International Journal of
Science and Engineering Investigations
vol. 5, issue 58, November 2016
ISSN: 2251-8843

Moving Obstacle Avoidance Control of Cars Based on Big Data and Just-In-Time Modeling

Tatsuya Kai
Tokyo University of Science
(kai@rs.tus.ac.jp)

Abstract- In this paper a moving obstacle avoidance control problem for cars is considered and a control method based on big data and just-in-time modeling is developed. Just-in-time modeling is a new kind of data-driven control techniques in the big data age and has been applied in various real systems. The main feature of the proposed method via just-in-time modeling is that a gain of the control input to avoid an encountered moving obstacle can be computed from a data-base which includes a lot of driving data in different situations. Especially, the noteworthy advantage of the new control method is small computation time, and hence real-time control can be achieved. From numerical simulations, it turns out that the car can avoid various moving obstacles by the proposed method.

Keywords- *Moving Obstacle Avoidance Control, Car, Automatic driving, Big Data, Just-In-Time Modeling*

I. INTRODUCTION

Nowadays, various research groups and companies develops automatic operation techniques for self-driving cars, which includes automatic obstacle avoidance, automatic braking, automatic parking, automatic driving, automatic tracking, and so on [1-3]. Some techniques have already been turned into actual utilization. On the other hand, the word "big data" is spreading rapidly and it is utilized in various research fields [4-6]. We are able to expect that the big data technique can be applied to automatic operation problems of cars. However, the number of the researches on automatic operation techniques from the viewpoint of big data is quite small and they are needed in terms of actual utilization of self-driving cars.

The objective of this research is to develop an avoidance control method of encountered moving obstacles for a car based on big data and 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 big data.

The content of this paper is as follows. First, in Section 2, the problem formulation on the moving obstacle avoidance control problem for a car is presented. Next, Section 3 gives a summary of just-in-time modeling and develops a new control method for moving obstacle avoidance control of the car. Then, some numerical simulations are demonstrated in order to check the effectiveness of the new method in Section 4.

II. PROBLEM FORMULATION

In this section, the problem formulation is presented. In this paper, we consider the next nonholonomic car model on the 2-dimensional plane as a mathematical model of a car:

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

where (x, y) is the center position of the car, θ is the heading angle of the car. In addition, u_1, u_2 are the control inputs of the car, where u_1 is the heading velocity of the car and u_2 is the rotating angular velocity of the car as shown in Fig .1.

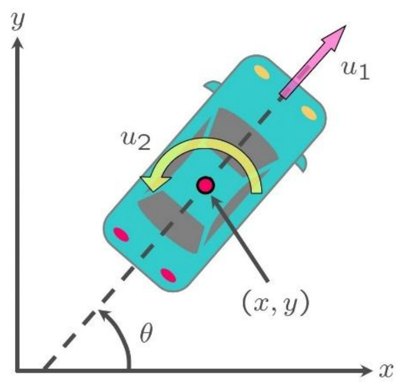


Figure 1. The nonholonomic car model.

Next, the problem setting of a moving obstacle is explained. We assume that the shape of the obstacle is circular and the center of the obstacle is located on the positive x –axis. We denote the center of the obstacle by $(X_o, 0)$ and the radius of the obstacle by R_o. It is also assumed that the obstacle moves to the positive y -direction with a constant velocity V_o as shown in Fig. 2.

Now, in order to avoid the collision with the moving obstacle, we consider the next control inputs u_1, u_2:

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

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

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

- Problem [Moving Obstacle Avoidance Control Problem]

For a given data: a velocity of the car V_c, an avoidance time T_c, a circular moving obstacle X_o, R_o, V_o, find a gain A of the control input u_2 which can avoid collision with the circular moving obstacle as shown in Fig. 2.

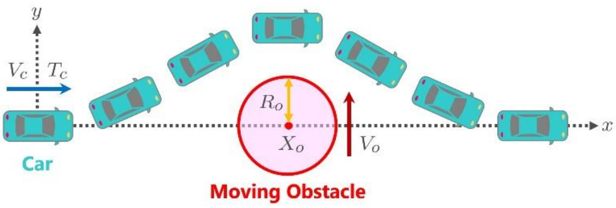


Figure 2. Moving obstacle avoidance control of a car.

In the next section, a new method to compute the gain based on just-in-time modeling will be derived as a solution of the above problem.

III. Just-In-Time Modeling Approach to Obstacle Avoidance Control

This section develops a new method of moving obstacle avoidance control of a car based on the just-in-time modeling approach.

A. Summary of Just-In-Time Modeling

First, this subsection shows a brief summary on just-in-time modeling. 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 predicted model at the query data from the obtained neighborhood data.

(4) Using the local linear predicted model, compute the predicted output for the query data.

