A Study on an Obstacle Avoidance Mobile Robot Using a Ceiling-mounted Camera System

천정설치형 카메라 시스템을 사용한 이동로봇의 장해물 회피에 관한 연구

by

[illegible] Kim Tran

Advisor: Professor Sang Bong Kim

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of

Master of Engineering

To the Department of Mechatronics Engineering, Graduate School,
Pukyong National University

August 2004

A Study on an Obstacle Avoidance Mobile Robot Using a Ceiling-mounted Camera System

A Dissertation
by

Anh Kim Tran

Approved of styles and contents by:

Chairman **Prof. Young Bok Kim**

Member **Prof. Yeon Wook Choe**

Member **Prof. Sang Bong Kim**

August 2004

Acknowledgements

It is a pleasure to express my gratefulness to all who have been witnesses to my persistent endeavor for a long time studying and living in Pusan, Korea. The first special thanks must go to Professor Sang Bong Kim who has been an active advisor Professor and a honorable sponsor since my first days studying and living in Pukyong National University. So far, there have been my unaccountable chances to perceive his valuable rich knowledge as well as his living experiences. Like other students who had been under his advice, I would like to wish my Professor and his family to have the long-lived health and happiness. Also, on behalf of Vietnamese students, I like to express our great thankfulness for his kindly supports and hope that his generous mind would greatly contribute to a closer brothership between the two countries. I would like to thank the members of my thesis committee: Professor Young Bok Kim and Professor Yeon Wook Choc, for the helpful comments and suggestions.

It would be a great mistake if I forget to recall my Professors and Lecturers in Ho Chi Minh University of Technology who had been giving me the first lessons of science as well as the lessons of life. Prof. Tan Tien Nguyen is the first one who had brought me up by showing his nice enthusiastic mind toward students. Prof. Huu Phuc Nguyen and Prof. Hoai Quoc Le who had written the introductory letters for my studying here. Prof. Tan Tung Phan and Prof. Thien Phuc Nguyen are great teachers who have been lastingly giving me valuable experiences and solutions whenever I got stuck. Phd. Trong Hieu Bui, my old classmate, is the one who helped me to have connection with my current advisor Professor and guided me to be adapted to a new life in Korea.

Dear brother, Mr. Tan Lam Chung, you have been working so hard, which makes a great contribution to the completion of this thesis, especially for the experiment part. His kindness makes me be positive as well. Truthfully, he is a bearer of my debt and I really appreciate what he has thought and done for me. I would like to express millions thanks to Phd. Hak Kyeong Kim who has been with me a lot of time and gives me the solution of my difficulties with his rich knowledge, rigorousness and generosity. I would like to say thanks all other Korean friends, Mr. SoungJea Park, Mr. JeaSung Im, Mr. Young-Gyu Kim, Mr. Sangch'an Kim… or to Mr. Hurchul, a Chinese Ph.d fellow, and Mr. Dung for all of their supports during my time here and for all the pleasures that we had been enjoyed altogether.

Dear family, my mom, Nhat, Mi, and my sweetheart, Ms. Dung! You all have been experienced the hardest time without me by your side. It was also the hardest time of mine since the first time in my life, I had to stay far away from my lovely ones and miss you a lot. Now I have made a giant leap and I know your great moral support and guidance will give me more strength to go on along my chosen route.

Last but not least, it is my favor to say thanks to all of my Vietnamese friends who are studying in Pukyong National University, friends in Pusan Foreign University, Pusan Maritime National University, Pusan National University, many friends in other places around Korea, and all of my other friends in Vietnam as well as in Korea. All of you had ever brought me up with wonderful pleasure time and moments. All those things are unforgettable and would be invaluable gifts in my life…

Pusan, June 16, 2004

Table of Contents

A Study on an Obstacle Avoidance Mobile Robot Using a Ceiling-mounted Camera System

천정설치형 카메라 시스템을 사용한 이동로봇의 장해물 회피에 관한 연구

Anh Kim Tran

Department of Mechatronics Engineering
Pukyong National University
Graduate School

Abstract

An implementation of a nonholonomic mobile robot (MR) control in an unstructured environment is addressed in this paper. Generally, the MR needs a reference path before starting its motion. Therefore, a path planning method is needed to generate a collision-free path, says, a tracking trajectory. Two of the path planning algorithms, namely, HJB-based algorithm and polynomial path planning approach. are introduced. A control system that is composed of software and hardware is designed to drive the MR to follow the trajectory. The work space is observed by a ceiling-mounted USB camera, calibrated in advance, as a part of a computer vision system. By using this system, the environment information is initially obtained by processing captured images of the workspace. By using the image processing techniques, the obstacles' positions are extracted from image features. This information is indispensable for the path planning process. As the result, a path, which is generated through a path planning algorithm, plays the role

as a flexible configuration for the MR to reach the given target point. In addition, through the image processing process, the positional information of the MR is obtained and updated frequently. Thus, the MR is able to get the useful inputs for its nonlinear backstepping tracking controller. The whole control system is realized by integrating a personal computer (Intel Pentium IV, 2.8Ghz, 1GB Ram), a USB camera and PIC-based microprocessors using wireless communication between high level and low level controllers: the computer vision system, composed of the USB camera and the computer, is in charge of coordinate tracking, path planning and control law computing serves as high level control while the device control, known as the PIC-based microprocessor control, stands for low level PIC microprocessor control.

1

Introduction

1.1 Background, Motivation and Problem Statement

A two-wheeled mobile robot (MR) is suitable for a variety of application in unstructured environment where a high degree of autonomy is required. The desired autonomous or intelligent behavior of the MR has motivated an intensive research in the last decade. To control the MR motion, several researches have been reported to solve the motion problem under nonholonomic constraints using the kinematics model of the MR[1-4]. The main objective of the researches is to construct a control law in which the velocity control inputs are able to stabilize the close-loop system[5, 6]. All these researches considered only the kinematic model of the MR, and it is assumed that the actual vehicle control inputs can generate a "perfect velocity tracking"[7]. On the other hand, a few later literatures have been devoted to the problem of integration of the nonholonomic kinematic controller and the dynamics of the platform[8-11]. That is, a dynamical extension is necessary for a good controller. Fierro *et al.* proposed a combined kinematic/torque control law using backstepping control approach[11]. This general control structure generates other control techniques. A torque-computed controller is generated when the system parameters of MR's model are known exactly. An adaptive controller is produced when the system parameters of the MR's model are uncertainly known[8-10]. Nowadays. the controllers such as adaptive or robust ones are used often for industrial fields because controllers can control the MR effectively and keep its stability even in the presence of external disturbances or the perturbation of system parameters. For all of these controllers, the Lyapunov theory is used to guarantee the asymptotical stability of the control system.

There are evidences for which the MR needs a good navigation capability to win a successful performance. A motion controller, therefore, exhibits a navigation capability when the MR is able to successfully obtain a certain desired objective. For instance, the desired objective is classified into three typical tasks: trajectory tracking, path following and point stabilization[12]. Besides, the MR needs other capabilities in order to meet a high autonomy requirement for a robotic vehicle. One related example is the navigation of the MR in an environment with obstacles. In this application, the MR has to find its own way from an initial point to a given target. This process is called path planning or trajectory planning

The path planning is one of the standard tasks in the robot navigation and numerous works on it have been reported[13-17]. An innovative path planning approach is necessary to generate a good, reasonable path for the MR. There are some works considering path optimality[14, 18, 19], e.g. the HJB-based path planning method generates the shortest trajectory from an initial point to the target[14]. The other works concentrate path characteristics, such as smoothness or continuousness, which facilitate a convenient path tracking (or path following) controller. A smooth polynomial path can be chosen for the MR's trajectory tracking[13]. Hence, to get the appropriate path depending on actual requirements, a planning approach must be developed.

In the vision-based navigation work reported in the past, obstacle avoidance was carried out using ultrasonic and camera sensors[20-23]. Ultrasonic sensor-based methods take over the control of the robot as long as obstacles are detected in the vicinity. The control is then handled back to vision-based processing once the obstacles are no longer a factor. While it is expedient to use ultrasonic sensors in such a manner, where the ultrasonic sensor-based and the vision-based navigation controls are alternated according to actual situation, it is empirically more interesting to use merely vision-based processing for obstacle avoidance. This can be supported by several of reliable vision systems developed recently[24-28]. At the matter of fact, a good computer vision system can extract accurately image features from a captured image. These image features are then processed to obtain useful information. In a previous work of *Dixon et al.*, by using an un-calibrated computer vision system, a mobile robot was controlled by applying work-space-to-camera-space transformation approach. That is, the control closed-loop is closed in the camera-space instead of the work-space. Nevertheless, it is not straightforward to handle this method due to the implementation of a complicated transformation between the work-space and the camera-space in order to control the MR. Besides, the work-space position of the MR is unknown, rather, only pixel coordinates in the camera-space are known.

In fact, it is very difficult to obtain an instantaneous Cartesian position of a MR in its work-space by using the computer vision system. Fortunately, this can be solved by developing a mapping technique to a computer vision system with a single CCD camera. Specifically, in this paper, a single USB camera is applied to the system with prior calibration steps. Then, a pin-hole lens model is applied to the camera system to express the transformation from the work-space to the camera-space. An inverse mapping technique is developed to determine the Cartesian coordinates of an interesting point in the work-space.

1.2 Contribution to the problem and scope of application

This paper is presented specifically to solve the obstacle avoidance problem of a wheeled mobile robot. Our aim is to incorporate path planning and tracking controller of the MR. The concept of this work appears partly in a few recent literatures[29-31], and there are still remained some interesting technical problems, such as localization techniques or navigation controllers of the MR, that are needed to be realized.

This paper's contents especially support to the use of a computer vision system in order to facilitate the MR control. This paper also employs several path planning approaches, e.g. HJB-based and polynomial path planning approach, in order to generate a reference trajectory for the MR. There are some unreal assumptions in the paper: actual size of the MR is larger than that used in the simulation and zero obstacle height is assumed. However, the outcomes of the experimental realization demonstrate the effectiveness of the control method.

In this paper, it is assumed that all the obstacles are located at fixed positions in the screen range of a ceiling-mounted USB camera in a computer vision system. By using this system, the environment information is initially obtained by processing a captured image of the workspace. The obstacles' positions are obtained from the image features of the obstacles. Also, the instantaneous Cartesian coordinate of the MR is able to be obtained by processing similarly to the image features attached to the MR. As the result, a path, which is generated by the path planning method, plays the role as a flexible reference trajectory for the MR to reach the given target point. In addition, the visioning system is used for frequently updating MR's position and orientation that are needed by the nonlinear backstepping trajectory tracking controller.

The approach to the computer vision system has remarkably facilitated the good navigation of the MR. It should be noted that all the necessary parameters for the vision system are obtained through a calibration process. In this process, a pin-hole lens model is applied to the USB camera in order to formulate the transformation between work-space and camera-space. Specifically, an inverse mapping technique is developed to determine the Cartesian coordinates of an image point in the space.

In order to realize the *nonlinear backstepping controller* that is designed to track a reference path[11], design and development for the hardware and software are indispensable tasks. The developed control system is based on the integration of a personal computer (Intel Pentium IV, 2.8GHz, 1GB Ram), a USB camera and PIC-based microprocessors using wireless communication between high level and low level controllers: the high level control which consists of the USB camera and the computer, known as the computer vision system, is in charge of coordinate tracking, path planning and control law computing, while the device control which is known as the PIC-based microprocessor control stands for low level PIC microprocessor control.

To drive the MR tracking the reference trajectory, the PIC-based servo controller is developed. This controller is composed of two modules that communicate with each other via I2C. One module acts as a master while another acts as a slave. The master microprocessor, which receives the signal from the central computer via Bluetooth wireless communication module Promi SD202, directs the control commands to the slave microprocessors to drive the left and right motors. The motors are driven via LMD18200 dual full-bridge driver. The simulation and experimental results show the good tracking control performance and the effectiveness of the proposed method.

1.3 Summary Outline

This thesis consists of nine chapters. The content in each chapter are summarized as follows:

- ***Chapter 1: Introduction***

  A complete statement of the problem, the background and reasons for conducting the study, the contribution of this research, and the summary outline of contents of this thesis are introduced.

- ***Chapter 2: A Non-holonomic Wheeled Mobile Robot***

  A kinematic and dynamic model of a two-wheeled mobile robot are presented.

- ***Chapter 3: Ceiling-mounted Camera System***

  A powerful means of visual localization is integrated into a computer vision system with a ceiling-mounted camera. In this system, a single CCD camera is employed to capture images of the work space. The images are then processed to obtain useful information such as obstacles positions, mobile robot position and orientation, etc. Also, the computer vision systems with calibration and mapping approaches are introduced.

- ***Chapter 4: Obstacle Avoidance Path Planning Approaches***

  Two of the path planning algorithms, namely, the Hamilton-Jacobi-Belman Equation based algorithm and the polynomial path planning approach, are presented. Simulations are shown for the two methods.

- ***Chapter 5: Trajectory Tracking Controller Design***

  A nonlinear trajectory tracking controller is designed based on the backstepping control design for kinematic/dynamic model of the MR.

- ***Chapter 6: Hardware Design and Implementation***

  A USB camera system and the PIC-based mobile robot controller is presented. The experiment method is introduced.

- ***Chapter 7: Simulation, Experimental Results, and Discussions***

  Show and discuss the simulation and experimental results. Also, the conclusions are given.

- ***Chapter 8: Discussion and Conclusions***

  Summarize the results of this thesis.

- ***Chapter 9: Appendices***

2

A Nonholonomic Mobile Robot

In this chapter, the kinematics and dynamics models of a two-wheeled mobile robot under the nonholonomic constraints are constructed, step by step. Using this model, the kinematic and dynamic controllers are able to be derived. In addition, the mobile robot control application in an unstructured environment is dug up through concise analyses.

2.1 Wheeled Mobile Robot and Its Application

A two-wheeled mobile robot is suitable for a variety of application in unstructured environments where a high degree of autonomy is required. The desired autonomous or intelligent behavior of the MR has motivated an intensive research in the last decade. A specific application to the obstacle avoidance mobile robot has been raised much concern recently. That is, a mobile robot must be able to avoid both static and dynamic obstacles in order to get a target. This essential task often relies on navigation means such as ultrasonic or computer visions. The vision systems are passive and can provide lateral and depth resolution exceeding that of ultrasonic devices[32]. For a computer vision system equipped with a CCD camera, some calibration steps must be carried out prior to operation in order to successfully observe the work space[33]. Consequently, a mobile robot can have its path planned and tracked basing on the information obtained by the navigation devices simultaneously. The problem of path planning and computer vision will be discussed consecutively in the followed chapters. First, let us discuss about the modeling of a wheeled mobile robot.

Much research effort has been oriented to solving the problem of motion under nonholonomic constraints using the kinematic model of a mobile robot. All the controllers for kinematic model assume to have a "perfect velocity tracking". That is, velocity control inputs are proposed to stabilize the close-loop system. In practice one almost always controls forces and torques that actuaters should exert instead of velocities. Therefore, in the next section, a complete system which consists of both kinematic and dynamic models will be described.

2.2 Modeling of a Wheeled Mobile Robot

2.2.1. Kinematic Equations

The mobile robot with two actuated wheels is depicted in Fig. 2-1. $2b$ is the width of the mobile robot and r is the radius of the wheel. O-XY is the world coordinate system and C-xy is the coordinate system fixed to the mobile robot. Let us fix some notations:

m_c: mass of the vehicle without the wheels.

m_w: mass of a wheel.

d: displacement from the point C to the mass center of the mobile robot, which is assumed to be on the axis of symmetry.

I_c: moment of inertia of the vehicle without the driving wheels and motor rotors about vertical axis passing through the point C.

I_w: moment of inertia of each driving wheel and the motor rotor about the wheel axis.

I_m: moment of inertia of the wheel about a wheel diameter.

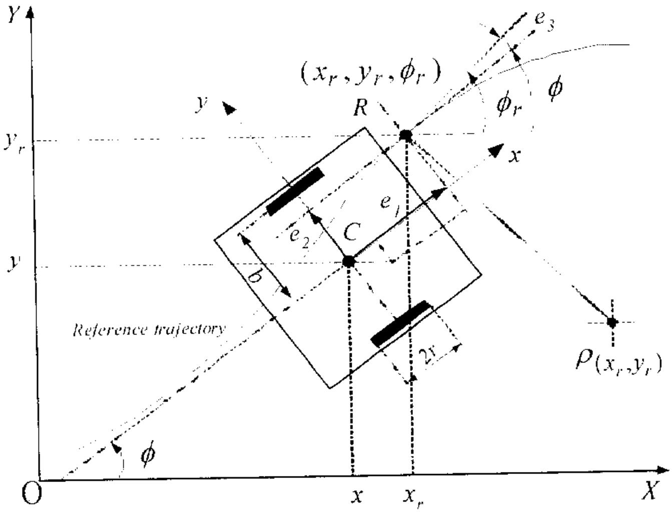


Fig. 2-1 Mobile robot with two actuated wheels

The configuration of the mobile robot can be described by five generalized coordinates[11, 34-36].

$$\mathbf{q} = [x, y, \phi, \theta_r, \theta_l]^T = [\mathbf{q}_1^T \quad \theta_r \quad \theta_l]^T \tag{2.1}$$

where (x, y) are the coordinates of C in the world coordinate system; ϕ is the heading angle of the mobile robot; and θ_r and θ_l are the angles of the right and left driving wheels. $\mathbf{q}_1^T = [x, y, \phi]$ is a sub-generalized coordinate when only three states x、y、ϕ except θ_r、θ_l are concerned. The kinematics of the system can be written as follows:

$$\dot{\mathbf{q}}_1 = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = \mathbf{S}_1(\mathbf{q}_1)\mathbf{v} \tag{2.2}$$

with

$$\mathbf{S}_1(\mathbf{q}_1) = \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix}, \ \mathbf{v} = \begin{bmatrix} \mathrm{v} \\ \omega \end{bmatrix}$$

where v, ω are tangential and angular velocity of the MR's center, the point C. $\mathbf{v}$ is considered as control input of the kinematic model (2.2).

In fact, it is intuitive to regard the control inputs as the angular velocities of left and right wheels of the MR, denoted by v_l and v_r respectively. The relationship between $[v, \omega]^T$ and $[v_r, v_l]^T$ is the following:

$$\begin{bmatrix} v_r \\ v_l \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{b}{r} \\ \frac{1}{r} & -\frac{b}{r} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{2.3}$$

Therefore, with a given control command $[v, \omega]^T$, one can obtain the direct control command to the right and left wheels by using Eq. (2.3).

