



# Rapid Navigation Function Control for Two-Wheeled Mobile Robots

Wojciech Kowalczyk<sup>1</sup>

Received: 7 November 2017 / Accepted: 22 May 2018 / Published online: 14 June 2018  
© The Author(s) 2018

## Abstract

This paper presents a kinematic controller for a differentially driven mobile robot. The controller is based on the navigation function (NF) concept that guarantees goal achievement from almost all initial states. Slow convergence in some cases is a significant disadvantage of this approach, especially when narrow passages exist in the environment and/or specific values of design parameters are set. The main reason of this phenomenon is that the velocity control strongly depends on the slope of the NF. The algorithm proposed in this paper is based on a method introduced in Urakubo (Nonlin. Dyn. **81**(3): 1475–1487 2015), that extends NF to nonholonomic mobile platforms and allows stabilizing not only the position of robots but also their orientation. This algorithm is used as a reference in experimental performance comparison. In the new algorithm, the gradient of the NF is used to generate motion direction but the velocity is computed as a function of position and orientation errors. This approach results in much better state converge. Analysis of the convergence shows how the location of the eigenvalues of linearized system affects time of goal achievement. The paper describes saddle point detection and avoidance methodology and presents their experimental verification. It also shows what happens in practice if initial position is located exactly in the saddle point and its detection/avoidance procedures are turned off.

**Keywords** Mobile robot control · Navigation function · Set point control · Obstacle avoidance · Saddle point detection

## 1 Introduction

The breakthrough idea to use artificial potential fields to control manipulators and mobile robots was introduced by Khatib [4] in 1986. In this approach both attraction to the goal and repulsion from the obstacles are negated gradients of the artificial potential functions. The original solution had one major drawback: local minima may occur depending on the configuration of obstacles in the task space and inadequate potential functions choice. The problem is caused by the fact that attracting and repulsive components of the control are combined by simple addition. In some states which are far from the target location the addition of attracting and repulsive vectors may result in zero vector.

This causes the robot to get stuck far from the target location.

The solution of this problem was presented in several articles published at the turn of the 1980s and 1990s by Rimon and Koditschek. They introduced *navigation function* (NF) [12–14] that solves the problem of local minima. In this method the addition used to combine attraction and repulsion was replaced by division. In [12], the algorithm for the generalized sphere-shaped obstacles (sphere worlds) was described. It can be directly used to control mobile robots without nonholonomic constraints. In the next publications this approach was expanded to include more complex environments [13, 14] e.g. star-shaped-obstacles (star worlds) and trees-of-star-shaped-obstacles. All these cases assume that the shapes enclosing obstacles can be set by the analytic functions that fulfill given conditions.

A proposition of the NF extension to nonholonomic mobile robot was presented by Urakubo [21] in 2004 and later extended [20]. The Author proposed that state variable representing orientation be included in the NF that results in a three dimensional gradient vector for a planar case. The

---

✉ Wojciech Kowalczyk  
wojciech.kowalczyk@put.poznan.pl

<sup>1</sup> Institute of Automation and Robotics, Faculty of Computing,  
Poznan University of Technology, Piotrowo 3A,  
Poznan, Poland

gradient cannot be used directly to control the robot like in [13, 14]. It is used as a reference for a nonlinear controller that generates linear and angular velocities applied to the mobile platform. In contrast to the Rimon and Koditschek algorithm the orientation also converges to the desired value. Convergence proof was enclosed in the paper.

The author of this paper together with its co-authors conducted extensive tests of this algorithm including simulations for sphere worlds [10], star worlds [6], and experiments for sphere worlds [7–9].

The beginning of the 21st century saw publication of many propositions of extension of the NF approach for the groups of mobile robots [2, 15, 16]. The publications addressed the problem of collision avoidance in multi-agent robotic systems. The first paper introduces multi-robot navigation function (MRNF). The second and third use prioritization to solve conflicts in the case of concurrent goals. In the [11] a new NF is proposed that can be computed locally, and only the knowledge of the target and obstacles in the robots' neighborhood is required. In [17, 18] NF is used to control aircrafts.

This work is motivated by the poor time performance of known NF control methods. This is caused by the direct dependency of the velocity control on the module of the NF gradient. The gradient depends on many conditions, including a number of obstacles, their relative positions, obstacle functions that avoid collisions, goal location, and some design parameters that must be tuned to avoid local minima. As a result, the module of the gradient vector is not usually the best choice for the velocity control.

The main contribution of this paper is to propose modification of the method described in [20] that makes the convergence to desired coordinates much faster. The convergence of closed-loop system is analyzed by linearization in the neighborhood of the critical point. It is shown that by properly selecting the new design parameters, the real parts of eigenvalues of the linearized system can be easily moved to greater negative values. This results in faster convergence of the system to the desired values. The effectiveness of the proposed method is verified by experiments. In addition, experimental verification of the NF saddle point detection and avoidance is included. This kind of results has never been published previously to the best of the author's knowledge.

Section 2 introduces the model of the differentially driven mobile platform and the environment. Section 3 presents control algorithm. In Section 4 convergence proof is given. Section 5 describes method of saddle point detection and avoidance. Section 6 presents experimental test-bed and results of experiments that illustrate how convergence of the new algorithm has improved in comparison to the reference algorithm. Detection of the

saddle point and algorithm that drives the robot out of this point is presented. The final section includes conclusion.

## 2 Model of the System

A kinematic model of a differentially driven mobile robot is given by the following equation:

$$\dot{\mathbf{q}} = B(\mathbf{q})\mathbf{u} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u} \quad (1)$$

where vector  $\mathbf{q}^\top \triangleq [x \ y \ \theta]$  denotes the pose,  $x$ ,  $y$  are position coordinates and  $\theta$  is orientation of the platform expressed in a global, fixed coordinate frame. Vector  $\mathbf{u}^\top \triangleq [v \ \omega]$  is the control vector with  $v$  denoting linear velocity and  $\omega$  denoting angular velocity of the robot.

The obstacles are circle-shaped and modeled with obstacle function:

$$\beta_i \triangleq \|\mathbf{r} - \mathbf{p}_i\|^2 - \rho_i^2. \quad (2)$$

where  $\rho_i$  is the radius of the  $i$ -th obstacle ( $i = 1, \dots, N$ ),  $\mathbf{p}_i$  represents its center,  $\mathbf{r} = [x \ y]^\top$  is the location of the robot and  $N$  is the number of obstacles. The obstacle function has a zero value at the boundary of the obstacle and increases if the distance to the obstacle grows. It must be at least twice differentiable.

The task space has an external boundary described by the additional obstacle function that can be considered as representing one more obstacle denoted in the further equations with index zero:

$$\beta_0 \triangleq \rho_0^2 - \|\mathbf{r} - \mathbf{p}_0\|^2, \quad (3)$$

where  $\rho_0$  is the radius of the task space and  $\mathbf{p}_0$  is the center of the task space (usually the origin of the global coordinate frame).

