J Intell Robot Syst (2009) 54:709–731
DOI 10.1007/s10846-008-9285-7

Wheeled Mobile Robots Control in a Linear Platoon

Gregor Klančar · Drago Matko · Sašo Blažič

Received: 11 February 2008 / Accepted: 12 September 2008 / Published online: 5 October 2008

Abstract Future transportation systems will require a number of drastic measures, mostly to lower traffic jams and air pollution in urban areas. Automatically guided vehicles capable of driving in a platoon fashion will represent an important feature of such systems. Platooning of a group of automated wheeled mobile robots relying on relative sensor information only is addressed in this paper. Each vehicle in the platoon must precisely follow the path of the vehicle in front of it and maintain the desired safety distance to that same vehicle. Vehicles have only distance and azimuth information to the preceding vehicle where no inter-vehicle communication is available. Following vehicles determine their reference positions and orientations based on estimated paths of the vehicles in front of them. Vehicles in the platoon are then controlled to follow the estimated trajectories. Then presented platooning control strategies are experimentally validated by experiments on a group of small-sized mobile robots and on a Pioneer 3AT mobile robot. The results and robustness analysis show the proposed platooning approach applicability.

Keywords Mobile robots · Platooning · Path following control · Reactive multiagent system

1 Introduction

Intelligent transportation systems are a growing domain and have been a subject of extensive research in the mobile robotics field and in intelligent vehicles. The

G. Klančar (✉) · D. Matko · S. Blažič
Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia
e-mail: gregor.klancar@fe.uni-lj.si

D. Matko
e-mail: drago.matko@fe.uni-lj.si

S. Blažič
e-mail: saso.blazic@fe.uni-lj.si

capability of the vehicle to precisely follow the leading vehicle has potential in automated highway systems such as automated or semi-automated adaptive cruise control systems and vehicle platoon systems. Further improvements include avoiding traffic jams due to the increase in vehicle density on highways and security improvements in car travel due to an automated or semi-automated driving assistance, such as adaptive cruise control, obstacle detection and avoidance, automatic car parking and the like. Most of these platoon systems are based on a linear configuration (virtual train of vehicles).

Among the several problems associated with the control of platooning systems, longitudinal and lateral control are the most important ones. Longitudinal control involves controlling the braking and acceleration in order to stabilize the distance between the leading vehicle and the following vehicle. This control takes as a parameter the distance between the leading and the following vehicles. Sheikholeslam and Desoer [25] proposed a form of longitudinal control based on linearization methods. Ioannou and Xu [11] controlled the brakes and the acceleration using a fixed-gain PID control with gain scheduling. In contrast, Hedrick, Tomizuka and Varaiya [10] used a control mode based on a non-linear method with PID. Lee, Tomizuka, Jung and Kim [16, 17] proposed a longitudinal control based on fuzzy logic.

Lateral control involves aligning the vehicle's direction relative to the vehicle in front of it. Daviet and Parent [6] proposed a form of lateral control using a PID controller. This control consists of keeping the angle between the leading and the following vehicles close to zero. In literature, articles can be found dealing with lateral and longitudinal control using physics-inspired models. For instance, Gehrig and Stein [8] designed a model based on particles' submissive forces, whereas Yi and Chong [26] developed an impedance-control immaterial hook model. Halle and Chaib-draa [9] used a Multi-Agent System (MAS) in order to model immaterial vehicles using constant values from [6]. Contet, Gechter, Gruer and Koukam [4] proposed a solution for longitudinal and lateral control using Newtonian forces in an interactive model. In Bom et al. [2] a global platooning control strategy is proposed using a nonlinear control law which decouples the lateral and longitudinal control. Reynolds [23] proposed a distributed behavior model to mimic animal behaviors in herds, schools and flocks for computer animations use. Group motions and formations of animals were simulated there, using independent agents having local perception of dynamic environment. Fredslund and Matarić [7] present the formations control algorithm of a group of distributed robots where a relative position information to other robots is obtained by means of local sensing. Similarly, Cortés et al. [5] presented coverage control algorithm for a group of autonomous vehicles in a distributed sensing task, where each vehicle depends only on information of its neighbors.

In this work an approach to a platoon of nonholonomic vehicles using the state-space control of nonholonomic systems is presented. The vehicle platooning control strategy relies on relative information to preceding vehicles only and therefore no explicit inter-vehicle data exchange and global information (such as GPS) is required. The important advantage here is that relative information can be measured with low-cost sensor sets. Additionally, the method to obtain on-line objectives for the follower vehicles control is presented, where the inter-vehicle distance is curvilinear as also proposed in [2]. Nonholonomic systems have motion limitations resulting from their kinematic model. Therefore, some directions of motion are

not possible. This paper deals with a differential drive which cannot move in the direction perpendicular to the wheels. Although the proposed algorithm is derived for the differential type of vehicle, the proposed approach is also applicable to other types of vehicles. The vehicles whose kinematics model can be approximated with the differential drive can apply this approach directly. In a more general case only the trajectory-tracking controller structure consisting of feed-forward control inputs and a feed-back controller should be adapted according to the vehicles kinematics, while the rest of the presented approach remains the same. Vehicles in the platoon are reactive agents with trajectory tracking behavior which forces each vehicle in the platoon to follow the path of the vehicle preceding it. This behavior is achieved by the proposed control approach where each following agent has the ability to memorize the course of the preceding vehicle path. The leading vehicle in the platoon follows the course of the reference trajectory or it is operated by a human.

Controlling nonholonomic systems as they follow a reference path is a well-known problem that has been studied by many authors [14, 18, 19, 24]. The control of such a group of vehicles is solved by considering its first-order kinematic model. The obtained motion can later be upgraded to include the dynamic properties (inertia and mass) as well, where the vehicle-control problem is transferred to controlling a second-order kinematic model [19, 24]. The control of vehicles, especially mobile robots, by considering only the first-order kinematics is very common in literature [1, 3, 15, 22] as well as in practice. This is mainly because the control problem is easy to solve, the system dynamics can usually be neglected (because of the powerful engines), especially at moderate speeds, and because the vehicles' design sometimes does not allow the torque or acceleration to be forced at the vehicle input (only the reference speed can be set).

Next, the vehicle has to consider nonholonomic constraints, so its path cannot be arbitrary. The vehicle usually moves in an environment with obstacles, limitations and other demands, all of which define the desired path of the vehicle. All these facts suggest the vehicle should be controlled on a reference path, which should follow all the kinematic constraints. Nonholonomic systems usually have feed-forward control [3, 19, 22, 24], where the system inputs are calculated from the known trajectory and all the kinematic constraints are implicitly considered by the trajectory design. However, the use of only open-loop control (just feed-forward) is practically useless because it is not robust to errors in the initial system states and other disturbances during the operation. A closed loop is, therefore, added for practical use. The above-mentioned combination (feed-forward and feed-back control) is intuitive and suitable for most nonholonomic mechanical systems.

On the basis of the vehicle's kinematic equations the control is designed to follow an arbitrary reference path with a predefined velocity profile. The proposed control algorithm is validated by experiments on real small-size soccer robots with differential drive and on a four-wheel mobile robot Pioneer 3AT.

The work presented in this paper is organized as follows. In Section 2 a model of nonholonomic systems and the corresponding control law that can be applied to such systems are presented. The application of the proposed control law to the platoon system is derived in Section 3. The results of the experiments obtained on the two different mobile robot types are presented in Section 4 and the conclusions are given in Section 5.

2 Modeling and Control of Nonholonomic Systems

In the following, the direct and inverse kinematics for mobile vehicles with a differential drive are determined. The vehicle's architecture, together with its symbols, is shown in the left part of Fig. 1. The pose of the vehicle on a plane is defined by the vector of system states $\mathbf{q}(t) = [x(t), y(t), \theta(t)]^T$. The kinematic equations are equivalent to those for a unicycle. Robots with such an architecture have a nonintegrable constraint in the form $\dot{x}\sin\theta - \dot{y}\cos\theta = 0$, resulting from the assumption that the robots cannot slip in lateral direction. The kinematic model whose vector fields span the null space of the nonintegrable constraint is as follows

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{1}$$

where v and ω are the tangential and angular velocities of the platform shown in the left part of Fig. 1. The velocities of the vehicle's right and left wheel (needed for a low-level velocity control) are then expressed as $v_R = v + \frac{\omega B}{2}$ and $v_L = v - \frac{\omega B}{2}$, respectively, where B is the distance between the robot wheels. The low-level control is implemented in the robot microcontroller. Each wheel (motor) speed is calculated by a discrete PID controller; this ensures that the robot drives with the required reference wheel speeds.

