International Journal of Robotics and Automation (IJRA)
Vol. 7, No. 1, March 2018, pp. 18~38
ISSN: 2089-4856, DOI: 10.11591/ijra.v7i1.pp18-38

Investigation of Formation Control Approaches Considering the Ability of a Mobile Robot

Hannes Wind[1], Oliver Sawodny[2], Thomas Bräunl[3]
[1,2]Institute for System Dynamics, University of Stuttgart, Stuttgart, Germany
[3]Department of Electrical, Electronic and Computer Engineering, University of Western Australia, Crawley, Australia

Article Info

Article history:

Received Sept 11, 2017
Revised Nov 22, 2017
Accepted Dec 15, 2017

Keyword:

Controller
Formation control
Implementation of formation
Realistic model of mobile robot

ABSTRACT

This work investigates and compares various formation control approaches for mobile robots. A comprehensive literature review was conducted, with particular focus on the approaches' applicability to be implemented on real mobile robots with limited hard and software capabilities. A realistic model of mobile robots is introduced and its parameters are identifed with measurements from actual mo-bile robots. Later on, the model is extended and used within simulation studies of the various investigated approaches. A collision avoidance controller based on a formation controller is proposed and simulations are carried out. Experiments on real mobile robots are conducted for two formation controllers and for the pro-posed collision avoidance controller. It is shown that if the requirements resulting from the simulation studies are satisfied, an implementation on the real robots is possible

Corresponding Author:

Hannes Wind,
Institute for System Dynamics,
University of Stuttgart,
Keplerstraße 7, 70174, Stuttgart, Germany.
Email: hannes.wind@isys.uni-stuttgart.de

1. INTRODUCTION

There are many advantages of controlling multiple mobile robots. For achieving a task, which might require one sophisticated and expensive robot, a group of simple robots can be sufficient [1]. Using multiple robots can also contribute to the robustness and the efficiency of the system formed by robots [2], [3]. The applications are widespread and range from automated highway systems [4], [5] to achieving formations with satellites [6].

This paper considers the implementation of formation controllers in a leader/follower manner on given mobile robots. The appearance of one mobile robot can be seen in Figure 1. This robot is driven by two independently actuated motors in differential drive mode. For measuring the position and the orientation of the robot, encoders are attached to each motor and measure the displacement of each wheel. Furthermore, position sensitive devices (PSD) are installed which measure the distance to an object. These PSDs are mounted in front and on the side of the robot. The hardware in terms of computational ability is divided into two parts. The high level is a Raspberry Pi board which is designed to run the application program and the low level is an Arduino which request data of the sensors and drive the actuators.

The remainder of this paper is organized as follows: in section 2, a literature review about formation control is given. Especially, the different tasks to achieve with mobile robots are emphasized. Section 3 deals with models of a mobile robot. Firstly, a kinematic model is presented. Secondly, a complex dynamic model is introduced. This model is reduced afterwards and the parameters are identified. Finally, a speed controller is designed based on the identified model. In Section 4, two formation controllers based on a leader/follower approach are presented. Furthermore, a collision avoidance controller is proposed based on a leader/follower

approach. This Section is followed by simulation investigations in Section 5. Therefore, considered models for these simulation investigations are introduced. Based on these models, simulations for each of the three presented formation controller of Section 4 are performed. The experimental results of each considered formation controller are shown in Section 6. Section 7 gives a conclusion as well as an outlook on further work.

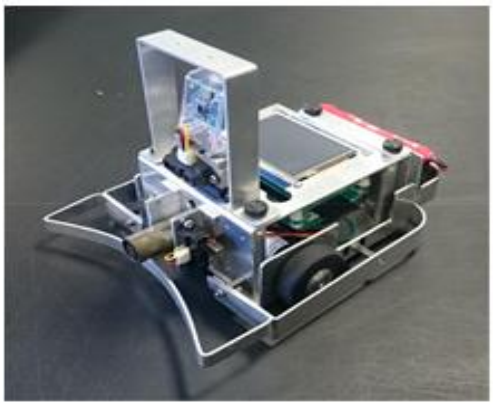

Figure 1. Appearance of the mobile robot

2. LITERATURE REVIEW

There are several control tasks to achieve with mobile robots. The most basic task is to follow a given trajectory [7], known as trajectory tracking. Another motion task is to achieve group formation. Thereby, the goal is to change the position of each robot in order to generate a group formation [8]. This formation control could be considered as an extension to the trajectory tracking problem where robots follow a given trajectory while performing a formation.

The trajectory tracking problem was solved in [9] by using a time-varying control law as a feedback. Another way to solve this problem is through the use of dynamic feedback linearization [10], [11] and [12]. In [13] and [14] the technique of backstepping is used for trajectory tracking of nonholonomic systems. The approach in [15] takes advantage of the cascade structure of the robot model.

The group formation control is considered as a control law which controls the mobile robots in a certain formation. In [16], the authors proposed an approach to achieve this goal by a time-varying feedback control. Zhiyun et al. give necessary and sufficient graphical conditions for the formation stabilization to a point and to more general geometric pattern [17].

There are various approaches to formation tracking, like the behavior-based method [18], the virtual structure strategy [19] and the leader-follower approach [20], [21] and [22]. The behavior-based approach defines a desired behavior for each robot. In order to derive the control law, the relative importance of each behavior is weighted [23]. This approach is used, for example, to apply the social characteristics of insects and animals to multirobot systems [24]. The virtual structure method treats the entire formation as a single entity. To derive the control law for each robot, the motion of the virtual structure is considered and transformed into the motion of the robot [25] and [26].

In this paper, we focus on the leader-follower approach. To accomplish the leader-follower strategy there are several approaches. Lei and Antsaklis made use of the cascade approach as in [15] to achieve a consensus-based controller [27]. Desai et al. applied feedback linearization to exponentially stabilize the distance and the orientation between the leader and the follower [20]. In addition to these approaches, model predictive control (MPC) has become an accepted method in solving the formation control problem. In [28], a nonlinear MPC is applied. Since the computational e ort is higher than applying a linear MPC, Kamel and Youmin use a linear MPC in combination with an input-output feedback linearization to achieve a leader-follower formation control [29].

Related works, where an implementation of a formation controller takes place are [30], [31], [32] and [33]. In these papers simulation results of the formation controller are presented. However, none of them considers a dynamic model, uncertainties of the sensors or the communication structure. Therefore, this work takes an extended model of the robot into account and performs further simulations. In Figure 2 is the considered robot with the relevant parameters.

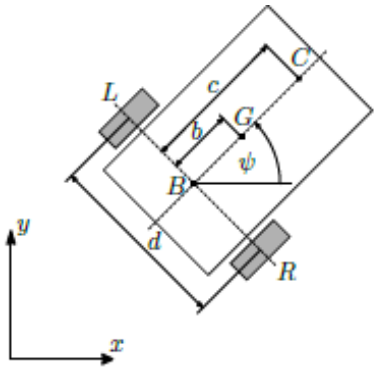

Figure 2. This Figure shows the considered robot with the relevant parameters

3. MODEL SETUP

In this section, a kinematic and a dynamic model is introduced. The kinematic model describes the motion of the mobile robot with the velocity and angular velocity as an input. This model is used for the controller design. Opposed to the kinematic model, the dynamic model describes the motion of the mobile robot based on the acting forces. The dynamic model is too detailed to use for the controller design but it can be used to validate the designed controller.

3.1. Kinematic Model

In this work, differentially driven wheeled mobile robots are considered. Each robot has two wheels that can be actuated independently. The description of a robot in a two-dimensional space can be done by the current position (x; y) and the orientation angle ϕ. Hence, the state vector results to $\boldsymbol{q}(t) = [x(t), y(t), \phi(t)]^T$. The kinematic equations of the robot are represented in Equation (1),

$$\begin{aligned} \dot{x} &= v\cos(\phi) \\ \dot{y} &= v\sin(\phi) \\ \dot{\phi} &= \omega \\ x(0) = x_0, y(0) &= y_0, \phi(0) = \phi_0, \end{aligned} \tag{1}$$

with the velocity v and the angular velocity ω as an input $u = [v, \omega]^T$. The connectedness of the velocity and angular velocity of the robot to the angular velocity of both wheels $(\omega_{W,R}, \omega_{W,L})$ is given by

$$\begin{aligned} v &= r\frac{\omega_{W,R} + \omega_{W,L}}{2} \\ \omega &= r\frac{\omega_{W,R} - \omega_{W,L}}{d} \end{aligned} \tag{2}$$

with r being the radius of a wheel.

3.2. Dynamic Model

3.2.1. Unreduced Model

The geometric parameters of a mobile robot are shown in Figure 2 where *B* is the wheel baseline center, *G* is the center of gravity, *C* is the position of the castor, *L* is the position of the left wheel and *R* is the position of the right wheel. The geometric distances are expressed by b, c and d.

The dynamic model of a robot is derived in [34]. The full dynamic equations of [34] are given in Equation (3), where *m* is the mass of the robot. The uncertainty vector $\boldsymbol{\delta} = [\delta_x, \delta_y, 0, \delta_v, \delta_\omega]^T$ includes the slip speed of the wheels, the viscous friction forces and the resistance force of the castor. The input of the model is given by $^{\tau_v}$dan$^{\tau_\omega}$.

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\phi}\\ \dot{v}\\ \dot{\omega}\end{bmatrix} = \begin{bmatrix} v\cos(\phi)\\ v\sin(\phi)\\ \omega\\ \frac{mbr^2}{\theta_v}\omega^2\\ -\frac{2mbr^2}{\theta_\omega}v\omega\end{bmatrix} + \begin{bmatrix}0 & 0\\ 0 & 0\\ 0 & 0\\ \frac{2r}{\theta_v} & 0\\ 0 & \frac{2rd}{\theta_\omega}\end{bmatrix}\begin{bmatrix}\tau_v\\ \tau_\omega\end{bmatrix} + \begin{bmatrix}\delta_x\\ \delta_y\\ 0\\ \delta_v\\ \delta_\omega\end{bmatrix},$$
$$x(0) = x_0, y(0) = y_0, \phi(0) = \phi_0, v(0) = v_0, \omega(0) = \omega_0. \quad (3)$$

with

$$\theta_v = mR_t + 2I_e, \quad \theta_\omega = I_e d^2 + 2R_t^2(I_z + mb^2), \quad \tau_v = \frac{1}{2}(k_t i_L + k_t i_R), \quad \tau_\omega = \frac{1}{2}(k_t i_L - k_t i_R) \quad (4)$$

Equation (3) and Equation (4) according to [34], where R_t is the nominal radius of the tire, I_e is the moment of inertia of the combined motor rotor and wheel, I_z is the moment of inertia of the robot about the vertical axis, k_t is the motor torque constant and i_L and i_R are the motor currents of the left and the right motors.