## 3 Control Algorithm

The goal is to stabilize robot in the origin with orientation equal to zero. The total NF is as follows:

$$V \triangleq \frac{C}{(C^\kappa + \beta)^{\frac{1}{\kappa}}}, \quad (4)$$

where  $\kappa$  is a positive, constant design parameter and

$$C \triangleq \|\mathbf{r}\|^2 + \theta^2 \frac{k_w}{k_w + \|\mathbf{r}\|^2}. \quad (5)$$

Design parameter  $k_w$  in Eq. 5 is a positive constant that allows tuning the function representing the importance of

the orientation with respect to the Euclidean distance to the target.

A common obstacle function is obtained by the aggregation of individual obstacle functions with the following equation:

$$\beta \triangleq \prod_{i=0}^N \beta_i. \tag{6}$$

The internal obstacles are represented by  $i = 1, \dots, N$  while index zero denotes the task space boundary.

The new control algorithm is as follows:

$$\mathbf{u} \triangleq - \left\{ a \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + b \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \right\} \gamma, \tag{7}$$

where  $a$  is a positive, constant design parameter,

$$b \triangleq -\bar{b} \frac{\mathbf{L}^T \nabla V}{h(g)} \tag{8}$$

and

$$\gamma = \frac{\mathbf{B}^T \nabla V (\|\mathbf{q}^T\| + \epsilon_e)}{(g^2 + \epsilon_f)}. \tag{9}$$

Design parameters  $\bar{b}$ ,  $\epsilon_e$ ,  $\epsilon_f$  in Eqs. 8 and 9 are positive constants,  $\mathbf{L}^T \triangleq [\sin \theta \quad -\cos \theta \quad 0]$  and  $g \triangleq \|\mathbf{B}^T \nabla V\| \geq 0$ . Function  $h(g)$  is as follows:

$$h(g) \triangleq g^2 + \epsilon_g \sqrt{g}, \tag{10}$$

where  $\epsilon_g$  is a small positive constant. Note that  $h(g)$  is non-decreasing function and  $h(0) = 0$ .

Finally,  $\nabla V$  denotes the gradient of the navigation function with respect to variables  $x$ ,  $y$  and  $\theta$ . Regardless of the number of obstacles, the gradient can be obtained in the analytical form as:

$$\nabla V = \frac{\nabla C (C^\kappa + \beta)^{\frac{1}{\kappa}} - \frac{C}{\kappa} (C^\kappa + \beta)^{(\frac{1}{\kappa}-1)} (\kappa C^{\kappa-1} \nabla C + \nabla \beta)}{(C^\kappa + \beta)^{\frac{2}{\kappa}}}, \tag{11}$$

where

$$\nabla C = \begin{bmatrix} \frac{\partial C}{\partial x} \\ \frac{\partial C}{\partial y} \\ \frac{\partial C}{\partial \theta} \end{bmatrix}^T = \begin{bmatrix} 2x(1 - \frac{k_w \theta^2}{(k_w + \|\mathbf{r}\|^2)^2}) \\ 2y(1 - \frac{k_w \theta^2}{(k_w + \|\mathbf{r}\|^2)^2}) \\ 2\theta \frac{k_w}{k_w + \|\mathbf{r}\|^2} \end{bmatrix}^T \tag{12}$$

and

$$\nabla \beta = \sum_{i=0}^N \left\{ \frac{\partial \beta_i}{\partial \mathbf{q}} \prod_{j=0, j \neq i}^N \beta_j \right\}. \tag{13}$$

As noted in [14] all undesired local minima of the NF (4) disappear as the parameter  $\kappa$  is increased. An algorithm for automatically tuning  $\kappa$  for sphere worlds is presented in [3]. The tuning parameter must satisfy a lower bound to

ensure convergence to the desired value. For the sufficiently high value of the parameter  $\kappa$  NF (4) has a critical point associated with each isolated obstacle, the saddle point.  $V$  has no other critical points other than these points and the goal.

Saddle points are unstable equilibrium points. Urakubo et al. [21] proposes a special control procedure for saddle point avoidance. It uses a time varying trigonometric function to push the robot away from the unstable equilibrium point. In Section 5, this method is recalled together with a detailed description of saddle point detection. Section 6 shows the results of experimental verification.

The NF  $V$  (4) has the following properties:

- $V$  is at least twice continuously differentiable and all elements of  $\nabla V$  and Hessian matrix  $\partial(\nabla V)/\partial \mathbf{q}$  are bounded
- $V \geq 0$ , and  $V = 0 \Leftrightarrow \mathbf{q} = 0$
- the critical points of  $V$  are a minimum at  $\mathbf{q} = 0$  and saddle points associated with the obstacles;  $V$  does not have any other critical points
- $V$  is a Morse function - the Hessian matrix is nonsingular at the critical points
- $V \leq V_b$  where  $V_b$  is the value of the NF at the obstacle boundary
- $\partial V/\partial \theta = 0 \Leftrightarrow \theta = 0 \Rightarrow \partial^2 V/\partial x \partial \theta = \partial^2 V/\partial y \partial \theta = 0$  and  $\partial^2 V/\partial \theta^2 > 0$

### 4 Convergence of the Closed-Loop System

This section presents analysis of the system convergence. It only includes steps that differ from the analysis shown in [20]. Substituting (7) with (1) one obtains:

$$\dot{\mathbf{q}} = -\mathbf{B}(\mathbf{q}) \left\{ a \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + b \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \right\} \gamma. \tag{14}$$

As the above equation is not differentiable it is modified as follows to get a differentiable one:

$$\dot{\mathbf{q}} = -\mathbf{B}(\mathbf{q}) \left\{ a \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \tanh\left(\frac{h(g)}{\epsilon}\right) b \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \right\} \gamma. \tag{15}$$

Note that the differentiable model of the closed-loop system (15) is needed for the further analysis only. It is not used in the implementation.

For  $\epsilon \rightarrow 0$  (15) approaches (14). By linearizing (15) in the neighborhood of equilibrium point  $\mathbf{q} = \mathbf{q}_0 \in \mathcal{H}$  where set  $\mathcal{H} = \{\mathbf{q} \mid \mathbf{B}^T \nabla V = 0\}$  one obtains

$$\delta \dot{\mathbf{q}} = \mathbf{A}(\mathbf{q}_0) \delta \mathbf{q}. \tag{16}$$

The  $\mathbf{A}(\mathbf{q}_0)$  matrix is computed taking into account that  $\|\mathbf{B}^T \nabla V\| \rightarrow 0$  as  $t \rightarrow \infty$ . This condition can be rewritten

as  $\sqrt{(\frac{\partial V}{\partial x} \cos \theta + \frac{\partial V}{\partial y} \sin \theta)^2 + (\frac{\partial V}{\partial \theta})^2} \rightarrow 0$ . This leads to the conclusion that  $\frac{\partial V}{\partial \theta} \rightarrow 0$  and  $\frac{\partial V}{\partial x} \rightarrow 0$  as the system approaches the equilibrium point. Moreover, assuming that the nominator of the fraction on the right hand side of Eq. 8  $\mathbf{L}^\top \nabla V \rightarrow 0$  as  $t \rightarrow \infty$  (that may be rewritten as follows:  $\frac{\partial V}{\partial x} \sin \theta - \frac{\partial V}{\partial y} \cos \theta \rightarrow 0$ ) and knowing that  $\frac{\partial V}{\partial x} \rightarrow 0, \theta \rightarrow 0$  one can state that  $\frac{\partial V}{\partial y} \rightarrow 0$ . Summarizing, due to the fact that  $\nabla V \rightarrow 0$ , the closed-loop system is asymptotically stable to the equilibrium point ( $\lim_{t \rightarrow \infty} x, y, \theta = 0$ ).