Fig. 3 shows an illustration of the just-in-time modeling method. The advantages of just-in-time modeling is as follows; we do not need to 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.

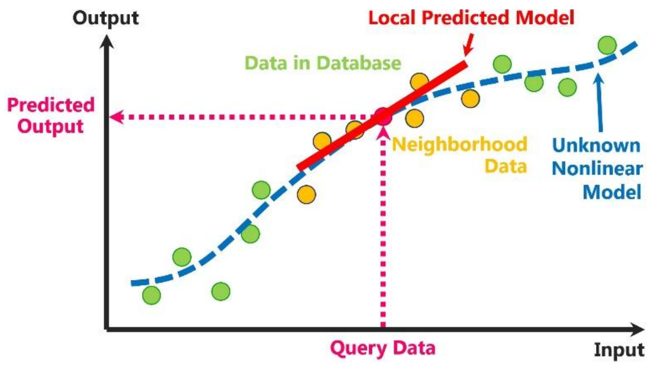


Figure 3. An illustration of just-in-time modeling.

B. Construction of Database

Next, we consider how to construct a database on moving obstacle avoidance of the car. At first, we have to decide the range of parameters in a database. For the heading 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

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

It must be noted 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 numbers of T_c, X_o, R_o, V_o as:

$$\begin{aligned} N_{T_c} &= \frac{T_c^{\max} - T_c^{\min}}{h_{T_c}} + 1, \\ N_{X_o} &= \frac{X_o^{\max} - X_o^{\min}}{h_{X_o}} + 1, \\ N_{R_o} &= \frac{R_o^{\max} - R_o^{\min}}{h_{R_o}} + 1, \\ N_{V_o} &= \frac{V_o^{\max} - V_o^{\min}}{h_{V_o}} + 1. \end{aligned} \quad (4)$$

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 $(V_c, T_c, X_o, R_o, V_o)$, 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:

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

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

$$N = N_{V_c} N_{T_c} N_{X_o} N_{R_o} N_{V_o} - N_{fail}, \quad (6)$$

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

C. Computation of Gain

Then, a computation method of a predicted 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 $(V_c^i, T_c^i, X_o^i, R_o^i, V_o^i)$ and the query data $(V_c^*, T_c^*, X_c^*, R_c^*, V_c^*)$ as:

$$d^i = \sqrt{\begin{aligned}&(V_c^i - V_c^*)^2 + (T_c^i - T_c^*)^2 \\ &+ (X_o^i - X_o^*)^2 + (R_o^i - R_o^*)^2 + (V_o^i - V_o^*)^2\end{aligned}}. \quad (7)$$

The distance d^i is also saved in the database as $(V_c^i, T_c^i, X_o^i, R_o^i, V_o^i, A^i, d^i)$ $(i = 1, \cdots, N)$. Next, we sort all the data in ascending order with respect to the distance d^i: $(\tilde{V}_c^i, \tilde{T}_c^i, \tilde{X}_o^i, \tilde{R}_o^i, \tilde{V}_o^i, \tilde{A}^i, \tilde{d}^i)$ $(i = 1, \cdots, N)$, and extract K $(K < N)$ data from the sorted database (the tilde over variables means sorted data). Finally, by using the K extracted data $(\tilde{V}_c^i, \tilde{T}_c^i, \tilde{X}_o^i, \tilde{R}_o^i, \tilde{V}_o^i, \tilde{A}^i, \tilde{d}^i)$ $(i = 1, \cdots, K)$, we calculate the predicted gain A^* of the query data $(V_c^*, T_c^*, X_c^*, R_c^*, V_o^*)$ as weighted mean:

$$A^* = \frac{\sum_{i=1}^{K} \frac{\tilde{A}^i}{\tilde{d}^i}}{\sum_{i=1}^{K} \frac{1}{\tilde{d}^i}}, \quad (8)$$

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

D. Proposed Algorithm

The procedure of obstacle avoidance control of the car based on just-in-time modeling is summarized as the next algorithm (see also Fig. 4).

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

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

Step 2. For a query data $(V_c^*, T_c^*, X_c^*, R_c^*, V_c^*)$, calculate distances d^i $(i = 1, \cdots, N)$ for all the data $(V_c^i, T_c^i, X_o^i, R_o^i, V_o^i)$ $(i = 1, \cdots, N)$ 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: $(\tilde{V}_c^i, \tilde{T}_c^i, \tilde{X}_o^i, \tilde{R}_o^i, \tilde{V}_o^i, \tilde{A}^i, \tilde{d}^i)\ (i = 1, \cdots, N)$ and extract $K\ (K < N)$ data from the sorted database $(\tilde{V}_c^i, \tilde{T}_c^i, \tilde{X}_o^i, \tilde{R}_o^i, \tilde{V}_o^i, \tilde{A}^i, \tilde{d}^i)\ (i = 1, \cdots, K)$.