From kinematic system (2.2), the system must be derived from a non-holonomic constraint:

$$-\dot{x}\sin\phi + \dot{y}\cos\phi = 0 \tag{2.4}$$

which states that the vehicle can not move in the direction perpendicular to the symmetric axis of the vehicle directly, that is, the velocity of C must be in the direction of the axis x. Furthermore, Eq. (2.3) is resulted from the assumption that the wheels roll and do not slip, expressed as follows

$$\dot{x}\cos\phi + \dot{y}\sin\phi + b\dot{\phi} = r\dot{\theta}_r \tag{2.5}$$

$$\dot{x}\cos\phi + \dot{y}\sin\phi - b\dot{\phi} = r\dot{\theta}_l \tag{2.6}$$

Subtracting and adding of Eqs. (2.5) and (2.6) yield

$$2b\dot{\phi} = r\left(\dot{\theta}_r - \dot{\theta}_l\right) \tag{2.7}$$

$$\dot{x}\cos\phi + \dot{y}\sin\phi = \frac{r}{2}\left(\dot{\theta}_r + \dot{\theta}_l\right) \tag{2.8}$$

It is clear that Eq. (2.7) can be integrated with appropriate initial conditions. As the result, it is guaranteed that ϕ depends on θ_r and θ_l as follows

$$\phi = \frac{r}{2b}\left(\theta_r - \theta_l\right) \tag{2.9}$$

Therefore, ϕ can be eliminated from the generalized coordinates. Without loss of generality, it is shown that $\mathbf{q} = \left[x, y, \theta_r, \theta_l\right]^T$. It turns out that Eq. (2.8) is another non-holonimic constraint since it cannot be integrated to obtain an explicit relationship among coordinates. The two non-holonomic constraints (2.4) and (2.8) are expressed in matrix form as follow

$$\mathbf{A}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0} \tag{2.10}$$

where $$\mathbf{A}(\mathbf{q})=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{bmatrix}=\begin{bmatrix} -\sin\phi & \cos\phi & 0 & 0 \\ -\cos\phi & -\sin\phi & \frac{r}{2} & \frac{r}{2} \end{bmatrix} \tag{2.11}$$

Then, the state space realization of a non-holonomic mobile robot can be established

$$\dot{\mathbf{q}}=\mathbf{S}(\mathbf{q})\boldsymbol{\eta} \tag{2.12}$$

where $\boldsymbol{\eta}=\left[\dot{\theta}_r,\dot{\theta}_l\right]^T=\left[v_r,v_l\right]^T$, $\mathbf{S}(\mathbf{q})\in R^{4\times 2}$ is a matrix whose columns are in the null space of $\mathbf{A}(\mathbf{q})$, *i.e.*, $\mathbf{A}(\mathbf{q})\mathbf{S}(\mathbf{q})=\mathbf{0}$. The specific choice of $\mathbf{S}(\mathbf{q})$ is as follow

$$\mathbf{S}(\mathbf{q})=\left[\mathbf{s}_1(\mathbf{q}) \quad \mathbf{s}_2(\mathbf{q})\right]=\begin{bmatrix} \frac{r}{2}\cos\phi & \frac{r}{2}\cos\phi \\ \frac{r}{2}\sin\phi & \frac{r}{2}\sin\phi \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.13}$$

Hence, the kinematic model of a mobile robot can be expressed either by Eq.(2.2) or Eq. (2.12) since Eq. (2.2) can be obtained by substituting Eq. (2.3) into Eq. (2.12) and using constraint (2.7) to derive $\dot{\phi}$ from $\dot{\theta}_r$ and $\dot{\theta}_l$.

2.2.2 Dynamic Equations

The Lagrangian equation is used to establish equations of motion for the mobile robot. The total kinetic energy of the mobile platform and the two wheels is

$$K = \frac{1}{2} m\left(\dot{x}^2 + \dot{y}^2\right) + m_c d\dot{\phi}\left(\dot{y}\cos\phi - \dot{x}\sin\phi\right) + \frac{1}{2} I\dot{\phi}^2 + \frac{1}{2} I_w\left(\dot{\theta}_r^2 + \dot{\theta}_l^2\right) \quad (2.14)$$

where $$m = m_c + 2m_w \quad (2.15)$$

$$I = I_c + 2m_w b^2 + 2I_m \quad (2.16)$$

Lagrangian equations of motion for the non-holonomic mobile robot system are governed by M. R. Reinhardt[37]

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{q}_i}\right) - \frac{\partial K}{\partial q_i} = \tau_i - a_{1i}\lambda_1 - a_{2i}\lambda_2, \qquad i = 1,\ldots,4 \quad (2.17)$$

where q_i is the generalized coordinate, τ_i is the generalized force, a_{ij} is from the constraint Eq. (2.11), and λ_1, λ_2 are the Lagrangian multipliers. Substituting the total kinetic energy Eq. (2.11) into Eq. (2.12), the followings are obtained

$$m\ddot{x} - m_c d\left(\ddot{\phi}\sin\phi + \dot{\phi}^2\cos\phi\right) = \lambda_1\sin\phi + \lambda_2\cos\phi \quad (2.18)$$

$$m\ddot{y} + m_c d\left(\ddot{\phi}\cos\phi - \dot{\phi}^2\sin\phi\right) = -\lambda_1\cos\phi + \lambda_2\sin\phi \quad (2.19)$$

$$m_c d\frac{r}{2b}\left(\ddot{y}\cos\phi - \ddot{x}\sin\phi\right) + \left(I\frac{r^2}{4b^2} + I_w\right)\ddot{\theta}_r - I\frac{r^2}{4b^2}\ddot{\theta}_l = \tau_r - \frac{r}{2}\lambda_2 \quad (2.20)$$

$$-m_c d\frac{r}{2b}\left(\ddot{y}\cos\phi - \ddot{x}\sin\phi\right) - I\frac{r^2}{4b^2}\ddot{\theta}_r + \left(I\frac{r^2}{4b^2} + I_w\right)\ddot{\theta}_l = \tau_l - \frac{r}{2}\lambda_2 \quad (2.21)$$

where $\boldsymbol{\tau} = \begin{bmatrix} \tau_r & \tau_l \end{bmatrix}^T$ are the torques acting on the two wheels. The Eqs. (2.18)-(2.21) can be written in the matrix form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{V}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{E}(\mathbf{q})\boldsymbol{\tau} - \mathbf{A}^T(\mathbf{q})\boldsymbol{\lambda} \tag{2.22}$$

where $\mathbf{A}(\mathbf{q})$ is defined in Eq. (2.11) and

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} m & 0 & -m_c \dfrac{rd}{2b}\sin\phi & m_c \dfrac{rd}{2b}\sin\phi \\ 0 & m & m_c \dfrac{rd}{2b}\cos\phi & -m_c \dfrac{rd}{2b}\cos\phi \\ -m_c \dfrac{rd}{2b}\sin\phi & m_c \dfrac{rd}{2b}\cos\phi & I\dfrac{r^2}{4b^2} + I_w & -I\dfrac{r^2}{4b^2} \\ m_c \dfrac{rd}{2b}\sin\phi & -m_c \dfrac{rd}{2b}\cos\phi & -I\dfrac{r^2}{4b^2} & I\dfrac{r^2}{4b^2} + I_w \end{bmatrix} \tag{2.23}$$

$$\mathbf{V}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} -m_c d\dot{\phi}^2 \cos\phi \\ -m_c d\dot{\phi}^2 \sin\phi \\ 0 \\ 0 \end{bmatrix}; \ \mathbf{E}(\mathbf{q}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}; \ \boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} \tag{2.24}$$

Therefore the dynamic equations of the system is expressed into Eq. (2.22). However, it would be more convenient if the Lagrangian multipliers are eliminated from Eq. (2.22). Then the derivative of the Eq. (2.12) yields

$$\ddot{\mathbf{q}} = \mathbf{S}(\mathbf{q})\dot{\boldsymbol{\eta}} + \dot{\mathbf{S}}(\mathbf{q})\boldsymbol{\eta} \tag{2.25}$$

Now, by multiplying the both sides of Eq. (2.22) by $\mathbf{S}^T(\mathbf{q})$ and noticing that $\mathbf{S}^T(\mathbf{q})\mathbf{A}^T(\mathbf{q}) = \mathbf{0}$ and $\mathbf{S}^T(\mathbf{q})\mathbf{E}(\mathbf{q}) = \mathbf{I}_{2\times 2}$, the following is obtained

$$\mathbf{S}^{T}(\mathbf{q})\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{S}^{T}(\mathbf{q})\mathbf{V}(\mathbf{q},\dot{\mathbf{q}}) = \boldsymbol{\tau} \tag{2.26}$$

By substituting Eq. (2.25) into the above equation, the following is obtained

$$\mathbf{S}^{T}(\mathbf{q})\mathbf{M}(\mathbf{q})\mathbf{S}(\mathbf{q})\dot{\boldsymbol{\eta}} + \mathbf{S}^{T}(\mathbf{q})\mathbf{M}(\mathbf{q})\dot{\mathbf{S}}(\mathbf{q})\boldsymbol{\eta} + \mathbf{S}^{T}(\mathbf{q})\mathbf{V}(\mathbf{q},\dot{\mathbf{q}}) = \boldsymbol{\tau} \tag{2.27}$$

Adding the unmodeled parts and the estimated disturbances to Eq. (2.27), the following is obtained

$$\overline{\mathbf{M}}\dot{\boldsymbol{\eta}} + \overline{\mathbf{V}}(\dot{\mathbf{q}})\boldsymbol{\eta} + \overline{\mathbf{F}}(\boldsymbol{\eta}) + \bar{\boldsymbol{\tau}}_{d} = \boldsymbol{\tau} \tag{2.28}$$

where

$$\overline{\mathbf{V}}(\dot{\mathbf{q}}) = \begin{bmatrix} 0 & \dfrac{r^{2}}{2b} m_{c} d\dot{\phi} \\ -\dfrac{r^{2}}{2b} m_{c} d\dot{\phi} & 0 \end{bmatrix} \tag{2.29}$$

$$\overline{\mathbf{M}} = \mathbf{S}^{T}(\mathbf{q})\mathbf{M}(\mathbf{q})\mathbf{S}(\mathbf{q}) = \begin{bmatrix} \dfrac{r^{2}}{4b^{2}}\left(mb^{2} + I\right) + I_{w} & \dfrac{r^{2}}{4b^{2}}\left(mb^{2} - I\right) \\ \dfrac{r^{2}}{4b^{2}}\left(mb^{2} - I\right) & \dfrac{r^{2}}{4b^{2}}\left(mb^{2} + I\right) + I_{w} \end{bmatrix} \tag{2.30}$$

$\bar{\boldsymbol{\tau}}_{d}$ represents estimated disturbance with bounded value of unknown parameters. $\overline{\mathbf{F}}(\boldsymbol{\eta})$ represents friction vector against dynamics.

Therefore, the complete dynamics of the mobile robot consists of the kinematic steering system Eq. (2.2), extra dynamics Eq. (2.28) and Eq. (2.3).

3

Ceiling-mounted Camera System

In this chapter, the camera system is shown. The role of the system is to get the position information of the objects in the work space. This information is very useful for the controller of mobile robot to calculate the velocities or torques commands that are exerted on actuators. Also the locations of the obstacles can be obtained by the path planning algorithm. Before putting into operation, the camera needs to be calibrated in order to get familiar with the environment that it observes. Therefore, the content in the following sections describes features of a calibrated camera system, the calibration approaches and the mapping between a 3D Cartesian coordinates point and an image point. As the result, the OpenCV library is adopted as a calibration means for our implementation.

3.1 Calibrated Camera System

A camera system should needs certain calibrated parameters before putting into operation. There are two kinds of parameters, namely, extrinsic and intrinsic parameters. The calibrating process helps to determine a set of parameters that prescribe the mapping between 3-D reference coordinates and 2-D image coordinates. Several advanced methods for camera calibration are found from the literatures[38, 39]. The following descriptions will help a more insightful understanding of the system.

3.1.1 Notation

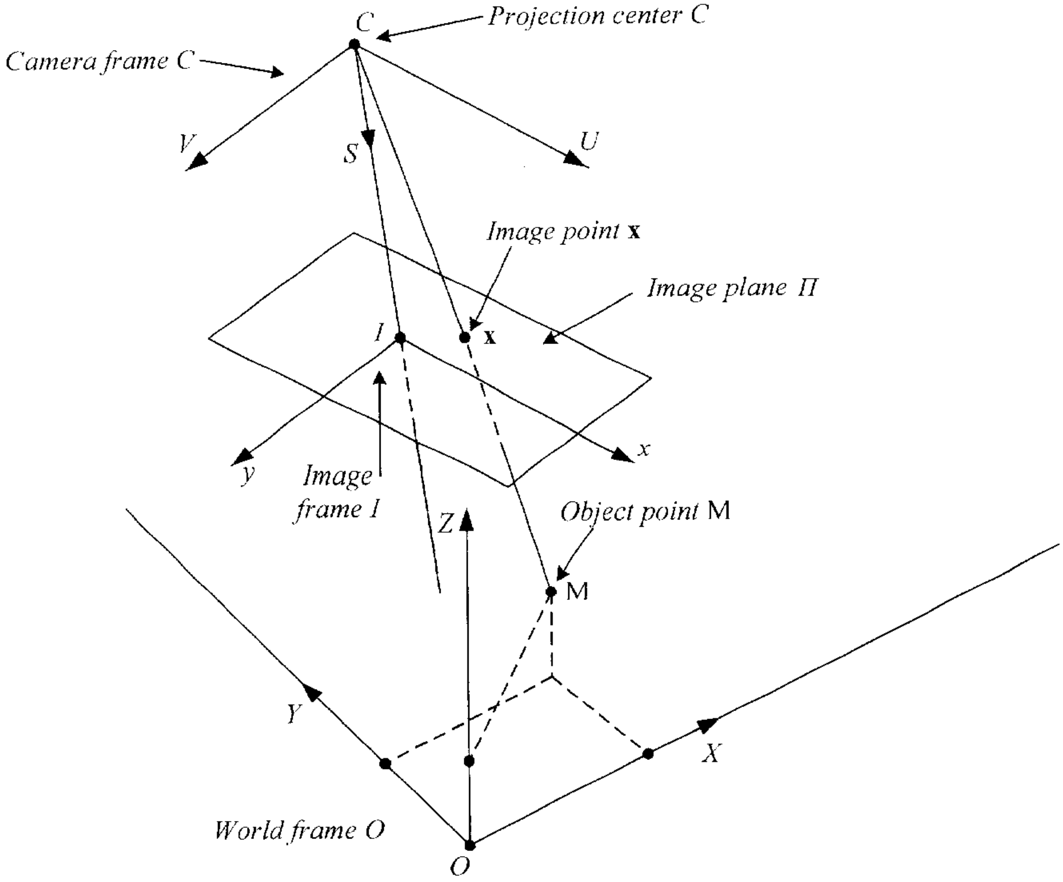


Fig. 3-1 Pinhole camera model

A 2D point is denoted by $\mathbf{m}=[u,v]^T$. A 3D point is denoted by $\mathrm{M}=[X,Y,Z]^T$. $\tilde{\mathbf{x}}$ is denoted as the augmented vector by adding 1 as the last element of an vector: $\tilde{\mathbf{m}}=[u,v,1]^T$ and $\tilde{\mathrm{M}}=[X,Y,Z,1]^T$. A camera is modeled by the usual pinhole: the relationship between a 3D point M and its image projection **m** is given by

$$s\tilde{\mathbf{m}}=\mathbf{A}[\mathbf{R}\quad\mathbf{t}]\tilde{\mathrm{M}} \tag{3.1}$$

where s is an arbitrary scale factor, $(\mathbf{R},\mathbf{t})$, called the extrinsic parameters, is the composition of a rotation matrix and a translation vector, which relates the world coordinate system to the camera coordinate system, and **A**, called the camera intrinsic matrix, is given by

$$\mathbf{A}=\begin{bmatrix}\alpha & \gamma & u_0\\ 0 & \beta & v_0\\ 0 & 0 & 1\end{bmatrix}$$

with (u_0, v_0) is the coordinates of the principal point, α and β are the scale factors in image u and v axes, and γ is the parameter describing the skewness of the two image axes.

3.1.2 Homography Between the Model Plane and Its Image

Without loss of generality, it is assumed that the model plane is on $Z = 0$ of the world coordinate system. Let's denote the i^{th} column of the rotation matrix **R** by r_i. From (3.1), the following is obtained

$$s\begin{bmatrix}u\\ v\\ 1\end{bmatrix}=\mathbf{A}[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3 \quad \mathbf{t}]\begin{bmatrix}X\\ Y\\ 0\\ 1\end{bmatrix}=\mathbf{A}[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}]\begin{bmatrix}X\\ Y\\ 1\end{bmatrix}$$

$\mathrm{M} = [X, Y]^T$ can be expressed since Z is always equal to 0. In turn, $\widetilde{\mathrm{M}} = [X, Y, 1]^T$. Therefore, a model point M and its image **m** is related by a homography **H**:

$$s\tilde{\mathbf{m}} = \mathbf{H}\tilde{\mathrm{M}} \text{ with } \mathbf{H} = \mathbf{A}[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}]. \tag{3.2}$$

Clearly, the 3x3 matrix **H** is defined up to a scale factor.

3.1.3 Constraints on the Intrinsic Parameters

Given an image of the model plane, an homography can be estimated (*Appendix A*). Let's denote it by $\mathbf{H} = [\mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3]$ From (3.2), the following is reduced

$$[\mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3] = \lambda\mathbf{A}[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}],$$

where λ is an arbitrary scalar. By the fact that $\mathbf{r}_1$ and $\mathbf{r}_2$ are orthonormal, the following are obtained

$$\mathbf{h}_1^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_2 = 0 \tag{3.3}$$

$$\mathbf{h}_1^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_1 = \mathbf{h}_2^T \mathbf{A}^{-T} \mathbf{A}^{-1} \mathbf{h}_2 \tag{3.4}$$

These are the two basic constraints on the instrinsic parameters, given one homography. Because a homography has 8 degrees of freedom and there are 6 extrinsic parameters (3 for rotation and 3 for translation), only 2 constraints on the intrinsic parameters are obtained. Note that $\mathbf{A}^{-T}\mathbf{A}^{-1}$ actually describes the image of the absolute conic[40].

3.1.4 Solving Camera Calibration

3.1.4.1 Closed-Form Solution

Let

$$\mathbf{B} = \mathbf{A}^{-T}\mathbf{A}^{-1} \equiv \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix}$$
$$= \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2\beta} & \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} \\ -\frac{\gamma}{\alpha^2\beta} & \frac{\gamma^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0\gamma - u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix} \quad (3.5)$$

Note that **B** is symmetric, defined by a 6D vector

$$\mathbf{b} = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T \quad (3.6)$$

Let the i^{th} column vector of **H** be $\mathbf{h}_i = [h_{i1}, h_{i2}, h_{i3}]^T$. Then, the following is satisfied

$$\mathbf{h}_i^T \mathbf{B} \mathbf{h}_j = \mathbf{v}_{ij}^T \mathbf{b} \quad (3.7)$$

with

$$\mathbf{v}_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^T$$

Therefore, the two fundamental constraints (3.3) and (3.4), from a given homography, can be rewritten as 2 homogeneous equations in **b**:

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ (\mathbf{v}_{11} - \mathbf{v}_{22})^T \end{bmatrix} \mathbf{b} = \mathbf{0} \qquad (3.8)$$

If n images of the model plane are observed, by stacking such n equations as (3.8), the combination is expressed as

$$\mathbf{Vb} = \mathbf{0} \qquad (3.9)$$

where **V** is a $2n$ X 6 matrix. If $n \geq 3$, then a unique solution of Eq. 3.9, **b**, is defined up to a scale factor. If $n = 2$, the skewless constraint $\gamma = 0$, i.e, $[0,1,0,0,0,0]\mathbf{b} = 0$, can be added as an additional equation to (3.9). If $n = 1$, only two camera intrinsic parameters, e.g., α and β, can be solved when u_0, v_0 are known (e.g., at the image center) and $\gamma = 0$. The solution of (3.9) is well known as the eigenvector of $\mathbf{V}^T\mathbf{V}$ associated with the smallest eigenvalue. Once **b** is estimated, all camera intrinsic matrix **A** can be computed. See *Appendix B* for the details.

