California AHMCT Program
University of California at Davis
California Department of Transportation

TRACKING CONTROL ALGORITHMS FOR THE TETHERED MOBILE ROBOT

YULIN ZHANG
STEVEN A. VELINSKY

AHMCT Research Report
UCD-ARR-94-06-30-01

Interim Report of Contract
IA65Q168-MOU 92-9

June 30, 1994

TRACKING CONTROL ALGORITHMS FOR THE TETHERED MOBILE ROBOT

Interim Report

YULIN ZHANG
Department of Astronautics and Mechanics
Changsha Institute of Technology, Changsha, Hunan, China
Currently, Visiting Research Scientist
Department of Mechanical and Aeronautical Engineering
University of California, Davis, California

STEVEN A. VELINSKY
Department of Mechanical and Aeronautical Engineering
University of California, Davis, California

June 30, 1994

Advanced Highway Maintenance and Construction Technology Program

DISCLAIMER/DISCLOSURE

"The research reported herein was performed as part of the Advanced Highway Maintenance and Construction Technology Program (AHMCT), within the Department of Mechanical and Aeronautical Engineering at the University of California, Davis and the Division of New Technology and Materials Research at the California Department of Transportation. It is evolutionary and voluntary. It is a cooperative venture of local, state and federal governments and universities."

"The contents of this report reflect the views of the author(s) who is (are) responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the STATE OF CALIFORNIA or the FEDERAL HIGHWAY ADMINISTRATION and the UNIVERSITY OF CALIFORNIA. This report does not constitute a standard, specification, or regulation."

ABSTRACT

The purpose of this research is to provide a theoretical basis for the tracking control of a Tethered Mobile Robot in highway maintenance application. The Tethered Mobile Robot is a mobile robot with two independent driven wheels. The tracking control algorithms for this type of mobile robot are thoroughly and systematically studied in this report. The tracking control algorithms are developed both on the basis of kinematic models and dynamic models. The main effort is to develop tracking control algorithms with strong robustness to uncertainties, such as system perturbations and external disturbances.

Kinematic models with wheel slippage are developed. The slippage influence is treated to be equivalent to parameter uncertainties in the system. An orientation equation for this kind of wheeled mobile robot is developed. It reveals the inherent relation between position tracking and orientation tracking. The influence of the location of the tracking point on the tracking ability is thoroughly studied in this report. Two globally stable tracking control algorithms are developed according to the tracking point location. A kinematic robust tracking control algorithm is developed on the basis of the kinematic model with uncertainty.

On the basis of a full dynamic model with detailed tire-ground contact model, reduced order dynamic models are developed. Uncertainty influence on the dynamic system is analyzed and a match condition is proven. An exponential position dynamic tracking control algorithm is developed. The control algorithm is based on the reduced order dynamic model and possesses strong robustness to system uncertainties. Variable structure control theory is used to construct a variable structure dynamic tracking control algorithm for the wheeled mobile robot. Because the match condition is hold for the uncertainties in the dynamic system, the exponential tracking control performance of the variable dynamic tracking control algorithm is invariant to system uncertainties, such as parameter perturbations, external disturbances, and unmodeled system dynamics.

The performances of the tracking control algorithms are simulated by numerical methods. It is worth pointing out that the full dynamic model with a detailed tire model is used in all the simulations for the dynamic tracking control algorithms. The path planning problem related to highway maintenance is discussed. Since the tracking control algorithm has trajectory tracking features, a speed manipulation algorithm for mobile robot is proposed to produce a reference trajectory along the reference path.

From a tracking control point of view, the Tethered Mobile Robot is no more than a ground vehicle with two independent driven wheels. As such, the tracking control algorithms presented in this report are valid for any differentially steered wheeled mobile robots or vehicles.

EXECUTIVE SUMMARY

A Tethered Mobile Robot (TMR) is being developing for highway maintenance, such as highway crack sealing (Winters and Velinsky, 1992) in the Advanced Highway Maintenance and Construction Technology (AHMCT) Center at the University of California, Davis. The operations proposed for the TMR in highway maintenance are basically tasks along a desired path, such as a highway crack. Because of the real highway working environment, the payload and speed of the TMR are much higher than that of a common wheeled mobile robot. Two of the vital requirements for the TMR design and development are

- The tooling/equipment can accurately track a reference path;
- The tracking ability is robust to uncertainties, such as unmodeled dynamics, parameter perturbations, and external payload disturbances.

The above two requirements are all related to the performance of the tracking control system of the TMR. The TMR is basically a mobile robot with two differentially driven wheels. The tracking control algorithm of this type of mobile robot has been studied extensively in the last decade. However, several fundamental problems are still not well solved. These include the relation between position tracking and orientation tracking, the influence of the location of tracking point on the tracking ability, and the robustness of the tracking control algorithm. These problems have critical influence on the tracking control performance of the wheeled mobile robot.

To meet the requirements for the TMR tracking control performance, fundamental modeling and control algorithms have been studied in the last several months in the AHMCT Center, and novel results on the tracking control algorithms of the wheeled mobile robot have been obtained. In this report, we present the main theoretical progress on the modeling and tracking control algorithms in our research.

The kinematic model is the basis for the tracking control of wheeled mobile robots. Kinematic models of mobile robots have been studied by many authors. However, for the purposes of our work, it is necessary to have a detailed study on the kinematic model of mobile robots with two

independent wheels. In this report, we first describe the kinematics equations of the mobile robots, and then study the kinematic features and their influence on the tracking control performance. The tracking control performance of a mobile robot with two independent driven wheels is strongly determined by its kinematics features and the location of the tracking point on the mobile robot as well. Through the analysis of the kinematics characteristics, fundamental tracking control algorithm structures for this kind of mobile robot are proposed. This includes the tracking variable assignment, the tracking singularity and position-orientation tracking decoupling problems.

For the problem of accurate path tracking control, a kinematic or dynamic model based control algorithm is essential. This is because the dynamic model of the mobile robot can have considerable uncertainty in modeling the driven force and payload, and its computational complexity is relatively high. Kinematic models play an important role in control algorithm development. Global convergence tracking control algorithms are developed for two different cases, for the tracking point on and off the baseline. When the tracking point is not on the baseline, only the position of the tracking point can exactly track a reference path, and it is impossible to exactly track the heading direction of the mobile robot. When the tracking point is located on the baseline, both the position of the tracking point and the orientation of the mobile robot can exactly track a reference path. Numerical simulation examples are provided which illustrate the performance of the control algorithms developed herein.

Since the TMR is being developed for application in highway maintenance, the accurate tracking ability of the TMR to a reference path is very important for most maintenance operations. However, since the working environment of the TMR will be in an unstructured environment (i.e., highway), there are many uncertainties that exist. Among them are the changes of roadway surface conditions, the varying operation payloads, the different maintenance tasks, etc. These uncertainties combined with the relative high speeds will result in wheel slippage and a change in radius of the pneumatic tires. These uncertainties will appear in the mobile robot kinematics equation as uncertainties of effective wheel radius. Therefore, to meet the accurate tracking

requirements, any kinematics based tracking control algorithm must have enough robustness to overcome the tracking performance deterioration caused by the kinematic uncertainties.

We thus develop a mobile robot kinematic tracking control algorithm based on the kinematic model with uncertainties. This algorithm is developed to guarantee the exponential tracking control property when uncertainties exist, and the algorithm possesses the similar calculation complexity as that of a PID based algorithm and has the advantage of being suitable for real time application. Through numerical simulation, it is determined that the robust control algorithm not only possesses the robustness ability under the condition of uncertainty, but also improves the tracking control performance under the condition of an exact kinematics model.

Based on the dynamic modeling work of Boyden and Velinsky (1993), it was believed that a kinematics based control algorithm is only valid for very low speed and very low payload robots. Therefore, a dynamic model based control algorithm is necessary for accurate path tracking control, and one is developed herein. We describe the full dynamic model of the mobile robot, and introduce the concept of a perfect dynamic model. The influence of the uncertainties, such as unmodeled dynamics, slippage, payload variation, parameter perturbation, etc., on the system dynamics is analyzed. Since exact position tracking control is very important for the Tethered Mobile Robot, we make a coordinate transformation to get the tracking point position dynamic model. As a fundamental basis of the robust tracking control algorithm, the matching condition for the mobile robot uncertainty in the dynamic model is proven.

Additionally, we take the full dynamic model as a basis, and an exponential tracking control algorithm is then developed based on the perfect dynamic model, and robustness to uncertainties is guaranteed. In this way, the robust tracking control algorithm gives high tracking control performance when unmodeled dynamics and external disturbances exist.

The main difficulty in the mobile robot's dynamic model based control algorithm development and application is the influence of uncertainty. This is because real-time operation limits the complexity of the dynamic model, while it is impossible to accurately describe the motion with low order and simple equations. Robustness is the most important property for the dynamic model

based control algorithm. A variable structure control system has the feature of sliding mode invariance to both system perturbations and external disturbances (Utkin 1978, 1992), and therefore, it is a suitable design method for the dynamic model based tracking control of mobile robots. Thus, we first review the main aspects of variable structure control, and then produce a variable structure control algorithm for the mobile robot's tracking control problem.

The path planning problem related to highway maintenance is also discussed. Since the tracking control algorithm has trajectory tracking features, a speed manipulation algorithm for mobile robot is proposed to produce a reference trajectory along the reference path.

From a tracking control point of view, the Tethered Mobile Robot is no more than a ground vehicle with two independent driven wheels. As such, the tracking control algorithms presented in this report are valid for any differentially steered wheeled mobile robots or vehicles.

TABLE OF CONTENTS

APPENDIX A - MATLAB SIMULATION CODE

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

A Tethered Mobile Robot (TMR) is being developing for highway maintenance, such as highway crack sealing (Winters and Velinsky, 1992) in the Advanced Highway Maintenance and Construction Technology (AHMCT) Center at the University of California, Davis. The operations proposed for the TMR in highway maintenance are basically tasks along a desired path, such as a highway crack. Because of the real highway working environment, the payload and speed of the TMR are much higher than that of a common wheeled mobile robot. Two of the vital requirements for the TMR design and development are

- The tooling/equipment can accurately track a reference path;
- The tracking ability is robust to uncertainties, such as unmodeled dynamics, parameter perturbations, and external payload disturbances.

The above two requirements are all related to the performance of the tracking control system of the TMR. The TMR is basically a mobile robot with two differentially driven wheels. The tracking control algorithm of this type of mobile robot has been studied extensively in the last decade. However, several fundamental problems are still not well solved. These include the relation between position tracking and orientation tracking, the influence of the location of tracking point on the tracking ability, and the robustness of the tracking control algorithm. These problems have critical influence on the tracking control performance of the wheeled mobile robot.

To meet the requirements for the TMR tracking control performance, fundamental modeling and control algorithms have been studied in the last several months in the AHMCT Center, and novel results on the tracking control algorithms of the wheeled mobile robot have been obtained. In this report, we present the main theoretical progress on the modeling and tracking control algorithms in our research.

In this chapter, we first introduce the configuration features of the TMR related to the tracking control problem; then, we review the main previous works on the wheeled mobile robot tracking control problem; and finally, we consider the tracking control problem statement for the TMR.

1.2 Tethered Mobile Robot Configuration

The TMR is configured to be a self-propelled wheeled mobile robot working in proximity to a support vehicle (see Fig. 1.1).

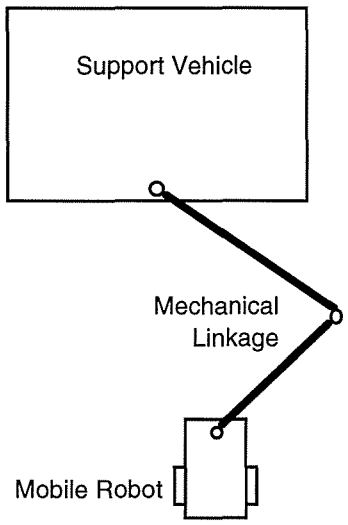


Fig. 1.1 Tethered Mobile Robot Configuration

The TMR has two differentially driven wheels and a castor. The driven wheels have pneumatic tires and are driven by two D. C. motors. A two degree of freedom planar mechanical linkage is designed to link the mobile robot and the vehicle for purposes of power, materials, and allow for the measurement of the position and orientation of the robot relative to the support vehicle with high accuracy. The support vehicle contains the associated maintenance supplies, power supply, and the primary maintenance operation sensing devices. The mobile robot is supplied with the necessary maintenance materials and power through the tether to the support vehicle. Two linear

transducers and an encoder are used to determine the relative position and orientation of the robot to the support vehicle.

From the tracking control point of view, the important feature of the TMR configuration is that its position and orientation can be measured with high accuracy. This makes accurate and robust tracking control possible for this wheeled mobile robot.

1.3 Previous Work on Wheeled Mobile Robot Tracking Control

Extensive research has been done in the last decade on the tracking control of wheeled mobile robots. Most of the authors take a kinematic model as the basis. Some authors studied the dynamic tracking control problem based on an ideal wheel ground contact model. Because of the nonholonomic feature of the wheeled mobile robot's kinematics constraints, many authors concentrated on the kinematic controllability and path planning problem. In this section, we first review the previous work on the kinematic and dynamic modeling of the differentially driven wheeled mobile robots, then, we briefly outline main results on the tracking control algorithm of the differentially driven mobile robots.

1.3.1 *Kinematic Modeling*

Many authors contributed to the kinematic modeling of wheeled mobile robots. Here, we only review two comprehensive research results in this area.

Muir and Neuman (1987) studied the kinematic modeling of a general wheeled mobile robot. Different wheel configurations, such as conventional, omnidirectional, and ball wheels, are considered. Basic assumptions used are as following:

- The wheeled mobile robot moves on a planar surface;
- The mobile robot does not contain flexible parts;
- The friction at the contact point between the wheel and the surface is sufficiently large.

Wheeled mobile robot coordinate systems and their transformation matrices are discussed. Matrix coordinate transformation algebra is described. Position kinematics, velocity kinematics, and acceleration kinematics for general wheeled mobile robot configurations are developed. Kinematic

characteristics of wheeled mobile robots, such as the robot mobility, kinematics solution, inverse solution, robot actuation, and robot sensing are studied. Application of the kinematic model to a wheeled mobile robot Uranus was shown as an illustrative example. Rotational slip was treated by means of least-squares method with sensed kinematics data.

Alexander and Maddocks (1989) also comprehensively studied the kinematics and inverse kinematics of wheeled mobile robots. They investigated the slippage by means of minimization of a non smooth convex dissipation functional that is derived from Coulomb's law of friction.

In the kinematic models discussed above a fundamental assumption is that there is perfect wheel-surface contact, although these authors treated the wheel slippage to a varying extent. Muir and Neuman (1987) used sensed motion data to the solution of the kinematics equations, therefore, the slippage influence can be treated by means of least-square method. Alexander and Maddocks (1989) were trying to estimate the slippage on the basis of quasi-static motion assumption. The common point of their methods is trying to determine the slippage quantities in the kinematic models.

Wheel slippage is completely determined by the wheeled mobile robot dynamics. It is impossible to solve the slip problem based on kinematics. From a tracking control point of view, it is necessary to have the kinematics equations including slippage terms but it is not necessary to determine the slippage quantities through the kinematic relations.

Muir and Neuman (1987) and Alexander and Maddocks (1989) concentrated on the general mathematics descriptions for all types of wheeled mobile robot configurations. For the specific configuration of a mobile robot with two differentially driven wheels, which we are concerned with in our research, a deep understanding of the special kinematic features is required for the purpose of accurate path tracking control.

1.3.2 *Dynamic Modeling*

There are few tracking control algorithms for wheeled mobile robots that are based on full dynamic models. Some authors used simplified dynamic models based on the assumption of perfect wheel-surface contact conditions in their control algorithm development.

Hamdy and Badreddin (1992) developed a dynamic model of wheeled mobile robot for navigation and control purposes. The robot is a cart with two differently driven wheels and one castor. There is a rotational tower mounted on the cart. The dynamic model includes the vehicle motion dynamics and driven motors' dynamics. A basic assumption was made is that there exists only longitudinal slippage and no lateral slippage. Longitudinal slip was described by kinematic relations. Two types of dynamic equations for the wheeled mobile robot are developed, one is for the case of no slip and another is for the case of slip presence. In simulation, the wheel-ground contact forces are described by nonlinear functions of the corresponding slip speeds. This includes sticking, dry, and viscous friction. The wheel roll resistance is modeled as a constant coefficient multiplied by half of the robot weight. A slip speed limit was tested to determine whether or not a slip model should be used in the simulation process.

Boyden and Velinsky (1993) developed dynamic models for wheeled mobile robots towards the application of the Tethered Mobile Robot simulation. Full vehicle dynamics with Dugoff's tire friction model are included in the dynamic models. Since the Tethered Mobile Robot is developed to operate in a real highway maintenance environment with relatively high motion speed and operation payload, wheels with pneumatic tires are selected in order to get higher driven forces. For such a wheel configuration and operation conditions, it is very important to use a more realistic tire model in the dynamic modeling. Full dynamic models and simplified dynamic models for both conventionally steered and differentially steered mobile robots are described. The effectiveness of the dynamic models are illustrated by numerical simulations. Tire driven forces are calculated by Dugoff's friction circle method. These forces are nonlinear functions of longitudinal and lateral speed of the vehicle. The total wheel payload is assumed to be the vehicle weight.

Kinematic models cannot effectively describe wheel slippage. However, the influence of slippage must be considered in the tracking control problem of wheeled mobile robots in the case of high motion speed and payload, such as Tethered Mobile Robot. Therefore, a full dynamic model with detailed tire-ground forces model is necessary. However, a full dynamic model with accurate tire-ground force description is far too complex for any tracking control algorithm. This is

because Dugoff's friction circle method is not a closed form description. Even in the simplified dynamic model developed by Hamdy and Badreddin (1992), limited test procedures must be performed in the simulation process to decide which model, slip or no slip, could be used. The slip model possess higher order than the no slip model. From the tracking control point of view, a reduced order dynamic model with clear uncertainty influence is necessary. This dynamic model should have the simplicity of a perfect dynamic model. The influence of the uncertainties could be well treated by a robust design method. In this way the tracking control performance can be guaranteed.

1.3.3 *Kinematic Model Based Tracking Control Algorithms*

Based on the kinematic model of the mobile robot with two driven wheels, controllability, motion planing, and related control problems are studied by many authors, for example, Kanayama, Nilipour, and Lelm (1988), Kanayama, Kimura, Miyazaki, and Noguchi (1990), d'Andréa-Novel, Bastin, and Campion (1991), Badreddin (1992), Pomet, Thuilot, Bastin, and Campion (1992), Wit and Sørdalen (1992), Sørdalen and Wit (1993), Lee and Williams (1993), Walsh, Tilbury, Sastry, Murray, and Laumond (1994), and Reister and Pin (1994).

Kanayama, Nilipour, and Lelm (1988) developed a PID based tracking control algorithm for mobile robots with two driven wheels. The tracking point is set to be the center of the baseline. The vehicle posture errors are translated into the errors in the vehicle fixed coordinate system; that is, the errors in the forward and lateral directions and rotation, respectively. The translated error components are then PID filtered to generate the desired vehicle longitudinal speed and angular rotation speed. According to the kinematic relations, the control inputs of the kinematic based control algorithm (i.e., the wheel velocities of the left and right wheel) are calculated from the desired vehicle longitudinal speed and angular rotation speed. The vehicle posture and velocity are measured by means of the dead-reckoning method. Kanayama and Kimura, et al. (1990) developed a nonlinear feedback tracking control algorithm on the basis of the translated vehicle posture error. The tracking point is on the center of the baseline. Lyapunov's direct method was used to prove the stability of the tracking control algorithm. Unfortunately, the Lyapunov function

selected is not quadratic on the orientation error, and an unnecessary assumption was made to guarantee the non-positive time derivative of the Lyapunov function along the trajectory of the error equations. Effects of control parameters on the tracking performance and velocity or acceleration limits are discussed. The efficiency of the tracking control algorithm was proven by experimental results.

Lee and Williams (1993) developed a tracking control algorithm for mobile robots with two driven wheels. An important feature in the control system is that the tracking point is not on the baseline. The robot posture error in the world coordinate system is translated into the posture error in the vehicle fixed coordinate system. The desired rotation speed of the two driven wheels are calculated from the vehicle kinematics relations. Since the tracking point is not on the baseline, there exist three kinematic constraint equations on the two driven wheel speeds. It is well known that the lateral speed equation and the rotation speed equation of the mobile robots are not independent. Therefore, the two control inputs are determined from the three posture errors by using the least-squares method. The driven motors are controlled by a PID control algorithm. The authors' simulation results indicated that the convergence speed of the lateral tracking error is very slow. Since the position and orientation errors are not independent and a least-squares method is used to determine the desired control inputs, the position and orientation tracking control errors of the algorithm are strongly coupled and the tracking control performance is quite poor.

Walsh and Tilbury, et al. (1994) used a Taylor series expansion to get time-varying linearized tracking error equations, and Lyapunov's direct method was used to construct a linear time-varying feedback law. The control algorithm is proven to be exponentially stable. However, the convergence of the proposed control algorithm is directly determined by the specific reference trajectory, and the time-varying linear feedback control law causes high computational complexity in application.

Wit and Sørdalen (1992) produced a circular transformation to the wheeled mobile robot's kinematic equations, and an exponential control algorithm is developed on the transformed space to

control the robot to a given position and orientation. Sørdalen and Wit (1993) lately extended the method to the path following problem. A piecewise smooth feedback control law is proposed.

The location of the tracking point on the mobile robot is of critical influence on the tracking control property. However, this fact is not widely considered by most authors. It is impossible to track a reference trajectory both in orientation and position while the tracking point is not on the baseline. For industrial application of wheeled mobile robots, tracking control algorithm robustness is of vital importance; however, very few authors have considered this area.

1.3.4 *Dynamic Tracking Control Algorithm*

Kinematic model based tracking control algorithms are believed to be valid only for very low speed wheeled mobile robots. However, because of the difficulty to accurately model the wheel-surface contact forces, little progress has been made on a full dynamic model based tracking control algorithm.

d'Andréa-Novel, Bastin, and Campion (1992) studied the dynamic feedback linearization of wheeled mobile robots. Pure rolling and no slipping assumptions are made for the dynamic model. Campion, d'Andréa-novel, and Bastin (1991) studied the controllability and feedback stability of mobile robots based on a perfect dynamic model. Samson (1991) studied the torque control problem based on a perfect dynamic model also. Unfortunately, all noted research are restricted to very general mathematical descriptions and provide little contribution toward practical design and application.

Sarkar, Yun, and Kumar (1994) developed a dynamic tracking control algorithm for mechanical systems with both holonomic and nonholonomic constraints, and applied the control algorithm to wheeled mobile robots. The dynamic model is also a simplified perfect dynamic model. Numerical simulations are made to show the effectiveness of the control algorithm.

The major disadvantage of kinematic model based tracking control algorithm is that it is only valid for very low motion speeds and payloads. This is because wheel slip will occur while the motion speed is high or the payload is high and the perfect wheel-surface contact condition will no longer hold. In this case the ideal kinematic model is no longer valid. However, the perfect

dynamic model is still based on the assumption of perfect wheel-surface contact, and therefore, the dynamic tracking control algorithm based on a perfect dynamic model cannot completely overcome the disadvantages of the kinematic model based tracking control algorithm.

From the above reviewed previous work, it is clear that in order to meet the requirements for tracking performance for the Tethered Mobile Robot, attention must directed to the following:

- The influence of the kinematic features on the tracking performance;
- Uncertainty influence in kinematic model and dynamic model;
- Tracking control algorithm robustness;
- Dynamic tracking control algorithm.

1.4 Tracking Control Problem Statement

1.4.1 *Control Objective*

The control objective in this report is the Tethered Mobile Robot (TMR). The TMR is a mobile robot with two independently driven wheels and a castor. From the tracking control point of view, the TMR is the same as any mobile robot with two independently driven wheels except that its position and orientation can be accurately measured. Therefore, all the modeling and control algorithm results in this report is valid for all wheeled mobile robots with the same wheel configurations as TMR.

1.4.2 *Coordinate Systems*

The coordinate systems are selected to be consistent with the convention employed in the ground vehicle community. The *world coordinate system* is determined with the x axis in the longitudinal direction, the z axis in vertical direction and pointing towards the ground, the y axis is selected to make the x-y-z system a right-hand-system. The vehicle orientation is determined by the angle between the vehicle heading direction and the x axis. The *vehicle fixed coordinate system* is determined by the vehicle heading direction and lateral direction.

1.4.3 *Reference Path and Reference Trajectory*

Reference path is a curve in the x-y plane which is a desired path for the mobile robot to follow. *Reference trajectory* is the time parameterized curve in the x-y plane. The reference path is time independent, but the reference trajectory is time dependent. If the along path motion speed is determined along a reference path, then the reference path becomes a reference trajectory. Therefore, a reference trajectory is a reference path with motion speed determined. In most parts of this report we do not strictly distinguish between reference path and reference trajectory.

1.4.4 *Tracking Point and Baseline*

Tracking point is a point on the mobile robot with which the mobile robot follows a reference path. *Reference point* is a point on the reference path at which the tracking point on the mobile robot should align at time t. For the TMR, the tracking point is the tool equipment operating point. *Baseline* is the line on the mobile robot which links the centers (axles) of the two driven wheels.

1.4.5 *Tracking Control Problem*

The tracking control problem is to design a control algorithm to control the mobile robot to follow a reference path or trajectory in position and orientation. Since the tracking point of the TMR is the tool operation point which is not on the center of the baseline, we mainly study the position tracking control problem in this report. This is because the orientation is determined by the reference path while the tracking point is not on the center of the baseline.

1.4.6 *Exponential Tracking and Robustness*

If the position and orientation tracking errors exponentially converge to zero, we call the tracking control algorithm an *exponential tracking* control algorithm. *Robustness* is the property of a tracking control algorithm which guarantees perfect tracking control performance under system uncertainty or external disturbances.

1.5 Conclusions

In this chapter, we introduce the purposes and scope of this report. Previous research work on this area is reviewed with brief comments. This report concentrates on the development of a

tracking control algorithm for the Tethered Mobile Robot. All the results are valid for any mobile robot with two differently driven wheels.

CHAPTER 2

KINEMATIC MODELING AND ORIENTATION EQUATION

2.1 Introduction

The kinematic model is the basis for the tracking control of wheeled mobile robots. Kinematic models of mobile robots have been studied by many authors. Muri and Neuman (1987), and Alexander and Maddocks (1990) studied the general wheeled mobile robot kinematics problem, and concentrated on a general mathematical description of the kinematics for varied configurations of wheeled mobile robots. However, for the purpose of this report, it is necessary to have a detailed study on the kinematic model of mobile robots with two independent wheels. In this report, we first describe the kinematics equations of the mobile robots, and then study the kinematic features and their influence on the tracking control performance. The tracking control performance of a mobile robot with two independent driven wheels is strongly determined by its kinematics features and the location of the tracking point on the mobile robot as well. Through the analysis of the kinematics characteristics, fundamental tracking control algorithm structures of this kind of mobile robot are proposed. This includes the tracking variable assignment, the tracking singularity and position-orientation tracking decoupling problems.

2.2 Kinematics Equations

The structural parameters of the Tethered Mobile Robot are defined as shown in Fig. 2. The coordinate system x-o-y is the world coordinate with the z axis pointing into the paper to make the frame right handed. The local velocity coordinate frame $u-o'-v$ is fixed on the mobile robot with the origin point o' located at the tracking point G, and i and j are the unit vectors of heading speed u and lateral speed v, respectively.

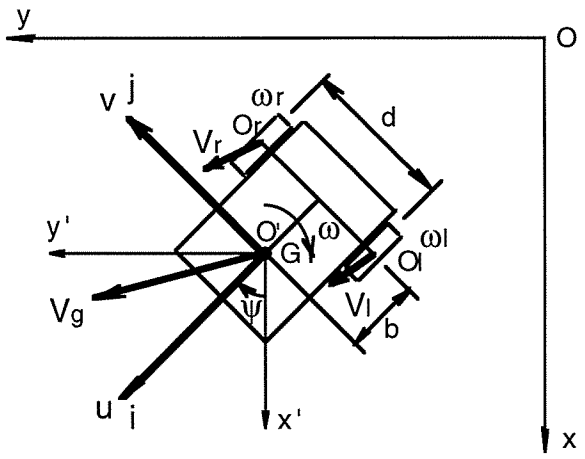


Fig. 2.1 Mobile Robot Kinematic Parameters Definition

When the longitudinal and lateral slippage exist, the velocity of the wheel center has both lateral and longitudinal components; that is

$$\vec{V}_l = u_l \vec{i} + v_l \vec{j} \tag{2.1}$$

and

$$\vec{V}_r = u_r \vec{i} + v_r \vec{j} \tag{2.2}$$

where, u_l *and* u_r are the wheel axis longitudinal speed and v_l *and* v_r are the wheel axis lateral speed, and l *and* r corresponding to the left and right wheels, respectively. The velocity of the tracking point G can be described as

$$\vec{V}_g = \vec{V}_l + \omega \vec{k} \times \vec{O_l G} \tag{2.3}$$

where, ω is the rotational angular speed and k is a vertical unit vector pointing into the paper. Then, we have

$$\begin{aligned}\vec{V}_g &= u_l \vec{i} + v_l \vec{j} + \omega \vec{k} \times (b \vec{i} + \frac{d}{2} \vec{j}) \\ &= (u_l - \frac{d}{2}\omega) \vec{i} + (v_l + b\omega) \vec{j} \\ &= u \vec{i} + v \vec{j}\end{aligned} \quad . \tag{2.4}$$

Therefore, we can obtain

$$u = u_l - \frac{d}{2}\omega \tag{2.5}$$

and

$$v = v_l + b\omega. \tag{2.6}$$

On the other hand

$$\begin{aligned}\vec{V}_g &= \vec{V}_r + \omega \vec{k} \times \vec{O_r G} \\ &= u_r \vec{i} + v_r \vec{j} + \omega \vec{k} \times (b \vec{i} - \frac{d}{2} \vec{j}) \\ &= (u_r + \frac{d}{2}\omega) \vec{i} + (v_r + b\omega) \vec{j} \\ &= u \vec{i} + v \vec{j}\end{aligned} \quad . \tag{2.7}$$

Accordingly, we have

$$u = u_r + \frac{d}{2}\omega \tag{2.8}$$

$$v = v_r + b\omega \tag{2.9}$$

From (2.5) and (2.8), we can obtain

$$\omega = \frac{1}{d}(u_l - u_r), \tag{2.10}$$

and from (2.6) and (2.9), we have

$$v_l = v_r =: v^s \tag{2.11}$$

and

$$v = \frac{b}{d}(u_l - u_r) + v^s \tag{2.12}$$

where v_s is referred to as the lateral slip speed. Substituting (2.10) into (2.5) or (2.8), we have

$$u = \frac{1}{2}(u_l + u_r). \tag{2.13}$$

2.3 Kinematic Model with Uncertainties

To arrive at the kinematic model with uncertainties, we make a basic assumption that there is not pure lateral sliding; i.e., lateral slip is due to the cornering force only. The longitudinal slip coefficient c is defined to be

$$c_i = 1 - \frac{V_i}{r_i \omega_i} \quad (i = l, r) \tag{2.14}$$

where, r is the radius of the wheel, and ω is the rotational angular speed of the wheel. Therefore,

$$V_i = \omega_i r_i (1 - c_i) \quad (i = l, r). \tag{2.15}$$

Because of the lateral slip, there is a slip angle α between the heading direction and the travel direction of the wheel. The longitudinal and the lateral speed of a wheel can then be determined as

$$u_i = V_i \cos\alpha_i = \omega_i r_i (1 - c_i)\cos\alpha_i = \omega_i r_{e,i} \quad (i = l, r) \tag{2.16}$$

and

$$v_i = V_i \sin\alpha_i = \omega_i r_{e,i} tg\alpha_i \quad (i = l, r) \tag{2.17}$$

where $r_e = r(1-c)\cos\alpha$ is defined to be the effective rolling radius of the wheel. Substituting (2.16) and (2.17) into (2.10), (2.12) and (2.13), we have

$$\begin{bmatrix} u(t) \\ v(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{2} r_{e,l} & \frac{1}{2} r_{e,r} \\ \frac{b_l}{d} r_{e,l} & -\frac{b_r}{d} r_{e,r} \\ \frac{1}{d} r_{e,l} & -\frac{1}{d} r_{e,r} \end{bmatrix} \begin{bmatrix} \omega_l(t) \\ \omega_r(t) \end{bmatrix} \tag{2.18}$$

where

$$b_l = b + \frac{d}{2} tg\alpha_l \tag{2.19}$$

and

$$b_r = b - \frac{d}{2} tg\alpha_r . \tag{2.20}$$

In the world coordinate frame, we have

$$\begin{bmatrix} \dot{x}_g(t) \\ \dot{y}_g(t) \\ \dot{\psi}(t) \end{bmatrix} = \begin{bmatrix} \cos\psi(t) & -\sin\psi(t) & 0 \\ \sin\psi(t) & \cos\psi(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u(t) \\ v(t) \\ \omega(t) \end{bmatrix}. \tag{2.21}$$

Combining (2.18) and (2.21), we have

$$\begin{bmatrix} \dot{q}_p(t) \\ \dot{\psi}(t) \end{bmatrix} = \begin{bmatrix} G_p(\psi) \\ g^T \end{bmatrix} \upsilon(t) \tag{2.22}$$

where

$$q_p(t) = \begin{bmatrix} x_g(t) & y_g(t) \end{bmatrix}^T \tag{2.23}$$

$$\upsilon(t) = \begin{bmatrix} \omega_l(t) & \omega_r(t) \end{bmatrix}^T \tag{2.24}$$

$$G_p(\psi)=\frac{1}{d}\begin{bmatrix} r_{e,l}[\frac{d}{2}\cos\psi(t)-b_l\sin\psi(t)] & r_{e,r}[\frac{d}{2}\cos\psi(t)+b_r\sin\psi(t)] \\ r_{e,l}[\frac{d}{2}\sin\psi(t)+b_l\cos\psi(t)] & r_{e,r}[\frac{d}{2}\sin\psi(t)-b_r\cos\psi(t)] \end{bmatrix} \tag{2.25}$$

and

$$g=\frac{1}{d}\begin{bmatrix} r_{e,l} & -r_{e,r} \end{bmatrix}^T. \tag{2.26}$$

Although the magnitudes of the longitudinal slip and lateral slip are determined by the dynamics and the payload of the mobile robot, the dynamic model of the tethered mobile robot possess more uncertain influences, such as the modeling of pneumatic tire and payload. The kinematic model of (2.22) describes all the influences of slippage as parameter uncertainties.

