A Just-In-Time Modeling Approach to Obstacle Avoidance Control of a Car

Tatsuya Kai[1] and Shunsuke Miyashita[1]

Abstract**—**This study is addressed to obstacle avoidance control of a car based on just-in-time modeling. Just-in-time modeling is a new kind of data-driven control techniques and is used in various real systems. The main characteristics of the proposed method is that a gain of the control input to avoid an encountered obstacle is computed from a database which includes a lot of driving data in different situations via just-in-time modeling. Especially, the noteworthy advantage of the new method is small computation time, and hence real-time control can be achieved. Numerical simulations show that the car can avoid various obstacles by the proposed method.

Keywords**—** just-in-time modeling, obstacle avoidance control, car, automatic driving.

I. Introduction

Recently, a lot of work on automatic operation of cars have been actively done by various companies and research groups [1-3]. Automatic operation technique includes automatic obstacle avoidance, automatic braking, automatic parking, automatic driving, automatic tracking, and so on. On the other hand, the word "big data" is spreading rapidly and it is utilized in various research fields [4-6]. It is expected that the big data technique can be applied to automatic operation problems of cars.

This research aims at obstacle avoidance control of a car based on just-in-time modeling. In the just-in-time modeling technique, we regard a database which include a lot of input and output data for a system as a mathematical model of the system, and we compute a control input by using some extracted data from the database [7-10]. That is to say, just-in-time modeling is one of the applications of bigdata. First, the problem formulation on the obstacle avoidance control problem is stated in Section 2. Next, a summary of just-in-time modeling is presented and a new method for obstacle avoidance control of a car is developed in Section 3. Then, Section 4 shows some numerical simulations in order to check the effectiveness of the new method.

II. Problem Formulation

In this section, the problem formulation will be presented. In this study, as a mathematical model of a car, we consider the following nonholonomic car model on the 2-dimensional plane:

$$\begin{cases} \dot{x} = \cos\theta \cdot u_1, \\ \dot{y} = \sin\theta \cdot u_1, \\ \dot{\theta} = u_2, \end{cases} \tag{1}$$

where (x, y) is the center point of the car, θ is the heading angle of the car, and u_1, u_2 are the control inputs.

Next, the problem setting of an obstacle is explained. We assume that the shape of the obstacle is circular and the center of the obstacle is located on the x–axis. We denote the center of the obstacle by $(X_o, 0)$ and the radius of the obstacle by R_o as shown in Fig. 1. Now, it is also assumed that the control inputs u_1, u_2 are represented in the form:

$$\begin{cases} u_1 = V_c, \\ u_2 = A\cos\dfrac{2\pi t}{T_c} \quad (0 \le t \le T_c), \end{cases} \tag{2}$$

where V_c is the velocity, A is the gain of u_2, and T_c is the avoidance time. If the control inputs (2) are applied to the car (1), the car moves as depicted in Fig. 1.

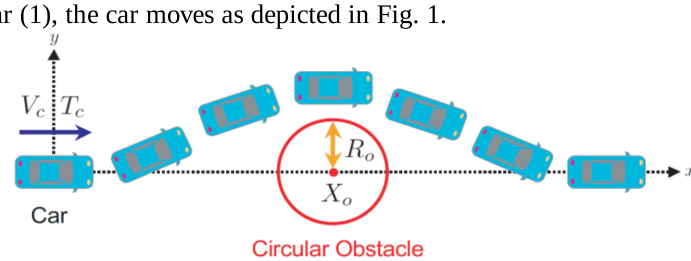


Fig. 1: Obstacle avoidance control of a car.

In this study, we consider the following problem on obstacle avoidance control of the car.

Problem [Obstacle Avoidance Problem]

For a given data: a velocity of the car V_c, an avoidance time T_c, a circular obstacle X_o, R_o, find a gain A of the control input u_2 which can avoid collision with the circular obstacle.

[1]Tokyo University of Science

In the next section, we will consider a new method based on just-in-time modeling as a solution of the above problem.