3.2.2. Reduced Model

The model in (3) has an uncertainty vector which is difficult to identify. This motivates some simplifications to get a reduced dynamic model. As a first simplification, the uncertainty vector $\boldsymbol{\delta}$ in Equation (3) will be neglected. This can be done by the assumption of no slipping wheels, no external disturbance and by neglecting the friction of the castor.

Furthermore, the input signal of the motor is the duty cycle of the pulse-width modulation (PWM) signal. Since the model in Equation (3) is using the current as an input, it has to be changed. For simplicity, the voltage of the motor is chosen as an input. The relation between the voltage and the current of a DC motor can be described as shown in Equation (5)

$$u = R_a i + L_a \dot{i} + k_v \omega \quad (5)$$

where R_a is armature resistance, L_a is the armature inductance, k_v is the motor velocity constant and u and $\dot{i}$ are the armature voltage and the armature current respectively. Since Equation (5) is a differential equation, it would add two further states to the dynamic model. This can be avoided if the dynamics of the current in Equation (5) are significantly faster than the fastest dynamic in Equation (3). This assumption is satisfied and the dynamics of the current can be neglected. This yields the following Equation (6) and Equation (7)

$$u = R_a i + k_v \omega \quad (6)$$

with these simplifications, the reduced dynamic model results to

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\phi}\\ \dot{v}\\ \dot{\omega}\end{bmatrix} = \begin{bmatrix} v\cos(\phi)\\ v\sin(\phi)\\ \omega\\ \frac{mbr^2}{\theta_v}\omega^2 - \frac{2k_t k_v}{\theta_v R_a}v\\ -\frac{2mbr^2}{\theta_\omega}v\omega - \frac{d^2 k_t k_v}{\theta_\omega R_a}\omega\end{bmatrix} + \begin{bmatrix}0 & 0\\ 0 & 0\\ 0 & 0\\ \frac{rk_t}{\theta_v R_a} & \frac{rk_t}{\theta_v R_a}\\ -\frac{rdk_t}{\theta_\omega R_a} & \frac{rdk_t}{\theta_\omega R_a}\end{bmatrix}\begin{bmatrix}u_L\\ u_R\end{bmatrix}$$
$$x(0) = x_0, y(0) = y_0, \phi(0) = \phi_0, v(0) = v_0, \omega(0) = \omega_0. \quad (7)$$

3.2.3. Model Identification

The goal of the model identification is to determine the unknown parameters in Equation (7). Since in Equation (7) parameters only occur on the right hand side of $[\dot{v}, \dot{\omega}]^T$ and these equations do not depend on $[x, y, \omega]^T$, the dynamic model for the identification can be reduced to the states $[v, \omega]^T$. Furthermore, the parameters in Equation (7) are combined to one parameter vector. This yields the following model description for the identification with the parameter vector $\boldsymbol{p} = [p_1, p_2, p_3, p_4, p_5, p_6]^T$. In order to identify the model, the velocity and the angular velocity needs to be measured. The measurement and the simulation with the identified parameters can be seen in Figure 3.

$$\begin{bmatrix}\dot{v}\\ \dot{\omega}\end{bmatrix} = \begin{bmatrix}p_1\omega^2 + p_2 v\\ p_4 v\omega + p_5\omega\end{bmatrix} + \begin{bmatrix}p_3 & p_3\\ -p_6 & p_6\end{bmatrix}\begin{bmatrix}u_\mathrm{L}\\ u_\mathrm{R}\end{bmatrix}$$
$$v(0) = v_0, \omega(0) = \omega_0 \tag{8}$$

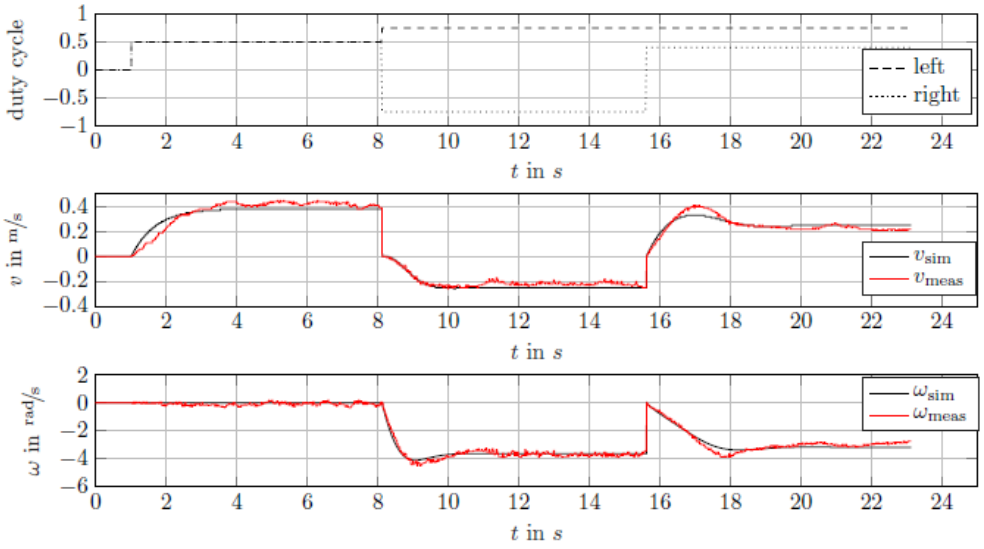

Figure 3. Simulation of the model with the identified parameters and the measurement

For the identification, there are three experiments considered. The input signal is chosen as a step function, whereby each experiment has different nal values. These experiments are chosen as typical movements of the robot. The first experiment is a straight drive, the second experiment is driving a turn and the last experiment is driving a curve. In order to obtain the parameters, the model according to Equation (8) is simulated with a parameter vector $\boldsymbol{p}_0$. The velocity and the angular velocity from the measurements are calculated in parallel. The output of the model $\boldsymbol{y}_\mathrm{sim} = [v_\mathrm{sim}, \omega_\mathrm{sim}]^T$ and the output of the measurements $\boldsymbol{y}_\mathrm{meas} = [v_\mathrm{meas}, \omega_\mathrm{meas}]^T$ are considered in a cost function.

$$J(\boldsymbol{p}) = \frac{1}{2}\sum_{i=1}^{M}((v_\mathrm{meas}(t_i) - v_\mathrm{sim}(t_i, \boldsymbol{p}))^2 + \quad (\omega_\mathrm{meas}(t_i) - \omega_\mathrm{sim}(t_i, \boldsymbol{p}))^2) \tag{9}$$

Thereby *M* is the quantity of measured points. To minimize this cost function, a nonlinear optimizer using the simplex search method is used [35].

3.2.4. Speed Controller

The formation controller, which is considered in this work, uses the kinematic model in Equation (1) to obtain the control law for each robot. Since the input of the real robot is the duty cycle of the PWM signal of each wheel, it is not guaranteed that the current speed of each wheel reaches the desired speed. Therefore, it is necessary to implement a speed controller for each wheel. For this purpose, a simple PID controller is considered. The input of the controller is the error between the desired speed and the current speed of each wheel. Equation (2) can be rearranged to obtain the desired speed from the velocity and angular velocity of the robot. The transfer function of the continuous-time PID controller is given by

$$G_\mathrm{C}(s) = K_\mathrm{P} + K_\mathrm{I}\frac{1}{s} + K_\mathrm{D}\frac{N_\mathrm{f} s}{s + N_\mathrm{f}} \tag{10}$$

with the control parameter K_P, K_I and K_D and the filter coefficient N_f . The parameters for the PID controller are obtained via a nonlinear optimization. The goal of this optimization is to minimize the integrated squared

error $e = v_{\text{des}} - v_{\text{sim}}$ between the desired speed and the simulated speed. Therefore, the reduced dynamic model Equation (7) with the identified parameter is used. Since the input u of the motor is constrained, this constraint has to be considered in the optimization. To avoid slipping wheels, the acceleration of the robot is constrained as well as the angular acceleration. The constrained optimization problem is given by

$$\begin{aligned} &\min_{K_P,K_I,K_D} \int_0^T e^2 dt \\ \text{subject to } &|u| - u_{\max} \leq 0 \\ &|\dot{v}| - a_{\max} \leq 0 \\ &|\dot{\omega}| - \alpha_{\max} \leq 0 \end{aligned} \tag{11}$$

with the maximum voltage of the motor u_{max}, the maximum acceleration a_{max} and the maximum angular acceleration a_{max}. For the implementation of the PID controller, the continuous-time controller has to be transformed into a discrete controller. This is done by means of zero-order hold discretization. Robot 1 is the leader and robot 2 is the follower. The goal is to maintain the relative distance and the orientation between the leader and the follower. Figure 4 is illustration of the leader follower constellation.

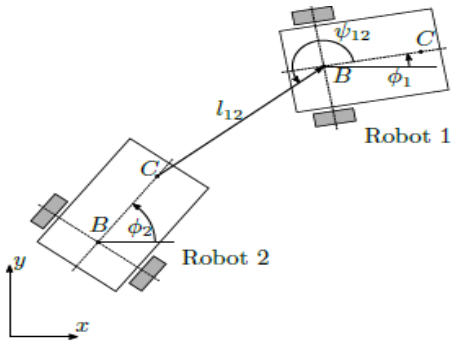

Figure 4. Illustration of the leader follower constellation of the l controller according to [20]

4. FORMATION CONTROL APPROACHES

In this section, two given formation controllers are outlined and the control laws are presented. Furthermore, a new approach to avoid a collision based on a formation controller is proposed.

4.1. $l-\psi$ Controller

The $l-\psi$ controller is proposed in [11]. This controller uses the leader-follower approach, whereby the relative distance and the orientation between the leader and the follower are controlled. This approach uses the techniques of input/output linearization to achieve the formation controller. Figure 4 illustrates the leader and the follower robot with the controlled values l_{12} dan ψ_{12}. The aim of the $l-\psi$ controller is to maintain the desired length l_{12}^{d} and the desired relative angle ψ_{12}^{d} while the leader is following a given trajectory. Thereby, the state of the follower is defined as $[l_{12}, \psi_{12}, \phi_2]^T$.

