Path Tracking Laws and Implementation of a Vision -Based Wheeled Mobile Robot

YING-SHING SHIAO, JUI-LIANG YANG, DING-TSAIR SU
Department of Electrical Engineering
National Chang-hua University of Education
Chang-hua 520 Taiwan
REPUBLIC OF CHINA.(TAIWAN)
shiaoing@cc.ncue.edu.tw, y5111@mail.elvs.chc.edu.tw, sdt@cc.ctu.edu.tw
http://www.ncue.edu.tw

Abstract: - This paper has studied techniques for building system configuration, control architecture and implementation of a vision-based wheeled mobile robot. The completed WMR has been built with the dead-reckoning method so as to determine the vehicle's velocity and posture by the numerical differentiation/integration over short travelling. The developed PID controllers show good transient performances, that is, velocity of right and left wheels can track the commands quickly and correctly. Moreover, the path tracking control laws have been also executed in the DSP-based controller in the WMR. The image-recognized system can obtain motion information 15 [frames/sec] by using H-S model, which is one of the well-known color detection methods. The better performance a vision system has, the more successful the control laws design. The WMR obtains its posture from the dead-reckoning device together with the vision system. Finally, we integrate these subsystems and complete the operators of the whole system. This complementing wheeled mobile robot system can be thought of as a platform for testing various tracking control laws and signal filtering method. To solve the problem of position/orientation tracking control of the WMR, two kinematical optimal nonlinear predictive control laws are developed to manipulate the vehicle to asymptotically follow the desired trajecto ries. A Kalman filtering scheme is used to reduce the bad effect of the imagine nose, thereby improving the accuracy of pose estimation. The experimental system is composed of a wireless RS232 modem, a DSP -based controller for the wheeled mobile robot and a vision system with a host computer. A computation-effective and high-performance DSP-based controller is constructed for executing the developed sophisticated path tracking laws. Simulation and experimental results are included to illustrate the feasibil ity and effectiveness of the proposed control laws.

Key-Words: - Predictive control; Mobile robot; Path tracking; Nonlinear characteristics, Kalman filtering

1 Introduction

This paper is devoted to developing methodologies and techniques for control architecture design, path tracking laws and implementation of a vision-based wheeled mobile robot (WMR). This study has been mainly excited by a wide variety of practical mobile robots applications due to their ability to work in various domains. WMRs have already gained widespread applications, such as planetary exploration, materials transportation, military tasks, manufacturing servicing, hazardous environment and mine excavation. To achieve the aforementioned tasks, the WMR requires sensing of the environments, intelligent trajectory planning, navigation and path tracking control. This desired autonomous or intelligent behavior has motivated an intensive research over the past decades.

To achieve path-tracking control for the WMRs, many sophisticated control approaches have been investigated by several researchers. The existing tracking control methods for the WMR can be classified into five categories: (i.) sliding mode [1]; (ii.) nonlinear control [2-3]; (iii.) fuzzy control [4]; (iv.) neural network control [5]; and (v.) adaptive backstepping control [6]. In 2007, Qiuling et al. used sliding mode control for tracking control, which is complicated and computationally expensive. The generated velocity command with respect to time is not a smooth curve in [1]. Lei et al. introduced the fuzzy tracking control approaches [4] in 2006. But it is very difficult to formulate the fuzzy rules, which are usually obtained from the trial-and-error procedure. In 2006, the computational expensive neural network was adopted by Heinen et al. [5]. The algorithm requires on-line learning in order to make the robot perform properly. Design of the path following controller for gyrobot, a gyroscopically stabilized single-wheeled robot, is presented [7].

In this paper, we introduced two optimal nonlinear predictive control approaches [8-9] for manipulators. The control laws minimized a quadratic performance index of the state predicted tracking error. The algorithms were shown to improve tracking accuracy of the manipulators.

In addition, Kalman filter police has been proposed in [10-11]. In this paper, we extend the Kalman filter method to deal with the pose estimation problem of the WMR with corrupted imaging noise.

This paper is organized as follows. Camera-Space Tracking Control is presented in Section 2, and section 3 introductions kinematic model and kinematic tracking control design. Section 4 aims at developing two optimal nonlinear predictive control approaches. Section 5, an extended Kalman filtering scheme is adopted to filter out the corrupted noise in the images. In Section 6 simulation and experimental results are presented. Section 7 concludes this paper.

2 Camera-Space Tracking Control

2.1 Vision-based Control System

The control objective for the vision-based tracking problem is to manipulate the WMR to follow the desired trajectory. This system is composed of a ceil-mounted fixed camera whose outputs are connected to a host computer, and a WMR with two atop different color, round marks (in Fig.1).

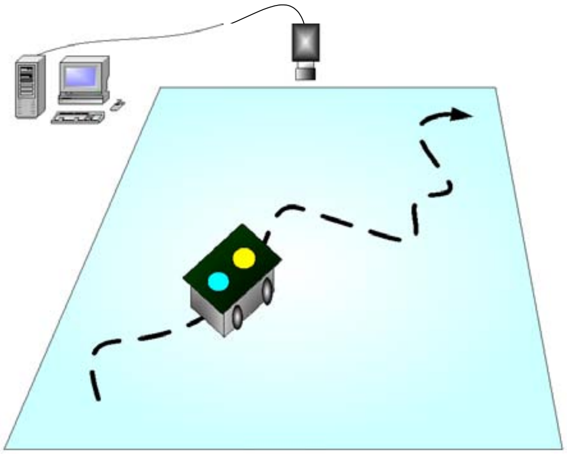


Fig. 1 Camera system configuration

The host computer is used to periodically provide the position and orientation information of the WMR for tracking the reference trajectory, via the wireless RS232 modem. The operation of such a control system can be easily understood from the system block diagram shown in Fig.2.

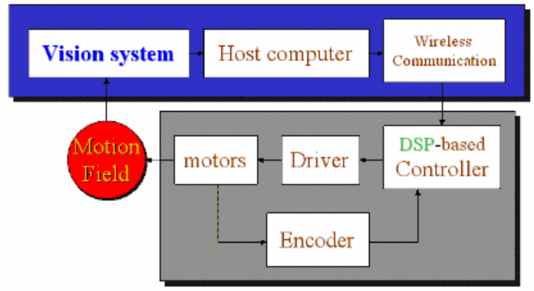


Fig.2 A block diagram of the vision-based control system

The digital DSP-based controller has been designed with multi-loop characteristics. Fig.3 details the DSP-Based control system with inner P current control loop, encoder-based PID motor velocity control and dead-reckoning-based path tracing control. The very structure of the DSP-based controller is very useful in proving high-performance operation and control for the WMR.

