IEEE/RSJ International Workshop on Intelligent
Robots and Systems IROS '91, Nov. 3-5, 1991,
Osaka, Japan. IEEE Cat. No. 91TH0375-6

Path Tracking Control of Trailer-Like Mobile Robot

M. Sampei, T. Tamura, T. Itoh and M. Nakamichi

Department of Electrical and Electronics Engineering
Chiba University
1-33 Yayoi-cho, Chiba 260, JAPAN

Abstract

In this paper, we will design a path tracking controller for a trailer-like mobile robot which has the following properties.

(1) The stability of the control can be evaluated analytically.
(2) The trailer-like mobile robot tracks the desired path even when it moves backward.
(3) The control varies according to the environment and/or the velocity of the robot without violating the stability of the control.

1. Introduction

Mobile robots are used to perform tasks with minimal operator assistance. In particular, trailer-like mobile robots are useful because they can carry much luggage compared to car-like mobile robots can do. Their path tracking control, however, is difficult especially when they move backward. For example, even if we would like them to follow a straight line while they are moving backward, they will soon end up jack-knifing and go out of control as Fig.1 shows. Though we need this kind of control for putting a robot into a garage, effective controllers for this have not been proposed yet.

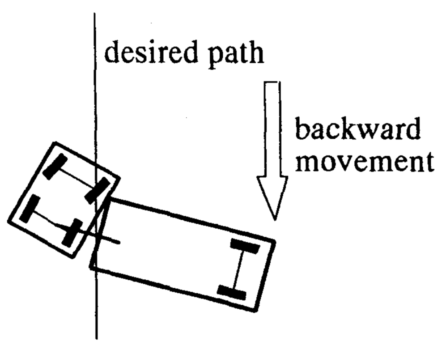


Fig.1 Jack-Knife

Here, we will design a path-tracking controller for a trailer-like mobile robot which has the following properties.

(1) The stability of the control can be evaluated analytically.
(2) The trailer-like mobile robot tracks the desired path even when it moves backward.
(3) The control varies according to the environment and/or the velocity of the robot without violating the stability of the control.

In this paper, we consider the trailer-like mobile robot shown in Fig.2. (X, Y) are Cartesian coordinates which represent the position of the robot. y_2 represents deviation from the desired path (the X-axis). We determine the following symbols.

Symbols in the tractor

- P_1 : Middle point of the tractor's rear wheels. Its Cartesian coordinates are (x_1, y_1).
- O_1 : Center of rotation of the tractor.
- Q : Steering axis ($O_1P_1 \perp P_1Q$).
- L_1 : Wheelbase of the tractor.
- θ_1 : Angle between the center line of the tractor and that of the trailer.
- α : Steering angle.
- z : Distance along the real path of P_1.

Symbols in the trailer

- P_2 : Middle point of the trailer's rear wheels. Its Cartesian coordinates are (x_2, y_2).
- O_2 : Center of rotation of the trailer.
- L_2 : Wheelbase of the trailer. (Distance between P_1 and P_2.)
- θ_2 : Angle between the center line of the trailer and the desired path. (Angle deviation of the trailer.)
- η : Distance along the real path of P_2.

The object of this paper is to design controllers which make the robot follow the X-axis, i.e., $y_2 \to 0$, $\theta_1 \to 0$ and $\theta_2 \to 0$ as it moves forward or backward.

We design controllers using the exact linearization method [1,2] and time scale transformation [3,4]. The controller is designed as follows: firstly, define a new time scale which is chosen to be identical to the distance along

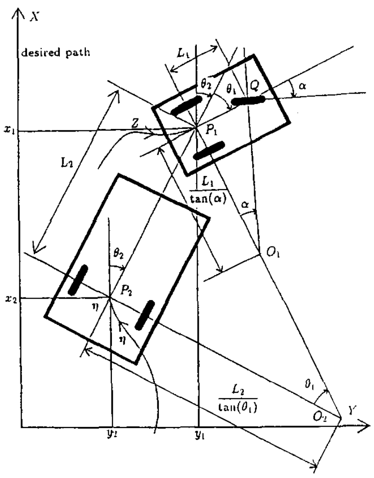


Fig.2 A Trailer-Like Mobile Robot

the desired path, i.e., x_2, and describe the dynamic model of the robot using a state equation with this time scale x_2. Then, exactly linearize this state equation with appropriate state and input transformation. Finally, design a linear controller (servo controller if necessary) for the linearized system. Since the robot's dynamics is exactly transformed to a linear system, we can analytically evaluate the stability and the performance of the trajectory control. Using further time scale transformation and linearization, we can design controllers which make the robot track the X-axis even when it moves backward, and/or which vary according to the velocity of the robot and/or the environment without violating the stability of the control.

