ELSEVIER

Robotics and Autonomous Systems 17 (1996) 65–80

Robotics and Autonomous Systems

Trajectory tracking control for navigation of the inverse pendulum type self-contained mobile robot

Yun-Su Ha *, Shin'ichi Yuta

Intelligent Robot Laboratory, Institute of Information Sciences and Electronics, University of Tsukuba, 1-1-1 Tennodai, Tsukuba 305, Japan

Abstract

In this paper, we discuss the trajectory control for a wheeled inverse pendulum type mobile robot. The robot has two independent driving wheels on the same axle, and a gyro type sensor to measure the inclination angular velocity of the body and rotary encoders to measure wheel rotation. The purpose of this work is to make a robot autonomously navigate in a plane while keeping its own balance. The control algorithm consists of three parts: balance and velocity control, steering control and straight line tracking control.

We designed and implemented a vehicle command system for such robot to control using the proposed algorithm. Experiments of the navigation in a real indoor environment have been performed using our experimental robot "Yamabico Kurara".

Keywords: Wheeled inverse pendulum; Posture and velocity control; Power wheeled steering; Trajectory tracking control

1. Introduction

The focus of this work is to make a wheeled inverse pendulum type mobile robot navigate autonomously in a plane while keeping its balance. The robot is assumed to have two independent driving wheels on the same axle which support and move the robot itself, and a vibration type gyro sensor to measure the inclination angular velocity of the body and rotary encoders to measure wheel rotation.

Previous research on wheeled inverse pendulum type robots have been reported. Yamafuji and Kawamura [6] proposed a posture and driving control algorithm for a similar vehicle. They assumed that the robot has a tactile sensor to detect the angle between the ground and the body. They succeeded at balance contol but the robot could not move freely in a plane because both of its wheels were driven by a single motor. Matsumoto et al. [4] presented the estimation algorithm of posture using the adaptive observer. The presented algorithm also ignored steering control on a plane. These two papers reported a real implementation on robots and results of experiments. However, their robots were not autonomous and were connected by wires to the external computers. Koyanagi et al. [3] proposed a two dimensional trajectory control algorithm for this type of robot and built an autonomous self-contained robot with the proposed algorithm. However, the algorithm only worked while the robot moved slowly.

In this paper, we propose an algorithm for trajectory tracking control of a wheeled inverse pendulum type mobile robot which can operate freely at relatively

* Corresponding author. E-mail: hys@roboken.is.tsukuba.ac.jp.

0921-8890/96/$15.00
SSDI 0921-8890(95)00062-3

high speeds in a plane. In Section 2, we introduce to features of the wheeled inverse pendulum type self-contained mobile robot which are discussed in this paper and describe the control scheme. Section 3 is devoted to the description of our modeling method and the algorithm for posture and velocity control of the wheeled inverse pendulum type robot. The algorithm for steering and trajectory tracking control is discussed in Sections 4 and 5. We also introduce a vehicle control command system for this type of robot in Section 6. The configuration of an experimental set-up of the wheeled inverse pendulum type self-contained mobile robot "Yamabico Kurara" and the results are given in Section 7.

2. The wheeled inverse pendulum type mobile robot and its control scheme

2.1. The wheeled inverse pendulum type mobile robot

The wheeled inverse pendulum type mobile robot, which is discussed in this paper, has the following features:

(1) The robot has two driving wheels on the left and right sides of its body and no other supporting wheels.
(2) The robot is self-contained in that all necessary functions for control, such as a computer system, sensor system and battery are installed within its body.
(3) The two driving wheels of the robot are driven by independent DC motors.
(4) The robot has a gyro sensor to measure inclination angular velocity of its body.

2.2. The scheme of control systems

The motion of the robot on a plane is represented by the position vector (x, y), the direction angle ψ, the locomotion velocity v and the direction angular velocity $\dot{\psi}$. The posture of the robot is represented by the inclination angle ϕ. The configuration is shown in Fig. 1. The problem is, how to find the torque of both the left and right side motors to keep the robot's balance and track the given trajectory on the x–y plane.

In this paper, we propose an algorithm for trajectory control of such robots, which is divided into three parts. The first part is for the balancing and translational velocity control on the x–y plane. The robot was modeled as a one dimensional wheeled inverse pendulum. The control of balance and moving velocity of the robot are handled together as one problem because the posture has a deep relationship with velocity control. When the rate gyro sensor is used to estimate the inclination angle of the body, the accumulated error is a serious problem. We solved this problem by augmenting one integrator in the forward path of the control system. In this control system, the integrator cancels the effects of accumulated errors.

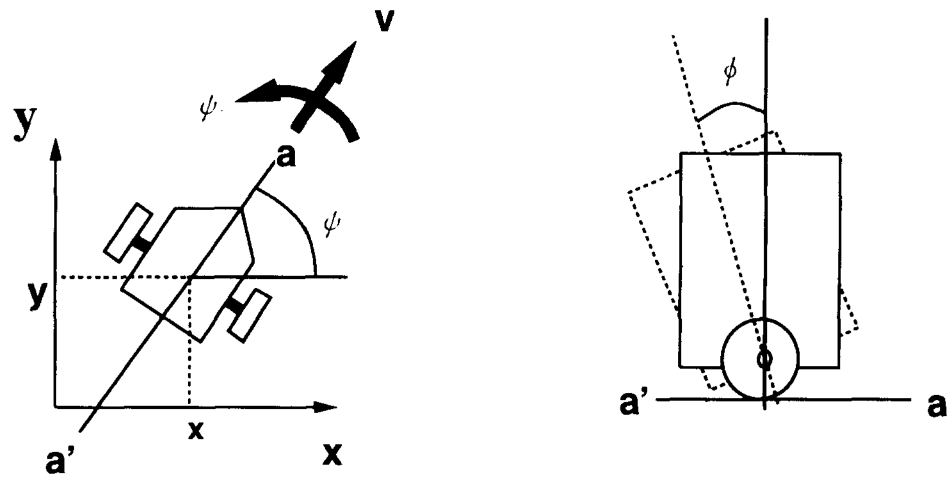


Fig. 1. Motion and posture of the robot on the x–y plane.

