Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 49-11 (2016) 168–174

Merging control of cooperative vehicles

A. Mosebach, S. Röchner and J. Lunze

Institute of Automation and Computer Control, Ruhr-Universität Bochum, Germany, e-mail: mosebach@atp.rub.de

Abstract: This paper presents a decentralized merging control algorithm for cooperative vehicles. The vehicles are controlled by a combination of a distance controller and a feed-forward control with a polynomial trajectory planning. In contrast to methods from recent literature, this approach stands out due to its low communication complexity. It is shown that the designed algorithm satisfies a collision-free merging of the vehicles. The effectiveness of the algorithm is illustrated by a simulation study.

Keywords: merging control, vehicles, tracking control, inter-vehicle communication

1. INTRODUCTION

The purpose of this paper is the design of a decentralized algorithm for the merging control of two vehicle traffic lanes (Fig. 1). The vehicles are assumed to be equipped with communication modules, which facilitate completely new strategies for the coordination of the vehicles. In particular, such communication systems allow vehicles to share information, which was previously only used for local purposes. Critical traffic situations, in which today's advanced driver assistance systems have to react to avoid e.g. collisions, could therefore be prevented before they even occur.

In this paper it is assumed that each vehicle knows its own position and the distance to the vehicle ahead. The control aim is the collision-free merging of two vehicle streams. The vehicles are allowed to exchange information inside the control region shown in Fig. 1. The point s_e of the control region is referred to as the merging point. The vehicles are required to leave the merging point s_e with the velocity v_e and with the distance d_e between them.

The vehicle merging problem is solved by a combination of a vehicle distance controller and a trajectory tracking controller, which is only activated inside the control region shown in Fig. 1. In the area in front of the control region, a string stable distance control algorithm is used for collision avoidance. The velocity of incoming vehicles is assumed to be less than or equal to the velocity v_e.

The designed merging algorithm can be summarized in the following three phases:

Data collection: Each vehicle entering the control region receives the merging time $t_{e,i-1}$ of the preceding vehicle.

Deceleration phase: Incoming vehicles decelerate until a collision-free passing of the merging point s_e can be guaranteed.

Acceleration phase: A tracking controller accelerates the vehicles such that the merging point s_e is reached with velocity v_e and distance d_e at the time instant $t_{e,i}$.

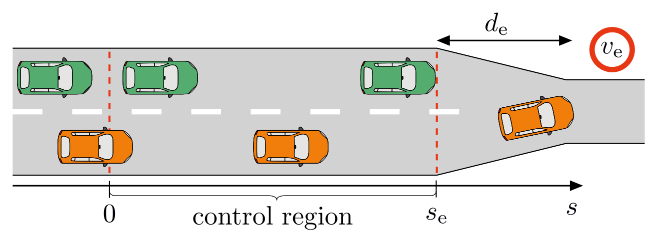

Fig. 1. Merging of two traffic lanes

The main result of this paper is a decentralized algorithm for the collision-free merging of two vehicle streams (Algorithm 1 and Theorem 3).

Literature survey. One of the first papers considering the vehicle-merging problem for two vehicle streams was presented by Athans (1969). The designed optimal controller requires an information exchanges between all the vehicles, but does not satisfy the requirement of a collision-free merging. In Schmidt and Posch (1983); Posch (1986); Lu and Hedrick (2000) and Ravari et al. (2007) additional centralized control approaches are presented that require a common coordinator. A decentralized control algorithm, which requires an extensive exchange of data between the vehicles is presented in Uno et al. (1999).

The coordination of vehicles on intersections without traffic lights is strongly related to the vehicle-merging problem. A centralized solution using the FCFS-strategy (First Come First Served) is presented in Dresner and Stone (2004, 2005), where an experimental evaluation is presented in Quinlan et al. (2010). An evolutionary algorithm was used in Wu et al. (2012) for the determination of the vehicle merging sequence. This algorithm handles a larger traffic volume more efficiently than the FCFS-strategy considered in Dresner and Stone (2004, 2005). In Naumann and Rasche (1997) and Naumann et al. (1997), decentralized strategies are used for the coordination of the vehicles. The intersection was divided into areas, which are only allowed to contain not more than one vehicle at the same time. The communication between the vehicles was realized with a token ring network, which results in large amount of data to exchange.