High-level control implemented on a personal computer is proposed in this study. High-level control enables each vehicle in the platoon to precisely follow the path of the vehicle in front of it. The calculated robot control inputs **u** (speeds) are the reference inputs for the robot's low-level control.

For a given reference trajectory $(x_r(t), y_r(t))$ defined in the time interval $t \in [0,T]$ the feed-forward control inputs v_{ff} and ω_{ff} are derived from the kinematics model (1). However, the calculated robot inputs drive the robot on a desired path

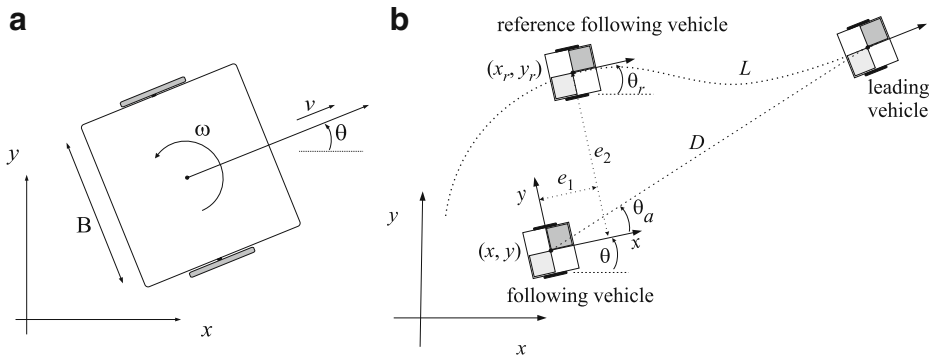

Fig. 1 Vehicle architecture and symbols (**a**) and illustration of the error transformation (**b**), where the following vehicle follows the leading vehicle path

only if there are no disturbances and no initial state errors. The tangential velocity v_{ff} and angular velocity ω_{ff} are calculated from the reference path as follows

$$v_{ff}(t) = \sqrt{\dot{x}_r^2(t) + \dot{y}_r^2(t)} \qquad \omega_{ff}(t) = \frac{\dot{x}_r(t)\ddot{y}_r(t) - \dot{y}_r(t)\ddot{x}_r(t)}{\dot{x}_r^2(t) + \dot{y}_r^2(t)} \tag{2}$$

The necessary condition in the path-design procedure is a twice-differentiable path and a nonzero tangential velocity $v_{ff}(t) \neq 0$. If for some time t the tangential velocity $v_{ff}(t)=0$, the robot rotates at a fixed point with the angular velocity $\omega_{ff}(t)$ which cannot be determined from (2) and therefore must be given explicitly.

When a vehicle is controlled to drive on a reference path, it usually has some state-following error. This state-tracking error, $\mathbf{e}(t) = [e_1(t)\ e_2(t)\ e_3(t)]^T$ expressed in terms of the real vehicle, as shown in the right part of Fig. 1, reads

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \tag{3}$$

In the right part of Fig. 1 the reference vehicle is an imaginary vehicle that ideally follows the reference path. In contrast, the real vehicle (when compared to the reference vehicle) has some error when following the reference path. Therefore, the control algorithm was designed to force the vehicle to precisely follow the reference path as proposed in [19, 20, 22]. It is as follows

$$\mathbf{u} = \begin{bmatrix} v \\ \omega \end{bmatrix} = \mathbf{u}_{ff} + \mathbf{u}_{fb} = \begin{bmatrix} v_{ff}\cos e_3 \\ \omega_{ff} \end{bmatrix} + \begin{bmatrix} v_{fb} \\ \omega_{fb} \end{bmatrix} \tag{4}$$

where v and ω are the reference velocities (set-points) for the low-level control controlling the wheels of the vehicle, while v_{fb} and ω_{fb} are the outputs of the feed-back controller given by

$$\begin{bmatrix} v_{fb} \\ \omega_{fb} \end{bmatrix} = \begin{bmatrix} k_1 & 0 & 0 \\ 0 & k_2 & k_3 \end{bmatrix} \cdot \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \tag{5}$$

The schematic of the obtained control is explained in Fig. 2. An intuitive explanation of the gain matrix's structure (5) can also be obtained from observing Fig. 1. To reduce the error in the driving direction e_1 the tangential robot velocity should be

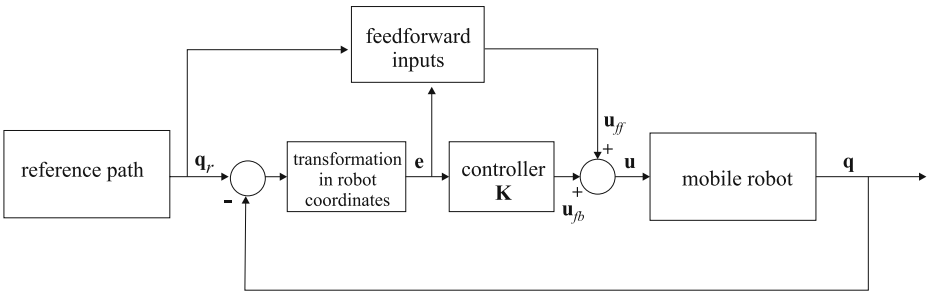

Fig. 2 Mobile-vehicle trajectory tracking control schematic consisting of feed-forward part u_{ff} and feed-back part u_{fb}

changed accordingly. Similarly, the orientation error e_3 can be efficiently manipulated by the robot's angular speed. Finally, the error orthogonal to the driving direction can be reduced by changing the angular velocity.

The gains k_1, k_2 and k_3 of the state feed-back controller $\mathbf{K}$ can be determined by trial and error, by comparing the actual and the desired characteristic polynomial equations as proposed in [20] or by a predictive trajectory tracking control approach which results in varying feedback gain matrix $K(t)$ as done in [13].

3 Application of the Controller to a Linear Platoon

It is supposed that there is no data communication between the leading and the following vehicles. The following vehicle measures the distance D and the azimuth θ_a (relative to its own orientation, see the right part of Fig. 1) of the leading vehicle. The distance D and the azimuth θ_a information are obtained from the laser range finder sensors which directly measure distances and angles to the objects. To ensure a stable control also a measurement of the orientation of the following vehicle (e.g. with a compass) is needed. No other sensors (e.g. GPS) are required. All the positions are treated in a coordinate system that is fixed to the ground. The following vehicle determines its own position using odometry. Having the current position $\mathbf{X}(k\Delta t) = [x(k\Delta t), y(k\Delta t)]^T$, the position in the next sample is determined by a simple Euler integration

$$\mathbf{X}((k+1)\Delta t) = \mathbf{X}(k\Delta t) + \begin{bmatrix} \cos\big(\theta(k\Delta t)\big) \\ \sin\big(\theta(k\Delta t)\big) \end{bmatrix} v(k\Delta t)\Delta t \tag{6}$$

where $\theta(k\Delta t)$ is the current (measured) orientation of the following vehicle, $v(k\Delta t)$ is the current speed of the vehicle, and Δt is the sample time. As shown later, the method of integration and the associated errors in the accuracy of the absolute position is not significant, since only the relative position of both vehicles is important.

Vehicle navigation using odometry information is the simplest method and easy to implement. Therefore it is preferred in practice as long as the error accumulation does not affect vehicle performance. To tackle this problem usually other global sensors (laser range scanners, GPS, stereo camera, etc.) are included in the vehicle navigation where the data can be additionally improved by using Kalman filtering or similar methods.