III. Just-In-Time Modelling Approach To Obstacle Avoidance Control

This section develops a new method of obstacle avoidance control of a car based on the just-in-time modeling approach. First, a summary on just-in-time modeling is presented. In general, just-in-time modeling is carried out in accordance with the following procedure.

(1) Construct a database that contains input-output data of the system.
(2) In order to compute an output for a query data, extract some neighborhood data of the query data from the database.
(3) Derive a local linear model at the query data from the obtained neighborhood data.
(4) Using the local linear model, compute the output for the query data.

Fig. 2 shows an illustration of just-in-time modeling. The advantages of just-in-time modeling is as follows; we do not derive a mathematical model of the system for control, it is available for not only linear systems but also nonlinear ones, computation of an output for a query data needs low calculation amount, However, we have some options such as definition of neighborhood, the total number of data in the database, the number of neighborhood data, derivation of local linear model, and so on.

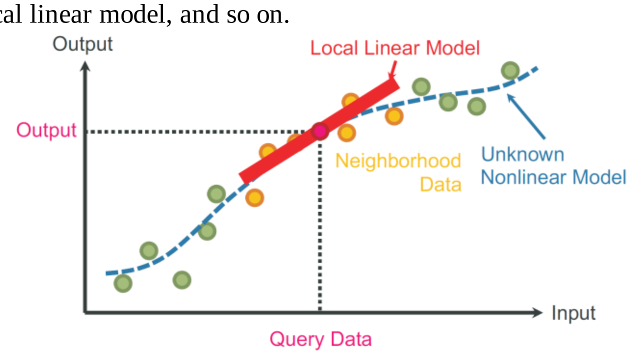


Fig. 2: An illustration of just-in-time modeling.

Second, we consider how to construct a database on obstacle avoidance of the car. At first, we have to decide the range of database. For the velocity of the car V_c, we denote the minimum and maximum values by $V_c^{\min}$ and $V_c^{\max}$, respectively. In addition, we also denote the interval of data for V_c by h_{V_c}. Then, the number of V_c can be calculated by

$$N_{V_c} = \frac{V_c^{\max} - V_c^{\min}}{h_{V_c}} + 1. \tag{3}$$

We have to note that $V_c^{\min}, V_c^{\max}, h_{V_c}$ have to be determined so that N_{V_c} is an integer. In a similar way, we can define the number of T_c, X_o, R_o as

$$N_{T_c} = \frac{T_c^{\max} - T_c^{\min}}{h_{T_c}} + 1,\ N_{X_o} = \frac{X_o^{\max} - X_o^{\min}}{h_{X_o}} + 1,\ N_{R_o} = \frac{R_o^{\max} - R_o^{\min}}{h_{R_o}} + 1. \tag{4}$$

Now, a computation method of gains which can avoid collision with obstacles is derived. For the range of the gain A in the control input u_2. We denote the minimum and maximum values by $A^{\min}$ and $A^{\max}$, respectively. For a data (V_c, T_c, X_o, R_o), we perform a numerical simulation of the car with the gain A by using the car model (1). It is noted that we change the value of A from $A^{\min}$ to $A^{\max}$ with the interval h_A in ascending order. For the simplicity, it is assumed that the shape of the car is approximated by a circle with a radius R_c. Then, the condition such that the car does not collide with the obstacle can be represented by

$$\sqrt{(x(t) - X_o)^2 + y(t)^2} > R_c + R_o + R_{off}, \tag{5}$$

where R_{off} is an offset distance between the car and the obstacle for safety. If the inequality holds while the simulation, obstacle avoidance control is achieved and save the gain A in the database. According to circumstances, the gain which avoids collision cannot be obtained because of the setting of $A^{\max}$, that is to say, the car may not avoid an obstacle in spite of the maximum value of the gain $A^{\max}$. Hence, the number of data in the database N satisfies

$$N = N_{V_c} N_{T_c} N_{X_o} N_{R_o} - N_{fail}, \tag{6}$$

where N_{fail} is the number of data for which obstacle avoidance control is not achieved. The number of data in the database have greater influence on the control performance and the computation amount in just-in-time modeling.