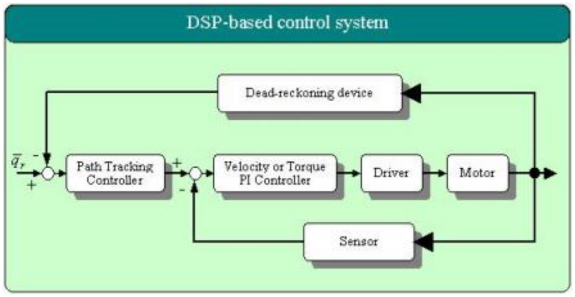


Fig. 3 DSP-based control system

2.2 PID Velocity Controller Design

In the following design, the wheeled mobile robot employs the two DC servo motors with two optical encoders. The digital optical encoder generates 16 QEP pulses per revolution, and the motors have the maximum velocity of MAX(w)=482 [rpm].

Fig.4 shows the chassis of the WMR, where the length of the driving axis between the wheels is L_0 = 15[cm], and the radius of the wheels is γ_0 = 2.7056[cm]. Hence we can calculate the maximum linear and angular velocities of the WMR, as below

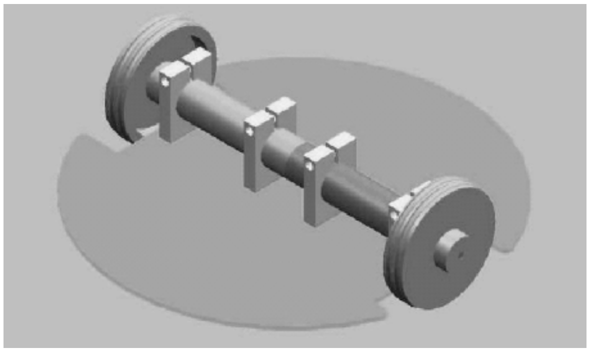


Fig. 4 The chassis of wheeled mobile robot

$$\upsilon = \frac{V_R + V_L}{2} = 20.13 \qquad [\text{cm/s}]$$

$$\omega = \frac{V_R - V_L}{L_0 / 2} = 5.368 \qquad [\text{rad/s}] \qquad (1)$$

The DSP-based controller demodulates the modem signals sent from the host computer, and obtains the position and orientation of the WMR. After performing the tracking control algorithm, with the sampling period of 20 [ms], the DSP-based controller outputs the following speed commands to the two driving wheels

$$V_R = {2\upsilon + \frac{L_0}{2}\omega} \Big/ 2$$

$$V_L = {2\upsilon - \frac{L_0}{2}\omega} \Big/ 2 \qquad (2)$$

Therefore, with the speed commands, the velocities of the right and left wheels of the WMR are regulated by the PID control laws, whose sampling period is 1 [ms].

2.3 Dead-reckoned Posture Estimation

The output signals of the encoder are directly connected to DSP. Counting the pulse numbers from the encoders composes the dead-reckoning device. Thus, we have

$$\theta_L = \frac{N_L}{64} \cdot 2\pi \qquad \theta_R = \frac{N_R}{64} \cdot 2\pi \qquad (3)$$

$$\Delta d_L = \theta_L \cdot R \qquad \Delta d_R = \theta_R \cdot r_0 \qquad (4)$$

The (r_0) is the mentioned radius of the wheels. Hence we can know the traveling distances of the right and left wheels, Δd_L and Δd_R . The orientation difference, $\Delta\theta$ and the traveling distance Δd can be employed to find the subsequent position and orientation of the vehicle. Thus, we have

$$\Delta\theta = \frac{\Delta d_R - \Delta d_L}{2L_0} \qquad \Delta d = (\Delta d_L + \Delta d_R)/2$$
$$\theta(k) = \theta(k-1) + \Delta\theta$$
$$x(k) = x(k-1) + \Delta d \cdot \cos\theta(k)$$
$$y(k) = y(k-1) + \Delta d \cdot \sin\theta(k) \qquad (5)$$

2.4 Accomplished WMR and Controller's Circuitry

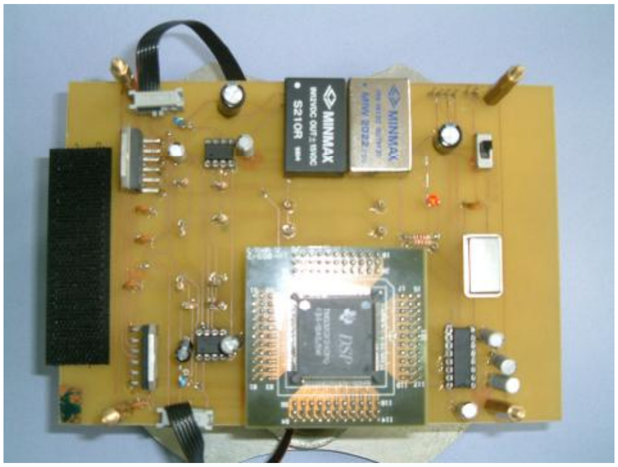


Fig. 5 Physical view of the integrated DSP-based control board

Fig.5 shows the completed DSP-based controller on a print circuit board (PCB). Fig.6 depicts a recent picture of the wheeled

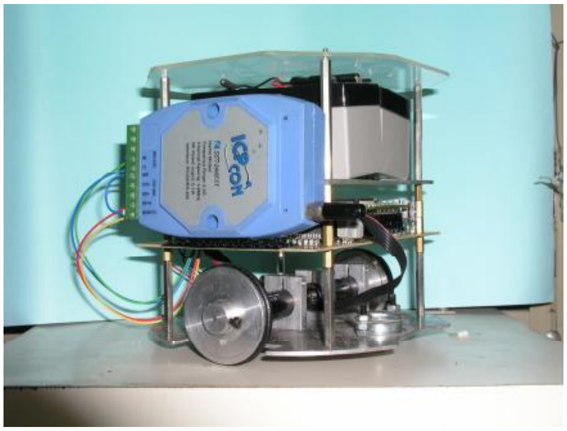


Fig. 6 A recent picture of the wheeled mobile robot

2.5 Vision System

To achieve quick and smooth motion of the robot, the vision servoing system plays an important key. Here we use a color CCD-based vision system to localize the position and orientation of the WMR.

