apply the angular velocities to the axis of the wheels. A R element (resistance) and a C element (stiffness) can be found in the model. These elements only act in the *z* direction of the base and represent the contact with the floor. Since the floor is very stiff and has a high resistance, the values of these two components are high. Without these elements, the robot model will accelerate in the negative *z* direction due to the gravity.

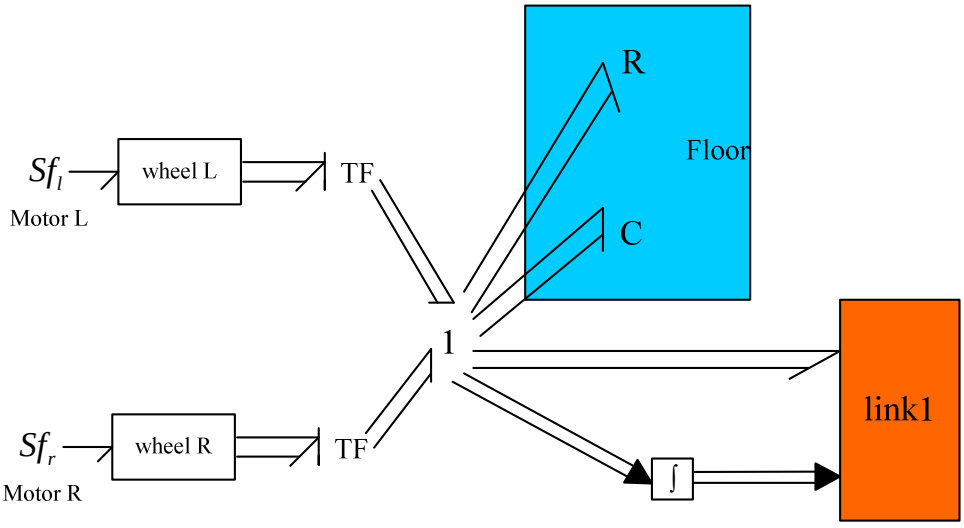


Figure 3.38: Bond graph model of connection between mobile platform and manipulators

As described above, the mobile platform is connected to the first link of the robotic manipulator (the waist). However, in the bond graph model they cannot be connected directly. The model of the mobile platform provides rotation and translation to the first link. These velocities are applied in a point of the link. R_{0B} is the rotation matrix between the first link (frame $\sum_0$) and the world coordinate frame (frame $\sum_B$). ω_0^B and v_0^B are the angular velocity and the linear velocity of the connecting point between the platform and the first link and written in the world coordinate frame.

Angular velocity of joint 1 expressed in frame $\sum_1$:

$$\omega_1^1 = R_{10}(\omega_0^0 + \omega_{act1}^0) = R_{10}R_{0B}\omega_0^B + R_{10}\omega_{act1}^0 \tag{3.118}$$

Linear velocity of joint 1 expressed in frame $\sum_1$:

$$v_1^1 = R_{10}v_0^0 = R_{10}R_{0B}v_0^B \tag{3.119}$$

Since the position of the base is required in calculating R_{0B} , an $\int$-block has to be added. This block calculates the position based on the current position and rotation of the base.

3.2.7 Bond Graph Model Validation

In order to validate the bond graph model of the personal assistant robot some simulations are carried out in this section. All simulations are based on a control system using Matlab. A toolbox BG V.2.1 in Matlab/Simulink is used for simulation and validation. Though many bond graph-based simulation software packages are available, at academic institutions level and in most industries, Matlab/Simulink is the software which is used more often for studying dynamical systems. Using this toolbox the bond graph model is directly built in Simulink. This toolbox only consists of 9 blocks which wholly realize the necessary functionality. Figure 3.39 shows the bond graph model of one link using this toolbox. By comparison, it is obvious that this model is consistent with the bond graph model shown in Figure 3.34.

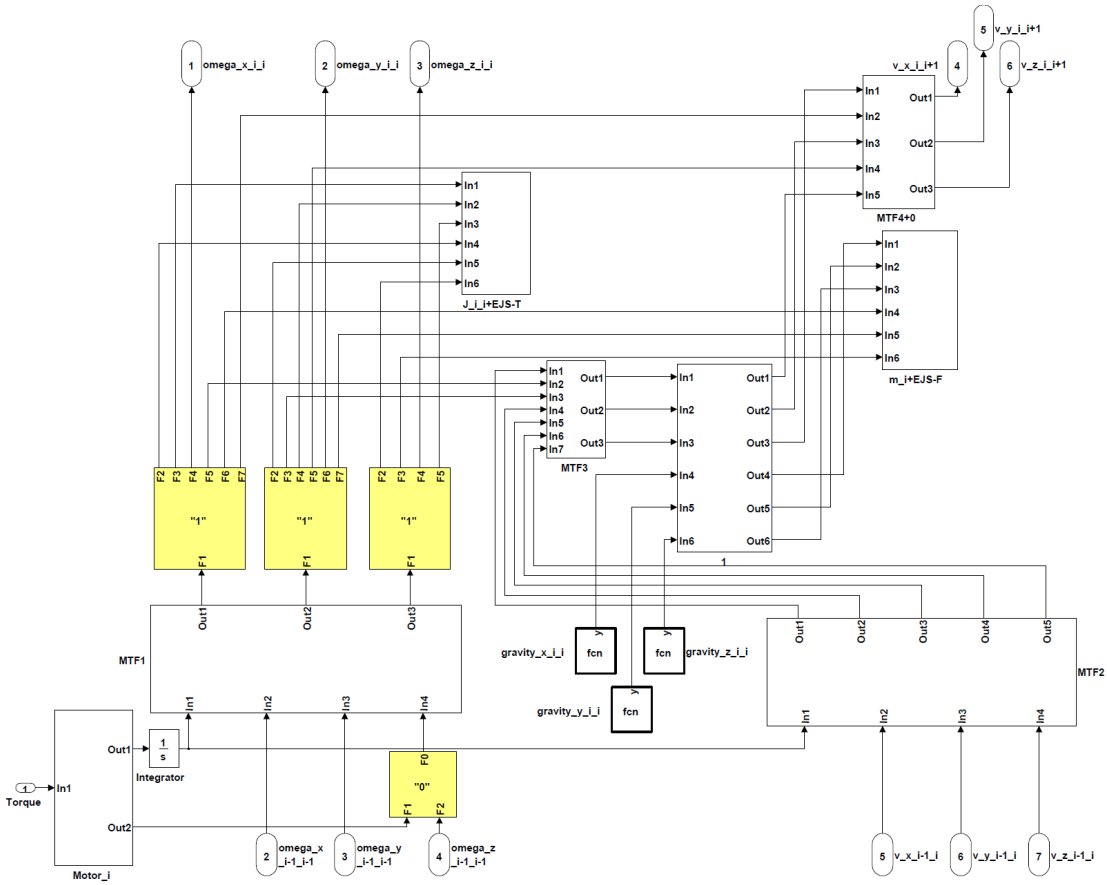
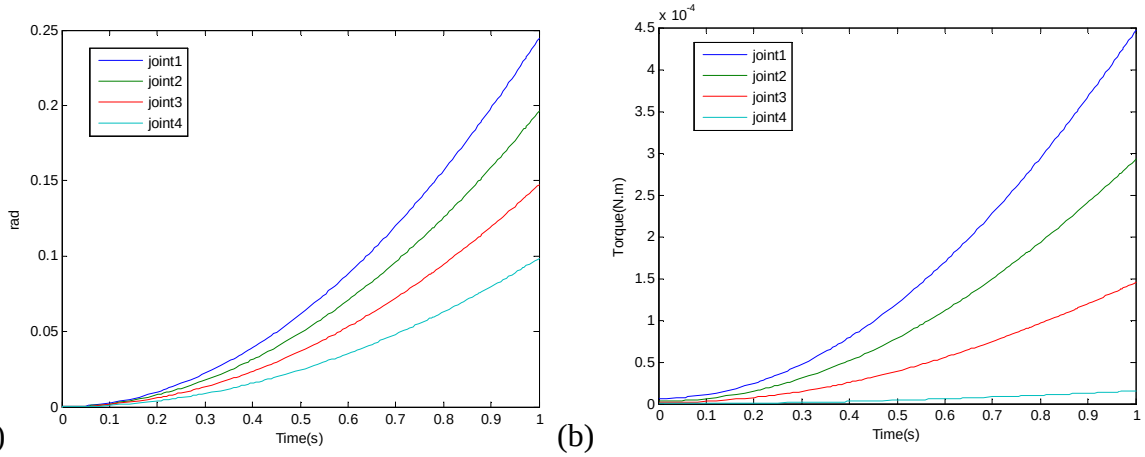


Figure 3.39: Bond graph model of using BG V.2.1

3.2.7.1 Bond Graph Validation of Dexterous Hand

For the validation, we also use the precise mathematical model to calculate the joint torques as the inputs of the hand's bond graph model. The desired joint trajectories for a finger are given as:

$$\begin{bmatrix} q_{f11} & q_{f12} & q_{f13} & q_{f14} \end{bmatrix}^T = \begin{bmatrix} 0.25t^2 & 0.2t^2 & 0.15t^2 & 0.1t^2 \end{bmatrix}^T$$



(a) (b)

Figure 3.40: Simulation results (a) Joint trajectories (b) Joint torques

Figure 3.40(a) gives each joint's trajectory and the torques obtained from the mathematical model are shown in Figure 3.40(b). We can see that the simulation results in the above figures are the same with those shown in Figure 3.25. In other words the bond graph model is accordance with the mathematical model. The other fingers' models can be validated by the same method. The simulation results have shown the effectiveness of the bond graph model of the dexterous hand.

3.2.7.2 Bond Graph Validation of Mobile Manipulator

In the previous subsection we have validated the bond graph model of the dexterous hand. In this subsection we will validate the bond graph model of the mobile manipulator excluding the hands. For simplicity, we also neglect the motions of the neck. This simplified robot has 18 DOFs (a 2 DOFs mobile platform, a 2 DOFs waist and two 7 DOFs arms). Assume no disturbances and uncertainties. We have known that, considering a typical wheeled mobile manipulator subject to kinematic constraints, the dynamics can be in the form:

$$\bar{M}\dot{\zeta} + \bar{C}\zeta + \bar{F} = \bar{E}\tau$$

Assume that all the parameters of the mobile manipulator are known. Based on the accurate mathematical dynamic model, an open-loop control system is established to obtain the joint torques. We apply these torques as the inputs of the bond graph model to track the reference trajectories. Let the desired output trajectories be specified as follows.

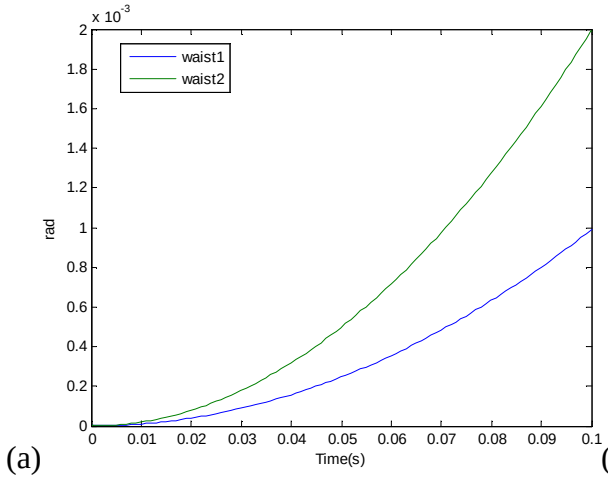
The reference velocity vector for the mobile platform is designed as:

$$\begin{bmatrix} v_d & \omega_d \end{bmatrix}^T = \begin{bmatrix} 0.1t & 0.05t \end{bmatrix}^T$$

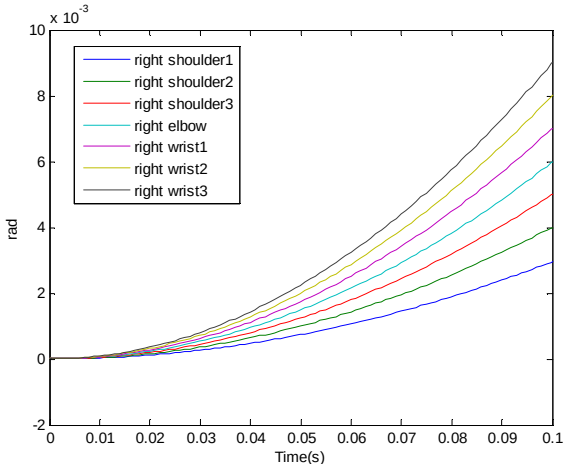
Meanwhile, the reference trajectories for the onboard waist and arms are given by:

$$q_{w1d} = 0.1t^2, q_{w2d} = 0.2t^2, q_{r1d} = 0.3t^2, q_{r2d} = 0.4t^2, q_{r3d} = 0.5t^2, q_{r4d} = 0.6t^2, q_{r5d} = 0.7t^2,$$
$$q_{r6d} = 0.8t^2, q_{r7d} = 0.9t^2, q_{l1d} = -0.3t^2, q_{l2d} = -0.4t^2, q_{l3d} = -0.5t^2, q_{l4d} = -0.6t^2,$$
$$q_{l5d} = -0.7t^2, q_{l6d} = -0.8t^2, q_{l7d} = -0.9t^2$$

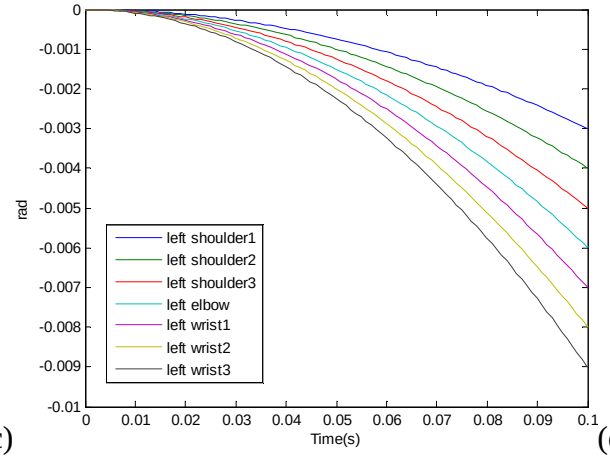
Numerical simulation results illustrated in Figure 3.41 show that the desired trajectories are well tracked through the open-loop control system. This indicates that the bond graph model of the dual-arm mobile manipulator is identical with the mathematical model. The results also demonstrate the usefulness of the bond graph approach in modeling a complex system.


x 10^-3
waist1
waist2
rad
Time(s)

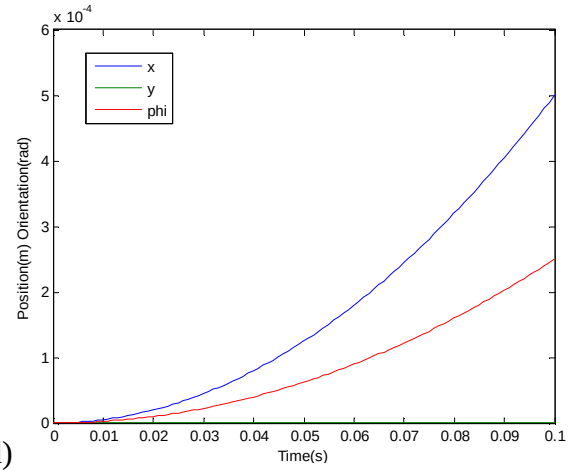
(a)


x 10^-3
right shoulder1
right shoulder2
right shoulder3
right elbow
right wrist1
right wrist2
right wrist3
rad
Time(s)

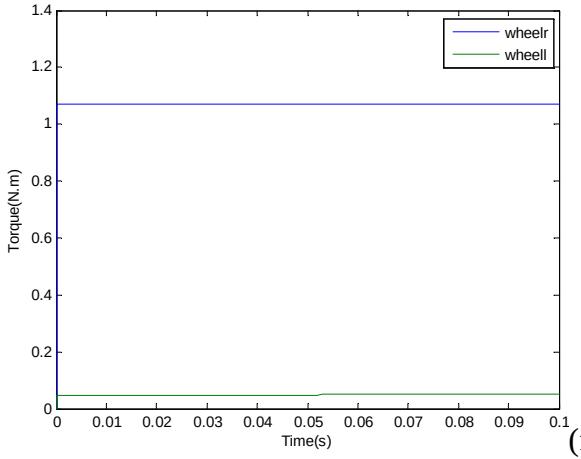
(b)


left shoulder1
left shoulder2
left shoulder3
left elbow
left wrist1
left wrist2
left wrist3
rad
Time(s)

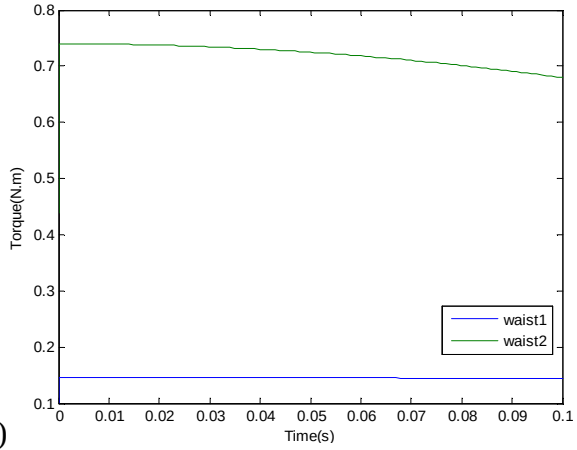
(c)


x 10^-4
x
y
phi
Position(m) Orientation(rad)
Time(s)

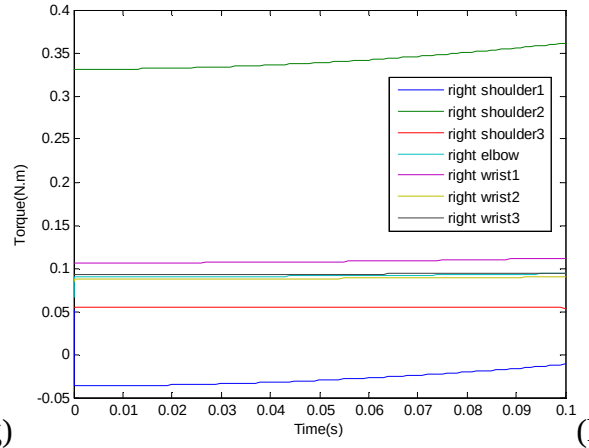
(d)


wheelr
wheell
Torque(N.m)
Time(s)

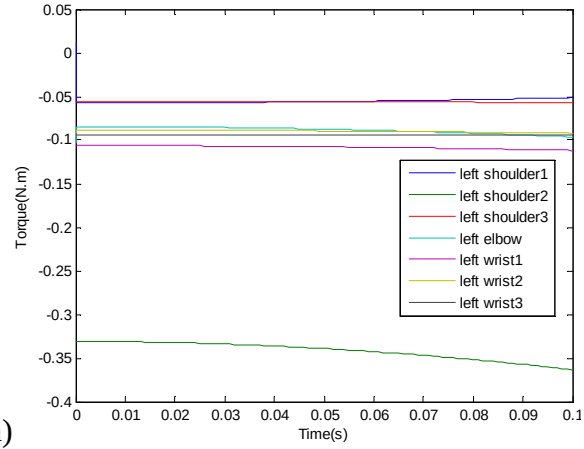
(e)


waist1
waist2
Torque(N.m)
Time(s)

(f)



(g)



(h)

Figure 3.41: Simulation results (a) Joint angles of waist (b) Joint angles of right arm (c) Joint angles of left arm (d) Posture of mobile platform (e) Joint torques of wheels (f) Joint torques of waist(g) Joint torques of right arm (h) Joint torques of left arm

CHAPTER4 MOTION PLANNING AND CONTROL

4.1 Path Planning Approach for Redundant Manipulator

4.1.1 Introduction

Personal assistant robots operating in human-centered environments should be able to plan collision-free motions for grasping and manipulation tasks. In this subchapter the problem of collision-free path planning in robot joint space for a redundant manipulator with the given desired pose (position and orientation) in task space is considered. An example for this kind of problems is described in Figure 4.1.

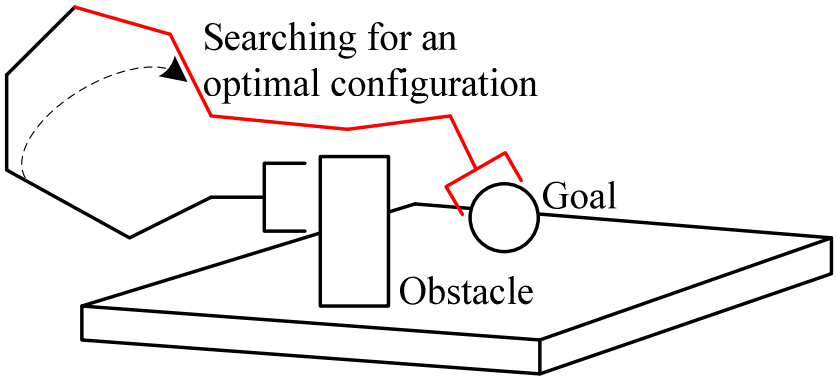


Figure 4.1: An example of collision-free path planning problem

Because of the ability to solve very complex, high dimensional planning problems and the relative ease of implementation, RRTs have been used in a huge range of motion planning problems. JT-RRT was proposed in [138], which interleaves growing a tree randomly and growing a tree towards a goal using Jacobian transpose. An advantage of the method is an analytical solution of IK is not required. [139] proposed two probabilistically complete algorithms, the IKBiRRT and an extended version of the JT-RRT.

In this subchapter, a new algothrim based on RRT is presented. For orientation, quaternion representation is applied to give singularity-free orientation control. A hybrid algorithm of combining RRT method with the Jacobian pseudoinverse algorithm is presented. The nearest configuration point to the task space with the goal pose is taken as the expanded node of the tree. The best expanding direction is calculated with the Jacobian pseudoinverse control algorithm. And the velocity of the end-effector and the joint velocities of the manipulator are restricted by using the bisection gradient-decent extend algorithm to avoid the occurrence of joint velocity mutation.

4.1.2 Problem Formulation

(a) Orientation tracking

Position is well described by a position vector p, but there is no general method to describe the orientation. The application of unit quaternion in robotics has increased in the latter years. Quaternion instead of Euler angles is used to estimate the orientation (pitch, roll and yaw angles) because of its singularity-free orientation representation. It seems to be the preferred method to formulate the end-effector orientation tracking control problem. It was invented by W.R.Hamilton in 1843 [140]. A quaternion has 4 components:

$$\mathbf{q} = (q_0 + iq_1 + jq_2 + kq_3) = q_w + \mathbf{q}_v \tag{4.1}$$

where $q_w = q_0,\ \mathbf{q}_v = (iq_1 + jq_2 + kq_3)$, $i \wedge 2 = j \wedge 2 = k \wedge 2 = -1,\quad jk = -kj = i,\quad ki = -ik = j,$ $ij = -ji = k$ and $\mathbf{q}$ is subject to the constraint:

$$\mathbf{q}^{\mathrm{T}}\mathbf{q} = 1 \tag{4.2}$$

Consider a n DOFs manipulator with generalized coordinates $\theta = \begin{bmatrix}\theta_1 & \dots & \theta_n\end{bmatrix}^T$. In order to distinguish quaternion from generalized coordinates, in this subchapter, the generalized coordinates are denoted by θ.

As mentioned in the kinematic analysis section, the rotation axis can be represented with R_n^0. It also can be represented with the composite product by a unit quaternion as follows [141]:

$$R_n^0(\theta) = (q_w^2 - \mathbf{q}_v{}^T\mathbf{q}_v)I_3 + 2\mathbf{q}_v\mathbf{q}_v{}^T - 2\mathbf{q}_v \times q_w \tag{4.3}$$

where I_3 is a 3×3 identity matrix.

Similarly to (4.3), R_{nd}^0 which represents the desired orientation can be related to a desired unit quaternion, denoted by $\begin{bmatrix}q_{wd} & \mathbf{q}_{vd}\end{bmatrix}^T \in \Re^4$, as follows:

$$R_{nd}^0(\theta) = (q_{wd}^2 - \mathbf{q}_{vd}{}^T\mathbf{q}_{vd})I_3 + 2\mathbf{q}_{vd}\mathbf{q}_{vd}{}^T - 2\mathbf{q}_{vd} \times q_{wd} \tag{4.4}$$

To quantify the difference between the current and the desired end-effector orientations, we define the rotation matrix $\tilde{R} \in \Re^{3\times3}$ as follows:

$$\tilde{\mathrm{R}} = \mathrm{R}\mathrm{R}_d^T \tag{4.5}$$

The unit quaternion tracking error $e_q \in \Re^4$ can be derived as follows [141]:

$$\mathbf{e}_q = \begin{bmatrix} e_w \\ \mathbf{e}_v \end{bmatrix} = \begin{bmatrix} q_w q_{wd} + \mathbf{q}_v{}^T \mathbf{q}_{vd} \\ q_w \mathbf{q}_{vd} - q_{wd} \mathbf{q}_v - \mathbf{q}_v \times \mathbf{q}_{vd} \end{bmatrix} \tag{4.6}$$

which satisfies the constraint:

$$\mathbf{e}_q^T \mathbf{e}_q = e_w^2 + \mathbf{e}_v^T \mathbf{e}_v = 1 \tag{4.7}$$

The control objective is formulated as follows:

$$\mathbf{e}_q = \begin{bmatrix} \pm 1 & 0 & 0 & 0 \end{bmatrix}^T \tag{4.8}$$

Orientation error in this subchapter is defined using the quaternion and the coordinate transformation matrix. For orientation control, in [142], the orientation error is formulated using the unit quaternion as:

$$\mathbf{e}_0 = \mathbf{e}_v \tag{4.9}$$

(b) Path planning of a manipulator

The personal assistant robot is designed to work in human-centered environments where there are many obstacles. So for grasping an object several tasks have to be solved in general, like reaching the desired grasping pose, solving IK and finding a collision-free grasping trajectory. The path planning problem can be described as follows: given an initial configuration θ_{init} in joint space and the desired pose $X_d = \begin{bmatrix} p_d{}^T & \mathbf{q}_d{}^T \end{bmatrix}^T$ of the end-effector in task space, find a continuous collision-free path which satisfies the constraint equation:

$$X_d \;=\; \text{Forwardkinematics } (\theta_{goal}) \tag{4.10}$$

where p_d and $\mathbf{q}_d$ respectively denote the desired position and orientation of the end-effector.

(c) Obstacle detection and avoidance

For obstacle avoidance, it is often demanded to determine the minimum distance from the points on the manipulator to the obstacles is greater than a pre-set minimum value. To generate a feasible path in an obstacle-ridden environment, the algorithm proposed by Sean Quinlan [143] is used for the present problem. The algorithm essentially gives an upper bound for the distance and any point on the geometry of the manipulator can move for a given change in the configuration. This method is proposed for computing some collision-free bubbles around the configuration of the manipulator with revolute joints. These bubbles represent the associated free space (Figure 4.2).

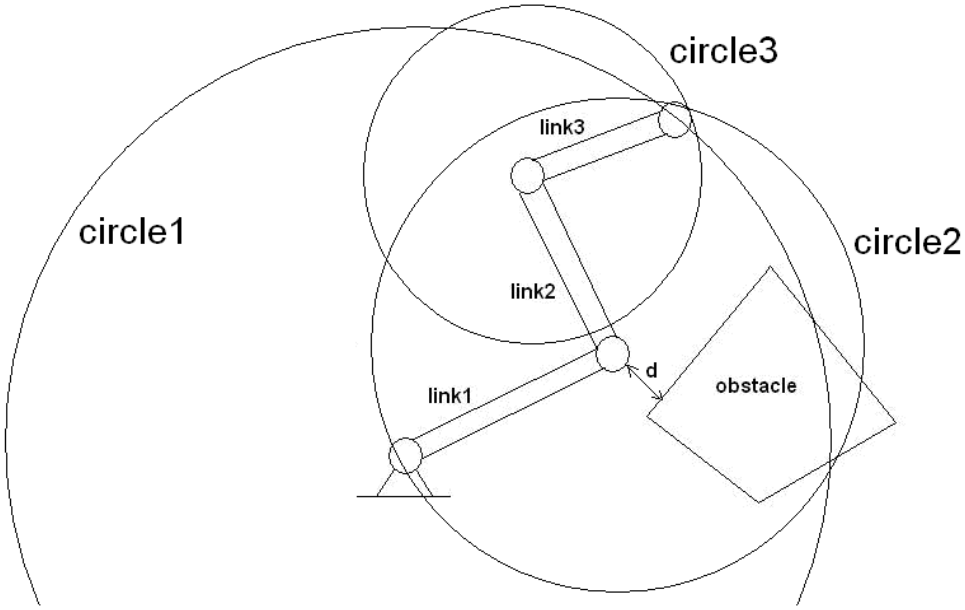


Figure 4.2: Collision-free bubbles

Let d be the minimum distance between the manipulator and the obstacles in the environment. Also, suppose that for each joint of the manipulator, a radius r is determined such that the cylinder centered along the axis of the joint contains all the subsequent links of the manipulator. The principle is shown in Figure 4.2, consider a 3 DOFs manipulator, since the manipulator is planar, the bounding cylinder becomes a circle. Circle one contains the entire manipulator, circle two contains link 2 and link 3, and circle three only contains link 3. Consider moving the manipulator to a configuration θ by first rotating joint 1 to position $\theta_1^{'}$. From the bounding cylinder at joint 1, we know that no part of the manipulator will move a distance greater than $r_1 | \theta_1^{'} - \theta_1 |$. Next, joint 2 moves to $\theta_2^{'}$ (link 1 does not move), and the other links move a distance no greater than $r_2 | \theta_2^{'} - \theta_2 |$. Each joint of the manipulator is moved in turn, and the total distance traveled by these motions is bounded by:

$$\sum_{i=1}^{n} r_i \mid \theta_i^{'} - \theta_i \mid \tag{4.11}$$

We can specify a joint space bubble by:

$$\beta_\theta = \left\{ \theta' : \sum_{i=1}^{n} r_i \mid \theta_i^{'} - \theta_i \mid < d \right\} \tag{4.12}$$

A set of bubbles are used to describe the local free space around the configuration of the manipulator.

4.1.3 Jacobian Pseudoinverse Algorithm

(a) Inverse kinematics solution

The differential kinematic equation, in terms of either the geometric or the analytical Jacobian, establishes a linear mapping between the joint space velocities and the task space velocity, and it can be utilized to find the joint velocities. The differential kinematic equation that expresses the relation between the end-effector velocity and the joint velocities has the following form:

$$V = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} \dot{\theta} \tag{4.13}$$

where $\dot{\theta} = \begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 & ... & \dot{\theta}_n \end{bmatrix}^T$ is the joint velocity vector, $v = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T$ and $\omega = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ are the linear and angular velocity vectors of the end-effector.

J_v can then be written as:

$$J_v = \frac{\partial p}{\partial \theta^T} = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \cdots & \frac{\partial x}{\partial \theta_n} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \cdots & \frac{\partial y}{\partial \theta_n} \\ \frac{\partial z}{\partial \theta_1} & \frac{\partial z}{\partial \theta_2} & \cdots & \frac{\partial z}{\partial \theta_n} \end{bmatrix} \tag{4.14}$$

where $p = \begin{bmatrix} x & y & z \end{bmatrix}^T$ denotes the position of the end-effector.