We also evaluated this control by experiments. We built a model trailer with sensors to measure the trailer's position and angle deviation. In the controller design, we will assume that the tractor has three wheels in order to ignore slide slips. The tractor used in the experiments, however, had four wheels, i.e., there were slide slips as a disturbance. But the model trailer successfully tracked the desired path as it moved forward or backward.

2. Modeling of the Trailer-Like Mobile Robot

Let us assume that there are no slide slips. Then we can geometrically obtain the following differential equations of the trailer-like mobile robot.

$$\frac{d}{dz}(\theta_1+\theta_2) = \frac{1}{L_1}\tan(\alpha). \tag{1}$$

$$\frac{d(\theta_2)}{d\eta} = \frac{1}{L_2}\tan(\theta_1). \tag{2}$$

$$\frac{dx_2}{d\eta} = \cos(\theta_2). \tag{3}$$

$$\frac{dy_2}{d\eta} = \sin(\theta_2). \tag{4}$$

$$\frac{d\eta}{dz} = \cos(\theta_1). \tag{5}$$

Where, (1) represents the dynamics of the tractor, (2) - (4) represent that of the trailer, and (5) represents the relation between the real path of the tractor and that of the trailer.

Since it will be hard to design a controller if we describe the robot's dynamics in the actual time scale t, we will introduce a new time scale[3,4]. The new time scale is x_2 which is identical to the distance along the desired path. The differential equations(1) - (5) can be summarized in the following state equation with this time scale x_2.

$$\frac{d}{dx_2}\begin{pmatrix} y_2 \\ \theta_1 \\ \theta_2 \end{pmatrix} = \begin{pmatrix} \tan(\theta_2) \\ -\frac{\tan(\theta_1)}{L_2\cos(\theta_2)} \\ \frac{\tan(\theta_1)}{L_2\cos(\theta_2)} \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{L_1\cos(\theta_1)\cos(\theta_2)} \\ 0 \end{pmatrix}\tan(\alpha). \tag{6}$$

3. Exact Linearization of the Dynamics

System (6) can exactly be linearized by defining the state:

$$\phi \stackrel{\text{def}}{=} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{pmatrix} \stackrel{\text{def}}{=} \begin{pmatrix} y_2 \\ \tan(\theta_2) \\ \frac{\tan(\theta_1)}{L_2\cos^3(\theta_2)} \end{pmatrix}. \tag{7}$$

By differentiating states ϕ_1, ϕ_2 and ϕ_3 with respect to x_2, we have

$$\frac{d\phi_1}{dx_2} = \frac{dy_2}{dx_2} = \tan(\theta_2) = \phi_2, \tag{8}$$

$$\begin{aligned} \frac{d\phi_2}{dx_2} &= \frac{d}{dx_2}\tan(\theta_2) = \{\frac{\partial}{\partial\theta_2}\tan(\theta_2)\}\frac{d\theta_2}{dx_2} \\ &= \frac{1}{\cos^2(\theta_2)}\frac{d\theta_2}{dx_2} = \frac{\tan(\theta_1)}{L_2\cos^3(\theta_2)} = \phi_3, \end{aligned} \tag{9}$$

$$\begin{aligned} \frac{d\phi_3}{dx_2} &= \frac{d}{dx_2}\frac{\tan(\theta_1)}{L_2\cos^3(\theta_2)} \\ &= \{\frac{\partial}{\partial\theta_1}\frac{\tan(\theta_1)}{L_2\cos^3(\theta_2)}\}\frac{d\theta_1}{dx_2} + \{\frac{\partial}{\partial\theta_2}\frac{\tan(\theta_1)}{L_2\cos^3(\theta_2)}\}\frac{d\theta_2}{dx_2} \\ &= \frac{3\sin^2(\theta_1)\tan(\theta_2)-\tan(\theta_1)}{L_2^2\cos^2(\theta_1)\cos^4(\theta_2)} \\ &\quad + \frac{1}{L_1L_2\cos^3(\theta_1)\cos^4(\theta_2)}\tan(\alpha). \end{aligned} \tag{10}$$

