

Different Kinematic Path Following Controllers for a Wheeled Mobile Robot of (2,0) Type

Joanna Płaskonka

Received: 14 January 2013 / Accepted: 30 June 2013 / Published online: 19 September 2013
© The Author(s) 2013. This article is published with open access at Springerlink.com

Abstract In the paper a comparative study of selected kinematic path following controllers for a wheeled mobile robot of (2,0) type has been presented. The control strategies are based on one of two approaches to the path parameterization—either the Serret-Frenet frame with an orthogonal projection or the Serret-Frenet frame with a non-orthogonal projection of a robot on the desired path. The complete control system for the wheeled mobile robot consists of two parts, a kinematic controller and a dynamic controller, because of a presence of nonholonomic constraints. The behaviour of the presented algorithms is demonstrated through computer simulations.

Keywords Path following · Wheeled mobile robot · Serret-Frenet frame · Nonholonomic constraints · Model-based control

1 Introduction

The number of possible applications of wheeled mobile robots is huge and is even increasing,

just to mention platforms for transport operations, planetary exploration, automatic cleaning, museum tours etc. Thus robust control algorithms for such robotic objects are needed. There are numerous papers related to control algorithms for wheeled mobile robots published in the literature. Three basic types of tasks realized by mobile platforms can be distinguished:

- point stabilization,
- trajectory tracking (the robot has to follow a desired curve which is time-parametrized),
- path following (the task of the robot is to follow a curve parametrized by a curvilinear distance from a fixed point).

While realizing a trajectory tracking task the robot needs to be in a particular position at a pre-specified time and thus the robot could be pushed to its performance limits. In contrary, a path following task requires the robot to converge to a geometric curve with any feasible speed profile. This is why a path following problem bypasses the limitations of the trajectory tracking task. The objective of the paper is to review different path following algorithms for a unicycle-type wheeled mobile robot and verify their robustness properties with respect to noise and velocity constraints. The considered type of a problem corresponds to e.g. automatic wall following or driving on a road while trying to maintain the distance between vehicle chassis and the side of the road constant.

J. Płaskonka (✉)
Institute of Computer Engineering,
Control and Robotics, Wrocław University
of Technology, Wrocław, Poland
e-mail: joanna.plaskonka@pwr.wroc.pl

Many solutions of the path following problem for the unicycle are dedicated to that particular type of a mobile platform. In the approach presented in [1] the position of the virtual vehicle to be tracked is simply defined by the orthogonal projection of the robot on the path. Such an approach is valid locally, only near the path. The another approach described in [2] overcomes initial condition constraints imposed on the robot by controlling explicitly the rate of progression of a virtual target. The consequence of this approach is that the amount of path following errors that have to converge to zero is increased by one comparing to the previously mentioned approach. The method proposed in [3] solves the problem of the bounded path curvature. It neither requires the computation of a projection of the robot position on the path, nor does it need to consider a moving virtual target to be tracked. Morin-Samson algorithm [4] uses the kinematic equations of the robot derived with respect to the Serret-Frenet frame transformed into a canonical form called the chained form. Some of the algorithms available in the literature also deal with obstacles avoidance, e.g. [5–7], or allow to realize a formation path following task, see e.g. [8, 9].

More general approaches to designing kinematic control strategies may be applied for various nonholonomic mobile robots, e.g. one presented in [10]. However a generality of the proposed algorithm was achieved at the expense of the occurrence of an undesirable chattering phenomenon. Usually better results are obtained when using algorithms dedicated to particular types of wheeled mobile robots.

In the paper different kinematic path following algorithms—which use the kinematic model of the robot expressed with respect to the Serret-Frenet frame—are compared: Samson algorithm [11], Morin-Samson algorithm [4], Soetanto-Lapierre-Pascoal algorithm [2] and two proposed alternative discontinuous algorithms. The first discontinuous algorithm results from the attempt of linearisation and decoupling of the equations describing the dynamics of path following errors, while the latter one is a modification of Samson algorithm.

Table 1 Classification of wheeled mobile robots

σ_m	3	2	2	1	1
σ_s	0	0	1	1	2

2 The Mathematical Model of the Robot

There are five classes of wheeled mobile robots introduced in [12]. The proposed classification is based on two indices—a degree of mobility σ_m and a degree of steerability σ_s , see Table 1. The robots of (3,0) type have full mobility in the plane, which means that they are able to move in any direction without any reorientation, while the other have a restricted mobility. In the article considerations will be restricted to type (2,0) robots. Such robots have no steering wheels, but either one or several fixed wheels with a common axle, [13].

The wheeled mobile robot of (2,0) type, also called a unicycle, was depicted in Fig. 1. (x, y) represents coordinates of the characteristic point M relative to the inertial frame X_0Y_0 and θ is a robot's orientation. The point M is located in the middle of the wheel axle of the vehicle. R is the radius of the robot's wheel and d is a half the width of the mobile platform.

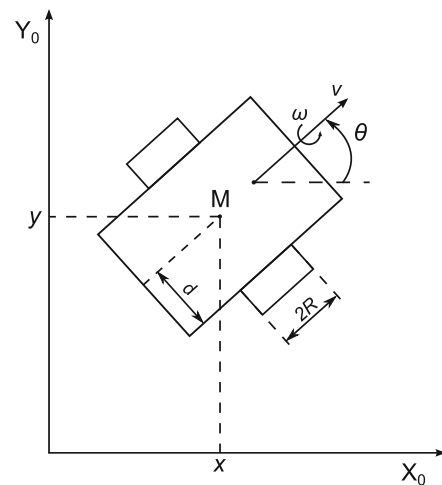


Fig. 1 The unicycle's parameters

2.1 The Kinematic Model

The motion of the unicycle can be described by generalized coordinates $\mathbf{q}_m \in R^n$ and generalized velocities $\dot{\mathbf{q}}_m \in R^n$. The assumption that the robot is moving on a plane without slippage of its wheels is made. It is equivalent to an assumption that the momentary velocity at the contact point between each wheel and the motion plane is equal to zero. This implies the existence of l ($l < n$) independent nonholonomic constraints expressed in Pfaffian form

$$\mathbf{A}(\mathbf{q}_m)\dot{\mathbf{q}}_m = 0, \tag{1}$$

where $\mathbf{A}(\mathbf{q}_m)$ is a full-rank matrix of $(l \times n)$ size. Since due to Eq. 1, the platform velocity is in a null space of $\mathbf{A}(\mathbf{q}_m)$, it is always possible to find a vector of special auxiliary velocities $\eta \in R^m$, $m = n - l$, such that

$$\dot{\mathbf{q}}_m = \mathbf{G}(\mathbf{q}_m)\eta, \tag{2}$$

where \mathbf{G} is an $n \times m$ full rank matrix satisfying the relationship $\mathbf{A}\mathbf{G} = 0$. The Eq. 2 is called the kinematics of the nonholonomic mobile platform. For the unicycle the Eq. 2 has the following form

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \end{pmatrix} = \begin{bmatrix} \cos \theta & \cos \theta \\ \sin \theta & \sin \theta \\ \frac{1}{d} & -\frac{1}{d} \\ 0 & \frac{2}{R} \\ \frac{2}{R} & 0 \end{bmatrix} \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix}, \tag{3}$$

where auxiliary velocities η_1 and η_2 are scaled rotation velocities of the unicycle’s wheels $\dot{\phi}_1$ and $\dot{\phi}_2$:

$$\eta_1 = \frac{R}{2}\dot{\phi}_2, \quad \eta_2 = \frac{R}{2}\dot{\phi}_1. \tag{4}$$

One can use notations

$$v = \eta_1 + \eta_2, \tag{5}$$

where v is the linear velocity of the robot and

$$\omega = \frac{1}{d}(\eta_1 - \eta_2), \tag{6}$$

where ω denotes angular velocity of the robot.

Using the symbols v and ω the kinematic model of the considered wheeled mobile robot can be described by the equations

$$\begin{cases} \dot{x} = v \cos \theta, \\ \dot{y} = v \sin \theta, \\ \dot{\theta} = \omega. \end{cases} \tag{7}$$

2.2 The Dynamic Model

To obtain the model of the unicycle’s dynamics, the d’Alembert Principle has to be evoked

$$\mathbf{M}(\mathbf{q}_m)\ddot{\mathbf{q}}_m = \mathbf{B}(\mathbf{q}_m)\tau + \mathbf{A}^T(\mathbf{q}_m)\lambda, \tag{8}$$

where:

- $\mathbf{M}(\mathbf{q}_m)$ inertia matrix of the mobile platform,
- $\mathbf{A}(\mathbf{q}_m)$ matrix of nonholonomic constraints for the platform,
- λ vector of Lagrange multipliers,
- $\mathbf{B}(\mathbf{q}_m)$ input matrix,
- τ vector of controls.

Matrix $\mathbf{A}(\mathbf{q}_m)$ is Pfaffian matrix for the platform, see Eq. 1, while $\mathbf{B}(\mathbf{q}_m)$ describes which state variable is directly controlled by the actuator. The model (8) can be also expressed using auxiliary velocities η . For this purpose $\ddot{\mathbf{q}}_m$ is computed

$$\ddot{\mathbf{q}}_m = \mathbf{G}(\mathbf{q}_m)\dot{\eta} + \dot{\mathbf{G}}(\mathbf{q}_m)\eta \tag{9}$$

and the vector of Lagrange multipliers is eliminated, using a condition $\mathbf{G}^T\mathbf{A}^T = 0$, by left-sided multiplying the mobile platform equations by \mathbf{G}^T matrix. After substituting $\ddot{\mathbf{q}}_m$ the Eq. 8 transforms into

$$\mathbf{M}^*(\mathbf{q}_m)\dot{\eta} = \mathbf{B}^*(\mathbf{q}_m)\tau \tag{10}$$

with

$$\mathbf{M}^* = \mathbf{G}^T\mathbf{M}\mathbf{G} \tag{11}$$

and

$$\mathbf{B}^* = \mathbf{G}^T\mathbf{B}. \tag{12}$$

The considered dynamic model of the wheeled mobile robot of (2,0) type is very simple as elements of the inertia matrix are constant, nevertheless introducing dynamic model allows to observe

an influence of the inertia of the robot on the realization of a desired task.