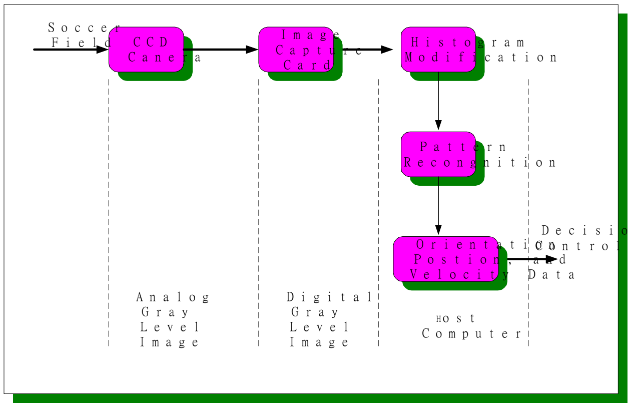


Fig.7 The operational procedure of the vision system

A camera mounted on the ceiling has a height of 3[m] above the robot workspace in order to continuously capture the motion images of the robot into the host computer through the image-capturing card. Fig 7 shows the operational procedure of the vision system. Before obtaining the positions of the color marks, we must be to check and calibrate the image distortion caused by the lens effect. Two kinds of distortions caused by the lens are illustrated in Fig. 8. Since not all parts of the lens can match the ideal pin-hole model, the image magnification factors should be compensated according to the distortion condition.

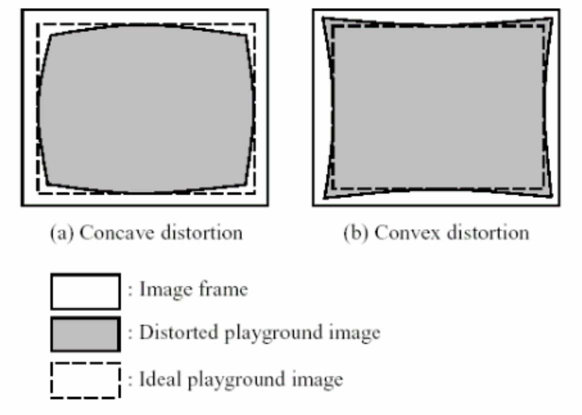


Fig. 8 Image distortions caused by the lens effects

From Fig. 8(a), we observe that the distortion type of the camera in our vision system belongs to the type of concave distortion. In Fig.9, to calibrate such a distortion, the following equation canceling the effects is expressed by

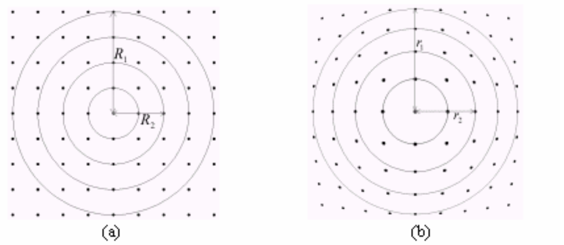


Fig. 9 Concave distortion effects (a) in the task-space, (b) in the camera-space

$$R = P \times r^2 + Q \times r \tag{6}$$

where R is the distance in the task-space, r is the distance in the camera-space, and the Parameters P and Q are two ratios. By finding out the relationship between (R1, R2) in the task-space and (r1, r2) in the camera-space, we obtain the following equation to computer the values of P ad Q,

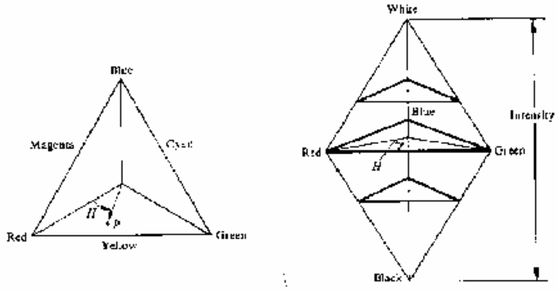


Fig. 10 HIS model

$$\begin{bmatrix} P \\ Q \end{bmatrix} = inv\left(\begin{bmatrix} r_1^2 & r_1 \\ r_2^2 & r_2 \end{bmatrix}\right) \times \begin{bmatrix} R_1 \\ R_2 \end{bmatrix} \tag{7}$$

Object recognition in a vision system can be achieved using edge detection or a gray level detection method. Here two different color patterns mounted on the top plate of the robot are adopted to estimate the WMR position/orientation. Aside from the previous detection approaches, an alternative to pattern recognition is a color detection method. There are several color models for image processes, such as RGB, HIS, YUV and so on. The RGB values of each pixel in the image can be grabbed utilizing MIL (Matrox Imaging Library programming) in the vision system. Although the pattern recognition can be completed via the RGB model, we choose HIS color model (see Fig. 10) as the basis of the recognition in contrast to the unstable behavior of the RGB model in each pixel.

The transform function of the RGB model to HIS model is described as follows:

$$I = \frac{1}{3}(R+G+B)$$
$$S = 1 - \frac{3}{(R+G+B)}[\min(R,G,B)] \quad (8)$$
$$H = \cos^{-1}\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{1/2}}\}$$

2.6 Concluding Remarks

This paper has studied techniques for building system configuration, control architecture and implementation of a vision-based wheeled mobile robot. The completed WMR has been built with the dead-reckoning method so as to determine the vehicle's velocity and posture by the numerical differentiation/integration over short traveling. The developed PID controllers show good transient performances, that is, velocity of right and left wheels can track the commands quickly and correctly. Moreover, the path tracking control laws have been also executed in the DSP-based controller in the WMR. The image-recognized system can obtain motion information 15 [frames/sec] by using H-S model, which is one of the well-known color detection methods. The better performance a vision system has, the more successful the control laws design. The WMR obtains its posture from the dead-reckoning device together with the vision system. Finally, we integrate these subsystems and complete the operators of the whole system. This complementing wheeled mobile robot system can be thought of as a platform for testing various tracking control laws and signal filtering method.

3 Preliminary Background

3.1 Kinematic Model

Consider a nonholonomic WMR under the assumption of pure rolling and non-slipping. Its kinematic model is given by

$$\dot{q} = S(q)\cdot u \quad (9)$$

Where $q(t), \dot{q}(t) \in \Re^3$ are defined by

$$q = \begin{bmatrix} x_c & y_c & \theta \end{bmatrix} \quad \dot{q} = \begin{bmatrix} \dot{x}_c & \dot{y}_c & \dot{\theta} \end{bmatrix}^T \quad (10)$$

$x_c(t)$, $y_c(t)$ and $\theta(t)$ represent the position/orientation of the WMR in Fig.11 respectively, the matrix $S(\cdot) \in \Re^{3\times 2}$ is defined as follows

$$S(q) = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \quad (11)$$

And the velocity vector $u(t) \in \Re^2$ is denoted by

$$u = \begin{bmatrix} v_1 & v_2 \end{bmatrix}^T = \begin{bmatrix} v_l & \dot{\theta} \end{bmatrix}^T \quad (12)$$

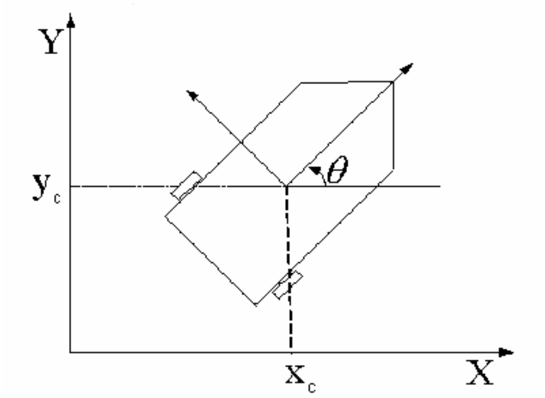


Fig. 11 Wheeled mobile robot

We defined one another Cartesian position/orientation in the camera space by $\bar{q} = \begin{bmatrix} \bar{x}_c & \bar{y}_c & \bar{\theta} \end{bmatrix}^T$, and thus, the kinematic model in the camera-space takes the same form

$$\dot{\bar{q}} = S(\bar{q})\cdot\bar{u} \quad (13)$$

Where the velocity vector $\bar{u}(t) = \begin{bmatrix} \bar{v}_1 & \bar{v}_2 \end{bmatrix}^T \in \Re^2$ represent the linear and angular velocities in the camera-space.