If we define new input v:

$$\begin{aligned} v &= \frac{3\sin^2(\theta_1)\tan(\theta_2)-\tan(\theta_1)}{L_2^2\cos^2(\theta_1)\cos^4(\theta_2)} \\ &\quad + \frac{1}{L_1L_2\cos^3(\theta_1)\cos^4(\theta_2)}\tan(\alpha), \end{aligned} \tag{11}$$

we can obtain the exactly linearized system

$$\frac{d\phi}{dx_2} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}\phi + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}v. \tag{12}$$

We can readily design any type of linear controller for this linearized system; for example, integrators can be introduced to achieve servo control.

This strategy allows us to analytically evaluate the stability of the trajectory control because the dynamics is expressed by a linear state equation. For example, we can define poles for path tracking control.

If we design the state feedback controller:

$$v = \begin{pmatrix} f_1 & f_2 & f_3 \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{pmatrix}, \tag{13}$$

then the steering angle α is

$$\alpha = \tan^{-1}\left[-\frac{L_1\cos(\theta_1)\{3\sin^2(\theta_1)\tan(\theta_2) - \tan(\theta_1)\}}{L_2} + L_1 L_2 \cos^3(\theta_1)\cos^4(\theta_2) v\right], \tag{14}$$

where,

$$v = f_1 y_2 + f_2 \tan(\theta_2) + f_3 \frac{\tan(\theta_1)}{L_2\cos^3(\theta_2)}. \tag{15}$$

This implies that α can be determined by only using y_2, θ_1 and θ_2. Moreover, if the feedback (13) stabilizes system (12), then $\phi \to 0$ as $x_2 \to \infty$. Thus, $y_2 \to 0$, $\theta_1 \to 0$ and $\theta_2 \to 0$ as x_2 increases. It can be readily shown that, with the stabilizing feedback (13), x_2 monotonically increases with respect to the actual time t, providing $\dot{z} \geq 0$, $-\pi/2 < \theta_1(0) < \pi/2$ and $-\pi/2 < \theta_2(0) < \pi/2$. Therefore, under these conditions, the trailer-like mobile robot tracks the X-axis as it moves forward.

We can also design a servo controller:

$$v = \int (y_2 - y_r)dx_2 + f_1\phi_1 + f_2\phi_2 + f_3\phi_3, \tag{16}$$

for linearized system (12), where y_r is the reference input. Since $dx_2 = \dot{\eta}\cos(\theta_2)\ dt$ from (3), v can numerically be calculated as

$$v = \int (y_2 - y_r)\dot{\eta}\cos(\theta_2)dt + f_1 y_2 + f_2\tan(\theta_2) + f_3\frac{\tan(\theta_1)}{L_2\cos^3(\theta_2)}, \tag{17}$$

and steering angle α should be

$$\alpha = \tan^{-1}\left[-\frac{L_1\cos(\theta_1)\{3\sin^2(\theta_1)\tan(\theta_2) - \tan(\theta_1)\}}{L_2} + L_1 L_2 \cos^3(\theta_1)\cos^4(\theta_2) v\right],$$

where, v is defined as (17). With an argument similar to that in the state feedback case, with this controller, the trailer-like mobile robot tracks the X-axis if $\dot{z} \geq 0$, $-\pi/2 < \theta_1(0) < \pi/2$ and $-\pi/2 < \theta_2(0) < \pi/2$. Furthermore, this controller eliminates steady-state position error, even if there exists constant error in the steering angle α.

4. Controller Design Using Further Time Scale Transformation

In the previous section, we have shown the basic strategy of controller design using exact linearization. This controller, however, can only be used when the trailer-like mobile robot moves forward ($\dot{z} > 0$) and cannot be used when it moves backward ($\dot{z} < 0$). If we consider the case of parking a trailer-like mobile robot in a garage, however, control of the robot moving backward is important.