2.4 Perfect Kinematic Model

We define the perfect kinematic model to be the situation where there is no slippage and parameters uncertainties. For the perfect kinematic model, the time derivatives of the position and orientation of a mobile robot with two independent driven wheels in a planar surface can be completely determined by the angular speed of the driven wheels. From the previous section, we set

$$v^s \equiv 0,\ c_l \equiv 0,\ c_r \equiv 0$$

and

$$r_l = r_r = r.$$

Then, we have

$$\omega(t)=\frac{r}{d}[\omega_l(t)-\omega_r(t)] \tag{2.27}$$

$$v(t)=\frac{br}{d}[\omega_l(t)-\omega_r(t)] \tag{2.28}$$

$$u(t)=\frac{r}{2}[\omega_l(t)+\omega_r(t)] \tag{2.29}$$

where $\omega(t)$ is the angular speed, and $u(t)$ and $v(t)$ are the forward and lateral speeds, respectively, r is the radius of the wheels, and ω_r *and* ω_l are the angular speeds of the right and left wheel, respectively. It is clear that the rotation speed $\omega(t)$ and lateral speed $v(t)$ are not independent. Therefore, there are only two independent variables in the local coordinate system.

The motion equations of the tracking point G in the world coordinates can be obtained through a rotation transformation; that is

$$\dot{x}_g(t) = u(t)\cos\psi(t) - v(t)\sin\psi(t) \tag{2.30}$$

$$\dot{y}_g(t) = u(t)\sin\psi(t) + v(t)\cos\psi(t) \tag{2.31}$$

$$\dot{\psi}(t) = \omega(t) \tag{2.32}$$

where, x and y are the position of the tracking point G in the world coordinate frame x-o-y, and ψ is the orientation angle of the mobile robot. Combining (2.27)-(2.32), we have

$$\dot{x}(t) = G(\psi)v(t) \tag{2.33}$$

where

$$x(t) = \begin{bmatrix} x_g(t) & y_g(t) & \psi(t) \end{bmatrix}^T \tag{2.34}$$

$$v(t) = \begin{bmatrix} \omega_l(t) & \omega_r(t) \end{bmatrix}^T \tag{2.35}$$

and

$$G(\psi) = \frac{r}{d}\begin{bmatrix} -b\sin\psi + \frac{d}{2}\cos\psi & b\sin\psi + \frac{d}{2}\cos\psi \\ b\cos\psi + \frac{d}{2}\sin\psi & -b\cos\psi + \frac{d}{2}\sin\psi \\ 1 & -1 \end{bmatrix}. \tag{2.36}$$

2.5 Tracking Pairs and Singularity

Although many authors use all three posture variables as feedback variables in the control algorithms, the mobile robot with two independent driven wheels is of only two degrees of

freedom, and there are only two completely independent variables in the world coordinates. To track a reference path in position, the orientation angle can not be controlled arbitrarily at the same time. Since the only feasible reference orientation is the tangential direction of the reference path, the orientation is inherently determined by the path configuration. We will refer to the tangential direction of a given reference path as the reference orientation throughout this paper.

Since there are two degrees of freedom for the mobile robot, we can exactly track two variables. We can chose one of the variable pairs (x_g, y_g) *or* (x_g, ψ) *or* (y_g, ψ) to be the tracking control feedback variables. Denoting matrix $G(\psi)$ as

$$G(\psi) = \begin{bmatrix} g_x^T \\ g_y^T \\ g_\psi^T \end{bmatrix} \tag{2.37}$$

where, $g_i^T \in R^2, (i = x, y, \psi)$, we can obtain the following results:

$$\kappa_{x,y} = det\begin{bmatrix} g_x^T \\ g_y^T \end{bmatrix} = -\frac{r^2}{d} b \tag{2.38}$$

$$\kappa_{x,\psi} = det\begin{bmatrix} g_x^T \\ g_\psi^T \end{bmatrix} = -\frac{r^2}{d} cos\,\psi \tag{2.39}$$

$$\kappa_{y,\psi} = det\begin{bmatrix} g_y^T \\ g_\psi^T \end{bmatrix} = -\frac{r^2}{d} sin\,\psi. \tag{2.40}$$

From (2.38)-(2.40), we arrive at the following observations:

- Since $\kappa_{x,y}$ is always nonzero when $b \neq 0$, therefore, the tracking pair $(x_g(t), y_g(t))$ is nonsingular for any reference path as long as the tracking point is not on the center of baseline;
- It is clear that $\kappa_{x,\psi}$ is zero when $\psi = \pm\frac{\pi}{2}$, therefore, the tracking pair $(x_g(t), \psi(t))$ is singular when $\psi = \pm\frac{\pi}{2}$; that is, this tracking pair cannot track any reference path which is parallel with the y axis;

- For the tracking pair $(y_g(t), \psi(t))$, the singularity occurs when $\kappa_{y,\psi}$ is zero at $\psi = 0$ *or* $\psi = \pm\pi$; this means that this tracking pair cannot track any reference path parallel with the x axis.

Therefore, when $b \neq 0$, we can only track a reference path in position. When b=0, that is, the tracking point G is on the center of baseline, there is no lateral speed at the tracking point G on the local coordinate frame. The basic kinematic equations in the local coordinate frame becomes

$$\omega(t) = \frac{r}{d}[\omega_l(t) - \omega_r(t)] \tag{2.41}$$

$$u(t) = \frac{r}{2}[\omega_l(t) + \omega_r(t)] \tag{2.42}$$

and in the world coordinates the kinematic equations are of the following form:

$$\dot{x}_g(t) = u(t)\cos\psi(t) \tag{2.43}$$

$$\dot{y}_g(t) = u(t)\sin\psi(t) \tag{2.44}$$

$$\dot{\psi}(t) = \omega(t). \tag{2.45}$$

Then, the matrix $G(\psi)$ defined by (2.36) becomes

$$G(\psi) = \begin{bmatrix} \frac{r}{2}\cos\psi & \frac{r}{2}\cos\psi \\ \frac{r}{2}\sin\psi & \frac{r}{2}\sin\psi \\ \frac{r}{d} & -\frac{r}{d} \end{bmatrix}. \tag{2.46}$$

Sine $\kappa_{x,y} \equiv 0$ in this case, the pure position tracking control is completely singular, and the orientation variable must be taken as the tracking control variable.

2.6 Orientation Equation

The orientation of the mobile robot during the path tracking process is determined by the features of the reference path. We define the tangential direction of the reference path as the reference orientation $\psi_f(t)$. We can see that when $b \neq 0$ and if we want to accurately track a reference path in position, it is impossible to have accurate orientation tracking in this case. A difference between the mobile robot orientation and the reference orientation is inherently necessary when the position tracking is exact.

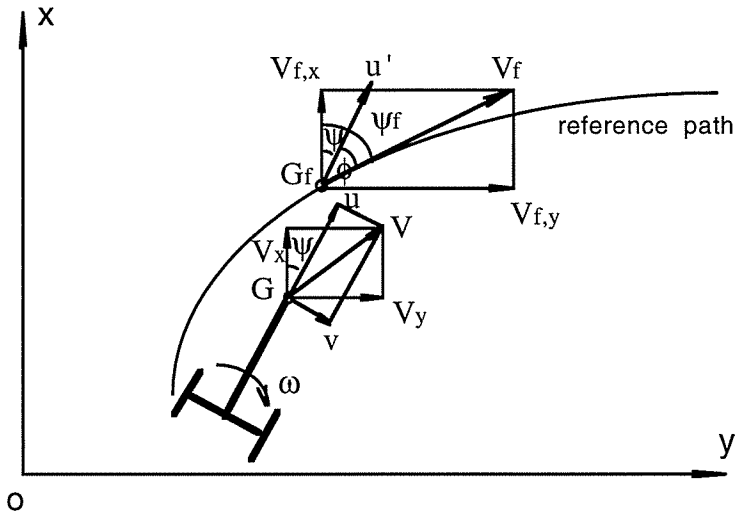


Fig. 2.2 Orientation Tracking Error

Denoting the angle difference between the mobile robot heading direction and the reference path tangential direction as ϕ as shown in Fig. 2.2, the value of the speed of the tracking point G on the mobile robot is written as

$$V(t) = \sqrt{V_x^{\,2}(t) + V_y^{\,2}(t)} = \sqrt{\dot{x}_g^{\,2}(t) + \dot{y}_g^{\,2}(t)} \tag{2.47}$$

and the corresponding value of the speed of the reference point Gf on the reference path is

$$V_f(t) = \sqrt{V_{f,x}{}^2(t) + V_{f,y}{}^2(t)} = \sqrt{\dot{x}_{gf}{}^2(t) + \dot{y}_{gf}{}^2(t)}\,. \tag{2.48}$$

Since

$$b\dot{\psi}(t) = v(t) \tag{2.49}$$

and

$$v(t) = V_f(t)\sin\phi - \Delta V(t)\sin(\psi + \delta) \tag{2.50}$$

where

$$\Delta V(t) = \sqrt{\dot{e}_x{}^2(t) + \dot{e}_y{}^2(t)} \tag{2.51}$$

$$e_x(t) = x_g(t) - x_{gf}(t)$$

$$e_y(t) = y_g(t) - y_{gf}(t)$$

$$\delta = a\tan\left(\frac{\dot{e}_y(t)}{\dot{e}_x(t)}\right), \tag{2.52}$$

and noting that

$$\psi(t) = \psi_f(t) - \phi(t) \tag{2.53}$$

and

$$\psi_f(t) = a\tan\left(\frac{\dot{y}_{gf}(t)}{\dot{x}_{gf}(t)}\right) \tag{2.54}$$

$$\dot{\psi}_f(t) = \frac{\dot{x}_{gf}(t)\ddot{y}_{gf}(t) - \ddot{x}_{gf}(t)\dot{y}_{gf}(t)}{\dot{x}_{gf}{}^2(t) + \dot{y}_{gf}{}^2(t)}, \tag{2.55}$$

we have

$$\dot{\phi}(t) + \frac{1}{b}\sqrt{\dot{x}_{gf}{}^2(t) + \dot{y}_{gf}{}^2(t)}\sin\phi(t) = \dot{\psi}_f(t) + \varepsilon(t) \tag{2.56}$$

where

$$\varepsilon(t) = \frac{1}{b}\sqrt{\dot{e}_x{}^2(t) + \dot{e}_y{}^2(t)}\sin(\psi_f(t) - \phi(t) + \delta(t)). \tag{2.57}$$

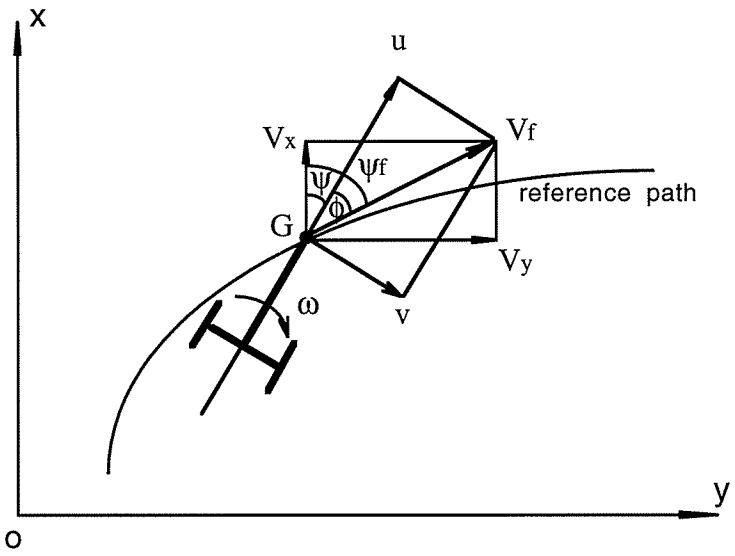


Fig. 2.3 Orientation Tracking Error When Position Tracking Is Exact

The last term in equation (2.56) only depends on the initial position errors. When the initial position errors are zero, or when $t \to \infty$, $\dot{e}_p(t) \to 0$, $\varepsilon(t) \to 0$, we have

$$\dot{\phi}(t) + \frac{1}{b}\sqrt{\dot{x}_{gf}^{\,2}(t) + \dot{y}_{gf}^{\,2}(t)}\,\sin\phi(t) = \frac{\dot{x}_{gf}(t)\ddot{y}_{gf}(t) - \ddot{x}_{gf}(t)\dot{y}_{gf}(t)}{\dot{x}_{gf}^{\,2}(t) + \dot{y}_{gf}^{\,2}(t)} \quad . \tag{2.58}$$

From (2.58), when the mobile robot accurately tracks a reference path in position, there may be a nonzero $\phi(t)$ determined by the reference path configuration, as shown in Fig. 2.3. In this case, the tracking point G is overlapping the reference point Gf on the reference path, and both the value and direction of the velocity of point G is the same as that of the desired velocity at the reference point Gf. If the reference path has a nonzero curvature at point Gf, the mobile robot must have a nonzero rotation speed. Therefore, there is a nonzero lateral speed v at point G, and this means the orientation ψ of the mobile robot cannot be the same with the tangential direction ψ_f of the reference path at point Gf.

The zero input equation of (2.58) is of the form:

$$\dot{\phi}(t)+\frac{1}{b}q(t)\sin\phi(t)=0 \quad (q(t)>0, \forall t \geq 0). \tag{2.59}$$

We can see that $\phi = 0 \; or \pm \pi$ are the equilibrium solutions of (2.59); that is, when $\phi = 0 \; or \pm \pi$ there is $\dot{\phi}(t) = 0$. When b>0, the tracking point G is in front of the baseline, equation (2.59) has a stable equilibrium solution of ϕ=0 but $\phi = \pm\pi$ is unstable. In this case, the mobile robot is heading forward during the tracking process. When b<0, the tracking point G is behind the baseline, equation (2.59) then has a stable equilibrium solution of $\phi = \pm\pi$ while ϕ=0 is not stable; this means the mobile robot will track the path while heading rearward. When b=0, the tracking point G is the center of the baseline, and from (2.49) the lateral speed is zero. The value and the direction of the forward speed $u(t)$ are the same as that of the desired velocity at the reference point G_P on the reference path, and therefore $\phi(t) \equiv 0$.

From the above discussions, the orientation of the mobile robot is completely determined by the reference path and the location of the tracking point on the mobile robot. Under the definition of reference orientation being the tangential direction of the reference path, exact orientation tracking occurs only when the reference path is a straight line or the tracking point is the center point of the baseline. The mobile robot may be in a heading-forward or heading-rearward direction during the tracking process depending on the location of the tracking point (in the front or behind the baseline). The orientation tracking error increases as the distance from the tracking point to the baseline increases.

It is evident that both exact position and orientation tracking control at the same time is impossible for this kind of mobile robot when the tracking point is not on the baseline. If both position and orientation accurate tracking is required, a third degree of freedom for the posture control is necessary in the mobile robot configuration. Orientation tracking error can be reduced by decreasing the distance b or decreasing the curvature of the reference path.

2.7 Conclusion

Kinematic models of the mobile robot with two driven wheels is developed in this chapter. General kinematics equations, uncertainty kinematic model, and perfect kinematic model are produced. Since there are only two control inputs and three posture variables, tracking control pairs selection and their singularity are also discussed in this chapter.

An orientation equation under the condition of exact position tracking is proposed, and the fundamental relation between position tracking and orientation tracking is given. When the mobile robot exactly tracks a reference path, the orientation of the mobile robot is completely determined by the location of the tracking point on the mobile robot and the desired speed and acceleration of the reference path.

CHAPTER 3

KINEMATIC MODEL BASED TRACKING CONTROL

3.1 Introduction

With respect to the tracking control problem for wheeled mobile robots, algorithms from basic PID based ones to fuzzy logic and neural network based algorithms were developed by many authors. The knowledge based algorithms are mainly suitable for real environment guidance type problems. However, for the problem of accurate path tracking control, a kinematic or dynamic model based control algorithm is essential. This is because the dynamic model of the mobile robot can have considerable uncertainty in modeling the driven force and payload, and its computational complexity is relatively high. Kinematic models play an important role in control algorithm development. Kanayama, Nilipour, and Lelm (1988) proposed a PID control algorithm for mobile robot tracking control. The kinematic model was used to transform the posture errors in world-coordinates to errors in a robot-fixed frame. A similar control algorithm was developed by Lee and Williams (1993). Kanayama, Kimura, Miyazaki, and Noguchi (1990) proposed a nonlinear control algorithm based on a kinematic model, and they also gave a proof of the algorithm's stability. Winters and Velinsky (1992) used a PID based control algorithm in a Tethered Mobile Robot tracking control study which included both simulation and experiment of a prototype mobile robot. Since the fundamental control structure related to the kinematic features was uncertain, in some algorithms, the inherent consistence with the kinematics of the mobile robot was not completely matched. As such, a strong disturbance between position and orientation tracking was reported, and the tracking performance was poor.

In this chapter global convergence tracking control algorithms are developed for two different cases, for the tracking point on and off the baseline. When the tracking point is not on the baseline, only the position of the tracking point can exactly track a reference path, and it is impossible to exactly track the heading direction of the mobile robot. When the tracking point is

located on the baseline, both the position of the tracking point and the orientation of the mobile robot can exactly track a reference path. Numerical simulation examples are provided which illustrate the performance of the control algorithms.

3.2 Global Exponential Position Tracking Control

When the tracking point is not on the baseline, the mobile robot position variables $(x_g(t), y_g(t))$ can be taken as control variables. We now denote

$$G_p(\psi)=\begin{bmatrix} g_x^T \\ g_y^T \end{bmatrix}=\frac{r}{d}\begin{bmatrix} -b\sin\psi+\frac{d}{2}\cos\psi & b\sin\psi+\frac{d}{2}\cos\psi \\ b\cos\psi+\frac{d}{2}\sin\psi & -b\cos\psi+\frac{d}{2}\sin\psi \end{bmatrix} \tag{3.1}$$

and take the control inputs as

$$\upsilon(t)=[G_p^{-1} \quad (g_\psi^T g_\psi)^{-1} g_\psi]\begin{bmatrix} \upsilon_x \\ \upsilon_y \\ --- \\ \upsilon_\psi \end{bmatrix}. \tag{3.2}$$

From (2.33) and (3.2), we have

$$\begin{bmatrix} \dot{x}_p(t) \\ \dot{\psi}(t) \end{bmatrix}=\begin{bmatrix} I_{2\times 2} & f(\psi) \\ h^T(\psi) & 1 \end{bmatrix}\begin{bmatrix} \upsilon_p(t) \\ \upsilon_\psi(t) \end{bmatrix} \tag{3.3}$$

where,

$$x_p(t):=\begin{bmatrix} x_g(t) & y_g(t) \end{bmatrix}^T \tag{3.4}$$

$$\upsilon_p(t)=\begin{bmatrix} \upsilon_x(t) & \upsilon_y(t) \end{bmatrix}^T \tag{3.5}$$

$$h(\psi)=\begin{bmatrix} -\frac{\sin\psi}{b} & \frac{\cos\psi}{b} \end{bmatrix}^T \tag{3.6}$$

and

$$f(\psi)=\left[-b\sin\psi \quad b\cos\psi\right]^T. \tag{3.7}$$

From (3.6) and (3.7), we have

$$\begin{aligned} h^T(\psi)f(\psi)&=1,\\ h^T(\psi)\dot{x}_p(t)&=\dot{\psi}(t) \end{aligned} \tag{3.8}$$

and

$$f(\psi)\dot{\psi}(t)=[\dot{x}_p(t)]_{rotation} \tag{3.9}$$

where $[\dot{x}_p(t)]_{rotation}$ represents the component of speed caused by the pure rotation with angular speed $\dot{\psi}(t)$. Therefore, if the transformed control inputs $v_p(t)$ *and* $v_\theta(t)$ correspond to the reference value of $\dot{x}_p(t)$ *and* $\dot{\psi}(t)$, respectively, then the rotation speed is fully determined by the reference position speed, and any control to rotational speed will disturb the accurate control of the position. Considering that

$$rank\begin{bmatrix} I_{2\times 2} & f(\psi) \\ h^T(\psi) & 1 \end{bmatrix}=2 \tag{3.10}$$

then a direct full decoupling control algorithm between $v_p(t)$ *and* $v_\psi(t)$ cannot be obtained.

For the position tracking control, we take

$$v_\theta=0 \tag{3.11}$$

and

$$v_p=\dot{x}_p(t)-K_p e_p(t) \tag{3.12}$$

where

$$e_p(t)=x_p(t)-x_{pf}(t)$$

$$K_p = \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix} \quad (k_x > 0, k_y > 0) \tag{3.13}$$

and $x_{pf}(t)$ is the reference path. Then, from (3.3) we have

$$\dot{e}_p(t) = -K_p e_p(t) \tag{3.14}$$

and

$$\dot{\psi}(t) = h^T(\psi)[\dot{x}_{pf} - K_p e_p(t)]. \tag{3.15}$$

From (3.14), we have

$$e_p(t) = e^{-K_p t} e_p(0) \tag{3.16}$$

which implies that under control (3.11) and (3.12), the mobile robot exponentially tracks the reference path in position with zero steady state error, and the algorithm is a globally convergent.

From (3.12) and (3.14), we can see that there is no need to have a derivative feedback term or an integral feedback term in the control (3.12). A derivative feedback term is equivalent to decreasing the value of Kp, and an integral term will deteriorate the stability margin.

Since $h^T(\psi)\dot{x}_{pf}(t) \to \dot{\psi}(t)$ *and* $e_p(t) \to 0$, *when* $t \to \infty$, therefore, equation (3.15) gives no information about orientation. Combining (3.2), (3.11), and (3.12), the feedback linearized exponential tracking control algorithm is written as

$$v(t) = G_p^{-1}(\psi)[\dot{x}_{pf}(t) - K_p e_p(t)]. \tag{3.17}$$

Because the matrix Gp is second order and its determinate is a constant as shown in (2.38), there is no extra calculation needed to determine the inverse of Gp in (3.17). The control algorithm (3.17)

is of the same calculation complexity as a common PID control algorithm and it is therefore suitable for real time application.

3.3 Global Position and Orientation Tracking Control

When the tracking point is on the baseline, exactly tracking both position and orientation is possible as described in section 2.2. In this case, we take the local variables $u(t)$ *and* $\omega(t)$ to be tracking variables. Then from (2.41) and (2.42) we have

$$\begin{bmatrix} u(t) \\ \omega(t) \end{bmatrix} = H\upsilon(t) \tag{3.18}$$

where

$$H = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{d} & -\frac{r}{d} \end{bmatrix}. \tag{3.19}$$

Taking the control to be

$$\upsilon(t) = H^{-1}\begin{bmatrix} \upsilon_u \\ \upsilon_\psi \end{bmatrix} \tag{3.20}$$exactly tracking

results in

$$\begin{bmatrix} u(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} \upsilon_u \\ \upsilon_\psi \end{bmatrix}. \tag{3.21}$$

Although the control system (3.21) is linear in the local coordinate frame, the control law for the transformed control inputs υ_u *and* υ_ψ must be designed based on world coordinate variables. Considering the expression of $u(t)$ in world coordinates, we have

$$u(t) = \dot{x}_g(t)\cos\psi(t) + \dot{y}_g(t)\sin\psi(t). \tag{3.22}$$

We now set the control algorithm to be

$$v_u(t) = u_f(t) - k_u[e_x(t)\cos\psi(t) + e_y(t)\sin\psi(t)] \tag{3.23}$$

$$v_\psi(t) = \dot{\psi}_f(t) - k_\psi\sigma(e_\psi) + \xi(e_x, e_y, \psi) \tag{3.24}$$

where, $k_u > 0,\ k_\psi > 0$, and $\sigma(\cdot)\ and\ \xi(\cdot)$ are unknown functions to be determined as follows. Substituting (3.23) and (3.24) into (3.21), we have

$$u(t) = u_f(t) - k_u[e_x(t)\cos\psi(t) + e_y(t)\sin\psi(t)] \tag{3.25}$$

and

$$\dot{\psi}(t) = \dot{\psi}_f(t) - k_\psi\sigma(e_\psi) + \xi(e_x, e_y, \psi). \tag{3.26}$$

Taking the Lyapunov function candidate to be

$$V = \frac{1}{2}[e_x^2(t) + e_y^2(t)] + 4\sin^2(\frac{e_\psi(t)}{4}), \tag{3.27}$$

then the time derivative of V along the trajectory of the system equation is expressed as

$$\dot{V} = e_x\dot{e}_x + e_y\dot{e}_y + 2\sin(\frac{e_\psi}{4})\cos(\frac{e_\psi}{4})\dot{e}_\psi. \tag{3.28}$$

Considering (2.43) and (2.44), we have

$$\begin{aligned} \dot{e}_x(t) &= u(t)\cos\psi(t) - u_f\cos\psi_f(t) \\ \dot{e}_y(t) &= u(t)\sin\psi(t) - u_f\sin\psi_f(t) \end{aligned}. \tag{3.29}$$

Therefore, we arrive at the following expression

$$\begin{aligned}\dot{V} &= (e_x \cos\psi + e_y \sin\psi)u - (e_x \cos\psi_f + e_y \sin\psi_f)u_f + \sin(\frac{e_\psi}{2})\dot{e}_\psi \\ &= -k_u(e_x \cos\psi + e_y \sin\psi)^2 + \sin(\frac{e_\psi}{2})\dot{e}_\psi \\ &\quad + (e_x \cos\psi + e_y \sin\psi)u_f - (e_x \cos\psi_f + e_y \sin\psi_f)u_f \\ &= -k_u(e_x \cos\psi + e_y \sin\psi)^2 + \sin(\frac{e_\psi}{2})\dot{e}_\psi \\ &\quad + 2u_f(e_y \cos\psi - e_x \sin\psi)\sin(\frac{e_\psi}{2}) \\ &= -k_u(e_x \cos\psi + e_y \sin\psi)^2 - k_\psi \sin(\frac{e_\psi}{2})\sigma(e_\psi) \\ &\quad + [\xi(e_x, e_y, \psi) + 2u_f(e_y \cos\psi - e_x \sin\psi)]\sin(\frac{e_\psi}{2})\end{aligned} \quad . \tag{3.30}$$

Now, we let

$$\sigma(e_\psi) = \sin\frac{e_\psi}{2} \tag{3.31}$$

and

$$\xi(e_x, e_y, \psi) = -2u_f(e_y \cos\psi - e_x \sin\psi) \tag{3.32}$$

which leads to

$$\begin{aligned}\dot{V} &= -k_u(e_x \cos\psi + e_y \sin\psi)^2 - k_\psi \sin^2\frac{e_\psi}{2} \\ &\leq 0\end{aligned} \tag{3.33}$$

and V is a Lyapunov function. Combining (3.20), (3.23) and (3.24), we have

$$\upsilon(t) = H^{-1}\begin{bmatrix} u_f(t) - k_u[e_x(t)\cos\psi(t) + e_y(t)\sin\psi(t)] \\ \dot{\psi}_f(t) - k_\psi \sin\left(\frac{e_\psi(t)}{2}\right) - 2u_f(t)[e_y(t)\cos\psi(t) - e_x(t)\sin\psi(t)] \end{bmatrix}. \tag{3.34}$$

Since

$$0 \le |e_\psi| < 2\pi,$$

we have

$$0 \le |\sin\tfrac{e_\psi(t)}{4}| \le |\sin\tfrac{e_\psi(0)}{4}|$$

which implies that

$$0 \le |e_\psi(t)| \le |e_\psi(0)|,$$

and therefore, the control algorithm (3.34) is globally convergent for tracking a reference path both in position and orientation with the tracking point G on the baseline.

3.4 Illustrative Examples

We take three types of reference paths, straight line, circle, and sine, as illustrative examples to show the tracking control ability of the above control algorithms and the orientation tracking problem.

Example 3.4.1 *Straight line reference path with initial position errors.* Taking reference path to be as follows

$$\begin{cases} x_{gf}(t) = 2t \\ y_{gf}(t) = 2 + t \end{cases}$$

then the reference orientation is

$$\begin{aligned} \psi_f(t) &= a\tan\left(\frac{\dot{y}_{gf}}{\dot{x}_{gf}}\right) = a\tan(\frac{1}{2}). \\ \dot{\psi}_f(t) &= 0 \end{aligned}$$

The steady state orientation equation (2.58) is of the form

$$\dot{\phi}(t)+\frac{1}{b}\sqrt{5}\sin\phi(t)=0$$

If b>0, for $|\phi|<<1$, we have

$$\phi(t)=\phi(t_0)e^{-\frac{\sqrt{5}}{b}(t-t_0)}\to 0,\ t\to\infty \Rightarrow \psi(t)\to\psi_f(t)$$

and if b<0, for $\phi=\phi'-\pi,\ |\phi'|<<1$, we have

$$\phi'(t)=\phi'(t_0)e^{-\frac{\sqrt{5}}{|b|}(t-t_0)}\to 0,\ t\to\infty \Rightarrow \psi(t)\to\psi_f(t)-\pi\ .$$

The numerical simulation results are shown in Fig. 3.1 and Fig. 3.2. The initial conditions are set to be

$$x_p(0)=0 \text{ and } \psi(0)=0.$$

and the control parameters are set to be

$$k_x=k_y=3.$$

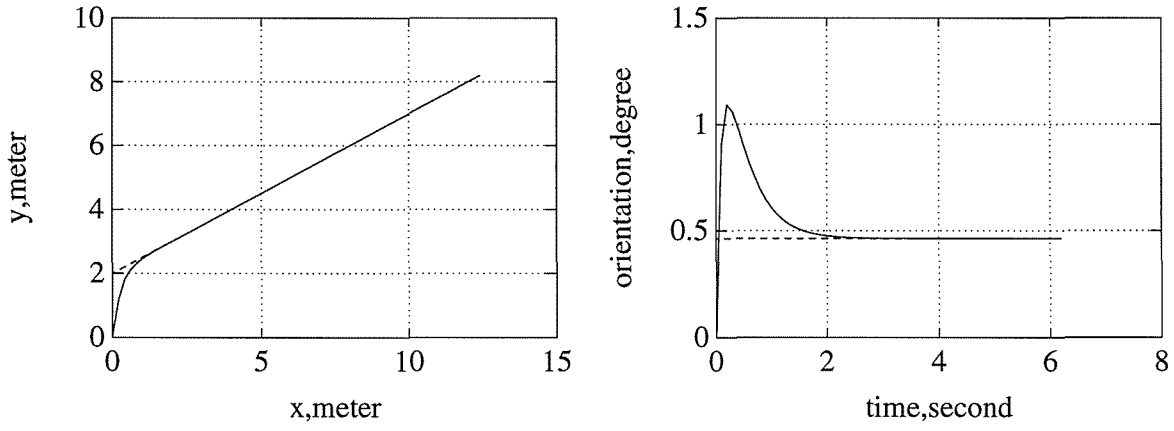


Fig. 3.1 Straight Line Reference Path with Initial Position Errors and b>0

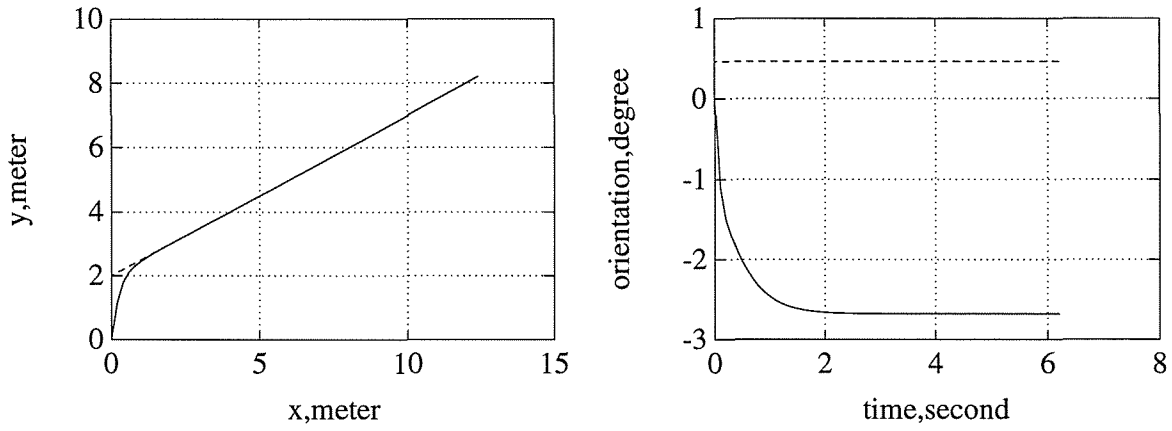


Fig. 3.2 Straight Line Reference Path with Initial Position Errors and b<0

Because there are initial position errors, the control algorithm exponentially converges to the reference path. The mobile robot orientation convergence to the tangential direction in heading-forward or heading-rearward depends on the sign of b. Fig. 3.1 corresponds to the situation when b>0, and Fig. 3.2 corresponds to the case of b<0. Numerical simulation results verify that whether the orientation error converges to zero or to π depends only on the sign of b regardless of the initial value of the orientation when $b \neq 0$.