And J_ω can then be written as:

$$J_\omega = \begin{bmatrix} a_{1x} & a_{2x} & \dots & a_{nx} \\ a_{1y} & a_{2y} & \dots & a_{ny} \\ a_{1z} & a_{2z} & \dots & a_{nz} \end{bmatrix} \tag{4.15}$$

where $\begin{bmatrix} a_{ix} & a_{iy} & a_{iz} \end{bmatrix}^T$ which was mentioned in equation (3.3) denotes the rotation axis z of joint *i* in the base frame.

Based on the kinematic analysis, the end-effector orientation is expressed by the unit quaternion **q** which is related to the angular velocity of the end-effector via the following differential equation [141]:

$$\dot{\mathbf{q}} = B(\mathbf{q})\omega \tag{4.16}$$

Thus, the relation between the joint velocities and the end-effector velocity can also be expressed as follows:

$$\begin{bmatrix} \dot{p} \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} J_v(\theta) \\ B(\mathbf{q})J_\omega(\theta) \end{bmatrix} \dot{\theta},\ J_r = \begin{bmatrix} J_v & B(\mathbf{q})J_\omega \end{bmatrix}^T \tag{4.17}$$

Due to the non-square Jacobian matrix for the redundant manipulator, the basic inverse solution of (4.13) is obtained by using the pseudoinverse $J_r^\dagger$ of the matrix J_r and the inverse solution can then be written as:

$$\dot{\theta} = J_r^\dagger V \tag{4.18}$$

where the pseudoinverse $J_r^\dagger$ can be computed as $J_r^\dagger = J_r^T (J_r J_r^T)^{-1}$.

In the absence of any obstacles, internal collisions, or joint limits, this simple controller is guaranteed to reach the goal. However, the pseudoinverse tends to have stability problems in the neighborhoods of singularities. If the configuration is close to a singularity, then the pseudoinverse method will lead to very large changes in the joint angles, even for small movement in the target position. The pseudoinverse has the further property that the matrix $(I - J_r^\dagger J_r)$ performs a projection onto the null space of J_r.

A common method including the null space in the solution is given by [144]:

$$\dot{x} = J^{\dagger}\dot{y} + (I - J^{\dagger}J)z \tag{4.19}$$

The first term is the particular solution to the inverse problem $J\dot{x} = \dot{y}$, and the second term represents the homogeneous solution to the problem $J(I - J^{\dagger}J)z = 0$.

By using (4.19), the general inverse solution for the redundant manipulator is written as:

$$\dot{\theta} = J_r^{\dagger}V + (I - J_r^{\dagger}J_r)\dot{\varphi} \tag{4.20}$$

where $\dot{\varphi}$ is an arbitrary vector that can be used for optimization purposes and the matrix $(I - J_r^{\dagger}J_r)$ projects the joint vector $\dot{\varphi}$ onto the null space $N(J_r)$.

(b) Redundancy resolution

In velocity based control, given the desired task space position p_d and orientation $\mathbf{q}_d$ (as a quaternion), the linear velocity v and the angular velocity ω in task space are derived with position control and are defined as:

$$v = k_p(p_d - p) \tag{4.21}$$

$$\omega = k_0\mathbf{e}_0 \tag{4.22}$$

where k_p and k_0 are positive.

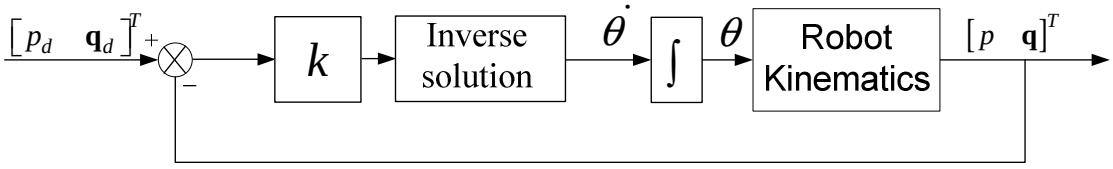


Figure 4.3: Jacobian pseudoinverse algorithm

According to Figure 4.3, we obtain the following control algorithm:

$$\dot{\theta} = kJ_r^{\dagger}e + (I - J_r^{\dagger}J_r)\dot{\varphi} \tag{4.23}$$

where the position and orientation errors in task space are defined as $e = \begin{bmatrix} e_p \\ \mathbf{e}_0 \end{bmatrix} = \begin{bmatrix} p_d - p \\ \mathbf{e}_v \end{bmatrix} \in \Re^6$ and $k = \begin{bmatrix} k_p & 0_{3\times3} \\ 0_{3\times3} & k_0 \end{bmatrix}$.

A typical choice of the null space joint velocity vector is given by [144]:

$$\dot{\varphi} = k_w (\frac{\partial w(\theta)}{\partial \theta})^T \tag{4.24}$$

with $k_w > 0$. $w(\theta)$ is a scalar objective function of the joint variables and $(\frac{\partial w(\theta)}{\partial \theta})^T$ is the vector function representing the gradient of w.

By suitably choosing w, one can try to achieve secondary goals in addition to making the end-effector track the goal pose. In this research, just the joint limit avoidance is considered as the optimization objective, and the objective function is the distance from the mechanical joint limits, which can be defined as:

$$w(\theta) = \frac{1}{n} \sum_{i=1}^{n} (\frac{\theta_i - \overline{\theta}_i}{\theta_{iM} - \theta_{im}})^2 \tag{4.25}$$

where $\theta_{iM}(\theta_{im})$ denotes the maximum (minimum) limit for θ_i, and $\overline{\theta}_i$ is the middle value of the joint range. Thus the redundancy may be exploited to keep the manipulator off the joint limits.

Note that the algorithm requires that all the expected joint angular velocities can be reached, but because of the joint angle constraints and the obstacles, the joint cannot change the configuration in any direction. To solve this problem, in this section, bisection method is used to choose the displacement of the end-effector and determine the feasibility of the joint angular velocities to limit the joint velocities and the velocity of the end-effector.

4.1.4 RRT Method based on Jacobian Pseudoinverse Algorithm

This section improves the basic RRT algorithm and uses this proposed method for the motion planning of the redundant manipulator. RRT is a tree composed of nodes which represent configurations in the search space. The RRT growth process causes it to rapidly expand throughout the space. Considering the characteristic of multi-joint, we use this method in joint space. The basic principle of RRT algorithm can be described as follows.

Figure 4.4 describes an example of how the basic RRT works. The blue circles represent the configurations that are already in the tree. Initially the tree has only one node located at the starting state. Firstly a search tree is generated from this initial configuration θ_{init}, and we take this configuration point as a parent node. Then we choose a point θ_{rand} at random in the configuration space. We find θ_{rand}'s nearest neighbor in the tree and that configuration is denoted as θ_{near}. Next we try to connect θ_{rand} and θ_{near} using a simple local planner, which essentially just moves from one configuration to the other until it encounters an obstacle. If the local planner succeeds in reaching θ_{rand}, it is added as a new node in the tree. This step is repeated until the manipulator reaches the goal. Typically the search is only allowed to proceed in a small step $\delta\theta$, in which case the location reached (called θ_{new}) is added instead. The new node θ_{new} regards θ_{near} as its parent node.

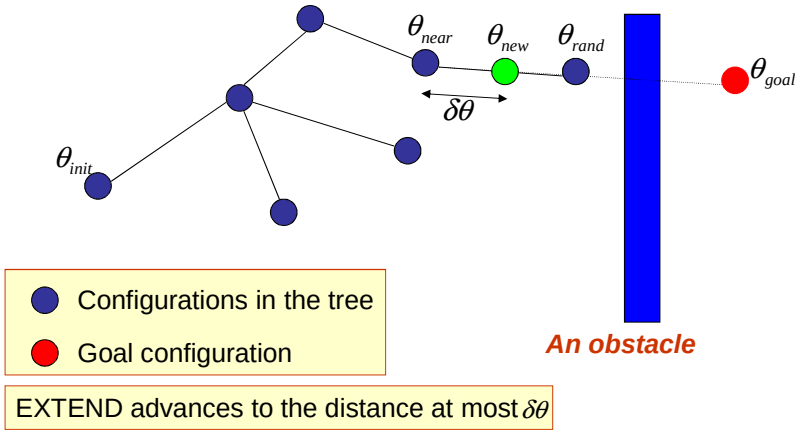


Figure 4.4: Basic RRT principle

Basic RRT algorithm
GENERATE_RRT(θ_{init} , M, Δt)
1. T.init(θ_{init});
2. For m = 1 to M do
3. $\theta_{rand} \leftarrow$ RANDOM_STATE();
4. $\theta_{near} \leftarrow$NEAREST_NEIGHBOR(θ_{rand} , T);
5. EXTEND(θ_{near} , θ_{rand})
6. $\delta\theta \leftarrow$ SELECT_INTPUT(θ_{rand} , θ_{near});
7. $\theta_{new} \leftarrow$ NEW_STATE(θ_{near} , $\delta\theta$, Δt);
8. T.add (θ_{new});
9. Return T;

As can be seen from the above algorithm, before planning, a goal configuration should be given in joint space in advance. In practice, the goal point is usually given in task space, and then the configuration is obtained by inverse kinematics. The pseudoinverse of the Jacobian matrix is used to improve the RRT algorithm as follows.

During the expansion of the search tree, with probability p_r the tree is grown towards the goal and with probability $1-p_r$ it is grown randomly in joint space. When growing randomly, it is exactly as in the basic RRT algorithm. When growing towards the goal, we select the node closest in task space to the goal pose X_{goal} and then use the pseudoinverse of the Jacobian matrix on this node to determine the expansion direction. To avoid the occurrence of mutations of joint and end-effector velocities in the expansion direction, RRT selects a new node along the expansion direction which is determined by the bisection gradient-decent extend algorithm. During the expansion, a failure count of a node is incremented when an attempt to add a new branch fails because the new node is not collision-free (computing collision-free bubbles) or does not yield a lower goal distance than the old node. When the number of failures exceeds the predetermined value, the node is removed from the search tree. This phenomenon indicates that there is not a feasible path from this node to the goal node. The parent node of this removed node is returned. Then RRT repeats the above process and randomly selects a node to expand the search tree until the distance between the end-effector and the goal is less than the predetermined value. The concrete realization of this algorithm is as follows:

① RRT method based on Jacobian pseudoinverse algorithm:

Θ_{new} = { θ_{init} }
while (**DistanceToGoal**(Θ_{new}) > DistanceThreshold)
    p= **Rand** (0, 1)
    If p< pr
        Θ_{new} =**GradientDecentExtend** ( )
    Else
        Θ_{new} =**BasicExtend** ( )

② Gradient DecentExtend ( )
θ_{old} = **ClosestNodeToGoal** ( );
Xold = **KinematicsSolve** (θ_{old}) //End-effector position and orientation

ΔX = Xgoal – Xold //Distance to goal

For j = 0: 1: N //Bisection gradient-decent extend

ΔXstep =ΔP / 2^j

If |ΔXstep | <ΔXmax //Limiting the velocity of end-effector

Θ_{new} = **ComputeNewNode** (θ_{old} , ΔXstep)

Θ_{new} = Θ_{new} + { θ_{new} }

Break

Else

Continue

③ Expansion procedure:

ComputeNewNode (θ_{old} , ΔXstep)

If |ΔXstep | <ε

return θ_{old}

Else

J =**ComputeJacobian** (θ_{old})

$\Delta\theta = kJ_r^{\dagger}\Delta\text{Xstep} + (I - J_r^{\dagger}J_r)\dot{\varphi}$

If $|\Delta\theta| < \Delta\theta\max$ //Avoiding a sudden change of joints

θ_{new} = θ_{old} +$\Delta\theta$

If **CollisionFree**$(\theta_{new}) \cap (\text{DistanceToGoal}(\theta_{new}) - \text{DistanceToObstacle}(\theta_{new}))$

$< (\text{DistanceToGoal}(\theta_{near}) - \text{DistanceToObstacle}(\theta_{near}))$

return θ_{new}

Else

IncreaseFailureCount (θ_{old} , 1)

If failureCount > f

remove (θ_{old})

θ_{old} ⟵ **Parent**(θ_{old})

return θ_{old}

Else

Continue

Else

Continue

where function **KinematicsSolve** ( ) gives the forward kinematics solution for the robot.

4.2 Path Planning Approach for Mobile Platform

[145] extended the basic RRT concept to create an algorithm called RRT-Connect. RRT-Connect which is a variation of RRT is applied to the path planning problem of a mobile robot that works in an environment with obstacles. Figure 4.5 is an example how RRT-Connect works. The planner maintains two RRTs, denoted T_a and T_b rooted at q_{init} and q_{goal} respectively. Their nodes are depicted in different colors. At the beginning, the trees only contain the initial and the goal state, respectively. The planner starts by picking a random node and extends the tree on the left to this random node. Then, the tree on the right hand side is extended to the newly added node in the left tree. The planner then switches the roles of the two trees: in the following iteration, we extend the tree on the right hand side and extend the tree on the left to the new node in the right tree. Eventually, the two trees meet and the planner returns the corresponding trajectory.

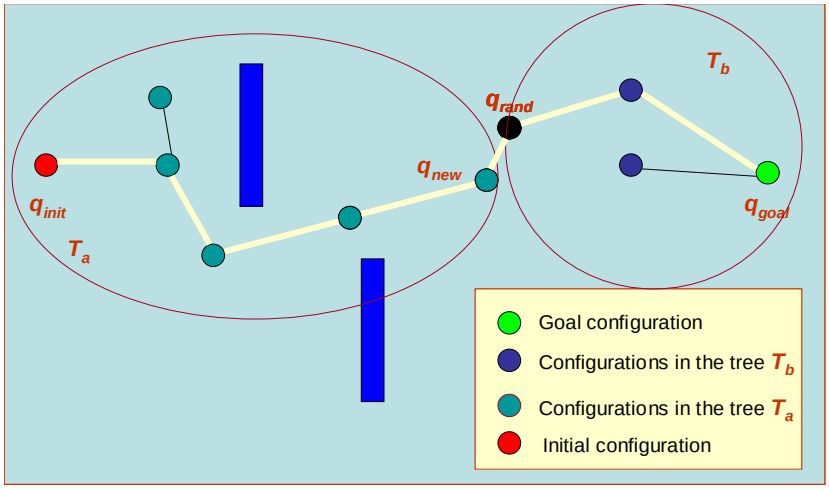


Figure 4.5: An example of RRT-Connect

Although RRT-Connect algorithm has been proved to be very successful, we still can find an optimal path in a local search task. Figure 4.6 illustrates this problem. In Figure 4.6, *A*, *B* and *C* are three adjacent nodes. When the robot moves from *A* to *C*, the local path searched by RRT-Connect is composed of segments *AB* and *BC*. But if the robot moves along the segments *AD* and *DC*, obviously, the length will be shortened. Consequently, this path is closer to the optimal path. Here, a path optimization algorithm based on modified node enhancing strategy is proposed. The node enhancing method is used to optimize the initial planning by using the basic RRT-Connect algorithm, the original path nodes are gradually substituted by some new nodes, and the number of the inflection points on the path will be reduced greatly, thus the path length will be shorten.

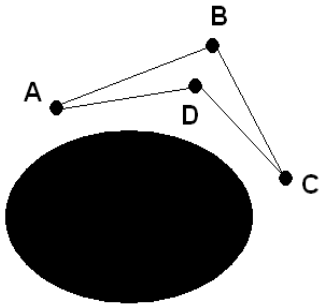


Figure 4.6: Local path

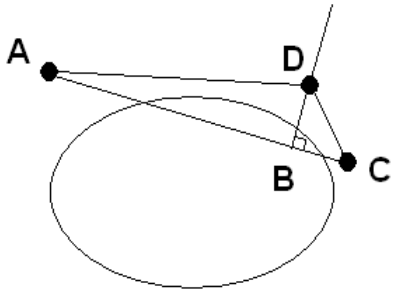


Figure 4.7: Principle of node enhancing method

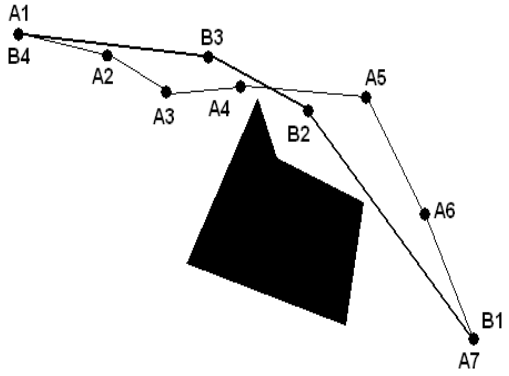


Figure 4.8: Improved node enhancing method

Node enhancing method chooses some new nodes to replace some old nodes in the original path in order to decrease the inflection points and then shorten the length of path. The basic principle is shown in Figure 4.7. The optimization steps are as follows:

- We adopt binary search algorithm to detect collision on the segment *AC*. If one point is detected collide with some obstacles, we stop the detection and record the collision point denoted as *B*.

- We take *B* as the pedal to draw a vertical line of segment *AC* and take *h* as unit length to extend the line along both sides of *AC* in turn in order to find point *D*. *D* is the first point that makes segment *AD* and segment *CD* satisfy the collision free detection. Unit length *h* is the smaller the better, but at the same time the computational cost will become large. So *h* must be properly selected. As mentioned above, we adopt alternate collision detection along both sides of *AC* which lowers the computational cost.

On the basis of basic node enhancing method, we improve the above algorithm. Figure 4.8 depicts this improved method. Line *A* which is composed of points A_i is the original path before optimization. It contains seven nodes. From left to right, they are marked as A_1, A_2, A_3, A_4, A_5, A_6 and A_7. Now we connect A_7 and other points (from A_6 to A_1) in sequence and successively determine the feasibility of A_7A_6, A_7A_5… until encountering the first node which is not feasible (A_4 in the figure). Here, the feasibility means collision-free. The segment A_7A_4 collides with an obstacle. A new node B_2 is added to the path between A_7 and A_4 according to

node enhancing method. Then the nodes between A_7 and A_5 are deleted (including A_5). Taking A_7 as B_1 and taking B_2 as the start point, the above search is executed again until encountering an infeasible point (A_3). Then we use the same method to add B_3 to the path. The nodes between B_2 and A_3 are deleted (including A_3). On the analogy of this, we find new nodes until arriving the endpoint and denote the endpoint as B_{i+1}. At last, a new path is found (line B in Figure 4.8).

4.3 RBF Neural Network Adaptive Control of Mobile Manipulator

4.3.1 Introduction

We have known that it is difficult to control a mobile manipulator due to nonholonomic constraints on the mobile platform and the dynamic interaction, especially when the dynamics of the robot is partially or completely unknown or there exist disturbances. NN has strong self-learning ability and has good capability for the identification of complex nonlinear dynamic systems. In resent years there has been a growing interest in the motion control using NNs. A NN-based methodology is developed in [11] for the motion control of mobile manipulators subject to kinematic constraints. The dynamics of the mobile manipulator is assumed to be completely unknown, and is identified on-line by the NN estimators. In [146] an adaptive hybrid controller is proposed. This controller consists of two parts: one is for the tracking control of the mobile platform in kinematics. The other part is for the robotic arm in dynamics. RBFNN is adopted for unmodeled dynamics and external disturbances.

In this subchapter, a new motion control method using RBFNN, the GL matrix and its product operator is presented. The dynamics of the mobile manipulator is assumed to be completely unknow and there exist some disturbances. The difference between this proposed method and those proposed in [11] and [146] is that this new control law is not only applicable when the dynamics of the robot is completely unknown, but also capable of approximating all uncertain inertial system parameters separately.

4.3.2 RBF Neural Network Modeling of Mobile Manipulator

This section discusses the NN modeling of a mobile manipulator. Assume that the mobile manipulator has p generalized coordinates denoted as q and the dynamic parameters of the robot are completely unknown. Define p as $p = m + n$, where m and n are the numbers of generalized coordinates of the mobile base and the manipulator respectively. A RBFNN is used to identify the unknown mobile manipulator dynamics and disturbances. For

differentiating the number of nonholonomic constraints from the error dynamics, in this chapter, the nonholonomic kinematic constraint matrix $A_v(q_v)$ is assumed to have full rank r'.

We have already known that a general nonlinear function $f(x)$ can be approximated by a RBFNN as:

$$f(x) = W^T \varphi(x) + \varepsilon(x) \tag{4.26}$$

with $\varepsilon(x)$ a NN functional reconstruction error vector.

From the dynamic equation (3.14), both $\bar{M}(q)$ and $\bar{F}(q)$ are only the functions of q, hence, static NNs are sufficient to model each element in the matrices [147]:

$$m_{kj}(q) = \sum_l \theta_{kjl} \xi_{kjl}(q) + \varepsilon_{mkj}(q) = \theta_{kj}^T \xi_{kj}(q) + \varepsilon_{mkj}(q) \tag{4.27}$$

$$f_k(q) = \sum_l \beta_{kl} \sigma_{kl}(q) + \varepsilon_{fk}(q) = \beta_k^T \sigma_k(q) + \varepsilon_{fk}(q) \tag{4.28}$$

where θ_{kjl}, $\beta_{kl} \in \Re$ are the weights of the NNs; $\xi_{kjl}(q)$, $\sigma_{kl}(q) \in \Re$ are the corresponding Gaussian basis functions with input vector q; $\varepsilon_{mkj}(q)$, $\varepsilon_{fk}(q) \in \Re$ are respectively the modeling errors of $m_{kj}(q)$ and $f_k(q)$, and they are assumed to be bounded.

Whereas, $\bar{C}(q, \dot{q})$ is modeled using dynamic NN with the inputs q and $\dot{q}$:

$$c_{kj}(q, \dot{q}) = \sum_l \alpha_{kjl} \varsigma_{kjl}(z) + \varepsilon_{ckj}(z) = \alpha_{kj}^T \varsigma_{kj}(z) + \varepsilon_{ckj}(z) \tag{4.29}$$

where $z = \begin{bmatrix} q^T & \dot{q}^T \end{bmatrix}^T \in \Re^{2p}$; $\alpha_{kjl} \in \Re$ is the weight; $\xi_{kjl}(z) \in \Re$ is the corresponding Gaussian basis function with input vector z; $\varepsilon_{ckj}(z)$ is the modeling error of $c_{kj}(q, \dot{q})$, and it is also assumed to be bounded.

$$m_{kj}(q) = \theta_{kj}^T \xi_{kj}(q) + \varepsilon_{mkj}(q) \tag{4.30}$$

$$c_{kj}(q, \dot{q}) = \alpha_{kj}^T \varsigma_{kj}(z) + \varepsilon_{ckj}(z) \tag{4.31}$$

$$f_k(q) = \beta_k^T \sigma_k(q) + \varepsilon_{fk}(q) \tag{4.32}$$

$\bar{M}$, $\bar{C}$, $\bar{F}$ can be rewritten using GL matrix [147] as:

$$\bar{M} = \left[\{\Theta\}^T \bullet \{\Xi(q)\} \right] + E_{\bar{M}}(q) \tag{4.33}$$

$$\bar{C} = \left[\{A\}^T \bullet \{Z(z)\} \right] + E_{\bar{C}}(z) \tag{4.34}$$

$$\bar{F} = \left[\{B\}^T \bullet \{H(q)\} \right] + E_{\bar{F}}(q) \tag{4.35}$$

where Θ, $\Xi(q)$, A, $Z(z)$, B, $H(q)$ are respectively the GL matrices and vectors with the elements θ_{kj}, α_{kj}, β_k, ξ_{kj}, ς_{kj}, σ_k ; $E_{\bar{M}}(q)$, $E_{\bar{C}}(z)$, $E_{\bar{F}}(q)$ are the matrices and vector composed of the modeling errors ε_{mkj}, ε_{ckj}, ε_{fk}. The ability of RBFNN to approximate continuous functions has been widely studied [148].

4.3.3 Controller Design

We consider the control of a mobile manipulator subject to kinematic constraints when the parameters of the dynamic model are completely unknown. Unknown dynamic parameters and disturbances should be identified and compensated in closed-loop control using RBFNN.

For a differential wheeled mobile platform, let it track the reference trajectory $q_{vr} = \begin{bmatrix} x_r & y_r & \phi_r \end{bmatrix}^T$ and the reference velocity $V_r = \begin{bmatrix} v_r & \omega_r \end{bmatrix}^T$, where $\begin{bmatrix} x_r & y_r \end{bmatrix}^T$ are the desired coordinates of the center of mass in the world coordinate system; ϕ_r is the desired heading angle of the platform measured from the x-axis of the world coordinate system; v_r and ω_r are the desired linear and angular velocities of the mobile platform.

Define the tracking error as:

$$\tilde{q}_v = \begin{bmatrix} \tilde{x}_c & \tilde{y}_c & \tilde{\phi} \end{bmatrix}^T \tag{4.36}$$

where $\tilde{x}_c = x_r - x_c$, $\tilde{y}_c = y_r - y_c$, and $\tilde{\phi} = \phi_r - \phi$.

Representing this tracking error in the moving frame xyz as:

$$e_v = \begin{bmatrix} e_x & e_y & e_\phi \end{bmatrix}^T = T_r(\phi) \cdot \tilde{q}_v \tag{4.37}$$

where $T_r(\phi) = \begin{bmatrix} cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

To achieve the asymptotical position tracking ability of the mobile platform, the velocity control law for the system is adopted as [149]:

$$\alpha = \begin{bmatrix} v_d \\ \omega_d \end{bmatrix} = \begin{bmatrix} v_r \cos e_\phi - k_x e_x \\ \omega_r + k_\phi \sin e_\phi + v_r k_y e_y \end{bmatrix} \tag{4.38}$$

where k_x, k_y and k_ϕ are positive constants.

Assume $\eta_d(t) = \begin{bmatrix} v_d & \omega_d & \dot{q}_{1d} & \cdots & \dot{q}_{nd} \end{bmatrix}^T$ is the desired velocity vector of the mobile manipulator and $\eta(t) = \begin{bmatrix} v_c & \omega_c & \dot{q}_1 & \cdots & \dot{q}_n \end{bmatrix}^T$ is the practical velocity vector of the mobile manipulator, define:

$$\dot{e}(t) = \eta_d(t) - \eta(t) \tag{4.39}$$

$$r(t) = \dot{e} + \Lambda \int \dot{e} \tag{4.40}$$

where Λ is a positive definite matrix.

Assumption: The desired reference trajectory $\int \eta_d(t)$ is assumed to be bounded and uniformly continuous and has bounded and uniformly continuous derivatives up to the second order.

Lemma 4.1: Let $e(t) = h(t) * r(t)$ where "*" denotes convolution product and $h(t) = L^{-1}(H(s))$ with $H(s)$ is an $n \times n$ strictly proper, exponentially stable transfer function. Then $r \in L_2^n$ implies that $e \in L_2^n \cap L_\infty^n$, $\dot{e} \in L_2^n$, e is continuous, and $e \to 0$ as $t \to \infty$, if, in addition, $r \to 0$ as $t \to \infty$ then $\dot{e} \to 0$ [150], where $L_2 \triangleq \left\{ \|f\|_2 = \int_0^\infty |f|^2 \, dt < \infty, \text{finite energy} \right\}$, $L_\infty \triangleq \left\{ \|f\|_\infty = \sup_{t \in [0,\infty)} |f| < \infty, \text{bounded signal} \right\}$.

Define the tracking errors as

$$r = \eta_r - \eta \tag{4.41}$$

which lead to their time derivatives:

$$\dot{r} = \dot{\eta}_r - \dot{\eta} \tag{4.42}$$

The controller is proposed as follows:

$$S^T E\tau = \hat{\bar{M}}\dot{\eta}_r + \hat{\bar{C}}\eta_r + \hat{\bar{F}} + Kr + U_s \tag{4.43}$$

where $K \in R^{(p-r')\times(p-r')} > 0$ is a positive controller gain and $\hat{\bar{M}}$, $\hat{\bar{C}}$, $\hat{\bar{F}}$ represent the estimates of the true parameter matrices and vector. The first three terms of the control law are the model-based control, whereas the term Kr gives the PD control. The last term U_s is the robust controller for compensating for the approximation errors and uncertainties.

Applying the control law to the mobile manipulator dynamics yields the tracking error equation:

$$\begin{aligned} S^T E\tau &= \left\{\left[\{\Theta\}^T \bullet \{\Xi(q)\}\right] + E_{\bar{M}}(q)\right\}\dot{\eta} + \left\{\left[\{A\}^T \bullet \{Z(z)\}\right] + E_{\bar{C}}(z)\right\}\eta \\ &\quad + \left[\{B\}^T \bullet \{H(q)\}\right] + E_{\bar{F}}(q) + \tau_d \\ &= \left[\left\{\hat{\Theta}\right\}^T \bullet \left\{\hat{\Xi}(q)\right\}\right]\dot{\eta}_r + \left[\left\{\hat{A}\right\}^T \bullet \left\{\hat{Z}(z)\right\}\right]\eta_r + \left[\left\{\hat{B}\right\}^T \bullet \left\{\hat{H}(q)\right\}\right] \\ &\quad + Kr + U_s \end{aligned} \tag{4.44}$$

where $\hat{\Xi} = \Xi - \tilde{\Xi},\ \hat{Z} = Z - \tilde{Z},\ \hat{H} = H - \tilde{H}$.

Substituting the above three transformations into (4.44) yields:

$$\begin{aligned} &\left[\left\{\hat{\Theta}\right\}^T \bullet \left\{\hat{\Xi}(q)\right\}\right]\dot{\eta}_r + \left[\left\{\hat{A}\right\}^T \bullet \left\{\hat{Z}(z)\right\}\right]\eta_r + \left[\left\{\hat{B}\right\}^T \bullet \left\{\hat{H}(q)\right\}\right] + Kr + U_s \\ &= \left[\left\{\hat{\Theta}\right\}^T \bullet \{\Xi(q)\}\right]\dot{\eta}_r + \left[\left\{\hat{A}\right\}^T \bullet \{Z(z)\}\right]\eta_r + \left[\left\{\hat{B}\right\}^T \bullet \{H(q)\}\right] \\ &- \left[\left\{\hat{\Theta}\right\}^T \bullet \{\tilde{\Xi}(q)\}\right]\dot{\eta}_r - \left[\left\{\hat{A}\right\}^T \bullet \{\tilde{Z}(z)\}\right]\eta_r - \left[\left\{\hat{B}\right\}^T \bullet \{\tilde{H}(q)\}\right] + Kr + U_s \end{aligned} \tag{4.45}$$

Using (4.45), (4.41), (4.42) and rewriting (4.44) yield:

$$\begin{aligned}
&\left\{\left[\{\Theta\}^T \bullet \{\Xi(q)\}\right] + E_{\bar{M}}(q)\right\}(\dot{\eta}_r - \dot{r}) + \left\{\left[\{A\}^T \bullet \{Z(z)\}\right] + E_{\bar{C}}(z)\right\}(\eta_r - r) \\
&+\left[\{B\}^T \bullet \{H(q)\}\right] + E_{\bar{F}}(q) + \tau_d \\
&= \left[\{\hat{\Theta}\}^T \bullet \{\Xi(q)\}\right]\dot{\eta}_r + \left[\{\hat{A}\}^T \bullet \{Z(z)\}\right]\eta_r + \left[\{\hat{B}\}^T \bullet \{H(q)\}\right] \\
&-\left[\{\hat{\Theta}\}^T \bullet \{\tilde{\Xi}(q)\}\right]\dot{\eta}_r - \left[\{\hat{A}\}^T \bullet \{\tilde{Z}(z)\}\right]\eta_r - \left[\{\hat{B}\}^T \bullet \{\tilde{H}(q)\}\right] + Kr + U_s
\end{aligned} \tag{4.46}$$

Simplifying the above equation:

$$\begin{aligned}
&\left\{\left[\{\Theta\}^T \bullet \{\Xi(q)\}\right] + E_{\bar{M}}(q)\right\}\dot{r} + \left\{\left[\{A\}^T \bullet \{Z(z)\}\right] + E_{\bar{C}}(z)\right\}r \\
&+Kr + U_s \\
&= \left[\{\tilde{\Theta}\}^T \bullet \{\Xi(q)\}\right]\dot{\eta}_r + \left[\{\tilde{A}\}^T \bullet \{Z(z)\}\right]\eta_r + \left[\{\tilde{B}\}^T \bullet \{H(q)\}\right] \\
&+\left[\{\hat{\Theta}\}^T \bullet \{\tilde{\Xi}(q)\}\right]\dot{\eta}_r + \left[\{\hat{A}\}^T \bullet \{\tilde{Z}(z)\}\right]\eta_r + \left[\{\hat{B}\}^T \bullet \{\tilde{H}(q)\}\right] \\
&+E_1 + \tau_d
\end{aligned} \tag{4.47}$$

where $\tilde{\Theta} = \Theta - \hat{\Theta},\ \tilde{A} = A - \hat{A},\ \tilde{B} = B - \hat{B},\ E_1 = E_{\bar{M}}(q)\dot{\eta}_r + E_{\bar{C}}(z)\eta_r + E_{\bar{F}}(q)$.

