# An Adaptive Motion Controller for a Differential-Drive Mobile Robot

UM-MEAM-93-01

L. Feng, Y. Koren and J. Borenstein
Mobile Robot Laboratory
Department of Mechanical Engineering and Applied Mechanics
The University of Michigan
Ann Arbor, MI 48109-2125

## Abstract

This paper describes the design and implementation of an adaptive motion controller for a differential-drive mobile robot. The controller uses absolute position information to modify the control parameters in real time and, in turn, to compensate for the motion errors. Robot motion errors are classified into internal and external errors. The cross-coupling control method is used to compensate for the internal errors that can be detected by the wheel encoders. The adaptive controller provides compensation for the external errors. The adaptive controller is analyzed, and its stability and convergence are discussed. Both computer simulation and experiments are conducted to evaluate the control system. The results show significant improvements over conventional controllers.

# 1 Introduction

Mobile robotics is a fast advancing field that has found many new applications in nuclear power plant's maintenance, waste management, disabled person assistance, material handling, security and household service [1]. Currently most of the mobile robot research has concentrated on the application of mobile platforms to perform intelligent tasks, rather than on the development of methodologies for analyzing, designing, and controlling the mobile system. However, improved motion control systems will enable the application of mobile robots to tasks requiring accurate trajectory tracking even in unstructured environments [2]. Accordingly, this paper suggests a new motion control method.

As shown in Fig. 1, a typical motion control system is based on a static kinematic model whose accuracy depends on the operation of the robot and the structure of the control loop. In this paper, we introduce an adaptive controller with the following two improvements over the conventional controller of Fig. 1:

1. **Adaptation to the change of robot parameters and the changing environment [3].**

Some robot parameters change with the operation of the robot, e.g., the drive wheel diameters change with the load and load distribution. In order to navigate and control the robot motion precisely, a kinematic model is needed, whose parameters are adjusted according to changes in robot operating conditions.

2. **Direct control of the** *most significant error* [4].

In this paper, we introduce the term *most significant error*. This term is used to refer to the error that has the largest impact on the motion accuracy. In this study, we choose the orientation error as the most important error. The selection of the *most significant error* depends on the application and the kinematics of the robot.

Adaptive control has been used to solve similar problems in the area of robotics. particularly in the adaptive control of robot manipulators. The basic idea of adaptive control is to eliminate the effects of variations in the controlled system parameters by estimating the parameters in real time and using the estimates in the control process, or by generating correction signals to compensate for the errors. Adaptive control has been used to accurately control the motion of robot manipulators in the case where the parameters of the arm are not precisely known or change with the operation of the robot (e.g., flexible arm and changing load) [5, 6, 7]. Application of adaptive control in mobile robot control has not been widely studied. A self-tuning navigation algorithm has been suggested by Banta [3]. Ths algorithm is aimed at correcting motion errors caused by steering miscalibration, uneven tire wear and wheel misalignment. Banta's algorithm employs a least-square method for the parameter estimation and the estimates are used to adjust the control of the robot.

We have identified three reasons for the fact that adaptive control has not found wide application in mobile robotics:

A. **It is difficult to apply the methodologies for modeling and controlling stationary manipulators to mobile robots, because of the inherent differences between the two.** For examples: (1) A mobile robot contains multiple closed-link chains, whereas a stationary manipulator forms a closed-link chain only when in contact with a stationary object; and (2) The contact between a wheel and a planar surface is a high-pair joint, whereas stationary manipulators contain only low-pair joints;

B. **There has been very little study on the robot-environment system modeling [1].** The effects of some of the external factors, e. g., slippage, are very hard to model accurately. The relations between the motion errors and the error of each drive loop are not well known.

C. **The absolute motion information of a mobile robot is not readily available [8].** It is more difficult to obtain accurate motion information. Typicaly this information is very expensive, infrequent, and less accurate compared with the absolute motion information of robot manipulators.

This paper will concentrate on the problem of adaptive motion control of a mobile robot in the case where the physical model that describes the motion of the mobile robot is not well known or changes with the operation of the robot. The proposed controller dynamically adjusts its parameters according to the robot's operating and the environmental conditions. In addition, *Cross-coupling control* [4] (explained in Section 5.2) is used to control the orientation error by coordinating the motion of the two drive loops.

In the next Section, we will discuss the kinematics of the differential-drive mobile robots. In Section 3, error sources in robot motion will be discussed. In Section 4, we will introduce a model for the vehicle-environment system. In Section 5, we will briefly review the principles of adaptive control, different control structures and different design methods, and we will discuss the adaptive motion control for mobile robot. In Section 6, the performances of the proposed controller will be evaluated by both computer simulation and experiments. Finally, conclusions are drawn in the Section 7.

## 2 Kinematics of Differential-Drive Mobile Robots

Sensed forward kinematics is the relationship between the motion of the robot and the sensed motion of the robot wheels. For a differential-drive mobile robot as shown in Fig. 2, the velocity forward kinematics is given by

$$\dot{x} = \frac{v^L + v^R}{2} \sin\theta \tag{1}$$

$$\dot{y} = \frac{v^L + v^R}{2} \cos\theta \tag{2}$$

$$\dot{\theta} = \frac{v^R - v^L}{b_w}, \tag{3}$$

where $\dot{x}$. $\dot{y}$ and $\dot{\theta}$ are the linear and angular velocity of the robot, $v^L$ and $v^R$ are the velocities of the left and right drive wheels, $\theta$ is the orientation of the robot. $b_w$ is the distance between the two drive wheels (i.e., wheel base), respectively.

Inverse kinematics determines the motions of the drive wheels that are needed to obtain a desired robot motion. Since there are only two degrees of freedom. we can specify two parameters, say, the velocity in the x direction ($\dot{x}$) and the rate of change in orientation ($\dot{\theta}$). Then we solve $v^R$ and $v^L$ from Eqs. (1) and (3), and we obtain

$$v^R = \frac{\dot{x}}{\sin\theta} + \frac{b_w}{2}\dot{\theta} \tag{4}$$

$$v^L = \frac{\dot{x}}{\sin\theta} - \frac{b_w}{2}\dot{\theta} . \tag{5}$$

When $\theta = 0^\circ$ or $\theta = 180^\circ$, the above equations are not defined, and the drive wheel speeds can be determined from Eqs. (2) and (3). The above inverse kinematic analysis is based on the assumption that the two drive wheels are exactly parallel and every parameter needed is precisely known and not changing. For accurate trajectory control these assumptions are not adequate. Even frequent calibration of the robot is not very useful since some parameters are changing with the operation of the robot and the change in the environment.

# 3 Motion Error Sources for Mobile Robots

We can largely classify error sources into two categories: internal errors and external errors. The internal errors are the errors that can be detected by the wheel motion information. The external errors are the errors that only become apparent when the robot wheels interact with the environment and can only be detected by absolute robot motion measurements. The external errors can be further divided into systematic errors and non-systematic errors. The systematic errors are the errors that exist over a long period of time without changing their characteristics. The non-systematic errors happen in a random fashion and can only be described in a statistical sense.

## 3.1 Internal Errors

The main internal error sources are:

1. **Different drive loop parameters.** For a differential-drive robot, when the two drive loops have different parameters (e.g., time constants and loop gains), the responses of the two loops will be different, and the result is an error in the path.

2. **Different disturbances acting on the different drive loops.** One example is the difference in bearing frictions [9]. The difference of disturbances will affect the transient response and in some case the steady state response - depending on the type of controllers used in the control loops.

## 3.2 External Systematic Errors

External errors can be classified as systematic and nonsystematic errors. The main external systematic error sources are:

1. **Different wheel diameters.** When the two drive wheels have different diameters and the same angular speed, the robot will follow a circular instead of a straight line path [10] (Fig. 3). There are several causes for a difference in the wheel diameters: (1) Load or its distribution changes; (2) Uneven wear of the wheels; (3) Uneven inflation in case of inflated wheels; (4) Manufacturing tolerance of the wheels.

The difference in wheel diameters can be modeled as [10]

$$\Delta\theta = \frac{\omega\Delta t(d^R - d^L)}{b_w} \tag{6}$$

where $\omega$ is the angular velocity of the drive wheels, $\Delta\theta$ is the change in robot orientation, $d^L$ and $d^R$ are the left and right drive wheel diameters, and $\Delta t$ is the amount of time elapsed.

We can observe that: (1) $\Delta\theta \propto \frac{1}{b_w}$, i.e., the orientation error resulted from different wheel diameters is inversely proportional to the wheel base. (2) $\Delta\theta \propto d^R - d^L$, i.e., the orientation error is proportional to the wheel diameter difference. (3) $\Delta\theta \propto \omega\Delta t$, i.e., the orientation error is also proportional to the distance traveled.