Example 3.4.2 *Circular reference path with exact initial position.* A one fourth circle is taken to be the reference path which can be described as

$$\begin{cases} x_{gf}(t) = a\sin(\omega t) \\ y_{gf}(t) = a\cos(\omega t) \end{cases} \quad (0 \le t \le \frac{\pi}{2\omega})$$

where a>0 is the radius of the circle path. The reference orientation is then

$$\psi_f(t) = a\tan\left(\frac{\dot{y}_{gf}}{\dot{x}_{gf}}\right) = -\omega t$$
$$\dot{\psi}_f(t) = -\omega$$

The orientation equation (2.58) is now of the form

$$\dot{\phi}(t)+\frac{1}{b}a\omega\sin\phi(t)=-\omega\,.$$

The steady state solution is then expressed as

$$\phi_{ss}=\sin^{-1}\frac{b}{a}\quad(b>0)$$

or

$$\phi_{ss}=\pi+\sin^{-1}\frac{b}{a}\quad(b<0)\,.$$

When b<<a, the steady state value of ϕ is approximately equal to $\frac{b}{a}$ for b>0 or $\pi+\frac{b}{a}$ for b<0. This means that the orientation error is in proportion to the distance b and the curvature $\frac{1}{a}$ of the reference path. A numerical simulation result is shown in Fig. 3.3. In the simulation, the parameters of the reference path are set to be a=4 and $\omega=\frac{1}{4}$, and the control parameters are the same as in Example 3.5.1. An exact initial position is set for the simulation. The initial orientation is set to be the equal to the tangential direction of the reference path. The path of the mobile robot completely overlaps the reference path, and the orientation error increases from zero to a constant.

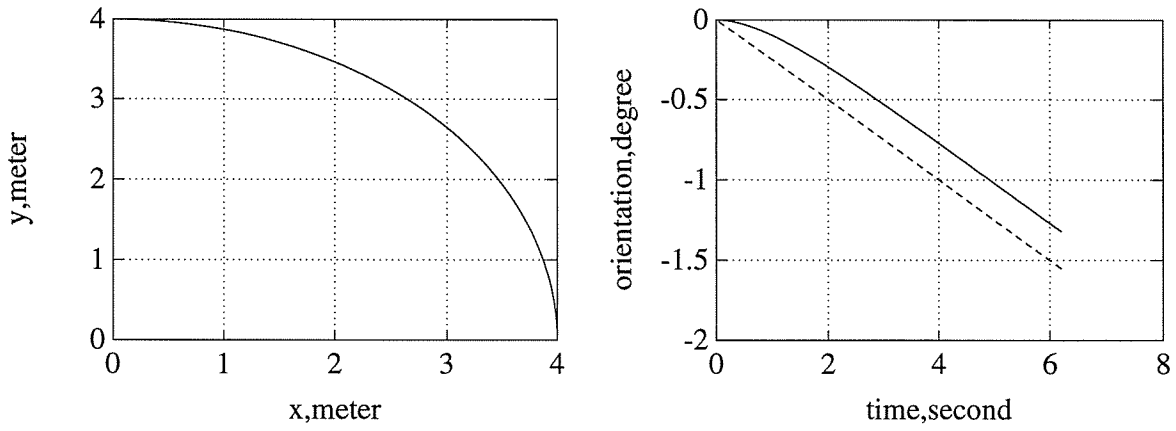


Fig. 3.3 Circle Reference Path with Exact Initial Position and b>0

Example 3.4.3 *Sine reference path with exact initial position.* The reference path is set to be

$$\begin{cases} x(t) = t \\ y(t) = c\sin(\omega t) \end{cases}$$

and the reference orientation is then expressed as

$$\psi_f(t) = a\tan(c\omega\cos\omega t).$$

The orientation equation is then written as

$$\dot{\phi}(t) + \frac{1}{b}\sqrt{1 + c^2\omega^2\cos^2\omega t}\,\sin\phi(t) = -\frac{c\omega^2\sin\omega t}{1 + c^2\omega^2\cos^2\omega t}.$$

In this case the reference orientation and the orientation error are all functions of time. Numerical simulation results are shown in Fig. 3.4. The value of c and ω in the reference path are set to be 4 and 1/4, respectively, and the control parameters are

$$k_x = k_y = 3.$$

The initial position is exact, and the initial orientation is zero. Because there is no initial position errors, the position path is again completely overlaps the reference path, and the orientation of the mobile robot delays the tangential direction of the reference path.

Example 3.4.4 *Circular reference path with the tracking point on the baseline.* A circular reference path is used to simulate the tracking control process of a mobile robot with the tracking point on the baseline, that is b=0. The initial conditions are set to be

$$x_p(0) = 0 \text{ and } \psi(0) = 0.$$

The control parameters are taken to be

$$k_\eta = 3 \text{ and } k_\theta = 5.$$

Fig. 3.5 shows that both position and orientation track the reference path with zero steady state error.

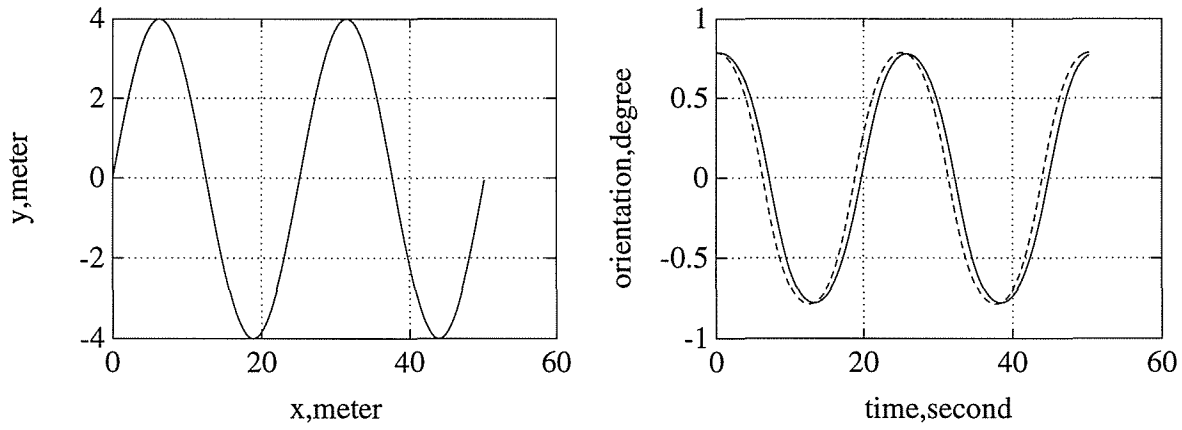


Fig. 3.4 Sine Reference Path with Exact Initial Position and b>0

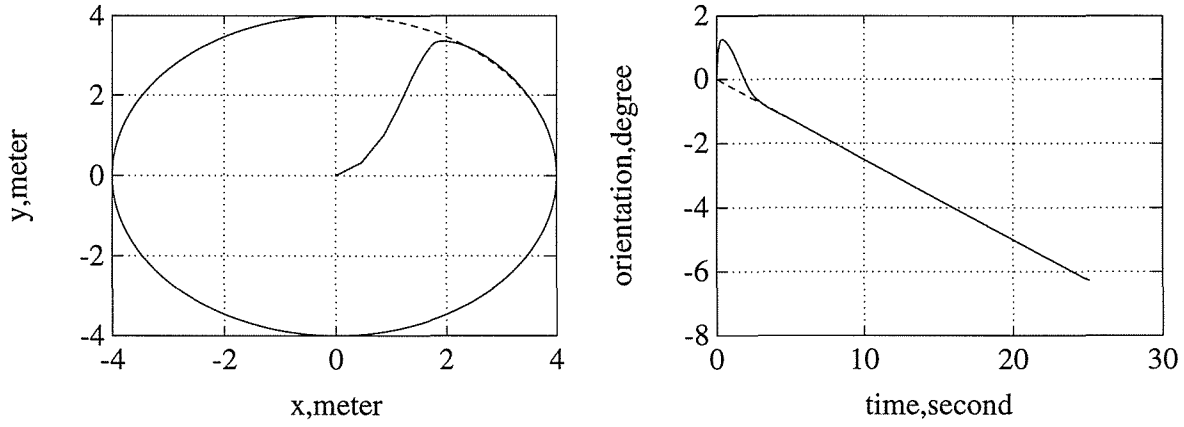


Fig. 3.5 Circular Reference Path with the Tracking Point on Baseline (b=0)

3.5 Conclusions

On the basis of a kinematic model, control algorithms for the mobile robot with two independent driven wheels are developed. A global exponentially convergent control algorithm and a globally convergent tracking control algorithm are developed according to the tracking point location relative to the baseline. These algorithms can track any differentiable reference path with zero steady state errors. Illustrative examples are provided to show the position tracking control ability and the orientation behavior for the control algorithms.

It can be concluded that it is impossible to exactly track both position and orientation concurrently when the tracking point is not on the baseline. When the tracking point is located on the center of baseline, the position and orientation can be tracked simultaneously.

CHAPTER 4

KINEMATIC MODEL BASED ROBUST TRACKING CONTROL

4.1 Introduction

Since the TMR is being developed for application in highway maintenance, the accurate tracking ability of the TMR to a reference path is very important for most maintenance operations. However, since the working environment of the TMR will be in an unstructured environment (i.e., highway), there are many uncertainties that exist. Among them are the changes of roadway surface conditions, the varying operation payloads, the different maintenance tasks, etc. These uncertainties combined with the relative high speeds will result in wheel slippage and a change in radius of the pneumatic tires. These uncertainties will appear in the mobile robot kinematics equation as uncertainties of effective wheel radius. Therefore, to meet the accurate tracking requirements, any kinematics based tracking control algorithm must have enough robustness to overcome the tracking performance deterioration caused by the kinematic uncertainties.

Winters and Velinsky (1992) developed a PID control algorithm for the TMR, and both numerical simulation and experiment are made based on a prototype TMR model. This control algorithm used the PID control algorithm on the basis of mobile robot inverse kinematics. Much research has been done on the wheeled mobile robot tracking control area. Kochekali and Velinsky (1994) have made a comprehensive survey of the mobile robot control problem. Kanayama (1988) proposed a PID control algorithm by kinematically transferring mobile robot posture errors in the world coordinate frame into errors in local coordinate frame. The posture errors in the local coordinate frame are then filtered by a PID algorithm to produce the control inputs, that is, the speed of the two driven wheels. Lee and Williams (1992) developed a similar PID control algorithm for wheeled mobile robots. The local position and orientation are used to determine the two driven wheels angular speeds by means of least-squares-method. The posture errors in the

world coordinate frame are transformed into errors in the local frame on the basis of kinematics. This kind of control algorithm has three basic features: one is the assumption of precise mobile robot kinematics model; the second is the PID control algorithm; and the third is that all three posture variables (two position variables and one orientation variable) are used as feedback. Since the character of the nonholonomic condition of a mobile robot's kinematics (i.e., there are three posture variables and only two control inputs), in the path tracking process, the mobile robot position and orientation are inherently coupled. Inadequate feedback of the three posture variables may influence the tracking performance. There is no guarantee of global stability and tracking performance for this kind of control algorithm. The robustness of this kind of control algorithm is also very limited.

Although much work has been done on robust control of robot manipulators, see Spong and Vidyasagar (1989), little research has been done on the robust control of a mobile robot with independent driven wheels. In this chapter, we will develop a mobile robot kinematic tracking control algorithm based on the kinematic model with uncertainties. On the basis of the kinematics features, a global exponential tracking control algorithm is proposed in the previous section. The algorithm decouples the kinematics inherent coupling of orientation control to position control, and a feedback linearized control law for position tracking is obtained. The algorithm revealed the fundamental tracking control structure of mobile robots with two independent driven wheels. A robust control algorithm is developed in this section to guarantee the exponential tracking control property when uncertainties exist. This algorithm possesses the similar calculation complexity as that of the PID based algorithm and has the advantage of being suitable for real time application. Numerical simulation is made to show the efficiency of the robust control algorithm. It is found that the robust control algorithm not only possesses the robustness ability under the condition of uncertainty, but also improves the tracking control performance under the condition of an exact kinematics model.

4.2 Robust Control Algorithm for Tethered Mobile Robot

For tracking control of the tethered mobile robot, a reasonable tracking point is the contact point of the tool and ground; that is, the tracking point is not on the center of the baseline. In Chapter 3, we proposed an exponential position tracking control algorithm for the mobile robot with the tracking point not located on the baseline. If we know the exact kinematic model of the mobile robot, from section 3.2 the control law is set to be

$$v(t) = G_p^{-1}(\psi)[\dot{q}_{p,f}(t) - K_p e_p(t)] \tag{4.1}$$

where, $q_{p,f}(t)$is the reference path, and

$$q_{p,f}(t) = \begin{bmatrix} x_{g,f}(t) & y_{g,f}(t) \end{bmatrix}^T \tag{4.2}$$

$$e_p(t) = q_p(t) - q_{p,f}(t) \tag{4.3}$$

$$K_p = \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix} \quad (k_x > 0, k_y > 0) \ . \tag{4.4}$$

Substituting the control (4.1) into the perfect kinematic model (2.33), we have

$$\dot{e}_p(t) = -K_p e_p(t) \ . \tag{4.5}$$

Taking the Lyapunov function to be

$$V = e^T(t) P e(t) \tag{4.6}$$

where P is the unique positive definite solution of the Lyapunov equation, we have

$$K_p P + P K_p = Q \tag{4.7}$$

and Q is a given positive definite symmetrical matrix. Taking the time derivative of the Lyapunov function V along the trajectory of system equation (4.5), we have

$$\dot{V} = -2k_m e^T(t)Pe(t) \tag{4.8}$$

where,

$$k_m = min[k_x, k_y].$$

That is, the control (4.1) exponentially tracks the reference path position at exponential rate k_m.

The only feasible reference orientation for the tracking control is the tangential direction of the reference path. In this case, the heading direction of the mobile robot is completely determined by the features of the reference path and the location of the tracking point on the mobile robot.

The exponential tracking control algorithm depends on the precise kinematics of the mobile robot. For the Tethered Mobile Robot, however, there are strong uncertainties in the kinematic model related to wheel slippage. Although it is difficult to get an exact kinematic model while slippage occurs, a robust control algorithm can be developed to ensure tracking control performance in this case. Robust control algorithms for model parameter uncertainties have been extensively studied, see for example, Leitmann (1981) and Corless (1993). A Lyapunov method based algorithm is used in this paper to design the robust tracking control algorithm.

We now let the control law (4.1) be determined based on the kinematic model without slippage; that is

$$\upsilon(t) = \hat{G}_p^{-1}(\psi)\alpha(t) \tag{4.9}$$

where

$$\hat{G}_p(\psi) = \frac{r}{d}\begin{bmatrix} \frac{d}{2}\cos\psi - b\sin\psi & \frac{d}{2}\cos\psi + b\sin\psi \\ \frac{d}{2}\sin\psi + b\cos\psi & \frac{d}{2}\sin\psi - b\cos\psi \end{bmatrix} \tag{4.10}$$

and r is the design radius of the wheels. Substituting the control (4.9) into the uncertainty kinematic model (2.22), we obtain

$$\dot{q}_p(t) = \alpha(t) + E(\psi)\alpha(t) \tag{4.11}$$

where

$$E(\psi) = G_p(\psi)\hat{G}_p^{-1}(\psi) - I_{2\times 2}. \tag{4.12}$$

Choosing the robust control law to be

$$\alpha(t) = \dot{q}_{p,f}(t) - K_p e_p(t) + \delta(t) \tag{4.13}$$

then we have

$$\dot{e}_p(t) = -K_p e_p(t) + \delta(t) + \eta_p(t) \tag{4.14}$$

and

$$\eta_p(t) = E(\psi)[\dot{q}_{p,f}(t) - K_p e_p(t) + \delta(t)]. \tag{4.15}$$

Since the reference path is always planned to be of limited motion speed, therefore,

$$\sup_{t\geq 0}\left\|\dot{q}_{p,f}(t)\right\| \leq \chi < \infty \ . \tag{4.16}$$

Because the longitudinal slip coefficient and the lateral slip angle are relatively very small, therefore, we suppose that

$$\left\|E(\psi)\right\| \leq \mu < 1 \ . \tag{4.17}$$

Then with a function $\rho(e_p(t)) > 0$, we take

$$\delta(t)=\begin{cases}-\rho(e_p(t))\dfrac{Pe_p(t)}{\left\|Pe_p(t)\right\|} & (\left\|Pe_p(t)\right\|\neq 0)\\ 0 & (\left\|Pe_p(t)\right\|=0)\end{cases}. \tag{4.18}$$

It can be seen that

$$\left\|\delta(t)\right\|\leq\rho(e_p(t)) \tag{4.19}$$

and the function $\rho(e_p(t),t)>0$ can be determined as follows. Consider

$$\left\|\eta_p(t)\right\|\leq\mu[\chi+\left\|K_p\right\|\cdot\left\|e_p(t)\right\|+\rho(e_p(t))] \tag{4.20}$$

and let

$$\mu[\chi+\left\|K_p\right\|\cdot\left\|e_p(t)\right\|+\rho(e_p(t))]=\rho(e_p(t)). \tag{4.21}$$

Then, we have

$$\rho(e_p(t))=\frac{\mu}{1-\mu}[\chi+\left\|K_p\right\|\cdot\left\|e_p(t)\right\|] \tag{4.22}$$

and

$$\left\|\eta_p(t)\right\|\leq\rho(e_p(t)) \ . \tag{4.23}$$

Taking the Lyapunov function to be

$$V=e_p^TPe_p,$$

then

$$\begin{aligned}\dot{V}&=\dot{e}_p^T(t)Pe_p(t)+e_p^T(t)P\dot{e}_p(t)\\&=-e_p^T(t)Qe_p^T(t)+2e_p^T(t)P[\delta(t)+\eta_p(t)]\\&\leq-2k_me_p^T(t)Pe_p^T(t)+2e_p^TP\delta(t)+2\left\|Pe_p(t)\right\|\cdot\left\|\eta_p(t)\right\|\\&\leq-2k_me_p^T(t)Pe_p^T(t)+2e_p^TP[\delta(t)+\frac{Pe_p(t)}{\left\|Pe_p(t)\right\|}\rho(e_p(t))]\\&=-2k_me_p^T(t)Pe_p^T(t)\end{aligned} \ . \tag{4.24}$$

Therefore, we have

$$\|e_p(t)\| \le \gamma \|e_p(0)\| e^{-k_m t} \quad (\gamma < \infty,\ k_m = min(k_x, k_y)). \tag{4.25}$$

That is, under the robust control algorithm, the mobile robot position exponentially tracks the reference path at an exponential rate k_m.

For the mobile robot with two independent driven wheels, the heading direction of the mobile robot is completely determined by the reference path features and the location of the tracking point on the mobile robot when the position converges to the reference path.

4.3 Numerical Simulation and Discussion

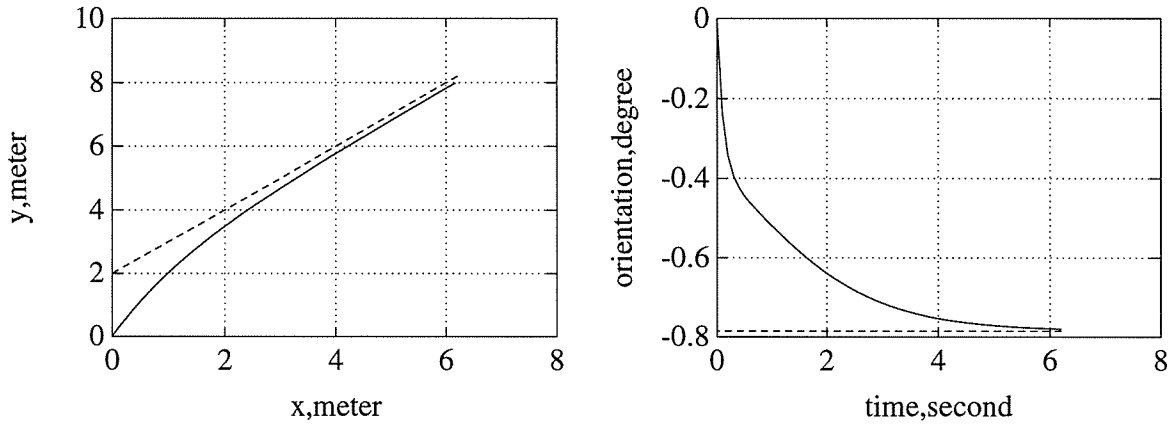
The robust exponential tracking control algorithm is simulated numerically with Matlab on a Macintosh Q80 computer. In the simulation, the feedback control parameters are set to be

$$k_x = k_y = 1.0.$$

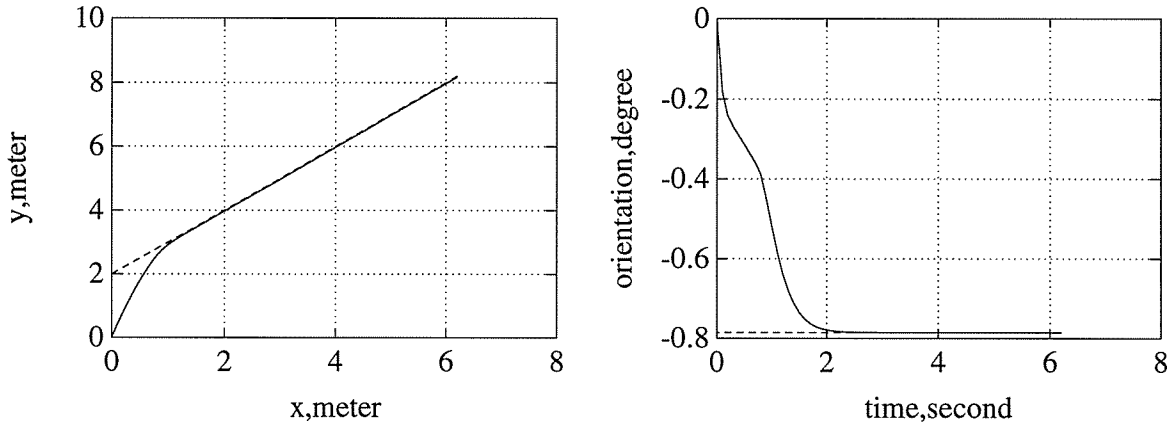
To make a comparison, the exponential tracking control (ETC) algorithm (4.1) and the robust tracking control (RTC) algorithm (4.13) are used in the simulation. We define the slippage factor to be the ratio of the effective rolling radius and the design radius of the wheel; that is

$$\sigma_i = \frac{r_{e,i}}{r} \quad (i = l, r) \ .$$

When there exists slippage, the tracking performance of the ETC algorithm deteriorates. Fig. 4.1(a) shows the tracking performance of the ETC with slippage factors to be $\sigma_l = 1.0$ *and* $\sigma_r = 0.8$ in the kinematics simulation, while the value of the slippage factors of the two wheels are taken to be 1.0, respectively, in the control law.



(a) Exponential control algorithm



(b) Robust control algorithm

Fig. 4.1 Straight Line Reference Path Tracking Control with Uncertainties

It is clear that when slip exists, the ETC algorithm cannot track the reference path exactly. Considering the control law (4.1), the control inputs (i.e., the rotation speed of the left and right wheels) are calculated based on the inverse kinematics.

When slip exists, there are uncertainties in the parameters of the kinematic model. The inverse kinematics calculation of (4.1) does not match the real kinematics of the mobile robot; therefore, tracking errors cannot be completely eliminated by the control system. The tracking performance of the robust control algorithm is as shown in Fig. 4.1 (b). Comparing with Fig. 4.1 (a) where the ETC algorithm is used under uncertainty conditions, it can be seen that the robust control algorithm

overcomes the tracking errors caused by the uncertainties. It also can be seen that the RTC algorithm has a higher convergence rate than the ETC algorithm. This is because the robust tracking control algorithm is inherently a nonlinear control algorithm, while the exponential control algorithm is basically a linear control algorithm. Despite the nonlinear features in the inverse kinematics calculation, the exponential control algorithm is a PD feedback control algorithm on the posture errors. The robust control algorithm is a nonlinear feedback control algorithm on the posture errors, and it gives higher control efforts when the value of the posture error is large.

To further illustrate the tracking ability of the robust control algorithm, a more complex reference path is used in the simulation with exact initial conditions as shown in Fig. 4.2. When slip exists, the ETC algorithm cannot exactly track the reference path as shown in Fig. 4.2(a) where the slippage factors for right and left wheel are all equal to 0.8. The tracking ability of the robust control algorithm under the uncertainty condition is shown in Fig. 4.2(b). It is clear that the RTC algorithm can track the reference path under the influence of uncertainty, and the tracking performance is greatly improved.

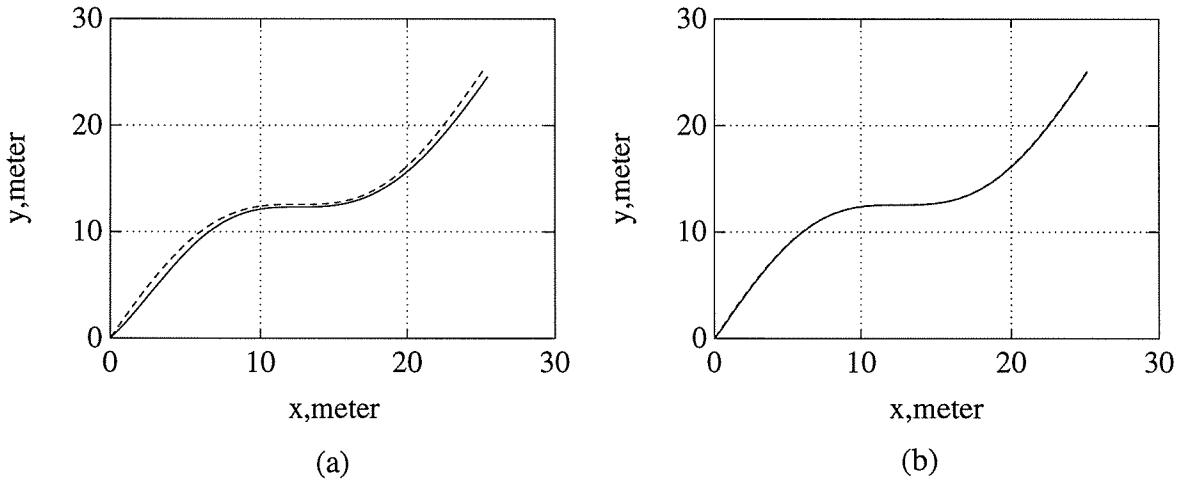


Fig. 4.2 Slope Sinusoidal Path Tracking Control with Uncertainties

4.4 Conclusion

On the basis of the kinematic features of a mobile robot with two independent driven wheels, a global exponential tracking control algorithm is proposed. This algorithm decouples the kinematics inherent coupling between orientation control and position control, and a feedback linearized control law for position tracking is obtained. This algorithm revealed the fundamental tracking control structure of mobile robots with two independent driven wheels. Considering the uncertainties existing in the mobile robot kinematics model, a robust control algorithm is developed to guarantee the exponential tracking control property when uncertainties, such as wheel slippage and pneumatic tire wheel radius errors, exist. The Tethered Mobile Robot (TMR) is being developing towards application in highway maintenance. Accurate tracking control is the basis of any maintenance operation while there are strong uncertainties due to changes of road surface conditions and different highway operation payloads. The TMR configuration has the advantage of accurate measurement of its position and orientation, and therefore, the robust control algorithm developed in this chapter is very suitable to guarantee the exponential tracking control of the TMR. The robust control algorithm is also feasible for any mobile robot with two independent driven wheels as long as its accurate position measurement is available. The efficiency of the robust control algorithm is illustrated by numerical simulation. It was found that the robust control algorithm improves the tracking control performance both when exact kinematics are known and when uncertainties exist.

CHAPTER 5

DYNAMIC MODELING

5.1 Introduction

The configuration of mobile robot moving along a planar surface is defined by three posture variables, the position coordinates and the heading direction. The purpose of tracking control cause the mobile robot to follow a desired reference path in both position and orientation. Based on the dynamic modeling work of Boyden and Velinsky (1993), it was believed that a kinematics based control algorithm is only valid for very low speed and very low payload robots. Therefore, a dynamic model based control algorithm is necessary for accurate path tracking control.

The dynamic model is the basis for the dynamic tracking control algorithm design and application. Vehicle dynamics was considered in some control algorithms for conventionally steered mobile robots. Hemami, Mehrabi and Cheng (1992) used a simplified linear dynamic model in the synthesis of a tracking control algorithm for a three-wheeled cart with front steering wheel. Nelson and Cox (1989) considered the motor dynamics in a control algorithm while still used a kinematics model in the control law design. A simplified dynamic model was used by Hemami, Mehrabi and Cheng (1990) to simulate the kinematics based control algorithm for a conventionally steered mobile robot. To control a high speed autonomous vehicle, Shin and Singh previewed the path dynamics and used acceleration limits to speed plan. A simplified dynamic model of a differential driven wheeled mobile robot is developed by Hamdy and Badreddin (1992). Only longitudinal slippage was considered in the model. Boyden and Velinsky (1993) developed dynamic models for both conventionally steered and differentially steered wheeled mobile robots with the use of Dugoff's tire friction model.

In this Chapter, we first describe the full dynamic model of the mobile robot, and introduce the concept of a perfect dynamic model. The influence of the uncertainties, such as unmodeled dynamics, slippage, payload variation, parameter perturbation, etc., on the system dynamics is then analyzed. Since exact position tracking control is very important for the Tethered Mobile

Robot, we make a coordinate transformation to get the tracking point position dynamic model. As a fundamental basis of the robust tracking control algorithm, the matching condition for the mobile robot uncertainty in the dynamic model is proven.

5.2 Full Dynamic Model

For development of the full dynamic model of the differential driven wheeled mobile robot, we assume that the robot is restricted to move on a two dimensional planar surface. A full dynamic model of a differential driven wheeled mobile robot is developed by Boyden and Velinsky (1993), and the tire payload is simplified in their model. In this section, the dynamic model is rewritten with addition of accurate tire payload and motor dynamics.

The coordinate system is selected as shown in Fig. 5.1 and Fig. 5.2, where x-o-y is the world coordinate system, and u-G-v is the local speed-coordinate fixed on the mobile robot with the origin at the mass center of the mobile robot, u in the heading direction of the robot, and v in the lateral direction.

The definition of the structural parameters are shown in Fig. 5.2. The point G is the mass center of the mobile robot. The point B is the center of the baseline. The points R and L are the wheel-ground contact points of right wheel and left wheel, respectively. The point C is the wheel-ground contact point of the castor. The point P is a mechanical linkage joint. The point E is the

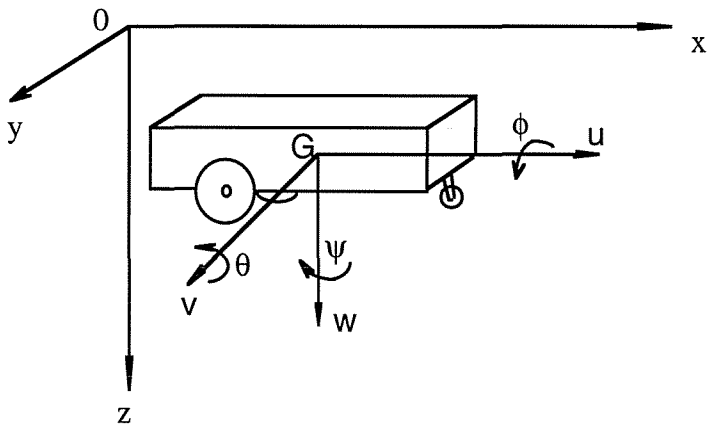


Fig. 5.1 Vehicle Coordinate System

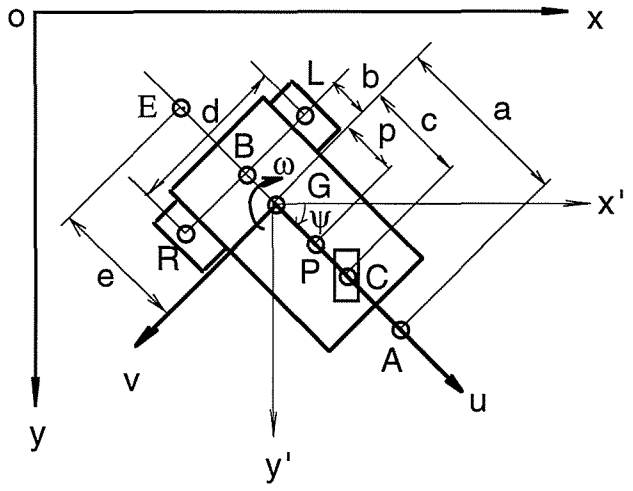


Fig. 5.2 Structural Parameters Definition

tool equipment and ground contact point. Point A is a tracking point, that is, the tracking control system controls the mobile robot to follow a desired reference path with this point. The tracking point may be a local vision sensor focus point or tool-ground contact point for an industrial mobile robot. The points R, L, E, P, and C are all the reaction force action points. The reaction force components acting on these points are as shown in Fig. 5.3, where, $F_{u,i}$, $F_{v,i}$, *and* M_i $(i = l, r, c, p, e)$ are the force component in the heading direction, the force component in the lateral direction, and the applied momentum, respectively. The force $F_{n,i}$ $(i = l, r, e, p, c)$ is the normal force acting in the negative direction of the z axis.