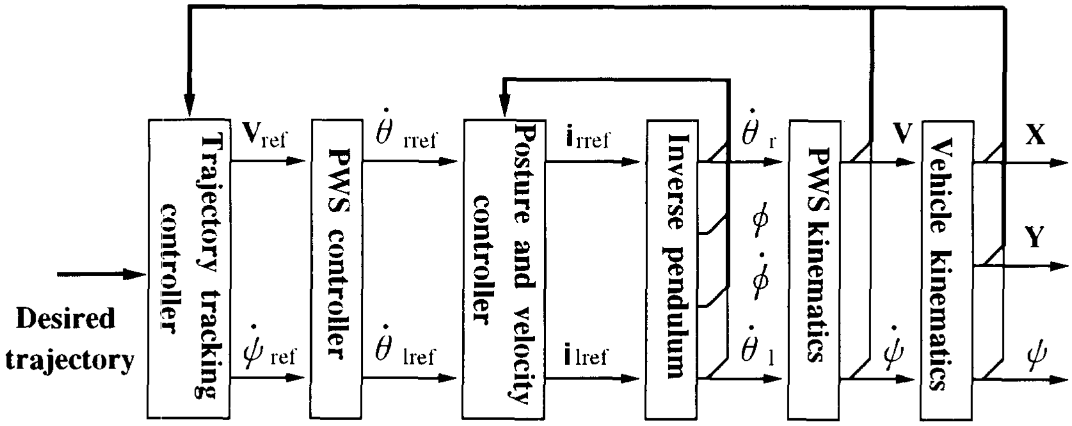


Fig. 2. Configuration of the trajectory control system for a wheeled inverse pendulum type mobile robot.

The second part is for the steering control of the robot. Mathematically, the robot cannot be modeled as a one dimensional wheeled inverse system while it is in a spinning or turning motion. However, the rotational angular velocity does not seriously affect the balance control if it is limited to small values. Hence, we assume that the steering control can be performed independently from the balance and velocity control without serious effects. The reference velocity of both the left and right wheels are calculated from the desired translational velocity and the directional angular velocity of the body. Each wheel is controlled independently to keep balance and to track the given moving velocity, where the algorithm used is based on the one dimensional model.

The third part is straight line tracking control. It is treated as a linear regulator so that the difference between the reference straight line given by the vehicle command system and the current position and direction angle of the robot goes to zero asymptotically. The algorithm we use for straight line tracking control was adapted from a similar algorithm used for a normal wheeled vehicle with supporting casters [2]. Fig. 2 shows configuration of the trajectory control system for the wheeled inverse pendulum type mobile robot.

3. Posture and velocity control

3.1. Model of the inversion and locomotion

The robot is modeled as a one dimensional inverse pendulum which rotates about the wheels' axles. Hence, the body's motion on a plane is determined by the inclination and translational motion. Fig. 3 shows the model of the robot, where θ and ϕ are the wheel's rotation angle and the inclination angle of the body, respectively, and β is the wheel's relative rotation angle to the body $\theta - \phi$. Lagrange's motion equation is used to analyze the dynamics of this model, which is given as

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T}{\partial \dot{\beta}}\right) - \frac{\partial T}{\partial \beta} + \frac{\partial U}{\partial \beta} + \frac{\partial D}{\partial \dot{\beta}} = Q_\beta, \tag{1}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T}{\partial \dot{\theta}}\right) - \frac{\partial T}{\partial \theta} + \frac{\partial U}{\partial \theta} + \frac{\partial D}{\partial \dot{\theta}} = Q_\theta, \tag{2}$$

where, T is the kinetic energy, U the potential energy, D the dissipation energy function, Q_β the external force to β axis and Q_θ the external force to θ axis, which are given as:

$$T = \tfrac{1}{2}M_\mathrm{w}(\dot{s}_2^2 + \dot{z}_2^2) + \tfrac{1}{2}M_\mathrm{b}(\dot{s}_1^2 + \dot{z}_1^2) + \tfrac{1}{2}I_\mathrm{w}\dot{\theta}^2 + \tfrac{1}{2}I_\mathrm{b}(\dot{\theta} - \dot{\beta})^2 + \tfrac{1}{2}I_\mathrm{M}\eta^2\dot{\beta}^2,$$

$$U = M_\mathrm{w}gr + M_\mathrm{b}gl\cos(\theta - \beta),$$
$$D = \tfrac{1}{2}(\mu_\mathrm{s}\dot{\beta}^2 + \mu_\mathrm{g}\dot{\theta}^2),$$
$$Q_\beta = \eta\tau_t u, \qquad Q_\theta = 0.$$

Assuming that the robot is to move in an upright posture ($\phi \simeq 0, \dot{\phi} \simeq 0$), the linearized motion equations near the upright state are given as

$$(M_\mathrm{b}l^2 + I_\mathrm{b} + \eta^2 I_\mathrm{M})\ddot{\phi} + (M_\mathrm{b}rl - \eta^2 I_\mathrm{M})\ddot{\theta} + \mu_\mathrm{s}\dot{\phi} - \mu_\mathrm{s}\dot{\theta} - M_\mathrm{b}gl\phi = -\eta\tau_t u \tag{3}$$

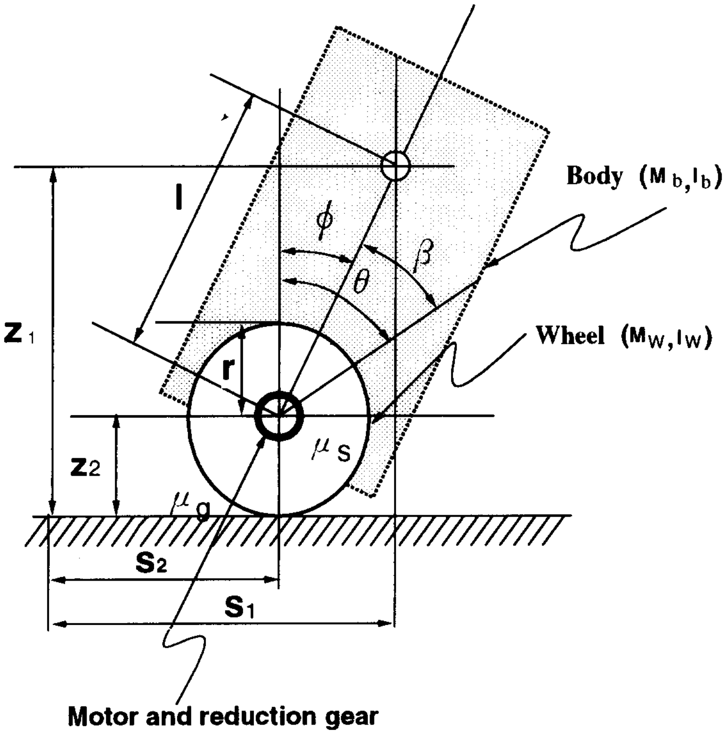


Fig. 3. The model of the wheeled inverse pendulum.

$$
\begin{aligned}
&(M_b r l + M_b l^2 + I_b)\ddot{\phi} + [(M_b + M_w)r^2 \\
&\quad + M_b r l + I_w]\ddot{\theta} + \mu_g \dot{\theta} - M_b g l \phi = 0. \qquad (4)
\end{aligned}
$$

