CONTROL OF ROBOTIC HIGHWAY SAFETY MARKERS

by

Xiangrong Shen

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Master of Science

Major: Mechanical Engineering

Under the Supervision of Professor Shane M. Farritor

Lincoln, Nebraska

July, 2003

TABLE OF CONTENTS

1 INTRODUCTION…… 1

1.1 Motivation…… 1

1.2 Robotic Safety Barrel…… 2

1.3 Literature Review…… 4

1.3.1 Highway Robotics…… 4

1.3.2 Multi-Robot Motion Planning…… 5

1.3.3 Mobile Robot Control…… 7

1.4 Thesis Objective…… 8

1.5 Thesis Outline…… 8

2 SYSTEM DESIGN…… 10

2.1 System Configuration…… 10

2.2 Lead Robot…… 10

2.3 Barrel Robot…… 13

2.3.1 Design Objectives…… 13

2.3.2 Mechanical Design…… 15

2.3.3 Electrical Design…… 16

3 GLOBAL PLANNING AND CONTROL…… 17

3.1 Introduction…… 17

3.2 The Global Planning Problem…… 18

3.2.1 Problem Definition…… 18

3.2.2 Problem Decomposition…… 19

3.3 Sub-problem (1): the Path Planning Problem (PPP)…… 19

3.3.1 Problem Definition…… 19

3.3.2 Solution to PPP…… 20

3.4 Sub-problem (2): the Velocity Planning Problem (VPP)…… 23

3.4.1 Problem Definition…… 23

3.4.2 Virtual Coordination Space………………………………………….. 23
3.4.3 Collision Region………………………………………………………25
3.4.4 Invisible Region…………………………………………………… 25
3.4.5 Solution to VPP………………………………………………………26
3.5 ***Algorithm Overview***……………………………………………………31

4 LOCAL CONTROL……………………………………………….. 32
4.1 ***Introduction***……..…………………………………..…………………..32
4.2 ***Analysis of Local Control Problem***……………..……………………... 32
4.2.1 Kinematic Model………………..………………..………….. 32
4.2.2 Difficulties………………..………………..…………………35
4.2.3 Existing Solutions………………..………………..…………. 35
4.3 ***Control Law***……………..………………..………………………… 36
4.3.1 Overview………………..………………..……………………36
4.3.2 Local Trajectory Generation………………..……………………37
4.3.3 Dead-Reckoning………………..………………...……………. 44
4.3.4 Controller Development………………..……………………….. 48
4.4 ***Simulation***……………..………………..……………………………50
4.4.1 Dynamic Model………………..………………..……………. 50
4.4.2 Controller Parameter Tuning………………..…………………... 56

5 STABILITY ANALYSIS………………..………………………63
5.1 ***Global Stability Analysis***……………..……………………………… 63
5.2 ***Cartesian Stability Analysis***……………..…………………………… 64
5.2.1 Definition………………..………………..………………….. 64
5.2.2 Analysis with Lyapunov Second Method………………………..66
5.3 ***Actuator Stability Analysis***……………..…………………………… 72
5.3.1 Actuator Control Loop………………..………………………… 72
5.3.2 Analysis with Lyapunov Second Method………………………… 72

6 TEST RESULTS………………..……………………………… 76

6.1 Introduction……………………..……………..………………………… 76
6.2 Local Control Test……………..…………………………………….. 76
6.2.1 Dynamic Response…………………..…………………………….. 76
6.2.2 Repeated Experiments…………………..…………………………. 79
6.3 System Field Test………………..………………..……………………82

7 CONCLUSIONS AND FUTURE WORK……………….….. 85
7.1 Contributions of This Work………………..………………..………. 85
7.1.1 Robot Motion Planning…………………..…………………………85
7.1.2 Mobile Robot Control…………………..…………………………. 86
7.2 Future Work…………………..………………..………………………87

REFERENCES…………………..…………………..………………………..88

TABLE OF FIGURES

Figure 1-1: Robotic Safety Barrel………………………………………... 2

Figure 2-1: Lead Robot……………………………………………………..11

Figure 2-2: Laser Rangefinder………………………………………………12

Figure 2-3: Laser Image ……………………………………………………... 13

Figure 2-4: Barrel Robot…………………………………………………... 15

Figure 3-1: Global Paths…………………………………………………... 20

Figure 3-2: Coordination of Two Barrel Robots…………………………. 24

Figure 3-3: Virtual Coordination Space…………………………………… 24

Figure 3-4: Laser Image ……………………………………………………... 26

Figure 3-5(a): Cartesian Paths of the Robots…………………………... 28

Figure 3-5(b): Two Possible Paths in the Virtual Coordination Space...28

Figure 4-1: Schematic Representation of the Nonholonomic Vehicle….33

Figure 4-2: Inverse Jacobian Cartesian Controller …………………... 37

Figure 4-3: Local Path Generation………………………………………. 38

Figure 4-4: Time-Distance Relationship…………………………………… 40

Figure 4-5: Waypoint Generation……………………………………….. 43

Figure 4-6: Differential Robot Motion…………………………………….. 45

Figure 4-7: Original Control Scheme……………………………………… 48

Figure 4-8: Revised Control Scheme……………………………………... 49

Figure 4-9: Motor Driving Circuit………………………………………… 51

Figure 4-10: Shaft Dynamics……………………………………………….. 52

Figure 4-11: Frame Dynamics……………………………………………… 53

Figure 4-12: Block Diagram………………………………………………... 58

Figure 4-13: Simplified Block Diagram for θ_L/θ_{dL}………………………… 59

Figure 4-14: Simplified Block Diagram for θ_R/θ_{dL}………………………… 60

Figure 4-15: Root Locus for PID Controller Tuning …………………... 61

Figure 5-1: Global Control Loop…………………………………………… 63
Figure 5-2: New Coordinate system…………………………………………67
Figure 5-3: Actuator Control Loop………………………………………… 72
Figure 6-1(a): Dynamic Response of the Left Wheel……………………. 77
Figure 6-1(b): Dynamic Response of the Right Wheel…………………... 77
Figure 6-2: Desired and Actual Paths……………………………………… 78
Figure 6-3: Test Results of One Hundred Experiments………………… 79
Figure 6-4: Distribution of the Magnitude of Error…………………….. 81
Figure 6-5: Lane Closure with Five Barrel Robots……………………… 82
Figure 6-6: Desired and Actual Paths During Field Test……………….. 83

CONTROL OF ROBOTIC HIGHWAY SAFETY MARKERS

Xiangrong Shen, M.S.

University of Nebraska, 2003

Adviser: Shane M. Farritor

Proper traffic control is critical in highway work zone safety. Traffic control devices such as signs, barricades, cones, and safety barrels are used. Automated safety devices could improve work zone design and housekeeping and therefore increase safety.

This thesis presents a mobile safety barrel robot. The robotic safety barrels can self-deploy and self-retrieve — removing workers from this dangerous task. The robots move independently so they can be deployed in parallel and can quickly reconfigure as the work zone changes. The system must be reliable and have a low per-robot cost.

A six robot system, which consists of a lead robot and five low cost barrel robots, is described. A hierarchical planning and control approach is presented, in which control is broken into two levels — the global planning and control, and the local control, with the purpose of reducing per-robot cost. The global planning and control is performed with one central sophisticated mobile robot that localizes each barrel, plans its path, sends out commands, and monitors the performances. The local control, as the lower level, is distributed to each barrel robot that does not have knowledge of other robots, but only performs the local tasks. With this control system applied, individual barrel robot does not need high computational ability or expensive localization and hazard avoidance sensing. Test results are also presented including a statistical analysis of the performance of the local robot controllers and field tests of the full system.

CHAPTER 1: INTRODUCTION

1.1 Motivation

The safety of highway construction and maintenance workers is an important issue. For every billion dollars spent on road construction, more than 33 people die in accidents related to that work (Anon, 1990). The nation's highway system is aging and is in need of constant repair. Increased maintenance and construction will result in more injuries. New technologies may be useful in reducing work zone accidents.

Work zone housekeeping may be the most important element in reducing accidents (Anon, 1990). Housekeeping involves tasks such as covering and uncovering signs and moving markers (barrels, cones, signs) as the work progresses. While housekeeping is important, it comes at a very high cost. Worker time is required to set up and tear down hundreds of construction zone safety devices. Safety markers often need to be deployed and retrieved at the beginning and end of each workday. Removing the markers or covering them is time-consuming, but leaving them in open view around the clock adds to unsafe conditions. Safety markers that are simply moved to the side of the road remain highly visible, and this constant presence leads to driver complacency. In addition, deployment and retrieval activities are extremely dangerous since the construction workers are very close to passing motorists. Aware of this, implementing a "smart" work zone, which consists of automated traffic control devices, is one method to improve safety.

1.2 Robotic Safety Barrel

This thesis presents the design of a control system for a mobile Robotic Safety Barrel (RSB), Figure 1-1. Safety barrels guide traffic and serve as a visible barrier between traffic and work crews. These barrels consist of a brightly colored plastic drum (approximately 130cm high and 50cm in diameter) that is attached to a heavy base. Often, hundreds of barrels are manually placed in a typical work zone. The robotic safety barrel replaces the heavy base of a traditional safety barrel with a mobile robot. The mobile robot transports the safety barrel, and robots work in teams to provide traffic control.

Figure 1-1: Robotic Safety Barrel (RSB)

Independent, autonomous barrel motion has several advantages. First, the barrels can self-deploy and self-retrieve, eliminating the dangerous task of manually placing barrels in busy traffic. Second, the barrel positions can be quickly and remotely re-configured as the work zone changes. Barrels could continuously follow work crews to maintain proper placement for safety. The robotic safety barrels are the first elements of a team of Robotic Safety Markers (RSMs) that includes signs, cones, and possibly barricades and arrestors.

This approach clearly has a higher equipment cost than traditional systems; however, there are possible cost reductions in labor and increased worker safety. The approach is not fitted for every work zone but there are many situations where it would be useful and practical. For example, temporary lane closures for tasks such as crack sealing, pothole repair, utility work, and incident management are very common in urban settings (Traynowichz, 2002). These work zones frequently change as work progresses (e.g. move to the next pothole). This example is used as a case study and a lane closure is demonstrated in Chapter 6.3.

Robotic safety markers must be reliable and have a low per-robot cost. Reliability is important because a robot that malfunctions could create a significant hazard. Low cost is important because multiple safety markers are used and barrels are often struck by vehicles. The robots are designed so that each is simple, reliable, and inexpensive.

The design of a six robot system that consists of a lead robot and five low-cost barrel robots is developed. A novel distributed planning and control approach is presented that reduces the per-robot cost by centralizing the intelligence and sensing while keeping communication bandwidth low by distributing local control. Test results are presented including a statistical analysis of the local controller and field tests of the full system.

1.3 Literature Review

1.3.1 Highway Robotics

There is much research investigating “smart” highways, such as the Intelligent Vehicle Highway Systems (IVHS), that use technology to provide information to travelers, improve traffic control and congestion, and increase the efficiency of commercial vehicle and transit operations [e.g. Ravani et al, 1994]. Several projects have applied robotic technologies to highway construction equipment. One project created a remotely driven dump truck that is used as a barrier by shadowing a vehicle during slow-moving maintenance operations (Shah, 1993). Other projects developed an automated paving machine (White, 1995) and a robotic paint removal system (Woo, 1995).

The Advanced Highway Maintenance and Construction Technology (AHMCT) Research Center at the University of California, Davis has developed several automated highway maintenance systems including a laser-guided lane stripe painting machine (West et al, 1995), a tethered robot team for roadway crack sealing (Lasky, et al., 2000), and a flying tele-operated aerial bridge-inspection system (Woo, 1995).

There has also been research in the automated placement of safety devices. A tele-robotic system has been developed to apply raised, reflective pavement markers (Rihani and Bernold, 1996). Two other systems automate the placement and retrieval of rows of warning cones (Zhou and West, 1991; AHMCT, 2002). The systems are modified maintenance vehicles (pickups) that place cones as the vehicle travels down the road.

These previous systems are fundamentally different in that markers are placed serially. One system is used to deploy/retrieve the markers.

The advantage of serial deployment is that the markers themselves do not need to be modified, and the cost and complexity of the overall system is not related to the number of markers deployed. The disadvantages are that the deployment time grows with the number of markers, and the markers cannot be reconfigured to adapt to a changing work zone. The proposed approach makes each marker a robot. The markers can be moved/deployed/retrieved in parallel and can adapt to changes in the work zone. The disadvantage is that the cost of the overall system scales with the number of markers used.

Both approaches have advantages in different situations. The single robot that deploys numerous barrels is better where many seldom re-configured barrels are used (e.g. bridge construction). The proposed approach, where each barrel is a robot, would be useful where fewer barrels are used and are often moved (e.g. patch a pothole every 100 meters).

1.3.2 Multi-Robot Motion Planning

Several techniques have been developed for multi-robot motion planning, an overview can be found in (Todt et al., 2000). One approach developed a hybrid control scheme for formations of non-holonomic mobile robots (Fierro, et al., 2001; Desai et al., 1998). This method controls the motion of many robots, or agents, while maintaining a predetermined formation. Each robot senses others and follows a leader. This work uses high bandwidth communication and complex vision sensors on each robot. Other approaches to formation control have focused on dynamic programming (Gentili and

Martinelli, 2000), obstacle avoidance (Kang et al., 2000), trajectory generation (Belta and Kumar, 2001), and model-based coordination (Egerstedt and Hu, 2001).

A theoretical study of a fully distributed (i.e. no central controller or inter-agent communication) approach to multi-robot control has been proposed that is based on praxeic utility theory (Gold et al, 2000). This work implemented a new decision-making paradigm based on an epistemic model of how people make decisions rather than the classical paradigm based on optimization. Another decentralized approach has investigated the effects of communication rates between swarms of distributed autonomous robots (Stilwell and Bishop, 2000). The observer-based methods were developed in this study, and the complex agent-level dynamics was not utilized in the controller development. In both studies the decentralized approach is possible because it is assumed that each robot has some environmental sensing. Methods to evaluate distributed systems have also been proposed (Ye, et al, 2000).

A novel hierarchical control system is described here where control is broken into two levels — global and local control. This reduces the per-robot cost by centralizing sensing and intelligence. Local control is distributed; therefore individual barrels do not need high computational ability or expensive localization and hazard avoidance sensing. Also, only low-rate communication (0.2 Hz) with the central controller is required so expensive transceivers are eliminated. This novel approach is analogous to a shepherd/sheep or general/troops approach to system behavior. The proposed approach has unique aspects including the incorporation of constraints that explicitly consider the limitations of centralized sensing and intelligence.

1.3.3 Mobile Robot Control

There are different forms of mobile robot, according to the classification of steering and powering of the vehicle. An overall study of the mobile robot kinematics can be found in (Wang, et al, 1996). A certain type of mobile robot with two steerable wheels was modeled and a feedback controller was developed to move the robot from any initial configuration (position and orientation) to any final configuration (Yun, et al, 1996). A unicycle-type mobile robot was analyzed and an observer based kinematic tracking controller was developed for global trajectory tracking (Lefeber, et al, 2001).

The differential drive robot, such as the barrel robot developed in this work, is perhaps the simplest type of mobile robot. A simple bounded velocity model was used to develop the time optimal trajectories (Balkcom, et al, 2000). A piecewise continuous pure feedback law was developed (Sordalen, et al, 1992), and the controller is stable with regard to the arbitrary point in the state space. Both of the studies did not consider the dynamics of the robot, however. The backstepping approach is another method for the mobile robot control; it developed a multilevel hierarchical controller to control the relative position between a moving target and an autonomous vehicle (Palopoli, et al, 2000).

In this work, an inverse Jacobian Cartesian control scheme is developed to control the motion of the barrel robot. The error in the Cartesian space is converted to joint space by the inverse Jacobian method, or Dead Reckoning, and then this error is acted upon by the tradition PID controller to create torque commands for the driving motors. The proposed scheme has the advantage of being simple, reliable and no high computational capability needed.

1.4 Thesis Objective

The control of the highway robotic safety markers is studied in this thesis. A hierarchical control system is developed, in which control is broken into two levels: 1) global planning and control, and 2) local control. For each of these levels, control algorithm is developed for the corresponding objective.

For the global planning and control, the problem is to move the group of robots from the initial configuration to the final configuration. According to the requirements of the application, the main constraints considered are keeping the robots motion within the work area, and avoiding collisions between the robots. To solve this problem, different motion planning algorithms are compared, and a decoupled approach is chosen due to the advantage of reduced computational complexity and easiness of implementation. The objective of the global planning and control part is planning the motion to minimize the total time interval to complete the motion.

For the local control, the control of the nonholonomic differential drive mobile robot is studied, according to the design of the barrel robot. The important assumption of no slip between the ground and the wheels is made, and the kinematics of the robot is constructed based on this assumption. The objective of the local controller is to move the barrel robot to its desired waypoint independently with an acceptable precision, and an inverse Jacobian Cartesian controller is developed for this purpose.

1.5 Thesis Outline

This thesis is divided into seven chapters. This chapter serves as an introduction and overview of the work. Chapter 2 presents the system design. The configuration of the system is presented. The designs of the lead robot and barrel robot are analyzed in this chapter.

Chapter 3 gives a description of the global planning and control scheme. The problem is defined, and then it is decomposed into two sub-problems: 1) a Path Planning Problem, and 2) a Velocity Planning Problem. The global paths of parabolas are used to solve the Path Planning Problem, and a Tangent Graph algorithm is used to solve the Velocity Planning Problem.

Chapter 4 presents the local controller development of the barrel robot. The kinematics of the barrel robot and the difficulty of the control of nonholonomic mobile robot are analyzed. A novel inverse Jacobian Cartesian controller is developed thereafter. Chapter 5 analyzes the stability of the controller according to the Lyapunov stability theory. The Cartesian stability and actuator stability are proven.

Chapter 6 gives the test results of local controller experiments and a ground test. The dynamic response of the local control was tested as the first step, and then the repetitive accuracy of the local controller is tested with a hundred repeated experiments. A statistical analysis is given based on the result. After that, the result of the ground test is given, and the performance of the control system is evaluated.

Chapter 7 outlines the general conclusions regarding the control of robotic safety markers. Some suggestions of further work are given.

CHAPTER 2: SYSTEM DESIGN

2.1 System Configuration

The design of a six robot system that consists of a lead robot and five low-cost barrel robots is proposed. This configuration is called in this work a Shepherd-Sheep concept. The lead robot serves as the ‘shepherd’: it is equipped with high speed processors, a variety of sensors and communication equipments as well as a human-machine interface, thus it could receive the operator’s instructions, plan the motion of the whole team and send out commands to the barrel robots. The barrel robots, like sheep, do have some local control abilities.

2.2 Lead Robot

In the multi-robot system described in this thesis, an autonomous mobile robot is designed as the lead robot. The robot, called *Companion*, is a 4-wheel electric drive system with automobile-type steering, and its modes of operation span from direct human teleoperation to fully autonomous behavior (Figure 2-2).

Figure 2-1: Lead Robot

Its mobile platform consists of a modified electric wheelchair. The powered chair is different than some powered wheelchairs in that locomotion and steering are done with separate pairs of wheels instead of one pair doing both. The platform stores energy with two deep-cycle lead-acid battery at 24 volts, with a total energy capacity of 1.5 kW-hr. Actuator motors include a high torque brushed DC motor for drive, along with a 2-speed transmission, and a smaller DC motor with a worm and lead screw assembly for automobile-type (Ackermann) steering. A turning radius of just over two feet allows for very tight maneuvers, which is very helpful for most proposed applications of the system.

The sensors mounted on *Companion* include bumpers, proximity detectors, sonars, gyro, video and a laser rangefinder. The laser rangefinder, LMS 200 made by

SICK AG, Germany, is the main sensor for the localization of barrel robots (Figure 2-2). For the current application of moving the barrel robots from the initial positions to the desired final positions, both of which are within the range of the laser rangefinder, the rangefinder is placed on a fixed spot. For the mobile applications in the future, it will be mounted to the lead robot to find the relative position between the lead robot and barrel robots.

Figure 2-2: Laser Rangefinder

Figure 2-3 is an image scanned by the rangefinder. The rangefinder scans the area of 180˚ two-dimensionally, and the barrel robot positions can be calculated with a special computer algorithm. The typical range of the rangefinder is 32 meters, which meets the requirements of most applications.