3 Description of the Robot Relative to a Desired Path

To describe the position of the robot relative to a desired path, different approaches may be applied. The one of the most commonly used approaches is based on the Serret-Frenet frame—which consists of vectors tangent, normal and binormal to a curve—moving along a desired path. The robot moving on a plane is considered so the Serret-Frenet frame consists only of vectors tangent and normal to the path.

The path P is characterized by a curvature $\kappa(s)$, which is the inversion of the radius of the circle tangent to the path at a point characterized by the parameter s which is a curvilinear distance from the fixed point, i.e. from the beginning of the path. The desired orientation of the mobile platform satisfies the equation

$$\dot{\theta}_r = \pm \kappa(s) \dot{s}. \tag{13}$$

The sign on the right side of the Eq. 13 depends on the direction of moving along a desired curve (negative when the Serret-Frenet frame moves in the clockwise direction, positive otherwise).

3.1 The Serret-Frenet Frame with an Orthogonal Projection of a Robot on the Path

In this approach it is assumed that the robot has to follow a virtual target which is its orthogonal projection on the desired curve, see Fig. 2. The point M' is the orthogonal projection of the point M on the path P and l is the distance error between the actual vehicle and the virtual one. The point M' exists and is uniquely defined if the following conditions are satisfied:

- the curvature $\kappa(s)$ is bounded from above (the desired path cannot be too much curved),
- the robot cannot move too far from the path (the Serret-Frenet parametrization is local),

see for a detailed explanation e.g. [14]. The position of the point M relative to an inertial frame

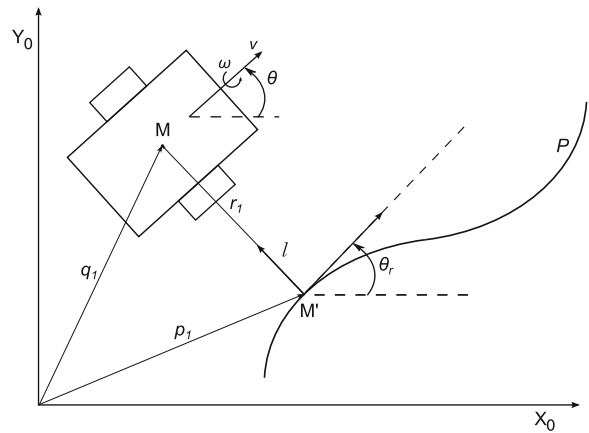


Fig. 2 The Serret-Frenet frame definition in a case of an orthogonal projection of a robot on the path

is equal to $\mathbf{t}_0 = \begin{pmatrix} q_1 \\ 1 \end{pmatrix}$, while its position relative to the Serret-Frenet frame is equal to $\mathbf{t}_1 = \begin{pmatrix} r_1 \\ 1 \end{pmatrix}$. The relationship between \mathbf{t}_0 and \mathbf{t}_1 can be described by the matrix equation

$$\begin{pmatrix} q_1 \\ 1 \end{pmatrix} = \begin{bmatrix} \mathbf{R}_{\theta_r} & \mathbf{p}_1 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} r_1 \\ 1 \end{pmatrix}, \tag{14}$$

where

$$\mathbf{R}_{\theta_r} = \mathbf{R} = \mathbf{Rot}(z, \theta_r) = \begin{bmatrix} \cos \theta_r & -\sin \theta_r & 0 \\ \sin \theta_r & \cos \theta_r & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{15}$$

Therefore the following equation holds

$$\mathbf{q}_1 = \mathbf{R} \mathbf{r}_1 + \mathbf{p}_1. \tag{16}$$

After differentiating and transforming the Eq. 16, one has

$$\mathbf{R} \dot{\mathbf{r}}_1 = \dot{\mathbf{q}}_1 - \dot{\mathbf{R}} \mathbf{r}_1 - \dot{\mathbf{p}}_1. \tag{17}$$

The Eq. 17 could be multiplied by \mathbf{R}^T on the left side

$$\dot{\mathbf{r}}_1 = \mathbf{R}^T \dot{\mathbf{q}}_1 - \mathbf{R}^T \dot{\mathbf{R}} \mathbf{r}_1 - \mathbf{R}^T \dot{\mathbf{p}}_1. \tag{18}$$

The transformation from the inertial frame $X_0 Y_0$ to the Serret-Frenet frame is equal to

$$\mathbf{K}_0^1 = \begin{bmatrix} \mathbf{R} & \mathbf{p}_1 \\ 0 & 1 \end{bmatrix}, \tag{19}$$

thus the velocity of point **M** relative to the Serret-Frenet frame can be calculated as follows

$$\mathbf{v}_B = (\mathbf{K}_0^1)^{-1} \dot{\mathbf{K}}_0^1 = \begin{bmatrix} \mathbf{R}^T \dot{\mathbf{R}} & \dot{\mathbf{R}}^T \dot{\mathbf{p}}_1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} [\omega_B] & \mathbf{v}_B \\ 0 & 0 \end{bmatrix}. \tag{20}$$

Using the relationships

$$\mathbf{r}_1 = (0 \quad l \quad 0)^T, \tag{21}$$

$$\mathbf{q}_1 = (x \quad y \quad 0)^T \tag{22}$$

and

$$\mathbf{v}_B = \dot{\mathbf{R}}^T \dot{\mathbf{p}}_1 = (\dot{s} \quad 0 \quad 0)^T, \tag{23}$$

as the reference vehicle is moving along a desired path in the direction of *X* axis of the Serret-Frenet frame, the Eq. 18 can be rewritten as

$$\begin{pmatrix} 0 \\ \dot{l} \\ 0 \end{pmatrix} = \begin{bmatrix} \cos \theta_r & \sin \theta_r & 0 \\ -\sin \theta_r & \cos \theta_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{s} \end{pmatrix} - \begin{bmatrix} 0 & -\dot{\theta}_r & 0 \\ \dot{\theta}_r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} 0 \\ l \\ 0 \end{pmatrix} - \begin{pmatrix} \dot{s} \\ 0 \\ 0 \end{pmatrix}. \tag{24}$$

Finally one can obtain equations for \dot{l} and \dot{s}

$$\dot{l} = (-\sin \theta_r \quad \cos \theta_r) \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix}, \tag{25}$$

$$\dot{s} = \frac{(\cos \theta_r \quad \sin \theta_r)}{1 \mp \kappa(s)l} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix}. \tag{26}$$

To avoid singularity $l = \pm \frac{1}{\kappa(s)}$ one has to ensure that during the control process an inequality $|l\kappa(s)| < 1$ holds, which means that the parametrization is local. In addition, the orientation error is determined

$$\tilde{\theta} = \theta - \theta_r \tag{27}$$

and its derivative

$$\dot{\tilde{\theta}} = \dot{\theta} - \dot{\theta}_r = \dot{\theta} \mp \kappa(s)\dot{s}. \tag{28}$$

For the unicycle the kinematic model expressed with respect to the Serret-Frenet frame with an

orthogonal projection of a robot on the path is given by the following system of equations

$$\begin{cases} \dot{l} = v \sin \tilde{\theta}, \\ \dot{s} = \frac{v \cos \tilde{\theta}}{1 \mp \kappa(s)l}, \\ \dot{\tilde{\theta}} = \omega \mp \frac{\kappa(s)v \cos \tilde{\theta}}{1 \mp \kappa(s)l}. \end{cases} \tag{29}$$

The model described by the Eq. 29 can be transformed into the three-dimensional chained system

$$\begin{cases} \dot{z}_1 = u_1, \\ \dot{z}_2 = u_2, \\ \dot{z}_3 = z_2 u_1 \end{cases} \tag{30}$$

using the change of coordinates

$$\begin{cases} z_1 = s, \\ z_2 = (1 \mp \kappa(s)l) \tan \tilde{\theta}, \\ z_3 = l \end{cases} \tag{31}$$

and control variables

$$\begin{aligned} u_1 &= \frac{v \cos \tilde{\theta}}{1 \mp \kappa(s)l}, \\ u_2 &= (\mp \dot{s})l \mp \kappa(s)u_1 z_2 \cdot \tan \tilde{\theta} \\ &\quad + \frac{1 \mp \kappa(s)l}{\cos^2 \tilde{\theta}} (\omega + u_1 \kappa(s)). \end{aligned} \tag{32}$$

The presented transformation into the chained system is local, $|\tilde{\theta}| < \frac{\pi}{2}$.