The kinematic relationships of Robot 1 shown in Figure 4 are given by Equation (1). The kinematic relationships of Robot 2 are given by

$$\begin{aligned} \dot{l}_{12} &= v_2 \cos(\gamma_1) - v_1 \cos(\psi_{12}) + c\omega_2 \sin(\gamma_1) \\ \dot{\psi}_{12} &= \frac{1}{l_{12}} \left(v_1 \sin(\psi_{12}) - v_2 \sin(\gamma_1) + c\omega_2 \cos(\gamma_1) - l_{12}\omega_1 \right) \\ \dot{\phi}_2 &= \omega_2, \end{aligned} \tag{12}$$

according to [20], with $\gamma_1 = \phi_1 + \psi_{12} - \phi_2$. It is required that $l_{12} > c$ in order to avoid a collision. The use of input-output linearization yields the following control law for the follower:

$$\omega_2 = \frac{\cos(\gamma_1)}{c}\{(\alpha_2 l_{12}\left(\psi_{12}^{\mathrm{d}} - \psi_{12}\right) - v_1 \sin(\psi_{12}) + l_{12}\omega_1 + \rho_{12}\sin(\gamma_1))\}$$
$$v_2 = \rho_{12} - c\omega_2 \tan(\gamma_1) \tag{13}$$

where

$$\rho_{12} = \frac{\alpha_1(l_{12}^{\mathrm{d}} - l_{12}) + v_1\cos(\psi_{12})}{\cos(\gamma_1)} \tag{14}$$

with these inputs, the $l-\psi$ variables become

$$\dot{l}_{12} = \alpha_1(l_{12}^{\mathrm{d}} - l_{12})$$
$$\dot{\psi}_{12} = \alpha_2(\psi_{12}^{\mathrm{d}} - \psi_{12}) \tag{15}$$

whereby Equation (13), Equation (14) and Equation (15) according to [20]. In order to calculate the control law in Equation. (13), the length $\boldsymbol{l_{12}}$ and the angle ψ_{12} are required. These can be calculated as

$${}^{I}\boldsymbol{l}_{12} = {}^{I}\boldsymbol{p}_1 - \left({}^{I}\boldsymbol{p}_2 + {}^{I}\boldsymbol{R}_{\mathrm{V}_2}\cdot\begin{bmatrix}c\\0\end{bmatrix}\right)$$
$$l_{12} = \|{}^{I}\boldsymbol{l}_{12}\|$$
$$\psi_{12} = \arccos\left(\frac{({}^{I}\boldsymbol{R}_{\mathrm{V}_1}\boldsymbol{e})^{T}\cdot(-{}^{I}\boldsymbol{l}_{12})}{\|{}^{I}\boldsymbol{R}_{\mathrm{V}_1}\boldsymbol{e}\|\cdot\|{}^{I}\boldsymbol{l}_{12}\|}\right) \tag{16}$$

with

$${}^{I}\boldsymbol{R}_{V_i} = \begin{bmatrix}\cos(\phi_i) & -\sin(\phi_i)\\ \sin(\phi_i) & \cos(\phi_i)\end{bmatrix}, \qquad i = 1,2, \qquad \boldsymbol{e} = \begin{bmatrix}1\\0\end{bmatrix} \tag{17}$$

where ${}^{I}\boldsymbol{R}\boldsymbol{v}_i$ is a transformation matrix which transforms a vector from the coordinate system $\boldsymbol{V_i}$ into the inertial coordinate system $\boldsymbol{I}$ and $\boldsymbol{Ip_i}$ is the position vector of the robot i referred to the inertial coordinate system. This vector contains not the full state of the robot i, but only the position (x_i, y_i).

To be able to calculate the control law in the follower robot, the whole state of the leader q1, the whole state of the follower $\boldsymbol{q_2}$ and the input signals of the leader (v_1, ω_1) are needed as well as the angular velocity of the follower ω_2. From this, the communication between the leader and the follower can be determined.

4.2. $l-l$ Controller

The $l-l$ controller is proposed in [20]. This controller also uses the leader-follower approach with the difference of two leaders and one follower. The aim of the $l-l$ controller is to control the distance between the follower and the two leaders.

Figure 5 illustrates the two leaders (Robot 1 and Robot 2) and the follower (Robot 3). The controlled values are the distances between the leaders and the follower l_{13} and l_{23}. The aim of this controller is to maintain the desired lengths l_{13}^{d} dan l_{23}^{d} while the leaders follow a given trajectory. Thus, the state of the follower is given as $[l_{13}, l_{23}, \phi_3]^T$. The requirements for the $l-l$ controller concern the lengths l_{13} and l_{23}. It is required that those lengths are greater than the distance c and the follower must not lie on a line connecting the two leaders [20].

The kinematic relationships for the leader robots shown in Figure 5 are given by Equation (1). The kinematic relationships of the follower are given by

$$\begin{aligned}\dot{l}_{13} &= v_3\cos(\gamma_1) - v_1\cos(\psi_{13}) + c\omega_3\sin(\gamma_1)\\ \dot{l}_{23} &= v_3\cos(\gamma_2) - v_2\cos(\psi_{23}) + c\omega_3\sin(\gamma_2)\\ \dot{\phi}_3 &= \omega_3\end{aligned} \tag{18}$$

according to [20], with $\gamma_i = \phi_i + \psi_{i3} - \phi_3, i = 1,2$.

The use of input-output linearization yields the following control law for the follower

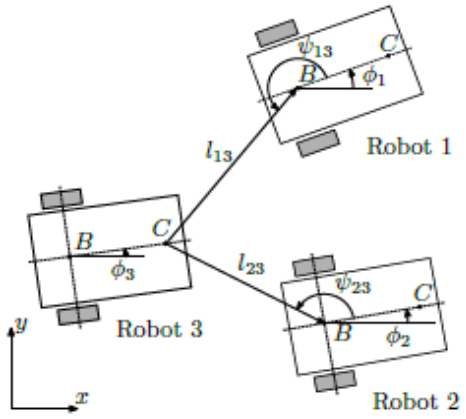

Figure 5. Illustration of the leader follower constellation of the $l-l$ controller according to [20]

$$\omega_3 = \frac{1}{c\sin(\gamma_1-\gamma_2)}\{\alpha_1(l_{13}^{\mathrm{d}} - l_{13})\cos(\gamma_2) + v_1\cos(\psi_{13})\cos(\gamma_2) - \alpha_2(l_{23}^{\mathrm{d}} - l_{23})\cos(\gamma_1) - v_2\cos(\psi_{23})\cos(\gamma_1)\}$$

$$v_3 = \frac{\alpha_1(l_{13}^{\mathrm{d}} - l_{13}) + v_1\cos(\psi_{13}) - c\omega_3\sin(\gamma_1)}{\cos(\gamma_1)} \tag{19}$$

with these inputs the $l-l$ variables become

$$\begin{aligned}\dot{l}_{13} &= \alpha_1(l_{13}^{\mathrm{d}} - l_{13})\\ \dot{l}_{23} &= \alpha_2(l_{23}^{\mathrm{d}} - l_{23})\end{aligned} \tag{20}$$

whereby Equation (19) and Equation (20) defined according to [20]. The calculation of the current length l_{13}^{d} and l_{23}^{d} follows Equation (16). To calculate the control law in the follower robot, the whole state of the two leaders $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$, the whole state of the follower $\boldsymbol{q_3}$ and the velocity of the leaders v_1 and v_2 are required as well as the angular velocity of the follower ω_3. Considering this information, the communication can be determined.

4.3. Collision Avoidance Controller

Following, an extension of the $l-\psi$ controller from Section 4.1 is proposed. The goal of this extension concerns the collision avoidance based on local sensors. First of all, static obstacles should be recognized by the local PSD sensors. Using this measurement, a defined safety distance to a static obstacle should be maintained. Furthermore, a collision with other robots in the formation needs to be avoided.

For the collision avoidance controller, a formation with one leader and $\boldsymbol{N}$ followers is considered. This formation is controlled by a $l-\psi$ controller. To be able to avoid a collision, the desired control values l_{1i}^{d} and ψ_{1i}^{d} $(i = 1,2,\ldots,N)$ are changed in an appropriated manner. This implies that the desired values are time dependent. Therefore, the following algorithm for changing the desired values and hence the formation is proposed

$$\begin{aligned} \dot{l}_{1i}^{\mathrm{d}} &= k_1 u_1 \\ \dot{\psi}_{1i}^{\mathrm{d}} &= k_2 u_2 \\ l_{1i}^{\mathrm{d}}(0) = l_{1i0}^{\mathrm{d}}, &\psi_{1i}^{\mathrm{d}}(0) = \psi_{1i0}^{\mathrm{d}}, \end{aligned} \tag{21}$$

with

$$u_1 = \begin{cases} 1, & \text{if}(PSD_{\mathrm{L}} < \epsilon_{\mathrm{L}} \text{ and } PSD_{\mathrm{R}} < \epsilon_{\mathrm{R}}) \text{ or } (PSD_{\mathrm{F}} < \epsilon_{\mathrm{F}}) \text{ or} \\ & (PSD_{\mathrm{L}} > \epsilon_{\mathrm{L}} \text{ and } PSD_{\mathrm{R}} > \epsilon_{\mathrm{R}} \text{ and } PSD_{\mathrm{F}} > \epsilon_{\mathrm{F}} \text{ and } l_{1i}^{\mathrm{d}} < l_{1i0}^{\mathrm{d}}) \\ -1, & \text{if}(PSD_{\mathrm{L}} > \epsilon_{\mathrm{L}} \text{ or } PSD_{\mathrm{R}} > \epsilon_{\mathrm{R}}) \text{ and } PSD_{\mathrm{F}} > \epsilon_{\mathrm{F}} \text{ and } l_{1i}^{\mathrm{d}} > l_{1i0}^{\mathrm{d}} \\ 0, & \text{else,} \end{cases}$$

$$u_2 = \begin{cases} 1, & \text{if}(PSD_{\mathrm{L}} < \epsilon_{\mathrm{L}} \text{ and } PSD_{\mathrm{R}} > \epsilon_{\mathrm{R}}) \text{ or} \\ & (PSD_{\mathrm{L}} > \epsilon_{\mathrm{L}} \text{ and } PSD_{\mathrm{R}} > \epsilon_{\mathrm{R}} \text{ and } \psi_{1i}^{\mathrm{d}} < \psi_{1i0}^{\mathrm{d}}) \\ -1, & \text{if}(PSD_{\mathrm{L}} > \epsilon_{\mathrm{L}} \text{ and } PSD_{\mathrm{R}} < \epsilon_{\mathrm{R}}) \text{ or} \\ & (PSD_{\mathrm{L}} > \epsilon_{\mathrm{L}} \text{ and } PSD_{\mathrm{R}} > \epsilon_{\mathrm{R}} \text{ and } \psi_{1i}^{\mathrm{d}} > l_{1i0}^{\mathrm{d}}) \\ 0, & \text{else,} \end{cases} \tag{22}$$

where subscript *L*, *R* and *F* refer to the left side, the right side and the front, respectively. The symbol PSD denotes the value of the PSD sensor and ϵ specifies the threshold. The value of the measurement from the PSD sensors is related to the distance, i.e., if the value is beyond a threshold, no threat concerning a collision exists.