It is also important to modify the trajectory tracking controller according to the robot's conditions. For example, we may use only a small steering angle while the robot moves fast. This is because centrifugal force increases with the increase of the mobile robot's velocity. On the other hand, we may use a large steering angle if it moves slowly. Thus, we need to design a velocity-dependent controller such that the steering angle will become smaller when it moves fast, and larger when it moves slowly. Furthermore, controllers should be designed to fit the environmental conditions. For example, if the working space of the robot is narrow, then tight tracking control is required.

In order to design the controller which allows us to control the robot moving backward, and/or which depends on the robot's condition and/or the environment, we utilize further time scale transformation.

We define another time scale ξ using the function $s(\lambda)$ as follows:

$$\frac{dx_2}{d\xi} \stackrel{\text{def}}{=} s(\lambda), \tag{18}$$

where λ represents the robot's velocity $\dot{z}$ and/or environmental factors. With use of this time scale transformation, system (12) becomes

$$\frac{d\phi}{d\xi} = s(\lambda)\frac{d\phi}{dx_2} = \begin{pmatrix} s(\lambda)\phi_2 \\ s(\lambda)\phi_3 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ s(\lambda) \end{pmatrix} v. \tag{19}$$

It can be readily shown that this system can exactly be linearized by using coordinate transformation:

$$\psi \stackrel{\text{def}}{=} \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{pmatrix} \stackrel{\text{def}}{=} \begin{pmatrix} \phi_1 \\ s(\lambda)\phi_2 \\ \frac{ds}{d\xi}\phi_2 + \{s(\lambda)\}^2\phi_3 \end{pmatrix}, \tag{20}$$

and input transformation:

$$\mu \stackrel{\text{def}}{=} \frac{d^2 s}{d\xi^2}\phi_2 + 3s(\lambda)\phi_3\frac{ds}{d\xi} + \{s(\lambda)\}^3 v. \tag{21}$$

With these transformations, the system becomes

$$\frac{d\psi}{d\xi} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}\psi + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}\mu. \tag{22}$$

Linear stabilizing controllers can be designed for this system as in the previous section. Stabilizing controllers ensure that $\psi_1 \to 0$, $\psi_2 \to 0$ and $\psi_3 \to 0$ as ξ increases.

According to the choice of the time scaling function $s(\lambda)$, we can obtain various kind of controllers.

4.1. Choice of Time Scaling Function for Backward Movement

Firstly, we set $s(\lambda) \equiv -1$. In this case, the states are

$$\psi_1 = \phi_1 = y_2, \quad (23)$$
$$\psi_2 = -\phi_2 = -\tan(\theta_2), \quad (24)$$
$$\psi_3 = \phi_3 = \frac{\tan(\theta_1)}{L_2\cos^3(\theta_2)}. \quad (25)$$

Thus, $\psi_1 \to 0$, $\psi_2 \to 0$ and $\psi_3 \to 0$ imply $y_2 \to 0$, $\theta_2 \to 0$ and $\theta_1 \to 0$, respectively. From the definition of the time scale transformation (18) and $s(\lambda) \equiv -1$, ξ is a strictly monotone-decreasing function with respect to x_2. This means the increase of ξ corresponds to the decrease of x_2. A stabilizing controller for system (22) ensures $y_2 \to 0$, $\theta_1 \to 0$ and $\theta_2 \to 0$ as ξ increases or x_2 decreases. Since x_2 decreases as the robot moves backward ($\dot{z} < 0$) providing $-\pi/2 < \theta_1(0) < \pi/2$ and $-\pi/2 < \theta_2(0) < \pi/2$, this controller ensures that the trailer-like mobile robot tracks the X-axis when it moves backward. This controller can be used for parking the robot in a garage.

4.2. Choice of Time Scaling Function for Velocity-Dependent Controller

If we define $s(\lambda)$ as a function of the velocity of the robot, then the controller will depend on the robot's velocity.