Notice that as the denominator of the fraction on the right hand side of Eq. 8 tends to zero ( $h(g) \rightarrow 0$ ) as  $t \rightarrow \infty$ , the nominator must convergence to zero ( $\mathbf{L}^\top \nabla V \rightarrow 0$ ) faster then the denominator to avoid  $b$  going to infinity and violating controllability of the system.

For the further analysis the control (7) can be rewritten as follows:

$$\mathbf{u} = -(\bar{\gamma} a \bar{\mathbf{I}} + \bar{\gamma} b \bar{\mathbf{J}}) \tag{17}$$

where

$$\bar{\mathbf{I}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} B^\top \nabla V, \quad \bar{\mathbf{J}} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} B^\top \nabla V \tag{18}$$

$$\text{and } \bar{\gamma} = \frac{(\|\mathbf{q}^\top\| + \epsilon_e)}{(g^2 + \epsilon_f)}.$$

The vectors  $\bar{\mathbf{I}}$  and  $\bar{\mathbf{J}}$  can be expressed as follows:

$$\bar{\mathbf{I}} = \begin{bmatrix} \cos \theta \frac{\partial V}{\partial x} + \sin \theta \frac{\partial V}{\partial y} \\ \frac{\partial V}{\partial \theta} \end{bmatrix} \tag{19}$$

and

$$\bar{\mathbf{J}} = \begin{bmatrix} \frac{\partial V}{\partial \theta} \\ -\cos \theta \frac{\partial V}{\partial x} - \sin \theta \frac{\partial V}{\partial y} \end{bmatrix}. \tag{20}$$

Note that vectors  $\bar{\mathbf{I}}$  and  $-\bar{\mathbf{J}}$  are perpendicular because their scalar product is zero:  $\bar{\mathbf{I}} \cdot (-\bar{\mathbf{J}}) = 0$ . It follows that the control is a composition of two perpendicular vectors  $\bar{\mathbf{I}}$  and  $\bar{\mathbf{J}}$  scaled by  $\bar{\gamma} a$  and  $\bar{\gamma} b$ , respectively.

Scalar  $b$  given by Eq. 8 is computed using Lie bracket  $\mathbf{L} = ad_{b_1} b_2$ , where  $b_1$  and  $b_2$  are columns of matrix  $B = [b_1(q) \ b_2(q)]$ . The third vector filed  $ad_{b_1} b_2$  fulfills controllability condition:

$$\det [b_1(q) \ b_2(q) \ ad_{b_1} b_2] = \det \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ \sin \theta & 0 & -\cos \theta \\ 0 & 1 & 0 \end{bmatrix} = 1. \tag{21}$$

Nonlinear control system is controllable.

Assuming that only first part of the control  $\bar{\mathbf{I}}$  is active ( $\bar{\mathbf{J}} = 0$ ) Lyapunov function decreases and stops in the set  $\{x, y, \theta\} \in \mathfrak{R} : \sqrt{(\frac{\partial V}{\partial x} \cos \theta + \frac{\partial V}{\partial y} \sin \theta)^2 + (\frac{\partial V}{\partial \theta})^2} = 0$  that is equivalent to  $\frac{\partial V}{\partial x} = 0$  and  $\frac{\partial V}{\partial \theta} = 0$ . Second component of the control (that contains  $b$ ) makes the effect of the Lie bracket  $\mathbf{L}$  that is responsible for convergence in the

$y$  coordinate. If both components of the control are active then  $V$  is decreasing and  $\frac{\partial V}{\partial q} = 0$  because the system is controllable. The condition  $\nabla V = 0$  leads to the conclusion that  $\dot{\mathbf{q}} = 0$  because the system is in set  $g = 0$ . It minds that  $x = 0, y = 0$  and  $\theta = 0$ . These intuitive arguments are extended to formal stability analysis that is presented in [20].

The matrix  $A(\mathbf{q}_0)$  is given by the equation:

$$A(\mathbf{q}_0) = \frac{\epsilon_e}{\epsilon_f} \left( a \begin{bmatrix} -\frac{\partial^2 V}{\partial x^2} & -\frac{\partial^2 V}{\partial x \partial y} & -\frac{\partial V}{\partial y} \\ 0 & 0 & 0 \\ 0 & 0 & -\frac{\partial^2 V}{\partial \theta^2} \end{bmatrix} + \tilde{b} \begin{bmatrix} 0 & 0 & -\frac{\partial^2 V}{\partial \theta^2} \\ 0 & 0 & 0 \\ \frac{\partial^2 V}{\partial x^2} & \frac{\partial^2 V}{\partial x \partial y} & \frac{\partial V}{\partial y} \end{bmatrix} \right). \tag{22}$$

In above equation  $\tilde{b}$  is as follows:

$$\tilde{b} = \frac{b \partial V}{\epsilon \partial y}. \tag{23}$$

By computing  $\det(\lambda I - A(\mathbf{q}_0)) = 0$  one obtains:  $\lambda(\lambda^2 + \lambda \eta_a + \eta_b) = 0$ , where

$$\eta_a = \frac{\epsilon_e}{\epsilon_f^2} \left[ a \left( \frac{\partial^2 V}{\partial \theta^2} + \frac{\partial^2 V}{\partial x^2} \right) - \frac{b}{\epsilon} \left( \frac{\partial V}{\partial y} \right)^2 \right],$$

$$\eta_b = \left( \frac{\epsilon_e}{\epsilon_f} \right)^2 \left[ a^2 + \left( \frac{b \partial V}{\epsilon \partial y} \right)^2 \right] \frac{\partial^2 V}{\partial x^2} \frac{\partial^2 V}{\partial \theta^2}.$$

The three eigenvalues of  $A(\mathbf{q}_0)$  are as follows:

$$\begin{aligned} \lambda_1 &= 0 \\ \lambda_{2,3} &= -\frac{1}{2} \left( \frac{\epsilon_e}{\epsilon_f} \right) \left\{ a \left( \frac{\partial^2 V}{\partial \theta^2} + \frac{\partial^2 V}{\partial x^2} \right) - \frac{b}{\epsilon} \left( \frac{\partial V}{\partial y} \right)^2 \right. \\ &\quad \pm \left[ \frac{b^2}{\epsilon^2} \left( \frac{\partial V}{\partial y} \right)^4 - 2a \left( \frac{\partial^2 V}{\partial \theta^2} + \frac{\partial^2 V}{\partial x^2} \right) \frac{b}{\epsilon} \left( \frac{\partial V}{\partial y} \right)^2 \right. \\ &\quad \left. \left. + a^2 \left( \frac{\partial^2 V}{\partial x^2} - \frac{\partial^2 V}{\partial \theta^2} \right)^2 - 4 \frac{b^2}{\epsilon^2} \left( \frac{\partial^2 V}{\partial y^2} \right)^2 \frac{\partial^2 V}{\partial x^2} \frac{\partial^2 V}{\partial \theta^2} \right]^{1/2} \right\}. \end{aligned}$$