3.2 The Serret-Frenet Frame with a Non-orthogonal Projection of a Robot on the Path

The similar reasoning can be carried out for the case when the position of the virtual target is defined by the non-orthogonal projection of the actual vehicle on the path (see Fig. 3). In that situation there are three path following errors—an orientation error and two distance errors, one in the direction of the *X* axis and second in the direction of *Y* axis of the Serret-Frenet frame. The velocity of **r**₂ is equal to

$$\dot{\mathbf{r}}_2 = \mathbf{R}^T \dot{\mathbf{q}}_2 - \mathbf{R}^T \dot{\mathbf{R}} \mathbf{r}_2 - \mathbf{R}^T \dot{\mathbf{p}}_2 \tag{33}$$

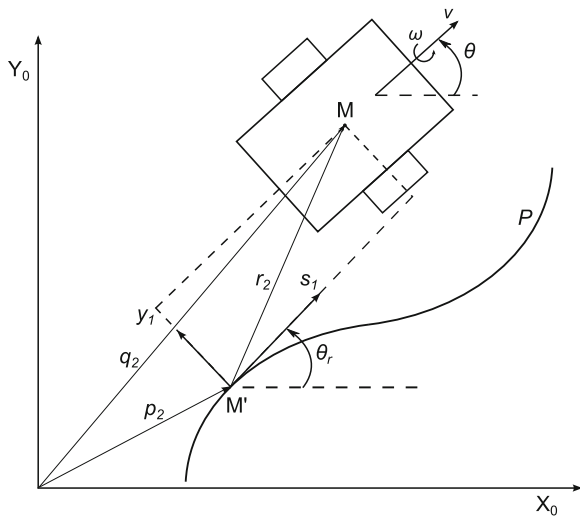


Fig. 3 The Serret-Frenet frame definition in a case of a non-orthogonal projection of a robot on the path

with

$$\mathbf{r}_2 = (s_1 \quad y_1 \quad 0)^T, \tag{34}$$

$$\mathbf{q}_2 = (x \quad y \quad 0)^T \tag{35}$$

and

$$\mathbf{R}^T \dot{\mathbf{p}}_2 = (\dot{s} \quad 0 \quad 0)^T. \tag{36}$$

From Eq. 33 one gets

$$\begin{pmatrix} \dot{s}_1 \\ \dot{y}_1 \\ 0 \end{pmatrix} = \begin{bmatrix} \cos \theta_r & \sin \theta_r & 0 \\ -\sin \theta_r & \cos \theta_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ 0 \end{pmatrix} - \begin{bmatrix} 0 & -\dot{\theta}_r & 0 \\ \dot{\theta}_r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} s_1 \\ y_1 \\ 0 \end{pmatrix} - \begin{pmatrix} \dot{s} \\ 0 \\ 0 \end{pmatrix} \tag{37}$$

and thus

$$\dot{s}_1 = (\cos \theta_r \quad \sin \theta_r) \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} - \dot{s}(1 - y_1 \kappa(s)), \tag{38}$$

$$\dot{y}_1 = (-\sin \theta_r \quad \cos \theta_r) \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} - \dot{s} \kappa(s) s_1. \tag{39}$$

There is no singularity related to distance errors in Eqs. 38–39. Thus a path parametrization based on the Serret-Frenet frame with a non-orthogonal projection of a robot on the path is not local.

For the unicycle the kinematic model expressed with respect to the Serret-Frenet frame with a non-orthogonal projection of a robot on the path is given by the following system of equations

$$\begin{cases} \dot{s}_1 = -\dot{s}(1 - \kappa(s)y_1) + v \cos \tilde{\theta}, \\ \dot{y}_1 = -\kappa(s)\dot{s}s_1 + v \sin \tilde{\theta}, \\ \dot{\tilde{\theta}} = \omega \mp \kappa(s)\dot{s}. \end{cases} \tag{40}$$

4 Control Problem Statement

The control problem for the considered robot is as follows:

Design a control law τ such that the wheeled mobile robot of (2,0) type with a fully known dynamics follows a desired path and path following errors converge to zero.

The desired path has to be admissible, i.e. it can be realized without slippage of robot’s wheels. It is assumed that the desired path is a smooth curve which has two smooth and bounded derivatives relative to s variable.

The complete model of a nonholonomic system consisting of kinematics and dynamics has a structure of two cascaded equations. To find a control law for such a cascade system, one may use the backstepping-like procedure [15]:

- kinematic controller η_r —generates a ‘velocity profile’ which ensures the realizability of a path following task in such a way that nonholonomic constraints are satisfied; the kinematic controller does not take into account the dynamics of the object,
- dynamic controller τ —its task is to regulate real velocities η towards the reference control η_r .

5 Path Following for a Unicycle—Kinematic Control

There are different approaches to formulation of a path following task. The robot’s motion can finish by stopping on a path or it may be continuous, cyclic. In this paper the latter approach is adopted. It is assumed that a direction of a movement along

the desired curve is opposite to the clockwise direction, this means that

$$\dot{\tilde{\theta}} = \kappa(s)\dot{s}. \tag{41}$$

Hence the equations describing path following errors for a case of an orthogonal projection of a robot on the path have a form as below

$$\begin{aligned} \dot{l} &= v \sin \tilde{\theta}, \\ \dot{\tilde{\theta}} &= \omega - \frac{\kappa(s)v \cos \tilde{\theta}}{1 - \kappa(s)l} \end{aligned} \tag{42}$$

and in a case of a non-orthogonal projection are equal to

$$\begin{aligned} \dot{s}_1 &= -\dot{s}(1 - \kappa(s)y_1) + v \cos \tilde{\theta}, \\ \dot{y}_1 &= -\kappa(s)\dot{s}s_1 + v \sin \tilde{\theta}, \\ \dot{\tilde{\theta}} &= \omega - \kappa(s)\dot{s}. \end{aligned} \tag{43}$$

5.1 Samson Control Algorithm

Path following errors for the unicycle can be expressed as follows

$$\dot{l} = v \sin \tilde{\theta}, \tag{44}$$

$$\dot{\tilde{\theta}} = u, \tag{45}$$

where $u = \omega - \frac{\kappa(s)\cos \tilde{\theta}}{1 - \kappa(s)l}v$ is a new control for the second equation. Samson kinematic control law [11] for the systems (44)–(45) is equal to

$$v = const, \quad u = -k_2lv \frac{\sin \tilde{\theta}}{\tilde{\theta}} - k_3\tilde{\theta}, \quad k_2, k_3 > 0. \tag{46}$$

The systems (44)–(45) with a closed-loop of the feedback signal (46) is asymptotically stable. That can be shown using the following Lyapunov function

$$V(l, \tilde{\theta}) = k_2 \frac{l^2}{2} + \frac{\tilde{\theta}^2}{2} \tag{47}$$

and Barbalat lemma, see for details e.g. [16].

5.2 Discontinuous Control Algorithm—First Proposal

The following discontinuous control law for the system (42) is proposed

$$\begin{aligned} v &= \begin{cases} \frac{-k_1l}{\sin \tilde{\theta}} \tilde{\theta} \neq 0, \\ v_c \tilde{\theta} = 0, \end{cases} \\ \omega &= -k_2\tilde{\theta} + \frac{\kappa(s)v \cos \tilde{\theta}}{1 - \kappa(s)l}, \end{aligned} \tag{48}$$

where $v_c = const$, $k_1, k_2 > 0$. The initial orientation error $\tilde{\theta}(0)$ can not be equal to zero. The system (42) with a closed-loop of a signal (48) has the form

$$\begin{aligned} \dot{l} &= \begin{cases} -k_1l \tilde{\theta} \neq 0, \\ 0 \tilde{\theta} = 0, \end{cases} \\ \dot{\tilde{\theta}} &= -k_2\tilde{\theta}. \end{aligned} \tag{49}$$

The inequality $|l\kappa(s)| < 1$ resulting from the restrictions imposed on the desired path is satisfied when $l(0)$ satisfies this inequality since the time-derivative of l is non-increasing.

The expression $\frac{-k_1l}{\sin \tilde{\theta}}$ is undefined for $(l, \tilde{\theta}) = (0, 0)$. Therefore the proposed algorithm is applied in a modified version

$$\begin{aligned} v &= \begin{cases} \frac{-k_1l}{\sin \tilde{\theta}} |\tilde{\theta}| > \epsilon, \\ v_c |\tilde{\theta}| \leq \epsilon, \end{cases} \\ \omega &= -k_2\tilde{\theta} + \frac{\kappa(s)v \cos \tilde{\theta}}{1 - \kappa(s)l} \end{aligned} \tag{50}$$

with ϵ to be a small positive number, e.g. $\epsilon = 0.001$.

5.3 Discontinuous Control Algorithm—Second Proposal

The another proposed kinematic path following algorithm for the system (42)

$$\begin{aligned} v &= \begin{cases} \frac{-k_1l}{\sin \tilde{\theta}} \tilde{\theta} \neq 0, \\ v_c \tilde{\theta} = 0, \end{cases} \\ \omega &= -k_2lv \frac{\sin \tilde{\theta}}{\tilde{\theta}} - k_3\tilde{\theta} + \frac{\kappa(s)v \cos \tilde{\theta}}{1 - \kappa(s)l} \end{aligned} \tag{51}$$

is a modification of the Samson algorithm [11]. In practice this algorithm is used in the following form

$$v = \begin{cases} -k_1 l \frac{\sin \tilde{\theta}}{\tilde{\theta}} > \epsilon, \\ v_c & |\tilde{\theta}| \leq \epsilon, \end{cases} \tag{52}$$

$$\omega = -k_2 l v \frac{\sin \tilde{\theta}}{\tilde{\theta}} - k_3 \tilde{\theta} + \frac{\kappa(s)v \cos \tilde{\theta}}{1 - \kappa(s)l},$$

where ϵ is a small positive number.

5.4 Morin-Samson Control Algorithm

For the kinematic equations expressed with respect to the Serret-Frenet frame with an orthogonal projection of the guidance point on a path which are transformed into the 3-dimensional chained system (30), the following control law is proposed [4]

$$\begin{aligned} u_1 &= \text{const}, \\ u_2 &= -u_1 k_3 z_3 - |u_1| k_2 z_2. \end{aligned} \tag{53}$$

In general u_1 does not have to be constant. It is sufficient if it is a bounded differentiable time function whose derivative is bounded and which does not tend to zero as t tends to infinity.