2. **Wheel misalignment.** The effect of the misalignment of the drive wheels is the robot constantly pulling to one side. Causes for this error include: (1) Manufacturing tolerance, and (2) The distribution and the amount of load carried by the robot.

3. **Contact area.** When the wheel contact with the floor, there is a contact area, rather than a contact point. This causes an uncertainty about the effective distance between the drive wheels (i.e., the wheelbase $b_w$) and thereby introduces inaccuracy in orientation calculation.

These three errors are the most important biased errors. They exist over a long period of time and their effect remains the same, so they can be measured and compensated for.

## 3.3 External Non-Systematic Errors

External nonsystematic errors include:

1. **Wheel slippage.** Slippage is a big problem in dead-reckoning, it is not a biased error and it can happen in a very short time period. However, the slippage normally happens only when the robot moves on a curved path (centrifugal force) and when the robot accelerates or decelerates.

2. **Floor roughness.** When the robot travels over a rough floor, the wheels move up and down over the bumps. Part of the motion recorded by the

wheel encoders is the vertical distance required to clear the bumps. Surface roughness and undulation cause the traveled distance to be overestimated.

The external nonsystematic errors are random in nature and there are no good ways to qualitaticely predict these errors. Therefore, we can at best compensate after they occured.

## 3.4 Error Decomposition

The motion error of the robot can be decomposed as follows (Fig. 4): The first is the orientation error $e_\theta$, which is defined as the difference between the real robot orientation and the desired robot orientation. It is the *most significant error* as far as motion accuracy is concerned because the orientation error will result in a contour error, which grows with the distance traveled. The contour error $e_c$ is defined as the distance between the actual robot position to the desired robot position in the direction perpendicular to the direction of travel. The contour error is the direct result of the orientation error. We can not control both errors at the same time for a differential-drive robot. The third error is the tracking error $e_t$, which is the distance between the actual position to the desired position in the direction of travel. The tracking error does not have a very significant effect on the motion accuracy of the robot, and we can control the tracking error by adjusting the input so that the robot has the desired traveling speed.

The main problem with motion errors of mobile robots is that the errors can grow without a bound, and that the errors increase nonlinearly with the distance traveled because of the *accumulation* of the orientation error. The key task is therefore to control the growth of the error. The unbounded growth of error is in most part caused by the systematic external errors. Therefore we will use biased error information from absolute position measurements for compensation.

# 4 Robot-Environment System Modeling

A fundamental difference between adaptive motion control and conventional motion control is that in the adaptive control both the robot and its environment are included in the system, whereas in the conventional control only the robot system is modeled. Although there is extensive research in the vehicle field, the problem of accurately modeling conventional vehicles has not been solved. Typically, only simple parametric vehicle models that are adjusted with experimental data are used [1].

A dead-reckoning model aimed at improving the path following accuracy by introducing error terms was introduced by Banta [3]. In Banta's work, the robot position $(x_k, y_k)$ and orientation $\theta_k$ are given by:

$$x_k = x_{k-1} + \Delta u_k \cos(\frac{\theta_{k-1} + \theta_k}{2}) + \beta_x \Delta u_k \tag{7}$$

$$y_k = y_{k-1} + \Delta u_k \sin(\frac{\theta_{k-1} + \theta_k}{2}) + \beta_y \Delta u_k \tag{8}$$

$$\theta_k = \theta_{k-1} + \Delta\theta_k + \beta_\theta \Delta u_k \, , \tag{9}$$

where

$$\Delta u_k = \frac{\Delta n_k^L d^L + \Delta n_k^R d^R}{2} \tag{10}$$

$$\Delta\theta_k = \frac{\Delta n_k^R d^R - \Delta n_k^L d^L}{b_w} \, , \tag{11}$$

growth of error is in most part caused by the systematic external errors. Therefore we will use biased error information from absolute position measurements for compensation.

# 4 Robot-Environment System Modeling

A fundamental difference between adaptive motion control and conventional motion control is that in the adaptive control both the robot and its environment are included in the system, whereas in the conventional control only the robot system is modeled. Although there is extensive research in the vehicle field, the problem of accurately modeling conventional vehicles has not been solved. Typically, only simple parametric vehicle models that are adjusted with experimental data are used [1].

A dead-reckoning model aimed at improving the path following accuracy by introducing error terms was introduced by Banta [3]. In Banta's work, the robot position $(x_k, y_k)$ and orientation $\theta_k$ are given by:

$$x_k = x_{k-1} + \Delta u_k \cos(\frac{\theta_{k-1} + \theta_k}{2}) + \beta_x \Delta u_k \qquad (7)$$

$$y_k = y_{k-1} + \Delta u_k \sin(\frac{\theta_{k-1} + \theta_k}{2}) + \beta_y \Delta u_k \qquad (8)$$

$$\theta_k = \theta_{k-1} + \Delta\theta_k + \beta_\theta \Delta u_k\ , \qquad (9)$$

where

$$\Delta u_k = \frac{\Delta n_k^L d^L + \Delta n_k^R d^R}{2} \qquad (10)$$

$$\Delta\theta_k = \frac{\Delta n_k^R d^R - \Delta n_k^L d^L}{b_w}\ , \qquad (11)$$

where $\Delta n_k^L$ and $\Delta n_k^R$ are the measured angular position changes of the left and right wheels in each sampling period, $\beta_x$, $\beta_y$, and $\beta_\theta$ are the errors coefficients, and the errors are assumed to be proportional to the distance traveled by the robot. $\Delta u_k$ is the distance traveled by the robot in each sampling period and $\Delta\theta_k$ is the change in robot orientation during the sampling period.

When we instruct the robot to move on a straight line, the resultant path might be a straight line that is at an angle with the desired line path (if $\beta_x \neq 0$, $\beta_y = 0$ and $\beta_\theta = 0$); or a circle (if $\beta_x = 0$, $\beta_y = 0$ and $\beta_\theta \neq 0$); or a circle plus a straight line (if $\beta_x \neq 0$, $\beta_y = 0$ and $\beta_\theta \neq 0$). The last two cases describe the nonlinear characteristic where the contour error increases with the distance traveled. Although this model is very simple, it is able to represent some very important error sources and it has been experimentally proved to provide reasonably good results [3, 10]. This model will be used in our adaptive motion control.

The error model given by Eq. (9) indicates that the orientation error is a linear function of the distance traveled by the robot. Our experimental results support this assumption. Fig. 5 shows the orientation error of the robot when it moves along a straight wall. In Fig. 5, A is the case where the drive wheels of the robot are not taped and B is the case where the right drive wheel is taped with masking tape. The orientation of the robot is calculated by using two sonars measuring the distance to the wall as shown in Fig. 6. From Fig. 5, we can observe that the orientation error of the robot is approximately linearly proportional to the distance traveled. The difference in drive wheel diameters will cause the robot to move on a circular path, i.e., the orientation changes as a linear function of the distance traveled. We can think of the composite effect of all the error sources as an *effective difference in wheel diameters.*

# 5 Adaptive Motion Control

In this section, we will discuss adaptive mobile robot motion control. In Section 5.1, we will review the basics of adaptive control and the adaptation mechanisms and in Section 5.2, we introduce the concept of cross-coupling control. In Section 5.3, we discuss the error decomposition and its effects on the controller design and the detailed design and analysis of the adaptive robot motion controller is presented in Section 5.4.

## 5.1 Adaptive Control

There are two main kinds of adaptive control schemes, the self-tuning adaptive control (STAC), and the model-reference adaptive control (MRAC). A STAC system (Fig. 7) first uses an estimator to determine on-line the parameters of the process, and then employs these estimated parameter values in a controller design block to tune the adjustable controller [11]. Recursive least-square is the most widely used estimation method. It provides only a suboptimal solution but is simple and robust [12]. Kalman filter based techniques have also been widely used and modified to take application-specific problems into account [12]. A STAC system for robots has been studied by Koivo [13]. Koive's system uses recursive least-square to estimate the parameters and the regulator is designed based on the extended minimum variance. However, on-line estimation of all system parameters and the control design make STAC computationally intensive [14]. The numerical estimation techniques tend to be numerically unstable as the number of parameters increases in the system model [15] and the convergence rate decreases with the increase in the number of parameters to be estimated. Prior knowledge of the plant model is needed, such as, the order and the form of the model. This knowledge may not be readily available for complex models.

In MRAC (Fig. 8), there are four main components: (1) the reference model that specifies the desired performances; (2) the adjustable system whose performances should be as close as possible to that of the reference model; (3) the subtractor that forms the error between the states or the outputs of the reference model and of the adjustable system (generalized error); (4) the adaptation mechanism that processes the generalized error in order to modify accordingly the control or the parameters of the adjustable system.