The path of the leading vehicle $\mathbf{X}_h(k\Delta t) = [x_h(k\Delta t), y_h(k\Delta t)]^T$ is calculated by the following vehicle using its current position and the current measurements of the distance $D(k\Delta t)$ and the azimuth $\theta_a(k\Delta t)$ (e.g. by using a laser range finder) as follows

$$\mathbf{X}_h(k\Delta t) = \mathbf{X}(k\Delta t) + \begin{bmatrix} \cos\big(\theta(k\Delta t) + \theta_a(k\Delta t)\big) \\ \sin\big(\theta(k\Delta t) + \theta_a(k\Delta t)\big) \end{bmatrix} D(k\Delta t) \tag{7}$$

This information is stored in the memory and represented in parametric form (with the parameter k). Intermediate points are obtained by means of linear interpolation

$$\mathbf{X}_h(t) = \mathbf{X}_h(k\Delta t) + \frac{t - k\Delta t}{\Delta t}\big[\mathbf{X}_h\big((k+1)\Delta t\big) - \mathbf{X}_h(k\Delta t)\big] \quad k\Delta t \le t < (k+1)\Delta t \tag{8}$$

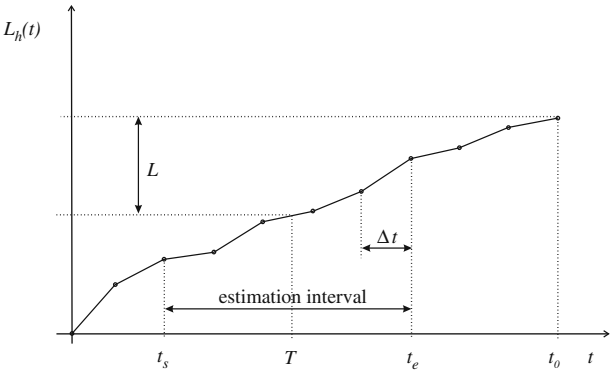

Fig. 3 The function $L_h(t)$ gives the distance travelled by the leading vehicle. The following vehicle tracks the leading vehicle at a distance L, i.e. at the current time t_0 it should follow the position of the leading vehicle at $t = T$

The following vehicle is supposed to track the leading vehicle at a distance L - measured on the path of the leading vehicle. In order to estimate the path length, the distance function $L_h(t)$ that gives the distance travelled by the leading vehicle against time is defined as

$$L_h(t) = \int_0^t \sqrt{\dot{x}_h^2(\tau) + \dot{y}_h^2(\tau)}d\tau \tag{9}$$

Note that the derivatives $\dot{x}_h(\tau)$ and $\dot{y}_h(\tau)$ formally do not exist in the sampling points $\tau = k\Delta t$ ($k \in \mathbb{Z}$), since $\dot{x}_h(\tau)$ and $\dot{y}_h(\tau)$ are not continuous in these points. Practically, this does not influence the calculation of $L_h(t)$ since only an integral of those functions is needed (the actual value of the derivative in discrete points does not influence the integral as long as the derivative is finite). An example of the function $L_h(t)$ is given in Fig. 3. From (9) it is obvious that the function $L_h(t)$ is continuous, monotonously nondecreasing, and $L_h(0) = 0$.

The following vehicle should track the leading vehicle at a distance L, i.e. at the current time t_0 it should follow the position of the leading vehicle at $t = T$ (see Fig. 3), where T is defined by

$$L_h(T) = \begin{cases} L_h(t_0) - L & L_h(t_0) \geq L \\ 0 & L_h(t_0) < L \end{cases} \tag{10}$$

Note that in the case of $\frac{dL_h(t)}{dt} = 0$ where the solution T of (10) is sought, the solution is not unique (this is practically impossible). In that case, the biggest T that satisfies (10) is used. Next, the path of the leading vehicle is expressed in the parametric polynomial form in the interval $[t_s, t_e]$ where $T \in [t_s, t_e]$:

$$\begin{aligned} \hat{x}_h(t) &= a_2^x t^2 + a_1^x t + a_0^x \\ \hat{y}_h(t) &= a_2^y t^2 + a_1^y t + a_0^y \end{aligned} \tag{11}$$

where the functions $\hat{x}_h(t)$ and $\hat{y}_h(t)$ approximate the functions $x_h(t)$ and $y_h(t)$, respectively, in the interval $[t_s, t_e]$. The coefficients of the polynomials a_i^x and a_i^y ($i = 0, 1, 2$) are calculated using the least-squares method with at least three samples around the time T (six points were used in our experiments). The second order polynomials (11) can successfully approximate the small piece of the leading vehicle path (consisting only of a few samples around the time T) because of the small

sampling time Δt compared to the vehicle driving dynamics. As already mentioned the position error due to dead reckoning (navigation with odometry) is not critical in the proposed algorithm because only the relative information (D and θ_a) between the leading and the following vehicle is important. Moreover, in the short time window (a few samples around T) where polynomials (11) are estimated, these errors do not significantly affect the trajectory estimation.

The reference position and the orientation of the following vehicle are determined using

$$\mathbf{X}_r = \begin{bmatrix} \hat{x}_h(T) \\ \hat{y}_h(T) \end{bmatrix} = \begin{bmatrix} a_2^x T^2 + a_1^x T + a_0^x \\ a_2^y T^2 + a_1^y T + a_0^y \end{bmatrix} \tag{12}$$

$$\theta_r = \arctan \frac{2a_2^y T + a_1^y}{2a_2^x T + a_1^x} \tag{13}$$

where arctan is a 4-quadrant version of the inverse tangent function. In the right part of Fig. 1 they are denoted as the reference vehicle. The tangential and angular velocities of the reference vehicle (needed for the feed-forward control (4)) are

$$v_{ff}(t) = \sqrt{\left(2a_2^x T + a_1^x\right)^2 + \left(2a_2^y T + a_1^y\right)^2} \tag{14}$$

and

$$\omega_{ff}(t) = \frac{\left(2a_2^x T + a_1^x\right) \times 2a_2^y - \left(2a_2^y T + a_1^y\right) \times 2a_2^x}{\left(2a_2^x T + a_1^x\right)^2 + \left(2a_2^y T + a_1^y\right)^2} \tag{15}$$

respectively. For the feed-back control (5) the error vector is given according to (3) by

$$\mathbf{e} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{X}_r - \mathbf{X} \\ \theta_r - \theta \end{bmatrix} \tag{16}$$

Using relations (14) to (15) the feed-forward part and the feedback-part of the linear platoon control law are completely defined taking into account (4) and (3). This enables vehicles in a platoon to drive in a linear formation.

3.1 Obstacle Avoidance

In the environment with fixed or dynamically changing obstacles it is possible that the obstacle appears between the follower and the leader. In this case the following robot must avoid these obstacles using vector fields avoidance algorithm as proposed in the following. The error vector given in (3) used in the feedback part of the control law is adapted to include also the influence of obstacles. The global reference pose on the trajectory $(x_r(t), y_r(t), \theta_r(t))$ in terms of vehicle reads e_1, e_2 and e_3(see Fig. 4).

According to Fig. 4, the attraction vector $\vec{a} = (e_1, e_2)$ to the reference point on the trajectory and the repulsion vector $\vec{r}$ showing away from the obstacle and perpendicular to the direction of $\vec{a}$ are defined. Such a choice of the repulsive vector

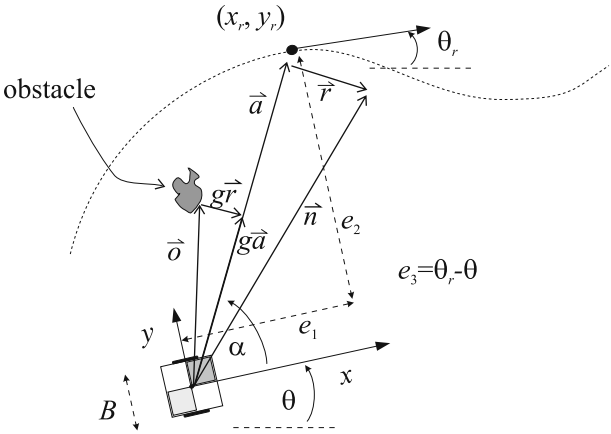

Fig. 4 Obstacle avoidance principle using attraction vector $\vec{a}$ and repulsive vector $\vec{r}$

avoids chattering behavior problems when the obstacle is in front of the vehicle. The closest distance of the obstacle to the line defined by $\vec{a}$ is

$$d = |o_y \cos\alpha - o_x \sin\alpha|$$

and the repulsive vector is then defined by

$$\vec{r} = \begin{cases} (0,0) & d \geq B \\ \vec{a} - \frac{\vec{o}}{g} & d < B \end{cases}$$

where o_x and o_y are the components of the obstacle vector $\vec{o}$, $g = \frac{\vec{a}\cdot\vec{o}}{\|\vec{a}\|^2}$, symbol $\cdot$ is the dot product and B is the vehicle dimension. In this way the obstacle avoidance effect is present only when the obstacle distance to the line defined by $\vec{a}$ is less than the vehicle dimension B. The resultant vector $\vec{n}$ defining a new driving direction is then

$$\vec{n} = \vec{a} + \vec{r} = \begin{cases} \vec{a} & d \geq B \\ 2\vec{a} - \frac{\vec{o}}{g} & d < B \end{cases}$$