Then we obtain:

$$\begin{aligned}
&\bar{M}(q)\dot{r} + \bar{C}(q,\dot{q})r + Kr + U_s \\
&= \left[\{\tilde{\Theta}\}^T \bullet \{\Xi(q)\}\right]\dot{\eta}_r + \left[\{\tilde{A}\}^T \bullet \{Z(z)\}\right]\eta_r + \left[\{\tilde{B}\}^T \bullet \{H(q)\}\right] \\
&+\left[\{\hat{\Theta}\}^T \bullet \{\tilde{\Xi}(q)\}\right]\dot{\eta}_r + \left[\{\hat{A}\}^T \bullet \{\tilde{Z}(z)\}\right]\eta_r + \left[\{\hat{B}\}^T \bullet \{\tilde{H}(q)\}\right] \\
&+E_1 + \tau_d
\end{aligned} \tag{4.48}$$

The uncertain term $E = E_1 + E_2 + \tau_d$ is assumed to be bounded by $\|E\| \le \psi$, and the upper bound ψ can be estimated on-line. E_2 is the higher-order term which will be discussed later.

The robust controller U_s is proposed as:

$$U_s = \hat{\psi}\,\mathrm{sgn}(r) \tag{4.49}$$

where $\hat{\psi}$ is the estimated value of the uncertain term bound ψ.

In this section, Gaussian function parameters centers c and widths b are the estimates of the optimal parameter vectors for the NN and are updated on-line. The hidden-layer output error $\tilde{\varphi}$ can be approximated in terms of Taylor series [151]:

$$\tilde{\varphi} = \varphi - \hat{\varphi} = \varphi_c \tilde{c} + \varphi_b \tilde{b} + O \tag{4.50}$$

where

$$\varphi_c = \left[\frac{\partial \varphi_1}{\partial c}, \frac{\partial \varphi_2}{\partial c}, \ldots, \frac{\partial \varphi_{n_h}}{\partial c} \right]_{c=\hat{c}}$$
$$\varphi_b = \left[\frac{\partial \varphi_1}{\partial b}, \frac{\partial \varphi_2}{\partial b}, \ldots, \frac{\partial \varphi_{n_h}}{\partial b} \right]_{b=\hat{b}}$$

n_h is the number of the hidden neurons; O denotes a vector containing the higher-order terms.

The parameter adaptation laws of the NN controller are updated on-line as follows:

$$\begin{aligned}
&\dot{\hat{\theta}}_k = \Gamma_k \bullet \{\xi_k(q)\} \dot{\eta}_r r_k \\
&\dot{\hat{\alpha}}_k = Q_k \bullet \{\varsigma_k(z)\} \eta_r r_k \\
&\dot{\hat{\beta}}_k = N_k \bullet \{\sigma_k(q)\} r_k \\
&\dot{\hat{\psi}}_k = T_k r_k \operatorname{sgn}(r_k) \\
&\dot{\hat{c}}_{\theta_k} = E_{\theta_k} \bullet \{\xi_{kc}\} \hat{\theta}_k \dot{\eta}_r r_k,\ \dot{\hat{c}}_{\alpha_k} = E_{\alpha_k} \bullet \{\varsigma_{kc}\} \hat{\alpha}_k \eta_r r_k,\ \dot{\hat{c}}_{\beta_k} = E_{\beta_k} \bullet \{\sigma_{kc}\} \hat{\beta}_k r_k, \\
&\dot{\hat{b}}_{\theta_k} = I_{\theta_k} \bullet \{\xi_{kb}\} \hat{\theta}_k \dot{\eta}_r r_k,\ \dot{\hat{b}}_{\alpha_k} = I_{\alpha_k} \bullet \{\varsigma_{kb}\} \hat{\alpha}_k \eta_r r_k,\ \dot{\hat{b}}_{\beta_k} = I_{\beta_k} \bullet \{\sigma_{kb}\} \hat{\beta}_k r_k.
\end{aligned} \tag{4.51}$$

where

$\Gamma_k = \Gamma_k^T > 0, Q_k = Q_k^T > 0, N_k = N_k^T > 0, T_k \in \Re > 0, E_{\theta_k} = E_{\theta_k}^T > 0, E_{\alpha_k} = E_{\alpha_k}^T > 0, E_{\beta_k} = E_{\beta_k}^T > 0,$ $I_{\theta_k} = I_{\theta_k}^T > 0$, $I_{\alpha_k} = I_{\alpha_k}^T > 0$, $I_{\beta_k} = I_{\beta_k}^T > 0$, $\hat{\theta}_k$ and $\hat{\alpha}_k$ are the vectors of $\hat{\theta}_{kj}$ and $\hat{\alpha}_{kj}$ respectively.

Theorem: Using the control law (4.43), if the parameters are updated by (4.51), then

- r converges to zero as t→∞.
- e and $\dot{e}$ converge to zero as t→∞.

Proof. Choose the following Lyapunov function candidate:

$$\begin{aligned} V = &\frac{1}{2} r^T \bar{M}(q) r + \frac{1}{2}\sum_{k=1}^{p-r'} \tilde{\theta}_k^T \Gamma_k^{-1} \tilde{\theta}_k + \frac{1}{2}\sum_{k=1}^{p-r'} \tilde{\alpha}_k^T Q_k^{-1} \tilde{\alpha}_k + \frac{1}{2}\sum_{k=1}^{p-r'} \tilde{\beta}_k^T N_k^{-1} \tilde{\beta}_k \\ &+ \frac{1}{2}\sum_{k=1}^{p-r'} \tilde{\psi}_k T_k^{-1} \tilde{\psi}_k + \frac{1}{2}\sum_{k=1}^{p-r'} \tilde{c}_{\theta k}^T E_{\theta k}^{-1} \tilde{c}_{\theta k} + \frac{1}{2}\sum_{k=1}^{p-r'} \tilde{c}_{\alpha k}^T E_{\alpha k}^{-1} \tilde{c}_{\alpha k} + \frac{1}{2}\sum_{k=1}^{p-r'} \tilde{c}_{\beta k}^T E_{\beta k}^{-1} \tilde{c}_{\beta k} \\ &+ \frac{1}{2}\sum_{k=1}^{p-r'} \tilde{b}_{\theta k}^T I_{\theta k}^{-1} \tilde{b}_{\theta k} + \frac{1}{2}\sum_{k=1}^{p-r'} \tilde{b}_{\alpha k}^T I_{\alpha k}^{-1} \tilde{b}_{\alpha k} + \frac{1}{2}\sum_{k=1}^{p-r'} \tilde{b}_{\beta k}^T I_{\beta k}^{-1} \tilde{b}_{\beta k} \end{aligned} \tag{4.52}$$

Differentiating V obtains:

$$\begin{aligned} \dot{V} = &r^T \bar{M} \dot{r} + \frac{1}{2} r^T \dot{\bar{M}} r + \sum_{k=1}^{p-r'} \tilde{\theta}_k^T \Gamma_k^{-1} \dot{\tilde{\theta}}_k + \sum_{k=1}^{p-r'} \tilde{\alpha}_k^T Q_k^{-1} \dot{\tilde{\alpha}}_k + \sum_{k=1}^{p-r'} \tilde{\beta}_k^T N_k^{-1} \dot{\tilde{\beta}}_k \\ &+ \sum_{k=1}^{p-r'} \tilde{\psi}_k T_k^{-1} \dot{\tilde{\psi}}_k + \sum_{k=1}^{p-r'} \tilde{c}_{\theta k}^T E_{\theta k}^{-1} \dot{\tilde{c}}_{\theta k} + \sum_{k=1}^{p-r'} \tilde{c}_{\alpha k}^T E_{\alpha k}^{-1} \dot{\tilde{c}}_{\alpha k} + \sum_{k=1}^{p-r'} \tilde{c}_{\beta k}^T E_{\beta k}^{-1} \dot{\tilde{c}}_{\beta k} \\ &+ \sum_{k=1}^{p-r'} \tilde{b}_{\theta k}^T I_{\theta k}^{-1} \dot{\tilde{b}}_{\theta k} + \sum_{k=1}^{p-r'} \tilde{b}_{\alpha k}^T I_{\alpha k}^{-1} \dot{\tilde{b}}_{\alpha k} + \sum_{k=1}^{p-r'} \tilde{b}_{\beta k}^T I_{\beta k}^{-1} \dot{\tilde{b}}_{\beta k} \end{aligned} \tag{4.53}$$

Through property 3.2, $\dot{\bar{M}}(q) - 2\bar{C}(q,\dot{q})$ is screw-symmetric then $r^T\left(\dot{\bar{M}} - 2\bar{C}\right) r = 0$, the following equation can be derived:

$$\begin{aligned} \dot{V} = &r^T \bar{M} \dot{r} + r^T \bar{C} r + \sum_{k=1}^{p-r'} \tilde{\theta}_k^T \Gamma_k^{-1} \dot{\tilde{\theta}}_k + \sum_{k=1}^{p-r'} \tilde{\alpha}_k^T Q_k^{-1} \dot{\tilde{\alpha}}_k + \sum_{k=1}^{p-r'} \tilde{\beta}_k^T N_k^{-1} \dot{\tilde{\beta}}_k \\ &+ \sum_{k=1}^{p-r'} \tilde{\psi}_k T_k^{-1} \dot{\tilde{\psi}}_k + \sum_{k=1}^{p-r'} \tilde{c}_{\theta k}^T E_{\theta k}^{-1} \dot{\tilde{c}}_{\theta k} + \sum_{k=1}^{p-r'} \tilde{c}_{\alpha k}^T E_{\alpha k}^{-1} \dot{\tilde{c}}_{\alpha k} + \sum_{k=1}^{p-r'} \tilde{c}_{\beta k}^T E_{\beta k}^{-1} \dot{\tilde{c}}_{\beta k} \\ &+ \sum_{k=1}^{p-r'} \tilde{b}_{\theta k}^T I_{\theta k}^{-1} \dot{\tilde{b}}_{\theta k} + \sum_{k=1}^{p-r'} \tilde{b}_{\alpha k}^T I_{\alpha k}^{-1} \dot{\tilde{b}}_{\alpha k} + \sum_{k=1}^{p-r'} \tilde{b}_{\beta k}^T I_{\beta k}^{-1} \dot{\tilde{b}}_{\beta k} \\ = &r^T\left(\bar{M}\dot{r} + \bar{C} r\right) + \sum_{k=1}^{p-r'} \tilde{\theta}_k^T \Gamma_k^{-1} \dot{\tilde{\theta}}_k + \sum_{k=1}^{p-r'} \tilde{\alpha}_k^T Q_k^{-1} \dot{\tilde{\alpha}}_k + \sum_{k=1}^{p-r'} \tilde{\beta}_k^T N_k^{-1} \dot{\tilde{\beta}}_k \\ &+ \sum_{k=1}^{p-r'} \tilde{\psi}_k T_k^{-1} \dot{\tilde{\psi}}_k + \sum_{k=1}^{p-r'} \tilde{c}_{\theta k}^T E_{\theta k}^{-1} \dot{\tilde{c}}_{\theta k} + \sum_{k=1}^{p-r'} \tilde{c}_{\alpha k}^T E_{\alpha k}^{-1} \dot{\tilde{c}}_{\alpha k} + \sum_{k=1}^{p-r'} \tilde{c}_{\beta k}^T E_{\beta k}^{-1} \dot{\tilde{c}}_{\beta k} \\ &+ \sum_{k=1}^{p-r'} \tilde{b}_{\theta k}^T I_{\theta k}^{-1} \dot{\tilde{b}}_{\theta k} + \sum_{k=1}^{p-r'} \tilde{b}_{\alpha k}^T I_{\alpha k}^{-1} \dot{\tilde{b}}_{\alpha k} + \sum_{k=1}^{p-r'} \tilde{b}_{\beta k}^T I_{\beta k}^{-1} \dot{\tilde{b}}_{\beta k} \end{aligned} \tag{4.54}$$

And (4.48) gives:

$$
\begin{aligned}
&\bar{M}(q)\dot{r}+\bar{C}(q,\dot{q})r \\
&=\left[\left\{\tilde{\Theta}\right\}^{T}\bullet\left\{\Xi(q)\right\}\right]\dot{\eta}_{r}+\left[\left\{\tilde{A}\right\}^{T}\bullet\left\{Z(z)\right\}\right]\eta_{r}+\left[\left\{\tilde{B}\right\}^{T}\bullet\left\{H(q)\right\}\right] \\
&+\left[\left\{\hat{\Theta}\right\}^{T}\bullet\left\{\tilde{\Xi}(q)\right\}\right]\dot{\eta}_{r}+\left[\left\{\hat{A}\right\}^{T}\bullet\left\{\tilde{Z}(z)\right\}\right]\eta_{r}+\left[\left\{\hat{B}\right\}^{T}\bullet\left\{\tilde{H}(q)\right\}\right] \\
&+E_{1}+\tau_{d}-Kr-U_{s}
\end{aligned}
\tag{4.55}
$$

Then (4.54) becomes:

$$
\begin{aligned}
\dot{V}=&-r^{T}Kr-\sum_{k=1}^{p-r'}\hat{\psi}_{k}r_{k}\operatorname{sgn}(r_{k})+r^{T}(E_{1}+\tau_{d})+\sum_{k=1}^{p-r'}\left\{\tilde{\theta}_{k}\right\}^{T}\bullet\left\{\xi_{k}(q)\right\}\dot{\eta}_{r}r_{k} \\
&+\sum_{k=1}^{p-r'}\left\{\tilde{\alpha}_{k}\right\}^{T}\bullet\left\{\varsigma_{k}(z)\right\}\eta_{r}r_{k}+\sum_{k=1}^{p-r'}\left\{\tilde{\beta}_{k}\right\}^{T}\bullet\left\{\sigma_{k}(q)\right\}r_{k}+\sum_{k=1}^{p-r'}\left\{\hat{\theta}_{k}\right\}^{T}\bullet\left\{\tilde{\xi}_{k}(q)\right\}\dot{\eta}_{r}r_{k} \\
&+\sum_{k=1}^{p-r'}\left\{\hat{\alpha}_{k}\right\}^{T}\bullet\left\{\tilde{\xi}_{k}(z)\right\}\eta_{r}r_{k}+\sum_{k=1}^{p-r'}\left\{\hat{\beta}_{k}\right\}^{T}\bullet\left\{\tilde{\sigma}_{k}(q)\right\}r_{k}+\sum_{k=1}^{p-r'}\tilde{\theta}_{k}^{T}\Gamma_{k}^{-1}\dot{\tilde{\theta}}_{k}+\sum_{k=1}^{p-r'}\tilde{\alpha}_{k}^{T}Q_{k}^{-1}\dot{\tilde{\alpha}}_{k} \\
&+\sum_{k=1}^{p-r'}\tilde{\beta}_{k}^{T}N_{k}^{-1}\dot{\tilde{\beta}}_{k}+\sum_{k=1}^{p-r'}\tilde{\psi}_{k}T_{k}^{-1}\dot{\tilde{\psi}}_{k}+\sum_{k=1}^{p-r'}\tilde{c}_{\theta k}^{T}E_{\theta k}^{-1}\dot{\tilde{c}}_{\theta k}+\sum_{k=1}^{p-r'}\tilde{c}_{\alpha k}^{T}E_{\alpha k}^{-1}\dot{\tilde{c}}_{\alpha k} \\
&+\sum_{k=1}^{p-r'}\tilde{c}_{\beta k}^{T}E_{\beta k}^{-1}\dot{\tilde{c}}_{\beta k}+\sum_{k=1}^{p-r'}\tilde{b}_{\theta k}^{T}I_{\theta k}^{-1}\dot{\tilde{b}}_{\theta k}+\sum_{k=1}^{p-r'}\tilde{b}_{\alpha k}^{T}I_{\alpha k}^{-1}\dot{\tilde{b}}_{\alpha k}+\sum_{k=1}^{p-r'}\tilde{b}_{\beta k}^{T}I_{\beta k}^{-1}\dot{\tilde{b}}_{\beta k}
\end{aligned}
\tag{4.56}
$$

Obviously,

$$
(\dot{\bullet})=(\bullet)-(\dot{\bullet})
\tag{4.57}
$$

where $(\bullet)=0$.

We obtain:

$$
\begin{aligned}
\dot{V}=&-r^{T}Kr-\sum_{k=1}^{p-r'}\hat{\psi}_{k}r_{k}\operatorname{sgn}(r_{k})+r^{T}(E_{1}+\tau_{d})+\sum_{k=1}^{p-r'}\left\{\tilde{\theta}_{k}\right\}^{T}\bullet\left\{\xi_{k}(q)\right\}\dot{\eta}_{r}r_{k} \\
&+\sum_{k=1}^{p-r'}\left\{\tilde{\alpha}_{k}\right\}^{T}\bullet\left\{\varsigma_{k}(z)\right\}\eta_{r}r_{k}+\sum_{k=1}^{p-r'}\left\{\tilde{\beta}_{k}\right\}^{T}\bullet\left\{\sigma_{k}(q)\right\}r_{k}-\sum_{k=1}^{p-r'}\tilde{\theta}_{k}^{T}\Gamma_{k}^{-1}\dot{\hat{\theta}}_{k} \\
&-\sum_{k=1}^{p-r'}\tilde{\alpha}_{k}^{T}Q_{k}^{-1}\dot{\hat{\alpha}}_{k}-\sum_{k=1}^{p-r'}\tilde{\beta}_{k}^{T}N_{k}^{-1}\dot{\hat{\beta}}_{k}-\sum_{k=1}^{p-r'}\tilde{\psi}_{k}T_{k}^{-1}\dot{\hat{\psi}}_{k}+\sum_{k=1}^{p-r'}\left\{\hat{\theta}_{k}\right\}^{T}\bullet\left\{\tilde{\xi}_{k}(q)\right\}\dot{\eta}_{r}r_{k} \\
&+\sum_{k=1}^{p-r'}\left\{\hat{\alpha}_{k}\right\}^{T}\bullet\left\{\tilde{\xi}_{k}(z)\right\}\eta_{r}r_{k}+\sum_{k=1}^{p-r'}\left\{\hat{\beta}_{k}\right\}^{T}\bullet\left\{\tilde{\sigma}_{k}(q)\right\}r_{k}-\sum_{k=1}^{p-r'}\tilde{c}_{\theta k}^{T}E_{\theta k}^{-1}\dot{\hat{c}}_{\theta k} \\
&-\sum_{k=1}^{p-r'}\tilde{c}_{\alpha k}^{T}E_{\alpha k}^{-1}\dot{\hat{c}}_{\alpha k}-\sum_{k=1}^{p-r'}\tilde{c}_{\beta k}^{T}E_{\beta k}^{-1}\dot{\hat{c}}_{\beta k}-\sum_{k=1}^{p-r'}\tilde{b}_{\theta k}^{T}I_{\theta k}^{-1}\dot{\hat{b}}_{\theta k}-\sum_{k=1}^{p-r'}\tilde{b}_{\alpha k}^{T}I_{\alpha k}^{-1}\dot{\hat{b}}_{\alpha k} \\
&-\sum_{k=1}^{p-r'}\tilde{b}_{\beta k}^{T}I_{\beta k}^{-1}\dot{\hat{b}}_{\beta k}
\end{aligned}
\tag{4.58}
$$

If the adaptation laws of on-line parameters $\dot{\hat{\theta}}_k$, $\dot{\hat{\alpha}}_k$, $\dot{\hat{\beta}}_k$ for the proposed NN controller are selected as in (4.51), then $\dot{V}$ becomes:

$$\begin{aligned}\dot{V} = &-r^T Kr - \sum_{k=1}^{p-r'} \hat{\psi}_k r_k \operatorname{sgn}(r_k) + r^T (E_1 + \tau_d) - \sum_{k=1}^{p-r'} \tilde{\psi}_k T_k^{-1} \dot{\hat{\psi}}_k \\ &+ \sum_{k=1}^{p-r'} \left\{\hat{\theta}_k\right\}^T \bullet \left\{\tilde{\xi}_k(q)\right\} \dot{\eta}_r r_k + \sum_{k=1}^{p-r'} \left\{\hat{\alpha}_k\right\}^T \bullet \left\{\tilde{\varsigma}_k(z)\right\} \eta_r r_k \\ &+ \sum_{k=1}^{p-r'} \left\{\hat{\beta}_k\right\}^T \bullet \left\{\tilde{\sigma}_k(q)\right\} r_k - \sum_{k=1}^{p-r'} \tilde{c}_{\theta k}^T E_{\theta k}^{-1} \dot{\hat{c}}_{\theta k} - \sum_{k=1}^{p-r'} \tilde{c}_{\alpha k}^T E_{\alpha k}^{-1} \dot{\hat{c}}_{\alpha k} \\ &- \sum_{k=1}^{p-r'} \tilde{c}_{\beta k}^T E_{\beta k}^{-1} \dot{\hat{c}}_{\beta k} - \sum_{k=1}^{p-r'} \tilde{b}_{\theta k}^T I_{\theta k}^{-1} \dot{\hat{b}}_{\theta k} - \sum_{k=1}^{p-r'} \tilde{b}_{\alpha k}^T I_{\alpha k}^{-1} \dot{\hat{b}}_{\alpha k} - \sum_{k=1}^{p-r'} \tilde{b}_{\beta k}^T I_{\beta k}^{-1} \dot{\hat{b}}_{\beta k}\end{aligned} \tag{4.59}$$

Using (4.50), $\tilde{\xi}_k$, $\tilde{\varsigma}_k$, $\tilde{\sigma}_k$ can be approximated as:

$$\begin{aligned}\tilde{\xi}_k &= \xi_{kc} \tilde{c}_{\theta_k} + \xi_{kc} \tilde{b}_{\theta_k} + O_{\theta_k} \\ \tilde{\varsigma}_k &= \varsigma_{kc} \tilde{c}_{\alpha_k} + \varsigma_{kc} \tilde{b}_{\alpha_k} + O_{\alpha_k} \\ \tilde{\sigma}_k &= \sigma_{kc} \tilde{c}_{\beta_k} + \sigma_{kb} \tilde{b}_{\beta_k} + O_{\beta_k}\end{aligned} \tag{4.60}$$

Substituting (4.60) into (4.59) gives:

$$\begin{aligned}\dot{V} = &-r^T Kr - \sum_{k=1}^{p-r'} \hat{\psi}_k r_k \operatorname{sgn}(r_k) + r^T (E_1 + \tau_d) - \sum_{k=1}^{p-r'} \tilde{\psi}_k T_k^{-1} \dot{\hat{\psi}}_k \\ &+ \sum_{k=1}^{p-r'} \left\{\hat{\theta}_k\right\}^T \bullet \left\{\xi_{kc} \tilde{c}_{\theta_k} + \xi_{kc} \tilde{b}_{\theta_k} + O_{\theta_k}\right\} \dot{\eta}_r r_k \\ &+ \sum_{k=1}^{p-r'} \left\{\hat{\alpha}_k\right\}^T \bullet \left\{\varsigma_{kc} \tilde{c}_{\alpha_k} + \varsigma_{kc} \tilde{b}_{\alpha_k} + O_{\alpha_k}\right\} \eta_r r_k \\ &+ \sum_{k=1}^{p-r'} \left\{\hat{\beta}_k\right\}^T \bullet (\sigma_{kc} \tilde{c}_{\beta_k} + \sigma_{kb} \tilde{b}_{\beta_k} + O_{\beta_k}) r_k \\ &- \sum_{k=1}^{p-r'} \tilde{c}_{\theta k}^T E_{\theta k}^{-1} \dot{\hat{c}}_{\theta k} - \sum_{k=1}^{p-r'} \tilde{c}_{\alpha k}^T E_{\alpha k}^{-1} \dot{\hat{c}}_{\alpha k} - \sum_{k=1}^{p-r'} \tilde{c}_{\beta k}^T E_{\beta k}^{-1} \dot{\hat{c}}_{\beta k} \\ &- \sum_{k=1}^{p-r'} \tilde{b}_{\theta k}^T I_{\theta k}^{-1} \dot{\hat{b}}_{\theta k} - \sum_{k=1}^{p-r'} \tilde{b}_{\alpha k}^T I_{\alpha k}^{-1} \dot{\hat{b}}_{\alpha k} - \sum_{k=1}^{p-r'} \tilde{b}_{\beta k}^T I_{\beta k}^{-1} \dot{\hat{b}}_{\beta k}\end{aligned} \tag{4.61}$$

Using the on-line parameter adaptation laws of $\dot{\hat{c}}_{\theta k}$, $\dot{\hat{c}}_{\alpha k}$, $\dot{\hat{c}}_{\beta k}$, $\dot{\hat{b}}_{\theta k}$, $\dot{\hat{b}}_{\alpha k}$, $\dot{\hat{b}}_{\beta k}$ in (4.51) then $\dot{V}$ is given by:

$$
\begin{aligned}
\dot{V} &= -r^T Kr - \sum_{k=1}^{p-r'} \hat{\psi}_k r_k \operatorname{sgn}(r_k) + r^T (E_1 + \tau_d) - \sum_{k=1}^{p-r'} \tilde{\psi}_k T_k^{-1} \dot{\hat{\psi}}_k \\
&\quad + \sum_{k=1}^{p-r'} \left\{\hat{\theta}_k\right\}^T \bullet \left\{O_{\theta_k}\right\} \dot{\eta}_r r_k + \sum_{k=1}^{p-r'} \left\{\hat{\alpha}_k\right\}^T \bullet \left\{O_{\alpha_k}\right\} \eta_r r_k \\
&\quad + \sum_{k=1}^{p-r'} \left\{\hat{\beta}_k\right\}^T \bullet \left\{O_{\beta_k}\right\} r_k \\
&= -r^T Kr - \sum_{k=1}^{p-r'} \hat{\psi}_k r_k \operatorname{sgn}(r_k) + r^T (E_1 + \tau_d) - \sum_{k=1}^{p-r'} \tilde{\psi}_k T_k^{-1} \dot{\hat{\psi}}_k + r^T E_2 \\
&= -r^T Kr - \sum_{k=1}^{p-r'} \hat{\psi}_k r_k \operatorname{sgn}(r_k) + r^T E - \sum_{k=1}^{p-r'} \tilde{\psi}_k T_k^{-1} \dot{\hat{\psi}}_k
\end{aligned}
\tag{4.62}
$$

where $E_2 = \sum_{k=1}^{p-r'} \left\{\hat{\theta}_k\right\}^T \bullet \left\{O_{\theta_k}\right\} \dot{\eta}_r r_k + \sum_{k=1}^{p-r'} \left\{\hat{\alpha}_k\right\}^T \bullet \left\{O_{\alpha_k}\right\} \eta_r r_k + \sum_{k=1}^{p-r'} \left\{\hat{\beta}_k\right\}^T \bullet \left\{O_{\beta_k}\right\} r_k$.

According to the adaptation law $\dot{\hat{\psi}}_k = T_k r_k \operatorname{sgn}(r_k)$, then (4.62) becomes:

$$
\begin{aligned}
\dot{V} &= -r^T Kr + r^T E - \psi r^T \operatorname{sgn}(r) \\
&= -r^T Kr - r^T (\psi \operatorname{sgn}(r) - E)
\end{aligned}
\tag{4.63}
$$

Assume $\psi \geq \|E\|$, (4.63) yields:

$$
\dot{V} \leq -r^T Kr \leq 0 \tag{4.64}
$$

From (4.64), V is bounded, which implies that $r \in L_\infty^{p-r'}$. We have $\int_0^t r^T Kr ds \leq V(0) - V(t)$, which leads to $r \in L_2^{p-r'}$. According to Lemma 4.1, from $r(t) = \dot{e} + \Lambda \int \dot{e}$, it can be obtained that $e \in L_\infty^{p-r'} \bigcap L_2^{p-r'}$ and $\dot{e} \in L_2^{p-r'}$. From the assumption, we conclude that $\int \eta(t),\ \eta(t),\ \int \eta_r(t),\ \eta_r(t) \in L_\infty^{p-r'}$ and they are continuous. Therefore, we can conclude that $\dot{r}$ and $\dot{\eta}$ are bounded. Thus, $r \to 0$ as $t \to \infty$ can be obtained. Consequently, we have $e \to 0$ and $\dot{e} \to 0$ as $t \to \infty$. It follows that e_q and $\dot{e}_q \to 0$ as $t \to \infty$, where e_q is the error vector of the generalized coordinates and $\dot{e}_q$ is its derivative, which completes the proof.

4.4 RBF Neural Network Adaptive Motion/Force Control

4.4.1 Introduction

For many tasks the mobile manipulator comes in contact with a constrained surface, and an interaction force develops between the end-effector and the environment. Therefore, the contact force control is at least as important as the position control. In such cases, the trajectory and the constraint force of the system are required to asymptotically converge to the desired ones, respectively. A unified form of holonomic and nonholonomic constrains is first introduced into the system's dynamic model reduction. A RBF network is constructed to identify the unknown dynamics of the robot. Based on the reduced model, a robust and adaptive control scheme which is similar to the control method presented in the previous subchapter is designed for the coordinated motion and force control problem of the nonholonomic mobile manipulator with constrained end-effector.

4.4.2 Dynamics of Robot with Holonomic and Nonholonomic Constraints

Assume that the end-effector position p_e of the mobile manipulator is constrained and the end-effector moves along a desired curve $h(p_e)=0$ on a frictionless and rigid surface as shown in Figure 4.9. The generalized position vector p_e of the end-effector can be expressed in terms of the generalized coordinates q of the mobile manipulator.