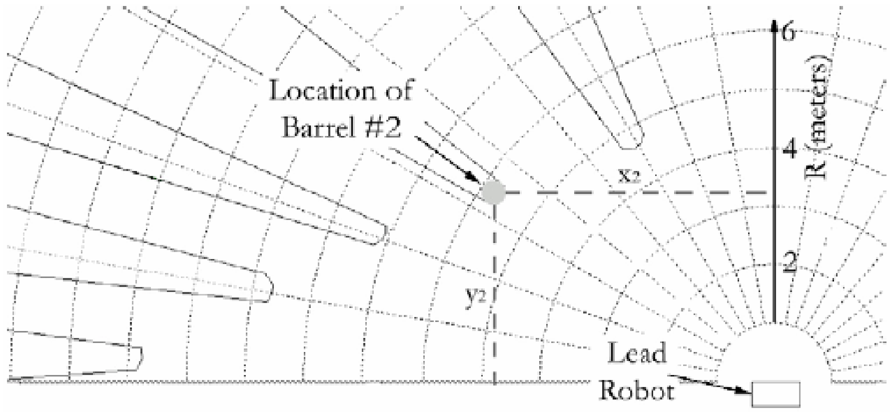

Figure 2-3: Laser Image

2.3 Barrel Robot

2.3.1 Design Objectives

There are two system-level design requirements for the barrel robot: 1) high reliability and 2) low per-robot cost. These constraints are specifically considered in the robot physical design described below and the design of the control and planning system in Chapter 4.

The system must be reliable in hardware and software. There is extensive work in fault tolerant robots and robots are now being used in critical operations such as surgery. Here, an encoded Radio Frequency (RF) "heartbeat" is broadcast locally so all robots can be stopped if a malfunction is detected. A loss of this RF heartbeat causes motor power to be cut through redundant fail-safe electrical relays. Also, each robot position is monitored by the individual robot and by the lead robot. Discrepancies in these redundant sensors (and processors) will cause either a local (robot) or global (heartbeat) shutdown. All communication between the lead and barrels must be successfully verified to prevent erroneous commands. Control system reliability and accuracy have been statistically quantified in Chapter 6.

Cost per robot is critical because multiple (often >100) markers are used and barrels are often destroyed. Robots struck by cars have performed their function — indicating that the driver is not in the correct position — but designing a robot to survive an impact would be difficult and make the robot more expensive.

There are also specific constraints to be considered. The robot and barrel will remain stationary in 96 km/h (60 mph) winds often caused by passing traffic. The robots weigh <12 kg so they can easily be moved by workers. The prototypes have the same footprint as traditional barrels so current transport methods can be used. The device also cannot become a safety hazard if an impact occurs. The robot needs to climb slopes < 7% grade and travel at 8 km/h (5 mph) (the speed of some maintenance operations e.g. striping). The robot needs to traverse small (<8cm) obstructions, bumps, and depressions. It must operate in inclement weather and maneuver within the work zone.

2.3.2 Mechanical Design

Figure 2-4: Barrel Robot

Mechanically, the robot consists of a platform that is similar to the base of a traditional safety barrel, as shown in Figure 2-2. The robot has two 20 cm diameter wheels that are independently driven by two motors. The rear of the robot has a passive caster. This kinematic arrangement allows the robot to turn on any radius — including turning in place. The motors have a high gear ratio so the robot is not easily moved when the motors are not powered. This allows the robot to stay in place for long periods while using little or no power. Since both the lead wheels are powered, the robot can climb small bumps and obstacles. Each wheel has an encoder so the robot can determine its position over short distances (statistically quantified in Chapter 6). The barrel snaps onto a ring around the robot and encloses all components. The robot itself stands less than 30 cm tall and raises the barrel height by 7 cm.

2.3.3 Electrical Design

Electrically, the robot is powered by a 12-volt lead acid battery — allowing it to operate continuously for approximately 20 hours. The barrel robot has three 8-bit processors: one is used for robot control (RS2000 Rabbit Semiconductor TM), one to manage communication with the outside world (PIC), and one to be a watchdog for the RF heartbeat (PIC). Each robot has an RF transceiver that can transmit and receive low bandwidth information.

CHAPTER 3: GLOBAL PLANNING AND CONTROL

3.1 Introduction

A hierarchical control system is developed in this work. Control is broken into two levels — the global planning and control, and the local control, with the purpose of reducing per-robot cost. The global planning and control is performed with one central sophisticated mobile robot that localizes each barrel, plans its path, sends out commands, and monitors the performances. The local control, as the lower level, is distributed to each barrel robot that does not have knowledge of other robots, but only performs the local tasks. With this control system, the individual barrel robot does not need high computational ability or expensive localization and hazard avoidance sensing. Moreover, only a low bandwidth communication is needed to communicate with the central controller, so expensive transceivers are also avoided.

Motion planning for mobile robots has been extensively studied. Several approaches have been developed such as behavior control (Brooks, 1986), artificial potential fields (Khatib, 1986), probabilistic maps (Borenstein and Koren, 1991), genetic planning (e.g. Farritor et al., 1998), and others. For the area of multiple robot coordination, a comprehensive analysis and classification has been developed by (Todt E. et al, 2000). In this work, the nomenclature is unified and the basic tools used in each method are listed and classified.

A decoupled method is used in the global planning of the control system. The Path – Velocity decomposition is an efficient planning method to solve the global planning problem (Kant, et al, 1986). The geometric paths and velocity profiles are planned separately, i.e. the coordination phase is separated from the path-planning phase. In this approach, the problem of global planning is decomposed into two sub problems: the Path Planning Problem (PPP) and the Velocity Planning Problem (VPP). Then different algorithms are developed to solve these problems consequently.

3.2 The Global Planning Problem

3.2.1 Problem Definition

The task in the global planning is to move the group of barrel robots from the initial configuration to the desired configuration with the following goals:

1) Minimize the time to complete the motion;
2) Easy implementation with reasonable computation complexity;

And, under the following constraints:

1) Maintain a minimum distance between barrel robots (for collision avoidance);
2) Travel slower than the robot's maximum velocity;
3) Keep all the barrel robots in the specified work area (i.e. don't wander into traffic).

The global planning can be taken as a trajectory-planning problem. Trajectory is defined as the geometric path plus a velocity associated with each configuration. The reason to include the geometric path and the corresponding velocity is that both of them are crucial for collision avoidance, and thus both should be considered in the solution.

3.2.2 Problem Decomposition

There are two different approaches with regard to the relationship between the geometric path planning and velocity planning: coupled and decoupled. The geometric path and the velocity profile are determined at the same time in the coupled methods, while the converse is true in decoupled methods.

In this work, a decoupled method is used for the advantage of reducing the computational complexity. The Trajectory-Planning problem is decomposed heuristically into two sub-problems:

1) Path Planning Problem: planning the paths to avoid collision with static obstacles and keep the robot motion within the work area;
2) Velocity Planning Problem: planning the velocity along the path to avoid collisions with moving obstacles.

3.3 Sub-problem (1): the Path Planning Problem (PPP)

3.3.1 Problem Definition

The objective of Path Planning Problem is to generate global paths that barrel robots will take. To be feasible, a geometric path should be

1) Without striking static obstacles;

2) Within the specified work area.

The second requirement comes from the operational conditions: a barrel robot moving out of the work area may either put much hazard to the traffic or have difficulty in traversing the roadway shoulder.

There are also other considerations in the path planning. The nonholonomic constraint of the mobile robot requires the path to be smooth, and the initial direction of the path is preferred to coincide with the initial orientation of the robot to avoid turning in place. Moreover, simple representation of the path is also preferred to reduce computational complexity.

3.3.2 Solution to PPP

The planning problem in this work is somewhat unique in that the area to be traversed, a roadway, is relatively well known, flat and obstacle free. The primary obstacles for the barrel robots are other robots. The roadway may have identifiable areas that cannot be traversed, such as potholes or a sharp drop off at the pavements edge, but these areas can be designated by the operator and easily avoided by the planner.

A polynomial is used to create the desired paths. For the path of each barrel robot, there are three boundary conditions: initial position, initial orientation, and final position. Considering these boundary conditions, second order polynomials, or parabolas, are used.

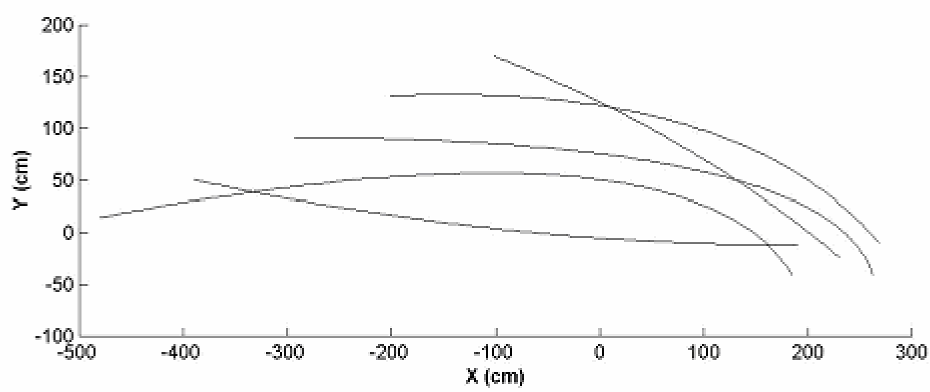

Figure 3-1: Global Paths

This approach ensures the nonholonomic constraints are considered. The final orientation of the path is not constrained so a new destination can be given while maintaining smooth motion (the path always begins with locally forward motion).

The procedures of finding the equations of parabolas are as follows. First, construct a local coordinate system $\{L\}$ with the origin at the robot's initial position, and the positive x- direction coinciding with the initial orientation. In the frame $\{L\}$, the parabola starts at the origin with the Y axis as its axis of symmetry. So the path representation has the simple form of

$$Y' = cX'^2$$

To obtain the coefficient c, the coordinates of the final position P_f in the frame {L} are calculated as:

$$X_f' = (X_f - X_s) * \cos(\Psi_s) + (Y_f - Y_s) * \sin(\Psi_s) \tag{3-1}$$

$$Y_f' = (X_f - X_s) * \sin(-\Psi_s) + (Y_f - Y_s) * \cos(\Psi_s) \tag{3-2}$$

Thus the coefficient c can be calculated:

$$c = Y_f' / (X_f'^2) \tag{3-3}$$

Now the path in the frame {L} can be written as:

$$Y' = cX'^2 \qquad 0 \le X' \le X_f' \text{ (for } X_f'<0\text{) or } 0 \ge X' \ge X_f' \text{ (for } X_f'>0\text{)} \tag{3-4}$$

To obtain the path representation in the global frame, the following coordinate transformation is used:

$$X' = (X - X_s) * \cos(\Psi_s) + (Y - Y_s) * \sin(\Psi_s) \tag{3-5}$$

$$Y' = (X - X_s) * \sin(-\Psi_s) + (Y - Y_s) * \cos(\Psi_s) \tag{3-6}$$

Substitute the Eqns (3-5) and (3-6) into (3-4):

$$(X - X_s) * \sin(-\Psi_s) + (Y - Y_s) * \cos(\Psi_s) = c * [(X - X_s) * \cos(\Psi_s) + (Y - Y_s) * \sin(\Psi_s)]^2 \tag{3-7}$$

After simplification, the path representation in the global frame can be derived as:

$$A * X^2 + B * Y^2 + C * XY + D * X + E * Y + F = 0 \tag{3-8}$$

where

$$A = c * \cos^2 \Psi_s$$

$$B = c * \sin^2 \Psi_s$$

$$C = 2c * \sin \Psi_s * \cos \Psi_s$$

$$D = \sin \Psi_s - 2c * \cos^2 \Psi_s * X_s - 2c * \sin \Psi_s \cos \Psi_s * Y_s$$

$$E = -\cos \Psi_s - 2c * \sin^2 \Psi_s * Y_s - 2c * \sin \Psi_s \cos \Psi_s * X_s$$

$$F = -\sin \Psi_s X_s + \cos \Psi_s Y_s + c \cos^2 \Psi_s X_s^2 + c \sin^2 \Psi_s Y_s^2 + 2c \sin \Psi_s \cos \Psi_s X_s Y_s$$

However, it is also possible that no solution exists. There are two such situations:

1) $X_f' = 0$. That means an infinite value of parabola coefficient c, thus no parabola is valid for this situation.
2) Part of the path traverses the boundary of the work area. This situation is also unacceptable.

To overcome this difficulty, a simple solution is used: turning the barrel robot in place to obtain a new initial orientation. Although a discontinuous motion at the beginning will result, the reduced computation and the continuous motion afterwards make it a feasible solution.

3.4 Sub-problem (2): the Velocity Planning Problem (VPP)

3.4.1 Problem Definition

With the path planned in the Path Planning Problem, the task of the velocity planning is to find the velocity along the path to avoid collision with moving obstacles. As is mentioned before, the area to be traversed is relatively obstacle free, so the main moving obstacles for each individual barrel robot are the other barrel robots in the same work area. Time to complete the motion is used as the criterion to judge the performance of the velocity function. The time to complete the motion is minimized. The main constraint considered in this problem is the maximal linear velocity of the barrel robot. This is limited by the physical saturation of the robot's drive motors.

The Velocity Planning Problem requires determining the velocities of the barrel robots along the paths to avoid collision with other barrel robots, with the target of minimizing the time to complete the motion, under the constraint of the maximum linear velocity of the barrel robot.

3.4.2 Virtual Coordination Space

Treated as a rigid body in the two dimensional space, the barrel robot has 3 degrees of freedom: x, y and ψ, where (x, y) is the global coordinates of the barrel robot center, and ψ is the orientation. Since the path has been defined in the global space, and the orientation of the barrel robot is assumed to be tangent to the path, therefore, there is only one variable needed to determine the global position of the barrel robot.

As shown in Figure 3-2, the global position (x, y) can be written as a function of the distance along the path s. Thus the velocity-planning problem can be expressed in the terms of new planning variable s.

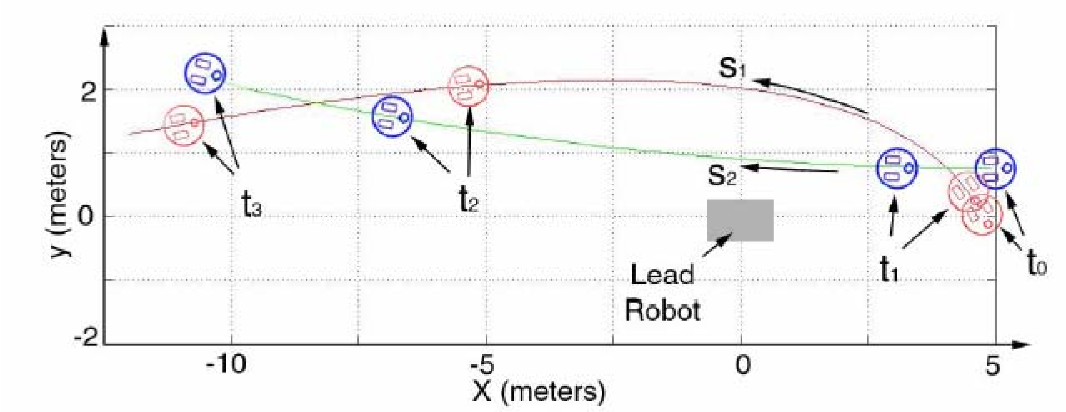

Figure 3-2: Coordination of Two Barrel Robots

Now consider the collision avoidance problem between barrel robots. For every two barrel robots, a Virtual Coordination Space can be constructed as Figure 3-3, and by planning an obstacle free path in this space the relative motion of barrel robots can be determined, thus collisions can be avoided.

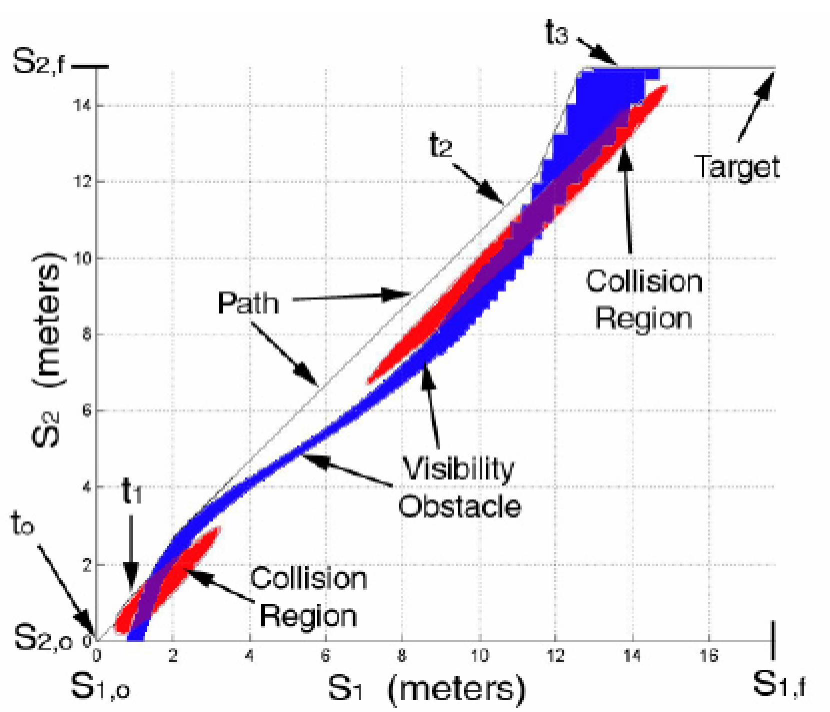

Figure 3-3: Virtual Coordination Space

3.4.3 Collision Region

In the Virtual Coordinate Space, the collision region is the main obstacle to avoid, as shown as the elliptical shapes in Figure 3-3. The definition of the collision region is the regions where the distance of the centers of the barrel robots are below a specified safety threshold such that the barrel robots may collide with each other. This safety threshold is set bigger than the diameter of the barrel robot to allow the errors in localizing and control.

3.4.4 Invisible Region

The other obstacle in the Virtual Coordination Space is the invisible region, or visibility obstacle. The reason to introduce the invisible region is to ensure light-of-sight contact between the lead robot and the barrel robots, so that the lead robot can always monitor the performance of each barrel robot.

The lead robot uses a laser sensor to scan the work area, and Figure 3-4 is an example of the image generated by the sensor. The sensor has a finite angular resolution, and a certain number of beams should be projected to the barrel robot to find its global position. This is the reason why the invisible region has a stepped boundary.

Usually the invisible regions are impassable, due to the need of continuous localization. There is one exception, though — if the invisible barrel robot is stationary, there is no need to localize it, and thus the invisible region can be traversed.

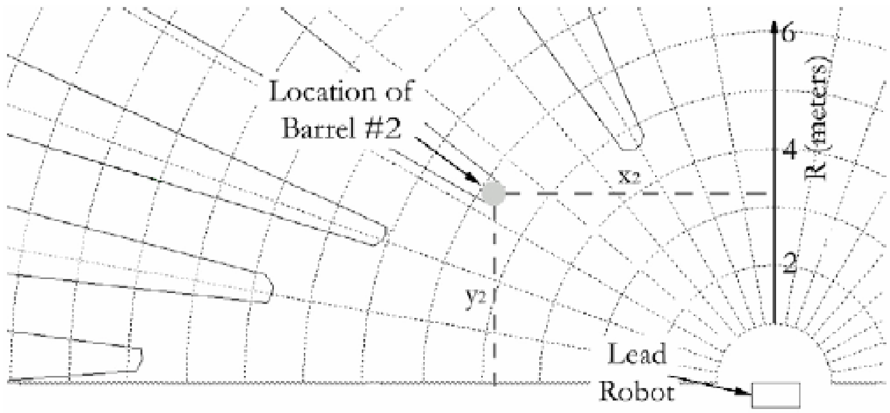

Figure 3-4: Laser Image

3.4.5 Solution to VPP

With the Virtual Coordination Space constructed and the obstacles plotted, feasible paths can now be planned from the origin to the goal in the coordination space, and thus the time-distance relationships can be computed accordingly. In order to generate such a path while avoiding the obstacles, the Tangent Graph (T-Graph) method was used here.

The Tangent Graph method is a powerful graphic method used in the mobile robot path planning (Liu, et al, 1991, 1994). Basically this method finds the collision free paths for the mobile robot in the cluttered area, and by searching the paths composed of the locally shortest path segments, the globally shortest path can be found.

There is an important difference between the path planning in the global space and that in the coordination space. For the global space path planning, the path with the shortest length is desired, while the path with the shortest time to complete the motion is desired for the coordination space path planning. The temporal cost is not explicit in the coordination space, while it could be computed with the actual path in the space and the kinematic constraints.

With respect to temporal cost, one important conclusion with the path is:

The tangent of the path is the ratio between the velocities of the two robots.