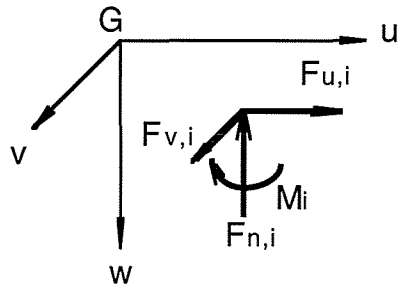


Fig. 5.3 Reaction Force Components at an Acting Point

5.2.1 *Dynamic Equations of Motion*

The dynamic equations of motion of the robotic vehicle can be deduced from Newton's second law. Because the robot moves on a planar surface, we have

$$\sum F_u = m(\dot{u} - v\omega) \tag{5.1}$$

$$\sum F_v = m(\dot{v} + u\omega) \tag{5.2}$$

$$\sum M_w = I_w \dot{\omega} \tag{5.3}$$

$$\sum F_w = 0 \tag{5.4}$$

$$\sum M_u = 0 \tag{5.5}$$

$$\sum M_v = 0 \tag{5.6}$$

where, m *and* I_w are the mass of the robot and the moment of inertia about the w axis, respectively, and

$$\sum F_u = F_{u,l} + F_{u,r} + F_{u,e} + F_{u,p} + F_{u,c} \tag{5.7}$$

$$\sum F_v = F_{v,l} + F_{v,r} + F_{v,e} + F_{v,p} + F_{v,c} \tag{5.8}$$

$$\sum M_w = \frac{d}{2}(F_{u,l} - F_{u,r}) - b(F_{v,l} + F_{v,r}) - eF_{v,e} + cF_{v,c} + pF_{v,p} + M_e + M_p \tag{5.9}$$

$$\sum F_w = mg - (F_{n,l} + F_{n,r} + F_{n,c} + F_{n,e} + F_{n,p}) \tag{5.10}$$

$$\sum M_u = \frac{d}{2}(F_{n,l} - F_{n,r}) - h_g(F_{v,l} + F_{v,r} + F_{v,c} + F_{v,e}) + h_p F_{v,p} \tag{5.11}$$

$$\sum M_v = cF_{n,c} + pF_{n,p} + h_g(F_{u,l} + F_{u,r} + F_{u,c} + F_{u,e}) - b(F_{n,l} + F_{n,r}) - eF_{n,e} - h_p F_{u,p} \tag{5.12}$$

where, h_g is the height from the mass center to the ground, and h_p is the height from the mechanical linkage joint point to the mass center. The tool equipment and mechanical linkage acting forces and momentum are external disturbance inputs to the system, and can be considered to be functions of time. That is,

$$F_{j,i} = \eta_{j,i}(t) \quad (j = u, v, n \quad i = e, p) \tag{5.13}$$

and

$$M_i = \tau_i(t) \quad (i = e, p). \tag{5.14}$$

The tire-ground contact forces $F_{j,i}$ $(j = u, v, n \quad i = l, r, c)$ are determined by the tire model. The tire model is the most complex part of the dynamic model and will be described late in this section.

5.2.2 *Wheel-motor System Dynamics*

The wheel driven system consists of pneumatic tired wheels, gearboxes, and DC motors. The dynamic model of the motors can be described as

$$I_m \dot{\omega}_{m,i}(t) + (B_m + \frac{K_t K_b}{R_a})\omega_{m,i}(t) = \frac{K_t}{R_a} v_i(t) - \tau_{m,i}(t) \quad (i = l, r) \tag{5.15}$$

where, I_m is the equivalent moment of inertia of the motor and the gearbox, B_m is the viscous friction coefficient of the motor and gearbox, K_t *and* K_b are the motor torque constant and the back emf constant, respectively, R_a is the armature resistance, and $\omega_{m,i}$, v_i, *and* $\tau_{m,i}$ are the motor angular speed, input applied armature voltage, and output payload torque. Here, $(i = l, r)$, and l *and* r correspond to the left and right wheel, respectively. The wheel mechanical motion dynamics is described a

$$I_{wh}\dot{\omega}_i(t) + B_{wh}\omega_i(t) = \tau_{m,i}(t) - F_{u,i}r \quad (i = l, r) \tag{5.16}$$

where, I_{wh} is the moment of inertia of the wheel, B_{wh} is the viscous friction of the wheel, r is the radius of the wheel, and ω_i is the angular speed of the wheel. Combining (5.15) and (5.16), we get the dynamic equations of the wheel-motor system as follows:

$$I_t\dot{\omega}_i(t) + B_t\omega_i(t) = \tau_i(t) - F_{u,i}r \quad (i = l, r) \tag{5.17}$$

and

$$I_t = I_{wh} + \sigma I_m \tag{5.18}$$

$$B_t = B_{wh} + \sigma(B_m + \frac{K_t K_b}{R_a}) \tag{5.19}$$

$$\tau_i = \frac{K_t}{R_a} v_i \quad (i = l, r) \tag{5.20}$$

where, σ is the ratio of the gearbox, and we refer to τ_l *and* τ_r as control system inputs. τ_l *and* τ_r are variables calculated from the product of the applied armature voltage and a constant coefficient.

5.2.3 *Velocity Constraints*

For wheeled mobile robots, kinematic equations provide special constraints on the velocity. In the general case, the velocity constraints for a differential driven mobile robot can be describe by the following equations (Zhang and Velinsky, 1994b):

$$u(t) = \frac{1}{2}[u_l(t) + u_r(t)] \tag{5.21}$$

$$v(t) = \frac{b}{d}[u_l(t) - u_r(t)] + v^s(t) \tag{5.22}$$

$$\omega(t) = \frac{1}{d}[u_l(t) - u_r(t)] \tag{5.23}$$

where, u_l *and* u_r are the longitudinal speed of the wheel center of the left wheel and right wheel, respectively, v^s is the lateral speed of the center of the wheels, that is the lateral slip speed. The longitudinal velocity of the center of the wheels can be further described as follows:

$$u_l(t) = r\omega_l(t) - u_l^s(t) \tag{5.24}$$

$$u_r(t) = r\omega_r(t) - u_r^s(t) \tag{5.25}$$

where, u_l^s *and* u_r^s are the longitudinal slip speed of the center of the left wheel and the right wheel, respectively. It is worthwhile to point out that the kinematic equations of (5.21)-(5.25) are valid in

general as long as the vehicle can be considered to be a rigid body. In the case of perfect wheel-ground contact, that is, $u_l^s = 0, u_r^s = 0 \text{ and } v^s = 0$, we have

$$v(t) = b\omega(t) \tag{5.26}$$

$$u_l(t) = r\omega_l(t) \tag{5.27}$$

$$u_r(t) = r\omega_r(t). \tag{5.28}$$

Equations (5.26)-(5.28) are only valid when there is no slippage.

5.2.4 *Pneumatic Tire Model*

Dugoff's friction circle concept is used to describe the pneumatic tire forces for mobile robots and common vehicle dynamics (Boyden and Velinsky, 1993, Guntur and Sankar, 1980). On the basis of Dugoff's tire friction model, the longitudinal and lateral forces acting on the wheels and the castor can be described as

$$F_{j,i} = \mu_{j,i} F_{n,i} \quad (j = u, v \quad i = l, r, c) \tag{5.29}$$

where

$$\mu_{j,l} = \mu_{j,l}(u, v, \omega, u_l, \omega_l, \tau_l) \quad (j = u, v) \tag{5.30}$$

$$\mu_{j,r} = \mu_{j,r}(u, v, \omega, u_r, \omega_r, \tau_r) \quad (j = u, v) \tag{5.31}$$

$$\mu_{j,c} = \mu_{j,c}(u, v, \omega) \quad (j = u, v). \tag{5.32}$$

From (5.4)-(5.6) and (5.10)-(5.12) and (5.29), we have

$$\begin{bmatrix} F_{n,l} \\ F_{n,r} \\ F_{n,c} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ \frac{d}{2} - h_g\mu_{v,l} & -\frac{d}{2} - h_g\mu_{v,r} & -h_g\mu_{v,c} \\ h_g\mu_{u,l} - b & h_g\mu_{u,r} - b & h_g\mu_{u,c} + c \end{bmatrix}^{-1} \begin{bmatrix} mg - \eta_{n,e} - \eta_{n,p} \\ h_g\eta_{v,e} - h_p\eta_{v,p} \\ e\eta_{n,e} + h_p\eta_{u,p} - h_g\eta_{u,e} - p\eta_{n,p} \end{bmatrix}. \tag{5.33}$$

5.2.5 *Full Dynamic System Equations*

From the above description, we can rewrite the dynamic system equations as follows. From (5.1)-(5.3), (5.7)-(5.9), and (5.17), we have

$$\dot{u} = \frac{1}{m}(F_{u,l} + F_{u,r} + F_{u,c} + \eta_{u,e} + \eta_{u,p}) + v\omega \tag{5.34}$$

$$\dot{v} = \frac{1}{m}(F_{v,l} + F_{v,r} + F_{v,c} + \eta_{v,e} + \eta_{v,p}) - u\omega \tag{5.35}$$

$$\dot{\omega} = \frac{1}{I_w}[\frac{d}{2}(F_{u,l} - F_{u,r}) - b(F_{v,l} + F_{v,r}) + cF_{v,c} - e\eta_{v,e} + p\eta_{v,p} + \tau_e + \tau_p] \tag{5.36}$$

$$\dot{\omega}_l(t) = \frac{1}{I_t}[\tau_l(t) - F_{u,l}r - B_t\omega_l(t)] \tag{5.37}$$

$$\dot{\omega}_r(t) = \frac{1}{I_t}[\tau_r(t) - F_{u,r}r - B_t\omega_r(t)]. \tag{5.38}$$

The motion of the tracking point A in the world coordinates can be described by the following equations:

$$\dot{x}(t) = u(t)\cos\psi(t) - [v(t) + (a-b)\omega(t)]\sin\psi(t) \tag{5.39}$$

$$\dot{y}(t) = u(t)\sin\psi(t) + [v(t) + (a-b)\omega(t)]\cos\psi(t) \tag{5.40}$$

$$\dot{\psi}(t) = \omega(t). \tag{5.41}$$

Equations (5.34)-(5.41) are the full dynamic system equations of the differentially driven wheeled mobile robot. The equations are also valid for any differentially driven wheeled ground vehicles. The main difficulty in applying the above full dynamic model to control system design is that the tire model is complex and nonlinear and it is impossible to get any closed form analytical control system synthesis results. Herein, we deal with the dynamic control problem by using a simplified dynamic model, which we refer as a perfect dynamic model, to synthesize the control system while maintaining the same control performance as that based on the full dynamic model.

5.3 Perfect Dynamic Model

We define the perfect dynamic model to be a wheeled mobile robot on which the perfect wheel-ground contact condition always holds and there exists no external disturbance inputs, such as equipment tool forces, mechanical linkage forces, etc. We also neglect the viscous friction force and the resistance force of the castor. For the perfect dynamic model, the system dynamic equations (5.34)-(5.38) become

$$\dot{u} = \frac{1}{m}(F_{u,l} + F_{u,r}) + v\omega \tag{5.42}$$

$$\dot{v} = \frac{1}{m}(F_{v,l} + F_{v,r}) - u\omega \tag{5.43}$$

$$\dot{\omega} = \frac{1}{I_w}[\frac{d}{2}(F_{u,l} - F_{u,r}) - b(F_{v,l} + F_{v,r})] \tag{5.44}$$

$$\dot{\omega}_l(t) = \frac{1}{I_t}[\tau_l(t) - F_{u,l}r] \tag{5.45}$$

$$\dot{\omega}_r(t) = \frac{1}{I_t}[\tau_r(t) - F_{u,r}r] \tag{5.46}$$

and, the kinematic equations (5.26)-(5.28) are valid; that is

$$u(t) = \frac{r}{2}[\omega_l(t) + \omega_r(t)] \tag{5.47}$$

$$\omega(t) = \frac{r}{d}[\omega_l(t) - \omega_r(t)] \tag{5.48}$$

and

$$v(t) = b\omega(t). \tag{5.49}$$

From (5.45), (5.46), and (5.47), we have

$$F_{u,l} + F_{u,r} = \frac{1}{r}(\tau_l + \tau_r) - \frac{2I_t}{r^2}\dot{u}. \tag{5.50}$$

From (5.45), (5.46), and (5.48), we obtain

$$F_{u,l} - F_{u,r} = \frac{1}{r}(\tau_l - \tau_r) - \frac{I_t d}{r^2}\dot{\omega}. \tag{5.51}$$

From (5.43) and (5.49), we have

$$F_{v,l} + F_{v,r} = m(b\dot{\omega} + u\omega). \tag{5.52}$$

Therefore, substituting (5.50) and (5.49) into (5.42), we have

$$\dot{u}(t) = \frac{mbr^2}{\Theta_u}\omega^2(t) + \frac{r}{\Theta_u}[\tau_l(t) + \tau_r(t)] \tag{5.53}$$

where

$$\Theta_u = mr^2 + 2I_t.$$

Substituting (5.51) and (5.52) into (5.44), we obtain

$$\dot{\omega}(t) = -\frac{2bmr^2}{\Theta_\omega}u(t)\omega(t) + \frac{rd}{\Theta_\omega}[\tau_l(t) - \tau_r(t)] \tag{5.54}$$

where

$$\Theta_\omega = I_t d^2 + 2r^2(I_w + mb^2).$$

Now, substituting (5.49) into the motion trajectory equations of (5.39)-(5.41), we obtain

$$\dot{x}(t) = u(t)\cos\psi(t) - a\omega(t)\sin\psi(t) \tag{5.55}$$

$$\dot{y}(t) = u(t)\sin\psi(t) + a\omega(t)\cos\psi(t) \tag{5.56}$$

$$\dot{\psi}(t) = \omega(t). \tag{5.57}$$

Denoting

$$\tau_u(t) = \frac{1}{2}[\tau_l(t) + \tau_r(t)] \tag{5.58}$$

$$\tau_\omega(t) = \frac{1}{2}[\tau_l(t) - \tau_r(t)] \tag{5.59}$$

and rewriting the equations (5.53)-(5.57) in a compact form, we have

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\psi}(t) \\ \dot{u}(t) \\ \dot{\omega}(t) \end{bmatrix} = \begin{bmatrix} u(t)\cos\psi(t) - a\omega(t)\sin\psi(t) \\ u(t)\sin\psi(t) + a\omega(t)\cos\psi(t) \\ \omega(t) \\ \frac{mbr^2}{\Theta_u}\omega^2(t) \\ -\frac{2bmr^2}{\Theta_\omega}u(t)\omega(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{2r}{\Theta_u} & 0 \\ 0 & \frac{2rd}{\Theta_\omega} \end{bmatrix} \begin{bmatrix} \tau_u(t) \\ \tau_\omega(t) \end{bmatrix}. \tag{5.60}$$

The equations (5.60) are the perfect dynamic model of the differentially driven wheeled mobile robot. Although the perfect dynamic model is only a simplified dynamic model, it is rigorous in mechanical principle under the perfect conditions. In other words, there is no vague physical process model being used in the perfect dynamic model; the only physical principle used in the model is Newton's second law and rigid body kinematics relations. Since the perfect dynamic model is developed on the basis of certain perfect conditions, there are uncertainties, such as unmodeled dynamics, external disturbance inputs, viscous resistance, parameter perturbations, etc., that exist in the perfect dynamic model compared to the full dynamic model. In order to develop a tracking control algorithm based on the perfect dynamic model while still guaranteeing high tracking performance, the control algorithm must possess robustness to all the uncertainties.

5.4 Reduced-Order Dynamic Model with Uncertainties

To analyze the uncertainties, we consider the kinematic equations of (5.21)-(5.25). These kinematic equations are valid for the system in any case. From (5.21) and (5.23), we have

$$u_l = u + \frac{d}{2}\omega \tag{5.61}$$

and

$$u_r = u - \frac{d}{2}\omega . \tag{5.62}$$

Then, from (5.24) and (5.25), we have

$$\omega_l = \frac{1}{r}(u + \frac{d}{2}\omega + u_l^s) \tag{5.63}$$

and

$$\omega_r = \frac{1}{r}(u - \frac{d}{2}\omega + u_r^s). \tag{5.64}$$

Substituting equations (5.37) and (5.38) into equation (5.34) and considering the kinematic equations (5.22) and (5.61)-(5.64), we obtain

$$\dot{u}(t) = \frac{mbr^2}{\Theta_u}\omega^2(t) + \frac{2r}{\Theta_u}\tau_u(t) + \delta_u \tag{5.65}$$

where

$$\begin{aligned}\delta_u = \frac{1}{\Theta_u}\{ & mr^2\omega(t)v^s(t) + r^2[\eta_{u,c}(t) + \eta_{u,e}(t) + \eta_{u,p}(t)] \\ & - B_t[2u(t) + u_l^s(t) + u_r^s(t)] - I_t[\dot{u}_l^s(t) + \dot{u}_r^s(t)]\}\end{aligned} . \tag{5.66}$$

Then, substituting (5.37), (5.38), and (5.35) into (5.36) and considering the kinematic equations (5.22) and (5.61)-(5.64), we have

$$\dot{\omega}(t) = -\frac{2bmr^2}{\Theta_\omega}u(t)\omega(t) + \frac{2rd}{\Theta_\omega}\tau_\omega(t) + \delta_\omega \tag{5.67}$$

where

$$\delta_\omega = \frac{1}{\Theta_\omega}\{-B_t d^2\omega(t) - B_t d[u_l^s(t) - u_r^s(t)] - I_t d[\dot{u}_l^s(t) - \dot{u}_r^s(t)] - 2mbr^2\dot{v}^s(t) + 2r^2[(b-e)\eta_{v,e}(t) + (a+b)\eta_{v,c}(t) + (c+b)\eta_{v,p}(t) + \tau_e(t) + \tau_p(t)]\} \quad (5.68)$$

Considering kinematic equations (5.22) and (5.23), the motion trajectory of the tracking point A in the world coordinates can be described as

$$\dot{x}(t) = u(t)\cos\psi(t) - a\omega(t)\sin\psi(t) - v^s(t)\sin\psi(t) \quad (5.69)$$

$$\dot{y}(t) = u(t)\sin\psi(t) + a\omega(t)\cos\psi(t) + v^s(t)\cos\psi(t) \quad (5.70)$$

and

$$\dot{\psi}(t) = \omega(t) \ . \quad (5.71)$$

Now, we denote

$$\delta_x = -v^s(t)\sin\psi(t)$$

$$\delta_y = v^s(t)\cos\psi(t)$$

and the system equations (5.65), (5.67), and (5.69)-(5.71) can be rewritten in a vector form corresponding to the perfect dynamic model equations (5.60) as

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\psi}(t) \\ \dot{u}(t) \\ \dot{\omega}(t) \end{bmatrix} = \begin{bmatrix} u(t)\cos\psi(t) - a\omega(t)\sin\psi(t) \\ u(t)\sin\psi(t) + a\omega(t)\cos\psi(t) \\ \omega(t) \\ \frac{mbr^2}{\Theta_u}\omega^2(t) \\ -\frac{2bmr^2}{\Theta_\omega}u(t)\omega(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{2r}{\Theta_u} & 0 \\ 0 & \frac{2rd}{\Theta_\omega} \end{bmatrix} \begin{bmatrix} \tau_u(t) \\ \tau_\omega(t) \end{bmatrix} + \begin{bmatrix} \delta_x \\ \delta_y \\ 0 \\ \delta_u \\ \delta_\omega \end{bmatrix} . \quad (5.72)$$

The system equations (5.72) are the exact dynamic equations of the mobile robot provided that $\delta_x, \delta_y, \delta_u, and\ \delta_\omega$ can be exactly measured. It is clear that $\delta_x, \delta_y, \delta_u, and\ \delta_\omega$ are uncertainties consisting of disturbances due to external inputs, unmodeled dynamics, viscous resistance, and slippage. Compared to equations (5.60), it is evident that the system equations (5.72) are the

perfect dynamic model (5.60) plus a vector of uncertainties. This means that the tracking control system of the mobile robot can be designed based on the perfect dynamic model as long as the control algorithm is robust to the uncertainties. We refer to (5.72) as the uncertainty dynamic model of the system.

5.5 Position Coordinate Transformation

In our previous work based on kinematic modeling in Chapter 2 and 3, we have pointed out that there exists an inherent heading direction error as long as the tracking point is not on the center of the baseline for this type of mobile robot. In this report we mainly consider the position tracking control problem with the tracking point not on the baseline. Therefore, we first introduce a position coordinate transformation to the reduced order dynamic system. Let us denote

$$q_1 = \begin{bmatrix} x \\ y \end{bmatrix} \tag{5.73}$$

and

$$q_2 = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}. \tag{5.74}$$

Then, we have

$$\dot{q}_1 = q_2 \tag{5.75}$$

and

$$\dot{q}_2 = \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix}. \tag{5.76}$$

From (5.69) and (5.70), we have

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} u \\ a\omega + v^s \end{bmatrix}. \tag{5.77}$$

Now, taking the time derivative of both sides of equation (5.77), we have

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} \dot{u} - a\omega^2 - v^s\omega \\ a\dot{\omega} + u\omega + \dot{v}^s \end{bmatrix}. \tag{5.78}$$

Substituting equations (5.65) and (5.67) into (5.78) gives

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} \dfrac{mr^2(b-a)-2aI_t}{\Theta_u}\omega^2 + \dfrac{2r}{\Theta_u}\tau_u + \delta_u - v^s\omega \\ \dfrac{I_t d^2 + 2r^2[I_w + mb(b-a)]}{\Theta_\omega}u\omega + \dfrac{2rad}{\Theta_\omega}\tau_\omega + a\delta_\omega + \dot{v}^s \end{bmatrix}. \tag{5.79}$$

From (5.77), we can solve

$$u(\psi, q_2) = \dot{x}\cos\psi + \dot{y}\sin\psi \tag{5.80}$$

and

$$\omega(\psi, q_2, v^s) = \frac{1}{a}(\dot{y}\cos\psi - \dot{x}\sin\psi - v^s). \tag{5.81}$$

Rearranging (79), we obtain

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = D(\psi)[s(\psi, q_2, \delta_m) + \tau + \varsigma(\psi, q_2, \delta_m)] \tag{5.82}$$

where,

$$D(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} \dfrac{2r}{\Theta_u} & 0 \\ 0 & \dfrac{2rad}{\Theta_\omega} \end{bmatrix} \tag{5.83}$$

$$s(\psi, q_2, \delta_m) = \begin{bmatrix} \dfrac{mr^2(b-a)-2aI_t}{2r}\omega^2(\psi, q_2, v^s) \\ \dfrac{I_t d^2 + 2r^2[I_w + mb(b-a)]}{2rad}u(\psi, q_2)\omega(\psi, q_2, v^s) \end{bmatrix} \tag{5.84}$$

$$\varsigma(\psi, q_2, \delta_m) = \begin{bmatrix} \frac{\Theta_u}{2r}[\delta_u(\psi, q_2, \delta_m) - v^s\omega(\psi, q_2, v^s)] \\ \frac{\Theta_\omega}{2rad}[a\delta_\omega(\psi, q_2, \delta_m) + \dot{v}^s] \end{bmatrix} \tag{5.85}$$

$$\omega(\psi, q_2, \delta_m) = \omega(\psi, q_2, v^s) \tag{5.86}$$

$$\tau = \begin{bmatrix} \tau_u \\ \tau_\omega \end{bmatrix} \tag{5.87}$$

and, δ_m represents the unmodeled uncertainty vector. Then, through the coordinate transformation (5.73) and (5.74), the system equations (5.72) become

$$\dot{q}_1 = q_2 \tag{5.88}$$

$$\dot{q}_2 = D(\psi)[s(\psi, q_2, \delta_m) + \tau + \varsigma(\psi, q_2, \delta_m)] \tag{5.89}$$

and the orientation equation is

$$\dot{\psi} = \omega(\psi, q_2, \delta_m). \tag{5.90}$$

From the definition of q_1 *and* q_2, equations (5.88) and (5.89) are the system equations describing the position of the mobile robot.

It is easy to verify that through the coordinate transformation (5.73) and (5.74) the perfect dynamic model equations (5.60) can be transformed into

$$\dot{q}_1 = q_2 \tag{5.91}$$

$$\dot{q}_2 = D(\psi)[s_0(\psi, q_2) + \tau] \tag{5.92}$$

and

$$\dot{\psi} = \omega_0(\psi, q_2) \tag{5.93}$$

where,

$$s_0(\psi, q_2) = \begin{bmatrix} \dfrac{mr^2(b-a) - 2aI_t}{2r}\omega_0^2(\psi, q_2) \\ \dfrac{I_t d^2 + 2r^2[I_w + mb(b-a)]}{2rad}u(\psi, q_2)\omega_0(\psi, q_2) \end{bmatrix} \tag{5.94}$$

and

$$\omega_0(\psi, q_2) = \frac{1}{a}(\dot{y}\cos\psi - \dot{x}\sin\psi). \tag{5.95}$$

5.6 Matching Condition

It is important to consider a special feature of the mobile robot dynamic model structure. We can see in the following that all the uncertainties in the uncertainty dynamic model equations enter the system through the input channel. This is called a 'matching condition'. It is well known that if the matching condition is hold, a robust control algorithm can then be developed for the tracking control. It is also well known that if the variable structure control method is applied to the control system design, the sliding mode of the system is invariant under the influence of the uncertainties which enter the system through input channel.

We rewrite the uncertainty dynamic system model, equations (5.88) and (5.89), in a compact form as

$$\dot{q} = f(q, \psi, \delta_m) + \xi(q, \psi, \delta_m) + B(\psi)\tau \tag{5.96}$$

where

$$B(\psi) = \begin{bmatrix} 0_{2\times 2} \\ D(\psi) \end{bmatrix} \tag{5.97}$$

$$f(q, \psi, \delta_m) = \begin{bmatrix} q_2 \\ 0_{2\times 2} \end{bmatrix} + B(\psi)s(\Psi, q_2, \delta_m) \tag{5.98}$$

and

$$\xi(q, \psi, \delta_m) = B(\psi)\varsigma(\psi, q_2, \delta_m). \tag{5.99}$$

In equations (5.96), $\xi(q, \psi, \delta_m)$ is the unmodeled uncertainty term, and the term $f(q, \psi, \delta_m)$ consists of unmodeled uncertainty as well. Furthermore, there may be parameter uncertainty in the term $f(q, \psi, \delta_m)$ and the input matrix $B(\psi)$.

Theorem 5.6.1 (Matching Condition): The uncertainty dynamic model (5.96) satisfies the matching condition, that is,

$$f(q, \psi, \delta_m) = f_0(q, \psi) + B_0(\psi)\tilde{f}(q, \psi, \delta_m) \tag{5.100}$$

$$B(\psi, \delta_m) = B_0(\psi) + B_0(\psi)\tilde{B}(\psi) \tag{5.101}$$

$$\xi(q, \psi, \delta_m) = B_0(\psi)\tilde{\xi}(q, \psi, \delta_m) \tag{5.102}$$

where,

$$B_0(\psi) = \begin{bmatrix} 0 \\ D_0(\psi) \end{bmatrix} \tag{5.103}$$

and $\tilde{f}(q, \psi, \delta_m)$, $\tilde{B}(q, \psi, \delta_m)$, *and* $\tilde{\xi}(q, \psi, \delta_m)$ are vector and matrix with appropriate dimensions.

Proof: Let us denote

$$B(\psi) = B_0(\psi) + \Delta B(\psi) \tag{5.104}$$

$$f(q, \psi, \delta_m) = f_o(q, \psi) + \Delta f(q, \psi, \delta_m) \tag{5.105}$$

where, $B_0(\psi)$ *and* $f_0(q, \psi)$ are known functions, and $\Delta B(\psi)$ *and* $\Delta f(q, \psi, \delta_m)$ are uncertainty terms. To check the matching condition of equation (5.100)-(5.102), we need only to prove that

$$\Delta B(\psi) = B_0(\psi)\tilde{B}(\psi) \tag{5.106}$$

$$\Delta f(q, \psi, \delta_m) = B_0(\psi)\tilde{f}(q, \psi, \delta_m) \tag{5.107}$$

$$\xi(q, \psi, \delta_m) = B_0(\psi)\tilde{\xi}(q, \psi, \delta_m) \tag{5.108}$$

where, $\tilde{B}(\psi)$, $\tilde{f}(q, \psi, \delta_m)$, *and* $\tilde{\xi}(q, \psi, \delta_m)$ are an appropriate matrix and vector with uncertainty, respectively. The matching conditions (5.106)-(5.108) can be proved directly from the definition in (5.83)-(5.86) and (5.97)-(5.99). It can be seen from (5.83) that the matrix $D(\psi)$ is nonsingular. Then, from (5.83) and (5.97), we have

$$\begin{aligned}\Delta B(\psi) &= \begin{bmatrix} 0 \\ \Delta D(\psi) \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ D_0(\psi) \end{bmatrix} D_0^{-1}(\psi)\Delta D(\psi). \\ &= B_0(\psi)D_0^{-1}(\psi)\Delta D(\psi) \\ &=: B_0(\psi)\tilde{B}(\psi)\end{aligned} \tag{5.109}$$

The condition (5.106) is then proven. Substituting (5.84) and (5.106) into (5.98), we have

$$\begin{aligned}\Delta f(q, \psi, \delta_m) &= [\begin{bmatrix} q_2 \\ 0 \end{bmatrix} + B(\psi)s(\psi, q_2, \delta_m)] - [\begin{bmatrix} q_2 \\ 0 \end{bmatrix} + B_0(\psi)s_o(\psi, q_2)] \\ &= [B_0(\psi) + \Delta B(\psi)][s_0(\psi, q_2) + \Delta s(\psi, q_2, \delta_m)] - B_0(\psi)s_o(\psi, q_2) \\ &= B_0(\psi)\Delta s(\psi, q_2, \delta_m) + \Delta B(\psi)[s_0(\psi, q_2) + \Delta s(\psi, q_2, \delta_m)] \\ &= B_0(\psi)\{\Delta s(\psi, q_2, \delta_m) + \tilde{B}(\psi)[s_0(\psi, q_2) + \Delta s(\psi, q_2, \delta_m)]\} \\ &=: B_0(\psi)\tilde{f}(q, \psi, \delta_m)\end{aligned} \tag{5.110}$$

Thus, the condition (5.107) is proven. Finally, substituting (5.85) and (5.106) into (5.99), we have

$$\begin{aligned}\xi(q, \psi, \delta_m) &= [B_0(\psi) + \Delta B(\psi)]\varsigma(\psi, q_2, \delta_m) \\ &= B_0(\psi)[I + \tilde{B}(\psi)]\varsigma(\psi, q_2, \delta_m) \\ &=: B_0(\psi)\tilde{\xi}(q, \psi, \delta_m)\end{aligned} \tag{5.111}$$

and the condition (5.108) is proven. Therefore, the matching conditions (5.100)-(5.102) hold for the system (5.96). The matching condition is a general basis for the robust tracking control algorithm.

In (5.100)-(5.102) the uncertainty influence includes the unmodeled dynamics, external disturbances, and model parameter perturbations. It is clear that (5.100)-(5.102) implies that all the uncertainties enter the dynamic system through the control input channel. Theorem 5.6.1 provides a fundamental basis for robust dynamic tracking control algorithms.

5.7 Conclusion

Three types of dynamic models for a mobile robot with two independent driven wheels are developed in this chapter. The full dynamic model is a detailed model with a pneumatic tire and ground friction model. A perfect dynamic model is developed based on the perfect tire and ground contact condition. A reduced order dynamic model with uncertainties is also developed. It is found that the uncertainty dynamic model is the perfect dynamic model plus an uncertainty vector term. The basic idea in this chapter is that though a full dynamic model is too complex for any control algorithm design, the influence of the uncertainties to the perfect dynamic model (i.e., unmodeled dynamics, external disturbance inputs, viscous friction influences, and parameters perturbations) can be analyzed. It is proven that the matching condition is hold for the mobile robot uncertainty dynamic model. The matching condition provides a fundamental basis for the robust control algorithm design for the dynamic tracking control algorithm of the mobile robot.

CHAPTER 6

DYNAMIC MODEL BASED TRACKING CONTROL

6.1 Introduction

Differentially driven wheeled mobile robots possess both the advantages of higher tracking ability and simple wheel configuration. Tracking performance is vital for mobile robots which are developed to perform certain industry operations, such as highway crack sealing. Tracking control algorithms for wheeled mobile robots have been extensively studied in recent years. Most of the tracking control algorithms are developed based on kinematic models with perfect wheel-ground contact assumption. In this chapter, we deal with the tracking control problem of differentially driven wheeled mobile robots based on dynamic models.

The configuration of a mobile robot moving along a planar surface is dictated by three posture variables, the position variables and the heading direction. The purpose of tracking control is to cause the mobile robot to follow a desired reference path in position and orientation. Tracking control algorithms for differentially driven wheeled mobile robots are developed by many authors, for example, Kanayama, Nilipour, and Lelm (1988), Kanayama, Kimura, Miyazaki, and Noguchi (1990), Nelson (1989), Badreddin (1992), and Lee and Williams (1993). The kinematic equations of the mobile robots include nonholonomic constraints on the time derivative of the posture variables. Many authors, such as Walsh, et al., (1994), Campion, Bastin, and d'Andrea-Novel (1993), d'Andrea-Novel, Bastin, and Campion (1991, 1992), and Pomet, et al. (1992) analyzed the nonholonomic features of the mobile robot control problem. A basic feature of these algorithms is that they are developed on the basis of the mobile robot kinematics. A general assumption among these algorithms is that there is no slippage and the wheel ground contact is perfect. In Chapter 2, we studied the influence of the tracking point on the tracking performance of differentially driven wheeled mobile robots. An exponential position tracking control algorithm and heading direction equation are developed in Chapter 3. On the basis of the kinematic model, a

robust tracking control algorithm considering slippage and parameter uncertainties is also developed in Chapter 4. It was believed that the kinematics based control algorithm would only be valid for very low motion speed and very low payloads. Because the real control input is the input voltage of the motors, the kinematics based control algorithm must be matched with a motor control loop algorithm. An inadequately synthesized motor control loop will degrade tracking control performance, while the synthesis of the motor control loop is strongly related to both motor dynamics and vehicle dynamics. Therefore, a dynamic model based control algorithm is necessary for accurate path tracking control.