The new error vector e^* used in the feed-back part of controller (3) and which extends the error vector (16) with an obstacle avoidance part then reads

$$\mathbf{e}^* = \begin{cases} \mathbf{e} & d \geq B \\ [n_x, n_y, \angle\vec{n}]^T & d < B \end{cases}$$

where $\mathbf{e} = [e_1, e_2, e_3]^T$ is defined by relation (16), n_x and n_y are vector $\vec{n}$ components and $\angle\vec{n}$ is the angle of $\vec{n}$ according to the robot coordinate system.

3.2 Merging and Splitting of the Platoon

An important and interesting issue of merging and splitting of the platoons (i.e. how to incorporate new members to the platoon and how the members leave the platoon) will be addressed in the following. As before, the vehicles are supposed to have local sensor information only.

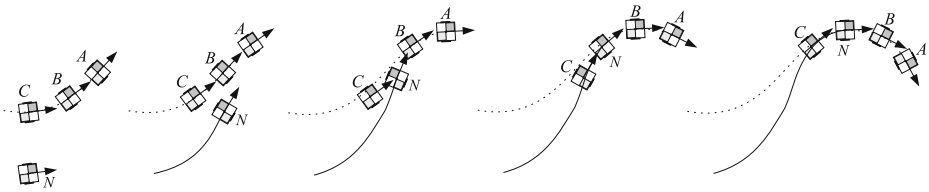

Fig. 5 Vehicle N joins the platoon between vehicles C and B. The path of the leading vehicle A is shown dashed, while the path of joining vehicle N is shown solid

Joining of the new vehicle to the platoon can be solved easily if the joining intention could be communicated to other vehicles in the platoon. Joining of the new vehicle is however also possible in case no communication is available if each vehicle follows the closest vehicle which is in the area in front of it. If the new joining vehicle is far from the platoon, it must first approach the platoon by defining the joining curve which drives it in the vicinity of the platoon. The new vehicle must approach the platoon (place of joining) so that it drives in parallel and behind some vehicle in the existing platoon. When the new vehicle comes in the vicinity of the platoon, it starts to follow the leading vehicle. The vehicles behind it will then automatically adapt to the new situation by the platoon controller described in Section 3. They will start to follow the new joining vehicle, respecting the requested safety distance among them. The joining of the new vehicle to the existing platoon can be thought of as a disturbance which affects only vehicles behind the joining vehicle. This disturbance vanishes after some transition phase. When the new vehicle joins the platoon as the last vehicle no disturbance is present.

The concept of a new vehicle joining the platoon is illustrated in Fig. 5

where vehicle N smoothly joins the platoon after vehicle B. This joining represents a smaller disturbance for vehicle C which starts to follow vehicle N and therefore leaves the path made by vehicle A. After short transition time this disturbance vanishes.

A more complicated situation arises when a vehicle wants to leave the formation (unless this is the last vehicle). This can be successfully achieved only if the other vehicles following the leaving one are informed about its intentions. The communication between the vehicles is therefore required. When the vehicle after the leaving vehicle receives the message about the leaving intentions of the preceding one, it starts to follow the next vehicle in the platoon (i.e. the preceding vehicle of the leaving vehicle).

3.3 Robustness Issues

How robust is the proposed platoon algorithm to unpredicted situations such as losing track of the leader vehicle, dynamically changing environment or obstacles obstructing the follower robot's view in the environment? In the Section 3 the presented control algorithm for multiple vehicles driving in a linear platoon works under given assumptions, such as: constantly available relative sensor information to the leader vehicle, no communication and no global information available. To deal with the mentioned unpredicted situations robustly, the proposed control algorithm should be upgraded with additional strategy as stated in the following.

In case the platoon is driving in environments with walls and corridors it is possible that the following vehicle loses view of the leading one. The following vehicle must still continue its path to the last position where the leading vehicle was seen. If the leading vehicle is observed, the basic algorithm can continue, otherwise the platoon stops. Further improvements of this situation are achieved if the leading vehicle also observes the progress (when and where it was last seen) of the follower. In case the follower is not seen or is more than $\mu \cdot L$ away, the leader stops and waits for the follower (L is the required safety distance and μ is threshold constant $\mu > 1$ e.g. $\mu = 1.5$). This waiting strategy then continues through the other following vehicles till the first one. The formation continues when the lost vehicle is seen again by its leader.

An important robustness issue concerns the case when more robots enter the platoon simultaneously. If the joining robots are positioned such that one lies in the visible area of the other, the proposed merging technique can still be applied, i.e., the robot that is behind the other one starts to follow the joining robot in front of it. The robot in front of it only follows the chosen robot in the platoon. This modification is quite easy to implement if the robots enter the platoon at the end of the formation. The situation is a little more complicated if they enter the platoon in the middle because the robot after the joining point must have all the joining robots within its view to make the correct decision about which robot to follow. The problem may also arise when the joining robots are side by side (without sensing each other) and both want to enter the platoon simultaneously. In such a case a collision is virtually inevitable without additional communication among the robots.

4 Experimental Results

The proposed algorithm for controlling a linear platoon of mobile robots was tested in simulations and experimentally on real small-size mobile robots as well as on a larger-size Pioneer 3AT mobile robot. In the following sections the results obtained from the experiments on real mobile robots are presented.

4.1 Experiments on Differential Type of Robots

The experimental results were obtained on a real mobile-robot platform explained in Fig. 6 which is a part of a robot-soccer set-up. The set-up consists of three Middle League MiroSot category robots of size 7.5 cm cubed, a digital color camera and a personal computer. The color camera mounted above the pitch is a global sensor. The vision part of the programme [12] processes the incoming image to identify the positions and orientations of the robots $\mathbf{q}_i(t) = [x_i(t), y_i(t), \theta_i(t)]$, $i = 1, 2, 3$. In the following, the explicit dependence of signals on time will be omitted to make the presentation clearer. The first (leading) robot (see Fig. 6) is controlled by the trajectory tracking controller (4) to follow the reference path $x_r(t)$, $y_r(t)$ defined in the time interval $t \in [0, T]$. According to the controller structure (4) the feed-forward control inputs v_{ff} and ω_{ff} are calculated by relations (2) while the feed-back part is determined by (5). The second (the first following) robot receives only the information about its distance and azimuth to the first robot and its own orientation. The third (the second following) robot receives only the information about its distance

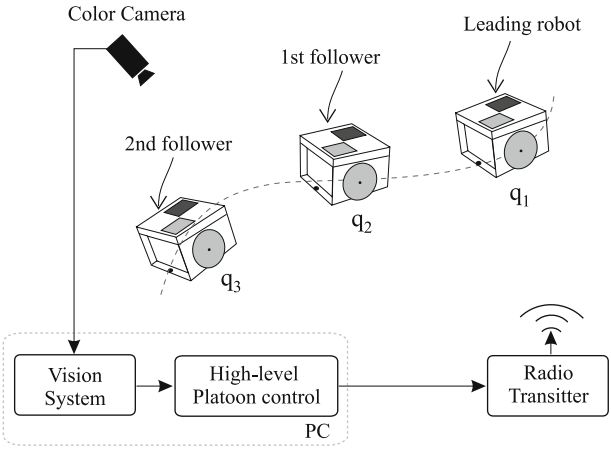

Fig. 6 Robot-soccer experiment set-up

and azimuth to the second robot and its own orientation. The noisy position estimates of the used camera sensor influence the calculated distance and azimuth information. The estimated noise standard deviation of the measured robots positions was 4 mm. The distances and azimuth orientations for each following robot i, $i = 2, 3$ are calculated by

$$D_i = \sqrt{(x_{i-1} - x_i)^2 + (y_{i-1} - y_i)^2}$$

$$\theta_{ai} = \arctan \frac{y_{i-1} - y_i}{x_{i-1} - x_i}$$

where arctan is a 4-quadrant version of the inverse tangent function.