Proof:

Suppose the positions of the two robots at time point t is $S_1(t)$ and $S_2(t)$, respectively. After a period of dt, their positions changed to $S_1(t+dt)$ and $S_2(t+dt)$. In the coordination space, they move from point ($S_1(t)$, $S_2(t)$) to point ($S_1(t+dt)$, $S_2(t+dt)$) along the path. Thus the tangent of the path is

$$\tan\theta = \lim_{dt\to 0}\frac{S_2(t+dt)-S_2(t)}{S_1(t+dt)-S_1(t)}$$

Divide the numerator and denominator of the right side with dt, we get

$$\tan\theta = \lim_{dt\to 0}\frac{[\frac{S_2(t+dt)-S_2(t)}{dt}]}{[\frac{S_1(t+dt)-S_1(t)}{dt}]}$$

The velocities of the robots are

$$V_1(t) = \lim_{dt\to 0}\frac{S_1(t+dt)-S_1(t)}{dt}$$

and

$$V_2(t) = \lim_{dt\to 0}\frac{S_2(t+dt)-S_2(t)}{dt}$$

Obviously

$$\tan\theta = \frac{V_2(t)}{V_1(t)} \tag{3-9}$$

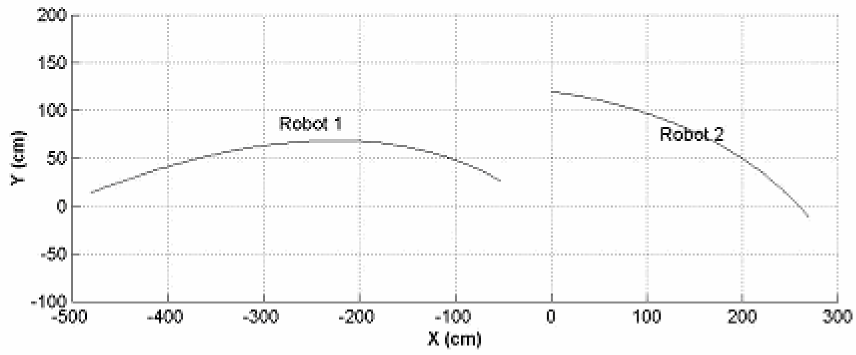

Figure 3-5 (a): Cartesian Paths of the Robots

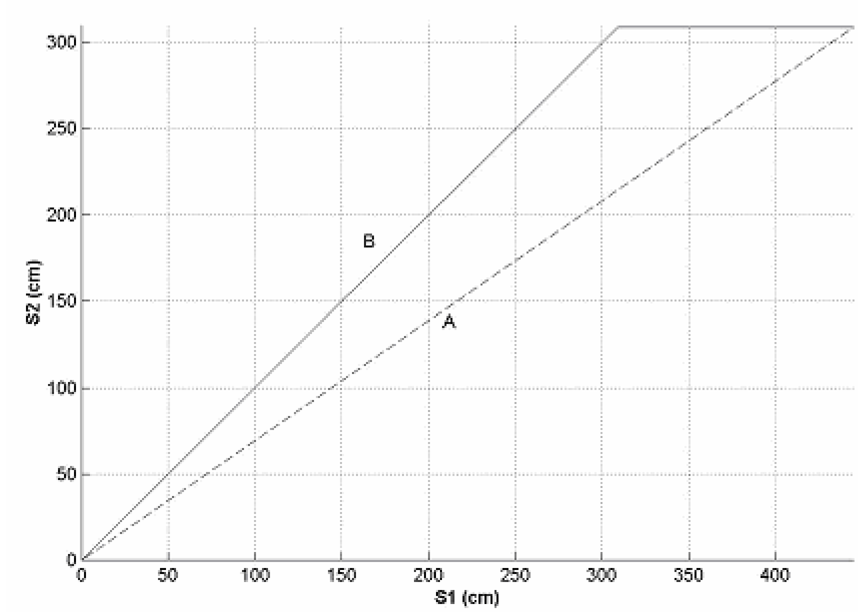

Figure 3-5 (b): Two Possible Paths in the Virtual Coordination Space

Now consider the simplest case of two robots moving toward their destinations without any possibility of collision and visibility blocking (Figure 3-5). The two robots are identical, with the same kinematic constraint of maximum velocity. In the coordination space there are two paths displayed. For path A, each robot moves at a constant velocity, and reaches the target simultaneously. This path is shown as a straight line in the coordination space. For path B, both robots move with the maximum velocity at the beginning, and after the Robot 2 reaches its destination, it stops and the Robot 1 continues moving with the maximum velocity until it reaches its destination as well. This path is shown as a combination of two straight line segments. For both of the two lines, the shortest possible time to complete the motion is

$$T = S_{2f} / V_{max} \qquad (3\text{-}10)$$

Thus the two paths have the same temporal cost. By taking the boundary of the coordination space as obstacles, it can be observed that by searching the graph with the straight line segments either with the direction of 45º or tangent to the obstacle boundary, the path with the lowest temporal cost can be found.

Extend this observation to the actual planning problem with obstacles, and the same approach can be applied to the planning problem of *n* robots. An iteration method is derived for the implementation on the real time control system.

The procedure of the algorithm is as follows:

1) For every two robots *i* and *j*, construct a virtual coordination space VCS(*i, j*), and plot the collision regions and invisible regions in the graph.
2) Searching the whole graph, find separated areas of obstacles;
3) For each separated obstacle, there are two priority options: *i* over *j*, or *j* over *i*. List all possible combinations of priorities; and for each combination, calculate the corresponding feasibility and minimal temporal cost;
4) With the combination of priority with the lowest temporal cost, choose the corresponding path and time-distance relationship as the desired trajectory.

In Step 3), for a certain combination of priorities, the detailed algorithm of calculating the feasibility and minimal temporal cost is:

1) Choose a time step Δt for iteration, considering the balance of precision and computation;
2) Each robot i which has not reached its final position moves forward with the maximum velocity for the time of Δt (Denoted as $S_i(m) \rightarrow S_i(m+1)$);
3) Plot the corresponding paths in all the virtual coordination spaces VCS(i, j) ($i=1 \ldots n, j=1 \ldots n, i \neq j$);
4) For those virtual coordination spaces in which the paths traversed into the obstacle areas, according to the priorities of the obstacles, put all the robots in a list with the descending consequence of priority. If such a list is logically impossible, then this combination of priorities is infeasible;
5) Starting from the robot with the second highest priority, replan the time-distance relationship of the robot i:
    a) For each robot j with a higher priority, check the VCS(i, j). Find the biggest value of distance $S_i(m+1)$ that the robot i can reach; Check the past waypoints $S_i(m)$ through $S_i(1)$. If $S_i(m+1)< S_i(k)$ and $S_i(m+1)> S_i(k-1)$, change the values of $S_i(k)$ through $S_i(m)$ to $S_i(m+1)$;
    b) Replan the waypoints $S(k)$ through $S(m)$ for all the robots with lower priorities to the robot i according to the changed values $S_i(k)$ through $S_i(m)$, using the same algorithm.
6) Check if all the robots have reached their final positions. If not, go back to step 2).

3.5 Algorithm Overview

To solve the global planning problem of moving a group of barrel robots from their initial positions to the desired final positions, a decoupled approach is developed in this work. The global paths and the time-distance relationships along the paths are planned serially, instead of simultaneously. The simple parabolas are used to connect the initial positions and final positions, generating the paths that satisfy the nonholonomic constraints. To avoid collisions between robots and minimize the time to complete the motion, a virtual coordination space is constructed for every two barrel robots, and the Tangent Graph method is used to find the optimal paths to avoid the obstacle regions and achieve minimal temporal cost.

The advantage of this algorithm is the reduced calculation resulted from the heuristic decomposition of the problem. This makes the algorithm easy to implement in the real time control system of the barrel robot system. For most situations this algorithm works well, while for those difficult situations turning the barrel robots in place is required to obtain new orientations, thus feasible planning can be completed thereafter.

1. CHAPTER 4: LOCAL CONTROL

4.1 Introduction

In the Robotic Safety Marker system, the mobile barrel robot is configured as a low cost autonomous mobile robot, with the ability of moving independently to a desired position with a reasonable precision. In the motion procedure, each barrel robot receives the desired waypoint (i.e. commanded position in global coordinates system) and the updated current position (i.e. measured position from outside sensors) from the lead robot. The barrel robot will then use a local control to obtain that position.

The barrel robot is designed as a differential drive mobile robot — the robot controls the motion and orientation by independently driving the two wheels. Based on this design, the technical difficulties were analyzed, and a real-time controller has been developed to meet the requirement.

4.2 Analysis of the Local Control Problem

4.2.1 Kinematic Model

Figure 4-1 is the schematic representation of the nonholonomic vehicle. It consists of a platform with two driving wheels on the same axis, and a free castor wheel stabilizing the platform. The motion and orientation are achieved by independent actuators, i.e. DC motors providing the torques to the rear wheels.

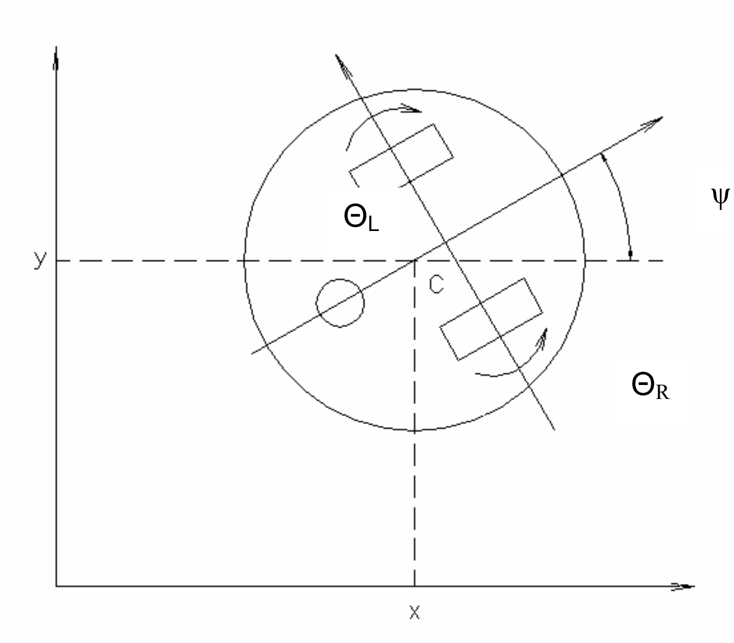

Figure 4-1: Schematic Representation of the Nonholonomic Vehicle

The position of the platform in the global Cartesian frame is completely specified by a state vector of three entries (n=3): $\underline{X} = [x, y, \psi]^T$, where (*x, y*) is the coordinates of the mass center C, and ψ is the orientation angle between the axis of symmetry and the X direction of the fixed Cartesian frame. Due to the small distance of the mass center to the axis of the driving wheels, the mass center C can be deemed as the intersection of the wheel axis and the axis of symmetry. This approximation facilitates our kinematic and dynamic analysis with small loss of precision.

An important assumption in the kinematic and dynamic analysis of the differential mobile robot is that the wheels roll on the ground without sliding, which can be expressed by the following equations:

$$v_L = \dot{\theta}_L \cdot r \tag{4-1}$$

$$v_R = \dot{\theta}_R \cdot r \tag{4-2}$$

where v_L ------ Velocity of the center of the left-hand wheel

v_R ------ Velocity of the center of the right-hand wheel

$\dot{\theta}_L$ ------ Angular velocity of the left-hand wheel

$\dot{\theta}_R$ ------ Angular velocity of the right-hand wheel

r ------ Radius of the wheels

The assumption of rolling without sliding also implies that the velocity of platform is on the direction of the symmetry axis, that is

$$\dot{x}\sin\psi - \dot{y}\cos\psi = 0 \tag{4-3}$$

This constraint is known as nonholonomic constraint, or non-integrable kinematic constraint. The nonholonomic constraint reduced the degree of freedom of the system to n-1 = 2, which is equal to the number of independent displacements of actuators. The forward kinematics can derived as

$$\begin{aligned} \dot{x} &= r\frac{\dot{\theta}_L+\dot{\theta}_R}{2}\cos\psi \\ \dot{y} &= r\frac{\dot{\theta}_L+\dot{\theta}_R}{2}\sin\psi \\ \dot{\psi} &= \frac{r}{D}(\dot{\theta}_L-\dot{\theta}_R) \end{aligned} \tag{4-4}$$

From the forward kinematic equations, it can be seen clearly that, with the same wheel rotation angles θ_L and θ_R, the final position will be different with the different velocity distribution during the motion.

4.2.2 Difficulties

Realizing a real-time controller for a mobile nonholonomic robot is a difficult problem from many points of view. The non-integrable constraint excludes a simple error based feedback controller, since the actuator controller does not have a static input. Instead, the desired value of the actuator changes with the integrated Cartesian position and orientation, which in turn comes from the dynamic response of the actuator to the inputted desired value. This is true even for the task of obtaining a static Cartesian position.

The difficulty of this task in our project is even increased due to the limitation of calculation capability and onboard sensor. The lower cost processor makes the execution time an important issue in the control algorithm, and sharing with other tasks such as communication makes the requirement stricter. The sole onboard sensor of wheel odometer requires frequent reading and numerical integration to obtain the current global position; moreover, the position error accumulates with the distance. All these constraints require an efficient controller that could obtain a desired position through a short path, with reasonable precision and calculation workload.

4.2.3 Existing solutions

The control of nonholonomic systems has been an active research area of nonlinear control for several years, and several methods have been developed to solve the problem. An important conclusion about this problem is that these systems can not be stabilized to the equilibrium point using a time-invariant, smooth (or even continuous) feedback control law (Brockett, 1983).

A piecewise continuous pure feedback law was introduced which makes the origin globally exponentially stable (Canudas de Wit, et al, 1991), and the result was extended to yield stabilization about an arbitrary point in the state space (Sordalen, et al, 1992). Both the position and orientation were controlled, and the exponential stability of the controller was proved. However, the velocity v and angular velocity ω of the robot are taken as the input to the controller, thus the dynamics of the robot driving mechanism was not considered.

The problem of controlling the relative position between a moving target and an autonomous nonholonomic vehicle was also studied (Palopoli, et al, 2000). A backstepping approach was applied to develop a multi-level hierarchical controller, and the digital implementation has been tested. The control algorithm proposed in this paper requires several levels of sub-controller and much calculation is need for real-time implementation.

4.3 Control Law

4.3.1 Overview

Based on the configuration of the barrel robot, a simple inverse Jacobian Cartesian scheme has been implemented. This is a common approach to robot control where an error ($\delta \underline{X}$) is created in Cartesian space using a desired Cartesian position ($\underline{X}_d$= [x_d, y_d, ψ_d]) and the measured position ($\underline{X}$= [x, y, ψ]). This error is transformed into joint space ($\underline{\delta\theta}$) using the inverse Jacobian derived below. The error in the joint space is then acted upon by a traditional PID controller to create torque commands ($\underline{\tau}$= [τ_l, τ_r]) for the robot's motors. Desired positions are created with a local parabola between the initial position and orientation and the desired waypoint. Again, specifying the initial orientation maintains smooth motion.

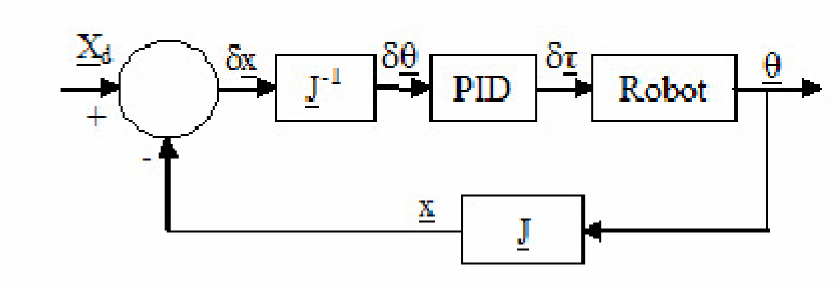

Figure 4-2: Inverse Jacobian Cartesian Controller

4.3.2 Local Trajectory Generation

As the input to the controller, the desired positions are created based on a trajectory generation algorithm that dictates a series of desired waypoints along a path with equal execution time interval, so that the controller can be implemented digitally at a fixed cycle. The trajectory generation algorithm, like the global trajectory planner, consists of two parts: path generation and waypoint generation. As the first step, a path is generated to connect the initial position and the desired final position. Then a certain time-distance relationship along the path is applied to find a series of waypoints with equal execution time interval as the input to the controller.

PATH GENERATION

The path can be created in many ways. Since the differential drive robot has the capability of turning in place, a possible method is turning in place until it faces the desired position, and then traveling in a straight line to that position. The path created in this way, however, will make the angular velocity of the driving wheels discontinuous, and thus produce unsmooth motion. Another common approach is to create a polynomial smooth path, with the initial direction coincide with the robot's initial orientation.

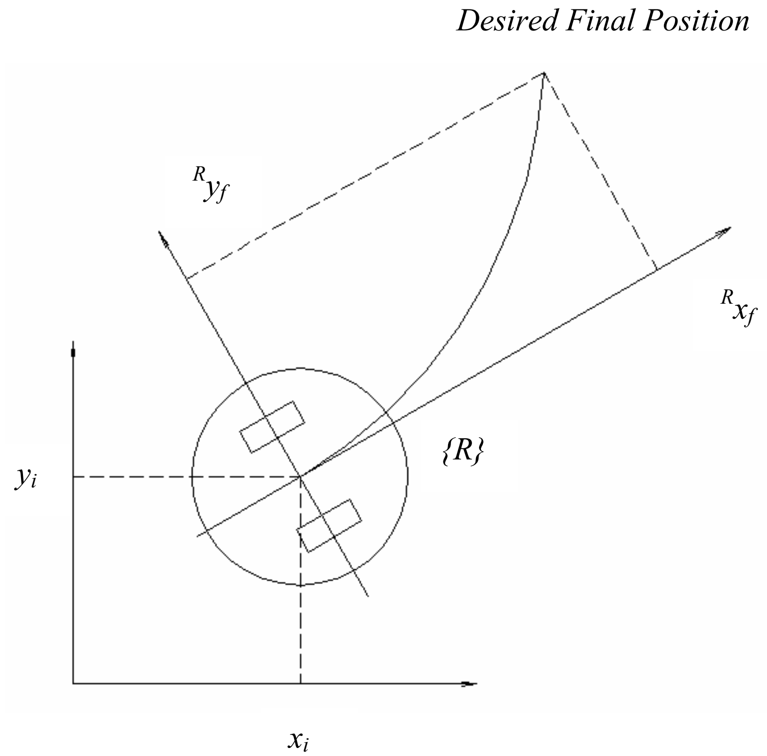

Figure 4-3: Local Path Generation

The procedure of path generation is as follows:

First, construct a fixed frame **{R}** with the origin at the initial position of the robot and the X- positive direction at the initial orientation of the robot. The desired final position and orientation in **{R}** are:

$$\begin{aligned} {}^{R}x_f &= \cos\psi_i \cdot (x_f - x_i) + \sin\psi_i \cdot (y_f - y_i) \\ {}^{R}y_f &= -\sin\psi_i \cdot (x_f - x_i) + \cos\psi_i \cdot (y_f - y_i) \\ {}^{R}\psi_f &= \psi_f - \psi_i \end{aligned} \tag{4-5}$$

where $[x_i, y_i, \psi_i]$ is the initial state of the robot.

Denote the path function in frame **{R}** as ${}^{R}y = F({}^{R}x)$, ${}^{R}x = 0 \sim {}^{R}x_f$. There are three constraints to the function:

(1) $F(0) = 0$;

(2) $\frac{dF}{d\,^{R}x}\Big|_{^{R}x=0} = 0$;

(3) $F(^{R}x_{f})=\,^{R}y_{f}$.

If the desired final orientation is also specified, there is the fourth constraint:

(4) $\frac{dF}{d\,^{R}x}\Big|_{^{R}x=^{R}x_{f}} = \tan(^{R}\psi_{f})$.

If only three constraints are applied, a polynomial function of order 2 will meet the requirements:

$$^{R}y=F(^{R}x)=C\cdot{}^{R}x^{2}, \qquad ^{R}x= 0\sim{}^{R}x_{f} \tag{4-6}$$

where C= $^{R}y_{f}/(^{R}x_{f}^{2})$.

If the fourth constraint also applies, a polynomial function of order 3 is needed:

$$^{R}y=F(^{R}x)= C_{3}\cdot{}^{R}x^{3}+C_{2}\cdot{}^{R}x^{2}, \qquad ^{R}x= 0\sim{}^{R}x_{f} \tag{4-7}$$

The coefficients C_3 and C_2 can be determined by the third and fourth constraints.

One problem with this path function is the singularity points with $^{R}x_{d}=0$. These points are located exactly on the axis of the driving wheels, i.e. exactly at the perpendicular direction to the initial direction. It can be proved that no polynomial function can satisfy these constraints simultaneously. For these singularity points an alternative path generation method should be used. For example, if continuity is not an important issue, the above-mentioned turn-and-go path can be applied: turning in place to the desired direction, and then traveling straight to the desired point. Another feasible method is the semicircle path, with the center located at the middle of the initial and desired final points. This method will require a more complicated path function, however.

TIME-DISTANCE RELATIONSHIP

With the path generated, the waypoints can be calculated as the input to the controller. The waypoints are supposed to be on the planned path, and be obtained at a fixed cycle by the robot. Since the path is fixed in the two dimensional space, the problem reduced to finding a time-distance relationship along the path.