2405-8963 © 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.08.026

In this paper a decentralized control algorithm is designed, which solves the vehicle-merging problem for two vehicle streams. In contrast to existing decentralized methods, the communication between the vehicles is designed to be as low as possible. In the worst case, each vehicle communicates two times. To the knowledge of the authors, for the first time a combination of a distance controller and a trajectory tracking controller is used. Whereas the majority of publications consider simple integrator dynamics, this paper deals with a second-order model for the vehicles.

Structure of this paper. The vehicle model and the trajectory tracking controller is presented in Section 2. The string stable distance controller is introduced in Section 3. The main result is the merging algorithm derived in Section 4. Section 5 illustrates the merging algorithm in simulations.

2. VEHICLE MODEL AND TRACKING CONTROLLER

The dynamics of the i-th vehicle are given by the linear state-space model

$$\dot{\boldsymbol{x}}_i(t) = \underbrace{\begin{pmatrix} -\frac{\gamma}{m} & 0 \\ 1 & 0 \end{pmatrix}}_{\boldsymbol{A}} \boldsymbol{x}_i(t) + \underbrace{\begin{pmatrix} \frac{1}{m} \\ 0 \end{pmatrix}}_{\boldsymbol{b}} u_i(t), \quad \boldsymbol{x}_i(0) = \boldsymbol{x}_{0,i} \tag{1}$$

$$s_i(t) = \underbrace{(0 \;\; 1)}_{\boldsymbol{c}^{\mathrm{T}}} \boldsymbol{x}_i(t), \tag{2}$$

where the state vector $\boldsymbol{x}_i^{\mathrm{T}}(t) = (v_i(t), s_i(t))$ is given by the stack of the velocity $v_i(t)$ and the position $s_i(t)$. The mass of the vehicles is given by m, the friction by γ and the initial state by $\boldsymbol{x}_{0,i}$. The control input $u_i(t)$ corresponds to the acceleration force and the output signal $s_i(t)$ to the position of the i-th vehicle.

The solution of the vehicle-merging problem is obtained with a trajectory tracking controller. In the following it is shown under which condition there exists a command trajectory $w_i(t)$ such that the requirement

$$s_i(t) \overset{!}{=} w_i(t), \quad t \geq 0 \tag{3}$$

can be satisfied. The inversion-based feedforward controller in Lunze (2014) shows that the trajectory tracking controller is given by

$$u_i(t) = m\ddot{w}_i(t) + \gamma\dot{w}_i(t). \tag{4}$$

This trajectory tracking controller ensures (3), if the derivatives $\dot{w}_i(t)$ and $\ddot{w}_i(t)$ of the command trajectory $w_i(t)$ exist and are continuous.

3. DISTANCE CONTROLLER

Each of the vehicles (1)−(2) is equipped with a distance controller, which is used to keep the velocity-dependent distance

$$w_{\mathrm{s},i}(t) = \alpha\, v_i(t) + d_{\min} \tag{5}$$

to the vehicle driving ahead. The parameter α describes the ratio between the desired distance $w_{\mathrm{s},i}(t)$ and the velocity $v_i(t)$. $d_{\min}$ is the required minimum distance. The structure of the distance controller is shown in Fig. 2. The signal $v_{\mathrm{s},i}(t)$ of the internal control loop is determined by

$$v_{\mathrm{s},i}(t) = \begin{cases} \frac{1}{\alpha}(d_i(t) - d_{\min}) & \text{if} \quad d_i(t) < w_{\mathrm{s},i}(t) \\ v_{\mathrm{e}} & \text{else} \end{cases} \tag{6}$$

with $d_i(t) = s_{\mathrm{i}-1}(t) - s_i(t)$ describing the distance to the vehicle driving in front. If the distance $d_i(t)$ is greater than or equal to the desired distance $w_{\mathrm{s},i}(t)$, then there is no need for a distance controller and the set-point $v_{\mathrm{s},i}(t)$ is set to the desired velocity v_{e}. Otherwise $(d_i(t) < w_{\mathrm{s},i}(t))$, the velocity $v_{\mathrm{s},i}(t)$ is specified by the control loop of the distance controller.