The first implementation of MRAC in robotics was done by Dwbowsky et. al [16], where a parametric optimization technique was used and stability was investigated for the uncoupled, linearized system model. Most of the recent research efforts have been focused on stability based methods, particularly, the hyperstability theory [6]. Stoten conducted experimental studies on the MRAC of manipulators [7] and Craig included nonlinearity compensation along with a feedback portion and parameter identification features [5]. In Craig's approach, the plant model does not have to be the same as the real plant and only performance convergence is needed instead of parameter convergence.

The model-reference adaptive control scheme was chosen for our system for the following reasons:

1. **It requires performance convergence rather than parameter convergence.** The objective is to achieve the desired performance. Normally parameter convergence can only be realized when some additional conditions are satisfied [17].

2. **The reference model is used to specify the desired performance and to monitor the state of the robot.** The reference model itself can be adjusted according to the operating conditions of the robot and the environment it is working in. Furthermore, when the difference between the model and the real system is small, the adaptation process will be fast and accurate.

3. **Small computational load.** The on-line identification and design procedures of a STAC system can be computational intensive and stability problems often occur when the number of the variables to be estimated becomes large [17]. This is not the case with the MRAC system.

## 5.2 Cross-Coupling Control

The objective of cross-coupling control is to reduce the orientation error by coordinating the control of both drive loops [18]. It is best suited for the applications where the motion of several axes must be coordinated to achieve an accurate trajectory [4]. For example, if a differential-drive mobile robot must follow a straight-line path, then the speed of the two drive wheels should be identical (assuming there are no external errors). This is done by the proposed cross-coupling control scheme, which is shown in Fig. 9. In this control scheme, in addition to the two conventional proportional control loops for controling the error in each drive loop, there is a proportional and integral controller that is used to control the error $e$ that is proportional to the orientation error of the robot and is calculated in real time. If there exists orientation error, a correction signal will be generated in addition to the corrections in each drive loop. A thorough analysis of the cross-coupling method was presented in [18] and the conclusions are summarized below:

1. **Cross-coupling control directly reduces (or eliminates) the orientation error by coordinating the velocities of the two control loops.**

The most important advantage of cross-coupling control is that it directly controls the *most significant error* (here the orientation error), while conventional controllers attempt to reduce the individual errors in each drive loop. The other advantage of cross-coupling is that the corrections occur in both control loops simultaneously, and, as a result, it has short settling time as well as excellent disturbance rejection capability.

2. **Combined cross-coupling and encoder compensation gain.**

Dual compensation gain uses the product of two gains, $c_1^L c_2^L$ and $c_1^R c_2^R$ to travel on curved paths and to compensate for known external errors. The cross-coupling gains $c_1^L$ and $c_1^R$, allows the robot to follow curved paths. For mobile robots, complicated curved paths can often be constructed from linear and circular segments. When the robot follows a circular path, the speed ratio of the two wheels should satisfy $v^R/v^L = (R + b_w/2)/(R - b_w/2) = c_1^L/c_1^R$ [8], where $R$ is the radius of the circle. Obviously, in order to follow different circular paths, different speed ratios are required.

The encoder compensation gains $c_2^L$ and $c_2^R$, are used to compensate for known external errors. For example, the robot may have different drive wheel diameters. Then the encoder compensation gains can be adjusted to compensate for this error. For example, if the left drive wheel diameter, $d^L$ is larger than that of the right drive wheel, $d^R$ and we give the same speed commands to both control loops, then the result is a circular path. However if we set $c_2^L = 1$ and $c_2^R = d^L/d^R$, the error is compensated for.

The encoder compensation gains are used as the adjustable parameters in the adaptive controller. Since there are many factors affecting the path errors and many of them are changing with the operation of the robot and the environment, a fixed set of encoder compensation gains can not provide satisfactory performance over a wide range of operating conditions. An adaptive controller is needed to adjust the compensation gains to compensate for the motion errors. The final dual compensation gain values are the product of the cross-coupling and encoder compensation gains.

3. **At steady state, $v^L/v^R = c^R/c^L$.**

4. **The steady state orientation error caused by the continuous disturbances is eliminated.**

## 5.3 Compensating for Robot Motion Errors

In Section 4, we introduced a decomposition model for the motion errors. As shown in Fig. 4, motion errors can be decomposed into three major components, orientation error, contour error and tracking error. This decomposition allows to decompose a complicated problem into several simpler problems. Among these three errors, the orientation error is the most significant one since it will result in contour error that will rapidly grow with the distance traveled.

The contour error is the direct result of the orientation error. The orientation error can be accurately modeled as a linear function of the distance traveled. The key factor in determining the orientation error is the ratio of the speeds of the two drive wheels in linear motion. In our motion controller, cross-coupling control is employed to compensate for the internal errors, while one adaptive loop is used to compensate for the external errors by adjusting encoder compensation gains.

Once accurate control of the *speed ratio* of the two drive-wheels is achieved, we may want to accurately control the *absolute speed* values. Let us assume for a moment that the right drive-wheel has a larger diameter than the left wheel, but we do not know the exact value of the diameters. The disturbances in the drive loops also affect the speed values. If we can measure the real linear speeds instead of the wheel speeds, we can add another adaptive loop to compensate for the speed value in the presence of both internal and external errors.

After the first two steps, we can accurately control both the speed ratio and speed values. However, there is still an important factor that must be considered. When the robot makes a turn, a fixed value for the wheel base $b_w$ is used in the dead-reckoning and in the inverse kinematics computations. However. the value of $b_w$ is usually not precisely known, or it can change because of the load and the contact area of the wheels. An adaptation can be dedicated to the estimation of this parameter, or simple calibration calculation can be used.

From the above discussion, it is easy to see that the first and third steps allow us to compensate for the orientation error, while the second step allows us to compensate for the tracking error. The contour error needs to be corrected separately because of the kinematic constraints of the differential-drive robot. The contour error can be corrected by making a turn toward the desired path, then turn back to the appropriate orientation after the robot comes back on the desired path.

In our work, we used sequential procedures for error compensation instead of a one-step process as in the case of contour error control for machine tools. There are three reasons: (1) In the case of a mobile robot, accurate absolute motion measurements are not available most of the time, while in machine tool control, accurate table motion information is obtained through encoders at all times. (2) In the case of machine tool control, the trajectory error is the contour error that is well defined by the error of each axis, while in the case of a mobile robot, the two axes are parallel and their motions are coupled. There are no accurate relations between the motion error of the robot and the error of each axis except for the orientation error. The position of the robot is estimated through dead-reckoning, which is not reliable for a long period of time. (3) A differential-drive mobile robot has only two degrees of freedom. It is impossible to control the position and orientation of the robot simultaneously.

In the next section, we will discuss the compensation of orientation error through encoder gain compensation. The other error compensations will not be discussed in this paper.

## 5.4 Encoder Gain Adaptation by the Hyperstability Method

There are three basic designs for a model reference adaptive control system [11]. The first one is based on local parametric optimization theory, the second one is based on Lyapunov Functions, and the third one is based on the hyperstability approach. Stability problems are inherent in MRAC design due to their time-varying nonlinear character. Therefore, a satisfactory MRAC system must first be shown to provide stability for the whole system. The adaptive control design based on the use of *hyperstability* and *positivity* concepts, is the most successful approach in the design of model reference adaptive systems [17]. In our design, the adjustable parameters are the encoder compensation gains and the basic assumptions are: (1) the adaptation takes place only when the robot moves on a straight line path with constant speed inputs; (2) the adaptation occurs at a much lower frequency compared to the sampling rate of the cross-coupling control; (3) the two drive loops have the same parameters and there is no disturbance.

In order to apply the hyperstability approach, one must first analyze the stability of the system, and then choose the best adaptation gains from all the stable systems. The design procedures consist of the following steps [17]: (1) Transform the MRAC system into the form of an *equivalent feedback* system composed of two blocks, one in the forward path and one in the feedback path as shown in Fig. 10; (2) Find solutions for the part of the adaptation laws which appears in the feedback path of the equivalent system such that the *Popov integral inequality* is satisfied [17]; (3) Find solutions for the remaining part of the adaptation law which appears in the forward path of the equivalent system such that the forward path is a hyperstable block [17]; (4) Specify the adaptation law explicitly for the original MRAC system. The adaptive motion control system for a differential-drive mobile robot is shown in Fig. 11.

