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 \((z_1, y_1)\).
- \(O_1\) : Center of rotation of the tractor.
- \(Q\) : Steering axis \((O_1P_1 \perp P_1Q)\).
- \(L_1\) : Wheelbase of the tractor.
- \(\theta_1\) : Angle between the center line of the tractor and that of the trailer.
- \(\alpha\) : 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 \((z_2, y_2)\).
- \(O_2\) : Center of rotation of the trailer.
- \(L_2\) : Wheelbase of the trailer.
  - (Distance between \(P_1\) and \(P_2\)).
- \(\theta_2\) : Angle between the center line of the trailer and the desired path.
  - (Angle deviation of the trailer.)
- \(\eta\) : 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., \(y_2 \rightarrow 0\), \(\theta_1 \rightarrow 0\) and \(\theta_2 \rightarrow 0\) 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...
the desired path, i.e., \( z_2 \), and describe the dynamic model of the robot using a state equation with this time scale \( z_2 \). 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. Model 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.

\[
\begin{align*}
\frac{d \theta_1}{dt} & = \frac{1}{L_1} \tan(\alpha). \\
\frac{d \theta_2}{dt} & = \frac{1}{L_2} \tan(\theta_1). \\
\frac{d \theta_2}{dz} & = \cos(\theta_2).
\end{align*}
\]  

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 \( z_2 \). The new time scale is \( z_2 \) 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 \( z_2 \).

\[
\frac{d}{dz_2} \begin{pmatrix}
\theta_1 \\
\theta_2 \\
\end{pmatrix}
= \begin{pmatrix}
\frac{\tan(\theta_2)}{L_2 \cos^2(\theta_2)} \\
\frac{\tan(\theta_1)}{L_2 \cos^2(\theta_2)} \\
0 \\
\end{pmatrix}
+ \begin{pmatrix}
0 \\
\tan(\alpha) \\
0 \\
\end{pmatrix} \cos(\theta_2).
\]

3. Exact Linearization of the Dynamics

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

\[
\phi \triangleq \begin{pmatrix}
\phi_1 \\
\phi_2 \\
\phi_3
\end{pmatrix}
= \begin{pmatrix}
\theta_2 \\
\tan(\theta_1) \\
\frac{\tan(\theta_2)}{L_2 \cos^2(\theta_2)}
\end{pmatrix}.
\]

By differentiating states \( \phi_1, \phi_2 \) and \( \phi_3 \) with respect to \( z_2 \), we have

\[
\begin{align*}
\frac{d \phi_1}{dz_2} & = \frac{d \phi_2}{dz_2} = \tan(\theta_2) = \phi_2, \\
\frac{d \phi_2}{dz_2} & = \frac{d}{dz_2} \tan(\theta_1) = \frac{\partial}{\partial \theta_1} \tan(\theta_1) \frac{d \theta_1}{dz_2} = \frac{1}{L_2 \cos^2(\theta_2)} \frac{d \theta_1}{dz_2} = \phi_3, \\
\frac{d \phi_3}{dz_2} & = \frac{d}{dz_2} \frac{\tan(\theta_2)}{L_2 \cos^2(\theta_2)} = \frac{3 \sin^2(\theta_2) \tan(\theta_2) - \tan(\theta_1)}{L_2^2 \cos^2(\theta_1) \cos^2(\theta_2)} + \frac{1}{L_2 \cos^2(\theta_1) \cos^2(\theta_2)} \tan(\alpha).
\end{align*}
\]

If we define new input \( v \):

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

we can obtain the exactly linearized system

\[
\frac{d \phi}{dz_2} = \begin{pmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0
\end{pmatrix} \phi + \begin{pmatrix}
0 \\
0 \\
1
\end{pmatrix} v.
\]
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:

\[
\begin{pmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\dot{x}_3
\end{pmatrix} = \begin{pmatrix}
f_1 & f_2 & f_3 \\
\phi_1 & \phi_2 & \phi_3
\end{pmatrix} \begin{pmatrix}
x_1 \\
x_2 \\
x_3
\end{pmatrix}
\]

then the steering angle \( \alpha \) is

\[
\alpha = \tan^{-1} \left( \frac{-L_1 \cos(\theta_1) (3 \sin^2(\theta_2) \tan(\theta_2) - \tan(\theta_3))}{L_2} + \frac{L_1 L_2 \cos^3(\theta_1) \cos^4(\theta_2) v}{L_2} \right),
\]

where,

\[
v = f_1 y_2 + f_2 \tan(\theta_2) + f_3 \tan(\theta_3)
\]