The state equation of the linearized model is obtained as

$$\dot{X} = AX + Bu, \qquad (5)$$

where

$$A = \begin{pmatrix} 0 & 1 & 0 \\ a_1 & a_3 & a_5 \\ a_2 & a_4 & a_6 \end{pmatrix}, \qquad B = \begin{pmatrix} 0 \\ b_1 \\ b_2 \end{pmatrix},$$

$$X = [\phi \; \dot{\phi} \; \dot{\theta}]^{\mathrm{T}},$$

$$\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \\ a_5 & a_6 \end{pmatrix} = \frac{1}{\Delta} \begin{pmatrix} (a_{22} - a_{12}) M_b g l & (a_{11} - a_{21}) M_b g l \\ -\mu_s a_{22} & \mu_s a_{21} \\ (\mu_s a_{22} + \mu_g a_{12}) & -(\mu_s a_{21} + \mu_g a_{11}) \end{pmatrix},$$

$$\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \frac{1}{\Delta} \begin{pmatrix} -a_{22} \eta \tau_t \\ a_{21} \eta \tau_t \end{pmatrix},$$

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} M_b l^2 + I_b + \eta^2 I_M & M_b r l - \eta^2 I_M \\ M_b r l + M_b l^2 + I_b & (M_b + M_w) r^2 + M_b r l + I_w \end{pmatrix},$$

$$\Delta = a_{11} a_{22} - a_{12} a_{21}.$$

The parameters and variables in Eq.(5) are defined in Table 1, where the values are for our experimental robot Yamabico Kurara.

3.2. The control law for posture and velocity control

The purpose of this section is to derive the control law to move at a given reference velocity while keeping the body balanced. If the reference rotational angular velocity $\dot{\theta}_{\mathrm{ref}}$ of the wheel is constant, the steady state vector X_s and steady control input u_s are derived as Eq. (6) from Eq. (5).

Table 1
Parameters and variables

Symbol	Parameter and variable name	Unit and value[a]	
M_b	Mass of the body	(kg)	9.01
M_w	Mass of the wheel	(kg)	0.51
I_b	Rotational inertia of the body	(kgm^2)	0.228
I_w	Rotational inertia of the wheel	(kgm^2)	5.1×10^{-4}
I_M	Rotational inertia of the motor axis	(kgm^2)	3.2×10^{-6}
r	Radius of the wheel	(m)	0.062
l	Length between the wheel axle and the center of gravity of the robot's body	(m)	0.138
μ_s	Coefficient of friction between the wheel axle including motor and gear	(Nm/(rad/s))	5.76×10^{-3}
μ_g	Coefficient of friction between the wheel and the ground	(N·m/(rad/s))	4.25×10^{-3}
τ_t	Torque constant of the motor	(Nm/A)	0.0235
η	Reduction ratio of gears		39.5
g	Acceleration due to gravity	(m/s^2)	9.8
ϕ	Inclination angle of the body	(rad)	
θ	Wheel's rotation angle	(rad)	
u	Motor's input current	(A)	

[a] Values are for Yamabico Kurara.

$$\begin{pmatrix} X_s \\ --- \\ u_s \end{pmatrix} = \begin{pmatrix} f(\dot{\theta}_{ref}) \\ --- \\ g(\dot{\theta}_{ref}) \end{pmatrix} = \begin{pmatrix} \dfrac{-(a_5 b_2 - a_6 b_1)}{(a_1 b_2 - a_2 b_1)} \dot{\theta}_{ref} \\ 0 \\ \dot{\theta}_{ref} \\ ------ \\ \dfrac{-(a_1 a_6 - a_2 a_5)}{(a_1 b_2 - a_2 b_1)} \dot{\theta}_{ref} \end{pmatrix}. \tag{6}$$

Here, defining ΔX and Δu as

$$\begin{pmatrix} \Delta X \\ \Delta u \end{pmatrix} = \begin{pmatrix} X - X_s \\ u - u_s \end{pmatrix} \tag{7}$$

the problem can be treated as the design of a state feedback regulator for the system represented by ΔX and Δu.

Generally, the modeling errors due to disregard of nonlinear factors such as the backlash of the reduction gear and nonlinear friction are unavoidable. Also an accurate identification of the parameters is not easy. Therefore, a counterplan against the modeling errors, the parameters variations and disturbance should be considered while designing controllers. In the case that the inclination angle is estimated by integration of measured values using the gyro sensor, accumulated errors are included. Hence, we augmented the control system to cope with these errors. Defining a new state variable z as

$$z = \int_0^t (\dot{\theta} - \dot{\theta}_{ref})\, dt \tag{8}$$

the augmented system ($\tilde{S}$) [5] is represented as

$$\Delta \dot{\tilde{X}} = \tilde{A} \Delta \tilde{X} + \tilde{B} \Delta u, \tag{9}$$

$$\Delta y = \tilde{C} \Delta \tilde{X}, \tag{10}$$

where

$$\Delta \tilde{X} = \begin{pmatrix} \Delta X \\ z \end{pmatrix}, \quad \tilde{A} = \begin{pmatrix} A & 0 \\ C & 0 \end{pmatrix},$$

$$\tilde{B} = \begin{pmatrix} B \\ 0 \end{pmatrix}, \quad \tilde{C} = \begin{pmatrix} C^T \\ 0 \end{pmatrix}^T, \quad C = [0\ 0\ 1].$$

The optimal controller which asymptotically stabilizes the feedback system of the augmented system and minimizes the performance index J:

$$J = \int_0^\infty (\Delta \tilde{X}^T \tilde{Q}\, \Delta \tilde{X} + \Delta u^T \tilde{R}\, \Delta u)\, dt, \tag{11}$$

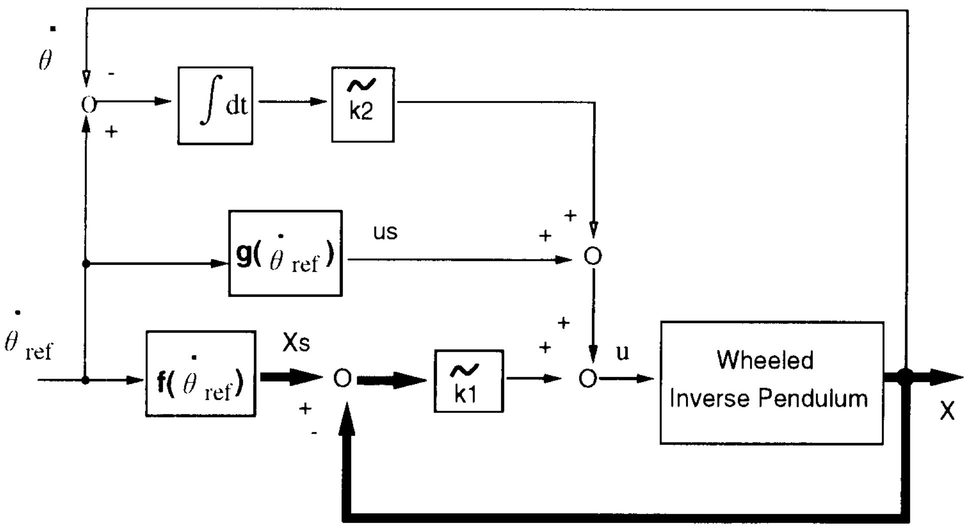


Fig. 4. Block diagram of the posture and velocity control system.

where $\tilde{Q} = \tilde{Q}^{\mathrm{T}} \geq 0$, $\tilde{R} = \tilde{R}^{\mathrm{T}} > 0$ is given by

$$u = u_s - \tilde{K}\Delta\tilde{X}$$
$$= u_s - \tilde{K}_1(X - X_s) - \tilde{k}_2 \int_0^t (\dot{\theta} - \dot{\theta}_{\mathrm{ref}})\, \mathrm{d}t, \qquad (12)$$

where $\tilde{K} = \tilde{R}^{-1}\tilde{B}^{\mathrm{T}}\tilde{P}$ and $\tilde{P}$ is the solution of the matrix Riccati equation.