What is more, if the following inequality holds

$$z_3^2(0) + \frac{1}{k_3} z_2^2(0) < \frac{1}{\kappa_{\max}^2} \tag{54}$$

with $\kappa_{\max} = \max_s |\kappa(s)|$, then the constraint $|l \kappa(s)| < 1$ is satisfied along any solution to the controlled system.

Proof The system containing only state variables z_2 and z_3 is considered. Such a system can be written in the below matrix form

$$\dot{\mathbf{z}}_c = \mathbf{\Lambda} \mathbf{z}_c, \tag{55}$$

where $\mathbf{z}_c = (z_2 \ z_3)^T$ and a matrix $\mathbf{\Lambda}$ is equal to

$$\mathbf{\Lambda} = \begin{bmatrix} -|u_1|k_2 & -u_1 k_3 \\ u_1 & 0 \end{bmatrix}. \tag{56}$$

The system (55) will be asymptotically stable if all eigenvalues of the matrix $\mathbf{\Lambda}$ have negative real part, i.e. the characteristic polynomial of the $\mathbf{\Lambda}$ matrix

$$p(\lambda) = \det(\lambda \mathbf{I}_2 - \mathbf{\Lambda}) = \lambda^2 + |u_1|k_2 \lambda + u_1^2 k_3 \tag{57}$$

must be Hurwitz polynomial. That condition is fulfilled for $k_i > 0$. Then consider the Lyapunov function

$$V(z_2, z_3) = \frac{1}{2}(z_2^2 + k_3 z_3^2). \tag{58}$$

For a positive coefficient k_3 the proposed Lyapunov function is non-negative

$$\begin{aligned} V &> 0, \quad (z_2, z_3) \neq (0, 0), \\ V &= 0, \quad (z_2, z_3) = (0, 0). \end{aligned} \tag{59}$$

The time-derivative of V calculated along a system's solution

$$\dot{V} = z_2 \dot{z}_2 + k_3 z_3 \dot{z}_3 = -|u_1| k_2 z_2^2 \tag{60}$$

is non-positive. Thus V is non-increasing and converges to some limit value $V_{\lim} \geq 0$. This implies that the variables z_2 and z_3 are bounded. If it is assumed that the control signal u_1 is bounded, the second derivative of V

$$\ddot{V} = 2u_1 |u_1| k_2 k_3 z_2 z_3 + 2u_1^2 k_2^2 z_2^2 \tag{61}$$

is a sum of bounded functions. Therefore \dot{V} is uniformly continuous. From Barbalat lemma one may conclude that \dot{V} tends to zero which implies that z_2 tends to zero and thus $\tilde{\theta} \rightarrow 0$. The time-derivative of $v^2 z_2$

$$\overline{v^2 z_2} = 2v \dot{v} z_2 - v^2 u_1 k_3 z_3 - v |u_1| k_2 z_2 \tag{62}$$

is the sum of two terms which tends to zero and a third which is uniformly continuous since its derivative is bounded. Using the fact that $v^2 z_2$ tends to zero and the extension of Barbalat's lemma [1], one may conclude that $\overline{v^2 z_2} \rightarrow 0$. Thus $v^2 u_1 k_3 z_3$ tends to zero, too. As the linear velocity of the robot v does not tend to zero and u_1, k_3 are constant, $\lim_{t \rightarrow \infty} z_3$ must be equal to zero which means that $V_{\lim} = 0$ and $l \rightarrow 0$.

Moreover the following inequality holds

$$2V \geq k_3 z_3^2 \tag{63}$$

which leads to

$$z_3^2 \leq \frac{2V}{k_3} \leq \frac{2V(0)}{k_3} < \frac{1}{\kappa_{\max}^2}. \tag{64}$$

Finally one has

$$|l \kappa_{\max}| < 1. \tag{65}$$

□

5.5 Soetanto-Lapierre-Pascoal Control Algorithm

The control algorithm proposed in [2] for the system (43) is equal to

$$\begin{aligned} \dot{s} &= v \cos \tilde{\theta} + k_1 s_1, \\ \omega &= \kappa(s)\dot{s} + \dot{\delta} - \gamma y_1 v \frac{\sin \tilde{\theta} - \sin \delta}{\tilde{\theta} - \delta} - k_2(\tilde{\theta} - \delta), \end{aligned} \quad (66)$$

$k_1, k_2 > 0,$

where the following assumptions have been made

- $\lim_{t \rightarrow \infty} v(t) \neq 0$, e.g. $v = const$,
- $\delta(0, v) = 0$,
- $\forall y_1 \forall v \ y_1 v \sin \delta(y_1, v) \leq 0$.

It guarantees the convergence of y_1, s_1 and $\tilde{\theta}$ to zero.

Proof Consider the Lyapunov function

$$V = \frac{1}{2}(s_1^2 + y_1^2) + \frac{1}{2\gamma}(\tilde{\theta} - \delta(y_1, v))^2 \quad (67)$$

with γ to be constant. The time-derivate of V

$$\begin{aligned} \dot{V} &= s_1 \dot{s}_1 + y_1 \dot{y}_1 + \frac{1}{\gamma}(\tilde{\theta} - \delta)(\dot{\tilde{\theta}} - \dot{\delta}) \\ &= s_1(v \cos \tilde{\theta} - \dot{s}) + y_1 v(\sin \tilde{\theta} - \sin \delta) \\ &\quad + y_1 v \sin \delta + \frac{1}{\gamma}(\tilde{\theta} - \delta)(\omega - \kappa(s)\dot{s} - \dot{\delta}) \\ &= -k_1 s_1^2 - k_2(\tilde{\theta} - \delta)^2 \end{aligned} \quad (68)$$

in non-positive. This means that $\lim_{t \rightarrow \infty} V(t) = V_{lim}$ and s_1, y_1 and $(\tilde{\theta} - \delta)$ are bounded. \dot{V} is uniformly continuous because its derivative is bounded as sum of bounded functions. By Barbalat's lemma \dot{V} tends to zero. Therefore $s_1 \rightarrow 0$ and $(\tilde{\theta} - \delta) \rightarrow 0$. Differentiating $(\tilde{\theta} - \delta)$ with respect to time gives

$$\frac{\dot{\tilde{\theta}} - \dot{\delta}}{\tilde{\theta} - \delta} = \gamma y_1 v \frac{\sin \tilde{\theta} - \sin \delta}{\tilde{\theta} - \delta} - k_2(\tilde{\theta} - \delta). \quad (69)$$

Hence

$$\begin{aligned} \frac{\dot{\tilde{\theta}} - \dot{\delta}}{v^2(\tilde{\theta} - \delta)} &= 2v\dot{v}(\tilde{\theta} - \delta) - \gamma y_1 v^3 \frac{\sin \tilde{\theta} - \sin \delta}{\tilde{\theta} - \delta} \\ &\quad + -v^2 k_2(\tilde{\theta} - \delta) \end{aligned} \quad (70)$$

is the sum of two terms which tend to zero and third term which is uniformly continuous. As $v^2(\tilde{\theta} - \delta)$ tends to zero, the extension of Barbalat's lemma tells us that $\frac{\dot{\tilde{\theta}} - \dot{\delta}}{v^2(\tilde{\theta} - \delta)}$ also tends to zero which implies that the term $\gamma y_1 v^3 \frac{\sin \tilde{\theta} - \sin \delta}{\tilde{\theta} - \delta} \rightarrow 0$. The linear velocity v does not tend to zero when t tends to infinity and γ is constant. The expression $\frac{\sin \tilde{\theta} - \sin \delta}{\tilde{\theta} - \delta}$ could be rewritten in the following way

$$\frac{\sin \tilde{\theta} - \sin \delta}{\tilde{\theta} - \delta} = \frac{\sin \frac{\tilde{\theta} - \delta}{2} \cos \frac{\tilde{\theta} + \delta}{2}}{\frac{\tilde{\theta} - \delta}{2}} \quad (71)$$

which for $(\tilde{\theta} - \delta) \rightarrow 0$ tends to $\cos \delta$. If only δ does not tend to $\frac{k\pi}{2}$, the the whole term $\frac{\sin \tilde{\theta} - \sin \delta}{\tilde{\theta} - \delta}$ does not tend to zero. Under this assumption y_1 tends to zero. This implies that V given by the Eq. 67 also tends to zero. From the assumption that $\delta(0, v) = 0$ and the convergence of y_1 to zero, $\delta \rightarrow 0$ which leads to $\tilde{\theta} \rightarrow 0$. □

6 Dynamic Control

The exact linearisation algorithm would be applied for a task of dynamic control of the unicycle. To the model (10) the following control is plugged

$$\tau = (\mathbf{B}^*(\mathbf{q}_m))^{-1} \mathbf{M}^*(\mathbf{q}_m) \omega_1, \quad (72)$$

where ω_1 is a new input for the linearised system. Then the system (10) with a closed-loop of the feedback signal (72) has a fom

$$\dot{\eta} = \omega_1. \quad (73)$$

ω_1 is chosen to be equal to

$$\omega_1 = \dot{\eta}_r - \mathbf{K}_m(\eta - \eta_r) \quad (74)$$

with $\mathbf{K}_m = \mathbf{K}_m^T > 0$. Then the equation describing the dynamics of errors

$$\dot{\mathbf{e}}_\eta + \mathbf{K}_m \mathbf{e}_\eta = 0, \quad \mathbf{e}_\eta = \eta - \eta_r \quad (75)$$

is asymptotically stable.