Eigenvalues  $\lambda_{2,3}$  differ from the ones obtained in [20] only by the coefficient  $(\epsilon_e/\epsilon_f^2)$ . All the conclusions for linearized system presented in [20] remain the same for the modified algorithm, in particular the necessary conditions for an equilibrium point to be stable are:

$$\left( \frac{\partial V}{\partial y} \right)^2 \leq \frac{\epsilon a}{b} \left( \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial \theta^2} \right), \quad \frac{\partial^2 V}{\partial x^2} \geq 0.$$

If the system has a zero eigenvalue, there exists in the neighborhood of 0 a function  $V(\mathbf{q})$  such that  $\frac{\partial V}{\partial \mathbf{q}} \dot{\mathbf{q}}$  is negative for  $\mathbf{q} \neq 0$  and which has the further property it remains negative definite for some reasonable class of perturbations [1]. The experiments carried out seem to confirm that in presented analysis the higher order terms (rejected as a result of approximation) belong to this class because the behavior of the closed-loop system, even for the states far

from the equilibrium point, is consistent with the results of the analysis based on the linear approximation. This local analysis seems to be applicable to the initial state points being quite far from the equilibrium point. The eigenvector corresponding to the zero eigenvalue lies in the tangent space of  $\mathcal{H}$  set at the equilibrium point. Velocity  $\dot{\mathbf{q}}$  in the direction of the eigenvector of the zero eigenvalue cannot be generated, even in nonlinear system (14). The motion in that direction is restricted due to a nonholonomic constraint. The two non-zero eigenvalues determine stability of the equilibrium point.

It can be clearly seen from the analysis of  $\lambda_{2,3}$  eigenvalues that by tuning  $\epsilon_e$  and  $\epsilon_f$  parameters the term  $(\epsilon_e/\epsilon_f^2)$  can be increased. This raises both real and imaginary components of these two eigenvalues. The real parts are relocated to the left (negative) side of the complex plane, an operation that allows better convergence of the system.

The eigenvalues can be ensured to have only real parts by fulfilling the condition:

$$2a\left(\frac{\partial^2 V}{\partial \theta^2} + \frac{\partial^2 V}{\partial x^2}\right)\frac{b}{\epsilon}\left(\frac{\partial V}{\partial y}\right)^2 + 4\frac{b^2}{\epsilon^2}\left(\frac{\partial^2 V}{\partial y^2}\right)^2 \frac{\partial^2 V}{\partial x^2} \frac{\partial^2 V}{\partial \theta^2} \leq \frac{b^2}{\epsilon^2}\left(\frac{\partial V}{\partial y}\right)^4 + a^2\left(\frac{\partial^2 V}{\partial x^2} - \frac{\partial^2 V}{\partial \theta^2}\right)^2.$$

This condition is fulfilled if the following, simpler inequalities are met:

$$\frac{\partial^2 V}{\partial x^2} \geq 0, \quad \frac{\partial^2 V}{\partial y^2} \geq 0.$$

In general, designing NF to fulfill these conditions may be quite difficult; however, they are not necessary to achieve asymptotic stability. They assure non-oscillatory behavior of the linearized system, which is a desired property in most applications.

If the robot is far from the equilibrium point the linear approximation may not be accurate, and thus, the improvement of the convergence is deduced by analyzing nonlinear control (7). In the NF  $V$  (4) function of distance to the goal  $C$  (5) and obstacles  $\beta$  (6) is mapped from  $(0, \infty)$  to  $(0, 1)$  - the counterdomain of the  $V$ . As a result, the influence of the distance to the goal on the control intensity (mainly for the locations far from the goal) is reduced. In the proposed modification the control is extended by including the term  $(\|\mathbf{q}^\top\| + \epsilon_e)$  that introduces dependency of control intensity on the distance to the goal. The term in the denominator  $(g^2 + \epsilon_f)$  intensifies the control for small  $\nabla V$  values (and thus  $B^\top \nabla V$ ). The described observations were confirmed by simulations and experiments but of course this is not formal analysis. Properties of the NF given in Section 3 are conducive to extend the region in which desired attributes of the linear approximation (resulting from the analysis of the eigenvalues) remain in force.

For improperly chosen  $\epsilon_e$  and  $\epsilon_f$ , the proposed algorithm may also characterize slow convergence, especially for distances far from the goal. Numerical tests show that even for these cases (far from the goal) the influence of  $(\epsilon_e/\epsilon_f^2)$  was consistent with the conclusions resulting from the analysis of the linearized system: by increasing  $(\epsilon_e/\epsilon_f^2)$  the convergence could be significantly improved. Of course, it is not guaranteed that this property always holds in the nonlinear system.

## 5 Saddle Points

Saddle points are unstable equilibrium points that occur when the attracting vector is balanced by the collision avoidance vector. There is one such point associated with each obstacle. From the theoretical point of view the saddle point is a zero measure set. One might suspect that the designer does not have to deal with this problem; however, it is not obvious how the behavior of the robot is affected in a real application by the saddle point. Some researchers states that saddle points present no problem in real world robotics since being trapped by saddle point can only occur in infinite precision dynamics [19, 22]. On the other hand, measurement quantization may cause the saddle point to expand to the some area surrounding its exact, theoretical location.

In this section some observations made during experiments are presented. It should be noted that behavior of the system may differ in other test-beds. Especially resolution of localization system, its precision, noises, repeatability and resolution of the actuator controllers may influence the system behavior.

### 5.1 Saddle Point Detection

The detection of the saddle point is based on analysis of the gradient  $\nabla V$  and Hessian of the navigation function  $H(V)$ . The saddle point is detected if two conditions are fulfilled: 1.  $\nabla V = \vec{0}$  (in practice  $\|\nabla V\| < \epsilon_s$ , where  $\epsilon_s$  - small positive constant), 2. one eigenvalue of  $H(V)$  is negative. In Section 6 detailed analysis of the influence of parameter  $\epsilon_s$  on the process of saddle point detection and avoidance is presented.