The posture and velocity control system is realized by a linear state feedback and feedforward controller as Eq. (12) (see Fig. 4).

4. Steering control

When the robot spins itself to change direction on a plane, i.e. $\dot{\psi} \neq 0$, the value $\dot{\psi}$ affects the dynamics of balancing control, and Eqs. (1) and (2) are no longer exact. However, we can assume the robot's dynamics keeps Eqs. (1) and (2) approximately, when the value $\dot{\psi}$ is small. Also, we regard the robot system as consisting of two independent inverse pendulums which are mounted on both wheels, which can be controlled independently. We can apply the Power Wheeled Steering method for heading control of the wheeled inverse pendulum type mobile robot based on these assumptions.

The algorithm to control such a non-holonomic system should be divided into two steps:

(a) Control to make a vehicle track the reference translational velocity v and directional angular velocity $\dot{\psi}$

(b) Determination of locomotion velocity v and directional angular velocity $\dot{\psi}$ of the vehicle based on the deviation between the current position and direction, and the reference trajectory.

Step (a) is discussed in this section, and step (b) is described in Section 5.

In the case of the power wheeled steering type vehicle (Fig. 5), the moving velocity v and directional angular velocity $\dot{\psi}$ relate to the left and right wheel's rotational angular velocity $\dot{\theta}_l, \dot{\theta}_r$ as

$$\begin{pmatrix} v \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} \frac{R_{\mathrm{r}}}{2} & \frac{R_{\mathrm{l}}}{2} \\ \frac{R_{\mathrm{r}}}{L} & -\frac{R_{\mathrm{l}}}{L} \end{pmatrix} \begin{pmatrix} \dot{\theta}_{\mathrm{r}} \\ \dot{\theta}_{\mathrm{l}} \end{pmatrix}, \qquad (13)$$

where L is the distance between the left and right wheels, and R_{l} and R_{r} are radius of the left and right wheels. The steering and moving velocity control of the robot are realized by controlling both the left and right wheel's rotational angular velocity. The reference rotational angular velocity for the posture and velocity control system (Fig. 4) is given as

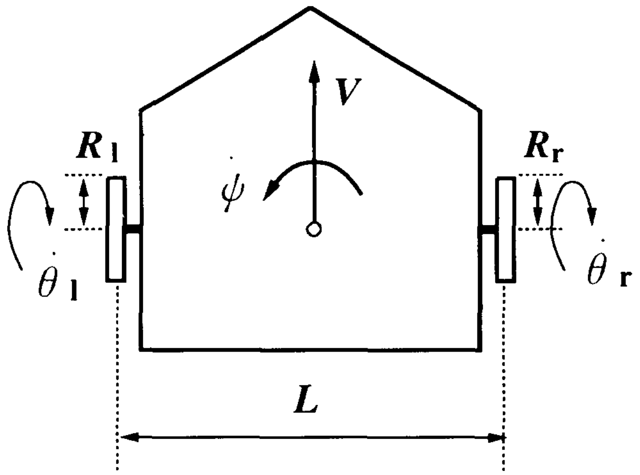


Fig. 5. Parameters for a vehicle controlled by the PWS method.

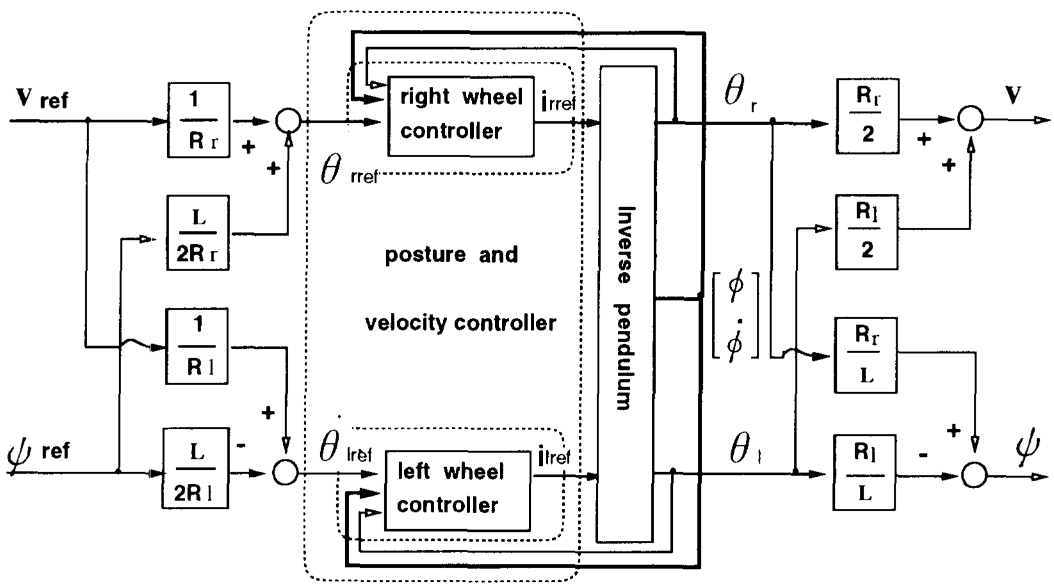


Fig. 6. The heading and velocity control system.

$$\begin{pmatrix} \dot\theta_{\text{rref}} \\ \dot\theta_{\text{lref}} \end{pmatrix} = \begin{pmatrix} \dfrac{1}{R_r} & \dfrac{L}{2R_r} \\ \dfrac{1}{R_l} & -\dfrac{L}{2R_l} \end{pmatrix} \begin{pmatrix} v_{\text{ref}} \\ \dot\psi_{\text{ref}} \end{pmatrix}. \tag{14}$$

The block diagram of the control system to perform step (a) is shown in Fig. 6. The inclination angle ϕ and angular velocity $\dot\phi$ are obtained from a gyro sensor, which are commonly used for the controller of both wheels.