In order to control the WMR in the c amera-space to track a camera-space desired trajectory (i.e., $\bar{x}_{rc}$, $\bar{y}_{rc}$ and $\bar{\theta}_r$), task-camera space transformations is required such that the proposed velocity vector in the task-space is able to effectively and correctly drive WMR in the camera–space.

Instead of generating the reference trajectory in the task-space using the vision system, we specify the desired velocity vector $\bar{u}_{rc}$ in the camera-space such that the reference trajectory can be produced in the camera-space via the following expression

$$\dot{\bar{q}} = S(\bar{q}_r)\cdot\bar{u}_r \quad (14)$$

3.2 Kinematic Tracking Control Design

To develop the control law, we have to find out the open loop error system in terms of $\tilde{x}(t)$, $\tilde{y}(t)$, and $\tilde{\theta}(t) \in \Re^1$ which denote the differences between the camera-space Cartesian posture and the desired posture by

$$\tilde{x} = \bar{y}_{rc} - \bar{y}_c \, , \tilde{y} = \bar{y}_{rc} - \bar{y}_c \, , \tilde{\theta} = \bar{\theta}_r - \bar{\theta} \qquad (15)$$

To facilitate the control design process, we employ a well-known global invertible transformation between $\tilde{x}(t)$, $\tilde{y}(t)$ and $\tilde{\theta}(t)$ the auxiliary error signal.

$$e(t) = \begin{bmatrix} e_1(t) & e_2(t) & e_3(t) \end{bmatrix}^T \in \Re^3$$

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\bar{\theta} & \sin\bar{\theta} & 0 \\ -\sin\bar{\theta} & \cos\bar{\theta} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{\theta} \end{bmatrix} \qquad (16)$$

We observe that $e_1(t), e_2(t), e_3(t)$ denotes the errors in the tangent direction, normal direction and orientation, respectively. With the invariability of transformation in Eq. (13), it is easy to prove that $\lim\limits_{x\to\infty} e_1(t), e_2(t), e_3(t) = 0 \Leftrightarrow \lim\limits_{x\to\infty} \tilde{x}(t), \tilde{y}(t), \tilde{\theta}(t) = 0$, by considering

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} \bar{v}_1 \\ \bar{v}_2 \end{bmatrix} + \begin{bmatrix} \bar{v}_{r1} \cos e_3 \\ \bar{v}_{r1} \sin e_3 \\ \bar{v}_{r2} \end{bmatrix} \qquad (17)$$

Let

$$P = \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & -1 \end{bmatrix}, u = \begin{bmatrix} \bar{v}_1 \\ \bar{v}_2 \end{bmatrix}, \Pi = \begin{bmatrix} \bar{v}_{r1} \cos e_3 \\ \bar{v}_{r1} \sin e_3 \\ \bar{v}_{r2} \end{bmatrix} \qquad (18)$$

4. Optimal Nonlinear Predictive Control

In this section we present kinematically tracking controllers based on minimization of the predictive state tracking errors. The two nonlinear predictive control schemes are developed to allow the vehicles position and orientation tracking of a desired reference trajectory.

4.1 A fixed End Point Predictive Controller

We improve tracking accuracy at next instant $(t+h)$, where h is a small time increment. That is, the tracking error is defined as:

$$e(t+h) \cong e(t) + h \cdot \dot{e}(t) \qquad (19)$$

$$\cong e(t) + h \cdot (p \cdot \bar{u} + \Pi) \qquad (20)$$

In order to find the control vector $u(t)$ that improves tracking error at next instant, we consider a dynamic performance index that penalizes the predictive tracking error and control efforts.

$$J_1 = \frac{1}{2} e^T (t+h) \cdot Q \cdot e(t+h) + \frac{1}{2} \bar{u}^T \cdot R \cdot \bar{u} \qquad (21)$$

$$\bar{u} = \begin{bmatrix} \bar{v}_1 \\ \bar{v}_2 \end{bmatrix}, Q = Q^T \geq 0, R = R^T > 0 \qquad (22)$$

where $Q = Q^T \geq 0 \in R^{3\times3}$ is a semi positive-definite matrix and $R = R^T > 0 \in R^{2\times2}$ is a positive definite matrix. The minimization of J_1 with respect to $u(t)$ yields:

$$\frac{\partial J_1}{\partial \bar{u}} = (\frac{\partial e}{\partial \bar{u}})^T \cdot Q \cdot e(t+h) + R \cdot \bar{u}^* = 0 \qquad (23)$$

Therefore, we have the optimal control vector as

$$\bar{u}^*(t) = -(P^T \cdot Q \cdot P)^{-1} \cdot P^T \cdot Q \cdot (h^{-1} \cdot e(t) + \Pi) = \begin{bmatrix} \bar{v}_1 \\ \bar{v}_2 \end{bmatrix} \qquad (24)$$

4.2 A Finite Horizon Nonlinear Predictive Control

Like the previous case, the tracking error at next instant $(t+T)$ is approximated by the following equation:

$$e(t+T) \cong e(t) + T \cdot \dot{e}(t) \cong e(t) + T \cdot (P \cdot \bar{u} + \Pi) \qquad (25)$$

The control goal is to find the best control vector $\bar{u}(t)$ so as to minimize the following nonsingular quadratic cost function:

$$J_2 = \frac{1}{2} \int_0^{h_c} e(t+T)^T \cdot Q \cdot e(t+T) dT + \frac{1}{2} \int_0^{h_c} \bar{u}^T (t+T) \cdot R \cdot \bar{u} \qquad (26)$$

where h_c is the predicted control signal horizon, $Q = Q^T \geq 0 \in R^{3\times3}$ is a positive semi-definite matrix and $R = R^T > 0 \in R^{2\times2}$ is a positive definite matrix. The minimization of J_2 with respect to $\bar{u}(t)$ yields:

$$\frac{\partial J_2}{\partial \bar{u}} = 0 \qquad (27)$$

$$\bar{u}^* = -(\frac{1}{2} h^2 P^T Q P + R)^{-1} \cdot (\frac{1}{2} h^2 P^T Q \Pi - P^T Q e(t)) \qquad (28)$$

5 Pose Estimation Using Kalman Filtering

5.1 Introduction

Filtering noise in real-time image sequences is required in many applications like visual servoing. There are also lots of examples in engineering where filtering is desirable. Kalman filter has been applied in areas as diverse as aerospace, marine navigation, nuclear power plant instrumentation, demographic modeling, manufacturing, and many others. A good filtering algorithm can remove measurement noise from measurements.