In order to investigate how the function effects the performance of the control, we will consider the following LQ optimal control problem. Consider the performance index:

$$J = \int \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{pmatrix}^T \begin{pmatrix} q_1 & 0 & 0 \\ 0 & q_2 & 0 \\ 0 & 0 & q_3 \end{pmatrix} \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{pmatrix} + r\mu^2 d\xi. \quad (26)$$

It is well known that the optimizing controller can be obtained in the form of state feedback:

$$\mu = f_1\psi_1 + f_2\psi_2 + f_3\psi_3. \quad (27)$$

From (7) (11) (20) (21), the steering angle α is determined as

$$\begin{aligned} v &= \frac{1}{\{s(\lambda)\}^3}\left\{\mu - \frac{d^2 s}{d\xi^2}\phi_2 - 3s(\lambda)\phi_3\frac{ds}{d\xi}\right\} \\ &= \frac{1}{\{s(\lambda)\}^3}\left[f_1 y_2 + \left\{s(\lambda)f_2 + \frac{ds}{d\xi}f_3 - \frac{d^2 s}{d\xi^2}\right\}\tan(\theta_2)\right. \\ &\quad \left. + \left(\{s(\lambda)\}^2 f_3 - 3s(\lambda)\frac{ds}{d\xi}\right)\frac{\tan(\theta_1)}{L_2\cos^3(\theta_2)}\right], \end{aligned} \quad (28)$$

$$\alpha = \tan^{-1}\left[-\frac{L_1\cos(\theta_1)\{3\sin^2(\theta_1)\tan(\theta_2) - \tan(\theta_1)\}}{L_2} + L_1 L_2\cos^3(\theta_1)\cos^4(\theta_2)v\right].$$

If $s(\lambda)$ is a constant value, then

$$v = \frac{1}{s^3}f_1 y_2 + \frac{1}{s^2}f_2\tan(\theta_2) + \frac{1}{s}f_3\frac{\tan(\theta_1)}{L_2\cos^3(\theta_2)}, \quad (29)$$

$$\alpha = \tan^{-1}\left[-\frac{L_1\cos(\theta_1)\{3\sin^2(\theta_1)\tan(\theta_2) - \tan(\theta_1)\}}{L_2} + L_1 L_2\cos^3(\theta_1)\cos^4(\theta_2)v\right].$$

It can be readily shown that this feedback minimizes the following performance index:

$$J = \int \begin{pmatrix} y_2 \\ \tan(\theta_2) \\ \frac{\tan(\theta_1)}{L_2\cos^3(\theta_2)} \end{pmatrix}^T \begin{pmatrix} q_1 & 0 & 0 \\ 0 & s^2 q_2 & 0 \\ 0 & 0 & s^4 q_3 \end{pmatrix} \begin{pmatrix} y_2 \\ \tan(\theta_2) \\ \frac{\tan(\theta_1)}{L_2\cos^3(\theta_2)} \end{pmatrix} + s^6 r v^2 dx_2. \quad (30)$$

This performance index shows that the weights for $\tan\theta_2$, $\frac{\tan(\theta_1)}{L_2\cos^3(\theta_2)}$ and v increase as s increases. Since v is defined as (11), this implies that the weights for θ_1, θ_2 and the steering angle α increase as s increases. In other words, if we choose a large s, then the controller keeps θ_1, θ_2 and the steering angle α small. Thus if we choose the function $s(\lambda)$ so that it increases with an increase in the robot's velocity, we can achieve the desired velocity-dependent control: the steering angle α will be small when the robot moves fast.

5. Simulation

Fig.3-1 shows the case where the trailer-like mobile robot moves forward. We designed a regulator and a servo controller. In both cases, the robot tracks the desired path.

Fig.3-2 is the case of backward movement. Since we use the proposed controller, the robot tracks the desired path even when it moves backward.

Fig.3-3, Fig.3-4 are the case that there exists constant error in the steering angle α. If we design a regulator,the robot cannot track the desired path and there exists constant error in its position. If we design a servo controller, however, it tracks the desired path without error.

In Fig.3-5, we use the velocity-dependent controller proposed in section 4.2, where $s(\dot{z}) = \dot{z}$ ($\dot{z}$ is the velocity of the robot). As it is shown in this figure, the steering angle α will be small when the robot moves fast, where $s(\dot{z}) = \dot{z}$ is large. Therefore, while the robot moves fast, it takes longer distance to track the desired path than the case where it moves slowly.

Fig.3-6 shows the backward movement with another initial condition, where $\theta_2(0) \neq 0$. This can be applied to the control of putting this kind of robot into a garage.