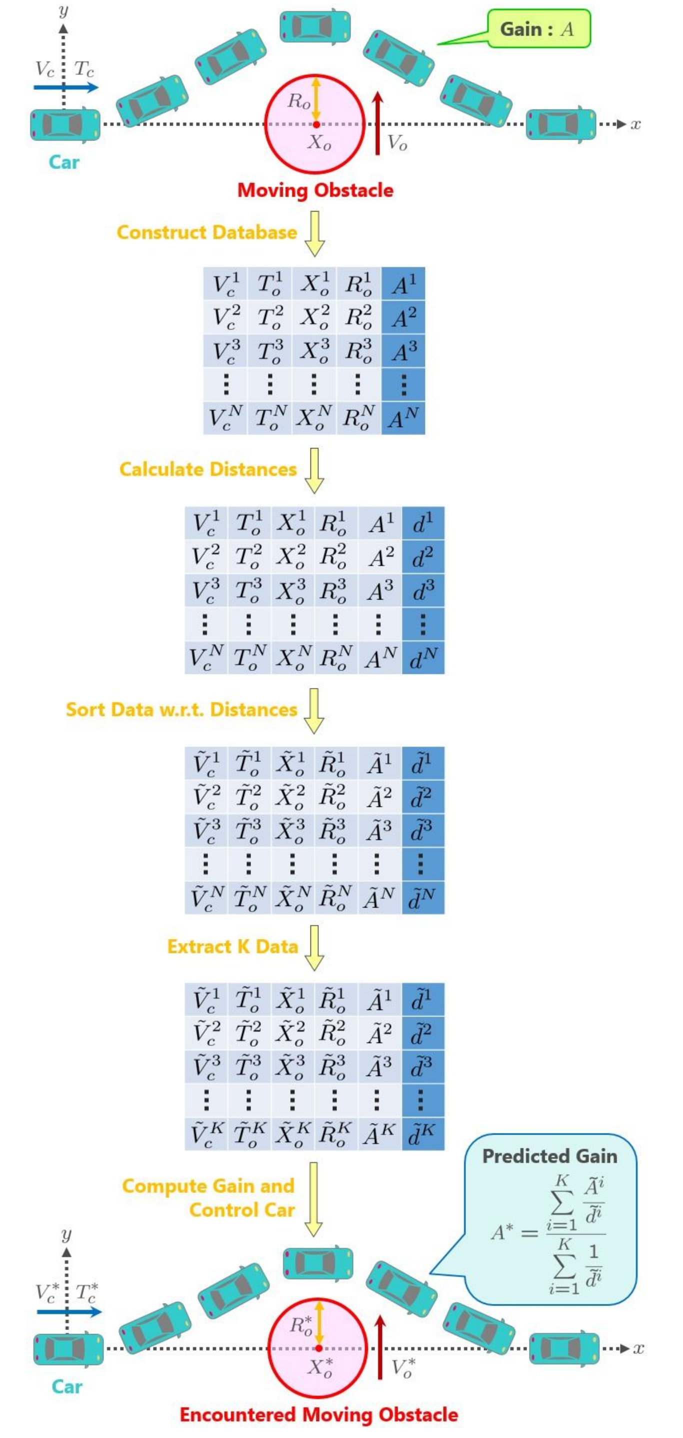


Figure 4. The algorithm on moving obstacle avoidance control of a car via just-in-time modeling.

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 predicted gain A^* by (8).

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

IV. NUMERICAL SIMULATIONS

In this section, some numerical simulations are demonstrated in order for confirmation of the effectiveness of the proposed method. First, a database on moving obstacle avoidance of the car is constructed for just-in-time modeling. For construction of a database, we set the ranges of input data as Table I.

TABLE I. PARAMETERS SETTING FOR CONSTRUCTION OF DATABASE

Parameters	Values	Parameters	Values
$V_c^{\min}$	8 m/s	$R_o^{\min}$	0.2 m
$V_c^{\max}$	12 m/s	$R_o^{\max}$	1.0 m
h_{V_c}	1 m/s	h_{R_o}	0.2 m
$T_c^{\min}$	8 s	$V_o^{\min}$	0 m/s
$T_c^{\max}$	12 s	$V_o^{\max}$	0.2 m/s
h_{T_c}	1 s	h_{V_o}	0.04 m/s
$X_o^{\min}$	40 m	$A^{\min}$	0.1
$X_o^{\max}$	60 m	$A^{\max}$	2.0
h_{X_o}	2 m	h_A	0.1

The parameter of the car is set as $R_c = 2m,\ R_{off} = 0.5m$. For the setting above, a database on moving obstacle avoidance control is generated by the method explained in Section III. A part of the database is shown in Table II. The number of data in the database is $N = 8250$, which is equal to

$$N = N_{V_c} N_{T_c} N_{X_o} N_{R_o} N_{V_o} = 5 \cdot 5 \cdot 11 \cdot 5 \cdot 6 = 8250, \quad (9)$$

hence $N_{fail} = 0$ holds.