Then, a computation method of a gain for a query data is shown. First, we have to extract some neighborhood data of the query data. We denote the number of data in the database by N and the index of the data by i $(i = 1, \cdots, N)$. In the simplest way, we use Euclidian distance and define the distance between a data $(V_c^i, T_c^i, X_o^i, R_o^i)$ and the query data $(V_c^*, T_c^*, X_c^*, R_c^*)$ as

$$d^i = \sqrt{(V_c^i - V_c^*)^2 + (T_c^i - T_c^*)^2 + (X_o^i - X_o^*)^2 + (R_o^i - R_o^*)^2}. \tag{7}$$

The distance d^i is also saved in the database as $(V_c^i, T_c^i, X_o^i, R_o^i, A^i, d^i)$ $(i = 1, \cdots, N)$. Next, we sort all the data in ascending order with respect to the distance d^i:

$(\tilde{V}_c^i, \tilde{T}_c^i, \tilde{X}_o^i, \tilde{R}_o^i, \tilde{A}^i, \tilde{d}^i)\ (i = 1, \cdots, N)$, and extract $K\ (K < N)$ data from the sorted database (the tilde over variables means sorted data). Finally, by using the K extracted data $(\tilde{V}_c^i, \tilde{T}_c^i, \tilde{X}_o^i, \tilde{R}_o^i, \tilde{A}^i, \tilde{d}^i)\ (i = 1, \cdots, K)$, we calculate the gain A^* of the query data $(V_c^*, T_c^*, X_c^*, R_c^*)$ as weighted mean:

$$A^* = \frac{\sum_{i=1}^{K} \frac{\tilde{A}^i}{\tilde{d}^i}}{\sum_{i=1}^{K} \frac{1}{\tilde{d}^i}} \tag{8}$$

where the inverse numbers of the distances $1/\tilde{d}^i$ are utilized as weights.

The procedure of obstacle avoidance control of the car based on just-in-time modeling is summarized as the next algorithm.

Algorithm [Obstacle Avoidance Control via Just-In-Time Modeling]

Step 1 For various data (V_c, T_c, X_o, R_o), compute a gain A which can avoid an obstacle. Then, construct the database which contains the inputs $(V_c^i, T_c^i, X_o^i, R_o^i)\ (i = 1, \cdots, N)$ and the output $A^i\ (i = 1, \cdots, N)$.

Step 2 For a query data $(V_c^*, T_c^*, X_c^*, R_c^*)$, calculate distances $d^i\ (i = 1, \cdots, N)$ for all the data $(V_c^i, T_c^i, X_o^i, R_o^i)\ (i = 1, \cdots, N)$ in the database with (7), add the distances to the database.

Step 3 Sort all the data in the database in ascending order with respect to the distance: $(\tilde{V}_c^i, \tilde{T}_c^i, \tilde{X}_o^i, \tilde{R}_o^i, \tilde{A}^i, \tilde{d}^i)\ (i = 1, \cdots, N)$ and extract $K\ (K < N)$ data from the sorted database $(\tilde{V}_c^i, \tilde{T}_c^i, \tilde{X}_o^i, \tilde{R}_o^i, \tilde{A}^i, \tilde{d}^i)\ (i = 1, \cdots, K)$.

Step 4 By using the gains and the distances in the extract K data $(\tilde{A}^i, \tilde{d}^i)\ (i = 1, \cdots, K)$, compute the gain A^* from (8).

Step 5 By using the control inputs (2) with the computed gain A^*, control the car.

IV. Numerical Simulations

This section shows some numerical simulations in order to confirm the effectiveness of the proposed method. We first construct a database on obstacle avoidance of the car for just-in-time modeling. For construction of a database, we set the ranges of input data as Table I.