6. Experiments

We evaluated the proposed controller by experiments using a model trailer. (We will show these experiments in the film in the presentation.) Fig.4 illustrates the path tracking control for the trailer-like mobile robot. Path tracking control for backward movement corresponds to the control of putting a robot into a garage if we put a garage at the origin O. Further symbols in this figure are as follows.

O : The axis of rotation of the sensor 1 at the origin.
R : The axis of rotation of the sensor 2 on the trailer.
β : Angle between the center line of the trailer and the optical axis of the sensor 2.
γ : Angle between the X-axis and the optical axis of the sensor 1.
l : Distance between O and R.

We have two sensors in Fig.4, and each of them faces each other to measure β, γ and l. If we do not control these sensors, however, they go out of this situation when the robot moves. Therefore, we should control two sensors to keep them facing each other.

Fig.5 shows the configuration of this control. Each sensor has PSD(Position Sensitive Detector), lens and infrared LEDs(Light-Emitting Diodes). PSD receives infrared light from the opposite sensor through lens, and it measures the angle between its own optical axis and the line defined by the infrared LEDs on the other sensor and focal point. Then we designed the state feedback controller to make this angle zero. With this controller, each sensor faces each other in the real-time.

Since each sensor always faces each other, we can measure β and γ using rotary encoders. And we measure l, i.e., the distance between O and R, using the ultrasonic distance sensor whose receiver is on the sensor 1 and transmitter is on the sensor 2. Using these β, γ and l, we can obtain θ_2 and y_2 which are used to determine the steering angle α. We calculated y_2 using the data of l and γ. And obviously $\theta_2 = \beta + \gamma$. θ_1 which is also used to determine the steering angle α was measured by the rotary encoder which was fixed to P_1.

We used the model trailer whose tractor had four wheels, though we assumed it had only three wheels to ignore slide slips when we designed the controller. The experimental results show that the robot successfully tracks the desired path as in the simulation even there exist slide slips as a disturbance. This is because the proposed controller is a feedback controller and is robust for such disturbance. (We will show these experiments in the film in the presentation.)

7. Conclusion

We proposed a method to design a path tracking controller for a trailer-like mobile robot. This controller is especially useful when a robot moves backward. We can use this controller when we put this kind of robot into a garage.

References

[1] R.Su: On the Linear Equivalents of Nonlinear Systems, Systems and Control Letters, vol.2, no.1, 48/52 (1982).

[2] B.Jakubczyk and W.Respondek: On Linearization of Control Systems, Bull. Acad. Polonaise Sci. Ser. Sci. Math., vol.28, 517/522 (1980).

[3] M.Sampei and K.Furuta: On Time Scaling for Nonlinear Systems: Application to Linearization, IEEE Trans. on Automatic Control, vol.AC-31, no.5, 459/462 (1986).

[4] M.Sampei and K.Furuta: Robot Control in the Neighborhood of Singular Points, IEEE Journal of Robotics and Automation, vol.4, no.3 303/309 (1988).