This implies that \( \alpha \) can be determined by only using \( y_2, \theta_1, \) and \( \theta_2 \). Moreover, if the feedback (13) stabilizes system (12), then \( \phi \rightarrow 0 \) as \( x_2 \rightarrow \infty \). Thus, \( y_2 \rightarrow 0, \theta_1 \rightarrow 0 \) and \( \theta_2 \rightarrow 0 \) 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:

\[
v = \int (y_2 - y_1) dx_2 + f_1 \phi_1 + f_2 \phi_2 + f_3 \phi_3,
\]

for linearized system (12), where \( y_1 \) is the reference input. Since \( dx_2 = \dot{z} \cos(\theta_2) dt \) from (3), \( v \) can numerically be calculated as

\[
v = \int (y_2 - y_1) \dot{z} \cos(\theta_2) dt + f_1 y_2 + f_2 \tan(\theta_2) + f_3 \tan(\theta_3),
\]

and steering angle \( \alpha \) should be

\[
\alpha = \tan^{-1} \left( \frac{-L_1 \cos(\theta_1) (3 \sin^2(\theta_2) \tan(\theta_2) - \tan(\theta_3))}{L_2} + \frac{L_1 L_2 \cos^3(\theta_1) \cos^4(\theta_2) v}{L_2} \right),
\]

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 \( \alpha \).

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 \( \xi \) using the function \( s(\lambda) \) as follows:

\[
\frac{dz_2}{d\xi} = s(\lambda),
\]

where \( \lambda \) represents the robot's velocity and/or environmental factors. With use of this time scale transformation, system (12) becomes

\[
\frac{d\phi}{d\xi} = s(\lambda) \frac{d\phi}{d\xi} + \left( \begin{array}{c} 0 \\ s(\lambda) \phi_2 \\ 0 \end{array} \right) v.
\]

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

\[
\psi \triangleq \left( \begin{array}{c} \psi_1 \\ \psi_2 \\ \psi_3 \end{array} \right) \triangleq \left( \begin{array}{c} \phi_1 \\ s(\lambda) \phi_2 \\ s(\lambda)^3 \phi_3 \end{array} \right),
\]

and input transformation:

\[
\mu \triangleq \frac{d^2 s}{d\xi^2} \phi_2 + 3s(\lambda)\phi_3 \frac{ds}{d\xi} + s(\lambda)^3 v.
\]

With these transformations, the system becomes

\[
\frac{d\psi}{d\xi} = \left( \begin{array}{ccc} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{array} \right) \psi + \left( \begin{array}{c} 0 \\ 0 \\ 1 \end{array} \right) \mu.
\]

Linear stabilizing controllers can be designed for this system as in the previous section. Stabilizing controllers ensure that \( \psi_1 \rightarrow 0, \psi_2 \rightarrow 0 \) and \( \psi_3 \rightarrow 0 \) as \( \xi \) increases.
According to the choice of the time scaling function \( s(\lambda) \), we can obtain various kinds of controllers.

### 4.1. Choice of Time Scaling Function for Backward Movement

Firstly, we set \( s(\lambda) = -1 \). In this case, the states are
\[
\begin{align*}
\psi_1 &= \phi_1 = y_1, \\
\psi_2 &= -\phi_2 = -\tan(\theta_2), \\
\psi_3 &= \phi_3 = \frac{\tan(\theta_1)}{L_2 \cos^2(\theta_2)}.
\end{align*}
\]

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 \( s(\lambda) = -1 \), \( \xi \) is a strictly monotone-decreasing function with respect to \( x_2 \). This means the increase of \( \xi \) corresponds to the decrease of \( x_2 \).