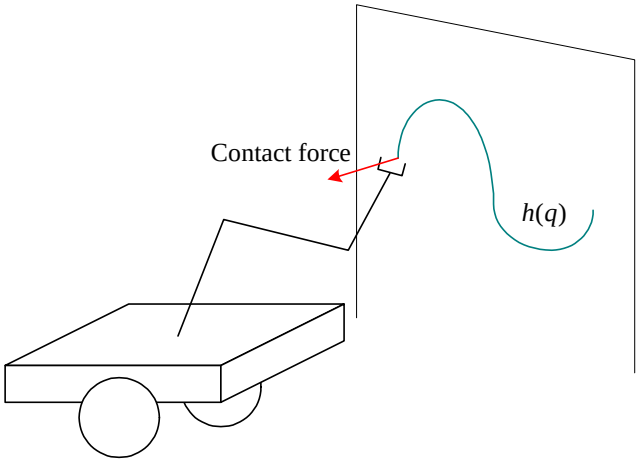


Figure 4.9: Robot with holonomic constraint

In this case, the dynamics of the mobile manipulator subject to holonomic and nonholonomic kinematic constraints can be obtained using the Lagrangian approach in the form:

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+F(q)+\tau_d = E(q)\tau+B^T(q)\lambda \tag{4.65}$$

where $q \in \Re^p$ denotes the p generalized coordinates; $M(q) \in \Re^{p \times p}$ is a symmetric and positive definite inertia matrix; $C(q,\dot{q}) \in \Re^{p \times p}$ represents a centripetal and Coriolis matrix, $F(q,\dot{q}) \in \Re^p$ is a friction and gravity vector; $B(q) = \begin{bmatrix} A_v & 0 \\ J_v & J_b \end{bmatrix}$, and $\lambda = \begin{bmatrix} \lambda_n^T & \lambda_h^T \end{bmatrix}^T$ is the vector of Lagrange multipliers corresponding to the nonholonomic and holonomic constraints; $B^T(q)\lambda$ denotes the constraint forces vector; $\tau_d \in \Re^p$ denotes the unknown bounded disturbances; $E(q) \in \Re^{p \times (p-r')}$ is the input transformation matrix; $\tau \in \Re^{p-r'}$ is a torque input vector.

Recall that the generalized coordinates of Loop *a* and Loop *b* may be separated into two sets $q = \begin{bmatrix} q_v^T & q_b^T \end{bmatrix}^T \in \Re^{m+n+2a}$, where $q_b = \begin{bmatrix} q_w^T & q_r^T & q_l^T \end{bmatrix}^T \in \Re^{n+2a}$. In the following, $n+2a$ is denoted by b. The mobile platform is subjected to the nonholonomic constraints, the r' non-integrable and independent kinematic constraints can be expressed as:

$$A_v(q_v)\dot{q}_v = 0 \tag{4.66}$$

where $A_v(q_v)$ is the kinematic constraint matrix which is assumed to have full rank r'.

The effect of the nonholonomic constraints on the mobile platform can be regarded as restricting the dynamics on the manifold $\Omega_n = \{(q_v, \dot{q}_v) \mid A_v(q_v)\dot{q}_v = 0\}$. The generalized constraint forces for the nonholonomic constraints can be given by $f_n = A_v^T(q_v)\lambda_n$, where $\lambda_n \in \Re^{r'}$ is known as the vector of Lagrangian multipliers.

There exists a full rank matrix $S_v(q_v) \in \Re^{m \times (m-r')}$ formed by $m-r'$ columns that spans the null space of $A_v(q_v)$ defined in (4.66):

$$S_v^T(q_v)A_v^T(q_v) = 0 \tag{4.67}$$

Assume that the mobile manipulator has r'' independent holonomic constraints (i.e. geometric constraints), which can be written as:

$$h(q) = 0,\ h(q) \in \Re^{r''} \tag{4.68}$$

where $h(q)$ is full rank.

The holonomic constraints could be further written as:

$$J(q)\dot{q} = 0 \tag{4.69}$$

where $J(q) = [J_v \quad J_b]$ is divided into two parts: $J_v = \frac{\partial h}{\partial q_v}$ for the mobile base and $J_b = \frac{\partial h}{\partial q_b}$ for the manipulator.

The holonomic constraint forces f_h can be expressed as $f_h = J^T(q)\lambda_h$ in joint space. Hence, the holonomic constraint manifold Ω_h on the robot's end-effector can be defined by $\Omega_h = \{(q, \dot{q}) \mid h(q) = 0, J(q)\dot{q} = 0\}$.

We have known that the generalized position vector p_e of the end-effector can be expressed in terms of the generalized coordinates q of the robot.

Thus

$$\frac{\partial h}{\partial q} = \frac{\partial h}{\partial p_e} \cdot \frac{\partial p_e}{\partial q} \tag{4.70}$$

where $\frac{\partial h}{\partial p_e}$ represents the gradient of the constraint function and $\frac{\partial p_e}{\partial q}$ is the Jacobian matrix of the mobile manipulator.

As discussed above, $J(q) = \frac{\partial h}{\partial q}$, the holonomic constraint forces can be expressed as:

$$f_h = (\frac{\partial p_e}{\partial q})^T \cdot (\frac{\partial h}{\partial p_e})^T \lambda_h \tag{4.71}$$

where $(\frac{\partial h}{\partial p_e})^T \lambda_h$ can be regarded as the normal force applied to the end-effector and $\frac{\partial p_e}{\partial q}$ is the Jacobian matrix. Thus this expression is corresponding to the term $J_h^T f_c$ in equation (3.83).

Let us define an auxiliary vector $v \in \Re^{(m-r')}$ and $\eta \in \Re^{(m-r')}$ is its integral. Considering the nonholonomic constraints and their derivatives, the dynamics of the mobile manipulator is as follows:

$$\begin{aligned}
&\begin{bmatrix} S_v^T M_v S_v & S_v^T M_{vb} \\ M_{bv} S_v & M_b \end{bmatrix} \begin{bmatrix} \dot{v} \\ \ddot{q}_b \end{bmatrix} + \begin{bmatrix} S_v^T M_v \dot{S}_v + S_v^T C_v S_v & S_v^T C_{vb} \\ M_{bv} \dot{S}_v + C_{bv} S_v & C_b \end{bmatrix} \begin{bmatrix} v \\ \dot{q}_b \end{bmatrix} \\
&+ \begin{bmatrix} S_v^T F_v \\ F_b \end{bmatrix} + \begin{bmatrix} S_v^T \tau_{dv} \\ \tau_{db} \end{bmatrix} \\
&= \begin{bmatrix} S_v^T E_v \tau_v \\ \tau_b \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ J_v S_v & J_b \end{bmatrix}^T \begin{bmatrix} 0 \\ \lambda_h \end{bmatrix}
\end{aligned} \tag{4.72}$$

From this equation, it can be seen that:

$$J'(q) = \begin{bmatrix} \dfrac{\partial h}{\partial \eta} & \dfrac{\partial h}{\partial q_b} \end{bmatrix} = \begin{bmatrix} J_v S_v & J_b \end{bmatrix} \tag{4.73}$$

It is recognized that there exists a proper partition [152] so that the vector q_b can be further rearranged and partitioned into $q_b = \begin{bmatrix} q_b^{1T} & q_b^{2T} \end{bmatrix}^T$.

where $q_b^1 \in \Re^{b-r''}$ describe the constrained motions of the manipulator, and $q_b^2 \in \Re^{r''}$ denote the remaining joint variables.

Thus

$$J'(q) = \begin{bmatrix} \dfrac{\partial h}{\partial \eta} & \dfrac{\partial h}{\partial q_b^1} & \dfrac{\partial h}{\partial q_b^2} \end{bmatrix} \tag{4.74}$$

From [152], it could be concluded that q is the function of $\varsigma = \begin{bmatrix} \eta^T & q_b^{1T} \end{bmatrix}^T$, that is, $q = q(\varsigma)$. There always exists a matrix $L(\varsigma) \in \Re^{(p-r')\times(p-r'-r'')}$ such that $\dot{q} = L(\varsigma)\dot{\varsigma}$, where $L(\varsigma) = \partial q / \partial \varsigma$, $\ddot{q} = L(\varsigma)\ddot{\varsigma} + \dot{L}(\varsigma)\dot{\varsigma}$ and $L(\varsigma)$, $J^{-1}(\varsigma) = J'(q(\varsigma))$ satisfies the relationship:

$$L^T(\varsigma) J^{-1T}(\varsigma) = 0 \tag{4.75}$$

The dynamic model (4.65), when it is restricted to the constraint surface, can be transformed into the reduced model:

$$\bar{M}(\varsigma)L(\varsigma)\ddot{\varsigma}+\bar{C}(\varsigma,\dot{\varsigma})\dot{\varsigma}+\bar{F}+\bar{\tau}_d=u+J^{'1T}(\varsigma)\lambda_h \tag{4.76}$$

where

$$\bar{M}=\begin{bmatrix} S_v{}^T M_v S_v & S_v{}^T M_{vb} \\ M_{bv}S_v & M_b \end{bmatrix}, \varsigma=\begin{bmatrix} \eta \\ q_b^1 \end{bmatrix},$$

$$\bar{C}=\begin{bmatrix} S_v{}^T M_v S_v & S_v{}^T M_{vb} \\ M_{bv}S_v & M_b \end{bmatrix}\dot{L}(\varsigma)+\begin{bmatrix} S_v{}^T M_v \dot{S}_v+S_v{}^T C_v S_v & S_v{}^T C_{vb} \\ M_{bv}\dot{S}_v+C_{bv}S_v & C_b \end{bmatrix}L(\varsigma),$$

$$\bar{F}=\begin{bmatrix} S_v{}^T F_v \\ F_b \end{bmatrix}, \bar{\tau}_d=\begin{bmatrix} S_v{}^T \tau_{dv} \\ \tau_{db} \end{bmatrix}, u=\bar{E}\tau, \bar{E}=\begin{bmatrix} S_v{}^T E_v & 0 \\ 0 & I_{b\times b} \end{bmatrix}.$$

Multiplying L^T by both sides of (4.76), we can obtain:

$$M_L\ddot{\varsigma}+C_L\dot{\varsigma}+F_L+\tau_{dL}=L^T u \tag{4.77}$$

where $M_L=L^T\bar{M}L,\ C_L=L^T\bar{C},\ F_L=L^T\bar{F},\ \tau_{dL}=L^T\bar{\tau}_d$.

Property 4.1: The matrix M_L is symmetric and positive definite; the matrix $\dot{M}_L-2C_L$ is skew-symmetric.

Property 4.2: For holonomic systems, matrices $\bar{M}$, $\bar{F}$, $J^{'1}(\varsigma)$, $L(\varsigma)$ are uniformly bounded and uniformly continuous if ς is uniformly bounded and continuous, respectively. $\bar{C}$ is uniformly bounded and uniformly continuous if ς and $\dot{\varsigma}$ are uniformly bounded and continuous.

Property 4.3: There exist some finite positive constants $c_i>0(1\le i\le 5)$ such that $\|M_L\|\le c_1$, $\|C_L\|\le c_2+c_3\|\dot{\varsigma}\|$, $\|F_L\|\le c_4$, and $\|\tau_{dL}\|\le c_5$.

The force multipliers λ_h can be obtained by (4.76):

$$\lambda_h=Z(\varsigma)(\bar{C}(\varsigma,\dot{\varsigma})\dot{\varsigma}+\bar{F}+\bar{\tau}_d-u) \tag{4.78}$$

where $Z=(J^{'1}(\bar{M})^{-1}J^{'1T})^{-1}J^{'1}(\bar{M})^{-1}$.

Control input u in (4.77) is selected as:

$$u = u_a - J'^{1T} u_b \tag{4.79}$$

Then (4.77) and (4.78) can be written as:

$$M_L \ddot{\varsigma} + C_L \dot{\varsigma} + F_L + \tau_{dL} = L^T u_a \tag{4.80}$$

and

$$\lambda_h = Z(\varsigma)(\bar{C}(\varsigma, \dot{\varsigma})\dot{\varsigma} + \bar{F} + \bar{\tau}_d - u_a) + u_b \tag{4.81}$$

4.4.3 NN Modeling with Nonholonomic and Holonomic constraints

From the dynamics equation (4.80), both $M_L\left(\eta, q_b^1\right)$ and $F_L\left(\eta, q_b^1\right)$ are only the functions of η and q_b^1, hence, static NNs are sufficient to model the matrix elements:

$$m_{kj}\left(\eta, q_b^1\right) = \sum_l \theta_{kjl}\xi_{kjl}\left(\eta, q_b^1\right) + \varepsilon_{mkj}\left(\eta, q_b^1\right) = \theta_{kj}^T \xi_{kj}\left(\eta, q_b^1\right) + \varepsilon_{mkj}\left(\eta, q_b^1\right) \tag{4.82}$$

$$f_k\left(\eta, q_b^1\right) = \sum_l \beta_{kl}\sigma_{kl}\left(\eta, q_b^1\right) + \varepsilon_{fk}\left(\eta, q_b^1\right) = \beta_k^T \sigma_k\left(\eta, q_b^1\right) + \varepsilon_{fk}\left(\eta, q_b^1\right) \tag{4.83}$$

where $\theta_{kjl}, \beta_{kl} \in \Re$ are the weights of the NNs, $\xi_{kjl}\left(\eta, q_b^1\right)$, $\sigma_{kl}\left(\eta, q_b^1\right) \in \Re$ are the corresponding Gaussian basis functions with input vector $\left[\eta^T \quad q_b^{1T}\right]^T$. $\varepsilon_{mkj}\left(\eta, q_b^1\right)$, $\varepsilon_{fk}\left(\eta, q_b^1\right) \in \Re$ are respectively the modeling errors of $m_{kj}\left(\eta, q_b^1\right)$ and $f_k\left(\eta, q_b^1\right)$, and they are assumed to be bounded.

Whereas, $C_L\left(\eta, \dot{\eta}, q_b^1, \dot{q}_b^1\right)$ is modeled using dynamic NN with the inputs η, $\dot{\eta}$, q_b^1 and $\dot{q}_b^1$:

$$c_{kj}\left(\eta, \dot{\eta}, q_b^1, \dot{q}_b^1\right) = \sum_l \alpha_{kjl}\varsigma_{kjl}(z) + \varepsilon_{ckj}(z) = \alpha_{kj}^T \varsigma_{kj}(z) + \varepsilon_{ckj}(z) \tag{4.84}$$

where $z = \left[\eta^T \quad \dot{\eta}^T \quad q_b^{1T} \quad \dot{q}_b^{1T}\right]^T$; $\alpha_{kjl} \in \Re$ is the weight; $\varsigma_{kjl}(z) \in \Re$ is the corresponding Gaussian basis function with input vector z; $\varepsilon_{ckj}(z)$ is the modeling error of $c_{kj}(z)$, and it is also assumed to be bounded.

$$m_{kj}\left(\eta, q_b^1\right) = \theta_{kj}^T \xi_{kj}\left(\eta, q_b^1\right) + \varepsilon_{mkj}\left(\eta, q_b^1\right) \tag{4.85}$$

$$c_{kj}\left(\eta, \dot{\eta}, q_b^1, \dot{q}_b^1\right) = \alpha_{kj}^T \varsigma_{kj}\left(z\right) + \varepsilon_{ckj}\left(z\right) \tag{4.86}$$

$$f_k\left(\eta, q_b^1\right) = \beta_k^T \sigma_k\left(\eta, q_b^1\right) + \varepsilon_{fk}\left(\eta, q_b^1\right) \tag{4.87}$$

M_L, C_L, F_L can be rewritten using GL matrix as:

$$M_L = \left[\{\Theta\}^T \bullet \left\{\Xi\left(\eta, q_b^1\right)\right\}\right] + E_{M_L}\left(\eta, q_b^1\right) \tag{4.88}$$

$$C_L = \left[\{A\}^T \bullet \{Z(z)\}\right] + E_{C_L}(z) \tag{4.89}$$

$$F_L = \left[\{B\}^T \bullet \left\{H\left(\eta, q_b^1\right)\right\}\right] + E_{F_L}\left(\eta, q_b^1\right) \tag{4.90}$$

where Θ, $\Xi\left(\eta, q_b^1\right)$, A, $Z(z)$, B, $H\left(\eta, q_b^1\right)$ are respectively the GL matrices and vectors with the elements θ_{kj}, α_{kj}, β_k, ξ_{kj}, ς_{kj}, σ_k. $E_{M_L}\left(\eta, q_b^1\right)$, $E_{C_L}(z)$, $E_{F_L}\left(\eta, q_b^1\right)$ are the matrices and vector composed of the modeling errors ε_{mkj}, ε_{ckj}, ε_{fk}.

4.4.4 Controller Design

Assume a desired motion trajectory of the mobile manipulator $\varsigma_d(t) = \left[\eta_d^T \quad q_{bd}^{1T}\right]^T$ and a desired constraint force f_d, or, equivalently, a desired multiplier λ_{hd}. $\varsigma_d(t)$ is assumed to be bounded and uniformly continuous and has bounded and uniformly continuous derivatives up to the second order. The desired Lagrangian multiplier λ_{hd} is also bounded and uniformly continuous. The trajectory and force tracking control is to design a control law such that the states of the system asymptotically converge to the desired trajectory, and the force error remains bounded simultaneously.

Define

$$e(t) = \varsigma_d(t) - \varsigma(t) \tag{4.91}$$

$$r(t) = e + \Lambda \dot{e} \tag{4.92}$$

$$e_\lambda = \lambda_{hd} - \lambda_h \tag{4.93}$$

where Λ is a positive definite matrix.

Define

$$\ddot{\varsigma} = \ddot{\varsigma}_r - \dot{r} \tag{4.94}$$

$$\dot{\varsigma} = \dot{\varsigma}_r - r \tag{4.95}$$

The controller is proposed as follows:

$$L^T u_a = \hat{M}_L \ddot{\varsigma}_r + \hat{C}_L \dot{\varsigma}_r + \hat{F}_L + Kr + U_s \tag{4.96}$$

$$u_b = \lambda_{hd} + K_p e_\lambda + K_i \int e_\lambda dt \tag{4.97}$$

where $K \in \Re^{(p-r'-r'')\times(p-r'-r'')} > 0$; K_p and K_i are positive definite; $\hat{M}_L$, $\hat{C}_L$, $\hat{F}_L$ represent the estimates of the true parameter matrices. The first three terms of the control law are the model-based control, whereas the term Kr gives the PD control. The last term U_s is the robust controller for compensating for the approximation errors and uncertainties.

Applying the control law to the mobile manipulator dynamics yields the tracking error equation:

$$\begin{aligned} L^T u_a &= \left\{\left[\{\Theta\}^T \bullet \left\{\Xi\left(\eta, q_b^1\right)\right\}\right] + E_{M_L}\left(\eta, q_b^1\right)\right\}\ddot{\varsigma} + \left\{\left[\{A\}^T \bullet \{Z(z)\}\right] + E_{C_L}(z)\right\}\dot{\varsigma} \\ &\quad + \left[\{B\}^T \bullet \left\{H\left(\eta, q_b^1\right)\right\}\right] + E_{F_L}\left(\eta, q_b^1\right) + \tau_d \\ &= \left[\left\{\hat{\Theta}\right\}^T \bullet \left\{\hat{\Xi}\left(\eta, q_b^1\right)\right\}\right]\ddot{\varsigma}_r + \left[\left\{\hat{A}\right\}^T \bullet \left\{\hat{Z}(z)\right\}\right]\dot{\varsigma}_r + \left[\left\{\hat{B}\right\}^T \bullet \left\{\hat{H}\left(\eta, q_b^1\right)\right\}\right] \\ &\quad + Kr + U_s \end{aligned} \tag{4.98}$$

where $\hat{\Xi} = \Xi - \tilde{\Xi},\ \hat{Z} = Z - \tilde{Z},\ \hat{H} = H - \tilde{H}$.

Substituting the above three transformations into (4.98) yields:

$$
\begin{aligned}
&\left\{\left[\{\Theta\}^T \bullet \left\{\Xi\left(\eta, q_b^1\right)\right\}\right] + E_{M_L}\left(\eta, q_b^1\right)\right\}\ddot{\varsigma} + \left\{\left[\{A\}^T \bullet \{Z(z)\}\right] + E_{C_L}(z)\right\}\dot{\varsigma} \\
&+\left[\{B\}^T \bullet \left\{H\left(\eta, q_b^1\right)\right\}\right] + E_{F_L}\left(\eta, q_b^1\right) + \tau_d \\
&=\left[\left\{\hat{\Theta}\right\}^T \bullet \left\{\Xi\left(\eta, q_b^1\right)\right\}\right]\ddot{\varsigma}_r + \left[\left\{\hat{A}\right\}^T \bullet \{Z(z)\}\right]\dot{\varsigma}_r + \left[\left\{\hat{B}\right\}^T \bullet \left\{H\left(\eta, q_b^1\right)\right\}\right] \\
&-\left[\left\{\hat{\Theta}\right\}^T \bullet \left\{\tilde{\Xi}\left(\eta, q_b^1\right)\right\}\right]\ddot{\varsigma}_r - \left[\left\{\hat{A}\right\}^T \bullet \left\{\tilde{Z}(z)\right\}\right]\dot{\varsigma}_r - \left[\left\{\hat{B}\right\}^T \bullet \left\{\tilde{H}\left(\eta, q_b^1\right)\right\}\right] \\
&+Kr + U_s
\end{aligned}
\tag{4.99}
$$

Considering (4.94), (4.95) and (4.99), (4.98) becomes:

$$
\begin{aligned}
L^T u_a &= \left\{\left[\{\Theta\}^T \bullet \left\{\Xi\left(\eta, q_b^1\right)\right\}\right] + E_{M_L}\left(\eta, q_b^1\right)\right\}(\ddot{\varsigma}_r - \ddot{r}) \\
&+\left\{\left[\{A\}^T \bullet \{Z(z)\}\right] + E_{C_L}(z)\right\}(\dot{\varsigma}_r - \dot{r}) \\
&+\left[\{B\}^T \bullet \left\{H\left(\eta, q_b^1\right)\right\}\right] + E_{F_L}\left(\eta, q_b^1\right) + \tau_d \\
&=\left[\left\{\hat{\Theta}\right\}^T \bullet \left\{\Xi\left(\eta, q_b^1\right)\right\}\right]\ddot{\varsigma}_r + \left[\left\{\hat{A}\right\}^T \bullet \{Z(z)\}\right]\dot{\varsigma}_r + \left[\left\{\hat{B}\right\}^T \bullet \left\{H\left(\eta, q_b^1\right)\right\}\right] \\
&-\left[\left\{\hat{\Theta}\right\}^T \bullet \left\{\tilde{\Xi}\left(\eta, q_b^1\right)\right\}\right]\ddot{\varsigma}_r - \left[\left\{\hat{A}\right\}^T \bullet \left\{\tilde{Z}(z)\right\}\right]\dot{\varsigma}_r - \left[\left\{\hat{B}\right\}^T \bullet \left\{\tilde{H}\left(\eta, q_b^1\right)\right\}\right] \\
&+Kr + U_s
\end{aligned}
\tag{4.100}
$$

Simplifying the above equation:

$$
\begin{aligned}
&\left\{\left[\{\Theta\}^T \bullet \left\{\Xi\left(\eta, q_b^1\right)\right\}\right] + E_{M_L}\left(\eta, q_b^1\right)\right\}\dot{r} + \left\{\left[\{A\}^T \bullet \{Z(z)\}\right] + E_{C_L}(z)\right\}r \\
&+Kr + U_s \\
&=\left[\left\{\tilde{\Theta}\right\}^T \bullet \left\{\Xi\left(\eta, q_b^1\right)\right\}\right]\ddot{\varsigma}_r + \left[\left\{\tilde{A}\right\}^T \bullet \{Z(z)\}\right]\dot{\varsigma}_r + \left[\left\{\tilde{B}\right\}^T \bullet \left\{H\left(\eta, q_b^1\right)\right\}\right] \\
&+\left[\left\{\hat{\Theta}\right\}^T \bullet \left\{\tilde{\Xi}\left(\eta, q_b^1\right)\right\}\right]\ddot{\varsigma}_r + \left[\left\{\hat{A}\right\}^T \bullet \left\{\tilde{Z}(z)\right\}\right]\dot{\varsigma}_r + \left[\left\{\hat{B}\right\}^T \bullet \left\{\tilde{H}\left(\eta, q_b^1\right)\right\}\right] \\
&+E_1 + \tau_d
\end{aligned}
\tag{4.101}
$$

where $\tilde{\Theta} = \Theta - \hat{\Theta},\ \tilde{A} = A - \hat{A},\ \tilde{B} = B - \hat{B}$, $E_1 = E_{M_L}\left(\eta, q_b^1\right)\ddot{\varsigma}_r + E_{C_L}(z)\dot{\varsigma}_r + E_{F_L}\left(\eta, q_b^1\right)$.

Then we obtain:

$$
\begin{aligned}
&M_L\left(\eta, q_b^1\right)\dot{r} + C_L(z)r + Kr + U_s \\
&= \left[\left\{\tilde{\Theta}\right\}^T \bullet \left\{\Xi\left(\eta, q_b^1\right)\right\}\right]\ddot{\varsigma}_r + \left[\left\{\tilde{A}\right\}^T \bullet \left\{Z(z)\right\}\right]\dot{\varsigma}_r + \left[\left\{\tilde{B}\right\}^T \bullet \left\{H\left(\eta, q_b^1\right)\right\}\right] \\
&+ \left[\left\{\hat{\Theta}\right\}^T \bullet \left\{\tilde{\Xi}\left(\eta, q_b^1\right)\right\}\right]\ddot{\varsigma}_r + \left[\left\{\hat{A}\right\}^T \bullet \left\{\tilde{Z}(z)\right\}\right]\dot{\varsigma}_r + \left[\left\{\hat{B}\right\}^T \bullet \left\{\tilde{H}\left(\eta, q_b^1\right)\right\}\right] \\
&+ E_1 + \tau_d
\end{aligned}
\tag{4.102}
$$

The uncertain term $E = E_1 + E_2 + \tau_d$ is assumed to be bounded by $\|E\| \le \psi$, and the upper bound ψ can be estimated on-line.

The robust controller U_s is proposed as:

$$
U_s = \hat{\psi}\,\mathrm{sgn}(r) \tag{4.103}
$$

where $\hat{\psi}$ is the estimated value of the uncertain term bound ψ.

The parameter adaptation laws of the NN controller are updated on-line as follows:

$$
\begin{aligned}
&\dot{\hat{\theta}}_k = \Gamma_k \bullet \left\{\xi_k\left(\eta, q_b^1\right)\right\}\dot{\eta}_r r_k \\
&\dot{\hat{\alpha}}_k = Q_k \bullet \left\{\varsigma_k(z)\right\}\eta_r r_k \\
&\dot{\hat{\beta}}_k = N_k \bullet \left\{\sigma_k\left(\eta, q_b^1\right)\right\} r_k \\
&\dot{\hat{\psi}}_k = T_k r_k\,\mathrm{sgn}(r_k) \\
&\dot{\hat{c}}_{\theta_k} = E_{\theta_k} \bullet \left\{\xi_{kc}\right\}\hat{\theta}_k\dot{\eta}_r r_k,\ \dot{\hat{c}}_{\alpha_k} = E_{\alpha_k} \bullet \left\{\varsigma_{kc}\right\}\hat{\alpha}_k\eta_r r_k,\ \dot{\hat{c}}_{\beta_k} = E_{\beta_k} \bullet \left\{\sigma_{kc}\right\}\hat{\beta}_k r_k, \\
&\dot{\hat{b}}_{\theta_k} = I_{\theta_k} \bullet \left\{\xi_{kb}\right\}\hat{\theta}_k\dot{\eta}_r r_k,\ \dot{\hat{b}}_{\alpha_k} = I_{\alpha_k} \bullet \left\{\varsigma_{kb}\right\}\hat{\alpha}_k\eta_r r_k,\ \dot{\hat{b}}_{\beta_k} = I_{\beta_k} \bullet \left\{\sigma_{kb}\right\}\hat{\beta}_k r_k.
\end{aligned}
\tag{4.104}
$$

where $\Gamma_k = \Gamma_k^T > 0$, $Q_k = Q_k^T > 0$, $N_k = N_k^T > 0$, $T_k > 0$, $E_{\theta_k} = E_{\theta_k}^T > 0$, $E_{\alpha_k} = E_{\alpha_k}^T > 0$, $E_{\beta_k} = E_{\beta_k}^T > 0$, $I_{\theta_k} = I_{\theta_k}^T > 0$, $I_{\alpha_k} = I_{\alpha_k}^T > 0$, $I_{\beta_k} = I_{\beta_k}^T > 0$, $\hat{\theta}_k$ and $\hat{\alpha}_k$ are the vectors of $\hat{\theta}_{kj}$ and $\hat{\alpha}_{kj}$.

Theorem: Using the control law (4.96) and (4.97), the following hold for any $(\varsigma(0), \dot{\varsigma}(0)) \in \Omega_h$

- r converges to zero as t→∞.
- e_ς and $\dot{e}_\varsigma$ converge to zero as t→∞.

- e_λ is bounded for all t ≥ 0.

Proof:

(a) Choose the same Lyapunov function candidate with (4.52).

r converges to zero as t→∞ and e_ς and $\dot{e}_\varsigma$ converge to zero as t→∞ can be proved. The proofs are similar to those in section 4.3.3 and are therefore omitted here.

(b) Substituting the control (4.96) and (4.97) into the reduced order dynamics (4.81) yields:

$$\begin{aligned} &-(I+K_p)e_\lambda - K_i\int e_\lambda dt \\ &= -ZL^{\dagger T}D \end{aligned} \tag{4.105}$$

where $L^\dagger = (L^T L)^{-1}L^T$ and $D = \hat{M}_L\ddot{\varsigma}_r + \hat{C}_L\dot{\varsigma}_r + \hat{F}_L + Kr + U_s - C_L(\varsigma,\dot{\varsigma})\dot{\varsigma} - F_L - \tau_{dL}$.

Using the result of (a), we know $\hat{M}_L\ddot{\varsigma}_r + \hat{C}_L\dot{\varsigma}_r + \hat{F}_L + Kr + U_s \to M_L\ddot{\varsigma} + C_L(\varsigma,\dot{\varsigma})\dot{\varsigma} + F_L + \tau_{dL}$.

Thus $\hat{M}_L\ddot{\varsigma}_r + \hat{C}_L\dot{\varsigma}_r + \hat{F}_L + Kr + U_s - C_L(\varsigma,\dot{\varsigma})\dot{\varsigma} - F_L - \tau_{dL} \to M_L\ddot{\varsigma}$.

M_L, $\ddot{\varsigma}$ are bounded, then we can prove that D is bounded. Since Z, $L^{\dagger T}$ are all bounded, $ZL^{\dagger T}D$ is also bounded. Therefore, the size of e_λ can be adjusted by choosing the proper gain matrices K_p and K_i such that $e_\lambda \to 0$, then $\lambda_h \to \lambda_{hd}$.

4.5 Manipulation with a Dexterous Robotic Hand

4.5.1 Introduction

Traditionally, the robot grasping process is divided into four stages: initializing the configurations of the fingers and object, pre-grasping, grasping and manipulating. The main problem of grasp planning for object manipulation is to plan the movement of the finger joints according to the desired motion of the object. Figure 4.10 is a flow chart illustrating an intelligent grasping process. First, the desired trajectory of the object and the initial configuration of the hand-object system are given. Secondly, we compute the finger joint angles for pre-grasping. Then, with the initial joint velocities, joint angles, object velocity and contact coordinates, the contact kinematics can be computed for updating the contact coordinates and acquiring the grasp map. After that, a force optimization algorithm is applied to obtain the contact forces. Next, the desired object trajectory is tracked by the proposed

control method. If the task has not been completed, the robot continues the manipulation. The joint torques can be obtained and the DC motors drive the fingers to acquire a stable grasp. Accordingly, the joint velocities and angles are updated. Then the process is executed again. Using the process described above, the robot can successfully grasp the object and makes the object follow the desired trajectory.