7 Simulations

The simulations were carried out to illustrate the behaviour of the wheeled mobile robot of

(2,0) type with the presented controllers and the dynamic model of the robot was simulated. The initial position of the platform was equal to $(x_0, y_0, \theta_0) = (1.5, 0, \frac{\pi}{3})$ for all algorithms except Soetanto-Lapierre-Pascoal algorithm and $(x_0, y_0, \theta_0) = (5, 1, \frac{\pi}{3})$ for Soetanto-Lapierre-Pascoal algorithm. The desired path was the circle described by the equations

$$x(s) = R \cos\left(\frac{s}{R}\right),$$

$$y(s) = R \sin\left(\frac{s}{R}\right),$$

where $R = 2$ m. The simulation time was set for 25 s for all algorithms except Soetanto-Lapierre-Pascoal algorithm and for 50 s for SLP algorithm.

Simulations were carried out in ideal conditions, i.e. without modelling any kind of a distortion, and in the case when the robot's orientation was distorted by a Gaussian white noise with a variance σ^2 . For the wheeled mobile robot of (2,0) type the parameter that has the biggest influence on the control is the orientation—this is why θ was chosen to be noised. It was assumed that $\sigma = 0.1$ which may be treated as the total measurement inaccuracy of devices at the level of approximately ± 5 degrees. Separate simulations with the white noise added to velocities of wheels to verify the influence of a motor quality on the realization of the path following task by the presented algorithms were carried out as well. In that case a noise variance was chosen to be equal to 0.1. To illustrate the influence of an inertia on the performance of the different control strategies in the perturbed cases two graphs have been combined together—the one when only the kinematic model was simulated and the second when the dynamical model was simulated.

Some additional simulations which take into account velocity constraints were carried out as well. The velocity constraints imposed on the robot are chosen as

$$-1 \leq v \leq 1 \text{ [m/s]},$$

$$-\frac{\pi}{5} \leq \omega \leq \frac{\pi}{5} \text{ [rad/s]}.$$

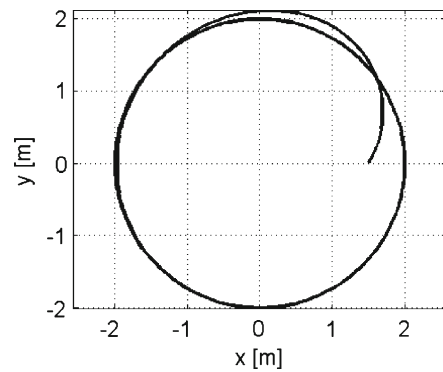


Fig. 4 The path following for the unicycle, XY plot (Samson algorithm, ideal conditions)

Parameters of the kinematic controllers were set to the values presented below:

- Samson algorithm: $k_2 = k_3 = 1, v = 1,$
- discontinuous algorithm, first proposal: $k_1 = k_2 = 1, v_c = 1,$
- discontinuous algorithm, second proposal: $k_1 = k_2 = k_3 = 1, v_c = 1,$
- Morin-Samson algorithm: $k_2 = 10, k_3 = 100, u_1 = 1,$
- Soetanto-Lapierre-Pascoal algorithm: $\gamma = 1, k_1 = 10, k_2 = 1000, \delta = -\text{sign}(v)\theta_a \tanh y_1, \theta_a = \frac{\pi}{4}$

and the matrix K_m in the dynamic controller is equal to $50 \cdot I_2$. The results of the simulations have been presented in Figs. 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43.

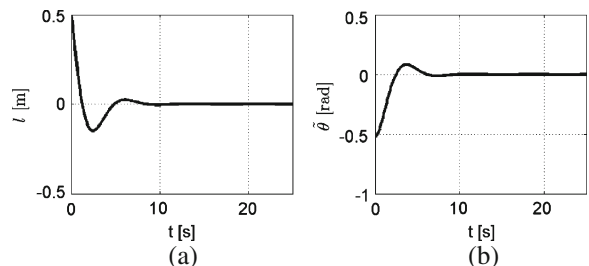


Fig. 5 The path following for the unicycle (Samson algorithm, ideal conditions): **a** the distance error l , **b** the orientation error $\hat{\theta}$

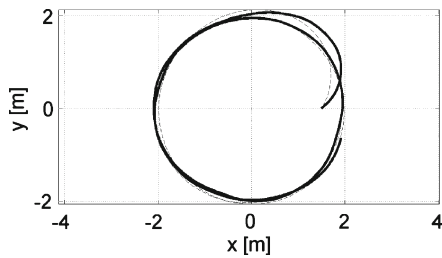


Fig. 6 The path following for the unicycle, XY plot (Samson algorithm with a noised orientation, *thin dashed line*—only kinematic model simulated, *thick solid line*—dynamical model simulated)

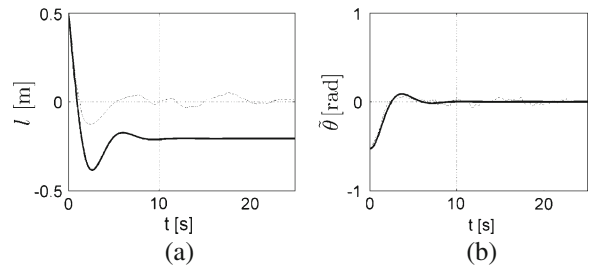


Fig. 9 The path following for the unicycle (Samson algorithm with noised wheels velocities, *thin dashed line*—only kinematic model simulated, *thick solid line*—dynamical model simulated): **a** the distance error l , **b** the orientation error $\hat{\theta}$

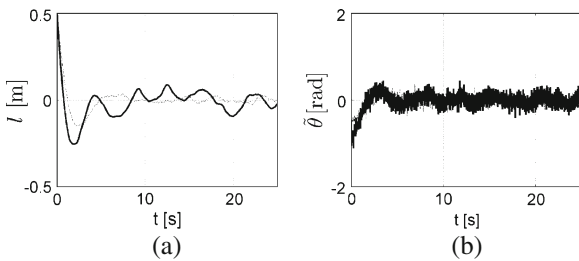


Fig. 7 The path following for the unicycle (Samson algorithm with a noised orientation, *thin dashed line*—only kinematic model simulated, *thick solid line*—dynamical model simulated): **a** the distance error l , **b** the orientation error $\hat{\theta}$

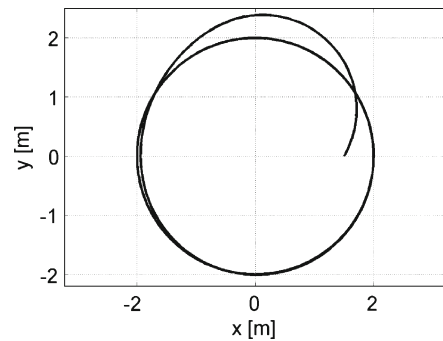


Fig. 10 The path following for the unicycle, XY plot (Samson algorithm, velocity constraints)

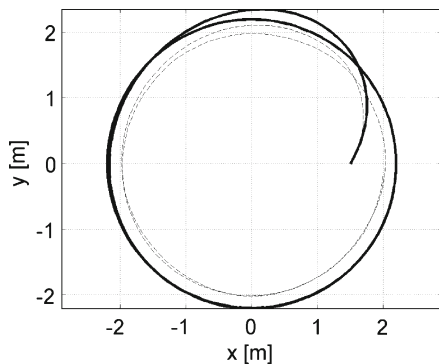


Fig. 8 The path following for the unicycle, XY plot (Samson algorithm with noised wheels velocities, *thin dashed line*—only kinematic model simulated, *thick solid line*—dynamical model simulated)

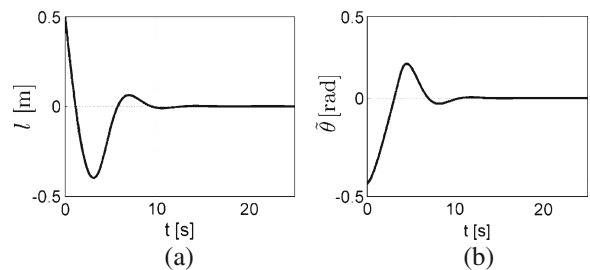


Fig. 11 The path following for the unicycle (Samson algorithm, velocity **a** the distance error l , **b** the orientation error $\hat{\theta}$

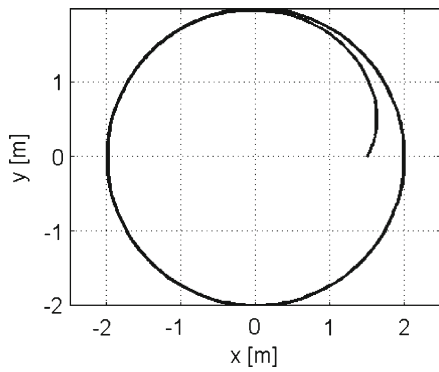


Fig. 12 The path following for the unicycle, XY plot (discontinuous control algorithm I, ideal conditions)

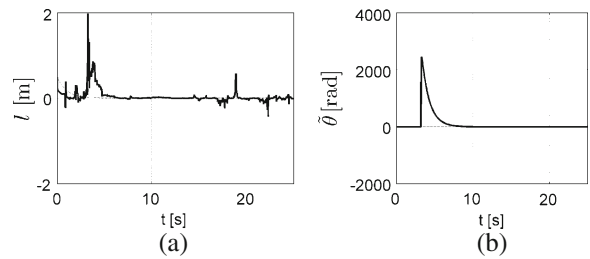


Fig. 15 The path following for the unicycle (discontinuous control algorithm I with a noised orientation, *thin dashed line*—only kinematic model simulated, *thick solid line*—dynamical model simulated): **a** the distance error l , **b** the orientation error $\tilde{\theta}$