The control error of the internal control loop is defined by $e_i(t) = v_{\mathrm{s},i}(t) - v_i(t)$. The controller $K(s)$ considering the velocity of the vehicle is described in the time-domain by the state-space model

$$\dot{\boldsymbol{x}}_{\mathrm{r},i}(t) = \underbrace{\begin{pmatrix} 0 & 0 \\ 1 & -s_1 \end{pmatrix}}_{\boldsymbol{A}_{\mathrm{r}}} \boldsymbol{x}_{\mathrm{r},i}(t) + \underbrace{\begin{pmatrix} k \\ 0 \end{pmatrix}}_{\boldsymbol{b}_{\mathrm{r}}} e_i(t), \quad \boldsymbol{x}_{\mathrm{r},i}(0) = \boldsymbol{x}_{\mathrm{r0,i}} \tag{7}$$

$$u_i(t) = \underbrace{(m \quad \gamma - ms_1)}_{\boldsymbol{c}_{\mathrm{r}}^{\mathrm{T}}} \boldsymbol{x}_{\mathrm{r},i}(t). \tag{8}$$

The state vector $\boldsymbol{x}_{\mathrm{r},i}^{\mathrm{T}}(t) = (x_{\mathrm{I},i}(t), x_{\mathrm{K},i}(t))$ consists of $x_{\mathrm{I},i}(t)$ (integrator state) and $x_{\mathrm{K},i}(t)$ (state of the lead–lag compensator). The control parameters α, k and s_1 are designed to fulfill the requirement on string stability, which is given by

$$||G_e(j\omega)||_\infty = \sup_\omega |G_e(j\omega)| \leq 1 \tag{9}$$

with $G_{\mathrm{e}}(s) = (s_{i-1}(t) - s_i(t))/(s_{i-2}(t) - s_{i-1}(t))$ (cf. Swaroop and Hedrick (1996)).

From the steady state of the control loop it follows with $\dot{v}_i(t) = 0$ and $\lim_{t\to\infty} v_i(t) = \bar{v}_i$ that

$$\lim_{t\to\infty} \boldsymbol{x}_{\mathrm{r},i}(t) = \begin{pmatrix} \bar{x}_{\mathrm{I},i} \\ \bar{x}_{\mathrm{K},i} \end{pmatrix} = \begin{pmatrix} s_1 \\ 1 \end{pmatrix} \bar{v}_i \tag{10}$$

holds.

4. MERGING ALGORITHM

4.1 Idea of the algorithm

Figure 3 illustrates the idea of the proposed merging algorithm for two neighboring vehicles entering the control region. The vehicle entering the control region first sends an information request to prove whether there are other vehicles inside the control region or not. If the control region is empty, the vehicle can pass through the control region with the constant velocity v_{e} (dashed line in Fig. 3). The solid line in Fig. 3 shows the case where there already is another vehicle in the control region. In this situation, the vehicle has first to be decelerated and subsequently to be accelerated in order to satisfy a collision-free merging with the desired distance $d_{\mathrm{e}} = \alpha\, v_{\mathrm{e}} + d_{\min}$.

The collision-free merging of the vehicles is achieved by the splitting of the vehicle's trajectory into two parts given by the deceleration phase ($t \in [0,\, t_{\mathrm{p},i}]$) and the acceleration phase ($t \in [t_{\mathrm{p},i},\, t_{\mathrm{e},i}]$). The determination of the merging time instant is thereby specified by the First-In-First-Out-strategy (FIFO). The deceleration phase is used to reduce