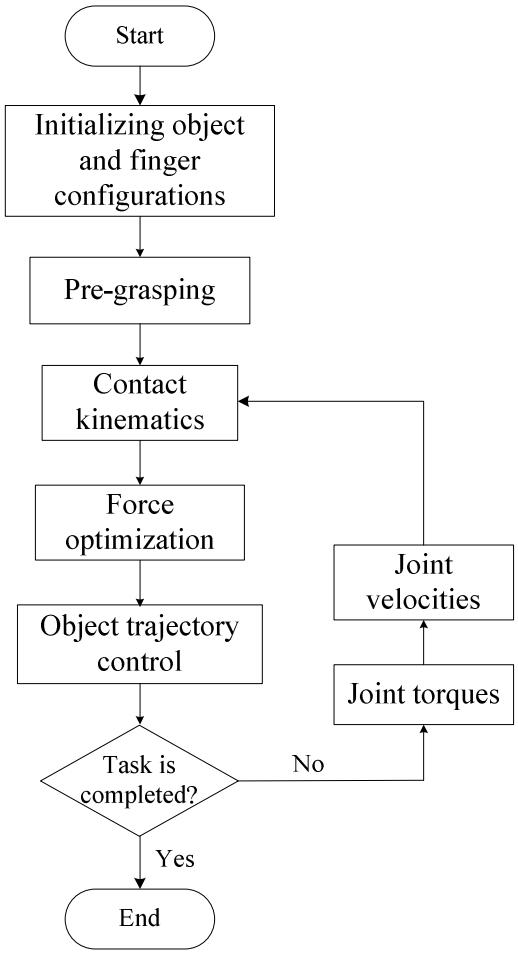


Figure 4.10: Grasping control flow chart

From this flowchart, it can be seen that the force optimization and the object trajectory are two important problems. For GFO problem, in section 1.2.7, we have discussed some approaches. For example, Han et al. [109] casted the friction cone constraints further into LMI and solved the grasping force problem by using the interior point algorithm; Buss et al. [110] reformulated the contact force optimization problem as a semi-definite programming problem. In section 3.1.4, the dynamics of the hand-object system and the robot-object system has been derived respectively. Assume that the dynamic and kinematic parameters of the systems are known accurately. For the second problem many classical dynamic model based trajectory tracking methods such as computed torque method and feedforward control law can be applied.

In this subchapter, LMI is used to optimize the grasping forces. Additionally, an object manipulation method is proposed by the computed torque method based on the accurate model for both the hand-object system and the robot-object system.

4.5.2 Grasping Force Optimization

Following is a method to ensure the stable grasp and prevent the slippage of the object. As already mentioned above, it's possible to find the internal forces (F_N) to ensure that the contact forces (f_c) are within the friction cones, without altering the wrench given to the center of the object.

Note that all contact forces are calculated as follows:

$$f_c = G^{\dagger} F_o + F_N \tag{4.106}$$

where F_o is assigned once the control law is chosen.

Recall that for the hard-finger contact model, the definition of the cone of friction is as following:

$$FC_{c_i} = \{ f_{c_i} \in \Re^3 : \frac{1}{\mu_i} \sqrt{f_{i1}^2 + f_{i2}^2} \le f_{i3},\ f_{i3} \ge 0 \} \tag{4.107}$$

where f_{i1} and f_{i2} denote the tangent components of the friction force; f_{i3} denotes the normal component of the contact force. For all fingers FC_c is defined.

The problem consists in finding a vector:

$$F_N \ :\ GF_N = 0,\ f_c = G^{\dagger} F_o + F_N \ \in \ FC_c \tag{4.108}$$

In this section, the problem described above is taken as a convex optimization problem [109] with LMI which can be accomplished by using convex optimization toolbox in Matlab. For contact *i* which is a point contact with friction the LMI is given by [110]:

$$P_i = \begin{bmatrix} \mu_i f_{i3} & 0 & f_{i1} \\ 0 & \mu_i f_{i3} & f_{i2} \\ f_{i1} & f_{i2} & \mu_i f_{i3} \end{bmatrix} \ge 0 \tag{4.109}$$

If we also impose constraints (maximum and minimum) on the normal forces to be exerted at the contact points, we can build the following LMI:

$$Sf_c \geq f_{\min}$$
$$-Sf_c \geq -f_{\mathrm{MAX}}$$

where $f_{\min}$ and f_{MAX} represent, respectively, the permitted maximum and minimum normal forces $S[\, f_{c1} \quad f_{c2} \quad \cdots \quad f_{cN}\,] = [\, f_{13} \quad f_{23} \quad \cdots \quad f_{N3}\,]$.

In the case of hard-finger contact, the matrix S is calculated as $S = [0 \quad 0 \quad 1]$.

The actuator torque capacity of the robotic hand is combined into the objective function to guide the optimization:

$$-\tau_{f_{ij}} \geq -\tau_{\mathrm{MAX}}$$

where $\tau_{f_{ij}}$ denotes the torque of each joint and τ_{MAX} is the permitted maximum actuator torque.

It can be shown that the constraints $P(f_c) \geq 0$, $Sf_c \geq f_{\min}$, $-Sf_c \geq -f_{\mathrm{MAX}}$ and $-\tau_{f_{ij}} \geq -\tau_{\mathrm{MAX}}$ are convex, for which we can apply a convex optimization algorithm to find the vector F_N. We can choose an objective function for the optimization problem to minimize the internal forces, appropriately weighted by a vector ω, and at the same time remain as far away as possible from the edges of the friction cones ($-\log(\det(P))$): $\arg\min(\omega^T F_N - \log(\det(P)))$ [109]. We obtain the following convex optimization problem:

$$\arg\min(\omega^T F_N - \log(\det(P)))$$
subject to
$$GF_N = 0$$
$$f_c = G^{\dagger} F_o + F_N$$
$$P(f_c) \geq 0$$
$$Sf_c \geq f_{\min}$$
$$-Sf_c \geq -f_{\mathrm{MAX}}$$
$$-\tau_{f_{ij}} \geq -\tau_{\mathrm{MAX}}$$

4.5.3 Controller Design

We present a control law that permits $X_o(t)$ to follow a desired trajectory $X_d(t)$ and at the same time ensures that all contacts in the grasping system meet the constraints of kinematic closure with the internal force F_N optimized above.

We put (3.45) into (3.84) and obtain:

$$f_c = G^{\dagger}(M_o\ddot{X}_o + C_o\dot{X}_o + N_o) + F_N \tag{4.110}$$

Then (3.83) can be written as:

$$\begin{aligned} &M_f J_h^{-1}(\dot{G}^T P^{-1}\dot{X}_o + G^T\dot{P}^{-1}\dot{X}_o + G^T P^{-1}\ddot{X}_o - \dot{J}_h J_h^{-1} G^T P^{-1}\dot{X}_o) \\ &+C_f J_h^{-1} G^T P^{-1}\dot{X}_o + F_f \\ &= \tau_f - J_h^T(G^{\dagger}(M_o\ddot{X}_o + C_o\dot{X}_o + N_o) + F_N) \end{aligned} \tag{4.111}$$

And rearranging it we have:

$$\begin{aligned} &(M_f J_h^{-1} G^T P^{-1} + J_h^T G^{\dagger} M_o)\ddot{X}_o \\ &= \tau_f - J_h^T(G^{\dagger}(C_o\dot{X}_o + N_o) + F_N) - F_f - C_f J_h^{-1} G^T P^{-1}\dot{X}_o \\ &-M_f J_h^{-1}(\dot{G}^T P^{-1}\dot{X}_o + G^T\dot{P}^{-1}\dot{X}_o - \dot{J}_h J_h^{-1} G^T P^{-1}\dot{X}_o) \end{aligned} \tag{4.112}$$

Both canceling the nonlinearity in the system and tracking the desired trajectory $X_d(t)$ can be achieved by using the proposed control scheme using computed torque method:

$$\begin{aligned} \tau_f = {} & J_h^T(G^{\dagger}(C_o\dot{X}_o + N_o) + F_N) + F_f + C_f J_h^{-1} G^T P^{-1}\dot{X}_o \\ & + M_f J_h^{-1}(\dot{G}^T P^{-1}\dot{X}_o + G^T\dot{P}^{-1}\dot{X}_o - \dot{J}_h J_h^{-1} G^T P^{-1}\dot{X}_o) \\ & + (M_f J_h^{-1} G^T P^{-1} + J_h^T G^{\dagger} M_o)(\ddot{X}_d + K_v\dot{E}_{X_o} + K_p E_{X_o}) \end{aligned} \tag{4.113}$$

where $\dot{E}_{X_o} = \dot{X}_d - \dot{X}_o$ and $E_{X_o} = X_d - X_o$, respectively, are the errors of the velocity and the trajectory of the object; while K_v and K_p are diagonal matrices whose values are, respectively, the derivative and the proportional coefficient.

The closed-loop system is uniformly ultimately bounded and the error vector satisfies the following equation:

$$\ddot{E}_{X_o} + K_v \dot{E}_{X_o} + K_p E_{X_o} = 0 \tag{4.114}$$

which represents the dynamics of the errors.

4.6 Object Manipulation with Mobile Manipulator

The control law used in this section is similar as above.

Recall that:

$$\begin{aligned}
&\bar{M}'\dot{\xi} + \bar{C}'\xi + \bar{F}' = \bar{E}'\tau' - J_B'^T f_c \\
&\xi = J_B'^{\dagger} G^T P^{-1} \dot{X}_o + N\sigma \\
&\dot{\xi} = J_B'^{\dagger}(\dot{G}^T P^{-1} \dot{X}_o + G^T \dot{P}^{-1} \dot{X}_o + G^T P^{-1} \ddot{X}_o - \dot{J}_B'(J_B'^{\dagger} G^T P^{-1} \dot{X}_o + N\sigma)) + N\sigma'
\end{aligned}$$

Substituting ξ, $\dot{\xi}$ and (4.106) into the dynamic equitation of the complete dynamic model which considers the external wrench:

$$\begin{aligned}
&\bar{M}'(J_B'^{\dagger}(\dot{G}^T P^{-1} \dot{X}_o + G^T \dot{P}^{-1} \dot{X}_o + G^T P^{-1} \ddot{X}_o - \dot{J}_B'(J_B'^{\dagger} G^T P^{-1} \dot{X}_o + N\sigma)) \\
&+ N\sigma') + \bar{C}'(J_B'^{\dagger} G^T P^{-1} \dot{X}_o + N\sigma) + \bar{F}' \\
&= \bar{E}'\tau' - J_B'^T (G^{\dagger}(M_o \ddot{X}_o + C_o \dot{X}_o + N_o) + F_N)
\end{aligned} \tag{4.115}$$

And rearranging it we have:

$$\begin{aligned}
&(\bar{M}' J_B'^{\dagger} G^T P^{-1} + J_B'^T G^{\dagger} M_o) \ddot{X}_o \\
&= \bar{E}'\tau' - J_B'^T (G^{\dagger}(C_o \dot{X}_o + N_o) + F_N) - \bar{F}' - \bar{C}'(J_B'^{\dagger} G^T P^{-1} \dot{X}_o + N\sigma) \\
&- \bar{M}'(J_B'^{\dagger}(\dot{G}^T P^{-1} \dot{X}_o + G^T \dot{P}^{-1} \dot{X}_o - \dot{J}_B'(J_B'^{\dagger} G^T P^{-1} \dot{X}_o + N\sigma)) + N\sigma')
\end{aligned} \tag{4.116}$$

We adopt the following method to track the desired trajectory $X_d(t)$:

$$\begin{aligned}
\bar{E}'\tau' = {} & J_B'^T (G^{\dagger}(C_o \dot{X}_o + N_o) + F_N) + \bar{F}' + \bar{C}'(J_B'^{\dagger} G^T P^{-1} \dot{X}_o + N\sigma) \\
& + \bar{M}'(J_B'^{\dagger}(\dot{G}^T P^{-1} \dot{X}_o + G^T \dot{P}^{-1} \dot{X}_o - \dot{J}_B'(J_B'^{\dagger} G^T P^{-1} \dot{X}_o + N\sigma)) + N\sigma') \\
& + (\bar{M}' J_B'^{\dagger} G^T P^{-1} + J_B'^T G^{\dagger} M_o)(\ddot{X}_d + K_v \dot{E}_{X_o} + K_p E_{X_o})
\end{aligned} \tag{4.117}$$

where $\dot{E}_{X_o} = \dot{X}_d - \dot{X}_o$ and $E_{X_o} = X_d - X_o$, respectively, are the errors of the velocity and the trajectory of the object while K_v and K_p are diagonal matrices whose values are, respectively, the derivative and the proportional coefficients.

Proof: We replace τ' with (4.117) in (4.116) and obtain:

$$\begin{aligned}\bar{M}'J_B^{\dagger}G^T P^{-1}\ddot{X}_o = (\bar{M}'J_B^{\dagger}G^T P^{-1} + J_B'^T G^{\dagger} M_o)(\ddot{X}_d + K_v\dot{E}_{X_o} + K_p E_{X_o}) \\ - J_B'^T G^{\dagger} M_o \ddot{X}_o\end{aligned} \tag{4.118}$$

Then the error dynamics is as following:

$$\ddot{E}_{X_o} + K_v\dot{E}_{X_o} + K_p E_{X_o} = 0 \tag{4.119}$$

The uniform ultimate boundedness of the tracking error can be proved from (4.119).

4.7 Simulation and Results

4.7.1 Path Planning Approach for Redundant Manipulator

The structure of a 7 DOFs manipulator discussed in this section is shown in Figure 4.11. It is an important part of the mobile manipulator which will be used in domestic environments as an assistant. This section takes the right arm as an example. As shown in Figure 4.11, the redundant manipulator is composed of a shoulder, an elbow and a wrist as the human arm. The robot's task is to reach one arbitrary posture $\begin{bmatrix} x & y & z & \alpha & \beta & \gamma \end{bmatrix}^T$ in task space when the obstacles are stationary in the environment.

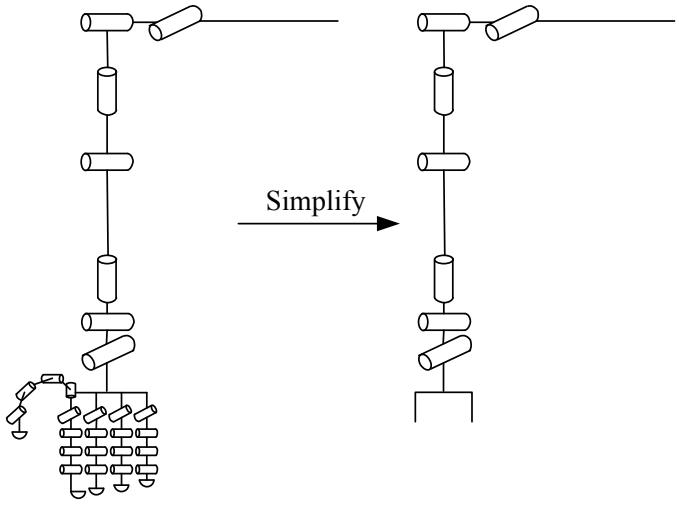


Figure 4.11: Structure and dimensions of a 7 DOFs manipulator

For the validation of the proposed RRT path planning algorithm, a complete model of the mobile manipulator has been built in Matlab. The structure of this robot is correspondent with the real one. All the links are simplified and are represented with lines. The task is designed as follows:

Given an initial configuration $\theta_{init} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T$ in configuration space and a desired posture $(p_d, \mathbf{q}_d)$ of the end-effector in task space, then find a continuous collision-free path in the environment, where $p_d = [0.25 \quad 0.3367 \quad 0.349]^T$ and the unit quaternion $\mathbf{q}_d$ can be obtained by the transformation of the desired Euler angles $[\frac{\pi}{2} \quad 0 \quad \frac{\pi}{2}]^T$.

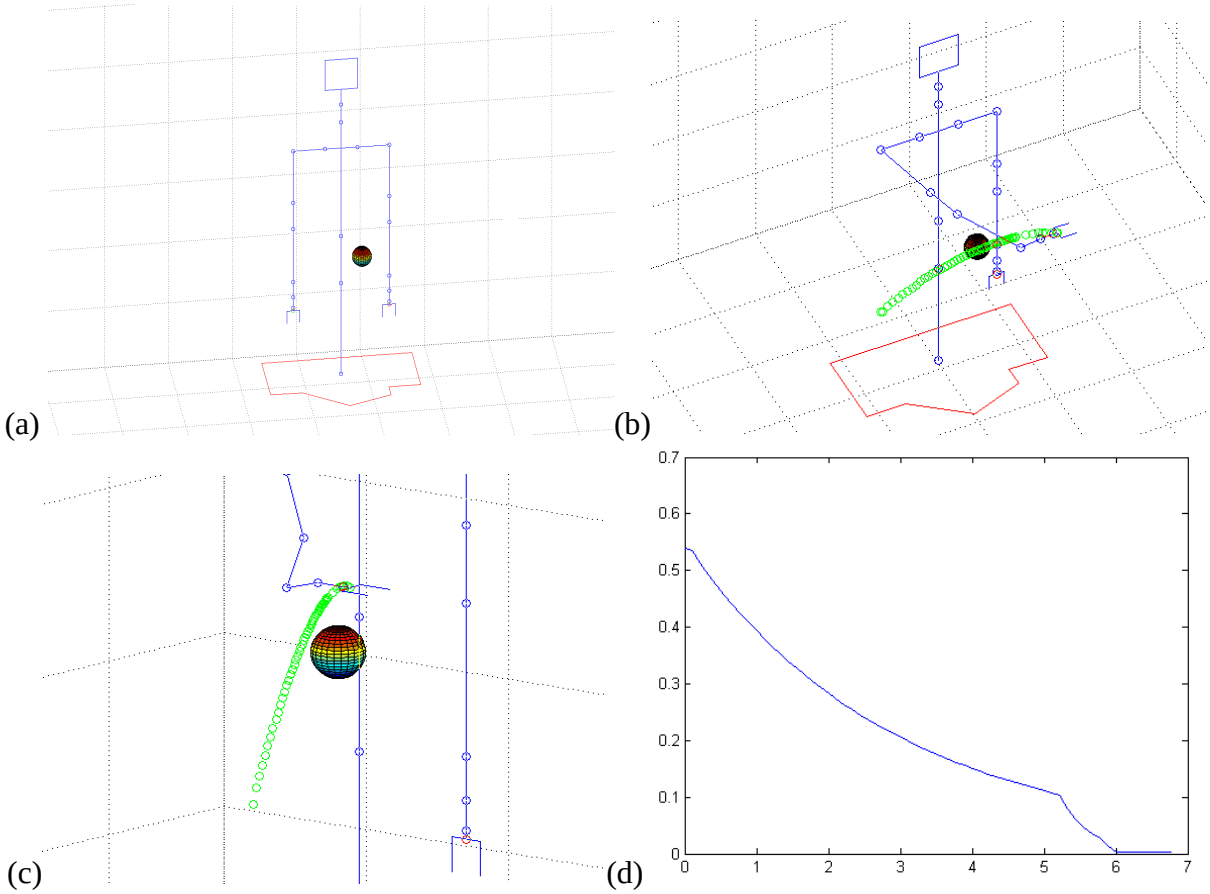


Figure 4.12: Simulation results (a) Initial state of robot (b) Computed trajectory towards goal posture (c) Collision-free path (d) Error to the goal

Figure 4.12 shows the simulation results based on the path planning approach. In these figures, the ball is taken as an obstacle. Figure 4.12(a) shows the initial configuration of the robot. From Figure 4.12(b), it can be seen that the pseudoinverse of Jacobian matrix control algorithm to guide the expansion of search tree makes the robot continually move towards the goal and avoid the redundant movements of the joints. The quaternion based orientation control is successfully implemented. We achieve a good overall performance in the regulation of the desired pose of the end-effector. The bisection gradient-decent extend algorithm limits each joint's angular velocity so that the movements of the joints are much smoother. The simulation results show the rapidity and validity of the proposed hybrid algorithm. From Figure 4.12(c), it is obvious that a collision-free path is generated by our improved RRT

algorithm. The error is shown in Figure 4.12(d) which indicates that the desired position and orientation of the redundant manipulator are successfully achieved. The error curve in Figure 4.12(d) is obtained by $e^T e$.

where $e = \begin{bmatrix} e_p \\ \mathbf{e}_0 \end{bmatrix} = \begin{bmatrix} p_d - p \\ \mathbf{e}_v \end{bmatrix} \in \Re^6$.

4.7.2 Path Planning Approach for Mobile Platform

We define a point to point task in an obstacle environment to test the RRT-Connect method. In Figure 4.13, the circle denotes an obstacle; the red line is the original path generated by basic RRT-Connect; the blue line is the optimized path. The result shows that a collision-free path has been searched and the optimized path is much shorter than the original one.

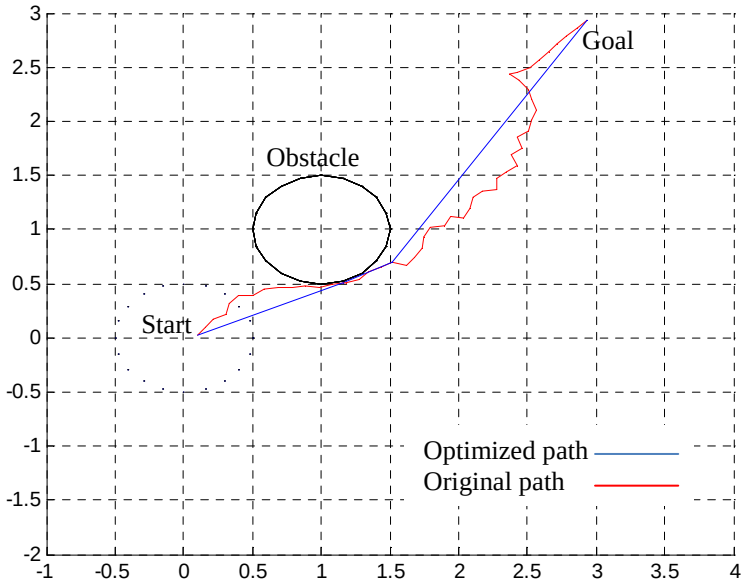


Figure 4.13: Comparison of original path and optimized path

The algorithm is applied to the mobile manipulator system. In Figure 4.14, the yellow points represent the trajectory of the mobile platform searched by the RRT-Connect algorithm; the red and green points denote the trajectories of left and right arms respectively. From this figure, we can clearly see that all these three trajectories are away from the obstacles.

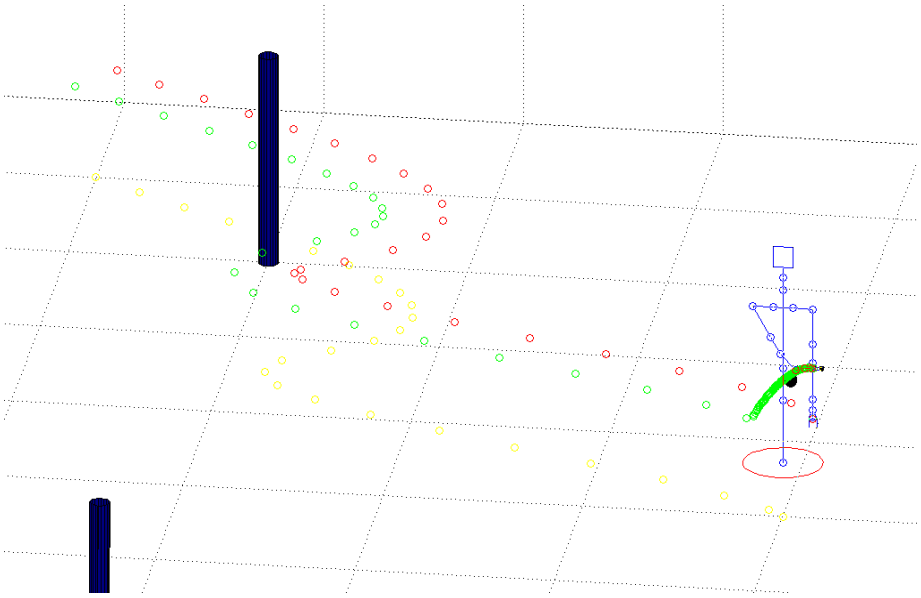


Figure 4.14: Generated trajectory using RRT-Connect path planning algorithm

4.7.3 Coordinated RBF Neural Network Adaptive Controller

In subsection 3.1.5.2, the computed torque control method is used to construct a closed-loop control system. This scheme is a model-based method which requires a reasonably accurate mathematical model. However, the control of mobile manipulators with unmodeled uncertainties and disturbances is essential in many practical applications. In this section, the virtual prototype and the bond graph model which have been validated are used to verify the efficacy of the proposed RBFNN adaptive control method. This method does not require any knowledge of the mobile manipulator.

For simplicity, we assume that the joints of the arms and the fingers are all fixed so that the mobile manipulator only has four DOFs (a mobile platform with two driving wheels and a two-link onboard waist). Figure 4.15 shows this case.

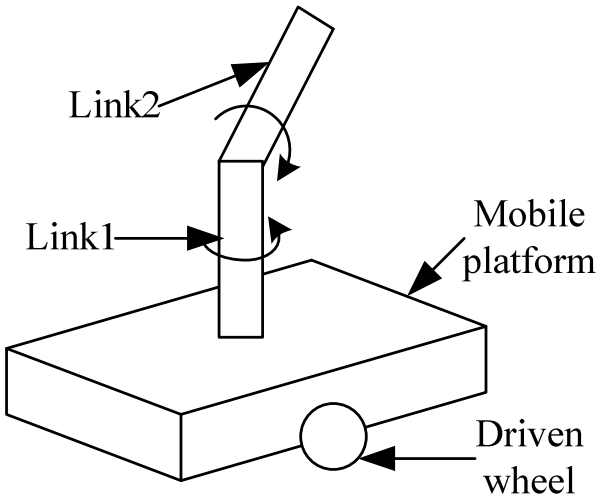


Figure 4.15: Model of a simplified nonholonomic mobile manipulator

Assume that we have no knowledge about the system and there is no payload. For each element of $\bar{M}$, $\bar{F}$ and $\bar{C}$ a 200-node NN is used. The gains for the controller are chosen as:

$\Gamma_k = diag[10]$, $Q_k = diag[10]$, $N_k = diag[10]$, $T_k = 10$, $E_{\theta_k} = E_{\alpha_k} = E_{\beta_k} = diag[10]$, $I_{\theta_k} = I_{\alpha_k} = I_{\beta_k} = diag[10], K = diag[30]$.

The control gains used in the kinematic control of the mobile platform are set by $k_x = 5,\ k_y = 4,\ k_\phi = 10$.

The reference velocity vector for the mobile platform is designed as:

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

Meanwhile, the reference trajectories for the onboard manipulator are given as:

$$[q_{1d} \quad q_{2d}]^T = [q_{10} + \sin(0.1\pi t) \quad q_{20} + \sin(0.1\pi t)]^T$$

where $[q_{10} \quad q_{20}]^T$ is the initial configuration of the manipulator.

To verify the effectiveness of the proposed method, two simulations are performed: (1) using the virtual prototype developed in ADAMS and (2) using the bond graph model.

(a) Validation using virtual prototype

The simulation results of the first case are illustrated in Figure 4.16. Figure 4.16(a) shows the practical trajectories, Figure 4.16(b) shows the trajectory tracking errors, Figure 4.16(c) depicts the practical velocities, Figure 4.16(d) gives the velocity tracking errors, Figure 4.16(e) shows the accelerations of the joints and Figure 4.16(f) describes the torques computed by the proposed control method. The simulation results prove the validity of the proposed algorithm.

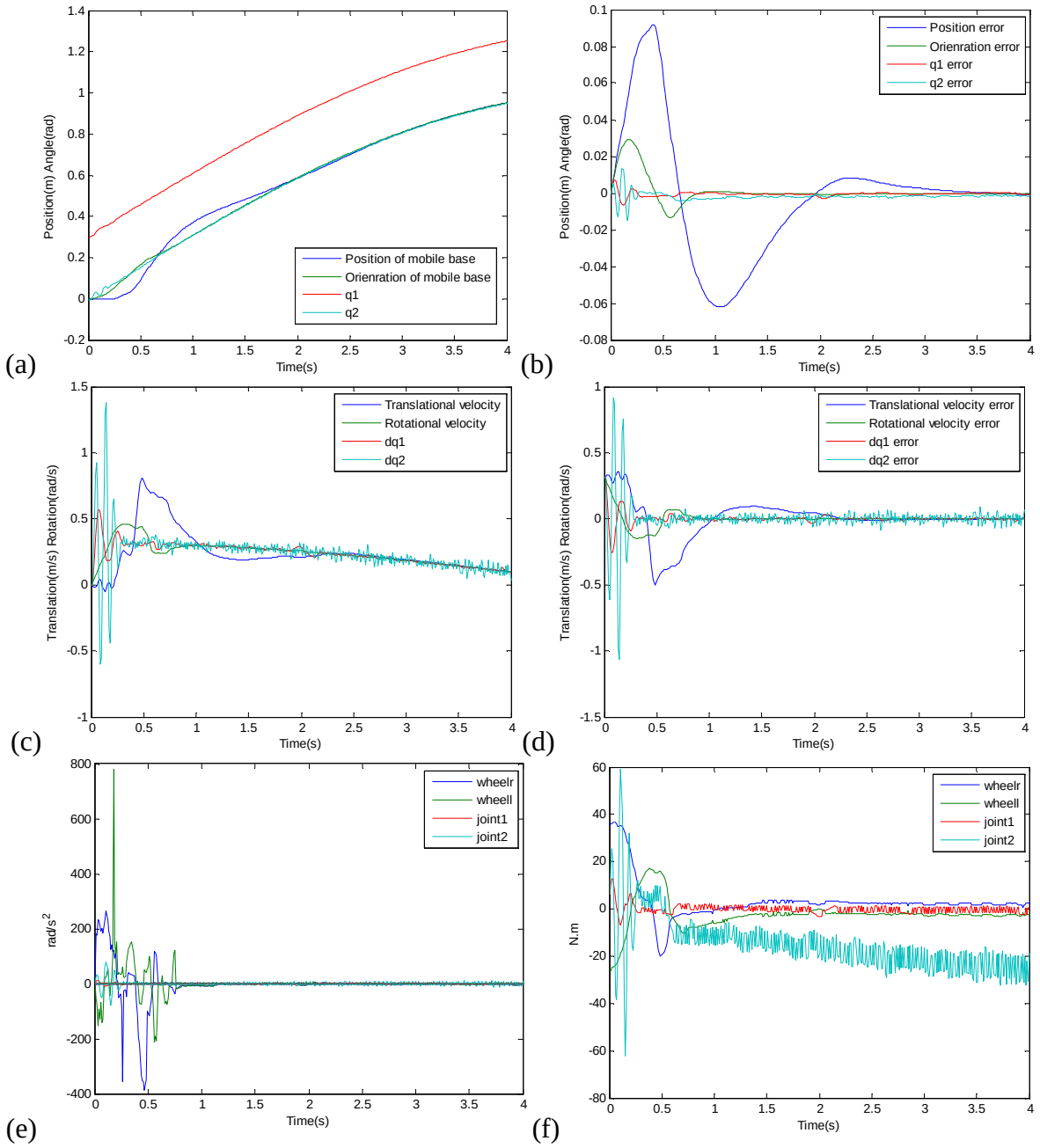


Figure 4.16: Simulation results based on ADAMS (a) Position tracking (b) Position tracking errors (c) Velocity tracking (d) Velocity tracking errors (e) Joint accelerations (f) Joint torques

(b) Validation using bond graph model

A. Adaptive control without disturbances

In case A, assume that there are no disturbances during the tracking. The simulation results are shown in Figure 4.17. Figure 4.17(a) shows the practical trajectories, Figure 4.17(b) shows the trajectory tracking errors, Figure 4.17(c) describes the practical velocities, and Figure 4.17(d) shows the velocity tracking errors.