5. SIMULATIVE INVESTIGATIONS AND RESULTS

In this section, the three formation controllers from Section 4 are investigated for the purpose of a possible implementation on the given robots. Specifically, several simulations of each formation controller are carried out with different models of the robot. These models are changed in an appropriated manner to achieve a more detailed model of the real robots. According to the results of these simulations, a conclusion is given regarding the possibility of implementing a formation controller on the robots.

5.1. Considered Simulation Models

Before the simulations take place, the different models for the simulation and different simulation environments are introduced. The models are changed from an easy kinematic model to a very detailed model which also considers communication. With these different models it is possible to gure out the crucial parts of the whole system concerning an implementation of the formation controllers on the real robots. These models are modelled and simulated in Matlab/Simulink. As a further simulation environment, EyeSim is introduced [36], [37].

As already mentioned, the first model will be a kinematic model as introduced in (1). The first improvement on the simulation model concerns the model of the robot. Instead of using the kinematic model of the robot on which the control design is based, a dynamical model is used. Therefore, the model Equation (7) of Section 3.2.2 with the identified parameters is used. Since the velocity v and the angular velocity ω are not controlled in Equation (7), the speed controller of Section 3.2.4 is applied. Therefore, not only the optimized control parameters of the PID controller are used. With the change of these parameters and therefore the change of the performance of the velocity controller, some requirements on the speed controller can be formulated.

To improve the dynamic model, a further characteristic of the robot is included. A crucial aspect from implementing a controller in hardware is the discretization. Due to the characteristic of a microcontroller, it is not possible to achieve a continuous output signal of the controller. The control output will therefore be a discrete signal.

Due to the fact that these formation controllers need communication, the method and impact of communication is investigated. For this work, two types of error causes are considered. The first error results from network delays. There are many causes for these delays e.g. the limited network transfer rate or sending packages through several layers. In summary all these delays can be considered as an additional amount of time in the sampling time. The second considered error concerns the communication protocol UDP, which is used for transmitting data. This protocol is unreliable and there is no guarantee of delivery. Therefore, the model of the system is extended to cover the occurrence of package loss in the network. The event of a

package loss is assumed to be uniformly distributed and if one package is lost, the whole calculation of the controller is skipped in this sample.

EyeSim is used as a second environment for simulating the formation controller [36], [37]. This program is a simulation environment for multiple mobile robots which provides the possibility to simulate different kind of robots [36], [37]. One of those robots is similar to the robots this paper takes as a basis. An advantage of EyeSim is the application programmer interface (API) which is the same as the API of the real robot. This means the whole functionality of the high level is provided in EyeSim. The relevant functionality for this purpose is limited to manipulating the speed of the robot, the communication between robots and the retrieving of the sensor data. The use of EyeSim is also a good opportunity to check the control law in combination with the high level functionality. Figure 6 is visualization of the $l-\psi$ controller with a kinematic model. The Robot 1 is denoted as +, Robot 2 as o and Robot 3 is denoted as *.

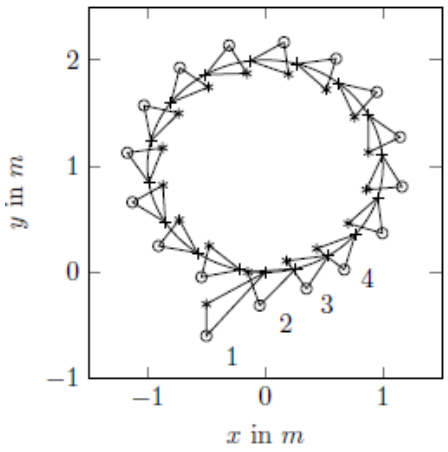

Figure 6. Visualization of the $l-\psi$ controller with a kinematic model

5.2. Simulations of the $l-\psi$ Controller

In this subsection, several simulations firstly with Matlab/Simulink and secondly with EyeSim are carried out. Finally, the applicability to implement this $l-\psi$ controller is discussed.

5.2.1. Simulation in Matlab/Simulink

The first simulation of the $l-\psi$ controller is done with the kinematic model to investigate its behavior. Therefore, a constellation with one leader and two followers is considered. The leader drives a curve with the input values v_r = 0.1 and $\omega_r = 0.1$. The formation specification for the first follower is defined as $l_{12}^d = 0.3, \psi_{12}^d = 210°$ and for the second follower the desired control values are $l_{13}^d = 0.3, \psi_{13}^d = 150°$. The initial pose for each robot is the control parameters are chosen as $\alpha_1 = 0.5$ and $\alpha_2 = 0.5$ as shown in Equation (23).

$$\boldsymbol{q}_1(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{q}_2(0) = \begin{bmatrix} -0.5 \\ -0.6 \\ 0 \end{bmatrix}, \boldsymbol{q}_3(0) = \begin{bmatrix} -0.5 \\ -0.3 \\ 0 \end{bmatrix} \tag{23}$$

The results of the simulation with the kinematic model can be seen in Figure 6. After an initial convergence of the two followers it can be observed that the desired length and the desired angle are maintained. The follower robot follows the leader with the specified parameters.

The second simulation is carried out with a dynamic model as mentioned in 5.1. There are two crucial outcomes of these simulations. Firstly, the speed controller plays an important role in the whole system. If the velocity v and the angular velocity ω of each robot do not converge to the required value, the behavior of the dynamic model is different from the kinematic model. In addition to this, the length $\boldsymbol{l_{1i}}$ and the angle ψ_{1i} (where i is the number of the follower) do not converge to the desired values, if the leader robot converges to the kinematic model and the followers do not converge. To achieve this goal, the integral

part of the PID controller is necessary. Considering the method of determining the control parameters it is ensured that both the velocity and the angular velocity converge.

Secondly, the shape of the reference trajectory combined with the parameters of the speed controller has an influence on the convergence of the formation values, namely the length $\boldsymbol{l_{1i}}$ and the angle ψ_{1i}. If the speed controller is faster, the influence on a change in the reference trajectory regarding the formation values is less than with a slower speed controller. Therefore, a fast speed controller is required. This demand is covered by the proposed method of determining the control parameters. The results of two simulations can be seen in Figure 7.

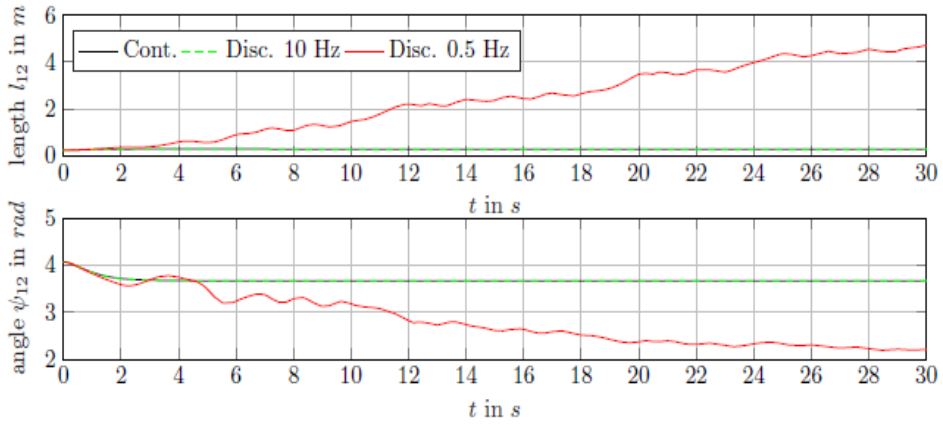

Figure 7. Visualization of the $l-\psi$ controller with a dynamic model and a discrete controller

The next simulation takes the dynamic model with the discrete controller as a basis. Since the network delays can be considered as an increase in the sample time, these delays will be considered here as well. For this purpose, a constellation with one leader and one follower is considered with the desired length $l_{12}^{\mathrm{d}} = 0.3m$ and the desired angle $\psi_{12}^{\mathrm{d}} = 210°$. The upper subplot visualizes the current length and the lower subplot the current angle. In each subplot, three simulations are shown. The first one uses a continuous $l-\psi$ controller whereas the other simulations use a discrete controller, one with a sample frequency of 10 Hz and one with a sample frequency of 0.5 Hz, respectively. It can be observed that the controller with the higher sample frequency is close to the continuous one. A too low sample frequency leads to a non-converging behavior. This characteristic of the discretization has to be considered by the implementation of the control law on the high level.

There are two outcomes of the simulations with the last improvement of the model regarding the package loss. Firstly, the stability of the $l-\psi$ controller depends on the percentage of the package loss. If the package loss is very high (in this case beyond 80%), the stability is not further guaranteed. Nevertheless the system tolerates a huge amount of package loss, which is a positive result regarding the implementation. Secondly, the amount of tolerated package loss depends on the shape of the leader's trajectory. The more changes occur in this trajectory, the fewer lost packages can be tolerated to achieve an adequate performance.

5.2.2. Simulation in EyeSim

The results of the simulation show a similar behavior of the $l-\psi$ controller to that obtained from the Matlab/Simulink simulations. Therefore, no diagram will be shown.

5.2.3. Applicability

Following, a summary of all the investigated situations concerning the implementation of the $l-\psi$ controller on the robots is given. If the speed controller is implemented as suggested in the Subsection 3.2.4, the formation can be achieved. Another point mentioned recurrently is the shape of the reference trajectory. This has to be chosen in an appropriate manner.

Critical open points are the time demand on the robot for one execution of the control algorithm and the package loss in the network. If the suggestions mentioned in the section before are considered, it is possible to implement the $l-\psi$ controller on the given robots.

5.3. Simulations of the $l-l$ Controller

In this subsection, several simulations firstly with Matlab/Simulink and secondly with EyeSim are carried out. Finally, the applicability of the $l-l$ controller is discussed. The Robot 1 is denoted as +, Robot 2 as o and Robot 3 is denoted as *. The results of the simulation can be seen in Figure 8.