5. Straight line tracking control

To make the robot track along a given trajectory in a plane, the reference value of moving velocity v_{ref} and

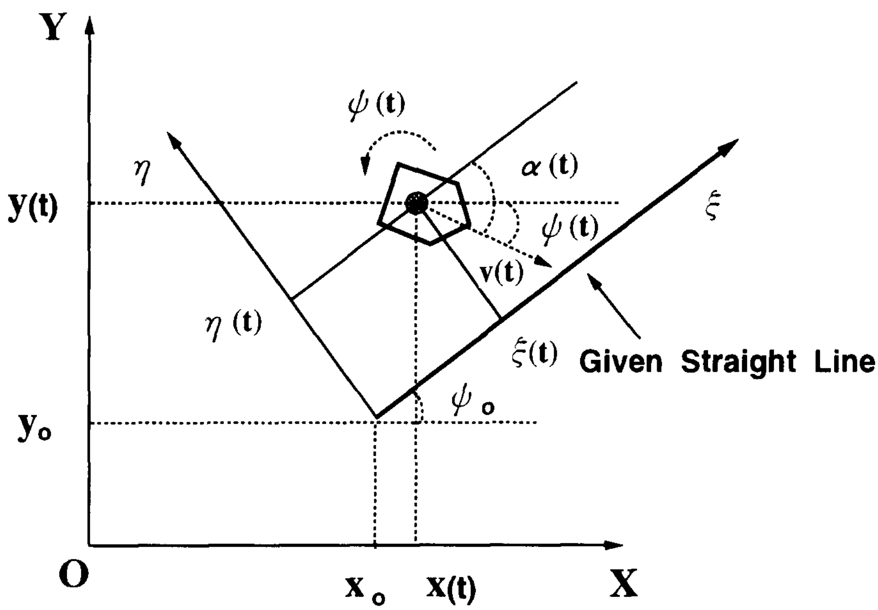


Fig. 7. Configuration of straight line tracking control.

direction angular velocity $\dot{\psi}_{\mathrm{ref}}$ must be calculated. In this section, we describe that algorithm.

5.1. Heading control for straight line tracking

Let us consider the case where the robot is to track along a straight line on a plane passing through a point (x_o, y_o) with angle ψ_o from the x axis. The robot's current position and direction are estimated as $x(t)$, $y(t)$ and $\psi(t)$, by dead reckoning or some other positioning system. At first, we represent the robot position and direction on the ξ–η coordinate, on which the ξ axis denotes the given straight line to track. The robot position and direction is denoted by $\xi(t)$, $\eta(t)$ and $\alpha(t)$. Then, the problem is formulated as "decide the angular velocity $\dot{\psi}(t)$ so that $\eta(t)$ and $\alpha(t)$ go to zero asymptotically, while $v(t)$ remains constant and $\xi(t)$ increases with given velocity". So the problem can be treated as a regulator problem after linearization based on the assumption that $\eta(t)$ and $\alpha(t)$ are small. The reference directional angular velocity at the next sampling instant should be calculated from negative feedback of $\eta(t)$ and $\alpha(t)$, i.e.

$$\dot{\psi}_{\mathrm{ref}}(t+\Delta t) = -k_\eta \eta(t) - k_\alpha \alpha(t). \tag{15}$$

5.2. Translational velocity control for starting and stopping

A sudden change of the reference velocity for starting and stopping may cause the robot to be unstable. Hence, the reference velocity is given in three stages: the acceleration section (t_0 - t_a), the ordinary operating section (t_a - t_d), the deceleration section (t_d - t_s), as shown in Fig. 8. The reference velocity to start and stop are calculated as follows:

(1) *Calculation of the reference velocity in acceleration section.* The reference velocity is given as

$$v_{\mathrm{ref}}(t+\Delta t) = v_{\mathrm{ref}}(t) + a \cdot \Delta t \quad : v_{\mathrm{ref}}(t) < v_d, \tag{16}$$

where Δt is sampling interval, a is given acceleration ratio and v_d is the desired velocity from the locomotion control command.

(2) *Calculation of reference velocity in deceleration section for stopping.* The calculation of reference velocity to make the robot stop at the goal position exactly is not a simple problem. The starting time of deceleration cannot be predicted exactly because of the effect of response delay time or steady-state error in the control system.

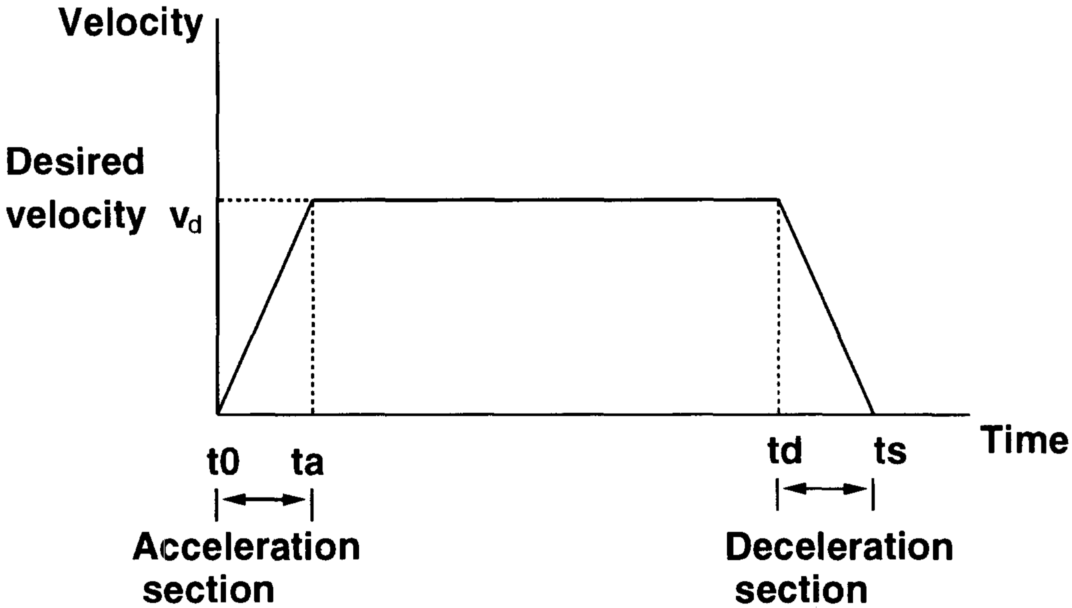


Fig. 8. Reference velocity for the robot.