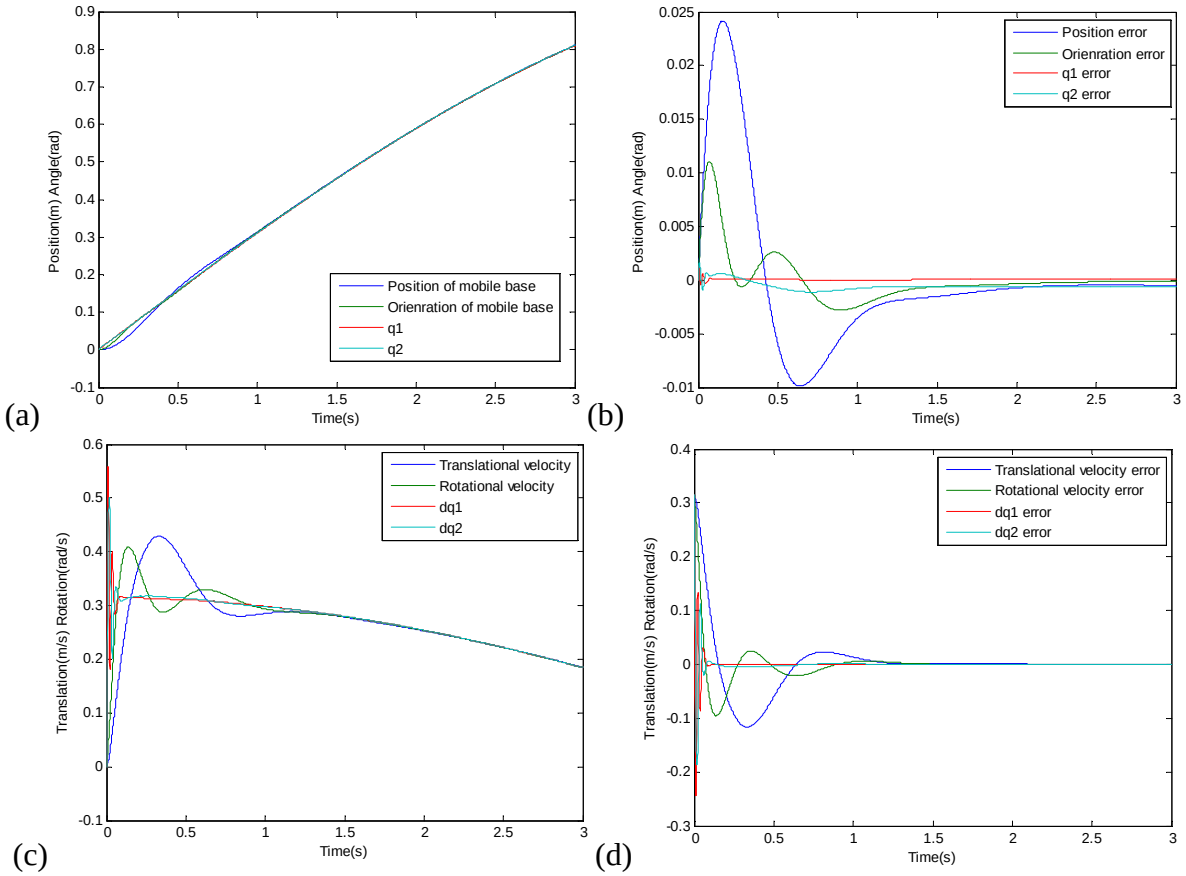


Figure 4.17: Simulation results in Case A (a) Position tracking without disturbances (b) Position tracking errors without disturbances (c) Velocity tracking without disturbances (d) Velocity tracking errors without disturbances

B. Adaptive control with disturbances

In case B, a continuous disturbance vector is added to each joint which is defined as:

$$\tau_d = \begin{bmatrix} 0.01\sin(2t) & 0.01\sin(2t) & 0.01\sin(2t) & 0.01\sin(2t) \end{bmatrix}^T$$

Figure 4.18(a), (b), (c) and (d) show the practical trajectories, the trajectory tracking errors, the practical velocities and the velocity tracking errors, respectively. It has been clearly observed from the above figures that, although we do not know the dynamic parameters completely, the proposed NN controller effectively attenuates the effects of uncertainties of the robot dynamics and disturbances, and the tracking errors converge to small values.

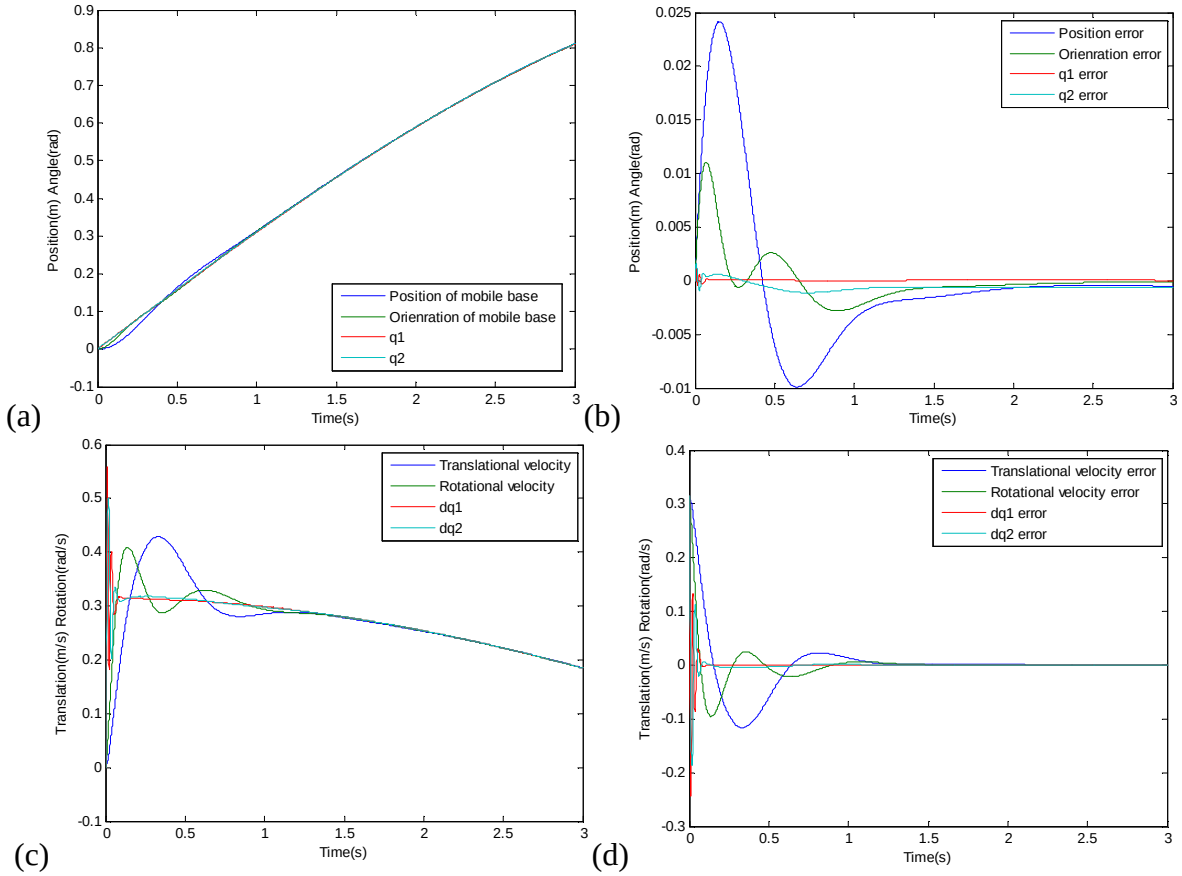
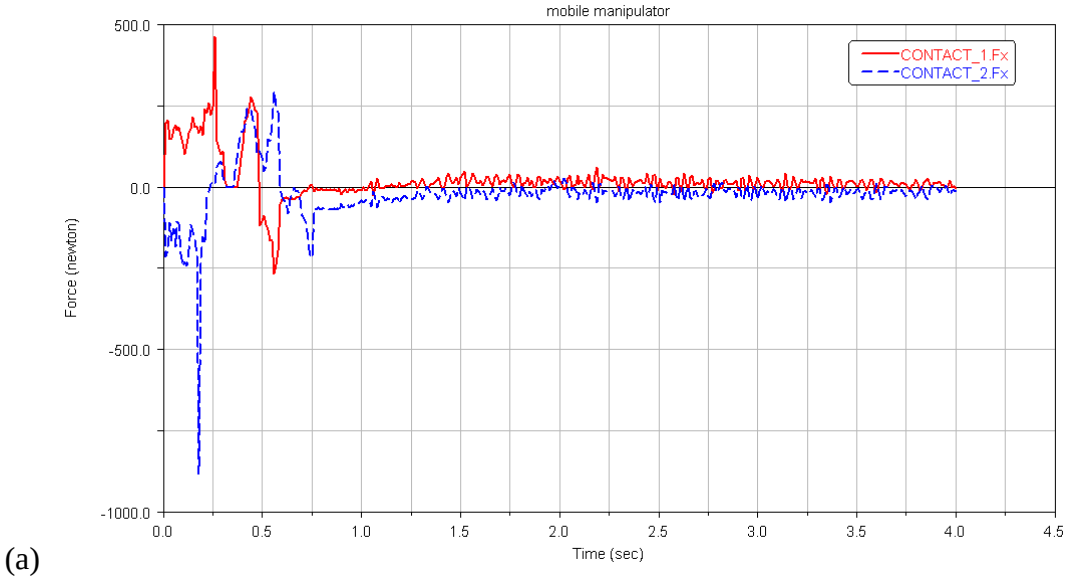
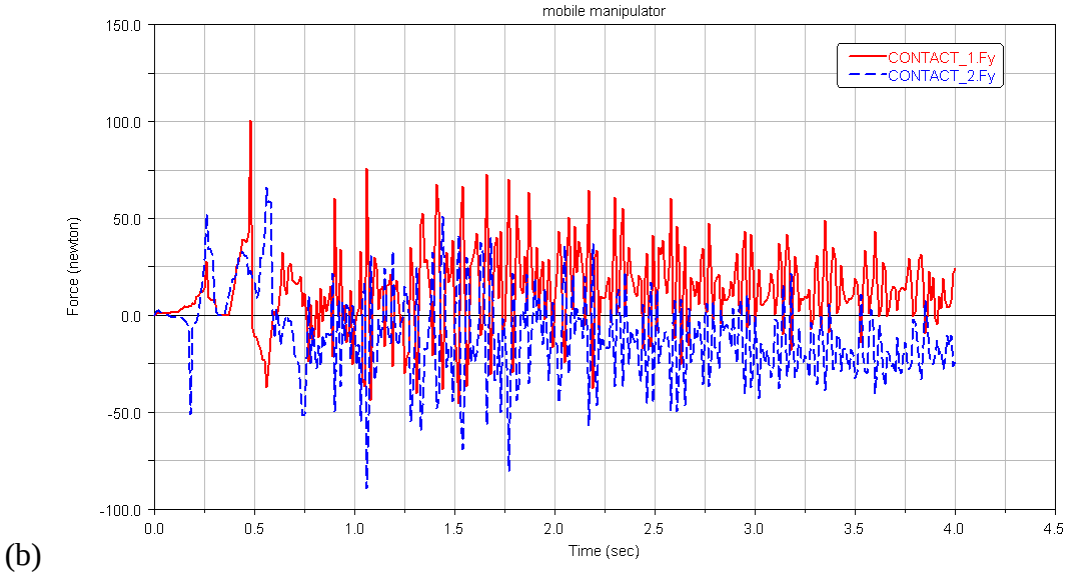


Figure 4.18: Simulation results in Case B (a) Position tracking with disturbances (b) Position tracking errors with disturbances (c) Velocity tracking with disturbances (d) Velocity tracking errors with disturbances

Through the comparison of the results in Figure 4.16, Figure 4.17 and Figure 4.18, it is obvious that the virtual prototype developed in ADAMS provides a better representation than the bond graph model to investigate dynamic behavior in a 3D environment. In the ADAMS environment, the contacts between the wheels and the ground, the stability of the robot and the frictions of the joints all can be analyzed. But in the bond graph model these could not be well modeled and expressed. However these effects which can be regarded as the disturbances and uncertainties affect the robot's performance greatly and it is necessary to add these factors to build a more realistic simulation environment. The frictions between the wheels and the ground detect by ADAMS are shown in Figure 4.19.



(a)



(b)

Figure 4.19: Frictions between the wheels and the ground (a) Frictions in *x*-direction (b) Frictions in *y*-direction

4.7.4 Motion/Force Control

The simplified 4 DOFs model in Figure 4.15 is used. Assume that we have no knowledge about the system and there is no payload.

Recall that the Lagrange coordinates of the mobile platform are $q_v = \begin{bmatrix} x_c & y_c & \phi \end{bmatrix}^T$ which represent the position of the robot in the world coordinate system and the constraints can be written in the form:

$$A_v(q_v)\dot{q}_v = 0 \tag{4.120}$$

where $A_v(q_v) = \begin{bmatrix} -\sin\phi & \cos\phi & 0 \end{bmatrix}$.

Define $v = \begin{bmatrix} \dot{\theta}_r & \dot{\theta}_l \end{bmatrix}^T$, then its integral $\eta = \begin{bmatrix} \theta_r & \theta_l \end{bmatrix}^T$.

Let the Lagrange coordinates of the manipulator be $q_b = \begin{bmatrix} q_1 & q_2 \end{bmatrix}^T$. Assume that the end-effector of the manipulator is subject to the geometric constraint: $h = l_1 + l_2 \sin(q_2) = 0$, thus

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, J^{'1} = \begin{bmatrix} 0 & 0 & 0 & l_2 \cos(q_2) \end{bmatrix}.$$

For each element of M_L, F_L and C_L, a 200-node NN is used. The gains for the controller are chosen as:

$\Gamma_k = diag[10]$, $Q_k = diag[10]$, $N_k = diag[10]$, $T_k = 10$, $E_{\theta_k} = E_{\alpha_k} = E_{\beta_k} = diag[10]$, $I_{\theta_k} = I_{\alpha_k} = I_{\beta_k} = diag[10], K = diag[30]$, $K_p = 5, K_i = 5$.

The reference trajectories for the mobile platform are designed as:

$$\eta_d = \begin{bmatrix} \sin(0.1\pi t) & \sin(0.1\pi t) \end{bmatrix}^T$$

Meanwhile, the reference trajectory for the onboard manipulator is given by:

$$q_{1d} = \sin(0.1\pi t)$$

The desired constraint force λ_{hd} is $10N$.

A continuous disturbance vector is added to the system which is defined as:

$$\tau_d = \begin{bmatrix} 0.01\sin(2t) & 0.01\sin(2t) & 0.01\sin(2t) & 0.01\sin(2t) \end{bmatrix}^T$$

The simulation is conducted using Matlab. The several results of the motion/force tracking control using the proposed control scheme are shown in Figure 4.20. Figure 4.20(a) shows the torques applied to the joints, Figure 4.20(b) shows the velocity tracking errors of the joints, Figure 4.20(c) is the tracking error of the constraint force, and Figure 4.20(d) illustrates the constraint force.

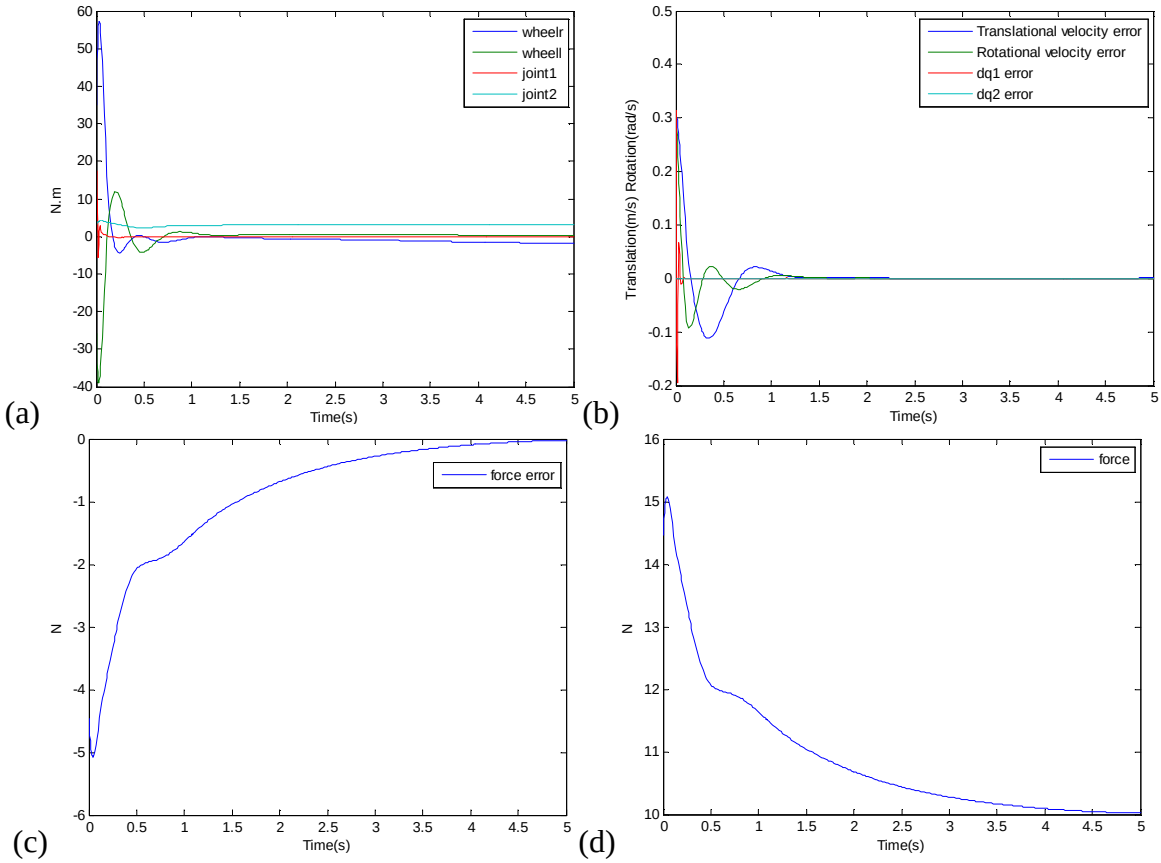


Figure 4.20: Simulation results of motion/force control (a) Joint torques (b) Velocity tracking errors of the joints (c) Tracking error of the constraint force (d) Constraint force

From the above figures, we can see that although we do not know the dynamic parameters of the model, the proposed RBFNN controller can effectively decrease the effects of uncertainties and disturbances. The trajectories and the constraint force converge to the desired ones, respectively. The simulation results validate the usefulness of the proposed control method.

4.7.5 Object Manipulation by Dexterous Hand

Cutkosky and Howe have specified a well known grasp taxonomy for a single hand grasp. The grasp hierarchy offers a classification scheme for typical grasps occurring in manufacturing tasks. According to Cutkosky's grasp taxonomy [99], we use one precision grasp (type 13) for object manipulation. This type is one of the most commonly used types of human grasps. It is able to operate a variety of tasks and also has strong adaptability to the shape of the object. The distribution of the fingers on the object and each joint's rotation angle for pre-grasp planning are dependent on the specific grasping tasks.

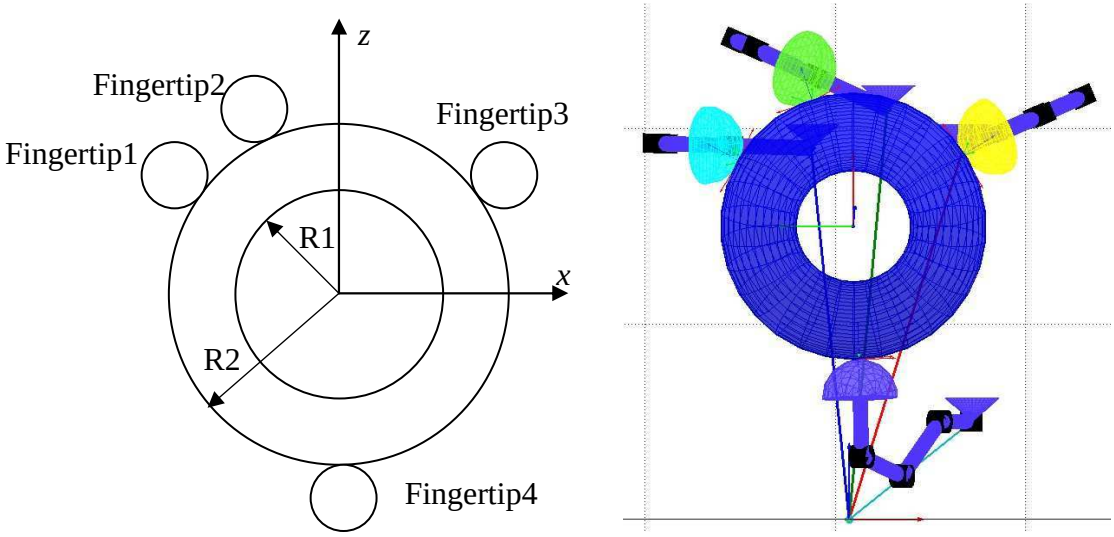


Figure 4.21: Distribution of fingers on the object for pre-grasp

We take grasping a torus as an example. The torus' minor radius a_o is 10mm and its major radius b_o is 25mm. Each fingertip is a semisphere whose radius is 10mm. The distribution of the fingertips on the torus is shown in Figure 4.21. The initial position and orientation of the object is $X_{oinit} = \begin{bmatrix} 0 & -0.055 & 0.075 & \frac{1}{2}pi & -\frac{5}{12}pi & 0 \end{bmatrix}^T$. In simulations, let end time be 0.01 seconds, step size be 0. 00001. The reference trajectory for the center of the object is given by $x = 0,\ y = -0.055 - 100t^2,\ z = 0.075$. The orientation of the torus does not change.

A comparison is made between the calculated results using the mathematical model and the operation results of the virtual simulation model established by ADAMS. Figure 4.22 shows the calculated results based on the mathematical model using the proposed manipulation control scheme. The trajectory of the torus, the torques applied to the joints and the contact forces varying with respect to time are obtained in Matlab. Figure 4.23 gives the joint toques and the normal forces when the actuator capacity is taken into account during the GFO.

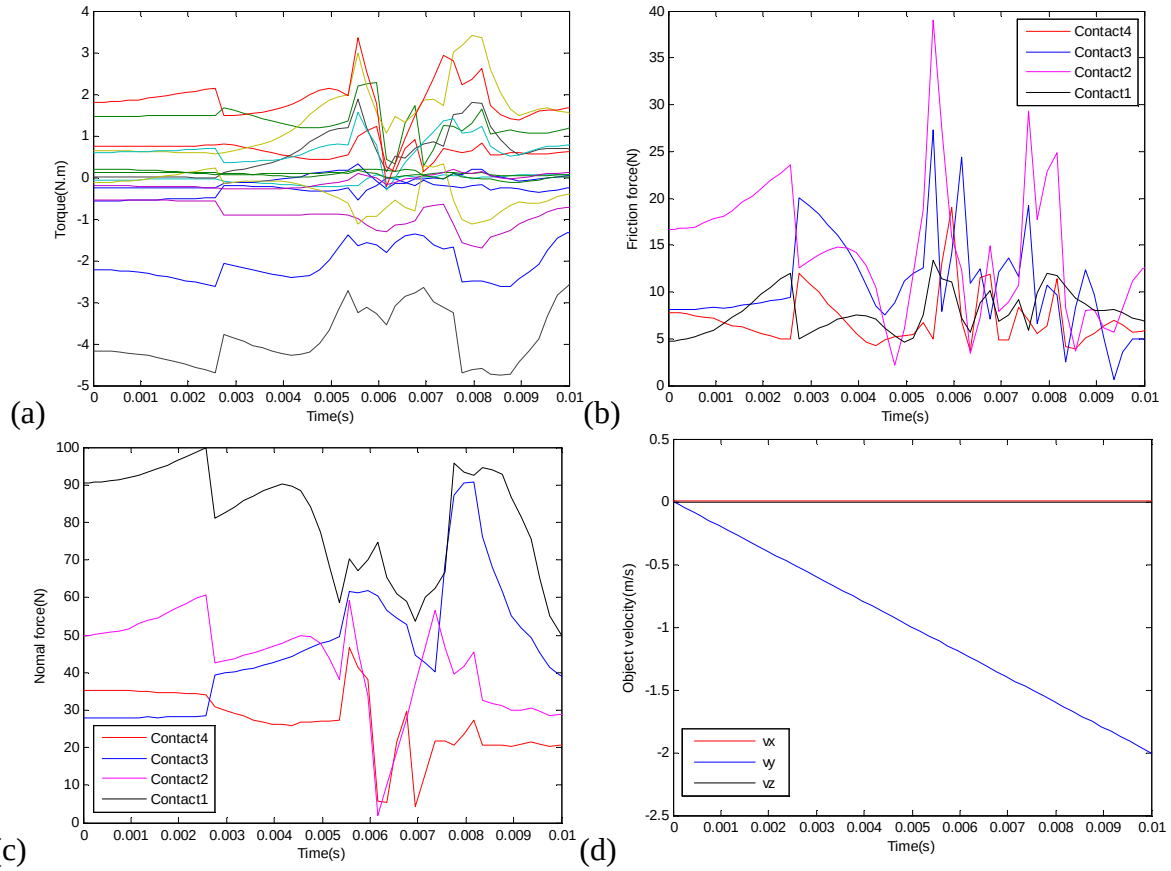


Figure 4.22: Calculated simulation results (a) Torques applied to the joints (b) Friction forces (c) Normal forces (d) Object velocity analysis

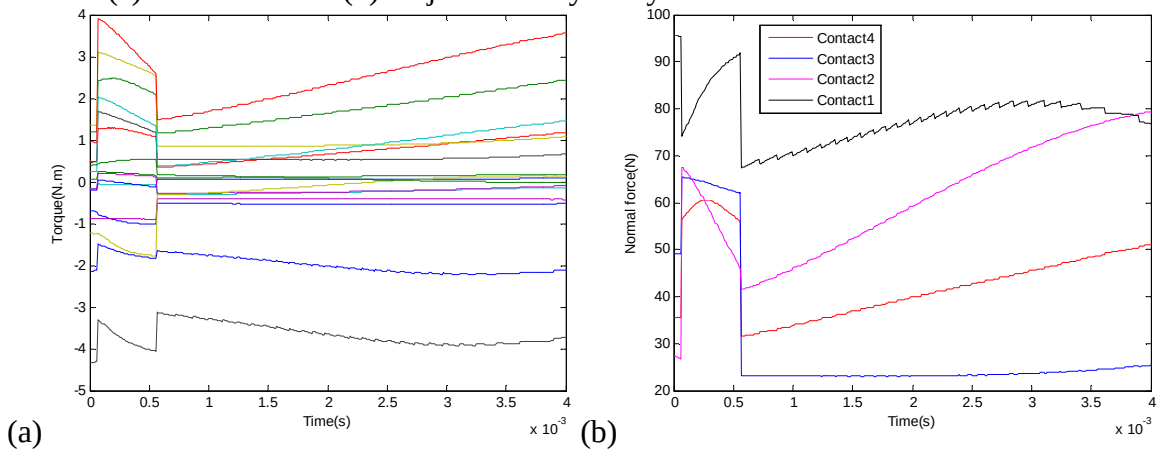
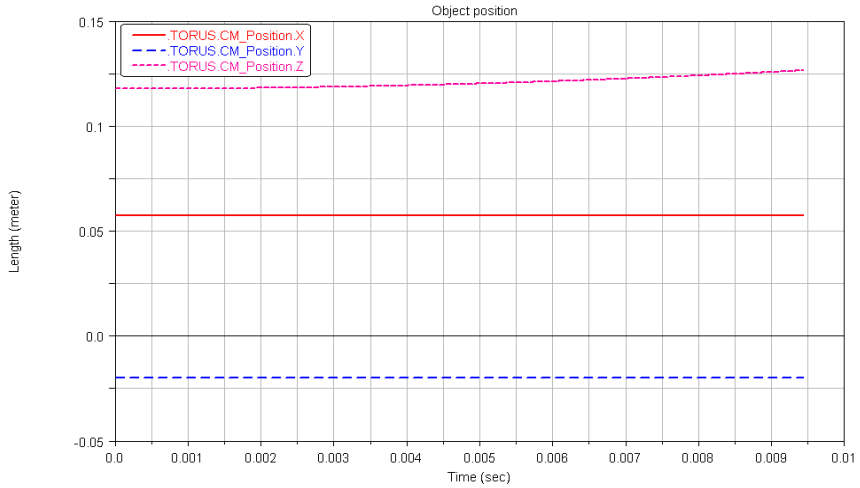
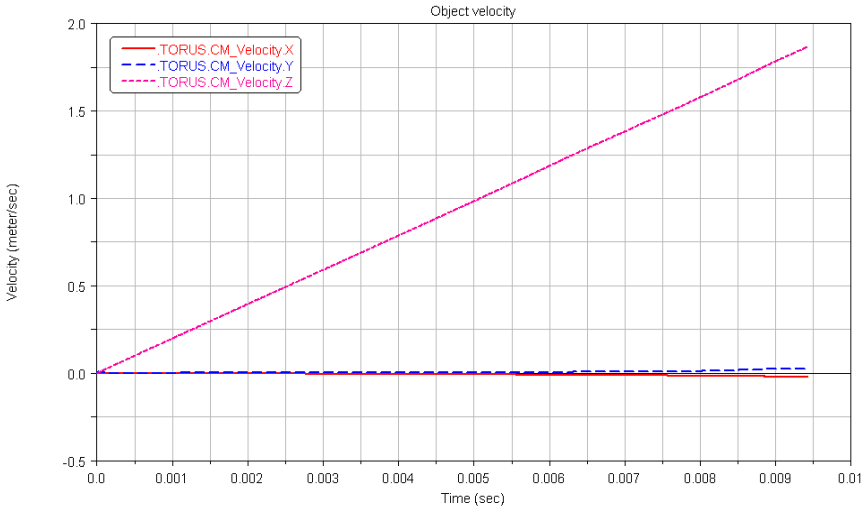


Figure 4.23: Joint torque optimization (a) Joint torques (b) Normal forces

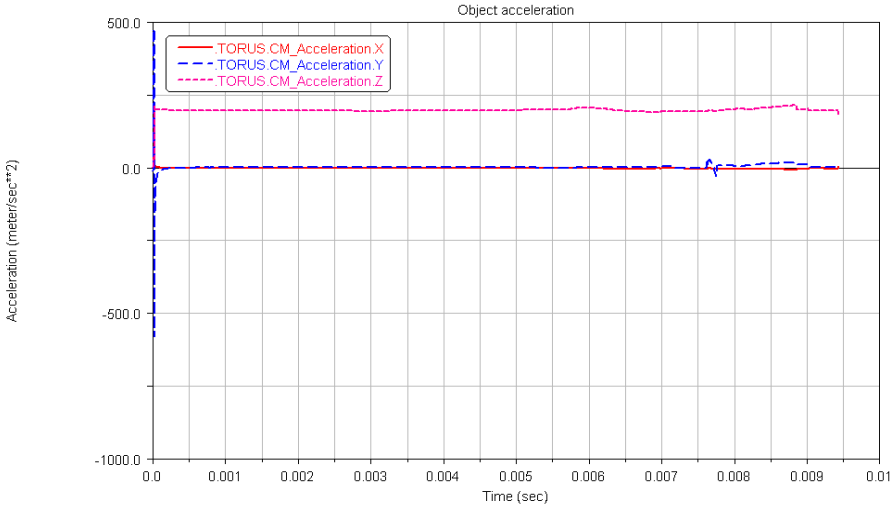
Secondly, ADAMS is used to simulate and detect the trajectory of the object and the contact forces between the hand and the object during grasp simulation, as shown in Figure 4.24. Figure 4.24(a), (b), (c) and (d) show the object position, object velocity, object acceleration and the contact forces which are the sums of normal and tangential forces. Note that in ADAMS, the coordinate system is defined as *x-zy* in correspondence to that defined in Matlab *xyz*. Additionally, due to the different coordinate of the original point in ADAMS, the initial position of the object's center of mass is $\begin{bmatrix} 0.0575 & -0.0198 & 0.118 \end{bmatrix}^T$.