The Hessian of the navigation function is as follows:

$$H(V) = \frac{\partial(\nabla V)}{\partial \mathbf{q}} = \frac{H(C)}{(\beta + C^\kappa)^{1/\kappa}} + \frac{C(H(\beta) + \kappa C^{\kappa-1}H(C) + \kappa C^{\kappa-2}\nabla C^\top \nabla C(\kappa - 1))}{\kappa(\beta + C^\kappa)^{\frac{1}{\kappa}+1}} - \frac{2(\nabla \beta + \kappa C^{\kappa-1}\nabla C)^\top \nabla C}{\kappa(\beta + C^\kappa)^{\frac{1}{\kappa}+1}} \tag{24}$$

$$+ \frac{C(\frac{1}{\kappa} + 1)(\nabla\beta + \kappa C^{\kappa-1}\nabla C)^\top (\nabla\beta + \kappa C^{\kappa-1}\nabla C)}{\kappa(\beta + C^\kappa)^{\frac{1}{\kappa}+2}}.$$

The Hessian of the attraction function is given by the following equation:

$$H(C) = \frac{\partial(\nabla C)}{\partial \mathbf{q}} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 C}{\partial x^2} & \frac{\partial^2 C}{\partial x \partial y} & \frac{\partial^2 C}{\partial x \partial \theta} \\ \frac{\partial^2 C}{\partial y \partial x} & \frac{\partial^2 C}{\partial y^2} & \frac{\partial^2 C}{\partial y \partial \theta} \\ \frac{\partial^2 C}{\partial \theta \partial x} & \frac{\partial^2 C}{\partial \theta \partial y} & \frac{\partial^2 C}{\partial \theta^2} \end{bmatrix} \tag{25}$$

where

$$h_{11} = \frac{8k_w\theta^2x^2}{(\|\mathbf{r}\|^2 + k_w)^3} - \frac{2k_w\theta^2}{(\|\mathbf{r}\|^2 + k_w)^2} + 2$$

$$h_{22} = \frac{8k_w\theta^2y^2}{(\|\mathbf{r}\|^2 + k_w)^3} - \frac{2k_w\theta^2}{(\|\mathbf{r}\|^2 + k_w)^2} + 2$$

$$h_{33} = \frac{2k_w}{\|\mathbf{r}\|^2 + k_w}$$

$$h_{12} = h_{21} = \frac{8k_w\theta^2xy}{(\|\mathbf{r}\|^2 + k_w)^3}$$

$$h_{13} = h_{31} = -\frac{4k_w\theta x}{(\|\mathbf{r}\|^2 + k_w)^2}$$

$$h_{23} = h_{32} = -\frac{4k_w\theta y}{(\|\mathbf{r}\|^2 + k_w)^2}$$

Hessian of the obstacle function is given by equation:

$$H(\beta) = \frac{\partial(\nabla\beta)}{\partial \mathbf{q}} = \tag{26}$$

$$= \sum_{i=0}^N \left( \frac{\partial^2 \beta_i}{\partial \mathbf{q}^2} \prod_{j=0, j \neq i}^N \beta_j + \frac{\partial \beta_i}{\partial \mathbf{q}} \sum_{k=0}^N \left\{ \frac{\partial \beta_k}{\partial \mathbf{q}} \prod_{j=0, j \neq i, j \neq k}^N \beta_j \right\} \right).$$

### 5.2 Control in the Saddle Point

The control given by Eq. 7 is temporarily replaced with the following one [21]:

$$\mathbf{u}_a \triangleq [a_1 \sin(wt) + b_1 \quad a_2 \cos(wt)]^T \tag{27}$$

where  $a_1, a_2, w$  and  $b_1$  are chosen such that  $[b_1 \quad \frac{a_1 a_2}{2w} \quad 0]^T$  is parallel to the  $H(V)$  eigenvector of the negative eigenvalue and  $|\frac{a_1}{w}| \ll 1, |\frac{a_2}{w}| \ll 1$  and  $|b_1| \ll 1$ . In [21], a detailed analysis of the system with this control is given. It is shown that by moving along the eigenvector of the negative eigenvalue and taking into account that the value of the NF in the area to which the robot is driven is less than in the saddle point the proper behavior is obtained: the robot does not approach the saddle point and it converges to the goal.

## 6 Experiments

In this section experimental test-bed is described and results of conducted tests are presented.

### 6.1 Experimental Setup

The mobile platform that was used in the presented experiments is the differentially driven MTracker robot (Fig. 1). It was designed at Poznan University of Technology. It is controlled by a two-level hardware controller: the low-level motion controller uses the signal processor TMS320F28335 150MHz and the high-level one is a single-core Intel Atom 1.2GHz board, equipped with WiFi radio used for remote management, task setting and communication with the external localization system. Depending on the requirements the high-level controller works on the Linux Ubuntu or Windows XP operating systems. The MTracker robot is a small platform: its diameter is 0.14m, its height is 0.13m, its weight is 1.4kg and its wheels have a diameter of 0.05m. The on-board power supply is LiIo 3.7Ah battery that allows two hour active operation.

During the test the robot is localized by the OptiTrack motion capture system. On the top of the robot four infra-red reflecting markers were mounted.

Wheel velocity control signals were scaled down when their value(s) exceeded the limit. This limit was set to 12rd/s, while the physical limitation of actuators is 24rd/s. The lower value of the limit prevented robot wheels from longitudinal slip. The obstacles were known *a priori* to prevent the influence of measurement inaccuracies on the experimental results.

A special scaling procedure is applied to the wheel controls. The desired wheel velocities are scaled down when at least one of the wheels exceeds the assumed limitation. The scaled control signal  $\mathbf{u}_s$  is calculated as follows:

$$\mathbf{u}_s = s\mathbf{u}, \tag{28}$$

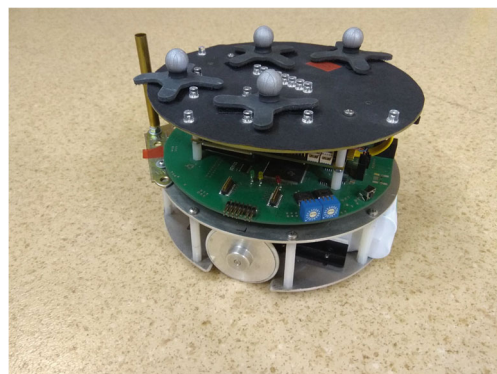


Fig. 1 MTracker robot (Poznan University of Technology)