Letting the velocity and position at any time t be $v(t)$ and $x(t)$, $v(t)$ and the position x_{stop} where robot should stop are given by

$$v(t) = a \cdot (t_s - t), \tag{17}$$

$$x_{\text{stop}} = x(t) + \int_{t}^{t_s} a \cdot (t_s - t)\,\mathrm{d}t, \tag{18}$$

where t_s is the time when the robot will stop. The velocity $v(t)$ can be rewritten from Eqs. (17) and (18):

$$v(t) = \sqrt{2a(x_{\text{stop}} - x(t))}. \tag{19}$$

Hence, if the control system has no response delay time and steady-state error, the reference velocity can be calculated as follows.

$$v_{\text{ref}}(t + \Delta t) = \begin{cases} v_{\text{ref}}(t) \\ \quad : v(t) < \sqrt{2a(x_{\text{stop}} - x(t))}, \\ v_{\text{ref}}(t) - a \cdot \Delta t \\ \quad : v(t) \geq \sqrt{2a(x_{\text{stop}} - x(t))}. \end{cases} \tag{20}$$

To avoid the run-over at the stop position x_{stop} caused by such effects, we propose a formula to calculate the reference velocity for stopping as Eq. (21).

$$v_{\text{ref}}(t + \Delta t) = \begin{cases} v_{\text{ref}}(t) \\ \quad : v(t) < \sqrt{2a[(x_{\text{stop}} - \varepsilon) - x(t)]}, \\ v_{\text{ref}}(t) - a \cdot \Delta t \\ \quad : v(t) \geq \sqrt{2a[(x_{\text{stop}} - \varepsilon) - x(t)]}, \end{cases} \tag{21}$$

where ε is an appropriate small value.

6. Locomotion control command system

A command system named "HYS", which is used to define the reference trajectory and other information for this type of robot has been designed. It is similar to the Spur [2] command system for autonomous mobile robots proposed by our group. The usual HYS commands denote the coordinate parameters on the two dimensional plane to define the line or position. Each command is represented as a function in the C language.

The important HYS commands are as follows:

(1) Straight line command: HYS_line_LC(x_{line}, y_{line}, ψ_{line}).

To define the straight line along which the robot should track.

(2) Stop command: HYS_stop_LC($x_{\rm stop}$, $y_{\rm stop}$, $\psi_{\rm stop}$) and HYS_stop( ).
To define the position and direction in which the robot should stop and emergency stop.

(3) Spin command: HYS_spin_LC($\psi_{\rm spin}$).
To make the robot spin to an angle of $\psi_{\rm spin}$ from the x axis.

(4) Position adjustment command:
HYS_adj_pos_LC($x_{\rm adj}$, $y_{\rm adj}$, $\psi_{\rm adj}$).
To replace the internally estimated position by the given parameters which are usually determined by detecting the environment with the external sensors.

(5) The other commands.
HYS_set_vel(v): To set the reference moving velocity of the robot.
HYS_set_angv($\dot{\psi}$): To set the reference directional angular velocity for the spin motion.
HYS_get_pos_LC($\&x, \&y, \&\psi$): To monitor the current position and direction.
HYS_get_vel_angv_pos($\&v, \&\dot{\psi}, \&\phi$): To monitor the current moving velocity, directional angular velocity and posture of the robot.

Using these commands, the navigation program can easily control the robot trajectory in real time based on the map information and external sensory data.

7. Experiments using Yamabico

7.1. Experimental system and determination of parameters

The robot used for experiments was a wheeled inverse pendulum type self-contained mobile robot Yamabico Kurara from which the front and rear casters had been removed from the standard type Yamabico robot. Yamabico are a series of self-contained autonomous mobile robot platforms for experimental research which were developed by the authors' group. Fig. 9 shows a self-contained autonomous inverse pendulum type experimental robot "Yamabico Kurara" which is controlling itself to keep its own balance. The robot has rotary encoders (resolution:2000) and a vibration type gyro sensor (TOKIMEC Co. TFG-160) to detect wheel rotation and the body's inclination angular velocity. Both wheels are driven by DC motors of 10 W. Ultra sonic sensors to recognize the environment are attached on the front, left and right sides of body.

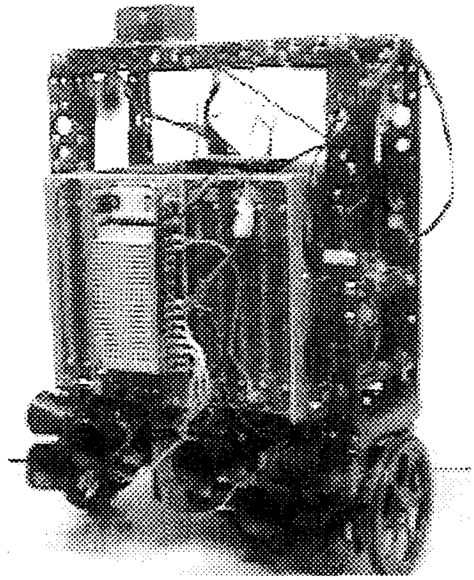


Fig. 9. The inverse pendulum type self-contained mobile robot Yamabico Kurara.

By using the parameters given in Table 1, the coefficient matrices $\tilde{A}$ and $\tilde{B}$ of the augmented system $(\tilde{S})$ are found to be

$$\tilde{A} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 38.23 & -0.085 & 0.121 & 0 \\ -51.76 & 0.347 & -0.56 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

$$\tilde{B} = \begin{pmatrix} 0 \\ -13.7 \\ 56.02 \\ 0 \end{pmatrix}$$

In the posture and velocity control experiment, the weighting matrices of performance index J were fixed as $\tilde{Q} = diag(1, 0.01, 0.01, 4)$ and $\tilde{R} = 300$. As a result, feedback gains of the control system represented in Eq. (12) were given as $\tilde{K}_1 = (-9.0, -1.58, -0.123)$, $\tilde{k}_2 = -0.115$, and

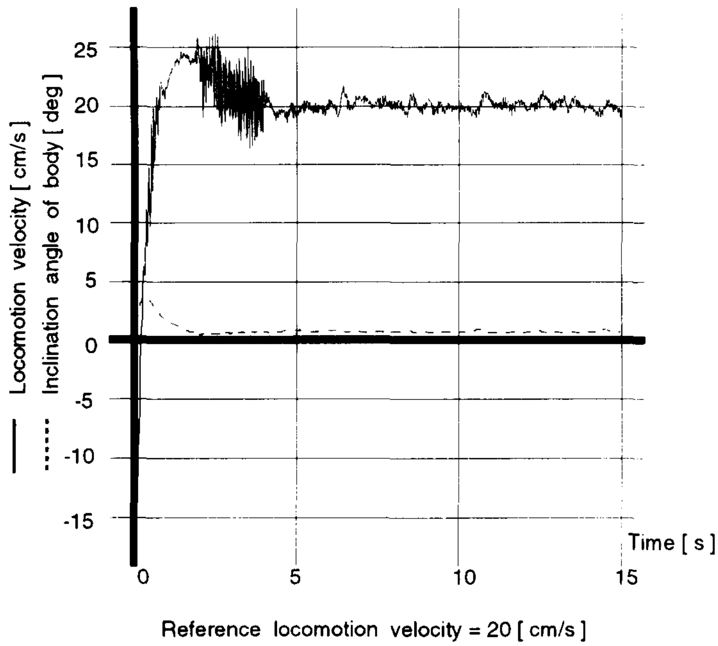


Fig. 10. Experimental results for posture and velocity control.

Fig. 11. Experimental scene on sloped rough ground.