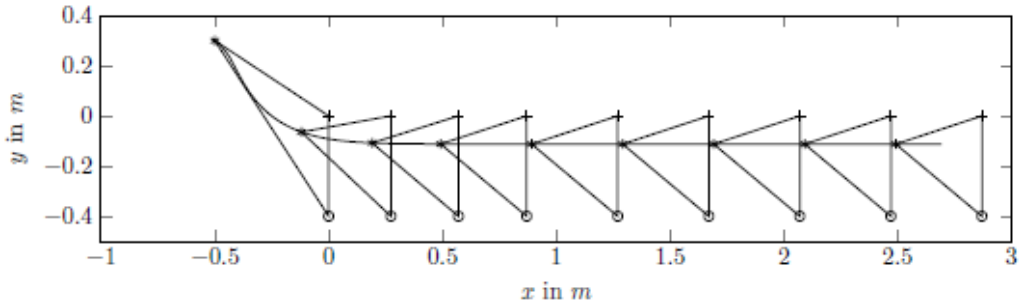

Figure 8. Visualization of the $l-l$ controller with a kinematic model, resulting from a simulation with Matlab/Simulink

5.3.1. Simulation in Matlab/Simulink

The first simulation of the $l-l$ controller is carried out with a kinematic model. Therefore, a simple constellation with two leaders and one follower is considered. This controller can only deal with straight lines. For that reason, the input values of both leaders are chosen as v_i = 0.1 and $\omega_i = 0 \ (i = 1, 2)$. The formation specification for the follower is defined as $l_{13}^{d} = 0.3$ and $l_{23}^{a} = 0.4$. The initial pose of each robot is given by

$$\boldsymbol{q}_1(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{q}_2(0) = \begin{bmatrix} 0 \\ -0.4 \\ 0 \end{bmatrix}, \boldsymbol{q}_3(0) = \begin{bmatrix} -0.5 \\ 0.3 \\ 0 \end{bmatrix} \tag{24}$$

and the control parameters are chosen as $\alpha_1 = 1$ and $\alpha_2 = 1$.

The results of the simulation can be seen in Figure 8. After an initial phase of convergence, the formation is maintained with the defined formation specification. In the further simulations the setup considering the initial pose, the desired values and the leader's trajectory will be the same as in the simulation of the kinematic model

The first simulation beyond the kinematic model is to simulate the $l-l$ controller with a dynamic model. The crucial outcome of these simulations concerns the speed controller. The requirements for the speed controller correspond with the demands in the Subsection 5.2.1. It must have a converging behavior considering the velocity v and the angular velocity ω. Since there is no change in the leaders' reference trajectory, the demands regarding the transient time are not as strict as for the $l-\psi$ controller.

In the next simulation, different sample times are considered. In Figure 9, the results of a simulation with a sample frequency of 10 Hz and 1 Hz can be seen. For comparison reasons, a simulation with a continuous $l-l$ controller is also visualized. Considering the higher sampled controller, it can be observed that the current length is close to the current length of the continuous controller. In fact both are converging to the desired length. As opposed to this, the l l controller with a sample frequency of 1 Hz shows an unstable behavior. Comparing the behavior of a discrete $l-\psi$ controller and a discrete $l-l$ controller, it can be observed that the $l-l$ controller tends to an unstable behavior earlier as the $l-\psi$ controller, when the sampling frequency is decreased. These properties of the $l-l$ controller need to be reconsidered in the implementation of the controller.

The delay in the network communication is considered as an additional discrete delay as in Subsection.5.2.1 Therefore, The Next Investigation Considers The Package Loss In The Network.Several simulations with different percentage of package loss are performed. The crucial outcome concerns the tolerated amount of package loss. The more packages are lost, the longer it takes to converge to the desired formation. This is valid until a certain amount of package loss occurs. Beyond this point, the $l-l$ controller becomes unstable.

5.3.2. Simulation in EyeSim

Since the results of the simulation show a similar behavior of the $l-\psi$ controller to that obtained from the Matlab/Simulink simulations, no diagram will be shown.

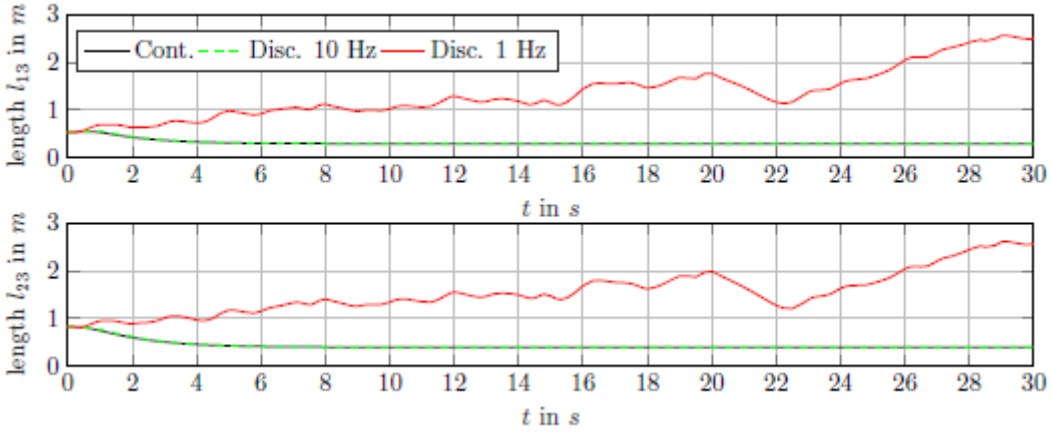

Figure 9. Visualization of the controller with a dynamic model and a discrete controller

5.3.3. Applicability

The results discussed above are reconsidered in order to summarize the investigation of the $l-l$ controller. In comparison with the $l-\psi$ controller, the demands on the speed controller are less restrictive. However, the requirements on the sample time are higher. This is not a disqualifier, since the sample frequency is very low when the instabilities occur. Nevertheless, the execution time of the control algorithm including all other necessary executions must be considered during the implementation. Furthermore, the occurrence of a package loss has to be reconsidered for the implementation. This is covered with the investigation of the $l-\psi$ controller.

5.4. Collision Avoidance Controller

In this subsection, simulations with the collision avoidance controller are carried out. The simulation in Matlab/Simulink is skipped for the collision avoidance controller for two reasons. Firstly, this controller has the $l-\psi$ controller as a basis and is has already been investigated with an extended model in Matlab/Simulink (see Section 5.2.1.). Secondly, the collision avoidance controller uses PSD sensors. EyeSim provides a realistic behavior of the PSD sensors. This also includes the measurement of the distance to another robot considering the appearance of it. For these reasons, only a simulation in EyeSim is performed

5.4.1. Simulation in EyeSim

The first simulation concerns the static obstacle. Therefore, a constellation with one leader and two followers is considered. The formation speciation is defined as $l_{12}^{\mathrm{d}}=0.4, \psi_{12}^{\mathrm{d}}=150^{\circ}$ for the first follower and $l_{13}^{\mathrm{d}}=0.4, \psi_{13}^{\mathrm{d}}=210^{\circ}$ for the second follower. The initial pose is given as

$$\boldsymbol{q}_1(0)=\begin{bmatrix}0\\0\\0\end{bmatrix}, \boldsymbol{q}_2(0)=\begin{bmatrix}-0.5\\0.4\\0\end{bmatrix}, \boldsymbol{q}_3(0)=\begin{bmatrix}-0.5\\0\\0\end{bmatrix} \tag{25}$$

The simulation setup is chosen as follows. The leader performs a straight line with two followers. After a de ned distance, an obstacle occurs for a certain length. This obstacle forces the second follower to change its position. The results of this simulation can be seen in Figure 10. It shows the position of each robot to certain time steps. It can be observed, that the second follower is changing the position to maintain the desired minimum distance to the obstacle. The change appears firstly in the angle until the distance to the first follower is under a threshold. After that, the desired length is increased. If the formation passed the obstacle, it returns to the initial formation. This simulation does not show an avoidance of a collision, since

the obstacle is not a potential collision. But it demonstrates the possibility to maintain a specified distance around each robot based on the PSD sensors

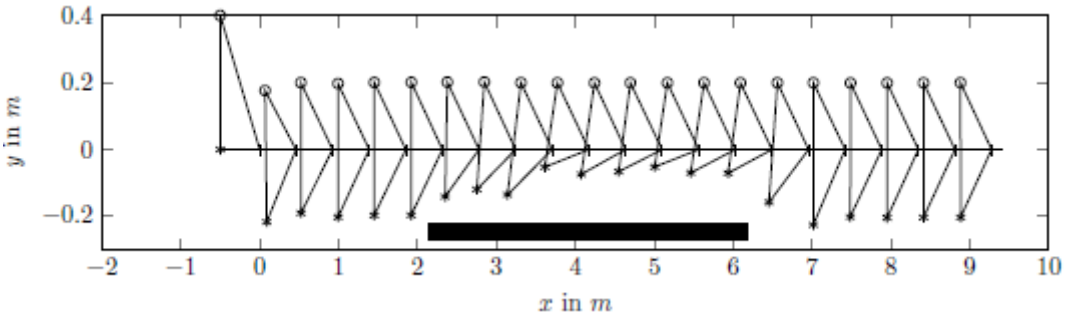

Figure 10. Visualization of the collision avoidance controller with an obstacle on the right, resulting from a simulation with EyeSim

The corresponding current length l_{13} and current angle ψ_{13} of the second follower can be seen in Figure 11. Additional to this, the desired formation parameters l_{13}^{d} and ψ_{13}^{d} are plotted. The situation according to the change in the desired formation parameters can be observed as described above. Furthermore, a converging of the current length l_{13} and current angle ψ_{13} to the desired ones can be seen. The Robot 1 is denoted as +, Robot 2 as o and Robot 3 is denoted as