TABLE I
PARAMETER SETTING FOR CONSTRUCTION OF DATABASE

Parameters	Values	Parameters	Values
$V_c^{\min}$	10 m/s	$X_o^{\min}$	40 m
$V_c^{\max}$	15 m/s	$X_o^{\max}$	60 m
h_{V_c}	1 m/s	h_{X_o}	1 m
$T_c^{\min}$	10 s	$R_o^{\min}$	1 m
$T_c^{\max}$	15 s	$R_o^{\max}$	5 m
h_{T_c}	1 s	h_{R_o}	1 m

The parameter of the car is set as $R_c = 2m,\ R_{off} = 0.5m$ and the range of the gain A is also set as $A^{\min} = 0.02,\ A^{\max} = 0.2,\ h_A = 0.02$. For the number of data in the database is $N = 3780$, which is equal to $N = N_{V_c} N_{T_c} N_{X_o} N_{R_o} = 6 \cdot 6 \cdot 21 \cdot 5 = 3780$, hence $N_{fail} = 0$ holds. By the method explained in Section 3, a database on obstacle avoidance control is obtained. A part of the database is shown in Table II.

TABLE II
DATABASE OBTAINED BY PROPOSED METHOD

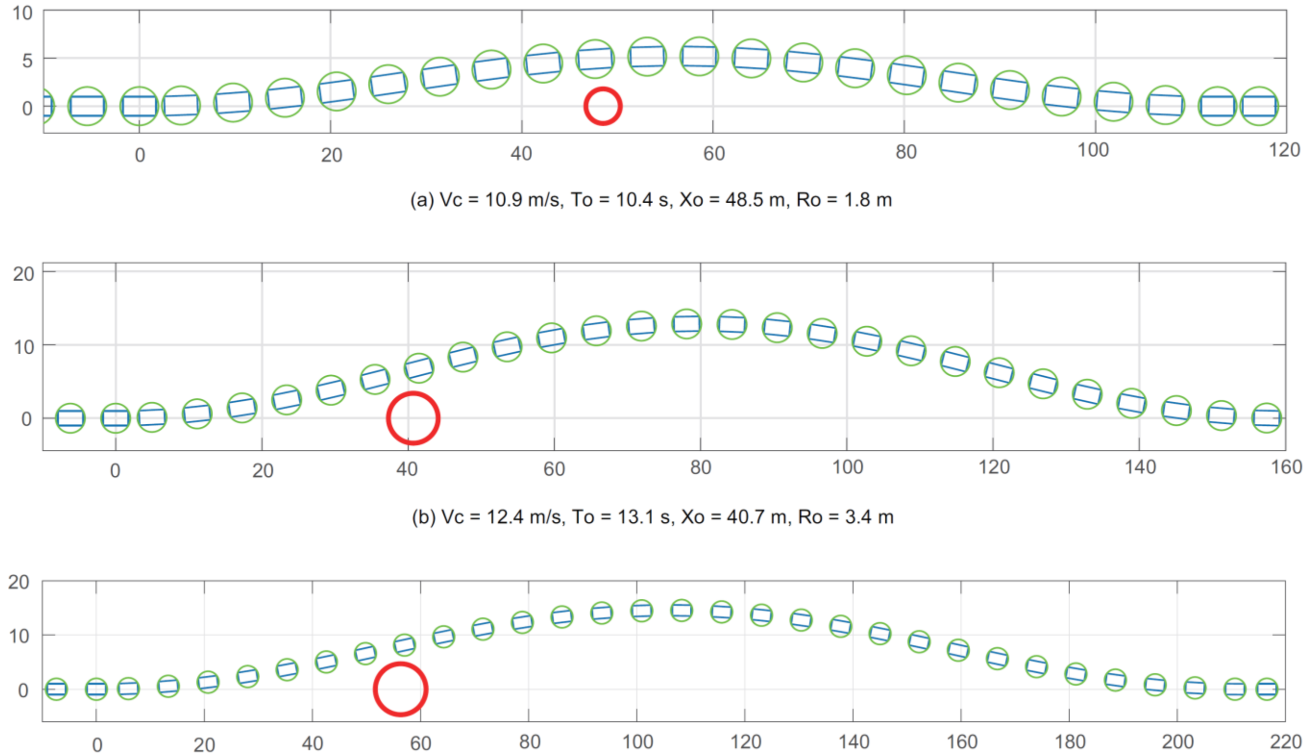
Data No.	V_c	T_c	X_o	R_o	A
1	10	10	40	1	0.08
2	10	10	40	2	0.1
3	10	10	40	3	0.14
4	10	10	40	4	0.16
5	10	10	40	5	0.18
⋮	⋮	⋮	⋮	⋮	⋮
3779	15	15	60	4	0.08
3780	15	15	60	5	0.08