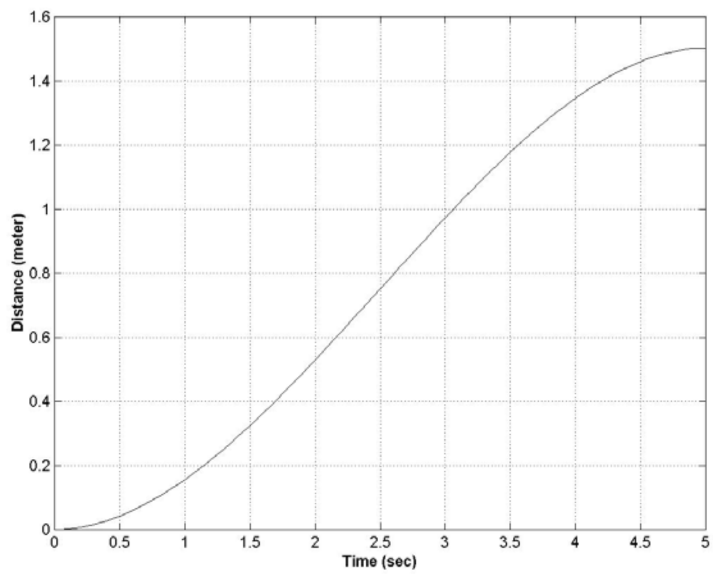

Figure 4-4: Time-Distance Relationship

Denote the distance to the initial point (x_0, y_0) along the planned path at a certain time point t with $s(t)$. There are two constraints on the function's value that come from the initial and final values:

$s(0)=0$

$s(T_f)=L_f$

where T_f is the total motion time interval, and L_f is the total length of the planned path.

Since a smooth motion is desired with the velocity continuous, the initial velocity should be zero. And the robot is also supposed to stay at the obtained position after the motion, so the final velocity is also set to zero. These requirements impose an additional two constraints:

$\dot{s}(0)=0$

$\dot{s}(T_f)=0$

These four constraints can be satisfied by a polynomial of at least third order. The function of third order polynomial, or cubic polynomial, ensures the acceleration to be linear to produce this motion. A cubic has the form:

$$s(t)=a_3t^3+a_2t^2+a_1t+a_0 \tag{4-8}$$

So the velocity along the path is

$\dot{s}(t)= 3a_3t^2+2a_2t+a_1$

Applying the constraints:

$s(0)=a_0=0$

$s(T_f)= a_3{*}T_f^3+a_2{*}T_f^2+a_1{*}T_f+a_0=L_f$

$\dot{s}(0)=a_1=0$

$\dot{s}(T_f)= 3a_3{*}T_f^2+2a_2{*}T_f+a_1=0$

Solve these equations for the coefficients:

$a_0 = 0$

$a_1 = 0$

$$a_3 = -\frac{2 * L_f}{T_f^{\ 3}}$$

$$a_2 = \frac{3 * L_f}{T_f^{\ 2}}$$

WAYPOINT GENERATION

Now we have planned the path, and derived the time-distance relationship along the path. To obtain the waypoint coordinates at a certain time point, the only problem left in the waypoint generation is the solution of waypoint coordinates from the distance along the path.

Suppose the path is planned as a parabola

${}^{R}y=F({}^{R}x)=C\cdot{}^{R}x^{2}, \qquad {}^{R}x= 0\sim{}^{R}x_{d}$

The length from the origin to the point $({}^{R}x, {}^{R}y)$ can be derived by integration:

$$s({}^{R}x)=\frac{1}{2}\left|{}^{R}x\right|\sqrt{1+(2C\,{}^{R}x)^{2}}+\frac{1}{4|C|}\ln(2\left|C\,{}^{R}x\right|+\sqrt{1+(2C\,{}^{R}x)^{2}}) \tag{4-9}$$

To solve for ${}^{R}x$ from $s({}^{R}x)$, the analytical solution is hard to obtain. Hence, for the implementation of real-time controller, a numerical solution with a certain level of precision could be implemented instead. One possible solution is the Taylor Expansion, with which ${}^{R}x$ can be expressed as a polynomial function of s. However, a high order polynomial is needed to obtain an acceptable precision, especially for a small value of C. For the real-time application, it may be very calculation expensive.

To overcome this difficulty, a simple iteration method is used in the waypoint generation. Consider the segment of parabola in Figure 4-5. Suppose the point i-1 has been obtained at time point t_{i-1}, and its coordinates $({}^{R}x_{i-1}, {}^{R}y_{i-1})$ and direction ${}^{R}\psi_{i-1}$ are all known. What we want to solve is the coordinates and orientation of the next waypoint t_i.

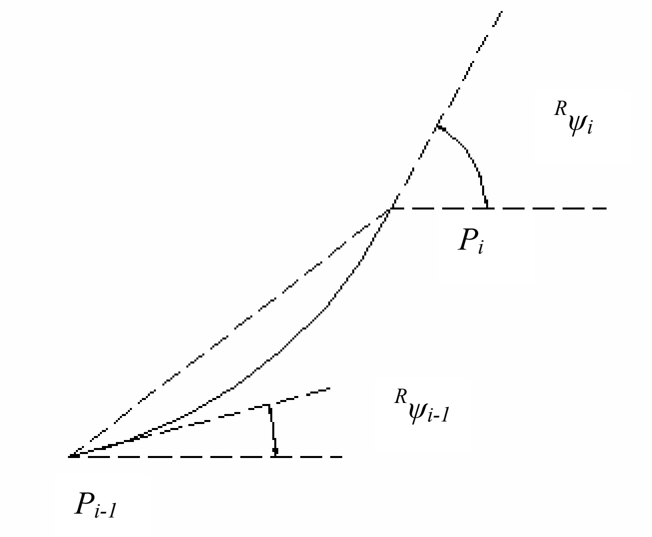

Figure 4-5: Waypoint Generation

First, the length of the arc $\overset{\frown}{P_{i-1}P_i}$ can be calculated:

$$\overset{\frown}{P_{i-1}P_i} = s(t_i) - s(t_{i-1})$$

Approximate the length of straight line segment $\overline{P_{i-1}P_i}$ with the arc length $\overset{\frown}{P_{i-1}P_i}$, and use the initial direction ${}^{R}\psi_{i-1}$ as the direction of the straight line. ${}^{R}x_i$ can be solved by geometry (suppose the desired final position (${}^{R}x_d$, ${}^{R}y_d$) lies in the first quadrant; final positions in other quadrants require corresponding sign change):

$$ {}^{R}x_i = {}^{R}x_{i-1} + [s(t_i) - s(t_{i-1})] * \cos {}^{R}\psi_{i-1} \tag{4-10}$$

From the path function, ${}^{R}y_i$ and ${}^{R}\psi_i$ can also be solved:

$$ {}^{R}y_i = C * {}^{R}x_i^2 \tag{4-11}$$

$$ {}^{R}\psi_i = \arctan(2 * C * {}^{R}x_i) \tag{4-12}$$

The error with the iteration tends to accumulate with the extension of the path, but the precision is acceptable with the current application. For a typical task setting:

obtaining a position (1, 1) with regard to frame {R} in 5 seconds with a control cycle of 0.05 seconds, the accumulated error with this method is less than 0.7%.

4.3.3 Dead-Reckoning

The term *Dead-Reckoning* refers to a mathematical process of projecting a point from a known starting location in the Cartesian coordinate space. In applying the method to the mobile robot, the "point" mentioned above is a reference position and orientation of the moving platform. The purpose of Dead-Reckoning in the mobile robot control is the derivation of a Jacobian relationship between the differential motion in Cartesian space and the differential motion in joint space. With the information of differential motion in the joint space (odometer readings of two driving wheels), the corresponding Cartesian differential motion can be solved to locate the robot in the Cartesian space. Conversely, with the information of the desired Cartesian differential motion, the corresponding desired joint differential motion can be solved as the input to the actuator controller.

Consider the differential robot motion in Figure 4-6. The robot moved an amount of $\delta\underline{X}=[\delta x, \delta y, \delta\psi]$ between t_{i-1} and t_i. The corresponding wheel rotation angle is $\delta\underline{\theta}=[\theta_L, \theta_R]$.

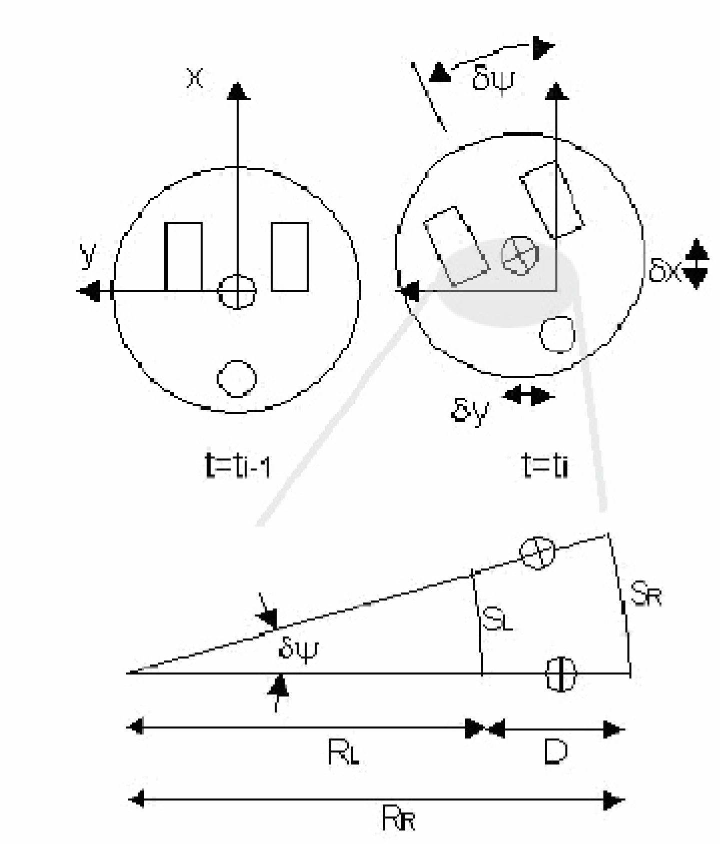

Figure 4-6: Differential Robot Motion

During this differential time the robot turn about a constant radius by an amount of $\delta\psi$. Here, S_L and S_R are the path lengths of the driving wheels. No slip is assumed so that the path lengths are related to the rotation angles of the wheels θ_L and θ_R by:

$$S_L=r^*\theta_L$$

$$S_R=r^*\theta_R$$

where r is radius of the wheels.

If small motions are assumed, the differential motion in Cartesian space can be found based on the differential motions in joint space. For the first step, solve for the direction change $\delta\psi$. The arc lengths of the wheels can be written as

$$S_L = R_L{}^* \delta\psi$$

$S_R = R_R * \delta\psi$

While the difference between R_L and R_R is equal to the distance between two wheels D:

$R_R - R_L = D$

From these equations $\delta\psi$ can be solved as:

$$\delta\psi = \frac{\mathrm{r}}{\mathrm{D}}(\delta\theta_R - \delta\theta_L) \tag{4-13}$$

Now δx and δy can be solved. Here an approximation was made that the straight-line distance $\sqrt{\delta x^2 + \delta y^2}$ equals to the arc length $(\frac{S_L + S_R}{2})$. Thus the position change can also be calculated:

$$\begin{aligned} \delta x &= (\frac{S_L + S_R}{2}) \cdot \cos(\frac{\delta\psi}{2}) = \frac{r}{2}(\delta\theta_L + \delta\theta_R)\cos[\frac{r}{2D}(\delta\theta_R - \delta\theta_L)] \\ \delta y &= (\frac{S_L + S_R}{2}) \cdot \sin(\frac{\delta\psi}{2}) = \frac{r}{2}(\delta\theta_L + \delta\theta_R)\sin[\frac{r}{2D}(\delta\theta_R - \delta\theta_L)] \end{aligned} \tag{4-14}$$

Note that this position change is observed in the robot-fixed coordinate system. To obtain the position change in the Cartesian space, a coordinate transformation should be taken:

$$\begin{aligned} dx &= \delta x \cdot \cos\psi - \delta y \cdot \sin\psi \\ dy &= \delta x \cdot \sin\psi + \delta y \cdot \cos\psi \end{aligned} \tag{4-15}$$

The position of the robot in Cartesian space can be determined from the changes in wheel position for small motions of the robot. For large motions the Cartesian position is non-observable based on wheel angle. The position of the robot is determined by accumulating differential motions:

$\underline{X}_i = \underline{X}_{i-1} + d\underline{X}$

This process, however, also accumulates measurement errors. Repeated laboratory testing has shown this error to be less than 5% of the overall motion on average. This error accumulation is not critical since the global motion of the robot is determined by the external measurement of the global planner, thus the location error can be eliminated periodically.

Conversely, the joint displacement can also be determined from the changes in Cartesian position for small motion. The inverse Jacobian controller of the barrel robot uses the inverse process to obtain the desired wheel rotation angle as the input to PID controller.

There is one problem with the inverse process, however. Three inputs, δx, δy and $\delta\psi$ may be used to solve for two unknown variables $\delta\theta_L$ and $\delta\theta_R$, and the equation set is over-determinant. Further observation of the three equations shows that they are not independent for smooth motion, and the simulation results verified this conclusion. Hence the desired wheel rotation angles can be calculated as (suppose $\delta x>0$ and $\delta y>0$, derived from Equ. 4-14):

$$\begin{aligned} \delta\theta_L &= \frac{2}{r}\sqrt{\delta x^2+\delta y^2} - \frac{2D}{r}\cdot \arctan(\frac{\delta y}{\delta x}) \\ \delta\theta_R &= \frac{2}{r}\sqrt{\delta x^2+\delta y^2} + \frac{2D}{r}\cdot \arctan(\frac{\delta y}{\delta x}) \end{aligned} \tag{4-16}$$

4.3.4 Controller Development

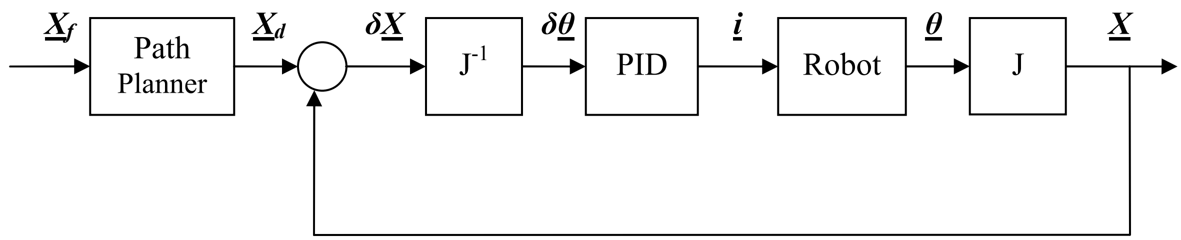

Figure 4-7: Original Control Scheme

The original control scheme is shown in Figure 4-7. A series of waypoints $\underline{X}_d$ are generated by the path planner as the input to the control loop. Then the error in Cartesian space $\delta\underline{X}$ is converted to the error in joint space $\delta\underline{\theta}$, which is acted upon by a tradition PID control law to create left and right motor input commands.

$$
\begin{aligned}
i_L &= K_P \cdot \delta\theta_L + K_I \cdot \int d\theta_L + K_D \cdot \frac{d\theta_L}{dt} \\
i_R &= K_P \cdot \delta\theta_R + K_I \cdot \int d\theta_R + K_D \cdot \frac{d\theta_R}{dt}
\end{aligned}
\tag{4-17}
$$

This scheme, however, is quite time consuming when implemented on the onboard microprocessor on the barrel robot, in that the complicated calculations of the Jacobian and the inverse Jacobian are both in the control loop. In fact these calculations are highly nonlinear, but we still use the name of Jacobian to indicate the conversion between the Cartesian space and the joint space. The outcome is that the control cycle is quite long (0.2 seconds on TFX-11 microprocessor), and the time delay in the control loop worsened the dynamic response.

Aware of this, a new idea of moving these calculations out of the control loop is proposed. After the waypoints $\underline{X}_d$ are generated, the corresponding rotation angles of the actuators $\underline{\theta}_d$ are calculated immediately, supposing that the robot follows the desired trajectory exactly. For each waypoint $\underline{X}_{d,i}$, the corresponding rotation angle $\underline{\theta}_{d,i}$ is calculated and stored in an array as the input to the controller of the actuator in the corresponding control cycle. The schematic figure is shown in Figure 4-8.

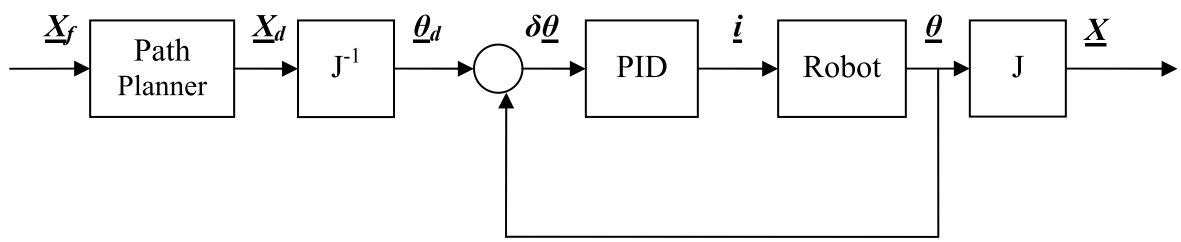

Figure 4-8: Revised Control Scheme

The control loop in the revised scheme is much simplified, as can be seen from the figure. The only calculation in the control loop is the PID control, and the control cycle can be reduced by more than a half.

This scheme, however, also causes error in the motion, since the final position not only depends on the final rotation angles of the wheels, but also depends on the angular velocity distribution in the procedure. If the actual rotation angles do not match the desired angles exactly during the whole procedure, the actually obtained position will be different from the desired position, even with the desired final rotation angles obtained exactly. Hence, the tracking error (error between $\underline{\theta}_{d,i}$ and $\underline{\theta}_i$) plays an important role in the final location precision. To reduce the tracking error, the PID controller should be finely tuned to improve the dynamic response and the control cycle be as short as possible.

With repeated experiments it can be found that, the location error with a big proportional gain K_P (20 in the current application) is much better than that with smaller K_P, and the resulting overshoot problem could be solved by applying a relatively smaller derivative gain K_D (5). An even smaller integral gain K_I (0.8) is applied to reduce the final actuator error, and that also contributed to the improvement of location precision. With repeated experiments under this setting, it can be shown that, for a typical task setting: obtaining a position (1, 1) with regard to frame {R} in 5 seconds with a control cycle of 0.05 seconds, the accumulated error less than 10%, which also included the error incurred by the onboard sensors (about 5%). Chapter 6 gives the test results and analysis in more detail.

4.4 Simulation

4.4.1 Dynamic Model

For the development of controller and the tuning of PID parameters, a dynamic model is developed base on the analysis of the mechanical and electrical design of the system.

ELECTRICAL CIRCUIT

The actuators of the barrel robot are two DC motors connected to two LM18245 H-bridge driver chips. The H-bridge motor driver can be modeled as a controlled current source, which provides a certain amount of current to the DC motor according to the digital control signal. The current provided by the motor driver can be derived by the I/O characteristics of LM18245:

$i(t) = \tau(t)*(V_{REF\ DAC}/16)*(4000/R_S)$

when $I(t) \leq [(V_{CC}-V_{BEMF})/R]*(16/V_{REF\ DAC})*(R_S/4000)$ (4-18)

or $i(t) = sign[\tau\ (t)]*\ (V_{CC}-V_{BEMF})/R$

when $|\ \tau\ (t)| > [(V_{CC}-V_{BEMF})/R]*(16/V_{REF\ DAC})*(R_S/4000)$.

where $\tau\ (t)$ ------ Output of the digital controller (-15 ~ 15, integer)

V_{CC} ------- Power supply voltage

V_{BEMF} ---- Back EMF voltage

$V_{REF\ DAC}$ -- D/A converter reference voltage

R_S ------ Reference resistance

The schematic figure of the motor driving circuit is as follows:

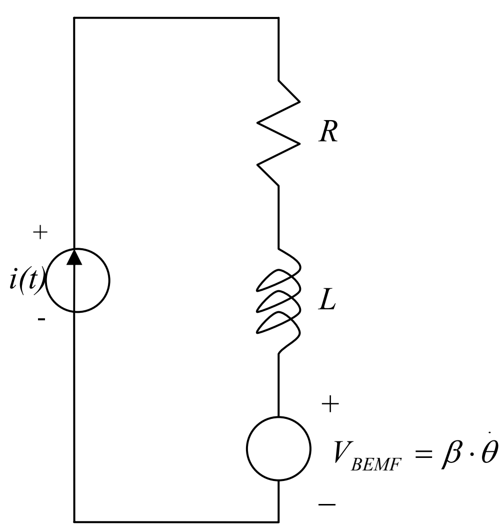

Figure 4-9: Motor Driving Circuit

MOTOR SHAFT DYNAMICS

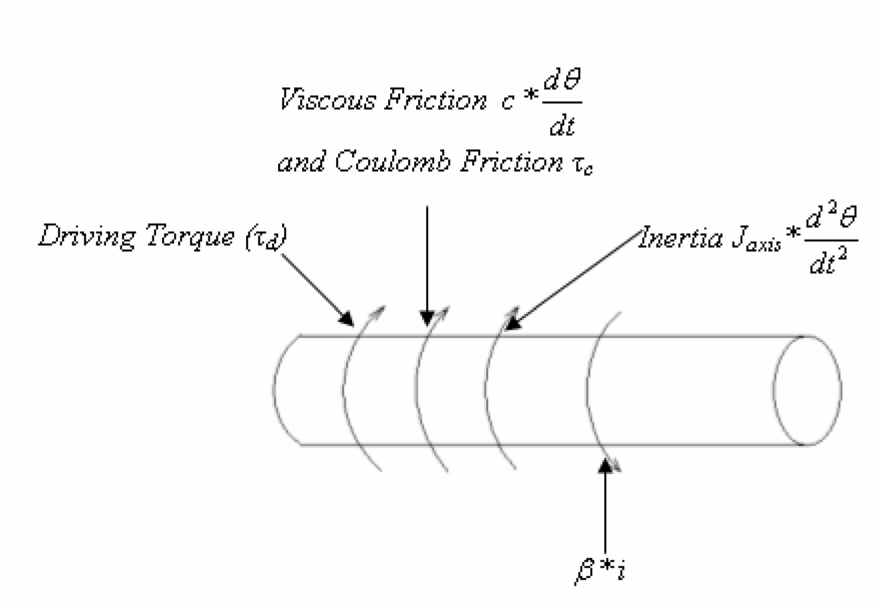

Figure 4-10: Shaft Dynamics

The differential equation governing the motion of the motor shaft is:

$$\beta_{motor} * i - \tau_{dmotor} - [C * \frac{d\theta_{motor}}{dt} + \tau_{cmotor}] - J_{axis} * \frac{d^2\theta_{motor}}{dt^2} = 0 \qquad (4\text{-}19)$$

where τ_{dmotor} is the motor driving torque, and τ_{cmotor} is the torque of Coulomb friction.