In this approach, we concentrate on the orientation error of the robot. Based on the orientation error model given in Eq. (9), the reference model is chosen as

$$\theta_m(k) = \theta_m(k-1) + \frac{\Delta u_k^R - \Delta u_k^L}{b_w} \,. \tag{12}$$

Since the adaptation occurs at a much lower frequency, we can assume that at each adaptation step, the cross-coupling loop has reached its steady state [8], i.e.,

$$\begin{aligned} v^L &= \frac{hk_ak_bc^R(R^L+R^R)}{(1+hk_ak_b)(c^L+c^R)} && (13) \\ v^R &= \frac{hk_ak_bc^L(R^L+R^R)}{(1+hk_ak_b)(c^L+c^R)} \,, && (14) \end{aligned}$$

where $R^L$ and $R^R$ are the input velocity commands, $k_a$ is the proportional loop gain, $k_b$ is the motor gain and $h$ is the encoder gain. Since

$$\begin{aligned} \frac{\Delta u_k^R - \Delta u_k^L}{b_w} &= \frac{v^R\Delta t - v^L\Delta t}{b_w} \\ &= \frac{c^L-c^R}{c^L+c^R}\frac{hk_bk_a}{(1+hk_bk_a)b_w}(R^L+R^R)\Delta t \,. \end{aligned}$$

The reference model can be written as

$$\theta_m(k+1) = \theta_m(k) + p_m u_1(k) \, , \tag{15}$$

where

$$u_1 = \frac{hk_bk_a}{(1+hk_bk_a)b_w}(R^L + R^R)\Delta T \, ,$$

$\Delta T$ is the sampling period of the adaptive loop, and $p_m = (c^L - c^R)/(c^L + c^R)$. If we let $c^L = \alpha R^R$ and $c^R = \alpha R^L$, then $p_m = (R^R - R^L)/(R^R + R^L)$. If following straight line is the goal, then the reference model is simple, namely, $p_m = 0$.

The controlled system can be modeled as

$$\dot{\theta} = \frac{v^R - v^L}{b_w} + \beta_\theta \frac{v^R + v^L}{2} \, . \tag{16}$$

Since we are only interested in the ratio of the two speeds $v^L$ and $v^R$. The effects of $\beta_\theta$ can be absorbed into the encoder compensation gains. The above equation can be rewritten as

$$\theta(k+1) = \theta(k) + p(k+1)u_1(k) \, , \tag{17}$$

where $p(k+1) = (c^L(k+1) - c^R(k+1))/(c^L(k+1) + c^R(k+1))$. To simplify the implementation, we can set $c^L = 1$ and only changes $c^R$. Then we will have $p(k+1) = (1 - c^R(k+1))/(1 + c^R(k+1))$ and $c^R$ can be found by $c^R(k+1) = (1 - p(k+1))/(1 + p(k+1))$.

The error is defined as

$$\epsilon(k) = \theta(k) - \theta_m(k) \, . \tag{18}$$

The adaptation algorithm is chosen to be the integral algorithm, which is given as

$$p(k+1) = p(k) + \gamma e(k+1)u_1(k)\,, \tag{19}$$

where $e(k+1)$ can be found from the following equations:

$$\begin{aligned} e(k+1) &= \theta_m(k+1) - \theta(k+1) \\ &= \theta_m(k) - p_m u_1(k) - \theta(k) - p(k+1)u_1(k)\,. \end{aligned} \tag{20}$$

Substituting the expression of $p(k+1)$ into the above equation, we can find $e(k+1)$ as

$$e(k+1) = \frac{\theta_m(k) - \theta(k) + (p_m - p(k))u_1(k)}{1 + \gamma u_1^2(k)}\,. \tag{21}$$

In order to show the stability and convergence of the system, we first decompose the system into a linear time-invariant system plus a non-linear time-variant feedback system. The system can be rewritten as

$$\begin{aligned} e(k+1) &= \theta_m(k+1) - \theta(k+1) \\ &= e(k) + p_m u_1(k) - p(k+1)u_1(k) \\ &= e(k) + m(k+1)\,, \end{aligned} \tag{22}$$

where $m(k+1) = p_m u_1(k) - p(k+1)u_1(k)$. The decomposed system is shown in Fig. 12. In order to prove the stability of the system, the two blocks can be examined separately.

**The hyperstability theorem** states that[17]: *If the feedforward block is such that the feedback system is globally (asymptotically) stable for all feedback blocks satisfying the Popov integral inequality, one then says that the feedback system is (asymptotically) hyperstable and that feedforward block is a hyperstable block.*

First let us show that the linear forward loop have a positive real transfer function. The transfer function of the linear block can be written as

$$G(z) = \frac{z}{z-1}\,. \tag{23}$$

This transfer function satisfies the following conditions:

1). It has a simple pole at $z = 1$;

2). $G(e^{j\omega}) + G(e^{-j\omega}) = 1 > 0$.

From the definition of a positive real system, we can determine that the linear block is positive real. Next we need to examine if the non-linear block satisfies the *Popov integral inequality* [17],

$$\eta(N) = \sum_{k=0}^{N} v^t(k)w(k) \geq -r_0^2, \;\; N >= 0 \tag{24}$$

where $v$ is the input vector, $w$ is the output vector of the feedback block, and $r_0^2$ is a finite positive constant.

Now we begin to show that the feedback block satisfies the Popov integral inequality.

$$\begin{aligned} m(k+1) &= (p(k+1) - p_m)u_1(k) \\ &= (p(0) + \gamma\sum_{i=0}^{k} e(i+1)u_1(i) - p_m)u_1(k)\,. \qquad (25) \\ e(k+1)w(k+1) &= e(k+1)(-m(k+1)) \\ &= (\gamma\sum_{i=0}^{k} e(i+1)u_1(i) + p(0) - p_m)e(k+1)u_1(k) \\ &\geq -\frac{1}{2\gamma}(p(0) - p_m)^2\,. \qquad (26) \end{aligned}$$

The inequality is obtained from the following known inequality [17]:

$$\sum_{k=0}^{N} f(k)(\alpha \sum_{0}^{k} f(i) + c) \geq -\frac{1}{2\alpha} c^2, \alpha = constant\,. \tag{27}$$

From the above analysis, we can observe that the system is always stable no matter what the adaptation gain is. However, this result is obtained by assuming that there is no measurement noise and other disturbances. In practice, a useful adaptation gain will be limited by the accuracy of the measurements and the disturbances. This approach gives a system that is always stable, and a large adaptation gain can be used to speed up the convergence. We can see that the encoder compensation gain adaptation offers a very simple controller structure and very simple adaptation algorithms. In general it can be used to compensate for the circular path error when the robot is required to move on a straight line. We know, however, that there are more error sources and we generally can not have perfect path following using the above algorithm although it can effectively compensate for a major portion of the errors.

# 6 Computer Simulation and Experimental Investigation

In this section, we will present a computer simulation and an experimental evaluation of the above MRAC motion controller.

## 6.1 Computer Simulation Evaluations of the Adaptive Motion Controller

To evaluate the adaptive motion controller we simulated a wall following situation where the robot was instructed to follow a straight wall. In this simulation we assume that measurements of the orientation of the robot relative to the wall were available. The orientation error was modeled as a second order function of $\Delta u^L$ and $\Delta u^R$, i.e., $\Delta\theta = \alpha_1 \Delta u^L + \beta_1 \Delta u^R + \alpha_2 \Delta u^L \Delta u^L + \beta_2 \Delta u^R \Delta u^R$. In this particular simulation, we chose $\alpha_1 = 2 \times 10^{-5}$, $\alpha_2 = 2 \times 10^{-5}$. $\beta_1 = 6 \times 10^{-5}$ and $\beta_2 = 4 \times 10^{-8}$. Random noise was added to the orientation measurements. Fig. 13(a) shows the orientation error of the robot without (A) and with (B) the adaptive controller. Fig. 13(b) shows the convergence of the encoder adaptation gain and Fig. 13(c) shows the resultant path of the robot trajectory without (A) and with (B) adaptive control. From the simulation results, we can observe that the proposed adaptive controller can effectively compensate for the systematic external errors and it exhibits good convergence characteristics.

## 6.2 Experimental Evaluations of the Adaptive Motion Controller

We tested the performance of our adaptive control method on the commercially available *LabMate* platform (Fig. 14). The *LabMate* has a square shape of 75 cm by 75 cm, and has a maximum speed of 1 m/sec. The absolute robot motion information is provided by ultrasonic sensors since they can provide positional information at very high frequency and with relatively high accuracy. The robot is equipped with three ultrasonic sensors (Fig. 6). In the experiments, the robot moved along a straight wall, while the ultrasonic sensors measured the distance between the sensor and the wall to allow compensation of the orientation of the robot.