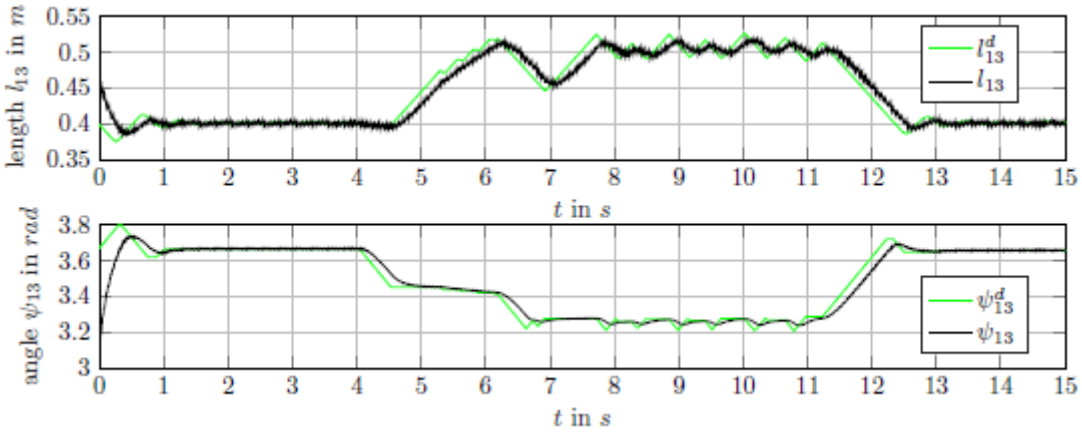

Figure 11. Visualization of the formation specification of the collision avoidance controller with an obstacle on the right, resulting from a simulation with EyeSim

A further simulation considers the collision avoidance with two robots. Therefore, a formation constellation with one leader and two followers is considered. In difference to the simulation before, this simulation uses the same formation parameters for each robot. They are defined as $l_{1i}^{\mathrm{d}} = 0.5$ and $\psi_{1i}^{\mathrm{d}} = 180°$ (i = 1,2) with the initial pose

$$\boldsymbol{q}_1(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{q}_2(0) = \begin{bmatrix} -0.5 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{q}_3(0) = \begin{bmatrix} -0.8 \\ 0 \\ 0 \end{bmatrix} \tag{26}$$

The leader is performing a straight line and the follower robots are trying to follow this line at the same position. Since this is physically not possible, the collision avoidance controller is taken into account. The results of this simulation can be seen in Figure 12 and 13. Once the second follower gets closer to the

first follower, the desired length is increased according to the algorithm in (22). Hence, no collision occurs. This new formation is maintained while the second follower has a new desired length.

5.4.2. Applicability

The simulations show the possibility of maintaining a defined distance on each side of the robot and the ability of avoiding a collision. If the crucial parts in Subsection 4.1 are considered for the $l-\psi$ controller, an implementation with these specifications is possible.

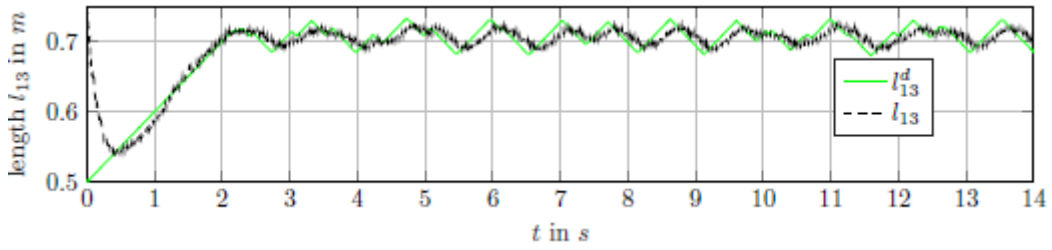

Figure 12. Visualization of the current length and the desired length of the follower two with the collision avoidance controller, resulting from a simulation with EyeSim

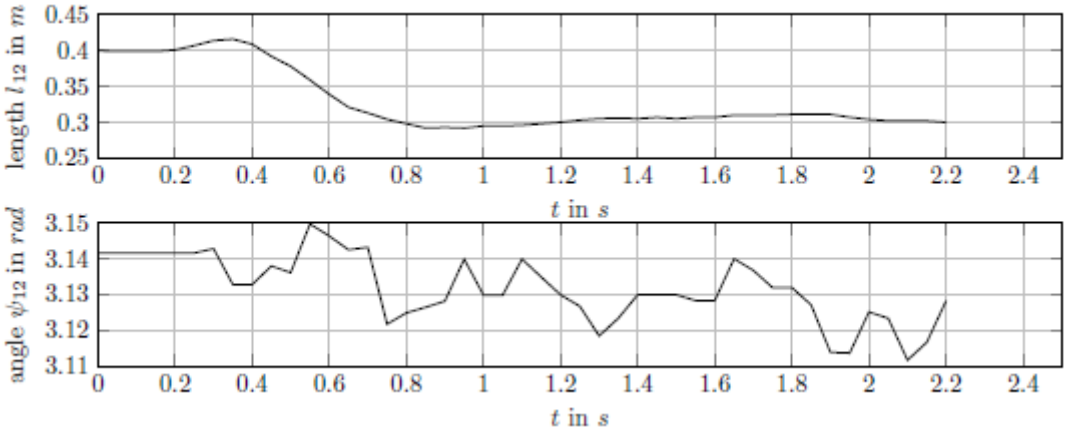

Figure 13. Measured length l_{12} and angle$_{12}$ of the first experiment with the $l-\psi$ controller

6. EXPERIMENTAL RESULTS

In this section, the experimental results are discussed. This is basically done for each controller itself. Since the $l-\psi$ controller was the first implemented controller, these studies are more extended and in some point more general than the others.

6.1. Experiments of the $l-\psi$ Controller

In order to gain a first impression of the connected real system, a simple experiment with one leader and one follower is considered, whereby the leader performs a straight line. The formation specication is dened as $l_{12}^{\mathrm{d}}=0.3$ and $\psi_{12}^{\mathrm{d}}=180°$ with initial pose

$$\boldsymbol{q}_1(0)=\begin{bmatrix}0\\0\\0\end{bmatrix},\boldsymbol{q}_2(0)=\begin{bmatrix}-0.45\\0\\0\end{bmatrix} \tag{27}$$

The results can be seen in Figure 13. It can be observed that the current length is converging to the desired length after a settling time. The increasing length at the beginning of the experiment is caused by the static friction of the robot. Since the leader has a step function in the velocity *v*, the robot can overcome the

static friction faster. Furthermore, the desired velocity of the follower, resulting from the formation controller, is relatively low. Thus, it takes longer to overcome the static friction. For the angle ψ_{12} this behavior cannot be observed, since the initial pose of the leader and the follower already fulfill the desired angle. Nevertheless it can be seen, that the desired angle ψ_{12}^{d} is maintained with a certain range during this experiment.

The second experiment considers one leader and two followers with the following formation specification. For the first follower the desired length is $l_{12}^{\mathrm{d}} = 0.3$ and the desired angle is defined as $\psi_{12}^{\mathrm{d}} = 150°$ whereas the formation specification for the second follower is chosen as $l_{13}^{\mathrm{d}} = 0.3, \psi_{13}^{\mathrm{d}} = 210°$. The initial pose of each robot is defined as shown in Figure 14.

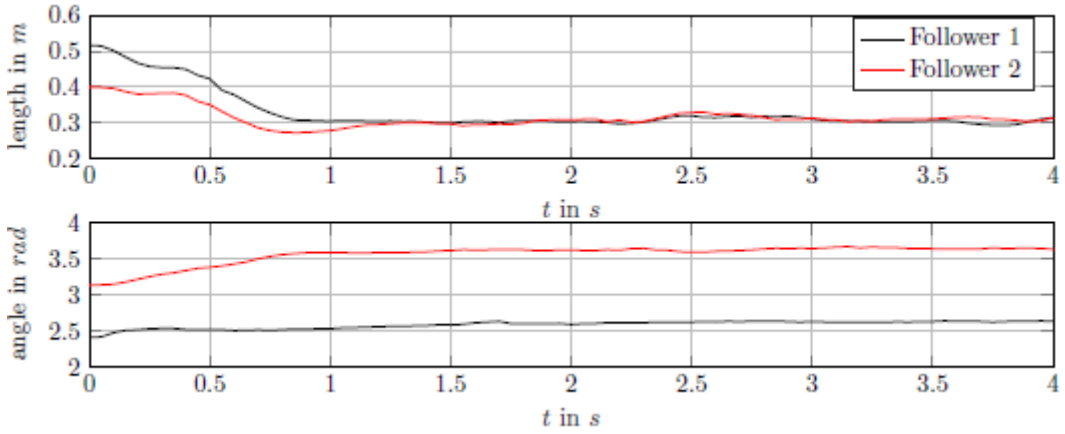

Figure 14. Measured length and angle between the leader and the two followers of the second experiment with the $l - \psi$ controller

In Figure 15, the formation of the three robots can be seen. For reasons of clarity and comprehensibility, this gure only shows the straight line. The convergence behavior as described above is clearly visible. Furthermore, the maintaining of the formation while performing a straight line is recognizable.

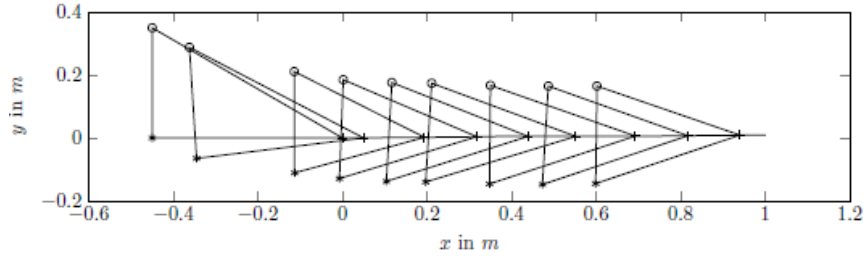

Figure 15. Visualization of the formation of the second experiment with the controller

$$\boldsymbol{q}_1(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{q}_2(0) = \begin{bmatrix} -0.45 \\ 0.35 \\ 0 \end{bmatrix}, \boldsymbol{q}_3(0) = \begin{bmatrix} -0.45 \\ 0 \\ 0 \end{bmatrix} \tag{28}$$

The leader performs a straight line for one meter with a constant velocity. After the straight line, the leader performs a circle with a radius of 0.25 m and an angle of 180°. The current length l_{12} and l_{13} of both follower as well as the current angle ψ_{12} and ψ_{13} are shown in Figure 14. It can be seen that the current length and the current angle are converging to the desired ones. At the time t = 2.3s the current length of both

follower increases until 2.5s. This is the point in time when the leader changes from the straight line to the circle. Thus, the followers need to converge to the desired length again. The difference in the current angle is not as significant as the difference in the current length.