Object position
TORUS.CM_Position.X
TORUS.CM_Position.Y
TORUS.CM_Position.Z
Length (meter)
0.15
0.1
0.05
0.0
-0.05
0.0
0.001
0.002
0.003
0.004
0.005
0.006
0.007
0.008
0.009
0.01
Time (sec)

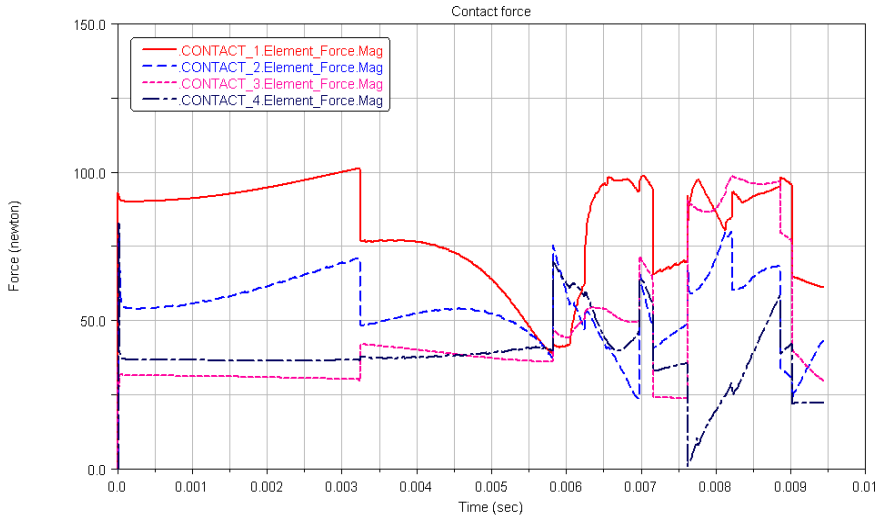
(a)


Object velocity
TORUS.CM_Velocity.X
TORUS.CM_Velocity.Y
TORUS.CM_Velocity.Z
Velocity (meter/sec)
2.0
1.5
1.0
0.5
0.0
-0.5
0.0
0.001
0.002
0.003
0.004
0.005
0.006
0.007
0.008
0.009
0.01
Time (sec)

(b)


Object acceleration
TORUS.CM_Acceleration.X
TORUS.CM_Acceleration.Y
TORUS.CM_Acceleration.Z
Acceleration (meter/sec**2)
500.0
0.0
-500.0
-1000.0
0.0
0.001
0.002
0.003
0.004
0.005
0.006
0.007
0.008
0.009
0.01
Time (sec)

(c)



(d)

Figure 4.24: Simulation of object manipulation in ADAMS (a) Object position (b) Object velocity (c) Object acceleration (d) Contact forces

By comparison, the motion of the object in ADAMS is in accordance with that in the calculated results. The sums of the friction forces in Figure 4.22(b) and the normal forces in Figure 4.22(c) are consistent with the contact forces detected by ADAMS shown in Figure 4.24(d). These simulation results show that the model of the dexterous hand built in ADAMS is effective and it lays a solid foundation for designing and manufacturing the real prototype. The results also verify the reasonableness of the control scheme for object manipulation.

4.7.6 Object Manipulation by Dual-Hand Mobile Manipulator

This section gives some simulation results of grasping an object with the complete robotic system. Before the simulation using ADAMS, the robot with the dexterous hands is developed using Matlab (Figure 4.25) which has the same structure as the real one. And all the links are simplified and are represented with columns. The initial configuration of the center of mass of the object is $X_{oinit} = [x_0 \quad y_0 \quad z_0 \quad \alpha_0 \quad \beta_0 \quad \gamma_0]^T$, where the first three components represent the position of the object relative to the base frame, and the last three ones describe the orientation of the object.

Assume that a precision grasp is applied to the object. The task is designed as follows: starting from the initial configuration, the personal assistant robot grasps an object from the right hand side while making the left hand-arm system keep the initial configuration; given an initial configuration X_{oinit}, the object moves along a desired trajectory described by following function $X_o = [x_0 + t \quad y_0 \quad z_0 \quad \alpha_0 \quad \beta_0 \quad \gamma_0]^T$.

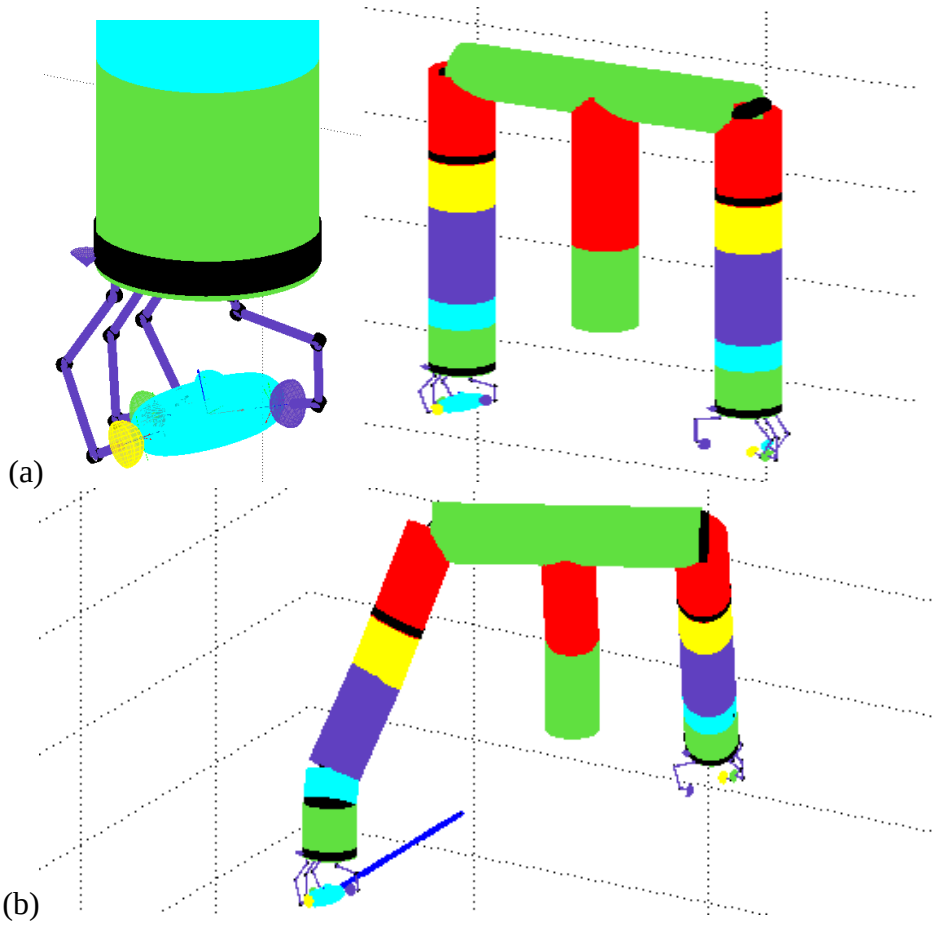


Figure 4.25: Object manipulation by mobile manipulator (a) Initial configuration of mobile manipulator and object system (b) Final configuration of mobile manipulator and object system

Figure 4.25 and Figure 4.26 show the numerical simulation results of object manipulation by the complete mobile manipulator. Figure 4.25(a) and Figure 4.25(b) illustrate the initial and final configurations of the composite system during one run of the grasping algorithm. From Figure 4.25(b), it is observed that the robotic system can successfully manipulate the object without slippage, and at the same time the joints of left hand-arm system keep stationary.

The object can well follow the desired trajectory as illustrated in Figure 4.26(a) which shows the motions in the x, y and z directions. The Euler angles which represent the orientation are shown in Figure 4.26(b). In Figure 4.26(c), threes curves depict the motion of the mobile platform which assists the hand to achieve the task. Figure 4.26(d), (e) and (f) show the joint torques of the wheels, the waist and dual arms calculated by the algorithm, respectively. The simulation results validate the usefulness of the proposed control method.

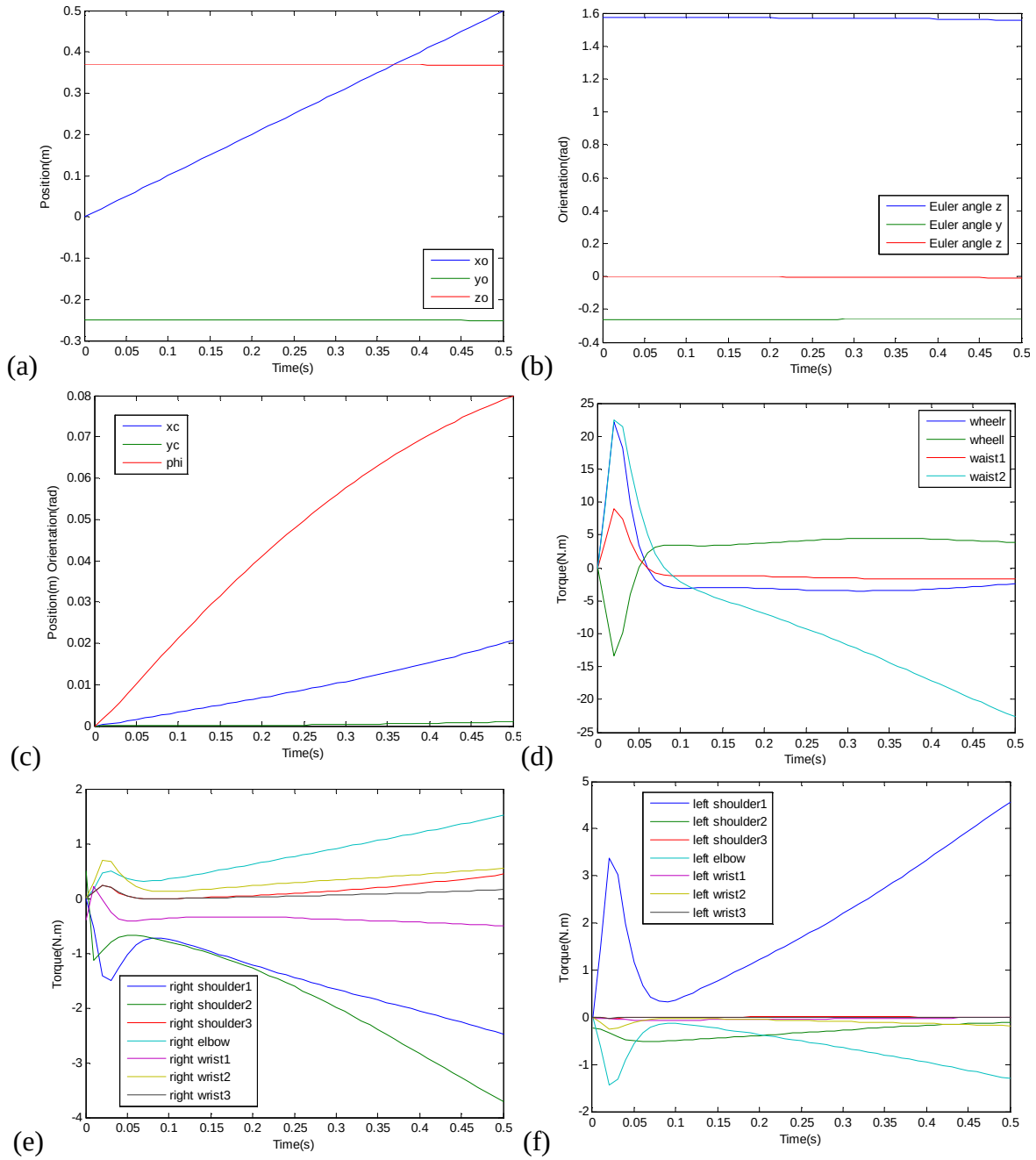


Figure 4.26: Simulation results (a) Motions in *x*, *y*, *z*-direction (b) Orientation change (c) Motions of mobile base (d) Joint torques of wheels and waist (e) Joint torques of right arm (f) Joint torques of left arm

The optimal normal and tangential forces at all contact points are described in Figure 4.27. It can be seen that the contact forces all satisfy the friction cone constraints.

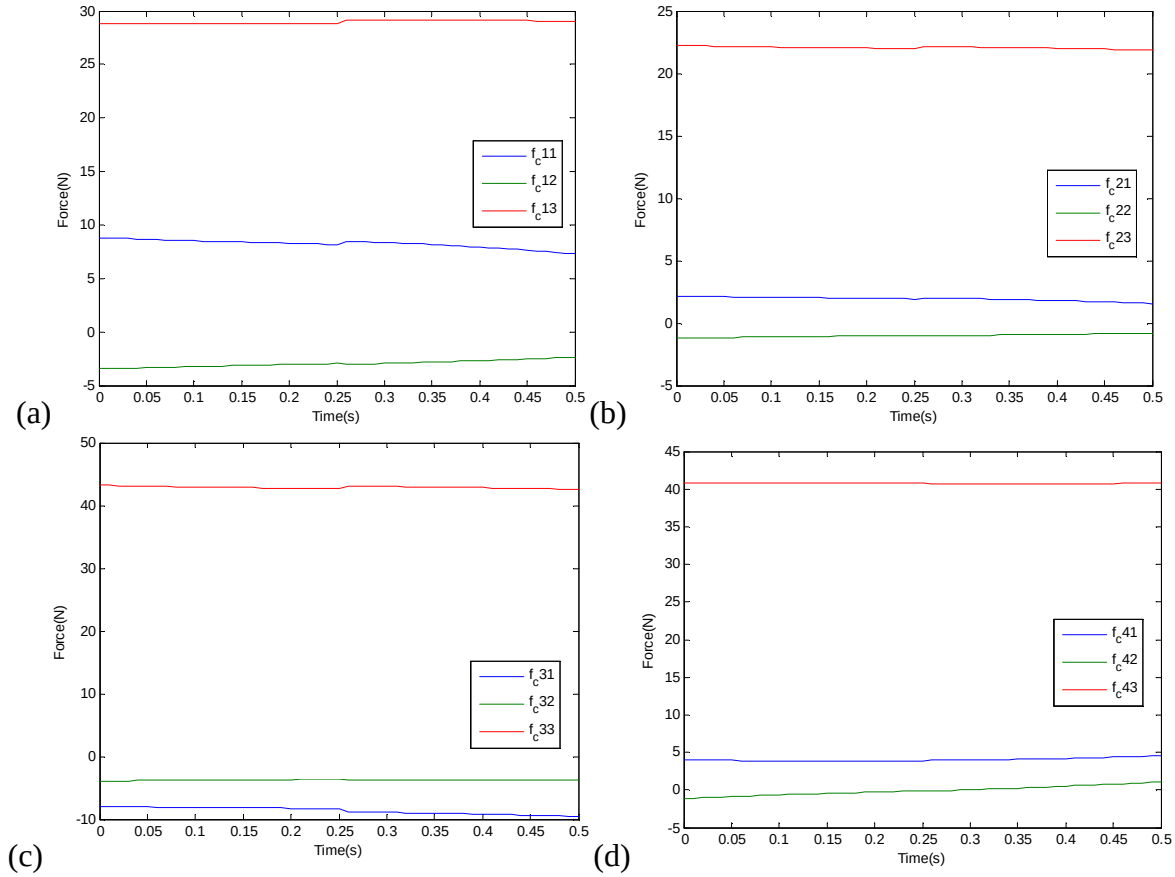


Figure 4.27: Magnitude of contact forces (a) Contact forces of finger 1 (b) Contact forces of finger 2 (a) Contact forces of finger 3 (b) Contact forces of finger 4

Conclusions, Contributions and Future works

Conclusions

This thesis designed and modeled a personal assistant robot as a daily helper and proposed some control schemes. Such a complex robot is reduced to a rigid body system with 56 DOFs consisting of a nonholonomic mobile base, a two DOFs waist, a two DOFs neck, two 7 DOFs human-like arms and two dexterous hands.

In this thesis, the mechanical, mathematical, and bond graph models are built respectively. A virtual prototype is created with MSC.ADAMS. First a solid model has been elaborated in Solidworks and then been imported into ADAMS environment. In ADAMS, the model has been carried out by introducing specific design characteristics and constraints. The proper characteristics give a realistic behavior for the simulation of robot operations. Using the virtual prototype, kinematic and dynamic simulations have been elaborated. The robot behavior of performing practical applications has been checked by simulating a variety of human-like operations. The simulation results have been used to validate the operation feasibility as well as to adjust the overall design of the robot. Consequently, ADAMS permits to decrease the time for the design steps and avoid the possibility of errors during the construction of a real robot. And the cost of the robot development can be reduced too.

The mathematical model consists of a set of differential equations describing the dynamics of the system. Based on the MDH method and the Euler-Lagrange method, the kinematic and dynamic equations of the robot's motion are instructed detailedly. We use the mathematical model to establish a control sytem for controlling the ADAMS model which has been validated. Through co-simulation, the mathematical model is proved to be in accordance with the virtual prototype. Additionally, modeling of the complex mobile manipulator system using bond graph approach is introduced. By writing the model in the bond graph language step by step, a model with reusable components has been created. The simulation structure is implemented using Simulink. This model has been validated under an open-loop controller through the precise mathematical model to show its effectiveness. It is proved that the application of bond graph facilitates the modeling of this complex electromechanical system, as well as the extension of a control system.

We have presented an improved RRT algorithm based on inverse kinematics for the motion planning of a redundant manipulator. We have found it to be very effective for this problem. RRT method can efficiently solve the collision-free path planning problem for the high-dimensional robot. Quaternion is applied in the computation of orientation error, and its feedback is augmented into the control scheme for concurrent control. Jacobian pseudoinverse algorithm determines the optimal expansion direction of search tree, and this method speeds up the movement to the goal node. This presented algorithm applies not only to the static environment, but also can solve the path planning in dynamic environment.

We have presented a novel robust adaptive NN control method for a mobile manipulator to track a desired trajectory in joint space. The proposed scheme consists of RBFNN, PD type controller and robust controller for compensating for the approximation errors, uncertainties and disturbances. The weights of NN are trained on-line and the upper bound of errors, Gaussian function parameters centers c and widths b are also estimated on-line. All the signals in the closed-loop control system are guaranteed to be asymptotically stable. The simulation results validate the usefulness of the proposed control method. A similar RBFNN control method is presented for coordinated motion/force control of a holonomic constrained nonholonomic mobile manipulator. The trajectory and the constraint force converge to the desired ones, respectively.

In this thesis a separate model is developed for the robotic dexterous hand using Solidworks and ADAMS. Object manipulation is realized in ADAMS environment. The simulation results show that the model is effective and it lays a solid foundation for designing and manufacturing a real prototype. We have presented an efficient control method for object manipulation and demonstrated the ability of our algorithm to consistently generate force-closure grasps in a few seconds. This algorithm also can be applied to the complete dynamic model of the humanoid mobile manipulator system grasping an object. And it is also able to provide firm grip, avoid slippage and well track the desired trajectory given to the object.

Contributions

In summary, this research has provided the following main contributions:

- Chapter 2 gives the application design for the personal assistant robot. This robot's ultimate goal is the execution of complex manipulation tasks in domestic environments. And the requirements for realizing those applications are discussed.

- Chapter 2 presents the details of the mechanical design. A simulation environment for this mobile manipulator is constructed based on co-simulation between ADAMS and Simulink.

- The complete dynamic model of the robot is established using Lagrange's approach and bond graph respectively. (Chapter 3)

- A collision-free path planning algorithm for a redundant manipulator with the given desired pose is presented which combines the Jacobian pseudoinverse algorithm with RRT. (Chapter 4)

- A robust RBF network-based control scheme for motion control of a nonholonomic mobile manipulator in the presence of uncertainties and disturbances is developed. (Chapter 4)

- A robust RBF network-based control scheme for coordinated force/motion control of a nonholonomic mobile manipulator with holonomic constraints is developed. (Chapter 4)

- Considering the kinematics and dynamics of a dexterous hand manipulating an object with known shape by rolling contacts, a control scheme is designed to ensure firm grip, avoid slippage and well track a given motion imposed to the object. (Chapter 7) This control method is also applicable to the dual-arm/hand mobile manipulator system. (Chapter 4)

Future Research

There is still plenty of work that needs to be done.

In the short term

The models presented in this thesis can be further developed with complicated path planning and control algorithms so that the robot can accomplish more advanced tasks. For example, the motion of soft-fingertip manipulation will be analyzed. This is usually true not only for human grasping, but also for robotic hands using fingertips made of soft materials. In order to analyze the motion of using soft fingertips, the dynamic model will be built and some dynamic manipulation control schemes will be designed. The proposed motion planning algorithm will be verified in a dynamic environment with moving obstacles. Additionally, coordinating the motions of multiple robots operating in a shared workspace without collisions will be studied. What's more, the motion stability of the robot will also be discussed in future research.

In the long term

Our ultimate goal is to create a real personal assistant robot for assisting humans in domestic environments. The satisfactory results in the illustrated simulations of the robot are encouraging to proceed in the project to build a physical prototype for experimental tests in a near future. The details of our mechanical design related to the various modules or subsystems will be developed. For example, mass distribution, link and joint design, mechanical component of the sensors, actuator including reduction gear and DC motor selection etc. The selection of motor and gear ratio is based on the speed at the motor shaft and load torque. Thus, the simulation parameters (e.g. all joint driving forces) which have been obtained from the 3D virtual prototype simulations provide an important basis for the precise motor selection. Due to the modification of mechanical structure, the 3D solid model in Solidworks will be modified. Correspondingly, the parameters of the quality, inertia and position of each part's center of mass will be changed and acquired using ADAMS.

After the physical prototype has been created, based on the models (mathematical and bond graph models) built in this thesis according to the ideal design parameters, small changes to these models are required by identifying the parameters of the real robot.

REFERENCES

[1] T. Asfour, K. Regenstein, and P. Azad et al, “ARMAR-III: An integrated humanoid platform for sensory-motor control,” in Proc. 6th IEEE-RAS International Conference on Humanoid Robots, 2006.

[2] Meka webpage: http://mekabot.com/products/m1- mobile-manipulator/.

[3] David Thomas Pitts, “Dynamic analysis of mobile manipulators,” University of Toronto.

[4] QRIO, available: http://www.sony.net/SonyInfo/News/Press_Archive/200312/03-060E/.

[5] ASIMO, available: http://asimo.honda.com/default.aspx.

[6] Dexter, http://www-robotics.cs.umass.edu/Research/Humanoid/humanoid_index.html.

[7] Domo, available: http://people.csail.mit.edu/edsinger/domo.htm.

[8] Z. P. Jiang and H. Nijmeijer, “Tracking control of mobile robots: a case study in backstepping,” Automatica, vol. 33, no. 7, pp. 1393-1399, 1997.

[9] Y. Yamamoto and X. Yun, “A modular approach to dynamic modeling of a class of mobile manipulators,” International Journal of Robotics and Automation, vol. 12, no. 2, pp. 41-48, 1997.

[10] H. G. Tanner and K. J. Kyriakopoulos, “Mobile manipulator modeling with Kane’s approach,” Robotica, vol. 19, pp. 675-690, 2001.

[11] L. Sheng, A. A. Goldenberg, “Neural-network control of mobile manipulators,” IEEE Transactions on Neural Networks, vol. 12, no. 5, pp. 1121-1133, 2001.

[12] P. C. Breedveld, “Port-based modeling of mechatronic systems,” Mathematics and Computers in Simulation, 66(2-3):99 –128, 2004, selected papers from the 4th IMACS Symposium on Mathematical Modelling.

[13] Wolfgang Borutzky, Bond Graph Methodology, Springer, 2010.

[14] A. Mukherjee, R. Karmakar, “Modeling and simulation of engineering systems through Bondgraphs,” Narosa Publishing House, New Delhi, 2000.

[15] A. M. Bos, M. J. L. Tiernego, “Formula manipulating in the Bond Graph modeling and simulation of large mechanical systems,” Journal of Franklin Institute, vol. 319, no. 1/2, pp. 51-55, 1985.

[16] E. P. Fahrenthold, J.D. Worgo, “Vector Bond Graph analysis of mechanical systems”, Trans. of ASME, vol. 113, pp. 344-353, 1991.

[17] D. Karnopp, “Approach to derivative causality in Bond Graph models of mechanical systems”, Journal of Franklin Institute, vol. 329, no. 1, 1992.

[18] J.J. Granda, “Computer aided design of dynamic systems,” http://gaia.csus.edu/~grandajj/.

[19] Y. Yamamoto, X. Yun, “Coordinating locomotion and manipulation of a mobile manipulator,” in Proc. the 31th Conference on Decision and Control, Tucson, AZ, pp. 2643-2648, Dec.1992.

[20] H. Seraji, “Motion Control of mobile manipulators,” in Proc. IEEE International Conference on Intelligent Robots and Systems, pp. 2056-2063, 1993.

[21] W. F. Carriker, P. K. Khosla and B. H. Krogh, “Path planning for mobile manipulators for multiple task execution,” IEEE Trans. on Robotics and Automation, vol. 7, no. 3, pp. 403-408, 1991.

[22] G. Foulon, J. Y. Fourquet, M. Renaud, “Planning point to point paths for nonhonolomic mobile manipulators,” in Proc. IEEE International Conference on Intelligent Robots and Systemsm, pp. 374-379, 1998.

[23] K. Tchon, J. Jakubiak, R. Muazynski, “Regular jacobian motion planning algorithms for mobile manipulators,” IFAC[C], 2002.

[24] H. G. Tanner, K. J. Kyriakopoulos, “Nonholonomic motion planning for mobile manipulators,” in Proc. IEEE International Conference on Intelligent Robots and Automation, pp. 1233-1238, 2000.

[25] M. Akira, F. Seiji, M. Yamamoto, “Trajectory planning of mobile manipulator with end-Effector’s specified path,” in Proc. IEEE International Conference on Intelligent Robots and Systems, pp. 2264-2269, 2001.

[26] J. Tan and N. Xi, “Unified model approach for planning and control of mobile manipulators,” in Proc. IEEE International Conference on Robotics and Automation, vol. 3, pp. 3145–3152, 2001.

[27] Q. Huang, K. Tanie, S. Sugano, “Coordinated motion planning for a mobile manipulator considering stability and manipulation,” International Journal of Robotics Research, vol. 19, no. 8, pp. 732-742, 2000.

[28] M. Haddad, S. Hanchi, and H. E. Lehtihet, “Point-to-point trajectory planning of wheeled mobile manipulators with stability constraint. Extension of the random-profile approach,” Eur. J. Mech., A/Solid, vol. 28, pp. 477- 493, 2009.

[29] I. Kolmanvsky, N. H. McClamroch, “Developments in nonholonomic control problems,” Control Systems Magazine, pp. 20-36, 1995.

[30] Y. M. Hu, Q. J. Zhou, H. L. Pei. Theory and application of nonholonomic control system, 1996.

[31] L. E. Kavraki, M. N. Koulountzakis, and J. C. Latombe, “Analysis of probabilistic roadmaps for path planning,” IEEE Trans. on Robotics and Automation, vol. 14, no. 1, pp. 166–171, 1996.

[32] N. M. Amato and Y. Wu, “A randomized roadmap method for path and manipulation planning,” in Proc. IEEE International Conference on Robotics and Automation, 1996.

[33] S. LaValle, “Rapidly-exploring random trees: A new tool for path planning,” Comput. Sci. Dept., Iowa State Univ., Ames, IA, Tech. Rep. TR 98-11, 1998.

[34] J. Kuffner and S. LaValle, “RRT-Connect: An efficient approach to single-query path planning,” in Proc. IEEE International Conference on Robotics and Automation, pp. 995–1001, 2000.

[35] S. S. Ge and Y. J. Cui, “New potential functions for mobile robot path planning,” IEEE Trans. Robotics and Automation, vol. 16, no. 5, pp. 615–620, October 2000.

[36] Y. Koren and J. Borenstein, “Potential field methods and their inherent limitations for mobile robot navigation,” in Proc. IEEE Conference on Robotics and Automation, pp. 1398–1404, April 1991.

[37] F. Lingelbach, “Path planning using probabilistic cell decomposition,” in Proc. International Conference on Robotics and Automation, 2004. p.1.

[38] G. J. Weins, “Effect of dynamic coupling in mobile robotic systems,” in Proc. SME Robotic Research World Conference, pp.43-57, 1989.

[39] G. J. Wiens, W. M. Jang, “Passive joint control of dynamic coupling in mobile robots,” International Journal of Robotics Research, vol. 13, no. 3, pp. 209-220, 1994.

[40] Y. Yamamoto, X. P. Yun, “Control of mobile manipulators following a moving surface,” in Proc. IEEE International Conference on Intelligent Robots and Automation, vol. 3, pp. 1-6, 1993.

[41] H. Lee, T. Takubo, H. Arai, and K. Tanie, “Control of mobile manipulators for power assist systems,” in Proc. IEEE Int. Conf. on Systems, Man and Cybernetics (SMC’99), vol.4, pp. 989-994, 1999.

[42] S. Jagannathan, S. Q. Zhu and F. L. Lewis, “Path planning and control of a mobile base with nonholonomic constraints,” Robotica, vol. 12, part 6, pp. 529-540, 1994.

[43] P. Evangelos, P. John, “Planning and model-based control for mobile manipulators,” in Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1810-1815, 2000.

[44] K. Liu, F. L. Lewis, “Decentralized continuous robust controller for mobile robots,” in Proc. IEEE International Conference on Intelligent Robots and Automation, pp. 1822-1827, 1990.

[45] L. Sheng, A. A. Goldenberg, “Robust damping control of mobile manipulators,” IEEE Transaction on systems, Man and Cybernetics, part B, vol. 32, no. 1, pp. 126-132, 2002.

[46] J. H. Chung, S. A. Velinsky and R. A. Hess, “Interaction control of a redundant mobile manipulator,” International Journal of Robotics Research, vol. 17, no. 12, pp. 1302-1309, 1998.

[47] Y. Yamamoto, X. P. Yun, “Coordinated obstacle avoidance of a mobile manipulator,” in Proc. IEEE International Conference on Intelligent Robots and Automation, pp. 2255-2260, 1995.

[48] P. Ogren, M. Egerstedt, X. Hu, “Reactive mobile manipulation using dynamic trajectory Tracking,” in Proc. IEEE International Conference on Intelligent Robots and Automation, pp. 3473-3478, 2000.

[49] J. D. Tan, N. Xi, Y. Wang, “Hybrid force/position control of redundant mobile manipulators,” IFAC[C], 2002.