The two sonars on the side can be used to measure the orientation of the robot relative to the wall. Each sonar is fired at least once every 80 ms. Sonar measurements were valid if they were within a certain range. To further improve the measurement accuracy, 10 readings from each sensor were gathered, then sorted in ascent order. Next the largest and smallest readings were compared. If the difference was larger than a certain threshold, these two readings were discarded. This procedure was repeated with the remaining readings, until the difference falls in the acceptable range or there were too few readings left. If there are too few readings left, the measurement will be discarded. Otherwise, the measurement is obtained by averaging the remaining acceptable readings. Experimental results show that the accuracy of our orientation measurement was about 1 degree, while the repeatability of the angle measurement was within $\pm 0.2^o$. It took about 0.7 s to get a valid orientation measurement. The controller hardware configuration is shown in Fig. 15.

In the experiments, the right drive wheel of the robot was covered with three layers of masking tapes to make the wheel diameters different. The experiment is composed of two stages: first calibration of the adaptation gain and then its usage in regular travel. In the first experiment, the robot was instructed to follow a straight wall, and ultrasonic sensors were used to measure the angle between the robot and the wall. This information was used for adaptation. Proportional and integral adaptation rule was used. The result of one experiment is shown in Fig. 16. Fig. 16(a) shows the orientation error without (A) and with (B) the adaptive controller. Clearly, when the cross-coupling control is used alone, it can not compensate for the external errors since no external sensory information is used in the cross-coupling controller. However, the adaptive motion controller successfully compensated for the external errors by adjusting encoder compensation gains based on the absolute orientation measurements. Fig. 16(b) shows the convergence of the adaptation gain. The final value is stored. From the figures, we can observe that the adaptive controller works well under the experimental conditions, and effectively compensate for the motion errors.

In the next stage of the experiment, the converged value of the encoder compensation gain was used to compensate for the motion error. The robot was instructed to follow a 2mx2m square path clockwisely with and without the compensation. The result is shown in Fig. 17. We can observe from Fig. 17 that the compensation is very effective and the motion accuracy improved significantly. We can also observe that the absolute motion accuracy is also improved significantly if we compare the result of the adaptive control with the result of cross-coupling control with untapped wheels. The reason is that there are systematic errors even when the wheel is not taped, and the adaptive controller compensates for them as well.

# 7 Conclusions

We described the design and implementation of an adaptive motion controller on a differential-drive mobile robot. The adaptive controller is used to compensate systematic motion errors through "learning," i.e., it detects the errors from absolute position measurements and attempts to correct for these errors. In order to apply the adaptation process, we used one section of a straight wall. The adaptation algorithm is based on the hyperstability theory to provide good convergence characteristics and stability, and it was found to improve the motion accuracy of the robot significantly.

*Acknowledgment*: This research is supported by the Department of Energy grant DE-FG02-86NE37969. We appreciate the efforts of Mr. Harry Alter from DOE and Frank Sweeny from ORNL in coordinating this national project.

# 8 Reference

[1] D. Nitzan, "Development of Intelligent Robots: Achievements and Issues," *IEEE Journal of Robotics and Automation*, Vol. RA-1, No. 1, pp. 3-13, Mar. 1985.

[2] P. F. Muir and C. P. Neuman, "Kinematic Modeling of Wheeled Mobile Robot," *Technical Report, CMU-RI-TR-86-12*, The Robotics Institute, Carnegie Mellon University, June 1986.

[3] L. Banta, "A Self Tuning Navigation Algorithm," *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, pp. 1313-1314, 1988.

[4] C. C. Lo, "Cross-Coupling Control of Multi-Axis Manufacturing Systems," Ph.D. Thesis, Dept. of Mechanical Engineering, The University of Michigan, Ann Arbor, MI, Apr. 1992.

[5] J. J. Craig, *Adaptive Control of Mechanical Manipulators.* Readings, MA: Addison-Wesley, 1988.

[6] J. E. Slotine and W. Li, "On the Adaptive Control of Robot Manipulators," *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, 1987.

[7] D. P. Stoten, *Model Reference Adaptive Control of Manipulators*, Somerset, England: Research Studies Press LTD., 1990.

[8] L. Feng, Y. Fainman, and Y. Koren, "Estimation of Absolute Position of Mobile System by Opto-electronic Processor," *IEEE Transactions on Man, Machine and Cybernetics*, Nov./Dec. 1992.

[9] J. Borenstein and Y. Koren, "Motion Control Analysis of a Mobile

Robot", *Journal of Dynamic Systems, Measurement, and Control*, Vol. 109. pp.73-79, June 1987.

[10] L. Banta, "Advanced Dead Reckoning Navigation for Mobile Robots." PhD Dissertation, Georgia Institute of Technology, Atlanta, Ga, 1987.

[11] K. J. Astrom and B. Wittenmark, *Adaptive Control.* Readings, MA: Addison-Wesley, 1989.

[12] G. C. Goodwin and K. S. Sin, *Adaptive Filtering, Prediction and Control.* Englewood Cliffs: Prentice Hall, 1984.

[13] A. J. Koivo and T. Guo, "Adaptive Linear Controller for Robotic Manipulators," *IEEE Transactions on Automatic Control.* Vol. 28, No. 2, 1983.

[14] S. Tosunoglu and D. Tesar, "State of the Art in Adaptive Control of Robotic Systems," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 24, No. 5, pp. 552-561, Sept. 1988.

[15] K. J. Astrom, "Theory and Applications of Adaptive Control - a Survey," *Automatica*, Vol.19, pp. 471-486, 1983.

[16] S. Dubowsky and D. T. DesForges, "The Application of Model-Referenced Adaptive Control to Robotic Manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 101, 1979.

[17] Y.D. Landau, *Adaptive Control, The Model Reference Approach.* New York: Marcel Dekker, Inc., 1979.

[18] L. Feng, Y. Koren and J. Borenstein, "A Cross-Coupling Controller for a Differential- Drive Mobile Robot", (to be submitted to publication).

# 9 Figure Captions

Fig. 1. Motion Control of Mobile Robots

Fig. 2. The Kinematics of a Differential-Drive Mobile Robot

Fig. 3. The Effects of Unequal Wheel Diameters

Fig. 4. Orientation is a Linear Function of the Distance Traveled

Fig. 5. Motion Error Decomposition

Fig. 6. Ultrasonic Sensors Are Used to Measure Robot Orientation

Fig. 7. A Self-Tuning Adaptive System

Fig. 8. A Model-Reference Adaptive System

Fig. 9. The blockdiagram of the Cross-Coupling Controller

Fig. 10. Equivalent Feedback System

Fig. 11. The Proposed Model-Reference Adaptive Control System

Fig. 12. Resultant Equivalent Feedback System

Fig. 13. The Simulated Performance of the Proposed Adaptive Motion Controller

(a). The Robot Orientation Error with and without the Adaptive Controller

(b). The Convergence of the Encoder Compensation Gain

(c). The Calculated Robot Trajectory with and without Compensation

Fig. 14. Experimental Robot

Fig. 15. The Motion Controller Hardware Configuration

Fig. 16. The Performance of Adaptive Motion Controller

(a). The Orientation Error under the Adaptive Controller

(b). The Convergence of the Encoder Compensation Gain

Fig. 17. Comparison of Repeatability for a 2mx2m Square Path under the Adaptive Controller and the Cross-Coupling controller

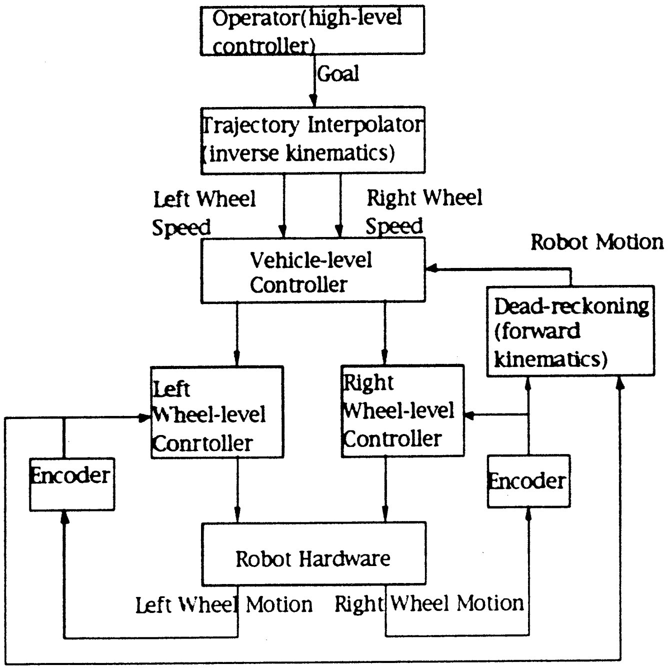
Operator(high-level controller)
Goal
Trajectory Interpolator (inverse kinematics)
Left Wheel Speed
Right Wheel Speed
Robot Motion
Vehicle-level Controller
Dead-reckoning (forward kinematics)
Left Wheel-level Conrtoller
Right Wheel-level Controller
Encoder
Encoder
Robot Hardware
Left Wheel Motion
Right Wheel Motion