Due to the small mass of the shaft, the inertia term $J_{axis} * \frac{d^2\theta_{motor}}{dt^2}$ can be neglected.

Then (4-19) reduced to

$$\tau_{dmotor} = \beta_{motor} * i - [C * \frac{d\theta_{motor}}{dt} + \tau_{cmotor}] \qquad (4\text{-}20)$$

FRAME DYNAMICS

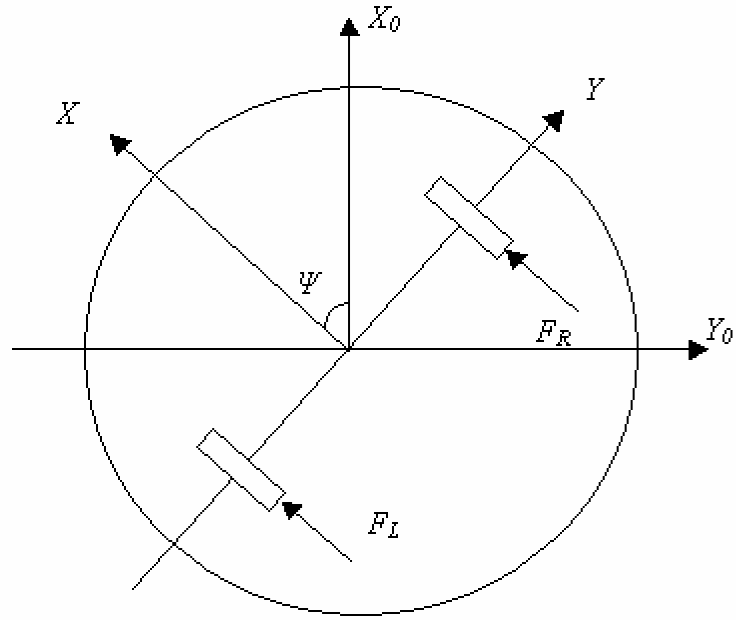

Figure 4-11: Frame Dynamics

Treat the frame of the barrel robot as a rigid body, and apply the Newton's second law:

$$F_L + F_R = m * \frac{d^2 x}{dt^2} \tag{4-21}$$

$$(F_R - F_L) * \frac{D}{2} = J * \frac{d^2 \psi}{dt^2} \tag{4-22}$$

where D is the wheel distance in the axis direction.

Neglecting the friction in the drive train, we have

$$F_L = \tau_{motorL} * k / r \tag{4-23}$$

$$F_R = \tau_{motorR} * k / r \tag{4-24}$$

where k is the gear ratio of the drive train, and r is the radius of the wheel.

Replacing the τ_{motor} in (4-23) and (4-24) with the expression in (4-20) for the left and right wheels, we have

$$F_L = (\beta_{motor} * i_L - C * \frac{d\theta_{motorL}}{dt} - \tau_{cmotorL}) * K / r$$

$$F_R = (\beta_{motor} * i_R - C * \frac{d\theta_{motorR}}{dt} - \tau_{cmotorR}) * K / r$$

Denoting $\beta=\beta_{motor}*k;\ c=C*k^2;\ \tau_c=\ \tau_{cmotor}$ *k, and note that $\theta_{motor}=\ \theta*k$, these two equations can be simplified as

$$F_L = (\beta * i_L - c * \frac{d\theta_L}{dt} - \tau_{cL}) / r \tag{4-25}$$

$$F_R = (\beta * i_R - c * \frac{d\theta_R}{dt} - \tau_{cR}) / r \tag{4-26}$$

Putting (4-25) and (4-26) into (4-21) and (4-22), the following equations can be derived:

$$(\beta * i_L - c * \frac{d\theta_L}{dt} - \tau_{cL} + \beta * i_R - c * \frac{d\theta_R}{dt} - \tau_{cR}) / r = m * \frac{d^2 x}{dt^2} \tag{4-27}$$

$$(\beta * i_R - c * \frac{d\theta_R}{dt} - \tau_{cR} - \beta * i_L + c * \frac{d\theta_L}{dt} + \tau_{cL}) * \frac{D}{2} / r = J * \frac{d^2 \psi}{dt^2} \tag{4-28}$$

From the kinematics of the frame, we have the following equations of the relationship between x, Ψ and θ_L, θ_R:

$x = (\theta_L + \theta_R)*r/2$

$\psi = (\theta_R - \theta_L)*r/D$

Replacing x, Ψ in (4-27) and (4-28), we have

$$(\beta * i_L - c * \frac{d\theta_L}{dt} - \tau_{cL} + \beta * i_R - c * \frac{d\theta_R}{dt} - \tau_{cR}) / r = \frac{mr}{2}(\frac{d^2\theta_L}{dt^2} + \frac{d^2\theta_R}{dt^2}) \tag{4-29}$$

$$(\beta * i_R - c\frac{d\theta_R}{dt} - \tau_{cR} - \beta * i_L + c\frac{d\theta_L}{dt} + \tau_{cL}) * \frac{D}{2} / r = \frac{Jr}{D}(\frac{d^2\theta_R}{dt^2} - \frac{d^2\theta_L}{dt^2}) \tag{4-30}$$

or $$\frac{d^2\theta_L}{dt^2} = (-\frac{c}{mr^2} - \frac{cD^2}{4Jr^2})\frac{d\theta_L}{dt} + (-\frac{c}{mr^2} + \frac{cD^2}{4Jr^2})\frac{d\theta_R}{dt} + (-\frac{1}{mr^2} - \frac{D^2}{4Jr^2})\tau_{cL}$$

$$+(-\frac{1}{mr^2}+\frac{D^2}{4Jr^2})\tau_{cR}+(\frac{\beta}{mr^2}+\frac{\beta D^2}{4Jr^2})i_L+(\frac{\beta}{mr^2}-\frac{\beta D^2}{4Jr^2})i_R \tag{4-31}$$

$$\frac{d^2\theta_R}{dt^2}=(-\frac{c}{mr^2}+\frac{cD^2}{4Jr^2})\frac{d\theta_L}{dt}+(-\frac{c}{mr^2}-\frac{cD^2}{4Jr^2})\frac{d\theta_R}{dt}+(-\frac{1}{mr^2}+\frac{D^2}{4Jr^2})\tau_{cL}$$

$$+(-\frac{1}{mr^2}-\frac{D^2}{4Jr^2})\tau_{cR}+(\frac{\beta}{mr^2}-\frac{\beta D^2}{4Jr^2})i_L+(\frac{\beta}{mr^2}+\frac{\beta D^2}{4Jr^2})i_R \tag{4-32}$$

Denoting $x_1=\theta_L$, $x_2=\frac{d\theta_L}{dt}$, $x_3=\theta_R$, $x_4=\frac{d\theta_R}{dt}$, the system dynamics can be written in the state space form:

$$\dot{x}_1=x_2$$

$$\dot{x}_2=(-\frac{c}{mr^2}-\frac{cD^2}{4Jr^2})x_2+(-\frac{c}{mr^2}+\frac{cD^2}{4Jr^2})x_4+(-\frac{1}{mr^2}-\frac{D^2}{4Jr^2})\tau_{cL}$$

$$+(-\frac{1}{mr^2}+\frac{D^2}{4Jr^2})\tau_{cR}+(\frac{\beta}{mr^2}+\frac{\beta D^2}{4Jr^2})i_L+(\frac{\beta}{mr^2}-\frac{\beta D^2}{4Jr^2})i_R$$

$$\dot{x}_3=x_4$$

$$\dot{x}_4=(-\frac{c}{mr^2}+\frac{cD^2}{4Jr^2})x_2+(-\frac{c}{mr^2}-\frac{cD^2}{4Jr^2})x_4+(-\frac{1}{mr^2}+\frac{D^2}{4Jr^2})\tau_{cL}$$

$$+(-\frac{1}{mr^2}-\frac{D^2}{4Jr^2})\tau_{cR}+(\frac{\beta}{mr^2}-\frac{\beta D^2}{4Jr^2})i_L+(\frac{\beta}{mr^2}+\frac{\beta D^2}{4Jr^2})i_R$$

or

$$
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{c}{mr^2}-\frac{cD^2}{4Jr^2} & 0 & -\frac{c}{mr^2}+\frac{cD^2}{4Jr^2} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{c}{mr^2}+\frac{cD^2}{4Jr^2} & 0 & -\frac{c}{mr^2}-\frac{cD^2}{4Jr^2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -\frac{1}{mr^2}-\frac{D^2}{4Jr^2} & -\frac{1}{mr^2}+\frac{D^2}{4Jr^2} \\ 0 & 0 \\ -\frac{1}{mr^2}+\frac{D^2}{4Jr^2} & -\frac{1}{mr^2}-\frac{D^2}{4Jr^2} \end{bmatrix}
$$

$$
\begin{bmatrix} \tau_{cL} \\ \tau_{cR} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{\beta}{mr^2}+\frac{\beta D^2}{4Jr^2} & \frac{\beta}{mr^2}-\frac{\beta D^2}{4Jr^2} \\ 0 & 0 \\ \frac{\beta}{mr^2}-\frac{\beta D^2}{4Jr^2} & \frac{\beta}{mr^2}+\frac{\beta D^2}{4Jr^2} \end{bmatrix} \begin{bmatrix} i_L \\ i_R \end{bmatrix}
$$

(4-33)

4.4.2 Controller Parameter Tuning

From the dynamic model it can be seen that there are several nonlinear links existing in the control loop. First, the output of the digital controller is rounded to an integer between -15 and 15, which is the range of H-bridge inputs, while the over limit part is truncated. Second, the relationship between the digital controller output and the motor driving current is linear in a certain range, which is in turn determined by the back EMF induced by the shaft rotation. The Coulomb friction is the other major nonlinear link, since it affects the dynamic response of the start stage greatly. The existence of these nonlinear links makes the controller parameter tuning through simulation difficult and error prone.

Aware of this, the parameter tuning in the barrel robot control is based on the combination of the simulation analysis and experiment. A linearized model is derived from the foregoing analysis, and then the root locus method is used to get a rough estimation of gain K_P, which is the major parameter of PID controller. With K_P

determined, repeated experiments with different sets of integral gain K_I and derivative gain K_D are done to find the parameters that give the highest location precision.

Ignoring the Coulomb friction terms, and apply the Laplace transformation to the dynamic equations (4-31), (4-32), we get

$$\Theta_L s^2 = (-\frac{c}{mr^2} - \frac{cD^2}{4Jr^2})\Theta_L s + (-\frac{c}{mr^2} + \frac{cD^2}{4Jr^2})\Theta_R s + (\frac{\beta}{mr^2} + \frac{\beta D^2}{4Jr^2})I_L + (\frac{\beta}{mr^2} - \frac{\beta D^2}{4Jr^2})I_R \tag{4-34}$$

$$\Theta_R s^2 = (-\frac{c}{mr^2} + \frac{cD^2}{4Jr^2})\Theta_L s + (-\frac{c}{mr^2} - \frac{cD^2}{4Jr^2})\Theta_R s + (\frac{\beta}{mr^2} - \frac{\beta D^2}{4Jr^2})I_L + (\frac{\beta}{mr^2} + \frac{\beta D^2}{4Jr^2})I_R \tag{4-35}$$

Then the transfer functions $\frac{\Theta_L(s)}{I_L(s)}, \frac{\Theta_L(s)}{I_R(s)}, \frac{\Theta_R(s)}{I_L(s)}, \frac{\Theta_R(s)}{I_R(s)}$ can be derived as

$$\frac{\Theta_L(s)}{I_L(s)} = \frac{\Theta_R(s)}{I_R(s)} = \frac{(4J\beta r^2 + m\beta r^2 D^2)s + 4c\beta D^2}{4mJr^4 s^3 + (8Jcr^2 + 2mcD^2 r^2)s^2 + 4c^2 D^2 s} \tag{4-36}$$

$$\frac{\Theta_L(s)}{I_R(s)} = \frac{\Theta_R(s)}{I_L(s)} = \frac{m\beta D^2 r^2 - 4J\beta r^2}{4mJr^4 s^2 + (8Jcr^2 + 2mcD^2 r^2)s + 4c^2 D^2} \tag{4-37}$$

Apply a simple proportional control law, the output of the digital controller is

$$\tau_L = K_P \cdot (\theta_{dL} - \theta_L) \tag{4-38}$$

$$\tau_R = K_P \cdot (\theta_{dR} - \theta_R) \tag{4-39}$$

And according to the characteristic of the motor driver, and ignoring the saturation effect, the currents of the motors are:

$$i_L = (\frac{V_{REFDAC}}{16})(\frac{4000}{R_S})\tau_L \tag{4-40}$$

$$i_R = (\frac{V_{REFDAC}}{16})(\frac{4000}{R_S})\tau_R \tag{4-41}$$

So the block diagram of the control loop can be shown as Figure 4-12.

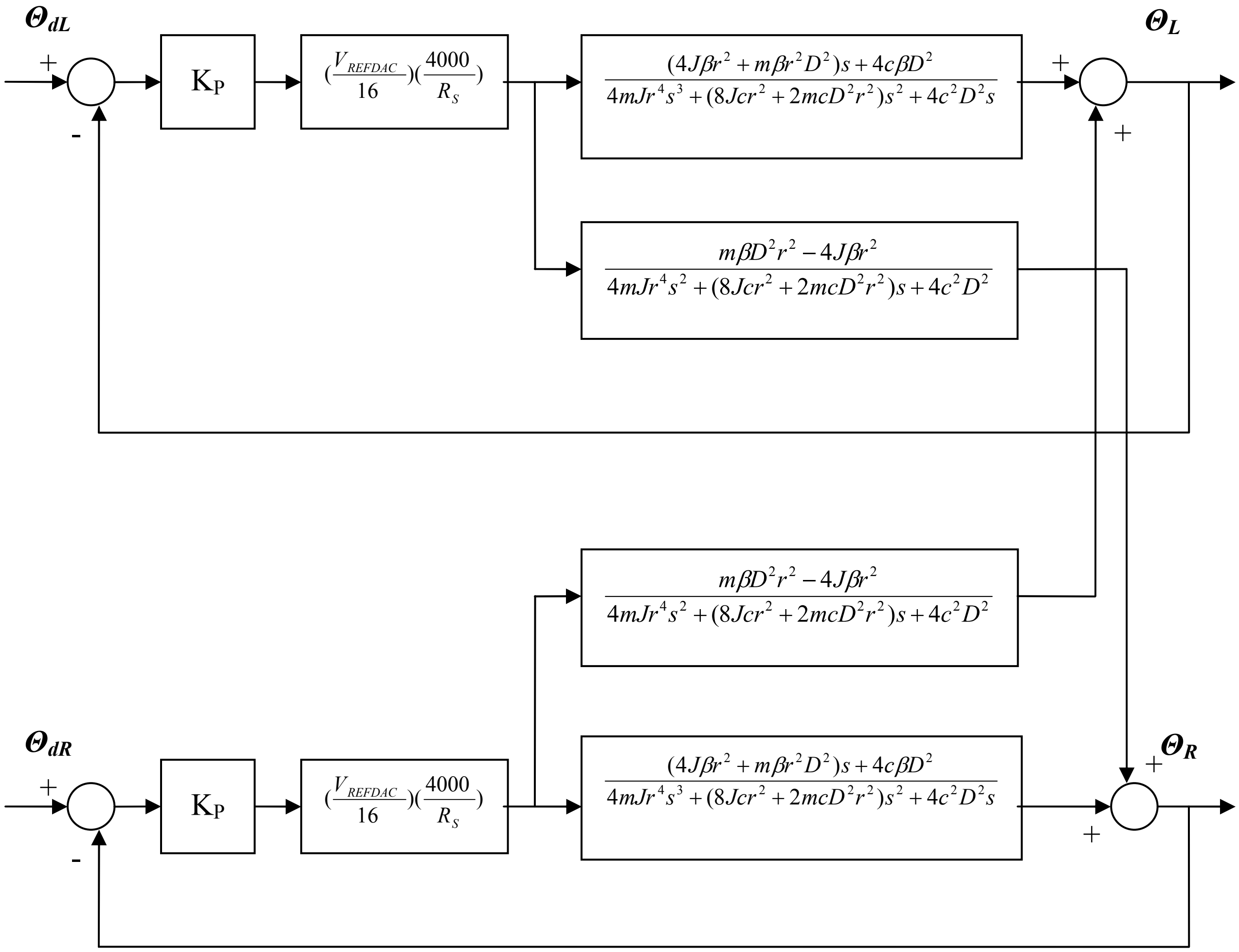

Figure 4-12: Block Diagram

To derive the transfer function $\frac{\Theta_L(s)}{\Theta_{dL}(s)}$ and $\frac{\Theta_R(s)}{\Theta_{dL}(s)}$, simplify the block diagram first. Denote $\frac{(4J\beta r^2 + m\beta r^2 D^2)s + 4c\beta D^2}{4mJr^4s^3 + (8Jcr^2 + 2mcD^2r^2)s^2 + 4c^2D^2s}$ as A,

$\frac{m\beta D^2 r^2 - 4J\beta r^2}{4mJr^4 s^2 + (8Jcr^2 + 2mcD^2 r^2)s + 4c^2 D^2}$ as B, $(\frac{V_{REFDAC}}{16})(\frac{4000}{R_S})$ as C_H, and set $\Theta_{dR}(s)$ as zero, the simplified block diagram is shown in Figure 4-13 and 4-14.