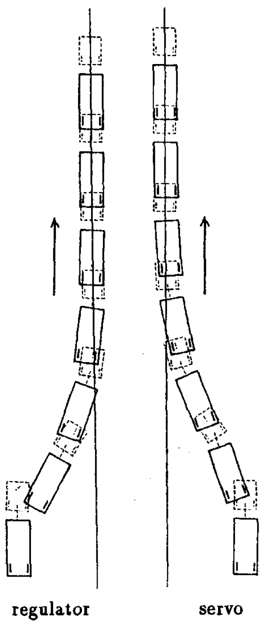


Fig.3-1 Forward Movement of Trailer

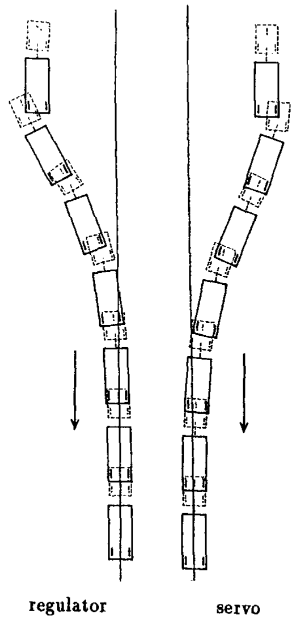


Fig.3-2 Backward Movement of Trailer

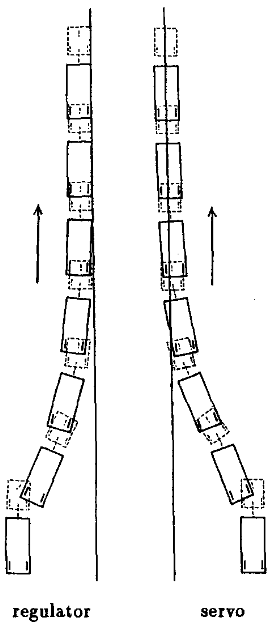
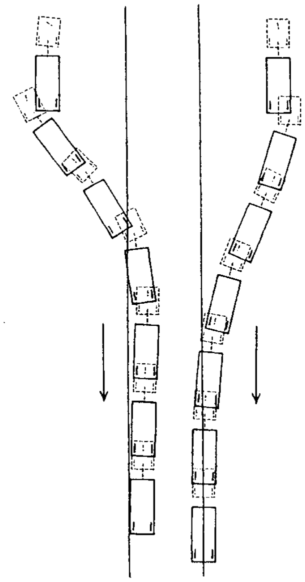
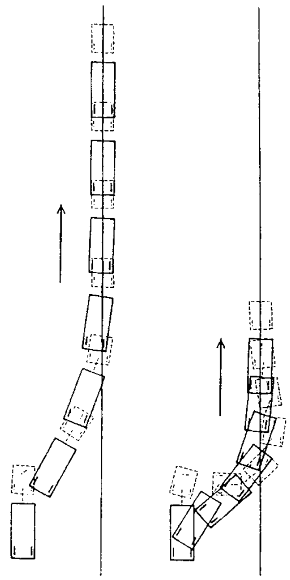


Fig.3-3 Forward Movement of Trailer with Constant Error in Steering Angle



regulator servo

Fig.3-4 Backward Movement of Trailer with Constant Error in Steering Angle



$\dot{z} = 2$ $\dot{z} = 1$

Fig.3-5 Forward Movement of Trailer with Velocity-Dependent Controller ($s(\dot{z}) = \dot{z}$)

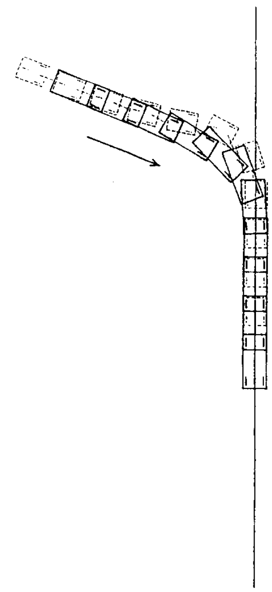


Fig.3-6 Backward Movement of Trailer with $\theta_2(0) \neq 0$ (Parking Control of Trailer)

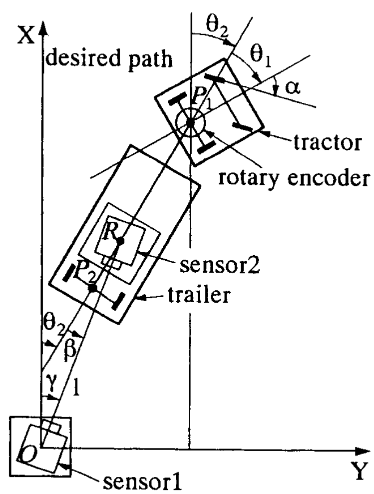


Fig.4 Path Tracking Control for a Trailer-Like Mobile Robot

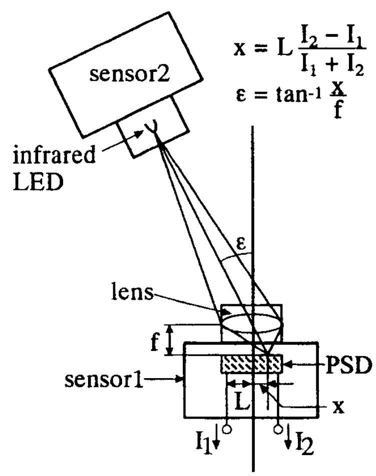


Fig.5 Control for Sensors