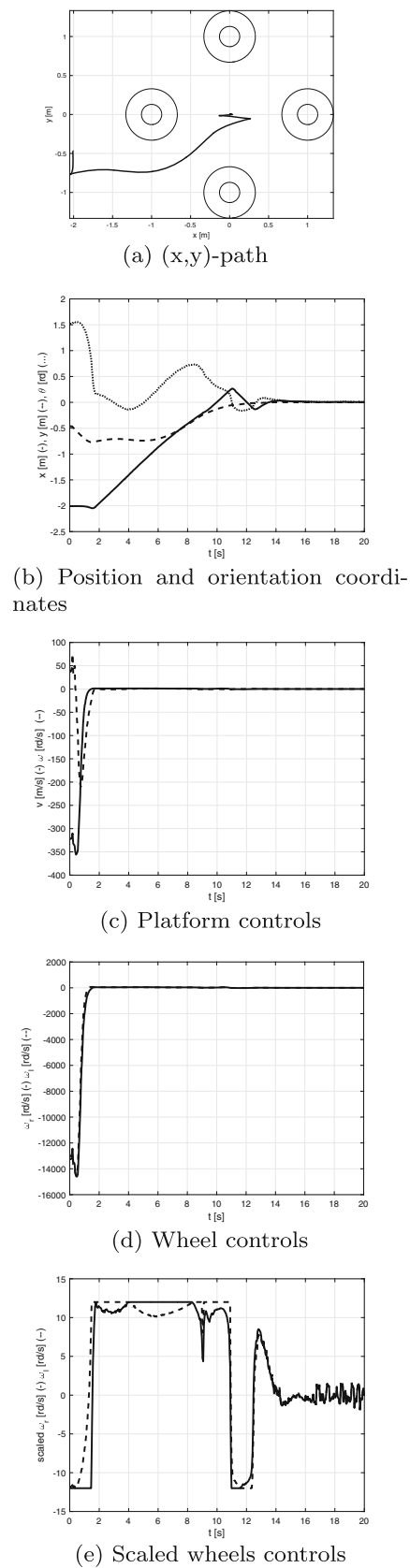


Fig. 2 Experiment 1: new algorithm

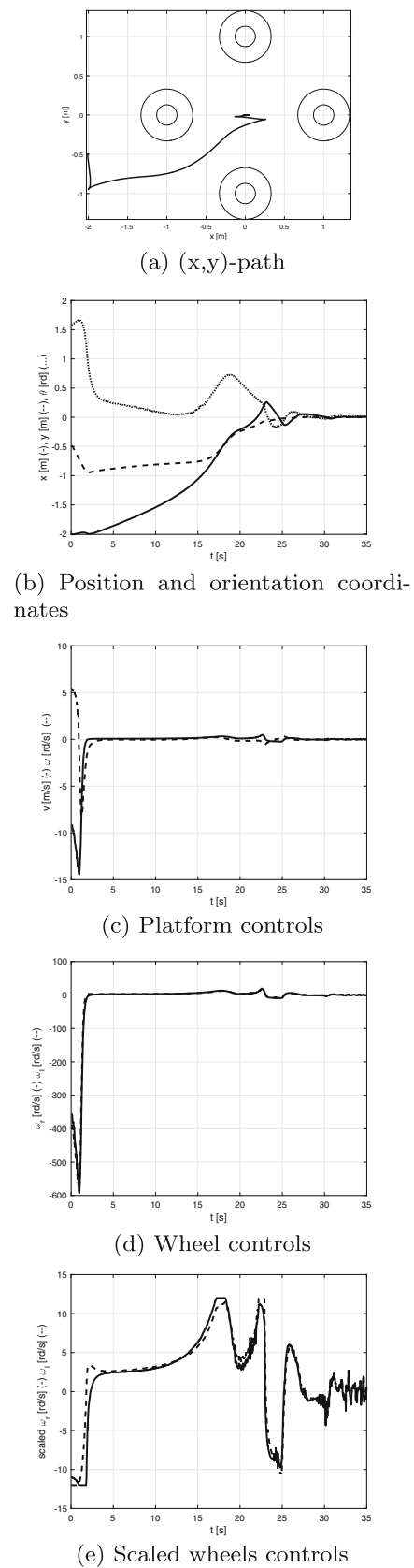


Fig. 3 Experiment 2: algorithm from [20]

where

$$s = \begin{cases} \frac{\omega_{max}}{\omega_o} & \text{if } \omega_o > \omega_{max} \\ 1 & \text{otherwise} \end{cases}, \tag{29}$$

and

$$\omega_o = \max\{|\omega_r|, |\omega_l|\}, \tag{30}$$

where  $\omega_r, \omega_l$  denote right and left wheel angular velocity,  $\omega_{max}$  is the predefined maximal allowed angular velocity for each wheel. This scaling procedure preserves the direction of motion of the mobile platform.

### 6.2 New Algorithm Comparison

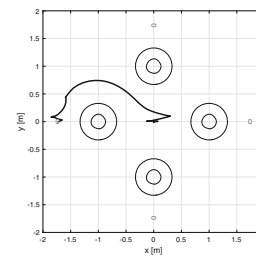
This section discusses results of experiments. In all the presented cases the desired state was  $[0.0 \ 0.0 \ 0.0]^T$ . The circular obstacles ( $N = 4$ ) were located at the following locations:  $[0.0 \ -1.0]^T, [0.0 \ 1.0]^T, [-1.0 \ 0.0]^T, [1.0 \ 0.0]^T$ , their radii  $0.13m$  extended with  $0.2m$  safety zone. Task space is bounded by the circle with the center in the origin of the coordinate frame and radius  $5m$ . The algorithm parameters were as follows:  $\kappa = 5, k_w = 0.1, a = 0.5, \bar{b} = 1.5, \epsilon_g = 10^{-4}, \epsilon_e = 0.1$  and  $\epsilon_f = 0.01$ .

In the first experiment the results for the new control algorithm are presented. The second experiment shows the results for the algorithm from [20] that can be obtained by replacing  $\gamma$  in Eq. 7 with

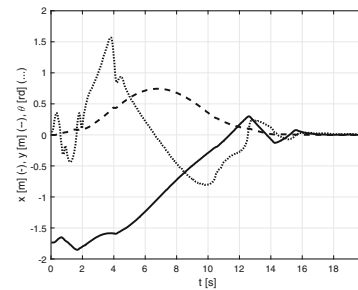
$$\bar{\gamma} = B^T \nabla V. \tag{31}$$

In both experiments  $(x, y)$ -paths of the platform center (Figs. 2a, 3a) are similar. The initial position and orientation vector was  $[-2.0 \ -0.48 \ 1.5]^T$ . Figures 2b and 3b show time graphs of position and orientation coordinates. It can be seen that the new algorithm drives system error to zero in 15s while the algorithm from [20] requires 30s to reach this state.

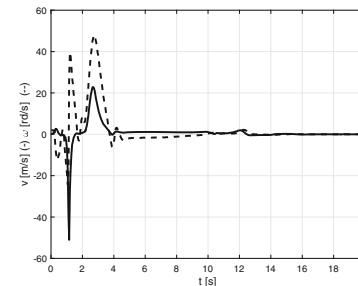
The controls generated by the new algorithm are larger by the order of magnitude (Figs. 2c and 3c). As a result, also the controls of robots wheels are much higher (Figs. 2d and 3d). After transformation by the scaling procedure the control signal goes into saturation in the beginning and stays in this state until 12.5s (Figs. 2e and 3e). In the reference algorithm control was saturated a few times but only during short time intervals. To summarize, both algorithms produce a similar path shape but the new algorithm results in twice faster convergence. As shown in Kowalczyk [5], for the initial location further from the origin the improvement may even be much better. In some cases, the new algorithm works properly even in the cases in which the reference algorithm may be considered useless. Note that velocity limits on actuators also reduce the observed improvement. Applying the new algorithm is more beneficial for larger  $\kappa$  that must be increased as narrower passages between obstacles exist in the task space.