By the database and the proposed algorithm, some numerical simulations are carried out and the results are illustrated in Fig. 3. Fig. 3 shows the trajectories of the car for three types of situations. From this results, we can confirm that the car can avoid the obstacles for various situations. In addition, it is also confirmed that it requires small computation time. Consequently, the results indicate the effectiveness of the proposed method.

V. Conclusions

In this study, a new obstacle avoidance control method of a car has been developed via just-in-time modeling. The new method can compute a gain of the control input to avoid an encountered obstacle with small computation time. Some numerical simulations show that the car can avoid various obstacles, and hence the effectiveness of the proposed method can be checked.

Future work on automatic driving of cars includes the next topics: an extension to avoidance control for moving obstacles, automatic parking and automatic driving based on just-in-time modeling.



(a) Vc = 10.9 m/s, To = 10.4 s, Xo = 48.5 m, Ro = 1.8 m

(b) Vc = 12.4 m/s, To = 13.1 s, Xo = 40.7 m, Ro = 3.4 m

(c) Vc = 14.8 m/s, To = 14.5 s, Xo = 56.3 m, Ro = 4.7 m

Fig. 3: Numerical simulation results.

REFERENCES

[1] N. Navet and F. Simonot-Lion ed., Automotive Embedded Systems Handbook (Industrial Information Technology), Society of Automotive Engineers Inc., 2013

[2] R. K. Jurgen, *Autonomous Vehicles for Safer Driving*, Society of Automotive Engineers Inc., 2013 http://dx.doi.org/10.4271/PT-158

[3] Konrad Reif ed., Automotive Mechatronics: Automotive Networking, Driving Stability Systems, Electronics (Bosch Professional Automotive Information), Springer Vieweg, 2014

[4] R. Kitchin, The Data Revolution: Big Data, Open Data, Data Infrastructures and Their Consequences, SAGE Publications Ltd; New., 2014

[5] N. Marz and J. Warren, Big Data: Principles and Best Practices of Scalable Real-Time Data Systems, Manning Pubns Co., 2015

[6] R. Buyya, R. N. Calheiros, and A. V. Dastjerdi, *Big Data: Principles and Paradigms*, Morgan Kaufmann, 2016

[7] A. Stenman, F. Gustafsson, and L. Ljung, "Just in time models for dynamical systems," in *Proc. of the 35th IEEE Conf. on Decision and Control*, 1996, pp. 1115-1120. http://dx.doi.org/10.1109/CDC.1996.574658

[8] Q. Zheng and H. Kimura, "Just-in-time modeling for function prediction and its applications," *Asian Journal of Control*, vol. 3, pp. 35-44, 2001. http://dx.doi.org/10.1111/j.1934-6093.2001.tb00040.x

[9] S. Wakao, D. Hosogoshi, and T. Yamamura, "Fundamental study on the application of Just-In-Time Modeling to wind power estimation," in *Proc. of European Wind Energy Conference and Exhibition 2006*, 2006, pp. 412-416.

[10] Z. Ge and Z. Song, "A comparative study of just-in-time-learning based methods for online soft sensor modeling," *Chemometrics and Intelligent Laboratory Systems*, vol. 104, pp. 306-317, 2010. http://dx.doi.org/10.1016/j.chemolab.2010.09.008