6.2. Experiments of the Controller

The first experiment considers a constellation with two leaders and one follower. Both leaders perform a straight line with the input values $v_i = 0.1$ and $\omega_i = 0$ $(i = 1, 2)$. The formation specification for the follower is chosen as $l_{13}^{\mathrm{d}} = 0.4$ and $l_{23}^{\mathrm{d}} = 0.4$ with the initial pose of each robot as shown in Figure 16.

$$\boldsymbol{q}_1(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{q}_2(0) = \begin{bmatrix} 0 \\ 0.4 \\ 0 \end{bmatrix}, \boldsymbol{q}_3(0) = \begin{bmatrix} -0.45 \\ 0.15 \\ 0 \end{bmatrix}$$

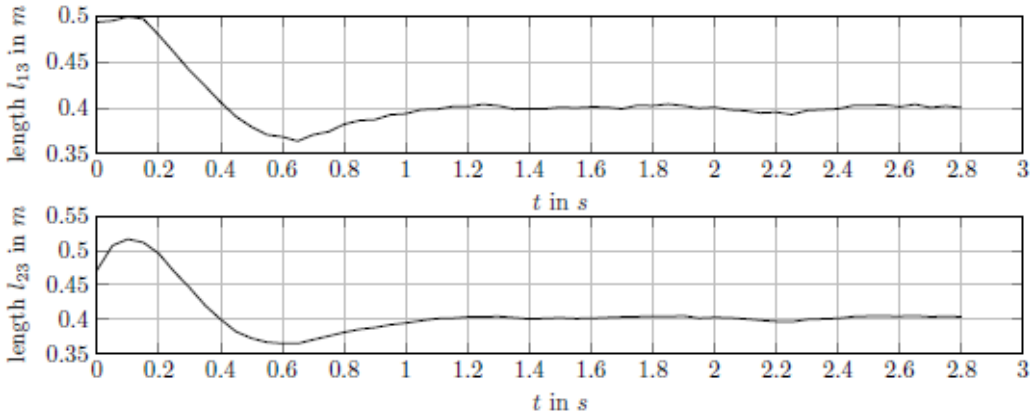

Figure 16. Results of experiments on real robots with the $l - l$ controller

Figure 17 visualizes the formation. It shows that both leaders perform a straight line while the follower maintains the desired lengths and thus the formation. The initial convergence to the specified formation as described above can be seen as well. The Robot 1 is denoted as +, Robot 2 as o and Robot 3 is denoted as *.

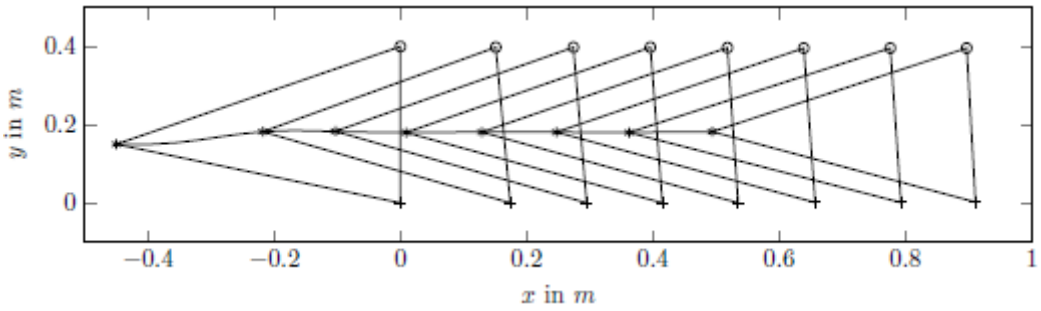

Figure 17. Visualization of the formation of the $l - l$ controller on the real robots

The results can be seen in Figure 16. It can be observed that the current length l_{13} and l_{23} are converging to the desired length l_{13}^{d} and l_{23}^{d} after a settling time. These desired lengths are maintained during the whole experiment, as required.

6.3. Experiments of the Collision Avoidance Controller

The setup of the experiments which are performed for the collision avoidance controller are similar to those in Subsection 5.4.1. The first experiment considers a static wall on the right side of the robots. Therefore, a constellation with one leader and one follower is considered. The formation specification is defined as $l_{12}^{\mathrm{d}} = 0.3$ and $\psi_{13}^{\mathrm{d}} = 180°$ with the initial pose.

$$\boldsymbol{q}_1(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{q}_2(0) = \begin{bmatrix} -0.43 \\ 0 \\ 0 \end{bmatrix} \tag{30}$$

The current length and the desired length as well as the current angle and the desired angle can be seen in Figure 18. Typical for the collision avoidance controller is the change in the desired length and desired angle. It can be observed that the current length and the current angle are close to the desired values with a bit of a time delay. These results are expected from the simulation with EyeSim in Subsection 5.4.1. The occurrence of the static obstacle on the right side can be seen at time t = 2.5 seconds. At that time a significant change in the desired angle can be observed. The angle is decreasing until t = 4.3 seconds. Subsequently, the robot overcomes the obstacle and the desired angle increases until the current angle reaches the initial angle ψ_{120}^{d}.

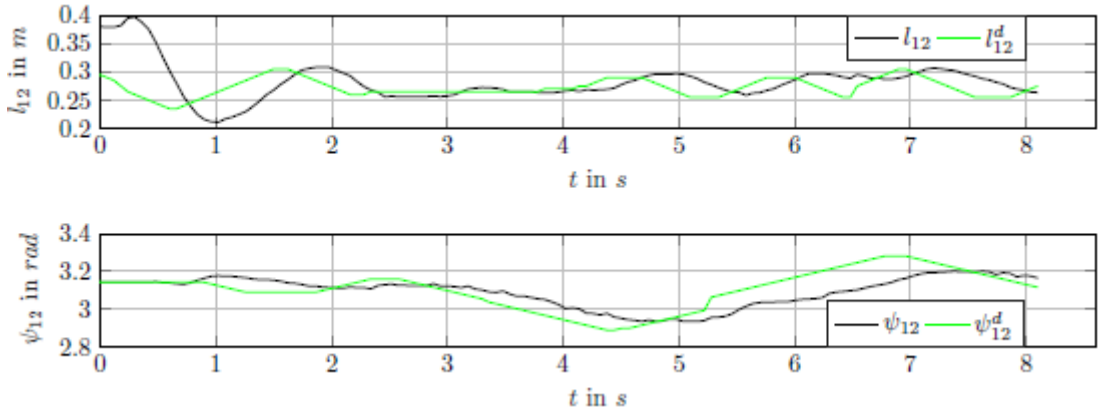

Figure 18. Results of experiments with the collision avoidance controller on real robots with an obstacle on the right

In Figure 19 a visualization of the formation can be seen. It can be observed that the follower is changing its position when the obstacles occur. Thus, a specified minimum distance to the obstacle is maintained. After the follower passes the obstacle, the desired angle changes back to the initial one.

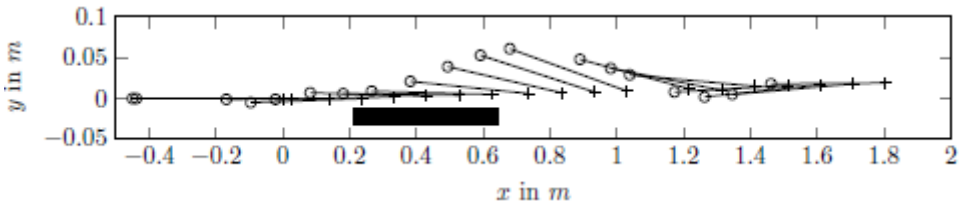

Figure 19. Visualization of the formation of the collision avoidance controller on real robots with an obstacle on the right. The Robot 1 is denoted as+and Robot 2 is denoted as o.

The second experiment concerns the avoidance of a collision with a robot to its front. This one is similar to the experiment in Subsection 5.4.1., with one leader and two followers. These two followers are trying to achieve the same formation, which is defined as $l_{1i}^{\mathrm{d}} = 0.3$ and $\psi_{1i}^{\mathrm{d}} = 180°$ $(i = 1, 2)$ with the initial pose

$$\boldsymbol{q}_1(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{q}_2(0) = \begin{bmatrix} -0.45 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{q}_3(0) = \begin{bmatrix} -0.9 \\ 0 \\ 0 \end{bmatrix} \tag{31}$$

The collision avoidance controller considers the distance to another object, measured by the PSD sensors. For practical purposes, another safety feature is added. If the measured distance is below a critical distance, the robot is forced to stop. This method is necessary at the beginning of the experiment due to the initial position of the second follower. Therefore, the speed at the beginning is too high to stop the robot with the collision avoidance controller. The results of this experiment can be seen in Figure 20. Due to the fact of another robot being in front of the follower two, the desired length l_{13}^{d} is increased. After the safety distance is maintained, the robot achieves a new desired length. With this desired length, no collision occurs. The stopping of the robot can be observed at the beginning of the experiment as a decreasing and increasing of the current length.

7. CONCLUSION

This paper investigates formation controllers for the practical implementation on given robots. An existing parametric dynamic model is used as a basic model for the investigation of different formation controllers. A model identification is performed to obtain the parameters of the dynamic model. Based on this model, the parameters of the PID controller for the speed of each wheel are determined. After introducing the considered formation controllers and proposing a new collision avoidance controller, several simulations are carried out. Studying the results of the simulations, certain requirements concerning the implementation can be imposed. Crucial requirements are that the velocity controller is fast enough and that the trajectory of the leader has to be planned concerning the ability of the robots.

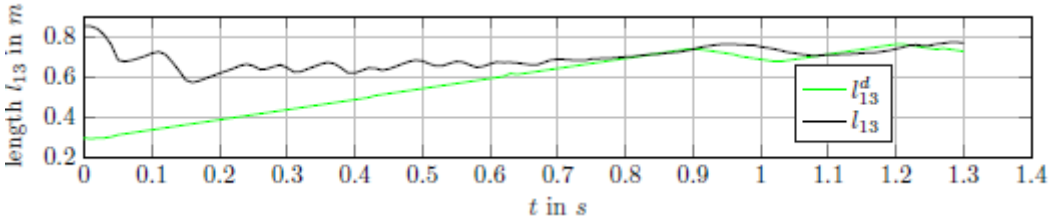

Figure 20. Current length and desired length of the follower two with the collision avoidance controller on real robots

Considering all these demands, the implementation is performed successfully. The results of the experiment show a similar behavior to the simulations and thus validates the simulations. The successful implementation and experiments show the ability of applying a formation controller on real robots which was developed for a kinematic model of a mobile robot. In addition to this, it shows that the given mobile robots are able to perform a formation with these formation controllers. The results of the collision avoidance controller show the ability of maintaining a safe distance to an obstacle and avoiding a collision.