Once **A** is known, the extrinsic parameter for each image is readily computed. From (3.2), the solution is given

$$\mathbf{r}_1 = \lambda \mathbf{A}^{-1}\mathbf{h}_1\,;\ \mathbf{r}_2 = \lambda \mathbf{A}^{-1}\mathbf{h}_2\,;\ \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2\,;\ \mathbf{t} = \lambda \mathbf{A}^{-1}\mathbf{h}_3$$

with $\lambda = 1/\|\mathbf{A}^{-1}\mathbf{h}_1\| = 1/\|\mathbf{A}^{-1}\mathbf{h}_2\|$. $\|\cdot\|$ is defined as the two-norm of a vector. Of course, due to the noise in data, the computed matrix $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$ does not in general satisfy the properties of a rotation matrix. *Appendix C* describes a method to estimate the best rotation matrix from a general 3 X 3 matrix.

3.1.4.2 Maximum Likelihood Estimation

The above solution is obtained by minimizing an algebraic distance which is not physically meaningful. It is refined through maximum likelihood inference.

Assuming n images of a model plane and m points on each model plane are given. In addition, assume that the image points are corrupted by independently and identically distributed noise. The maximum likelihood estimate can be obtained by minimizing the following function:

$$J_1 = \sum_{i=1}^{n}\sum_{j=1}^{m}\left\|\mathbf{m}_{ij} - \hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}_i, \mathbf{t}_i, \mathrm{M}_j)\right\|^2, \tag{3.10}$$

where $\hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}_i, \mathbf{t}_i, \mathrm{M}_j)$ is the projection of a point M_j in image i, according to Eq. (3.2). A rotation matrix $\mathbf{R}$ is parameterized by a vector of 3 parameters, denoted by $\mathbf{r}$. The vector is parallel to the rotation axis and its magnitude is equal to the rotation angle. $\mathbf{R}$ and $\mathbf{r}$ are related by the Rodrigues formula[41]. Minimizing (3.10) is a nonlinear minimization problem, which is solved with the Levenberg-Marquardt Algorithm as implemented in Minpack[42]. It requires an initial approximation of $\mathbf{A}, \{\mathbf{R}_i, \mathbf{t}_i | i = 1,\ldots,n\}$ which can be obtained using the technique described in the previous subsection (**3.1.4.1**).

3.1.4.3 Dealing with Radial Distortion

Up to now, lens distortion of a camera has not been considered. However, a desktop camera usually exhibits significant lens distortion, especially radial distortion. In this section, the only consideration is the first two terms of radial distortion.

Let (u, v) be the ideal (nonobservable distortion-free) pixel image coordinates, and $(\breve{u}, \breve{v})$ the corresponding real observed image coordinates. The ideal points are the projection of the model points according to the pinhole model. Similarly, (x, y) and $(\breve{x}, \breve{y})$ are the ideal (distortion-free) and real (distorted) normalized image coordinates. The relations are given[43]

$$\breve{x} = x + x[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]$$

$$\breve{y} = y + y[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]$$

where k_1, k_2 are the coefficients of the radial distortion. The center of the radial distortion is the same as the principal point. From $\breve{u} = u_0 + \alpha.\breve{x} + \gamma.\breve{y}$; $\breve{v} = v_0 + \beta\breve{y}$ and $u = u_0 + \alpha.x + \gamma.y$; $v = v_0 + \beta y$ the followings are obtained

$$\breve{u} = u + (u - u_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \tag{3.11}$$

$$\breve{v} = v + (v - v_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \tag{3.12}$$

Estimating Radial Distortion by Alternation.

As the distortion is expected to be small, one would expect to estimate the other five intrinsic parameters, using the technique described in sub-section **(3.1.4.2)**, reasonable well by simply ignoring distortion. One strategy is then to estimate $k_1.k_2$ after having estimated the other parameters, which will give us the ideal pixel coordinates (u,v). Then, from (3.11) and (3.12), two equations for each point in each image are given as

$$\begin{bmatrix} (u-u_0)(x^2+y^2) & (u-u_0)(x^2+y^2)^2 \\ (v-v_0)(x^2+y^2) & (v-v_0)(x^2+y^2)^2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} \breve{u}-u \\ \breve{v}-v \end{bmatrix}$$

Given m points in n images, all equations can be stacked together to obtain $2mn$ equations, or in matrix form as $\mathbf{Dk}=\mathbf{d}$, where $\mathbf{k}=[k_1,k_2]^T$. The linear-least-quares solution is given by

$$\mathbf{k}=\left(\mathbf{D}^T\mathbf{D}\right)^{-1}\mathbf{D}^T\mathbf{d} \qquad (3.13)$$

Once k_1 and k_2 are estimated, the estimate of the other parameters can be refined by solving (3.10) with $\hat{\mathbf{m}}(\mathbf{A},\mathbf{R}_i,\mathbf{t}_i,\mathrm{M}_j)$ replaced by (3.11) and (3.12). These two procedures can be alternated until convergence.

Complete Maximum Likelihood Estimation.

Experimentally, the convergence of the above alternation technique is found being slow. A natural extension to (3.10) is then to estimate the complete set of parameters by minimizing the following function

$$J_2 = \sum_{i=1}^{n}\sum_{j=1}^{m}\left\|\mathbf{m}_{ij} - \hat{\mathbf{m}}(\mathbf{A}, k_1, k_2, \mathbf{R}_i, \mathbf{t}_i, \mathrm{M}_j)\right\|^2, \tag{3.14}$$

where $\hat{\mathbf{m}}(\mathbf{A}, k_1, k_2, \mathbf{R}_i, \mathbf{t}_i, \mathrm{M}_j)$ is the projection of point M_j in image i according to Eq. (3.2), followed by distortion according to (3.11) and (3.12). This is a nonlinear minimization problem, which is solved with the Levenberg-Marquardt Algorithm as implemented in Minpack. A rotation matrix is again parameterized by a 3-vector $\mathbf{r}$ as described in sub-section (**3.1.4.2**). An initial approximation of $\mathbf{A}$ and $\{\mathbf{R}_i, \mathbf{t}_i | i = 1, \ldots, n\}$ can be obtained using the technique described in sub-section (**3.1.4.1**) and (**3.1.4.2**). An initial approximation of k_1 and k_2 can be obtained with the technique described in the last paragraph, or simply by setting them to 0.

3.2 OpenCV Calibration Approach

In this section the calibration approach proposed by Jean-Yves Bouguet[44] is presented. The internal camera model is very similar to that presented above and the model used by Heikkila and Silven[33]. The lens distortion includes 5 terms, denoted by $k_c = [k_1, k_2, k_3, k_4, k_5]^T$. The three additional terms represent tangential distortion coefficients, while the first two terms are radial distortion coefficients as stated before. The tangential distortion is due to "decentering", or imperfect centering of the lens components and other manufacturing defect in a compound lens. The relationship between (x, y) and $(\breve{x}, \breve{y})$, being the ideal (distortion-free) and real (distorted) normalized image coordinates respectively, is as follows

$$\breve{x} = \left[1 + k_1 r^2 + k_2 r^4 + k_5 r^6\right]x + 2k_3 xy + k_4\left(r^2 + 2x^2\right) \tag{3.15}$$

$$\breve{y} = \left[1 + k_1 r^2 + k_2 r^4 + k_5 r^6\right] y + k_3\left(r^2 + 2y^2\right) + 2k_4 xy \tag{3.16}$$

where $r^2 = x^2 + y^2$. From $\breve{u} = u_0 + \alpha\breve{x} + \gamma\breve{y}$ and $\breve{v} = v_0 + \beta\breve{y}$, the relationship between (u, v) and $(\breve{u}, \breve{v})$, being the ideal (nonobservable distortion-free) pixel image coordinates and the corresponding real observed image coordinates, is as follows

$$\begin{aligned} \breve{u} = u + \left(u - u_0\right)\left(k_1 r^2 + k_2 r^4 + k_5 r^6\right) + k_3\left[2\alpha xy + \gamma\left(r^2 + 2y^2\right)\right] \\ + k_4\left[\alpha\left(r^2 + 2x^2\right) + 2\gamma xy\right] \end{aligned} \tag{3.17}$$

$$\breve{v} = v + \left(v - v_0\right)\left(k_1 r^2 + k_2 r^4 + k_5 r^6\right) + k_3\beta\left(r^2 + 2y^2\right) + 2k_4\beta xy \tag{3.18}$$

The equation (3.17) and (3.18) can be expressed into matrix form as follow

$$\begin{bmatrix} r^2 & r^4 & \dfrac{2\alpha xy + \gamma\left(r^2 + 2y^2\right)}{u - u_0} & \dfrac{\alpha\left(r^2 + 2x^2\right) + 2\gamma xy}{u - u_0} & r^6 \\ r^2 & r^4 & \beta\left(r^2 + 2y^2\right) & 2\beta xy & r^6 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \\ k_5 \end{bmatrix} = \begin{bmatrix} \dfrac{\breve{u} - u}{u - u_0} \\ \dfrac{\breve{v} - v}{v - v_0} \end{bmatrix} \tag{3.19}$$

The above equation can be used to estimate the distortion coefficients by applying the method presented in section **3.1**.

3.3 Inverse Mapping

The inverse mapping is a technique to compute the coordinates of a 3D point $\mathrm{M} = [X, Y, Z]^T$ in the space from the observed pixel coordinates $\breve{\mathbf{m}} = [\breve{u}, \breve{v}]^T$.

There are two steps. First, the normalized image projection vector $\mathbf{x} = [x, y]^T$ should be determined by the inverse mapping with the intrinsic parameters and the observed pixel coordinates. Second, the actual coordinates of a point M can be obtained by the inverse mapping with extrinsic parameters, normalized vector $\mathbf{x} = [x, y]^T$ and an arbitrary scalar s. In fact, a $\breve{\mathbf{m}} = [\breve{u}, \breve{v}]^T$ and $\mathbf{x} = [x, y]^T$ may consist of more than one point coordinate M because there exist an arbitrary scalar s. Therefore, one more step is needed to determine the most suitable scale factor for each point M in the space. This task is not easy and time-consuming, so it is necessary to carry out several experiments with the camera system.

<The first step>

The internal coordinate mapping is realized by using the following equation

$$\tilde{\breve{\mathbf{m}}} = \begin{bmatrix} \breve{u} \\ \breve{v} \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \breve{x} \\ \breve{y} \\ 1 \end{bmatrix} = \mathbf{A}\tilde{\breve{\mathbf{x}}} \qquad (3.20)$$

From above equation, get $\tilde{\breve{\mathbf{x}}} = \mathbf{A}^{-1}\tilde{\breve{\mathbf{m}}}$ and substitute it to (3.15) and (3.16) to find $\mathbf{x} = [x, y]^T$.

<The second step>

External coordinate relationship, which relate the world coordinate to the camera coordinate system, is given in the following equation

$$s\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \widetilde{\mathrm{M}} = \mathbf{R}\,\mathrm{M} + \mathbf{t} \tag{3.21}$$

One can get

$$\mathrm{M} = \mathbf{R}^{-1} \begin{bmatrix} sx - t_1 \\ sy - t_2 \\ s - t_3 \end{bmatrix} \tag{3.22}$$

where

$$\mathbf{t} = [t_1, t_2, t_3]^T$$

Through the experiment, it is shown that a planar surface which parallels to the projection plane contains points mapped with a common scalar s. Therefore, a camera should be set up with an appropriate orientation to observe a certain planar surface. The grid extraction from the surface is necessary to estimate the extrinsic parameters and scale factor s as well. Once **R**, **t** and s are obtained, M is computed by substituting them in Eq. 3.22.

4

Obstacle Avoidance Path Planning Approaches

The motion planning is a standard task in the robot navigation field and there can be found numerous works on it[13, 14, 45]. In this chapter, two well-known approaches are shortly introduced for this essential task. These approaches are to generate a collision free reference path for obstacle avoidance of a mobile robot. Throughout this chapter, the obstacle avoidance problem concerning path optimality is specifically addressed. In the first approach based on Hamilton-Jacobi-Bellman Equation, path length is subject to be optimized. In addition, a remarkable high of path generating rate makes it possible for online application wherein the path is modified by promptly following any change of obstacle arrangement. The second approach, namely, the polynomial path planning, is to generate a smooth polynomial path. The smooth characteristic of the path permits the mobile robot to obtain a good tracking performance. Also, in the second path planning approach, the generated path must be the shortest among collision-free polynomials.

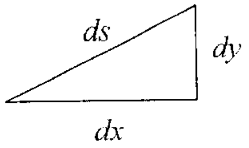
4.1 Hamilton-Jacobi-Bellman Equation Base Algorithm[14]

4.1.1 Hamilton-Jacobi-Bellman Equation

Consider a curve $\boldsymbol{\alpha}(s) \in R^n$ with s is the arc length and $\boldsymbol{\alpha}(s)$ represents a point on the curve with respect to the accumulated length s from the origin $\boldsymbol{\alpha}_0$. Differentiating the curve with respect to s yields the path tangent $\boldsymbol{\alpha}_s \in R^n$:

$$\frac{d\boldsymbol{\alpha}(s)}{ds} = \boldsymbol{\alpha}_s \tag{4.1}$$

Here are some constraints prescribed to the curve $\boldsymbol{\alpha}(s)$:



$$\Rightarrow \quad \|\boldsymbol{\alpha}_s\| = \frac{\|d\boldsymbol{\alpha}(s)\|}{ds} = \frac{ds}{ds} = 1 \tag{4.2}$$

$$\begin{aligned} \boldsymbol{\alpha}(0) &= \boldsymbol{\alpha}_0; \\ \boldsymbol{\alpha}(s_f) &= \boldsymbol{\alpha}_f \end{aligned} \tag{4.3}$$

Treating distance s as time t and the tangent vector $\boldsymbol{\alpha}_s$ as the control $\mathbf{u}$, Eq. (4.1) can be written as the first order dynamical system

$$\dot{\boldsymbol{\alpha}} = \mathbf{u}; \qquad \boldsymbol{\alpha}, \mathbf{u} \in R^n \tag{4.4}$$

The shortest path problem is formulated as the following time-optimal control problem:

$$\min_{\mathbf{u}} \int_0^{t_f} 1\, dt \tag{4.5}$$

Constraint (4.2) and boundary conditions (4.3) can be expressed into

$$\|\mathbf{u}\| = 1 \tag{4.6}$$

$$\boldsymbol{\alpha}(0) = \boldsymbol{\alpha}_0; \\ \boldsymbol{\alpha}(t_f) = \boldsymbol{\alpha}_f \tag{4.7}$$

and obstacle constraints: $\mathbf{g}(\boldsymbol{\alpha}) \geq \mathbf{0}; \quad \mathbf{g} \in R^m$ (4.8)

where m is known as the number of obstacles.

In addition, it is assumed that the free space is connected, and that the obstacles are nonintersecting, finite in size and not overlapping with the goal. Further, assume that the obstacles are circular, which implies that **g** is strictly convex and differentiable. Therefore, the set of obstacles $\{O\}$ is denoted as follows:

$$\{O\} = \{\boldsymbol{\alpha} : \mathbf{g}(\boldsymbol{\alpha}) < \mathbf{0}\} \tag{4.9}$$

Without loss of generality, the problem (4.5), with constraints (4.6), (4.7), (4.8), and the goal $\boldsymbol{\alpha}_f$, denoted by $\mathbf{0}$, to be exchanged with the origin $\boldsymbol{\alpha}_0$, can be formulated with Hamilton-Jacobi-Bellman function with state constraints. There can be found the solution according to *Theorem 1*[14], quoted as follows:

Theorem 1[18]: The control $\mathbf{u}^*(\boldsymbol{\alpha})$ is the solution of the problem (4.5) if it satisfies, the following HJB equation on $\{ R^n \setminus \{\mathbf{0}\} \setminus \{O\} \}$:

$$\min_{\mathbf{u}} \langle v_{\alpha}(\boldsymbol{\alpha}), \mathbf{u} \rangle = -1 \tag{4.10}$$

From (4.6) and (4.8), where $v(\boldsymbol{\alpha})$ is a mapping, $v(\boldsymbol{\alpha}): \{R^n \setminus \{\mathbf{0}\} \setminus \{O\}\} \mapsto R^1$, and satisfies the following constraints:

$$v(\mathbf{0}) = 0 \tag{4.11}$$

$$v(\boldsymbol{\alpha}) > 0, \qquad \boldsymbol{\alpha} \neq \mathbf{0} \tag{4.12}$$

where $v_{\alpha}(\boldsymbol{\alpha}) = \nabla_{\alpha} v(\boldsymbol{\alpha})$, is gradient of $v(\boldsymbol{\alpha})$, and the subscript α denotes the partial derivatives with respect to $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T$. The scalar function $v(\boldsymbol{\alpha})$ is the *return function*[46] representing the minimum *distance-to-go* to the origin.

Consequently, the solution of (4.10) subjected to the constraints (4.6) is the optimal control $\mathbf{u}^*(\boldsymbol{\alpha})$:

$$\mathbf{u}^*(\boldsymbol{\alpha}) = -\frac{v_{\alpha}(\boldsymbol{\alpha})}{\|v_{\alpha}(\boldsymbol{\alpha})\|} \tag{4.13}$$

Substituting $\mathbf{u}^*(\boldsymbol{\alpha})$ into (4.10) yields

$$\|v_{\alpha}(\boldsymbol{\alpha})\| = 1 \tag{4.14}$$

Substituting (4.13) and (4.14) in the system dynamics (4.4) results in

$$\dot{\boldsymbol{\alpha}} = -v_{\alpha}(\boldsymbol{\alpha}) \tag{4.15}$$

Obstacle free paths are, thus, generated by following $-v_{\alpha}(\boldsymbol{\alpha})$, the negative gradient of the *return function*. The *return function* generates paths that are globally optimal since $v(\boldsymbol{\alpha})$ has a unique minimum at the goal (the origin), as stated in the *Corollary 1*[14], quoted as the following

Corollary 1: The function $v(\boldsymbol{\alpha})$, satisfying (4.10), has a unique minimum at the goal.

It is generally difficult to find an analytical solution to (4.10), even for simple cases. It would be more convenient, for the shortest path problem, if $v(\boldsymbol{\alpha})$ is derived geometrically taking advantage of the known structure of the shortest path, like what will be presented in the next section.

4.1.2 Return Function

Now, the derivation of the return function for one circular obstacle in R^2 is presented. The extension to the manifold R^n is conceptually simple and can be discussed later.