Fig.1 A motion control architecture for a differential-drive mobile robot

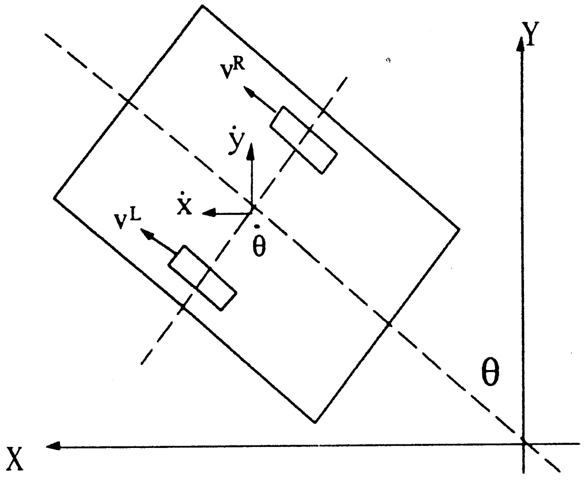

Fig.2 Kinematics of a differential-drive mobile robot

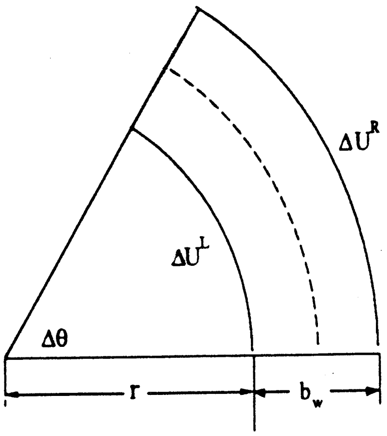

Fig.3 The effect of unequal wheel diameters

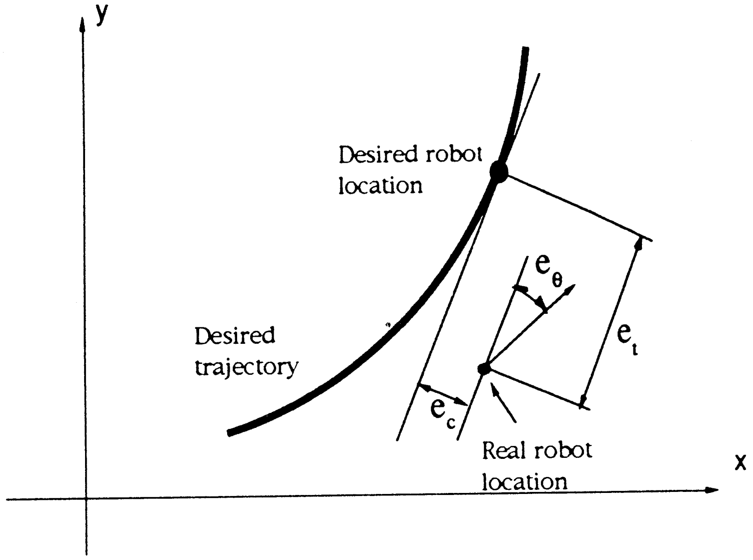

Fig.4 Motion error decomposition

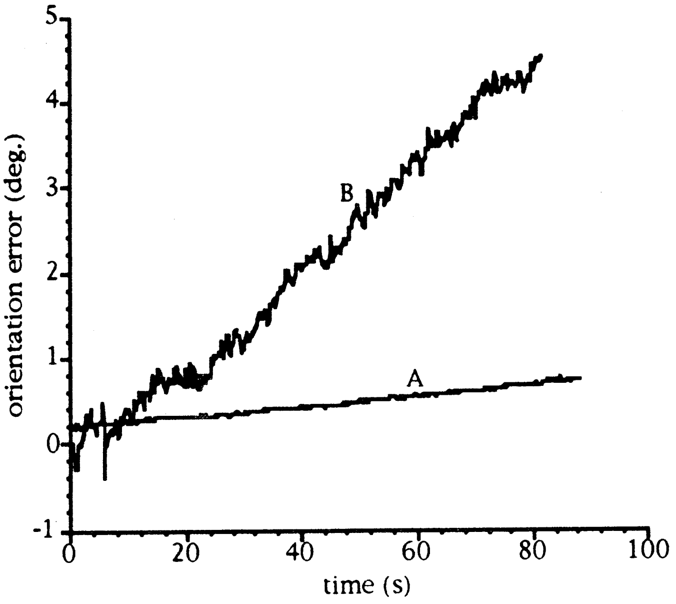

Fig.5 Orientation error as a linear function of the distance traveled A - right drive wheel is not taped, B - right drive wheel is taped