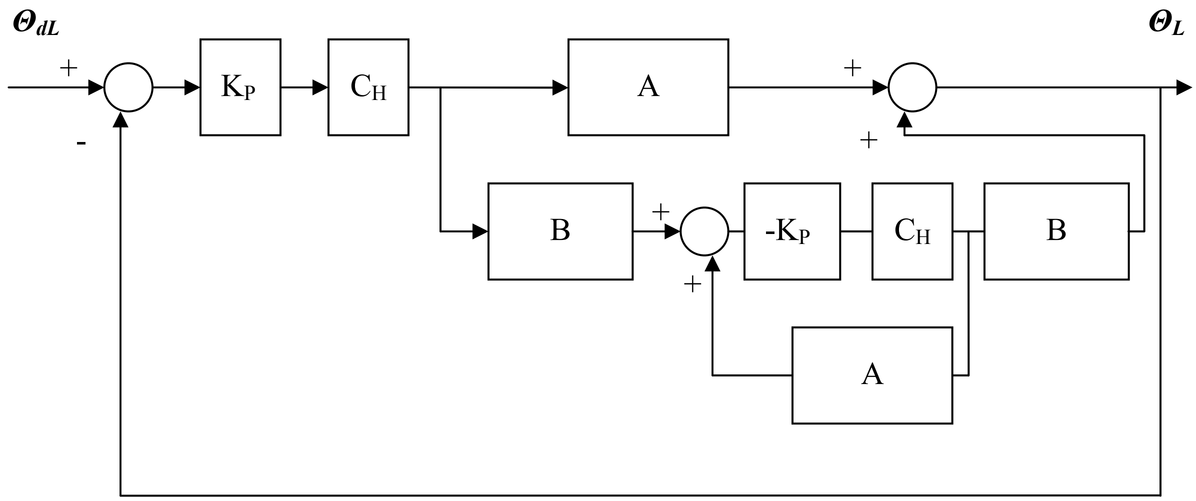

Figure 4-13: Simplified Block Diagram for θ_L/θ_{dL}

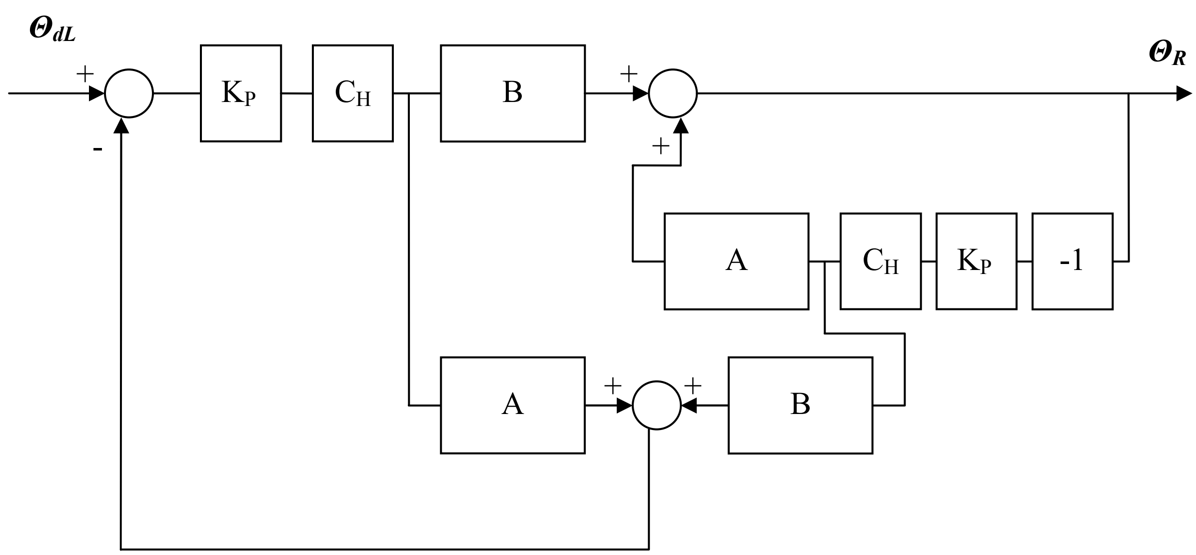

Figure 4-14: Simplified Block Diagram for θ_R/θ_{dL}

From the block diagram, the transfer functions can be derived as

$$\frac{\Theta_L(s)}{\Theta_{dL}(s)} = \frac{(K_P C_H \beta r^2 D^2 m + 4K_P C_H \beta J r^2)s^2 + 4K_P C_H \beta c D^2 s + 4{K_P}^2 {C_H}^2 \beta^2 D^2}{4mr^4 J s^4 + (2mD^2 r^2 c + 8Jcr^2)s^3 + (2K_P C_H \beta m D^2 r^2 + 8K_P C_H \beta J r^2 + 4c^2 D^2)s^2 + 8K_P C_H \beta c D^2 s + 4{K_P}^2 {C_H}^2 \beta^2 D^2}$$

(4-42)

$$\frac{\Theta_R(s)}{\Theta_{dL}(s)} = \frac{K_P C_H \beta r^2 (mD^2 - 4J)s^2}{4mr^4 J s^4 + (2mD^2 r^2 c + 8Jcr^2)s^3 + (2K_P C_H \beta m D^2 r^2 + 8K_P C_H \beta J r^2 + 4c^2 D^2)s^2 + 8K_P C_H \beta c D^2 s + 4{K_P}^2 {C_H}^2 \beta^2 D^2}$$

(4-43)

And the transfer function $\frac{\Theta_L(s)}{\Theta_{dR}(s)}$ and $\frac{\Theta_R(s)}{\Theta_{dR}(s)}$ can also be derived by the same procedure as:

$$\frac{\Theta_L(s)}{\Theta_{dR}(s)} = \frac{K_P C_H \beta r^2 (mD^2 - 4J)s^2}{4mr^4 J s^4 + (2mD^2 r^2 c + 8Jcr^2)s^3 + (2K_P C_H \beta m D^2 r^2 + 8K_P C_H \beta J r^2 + 4c^2 D^2)s^2 + 8K_P C_H \beta c D^2 s + 4{K_P}^2 {C_H}^2 \beta^2 D^2}$$

(4-44)

$$\frac{\Theta_R(s)}{\Theta_{dR}(s)} = \frac{(K_P C_H \beta r^2 D^2 m + 4K_P C_H \beta J r^2)s^2 + 4K_P C_H \beta c D^2 s + 4{K_P}^2 {C_H}^2 \beta^2 D^2}{4mr^4 J s^4 + (2mD^2 r^2 c + 8Jcr^2)s^3 + (2K_P C_H \beta m D^2 r^2 + 8K_P C_H \beta J r^2 + 4c^2 D^2)s^2 + 8K_P C_H \beta c D^2 s + 4{K_P}^2 {C_H}^2 \beta^2 D^2}$$

(4-45)

All the transfer functions $\frac{\Theta_L(s)}{\Theta_{dL}(s)}$, $\frac{\Theta_R(s)}{\Theta_{dL}(s)}$, $\frac{\Theta_L(s)}{\Theta_{dR}(s)}$ and $\frac{\Theta_R(s)}{\Theta_{dR}(s)}$ have the same characteristic function, and thus have the same systems poles. Since the characteristic function is the second order polynomial of K_P, the traditional root locus analysis is not applicable.

To overcome this difficulty, the approach used here is to calculate system poles directly with different values of K_P, and draw the loci of the roots on the complex plane. The root locus with K_P from 0 to 20 is shown in Figure 4-15.

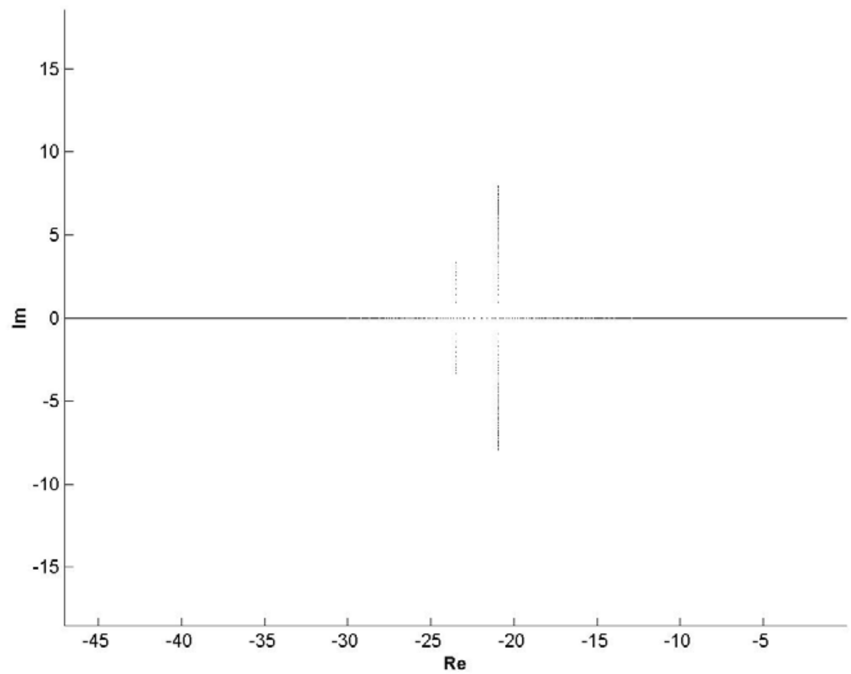

Figure 4-15: **Root Locus for PID Controller Tuning**

For $K_P = 0$, there are four real system poles located at about -46.8 (denoted as A), -41.7 (denoted as B), -0.2(denoted as C) and -0.2(denote as D) on the real axis. With the increase of the gain, the poles A and B move to the right, while the poles C and D move to the left. When the poles B and C meet with each other at about -21 of the real axis, they begin to diverge and move upwards and downwards vertically. The gain K_P at this point is 17.6, and the other poles A and D remain on the real axis at -31.2 and -15.8. With further increase of the gain, the two complex poles B and C continue move upwards and downwards, while the other two poles A and D continue to move in opposite directions. When the poles A and D meet at -23.5 of the real axis ($K_P = 19.6$), they begin to move upwards and downwards vertically. In order to obtain the best dynamic response with least overshoot, the gain could be selected between the range of 17.6 ~ 19.6.

The test result on the barrel robot matched this analysis result. With repeated experiments with different sets of PID gain values, the parameters giving the highest location precision were selected. The parameters of the PID controller were set as:

$K_P = 20$

$K_I = 0.8$

$K_D = 5$

CHAPTER 5: STABILITY ANALYSIS

5.1 Global Stability Analysis

Taking the group of barrel robots as the system, their positions are the controlled object of the planner and controller. So the global control loop could be constructed as shown in Figure 5-1.

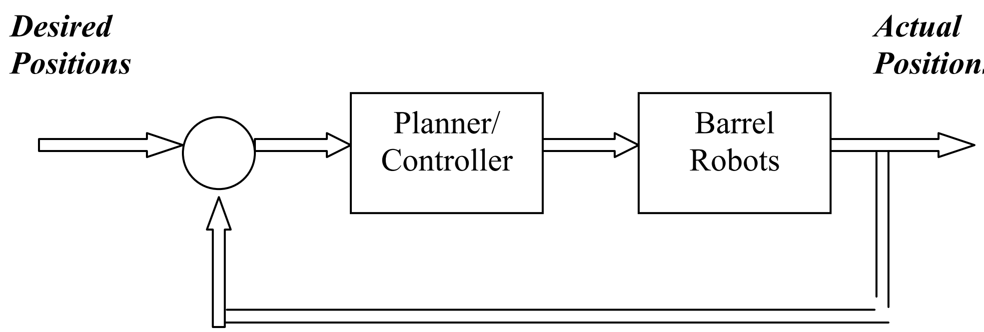

Figure 5-1: Global Control Loop

In the control loop, the link of *Planner/Controller* is very complicated, because it includes both the planning algorithms and the control laws, at both the global and local level. However, further analysis showed that the global planner is rather a link out of the control loop than an inside one, since in most situations it just runs once, and generates waypoints for the barrel robots. For the barrel robots, the task of their controller could be described as obtaining static positions in the Cartesian space. Thus, the control loop could be reduced to Figure 4-7, or Figure 4-8, as described in Chapter 4. Based on this analysis, the problem of global stability could be reduced to the problems of Cartesian stability and actuator stability, both of which will be discussed in detail in the following sections.

5.2 Cartesian Stability Analysis

5.2.1 Definition

It is commonly known that the stability is a very important issue in the control algorithm development, no matter for holonomic systems or nonholonomic systems. However, for the nonholonomic system, the existence of non-integrable constraints excludes the possibility of developing a control algorithm for the system to be stable to a single equilibrium point in the state space; instead, a feedback control can be developed so that the system is locally asymptotically stable to an equilibrium manifold (Bloch, et al, 1989). A manifold is a topological space that is locally Euclidean (i.e., around every point, there is a neighborhood which is topologically the same as the open unit ball in $\mathbb{R}^n$). The significance of this conclusion in the barrel robot control system can be illustrated as follows.

The dynamic system of the barrel robot has totally 5 configuration variables: $\underline{q}= [x, y, \psi, \theta_L, \theta_R]^T$, where $\underline{q}$ is the configuration vector, (x, y) is the Cartesian position of the robot, ψ is the orientation of the robot, and (θ_L, θ_R) is the positions of the two wheels. Assuming no slip between the wheels and the ground, the nonholonomic constraints can be written in the following equations of these variables:

$$\dot{x} = r\frac{\dot{\theta}_L + \dot{\theta}_R}{2}\cos\psi$$

$$\dot{y} = r\frac{\dot{\theta}_L + \dot{\theta}_R}{2}\sin\psi \qquad (5\text{-}1)$$

$$\dot{\psi} = \frac{r}{D}(\dot{\theta}_L - \dot{\theta}_R)$$

which is derived from the differential mobile robot kinematics, as shown in Chapter 4. These equations can also be written in matrix form:

$$\begin{bmatrix} 1 & 0 & 0 & -\frac{r\cos\psi}{2} & -\frac{r\cos\psi}{2} \\ 0 & 1 & 0 & -\frac{r\sin\psi}{2} & -\frac{r\sin\psi}{2} \\ 0 & 0 & 1 & -\frac{r}{D} & \frac{r}{D} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{\theta}_L \\ \dot{\theta}_R \end{bmatrix} = 0$$

or

$$J(\underline{q})\,\dot{\underline{q}} = 0$$

where $J(\underline{q})$ denotes a 3×5 matrix function with full rank, and $\dot{\underline{q}}$ is an 5-vector of velocity variables $[\dot{x}, \dot{y}, \dot{\psi}, \dot{\theta}_L, \dot{\theta}_R]^T$. According to the conclusion of (Bloch, et al, 1989), the equilibrium manifold is at least 3 dimensional, and the equilibrium manifold can be expressed as

$$E = \{(\underline{q}_1, \underline{q}_2, \dot{\underline{q}}_1, \dot{\underline{q}}_2) \mid \underline{q}_1 = 0, \dot{\underline{q}}_1 = 0, \dot{\underline{q}}_2 = 0\} \qquad (5\text{-}2)$$

where $\underline{q}_1$, $\underline{q}_2$ are the partitions of q which has 2 and 3 variables respectively.

Considering the purpose of the barrel robot controller, the partition can be done as

$\underline{q}_1 = [x, y]$

$\underline{q}_2 = [\psi, \theta_L, \theta_R]$ (5-3)

The physical meaning of the equilibrium manifold is that the position converges to (0, 0) with the orientation and wheel angles uncontrolled, and all the velocities are zero with regard to the equilibrium manifold.

The stability of the equilibrium manifold *E* can be defined as: the equilibrium manifold E is said to be asymptotically stable if it is stable in the sense of Lyapunov and $||(q_1, \dot{q}_1, \dot{q}_2)|| \rightarrow 0$ as $t \rightarrow \infty$ (Hui, et al, 1991). From this definition, it can be seen that the stability of the barrel robot can only be discussed with regard to the manifold instead of a point in the configuration space, while in the Cartesian space the robot is stable to a desired point with the orientation uncontrolled. This definition is suitable for the actual application of the barrel robot.

5.2.2 Analysis with Lyapunov Second Method

Considering the inverse Jacobian Cartesian controller in Figure 4-7. To obtain a certain point in 2 dimensional Cartesian space, a position error (dx, dy) is calculated by comparing the desired position (x_d, y_d) and the current position (x, y), and then it is transferred to the robot-fixed coordinate system by taking account in the current robot orientation ψ:

$$\begin{aligned} \delta x &= dx \cdot \cos\psi + dy \cdot \sin\psi \\ \delta y &= -dx \cdot \sin\psi + dy \cdot \cos\psi \end{aligned} \qquad (5\text{-}4)$$

The transformed error is transferred to the joint space using Eqn 4-16, and the calculated wheel angle ($\delta\theta_L, \delta\theta_R$) is used as the input to the actuator PID controller. This input, as can be seen, is changing with the current position and orientation of the robot, which in turn comes from the dynamic response to the input. With these highly nonlinear links existing in the control loop, it is very hard to find a proper Lyapunov function to judge the system stability.

Aware of this, a new approach of coordinate transformation is proposed to facilitate the controller stability analysis. The basic idea is to introduce a certain coordinate system so that the nonholonomic constraints have a trivial specification, and no coordinate transformation between the global frame and the robot-fixed frame is needed in the control loop with regard to the new coordinate system.

Considering the coordinate system in Figure 5-2. Frame {R} is attached to the robot, and in this frame the desired Cartesian position is indicated by a vector of two variables ${}^{R}\underline{X} = [a,\ \alpha]$, where a is the length of the arc with the positive direction coinciding with the robot's positive direction, and α is the angle between the robot's orientation and the direction of the desired position.

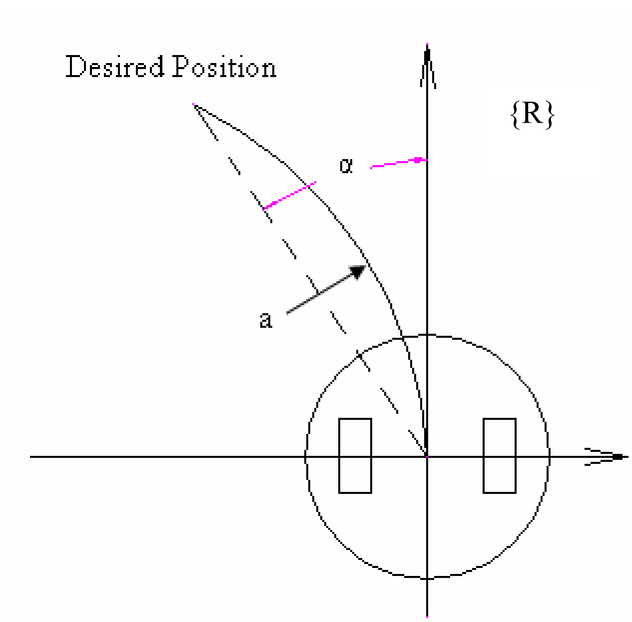

Figure 5-2: New Coordinate system

2. CONTROL LAW

With this coordinate system, the desired rotation angles of the two wheels can be written as:

$$\begin{aligned} \theta_L &= \frac{2}{r}a - \frac{2D}{r}\alpha \\ \theta_R &= \frac{2}{r}a + \frac{2D}{r}\alpha \end{aligned} \qquad (5\text{-}5)$$

These desired angles can be treated as the errors in the joint space, and the control law described in Chapter 4 can be applied to them to generate the torque inputs to the motors τ_L, τ_R. Since a very small integral gain is applied in the PID controller, the behavior of the controller can be deemed as a PD controller, and the equations are:

$$\begin{aligned} \tau_L &= -K_P \cdot \theta_L - K_D \cdot \frac{d\theta_L}{dt} \\ \tau_R &= -K_P \cdot \theta_R - K_D \cdot \frac{d\theta_R}{dt} \end{aligned}$$

Substitute Eqn (5-5) into the above equations:

$$\begin{aligned} \tau_L &= -K_P \cdot \frac{2}{r}a + K_P\frac{2D}{r}\alpha - K_D \cdot \frac{2}{r}\frac{da}{dt} + K_D \cdot \frac{2D}{r}\frac{d\alpha}{dt} \\ \tau_R &= -K_P \cdot \frac{2}{r}a - K_P\frac{2D}{r}\alpha - K_D \cdot \frac{2}{r}\frac{da}{dt} - K_D \cdot \frac{2D}{r}\frac{d\alpha}{dt} \end{aligned} \qquad (5\text{-}6)$$

Saturation and digitalization are ignored, since they will not affect the sign of Lyapunov function, and thus will not change the conclusion about stability.

3. LYAPUNOV FUNCTION

To search for a Lyapunov function to judge the controller stability, first consider the kinetic energy with the platform:

$$E = \frac{1}{2}mv^2 + \frac{1}{2}J\omega^2 \qquad (5\text{-}7)$$

and $$\dot{E} = mv\dot{v} + J\omega\dot{\omega} \qquad (5\text{-}8)$$

The robot velocity v and angular velocity ω are the derivative of a and α, respectively:

$$v = \frac{da}{dt} \tag{5-9}$$

$$\omega = \frac{d\alpha}{dt} \tag{5-10}$$

From the no wheel slip assumption we can get the relationship between v, ω and $\frac{d\theta_L}{dt}$, $\frac{d\theta_R}{dt}$:

$$v = \frac{r}{2}(\frac{d\theta_L}{dt} + \frac{d\theta_R}{dt}) \tag{5-11}$$

$$\omega = \frac{r}{D}(\frac{d\theta_R}{dt} - \frac{d\theta_L}{dt}) \tag{5-12}$$

where r is the radius of the wheels, and D is the distance between the wheels.