Consider a circular obstacle (Fig. 4-1), denoted **OB**. It has a radius r and a center at $\mathbf{Q} \in R^2$. An area **S** is defined as *obstacle shadow* as follows

$$\begin{aligned}\mathbf{S} = \{\boldsymbol{\alpha} : \angle\alpha \in [\angle T^{(1)}, \angle T^{(2)}], \|\boldsymbol{\alpha} - \mathbf{Q}\|^2 \geq r^2, \\ \|\boldsymbol{\alpha} - \boldsymbol{\alpha}_f\| \geq \|\mathbf{Q} - \boldsymbol{\alpha}_f\|\}\end{aligned} \tag{4.16}$$

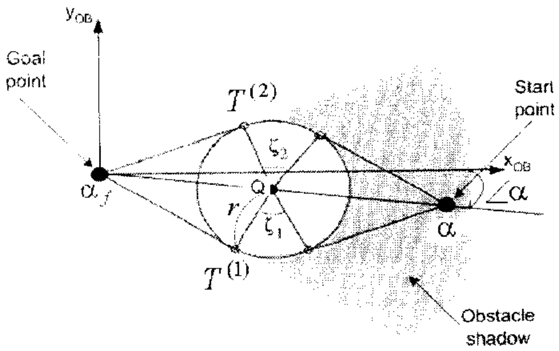


Fig. 4-1 One circular obstacle

where $\angle\alpha$ is the angle between a point $\boldsymbol{\alpha}$ and the x_{OB} axis, and $T^{(i)} \in R^2, i = 1,2$ are the contacted points on **OB** of the two tangents from the goal. Return function $\rho(\boldsymbol{\alpha}, \mathbf{Q}, r)$ of an arbitrary point $\boldsymbol{\alpha} \in R^2$ is equal to the length of optimal path. Return function can be written as:

$$\rho(\boldsymbol{\alpha}, \mathbf{Q}, r) = \begin{cases} w_Q(\boldsymbol{\alpha}, \mathbf{Q}, r) & \text{if } \boldsymbol{\alpha} \in \mathbf{S} \\ \left\|\boldsymbol{\alpha} - \boldsymbol{\alpha}_f\right\|, & \text{if } \boldsymbol{\alpha} \notin \mathbf{S} \end{cases} \tag{4.17}$$

$$w_Q(\boldsymbol{\alpha}, \mathbf{Q}, r) = \min\left\{\sqrt{\left\|\boldsymbol{\alpha} - \mathbf{Q}\right\|^2 - r^2} + r\zeta_i(\boldsymbol{\alpha}) + \sqrt{\left\|\mathbf{Q} - \boldsymbol{\alpha}_f\right\|^2 - r^2}\right\} \tag{4.18}$$

where $\zeta_i(\boldsymbol{\alpha}), i = 1,2$ are angles shown in Fig. 4-1. Note that the function $\left\|\boldsymbol{\alpha} - \boldsymbol{\alpha}_f\right\|$ is called the *unconstrainted return function* since it is the return function for problem (4.5) without obstacles. The function $w_Q(\boldsymbol{\alpha}, \mathbf{Q}, r)$, called the *constrainted return function*, is the return function for points inside the obstacle shadow **S** and depends on the obstacle parameters **Q** and r.

4.1.3 Pseudo-Return Function

This pseudo-return function is proposed to treat multiple obstacles by avoiding the obstacles optimally, one at a time. The avoidance procedure is simple: follow the gradient of the constrained return function of one (*nearest*) obstacle, denoted $\mathbf{OB}_k$, until leaving its obstacle shadow at one of the two tangent points $T_k^{(j)}$, $j = 1,2$. If at this point, the path enters the shadow of another obstacle, then, repeat this procedure using the constrained return function of the new obstacle. Otherwise, go to the goal, using the *unconstrained return function* (4.17)

The *nearest* obstacle to be avoided at a given point can be selected from the **J**, defined as:

$$\mathbf{J} = \left\{ j : \left\| \boldsymbol{\alpha} - \mathbf{Q}_j \right\| = \min_{\{i : \boldsymbol{\alpha} \in \mathbf{S}_i\}} \left\{ \left\| \boldsymbol{\alpha} - \mathbf{Q}_i \right\| \right\} \right\} \tag{4.19}$$

Another choice of the *nearest* obstacle is the maximum cost obstacle, i.e., the one that maximizes the value of the return function (4.17) at the current point, $\boldsymbol{\alpha}$. The set **J** is then defined as:

$$\begin{aligned} \mathbf{J} = \{ & j : \rho_j(\boldsymbol{\alpha}, \mathbf{Q}_j, r_j) > \|\boldsymbol{\alpha}\|, \rho_j(\boldsymbol{\alpha}, \mathbf{Q}_j, r_j) \geq \rho_i(\boldsymbol{\alpha}, \mathbf{Q}_i, r_i) \} \\ & \forall i \neq j, \qquad i = 1, \ldots, m \end{aligned} \tag{4.20}$$

where $\rho(\boldsymbol{\alpha}, \mathbf{Q}_j, r_j)$ represents the return function for the *j*th obstacle $\mathbf{OB}_j$.

Note that to compute **J**, it is necessary to consider only those obstacles with shadows containing $\boldsymbol{\alpha}$. If $\boldsymbol{\alpha}$ does not lie in the shadow of any obstacle, then the set **J** is empty, and the optimal solution is the straight line to the goal.

Consider the nearest circular obstacle with a radius r_k as $\mathbf{Q}_k \in R^2$ and $\boldsymbol{\alpha}$ as a current point. From Eq. (4.17) the path is thus generated by following the negative gradient of the pseudo-return function, $\psi(\boldsymbol{\alpha}, k)$, which is defined as follows:

$$\psi(\boldsymbol{\alpha}, k) = \begin{cases} d_1^{(k)} + d_2^{(k)} + r_k \min[\zeta_1^{(k)}, \zeta_2^{(k)}] & \text{if } k \neq 0 \\ \left\| \boldsymbol{\alpha} - \boldsymbol{\alpha}_f \right\| & \text{if } k = 0 \end{cases} \tag{4.21}$$

where

$$\begin{aligned} d_1^{(k)} &= \sqrt{\left\| \boldsymbol{\alpha} - \mathbf{Q}_k \right\|^2 - r_k^2}, \\ d_2^{(k)} &= \sqrt{\left\| \mathbf{Q}_k - \boldsymbol{\alpha}_f \right\|^2 - r_k^2} \end{aligned} \tag{4.22}$$

and the $\zeta_1^{(k)}(\boldsymbol{\alpha})$, $\zeta_2^{(k)}(\boldsymbol{\alpha})$are angles shown in Fig. (4.1), k is the index of the nearest obstacle that is selected from the set $\mathbf{J}$. $k = 0$ if $\mathbf{J}$ is empty. The $\psi(\boldsymbol{\alpha}, k)$ is called *pseudo-return function* since it does not satisfy the properties of a return function. In particular, it is discontinuous along the boundaries of the obstacles shadows, where k changes and the return function "jumps" from one obstacle to another. The algorithm with *pseudo-return function* can be summarized as follows:

Algorithm 1:

Step 1: Determine the nearest obstacle, $\mathbf{OB}_k$, using (4.19). If $k = 0$, go to step 3.

Step 2: If $k \neq 0$, follow the negative gradient of the *pseudo-return function* (4.21), $-\psi_{\alpha}(\boldsymbol{\alpha}, k)$, until reaching one of tangent points $T_k^{(j)}$, $j = 1,2$. Go to step 1.

Step 3: If $k = 0$, follow the negative gradient of the *unconstrained return function* (4.17), $\|\boldsymbol{\alpha} - \boldsymbol{\alpha}_f\|$ until reaching the goal. Stop.

A path generated by *Algorithm 1* is called a simple path since it ignores the special cases discussed later. The simple paths are guaranteed to reach the goal for all initial condition $\boldsymbol{\alpha}_0 \in R^2 \setminus \{O\}$. Where $\{O\}$ represents spaces occupied by obstacles.

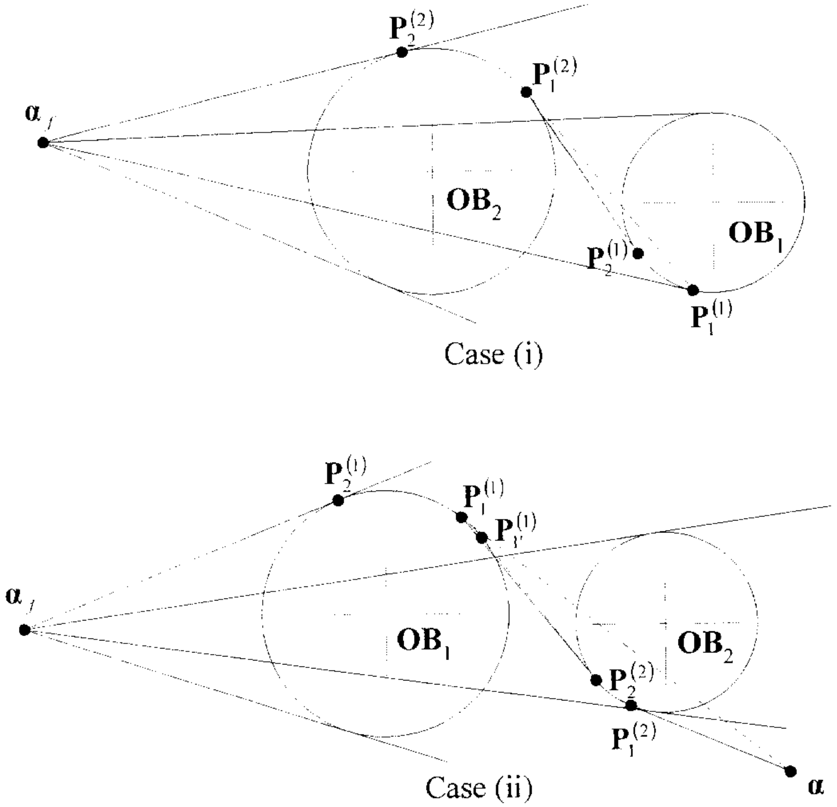
4.1.4 Special Cases

Algorithm 1 ignores special cases in which the simple path may intersect other obstacles while avoiding the nearest obstacle. There are two cases, as shown in Fig. (4-2): i) the path intersects the obstacle, which has just been avoided and ii) the path intersects another obstacle in the shadow of the nearest obstacle.

Both cases can be treated by defining the tangent point to the next obstacle as an intermediate goal, as shown schematically in Fig. (4.2).

Case i), the path exists the shadow of **OB**$_1$ at point $\mathrm{P}_1^{(1)}$, switching to the return function of the nearest obstacle, **OB**$_2$. However, the straight line path that avoids **OB**$_2$ from $\mathrm{P}_1^{(1)}$, i.e., the simple path, intersects **OB**$_1$. Therefore, $\mathrm{P}_1^{(2)}$ can be defined as an intermediate goal. In that case, the current point ($\mathrm{P}_1^{(1)}$) lies in the shadow of **OB**$_1$ to the intermediate goal ($\mathrm{P}_1^{(2)}$). The path follows the boundary of **OB**$_1$ until point $\mathrm{P}_2^{(1)}$. At this point ($\mathrm{P}_2^{(1)}$) the algorithm switches to the return function of the nearest obstacle, **OB**$_2$, to the original goal.

Case ii) is treated similarly, as shown in Fig. (4-2). Here, the current point lies in the shadow of **OB**$_1$ but not of **OB**$_2$. However, the path that avoids **OB**$_1$ intersects **OB**$_2$. $\mathrm{P}_1^{(1)}$ can be defined as the intermediate goal, and use *Algorithm 1* to avoid **OB**$_2$. After $P_2^{(2)}$ has been reached, *Algorithm 1* is used with the original goal.



Case (i)

Case (ii)

Fig. 4-2 Special cases

4.1.5 Discussion

This method is especially efficient for on-line application and for environment with high obstacle density since the computational effort at a typical point along the path is independent on the number of obstacles. The essential points to be considered for all obstacles are the tangential point $T_k^{(j)}$, $j = 1,2$, where the path switches from one obstacles shadow to another. As the result, the generated paths have the nearly shortest length since the *pseudo return function*, an approximation for the *return function*, is used in *Algorithm 1*. However, it is priorly stated that the discontinuity along the boundaries of the obstacles shadows, where k changes and the return function "jumps" from one obstacle to another, results in the discontinuities to the whole path. Consequently, these discontinuities may degrade the motion tracking of a mobile robot along the path, particularly for the vehicle running with high velocity.

For a mobile robot with the half-width, denoted as b, it is necessary to add b to the obstacles radius, r_k, then the obstacles radius should becomes $r_k + b$ while executing the *Algorithm 1*. Also, it should be noted that this increment must not violate the assumption stating that the free space is connected, and that the obstacles are non-intersected.

4.2 Polynomial Path Planning Approach[13]

4.2.1 Algorithm

This algorithm, in a known planar environment, is to generate the shortest polynomial path that avoids obstacles and connects the start point with the goal points. The first end of polynomials, $\boldsymbol{\alpha}_0$, is denoted as a start point, and another end, $\boldsymbol{\alpha}_f$, is denoted as the goal point. These points are assumedly known. $\boldsymbol{\alpha}(p)$, being a point on the curve, has coordinates defined by the following polynomial functions of a parameter $p \in R^1, 0 \le p \le 1$

$$\boldsymbol{\alpha}_0 = [x(0), y(0)]^T = \boldsymbol{\alpha}(0); \; \boldsymbol{\alpha}_f = [x(1), y(1)]^T = \boldsymbol{\alpha}(1); \qquad (4.23)$$

$$\boldsymbol{\alpha}(p) = [x(p), y(p)]^T = \left[\sum_{i=0}^{n-1} k_{xi} p^i, \sum_{i=0}^{n-1} k_{yi} p^i\right]^T \qquad (4.24)$$

where $(n-1)$ is the order of the polynomial. The obstacle assumptions are similar to those employed by the HJB based path planning method. Therefore, the set of obstacle $\{O\}$ is denoted as Eq. (4.9). Furthermore, increments of vehicle's half-width, b, to obstacles' radius are indispensable to a real implementation. Consequently, the obstacle constraints is given as Eq. (4.8), i.e., $\mathbf{g}(\boldsymbol{\alpha}) \geq \mathbf{0}$; $\mathbf{g} \in R^m$.

A second-order polynomial path from $\boldsymbol{\alpha}_0$ to $\boldsymbol{\alpha}_f$ can be determined by coefficients k_{xi} and k_{yi}, with $i = 0,1,2$, for polynomial parametric functions. With the two boundary constraints as shown in Eq. (4.23), one more point constraint is needed to determine a complete set of parameters. For this reason, a set of third points can be defined as the third constraints for l polynomials as follows

$$\boldsymbol{\alpha}_j\left(\frac{1}{2}\right) = \left[x\left(\frac{1}{2}\right), y\left(\frac{1}{2}\right)\right]^T = \boldsymbol{\alpha}_0 + \frac{\|\boldsymbol{\alpha}_0 - \boldsymbol{\alpha}_f\|}{2}\left[\cos(j\theta), \sin(j\theta)\right]^T \tag{4.25}$$

$$j \in Z^1;\ j = -\frac{l}{2}, \ldots, -1, 0, 1, \ldots, \frac{l}{2} \tag{4.26}$$

where l is the number of polynomials,

j is an integer denotes the index or label posted to each polynomial,

θ is a constant angle. This angle decides coverage of the polynomial set since each polynomial is varied by the product of this angle by an index j (Eq. 4.25).

From Eq. (4.25), the third points are given as a set of equi-angular (θ) points from $\boldsymbol{\alpha}_0$ at a radius $\|\boldsymbol{\alpha}_0 - \boldsymbol{\alpha}_f\|/2$. The number of polynomials l and the equi-angular step θ control the coverage of the set. These parameters should be appropriately chosen based on environment features and the required calculation rate.

When the parameters are known, process of determining the coefficients of parametric polynomial function is as the followings

The constraints for each jth second-order polynomial are written as follows

$$\boldsymbol{\alpha}(p) = [x(p), y(p)]^T = \left[k_{x2}p^2 + k_{x1}p + k_{x0}, k_{y2}p^2 + k_{y1}p + k_{y0}\right]^T \tag{4.27}$$

$$\boldsymbol{\alpha}(0) = \boldsymbol{\alpha}_0 = \left[k_{x2} * 0^2 + k_{x1} * 0 + k_{x0}, k_{y2} * 0^2 + k_{y1} * 0 + k_{y0}\right]^T \tag{4.28}$$

$$\begin{aligned}\boldsymbol{\alpha}_j\left(\frac{1}{2}\right) &= \left[x\left(\frac{1}{2}\right), y\left(\frac{1}{2}\right)\right]_j^T \\ &= \left[k_{x2} * \left(\frac{1}{2}\right)^2 + k_{x1} * \frac{1}{2} + k_{x0}, k_{y2} * \left(\frac{1}{2}\right)^2 + k_{y1} * \frac{1}{2} + k_{y0}\right]_j^T\end{aligned} \tag{4.29}$$

$$\boldsymbol{\alpha}(1) = \boldsymbol{\alpha}_f = \left[k_{x2} * 1^2 + k_{x1} * 1 + k_{x0}, k_{y2} * 1^2 + k_{y1} * 1 + k_{y0}\right]^T \tag{4.30}$$

where $\boldsymbol{\alpha}_0$ and $\boldsymbol{\alpha}_f$ are given, $\boldsymbol{\alpha}_j\left(\frac{1}{2}\right)$ is obtained by Eq. (4.25).

By denoting $\boldsymbol{\alpha}_0 = [x_0, y_0]^T$, $\boldsymbol{\alpha}_j\left(\frac{1}{2}\right) = [x_1, y_1]^T$, $\boldsymbol{\alpha}_f = [x_2, y_2]^T$

Eqs. (4.28~4.30) can be expressed in matrix form for each jth polynomial

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1/2 & 1/4 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} k_{x0} \\ k_{x1} \\ k_{x2} \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \tag{4.31}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1/2 & 1/4 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} k_{y0} \\ k_{y1} \\ k_{y2} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix} \tag{4.32}$$

Therefore, k_{xi} and k_{yi} can be obtained by solving Eq. (4.31) and (4.32).

After obtaining a set of the obstacle-free polynomials, by applying the obstacle constraints (4.8) to the polynomial coverage, the shortest path among them is obtained as a solution of the approach. For a given $(n-1)$ order parametric polynomial curve, as shown in Eq. (4.24), its length can be roughly approximated as

$$length \approx \sum_{i=1}^{n-1} \sqrt{\left[x\left(\frac{i}{n-1}\right) - x\left(\frac{i-1}{n-1}\right)\right]^2 + \left[y\left(\frac{i}{n-1}\right) - y\left(\frac{i-1}{n-1}\right)\right]^2} \tag{4.33}$$

Then, a sampling rate in the parametric variable p is defined as:

$$\Delta p = \frac{1}{2 \cdot length} \tag{4.34}$$

Discretizing p by Δp increments, i.e., $p = 0, \Delta p, 2\Delta p, \ldots, 1$, a set of p can be obtained. Then, by substituting them into parametric functions, i.e., like Eq. (4.27), with known coefficients, a polynomial can be enumerated by a set of discrete coordinates points.

4.2.2 Discussion

The polynomial path planning algorithm was presented for the planar floor plan with circular obstacles. In this environment, many reasonable paths for navigation may be modeled by two-dimensional polynomials in the plane of the floor. The emphasis of this algorithm is put on speed and smoothness. Therefore, in an environment observed by a camera, this method can take advantage of the vision system to locate the obstacles in the space so that the obstacle constraint (4.8) can be instantaneously referred by the path planning process. Thus, the path generating rate is increased remarkably.