Additionally, the system delay, d, mainly originating from the vision system, was compensated. As shown in Fig. 6 the state measurements of the robots are obtained from color camera and computer vision algorithm. The system delay is therefore caused by picture grabbing hardware and by computationally demanding computer vision estimation algorithm. Other sources of system delay are due to the computational time of control algorithms and wireless connection. From experiments the estimated common system delay time is $2\Delta t$ ($d = 2$) where sampling time is $\Delta t = 33$ ms. The undelayed system output $\bar{\mathbf{q}}$ can be estimated from the delayed system output $\mathbf{q}$ and from the difference between the undelayed and delayed model outputs using the same inputs as the system [21], i.e.,

$$\bar{\mathbf{q}}(k\Delta t) = \mathbf{q}(k\Delta t) + \mathbf{q}_m(k\Delta t) - \mathbf{q}_m((k-d)\Delta t)$$

where $\mathbf{q}_m(k\Delta t)$ is the output of the simulated system model without delay and $\mathbf{q}_m((k-d)\Delta t)$ is the output of the simulated system model including the estimated system delay.

The controller gain matrix (5) for controlling each mobile robot is

$$\mathbf{K} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 20 & 2 \end{bmatrix}$$

The results of the tests are shown in the right part of Fig. 7, where each robot in the platoon must follow the path of the robot in front of it and maintain the desired

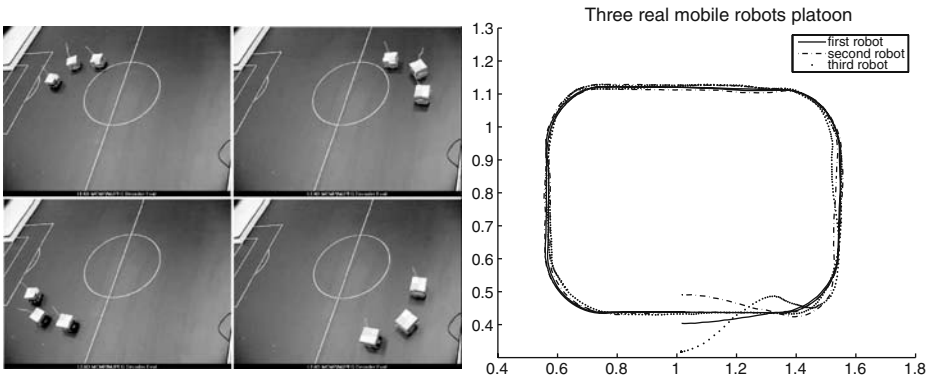

Fig. 7 Real set-up experiment (left) and results of real experiments (right)

safety distance $L = 20$ cm to that same robot. The film of the real experiments can be seen at [27].

In the left part of Fig. 8 the time course of the distance between the robots is presented. It is clear that after a transition phase (the second robot waits for the first robot to travel the distance L away before it starts to move, similarly the third robot waits for the second one, etc.) the second and the third robot follow with acceptable accuracy. From the results of the experiment in Fig. 7 it can be seen that the following robots nicely follow the paths of their preceding robots. Some smaller deviations from the first robot path are due to the noise in the position estimation, defined trajectory tracking controller dynamics and due to the time delay of the optical tracking and recognition. The left part of Fig. 8 shows the distances between the robots driving in the formation as illustrated in Fig. 7. The distances between the robots are close to the required safety distance L. The prominent difference from the required safety distance L at the beginning of the left graph in Fig. 8 appears due to initial robots position (robots were close to each other). When the first robot starts, the second one must wait until the first robot travels the distance L away, similarly the third robot waits for the second one. After some transition time ($3s$) the

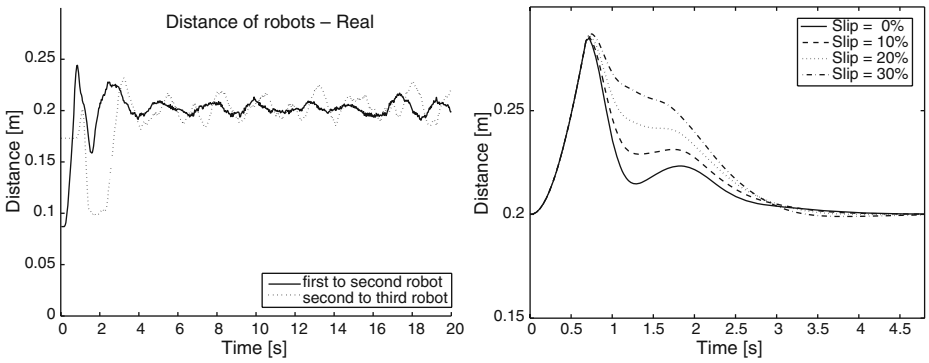

Fig. 8 Distance between robots — experiment (left) and distance between robots with slip in a straight path — simulation (right)

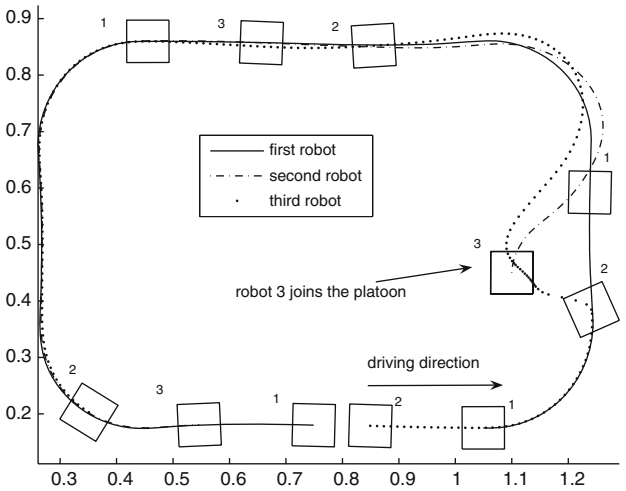

Fig. 9 Example of joining the new robot to the platoon. Robot 3 joins the platoon between robot 1 and robot 2

robots are arranged correctly and the distance among them is close to the required one. The accuracy of the integration method (6) and the associated error, which is equivalent to the slipping of the vehicle's wheels, is analysed and illustrated in the right part of Fig. 8, where the distance between the leading and the following platoon robots in a straight path is illustrated. Four simulated experiments were done. In each experiment a different but constant wheel slipping is simulated. In the first run the wheels did not slip (0% slip), in three other experiments the robot body velocity is only the part of the wheels velocity due to the slipping effect (10% slip means only 90% robot velocity). It can be seen that the constant slipping of the wheels has no influence on the steady-state distance of the platoon vehicles. The proposed control law compensates the slipping effect. This is expectable since wheel sleeping in our case does not influence servoing accuracy noticeably because relative information among vehicles (distances and azimuth orientations) are always obtained from accurate relative sensors.

The simulated example of joining the new robot to the existing platoon is shown in Fig. 9. In this example the strategy for joining robots to the platoon described in Subsection 3.3 is followed. The platoon starts with two robots (robot 1 and 2), and after some time robot 3 joins the platoon after robot 1 and before robot 2, as shown in Fig. 9. It is clear that robot 2 starts to follow robot 3 after robot 3 joins the platoon. As already described in Subsection 3.3, the joining of the new robot (robot 3) is seen as a disturbance for robot 2, which starts to follow robot 3 and therefore leaves the path made by robot 1. This disturbance vanishes after some transitional phase.