In practice, the individual state variables of a dynamic system may not be determined exactly by direct measurements. In contrast, we usually find that the measurements are corrupted by random noise. The system itself may also be subjected to random disturbances. It is then required to estimate the state variables from the noisy observations. In 1960, Kalman described a Kalman filtering scheme recursively find solutions to the discrete-data linear filtering problems. Fig.12 and Fig.13 illustrate the basic flowchart and calculation modules of the Kalman filter design. The main purpose of this chapter is to use Kalman filter to develop a pose estimate of the WMR.

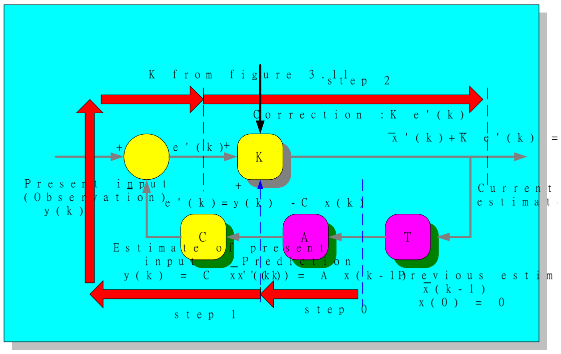


Fig.12 Computation steps in Kalman filter

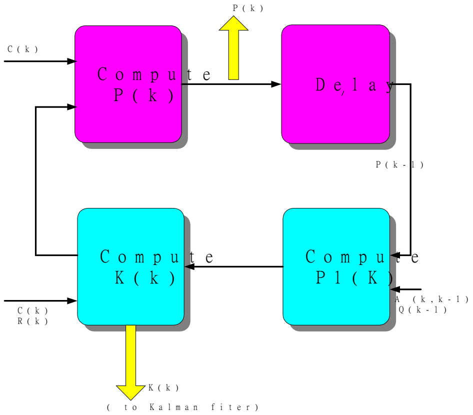


Fig.13 Subroutine calculations for Kalman filter

5.2 Kalman Filtering Method

Under the assumptions of no slipping and pure rolling, the dynamic behavior of the WMR can be described by the following discrete time state equation

$$X(k)=\begin{bmatrix}x(k)\\y(k)\\\theta(k)\end{bmatrix}=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\cdot\begin{bmatrix}x(k-1)\\y(k-1)\\\theta(k-1)\end{bmatrix}+\begin{bmatrix}T\cos\theta(k-1)&0\\T\sin\theta(k-1)&0\\0&T\end{bmatrix}\cdot\begin{bmatrix}v_1(k-1)\\v_2(k-1)\end{bmatrix}+\begin{bmatrix}w_1(k-1)\\w_2(k-1)\\w_3(k-1)\end{bmatrix}$$

$$=A(k-1)\cdot X(k-1)+B(k-1)\cdot u(k-1)+w(k-1) \quad (29)$$

where $w=(k+1)$ is a random process noise vector and T is the sampling period. Furthermore the measurement at time k can be denoted by $Z_k = P_k + \xi_k$ where ξ_k is random measurement noise, that is,

$$Z(k)=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\cdot\begin{bmatrix}x(k)\\y(k)\\\theta(k)\end{bmatrix}+\xi(k)$$

$$=C(k)\cdot X(k)+\xi(k) \quad (30)$$

The Kalman filter is formulated as follows. We assume that the process noise w_k and the measurement noise $\xi(k)$ are two white Gaussian noise vectors with the covariance matrix Q and R, respectively

$$E[w(k)\cdot w(k)^T]=Q(k) \quad (31)$$

$$E[\xi(k)\cdot\xi(k)^T]=R(k) \quad (32)$$

We propose a pose estimation algorithm for the WMR with the following two criteria:

The estimate value is finally equal to the true value of the state i.e., $\hat{x}\to x$ as $t\to\infty$. The algorithm method minimizes the expected value of the square of the estimation error as $E[(x-\hat{x})^2/z]$.

The Kalman Filter uses two sets of equations to predict value of the state variable. The first one is the time update equation that is responsible for predicting the current state and covariance matrix, used in time $k+1$ to predict the previous state. The second one is Measurement Update Equation is responsible for correcting the errors. In a sense, it is back propagating to get a new value for the prior state to improve the guess for the next state. The equations for a general Kalman filter are given by

A) Time Update Equations

- $\hat{x}(k)=A(k-1)\cdot\hat{x}(k-1)+B(k-1)\cdot u(k)$ (33)
- $P(k)=A(k-1)\cdot P(k-1)\cdot A^T(k-1)+Q(k-1)$ (34)

B) The Measurement Equations

- $K(k)=P(k)\cdot C^T(k)\cdot(C(k)\cdot P(k)\cdot C^T(k)+R(k))^{-1}$ (35)
- $\hat{x}(k)=\hat{x}(k)+K\cdot(Z(k)-C(k)\cdot\hat{x}(k))$ (36)
- $P(k)=(I-K(k)\cdot C(k))\cdot P(k)$ (37)

where $k(k)$, and $p(k)$ are the Kalman filtering gain matrix and covariance matrix, respectively.