4.3 Examples and Simulations

4.3.1 Hamilton-Jacobi-Bellman Function Based Algorithm

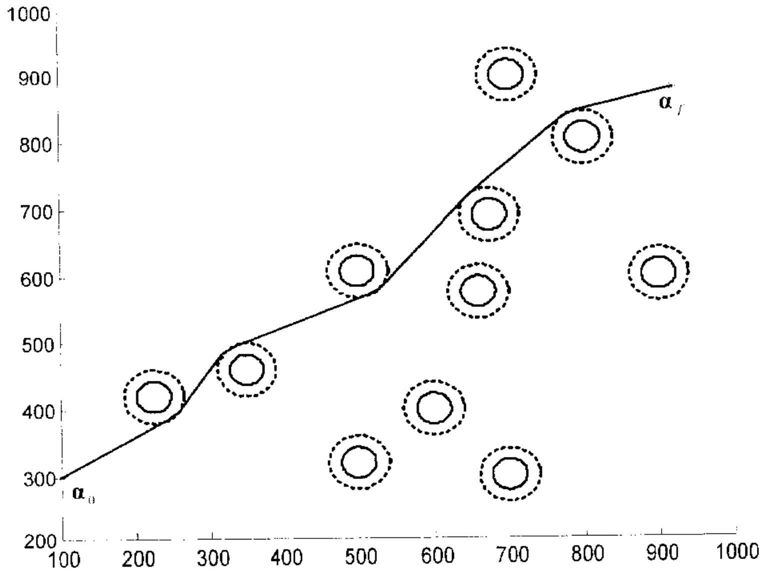
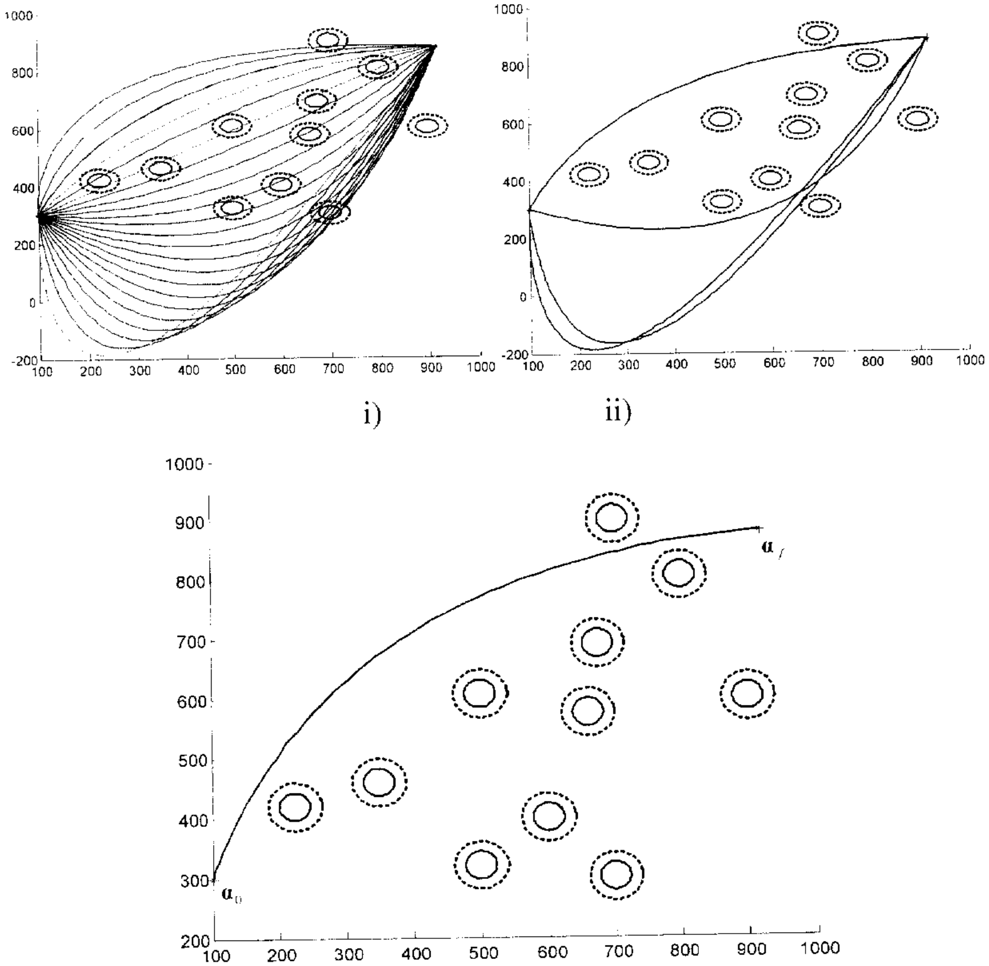


Fig. 4-3 Obstacle-free path generated by *theorem 1*

As shown (Fig. 4-3), the path is generated by *Algorithm 1*. The inside circles depict the true obstacle where the outside circles stand for obstacles that are used for path planning algorithm. The increment of the obstacles' radius is referred to the half-width of the mobile robot as stated in section 4.1.

4.3.2 Polynomial Path Planning Approach



i) ii)

iii)

Fig. 4-4 i) Polynomials coverage; ii) Obstacle-free polynomials; iii) Obstacle-free polynomial path

Above are illustrations of the polynomial path planning for a given floor plan with circular obstacles. The number of polynomials and the equi-angular step are chosen to be $l = 23$ and $\theta = 6^0$. The given start point $\boldsymbol{\alpha}_0$ and the goal point $\boldsymbol{\alpha}_f$ have a distance of around $850mm$. As shown in Fig. (4-4), in (i) a coverage of polynomials is found; in (ii) a set of obstacle-free polynomials is found; finally, in (iii) a shortest obstacle-free polynomial path is determined as the solution for the path planning algorithm.

5

Trajectory Tracking Controller Design

In the chapter 4, the path planning problem and algorithms to generate reasonable paths in a planar floor plan with obstacles have been discussed. To the mobile robot control problem, it is necessary to derive a good control law for the vehicle to track a given planned path. The trajectory tracking controller is designed by determining such control law. In the first section of this chapter, a nonlinear tracking controller for a kinematic model of mobile robot is derived[7]. Then a controller for a dynamic model are designed[11]. The control laws are stable in the sense of the Lyapunov function.

5.1 Nonlinear Tracking Controller for a Kinematic Model

The kinematic model of mobile robot is given in Eq. (2.2) as the following

$$\dot{\mathbf{q}}_1 = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = \mathbf{S}_1(\mathbf{q}_1)\mathbf{v} \tag{5.1}$$

Put $[x_r, y_r, \phi_r]^T$ as a reference tracking coordinate and $[v_r, \omega_r]^T$ as a reference input of the mobile robot.

The tracking error $[e_1, e_2, e_3]^T$ is defined as the following

$$\mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \phi_r - \phi \end{bmatrix} \tag{5.2}$$

The kinematic control law, which calculates a target velocity input $\mathbf{v}_t = [v_t, \omega_t]^T$ using the tracking error $\mathbf{e} = [e_1, e_2, e_3]^T$ and the reference velocity input $\mathbf{v}_r = [v_r, \omega_r]^T$, is designed by converging e_1, e_2, e_3 to zeros. The control law, with perfect velocity tracking assumption, is proposed as the following[7]

$$\begin{aligned} v_t &= v_r \cos e_3 + K_1 e_1 \\ \omega_t &= \omega_r + v_r (K_2 e_2 + K_3 \sin e_3) \end{aligned} \tag{5.3}$$

where K_1, K_2, K_3 are positive constants, and $\mathbf{K} = [K_1, K_2, K_3]^T$.

From Eq. (5.2), the derivative of the tracking error can be derived into

$$\dot{\mathbf{e}} = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = v \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} + \omega \begin{bmatrix} e_2 \\ -e_1 \\ -1 \end{bmatrix} + \begin{bmatrix} v_r \cos e_3 \\ v_r \sin e_3 \\ \omega_r \end{bmatrix} \tag{5.4}$$

A scalar function V_0 is chosen as a Lyapunov function candidate:

$$V_0 = \frac{1}{2}\left(e_1^2 + e_2^2\right) + \frac{1 - \cos e_3}{K_2} \tag{5.5}$$

Then, the derivative of V_0 yields

$$\dot{V}_0 = e_1\dot{e}_1 + e_2\dot{e}_2 + \dot{e}_3\frac{\sin e_3}{K_2} = -K_1 e_1^2 - \frac{K_3 \sin^2 e_3}{K_2} \leq 0 \qquad (5.6)$$

If $\mathbf{e} \neq \mathbf{0}$, it is clearly $V_0 > 0$ and $\dot{V}_0 < 0$, then V becomes a Lyapunov function. According to the Lyapunov theorem, the equilibrium point $\mathbf{e} = \mathbf{0}$ is stable in the sense of Lyapunov.

If $\mathbf{e} \cong \mathbf{0}$, the *proposition 2* proposed by Kanayama[7] is used to demonstrate that the uniformly asymptotically stability around $\mathbf{e} = \mathbf{0}$ under some conditions:

a) v_r, ω_r are continuous,

b) v_r, ω_r, K_1, K_3 are bounded,

c) $\dot{v}_r, \dot{\omega}_r$ are sufficiently small.

Under these conditions, $[e_1, e_2, e_3]^T = \mathbf{0}$ is uniformly asymptotically stable over $[0, \infty)$.

The proof of this proposition can be found in the *Appendix D*.

5.2 Nonlinear Tracking Controller for a Dynamic Model

5.2.1 A Nonlinear Feedback Controller Based on the Backstepping Control Design

From the chapter 2, a complete dynamics of the mobile robot is given by Eqs. (2.2), (2.3) and (2.28). Eq. (2.2) describes kinematics of steering system, Eq. (2.3) describes the relationship between different velocity inputs, and Eq. (2.28) describes the internal dynamics of the mobile robot. The equations can be reposted here for a greater convenience

$$\dot{\mathbf{q}}_1 = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = \mathbf{S}_1(\mathbf{q}_1)\mathbf{v} \tag{5.7}$$

$$\boldsymbol{\eta} = \begin{bmatrix} v_r \\ v_l \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{b}{r} \\ \frac{1}{r} & -\frac{b}{r} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = \overline{\mathbf{J}} \cdot \mathbf{v} \tag{5.8}$$

$$\overline{\mathbf{M}}\dot{\boldsymbol{\eta}} + \overline{\mathbf{V}}(\dot{\mathbf{q}})\boldsymbol{\eta} + \overline{\mathbf{F}}(\boldsymbol{\eta}) + \overline{\boldsymbol{\tau}}_d = \boldsymbol{\tau} \tag{5.9}$$

where $\boldsymbol{\tau} = [\tau_r \quad \tau_l]^T$ is the torque vector acting on the two wheels, $\overline{\boldsymbol{\tau}}_d$ represents estimated disturbance vector with bounded value of unknown parameters. $\overline{\mathbf{F}}(\boldsymbol{\eta})$ represents friction vector against dynamics. $\overline{\mathbf{M}} \in R^{2\times 2}$, shown in Eq. (2.30), is a symmetric, positive definite matrix. $\overline{\mathbf{V}}(\dot{\mathbf{q}}) \in R^{2\times 2}$, shown in Eq. (2.29), is a skew-symmetric matrix. $\mathbf{v} = [v, \omega]^T$ is considered as control input of the kinematic model (5.7) of a mobile robot. v and ω are known as the linear velocity and angular velocity, respectively. In fact, it is intuitive to experience the control inputs as the angular velocities of left and right wheels of the MR, denoted by v_l and v_r respectively, with $\boldsymbol{\eta} = [v_r, v_l]^T$. The relationship between $[v, \omega]^T$ and $[v_r, v_l]^T$ is expressed in Eq. (5.8).

According to the kinematic control law (5.3), it would be more convenient if one eliminates $\boldsymbol{\eta}$ from Eq. (5.9) by substituting (5.8) in (5.9). Then, by multiplying both sides of this new equation with $\overline{\mathbf{J}}^T$, the dynamics of the MR can be obtained as follows

$$\bar{\mathbf{J}}^T\bar{\mathbf{M}}\bar{\mathbf{J}}\dot{\mathbf{v}}+\bar{\mathbf{J}}^T\bar{\mathbf{V}}(\dot{\mathbf{q}})\bar{\mathbf{J}}\mathbf{v}+\bar{\mathbf{J}}^T\bar{\mathbf{F}}(\mathbf{v})+\bar{\mathbf{J}}^T\bar{\boldsymbol{\tau}}_d=\bar{\mathbf{J}}^T\boldsymbol{\tau} \tag{5.10}$$

Regenerating Eq. (5.10), one can obtain

$$(5.10)\Leftrightarrow \bar{\mathbf{M}}_1\dot{\mathbf{v}}+\bar{\mathbf{V}}_1(\dot{\mathbf{q}}_1)\mathbf{v}+\bar{\mathbf{F}}_1(\mathbf{v})+\bar{\boldsymbol{\tau}}_{d1}=\bar{\mathbf{B}}_1\boldsymbol{\tau} \tag{5.11}$$

where

$$\bar{\mathbf{M}}_1=\bar{\mathbf{J}}^T\bar{\mathbf{M}}\bar{\mathbf{J}}=\begin{bmatrix} m+2\frac{I_w}{r^2} & 0 \\ 0 & I_1 \end{bmatrix} \tag{5.12}$$

$$\bar{\mathbf{V}}_1(\dot{\mathbf{q}})=\bar{\mathbf{J}}^T\bar{\mathbf{V}}(\dot{\mathbf{q}})\bar{\mathbf{J}}=\begin{bmatrix} 0 & -m_c d\dot{\phi} \\ m_c d\dot{\phi} & 0 \end{bmatrix} \tag{5.13}$$

$$\bar{\mathbf{B}}_1=\bar{\mathbf{J}}^T=\begin{bmatrix} \frac{1}{r} & \frac{1}{r} \\ \frac{b}{r} & -\frac{b}{r} \end{bmatrix};\ \bar{\mathbf{F}}_1(\mathbf{v})=\bar{\mathbf{J}}^T\bar{\mathbf{F}}(\mathbf{v});\ \bar{\boldsymbol{\tau}}_{d1}=\bar{\mathbf{J}}^T\bar{\boldsymbol{\tau}}_d \tag{5.14}$$

$$\begin{aligned} I_1 &= I_c+2I_w\frac{b^2}{r^2}+2m_w b^2+2I_m \\ m &= m_c+2m_w \end{aligned} \tag{5.15}$$

where I_c is a moment of inertia of the vehicle without the driving wheels and the motor rotors about vertical axis passing through the point C. I_w is the moment of inertia of each driving wheel and the motor rotor about the wheel axis, I_m is the moment of inertia of the wheel about the diameter. m_c denotes mass of the vehicle without the wheels, m_w denotes mass of the wheel, and d denotes displacement from the point C to the mass center of the mobile robot, which is assumed to be on the axis of symmetry.

Therefore, the complete MR's dynamics turns out a set of two equations, i.e., Eqs. (5.7) and (5.11). Let $\mathbf{u} \in R^{2\times 2}$ be an auxiliary input, then by applying the nonlinear feedback, with $\det(\overline{\mathbf{B}}_1) \neq 0$, the following is obtained

$$\boldsymbol{\tau} = \overline{\mathbf{B}}_1^{-1}\left[\overline{\mathbf{M}}_1\mathbf{u} + \overline{\mathbf{V}}_1(\dot{\mathbf{q}}_1)\mathbf{v} + \overline{\mathbf{F}}_1(\mathbf{v}) + \overline{\boldsymbol{\tau}}_{d1}\right], \tag{5.16}$$

The dynamic control problem is converted into the kinematic control problem[11]

$$\dot{\mathbf{q}}_1 = \mathbf{S}_1(q_1)\mathbf{v} \tag{5.17.a}$$

$$\dot{\mathbf{v}} = \mathbf{u} \tag{5.17.b}$$

Eq. (5.17) represents a state-space description of the nonholonomic mobile robot and constitutes the basic framework for defining its nonlinear control properties[47, 48]. In performing the input transformation (5.16), it is assumed that all the dynamical quantities (e.g., $\overline{\mathbf{M}}_1, \overline{\mathbf{V}}_1(\dot{\mathbf{q}}_1), \overline{\mathbf{F}}_1(\mathbf{v})$) of the vehicle are exactly known. Otherwise, adaptive of robust control techniques are incorporated[8].

The last section shows a control law (5.3) that stabilizes the steering system (5.1) with the perfect velocity tracking assumption. It turns out that such assumption is not real because of the ignorance of the actual vehicle dynamics. With the desire to convert such a prescribed control $\mathbf{v}(t)$ into a torque control $\boldsymbol{\tau}(t)$ for the actual cart, Fierro *et al.* proposed the backstepping control design method[11]. This method is to select $\boldsymbol{\tau}(t)$ in (5.11) so that (5.7), (5.11) exhibits the desired behavior motivating the specific choice of the velocity $\mathbf{v}(t)$. The method, mainly focuses on the last statement that is known as the *Point Stabilization*, is quoted as follows:

Theorem 1 (***Point Stabilization***):

Given an arbitrary configuration $\mathbf{q}_{1r}$, find a smooth time-varying velocity control input $\mathbf{v}_f = \mathbf{g}_f(\mathbf{e}, \mathbf{v}_r, \mathbf{K}, t)$ such that $\lim\limits_{t\to\infty}(\mathbf{q}_{1r} - \mathbf{q}_1) = 0$.

Then define an auxiliary feedback control law $\mathbf{u} = \dot{\mathbf{v}}_f + \mathbf{K}_4(\mathbf{v}_f - \mathbf{v})$ such that $\mathbf{v} \to \mathbf{v}_f$ as $t \to \infty$. Finally, compute the torque $\boldsymbol{\tau} = \mathbf{g}_\tau(\mathbf{q}_1, \dot{\mathbf{q}}_1, \mathbf{v}, \mathbf{u})$ using (5.16).

5.2.2 Trajectory Tracking Controller Design

The controller design process is started with the kinematic control law (5.3). The derivative of the $\mathbf{v}_f$ becomes

$$\dot{\mathbf{v}}_f = \begin{bmatrix} \dot{v}_r \cos e_3 \\ \dot{\omega}_r + K_2 \dot{v}_r e_2 + K_3 \dot{v}_r \sin e_3 \end{bmatrix} + \begin{bmatrix} K_1 & 0 & -v_r \sin e_3 \\ 0 & K_2 v_r & K_3 v_r \cos e_3 \end{bmatrix} \dot{\mathbf{e}} \qquad (5.18)$$

where the derivative of the error, $\dot{\mathbf{e}}$, is given in Eq. (5.4). Eq. (5.4) can be expressed as

$$\dot{\mathbf{e}} = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \omega e_2 - v + v_r \cos e_3 \\ -\omega e_1 + v_r \sin e_3 \\ \omega_r - \omega \end{bmatrix} \qquad (5.19)$$

Then, by **Theorem 1**, the nonlinear feedback auxiliary control input $\mathbf{u} \in R^{2\times2}$ is obtained as follows

$$\mathbf{u} = \dot{\mathbf{v}}_f + \mathbf{K}_4(\mathbf{v}_f - \mathbf{v}) = \dot{\mathbf{v}}_f + K_4 \mathbf{I}(\mathbf{v}_f - \mathbf{v}) \qquad (5.20)$$

where K_4 is a positive scalar value and $\mathbf{I} \in R^{2\times2}$ is an identity matrix.