Vehicle dynamics have been considered in some control algorithms for conventionally steered mobile robots. Hemami, Mehrabi, and Cheng (1992) used a simplified linear dynamic model in the synthesis of a tracking control algorithm for a three-wheeled cart with front steeredg wheel. Nelson and Cox (1989) considered the motor dynamics in a control algorithm while still using a kinematics model in the control law design. A simplified dynamic model was used by Hemami, Mehrabi and Cheng (1990) to simulate the kinematics based control algorithm for a conventionally steered mobile robot. To control a high speed autonomous vehicle, Shin and Singh previewed the path dynamics and used an acceleration limit for speed planning. A simplified dynamic model of differentially driven wheeled mobile robot was developed by Hamdy and Badreddin (1992). Only longitudinal slippage was considered in their model. Boyden and Velinsky (1993) developed dynamic models for both differentially driven and conventionally steered wheeled mobile robots and used Dugoff's tire friction model to accurately account for the wheel ground interface. To the authors knowledge, there has been no control algorithm developed on the basis of a full dynamic model of a wheeled mobile robot. Due to the nonholonomic constraints and the complexity of modeling the interaction between the ground and the vehicle, a full dynamic model based control algorithm was previously considered not to be practical. In this paper, we take the full dynamic model as a basis, and an exponential tracking control algorithm is then developed based on the perfect dynamic model, and robustness to uncertainties is guaranteed. In this way, the robust

tracking control algorithm gives high tracking control performance when unmodeled dynamics and external disturbances exist.

6.2 Tracking Error Equations

For the tracking control problem, we desire the mobile robot to follow a reference trajectory $q_1^d(t) = [x^d(t), y^d(t)]^T$. It is convenient to describe the tracking control problem in the tracking error space. Let us form the tracking error vector as

$$e_1(t) = q_1(t) - q_1^d(t) \tag{6.1}$$

$$e_2(t) = q_2(t) - q_2^d(t) = \dot{q}_1(t) - \dot{q}_1^d(t) \tag{6.2}$$

and

$$e(t) = \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix}. \tag{6.3}$$

It can be seen from (5.83) that the matrix $D(\psi)$ is nonsingular. Let us set

$$\tau(t) = D_0^{-1}(\psi(t))[\upsilon(t) + \dot{q}_2^d(t)] - s_0(\psi(t), q_2(t)) \tag{6.4}$$

where $\upsilon \ni R^2$ is the new control input vector. The position tracking error equations for the perfect dynamic model equations (5.91)-(5.93) are

$$\dot{e}(t) = Ae(t) + B\upsilon(t) \tag{6.5}$$

where

$$A = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \tag{6.6}$$

and

$$B = \begin{bmatrix} 0 \\ I \end{bmatrix}. \tag{6.7}$$

It is clear that the feedback control law (6.4) linearizes the position tracking error equations of the perfect dynamic model. For the dynamic model (5.88)-(5.90), the tracking error equations become

$$\dot{e}(t) = Ae(t) + B[\upsilon(t) + \chi(\psi, q_2, \delta_m)] \tag{6.8}$$

where

$$\chi(\psi, q_2, \delta_m) = D(\psi)[\Delta s(\psi, q_2, \delta_m) + \varsigma(\psi, q_2, \delta_m)] \tag{6.9}$$

is the uncertainty term. It is reasonable to suppose that the uncertainty term is bounded; that is

$$\left\|\chi(\psi, q_2, \delta_m)\right\| = \left\|D(\psi)[\Delta s(\psi, q_2, \delta_m) + \varsigma(\psi, q_2, \delta_m)]\right\| \le \chi_m \tag{6.10}$$

where, $0 \le \chi_m < \infty$ is a given number.

The purpose of the tracking control problem is then to design a control algorithm for $\upsilon(t)$ to guarantee tracking error convergence to zero; that is, $e(t) \to 0, t \to \infty$. For the perfect dynamic system (6.5), we have the following lemma.

Lemma 6.2.1: There exist matrices $K_1 \in R^{2\times 2}$ *and* $K_2 \in R^{2\times 2}$ and the control law

$$\upsilon(t) = -K_1 e_1(t) - K_2 e_2(t) \tag{6.11}$$

such that the tracking error of the perfect dynamic model described by (6.5) exponentially converges to zero, or equivalently, for a given symmetric positive definite matrix $Q > 0$, there exists a unique symmetric positive definite matrix $P > 0$ to be the solution of the Lyapunov equation

$$\overline{A}P + P\overline{A} + Q = 0 \tag{6.12}$$

where $\overline{A} = A - BK$, *and* $K = [K_1 \quad K_2]$.

Proof: Considering that (A, B) is a controllable pair, then there exist matrices K_1 *and* K_2 such that the closed-loop system matrix $\overline{A} = A - BK$ is stable; that is, $Re\lambda(\overline{A}) < 0$. Therefore, the tracking error exponentially converges to zero.

6.3 Robust Tracking Control Algorithm

The main difficulty in the mobile robot dynamic model based control algorithm development and application is the influence of uncertainty. This is because real-time operation limits the complexity of the dynamic model, while it is impossible to accurately describe the motion with low order and simple equations. As shown in the previous section, low order dynamic system equations include uncertainty terms. Robustness is the most important property for the dynamic model based control algorithm. In this section, we will produce the robust tracking control algorithm based on the Lyapunov method.

From Lemma 4.1, the tracking error of the perfect dynamic system (6.5) exponentially converges to zero with the control (6.11). In the dynamic system equation (6.8), there exists an uncertainty term, and we can see that the uncertainty dynamic system (6.8) is equal to the perfect dynamic system (6.5) plus an uncertainty term $\chi(\psi, q_2, \delta_m)$ which enters the system through the input channel. We set the control for the uncertainty system (6.8) to be

$$\upsilon(t) = \upsilon_0(t) + \Delta\upsilon(t) \tag{6.13}$$

where

$$\upsilon_0 = -K_1 e_1 - K_2 e_2, \tag{6.14}$$

and $\Delta\upsilon$ represent the robust control term. With control (6.13), the uncertainty dynamic system equation (6.08) becomes

$$\dot{e} = \overline{A}e + B(\Delta\upsilon + \chi). \tag{6.15}$$

We take the robust control term to be

$$\Delta v(t) = \begin{cases} -\dfrac{B^T P e(t)}{\left\| B^T P e(t) \right\|} \chi_m & (\left\| B^T P e(t) \right\| \neq 0) \\ 0 & (\left\| B^T P e(t) \right\| = 0) \end{cases} \tag{6.16}$$

which leads to the following theorem.

Theorem 6.3.1: The tracking error of the uncertainty dynamic system (6.08) exponentially converges to zero under the control (6.13) with $v_0(t)$ *and* $\Delta v(t)$ determined by (6.14) and (6.16), respectively.

Proof: From Lemma 6.2.1, the matrices K_1 *and* K_2 can be selected to make the system matrix $\overline{A}$ in (6.15) stable, and for a given symmetric positive definite matrix $Q > 0$, there exists a unique symmetric positive definite matrix $P > 0$ that is the solution of the Lyapunov equation

$$\overline{A}P + P\overline{A} + Q = 0.$$

We take the Lyapunov function to be

$$V = e^T P e,$$

and then the derivative of the function V along the trajectory of the error equation (6.15) is expressed as

$$\begin{aligned} \dot{V} &= \dot{e}^T(t) P e(t) + e^T(t) P \dot{e}(t) \\ &= -e^T(t) Q e(t) + 2 e^T(t) P B [\Delta v(t) + \chi(t)] \\ &\leq -e^T(t) Q e(t) + 2 e^T P B \Delta v(t) + 2 \left\| B^T P e(t) \right\| \cdot \left\| \chi(t) \right\|. \\ &\leq -e^T(t) Q e(t) + 2 e^T P B [\Delta v(t) + \frac{B^T P e_p(t)}{\left\| B^T P e_p(t) \right\|} \chi_m] \end{aligned}$$

From (6.16), we have

$$\dot{V} \leq -e^T(t)Qe(t).$$

Because Q and P are symmetric positive definite matricies and the Lyapunov function is a positive definite quadratic form, therefore, the tracking error exponentially converges to zero.

Combining (6.4) and (6.13), the robust control algorithm for the uncertainty dynamic system is of the form

$$\tau = D^{-1}(\psi)[\ddot{q}_1^d - Ke + \Delta v] - s_0(\psi, q_2). \tag{6.17}$$

It is worth pointing out that the control algorithm (6.17) is a position tracking control algorithm. The heading direction of the mobile robot is determined by the reference path when the position tracking error becomes zero, as indicated in Chapter 2.

6.4 Numerical Simulation

Numerical simulation is used to illustrate the performance of the robust control algorithm (6.17). In the simulation, the full dynamic model (5.34)-(5.41) is used with the Dugoff's tire friction model to calculate the vehicle dynamics.

To illustrate the tracking control performance of the robust control algorithm, a non-robust control algorithm, which is produced by deleting the robust term Δv in (6.17), is also used in the simulations. The non-robust control algorithm is really the perfect dynamic system control algorithm through combining (6.4) and (6.11). A circle reference path with exact initial position is used in the simulation. The parameters in the feedback matrices of the robust control algorithm (6.17) and corresponding non-robust control algorithm are set to be $K_1 = diag[0.96 \quad 0.96]$ and $K_2 = diag[0.16 \quad 0.16]$.

Fig. 6.1 and Fig. 6.2 show the influence of the unmodeled dynamics on the tracking control performance of the non-robust and robust control algorithms, respectively. In the path and orientation graph, the dashed line indicates the reference path and the solid line indicates the simulated vehicle motion trajectory. In the longitudinal slip graph, the dashed and solid line correspond to the left and right wheel's slip, respectively. The longitudinal and lateral slip are calculated from the Equations (5.21)-(5.25). The oscillation in the longitudinal and lateral slip curves is due to the nonlinear and discontinuous features of Dugoff's friction model.

Because the non-robust algorithm is designed only on the basis of the perfect dynamic model, and the full dynamic model is used in the simulation, therefore, there exist unmodeled dynamics,

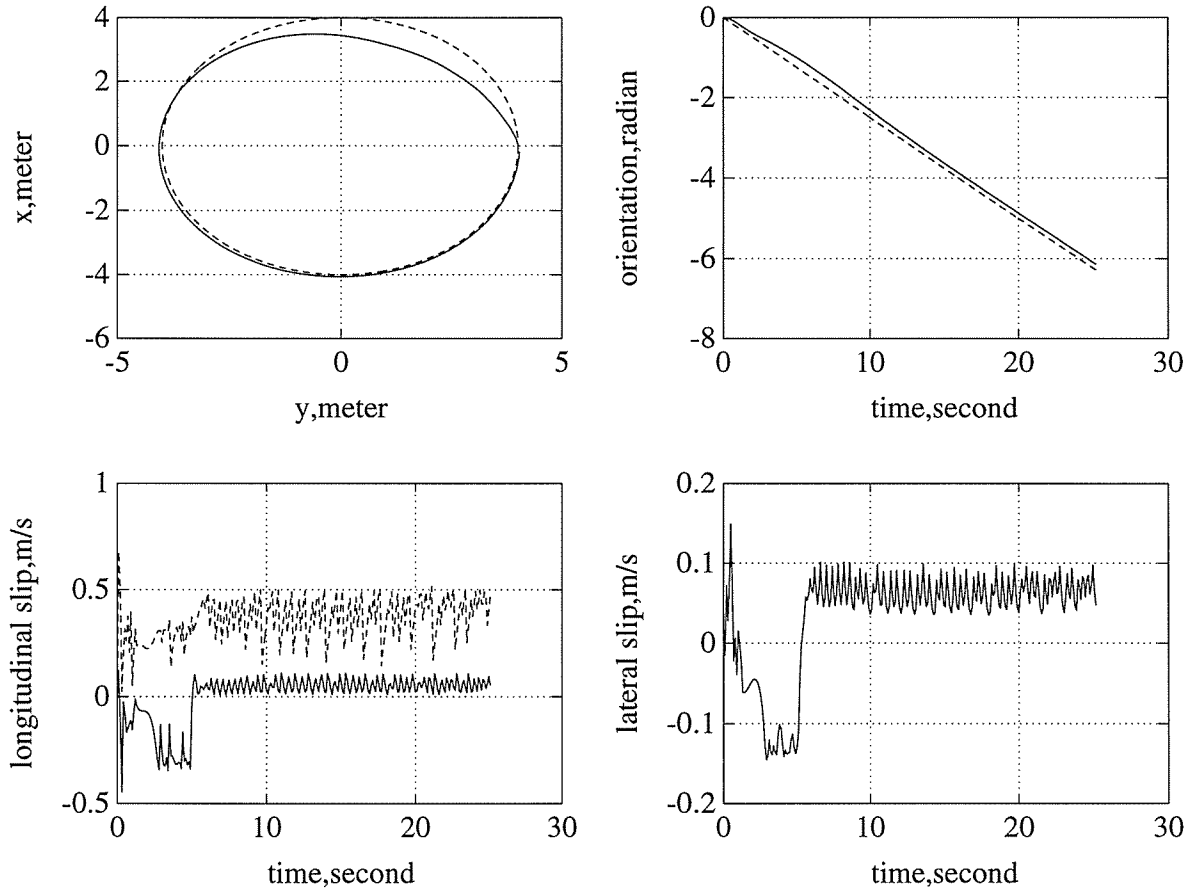


Fig. 6.1 Simulation Results of the Non-Robust Control Algorithm. The Vehicle Dynamics Is Calculated By the Full Dynamic Model and Dugoff's Tire Friction Model

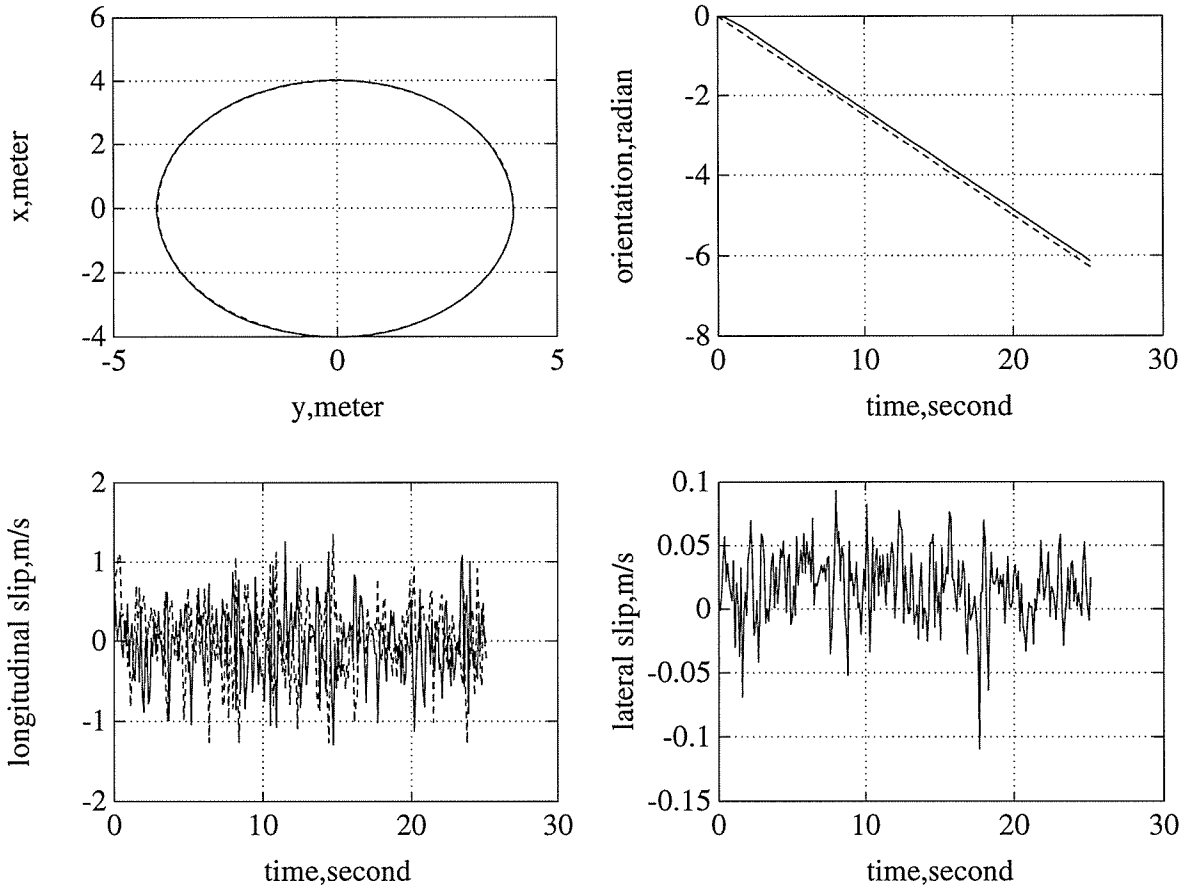
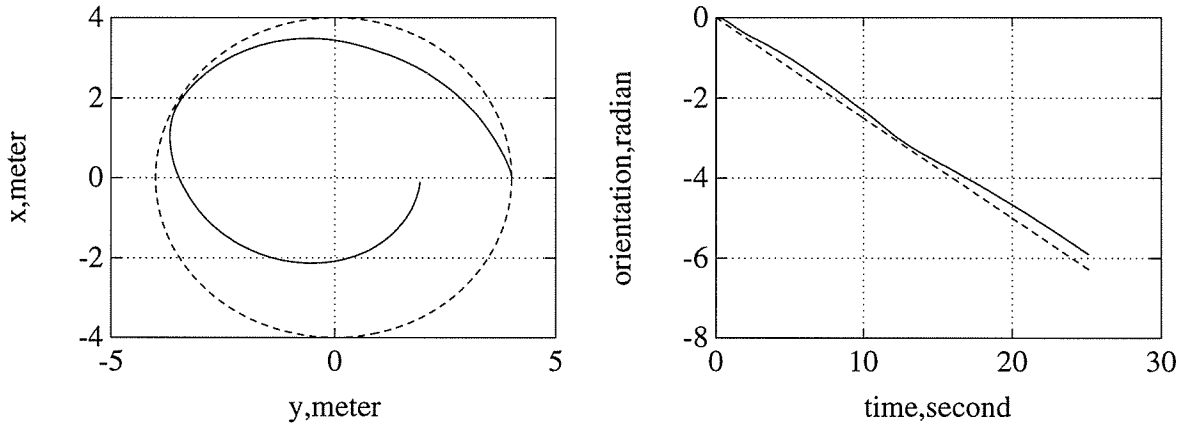


Fig. 6.2 Simulation Results of Robust Control Algorithm. The Vehicle Dynamics is Calculated by the Full Dynamic Model and Dugoff's Tire Friction Model

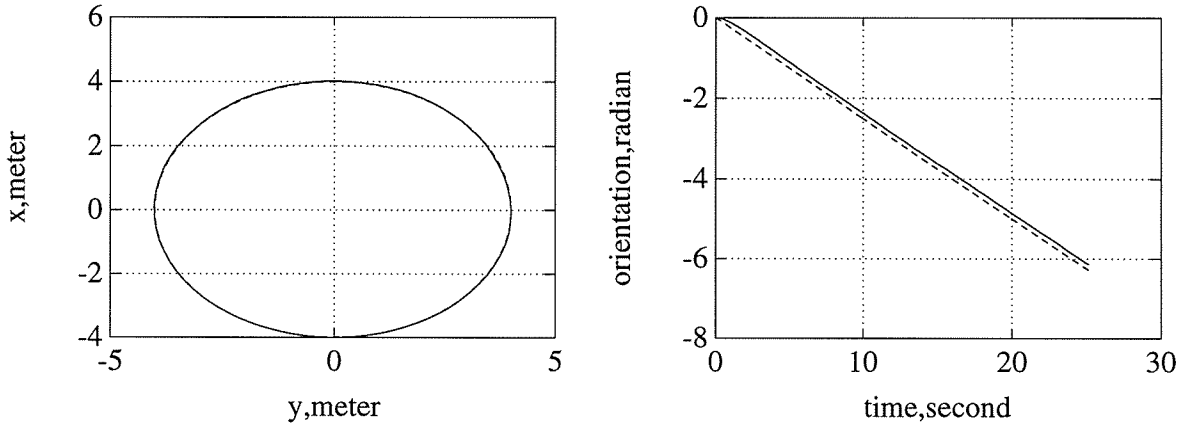
and the non-robust algorithm results in large tracking errors. It is also found that to further increase the feedback coefficients will degrade the tracking performance of the algorithm.

It can be seen from Fig. 6.2 that the robust control algorithm gives very good tracking results. The simulated vehicle trajectory overlaps the reference path in the graph. The oscillation in the slip curves of the robust algorithm is more serious than that on the non-robust algorithm. This is partially due to the stronger discontinuous features of Dugoff's tire model under stronger control action and partially due to the 'chattering' phenomenon of the robust control algorithm. It is very interesting to note that the average values of the longitudinal and the lateral slip of the robust control algorithm are smaller then that of the non-robust algorithm.

Fig. 6.3 provides simulation results of the tracking performance of the control algorithms under the condition of external disturbance. A tool-equipment action force, $\eta_{u,e}(t) = -200N$, is applied to the system. Fig. 6.3(a) is the simulation results of the non-robust algorithm, and Fig. 6.3(b) is the results of the robust control algorithm. It can be seen that the non-robust control algorithm cannot follow the reference path at all in this case, while the tracking trajectory of the robust control algorithm overlaps the reference path in the graph.



(a) Non-robust control algorithm



(b) Robust control algorithm

Fig. 6.3 Performance of the Robust Control Algorithm Under the Condition of External Disturbance. The Disturbance Is Set to Be a Tool-Equipment Acting Force $\eta_{u,e}(t) = -200N$ Which Is Applied for $t \geq 10$.

It is evident that the robust control algorithm gives very good tracking ability to both unmodeled dynamics and external disturbances.

The robustness of the control algorithm to external disturbance, such as the tool-equipment acting force, is very important for an industrial mobile robot. It is worth tnoting that the robustness of the robust control algorithm to the external disturbance is not unlimited. It is clear that, for example, if the magnitude of the tool force in the direction opposite the motion is greater than that of the maximum tire tractive force, then the vehicle will be completely uncontrollable physically; therefore, no control algorithm can work in this situation.

It is verified in the simulation that there exists a maximum external disturbance force beyond which the vehicle is uncontrollable. In practical application, a mobile robot must possess enough self weight and driven power to guarantee the robustness of the control algorithm.

6.5 Conclusion

Although a dynamic model based tracking control algorithm is vital for accurate tracking control of wheeled mobile robots, it is impossible to use a full dynamic model for the control system design because of the computational complexity. In this chapter, a systematic method is developed for the dynamic model based tracking control algorithm for differentially driven wheeled mobile robots. This method ensures the application of the nonlinear robust control system design method to the dynamic model based tracking control problem of the mobile robots. On the basis of a full dynamic model, a reduced order dynamic model is developed and the influence of uncertainty is analyzed. This leads to a tracking control algorithm which is exponentially convergent and robust to both unmodeled dynamics and external disturbances while maintaining calculation simplicity. Numerical simulation shows the tracking control ability with the presence of uncertainties.

CHAPTER 7

VARIABLE STRUCTURE DYNAMIC TRACKING CONTROL

7.1 Introduction

The main difficulty in the mobile robot's dynamic model based control algorithm development and application is the influence of uncertainty. This is because real-time operation limits the complexity of the dynamic model, while it is impossible to accurately describe the motion with low order and simple equations. As shown in the previous section, low order dynamic system equations include multiple uncertainty terms. Robustness is the most important property for the dynamic model based control algorithm. A variable structure control system has the feature of sliding mode invariance to both system perturbations and external disturbances (Utkin 1978, 1992), and therefore, it is a suitable design method for the dynamic model based tracking control of mobile robots. In this section, we first review the main aspects of variable structure control, and then produce a variable structure control algorithm for the mobile robot's tracking control problem.

7.2 Variable Structure System Method

The main principle of the Variable Structure System method is to design a control law to make the system react to the switching manifold

$$S(e) = Ce = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = 0. \tag{7.1}$$

When the system reachs the switching manifold, it will remain on the switching manifold. The system is then said to be in a "sliding mode". The tracking control performance is then determined by the sliding mode, that is

$$C_1 e_1 + C_2 \dot{e}_1 = 0. \tag{7.2}$$

If C_2 is nonsingular, then we have

$$\dot{e}_1 + C_2^{-1} C_1 e_1 = 0. \tag{7.3}$$

Therefore, the system position tracking dynamic response on the switching manifold is determined by (7.3).

One of the reaching conditions can be described by

$$S^T \dot{S} < 0. \tag{7.4}$$

It can be seen that the switching condition (7.4) will force the system to reach the switching manifold (7.1).

One of the Variable Structure Control types usually takes the form

$$\upsilon = \upsilon_e + \Delta\upsilon \tag{7.5}$$

where, υ_e is the equivalent sliding model control, and $\Delta\upsilon$ is an added term to satisfy the reaching conditions.

If the sliding mode of the Variable Structure Control system is not affected by system perturbations and external disturbances, it is said that the sliding mode is invariant to perturbations and disturbances.

It is well known that if the matching conditions (5.100)-(5.102) are satisfied, that is, all the system perturbations and external disturbances enter the system through the input channel, then, the sliding mode of the variable structure control system is invariant.

In Theorem 5.6.1 we have proven that all the uncertainties including the unmodeled dynamics, parameters perturbations, and external disturbances, in the uncertainty dynamic model of the mobile robot with two differently driven wheels, enter the system through the inputs channel; that is, the matching condition is hold. Therefore, the sliding mode of the system is invariant to all of these uncertainties.

7.3 Variable Structure Tracking Control

We first rewrite the perfect dynamic model (6.5) and the uncertainty dynamic model (6.8) as

$$\dot{e}(t) = Ae(t) + B\upsilon(t) \tag{6.5}$$

where

$$A = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}$$

and

$$B = \begin{bmatrix} 0 \\ I \end{bmatrix}.$$

Additionally, we have

$$\dot{e}(t) = Ae(t) + B[\upsilon(t) + \chi(\psi, q_2, \delta_m)] \tag{6.8}$$

where

$$\chi(\psi, q_2, \delta_m) = D(\psi)[\Delta s(\psi, q_2, \delta_m) + \varsigma(\psi, q_2, \delta_m)]$$

is the uncertainty term, and we suppose that the uncertainty term is bounded as follows

$$\left\|\chi(\psi, q_2, \delta_m)\right\| = \left\|D(\psi)[\Delta s(\psi, q_2, \delta_m) + \varsigma(\psi, q_2, \delta_m)]\right\| \le \chi_m$$

where, $0 \le \chi_m < \infty$ is a given number.

Theorem 7.4.1: Suppose that the switching manifold (7.1) is determined and C_2 is nonsingular. Then the sliding modes of the perfect dynamic system (6.5) and the uncertainty dynamic system (6.8) are equivalent.

Proof: Taking the time derivative of (7.1) along the trajectory of the perfect dynamic model (6.5), we have

$$\dot{S} = CAe + CB\upsilon = 0 \tag{7.6}$$

which results in

$$C_1 e_2 + C_2 \upsilon = 0. \tag{7.7}$$

Since C_2 is nonsingular, we can solve the equivalent control from (7.7) as follows:

$$\upsilon_{eq} = -C_2^{-1} C_1 e_2. \tag{7.8}$$

Substituting (7.7) into (6.5), the sliding mode of the perfect dynamic system (6.5) is then described as

$$\dot{e}(t) = \begin{bmatrix} 0 & I \\ 0 & -C_2^{-1} C_1 \end{bmatrix} e(t). \tag{7.9}$$

Similarly, taking the time derivative of (7.1) along the trajectory of the uncertainty dynamic model (6.8), we get the equivalent control as

$$\upsilon_{eq} = -C_2^{-1} C_1 e_2 - \chi. \tag{7.10}$$

Substituting (7.10) into (6.8), we can see that the sliding mode of the uncertainty dynamic system (6.8) is also described by equations (7.9). Therefore, the sliding modes of the two systems are equivalent.

Theorem 7.4.1 means that the sliding mode of the mobile robot's dynamic system equations is invariant to unmodeled system dynamics and external disturbances. Since the system dynamic response is determined by the sliding mode on the switching manifold, therefore, the uncertainty dynamic system (6.8) and the perfect dynamic system (6.5) have the same tracking performance on the same switching manifold.

We set matrix C in the switching manifold (7.1) to be

$$C = [C_1 \quad C_2] = [\Lambda \quad I] \tag{7.11}$$

where $\Lambda = diag[\lambda_1 \quad \lambda_2], (\lambda_1 > 0, \lambda_2 > 0)$. Then the sliding mode for both the perfect dynamic system and the uncertainty dynamic system is expressed as

$$\dot{e}_1(t) = -\Lambda e_1(t). \tag{7.12}$$

This means that the tracking error exponentially converges to zero. Next, we need to determine the control law for the uncertainty system (6.8) to guarantee the reaching condition (7.4). Taking the time derivative of the reaching condition (7.4) along the trajectory of (7.1), we have

$$\begin{aligned} S^T \dot{S} &= S^T CAe + S^T CB(\upsilon + \chi) \\ &= S^T CAe + S^T(\upsilon + \chi) \\ &= S^T(CAe + \upsilon + \chi) \end{aligned} \quad . \tag{7.13}$$

Now we set

$$\begin{aligned} \upsilon &= -CAe - \Delta\upsilon \\ &= -\Lambda e_2 - \Delta\upsilon \end{aligned} \tag{7.14}$$

and

$$\Delta\upsilon = \begin{cases} \dfrac{S}{\|S\|}\chi_m & (\|S\| \neq 0) \\ 0 & (\|S\| = 0) \end{cases} \tag{7.15}$$

Substituting (7.14) and (7.15) into (7.13), we see that, when $\|S\| \neq 0$ then

$$\begin{aligned} S^T\dot{S} &= S^T(-\Delta\upsilon + \chi) \\ &= -\|S\|\chi_m + S^T\chi \\ &\leq -\|S\|\chi_m + \|S\|\|\chi\|. \\ &= -\|S\|(\chi_m - \|\chi\|) \\ &< 0 \end{aligned}$$

That is, the reaching condition (7.4) is satisfied. Because the tracking error is controlled by (7.12) on the switching surface, then the tracking control exponentially converges to zero when the sliding mode is reached. Therefore, the following theorem applies.

Theorem 7.4.2: The control law (7.14) with $\Delta\upsilon$ determined by (7.15) is a variable structure control of the uncertainty dynamic system (6.8) for the switching manifold (7.11). Therefore, the tracking error of the uncertainty dynamic system (6.8) exponentially converges to zero under the control (7.14).

Now, set

$$C_1 = I, \text{ and } C_2 = \Lambda \ .$$

The sliding mode equation (7.9) becomes

$$\dot{e}_1 = e_2 \tag{7.16}$$

$$\dot{e}_2 = -\Lambda e_2 \tag{7.17}$$

whereas on the switch manifold (7.11) we have

$$e_2 = -\Lambda e_1$$

which leads to the following expression:

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \end{bmatrix} = \begin{bmatrix} -\Lambda & 0 \\ 0 & -\Lambda \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}. \tag{7.18}$$

Therefore, in the sliding mode the whole system is exponentially convergent.

One important feature of variable structure control is the chattering phenomena, which means that there exists a high frequency switching process in the control inputs. One common method to overcome the chattering is to modify (7.15) into

$$\Delta v = \begin{cases} \dfrac{S}{\|S\|}\chi_m & (\|S\| > \varepsilon) \\ \dfrac{S}{\varepsilon}\chi_m & (\|S\| \le 0) \end{cases} \tag{7.19}$$

where, $0 < \varepsilon \le 1$. We will call (7.19) a modified variable structure control.

Since in the TMR motor control system the input voltage is applied to the D. C. motors via a Pulse Width Modulation (PWM) circuit, chattering in the control inputs has a very weak direct influence.

7.4 Numerical Simulation

The tracking control performance of the variable structure dynamic tracking control algorithm is simulated based on the dynamic model described in Chapter 5. In the simulation, the dynamics of the wheeled mobile robot is calculated by the full dynamic model described by (5.34)-(5.41) with Dugoff's tire model. Therefore, unmodeled uncertainties are included in all simulatiions. We refer to the control (7.14) as a variable structure control if the nonlinear term Δv is determined by (7.15), or a *modified variable structure control* if the nonlinear term Δv is determined by (7.19). The C matrix in the switching manifold is determined by the structure of (7.11) with

$$\Lambda = \begin{bmatrix} 0.4 & 0 \\ 0 & 0.4 \end{bmatrix}.$$

The limit in the modified nonlinear term (7.19) is selected to be $\varepsilon = 0.1$ in the simulation. The dynamic model is integrated by means of a fourth order Runge-Kutta method. The simulation code is written with in Matlab on a Macintosh Q80 computer.

Fig. 7.1 is the simulation results of the variable structure control algorithms with a straight line reference path. There is initial position error, while the initial orientation is exact.