Replace the v and ω in Eqn (5-8) with Eqn (5-11) and (5-12):

$$\dot{E} = \frac{mr^2}{4}(\frac{d\theta_L}{dt} + \frac{d\theta_R}{dt})(\frac{d^2\theta_L}{dt^2} + \frac{d^2\theta_R}{dt^2}) + \frac{Jr^2}{D^2}(\frac{d\theta_R}{dt} - \frac{d\theta_L}{dt})(\frac{d^2\theta_R}{dt^2} - \frac{d^2\theta_L}{dt^2})$$

Denoting $x_1 = \theta_L$, $x_2 = \frac{d\theta_L}{dt}$, $x_3 = \theta_R$, $x_4 = \frac{d\theta_R}{dt}$, and substituting the state space equations in Chapter 4 into $\dot{E}$, we get

$$\begin{aligned}\dot{E} = {} & \frac{mr^2}{4}(x_2 + x_4)(-\frac{2c}{mr^2}\cdot x_2 - \frac{2c}{mr^2}\cdot x_4 - \frac{2}{mr^2}\cdot\tau_{cL} - \frac{2}{mr^2}\cdot\tau_{cR} + \frac{2\beta}{mr^2}\cdot\tau_L + \frac{2\beta}{mr^2}\cdot\tau_R) \\ & + \frac{Jr^2}{D^2}(x_4 - x_2)(\frac{cD^2}{2Jr^2}\cdot x_2 - \frac{cD^2}{2Jr^2}\cdot x_4 + \frac{D^2}{2Jr^2}\cdot\tau_{cL} - \frac{D^2}{2Jr^2}\cdot\tau_{cR} - \frac{\beta D^2}{2Jr^2}\cdot\tau_L + \frac{\beta D^2}{2Jr^2}\cdot\tau_R)\end{aligned}$$

After simplification:

$$\dot{E} = -c({x_2}^2 + {x_4}^2) - (x_2\tau_{cL} + x_4\tau_{cR}) + \beta(x_2\tau_L + x_4\tau_R) \tag{5-13}$$

In the expression of $\dot{E}$, the first term $-c(x_2{}^2 + x_4{}^2)$ is negative definite. And according to the characteristics of the Coulomb friction, when the angular velocity (x_2 or x_4) are not zero, the corresponding Coulomb friction (τ_{cL} or τ_{cR}) is always opposite to the angular velocity in direction, thus has the same sign according to the sign convention (see Figure 4-10). Therefore, the second term $-(x_2\tau_{cL} + x_4\tau_{cR})$ is always negative or zero, and thus also negative definite. In the analysis of the third term $\dot{E}_3 = \beta(x_2 i_L + x_4 i_R)$, substitute the motor current input τ_L and τ_R with the control law (5-6):

$$\dot{E}_3 = -\beta[x_2(K_P \cdot \frac{2}{r}a - K_P\frac{2D}{r}\alpha + K_D \cdot \frac{2}{r}\frac{da}{dt} - K_D \cdot \frac{2D}{r}\frac{d\alpha}{dt})$$

$$x_4(K_P \cdot \frac{2}{r}a + K_P\frac{2D}{r}\alpha + K_D \cdot \frac{2}{r}\frac{da}{dt} + K_D \cdot \frac{2D}{r}\frac{d\alpha}{dt})] \qquad (5\text{-}14)$$

Regrouping terms,

$$\dot{E}_3 = -\frac{2\beta K_P}{r}a(x_2 + x_4) - \frac{2\beta DK_P}{r}\alpha(x_4 - x_2) - \frac{2\beta K_D}{r}\frac{da}{dt}(x_2 + x_4)$$

$$-\frac{2\beta DK_D}{r}\frac{d\alpha}{dt}(x_4 - x_2) \qquad (5\text{-}15)$$

From Eqn (5-9) and (5-11), we have

$$x_2 + x_4 = \frac{2}{r} \cdot \frac{da}{dt} \qquad (5\text{-}16)$$

From Eqn (5-10) and (5-12), we have

$$x_4 - x_2 = \frac{D}{r} \cdot \frac{d\alpha}{dt} \qquad (5\text{-}17)$$

Substituting Eqns (5-16) and (5-17) into (5-15):

$$\dot{E}_3 = -\frac{4\beta K_P}{r^2}a \cdot \frac{da}{dt} - \frac{2\beta D^2 K_P}{r^2}\alpha \cdot \frac{d\alpha}{dt} - \frac{4\beta K_D}{r^2}(\frac{da}{dt})^2 - \frac{2\beta D^2 K_D}{r^2}(\frac{d\alpha}{dt})^2$$

(5-18)

The third term $-\frac{4\beta K_D}{r^2}(\frac{da}{dt})^2$ and the fourth term $-\frac{2\beta D^2 K_D}{r^2}(\frac{d\alpha}{dt})^2$ are negative definite. The signs of the first and second terms are dependent on the system state, however. Notice that these two terms are $a\frac{da}{dt}$ and $\alpha\frac{d\alpha}{dt}$ times a constant, respectively. Aware of this, a Lyapunov function is proposed as

$$V = E + E_{AP} \tag{5-19}$$

where $E_{AP} = \frac{2\beta K_P}{r^2}a^2 + \frac{\beta D^2 K_P}{r^2}\alpha^2$

Here an additional artificial potential energy term E_{AP} is introduced in the Lyapunov function. The reason to introduce this term is that its derivative $\dot{E}_{AP} = \frac{4\beta K_P}{r^2}a\cdot\frac{da}{dt} + \frac{2\beta D^2 K_P}{r^2}\alpha$ cancels out the first term $-\frac{4\beta K_P}{r^2}a\cdot\frac{da}{dt}$ and the second term $-\frac{2\beta D^2 K_P}{r^2}\alpha\cdot\frac{d\alpha}{dt}$ of $\dot{E}_3$, so the sign of the derivative of the Lyapunov function can be determined.

The derivative of the Lyapunov function is

$$\dot{V} = \dot{E} + \dot{E}_{AP} = \dot{E} + \frac{4\beta K_P}{r^2}a\cdot\frac{da}{dt} + \frac{2\beta D^2 K_P}{r^2}\alpha \tag{5-20}$$

Substitute Eqns (5-13) and (5-18) into (5-20),

$$\dot{V} = -c(x_2{}^2 + x_4{}^2) - (x_2\tau_{cL} + x_4\tau_{cR}) - \frac{4\beta K_D}{r^2}(\frac{da}{dt})^2 - \frac{2\beta D^2 K_D}{r^2}(\frac{d\alpha}{dt})^2$$

(5-21)

From the above analysis it has been shown that all the terms in $\dot{V}$ are negative definite, so $\dot{V}$ is negative definite. Since V is positive definite, it can be concluded that the Cartesian controller is asymptotically stable for all gains of K_P and K_D.

5.3 Actuator Stability Analysis

5.3.1 Actuator Control Loop

In Chapter 4, a revised control scheme is proposed to simplify the control loop and shorten the control cycle. In the revised control scheme, the inverse Jacobian is moved out of the control loop, and thus the actuator controller is the same as the traditional controller with a step input. The schematic figure is shown in Figure 5-3.

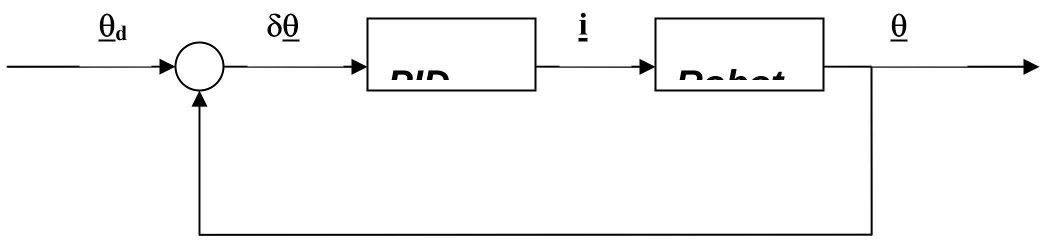

Figure 5-3: Actuator Control Loop

5.3.2 Analysis with Lyapunov Second Method

The actuator stability analysis is much more straightforward than the Cartesian stability, in that the actuator controller works in a similar way as the traditional PID controller with a step input. The methods in the linear feedback control theory, such as Root Locus and Nyquist Diagram, can be applied to the linearlized model. With the nonlinear parts existing in the model, Lyapunov second method is a more precise method in stability analysis, and the basic idea is identical to the Cartesian stability analysis.

CONTROL LAW

A constant input $\underline{\theta_d}$ is inputted to the control loop, and the error between $\underline{\theta_d}$ and $\underline{\theta}$ is acted upon by a PID controller to generate the input currents to the driving motors. Since the integral gain is very small, the control law can be written as:

$$\begin{aligned}\tau_L &= K_P \cdot (\theta_{dL} - \theta_L) + K_D \cdot \frac{d(\theta_{dL} - \theta_L)}{dt} \\ \tau_R &= K_P \cdot (\theta_{dR} - \theta_R) + K_D \cdot \frac{d(\theta_{dR} - \theta_R)}{dt}\end{aligned} \tag{5-22}$$

where θ_{dL} and θ_{dR} are the desired angles of the wheels.

Here saturation and digitalization are also ignored, since they will not affect the conclusion about stability.

LYAPUNOV FUNCTION

To search for a Lyapunov function to judge the actuator controller stability, first try the platform kinetic energy:

$$E = \frac{1}{2}mv^2 + \frac{1}{2}J\omega^2 \tag{5-23}$$

and then

$$\dot{E} = mv\dot{v} + J\omega\dot{\omega} \tag{5-24}$$

Following the same procedure as in the Cartesian stability, we substitute the state space equations into Eqn (5-24):

$$\begin{aligned}\dot{E} &= \frac{mr^2}{4}(x_2 + x_4)\left(-\frac{2c}{mr^2}\cdot x_2 - \frac{2c}{mr^2}\cdot x_4 - \frac{2}{mr^2}\cdot \tau_{cL} - \frac{2}{mr^2}\cdot \tau_{cR} + \frac{2\beta}{mr^2}\cdot \tau_L + \frac{2\beta}{mr^2}\cdot \tau_R\right) \\ &+ \frac{Jr^2}{D^2}(x_4 - x_2)\left(\frac{cD^2}{2Jr^2}\cdot x_2 - \frac{cD^2}{2Jr^2}\cdot x_4 + \frac{D^2}{2Jr^2}\cdot \tau_{cL} - \frac{D^2}{2Jr^2}\cdot \tau_{cR} - \frac{\beta D^2}{2Jr^2}\cdot \tau_L + \frac{\beta D^2}{2Jr^2}\cdot \tau_R\right)\end{aligned}$$

After simplification, we get

$$\dot{E} = -c(x_2{}^2 + x_4{}^2) - (x_2\tau_{cL} + x_4\tau_{cR}) + \beta(x_2\tau_L + x_4\tau_R) \tag{5-25}$$

With the similar analysis, both the first term and the second term of $\dot{E}$ can be shown as negative definite. So the analysis is focused on the third term

$$\dot{E}_3 = \beta(x_2\tau_L + x_4\tau_R) \tag{5-26}$$

Substitute the τ_L and τ_R in Eqn (5-26) with the control law (5-22):

$$\dot{E}_3 = \beta x_2[K_P \cdot (\theta_{dL} - \theta_L) + K_D \cdot \frac{d(\theta_{dL} - \theta_L)}{dt}] + \beta x_4[K_P \cdot (\theta_{dR} - \theta_R) + K_D \cdot \frac{d(\theta_{dR} - \theta_R)}{dt}] \tag{5-27}$$

Notice that $\underline{\theta}_d$ is a constant, and $x_1 = \theta_L$, $x_2 = \frac{d\theta_L}{dt}$, $x_3 = \theta_R$, $x_4 = \frac{d\theta_R}{dt}$, $\dot{E}_3$ can be simplified as

$$\dot{E}_3 = \beta K_P x_2(\theta_{dL} - x_1) - \beta K_D x_2{}^2 + \beta K_P x_4(\theta_{dR} - x_3) - \beta K_D x_4{}^2 \tag{5-28}$$

The terms $-\beta K_D x_2{}^2$ and $-\beta K_D x_4{}^2$ are negative definite, and the signs of $\beta K_P x_2(\theta_{dL} - x_1)$ and $\beta K_P x_4(\theta_{dR} - x_3)$ are undetermined. A similar Lyapunov function as (5-19) is proposed:

$$V = E + E_{AP} \tag{5-29}$$

where $E_{AP} = \frac{\beta K_P}{2}(\theta_{dL} - x_1)^2 + \frac{\beta K_P}{2}(\theta_{dR} - x_3)^2$

Here an additional artificial potential energy term E_{AP} is introduced in the Lyapunov function. The reason to introduce this term is that its derivative $\dot{E}_{AP} = -\beta K_P x_2(\theta_{dL} - x_1) - \beta K_P x_4(\theta_{dL} - x_3)$ cancels out the terms $\beta K_P x_2(\theta_{dL} - x_1)$ and $\beta K_P x_4(\theta_{dR} - x_3)$ in $\dot{E}_3$, so the sign of the derivative of the Lyapunov function can be determined.

The derivative of the Lyapunov function is

$$\dot{V} = \dot{E} + \dot{E}_{AP} = \dot{E} - \beta K_P x_2(\theta_{dL} - x_1) - \beta K_P x_4(\theta_{dL} - x_3) \tag{5-30}$$

Substituting Eqn (5-25) and (5-28) into (5-30):

$$\dot{V} = -c({x_2}^2 + {x_4}^2) - (x_2\tau_{cL} + x_4\tau_{cR}) - \beta K_D {x_2}^2 - \beta K_D {x_4}^2 \tag{5-31}$$

From the above analysis it has been shown that all the terms in $\dot{V}$ are negative definite, and thus $\dot{V}$ is negative definite. Since V is positive definite, it can be concluded that the actuator controller is asymptotically stable for all gains of K_P and K_D.

CHAPTER 6: TEST RESULTS

6.1 Introduction

Both the global and local planning and control schemes were tested to evaluate the performance of the barrel robot system. For the test of the barrel robot local controller, a typical motion of moving the barrel (1 meter forward and 1 meter left) was performed, and the dynamic response of the local controller was analyzed. This test was repeated for 100 times to determine the repetitive accuracy of the controller. To test the performance of the global planning and control scheme, an outdoor ground test was performed where the barrel robots were commanded to complete a typical task of moving into a wedge formation to close a highway lane. The desired and actual waypoints were recorded for each time step, and the performance of the system was evaluated by the observation of collision avoidance, as well as the comparison of the desired and actual waypoints.

6.2 Local Control Test

6.2.1 Dynamic Response

Since the local sensors on the barrel robots tend to accumulate error with the increase of the moving distance, usually the distance that the barrel moves independently is limited to less than 2 meters. Thus we pick up a typical motion of moving the barrel 1 meter forward and 1 meter to the left, and test the dynamic performance of the local controller.

As the object of the controller, the dynamic responses of the two wheels are shown in Figure 6-1.

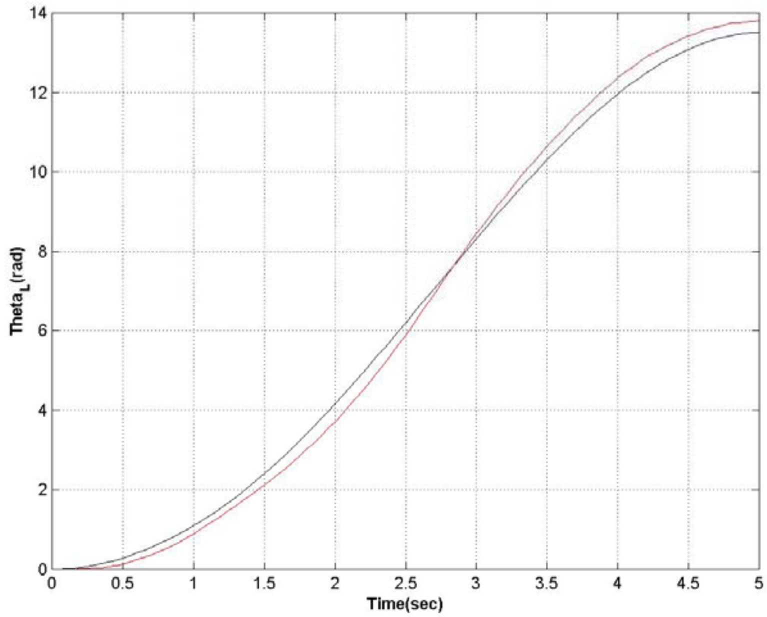

Figure 6-3(a): Dynamic Response of the Left Wheel

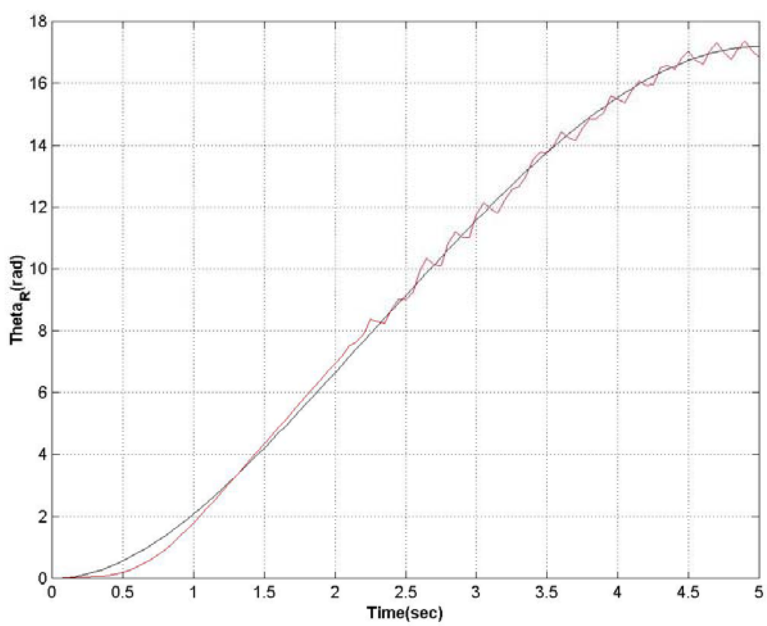

Figure 6-1(b): Dynamic Response of the Right Wheel

The output of the left wheel displayed a smooth curve, with a short rise time and a small overshoot. The right wheel, however, has a much more obvious oscillation in its output. The possible reasons include the backlash in the drive chain, and the saturation of the driving motors.

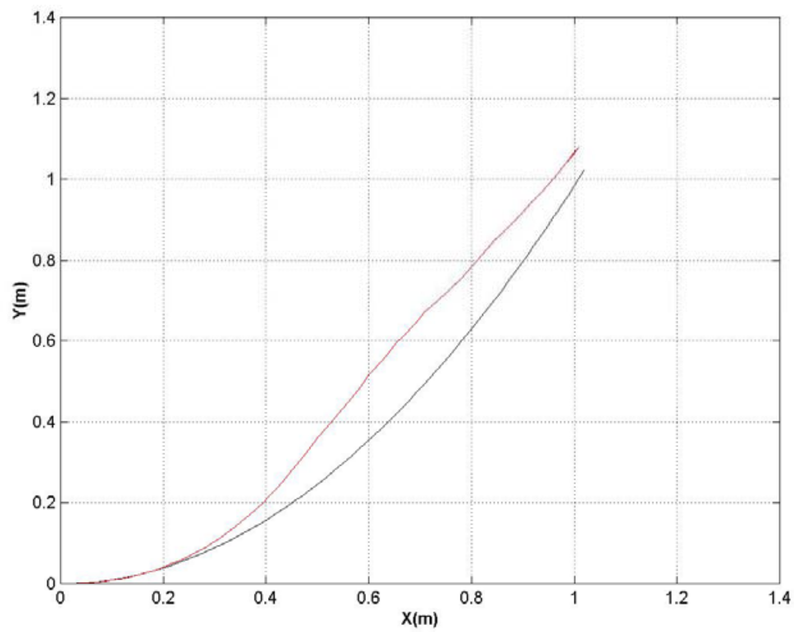

Figure 6-2: Desired and Actual Paths

The corresponding motion paths are shown in Figure 6-2. At the beginning the robot followed the desired trajectory, and then due to the overshoot of the right wheel the trajectory began to deviate to the left. The final localization error appeared to be less than 10 cm, which is less than 10% of the total moving length.

6.2.2 Repeated Experiments

Although single tests showed satisfactory results of localization precision, the same test was repeated for 100 times to test the controller's repetitive accuracy. The test results are shown in Figure 6-3.