To validate the derived controller, a theorem is given as follows[11]:

Theorem 2:

Given a nonholonomic system (5.7), (5.11) with n generalized coordinate vector $\mathbf{q}$, m independent constraints, and r actuators, let the following assumptions hold:

i) The number of actuators is equal to the number of degrees of freedom (i.e., $r = n - m$).

ii) The reference linear velocity is nonzero and bounded, $v_r > 0$ for all t. The angular velocity ω_r is bounded.

iii)A smooth auxiliary velocity control input $\mathbf{v}_f \in R^{r\times 1}$ is given by Eq. 5.3.

iv) $\mathbf{K} = [K_1, K_2, K_3]^T$ is a vector of positive constants.

v) K_4 is a sufficiently large positive constant.

The nonlinear feedback control $\mathbf{u} \in R^{2\times 2}$ is given by (5.20) and the vehicle input commands are given by (5.16). Then the origin $\mathbf{e} = \mathbf{0}$ is uniformly asymptotically stable, and the velocity vector of the mobile base satisfies $\mathbf{v} \to \mathbf{v}_f$ as $t \to \infty$.

Proof:

An auxiliary velocity error is defined as

$$\mathbf{e}_f = \mathbf{v} - \mathbf{v}_f = \begin{bmatrix} e_4 \\ e_5 \end{bmatrix} = \begin{bmatrix} v - v_r \cos e_3 - K_1 e_1 \\ \omega - \omega_r - K_2 v_r e_2 - K_3 v_r \sin e_3 \end{bmatrix} \tag{5.21}$$

From Eqs. (5.20) and (5.17.b), the following is obtained

$$\dot{\mathbf{e}}_f = -K_4 \mathbf{I} \mathbf{e}_f \tag{5.22}$$

Under assumption **v)**, the auxiliary velocity vector converges exponentially to zero. Therefore, the velocity vector of the mobile base satisfies $\mathbf{v} \to \mathbf{v}_f$ as $t \to \infty$.

Consider the following Lyapunov function candidate:

$$V = K_1\left(e_1^2 + e_2^2\right) + \frac{2K_1}{K_2}\left(1 - \cos e_3\right) + \frac{1}{2K_4}\left(e_4^2 + \frac{K_1}{K_2 K_3} e_5^2\right) \tag{5.23}$$

where $V \geq 0$, and $V = 0$ only if $\mathbf{e} = \mathbf{0}$ and $\mathbf{e}_f = \mathbf{0}$. Furthermore, by using (5.19), (5.21), (5.22)

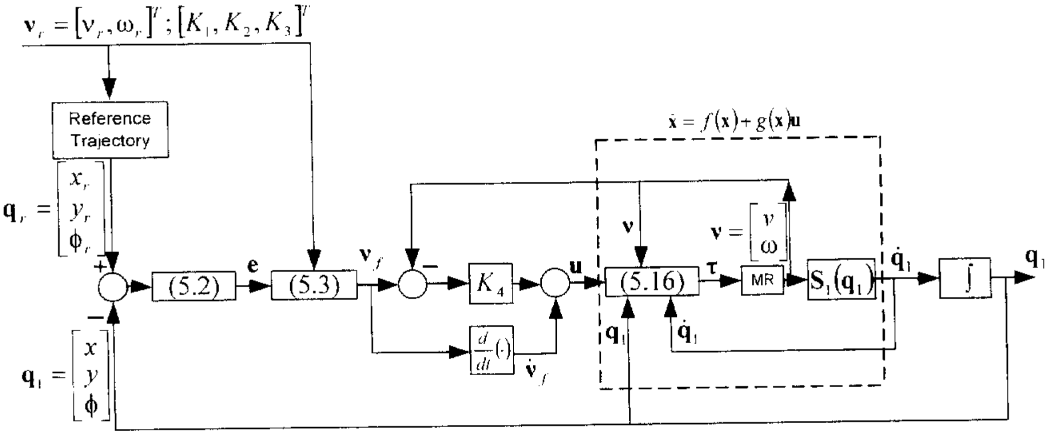
$$\dot{V} = -K_1^2 e_1^2 - \frac{K_1 K_3}{K_2} v_r \sin^2 e_3 - \left(e_4 + K_1 e_1\right)^2 - \frac{K_1}{K_2 K_3 v_r}\left(e_5 + K_3 v_r \sin e_3\right)^2 \tag{5.24}$$

Clearly, $\dot{V} \leq 0$ and the entire error $\boldsymbol{\varepsilon} = \left[\mathbf{e} \quad \mathbf{e}_f\right]^T$ is bounded. Using Eqs. (5.19), (5.21), (5.24), and assumption **iii)**, it is shown that $\|\boldsymbol{\varepsilon}\|$ and $\|\dot{\boldsymbol{\varepsilon}}\|$ are bounded, so that $\|\ddot{V}\| < \infty$, i.e., $\dot{V}$ is uniformly continuous. Since $V(t)$ does not increase and converge to some constant value, by Barbalat's lemma, $\dot{V} \to 0$ as $t \to \infty$. Considering that $\mathbf{e}_f = \left[e_4, e_5\right]^T \to \mathbf{0}$ as $t \to \infty$, then the limit of Eq. (5.24) is given as

$$0 = K_1 e_1^2 + \frac{K_1 K_3}{K_2} v_r \sin^2 e_3 \tag{5.25}$$

Eq. (5.25) implies that $\left[e_1 \quad e_3\right]^T \to 0$ as $t \to \infty$. Finally, using the definition of $\mathbf{e}_f$, it is easy to show that $e_2 \to 0$ as $t \to \infty$. Hence, the equilibrium point $\boldsymbol{\varepsilon} = \mathbf{0}$ is uniformly asymptotically stable.

Fig. 5-1 shows the block diagram of the proposed nonlinear feedback tracking controller.



$\mathbf{v}_r = [v_r, \omega_r]^T ; [K_1, K_2, K_3]^T$
Reference Trajectory
$\mathbf{q}_r = \begin{bmatrix} x_r \\ y_r \\ \phi_r \end{bmatrix}$
+
−
(5.2)
$\mathbf{e}$
(5.3)
$\mathbf{v}_f$
−
K_4
$\frac{d}{dt}(\cdot)$
$\dot{\mathbf{v}}_f$
$\mathbf{u}$
$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}$
$\mathbf{v}$
$\mathbf{v} = \begin{bmatrix} v \\ \omega \end{bmatrix}$
(5.16)
τ
MR
$S_1(\mathbf{q}_1)$
$\dot{\mathbf{q}}_1$
$\int$
$\mathbf{q}_1$
$\mathbf{q}_1$
$\dot{\mathbf{q}}_1$
$\mathbf{q}_1 = \begin{bmatrix} x \\ y \\ \phi \end{bmatrix}$

Fig. 5-1 Block diagram of nonlinear feedback tracking controller

6

Hardware Design and Implementation

6.1 Overall Control System

The experimental system is developed by using a computer vision system with a two-wheeled mobile robot. The configuration of the total control system is shown in Fig. 6-1 and the overall experimental environment is shown in Fig. 6-2.

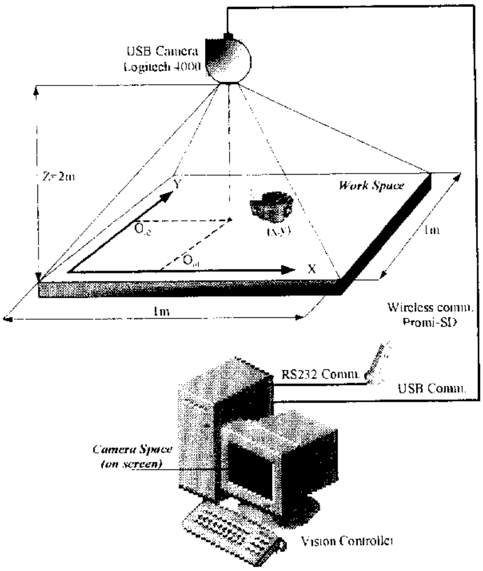


Fig. 6-1 Configuration of the total control system

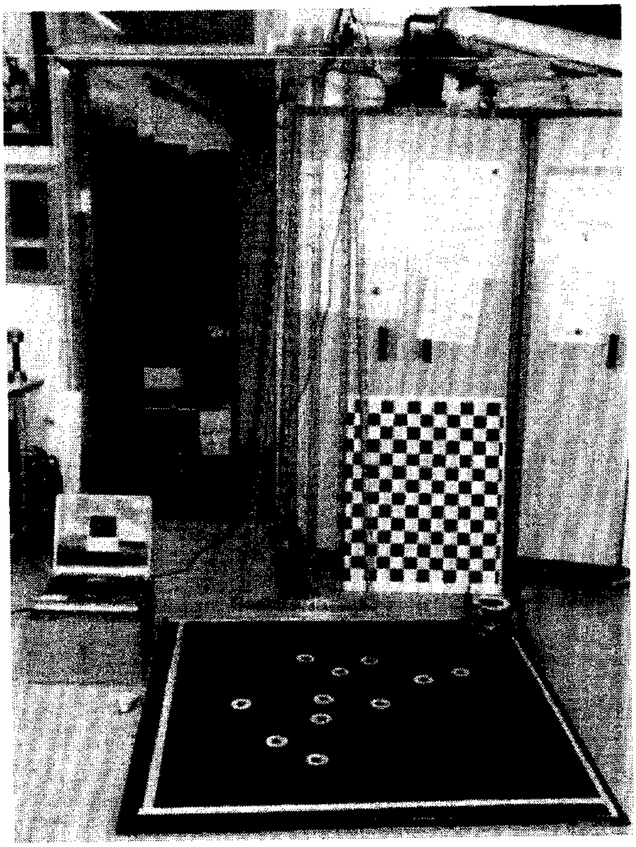


Fig. 6-2 Overall experimental system

As illustrated in Fig. 6-1, this developed control system is based on the integration of a personal computer (Intel Pentium IV, 2.8Ghz, 1GB RAM), a USB camera and PIC-based microprocessors. The control system is composed of two parts: high level computer control and low level microprocessor control. The former is used for image processing control and path planning module. The latter is used for device control. For the operation of the total control system, the image stream of the work space is captured into memory in bitmap format with size 320x240 via a USB camera Logitech 4000 at 30fps using QuickCam SDK. The image is processed by image processing library Intel OpenCV to get the coordinates of the obstacles at the initial time and the localization of the MR during the vehicle running time. The obstacles information is obtained by path planning module whereas the localization information of the MR is used for deriving control law. The computation of the control law is handled by the high level computer. Simultaneously, the control commands are sent to the low level microprocessor control via wireless communication. The software for the total control operation are programmed and integrated into one computer interface in Visual C++, namely Vision Controller CamMR V.1 (Fig. 6-3).

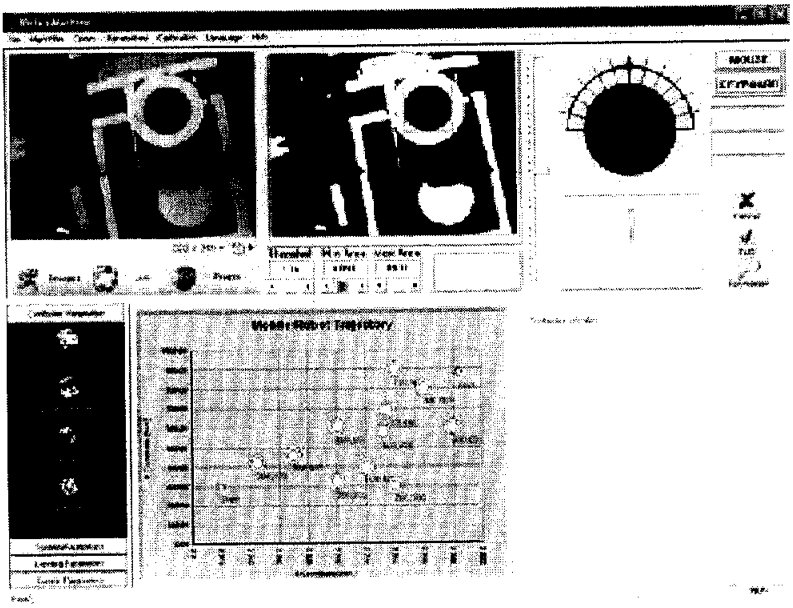


Fig. 6-3 Computer interface of the Vision controller CamMR V1.

As shown in the Fig. 6-2, the camera is mounted 2*m* above the work space. The position and the heading angle of the MR are achieved by means of the color markers on the MR. The positions of the markers are calculated using a threshold based approach that compares brightness values of pixels within a specific range. To detect the heading angle of the MR, coordinates of consecutive positions can be used while the MR is under motion. An approximate heading angle is given:

$$\phi(t) = \arctan\left(\frac{Y(t)-Y(t-1)}{X(t)-X(t-1)}\right) \tag{6.1}$$

where $X(t)$ and $Y(t)$ are MR's coordinates at an instantaneous time t. $\phi(t)$, the approximated heading angle, is the angle between the centroids of two consecutive robot positions.

The mobile robot using in this experiment has the parameters listed in the Table 6-1 when the photograph of the MR is shown in Fig. 6-5. The low level control, known as the PIC-based controller, is attached to the MR in order to implement its motion. The controller is composed of two parts: servo controller and main controller. The configuration diagram of the total control system is shown in Fig. 6-4.

Table 6-1 Parameter values of the MR

Parameters	**Value**
b [mm]	31.5
r [mm]	21.5
m_c [kg]	0.35
m_w [kg]	0.05
I_c [kgm^2]	$0.4\ .10^{-3}$
I_m [kgm^2]	$0.23\ .10^{-6}$
I_w [kgm^2]	$0.1\ .10^{-5}$

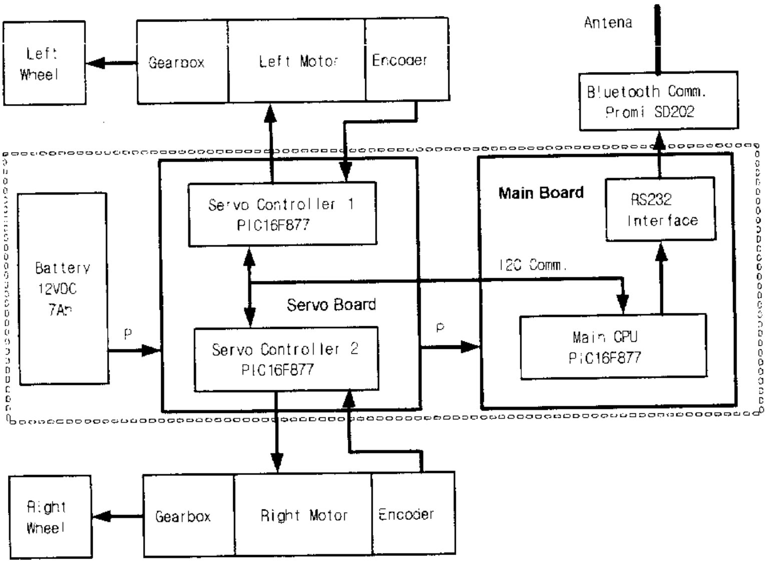


Fig. 6-4 Configuration diagram of the low level control

In this diagram (Fig. 6-4), two microprocessors using PIC16F877 are integrated into one module function as servo controllers for two motors of the left and the right wheels. The motors are driven via LMD18200 dual full-bridge driver. This servo controller can perform indirect velocity control using one encoder. Moreover, one microprocessor PIC16F877 is used as a master which receives the signal from central computer via Bluetooth wireless communication module Promi SD202. The master communicates with the servo module using I2C communication.

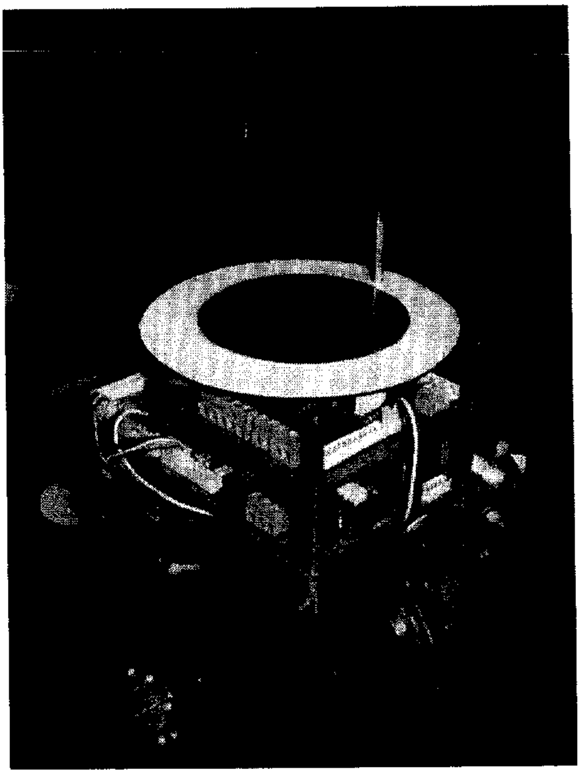


Fig. 6-5 Two-wheeled mobile robot

6.2 Implementation of Camera System Calibration

Once the MR is performing an assigned task, i.e., tracking a reference planned trajectory, it is necessary to observe its accurate position and orientation at each instantaneous time in order to enhance tracking performance. So far, several localization approaches using camera have been developed and they prove to be very meaningful aspect especially in the field of MR navigation. A computer vision system with prior calibration steps is able to serve for two purposes: i) obstacle positions mapping at the initial time, ii) mobile robot localization during operating time.

Therefore, before putting into operation, the camera needs to be calibrated in order to get familiar with the environment it observes. In this section, the calibration approach proposed by Jean-Yves Bouguet[44] is illustrated. The calibration process is carried out by working on images of a planar checkerboard, as shown in Fig. 6-6. In Fig. 6-7, 20 images of the checkerboard with different orientations are taken by the USB camera. The specification of the USB camera is given in the Table 6-4. By employing the OpenCV calibration toolbox library, the images are processed by the user. Once the calibration is terminated, the parameters are obtained and shown in Table 6-2 and 6-3. These parameters are used by in the methodology presented in chapter 3 in order to retrieve the desired information such as obstacle coordinate or mobile robot position and orientation in the work-space.

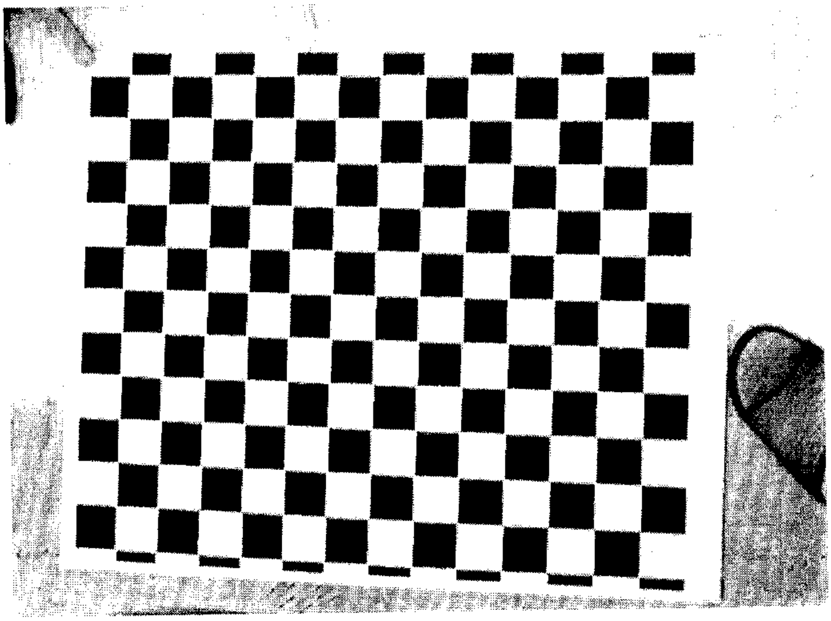


Fig. 6-6 Checkerboard used in the calibration process.