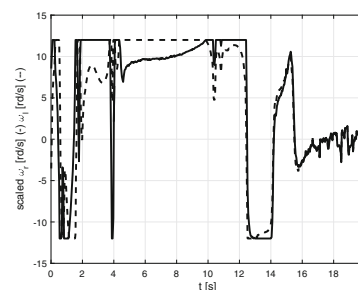
(a)  $(x,y)$ -path



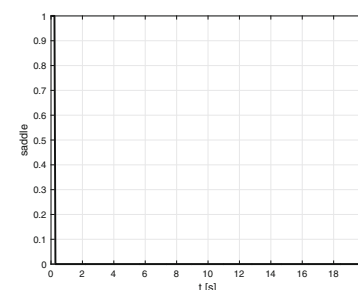
(b) Position and orientation coordinates



(c) Platform controls



(d) Scaled wheel controls



(e) Saddle

Fig. 4 Experiment 3: Saddle point - detection level  $\epsilon_s = 0.01$



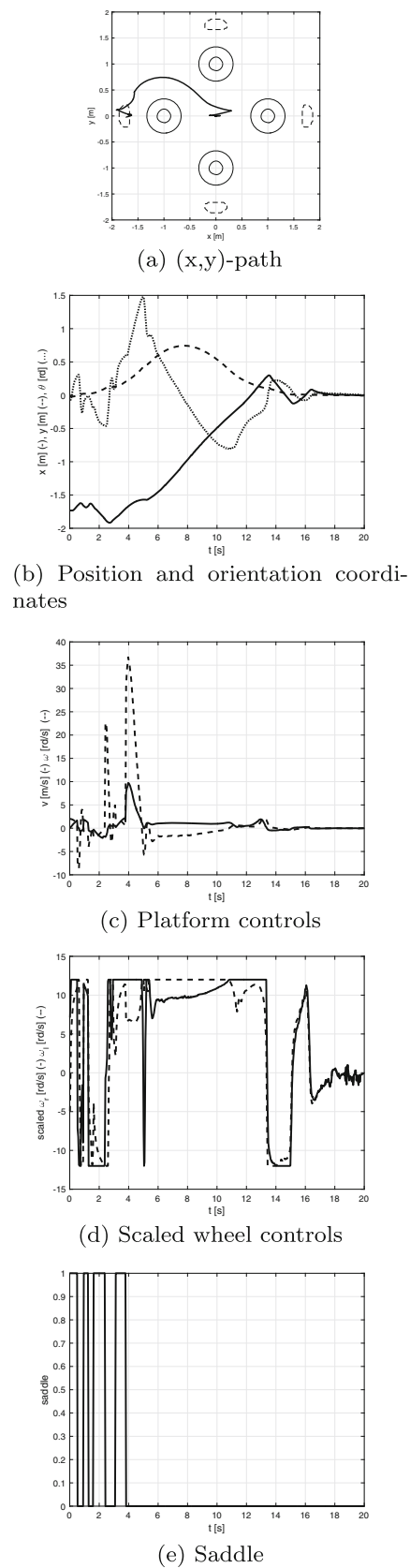


Fig. 5 Experiment 4: Saddle point - detection level  $\epsilon_s = 0.05$

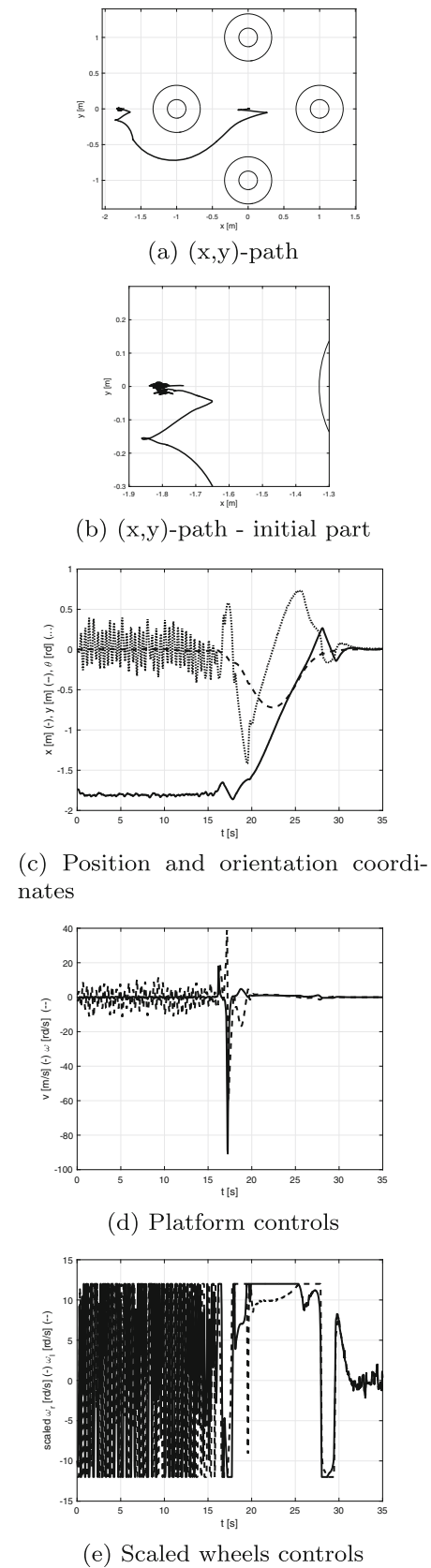


Fig. 6 Experiment 5: Saddle point - detection turned off

### 6.3 Saddle Point

Saddle point detection and avoidance is presented in the three subsequent experiments. In the first of them  $\epsilon_s = 0.01$ . In Fig. 4a gray contour at locations around  $[0.0 \ -1.73]^T$ ,  $[0.0 \ 1.73]^T$ ,  $[-1.73 \ 0.0]^T$ ,  $[1.73 \ 0.0]^T$  represent areas where the saddle point avoidance procedure is activated. This Figure shows also the path of the robot in  $(x, y)$ -plane. It starts from  $[-1.73 \ 0.0]^T$  (saddle point) with orientation  $0.0rd$ . Figure 4b presents time graphs of position and orientation coordinates. They converge to the desired values in  $17s$ . Figure 4c and d show time graphs of controls for the platform and wheels after scaling procedure. Figure 4e presents activity of the saddle point procedure. As can be observed it is active by the first  $0.5s$  of the experiment.

The next experiment was conducted for  $\epsilon_s = 0.05$ . In this case the detection areas surrounding the saddle points are much larger, marked with dashed lines in Fig. 5a. Figure 5b shows time graph of robots coordinates. Greater  $\epsilon_s$  resulted in slower convergence ( $2s$  more). Figure 5c shows linear and angular controls for the platform. Figure 5d presents robot wheel controls after scaling. Figure 5e shows the saddle avoidance activation variable. As one can see, the robot was got into the avoidance area four times in the first  $4s$  of the experiment.