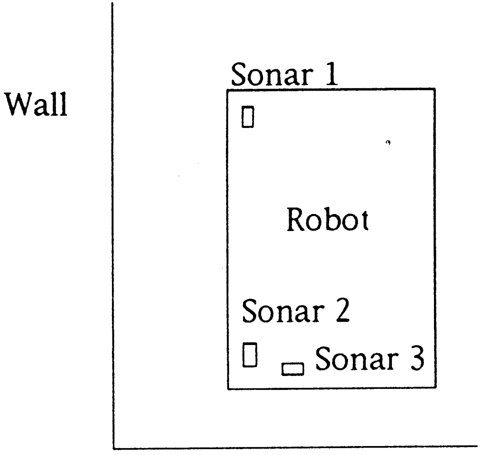

Fig.6 Ultrasonic sensors are used to measure robot orientation

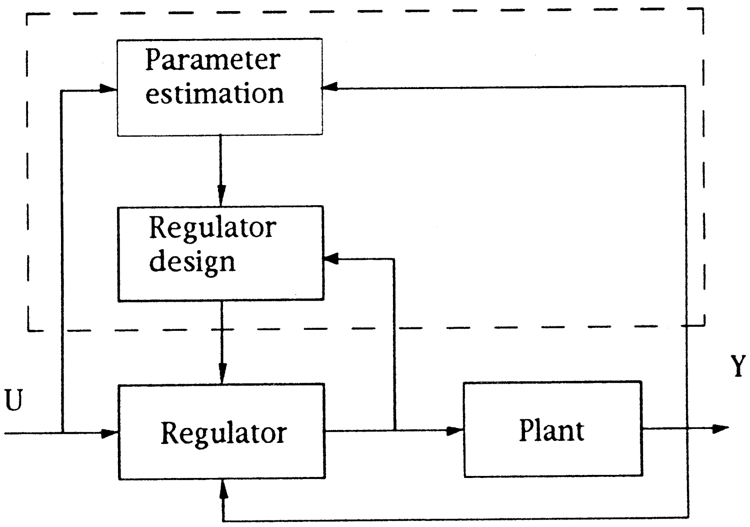

Fig.7 A self-tuning adaptive control system

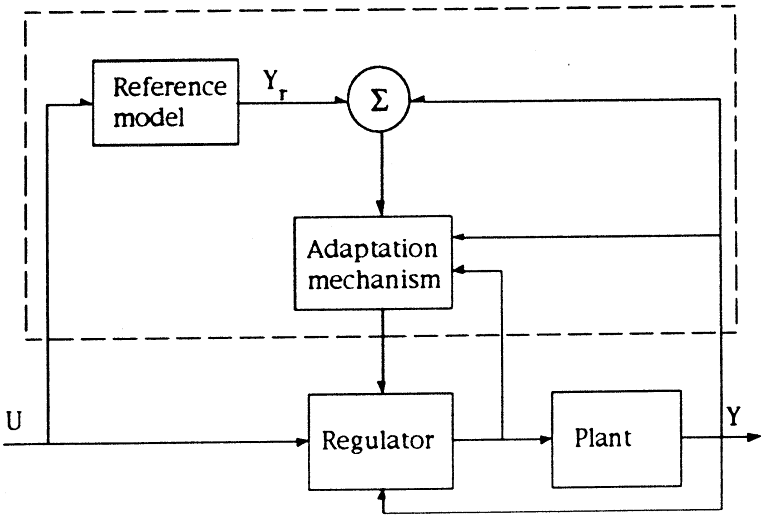

Fig.8 A model-reference adaptive control system

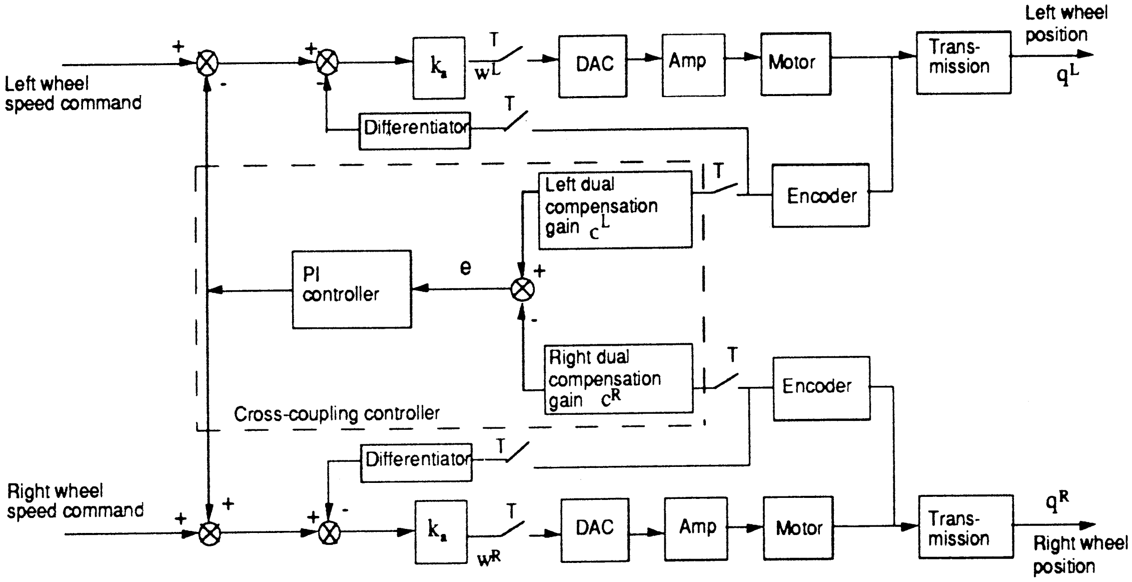

Fig.9 Block diagram of the cross-coupling controller

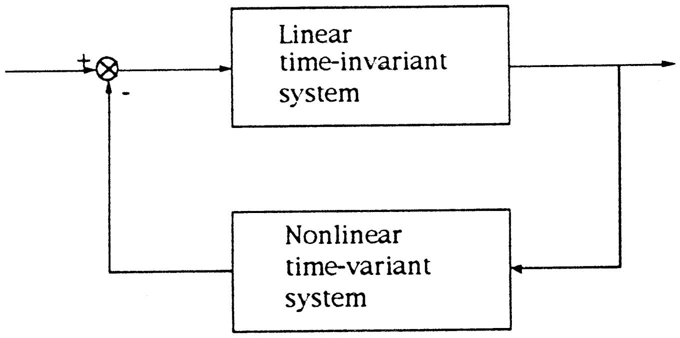

Fig.10 An equivalent feedback system

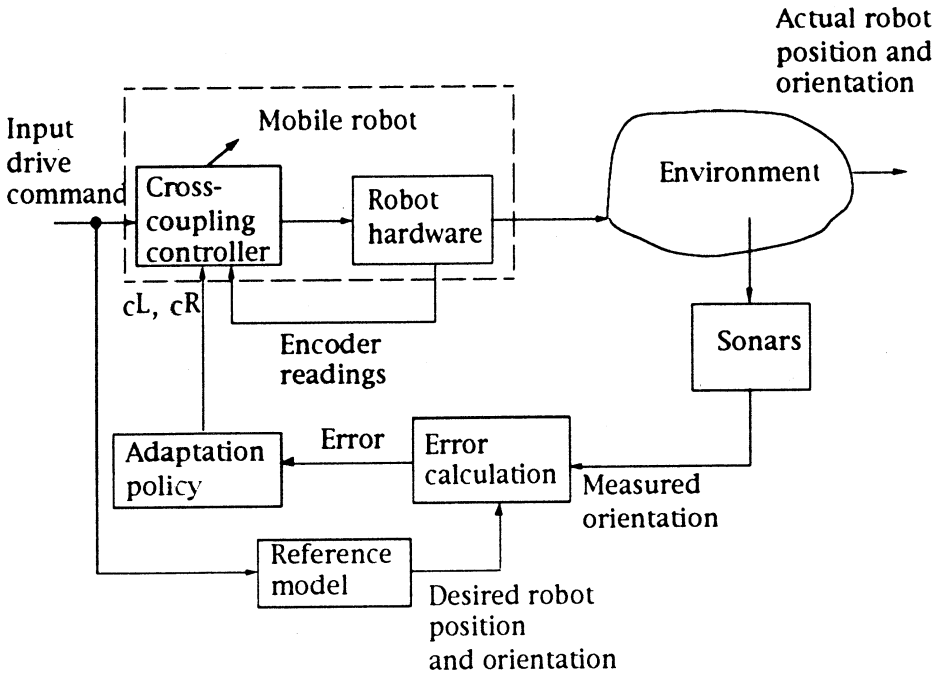

Fig.11 The proposed Adaptive motion control system

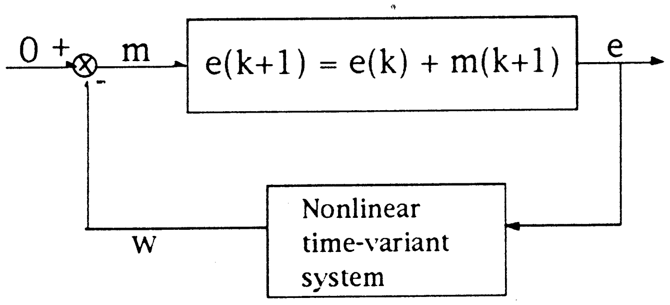

Fig.12 Resultant equivalent feedback system

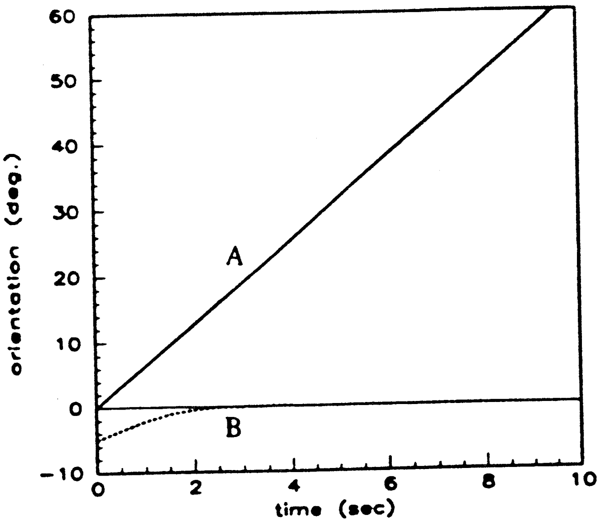

Fig.13 Computer simulation of the proposed adaptive motion controller
(a) The robot orientation error with and without the adaptive controller,
A - without compensation, B - with compensation