4.2 Experiment on Pioneer 3AT Mobile Robot

In this experiment in Fig. 10 the following robot measures the distance D and azimuth θ_a to the leading robot using a laser range scanner (Sick LMS200). The position of the following robot is obtained using odometry information (incremental encoders on the robot wheels) where the kinematic model of this robot is approximated by (1). The robot orientation is obtained by an orientation measurement

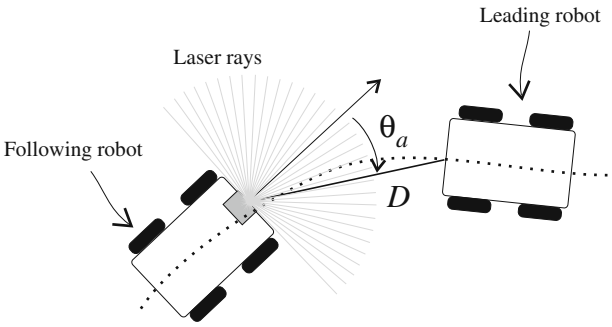

Fig. 10 Pioneer 3AT linear formation experiment set-up. The following robot measures the distance D and the angle θ_a to the leading robot

which ensures stable control. The orientation measurement can be obtained from a compass sensor where measurement noise can be filtered using Kalman filter. In this experiment the robot orientation is obtained by means of the already implemented localization algorithm with the predefined map of the environment. For unknown environments or dynamically changing environments simultaneous localization and mapping (SLAM) algorithms should be used. To resume, the required information which the following robot must have in order to follow the leading robot consists of: relative measurements to the leading robot (distance D and azimuth θ_a), odometrically estimated pose and information about its orientation in the environment. The controller gain matrix (5) for controlling the follower mobile robot is

$$\mathbf{K} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0.8 \end{bmatrix}$$

In the following, two different experiments using Pioneer 3AT are presented. In the first experiment the Pioneer 3AT robot follows the imaginary robot and in the second experiment it follows the soccer robot.

4.2.1 First Experiment: Pioneer 3AT Robot follows the Imaginary Robot

The results of the experiment are shown in Figs. 11 and 12. Because only one Pioneer 3AT robot was available, the experiment was performed using an imaginary (simulated) leading robot which drives on a reference path with a predefined velocity profile and the real robot which follows the leading one. In each sample time $\Delta t = 100$ ms the real following robot gets only relative information (D and θ_a) to the imaginary leading robot. This relative information is calculated from the known global position of the following robot and the position of the leading one. Global position of the leading robot is known from the implemented localization algorithm using an environment map. However, this global position is not used in the algorithm for controlling the robot to follow the leading one. Only the relative information (D and θ_a) to the imaginary leading robot and the following robot orientation are used to control the following robot.

The results of the test are shown in Fig. 11, where the following robot follows the path of the leading robot and tries to maintain the desired safety distance $L = 0.8$ m. The second robot follows the leading robot with acceptable accuracy as seen in the left part of Fig. 11. The safety distance between the robots (right part of Fig. 11) is also close to the desired one ($L = 0.8$). Because the imaginary head robot starts

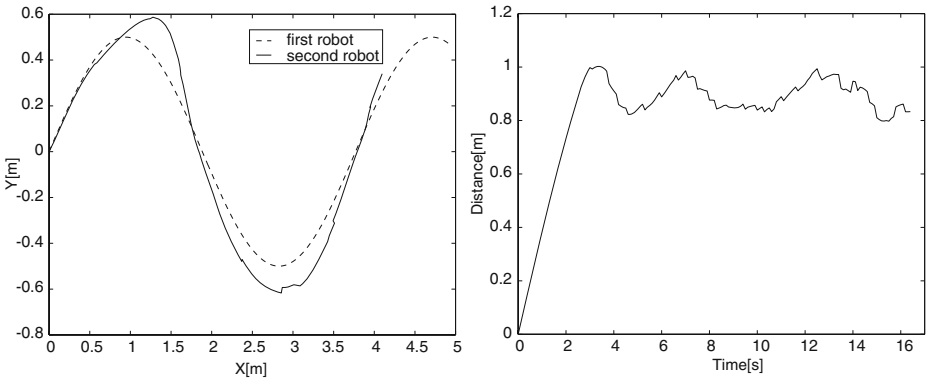

Fig. 11 Results of the experiment: robots' trajectory (left) and distance between the robots (right). The following robot position estimation is obtained by odometry

in the same position as the following one, the following robot must wait until the travelled distance of the leading robot is greater than L, then it starts to follow the leading robot's path. The reason for tracking error behavior is the robot kinematic model which is not well defined (as in the case of differential drive), especially when the robot rotates. It is very well-known that the four-wheel robot without steering mechanism does not allow rotation without sliding of the wheels. The prediction (odometry) based on an approximately estimated model (1) therefore results in tracking error.

The above experiment in Fig. 11 is repeated with additional sensor information to improve the robot position estimation. It is improved by laser range sensor information used in an implemented localization algorithm with a known environment map. The results of the experiment are shown in Fig. 12.

The difference in comparison with the previous experiments is that the robot pose is obtained by the before-mentioned localization algorithm and not by using the odometry as in Fig. 11. Now the second robot follows the path of the first robot with

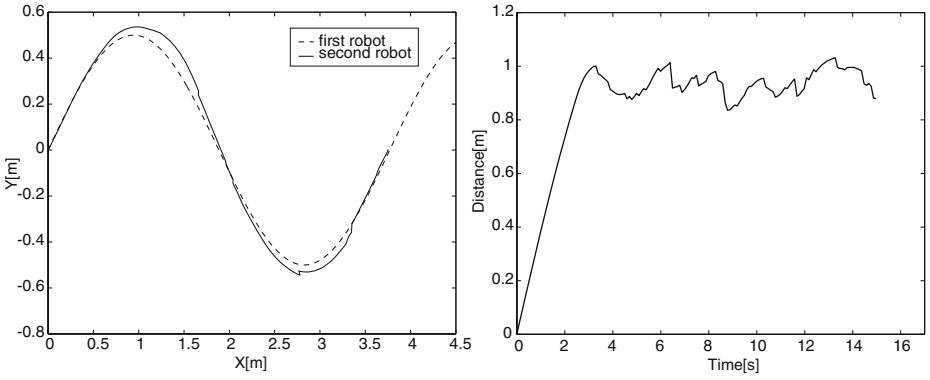

Fig. 12 Results of the experiment: robots' trajectory (left) and distance between the robots (right). The following robot position estimation is obtained from localisation (SLAM)

Fig. 13 View of the experiment (**a**) and screen shot of the control application (**b**)

a slightly better accuracy as in Fig. 11. The following robot still has some tracking error. However, to achieve a more precise tracking, some additional global sensor information is required such as: SLAM using laser range scanner, GPS sensor and the like.

4.2.2 Second Experiment: Pioneer 3AT Robot follows the Soccer Robot

The experimental environment is seen in the left part of Fig. 13 where the following robot - Pioneer 3AT (the bigger one) must follow the travelled path of the leading robot - soccer robot (the smaller one). The follower measures the distance and the relative orientation of the head robot using a laser range scanner sensor (Sick LMS200). The leading robot is made taller by an attached cylinder in order to be seen by the following robot's laser range scanner. The right part of Fig. 13 contains the view of the control application where the map of the environment and the laser range readings used in localization algorithm are seen. In the experiment the leading mobile robot was manually operated.

The results of the experiment are given in Fig. 14. As already mentioned the following robot follows the travelled path of the leading robot only by measured relative information D and θ_a to the leading robot. The position of the leading robot is not known. To draw the path of the leading robot in Fig. 14 the leading robot path must be calculated (similar to (7)) from the measured laser range readings (D and θ_a) and the known position (from localization algorithm) of the following (second) robot.