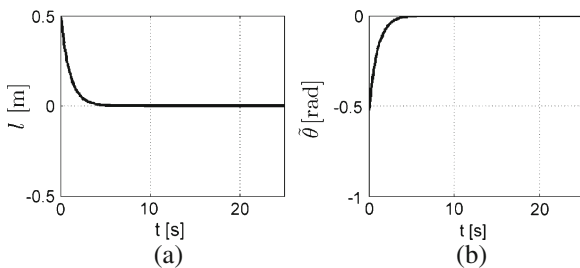


Fig. 13 The path following for the unicycle (discontinuous control algorithm I, ideal conditions): **a** the distance error l , **b** the orientation error θ

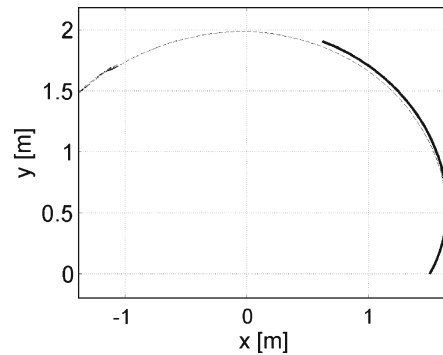


Fig. 16 The path following for the unicycle, XY plot (discontinuous control algorithm I with noised wheels velocities, *thin dashed line*—only kinematic model simulated, *thick solid line*—dynamical model simulated)

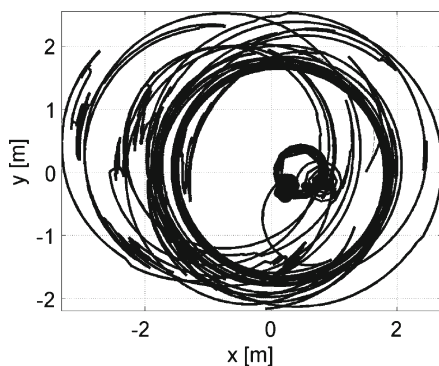


Fig. 14 The path following for the unicycle, XY plot (discontinuous control algorithm I with a noised orientation, *thin dashed line*—only kinematic model simulated, *thick solid line*—dynamical model simulated)

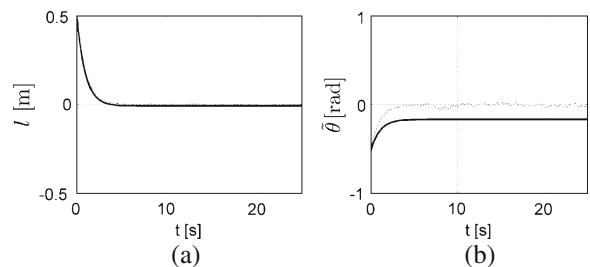


Fig. 17 The path following for the unicycle (discontinuous control algorithm I with noised wheels velocities, *thin dashed line*—only kinematic model simulated, *thick solid line*—dynamical model simulated): **a** the distance error l , **b** the orientation error $\tilde{\theta}$

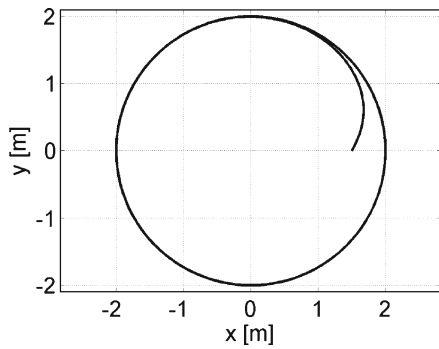


Fig. 18 The path following for the unicycle, XY plot (discontinuous control algorithm I, velocity constraints)

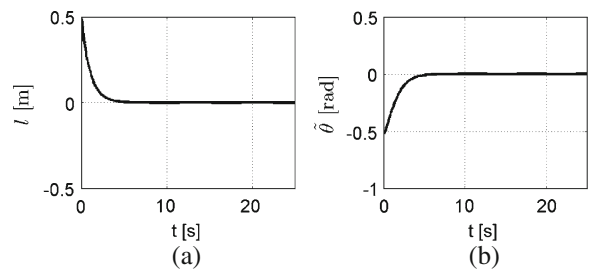


Fig. 21 The path following for the unicycle (discontinuous control algorithm II, ideal conditions): **a** the distance error l , **b** the orientation error $\tilde{\theta}$

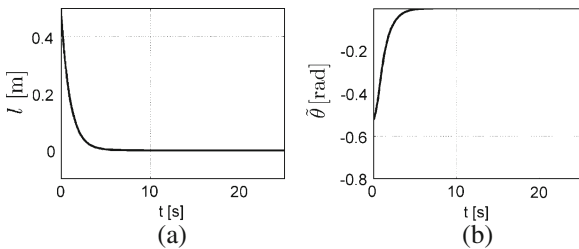


Fig. 19 The path following for the unicycle (discontinuous control algorithm I, velocity constraints): **a** the distance error l , **b** the orientation error $\tilde{\theta}$

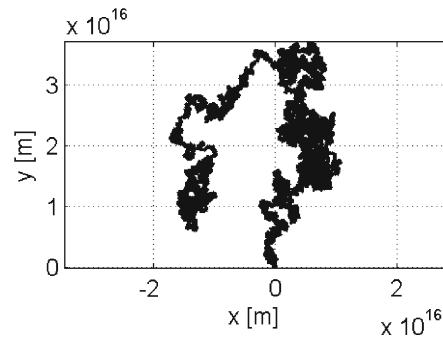


Fig. 22 The path following for the unicycle, XY plot (discontinuous control algorithm II with a noised orientation, *thin dashed line*—only kinematic model simulated, *thick solid line*—dynamical model simulated)

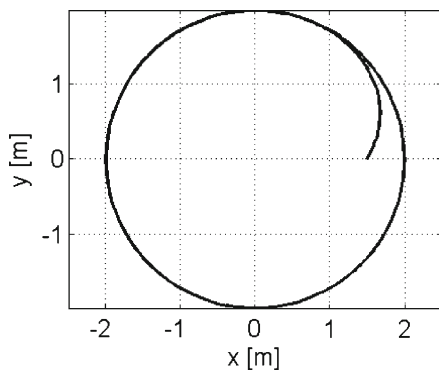


Fig. 20 The path following for the unicycle, XY plot (discontinuous control algorithm II, ideal conditions)

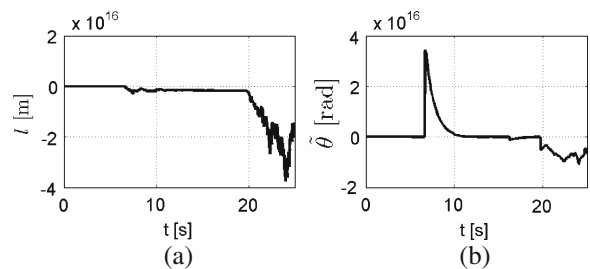


Fig. 23 The path following for the unicycle (discontinuous control algorithm II with a noised orientation, *thin dashed line*—only kinematic model simulated, *thick solid line*—dynamical model simulated): **a** the distance error l , **b** the orientation error $\tilde{\theta}$

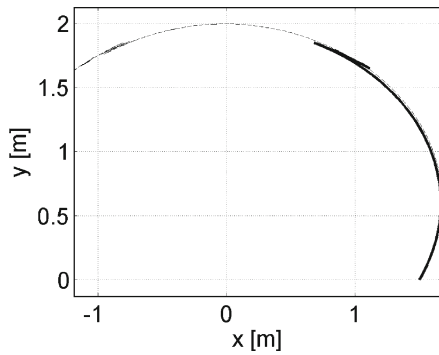


Fig. 24 The path following for the unicycle, XY plot (discontinuous control algorithm II with noised wheels velocities, *thin dashed line*—only kinematic model simulated, *thick solid line*—dynamical model simulated)

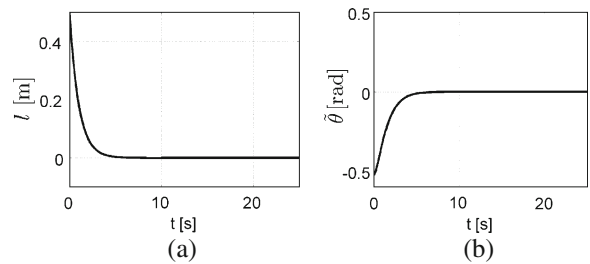


Fig. 27 The path following for the unicycle (discontinuous control algorithm II, velocity constraints): **a** the distance error l , **b** the orientation error $\tilde{\theta}$

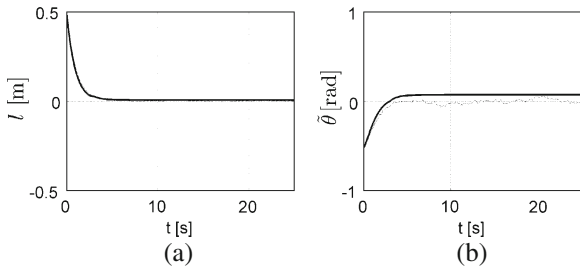


Fig. 25 The path following for the unicycle (discontinuous control algorithm II with noised wheels velocities, *thin dashed line*—only kinematic model simulated, *thick solid line*—dynamical model simulated): **a** the distance error l , **b** the orientation error $\tilde{\theta}$

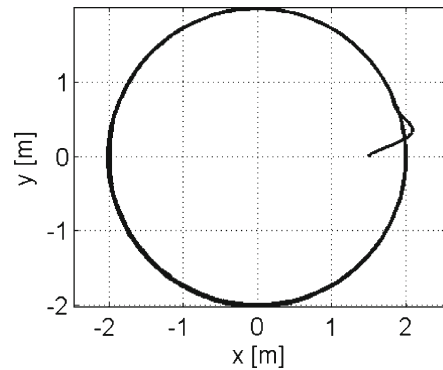


Fig. 28 The path following for the unicycle, XY plot (Morin-Samson algorithm, ideal conditions)