A stabilizing controller for system (22) ensures \( y_2 \to 0, \theta_2 \to 0 \) and \( \theta_1 \to 0 \) as \( x_2 \) decreases. Since \( x_2 \) decreases as the robot moves backward \( (\dot{z} < 0) \) providing \( -\pi/2 < \theta_2(0) < \pi/2 \) and \( -\pi/2 < \theta_3(0) < \pi/2 \), 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:
\[
J = \int \left( \psi_1^T Q_1 \psi_1 + \psi_2^T Q_2 \psi_2 + \psi_3^T Q_3 \psi_3 \right) + r \mu^2 d\xi. \tag{26}
\]

It is well known that the optimizing controller can be obtained in the form of state feedback:
\[
\mu = f_1 \psi_1 + f_2 \psi_2 + f_3 \psi_3. \tag{27}
\]

From (7) (11) (20) (21), the steering angle \( \alpha \) is determined as
\[
\alpha = \tan^{-1} \left[ \frac{-L_1 \cos(\theta_1) \{3 \sin^2(\theta_2) \tan(\theta_2) - \tan(\theta_1) \}}{L_2} \right. \\
+ \left. L_1 L_2 \cos^3(\theta_1) \cos^2(\theta_2) \nu \right].
\]

It can be readily shown that this feedback minimizes the following performance index:
\[
J = \int \left( \begin{array}{c}
y_2 \\
\tan(\theta_1) \\
\tan(\theta_2) \\
\end{array} \right)^T \left( \begin{array}{ccc}
Q_1 & 0 & 0 \\
0 & Q_2 & 0 \\
0 & 0 & Q_3 \\
\end{array} \right) \left( \begin{array}{c}
y_2 \\
\tan(\theta_1) \\
\tan(\theta_2) \\
\end{array} \right) + s^2 r \nu^2 d\xi. \tag{30}
\]

This performance index shows that the weights for \( \tan(\theta_2), \tan(\theta_1), \) and \( \nu \) increase as \( s \) increases. Since \( \nu \) is defined as (11), this implies that the weights for \( \theta_1, \theta_2 \) and the steering angle \( \alpha \) increase as \( s \) increases. In other words, if we choose a large \( s \), then the controller keeps \( \theta_1, \theta_2 \) and the steering angle \( \alpha \) 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 \( \alpha \) 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 \( \alpha \). 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 \( s(\dot{z}) = \dot{z} \). As it is shown in this figure, the steering angle \( \alpha \) will be small when the robot moves fast, where \( \dot{z} \) 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.
We have two sensors in Fig.4, and each of them faces each other to measure \( \theta \), \( \gamma \) 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 \( \beta \) and \( \gamma \) 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 \( \beta \), \( \gamma \), and \( l \), we can obtain \( \theta_2 \) and \( \gamma_2 \) which are used to determine the steering angle \( \alpha \). We calculated \( \gamma_2 \) using the data of \( l \) and \( \gamma \). And obviously \( \theta_2 = \beta + \gamma \). \( \theta_1 \) which is also used to determine the steering angle \( \alpha \) was measured by the rotary encoder which was fixed to \( R \).

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 especially useful when a robot moves backward. We can use this controller when we put this kind of robot into a garage.

References

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

Fig. 3-5 Forward Movement of Trailer with Velocity-Dependent Controller ($s(i) = i$)

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

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

Fig. 5 Control for Sensors

$$\begin{align*}
X & \text{ desired path} \\
\theta_2 & \text{ tractor} \\
\theta_1 & \text{ rotary encoder} \\
sensor2 & \text{ sensor} \\
1 & \text{ trailer} \\
sensor1 & \text{ sensor} \\
Y & \text{ Y}
\end{align*}$$

$$x = L \frac{L_2 - L_1}{L_1 + L_2}$$

$$\varepsilon = \tan^{-1} \frac{X}{f}$$

- 198 -
学霸图书馆

www.xuebalib.com

本文献由“学霸图书馆-文献云下载”收集自网络，仅供学习交流使用。

学霸图书馆（www.xuebalib.com）是一个“整合众多图书馆数据库资源，提供一站式文献检索和下载服务”的24小时在线不限IP图书馆。

图书馆致力于便利、促进学习与科研，提供最强文献下载服务。

图书馆导航：
图书馆首页 文献云下载 图书馆入口 外文数据库大全 疑难文献辅助工具