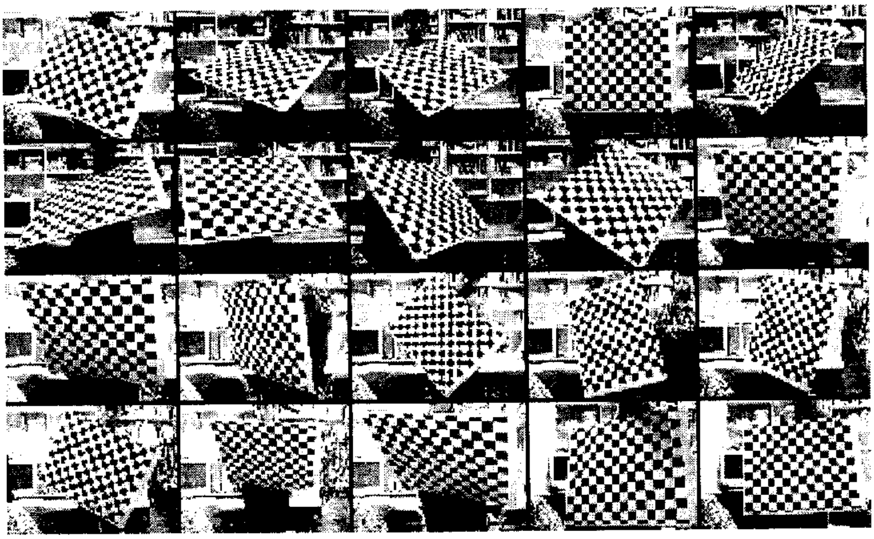


Fig. 6-7 Images of the checkerboard for the calibration process.

Table 6-2 Intrinsic parameters of the camera system and distortion coefficients

Parameters	**Value**	**Parameters**	**Value**
α	838.0078	k_1	0.0433
β	845.2424	k_2	-0.2989
γ	0.1997	k_3	-0.0012
$\bar{x}_0$	295.3777	k_4	-0.0102
$\bar{y}_0$	256.6679	k_5	0

Table 6-3 Extrinsic parameters of the camera system for the work-space

Parameters	**Value**	**Parameters**	**Value**
r_{11}	0.9887	r_{32}	-0.0201
r_{12}	-0.0448	r_{33}	0.9894
r_{13}	0.1431	t_1	-295
r_{21}	0.0414	t_2	-302
r_{22}	0.9988	t_3	1266.4
r_{23}	0.0263	s	1218.6
r_{31}	-0.1441	σ_s	12.741

Table 6-4 Specification of USB camera

Parameters	**Value**
Image size (*pixel* x *pixel*)	*320* x *240*
Branch name	Logitech 4000
Captured speed (frames/second)	30
Software	QuickCam SDK
Image processing library	Intel OpenCV
Resolution (*mm*/*pixel*)	3.5

For more convenient, the steps of calibration process are summarized as follow:

1. Grid extraction from figures (Fig. 6-7).
2. Calibration.
3. Re-calibration, if any.
4. Save the calibration results for intrinsic and extrinsic parameters.
5. Take a new figure of the checkerboard putting in the real work-space. The extrinsic parameters particularly for this figure (denoted Figure_R1) are obtained by using the routines in the calibration toolbox. Note that the upper left corner of the grid represents the origin of the world coordinate.
6. Calculate value of s for Figure_R1 by using grid data and extrinsic parameters specific for this figure. Check the distribution variance of the grid data, if the variance is large, repeat step 5 with a different orientation of checkerboard. If the variance σ_s is sufficiently small (it means that the camera image plane stays parallel to the work space plane), then the value of parameters can be saved and be ready for use, i.e., for proceed the next step (step 7). {call *zc_est.m* routine}
7. Call the routine to calculate the workspace coordinates from given pixel coordinates, intrinsic parameters, and extrinsic parameters (specific for Figure_R1). {call *inv_mapping.m* routine}

7

Simulation and Experimental Results

To verify the effectiveness of the proposed control method, simulation and experiments have been done for a two-wheeled mobile robot to track a path generated by a path planning approach in order to attain its final configuration, i.e. reaching a goal point $\boldsymbol{\alpha}_f$. The obstacle configuration is shown in Fig. 7-1. Our aim is to control the MR to get the target of $\boldsymbol{\alpha}_f = [900,880]^T$ from a given start point of $\boldsymbol{\alpha}_0 = [100,300]^T$. There are two path planning methods that are used separately to generate a reference trajectory for the MR's performance.

The parameters used for the simulation and the experiment are shown in Table 6-1. The controller parameters are $K_1 = 5; K_2 = 3000; K_3 = 110; K_4 = 1$.

7.1 Obstacle Avoidance with the HJB-based Path Planning Algorithm

By using the HJB-based path planning algorithm, the generated path has the optimal length of 1023.6 *mm*. The vehicle reference speed varies from 80*mm/sec* to 120*mm/sec*. It takes around 9.38 seconds to the target.

The trajectory tracking of the MR is shown in Fig. 7-1. Fig. 7-2 (a~c) shows tracking errors along the path. On the simulation trajectory of the MR, sudden errors occur at the discontinuous segments when the MR "jumps" from this obstacle's shadow to others'. However, it is shown that the behaviors of the errors in the experimental results are different from those in the simulation results.

In the Figure 7-2, the tracking errors differ from e_1 to e_3. Although the fluctuation of tracking error e_1 (Fig. 7-2a) is large in simulation result, it appears rather smooth in experimental result with the error less than $1mm$. For the tracking error e_2, the initial error varies within $2.5pixels$ (~$8mm$) and then reduces to $1pixel$ (~$3.5mm$) at the steady state. To reduce the risk of collision with the obstacle, the path must be planned with more safe space that compensates for the above tracking error e_2, i.e., increasing the half-width of the MR with respect to additional amount to the obstacles' radius and planning the path with these new-size obstacles. In the Fig. 7-2c, the error e_3 promptly approaches to stable state with a small error.

The experimental results show the effective aspect of this control plan. The path planning task is to generate or modify the path that is most appropriate to our tracking controller. As a result, a good performance of MR can be obtained as illustrating.

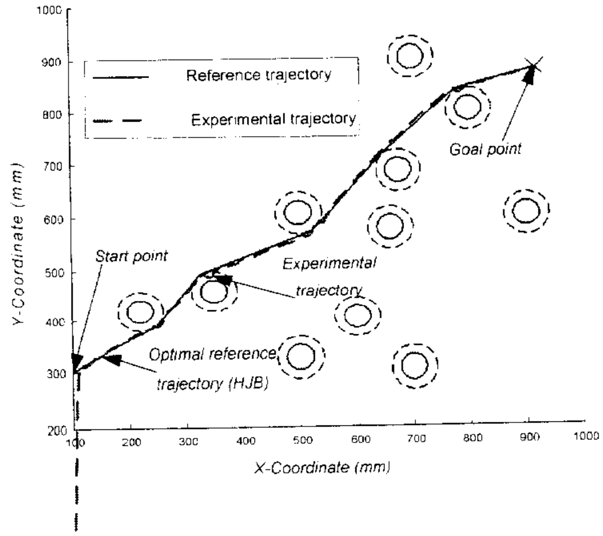
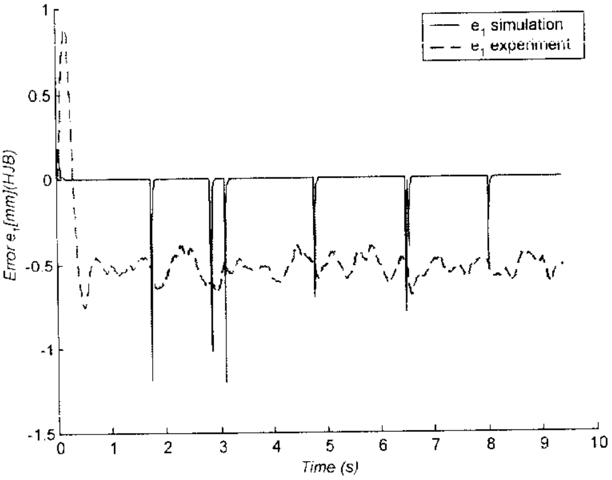
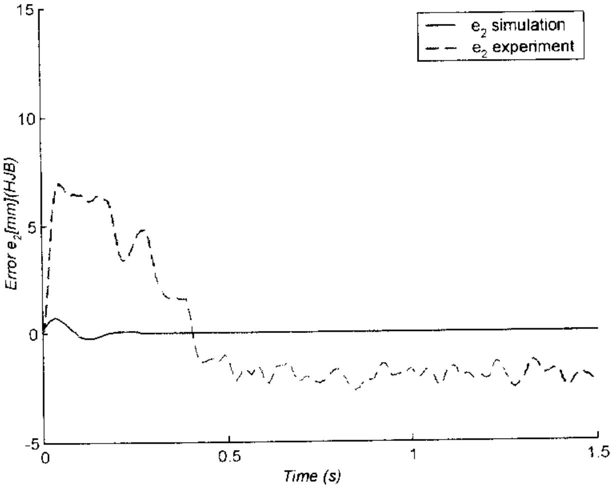


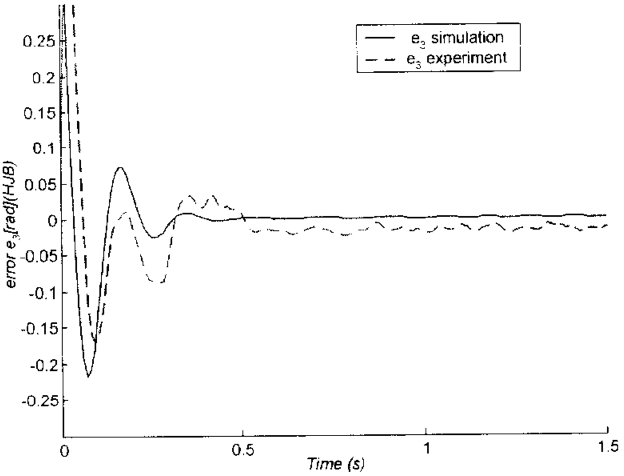
Fig. 7-1 Trajectory tracking of MR



(a) Error e_1



(b) Error e_2



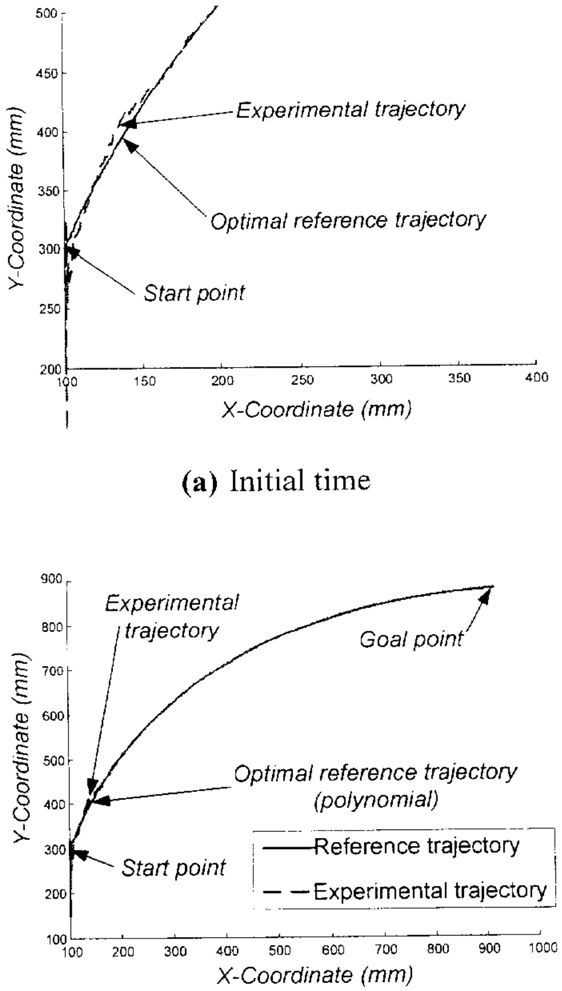
(c) Error e_3

Fig. 7-2 Tracking errors of MR along the path

7.2 Obstacle Avoidance with the Polynomial Path Planning Approach

Simulations and experiments are done for the two-wheeled MR tracking the trajectory generated by the polynomial path planning approach. The planned path has the optimal length of 1070*mm*. The vehicle reference speed is 80*mm/sec* in the initial time (the first second) and then increased to 120*mm/sec* for the rest of the path. It takes around 9.23*seconds* to the target.

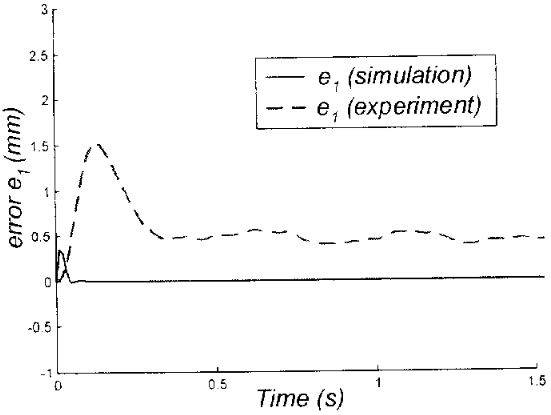
The tracking trajectory of the MR is given in Fig. 7-3. In Fig. 7-4, the tracking errors are exhibited. It appears that the MR follow well the planned path with acceptable tracking errors. Fig. 7-4a shows tracking performance of MR at the departing time. Fig. 7-3b shows the overall performance along the path. It is shown that the errors go to a stable state within one second.The good tracking results show the applicability and the feasibility of the proposed control plan.



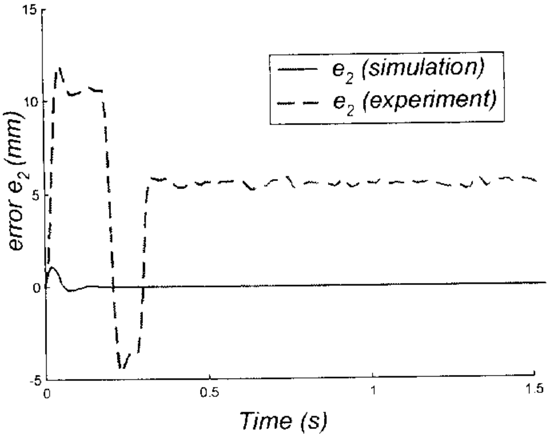
(a) Initial time

(b) Full time

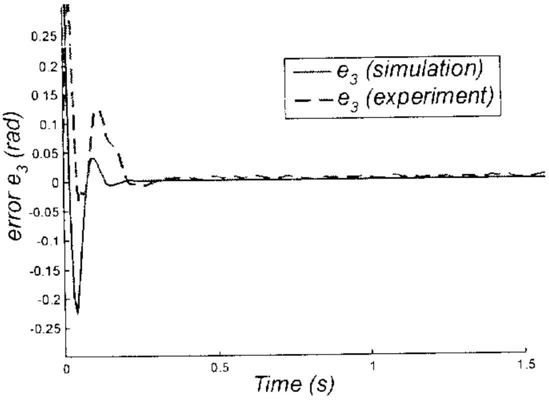
Fig. 7.3 Trajectory tracking of MR



(a) Error e_1



(b) Error e_2



(c) Error e_3

Fig. 7.4 Tracking errors of MR along the path

8

Discussions and Conclusions

A nonholonomic mobile robot control problem and its implementation in an unstructured environment have been addressed in this paper. Generally, a reference path for the MR is subject to formalize MR's performance behavior. Therefore, a path planning process is needed to generate a collision-free path, says, a tracking trajectory. Two of the path planning algorithms, namely, HJB-based algorithm and polynomial path planning approach, are employed. In the first case, by using the HJB-based path planning algorithm, the MR has chance to track the nearly shortest path. In the second case, a smooth trajectory which is generated by the polynomial path planning approach permits the MR to behave well in its performance.

A control system that is composed of software and hardware is designed to drive the MR to follow the trajectory. In this system, a nonlinear trajectory tracking controller which is based on the kinematic/dynamic model of a nonholonomic MR is employed.

In addition, by applying a ceiling-mounted camera system, the work-space is observed. Thus, some useful environmental information can be obtained. Before using, this computer vision system is subject to be calibrated in order to obtain necessary parameters for the mapping process. In using, the interesting image features (point, edge, line, area...) that were extracted from a captured image of the work space can be processed through the inverse mapping process to return the coordinate information. Hence, the environmental information is used as input for the path planning algorithm. Also, the information regarding to the position and orientation of MR are updated frequently and sent to MR controller as control inputs.

The whole control system is realized by integrating a personal computer (Pentium IV, 2.8GHz, 1GB RAM), a USB camera (Logitech 4000) and PIC-based microprocessors. By using wireless communication, the MR can receive controller commands from the PC. This total control system shows good performance and reliability in tracking performance by the simulation and experimental results.

9

Appendices

A. Estimation of the Homography Between the Model Plane and its Image

There are many ways to estimate the homography between the model plane and its image. In this part, a technique based on maximum likelihood criterion is presented. Let M_i and $\mathbf{m}_i$ be the model and image points, respectively. Ideally, they should satisfy Eq. (3.2). In practice, they do not satisfy Eq. (3.2) because of noise in the extracted image points. Let's assume that $\mathbf{m}_i$ is corrupted by Gaussian noise with mean $\mathbf{0}$ and covariance matrix $\mathbf{\Lambda}_{\mathbf{m}_i}$. Then, the maximum likelihood estimation of $\mathbf{H}$ is obtained by minimizing the following functional

$$\sum_i (\mathbf{m}_i - \hat{\mathbf{m}}_i)^T \mathbf{\Lambda}_{\mathbf{m}_i}^{-1} (\mathbf{m}_i - \hat{\mathbf{m}}_i) \qquad (9.1)$$

where $\hat{\mathbf{m}}_i = \frac{1}{\bar{\mathbf{h}}_3^T \mathrm{M}_i} \begin{bmatrix} \bar{\mathbf{h}}_1^T \mathrm{M}_i \\ \bar{\mathbf{h}}_2^T \mathrm{M}_i \end{bmatrix}$ with $\bar{\mathbf{h}}_i$ the i^{th} row of $\mathbf{H}$.