Fig. 7.2 is the simulation results of the modified variable structure control algorithm with the same reference path and initial condition as in Fig. 7.1. It is worth noting that the oscillations in the control input curves in Fig. 7.1 and Fig. 7.2 are not merely caused by the chattering of the

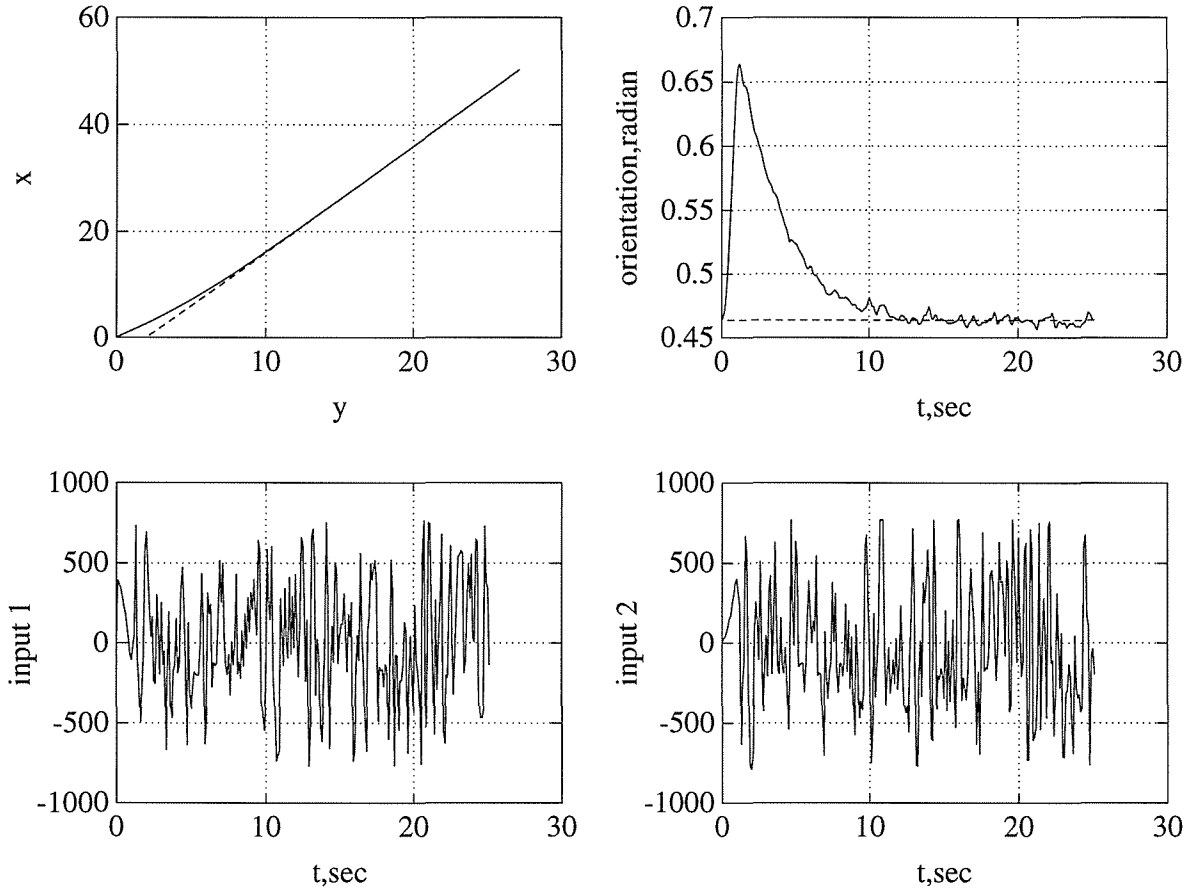


Fig. 7.1 Variable Structure Control with Straight Line Reference Path

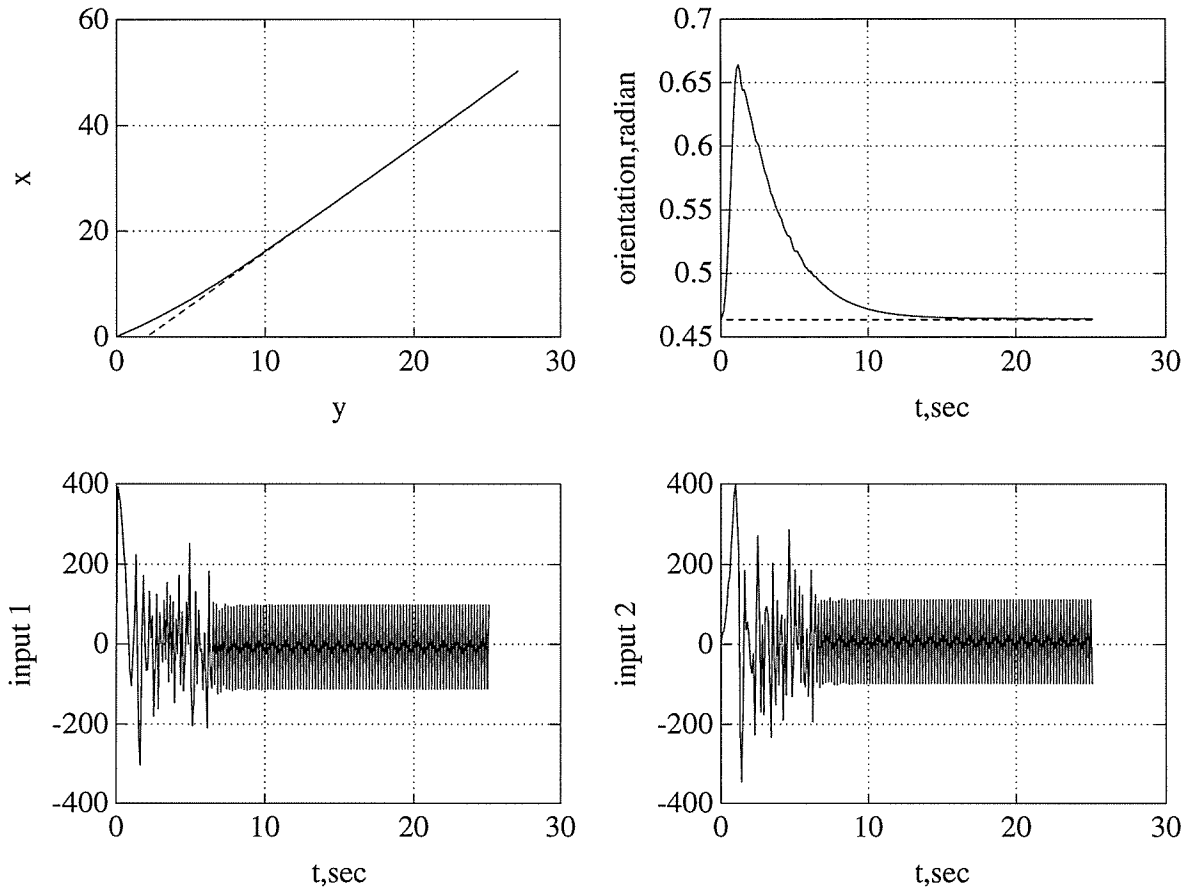


Fig. 7.2 Modified Variable Structure Control with Straight Line Reference Path

variable structure control. Because Dugoff's tire model in the full dynamic model is highly sensitivity to wheel slippage, the oscillation in the control inputs is necessary to overcome the wheel slippage. For the wheel slippage calculated by the full dynamic model one can reference the simulation results in Chapter 6.

It is clear that the chattering influence in the control inputs is greatly reduced by the modified variable structure control algorithm, and thus it will be used in all of the following simulation.

The robustness of the variable structure control to unmodeled system dynamics is proved by the simulation results in Fig. 7.1, Fig. 7.2, and all the simulation results in following part of this section. This is because the systems dynamics are calculated by the full dynamic model and it includes all the unmodeled uncertainties.

Fig. 7.3 is the simulation results of the modified variable structure control algorithm with a sinusoidal reference path and exact initial conditions.

Fig. 7.4 shows the tracking control performance of the modified variable structure control algorithm with a circular reference path. In addition to the unmodeled dynamics, an external disturbance tool force is applied to the system at t≥10.

Fig. 7.5 shows the tracking control performance of the modified variable structure control algorithm with a sinusoidal slope reference path and non zero initial orientation error. The uncertainty influence includes the unmodeled system dynamics and the external tool force.

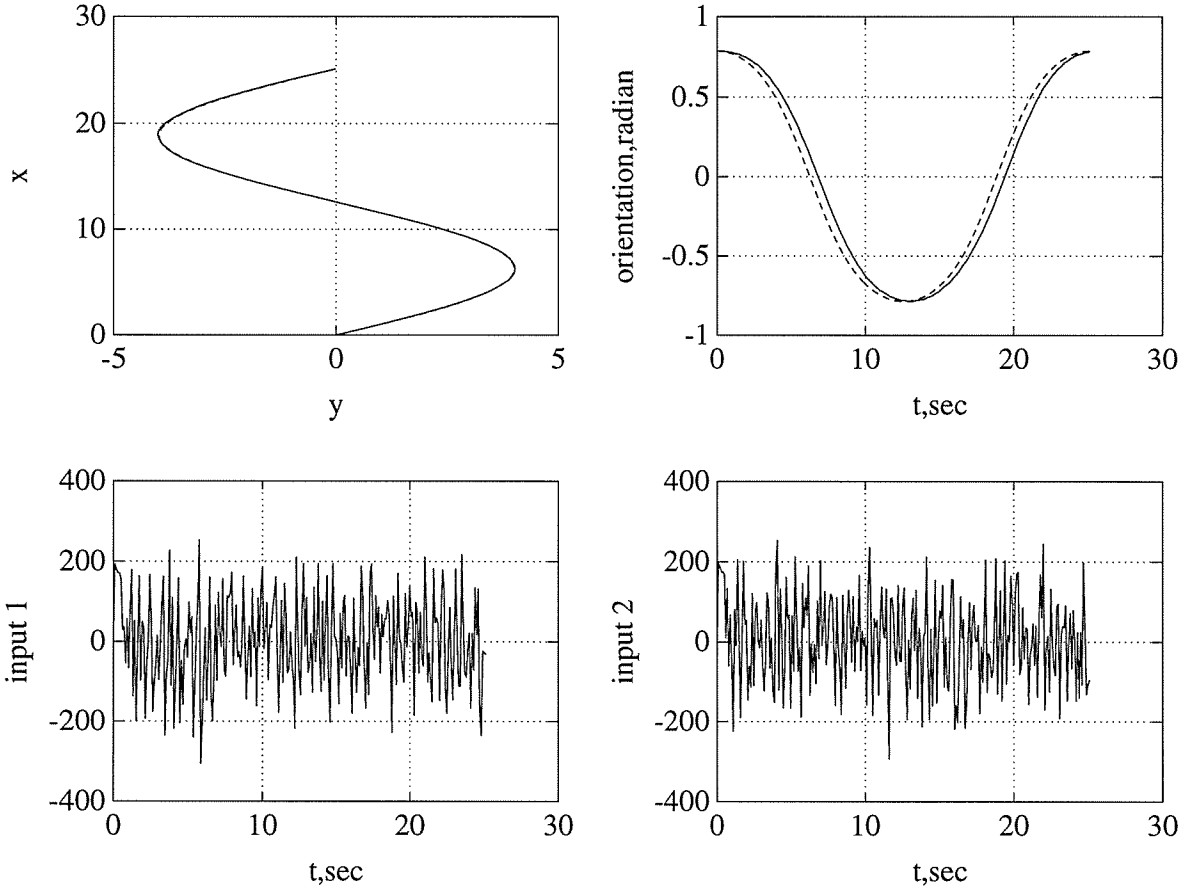


Fig. 7.3 Modified Variable Structure Control with Sinusoidal Reference Path

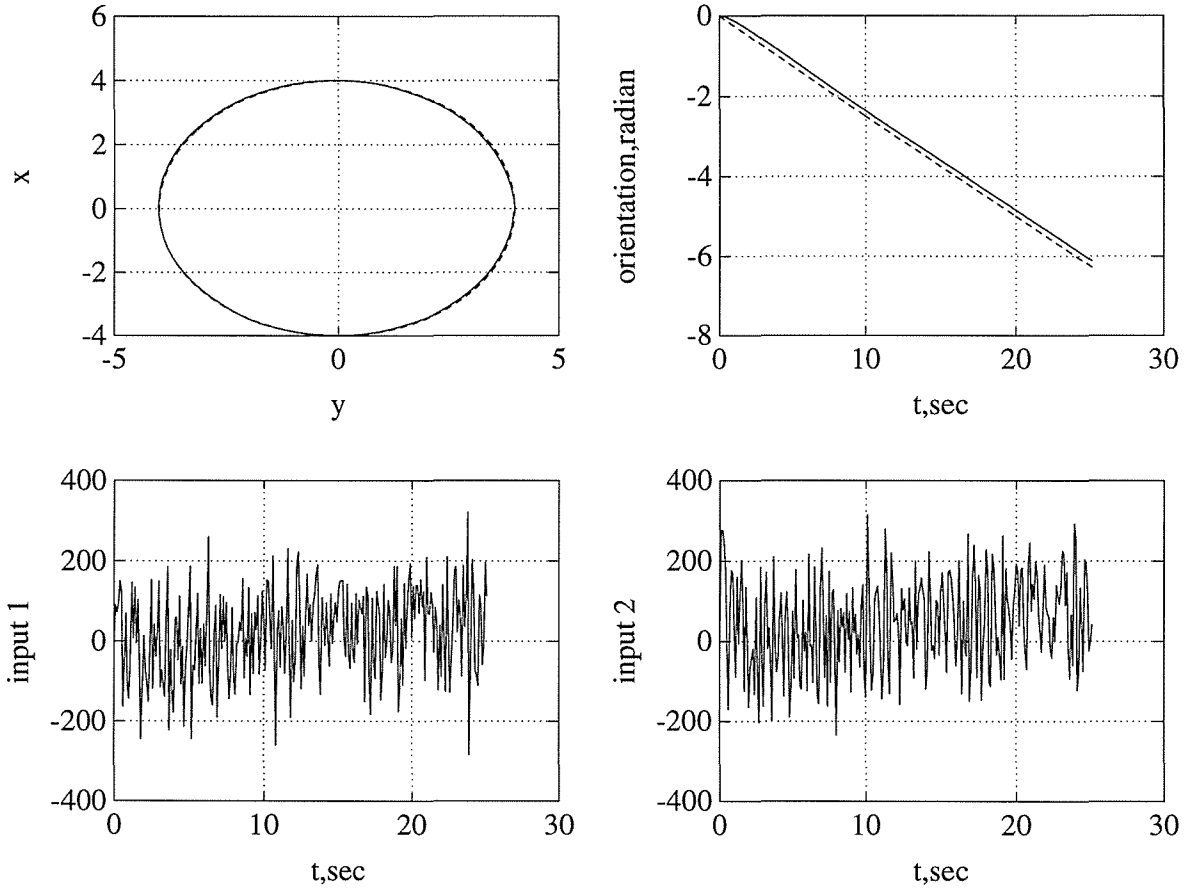


Fig. 7.4 Modified Variable Structure Control with Circular Reference Path and External Disturbance Tool Force $\eta_{u,e}(t) = -200N$ Applied for $t \geq 10$.

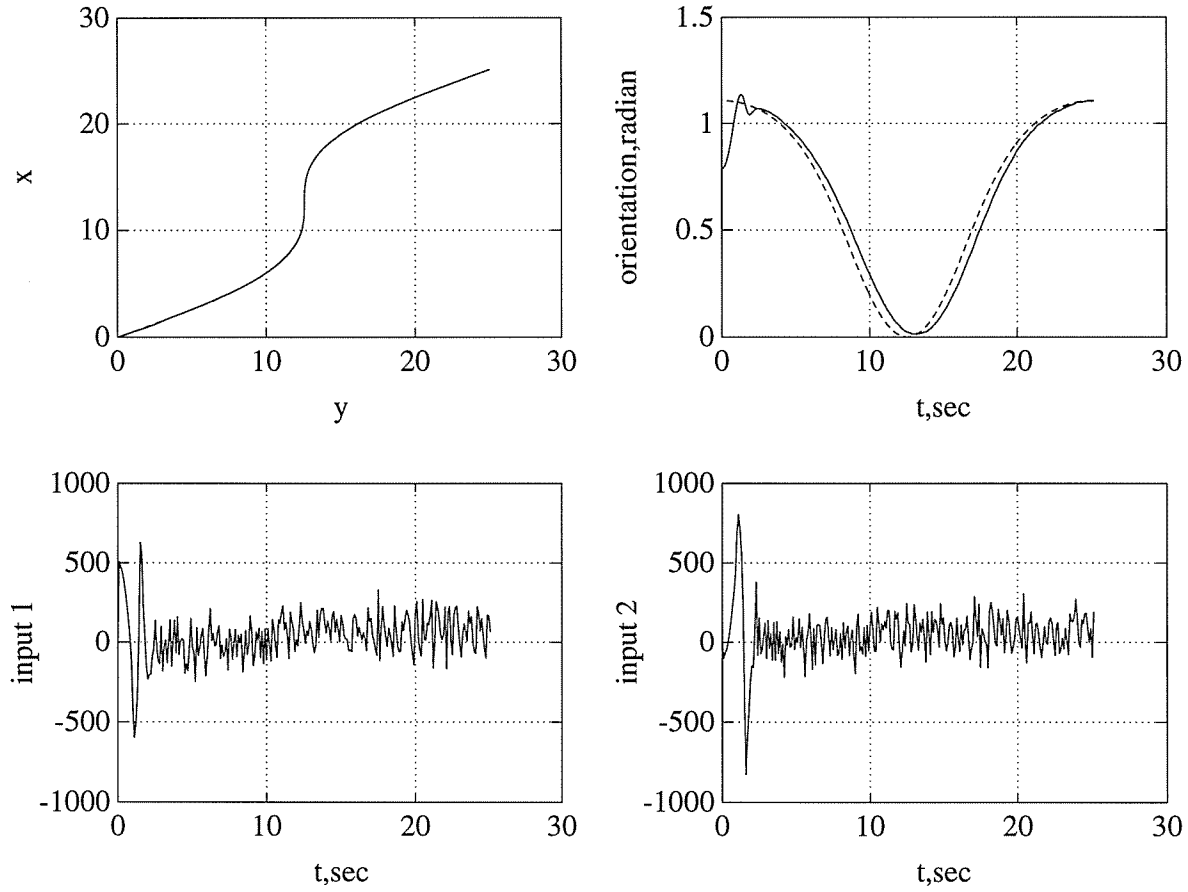


Fig. 7.5 Modified Variable Structure Control with Sinusoidal Slope Reference Path and External Disturbance Tool Force $\eta_{u,e}(t) = -200N$ Applied for $t \geq 10$.

In Fig. 7.6, the reference path and the initial conditions are the same as in Fig. 5 with the exception that a 20% error perturbation of system parameters is included in addition to the uncertainty influence of the unmodeled dynamics and external disturbances. The variable structure dynamic tracking control algorithm has high tracking control ability and possesses good robustness to all the system uncertainties.

7.5 Conclusion

On the basis of the variable structure control method, a variable structure dynamic tracking control algorithm for mobile robots with two independent driven wheels is developed. The

variable structure tracking control algorithm forces the dynamic system to reach a switching surface. The response of the dynamic system on the switching surface is called a sliding mode. In the sliding mode, the tracking error of the dynamic system exponentially converges to zero. Since we have proved in Chapter 5 that the uncertainties in the mobile robot dynamics all enter the system through the input channels (that is, the matching condition is hold for the mobile robot dynamics), then the sliding mode of the variable tracking control algorithm for the mobile robots is invariant to all the uncertainties. Simulation results show that the variable structure dynamic tracking control algorithm has high tracking control ability and possesses good robustness to all the system uncertainties.

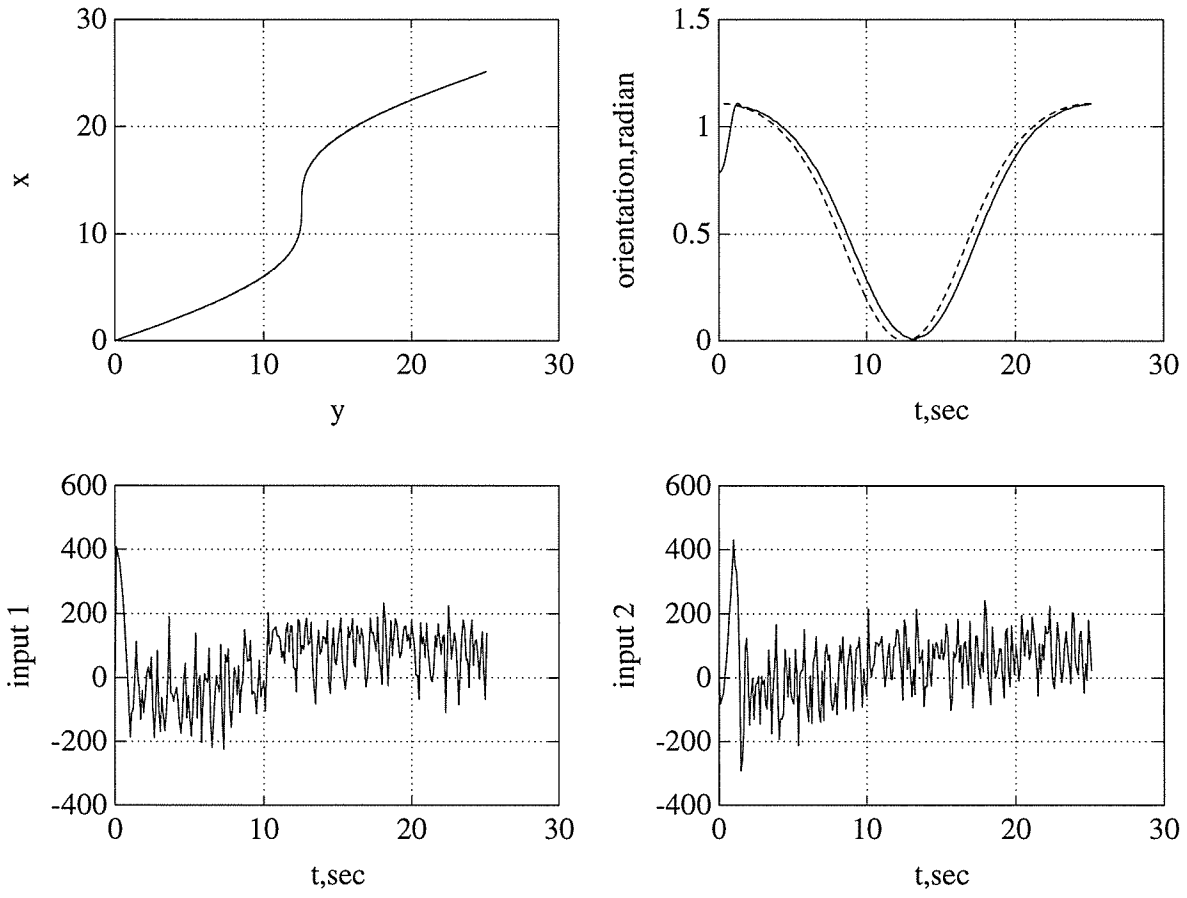


Fig. 7.6 Modified Variable Structure Control with Sinusoidal Slope Reference Path and External Disturbance Tool Force $\eta_{u,e}(t) = -200N$ Applied for $t \geq 10$ and Perturbation of 20% of System Parameter Errors

CHAPTER 8

PATH PLANNING IN HIGHWAY MAINTENANCE

8.1 Introduction

In the previous chapters, we have developed the kinematic model, dynamic model, and tracking control algorithms for mobile robots with two independent driven wheels. The purpose of our research is for tracking control of the Tethered Mobile Robot in highway maintenance operations. In this chapter, we will discuss the application of the developed modeling and control algorithms to highway maintenance.

8.2 Reference Trajectory and Reference Path

The tracking control algorithms developed in previous chapters are based on the trajectory tracking control problem, and we do not distinguish between reference path and reference trajectory. In the highway maintenance application, however, the mobile robot is typically required to accurately follow a reference path not a reference trajectory. For example, the reference path could be a crack in the pavement, which is independent of time. To use the tracking control algorithms described earlier to path following control, we must establish the relation between the reference path and the reference trajectory.

The reference path is considered to be a geographical curve on a planar surface. However, for a mobile robot to follow a reference path, the robot must move along the reference path with a certain speed, and in this way the reference path is related to time. Therefore, we consider the reference path to be a reference trajectory when the motion speed of the vehicle is determined. Thus, we do not strictly distinguish between the reference path and reference trajectory.

Due to the restriction that the mobile robot is on a planar surface, the reference path is a curve in the two dimensional plane represented as

$$\Im = \{ f(x, y) = 0 \mid f: \Re^2 \rightarrow \Re^1 \}. \tag{8.1}$$

If we use the curve length s , which is measured from the starting point on the curve to a point (x,y) on the curve, to be the parameter shown in Fig. 8.1, a reference path can be written as

$$\begin{cases} x = x(s) \\ y = y(s) \end{cases} \quad (s \geq 0). \tag{8.2}$$

The reference orientation is defined to be the angle between the x direction and the tangential direction of the reference path; that is

$$\psi(s) = a\tan\frac{y_s(s)}{x_s(s)} \tag{8.3}$$

where

$$x_s(s) = \frac{dx(s)}{ds}$$

and

$$y_s(s) = \frac{dy(s)}{ds}.$$

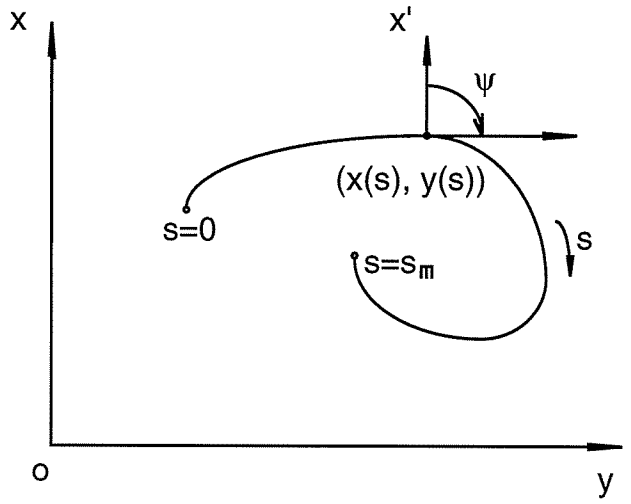


Fig. 8.1 Parameterized Reference Path

Since the parameter s is the along path distance, s is a function of time once the speed of the vehicle is determined; that is

$$s = s(t) \tag{8.4}$$

and

$$\dot{s}(t) = \frac{ds}{dt}$$

is the along path motion speed of the vehicle. Substituting (8.4) into (8.2) we have

$$\begin{cases} x(t) = x(s(t)) \\ y(t) = y(s(t)) \end{cases}. \tag{8.5}$$

Therefore, (8.5) becomes a reference trajectory. The reference orientation trajectory is thus expressed as

$$\psi(t) = a\tan\frac{y_s(s(t))}{x_s(s(t))}. \tag{8.6}$$

For the reference trajectory (8.5), we have the following time derivatives:

$$\dot{x}(t) = \frac{dx(t)}{dt} = \frac{dx(s)}{ds}\frac{ds}{dt} = x_s\dot{s} \tag{8.7}$$

$$\dot{y}(t) = \frac{dy(t)}{dt} = \frac{dy(s)}{ds}\frac{ds}{dt} = y_s\dot{s} \tag{8.8}$$

and

$$\ddot{x}(t) = \frac{dx_s}{dt}\dot{s} + x_s\ddot{s} = x_{ss}\dot{s}^2 + x_s\ddot{s} \tag{8.9}$$

$$\ddot{y}(t) = \frac{dy_s}{dt}\dot{s} + y_s\ddot{s} = y_{ss}\dot{s}^2 + y_s\ddot{s}\ . \tag{8.10}$$

The time derivatives for the reference orientation trajectory (8.6) that result are

$$\dot{\psi}(t) = \psi_s \dot{s} = \frac{x_s y_{ss} - y_s x_{ss}}{x_s^2 + y_s^2} \dot{s} \tag{8.11}$$

and

$$\ddot{\psi}(t) = \psi_{ss} \dot{s}^2 + \psi_s \ddot{s}. \tag{8.12}$$

From (8.5)-(8.12), we can see that when the reference path (8.2) and the motion speed of the vehicle are determined, the reference trajectory and the time derivatives are also determined.

In the highway crack sealing task, a vision sensor system is used to determine the crack location on the road surface. The sensed crack data is then filtered to reduce the noise and the reference path is then produced. The reference path with a determined vehicle speed will define the reference trajectory. Therefore, the crack following or reference path following problem can be solved by the previous described tracking control algorithms.

8.3 Speed Manipulation

The motion speed of the mobile robot is determined by the task's requirements. For the highway crack sealing application, a typical motion speed is 0.8939 m/s (2 mph). In application, it is often required to manipulate the motion speed of the mobile robot to match given initial and terminal conditions. Here, we use a quintic polynomial to describe the along path length function *s(t);* that is

$$s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5. \tag{8.13}$$

We set the initial conditions and terminal conditions to be

$$t = 0,\ s(0) = 0,\ \dot{s}(0) = 0,\ \ddot{s}(0) = 0 \tag{8.14}$$

and

$$t = T,\ s(T) = s_m,\ \dot{s}(T) = 0,\ \ddot{s}(T) = 0 \tag{8.15}$$

where, T is the terminal time, and s_m is the terminal along path length of the reference path.

The initial and terminal conditions described by (8.14) and (8.15) mean that the mobile robot originally stops at the start point, it then moves from the start point to the terminal point, and then stops at the terminal point.

Using the initial and terminal conditions of (8.14) and (8.15), the coefficients in the quintic polynomial (8.13) can be determined, and (8.13) becomes

$$s(t) = s_m[10(t/T)^3 - 15(t/T)^4 + 6(t/T)^5]. \tag{8.16}$$

The along path motion speed and acceleration of the mobile robot can be determined by the time derivatives of the function $s(t)$. From (8.16), we have

$$\dot{s}(t) = (s_m/T)[30(t/T)^2 - 60(t/T)^3 + 30(t/T)^4] \tag{8.17}$$

and

$$\ddot{s}(t) = (s_m/T^2)[60(t/T) - 180(t/T)^2 + 120(t/T)^3]. \tag{8.18}$$

The s(t) function and its time derivatives are as shown in Fig. 8.2.

Taking the time derivative of (8.18), we have

$$\dddot{s}(t) = (s_m/T^3)[60 - 360(t/T) + 360(t/T)^2]. \tag{8.19}$$

Now, setting

$$\dddot{s}(t) = 0$$

leads to

$$(t/T)_{a,m} = \frac{3 \pm \sqrt{3}}{6}. \tag{8.20}$$

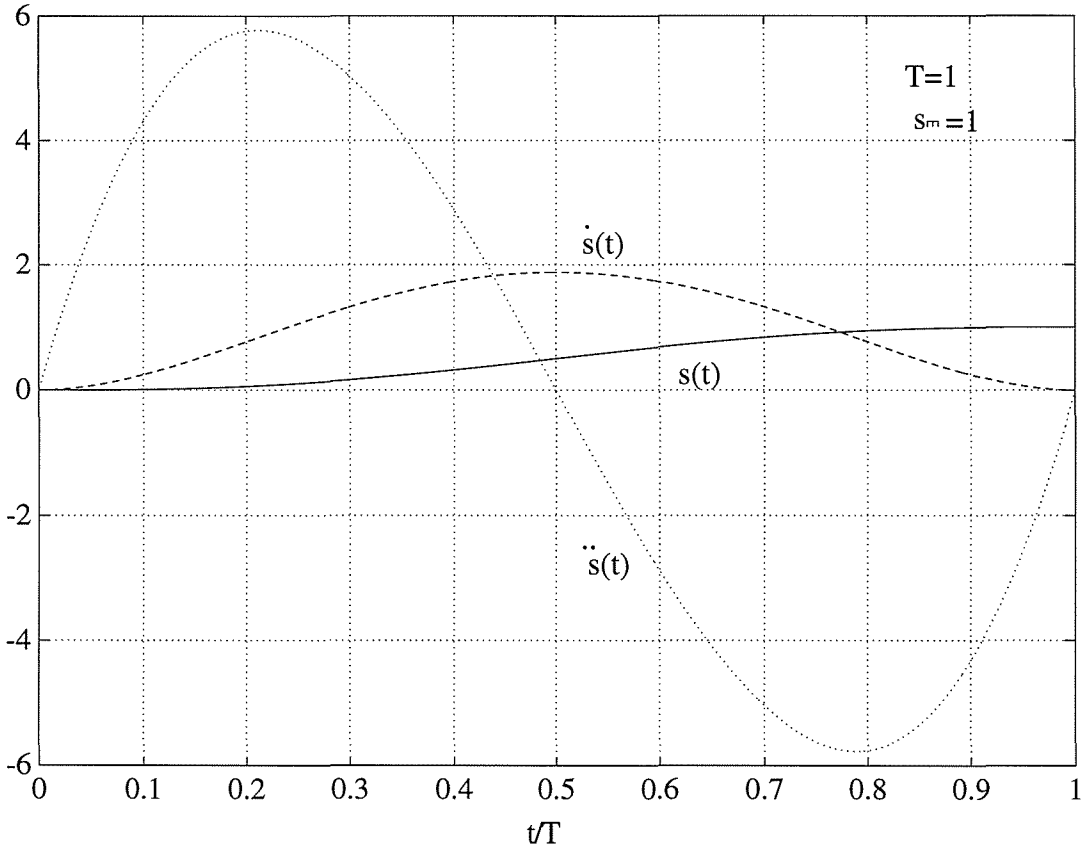


Fig. 8.2 Quintic Polynomial And Its Derivatives

Substituting (8.20) into (8.18), we get the maximum and minimum accelerations as

$$\ddot{s}_{max} = \ddot{s}(\frac{3-\sqrt{3}}{6}T) = \frac{10\sqrt{3}}{3}\frac{s_m}{T^2} \approx 5.7735\frac{s_m}{T^2} \tag{8.21}$$

and

$$\ddot{s}_{min} = \ddot{s}(\frac{3+\sqrt{3}}{6}T) = -\frac{10\sqrt{3}}{3}\frac{s_m}{T^2} \approx -5.7735\frac{s_m}{T^2} \ . \tag{8.22}$$

Setting

$$\dddot{s}(t) = 0$$

we can determine that

$$(t/T)_{v,m} = \frac{1}{2} \ . \tag{8.23}$$

Now substituting (8.23) into (8.17), we get the maximum speed as

$$\dot{s}_{max} = \dot{s}(\frac{1}{2}T) = \frac{15}{8}\frac{s_m}{T} = 1.875\frac{s_m}{T}. \tag{8.24}$$

The average acceleration is then

$$\overline{a} = \frac{s_m}{T^2}$$

and the average speed is expressed as

$$\overline{v} = \frac{s_m}{T} \ .$$

It can be seen that (8.21) and (8.22) give the maximum and minimum acceleration with respect to the average acceleration. The relation between the maximum speed and the average speed is described by (8.24).