The calculated path of the leading robot contains high noise which results from the noise of laser range measurements and a recognition algorithm which identifies the leading robot from a laser range scan. The following robot follows the travelled path of the leading robot as seen in the left part of Fig. 14. Smaller deviations from the leading robot path are due to odometry error position estimate, systems delays, noise of laser measurement and the recognition algorithm. The controller structure (5) also does not allow the robot to simultaneously suppress tracking errors in orientation and lateral directions (e_2 and e_3 in relation (3)). To control the orientation of a Pioneer 3AT mobile robot the wheels must slide, which again is a more demanding task than in the differential mobile robot type. Nevertheless, the following robot can

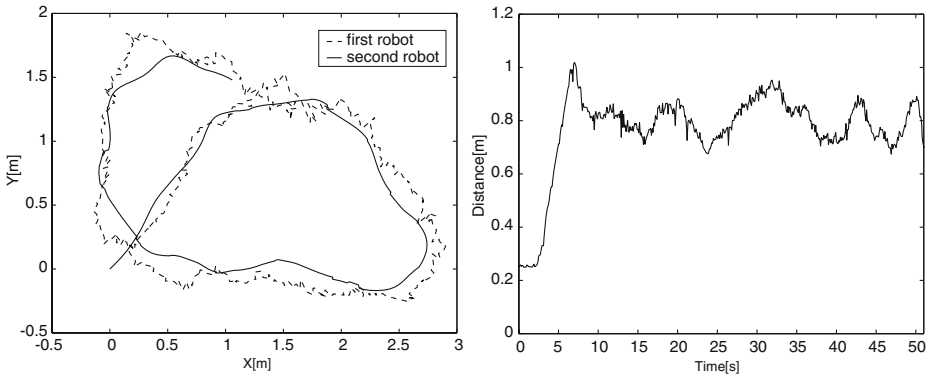

Fig. 14 Results of the experiment: robots' trajectory (left) and distance between the robots (right)

follow the leading robot satisfactorily and also the safety distance is close to the the required one. At the beginning of the experiment the following robot must wait until the travelled path of the leading robot is greater than L. The movies of the real experiments can be seen at [27].

4.3 Comparison with Related Work

In the following the proposed algorithm for vehicle platoons is compared to the related work as stated in the Introduction. The comparisons include local methods, as proposed in [5, 7, 23], and a global method, as proposed in [2]. These methods all deal with formation control for a group of vehicles, although each is specialized for a certain purpose, such as keeping the requested formation, platooning in urban transportation, sensing networks, etc. Nevertheless, some conclusions and an evaluation can be obtained from the comparisons on a simulated three-robot platoon experiment, as follows.

In Fig. 15 the resulting trajectories of the three-robot platoon obtained in this work's presented method is given.

Figure 16 gives the results of the platooning where the robots have information about their global position as well as the course of the reference path, and they do not need to estimate the reference path, as they do in the proposed approach. Using this approach each robot is controlled by a trajectory-tracking controller (4), using the global control strategy proposed in [2]. This approach enables very accurate reference-trajectory tracking, because each following robot just has to maintain the required safety distance to the robot in front of it. The safety distance is a curvilinear measurement (9) on a known reference path.

Figure 17 shows the result of the local strategy for controlling the robots in the platoon, where the robots have the same information as in the proposed approach in Fig. 15. The difference is that the vehicles do not estimate the trajectory of the robots in front of them, instead they react to current sensor information only. Each robot follows its assigned robot friend (the robot in front of it) and uses a control strategy to keep the robot friend in the center of the sensor range, as proposed in [7]. Each robot therefore measures the distance D and the azimuth θ_a to its robot friend. Considering the required safety distance L, the reference point in local robot coordinates (the

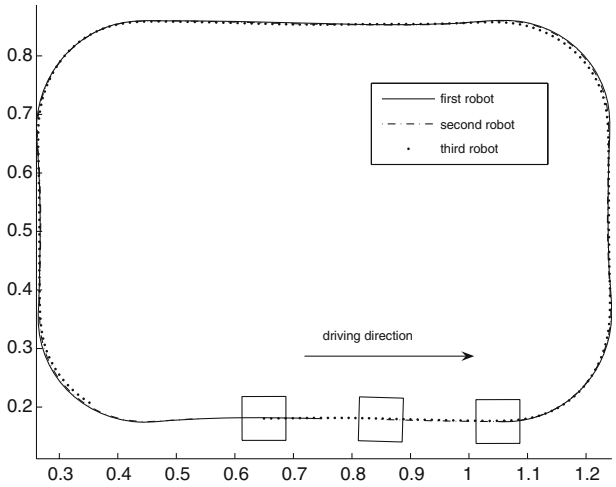

Fig. 15 A platoon of three mobile robots implemented by the proposed method

robot position is at the origin) is $\mathbf{X}_r = [(D-L)\cos\theta_a, (D-L)\sin\theta_a]^T$ and the feedback control is given by

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} k_1 & 0 & 0 \\ 0 & 0 & k_3 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{X}_r \\ \theta_a \end{bmatrix}$$

where the controller gains $k_1 = k_3 = 2$ if $D \geq L$ and $k_1 = k_3 = 0$ if $D < L$.

The tracking-performance comparison of the compared approaches is illustrated in Fig. 18, where the distance errors from the reference trajectory for each robot in the platoon for all three presented approaches (proposed, local and global) are shown. The overall calculated SSE_i index (sum of the squared tracking errors over the whole path) is given in Table 1, where i is the index of the robot in the platoon.

From Figs. 15–18 and Table 1 is clear that the best platooning performance is obtained with the global approach, where the tracking error does not accumulate

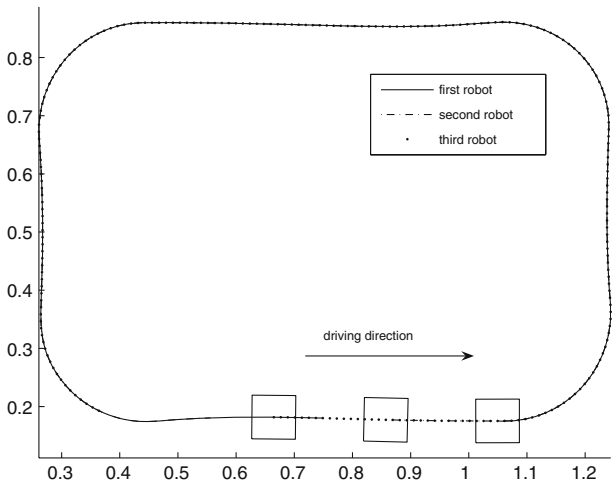

Fig. 16 A platoon of three mobile robots implemented by the global method

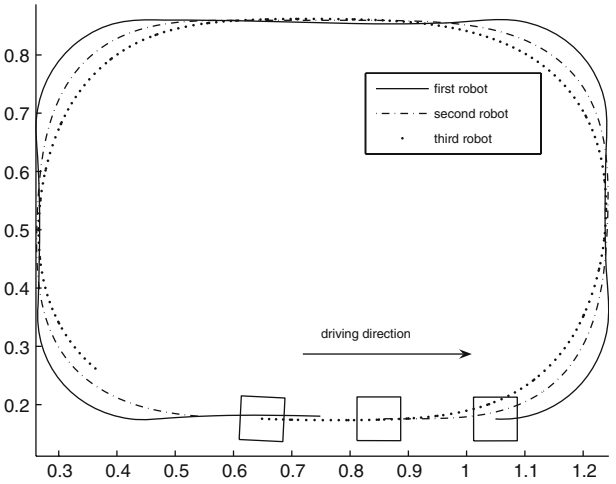

Fig. 17 A platoon of three mobile robots implemented by the local method

through the number of robots, because it is assumed that the reference path as well as the global robot poses are available. A poorer performance is obtained with the local approach, because the control algorithm for each robot has no information about the reference path. Each following robot in the platoon is simply controlled to keep its friend in the center of its sensor range and the required safety distance away ($D = L$ and $\theta_a = 0$). The tracking error in the case of the local approach accumulates through the number of robots in the platoon, as shown in Fig. 18. In this paper the proposed approach overcomes the disadvantages of the local approach because each following robot estimates the course of its robot friend's path (the robot in front of it) and follows it using the control strategy presented in Section 3. The obtained tracking results are close to the ones from the global approach. The tracking error still accumulates through the number of robots in the platoon, but much more slowly than in the local approach. The proposed approach has advantages over the global approach in situations when the global course of the path is not a-priori known to

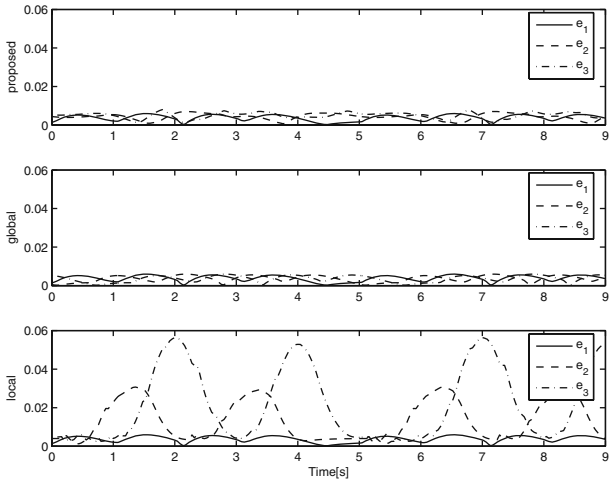

Fig. 18 Comparisons of the tracking errors from the reference trajectory for the proposed, global and local methods

Table 1 Comparison of the sum of the squared tracking errors over the whole path and for each robot in the platoon

METHOD	SSE_1	SSE_2	SSE_3
Proposed	0.0038	0.0049	0.0061
Global	0.0038	0.0038	0.0038
Local	0.0038	0.0641	0.1845

the robots or if it can change and when precise (better than one metre accuracy) global sensor information, such as GPS, is not available or is too expensive for the application.