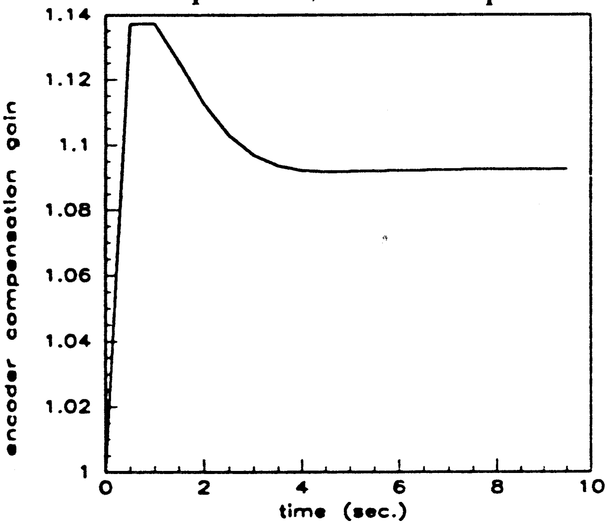

Fig.13(b) The convergence of the encoder compensation gains

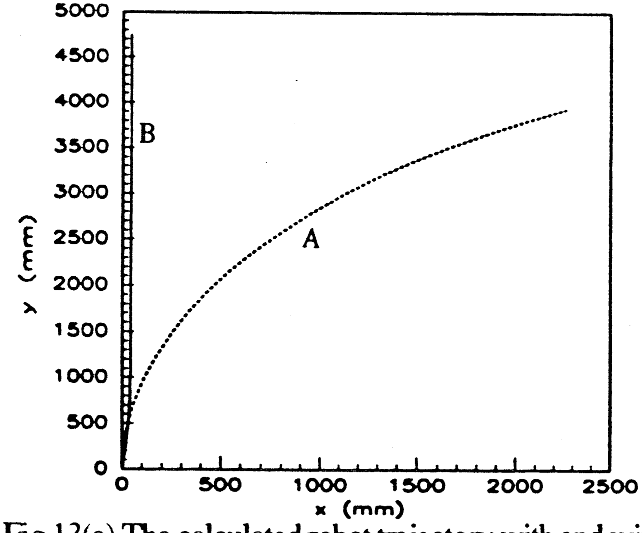

Fig.13(c) The calculated robot trajectory with and without adaptive controller

A - without compensation, B - with compensation

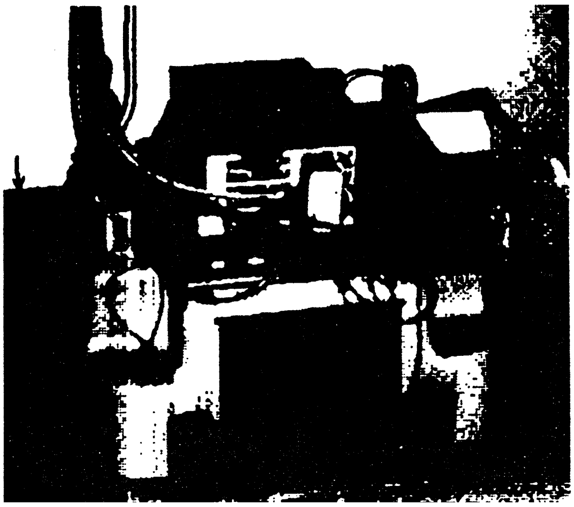

Fig.14 Experimental robot

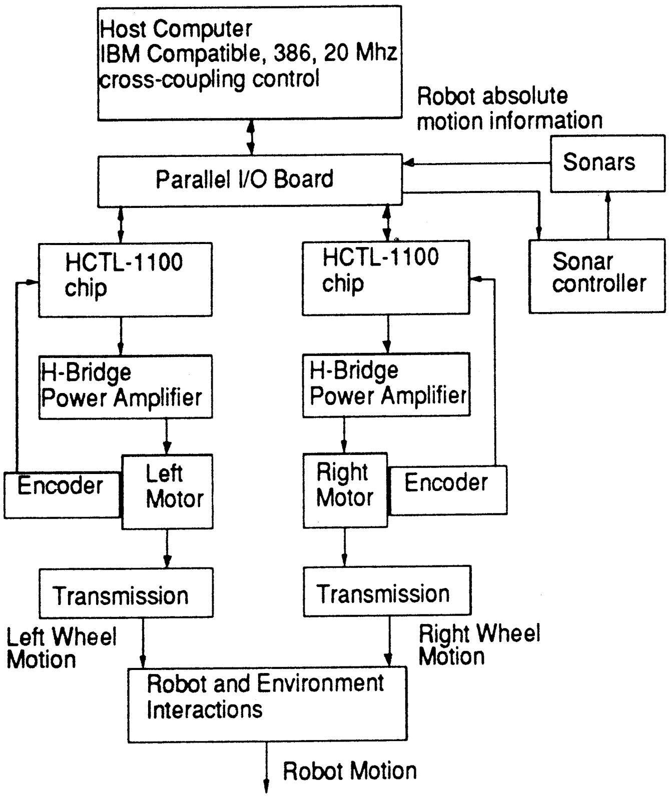

Fig.15 Motion controller hardware configuration

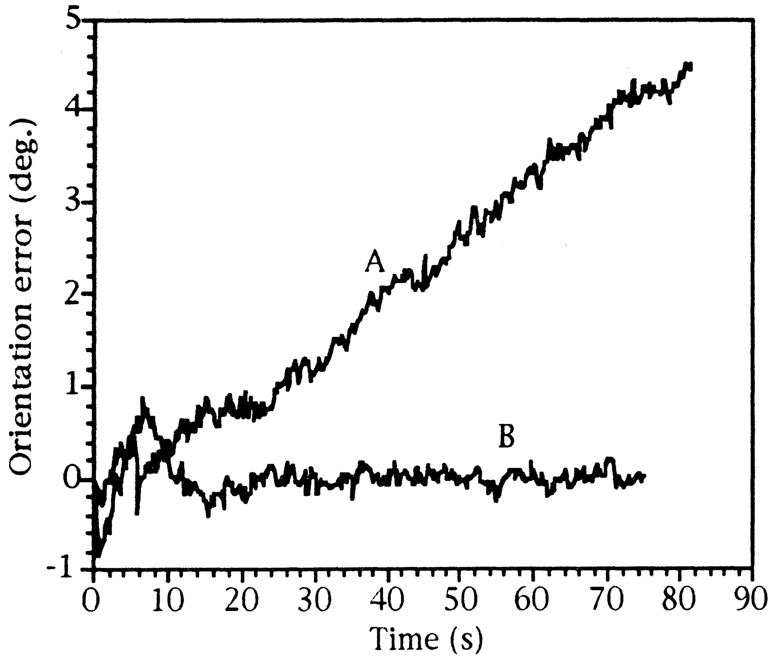

Fig.16 The performance of the proposed adaptive controller
(a) The Robot orientation error using the adaptive controller,
A - without compensation, B - with compensation

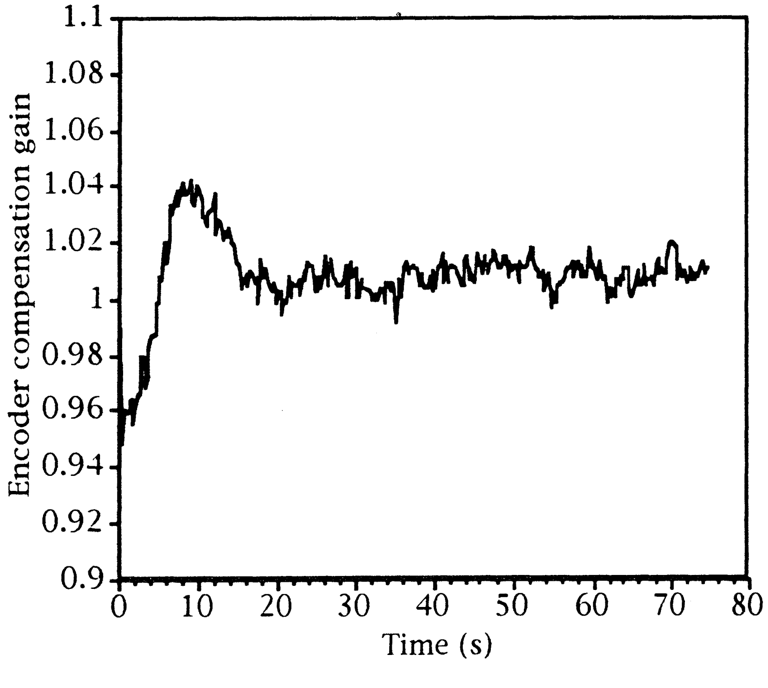

Fig.16(b) The convergence of the encoder compensation gains

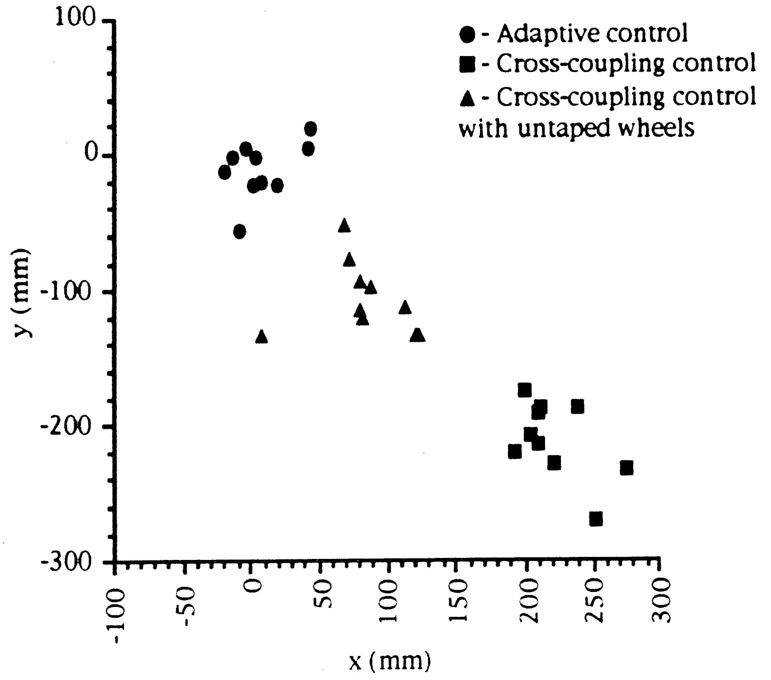

Fig.17 Comparison of motion control accuracy. The robot was preprogrammed to travel on a 2mx2m square path, starting at (0,0). The final positions after completing the path are shown.