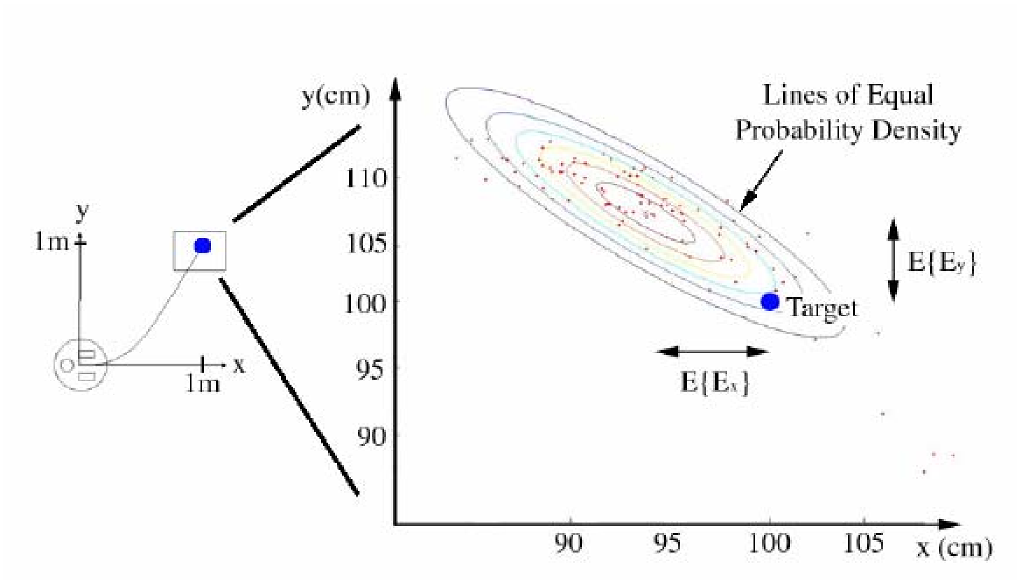

Figure 6-3: Test Results of One Hundred Experiments

The final position distribution, as shown in the right-hand side zoom-in figure of Figure 6-3, was recorded using a vision based external measurement device with an accuracy of <0.3 mm (Polaris, Northern Digital Inc.). The error is defined as the difference between the actual final position and the desired position. A Bivariate Normal Distribution was fit to the data to establish the expected values and standard deviations for the error in *x* and *y*. The probability density function is:

$$bn(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} e^{-(1/2)G(x,y;\mu_x,\mu_y,\sigma_x,\sigma_y)}$$

where

$$G(x, y; \mu_x, \mu_y, \sigma_x, \sigma_y) = \frac{1}{(1-\rho^2)}[\frac{(x-\mu_x)^2}{\sigma_x^{\ 2}} - 2\rho\frac{(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y} + \frac{(y-\mu_y)^2}{\sigma_y^{\ 2}}]$$

By fitting this model to the experiment data, the two dimensional distribution of the robot's final position is established. Lines of equal probability density are plotted as the ellipses shown in Figure 6-3.

The major axis of the ellipses is much greater than the minor axis, and from the distribution it can be said that the *x* and *y* errors are correlated with a correlation coefficient of 0.88. Furthermore, the angle of the major axis (54°) corresponds to the desired final orientation of the robot given by the planning parabola (63°). This correlation is a result of the sensors used by the control system to estimate the robots position. The wheel encoders are very accurate at measuring the forward motion of the robot but much less accurate at detecting small changes in the robot's heading, hence the correlation.

The distribution of the magnitude of the localization error is also studied. The magnitude of error ||E|| is given by $\sqrt{E_x^{\ 2} + E_y^{\ 2}}$. This distribution is continuous, lower bounded by zero, and has no upper bound, thus a Gamma distribution model is fit to the data. The probability density function of the Gamma distribution is:

$$f(x) = \begin{cases} \frac{\lambda^r}{\Gamma(r)} x^{r-1} e^{-\lambda x} & x \geq 0 \\ 0 & otherwise \end{cases}$$

where the Gamma function

$$\Gamma(r) = \int_0^\infty x^{r-1} e^{-x} dx \qquad r > 0$$

The plot of distribution is shown in Figure 6-4. This plot shows two distributions: 1) the error based on the position estimated from on-board sensors (i.e. where the robot "thinks" it stopped), and 2) the error from external measurements (i.e. where the robot actually stopped). Both are fit with the Gamma distribution.

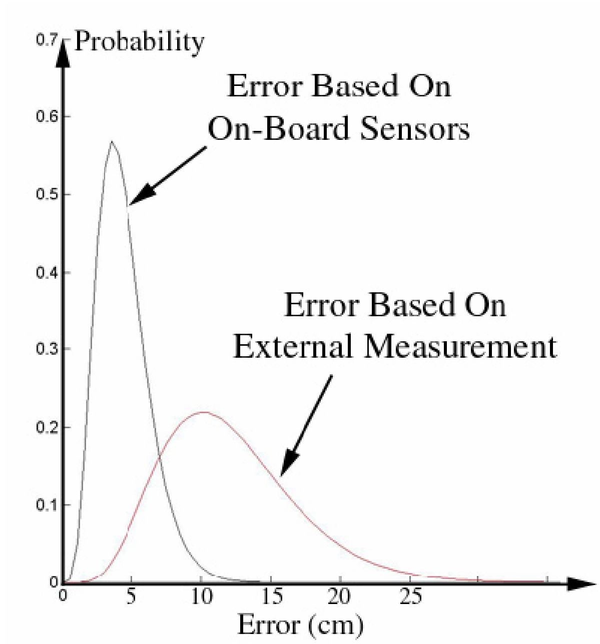

Figure 6-4: Distribution of the Magnitude of Error

The error based on on-board sensors had an expected value of 4.2 cm (variance = 1.48 cm^2), while the error based on external measurement had an expected value of 10.2 cm (variance = 9.6 cm^2). The corresponding shape factors *r* are 6.1 and 5.3, respectively. The plot shows that the robot "thought" it was very close to the target with a small distribution, but the actual error is larger with a bigger distribution. The robot needed to move ≈140 cm, so the actual error represents 7.3% of the total distance. A perfect control system could command the robot to zero error when measured with onboard sensors; however, the larger portion of the error could only be eliminated by improving the localization scheme. This portion of error is periodically eliminated using the global sensor.

4. 6.3 System Field Test

Both the global and local control schemes were tested in a realistic field environment. Five robots were randomly placed on the shoulder of the road and commanded to move into a taper or wedge, to close the right lane of a two-lane road, as shown in Figure 6-5. This is a very common task in many maintenance operations where the barrel robots could be useful (Traynowicz, 2002).

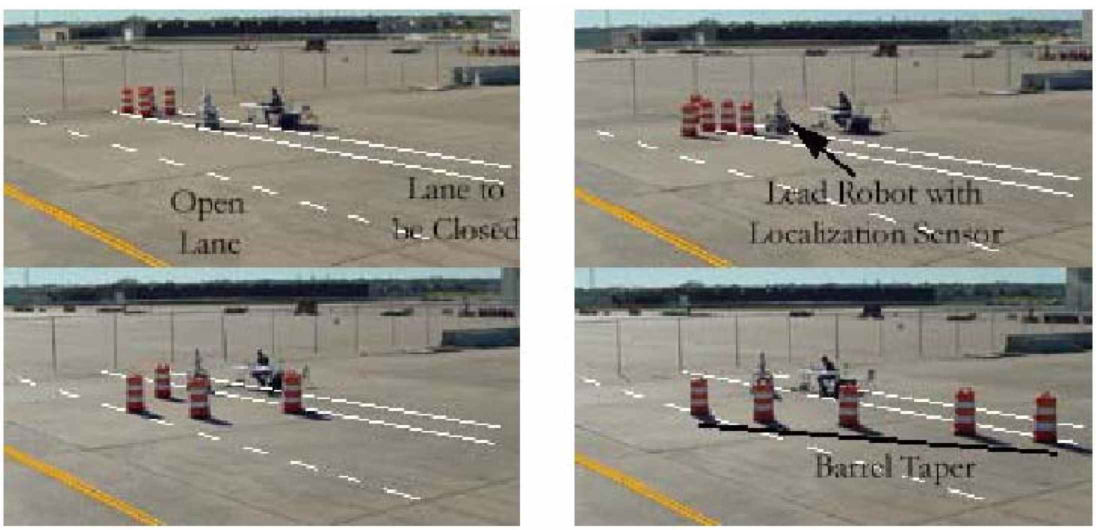

Figure 6-5: Lane Closure with Five Barrel Robots

Figure 6-6 shows the desired (dashed lines) and actual path (solid lines) for each robot during deployment. The symbols on each actual path represent where individual robots reached a waypoint and their position was updated. Lines connecting symbols only approximate robot motion between points.

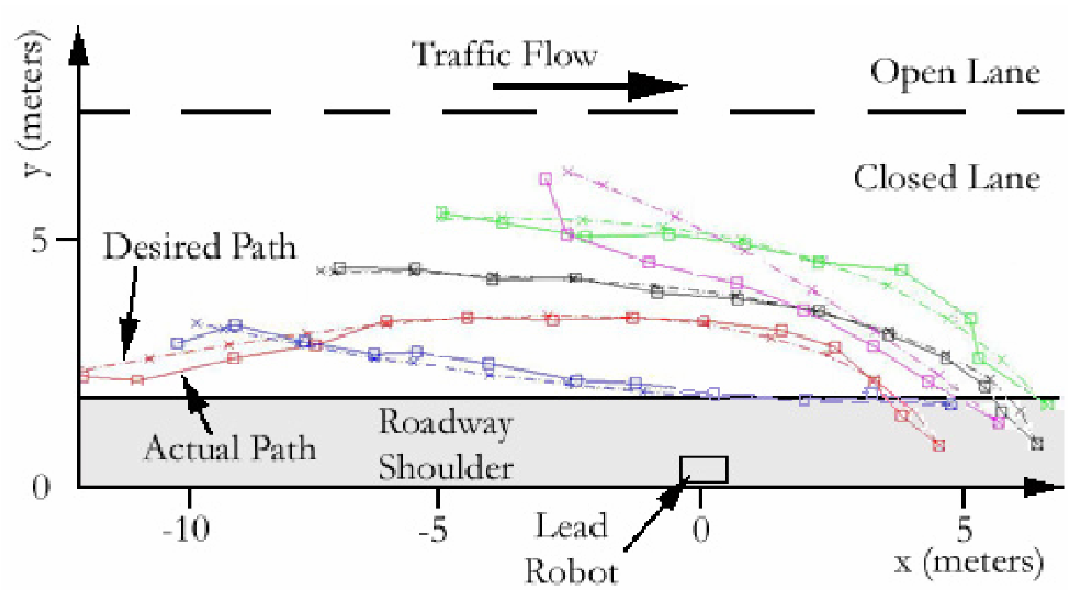

Figure 6-6: **Desired and Actual Paths During Field Test**

The global planner completed the required task. The global paths were generated with parabolas, and all the paths are within the work area. The waypoints were calculated thereafter along the paths, and the distances between waypoints at the same time point are all over the safe limit, which is set bigger than the barrel diameter to compensate for the robot control error.

The performance of the global and local scheme was also tested. The maximum deviation from the desired path was 23 cm and the maximum final error for all robots was 11 cm. This deviation may be caused by the error of the global localization error of the lead robot, and the local control error of the barrel robot. With regard to the application requirements, this accuracy is well within the requirements for barrel placement and exceeds the accuracy of current human deployment.

CHAPTER 7: CONCLUSIONS AND FUTURE WORK

7.1 Contributions of This Work

This thesis described the control of a system that uses mobile robots to transport safety markers for highway construction and maintenance. A hierarchical control system was proposed. Theoretical exploration was made in the area of robot motion planning and control theory to develop the planner and controller, and test results were presented which show the effectiveness of the control system.

7.1.1 Robot Motion Planning

The planning algorithm for the motion planning of a group of mobile robots was explored. The planning problem is constructed as moving a group of robots from the initial configuration to the final configuration, and a decouple method is proposed to solve the problem. In a heuristic way, the problem is decomposed into two sub problems: Path Planning Problem and Velocity Planning Problem.

The Path Planning Problem is to plan the global paths for the barrel robots. The main constraints considered are the nonholonomic constraint of the barrel robot and keeping the robots' motion within the work area. To generate smooth paths under these constraints, parabolic curves were used and the equations were derived in the Cartesian space.

With the global paths planned, the time-distance relationships along the paths were planned to get a complete motion plan. For each two barrel robots, a virtual coordination space (VCS) was constructed, and the obstacle areas plotted. The main obstacles considered are collision areas and invisible areas. A Tangent Graph method was used to find optimal paths in the VCS to achieve minimal temporal cost.

7.1.2 Mobile Robot Control

The control of the barrel robot was also explored in this work. Designed as a differential drive mobile robot, the barrel robot is subject to the nonholonomic constraint, or the non-integrable kinematic constraint. With the kinematic analysis taken, an inverse Jacobian Cartesian controller is developed to move the barrel robot to a desired position.

In the controller, a local planner generated waypoints as inputs to the control loop for each time step. Again, parabolas are used to get local paths, and the time-distance relationship is a 3rd polynomial with a continuous second order derivative(acceleration). For each time step, the desired waypoint was compared with the current position to get the error in the Cartesian space, and then this error was converted to the joint space with the method of Dead-Reckoning. The error in the joint space, which is the rotation angle of the wheel, was acted upon by a tradition PID controller to calculate the torque command to the motors to drive the robot. To simplify and accelerate the control loop, the Dead-Reckoning calculation was moved out of the control loop in the actual implementation, and the test results showed a satisfactory localization precision.

The stability of the barrel robot controller was also analyzed. Due to the nonholonomic constraint of the robot, it can only be stabilized to a manifold in the Euclidean space. Physically, it means that the robot can only converge to a certain position in the Cartesian space with the orientation uncontrolled. Both the Cartesian stability and the actuator stability were analyzed, and the Lyapunov stability theorem was used to reach the conclusion.

7.2 Future Work

Although the planning and control in both the global and local level have been completed in this work, there are still areas to be considered.

Further application of the barrel robot system in the highway construction and maintenance is yet to be explored. The planning and control algorithm explored in this work are mainly for the static applications. The mobile applications, such as painting, are the potential applications for further exploration.

For these application, new planning and control algorithm need to be developed, such as speed planning and control. The objects of the planning and control in this work are positions: robots are required to move to their desired positions; while in mobile applications the robots are required to move with a certain velocity, either following a leader or staying in a formation.

The improvement of the performance of the current system is another area of future work. For the global planning, global paths can be modified to avoid the small obstacles in the work area, or some coupled algorithms can be used to realize strict optimality. For the local control of barrel robot, as mentioned before, the localization error is composed of two parts: the controller dynamic error, and the error incurred by the onboard sensors. Correspondingly, improved nonlinear controller can be developed to reduce the dynamic error, and more precise onboard sensors can be used to reduce the sensing error.

5.

6. REFERENCES

Anon. “The 11 Best Ways to Improve Work Zone Safety.” Better Roads v60, n7 (1990): 20-23.

Belta, C., Kumar, V., “Motion Generation for Formations of Robots: A Geometric Approach,” IEEE International Conference on Robotics & Automation, pp. 1245-1250, May 2001.

Bloch, Anthony M., McClamroch, N. Harris, “Control of Mechanical Systems with Classical Nonholonomic Constraints”, Proceedings of the 28th Conference on Decision and Control, December 1989, pp.201-205.

Borenstein, J., Koren, Y., "The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots," IEEE Trans on Robotics and Automation, Vol. 7, No. 3 June 1991.

Brockett, R.W., “Asymptotic stability and feedback stabilization”, Differential Geometric Control Theory, pp. 181-208, Birkhauser, 1983.

Brooks, R., "A Robust Layered Control System for a mobile robot", IEEE Transactions on Robotics and Automation, Vol. 2, No. 1, 1986.

Canudas de Wit, C. and Sordalen, O. J., “Exponential Stabilization of Mobile Robots with Nonholonomic Constraints”, IEEE Conference on Decision and Control, Brighton, England, December 1991.

Desai, J., Ostrowski, J., Kumar, V., “Controlling formations of multiple mobile robots”, IEEE International Conference on Robotics & Automation, pp. 2864-2869, May 1998.

Egerstedt, M., Hu, X., “Formation Constrained Multi-Agent Control”, IEEE International Conference on Robotics & Automation, pp. 3961-3966, May 2001.

Farritor S., Hacot H. and Dubowsky S., “Physics-Based Planning for Planetary Exploration”, IEEE Intl. Conference on Robotic and Automation, 1998.

Feng, X., and Velinsky, S. A., “Distributed Control of a Multiple Tethered Mobile Robot System for Highway Construction and Maintenance”, Microcomputers in Civil Engineering, Vol. 12, pp. 383-392, November, 1997.

Fierro R., Das A.K., Kumar V., Ostrowski J.P., “Hybrid Control of Formations of Robots”, IEEE Intl. Conference on Robotic and Automation, pp. 3672-3677, May 2001.

Gumtau, R. “Richard Gumtau on Traffic Control and Work Zones.” Better Roads v63, n9, 1993.

Gentili, F., Martinelli, F., “Robot group formations: a dynamic programming approach for a shortest path computation,” IEEE Intl. Conf. on Robotics & Automation, pp. 3152-57, 2000.

Gold, T., Archibald, J., Frost, R., “A Utility Approach to Multi-Agent Coordination”, IEEE International Conference on Robotics & Automation, pp. 2052-2057, April 2000.

Hood, F. Hoff, W., King, R., "Evaluation of an Interactive Technique for Creating Site Models from Range Data", ANS Meeting on Robotics and Remote Systems, 1997.

Hui, Raymond, Goldenberg, Andrew A., “Stability of Non-holonomic Systems”, IEEE/RSJ International Workshop on Intelligent Robots and Systems, November 1991, pp.1260-1265.

Kang, W., Xi, N., Sparks, A., “Formation Control of Autonomous Agents in 3D workspace”, IEEE International Conference on Robotics & Automation, pp. 1755-1760, April 2000.

Kant, K., Zucker, Steven W., “Toward Efficient Trajectory Planning: the Path-Velocity Decomposition”, the International Journal of Robotics Research, Vol.5, No.3, Fall 1986, pp. 72 – 89.

Khatib, O. "Real-time obstacle avoidance for manipulators and mobile robots", International Journal of Robotics Research, 5 (1), 90-98, 1986.

Lasky, T., Ravani, B., “Sensor-based Path Planning and Motion Control for a Robotic System for Roadway Crack Sealing”, IEEE Trans. on Control Systems Tech. v8, n4 (2000): pp. 609-622.

Liu, Yun-Hui and Arimoto, Suguru, “Proposal of Tangent Graph and Extended Tangent Graph for Path Planning of Mobile Robots”, IEEE International Conference on Robotics & Automation, April 1991, pp. 312 – 317.

Liu, Yun-Hui and Arimoto, Suguru, “Computation of the Tangent Graph of Polygonal Obstacles by Moving-Line Processing”, IEEE International Conference on Robotics & Automation, December 1994, Vol. 10, No. 6, pp. 823 – 830.

Mohri, A., Yamamoto, M., Marushima, S., “Collision-Free Trajectory Planning for Two Manipulators Using Virtual Coordination Space”, IEEE Intl Conf on Robotics & Automation, pp. 674-9, 1993.

Murphy, R., Marzilli, A., Hughes, K., “When to explicitly Re-plan Paths for Mobile Robots”, IEEE International Conference on Robotics & Automation, pp. 3519-25, 1997.

Ollis, M., Stentz, A., “Vision-Based Perception for an Autonomous Harvester”, Proceedings of IEEE/RSJ Intl. Conf. on Intelligent Robotic Systems, Vol. 3, 1997, pp. 1838 - 1844.

Palopoli, L., Conticelli, F., Allotta, B., “Multi-level Stabilizing Control of an Nonholonomic Vehicle and Its Discrete-Time Multirate Implementation”, IEEE Conference on Robotics and Automation, April 2000, pp. 1830-1835.

Ravani, B., and Velinsky, S.A. West, T.H. “Requirements for Application of Robotics and Automation in Highway Maintenance and Construction Tasks”, ASCE Specialty Conference on Robotics for Challenging Environments (1994): pp. 356-364.

Rihani, Rami A., and Bernold, Leonhard E. “Telerobotic Pavement Marker Application.” ASCE Conference on Robotics for Challenging Environments (1996): pp. 171-177.

Shah, S. “New Work Zone Safety Devices.” Proceedings of the ASCE 3rd International Conf. on Applications of Advanced Technologies in Transportation Engineering (1993): pp. 308-315.

Sordalen, O. J. and Canudas de Wit, C., “Exponential Control Law for a Mobile Robot: Extension to Path following”, IEEE Conference on Robotics and Automation, Nice, France, May 1992.

Stilwell, D., Bishop, B., “A Framework for Decentralized Control of Autonomous Vehicles”, IEEE International Conference on Robotics & Automation, pp. 2358-2363, April 2000.

Todt, E., Raush, G., Suarez, R., “Analysis and Classification of Multiple Robot Coordination Methods”, IEEE Intl. Conference on Robotics & Automation, pp. 3158-3163, April 2000.

Velinsky, S.A., “Heavy Vehicle System for Automated Pavement Crack Sealing”, Heavy Vehicles Systems, International Journal of Vehicle Design, Vol. 1, No.1, pp. 114-128, 1993.

West, Thomas H., and Velinsky, Steven A., and Ravani, Bahram, “Advanced Highway Maintenance and Construction Technology Applications”, TR News (1995): pp. 17-23.

White, T., “Evolving Automation in the Asphalt Paving Industry”, TR News n176 (1995): pp. 4-6.

Woo, D., “Robotics in Highway Construction”, Public Roads v58, n3 (1995): p26-30.

Ye, W., Vaughan, R., Sukhatme, G., Heidemann, J., Estrin, E., Mataric, M., “Evaluating Control Strategies for Wireless-Networked Robots Using an Integrated Robot and Network Simulation”, IEEE Intl. Conference on Robotics & Automation, pp. 2941-2947, 2001.

Zhou, Tong, and West, Thomas, “Assessment of the State-of-the-Art of Robotics Applications in Highway Construction and Maintenance”, Proc. of the 2nd Intl. Conf. on Applications of Advanced technologies in Transportation Engineering (1991): pp. 56-60.