In tracking control, the motion speed of the mobile robot must be determined according to the task allowable maximum speed and maximum acceleration. The maximum acceleration is also limited by the drive system.

8.4 Basic Path Planning

There are two aspects of path planning in a highway maintenance task. One is the task determined path planning, such as a pavement crack; another is the initial locating path planning. For crack sealing, a vision sensing system is used to sense the crack and the reference path is then determined by the crack. When the task determined path is given, the mobile robot must move from its original location to the start point of the operation path and a locating reference path is then required.

8.4.1 *Task Operation Path Planning*

Highway crack sealing task path planning was studied by Lasky and Ravani (1993). A vision sensing system acquires gray scale images of the road using a line scan camera. Each line scan is 1.58 mm ($\frac{1}{16}$ in) deep and 3.6576 m (12 ft) wide (approximately one lane width). These scans are buffered up to build up an image consisting of 64.5 mm (2 in) tiles, each of which consists of a 32x32 grid of 1.58 mm ($\frac{1}{16}$ in) square pixels. The vision software then builds a histogram of each tile, and then computes a statistical moment of each histogram. Then the software compares the statistical moments within independent 5x5 tile areas. The software uses eight comparison patterns to detect both the presence and preferred direction of a crack. The raw vision data is insufficient, as it is just an array of potential crack locations, without any sense of a relationship between pixels, or tiles. A path planning algorithm processes this data by filtering out noise, filling in blank but potentially cracked areas, forming connected, ordered sets, and extracting the appropriate portions of the planned paths for the current workspace.

Since the planned path is an ordered set of discrete points on the planar surface, that is

$$\aleph = \{(x_k, y_k) | k = 1, 2, ..., N\}, \tag{8.25}$$

we can use an interpolating polynomial to get the parameterized reference path as

$$x(s) = p_x(s|\aleph) \tag{8.26}$$

and

$$y(s) = p_y(s|\aleph) \tag{8.27}$$

where, $p_x(\cdot)$ *and* $p_y(\cdot)$ are interpolating polynomials on the basis of the set of ordered points $\aleph$.

8.4.2 *Initial Locating Path Planning*

There are many ways to plan the initial locating path, and we use only straight line and circular curve segments for this purpose. The path starts from the point (x_1, y_1) and the mobile robot is initially located at the tracking point (x_0, y_0) with initial orientation ψ_0 as shown in Fig. 8.3.

We use a three-step method for the initial locating path planning. Let us draw a line through point (x_1, y_1) in the tangential direction of the reference path. The angle between this tangential line and the x axis is ψ_1. The point (x'_1, y'_1) is at the tangential line with a distance b to the point (x_1, y_1), where, b is the distance from the tracking point to the center of the baseline on the mobile robot. The center of the baseline of the mobile robot is initially located at point (x'_0, y'_0). Then, we draw a line through the two points $(x'_0, y'_0)$and (x'_1, y'_1). The angle between the x axis and the line through points $(x'_0, y'_0)$and (x'_1, y'_1) is ψ_{01}. We then denote

$$\Delta\psi_0 = \psi_{01} - \psi_0 \tag{8.28}$$

and

$$\Delta\psi_1 = \psi_{01} - \psi_1. \tag{8.29}$$

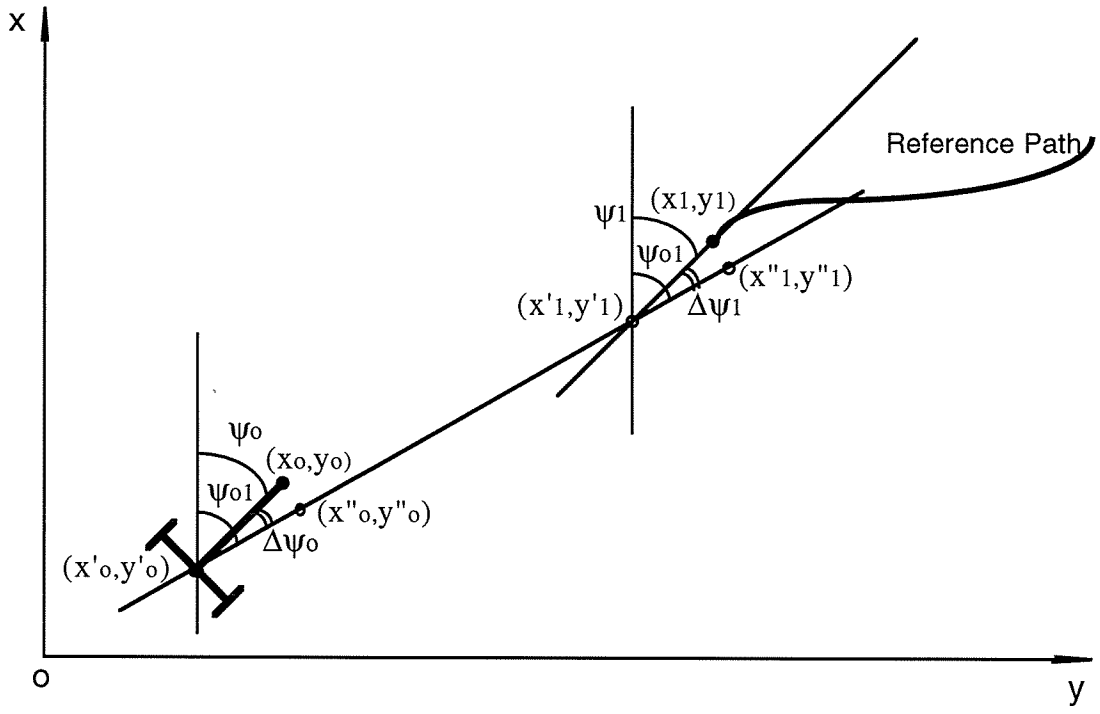


Fig. 8.3 Initial Locating Path Planning

The three-step reference path can then be planned as follows.

Step 1: Pure rotation with angular displacement $\Delta\psi_0$. The tracking point moves from point (x_0, y_0) to point (x''_0, y''_0). Here, the point (x''_0, y''_0) is on the line linking the points (x'_0, y'_0) and (x'_1, y'_1) with a distance b to the point (x'_0, y'_0);

Step 2: Pure straight line motion along the line through the points (x'_0, y'_0) and (x'_1, y'_1) with the tracking point moving from (x''_0, y''_0) to (x''_1, y''_1). Here, the point (x''_1, y''_1) is on the line through the points (x'_0, y'_0) and (x'_1, y'_1) with a distance b to the point (x'_1, y'_1). It is clear that the center of the baseline of the mobile robot moves form the point (x'_0, y'_0) to the point (x'_1, y'_1);

Step 3: Pure rotation with angular displacement $\Delta\psi_1$. The tracking point moves from point (x''_1, y''_1) to point (x_1, y_1). The point (x_1, y_1) is the starting point of the reference path, and the initial locating process is, therefore, completed.

Since (x_0, y_0), (x_1, y_1), ψ_0 are known, the three-step initial locating reference path can be mathematically described as follows.

We first determine the coordinates of related points and the value of the related angle. The original initial location of the center point of the baseline is

$$\begin{cases} x'_o = x_0 - b\cos\psi_0 \\ y'_o = y_0 - b\sin\psi_0 \end{cases}. \tag{8.30}$$

The coordinates of the point (x'_1, y'_1) is

$$\begin{cases} x'_1 = x_1 - b\cos\psi_1 \\ y'_1 = y_1 - b\sin\psi_1 \end{cases} \tag{8.31}$$

where

$$\psi_1 = a\tan\frac{y_s(0)}{x_s(0)}. \tag{8.32}$$

The angle ψ_{01} can be calculated as

$$\psi_{01} = a\tan\frac{y'_1 - y'_0}{x'_1 - x'_0}. \tag{8.33}$$

The coordinates of points (x''_0, y''_0) and (x''_1, y''_1) are

$$\begin{cases} x''_0 = x'_0 + b\cos\psi_{01} \\ y''_0 = y'_0 + b\sin\psi_{01} \end{cases} \tag{8.34}$$

and

$$\begin{cases} x''_1 = x'_1 + b\cos\psi_{01} \\ y''_1 = y'_1 + b\sin\psi_{01} \end{cases}, \tag{8.35}$$

respectively

Then, we describe the three-step reference path. The reference path in step 1 is the circular arc from point (x_0, y_0) to point (x''_0, y''_0) with a radius b; that is

$$\begin{cases} x(s) = x'_0 + b\cos(s/b + \psi_0) \\ y(s) = y'_0 + b\sin(s/b + \psi_0) \end{cases} \quad [0 \le s \le b(\psi_{01} - \psi_0)]. \tag{8.36}$$

The reference path in step 2 is the line segment from point (x''_0, y''_0) to (x''_1, y''_1); that is

$$\begin{cases} x(s) = x''_0 + \cos\psi_{01} s \\ y(s) = y''_0 + \sin\psi_{01} s \end{cases} \quad [0 \le s \le \sqrt{(x''_1 - x''_0)^2 + (y''_1 - y''_0)^2}]. \tag{8.37}$$

The reference path in step 3 is the circular arc from point (x''_1, y''_1) to point (x_1, y_1) with a radius b represented as

$$\begin{cases} x(s) = x'_1 + b\cos(\psi_{01} - s/b) \\ y(s) = y'_1 + b\sin(\psi_{01} - s/b) \end{cases} \quad [0 \le s \le b(\psi_{01} - \psi_0)]. \tag{8.38}$$

Therefore, the three-step initial locating reference path is described by Equations (8.36), (8.37), and (8.38).

8.5 Conclusion

Path planning of the Tethered Mobile Robot in highway maintenance is studied in this Chapter. Because the tracking control algorithm developed earlier in this report is based on reference trajectory tracking control, a parameterized reference path planning and quintic polynomial based speed manipulation method is used to produce the reference trajectory. The task determined reference path (e.g., crack following reference path) results from vision sensing and image recognition. A three-step initial locating path planning algorithm is developed to plan the reference path for the mobile robot so that it moves from the original parking position and orientation to the start point of a task determined reference path.

CHAPTER 9

CONCLUSIONS AND RECOMMENDATIONS

9.1 Conclusions

Tracking control algorithms for a mobile robot with two independent driven wheels are thoroughly and systematically studied in this report. Its purpose is to provide a theoretical basis for tracking control of the Tethered Mobile Robot as applied to highway maintenance. The tracking control algorithms developed in this report are valid for any differentially steered wheeled mobile robot or ground vehicle.

The tracking control algorithms are developed both on the basis of kinematic models and dynamic models. The main effort is made to develop a tracking control algorithm with strong robustness to uncertainties, such as system perturbations and external disturbances. The main contributions of this report is as follows:

- A kinematic model with wheel slippage is developed. The slippage influence is treated to be equivalent to parameter uncertainty in the system;
- An orientation equation for this kind of wheeled mobile robot is developed. It reveals the inherent relation between position tracking and orientation tracking;
- The influence of the location of the tracking point on the tracking ability is thoroughly studied. Two globally stable tracking control algorithms are developed according to the location of the tracking point relative to the baseline;
- A kinematic robust tracking control algorithm is developed on the basis of a kinematic model with uncertainty;
- Based on a full dynamic model with a detailed tire-ground contact model, reduced order dynamic models are developed. The influence of uncertainty on the dynamic system is analyzed and a matching condition is proven;

- An exponential position dynamic tracking control algorithm is developed. The control algorithm is based on the reduced order dynamic model and possesses strong robustness to system uncertainties;
- Variable structure control theory is used to construct a variable structure dynamic tracking control algorithm. Because the matching condition is hold for the uncertainties in the dynamic system, the exponential tracking control performance of the variable dynamic tracking control algorithm is invariant to system uncertainties, such as parameter perturbations, external disturbances, and unmodeled system dynamics.

The performance of the tracking control algorithms are studied through numerical simulations. The full dynamic model with detailed tire model is used in all simulations examining the dynamic tracking control algorithms.

The path planning problem related to highway maintenance is discussed. Since the tracking control algorithm is based on reference trajectory tracking, a mobile robot speed manipulation algorithm is proposed to produce the reference trajectory from the reference path.

9.2 Recommendations

As a final section of this report, we will make recommendations concerning the tracking control algorithm development and application. Although we have thoroughly studied both the kinematic and dynamic tracking control problem of the Tethered Mobile Robot, tracking control theory for this type of wheeled mobile robot is still an open research area. In order to improve the real-time performance and robustness, neural-network technology can be used in the tracking control algorithm to calculate the kinematic or dynamic compensation terms. An adaptive control algorithm is also a possible solution based on the strong uncertainties in the tracking control algorithm of the Tethered Mobile Robot. For tracking control of the Tethered Mobile Robot, application related control algorithms are also vital. Because of the computational complexity of the tracking control algorithm and the sensed data processing, a distributed computer control system should be developed to ensure flexibility, reliability and real-time performance. To improve total system

reliability, the control system must possess the self-diagnosis and fault detection. Application specific control software for this kind of mobile robot is also required.

REFERENCES

Badreddin, E. (1992) Real-time control of autonomous mobile robots. Robotics and Flexible Manufacturing Systems, Elsevier Science Publishers B. V., North-Holland, pp. 139-151.

Boyden, D. and Velinsky, S. A. (1993) Dynamic modeling of wheeled mobile robots. *AHMCT Research Report*, UCD-ARR-93-10-05-01, University of California at Davis.

Campion, G., d'Andrea-Novel, B., and Bastin, G. (1991) Controllability and state feedback stabilizability of nonholonomic mechanical systems. Advanced Robot Control, (Editor, Carlos Canudas de Wit), Springer-Verlag, Berlin, Germany.

Campion, G., Bastin, G., and d'Andrea-Novel, B., (1993) Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, pp. 462-469.

Corless, M. (1993) Control of uncertain nonlinear systems. *Journal of Dynamic Systems, Measurement, and Control*, Vol.115, pp.362-372.

Cox, I. J. and Wifong, G. T. (1990) Autonomous Robot Vehicles, Springer-Verlag.

d'Andrea-Novel, B., Bastin, G., and Campion, G. (1991) Modeling and control of nonholonomic wheeled mobile robots. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1130-1135.

d'Andrea-Novel, B., Bastin, G., and Campion, G. (1992) Dynamic feedback linearization of nonholonomic wheeled mobile robots. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2527-2532.

Gao, W. and Hung, J. C. (1993) Variable structure control of nonlinear systems: a new approach. *IEEE Transactions on Industrial Electronics*, Vol. 40, No.1, pp.45-55.

Guntur, R. and Sankar, S. (1980) A friction circle concept for Dugoff's tyre friction model. *International Journal of Vehicle Design*, Vol. 1, No. 4, pp. 373-377.

Hamdy, A. and Badreddin, E. (1992) Dynamic modeling of a wheeled mobile robot for identification and control. Robotics and Flexible Manufacturing Systems, Elsevier Science Publishers B. V., North-Holland, pp. 119-129.

Hemami, A., Mehrabi, M. G., and Cheng, R. M. H. (1990) A new control strategy for tracking in mobile robots and AGV's. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1122-1127.

Hemami, A., Mehrabi, M. G., and Cheng, R. M. H. (1992) Synthesis of an optimal control law for path tracking in mobile robots. *Automatica*, Vol. 28, No. 2, pp. 384-387.

Kanayama, Y., Nilipour, A., and Lelm C. A. (1988) A locomotion control method for autonomous vehicles. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1315-1317.

Kanayama, Y., Kimura, Y., Miyazaki, F., and Noguchi, T. (1990) A stable tracking control method for an autonomous mobile robot. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 384-389.

Kochekali, H. and Velinskey S. A., (1994) Adaptation of wheeled mobile robots to highway maintenance operations, *AHMCT Research Report UCD-ARR-94-1-5-01*, University of California at Davis.

Lasky, T. A., and Ravani, B., (1993) Path planning for robotic application in roadway crack sealing, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. .

Lee, S. S. and Williams, J. H. (1993) A fast tracking error control method for an autonomous mobile robot. *Robotica*, Vol. 11, pp. 205-215.

Leitmann, G. (1981) On the efficacy of nonlinear control in uncertain linear systems. *Journal of Dynamic Systems, Measurement, and Control*, Vol. 102, pp. 95-102.

Maeda, M., Maeda, Y., and Murakami, S. (1991) Fuzzy drive control of an autonomous mobile robot. *Fuzzy Sets and Systems*, Vol. 39, pp. 195-204.

Miyata, H. and Ohkita, M. (1992) Control of an autonomous mobile robot. Robotics and Flexible Manufacturing Systems, Elsevier Science Publishers. B. V. (North-Holland), pp. 151-160.

Muir, P. F. and Neuman, C. P. (1987) Kinematic modeling of wheeled mobile robots. *Journal of Robotic Systems*, Vol. 4, No. 2, pp. 281-340

Nelson, W. L. (1989) Continuous steering-function control of robot carts. *IEEE Transactions on Industrial Electronics*, Vol. 36, No. 3, pp. 330-337.

Nelson, W. L. and Cox, I. J. (1990) Local path control for an autonomous vehicle. Autonomous Robot Vehicles, Springger-Verlag, pp. 38-44.

Pomet, J. B., Thuilot, B., Bastin, G. and Campion, G. (1992) A hybrid strategy for the feedback stabilization of nonholonomic mobile robots. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 129-134.

Reister, D. B., Pin, F. G., (1994) Time-optimal trajectories for mobile robot with two independently driven wheels. *The International Journal of Robotics Research*, Vol. 13, No. 1, pp. 38-54.

Saga, K., Sugasaka, T., Sekiguchi, M., Nagata, S., and Asakawa, K. (1992) Mobile robot control by neural networks using self-supervised learning. *IEEE Transactions on Industrial Electronics*, Vol. 39, No. 6, pp. 537-541.

Samson, C., (1991) Velocity and torque feedback control of a nonholonomic cart. Advanced Robot Control, (Editor, Carlos Canudas de Wit), Springer-Verlag, Berlin, Germany, 1991.

Sarkar, N., Yun, X. , Kumar, V., (1994) Control of mechanical systems with rolling constraints: application to dynamic control of mobile robots. *The International Journal of Robotics Research*, Vol. 13, No. 1, pp. 55.69.

Shin, D. H. and Singh, S. (1992) Explicit path tracking by autonomous vehicles. *Robotica*, Vol. 10, pp. 539-554.

Spong, M. W. and Vidyasagar, M. (1989) Robot Dynamics and Control. John Wiley & Sons, Inc., 1989.

Sϕrdalen, O. J., Wit, C. C. de (1993), Exponential control law for a mobile robots: extension to path following. *IEEE Transactions on automatic control*, Vol. 37, No. 11, pp. 1791-1797.

Utkin, V. I. (1978) Sliding Modes and Their Application in Variable Structure Systems, Mir Publishers Moscow, 1978.

Utkin, V. I. (1992) Sliding Modes in Control Optimization, Springer-Verlag Berlin, Heidelberg, Germany, 1992.

Vidyasagar, M. (1993) Nonlinear Systems Analysis, Second Edition, Prentice Hall, Englewood Cliffs, New Jersey.

Walsh, G., Tilbury, D., Sastry, S., Murray, R., and Laumond J. P. (1994) Stabilization of trajectories for systems with nonholonomic constraints. *IEEE Transactions on Automatic Control*, Vol. 39, No. 1, pp. 216-222.

Winters, S. E., Hong, D., Velinsky, S. A., and Yamazaki, K. (1994) A new robotic system concept for automating highway maintenance operations. *Proc. of SPACE'94-ASCE Conference on Robotics for Challenging Environments*, pp. 374-382.

Winters, S. E. and Velinskey, S. A. (1992) Development of a tethered mobile robot (TMR) for highway maintenance. *AHMCT Research Report UCD-ARR-92-11-25-01*, University of California at Davis.

Wit, C. C. de, Sϕrdalen, O. J., (1992) Exponential stabilization of mobile robots with nonholomic constraints. *IEEE Transactions on automatic control*, Vol. 37, No. 11, pp. 1791-1797.

Zhang, Y. and Velinsky, S. A. (1994a) On the tracking control of mobile robot with differential driven wheels, to be published.

Zhang, Y. and Velinsky, S. A. (1994b) Robust control of a tethered mobile robot, to be published.

Zhang, Y. L., and Velinsky, S. A. (1994c) Dynamic model based tracking control of wheeled mobile robots, to be published.

APPENDIX A

MATLAB SIMULATION CODE

A.1 - Exponential Kinematic Tracking Control Algorithm

```
%#########################################################################
% MOBILE ROBOT EXPONENTIAL KINEMATIC TRACKING CONTROL
% ALGORITHM SIMULATION SOFTWARE
%
% Author: Dr. Yulin Zhang
%
%        Professor
%        Changsha Institute of Technology
%        Changsha, Hunan 410073, China
%
%        Visiting Research Scientist
%        Department of Mechanical and Aeronautical Engineering
%        University of California
%        Davis, California 95616, U. S. A.
%
% Copyright: Advanced Highway Maintenance and Construction Technology Center
%                 May 25, 1994
%
% Programming language: Matlab
%#########################################################################
clg
clear

% para: the file to set parameters to be
%       global variables;
% c_track:distance from the track point to
%         the center of base line;
% e_base:1/2 leigth of wheel base line;
% rwheel:radius of wheel;
%
rwheel=0.3048;
e_base=0.9144/2;
c_track=0.9144;
pai=3.1415926;

%setfb: set feedback matrix and solve
%       static Riccati equation;
Q=zeros(2,2);
Q(1,1) = 1.0;
Q(2,2) = 1.0;

Kp=zeros(2,2);
Kp(1,1) = 3.0;
Kp(2,2) = 3.0;
```

```
kc=1;

B=zeros(2,2);
A=-Kp
P=are(A,B,Q)

global rwheel e_base c_track Kp P kc pai;

tmax=4;
t0=0;tf=tmax*2*pai;
x0=[0 0 0.7854]';
%x0=[0 4 0]';
%x0=[0 0 0]';
h=0.1;
im=(tf-t0)/h;
tc=t0;
xc=x0;
t(1)=t0;
x(1,:)=x0';
for i=2:im+1
   xx=xc;
   tt=tc;
    [tc,xc]=rk42_on(tt,xx,h,3);
   t(i)=tc;
    x(i,:)=xc';
end
%
for i=0:tmax*20*pai;
    ti=0.1*i;
    tf(i+1)=ti;
    xff=refe_on(ti);
     xxf(i+1,:)=xff(:,1)';
end;
subplot(221),plot(x(:,1),x(:,2),xxf(:,1),xxf(:,2))
xlabel('x,meter');
ylabel('y,meter');
grid;
subplot(222),plot(t(:),x(:,3),tf,xxf(:,3))
xlabel('time,second');
ylabel('orientation');
grid;
subplot(223),plot(t(:),x(:,1),tf,xxf(:,1))
xlabel('time,second');
ylabel('x,meter');
grid;
subplot(224),plot(t(:),x(:,2),tf,xxf(:,2))
xlabel('time,second');
ylabel('y,meter');
grid;


%=============================================================
%rk4:function of Runge-Kutta method
function [t,x]=rk42_on(t0,x0,h,n)
```

```
x=zeros(n,1);
xt=zeros(n,1);
h2=h/2;
h6=h/6;
dxdt=diff_on(t0,x0);
th=t0+h2;
xt=x0+h2*dxdt;
dxt=diff_on(th,xt);
xt=x0+h2*dxt;
dxm=diff_on(th,xt);
xt=x0+h*dxm;
t=t0+h;
dxm=dxt+dxm;
dxt=diff_on(t,xt);
x=x0+h6*(dxdt+dxt+2.0*dxm);


%=============================================================
% function for kinematics
function xdot=diff_on(t,x)
% initialization
g=[0 0]';
f=[0 0]';
h=[0 0]';
u = [0 0]';
xdot = [0 0 0]';
xfdotp=[0 0]';
aa1=zeros(2,2);
inaa1=zeros(2,2);
gtg=[0 0]';

% calculation of the kinematics
cita=x(3);
aareal=kinemm_on(1.0,1.0,cita);
aa=kinemm_on(1.0,1.0,cita);
aa1(1,1)=aa(1,1);
aa1(1,2)=aa(1,2);
aa1(2,1)=aa(2,1);
aa1(2,2)=aa(2,2);
g(1)=aa(3,1);
g(2)=aa(3,2);
inaa1=inv(aa1);
gtg=inv(g'*g)*g;
f=aa1*gtg;
h=g'*inaa1;

% calculation of the reference path
xxf=refe_on(t);
xf=xxf(:,1);
xfdot=xxf(:,2);
xfdotp(1)=xfdot(1);
xfdotp(2)=xfdot(2);

% caculation of the errors
```

```
ep=zeros(2,1);
ep(1)=x(1)-xf(1);
ep(2)=x(2)-xf(2);
ec=x(3)-xf(3);

% calculation of linearized control law
v=xfdotp-Kp*ep+f*kc*ec;
v3=-kc*ec;

% caculation of control inputs
u=inaa1*v+g'*gtg*v3;
xdot=aareal*u;


%=========================================================
% KINEMM: function for wheeled mobile robot
% kinematics matrix;
% e_base: one second of the wheel base line
%         distance;
% c_track: distace of track point to the
%         center of wheel base line;
% rwheel: radius of wheel;
% e_base,c_track,rwheel are global variables;
% ccl: slippage coefficient of left wheel;
% ccr: slippage coefficient of right wheel;
% cita: orientation angular;
%
function aa=kinemm_on(ccl,ccr,cita)
aa=zeros(3,2);
aa(1,1) = ccl*rwheel*(c_track*sin(cita)+e_base*cos(cita));
aa(2,1) = ccl*rwheel*(-c_track*cos(cita)+e_base*sin(cita));
aa(3,1) = -ccl*rwheel;
aa(1,2) = ccr*rwheel*(-c_track*sin(cita)+e_base*cos(cita));
aa(2,2) = ccr*rwheel*(c_track*cos(cita)+e_base*sin(cita));
aa(3,2) = ccr*rwheel;
aa = aa/(2*e_base);


%=========================================================
% refe:function for caculating the
%          control reference;
% xf(i,1):position and orientation of
%         reference path;
% xf(i,2):derivatives of reference path;
%
function xf=refe_on(t)
xf=zeros(3,2);
xf(1,1)=t;
xf(2,1)=t+4*sin(t/4);
xf(3,1)=atan(1+cos(t/4));
xf(1,2)=1;
xf(2,2)=1+cos(t/4);
cs=1+cos(t/4);
cs2=1+cs*cs;
```

```
xf(3,2)=-sin(t/4)/4/cs2;
%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

A.2 - Global Convergence Kinematic Tracking Control Algorithm

```
%#########################################################################
% MOBILE ROBOT GLOBAL CONVERGENCE KINEMATIC TRACKING CONTROL
% ALGORITHM SIMULATION SOFTWARE
%
% Author: Dr. Yulin Zhang
%
%       Professor
%       Changsha Institute of Technology
%       Changsha, Hunan 410073, China
%
%       Visiting Research Scientist
%       Department of Mechanical and Aeronautical Engineering
%       University of California
%       Davis, California 95616, U. S. A.
%
% Copyright: Advanced Highway Maintenance and Construction Technology Center
%               May 25, 1994
%
% Programming language: Matlab
%#########################################################################
clg
clear

% para: the file to set parameters to be
%     global variables;
% c_track:distance from the track point to
%       the center of base line;
% e_base:1/2 leigth of wheel base line;
% rwheel:radius of wheel;
%
rwheel=0.3048;
e_base=0.9144/2;
c_track=0.9144;
pai=3.1415926;

%setfb: set feedback matrix and solve
%     static Riccati equation;
Q=zeros(2,2);
Q(1,1) = 1.0;
Q(2,2) = 1.0;

Kp=zeros(2,2);
Kp(1,1) = 3.0;
Kp(2,2) = 3.0;
kc=5;

B=zeros(2,2);
A=-Kp
P=are(A,B,Q)

global rwheel e_base c_track Kp P kc pai;
```

```
tmax=4;
t0=0;tf=tmax*2*pai;
%x0=[0 0 1.1071]';
%x0=[0 0 0.7854]';
%x0=[0 4 0]';
x0=[0 0 0]';
h=0.1;
im=(tf-t0)/h;
tc=t0;
xc=x0;
t(1)=t0;
x(1,:)=x0';
for i=2:im+1
   xx=xc;
   tt=tc;
    [tc,xc]=rk42_base(tt,xx,h,3);
   t(i)=tc;
    x(i,:)=xc';
end
%
for i=0:tmax*20*pai;
    ti=0.1*i;
    tf(i+1)=ti;
    xff=refe_on(ti);
     xxf(i+1,:)=xff(:,1)';
end;
subplot(221),plot(x(:,1),x(:,2),xxf(:,1),xxf(:,2))
xlabel('x,meter');
ylabel('y,meter');
grid;
subplot(222),plot(t(:),x(:,3),tf,xxf(:,3))
xlabel('time,second');
ylabel('orientation,degree');
grid;
subplot(223),plot(t(:),x(:,1),tf,xxf(:,1))
xlabel('time,second');
ylabel('x,meter');
grid;
subplot(224),plot(t(:),x(:,2),tf,xxf(:,2))
xlabel('time,second');
ylabel('y,meter');
grid;


%============================================================
%rk4:function of Runge-Kutta method
function [t,x]=rk42_base(t0,x0,h,n)
x=zeros(n,1);
xt=zeros(n,1);
h2=h/2;
h6=h/6;
dxdt=diff_base(t0,x0);
th=t0+h2;
```

```
xt=x0+h2*dxdt;
dxt=diff_base(th,xt);
xt=x0+h2*dxt;
dxm=diff_base(th,xt);
xt=x0+h*dxm;
t=t0+h;
dxm=dxt+dxm;
dxt=diff_base(t,xt);
x=x0+h6*(dxdt+dxt+2.0*dxm);


%=========================================================
% function for kinematics
function xdot=diff_base(t,x)
% initialization
u = [0 0]';
xdot = [0 0 0]';
xfdotp=[0 0]';
aa1=zeros(2,2);
inaa1=zeros(2,2);

% calculation of the kinematics
cita=x(3);
aareal=kinemm_base(1.0,1.0,cita);
aa1=uk_base(1.0,1.0);
inaa1=inv(aa1);

% calculation of the reference path
xxf=refe_on(t);
xf=xxf(:,1);
xfdot=xxf(:,2);
xfdotp(1)=xfdot(1);
xfdotp(2)=xfdot(2);
citaf=xf(3);

% caculation of the errors
ep=zeros(2,1);
ep(1)=x(1)-xf(1);
ep(2)=x(2)-xf(2);
ec=x(3)-xf(3);
cita=x(3);
eeta = ep(1)*cos(cita)+ep(2)*sin(cita);
ezta =-ep(1)*sin(cita)+ep(2)*cos(cita);

etaf=xfdotp(1)*cos(citaf)+xfdotp(2)*sin(citaf);
cfdot=xfdot(3);


% calculation of linearized control law
v=[0 0]';
km=Kp(1,1)';
v(1)=etaf-km*eeta;
v(2)=cfdot-kc*sin(ec/2)-2*etaf*ezta;
```

```
% caculation of control inputs
u=inaa1*v;
xdot=aareal*u;


%=====================================================
% KINEMM: function for wheeled mobile robot
% kinematics matrix;
% e_base: one second of the wheel base line
%         distance;
% c_track: distace of track point to the
%        center of wheel base line;
% rwheel: radius of wheel;
% e_base,c_track,rwheel are global variables;
% ccl: slippage coefficient of left wheel;
% ccr: slippage coefficient of right wheel;
% cita: orientation angular;
%
function aa=kinemm_base(ccl,ccr,cita)
aa=zeros(3,2);
aa(1,1) = 0.5*ccl*rwheel*cos(cita);
aa(2,1) = 0.5*ccl*rwheel*sin(cita);
aa(3,1) = -ccl*rwheel/(2*e_base);
aa(1,2) = 0.5*ccr*rwheel*cos(cita);
aa(2,2) = 0.5*ccr*rwheel*sin(cita);
aa(3,2) = ccr*rwheel/(2*e_base);


%=====================================================
% refe:function for caculating the
%          control reference;
% xf(i,1):position and orientation of
%        reference path;
% xf(i,2):derivatives of reference path;
%
function xf=refe_on(t)
xf=zeros(3,2);
xf(1,1)=2*t;
xf(2,1)=2+t;
xf(3,1)=atan(1/2);
xf(1,2)=2;
xf(2,2)=1;
xf(3,2)=0;