The last experiment was conducted with the saddle point avoidance procedure turned off. Figure 6a shows the  $(x, y)$ -path. Figure 6b presents the initial part of the path. It can be noted that the platform oscillates around the saddle point and finally leaves its surroundings and goes to the desired location. Figure 6c shows the time graph of the position and orientation coordinates. Note that largest oscillation amplitude can be observed in orientation signal ( $\pm 0.25rd$ ). After  $16s$  the robot leaves the trap. Oscillations are also observed in angular velocity control signal (Fig. 6d - dashed lines). Figure 6e shows wheel controls after scaling. By the first  $16s$  wheel controls switch between positive and negative velocity limits.

Note that whether the robot was driven left or right side of the obstacle depended on the time variable in Eq. 27. The best convergence time (that was mostly dependent on the time of leaving the saddle point) was obtained for  $\epsilon_s = 0.01$ . For much larger or smaller values of  $\epsilon_s$  the convergence was slower.

## 7 Conclusions

Navigation function algorithms are used for set-point control of the vehicles moving in 2D and 3D environments, with and without nonholonomic constraints, with

sphere-shaped obstacles, star-shaped obstacles and more complex environments in which obstacles can be modeled as trees-of-stars. These methods can be tuned to ensure one global minimum of the NF; however, in general they do not guarantee fast convergence to the target.

This paper proposes a rapidly converging NF control algorithm. The convergence of the closed-loop system is analyzed. The advantages of the new algorithm are significant for greater values of the  $\kappa$  parameter that must be increased if narrow passages between obstacles exist in the task space. The method was tested by experiments on an actual robot. It provides not only better convergence time, but also rapid convergence in cases in which other methods produce questionable results. The experiments presented illustrate clear improvement. The paper also includes experimental verification of the robots behavior in the saddle point. It was tested for the saddle point procedure turned off and for two different sizes of the saddle point detection regions.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## References

1. Brockett, R.W.: Asymptotic stability and feedback stabilization, *Differential Geometric Control Theory*, pp. 181–191. Birkhauser, Boston (1983)
2. Dimarogonasa, D.V., Loizou, S.G., Kyriakopoulos, K.J., Zavlanos, M.M.: A feedback stabilization and collision avoidance scheme for multiple independent non-point agents. *Automatica* **42**(2), 229–243 (2005)
3. Filippidis, I., Kyriakopoulos, K.J.: Adjustable navigation functions for unknown sphere worlds. In: *IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pp. 4276–4281 (2011)
4. Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robot. Res.* **5**(1), 90–98 (1986)
5. Kowalczyk, W.: Rapidly converging navigation function control for differentially driven mobile robots. In: *11th International Workshop on Robot Motion and Control (RoMoCo)*, p. 244 (2017)
6. Kowalczyk, W., Kozłowski, K.: Control of the differentially-driven mobile robot in the environment with a non-convex star-shape obstacle: Simulation and experiments. *Acta Polytechnica Hungarica* **13**(1), 123–135 (2016)
7. Kowalczyk, W., Przybyła, M., Kozłowski, K.: Rapid navigation function control for omnidirectional mobile platform. In: *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, p. 137 (2017)

8. Kowalczyk, W., Przybyła, M., Kozłowski, K.: Set-point control of mobile robot with obstacle detection and avoidance using navigation function-experimental verification. *J. Intell. Robot. Syst.* **85**(3-4), 539–552 (2017)
9. Kowalczyk, W., Przybyła, M., Kozłowski, K.: Control of a mobile robot and collision avoidance using navigation function-experimental verification. In: 2015 10th International Workshop on Robot Motion and Control (RoMoCo), pp. 148–152
10. Kowalczyk, W., Przybyła, M., Kozłowski, K.: Saddle point detection of the navigation function in nonholonomic mobile robot control. In: 2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR), pp. 936–941, Miedzyzdroje (2016). <https://doi.org/10.1109/MMAR.2016.7575263>
11. Lionis, G., Papageorgiou, X., Kyriakopoulos, K.: Locally computable navigation functions for sphere worlds. In: Proc. IEEE International Conference on Robotics & Automation (ICRA-2007), pp. 1998–2003. Roma (2007)
12. Rimon, E., Koditschek, D.E.: Exact robot navigation using cost functions: the case of distinct spherical boundaries. *IEEE Int. Conf. Robot. Autom.* **3**, 1791–1796 (1988)
13. Rimon, E., Koditschek, D.E.: The construction of analytic diffeomorphisms for exact robot navigation on star worlds. *Trans. Amer. Math. Soc.* **327**, 71–116 (1991)
14. Rimon, E., Koditschek, D.: Exact robot navigation using artificial potential functions. *IEEE Trans. Robot. Autom.* **8**(5), 501–518 (1992)
15. Roussos, G., Kyriakopoulos, K.J.: Completely decentralised navigation of multiple unicycle agents with prioritisation and fault tolerance. In: IEEE Conference on Decision and Control (CDC), pp. 1372–1377 (2010)
16. Roussos, G., Kyriakopoulos, K.J.: Decentralized and prioritized navigation and collision avoidance for multiple mobile robots. *Distrib. Autonom. Robot. Sys. - Springer Tracts Adv. Robot.* **83**, 189–202 (2013)
17. Roussos, G., Kyriakopoulos, K.: Decentralized navigation and conflict avoidance for aircraft in 3-D space. *IEEE Trans. Control Syst. Technol.* **20**(6), 1622–1629 (2012). <https://doi.org/10.1109/TCST.2011.2167974>
18. Roussos, G., Dimarogonas, D., Kyriakopoulos, K.: 3D navigation and collision avoidance for nonholonomic aircraft-like vehicles. *Int. J. Adapt. Control Signal Process. Special Issue: Air Traffic Manag.: Challenges Opport. Adv. Control* **24**(10), 900–920 (2010)
19. Trevisan, M., Idiart, M., Prestes, E., Engel, P.: Exploratory navigation based on dynamical boundary value problems. *J. Intell. Robot. Syst.* **45**(2), 101–114 (2006)
20. Urakubo, T.: Feedback stabilization of a nonholonomic system with potential fields: Application to a two-wheeled mobile robot among obstacles. *Nonlin. Dyn.* **81**(3), 1475–1487 (2015)
21. Urakubo, T., Okuma, K., Tada, Y.: Feedback control of a two wheeled mobile robot with obstacle avoidance using potential functions. *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)* **3**, 2428–2433 (2004)
22. Xu, Z., Hess, R., Schilling, K.: Constraints of potential field for obstacle avoidance on car-like mobile robots. In: IFAC Proceedings Volumes, vol. 45, no. 4, pp. 169–175. Wurzburg, Germany (2012)

**Wojciech Kowalczyk** received the M.Sc. degree in robotics from Poznan University of Technology (PUT), Poland in 1998. In 1999–2000 he was working as electronics design engineer and software developer (programming databases and communication applications for retail chains). Since 2000 he works at Poznan University of Technology where he received the Ph.D. degree in robotics in 2008. His scientific interests focus on the multi-agent systems, mainly on their application in the control of multiple mobile robots, collision avoidance and cooperative behaviours of non-holonomic robots at both kinematics and dynamics level. He was involved in medical research projects conducted at PUT. He teaches multi-agent systems, smart systems, robotics, object oriented programming, electrical engineering and electronics.