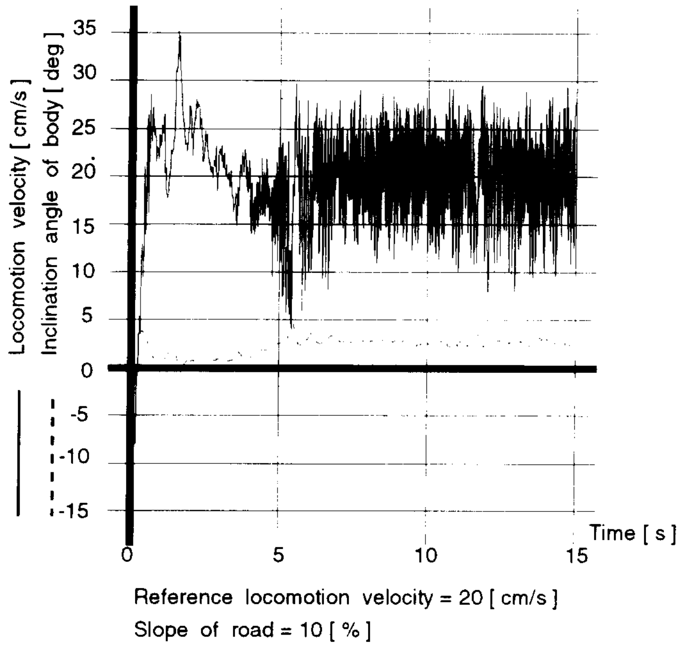


Fig. 12. Experimental results for posture and velocity control on sloped rough ground.

in this case poles of closed-loop system were calculated as λ_1, λ_2 = $(-6.177 \pm 0.119j)$ and λ_3, λ_4 = $(-1.529 \pm 0.1412j)$. The papramenters k_η, k_α in Eq. (15) were determined to be 0.00457 and 1, respectively.

An MC 68000 micro-processor was used as a controller, and all computation was performed in a fixed-point representation. The sampling interval of the control program is 5 ms. The motor input current is controlled using the feedforward current control method [1].

7.2. Experiments

(1) *Experiments for the posture and velocity control.* At first, the step response experiment to test the performance of the posture and velocity control system was performed. One of the results is shown in Fig. 10, which shows that the robot can track the reference velocity while keeping its balance. Fig. 11 shows the experimental scene on sloped rough ground. The result is shown in Fig. 12. In spite of the roughness and slope of the road, the robot tracked the reference velocity while keeping balance of posture. This result shows that the proposed algorithm is robust enough for the indoor environment.

(2) *Experiments to verify our assumption on applying PWS method.* We assume that the steering control can be done independently without serious effects upon the balance of the body if the rotational angular velocity is limited to small values. We checked the validity of this assumption by examining the posture of the body at corners A and B when the trajectory as shown in Fig. 13 is given for the robot. We found that the robot could track the given trajectory and maintain the stability of posture while performing a turning motion. The result is shown in Fig. 14.

(3) *Experiments for trajectory tracking and position control.* Fig. 15 shows the result of trajectory tracking control experiment when the desired straight line is given for the robot moving with velocity 25 cm/s at points A and B on the x-y plane. Fig. 16 shows one of the experimental result using the reference velocity is calculated by algorithm as Eq. (21) for position control.

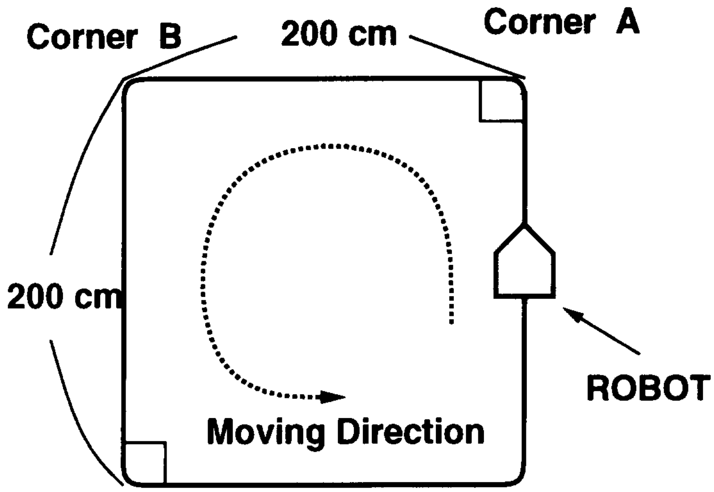


Fig. 13. The trajectory to test posture control in turning motion.

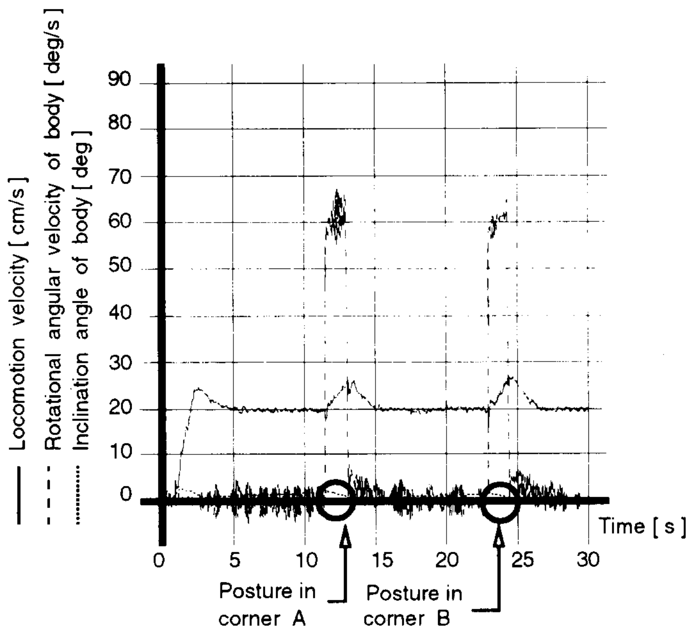


Fig. 14. Experimental results of posture stability in turning motion.

(4) *Experiments of indoor navigation.* We also performed an experiment in indoor navigation to test the performance of the total control system and the validity of the command system. The Intelligent Robot Laboratory at the University of Tsukuba was used for as the experimental environment. Fig. 17 shows the robot moving in a real environment.

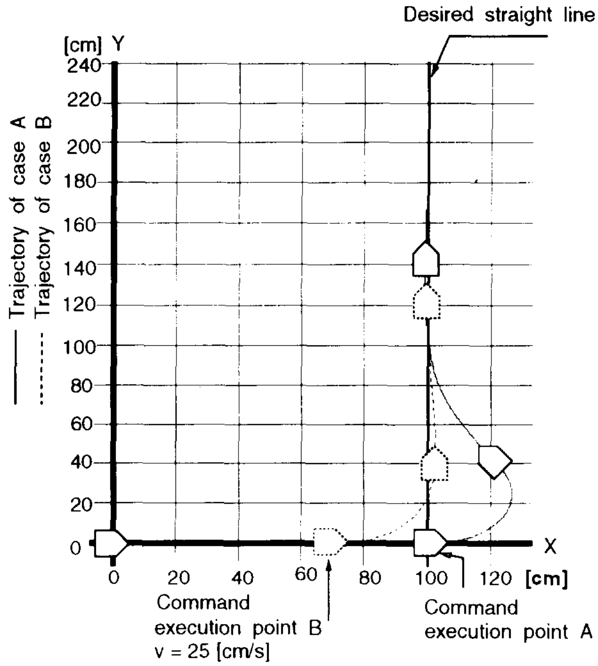


Fig. 15. Experimental results for trajectory tracking control.