%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

A.3 - Robust Kinematic Tracking Control Algorithm

```
%#########################################################################
% MOBILE ROBOT GLOBAL CONVERGENCE KINEMATIC TRACKING CONTROL
% ALGORITHM SIMULATION SOFTWARE
%
% Author: Dr. Yulin Zhang
%
%       Professor
%       Changsha Institute of Technology
%       Changsha, Hunan 410073, China
%
%       Visiting Research Scientist
%       Department of Mechanical and Aeronautical Engineering
%       University of California
%       Davis, California 95616, U. S. A.
%
% Copyright: Advanced Highway Maintenance and Construction Technology Center
%               May 25, 1994
%
% Programming language: Matlab
%#########################################################################
clg
clear
% para: the file to set parameters to be
%     global variables;
% c_track: distance from the track point to
%       the center of base line;
% e_base: 1/2 leigth of wheel base line;
% rwheel: radius of wheel;
%
rwheel=0.3048;
e_base=0.9144/2;
c_track=0.9144;
pai=3.1415926;

%setfb: set feedback matrix and solve
%     static Riccati equation;
Q=zeros(2,2);
Q(1,1) = 1.0;
Q(2,2) = 1.0;

Kp=zeros(2,2);
Kp(1,1) = 3.0;
Kp(2,2) = 3.0;
kc=2;

B=zeros(2,2);
A=-Kp
P=are(A,B,Q)

global rwheel e_base c_track Kp P kc pai;
```

```
t0=0;tf=2*pai;
x0=[0 4 -pai/2]';
%x0=[0 0 0]';
h=0.1;
im=(tf-t0)/h;
tc=t0;
xc=x0;
t(1)=t0;
x(1,:)=x0';
for i=2:im+1
   xx=xc;
   tt=tc;
    [tc,xc]=rk42(tt,xx,h,3);
   t(i)=tc;
    x(i,:)=xc';
end
%
for i=0:20*pai;
     ti=0.1*i;
     tf(i+1)=ti;
     xff=refe(ti);
      xxf(i+1,:)=xff(:,1)';
end;
subplot(221),plot(x(:,1),x(:,2),xxf(:,1),xxf(:,2))
xlabel('x,meter');
ylabel('y,meter');
grid;
subplot(222),plot(t(:),x(:,3),tf,xxf(:,3))
xlabel('time,second');
ylabel('orientation,degree');
grid;
subplot(223),plot(t(:),x(:,1),tf,xxf(:,1))
xlabel('time,second');
ylabel('x,meter');
grid;
subplot(224),plot(t(:),x(:,2),tf,xxf(:,2))
xlabel('time,second');
ylabel('y,meter');
grid;


%=========================================================
%rk4:function of Runge-Kutta method
function [t,x]=rk42(t0,x0,h,n)
x=zeros(n,1);
xt=zeros(n,1);
h2=h/2;
h6=h/6;
dxdt=diff2(t0,x0);
th=t0+h2;
xt=x0+h2*dxdt;
dxt=diff2(th,xt);
xt=x0+h2*dxt;
```

```
dxm=diff2(th,xt);
xt=x0+h*dxm;
t=t0+h;
dxm=dxt+dxm;
dxt=diff2(t,xt);
x=x0+h6*(dxdt+dxt+2.0*dxm);

%=============================================================
% function for kinematics
function xdot=diff2(t,x)
% initialization
g=[0 0]';
f=[0 0]';
h=[0 0]';
u = [0 0]';
xdot = [0 0 0]';
xfdotp=[0 0]';
aa1=zeros(2,2);
inaa1=zeros(2,2);
gtg=[0 0]';

% calculation of the kinematics
cita=x(3);
aareal=kinemm(1.0,1.0,cita);
aa=kinemm(1.0,1.0,cita);
aa1(1,1)=aa(1,1);
aa1(1,2)=aa(1,2);
aa1(2,1)=aa(2,1);
aa1(2,2)=aa(2,2);
g(1)=aa(3,1);
g(2)=aa(3,2);
inaa1=inv(aa1);
gtg=inv(g'*g)*g;
f=aa1*gtg;
h=g'*inaa1;

% calculation of the reference path
xxf=refe(t);
xf=xxf(:,1);
xfdot=xxf(:,2);
xfdotp(1)=xfdot(1);
xfdotp(2)=xfdot(2);

% caculation of the errors
ep=zeros(2,1);
ep(1)=x(1)-xf(1);
ep(2)=x(2)-xf(2);
ec=x(3)-xf(3);

% calculation of robust control dv
w=P*ep;
alf=1;
km=Kp(1,1);
dd=0.1;
```

```
ro=alf*(2+km*sqrt(ep'*ep));
aw=sqrt(w'*w);
if aw>=dd
  dv=-ro*w/aw;
else
  dv=-ro*w/dd;
end

% calculation of linearized control law
v=xfdotp-Kp*ep+f*kc*ec+dv;
v3=-kc*ec;

% caculation of control inputs
u=inaa1*v+g'*gtg*v3;
xdot=aareal*u;


%=============================================================
% KINEMM: function for wheeled mobile robot
% kinematics matrix;
% e_base: one second of the wheel base line
%        distance;
% c_track: distace of track point to the
%       center of wheel base line;
% rwheel: radius of wheel;
% e_base,c_track,rwheel are global variables;
% ccl: slippage coefficient of left wheel;
% ccr: slippage coefficient of right wheel;
% cita: orientation angular;
%
function aa=kinemm(ccl,ccr,cita)
aa=zeros(3,2);
aa(1,1) = ccl*rwheel*(c_track*cos(cita)-e_base*sin(cita));
aa(2,1) = ccl*rwheel*(c_track*sin(cita)+e_base*cos(cita));
aa(3,1) = -ccl*rwheel;
aa(1,2) = ccr*rwheel*(-c_track*cos(cita)-e_base*sin(cita));
aa(2,2) = ccr*rwheel*(-c_track*sin(cita)+e_base*cos(cita));
aa(3,2) = ccr*rwheel;
aa = aa/(2*e_base);


%=============================================================
% refe:function for caculating the
%          control reference;
% xf(i,1):position and orientation of
%        reference path;
% xf(i,2):derivatives of reference path;
%
function xf=refe(t)
xf=zeros(3,2);
xf(1,1)=4*sin(t/4);
xf(2,1)=4*cos(t/4);
xf(3,1)=-pai/2-t/4;
xf(1,2)=cos(t/4);
```

```
xf(2,2)=-sin(t/4);
xf(3,2)=-1/4;

%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

A.4 - Robust Dynamic Tracking Control Algorithm

```
%#########################################################################
% MOBILE ROBOT DYNAMIC TRACKING CONTROL SIMULATION SOFTWARE
%
% Author: Dr. Yulin Zhang
%
%       Professor
%       Changsha Institute of Technology
%       Changsha, Hunan 410073, China
%
%       Visiting Research Scientist
%       Department of Mechanical and Aeronautical Engineering
%       University of California
%       Davis, California 95616, U. S. A.
%
% Copyright: Advanced Highway Maintenance and Construction Technology Center
%               May 25, 1994
%
% Programming language: Matlab
%#########################################################################

clear
clg
%+++++++++++++ physical constant ++++++++++++++++++++++++++++++++++
g=9.801;
pai=3.1415926;
m=272;
Izz=407;
Itt=6.78;
Bt=0.0;
mu=0.8;
roc=0.0;
cx=40034;
cy=40034;
%  m---mass of mobile robot, kg
% Izz---yaw moment of inertia, kg.m.m
% Itt---combined wheel, gearbox, and motor
%      moments of inertia, kg.m.m
% mu---road/tire interface coefficient of friction
% cx---longitudinal tire stiffness, N/rad
% cy---lateral tire stiffness, N/rad
%+++++++++++++ structure parameters +++++++++++++++++++++++++++++++
a=0.762;
b=0.6096;
c=0.2286;
d=0.9144;
e=0.9144;
r=0.3048;
% a---distance from center of mass to the front castor, m
% b---distance from baseline to the center of mass, m
% c---distance from center of mass to the linkage point, m
% d---length of baseline of the two rear wheels, m
% e---distance from the center of mass to the tool
```

```
%     operation point, m
% r---tire radius, m
%+++++++++++++ control parameters ++++++++++++++++++++++++++++++++
Kd=zeros(2,2);
Kd(1,1)=0.96;
Kd(2,2)=0.96;

Kp=zeros(2,2);
Kp(1,1) =0.16;
Kp(2,2) =0.16;

O=zeros(2,2);
I=zeros(2,2);
I(1,1)=1;
I(2,2)=1;

Q=[I O;O I];
B=[O O;O O];
A=[O I;-Kp -Kd];
P=are(A,B,Q)
BI=[O;I];
global pai g m Izz Itt mu cx cy a b c d e r roc Bt Kp Kd P BI;

tmax=4;
t0=0;tf=tmax*2*pai;
%x0=[0 0 1.1071 0 0 0 0 0]';
%x0=[0 0 0.7854 0 0 0 0 0]';
x0=[0 4 0 0 0 0 0 0]';
%x0=[0 0 0 0 0 0 0 0]';
%x0=[0 2 0.4636 0 0 0 0 0]';
h=0.1;
im=(tf-t0)/h;
tc=t0;
xc=x0;
t(1)=t0;
x(1,:)=x0';
for i=2:im+1
   xx=xc;
   tt=tc;
    [tc,xc]=rk42_dyn(tt,xx,h,8);
   t(i)=tc;
    x(i,:)=xc';
end
%
for i=0:tmax*20*pai;
    ti=0.1*i;
    tf(i+1)=ti;
    xff=refe_path(ti);
     xxf(i+1,:)=xff(:,1)';
end;

vslip=x(:,5)-b*x(:,6);
uleft=x(:,4)+(d/2)*x(:,6);
uright=x(:,4)-(d/2)*x(:,6);
```

```
ulslip=r*x(:,7)-uleft;
urslip=r*x(:,8)-uright;

subplot(221),plot(x(:,2),x(:,1),xxf(:,2),xxf(:,1))
xlabel('y,meter');
ylabel('x,meter');
grid;
subplot(222),plot(t(:),x(:,3),tf,xxf(:,3))
xlabel('time,second');
ylabel('orientation,radian');
grid;
%subplot(223),plot(t(:),x(:,1),tf,xxf(:,1))
subplot(223),plot(t(:),ulslip,t(:),urslip);
xlabel('time,second');
%ylabel('x,meter');
ylabel('longitudinal slip,m/s');
grid;
%subplot(224),plot(t(:),x(:,2),tf,xxf(:,2))
subplot(224),plot(t(:),vslip);
xlabel('time,second');
%ylabel('y,meter');
ylabel('lateral slip,m/s');
grid;


%=============================================================
%rk4: function of Runge-Kutta method
function [t,x]=rk42_dyn(t0,x0,h,n)
x=zeros(n,1);
xt=zeros(n,1);
h2=h/2;
h6=h/6;
dxdt=diff_dyn(t0,x0);
th=t0+h2;
xt=x0+h2*dxdt;
dxt=diff_dyn(th,xt);
xt=x0+h2*dxt;
dxm=diff_dyn(th,xt);
xt=x0+h*dxm;
t=t0+h;
dxm=dxt+dxm;
dxt=diff_dyn(t,xt);
x=x0+h6*(dxdt+dxt+2.0*dxm);


%=============================================================
% function qdot=diff_dyn(t,q)
% t:time
% q(8): state variables {x,y,fei,u,v,w_yaw,w_left,w_right)}
function qdot=diff_dyn(t,q)
qdot=zeros(8,1);
Fx=[0 0 0]';
Fy=[0 0 0]';
Ft=[0 0 0]';
```

```
Tw=[0 0]';
% Ft: tool operation force (Fxt,Fyt,Fnt), Fnt=-Fzt
% Tw: torque applied to wheels (Tw_left,Tw_right)
% Fx: tire forces in x direction (Fxl,Fxr,Fxc)
% Fy: tire forces in y direction (Fyl,Fyr,Fyc)

fei=q(3);
u=q(4);
v=q(5);
w_yaw=q(6);
w_left=q(7);
w_right=q(8);

Ft=toolforce(t);
Tw=control(t,q);

Fnt=Ft(3);
T_left=Tw(1);
T_right=Tw(2);

Fx=[0 0 0]';
Fy=[0 0 0]';

Fnc=(b*m*g+(e-b)*Fnt)/(a+b);
Fn=(a*m*g-(a+e)*Fnt)/(a+b)/2;

u_left=u+w_yaw*d/2;
u_right=u-w_yaw*d/2;
v_slip=v-w_yaw*b;

u_c=u;
v_c=v+a*w_yaw;

[Fx(1),Fy(1)]=tiredriven(u_left,v_slip,w_left,T_left,Fn);
[Fx(2),Fy(2)]=tiredriven(u_right,v_slip,w_right,T_right,Fn);
[Fx(3),Fy(3)]=castor(u_c,v_c,Fnc);

udot=(Fx(1)+Fx(2)+Fx(3)+Ft(1))/m+v*w_yaw;
vdot=(Fy(1)+Fy(2)+Fy(3)+Ft(2))/m-u*w_yaw;
wdot=(0.5*d*(Fx(1)-Fx(2))-b*(Fy(1)+Fy(2))+a*Fy(3)-e*Ft(2))/Izz;
xdot=cos(fei)*u-sin(fei)*v;
ydot=sin(fei)*u+cos(fei)*v;

w_ldot=(Tw(1)-Bt*w_left-r*Fx(1))/Itt;
w_rdot=(Tw(2)-Bt*w_right-r*Fx(2))/Itt;

qdot(1)=xdot;
qdot(2)=ydot;
qdot(3)=w_yaw;
qdot(4)=udot;
qdot(5)=vdot;
qdot(6)=wdot;
qdot(7)=w_ldot;
qdot(8)=w_rdot;
```

```
%============================================================
function Ft=toolforce(t)
% function for calculating the tool operational forces
% Ft=(Fxt,Fyt,Fnt)', Fnt=-Fzt
Ft=[0 0 0]';
if t<=10,
   Ft(1)=0;
   Ft(2)=0;
   Ft(3)=0;
else
   Ft(1)=-200;
   Ft(2)=0;
   Ft(3)=0;
end;


%============================================================
%function [Fx_c,Fy_c]=castor(u_c,v_c,Fnc)
% for the calculation of castor resistance force
% u_c: longitudinal speed of the mobile robot
% v_c: lateral speed of the castor
function [Fx_c,Fy_c]=castor(u_c,v_c,Fnc)

vv=sqrt(u_c*u_c+v_c*v_c);
if vv==0,
  Fx_c=0;
  Fy_c=0;
else
  udir=u_c/vv;
  vdir=v_c/vv;
  Fx_c=-roc*udir*Fnc;
  Fy_c=-roc*vdir*Fnc;
end


%============================================================
% function for calculation driven wheel tire friction model
% u_w: longitudinal speed of wheel axis
% u_w: lateral speed of wheel axis
% w_w: rotational angular speed of wheel
% T_w: torque applied on wheel
% Fn: normal speed applied on wheel
% return in [Fx_w,Fy_w]
function [Fx_w,Fy_w]=tiredriven(u_w,v_w,w_w,T_w,Fn)
if u_w==0,
   if w_w==0,
      Fx_w=T_w/r;
   else
      Fx_w=sign(w_w)*mu*Fn;
   end;
   if v_w==0,
      Fy_w=0;
```

```
    else
       Fy_w= -sign(v_w)*mu*Fn;
    end;
 else
    lamd=abs(v_w/u_w);
    if w_w==0,
       slip=1;
    else
       if w_w*u_w>0,
          slip=1-r*w_w/u_w;
          if slip<=-3,
             slip=-3;
          end;
       else
          slip=1;
       end;
    end;
    clng=0.001*cx*Fn;
    clat=0.001*cy*Fn;
    scom=sqrt(slip*slip+lamd*lamd);
    mudyn=mu*(1-0.0034*abs(u_w)*scom);
    if mudyn<=0.7*mu,
      mudyn=0.7*mu;
    end;
    if slip==1,
       ccll=sqrt(clng*clng+clat*clat*lamd*lamd);
       if ccll<=0.000001,
          ccll=0.000001;
       end;
       muf=mudyn*Fn;
       Fx_w=-sign(u_w)*clng*muf/ccll;
       Fy_w=-sign(v_w)*clat*muf*lamd/ccll;
    else
       Fxd=-sign(u_w)*clng*slip/(1-slip);
       Fyd=-sign(v_w)*clat*lamd/(1-slip);
       mud=sqrt(Fxd*Fxd+Fyd*Fyd)/Fn;
       if mud<=mudyn/2,
          Fx_w=Fxd;
          Fy_w=Fyd;
       else
          mures=mudyn*(1-mudyn/mud/4);
          Fx_w=Fxd*(mures/mud);
          Fy_w=Fyd*(mures/mud);
       end;
    end;
 end


 %=========================================================
 % function for simulating the controller
 % t: time
 % q: state variables
 % Tw=[T_left, T_right]' torque applied to wheels
```

```
function Tw=control(t,q)
dz=[0 0]';
Tw=[0 0]';
ff=[0 0]';
z=[0 0]';
ed=[0 0]';
ep=[0 0]';
rqdot2=[0 0]';
rqdot=[0 0]';
rq=[0 0]';
oqp=[0 0]';
oqdot=[0 0]';
qf=zeros(3,3);
Gp=zeros(2,2);
D1=m*r*r+2*Itt;
D2=2*r*r*Izz+2*m*b*b*r*r+d*d*Itt;
fei=q(3);
u=q(4);
v=q(5);
w=q(6);
cosfei=cos(fei);
sinfei=sin(fei);
rD1=r/D1;
rbdD2=r*b*d/D2;
Gp(1,1)=rD1*cosfei-rbdD2*sinfei;
Gp(1,2)=rD1*cosfei+rbdD2*sinfei;
Gp(2,1)=rD1*sinfei+rbdD2*cosfei;
Gp(2,2)=rD1*sinfei-rbdD2*cosfei;
IttD1=2*Itt/D1;
IID2=(2*r*r*Izz+d*d*Itt)/D2;
ff(1)=-IttD1*v*w*cosfei-IID2*u*w*sinfei;
ff(2)=-IttD1*v*w*sinfei+IID2*u*w*cosfei;

oqdot(1)=u*cosfei-b*w*sinfei;
oqdot(2)=u*sinfei+b*w*cosfei;
oqp(1)=q(1);
oqp(2)=q(2);

qf=refe_path(t);
rqdot2(1)=qf(1,3);
rqdot2(2)=qf(2,3);
rqdot(1)=qf(1,2);
rqdot(2)=qf(2,2);
rq(1)=qf(1,1);
rq(2)=qf(2,1);

ed=oqdot-rqdot;
ep=oqp-rq;

%robust control algorithm
%dz=robust(ed,ep);
%z=rqdot2-Kd*ed-Kp*ep+dz;

%variable structure algorithm
```

```
dz=sliding(ed,ep);
z=rqdot2+dz;

Tw=inv(Gp)*(z-ff);


%==========================================================
function dz=robust(ed,ep)
dz=[0 0]';
ee=[ep;ed];
bpe=BI'*P*ee;
mm=bpe'*bpe;
mpe=sqrt(mm);
rro=0.5*(1.5+3.5*(e'*e));
if mpe>0.1,
  dz=-rro*bpe/mpe;
else
  dz=-rro*bpe/0.1;
end


%==========================================================
% refe: function for calculating the
%         control reference;
% xf(i,1): position and orientation of
%        reference path;
% xf(i,2): derivatives of reference path;
%
function xf=refe_path(t)
xf=zeros(3,3);
xf(1,1)=4*sin(t/4);
xf(2,1)=4*cos(t/4);
xf(3,1)= -t/4;
xf(1,2)=cos(t/4);
xf(2,2)=-sin(t/4);
xf(3,2)=-1/4;
xf(1,3)=-(1/4)*sin(t/4);
xf(2,3)=-(1/4)*cos(t/4);
xf(3,3)=0;

%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

A.5 - Variable Structure Dynamic Tracking Control Algorithm

```
%#########################################################################
% MOBILE ROBOT VARIABLE STRUCTURE DYNAMIC TRACKING CONTROL
% ALGORITHM SIMULATION SOFTWARE
%
% Author: Dr. Yulin Zhang
%        Professor
%        Changsha Institute of Technology
%        Changsha, Hunan 410073, China
%
%        Visiting Research Scientist
%        Department of Mechanical and Aeronautical Engineering
%        University of California
%        Davis, California 95616, U. S. A.
%
% Copyright: Advanced Highway Maintenance and Construction Technology Center
%                 May 25, 1994
%
% Programming language: Matlab
%#########################################################################
clear
clg

g=9.801;
pai=3.1415926;
m=272;
Izz=407;
Itt=6.78;
Bt=0.0;
mu=0.8;
roc=0.0;
cx=40034;
cy=40034;
%  m---mass of mobile robot, kg
% Izz---yaw moment of inertia, kg.m.m
% Itt---combined wheel, gearbox, and motor moments of inertia, kg.m.m
% mu---road/tire interface coefficient of friction
% cx---longitudinal tire stiffness, N/rad
% cy---lateral tire stiffness, N/rad

a=0.762;
b=0.6096;
c=0.2286;
d=0.9144;
e=0.9144;
r=0.3048;
% a---distance from centre of mass to the front castor, m
% b---distance from baseline to the center of mass, m
% c---distance from center of mass to the linkage point, m
% d---leigth of baseline of the two rear wheels, m
% e---distance from the center of mass to the tool
%     operation point, m
```

```
% r---tire radius, m

Kd=zeros(2,2);
Kd(1,1)=0.96;
Kd(2,2)=0.96;

Kp=zeros(2,2);
Kp(1,1) =0.16;
Kp(2,2) =0.16;

O=zeros(2,2);
I=zeros(2,2);
I(1,1)=1;
I(2,2)=1;

Q=[I O;O I];
B=[O O;O O];
A=[O I;-Kp -Kd];
P=are(A,B,Q)
BI=[O;I];
global pai g m Izz Itt mu cx cy a b c d e r roc Bt Kp Kd P BI;

tmax=4;
t0=0;tf=tmax*2*pai;

%initial conditions
%x0=[0 0 1.1071 0 0 0 0 0]';
x0=[0 0 0.7854 0 0 0 0 0]';
%x0=[0 4 0 0 0 0 0 0]';
%x0=[0 0 0 0 0 0 0 0]';
%x0=[0 0 0.4636 0 0 0 0 0]';

uc=[0 0]';

h=0.1;
im=(tf-t0)/h;
tc=t0;
xc=x0;
t(1)=t0;
x(1,:)=x0';
for i=2:im+1
   xx=xc;
   tt=tc;
    [tc,xc]=rk42_dyn(tt,xx,h,8);
   t(i)=tc;
    x(i,:)=xc';
   uc=control(tc,xc);
    ucc(i,:)=uc';
end

%calculating reference path
for i=0:tmax*20*pai;
    ti=0.1*i;
    tf(i+1)=ti;
```

```
    xff=refe_path(ti);
     xxf(i+1,:)=xff(:,1)';
end;

%calculating slippage
vslip=x(:,5)-b*x(:,6);
uleft=x(:,4)+(d/2)*x(:,6);
uright=x(:,4)-(d/2)*x(:,6);
ulslip=r*x(:,7)-uleft;
urslip=r*x(:,8)-uright;

%draw reference path and motion path
subplot(221),plot(x(:,2),x(:,1),xxf(:,2),xxf(:,1))
xlabel('y');
ylabel('x');
grid;

%draw orientation trajectory
subplot(222),plot(t(:),x(:,3),tf,xxf(:,3))
xlabel('t,sec');
ylabel('orientation,radian');
grid;

%draw x(t) and y(t)
%subplot(223),plot(t(:),x(:,1),tf,xxf(:,1))
%xlabel('t');
%ylabel('x');
%grid;
%subplot(224),plot(t(:),x(:,2),tf,xxf(:,2))
%ylabel('y');
%xlabel('t');
%grid;

%draw longitudinal and lateral slip
%subplot(223),plot(t(:),ulslip,t(:),urslip);
%xlabel('time,second');
%ylabel('longitudinal slip,m/s');
%grid;
%subplot(224),plot(t(:),vslip);
%xlabel('time,second');
%ylabel('lateral slip,m/s');
%grid;

% draw control inputs
subplot(223),plot(t(:),ucc(:,1));
xlabel('t,sec');
ylabel('input 1');
grid;
subplot(224),plot(t(:),ucc(:,2));
xlabel('t,sec');
ylabel('input 2');
grid;
```

```
%============================================================
%rk4:function of Runge-Kutta method
function [t,x]=rk42_dyn(t0,x0,h,n)
x=zeros(n,1);
xt=zeros(n,1);
h2=h/2;
h6=h/6;
dxdt=diff_dyn(t0,x0);
th=t0+h2;
xt=x0+h2*dxdt;
dxt=diff_dyn(th,xt);
xt=x0+h2*dxt;
dxm=diff_dyn(th,xt);
xt=x0+h*dxm;
t=t0+h;
dxm=dxt+dxm;
dxt=diff_dyn(t,xt);
x=x0+h6*(dxdt+dxt+2.0*dxm);


%============================================================
% function qdot=diff_dyn(t,q)
% t:time
% q(8):state variables (x,y,fei,u,v,w_yaw,w_left,w_right)
function qdot=diff_dyn(t,q)

qdot=zeros(8,1);
Fx=[0 0 0]';
Fy=[0 0 0]';
Ft=[0 0 0]';
Tw=[0 0]';
% Ft:tool operation force (Fxt,Fyt,Fnt), Fnt=-Fzt
% Tw:torque applied to wheels (Tw_left,Tw_right)
% Fx:tire forces in x direction (Fxl,Fxr,Fxc)
% Fy:tire forces in y direction (Fyl,Fyr,Fyc)

fei=q(3);
u=q(4);
v=q(5);
w_yaw=q(6);
w_left=q(7);
w_right=q(8);

Ft=toolforce(t);
Tw=control(t,q);

Fnt=Ft(3);
T_left=Tw(1);
T_right=Tw(2);

Fx=[0 0 0]';
Fy=[0 0 0]';

Fnc=(b*m*g+(e-b)*Fnt)/(a+b);
```

```
Fn=(a*m*g-(a+e)*Fnt)/(a+b)/2;

u_left=u+w_yaw*d/2;
u_right=u-w_yaw*d/2;
v_slip=v-w_yaw*b;

u_c=u;
v_c=v+a*w_yaw;

[Fx(1),Fy(1)]=tiredriven(u_left,v_slip,w_left,T_left,Fn);
[Fx(2),Fy(2)]=tiredriven(u_right,v_slip,w_right,T_right,Fn);
[Fx(3),Fy(3)]=castor(u_c,v_c,Fnc);

udot=(Fx(1)+Fx(2)+Fx(3)+Ft(1))/m+v*w_yaw;
vdot=(Fy(1)+Fy(2)+Fy(3)+Ft(2))/m-u*w_yaw;
wdot=(0.5*d*(Fx(1)-Fx(2))-b*(Fy(1)+Fy(2))+a*Fy(3)-e*Ft(2))/Izz;
xdot=cos(fei)*u-sin(fei)*v;
ydot=sin(fei)*u+cos(fei)*v;

w_ldot=(Tw(1)-Bt*w_left-r*Fx(1))/Itt;
w_rdot=(Tw(2)-Bt*w_right-r*Fx(2))/Itt;

qdot(1)=xdot;
qdot(2)=ydot;
qdot(3)=w_yaw;
qdot(4)=udot;
qdot(5)=vdot;
qdot(6)=wdot;
qdot(7)=w_ldot;
qdot(8)=w_rdot;


%=========================================================
function Ft=toolforce(t)
% function for calculating the tool operational forces
% Ft=(Fxt,Fyt,Fnt)', Fnt=-Fzt
Ft=[0 0 0]';
if t<=10,
   Ft(1)=0;
   Ft(2)=0;
   Ft(3)=0;
else
   Ft(1)=-200;
   Ft(2)=0;
   Ft(3)=0;
end;


%=========================================================
%function [Fx_c,Fy_c]=castor(u_c,v_c,Fnc)
% for the caculation of castor resistance force
% u_c:longitudinal speed of the mobile robot
% v_c:lateral speed of the castor
function [Fx_c,Fy_c]=castor(u_c,v_c,Fnc)
```

```
vv=sqrt(u_c*u_c+v_c*v_c);
if vv==0,
  Fx_c=0;
  Fy_c=0;
else
  udir=u_c/vv;
  vdir=v_c/vv;
  Fx_c=-roc*udir*Fnc;
  Fy_c=-roc*vdir*Fnc;
end


%=============================================================
% function for calculation driven wheel tire friction model
% u_w:longitudianl speed of wheel axis
% u_w:lateral speed of wheel axis
% w_w:rotational angular speed of wheel
% T_w:torque applied on wheel
% Fn:normal speed apllied on wheel
% return in [Fx_w,Fy_w]

function [Fx_w,Fy_w]=tiredriven(u_w,v_w,w_w,T_w,Fn)
if u_w==0,
   if w_w==0,
      Fx_w=T_w/r;
   else
      Fx_w=sign(w_w)*mu*Fn;
   end;
   if v_w==0,
      Fy_w=0;
   else
      Fy_w= -sign(v_w)*mu*Fn;
   end;
else
   lamd=abs(v_w/u_w);
   if w_w==0,
      slip=1;
   else
      if w_w*u_w>0,
         slip=1-r*w_w/u_w;
         if slip<=-3,
            slip=-3;
         end;
      else
         slip=1;
      end;
   end;
   clng=0.001*cx*Fn;
   clat=0.001*cy*Fn;
   scom=sqrt(slip*slip+lamd*lamd);
   mudyn=mu*(1-0.0034*abs(u_w)*scom);
   if mudyn<=0.7*mu,
     mudyn=0.7*mu;
```

```
    end;
    if slip==1,
       ccll=sqrt(clng*clng+clat*clat*lamd*lamd);
       if ccll<=0.000001,
          ccll=0.000001;
       end;
       muf=mudyn*Fn;
       Fx_w=-sign(u_w)*clng*muf/ccll;
       Fy_w=-sign(v_w)*clat*muf*lamd/ccll;
    else
       Fxd=-sign(u_w)*clng*slip/(1-slip);
       Fyd=-sign(v_w)*clat*lamd/(1-slip);
       mud=sqrt(Fxd*Fxd+Fyd*Fyd)/Fn;
       if mud<=mudyn/2,
          Fx_w=Fxd;
          Fy_w=Fyd;
       else
          mures=mudyn*(1-mudyn/mud/4);
          Fx_w=Fxd*(mures/mud);
          Fy_w=Fyd*(mures/mud);
       end;
    end;
end


%============================================================
% function for simulating the controller
% t:time
% q:state variables
% Tw=[T_left, T_right]' torque applied to wheels

function Tw=control(t,q)
dz=[0 0]';
Tw=[0 0]';
ff=[0 0]';
z=[0 0]';
ed=[0 0]';
ep=[0 0]';
rqdot2=[0 0]';
rqdot=[0 0]';
rq=[0 0]';
oqp=[0 0]';
oqdot=[0 0]';
qf=zeros(3,3);
Gp=zeros(2,2);
D1=m*r*r+2*Itt;
D2=2*r*r*Izz+2*m*b*b*r*r+d*d*Itt;
fei=q(3);
u=q(4);
v=q(5);
w=q(6);
cosfei=cos(fei);
sinfei=sin(fei);
rD1=r/D1;
```

```
rbdD2=r*b*d/D2;
Gp(1,1)=rD1*cosfei-rbdD2*sinfei;
Gp(1,2)=rD1*cosfei+rbdD2*sinfei;
Gp(2,1)=rD1*sinfei+rbdD2*cosfei;
Gp(2,2)=rD1*sinfei-rbdD2*cosfei;
IttD1=2*Itt/D1;
IID2=(2*r*r*Izz+d*d*Itt)/D2;
ff(1)=-IttD1*v*w*cosfei-IID2*u*w*sinfei;
ff(2)=-IttD1*v*w*sinfei+IID2*u*w*cosfei;

oqdot(1)=u*cosfei-b*w*sinfei;
oqdot(2)=u*sinfei+b*w*cosfei;
oqp(1)=q(1);
oqp(2)=q(2);

qf=refe_path(t);
rqdot2(1)=qf(1,3);
rqdot2(2)=qf(2,3);
rqdot(1)=qf(1,2);
rqdot(2)=qf(2,2);
rq(1)=qf(1,1);
rq(2)=qf(2,1);

ed=oqdot-rqdot;
ep=oqp-rq;

%variable structure algorithm
dz=sliding(ed,ep);
z=rqdot2+dz;

Tw=inv(Gp)*(z-ff);


%============================================================
function dz=sliding(ed,ep)
dz=[0 0]';

LL=zeros(2,2);
LL(1,1)=0.4;
LL(2,2)=0.4;

SS=LL*ep+ed;
CAE=LL*ed;

ss=SS'*SS;

vss=sqrt(ss);

if vss>0.1,
   dd=2.5*SS/vss;
else
   dd=2.5*SS/0.1;
end
```

```
dz=-CAE-dd;


%=========================================================
% refe:function for caculating the
%         control reference;
% xf(i,1):position and orientation of
%        reference path;
% xf(i,2):derivatives of reference path;
%
function xf=refe_on(t)
xf=zeros(3,3);

sint4=sin(t/4);
cost4=cos(t/4);
cs=1+cost4;
cs2=1+cs*cs;

xf(1,1)=t;
xf(2,1)=t+4*sint4;
xf(3,1)=atan(cs);

xf(1,2)=1;
xf(2,2)=1+cost4;
xf(3,2)=-sint4/4/cs2;

xf(1,3)=0;
xf(2,3)=-(1/4)*sint4;
xf(3,3)=-(1/16)*(cost4*cs2+2*sint4*sint4*cost4)/cs2/cs2;

%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```