[50] W. J. Dong, Y. S. Xu, Q. Wang, “On tracking control of mobile manipulators,” in Proc. IEEE International Conference on Intelligent Robots and Automation, pp. 3455-3460, 2000.

[51] W. J. Dong, W. L. Xu, “Tracking control of mobile manipulators,” Control and Decision, vol. 06, 2001.

[52] W. J. Dong, W. L. Xu, “Robust control of mobile manipulators,” Control Theory and Applications, vol. 03, 2002.

[53] W. J. Dong, W. L. Xu, “Robust control of an uncertain nonholonomic mobile manipulator,” Tsinghua Science and Technology, vol. 09, 2002.

[54] S. Jagannathan, “Discrete-time fuzzy logic control of a mobile robot with an onboard manipulator,” International Journal of Systems Science, vol. 28, no. 12, pp. 1195-1209, 1997.

[55] Y. Yamamoto and X. Yun, “Unified analysis on mobility and manipulability of mobile manipulators,” in Proc. IEEE International Conference on Robotics and Automation, pp. 1200–1206, 1999.

[56] S. Kang, K. Komoriya, K. Yokoi, “Reduced inertial effect in damping-based posture control of mobile manipulator,” in Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.488-493, 2001.

[57] Y. Yamamoto, X. P. Yun, “Effect of the dynamic interaction on coordinated control of mobile manipulators,” IEEE Transaction on Robotics and Automation, vol. 12, no. 5, pp.816-824, 1996.

[58] S. Ghasempoor, “A measure of stability for mobile manipulator,” in Proc. IEEE International Conference on Intelligent Robots and Automation, pp. 2249-2254, 1995.

[59] D. A. Rey and E. G. Papadopoulos, “On-line automatic tipover prevention for mobile manipulators,” in Proc, IEEE International Conference on Intelligent Robots and Systems, Grenoble, France, pp. 1273-1278, Sep. 1997.

[60] K. Yoneda, “Tumble stability criterion of integrated locomotion and manipulation,” Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 870-876, 1996.

[61] S. Dubowsky and E. E. Vance, “Planning mobile manipulator motions considering vehicle dynamic stability constraints,” in Proc. IEEE International Conference on Robotics and Automation, Scottsdale, pp.1271-1276, 1989.

[62] S. Sugano, “Stability control for mobile manipulator using potential method,” in Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.839-846, 1994.

[63] Q. Huang, S. Sugano, and K. Tanie, “Stability compensation of a mobile manipulator by manipulator motion: feasibility and planning,” Advanced Robotics, vol. 13, no. 1, pp. 25–40, 1999.

[64] N. Hogan, “Impedance control: an approach to manipulation: Part I-Theory; Part II - Implementation; Part III –Applications,” ASME J. Dynamic Systems, Measurement, and Control, vol. 107, no. 1, pp. 1-24, 1985.

[65] H. Kazerooni, et al, “Robust compliant motion for manipulators, Part I, II,” IEEE Journal of robotics and automation, vol. 2, no. 2, pp. 83-105, 1986.

[66] J. K. Salisbury, “Active stiffness control of a manipulator in Cartesian coordinates,” in Proc. IEEE Conference on Decision and Control, pp. 87-97, 1980.

[67] H. West, and H. Asada, “A method for the design of hybrid position/force controllers for manipulators constrained by contact with the environment,” IEEE Int. Conf. Robotics and Automation, pp. 251-259, St. Louis, MO, 1985.

[68] S. Chun-yi, Y. Stepanenko, “Robust motion/ force control of mechanical systems with classical nonholonomic constraints,” IEEE Trans on Automatic Control, vol. 39, no. 3, pp.609-614, 1994.

[69] M. Oya, C. Y. Su, and R. Katoh, “Robust adaptive motion/force tracking control of uncertain nonholonomic mechanical systems,” IEEE Trans. Robotics and Automation, vol. 19, no. 1, pp.175-181, 2003.

[70] W. Dong, “On trajectory and force tracking control of constrained mobile manipulators with parameter uncertainty,” Automatica, vol. 38, pp. 1475-1484, 2002.

[71] Z. J. Li, “Robust motion/force control of nonholonomic mobile manipulators using hybrid joints,” Advanced Robotics, vol. 21.no. 11, pp. 1231-1252.

[72] Z. P. Wang, S. S. Ge, and T. H. Lee, “Robust motion/force control of uncertain holonomic/nonholonomic mechanical systems,” IEEE Trans. Mechatronics, vol. 9, pp.118-123, March 2004.

[73] Z. J. Li, S. S. Ge, A. Ming, “Adaptive robust motion/ force control of holonomic-constrained nonholonomic mobile manipulators,” IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics, vol.37, no.3, pp. 607-616, 2007.

[74] Z. X. Xie, A. G. Ming, Z. J. Li. “Adaptive robust trajectory and force tracking control of constrained mobile manipulators,” International Conference on Mechatronics and Automation. Harbin, pp.1351-1355, 2007.

[75] T. Okada, “Computer control of multijointed finger system for precise object handling,” International Trends in Manufacturing Technology- Robot Grippers, 1986.

[76] K. S. Salisbury and B. Roth, “Kinematics and force analysis of articulated mechanical hands,” Journal of Mechanims, Transmissions and Actuation in Design, 1983, vol.105, pp.35-41.

[77] S. C. Jacobsen, E. K. Iversen, D. F. Knutti , R. T. Johnson, and K. B. Biggers, “Design of the Utah/MIT dextrous hand,” in Proc. IEEE Int. Conf. Robotics and Automation, San Francisco, pp. 1520-1532, April 7-10, 1986.

[78] W. T. Townsend “MCB-Industrial robot feature article-Barrett Hand grasper,” in Industrial Robot: An International Journal, vol.27, no. 3, pp.181-188, 2000.

[79] Barret hand webpage, http://www.barretttechnology.com.

[80] J. P. Gazeau, S. Zeghlout, M. Arsicault, and J. P. Lallemand, “The LMS Hand: force and position controls in the aim of the fine manipulation of objects,” in Proc. IEEE International Conference on Robotics and Automation, pp. 2642-2648, 2001.

[81] Caffaz and G. Cannata, “The design and development of the DIST-Hand dextrous gripper,” in Proc. IEEE International Conference on Robotics and Automation, vol. 3, pp. 2075–2080, 1998.

[82] Dist hand webpage, http://www.graal.dist.unige.it/research.

[83] Robonaut hand webpage, http://vesuvius.jsc.nasa.gov.

[84] C. S. Lovchik and M. A. Diftler, “The robonaut hand: A dextrous robot hand for space,” in Proc. IEEE International Conference on Robotics and Automation, pp. 907–912, 1998.

[85] Yong Kwun Lee, Isao Shimoyama, “A skeletal framework artificial hand actuated by pneumatic artificial muscles,” in Proc. IEEE International Conference on Robotics and Automation, pp. 926-931, May 10-15, 1999.

[86] J. Butterfass, M. Grebenstein, H. Liu, G. Hirzinger, “DLR-Hand II: next generation of a dextrous robot hand,” in Proc. IEEE International Conference on Robotics and Automation, pp. 109-114, 2001.

[87] N. Fukaya, S. Toyama, T. Asfour, R. Dillmann, “Design of the TUAT/Karlsruhe humanoid hand,” in Proc. IEEE/RSJ International conference on intelligent robots and systems, pp.1754-1759, 2000.

[88] S. Schulz, C. Pylatiuk, G. Bretthauer, “A new ultralight anthropomorphic hand,” in Proc. IEEE International Conference on Robotics and Automation, vol. 3, pp. 2437-2441, 2001.

[89] Haruhisa Kawasaki, Hisayuki Shimomura, Yuuji Shimizu, “Educational-industrial complex development of an anthropomorphic robot hand ‘Gifu hand’”, Advanced Robotics, vol. 15, no. 3, pp. 357-363, 2001.

[90] Shadow hand webpage, http://www.shadow.org.uk/.

[91] G. Berselli, G. Borghesan, M. Brandi, C. Melchiorri, C. Natale, G. Palli, S. Pirozzi, and G. Vassura, “Integrated mechatronic design for a new generation of robotic hands,” in Proc. IFAC Symposium on Robot Control, Gifu, Japan, 2009.

[92] Matthew Mason and Kenneth Salisbuy, Robot Hand and the Mechanics of Manipulation, MIT Press, 1985.

[93] M. R. Cutkosky, Robotic grasping and fine manipulation, Kluwer Academic Publisher, 1985.

[94] D. J. Montana, “The kinematics of multi-fingered manipulation,” IEEE Trans. Robot. Automat., vol. 11, no. 4, pp. 491–503, 1995.

[95] J. K. Salisbury. B. Roth, “Kinematics and force analysis of articulated mechanical hands,” Journal of Mechanisms, Transmissions and Automation in Design, vol. 105, pp. 35-41, 1983.

[96] Suguru Arimoto, Pham Thuc Anh Nguyen etc. “Dynamics and control of a set of dual fingers with soft tips,” Robotica vol.18, pp. 71-80.2000.

[97] Joseph Chan, Yunhui Liu, “Dynamic simulation of multi-fingered robot hands based on a unified model,” Robotics and Autonomous Systems, vol.32, pp.185-201, 2000.

[98] Ze Yu and J. Gu. “A survey on real-time controlled multi-fingered robotic hand,” Proc. Canadian Conf. on Electrical & Computer Eng., pp. 975-980, May, 2008.

[99] M. R. Cutkosky, “On grasp choice, grasp models, and the design of hands for manufacturing tasks,” Robotics and Automation, IEEE Transactions on, vol. 5, no. 3, pp. 269–279, 1989.

[100] A. Okamura, N. Smaby, M. R. Cutkosky, “An overview of dexterous manipulation,” in Proc. International Conference on Robotics and Automation, pp. 255-262, 2000.

[101] H. Olsson, K. J. Aström, Canudas, M. Gäfvert, and P. Lischinsky, “Friction models and friction compensation,” European Journal of Control, vol. 4, no. 3, pp.176–195, 1998.

[102] Y. Nakamura, K. Nagai, and T. Yoshikawa, “Dynamics and stability in coordination of multiple robotic mechanisms,” International Journal of Robotics Research, vol. 8, no. 2, pp. 44–61, 1989.

[103] P. R. Sinha and J. M. Abel, “A contact stress model for multifingered grasps of rough objects,” IEEE Trans. Robotics and Automation, vol. 8, pp. 7–22, Feb. 1992.

[104] V. Kumar and K. Waldron, “Suboptimal algorithms for force distribution in multifingered grippers,” IEEE Trans. Robotics and Automation, vol. 5, pp. 252–257, Apr. 1989.

[105] J. Kerr and B. Roth, “Analysis of multifingered hands,” International Journal of Robotics Research, vol. 8, no. 4, pp. 3–17, 1986.

[106] C. A. Klein and S. Kittivatcharapong, “Optimization force distribution for the legs of a walking machine with friction cone constraints,” IEEE Trans. Robotics and Automation, vol. 6, pp. 73–85, Feb. 1990.

[107] F. T. Cheng and D. E. Orin, “Efficient algorithm for optimal force distribution—the compact dual LP method,” IEEE Trans. Robotics and Automation, vol. 6, pp. 178–187, Apr. 1990.

[108] Y. H. Liu, “Qualitative test and force optimization of 3-D frictional form-closure grasps using linear programming,” IEEE Trans. Robotics and Automation, vol. 15, pp. 163–173, Apr. 1999.

[109] L. Han, J. C. Trinkle, and Z. X. Li, “Grasp analysis as linear matrix inequality problems,” IEEE Trans. Robotics and Automation, vol. 16, pp. 663–674, Dec. 2000.

[110] M. Buss, H. Hashimoto, and J. B. Moore, “Dextrous hand grasping force optimization,” IEEE Trans. Robotics and Automation, vol. 12, pp. 406–417, June 1996.

[111] G. F. Liu, J. J. Xu, and Z. X. Li, “Automatic real-time grasping-force determination for multifingered manipulation: Theory and experiments,” in Proc. IROS, pp. 1676–1680, 2002.

[112] O. Khatib, “Reduced effective inertia in macro/mini-manipulator systems,” in Proc. of American Control Conference, pp. 2140-2147, Atlanta, GA, June 1988.

[113] K. Nagai and T. Yoshikawa, “Grasping and manipulation by arm/multifingered-hand mechanisms,” in Proc. IEEE International Conference on Robotics and Automation, pp. 1040-1047, May 21-27, Nagoya, Japan, 1995.

[114] H. K. Khalil, Nonlinear Systems, 3rd edition, Prentice Hall, Upper Saddle River, New Jersey, 2002.

[115] F. L. Lewis and S. Jagannathan, Neural Network Control of Robot Manipulators and Nonlinear Systems, London: Taylor & Francis, 1999.

[116] K. S. Narendraand and A. M. haswmy, “A new adaptive law for robust adaptation without persistent excitation,” IEEE Trans. Automatic Control, vol. AC-32, no. 2, pp. 134-145, 1987.

[117] P. Picton, Neural Networks, PALGRALE, 2000.

[118] R. S. Ball, A Treatise on the Theory of Screws, Cambridge University Press, Cambridge, UK, 1900.

[119] J. M. Selig, Geometrical Methods in Robotics, Monographs in Computer Sciences. Springer Verlag, New York, first edition, 1996.

[120] S. Stramigioli and H. Bruyninckx, “Geometry and screw theory for robotics,” In Proc. IEEE International Conference on Robotics and Automation, 2001.

[121] H. I. Christensen, “Intelligent home appliances. In Robotics Research,” R.A. Jarvis and A. Zelinsky, Eds., no. 6 in Springer Tracts in Advanced Robotics (STAR). Springer Verlag, Heidelberg, Germany, pp. 319-330, January 2003.

[122] M. Lohse, F. Hegel, and B. Wrede, “Domestic applications for social robots - an online survey on the influence of appearance and capabilities,” Journal of Physical Agents, vol. 2, pp. 21–32, 2008.

[123] T. W. Fong, I. Nourbakhsh, and K. Dautenhahn, “A survey of socially interactive robots: concepts, design, and applications,” Robotics and Autonomous Systems, vol. 42, no. 3–4, pp. 142–166, 2002.

[124] C. D. Wickens, S. E. Cordon & Y. Liu, An Introduction to Human Factors Engineering, Addison Wesley LONGMAN, 1998.

[125] Jacques Denavit and Richard S. Hartenberg, “A kinematic notation for lower-pair mechanisms based on matrices,” ASME Journal of Applied Mechanics, pages 215–221, June 1955.

[126] Mark W. Spong and M. Vidyasagar. Robot Dynamics and Control. John Wiley & Sons, Ltd, New York, 1989.

[127] J. J. Craig, Introduction to Robotics, Addison Wesley, second ed., 1989.

[128] J. Chung, S. Velinsky, “Robust control of a mobile manipulator-dynamic modelling approach,” Proceedings of the American Control Conference, pp. 2435-2439, San Diego, California, USA, 1999.

[129] Q. Yu, I. Chen, “A general approach to the dynamics of nonholonomic mobile manipulator systems,” ASME Trans. of Dynamic Systems, Measurement, and Control, vol. 124, no. 4, pp. 512-521, 2002.

[130] H. Tanner, K. Kyriakopoulos, “Mobile manipulator modelling with Kane’s approach,” Robotica, vol. 19, pp. 675-690, 2001.

[131] W. Jingguo, L. Yangmin, “Dynamic modeling of a mobile humanoid robot,” In IEEE Int. Conf. on Robotics and Biomimetics, Bangkok, Thailand, pp. 639-644, 2009.

[132] R. M. Murray, Z. X. Li and S. S. Sastry, A Mathematical Introduction to Robotic Manipulation, CRC Press, 2000 Corporate Blvd., N.W., Boca Raton, Florida 33431, ISBN 0-8493-7981-4, 1994.

[133] Antonino Lo Biondo, Giulio Mancuso, Analisi e controllo manipolazione robotica (Analysis and Control of Robotic Manipulation), Technical Report, University of Pisa.

[134] Mutku Mvengei and john Kihiu, “Bond graph modeling of mechanical dynamics of an Excavator for hydraulic system analysis and design,” International Journal of Aerospace and Mechanical Engineering, 3:4, 2009.

[135] Dresscher, Douwe and Brodskiy, Yury and Breedveld, Peter and Broenink, Jan and Stramigioli, Stefano, “Modeling of the youBot in a serial link structure using twists and wrenches in a bond graph,” In 2nd International Conference on Simulantion, Modeling, and Programming for Autonomous Robots, SIMPAR 2010 Workshops, 15-18 November 2010, Darmstadt, Germany.

[136] A. Vaz, and S. Hirai, “Bond graph modelling of a hand prosthesis during contact interaction,” Proceedings of the IASTED International Conference on Applied Simulation and Modelling, Marbella, Spain, September 3-5, pp. 313–318, 2003.

[137] A. Vaz and S. Hirai, “Modeling a hand prosthesis with Word Bond Graph Objects,” Proc. Int. Conf. on Integrated Modeling & Analysis in Applied Control & Automation, no. IMAACA_BG-08, Genoa, Italy, Oct. 28-31, 2004.

[138] M. V. Weghe, D. Ferguson, and S. Srinivasa, “Randomized path planning for redundant manipulators without inverse kinematics,” in IEEE-RAS International Conference on Humanoid Robots, November, 2007.

[139] D. Berenson, S. S. Srinivasa, D. Ferguson, and J. Kuffner, “Manipulation planning on constraint manifolds,” in IEEE International Conference on Robotics and Automation, ICRA, 2009.

[140] Kuipers, B. Jack, Quaternions and rotation sequences: a primer with applications to orbits, Aerospace, and Virtual. Reality. Princeton University Press, 1999.

[141] B. Xian, M. S. de Queiroz, D. M. Dawson, and L. Walker, “Task-space tracking control of redundant robot manipulators via quaternion feedback,” IEEE. Transactions on Robotics and Automation, vol. 20, no. 1, pp.160-167, Feb. 2004.

[142] J. S. C. Yuan, “Closed-loop manipulator control using quaternion feedback,” IEEE Journal of Robotics and Automation, vol. 4, no. 4, pp. 434–440, 1988.

[143] S. Quinlan, “Real-time modification of collision-free paths,” Ph.D. dissertation, Stanford University, 1994.

[144] A. Liegeois, “Automatic supervisory control of the configuration and behavior of multibody mechanisms,” IEEE Trans. Systems, Man, and Cybernetics, vol. 7, no. 12, pp. 868–871, 1977.

[145] J. Kuffner and S. LaValle, “RRT-Connect: An efficient approach to single-query path planning,” in IEEE International Conference on Robotics and Automation, 2000, pp. 995–1001.

[146] X. M. Tan, D. B. Zhao, J. Q. Yi, D. Xu, “Adaptive hybrid control for omnidirectional mobile manipulators using neural-network,” In Proceedings of American Control Conference, IEEE, Seattle, Washington, USA, pp. 5174-5179, 2008.

[147] S. S. Ge, C. C. Hang, and L. C. Woon, “Adaptive neural network control of robot manipulators in task space,” IEEE Transactions on Industrial Electronics, vol. 44, no. 6, pp. 746–752, Dec. 1997.

[148] T. Poggio and F. Girosi, “A theory of neural networks for approximation and learning,” Artificial Intelligence Lab., MIT, Cambridge, MA, Memo. 1140, July 1989.

[149] R. Fierro and F. L. Lewis, “Control of a nonholonomic mobile robot: backstepping kinematics into dynamics,” Journal of Robotic Systems, vol. 14, pp. 149-163, 1997.

[150] C. Desoer and M. Vidyasagar, Feedback Systems: Input-Output Properties. New York: Academic, 1975.

[151] F. L. Lewis, “Nonlinear network structures for feedback control,” Asian Journal of Control, vol. 4, pp. 205-228, 1999.

[152] N. H. MaClamroch and D. Wang, “Feedback stabilization and tracking of constrained robots,” IEEE Transactions on Automatic Control, vol. 33, pp. 419-426, 1988.

Résumé étendu en français

La robotique d'assistance à la personne est devenue, ces dernières années, un secteur très actif. L'idée est d'utiliser un robot afin d'assister une personne dans des taches ingrates, répétitives, difficile ou dangereuses. La Figure 1 donne quelques exemples de robots d'assistance.

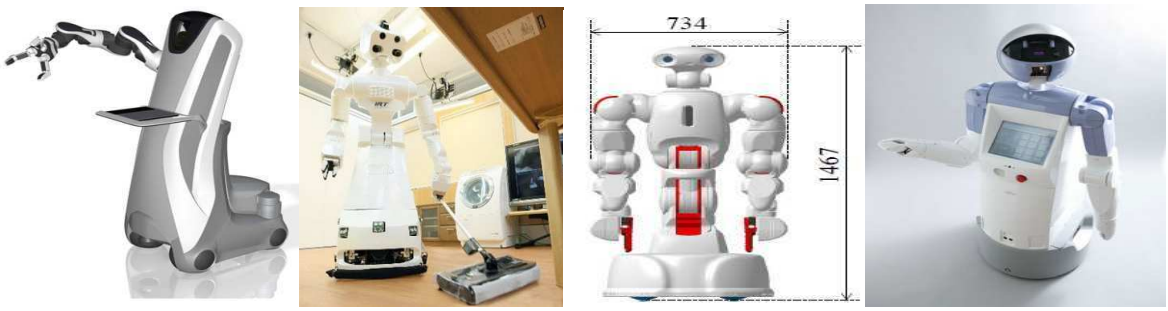


Figure 1: Exemples de robots assistance: Care-O-bot 3; Robot AR; TWENDY-ONE; Enon;

Le projet en cours poursuit le développement du prototype d'un robot d'assistance à la personne pour l'assistance dans des environnements domestiques. Il est une machine complexe qui exige des efforts particuliers pour le processus de conception. Cette thèse décrit la conception mécanique, la modélisation et la commande du manipulateur mobile à deux bras avec plusieurs doigts.

Le corps principal de la thèse est divisé en quatre parties.

Partie 1. État de l'art et préliminaires

Le chapitre 1 pose tout d'abord des principaux problèmes de recherche sur le robot d'assistance à la personne et puis introduit des recherches existantes pour les résolutions de problèmes. Ce chapitre aussi propose une brève introduction à la technique employée dans cette thèse en cinq sections. Dans la première section, certains préliminaires mathématiques à l'élaboration théorique dans les chapitres suivants sont d'abord présentés. Dans la deuxième section, l'outil bond graph est introduit. C'est une méthode graphique de modélisation. Ensuite, nous présentons et discutons l'architecture d'un réseau de neurones Radial Basis Function (RBF). Puis nous introduisons le concept de torseur. Dans la dernière section GL matrice et son opérateur sont présentés.

Partie 2. Conception et prototype virtuel

Dans le chapitre 2, nous comparons des robots d'assistance et puis les applications robotiques sont discutées. Le but ultime est de construire un robot autonome et capable d'exécuter des tâches complexes dans des environnements domestiques. Plusieurs fonctions sont détaillées. Et les exigences pour la réalisation de ces applications sont données.

Dans ce chapitre, selon les fonctions et les exigences décrites, nous concevons un robot d'assistance composé d'une plate-forme mobile à roues, deux bras et deux mains à 56 degrés de liberté. Dans la phase de conception mécanique, nous considérons les caractéristiques de l'architecture et la plage de rotation de chaque articulation. Ce robot est construit avec plusieurs chaînes répétées pour le tronc, les bras et les mains robotiques en utilisant le logiciel SolidWorks (Figure 2). Ensuite, la structure est exportée vers le logiciel ADAMS (Figure 3) pour effectuer des simulations de mouvements. Le prototype virtuel dans le système ADAMS fournit un outil pour explorer les nombreuses questions liées à la robotique d'assistance. Des simulations cinématiques et dynamiques du système robotique sont basées sur ADAMS.

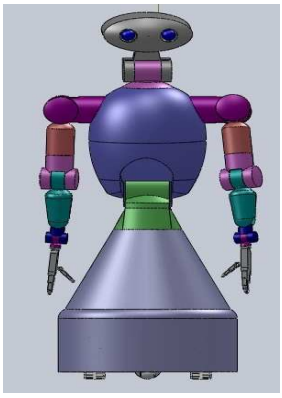


Figure 2: modèle sous Solidworks

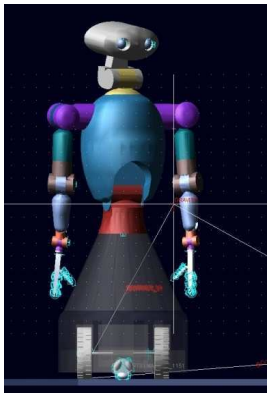


Figure 3: Prototype virtuel avec ADAMS

Partie 3. Modélisation du robot

Dans le chapitre 3, les modèles cinématiques et dynamiques du robot sont développés en utilisant respectivement le formalisme de Denavit-Hartenberg Modifié, et l'approche de Lagrange. L'approche bond graph, considéré comme la meilleure méthode graphique de modélisation des systèmes multi physiques, a été aussi utilisé pour modéliser le robot.

Partie 4. Planification et contrôle de mouvements

Le chapitre 4 développe de nouveaux algorithmes pour la planification et le contrôle de mouvements.

(a) Planification de mouvement

Le problème de planification de mouvement consiste à trouver un chemin sans collision pour un manipulateur avec la pose désirée (position et orientation). L'orientation est sous forme de quaternion. Un algorithme hybride qui combine la Pseudoinverse de la matrice Jacobienne et la méthode RRT (Rapidly-Exploring Random Tree RRT) est présenté.

(b) Commande de suivi de trajectoire par des réseaux de neurones

Le manipulateur mobile est un système non-linéaire complexe. Ainsi, le suivi de trajectoire dans l'espace articulaire est difficile à réaliser. Dans ce chapitre un contrôleur robuste basé sur les réseaux de neurones est introduit pour la commande coordonnée d'un manipulateur mobile. Cette méthode ne nécessite pas un modèle précis du manipulateur mobile. Les paramètres inconnus de la plate-forme mobile et le manipulateur sont identifiés et compensés en boucle fermée à l'aide de réseaux de neurones RBF (Radial Basis Function). Les erreurs causées par des perturbations peuvent être complètement éliminées par cette méthode.

(c) Commande force/position par des réseaux de neurones

Dans de nombreuses tâches le manipulateur mobile travaille en contact avec des surfaces, et une force d'interaction se développe entre l'effecteur et la surface. Par conséquent, le contrôle de force est aussi important que le contrôle de position. Un algorithme de contrôle similaire à celui de suivi de trajectoire est présenté pour la commande simultanée de la position et de la force d'un manipulateur mobile soumis à des contraintes holonomes et non holonomes en présence d'incertitudes et de perturbations. Nous prouvons la convergence des algorithmes proposés.

(d) Manipulation d'objets

Il existe peu de travaux relatifs à un système redondant compte tenu de la plate-forme mobile, les bras et les mains comme un système unique (comme ARMAR-III et Meka). La plupart des recherches ne tiennent pas compte de la coordination. Dans cette thèse, les modèles cinématique et dynamique du robot d'assistance à la personne manipulant un objet avec tes mains sont élaborés. Un algorithme de contrôle est présenté pour assurer une prise ferme, éviter les dérapages et bien suivre une trajectoire donnée à l'objet.

Conclusion

Dans cette thèse, nous avons proposé la conception mécanique d'un robot manipulateur mobile et utilisé une co-simulation ADAMS – Maltab/Simulink. Cela nous a permis de disposer d'un modèle virtuel très proche de la réalité, qui va servir comme outil de validation par la suite. Le modèle dynamique complet du robot a été est établi en utilisant les approches de Lagrange et bond graph. Un algorithme de planification de mouvement pour un manipulateur redondant combinant l'algorithme de Jacobi pseudoinverse et la méthode RRT a été proposé. En se basant sur le réseau RBF, deux commandes robustes ont été proposées. La première, pour contrôler le mouvement d'un manipulateur mobile non holonome en présence d'incertitudes et de perturbations. La seconde, pour le contrôle de la force/position d'un manipulateur mobile avec des contraintes non holonomes et holonomes. Compte tenu de la cinématique et de la dynamique d'une main robotique qui manipule un objet avec une forme connue, un système de commande est conçu pour assurer une prise ferme, éviter les dérapages et bien suivre un mouvement donné à l'objet. Cette méthode de contrôle est également applicable au système manipulateur mobile.

Conception et Commande d'un Robot d'Assistance à la Personne

Résumé: Ce travail s'inscrit dans le cadre de la conception et réalisation d'un robot d'assistance à la personne. Dans cette thèse, nous nous intéressons particulièrement à la conception, à la modélisation et à la commande d'un robot manipulateur mobile. La conception mécanique couplée à un outil de simulation dynamique multi-corps nous a permis d'obtenir un modèle virtuel très réaliste. Le modèle cinématique du système a été obtenu en utilisant la méthode D-H modifiée. L'approche Bond graph et la méthode de Lagrange ont permis de construire le modèle dynamique. Un algorithme hybride qui combine la pseudoinverse du jacobien et la méthode RRT a été proposé pour la planification de mouvement d'un manipulateur redondant et rechercher de configurations continues, stables et sans collision. Un contrôleur basé sur les réseaux de neurones a été introduit pour la commande coordonnée d'un manipulateur mobile. Cette méthode ne nécessite pas un modèle précis du robot. Les paramètres inconnus sont identifiés et compensés en utilisant des réseaux de neurones RBF. Un algorithme de contrôle similaire est présenté pour la commande force/position d'un manipulateur mobile qui est soumis à des contraintes holonomes et nonholonomes. L'étude de la main robotique a été effectuée séparément avant d'être couplée au reste du système. Les modèles cinématique et dynamique du système main-objet ont été obtenus en utilisant les approches mathématiques et bond graph. Un algorithme est proposé afin d'assurer une prise ferme, éviter les dérapages et suivre les mouvements désirés. Les validations des modèles et des différentes lois de commande ont été effectuées grâce à la co-simulation Matlab/modèle virtuel.

Mots-clefs: Robotique d'assistance à la personne; Prototype virtuel; Modélisation; Bond graph; Réseaux de neurones; Manipulation dextre; Simulation dynamique

Design and Control of a Personal Assistant Robot

Abstract: The purpose of this thesis is to design, model and control of a personal assistant robot used for domestic tasks. In order to make the robot's design more efficient, a virtual simulation system is built using dynamic simulation software. The kinematic model is set up based on modified D-H principle. The dynamic model is built using the Lagrange theorem and elaborated in Matlab. We also employ an energy-based approach for modeling and its bond graph notation ensures encapsulation of functionality, extendibility and reusability of each element of the model. A hybrid algorithm of combining the Jacobian pseudoinverse algorithm with Rapidly-Exploring Random Tree method is presented for collision-free path planning of a redundant manipulator. An intelligent robust controller based on neural network is introduced for the coordinated control of a mobile manipulator. This method does not require an accurate model of the robot. Unknown dynamic parameters of the mobile platform and the manipulator are identified and compensated in closed-loop control using RBF neural network. A similar control algorithm is presented for coordinated force/motion control of a mobile manipulator suffering both holonomic and nonholonomic constraints. Kinematics and dynamics of a dexterous hand manipulating an object with known shape by rolling contacts are derived. A computed torque control algorithm is presented to ensure firm grip, avoid slippage and well track a given motion imposed to the object. The validation of models and different control laws were made by the co-simulation Matlab / virtual model.

Keywords: Personal assistant robot; Virtual prototype; Modeling; Bond graph; Neural network; Dexterous manipulation; Dynamic simulation