The simulation models developed in this work can be used for further investigations of formation controllers for the application on real robots. With these models, it is possible to verify the ability of implementing a formation controller on real robots. Critical parts of the implementation can be determined and appropriate improvements can be carried out. A further improvement involves the dynamic model of the mobile robot. This dynamic model can be extended to consider the friction of the castor. Therefore, a suitable friction model should be chosen. In addition to that, new formation control approaches can be considered, such as a model predictive control approach. Since this controller requires high computational power, investigations should occur to assure sufficient amounts of computational power prior usage. The last improvement includes the collision avoidance controller. The ability of maintaining a desired distance to a static obstacle laterally can be extended to avoid obstacles in front of a robot. Therefore, the robots ability to sense obstacles in front has to be considered.

REFERENCES

[1] D. J. Stilwell and B. E. Bishop. "Platoons of underwater vehicles". IEEE Control Systems, 20(6):45{52, 2000.
[2] Quan Chen Yang and Wang Zhongmin. Formation control: a review and a new consideration. In International Conference on Intelligent Robots and Systems, pages 3181{3186, 2005.
[3] D. P. Scharf, F. Y. Hadaegh, and S. R. Ploen. "A survey of spacecraft formation ying guidance and control". part ii: control. In Proceedings of the American Control Conference, volume 4, pages 2976{2985, 2004.
[4] H. Raza and P. Ioannou. "Vehicle following control design for automated highway systems". IEEE Control Systems, 16(6): 43{60, 1996.
[5] J. G. Bender. "An overview of systems studies of automated highway systems. " *IEEE Transactions on Vehicular Technology*, 40(1):82{99, 1991.
[6] A. Serrani. "Robust coordinated control of satellite formations subject to gravity perturbations". In Proceedings of the American Control Conference, volume 1, pages 302{307, 2003.
[7] A. de Luca, G. Oriolo, and C. Samson. "Feedback control of a nonholonomic car-like robot". In J.-P Laumond, editor, Robot Motion Planning and Control, volume 229 of Lecture Notes in Control and Information Sciences, pages 171{253. Springer Berlin Heidelberg, 1998.
[8] Hiroaki Yamaguchi and Gerardo Beni. "Distributed autonomous formation control of mobile robot groups by swarm-based pattern generation". In Hajime Asama, Toshio Fukuda, Tamio Arai, and Isao Endo, editors, Distributed Autonomous Robotic Systems 2, pages 141{155. Springer Japan, 1996.
[9] C. Samson and K. Ait Abderrahim. "Feedback control of a nonholonomic wheeled cart in cartesian space". In IEEE International Conference on Robotics and Automation, pages 1136{1141, 1991.
[10] B. d'Andrea Novel, G. Campion, and G. Bastin."Control of nonholonomic wheeled mobile robots by state feedback linearization". Int. J. Rob. Res., 14(6):543{559, 1995.
[11] De Luca, Alessandro, Di Benedetto, Maria Domenica."Control of nonholonomic systems via dynamic compensation". Kybernetika, 29(6):593{608, 1993.
[12] G. Oriolo, A. de Luca, and M. Vendittelli. "Wmr control via dynamic feedback linearization: design, implementation, and experimental validation". IEEE Transactions on Control Systems Technology, 10(6):835{852, 2002.
[13] Ping Jiang Zhong and H. Nijmeijer. "A recursive technique for tracking control of nonholonomic systems in chained form". IEEE Transactions on Automatic Control, 44(2):265{279, 1999.
[14] Chung Lee Ti, Tai Song Kai, Hung Lee Ching, and Cheng Teng Ching. "Tracking control of unicycle-modeled mobile robots using a saturation feedback controller". IEEE Transactions on Control Systems Technology, 9(2):305{318, 2001.
[15] Elena Panteley, Erjen Lefeber, Henk Nijmeijer, and Petersbourg St. Russia. "Exponential tracking control of a mobile car using a cascaded approach". In In Proceedings of the IFAC Workshop on Motion Control, pages 221{226, 1998.
[16] H. Yamaguchi and J. W. Burdick. "Time-varying feedback control for nonholonomic mobile robots forming group formations". In Proceedings of the 37th IEEE Conference on Decision and Control, volume 4, pages 4156{4163, 1998.
[17] Lin Zhiyun, B. Francis, and M. Maggiore. "Necessary and sufficient graphical conditions for formation control of unicycles". IEEE Transactions on Automatic Control, 50(1):121{127, 2005.
[18] T. Balch and R. C. Arkin. "Behavior-based formation control for multirobot teams". *IEEE Transactions on Robotics and Automation*, 14(6):926{939, 1998.
[19] M.Anthony Lewis and Kar-Han Tan. "High precision formation control of mobile robots using virtual structures: Au-tonomous robots". Autonomous Robots, 4(4):387{403, 1997. ISSN 0929-5593.
[20] J. P. Desai, J. Ostrowski, and V. Kumar."Controlling formations of multiple mobile robots". In IEEE International Conference on Robotics and Automation, volume 4, pages 2864{2869, 1998.
[21] H. Takahashi, H. Nishi, and K. Ohnishi. Autonomous decentralized control for formation of multiple mobile robots considering ability of robot. IEEE Transactions on Industrial Electronics, 51(6):1272{1279, 2004.
[22] N. Cowan, O. Shakerina, R. Vidal, and S. Sastry. "Vision-based follow-the-leader". In International Conference on Intelligent Robots and Systems, volume 2, pages 1796{1801, 2003.
[23] J.R.T. Lawton, R. W. Beard, and B. J. Young. A decentralized approach to formation maneuvers. IEEE Transactions on Robotics and Automation, 19(6):933{941, 2003.
[24] G. Antonelli, F. Arrichiello, and S. Chiaverini. "Experiments of formation control with multirobot systems using the null-space-based behavioral control". *IEEE Transactions on Control Systems Technology*, 17(5):1173{1182, 2009. ISSN 1063-6536
[25] J. Ghommam, M. Saad, and F. Mnif."Formation path following control of unicycle-type mobile robots". In IEEE Interna-tional Conference on Robotics and Automation, pages 1966{1972, 2008.
[26] Yang Liu and Yingmin Jia. "An iterative learning approach to formation control of multi-agent systems". Systems & Control Letters, 61(1):148{154, 2012.
[27] Fang Lei and P. J. Antsaklis. "Decentralized formation tracking of multi-vehicle systems with nonlinear dynamics". In 14th Mediterranean Conference on Control and Automation, pages 1{6, 2006.
[28] Lim Heonyoung, Kang Yeonsik, Kim Changwhan, Kim Jongwon, and Jae You Bum. "Nonlinear model predictive controller design with obstacle avoidance for a mobile robot". In International Conference on Mechtronic and Embedded Systems and Applications, pages 494{499, 2008.

[29] M. A. Kamel and Zhang Youmin. "Decentralized leader-follower formation control with obstacle avoidance of multiple unicycle mobile robots". In IEEE 28th Canadian Conference on Electrical and Computer Engineering, pages 406{411, 2015.
[30] R. Fierro, A. K. Das, V. Kumar, and J. P. Ostrowski."Hybrid control of formations of robots". In IEEE International Conference on Robotics and Automation, volume 1, pages 157{162, 2001.
[31] H. Takahashi and K. Ohnishi. "Autonomous decentralized control for formation of multiple mobile-robots considering ability of robot". In The 29th Annual Conference of the IEEE Industrial Electronics Society, volume 3, pages 2041{2046, 2003.
[32] Lei Cheng and Wang Yongji."Communication-based multiple mobile robots rigid formation control. In Control, Automa-tion, Robotics and Vision Conference, volume 1, pages 729{734, 2004.
[33] In-Sung Choi and Jong-Suk Choi."*Leader-follower formation control using pid controller*". In Proceedings of the 5th International Conference on Intelligent Robotics and Applications - Volume Part II, ICIRA'12, pages 625{634, Berlin, Heidelberg, 2012. Springer-Verlag.
[34] Zhang Yulin, Hong Daehie, J. H. Chung, and S. A. Velinsky. "Dynamic model based robust tracking control of a differentially steered wheeled mobile robot". In Proceedings of the American Control Conference, volume 2, pages 850{855, 1998.
[35] Je rey C Lagarias, James A Reeds, Margaret H Wright, and Paul E Wright. "Convergence properties of the nelder{mead simplex method in low dimensions". SIAM Journal on optimization, 9(1):112{147, 1998.
[36] Thomas Br•aunl."Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems." Springer Publishing Company, Incorporated, 3rd ed. edition, 2008.
[37] Andreas Koestler and Thomas Br•aunl. "Mobile robot simulation with realistic error models". In International Conference on Autonomous Robots and Agents, pages 46{51 (6).

BIOGRAPHIES OF AUTHORS

Hannes Wind recieved his B.Eng. degree in mechatronics from the University of Ap-plied Sciences Aalen, Germany, in 2014 and he is currently studying mechatronics (M.Sc.) at the University of Stuttgart, Germany. He was a visitor at the University of Western Australia, Perth, in 2015

Oliver Sawodny received his Dipl.-Ing. degree in electrical engineering from the Univer-sity of Karlsruhe, Karlsruhe, Germany, in1991 and his Ph.D. degree from the University of Ulm, Ulm, Germany, in 1996. In 2002, he became a Full Professor at the Technical University of Ilmenau, Ilmenau, Germany. Since 2005, he has been the Director of the Institute for System Dynamics, University of Stuttgart, Stuttgart, Germany. His current research interests include methods of differential geometry, trajectory generation, and applications to mechatronic systems. He received important paper awards in major con-trol application journals such as Control Engineering Practice Paper Prize (IFAC, 2005) and IEEE Transaction on Control System Technology Outstanding Paper Award (2013).

Thomas Br•aunl is Professor at The University of Western Australia, Perth, where he directs the Renewable Energy Vehicle Project (REV) as well as the Robotics&Automation Lab. He has converted several road-licensed cars to battery-electric drive and autonomous driving and operates one of Australia largest EV charging network with 24 AC and DC charging stations. He is Technical Director of the West Australian Electric Vehicle Trial and the Principal Investigator of the ARC Electric Vehicle Fast-Recharging Project. He has worked on Driver-Assistance Systems with Daimler/Mercedes-Benz and on Electric Vehicle Charging Systems with BMW. Professor Br•aunl holds a Diploma from Univ. Kaiserslautern, Germany, a M.S. from USC, Los Angeles, and a PhD and Habilitation from Univ. Stuttgart.