5 Conclusion

A new algorithm for the control of vehicle platoons is proposed. The following vehicle only has information about its own orientation and about the distance and azimuth of the leading vehicle. Its own pose is determined using odometry and a compass. It calculates the reference path in a parametric polynomial form, and the parameters of the polynomials are determined by the least-squares method. Having the reference path, the feed-forward and the feed-back control are applied to the following vehicle. The following vehicle calculates its own position by means of a simple Euler integration. It was established that the error in the integration procedure (equivalent to the errors due to the wheel slipping) has a minor influence on the accuracy of the platoon distance in the case of differential drives. Additionally the solutions for obstacle avoidance and split and merge capabilities of the platoon members are briefly discussed.

Because each following vehicle in the platoon has the information about its preceding vehicle only, it also follows the effects of noise, disturbances and tracking errors of the preceding vehicle. This erratic behavior accumulates through the number of vehicles in the platoon, therefore the path of the last vehicle can diverge from the path of the first leading vehicle in case of a larger number of vehicles. This drawback can only be avoided if some additional information such as communication or global positions information is available or the course of the leading vehicle path is known to the following vehicles. For application to transportation systems in diverse environments with many vehicles the proposed local approach relying only on relative information should be upgraded. There the use of global information like precise GPS sensors, other sensors and communication capabilities would be necessary. Each vehicle should additionally track the vehicle in front as well as the vehicle behind. By adding communication abilities their positions and intensions such as joining of a new member to the platoon or the member leaving the platoon could be exchanged.

The proposed algorithm was tested on the robot-soccer test bed and on the Pioneer 3AT mobile robot. The results confirm the applicability of the proposed method.

Acknowledgements The authors would like to acknowledge the Slovenian Research Agency under CRP MIR M2-0116 project “Mobile Robot System for Reconnaissance, Research and Rescue” for funding this work.

References

1. Balluchi, A., Bicchi, A., Balestrino, A., Casalino, G.: Path Tracking Control for Dubin's Cars. In: Proceedings of the 1996 IEEE International Conference on Robotics and Automation, pp. 3123–3128. Minneapolis, Minnesota (1996)
2. Bom, J., Thuilot, B., Marmoiton, F., Martinet, P.: A Global Control Strategy for Urban Vehicles Platooning relying on Nonlinear Decoupling Laws. In: Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Alberta, pp. 1995–2000 (2005)
3. Canudas de Wit, C., Sordalen, O.J.: Exponential stabilization of mobile robots with nonholonomic constraints. IEEE Trans. Autom. Control **37**(11), 1791–1797 (1992)
4. Contet, J-M., Gechter, F., Gruer, P., Koukam, A.: Application of reactive multiagent system to linear vehicle platoon. In: Annual IEEE International Conference on Tools with Artificial Intelligence(*ICTAI*). Patras, Grece (2007)
5. Cortés, J., Martnez, S., Karatas, T., Bullo, F.: Coverage control for mobile sensing networks. IEEE Trans. Robot. Autom. **20**(2), 243–255 (2004)
6. Daviet, P., Parent, M.: Longitudinal and lateral servoing of vehicles in a pltoon. In: IEEE Intelligent Vehicles Symposium, Proceedings, pp. 41–46. Tokyo, Japan (1996)
7. Fredslund, J., Matarić, M.: A general algorithm for robot formations using local sensing and minimal communications. IEEE Trans. Robot. Autom. **18**(5), 837–846 (2002)
8. Gehrig, S.K., Stein, F.J.: Elastic bands to enhance vehicle following. In: IEEE Intelligent Transportation Systems Conference Proceedings, ITSC, pp. 597–602. Oakland, CA (2001)
9. Halle, S., Chaib-draa, B.: A collaborative driving system based on multiagent modeling and simulations. J. Trans. Res. Part C (TRC-C): Emergent Technol. **13**(4), 320–345 (2005)
10. Hedrick, J., Tomizuka, M., Varaiya P.: Control issues in automated highway systems. IEEE Control Syst. Magazine **14**(6), 21–32 (1994)
11. Ioannou, P., Xu Z.: Throttle and brake control systems for automatic vehicle following. IVHS J. **1**(4), 345–377 (1994)
12. Klančar, G., Kristan, M., Kovačič, S., Orqueda, O.: Robust and efficient vision system for group of cooperating mobile robots with application to soccer robots. ISA Trans. **43**, 329–342 (2004)
13. Klančar, G., Škrjanc, I.: Tracking-error model-based predictive control for mobile robots in real time. Robot. Auton. Syst. **55**(6), 460–469 (2007)
14. Kolmanovsky, I., McClamroch, N.H.: Developments in Nonholonomic Control Problems. IEEE Control Syst. **15**(6), 20–36 (1995)
15. Laumond, J.P.: Robot motion planning and control. In: Lecture Notes in Control and Information Science, Vol. 229. Springer-Verlag, New York (1998)
16. Lee, H., Tomizuka, M.: Adaptive vehicle traction force control for intelligent vehicle highway systems (ivhss). IEEE Trans. Ind. Electron. **50**(1), 37–47 (2003)
17. Lee, M., Jung, M., Kim, J.: Evolutionary programming-based fuzzy logic path planner and follower for mobile robots. In: Proceedings of the IEEE Conference on Evolutionary Computation, ICEC, Vol 1, pp. 139–144. San Diego, CA (2000)
18. Lepetič, M., Klančar, G., Škrjanc, I., Matko, D., Potočnik, B.: Time optimal planning considering acceleration limits. Robot. Auton. Syst. **45**, 199–210 (2003)
19. Luca, A., Oriolo, G.: Modelling and control of nonholonomic mechanical systems. In: Angeles, J., Kecskemethy, A., (eds.) Kinematics and Dynamics of Multi-Body Systems. Springer-Verlag, Wien, Austria (1995)
20. Luca, A., Oriolo, G., Vendittelli, M.: Control of wheeled mobile robots: An experimental overview. In: Nicosia, S., Siciliano, B., Bicchi, A., Valigi, P., (eds.) RAMSETE - Articulated and Mobile Robotics for Services and Technologies. Springer-Verlag, London, UK (2001)
21. Oblak, S., Škrjanc, I.: A comparison of fuzzy and CPWL approximations in the continuous-time nonlinear model-predictive control of time-delayed Wiener-type systems. J. Intell. Robot. Syst. **47**(2), 125–137 (2006)
22. Oriolo, G., Luca, A., Vandittelli, M.: WMR control via dynamic feed-back linearization: design, implementation, and experimental validation. IEEE Trans. Control Syst. Technol. **10**(6), 835–852 (2002)
23. Reynolds, C.W.: Flocks, herds, and schools: A distributed behavioral model. Comput. Graph. **21**(4), 25–34 (1987)

24. Sarkar, N., Yun, X., Kumar, V.: Control of mechanical systems with rolling constraints: application to dynamic control of mobile robot. Int. J. Rob. Res. **13**(1), 55–69 (1994)
25. Sheikholeslam, S., Desoer, C.A.: Longitudinal control of a platoon of vehicles with no communication of lead vehicle information: a system level study. IEEE Trans. Veh. Technol. **42**(4), 546–554 (1993)
26. Yi, S.-Y, Chong K.-T.: Impedance control for a vehicle platoon system. Mechatronics. **15**(5), 627–638 (2005)
27. http://msc.fe.uni-lj.si/PublicWWW/Klancar/RobotsPlatoon.html