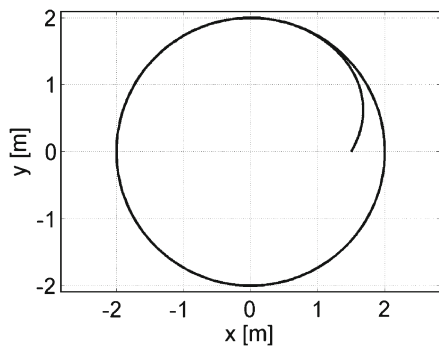


Fig. 26 The path following for the unicycle, XY plot (discontinuous control algorithm II, velocity constraints)

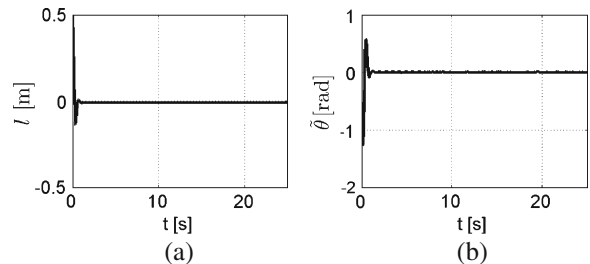


Fig. 29 The path following for the unicycle (Morin-Samson algorithm, ideal conditions): **a** the distance error l , **b** the orientation error $\tilde{\theta}$

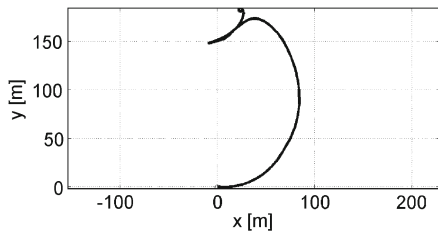


Fig. 30 The path following for the unicycle, XY plot (Morin-Samson algorithm with a noised orientation, *thin dashed line*—only kinematic model simulated, *thick solid line*—dynamical model simulated)

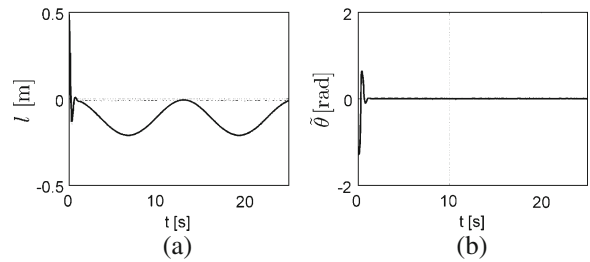


Fig. 33 The path following for the unicycle (Morin-Samson algorithm with noised wheels velocities, *thin dashed line*—only kinematic model simulated, *thick solid line*—dynamical model simulated): **a** the distance error l , **b** the orientation error $\tilde{\theta}$

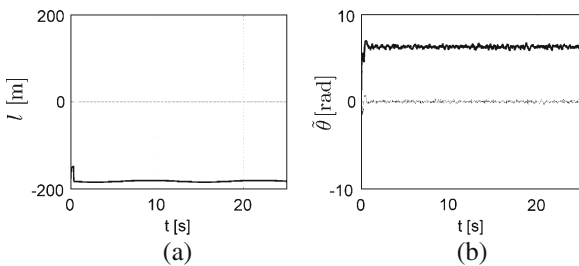


Fig. 31 The path following for the unicycle (Morin-Samson algorithm with a noised orientation, *thin dashed line*—only kinematic model simulated, *thick solid line*—dynamical model simulated): **a** the distance error l , **b** the orientation error $\tilde{\theta}$

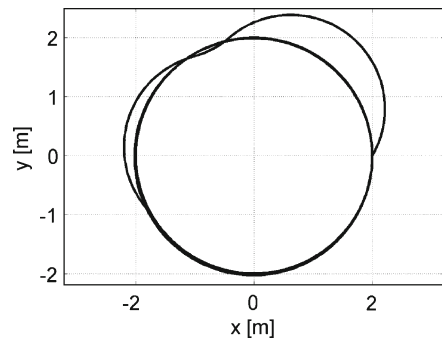


Fig. 34 The path following for the unicycle, XY plot (Morin-Samson algorithm, velocity constraints)

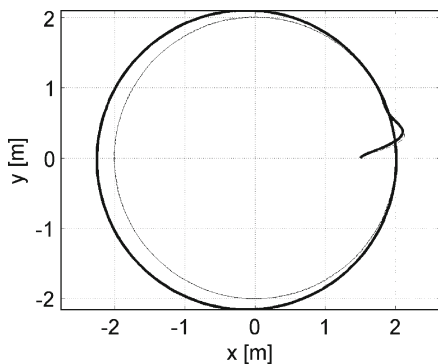


Fig. 32 The path following for the unicycle, XY plot (Morin-Samson algorithm with noised wheels velocities, *thin dashed line*—only kinematic model simulated, *thick solid line*—dynamical model simulated)

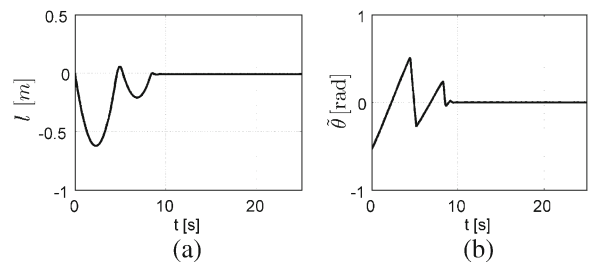


Fig. 35 The path following for the unicycle (Morin-Samson algorithm, velocity constraints): **a** the distance error l , **b** the orientation error $\tilde{\theta}$

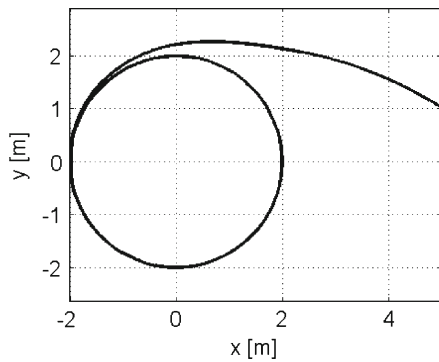


Fig. 36 The path following for the unicycle, XY plot (Soetanto-Lapierre-Pascoal algorithm, ideal conditions)

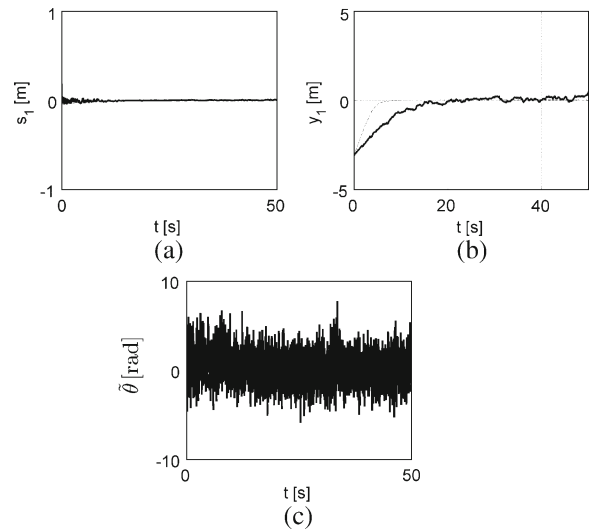


Fig. 39 The path following for the unicycle (Soetanto-Lapierre-Pascoal algorithm with a noisy orientation, *thin dashed line*—only kinematic model simulated, *thick solid line*—dynamical model simulated): **a** the distance error s_1 , **b** the distance error y_1 , **c** the orientation error $\tilde{\theta}$

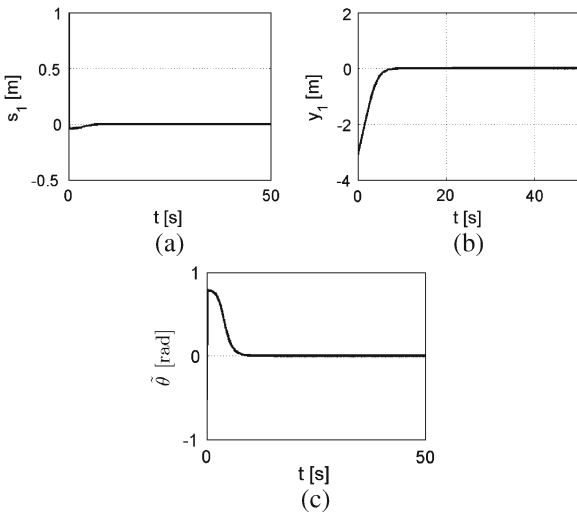


Fig. 37 The path following for the unicycle (Soetanto-Lapierre-Pascoal algorithm, ideal conditions): **a** the distance error s_1 , **b** the distance error y_1 , **c** the orientation error $\tilde{\theta}$

Samson algorithm seems to be the most noise-robust among presented algorithms. For the noisy orientation the path circled by the robot in XY plane is almost the same as the desired path—it is shifted a bit. Little worse results were obtained for the noisy velocity of wheels case because the robot followed a circle-shaped path with a bigger radius than the radius of the desired path.