In practice, $\Lambda_{\mathbf{m}_i} = \sigma^2\mathbf{I}$ is assumed for all i. To minimize (9.1), one has to solve a nonlinear least-squares problem. The nonlinear minimization is conducted with the Levenberg-Marquardt Algorithm as implemented in Minpack [42]. This requires an initial guess for the solution, which can be obtained as follows

Let $\mathbf{x} = \left[\bar{\mathbf{h}}_1^T, \bar{\mathbf{h}}_2^T, \bar{\mathbf{h}}_3^T\right]$. Then Eq. (3.2) can be rewritten as

$$\begin{bmatrix} \widetilde{\mathrm{M}}^T & \mathbf{0}^T & -u\widetilde{\mathrm{M}}^T \\ \mathbf{0}^T & \widetilde{\mathrm{M}}^T & -v\widetilde{\mathrm{M}}^T \end{bmatrix} \mathbf{x} = \mathbf{0} \tag{9.2}$$

When n points are given, n set of equations mentioned above are obtained. Therefore, the combination of such n equations can be expressed as

$$\mathbf{Lx} = \mathbf{0} \tag{9.3}$$

where $\mathbf{L}$ is a $2n \times 9$ matrix. As $\mathbf{x}$ is defined up to a scale factor, the solution is well known to be the right singular vector of $\mathbf{L}$ associated with the smallest singular value (or equivalently, the eigenvector of $\mathbf{L}^T\mathbf{L}$ associated with the smallest eigenvalue).

In $\mathbf{L}$, some elements are constant 1, some are in pixels, some are in world coordinates, and some are multiplication of both. This makes $\mathbf{L}$ poorly conditioned numerically. Much better results can be obtained by performing a simple data normalization such as the one proposed in [49], prior to running the above procedure.

B. Extraction of the Intrinsic Parameters from Matrix B

Matrix **B**, as described in Sect. **3.1.4.1**, is estimated up to a scale factor, i.e., $\mathbf{B} = \lambda\mathbf{A}^{-T}\mathbf{A}$ with λ an arbitrary scale. Without difficulty, one can uniquely extract the intrinsic parameters from matrix B.

$$v_0 = (B_{12}B_{13} - B_{11}B_{23})/(B_{11}B_{22} - B_{12}^2) \tag{9.4}$$

$$\lambda = B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})]/B_{11} \tag{9.5}$$

$$\alpha = \sqrt{\lambda / B_{11}} \tag{9.6}$$

$$\beta = \sqrt{\lambda B_{11} /(B_{11}B_{22} - B_{12}^2)} \tag{9.7}$$

$$\gamma = -B_{12}\alpha^2\beta / \lambda \tag{9.8}$$

$$u_0 = \gamma v_0 / \beta - B_{13}\alpha^2 / \lambda \, . \tag{9.9}$$

C. Approximating a 3 x 3 matrix by a Rotation Matrix

The problem considered in this section is to solve the best rotation matrix R to approximate a given 3 x 3 matrix Q. Here, "best" is in the sense of the smallest Frobenius norm of the difference R-Q. That is, the following problem is solved:

$$\min_{\mathbf{R}} \|\mathbf{R}-\mathbf{Q}\|_F^2 \qquad \text{subject to } \mathbf{R}^T\mathbf{R}=\mathbf{I} \tag{9.10}$$

Since

$$\|\mathbf{R}-\mathbf{Q}\|_F^2 = trace\left((\mathbf{R}-\mathbf{Q})^T(\mathbf{R}-\mathbf{Q})\right) = 3 + trace\left(\mathbf{Q}^T\mathbf{Q}\right) - 2trace\left(\mathbf{R}^T\mathbf{Q}\right)$$

problem (9.10) is equivalent to the one of maximizing trace $\left(\mathbf{R}^T\mathbf{Q}\right)$.

Let the singular value decomposition of $\mathbf{Q}$ be $\mathbf{USV}^T$, where $\mathbf{S} = diag(\sigma_1, \sigma_2, \sigma_3)$. If an orthogonal matrix $\mathbf{Z} = \mathbf{V}^T\mathbf{R}^T\mathbf{U}$ is defined, then

$$\begin{aligned} trace\left(\mathbf{R}^T\mathbf{Q}\right) &= trace\left(\mathbf{R}^T\mathbf{USV}^T\right) = trace\left(\mathbf{V}^T\mathbf{R}^T\mathbf{US}\right) \\ &= trace(\mathbf{ZS}) = \sum_{i=1}^{3} z_{ii}\sigma_i \le \sum_{i=1}^{3}\sigma_i \end{aligned} \tag{9.11}$$

It is clear that the maximum is achieved by setting $\mathbf{R} = \mathbf{UV}^T$ because then $\mathbf{Z} = \mathbf{I}$. This gives the solution to Eq. (9.10).

D. Proof for kinematic controller design: proposition 2[7]

Substituting control law (5.3) into Eq. (5.4), a nonlinear differential equation can be obtained

$$\dot{e}_1 = -K_1 e_1 + \omega_r e_2 + K_2 v_r e_2^2 + K_3 v_r e_2 \sin e_3 \tag{9.12}$$

$$\dot{e}_2 = -\omega_r e_1 - K_2 v_r e_1 e_2 - K_3 v_r e_1 \sin e_3 + v_r \sin e_3 \tag{9.13}$$

$$\dot{e}_3 = -K_2 v_r e_2 - K_3 v_r \sin e_3 \tag{9.14}$$

Linearizing Eq. (9.12~9.14) with assumption $\mathbf{e} \cong \mathbf{0}$ yields

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} -K_1 & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & -K_2 v_r & -K_3 v_r \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \tag{9.15}$$

$$\therefore \quad \dot{\mathbf{e}} = \mathbf{A}\mathbf{e} \tag{9.16}$$

$\mathbf{A}(\cdot)$ is clearly continuously differentiable and is bounded. Moreover, according to the Routh-Hurwitz criterion, the real parts of all characteristic roots of A are negative[7]. Therefore, by theorem 2.10 in [50], the system (9.16) is proved asymptotically stable at its origin $\mathbf{e} = \mathbf{0}$.

References

[1] T., T., N., F., T., S., and Hashimoto, M., "An Experimental System for Automatic Guidance of Robot Vehicle, Following the Route Stored in Memory," presented at 11th International Symposium on Industrial Robots, 1981.

[2] Kanayama, Y. and Yuta, S. i., "Vehicle Path Specification by Sequence Straight Lines," *IEEE Journal of Robotics and Automation*, vol. 4, pp. 265-276, 1988.

[3] Crowley, J. L., "Asynchronous Control of Orientation and Displacement in a Robotic Vehicle," presented at IEEE Conference on Robotics and Automation, 1989.

[4] Singh, S. and Shin, D. H., "Position Based Path Tracking for Wheeled Mobile Robots," presented at IEEE International Workshop on Intelligent Robots and Systems, Tsukuba, Japan, 1989.

[5] Kanayama, Y., Nilipour, A., and Lelm, C., "A Locomotion Control Method for Autonomous Vehicles," presented at IEEE Conference on Robotics and Automation, 1988.

[6] Nelson, W. and Cox, I., "Local Path Control for an Autonomous Vehicles," IEEE Conference on Robotics and Automation, 1988.

[7] Kanayama, Y., Kimura, Y., Miyazaki, F., and Noguchi, T., "A Stable Tracking Control Method for an Autonomous Mobile Robot," presented at IEEE Int. Conf. on Robotics and Auto., 1990.

[8] Bui, T. H., Chung, T. L., Nguyen, T. T., and Kim, S. B., "Adaptive Tracking Control of Two-Wheeled Welding Mobile Robot with Smooth Curved Welding Path," *KSME International Journal*, vol. 17, pp. 1682-1692, 2003.

[9] Kim, M.-S., Shin, J.-H., Hong, S.-G., and Lee, J.-J., "Designing a Robust Adaptive Dynamic Controller for Nonholonomic Mobile Robots under Modeling Uncertainty and Disturbances," *Mechatronics*, vol. 13, pp. 507-519, 2003.

[10] Pourboghrat, F. and Karlsson, M. P., "Adaptive Control of Dynamic Mobile Robots with Nonholonomic Constraints," *Computers & Electrical Engineering*, vol. 28, pp. 241-253, 2002.

[11] Fierro, R. and Lewis, F. L., "Control of a Nonholonomic Mobile Robot: Backstepping Kinematics into Dynamics," presented at 34th Conf. Deci. Ctrl., New Orleans, LA, 1995.

[12] Canudas de Wit, C., Khennouf, H., Samson, C., and Sordalen, O. J., "Nonlinear Control Design for Mobile Robots," in *Recent Trend in Mobile Robots*, Y. F. Zheng, Ed.: World Scientific, 1993, pp. 121-156.

[13] Hoover, A. and Olsen, B. D., "Path Planning for Mobile Robots Using a Video Camera Network," presented at IEEE/ASME Int. Conf. Adv. Intell. Mecha., Atlanta, USA, 1999.

[14] Sundar, S. and Shiller, Z., "Optimal Obstacle Avoidance Based on the Hamilton-Jacobi-Bellman Equation," *IEEE Trans. Robot. Automat.*, vol. 13, pp. 305-310, 1997.

[15] Connoly, C. I., Burns, J. B., and Weiss, R., "Path Planning Using Laplace's Equation," presented at IEEE Int. Conf. Robot. Automat., 1991.

[16] Beom, H. R. and Cho, H. S., "Path Planning for Mobile Robot Using Skeleton of Free Space," *Control Engineering Practice*, vol. 1, pp. 355-359, 1993.

[17] Rimon, E. and Koditschek, D. E., "Exact Robot Navigation Using Artificial Potential Functions," *IEEE Trans. Robot. Automat.*, vol. 8, pp. 501-518, 1992.

[18] Moskalenko, A. I., "Bellman Equations for Optimal Processes with Constraints on the Phase Coordinates," *A Translation Avtomatika I Telemekhanika Autom. Remote. Cont.*, vol. 4, pp. 1853~1864, 1967.

[19] Sallaberger, C. S. and D'Eleuterio, G. M. T., "Optimal Robotic Path Planning Using Dynamic Programming and Randomization*1," *Acta Astronautica*, vol. 35, pp. 143-156, 1995.

[20] Ohya, A., Kosaka, A., and Kak, A., "Vision-Based Navigation by a Mobile Robot with Obstacle Avoidance Using Single-Camera Vision and Ultrasonic Sensing," *IEEE Trans. on Robo. and Auto.*, vol. 14, pp. 969-978, 1998.

[21] Shimizu, S., Kato, T., Ocmula, Y., and Suematu, R., "Wide Angle Vision Sensor with Fovea-Navigation of Mobile Robot Based on Cooperation between Central Vision and Peripheral Vision," presented at Int. Conf. on Intell. Robot. and Sys., Maui, Hawaii, USA, 2001.

[22] Kim, Y.-J., Kim, J.-H., and Kwon, D.-S., "Evolutionary Programming-Based Univector Field Navigation Method for Fast Mobile Robots," *IEEE Trans. on Sys. Man. Cybern.-Part B: Cybern.*, vol. 31, pp. 450-458, 2001.

[23] Miyazaki, Y., Ohya, A., and Yuta, S. i., "Obstacle Avoidance Behavior of Autonomous Mobile Robot Using Fiber Grating Vision Sensor," 2000.

[24] Lee, W.-H., Roh, K.-S., and Kweon, I.-S., "Self-Localization of a Mobile Robot without Camera Calibration Using Projective Invariants," *Pattern Recognition Letters*, vol. 21, pp. 45-60, 2000.

[25] Achour, K. and Benkhelif, M., "A New Approach to 3d Reconstruction without Camera Calibration," *Pattern Recognition*, vol. 34, pp. 2467-2476, 2001.

[26] Dixon, W. E., Dawson, D. M., and Zergeroglu, E., "Adaptive Tracking Control of a Wheeled Mobile Robot Via an Uncalibrated Camera System," *IEEE Trans. on Sys. Man. Cybern.*, vol. 31, pp. 341-352, 2001.

[27] Stella, E. and Distante, A., "Self-Location of a Mobile Robot by Estimation of Camera Parameters," *Robotics and Autonomous Systems*, vol. 15, pp. 179-187, 1995.

[28] Klancar, G., Kristan, M., and Karba, R., "Wide-Angle Camera Distortions and Non-Uniform Illumination in Mobile Robot Tracking," *Robotics and Autonomous Systems*, vol. 46, pp. 125-133, 2004.

[29] Borenstein, J. and Koren, Y., "The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots," *IEEE Trans. on Robot. and Auto.*, vol. 7, 1991.

[30] Borenstein, J. and Koren, Y., "Real-Time Obstacle Avoidance for Fast Mobile Robots," *IEEE Trans. on Sys. Man. Cybern.*, vol. 19, No. 5, 1989.

[31] Meyer, J.-A. and Filliat, D., "Map-Based Navigation in Mobile Robots:: Ii. A Review of Map-Learning and Path-Planning Strategies," *Cognitive Systems Research*, vol. 4, pp. 283-317, 2003.

[32] Nourbakhsh, I. R., Andre, D., Tomasi, C., and Genesereth, M. R., "Mobile Robot Obstacle Avoidance Via Depth from Focus*1," *Robotics and Autonomous Systems*, vol. 22, pp. 151-158, 1997.

[33] Heikkila, J. and Silven, O., "A Four-Step Camera Calibration Procedure with Implicit Image Correction," presented at IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 1997.

[34] Yun, X. and Yamamoto, Y., "Internal Dynamics of a Wheeled Mobile Robot," presented at IEEE/RSJ Int. Conf. on Intell. Robots and Syst., Yokohama, Japan, 1993.

[35] Kozlowski, K. and Majchrzak, J., "A Backstepping Approach to Control a Nonholonomic Mobile Robot," presented at IEEE Int. Conf. on Robo. & Auto., Washington DC, 2002.

[36] Fukao, T., Nakagawa, H., and Adachi, N., "Adaptive Tracking Control of a Nonholonomic Mobile Robot," *IEEE Trans. Robot. Automat.*, vol. 16, pp. 609-615, 2000.

[37] Reinhardt, M. R., *Analytical Dynamics of Discrete Systems*. New York: Plenum Press, 1977.

[38] Heikkila, J., "Geometric Camera Calibration Using Circular Control Points," *IEEE Trans. on Pattern Analys. and Machine Intell.*, vol. 22, pp. 1066-1077, 2000.

[39] Zhang, Z., "A Flexible New Technique for Camera Calibration," *IEEE Trans. on Pattern Analys. and Machine Intell.*, vol. 22, pp. 1330-1334, 2000.

[40] Luong, Q.-T. and Faugeras, O. D., "Self-Calibration of a Moving Camera from Point Correspondences and Fundamental Matrices," *The International Journal of Computer Vision*, vol. 1, pp. 5-40, 1997.

[41] Faugeras, O., *Three-Dimension Computer Vision: A Geometric Viewpoint*: MIT Press, 1993.

[42] More, J., "The Levenberg-Marquardt Algorithm, Implementation and Theory," in *Numerical Analysys, Lecture Notes in Mathematics*, G. A. Watson, Ed.: Springer-Verlag, 1997, pp. 630.

[43] Zhang, Z., "A Flexible New Technique for Camera Calibration," Microsoft Research, Microsoft Corporation, One Microsoft Way Redmond, WA 98052 MSR-TR-98-71, Aug. 10, 2002 2002.

[44] Bouguet, J.-Y., "Camera Calibration Toolbox for Matlab," Intel 2001.

[45] Olsen, B. D. and Hoover, A., "Mobile Robot Navigation Using an Environment-Based Video Camera," 2000.

[46] Cesari, L., *Optimization-Theory and Applications: Problems with Ordinary Differential Equations*. New York: Springer-Verlag, 1983.

[47] Campion, G., d'Andrea-Novel, B., and Bastin, G., "Controllability and State Feedback Stabilizability of Nonholonomic Mechanical Systems," in *Lecture Notes in Control and Information Science*, C. Canudas de Wit, Ed.: Springer-Verlag, 1991, pp. 106-124.

[48] Campion, G., d'Andrea-Novel, B., and Bastin, G., "Modeling and State Feedback Control of Nonholonomic Mechanical Systems," presented at IEEE Conference on Decision and Control, Brighton, England, 1991.

[49] Hartley, R., "In Defence of the 8-Point Algorithm," presented at 5th International Conference on Computer Vision, Boston, MA, 1995.

[50] Furuta, K., Sano, A., and Atherton, D., *State Variable Methods in Automatic Control*: John Wiley & Son, 1988.

Publications and Conferences

A. Publications

[1] Anh Kim Tran, "Structure Design using Finite Element Method," Bachelor of Science Graduation Thesis, Ho Chi Minh University of Technology, Mechanical Engineering Department, Viet nam, 1998.

[2] Anh Kim Tran, Kwang Ju Kim, Tan Lam Chung, Hak Kyeong Kim, and Sang Bong Kim, "Path Design Method of a Mobile Robot for Obstacle Avoidance Using Ceilling-mounted Camera System and its Implementation," Journal of the Korean Society of Precision Engineering, Vol. 21, No. 08, 2004 (accepted for publishing on August, 2004).

[3] Anh Kim Tran, "A Study on an Obstacle Avoidance Mobile Robot Using a Ceiling-mounted Camera System," Master of Science Graduation Thesis, Pukyong National University, Department of Mechanical Engineering, Graduate School, Pusan, Korea, 2004, submitted.

B. Conferences

[1] Anh Kim Tran, Jin Ho Suh, Kwang Ju Kim, and Sang Bong Kim, "Study on Path Planning Algorithm of a Mobile Robot for Obstacle Avoidance using Optimal Design Method," *The Korean Society for Power System Engineering, Domestic Symposium*, Kunsan, Korea, April, 2003.

[2] Anh Kim Tran, Jin Ho Suh, Kwang Ju Kim, and Sang Bong Kim, "Path Planning Algorithm for Obstacle Avoidance of a Mobile Robot-An Optimal Design Method," *The 2003 International Symposium on Mechatronics*, Ho Chi Minh, Viet Nam. September, 2003.

[3] Anh Kim Tran, Jin Ho Suh, Kwang Ju Kim, and Sang Bong Kim, "Study on Path Planning Algorithm of A Mobile Robot for Obstacle Avoidance Using Optimal Design Method," *ICCAS 2003, Proceedings of the 2003 International Conference on Control, Automation and Systems*, Gyeongju, Korea, October 22-25, 2003.

[4] Anh Kim Tran, Kwang Ju Kim, Jin Ho Suh, and Sang Bong Kim, "Study on Obstacle Avoidance Path Planning Approaches for A Nonholonomic Mobile Robot-An Optimal Time Oriented Passage Designing Scheme," *KSPSE 2003, Proceedings of the International Symposium on Advanced Engineering,* Pusan, Korea, November 28, 2003.

[5] Anh Kim Tran, Kwang Ju Kim, Hak Kyeong Kim, Tan Lam Chung, and Sang Bong Kim, "Realization of an Obstacle Avoidance Mobile Robot Using a Ceilling-mounted Camera System," *The 2004 International Symposium on Advance Science and Engineering,* Ho Chi Minh, Viet Nam, May 19-23, 2004.

[6] Anh Kim Tran, Kwang Ju Kim, Hak Kyeong Kim, Tan Lam Chung, and Sang Bong Kim, "Obstacle Avoidance of a Mobile Robot using a Calibrated Computer Vision System," *The Korea Society of Power Engineering, Domestic Symposium*, Mokpo, Korea, May 28~29, 2004.

[7] Anh Kim Tran, Kwang Ju Kim, Tan Lam Chung, Hak Kyeong Kim, and Sang Bong Kim, "Implementation of an Obstacle Avoidance Mobile Robot using a Ceilling-mounted Camera System," *2004 International Conference on Dynamics, Instrumentation and Control,* Nanjing, China, August 18-20, 2004.