By the database and the proposed algorithm, three kinds of numerical simulations are carried out. The settings of the simulations are as follows:

(a) Simulation I:

$V_c = 9.7\,m/s, T_c = 9.3\,s, X_o = 51.2\,m,$

$R_o = 0.35\,m, V_o = 0.14\,m/s$

(b) Simulation II:

$$V_c = 8.1\, m/s, T_c = 11.9\, s, X_o = 59.9\, m,$$

$$R_o = 0.97\, m, V_o = 0.19\, m/s$$

(c) Simulation III:

$$V_c = 8.1\, m/s, T_c = 11.9\, s, X_o = 59.9\, m,$$

$$R_o = 0.97\, m, V_o = 0.19\, m/s$$

TABLE II. DATABASE OBTAINED BY PROPOSED METHOD

Data No.	V_c	T_c	X_o	R_o	V_o	A
1	8	8	40	1	0	0.08
2	8	8	40	1	0.04	0.1
3	8	8	40	1	0.08	0.14
4	8	8	40	1	0.12	0.16
5	8	8	40	1	0.16	0.18
⋮	⋮	⋮	⋮	⋮		⋮
8248	12	12	60	5	0.14	
8249	12	12	60	5	0.16	0.08
8250	12	12	60	5	0.2	0.08

The results are illustrated in Fig. 5. This figure shows the trajectories of the car for three types of situations. From these 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.

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

[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

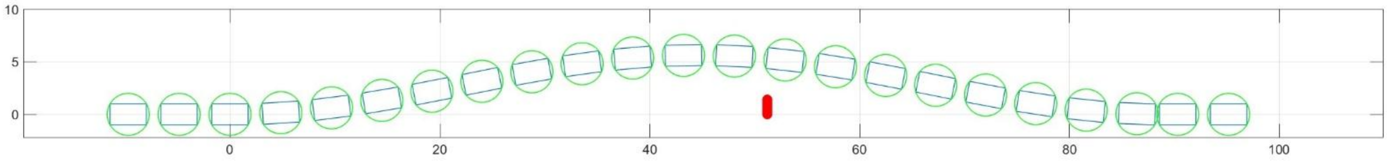
[6] R. Buyya, R. N. Calheiros, and A. V. Dastjerdi, *Big Data: Principles and Paradigms*, Morgan Kaufmann, 2016

[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.

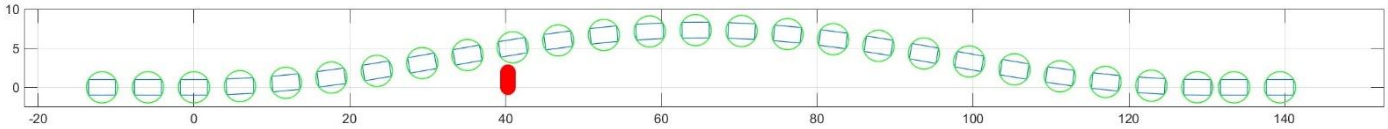
[8] Q. Zheng and H. Kimura, “Just-in-time modeling for function prediction and its applications,” *Asian Journal of Control*, vol. 3, pp. 35-44, 2001.

[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.

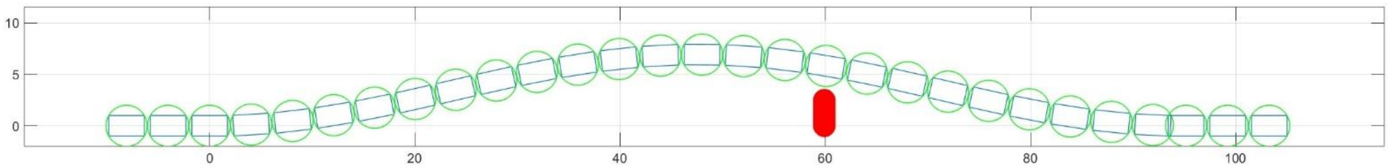
[10] Z. Ge and Z. Song, “A comparative study of just-in-time-learning based methods for online soft sensor modeling,” *Chemometrics and Intelligent Laboratory Systems*, vol. 104, pp. 306-317, 2010.



(a) Simulation I



(b) Simulation II



(c) Simulation III

Figure 5. Numerical simulation results (the circles in green: the car, the object in red : the moving obstacle).