The detailed procedure for proceeding Kalman filter for the pose estimation of the WMR is expressed as follows;

Step 0: initialize $\hat{x}(0/0) = E[x(0)]$ and $P(0/0) = \text{cov}[x(0)]$

Step 1: one-step prediction with given $\hat{x}(k-1/k-1) = E[x(k-1)/z(k-1)]$

$$\hat{x}(k/k-1) = E[x(k)/z(k-1)]$$

$$= E[A(k-1)\cdot x(k-1) + B(k-1)\cdot u(k-1) + G\cdot w(k-1)/z(k-1)]$$

$$= A(k-1)\cdot E[x(k-1)/z(k-1)] + B(k-1)\cdot u(k-1) + G\cdot E[w(k)]$$

$$= A(k-1)\cdot \hat{x}(k-1/k-1) + B(k-1)\cdot u(k-1) \qquad (38)$$

$$P_1(k/k-1) = E[(x(k) - \hat{x}(k/k-1)^2/z(k-1)]$$

$$= E[(A\cdot x(k-1) + B\cdot u(k) + G\cdot w(k-1) - A\cdot \hat{x}(k-1/k-1) - B\cdot u(k))^2/z(k-1)]$$

$$= E[(A(k-1)x(k-1) + G\cdot w(k-1) - A(k-1)\cdot \hat{x}(k-1/k-1))^2/z(k-1)]$$

$$= A(k-1)\cdot P_1(k-1/k-1)A^T(k-1) + G\cdot Q(k-1)\cdot G^T \qquad (39)$$

Step2: we compute kalman gain as

$$\hat{x}(k/k) = \hat{x}(k/k-1) + K\cdot[Z(k) - C\cdot \hat{x}(k/k-1)] \qquad (40)$$

$$K(k) = P_1(k/k-1)\cdot C^T[C\cdot P_1(k/k-1)\cdot C^T + R(k)]^{-1} \qquad (41)$$

$$P(k/k) = P_1(k/k-1) - K(k)\cdot C\cdot P_1(k/k-1)$$

$$= [I - K(k)\cdot C]\cdot P_1(k/k-1) \qquad (42)$$

When the noise in the real mobile robot occurs, Kalman filter police has been proposed in [10-11]. In this paper, we extend the Kalman filter method to deal with the pose estimation problem of the WMR with corrupted imaging noise.

6 Simulations and Experimental Results

6.1 Simulation Results

Before preceding the following experiment, numerical simulations using Matlab/Simulink were used to illustrate the feasibility of the proposed controllers and to verify the effectiveness of the proposed methods. Fig.14 shows a simulation block diagram for implementing the pin-hole lens model and the proposed controllers Eq.(24) and Eq.(28).

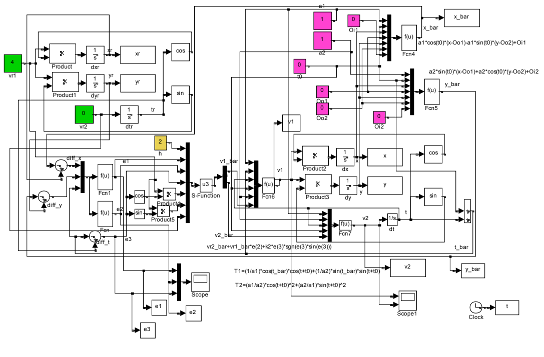
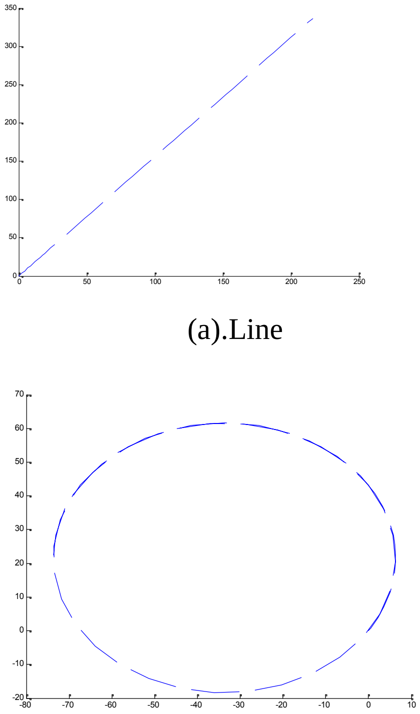

x_bar
a1*cos(t0)*(x-Oo1)+a1*sin(t0)*(y-Oo2)+Oi1
a2*sin(t0)*(x-Oo1)+a2*cos(t0)*(y-Oo2)+Oi2
y_bar
Product
Product1
Product2
Product3
S-Function
Fcn1
Fcn4
Fcn5
Fcn6
Fcn7
Scope
Scope1
Clock
e1
e2
e3
v1
v2
t_bar
T1=(1/a1)*cos(t_bar)*cos(t+t0)+(1/a2)*sin(t_bar)*sin(t+t0)
T2=(a1/a2)*cos(t+t0)^2+(a2/a1)*sin(t+t0)^2

Fig. 14 Simulation block diagram of the path tracking control system

In this section, two different reference trajectories, such as line, circular are considered (see Fig.15).

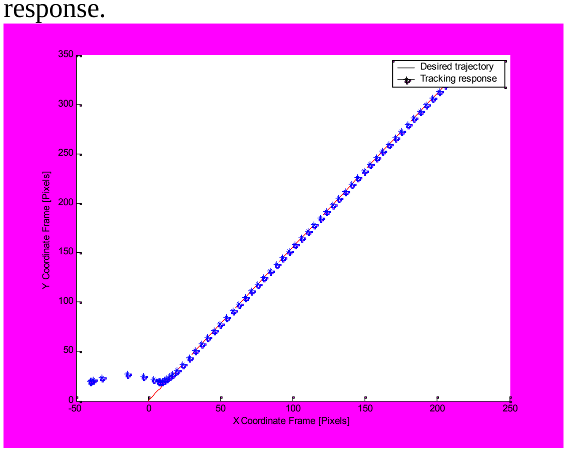


(a).Line

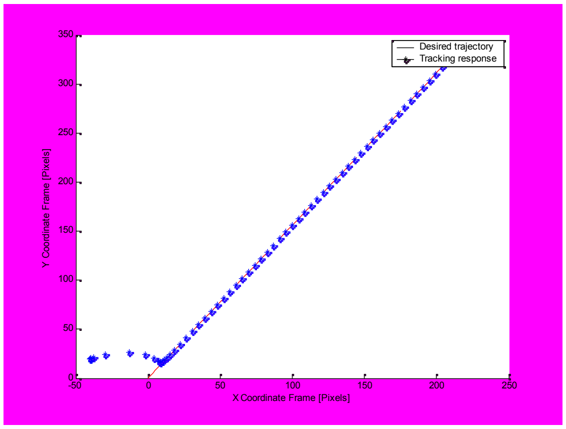
(b).Circle

Fig.15 Desired reference trajectories

Fig. 16 shows the Line trajectory tracking responses. A fixed-end point predictive controller: (a) tracking response. For the finite-horizon nonlinear predictive controller: (b) tracking response.