The discontinuous control algorithms work properly in ideal conditions and are characterized by a fast convergence of path following errors to zero without an occurrence of oscillations in the

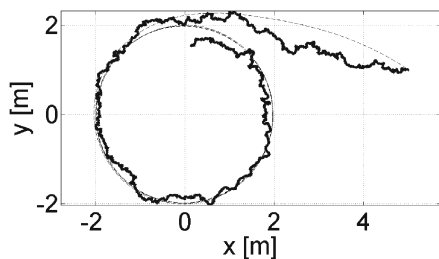


Fig. 38 The path following for the unicycle, XY plot (Soetanto-Lapierre-Pascoal algorithm with a noisy orientation, *thin dashed line*—only kinematic model simulated, *thick solid line*—dynamical model simulated)

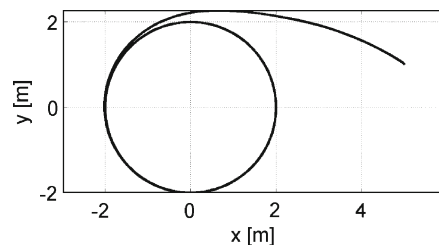


Fig. 40 The path following for the unicycle, XY plot (Soetanto-Lapierre-Pascoal algorithm with noisy wheels velocities, *thin dashed line*—only kinematic model simulated, *thick solid line*—dynamical model simulated)

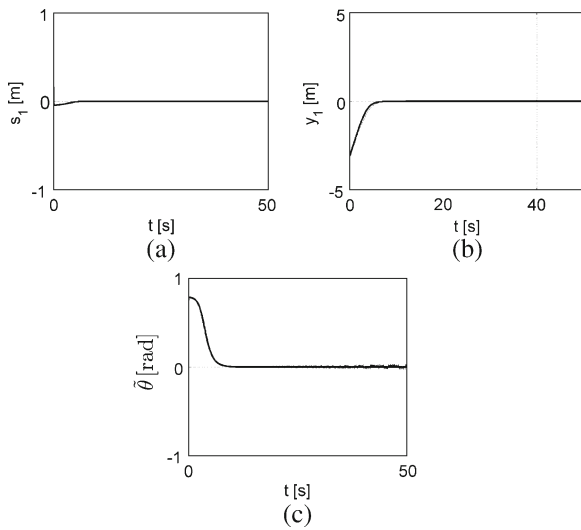


Fig. 41 The path following for the unicycle (Soetanto-Lapierre-Pascoal algorithm with noised wheels velocities, *thin dashed line*—only kinematic model simulated, *thick solid line*—dynamical model simulated): **a** the distance error s_1 , **b** the distance error y_1 , **c** the orientation error $\tilde{\theta}$

courses of the variables l and $\tilde{\theta}$ during the transient phase. However they are not noise-robust, even for a relatively small value of noise variance. It means that they cannot be used in practical applications where one has to take into account the probabilistic nature of the physical world. Thus it is recommended to design rather continuous control algorithms for the path following task instead of discontinuous ones.

Soetanto-Lapierre-Pascoal algorithm does not perform sufficiently well for the noised orienta-

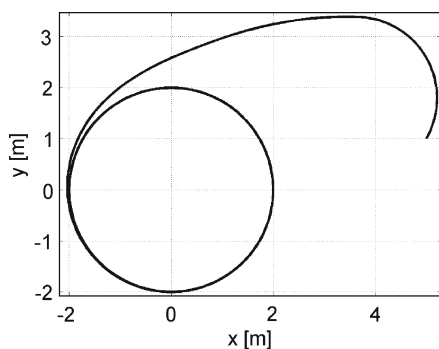


Fig. 42 The path following for the unicycle, XY plot (Soetanto-Lapierre-Pascoal algorithm, velocity constraints)

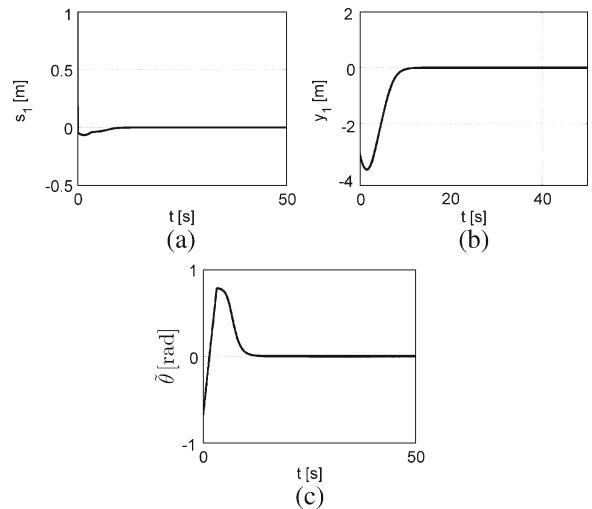


Fig. 43 The path following for the unicycle (Soetanto-Lapierre-Pascoal algorithm, velocity constraints): **a** the distance error s_1 , **b** the distance error y_1 , **c** the orientation error $\tilde{\theta}$

tion case. The robot does not move too far from the path, however it wanders around the path which is an undesirable behaviour.

The main difference between Morin-Samson and Samson algorithms is that before Morin-Samson algorithm could be used, two nonlinear transformations of the robot’s kinematics have to be computed, while for Samson algorithm one transformation is enough. Morin-Samson algorithm does not work properly for the noised orientation. However when one detaches the object’s dynamics and controls the robot only at the kinematic level, Morin-Samson algorithm works just a little worse than Samson algorithm. That issue has to be deeply investigated in the future.

What is more, the path following task is realized correctly by the considered wheeled mobile robot when the presented algorithms are modified by adding velocity constraints. Introducing such constraints slows down the convergence of path tracking errors to zero, but does not disrupt the realization of the task.

8 Conclusions

In the paper different kinematic path following algorithms for the wheeled mobile robot of (2,0)

type have been presented. For the two of the presented algorithms the proofs of their convergence have been performed.

The choice of the path parametrization is of a great importance in the path following task. The significant constraint of the parametrization which uses the Serret-Frenet frame with an orthogonal projection of the robot on the desired path is its local character. Therefore if the initial position of the robot is too far from the path, the robot has to at first get closer to the desired path before it can start the realization of the path following task. In contrary the approach using the Serret-Frenet frame with an non-orthogonal projection does not have such a drawback but it has a bigger amount of path following errors to be brought to zero and the designing of the control law may be more difficult.

The study showed that the impact of the noise on the realization of the path following task might be huge. This is why when using model-based control one has to measure the robot's posture as precisely as possible and ensure as high motor's quality as possible. The article is not comprehensive and only a selected group of algorithms which are based on the Serret-Frenet frame attached to a path was tested. In the future the other types of path following algorithms could be investigated taking into account their noise robustness. The extension of this work could be a study of the robustness properties of path following algorithms which deal with obstacles avoidance.

Acknowledgement This work was supported by the fellowship co-financed by European Union within European Social Fund.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

- Micaelli, A., Samson, C.: Trajectory tracking for unicycle-type and two-steering-wheels mobile robots. Technical Report No. 2097, INRIA, Sophia-Antipolis, France (1993)
- Soetanto, D., Lapierre, L., Pascoal, A.: Adaptive, non-singular path-following control of dynamic wheeled robots. *Proc. IEEE Conf. Decis. Control* **2**, 1765–1770 (2003)
- Morro, A., Sgorbissa, A., Zaccaria, R.: Path following for unicycle robots with an arbitrary path curvature. *IEEE Trans. Robot.* **27**(5), 1016–1023 (2011)
- Morin, P., Samson, C.: Motion control of wheeled mobile robots. In: Siciliano, B., Khatib, O.: (eds.) *Handbook of Robotics*. Springer (2008)
- Soueres, P., Hamel, T., Cadenat, V.: A path following controller for wheeled robots which allows to avoid obstacles during transition phase. *Proc. IEEE Int. Conf. Robot. Autom.* **2**, 1269–1274 (1998)
- Lapierre, L., Zapata, R., Lepinay, P.: Simultaneous path following and obstacle avoidance control of a unicycle-type robot. In: *IEEE International Conference on Robotics and Automation 2007*, pp. 2617–2622 (2007)
- Campani, M., Capezio, F., Rebor, A., Sgorbissa, A., Zaccaria, R.: A minimalist approach to path following among unknown obstacles. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS) 2010*, pp. 3604–3610 (2010)
- Consolini, L., Morbidi, F., Prattichizzo, D., Tosques, M.: Leader-follower formation control of nonholonomic mobile robots with input constraints. *Automatica*, **44**(5), 1343–1349 (2008)
- Ghommam, J., Mehrjerdi, H., Saad, M., Mnif, F.: Formation path following control of unicycle-type mobile robots. *Robot. Auton. Syst.* **58**(5), 727–736 (2010)
- Mazur, A.: Hybrid adaptive control laws solving a path following problem for nonholonomic mobile manipulators. *Int. J. Control* **77**(15), 1297–1306 (2004)
- Samson, C.: Path following and time-varying feedback stabilization of a wheeled mobile robots. In: *Proceedings of the IEEE International Conference on Advanced Robotics and Computer Vision*, pp. 1.1–1.5 (1992)
- Campion, G., Bastin, G., d'Andréa Novel, B.: Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Trans. Robot. Autom.* **12**(5), 47–61 (1996)
- Campion, G., Chung, W.: Wheeled robots. In: Siciliano, B., Khatib, O. (eds.) *Handbook of Robotics*. Springer (2008)
- Samson, C.: Control of chained systems—application to path following and time-varying point-stabilization of mobile robots. *IEEE Trans. Autom. Control* **40**(1), 64–77 (1995)
- Krstić, M., Kanellakopoulos, I., Kokotović, P.: *Nonlinear and Adaptive Control Design*. Wiley, Boca Raton (1994)
- Canudas de Wit, C., Siciliano, B., Bastin, G.: *Theory of Robot Control*. Springer, London (1996)