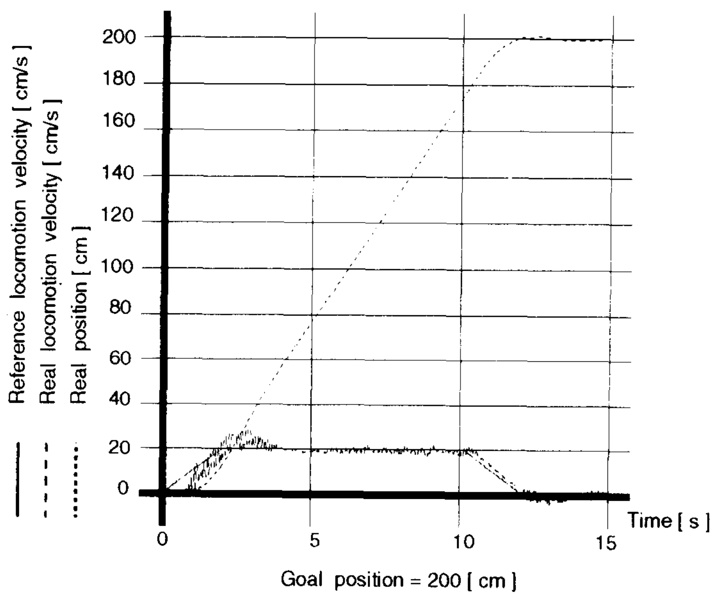


Fig. 16. Experimental results for position control.

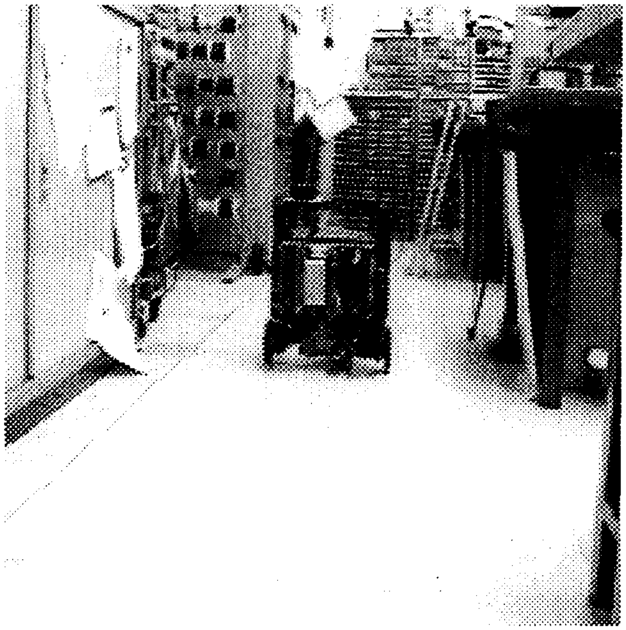


Fig. 17. Experiment in indoor navigation.

8. Conclusion

In this paper, we proposed a control algorithm to make the inverse pendulum type mobile robot move in a plane, with the desired constant velocity, and while keeping the balance of posture by itself. The control algorithm was implemented and tested on the experimental robot. Through experiments using the robot, we concluded:

(1) In spite of modeling errors, parameter variation, and the accumulated errors in the inclination angle, the robot can move robustly at the desired velocity and while keeping its balance.

(2) It is possible to separate the steering control from the posture and velocity control without locomotion velocity and direction angular velocity being constrained too much.

(3) Sensor based navigation, while keeping its balance, has been realized using the implemented command system.

References

[1] S. Iida and S. Yuta, Feedforward current control method using two-dimensional table for DC servo motor software servo system, *Proc. of the 1988 IEEE Int. Conf. of Industrial Electronics* (1988) 466–471.

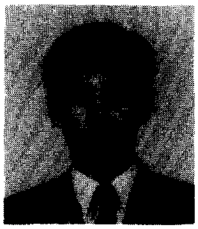
[2] S. Iida and S. Yuta, Vehicle command system and trajectory control for autonomous mobile robot, *Proc. of the 1991 IEEE Int. Workshop on Intelligent Robots and Systems* (1991) 212–217.

[3] E. Koyanagi, S. Iida, K. Kimoto and S. Yuta, A wheeled inverse pendulum type self-contained mobile robot and its two-dimensional trajectory control, *Proc. of ISMCR'92* (1992) 891–898.

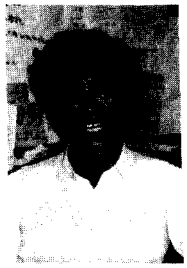
[4] O. Matsumoto, S. Kajita and K. Tani, Attitude estimation of the wheeled inverted pendulum using adaptive observer, *Proc. of 9th Academic Conf. of the Robotics Society of Japan* (1991) 909–910 (in Japanese).

[5] J. Medanic and Z. Uskokovic, The design of optimal output regulators for linear multivariable system with constant disturbances, *International Journal Control* 37(4) (1983) 809–830.

[6] K. Yamafuji and T. Kawamura, Postural control of a monoaxial bicycle, *Journal of the Robotics Society of Japan* 7(4) (1989) 74–79 (in Japanese).



Yun-Su Ha was born in Kyeongju, Korea, in 1962. He received B.E. and M.S. degrees in control engineering from Korea Maritime University, Pusan, Korea, in 1986 and 1990, respectively. From 1989 to 1992 he was an Assistant at Korea Maritime University. He is now a Ph.D. candidate in electronics and information science at University of Tsukuba, Ibaraki, Japan. He is interested in development of locomotion control system for a wheeled autonomous mobile robot.



Shin'ichi Yuta received B.E., M.Sc. and Ph.D. degrees in electrical engineering from Keio University, Yokohama, Japan, in 1970, 1972, and 1975, respectively. From 1975 to 1978 he was an Assistant Professor at Tokyo University of Agriculture and Technology, Tokyo, Japan, where his research subjects were medical imaging and ultrasonic image processing. Since 1978 he has been with the Institute of Information Science and Electronics, University of Tsukuba, Ibaraki, Japan, and is now a professor and leads the Intelligent Robot Lab. His current research interests include controller architecture of intelligent robots, robot sensors, robot vision, intelligent vehicle control, and construction of experimental autonomous mobile robots, as well as signal and image processing. Dr. Yuta is a member of the IEEE Robotics and Automation Society, RSJ and other academic societies.