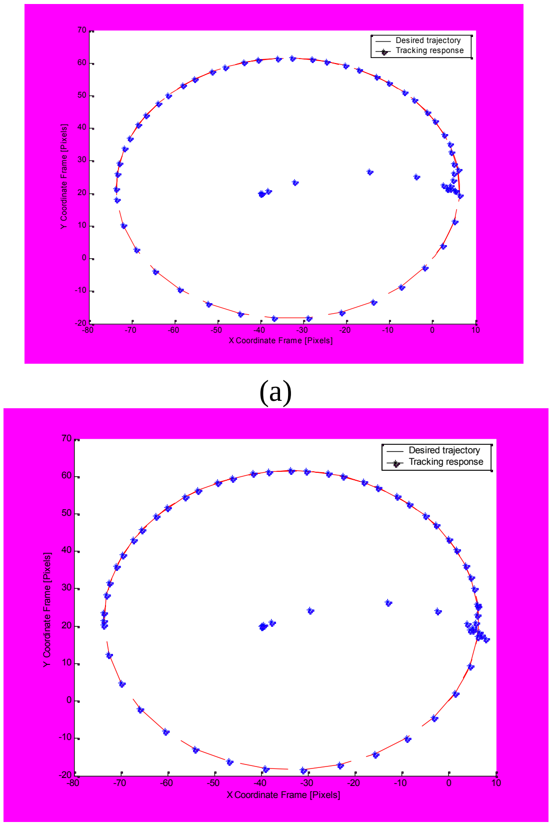
(a)



(b)

Fig. 16 Line trajectory tracking responses

Fig. 17 shows the Circle trajectory tracking responses. A fixed-end point predictive controller: (a) tracking response. For the finite-horizon nonlinear predictive controller: (b) tracking response.

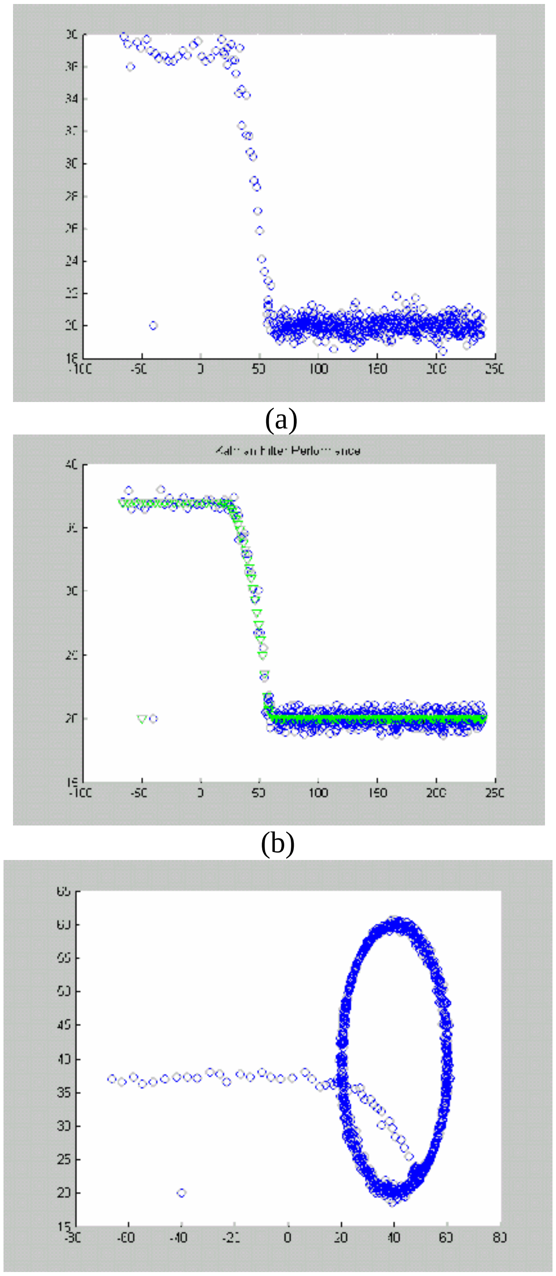


(a)

(b)

Fig. 17 Circle trajectory tracking responses

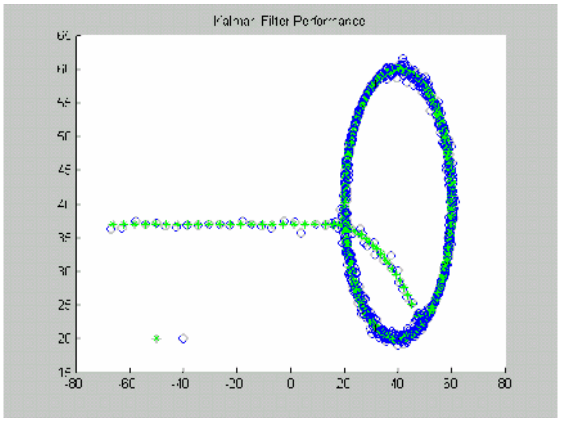
Fig. 18(a) depicts the line-tracking path that is corrupted with the noise. Fig. 18(b) shows that the corrupted noise is removed. Fig. 18(c) depicts the circle-tracking path that is corrupted with the noise. Fig. 18(d) shows the filtered circle trajectory. Fig. 18 reveals that the noise effect on posture estimation has been significantly reduced.



(a)

(b)

(c)



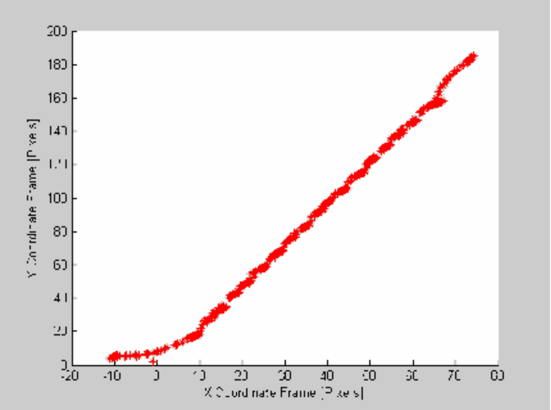
(d)

Fig. 18 The corrupted noise and Kalman filter trajectory tracking responses

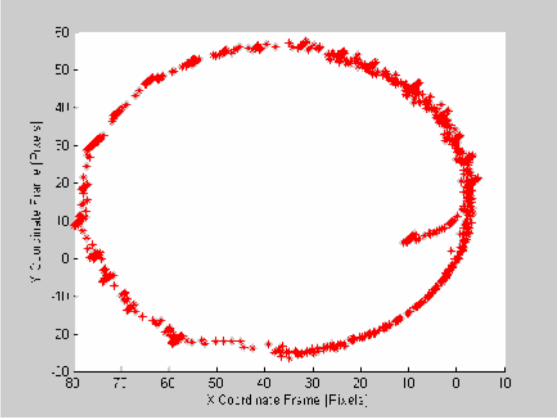
6.2 Experimental Results

Fig. 19(a) shows the resulting tracking responses for the camera-space desired trajectory line. Furthermore, Fig. 19(b) displays the resulting tracking responses for of the camera-space desired circle trajectory.

Fig. 20 (a) shows that the Kalman filter tracking responses for the desired line trajectory experimental result. Furthermore, Fig. 20(b) displays the resulting tracking responses for of the Kalman filter desired circle trajectory. The Kalman filter indeed removes out measurement noise from measurements.

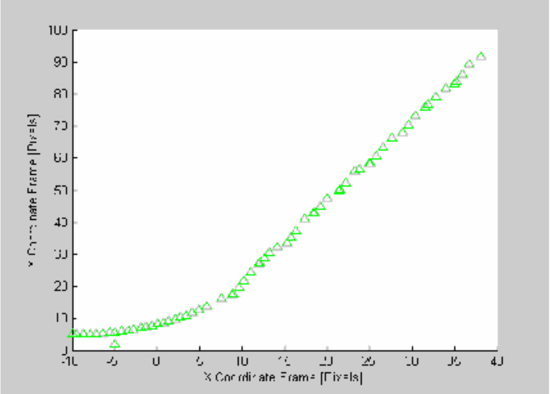


(a)

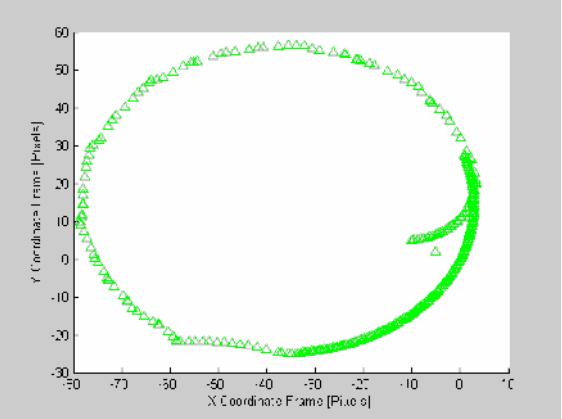


(b)

Fig. 19 The experimental tracking responses for the desired line trajectory



(a)



(b)

Fig. 20 The experimental Kalman filter tracking responses

The computer simulation results and experimental results have verified that two the predictive path tracking controllers really achieve the mission of path tracking control. From the experimental results illustrated in Fig. 13 there exist fluctuations caused by the image noise. Although sometimes the WMR seems to detour the desire trajectory in the camera space, the image-recognition process maybe cause the errors. This situation can be avoided if the experimental setup system is using Kalman filter.

7. Concludes

This paper has studied methodologies and techniques for control architecture design, path tracking and implementation of a vision-based wheeled mobile robot. The overall control system consists of three subsystems: the wireless communication system, the DSP-based motion control system for the WMR and the vision system for pattern recognition. Two kinematical, predictive path tracking control laws have been proposed in order to achieve asymptotical path tracking. The developed system obtains the posture of the WMR using combination of the vision system and dead-reckoning device, such that the accumulated errors caused by the numerical differentiation/integration will be reduced or even eliminated. The designed systems together with proposed control laws have been successfully used to manipulate the WMR follow the desired reference trajectories. The main results of this paper are summarized as below;

First, a DSP-based controlled WMR with internal current feedback loop has been constructed. The built wireless communication system has been implemented for signal transmission. Through the proposed PID control, the WMR achieves good transient performance to track velocity or torque commands. Aside form the simple PID control loop, the DSP-based low-level controller for the WMR is further constructed in order to perform sophisticated path tracking control laws. The WMR obtains the posture not only by the DR device but also by the vision system. The image-recognition system can process 15 frames per second by using one well-known color model, e.g. H-S model. These subsystems are integrated to test the path tracking control laws.

Second, two novel predictive control laws have been proposed for the system associated with the kinematical model, i.e. the model without the incorporation of its dynamic effects. In addition, the so-called pin-hole lens model has been supplied in the procedure of the control laws design. Computer simulations and experimental results have also shown the feasibility and effectiveness of the proposed predictive controller associated with the kinematical model.

Third, there is noise reduction for pose estimation of the WMR by applying the Kalman filter to remove the noise corrupted the captured images, so the Kalman filter can be applied in order to reduce the noise. Computer simulations and experimental results have also shown the feasibility and effectiveness of the proposed Kalman filter associated with imaging processing.

References

[1] J. Qiuling, X. Xiaojun, and L. Guangwen, Formation Path Tracking Controller of Multiple Robot System By High Order Sliding Mode, *Automation and Logistics, IEEE International Conference* .18-21 Aug. 2007 ,pp. 923 – 927

[2] J. Y. Lee, J. S. Yeon, and J. H. Park, Robust nonlinear control for flexible joint robot manipulators, *SICE, Annual Conference*. 17-20 Sept. 2007. pp.440 – 445

[3] B. Song, and J. W. Choi, Robust Nonlinear Control for Biped Walking with a Variable Step Size, *SICE-ICASE, 2006. International Joint Conference*, Oct. 2006 Page(s):3490 – 3495

[4] G. Lei, L. Qizheng, and W. Shiming, Design of Fuzzy Sliding-mode Controller for Bicycle Robot, *Nonlinear System Robotics and Biomimetics, ROBIO '06. IEEE International Conference.* Dec. 2006 Page(s):176 – 180

[5] M. R. Heinen, and F. S. Osorio, Neural Networks Applied to Gait Control of Physically Based Simulated Robots, *Neural Networks, SBRN '06. Ninth Brazilian Symposium* . Oct. 2006 Page(s):26 - 26

[6] D. G, and R. RD, Robust adaptive backstepping control for a nonholonomic mobile robot, *Wilson, Systems, Man, and Cybernetics, 2001 IEEE International Conference.* Vol. 5, 7-10 Oct. 2001 Page(s):3241 - 3245 vol.5

[7] A. Al-Mamun, Z. Zhu, P. Vadakkepat and T.H. Lee, Path Following Controller for Gyroscopically Stabilized Single-Wheeled Robot, *Proceedings of the 9th WSEAS International Conference on Automatic Control, Modeling & Simulation,* Istanbul, Turkey, May 27-29, 2007 11.

[8] De-Feng He, Hai-Bo Ji, Tao Zheng, On robustness of suboptimal min-max model predictive control, *WSEAS TRANSACTIONS on SYSTEMS and CONTROL,* Issue 8, Volume *2,* pp. 428-433. August 2007.

[9] R. Hedjar, R. Toumi, P. Boucher, and D. Dumur, Feedback nonlinear predictive control of rigid link robot manipulations, *Proceedings of American Control Conference,* Anchorage, AK, pp. 3594-3599, May 2002.

[10] S. Kwon, K. W. Yang, and S. Park, An Effective Kalman Filter Localization Method for Mobile Robots, *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference.* Oct. 2006 Page(s):1524 – 1529.

[11] J. Zhu, J. Park, and K. Lee, Robust Kalman filter of discrete-time Markovian jump system with parameter and noise uncertainty, *Proceedings of the 7th WSEAS International Conference on Simulation, Modelling and Optimization*, Beijing, China, September 15-17, 2007 .