Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 50-1 (2017) 4312–4317

Navigation Function Used for Parallel Parking in Restricted Area by a Differentially-Driven Mobile Robot

Mateusz PRZYBYŁA * **Wojciech KOWALCZYK** *
Krzysztof KOZŁOWSKI *

* *Poznań University of Technology*
Chair of Control and Systems Engineering
ul. Piotrowo 3A, Poznań, Poland
(e-mails: mateusz.przybyla@put.poznan.pl,
wojciech.kowalczyk@put.poznan.pl,
krzysztof.kozlowski@put.poznan.pl)

Abstract: This work has examined the use of navigation function for the motion task of parallel parking performed by mobile robot with kinematic constraints working in two-dimensional environment with obstacles. Classic navigation function (NF) approach assumes omnidirectional robot. In this paper NF extended with orientation component was tested. It can be tuned to guarantee global convergence to the target position with desired orientation. This paper provides results of experiments with simulated as well as physical robot, showing typical behavior of the method, as well as authors' remarks on its benefits and disadvantages.

Keywords: motion control, unicycle, parallel parking, obstacle avoidance, navigation function

1. INTRODUCTION

Navigation function was first introduced by Rimon and Koditschek as an answer to the Khatib's innovative proposition i.e. to use artificial potential fields to drive the robots motion [4]. Khatib's method treats obstacles as sources of positive potential and the target as a source of negative potential. The superposition of these factors produce a total potential field and its gradient is used to calculate control signals for the robot at any point in time and space. The superposition of the potential fields, however, may result in local potential minima, which are pitfalls during the execution of the control task. The navigation function approach has overcome this inconvenience.

The first prototype of NF was dedicated to N-dimensional sphere-like environments and assumed that the obstacles are also bounded with spheres [9]. By substituting N=2, it was easily converted to two dimensional, circular environments. Although NF approach has removed the problem of local minima, the existence of saddle points in the potential field still remained. Few years later the method was expanded to more complex environments [10, 11]. The handicap of these algorithms was that they assumed a point-like robot without kinematic constraints.

The nonholonomic constraints of the differentially driven mobile robots were taken into consideration in extension of NF provided by Urakubo and his colleagues in [14]. In their method the orientation component was included as a vital part of the artificial potential field. In general the path resulting from the gradient of such potential may not be admissible for the robots, however, the proper projection of gradient to control signal space corrects the motion to ensure convergence of both position and orientation.

In [2], [12] and [13] navigation function was applied to multi-robot system. In these methods robots are moving obstacles and affect the shape of the navigation function.

Problem of parallel parking was investigated many times. In [1] geometrical planning was used to park unicycle type robot. In [7] reactive control scheme was applied to solve car-like robot parking problem. Rigato's work [8] presents flatness-based controller for parallel parking between two obstacles located close to each other.

The main motivation of this research was to verify if NF approach that guarantees almost global convergence (excluding saddle points) generates acceptable trajectories in the case of demanding parallel parking task. It was tested both by numerical simulations and experiments with Kuka youBot mobile platform.

2. VEHICLE MODEL

The considered control strategy applies to driftless, differentially driven mobile robots with kinematic constraints, which model can be given by:

$$\dot{\boldsymbol{q}} \triangleq \boldsymbol{B}\boldsymbol{u} = \begin{bmatrix}\cos\theta & 0\\ \sin\theta & 0\\ 0 & 1\end{bmatrix}\boldsymbol{u}, \tag{1}$$

where vector $\boldsymbol{q} \triangleq [x, y, \theta]^\top$ denotes its state and x, y and θ represent the position coordinates and orientation of the robot with respect to a global, fixed coordinate frame. Vector $\boldsymbol{u} \triangleq [v, \omega]^\top$ denotes the control vector with

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2017.08.852

v and ω representing the linear and angular velocity of the platform, respectively.

3. ENVIRONMENT MODEL

The control algorithm restricts the robot task space to be bounded by a conceivable circle of radius ρ_0 and with its center located at $\boldsymbol{p}_0 = (x_0, y_0)$. For the sake of simplicity, we assume that the reference point (or the goal) lays at the center of this circle. From the task space configuration we can construct a potential-like factor which repels the robot from the boundary of the environment [9]:

$$\beta_0 \triangleq \rho_0^2 - ||\boldsymbol{r} - \boldsymbol{p}_0||^2, \tag{2}$$

where $\boldsymbol{r} = [x, y]^\top$ is the current position of the robot. Note that the repelling potential factor stays positive inside the task space and becomes negative outside.

In the task space, there can be many obstacles located at positions $\boldsymbol{p}_i = (x_i, y_i)$ $(i = 1, \cdot, N,\ N$ being the number of obstacles). The obstacles are assumed to be circular-shaped with radii ρ_i. The repelling potential factor for i-th obstacle is defined as [9]:

$$\beta_i \triangleq ||\boldsymbol{r} - \boldsymbol{p}_i||^2 - \rho_i^2. \tag{3}$$

Again, note that this factor stays positive only outside the obstacle and becomes negative inside.

4. CONTROL DESIGN

Given an environment with N obstacles and a task of stabilizing the robot in its origin the total navigation potential is defined as [9]:

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

where κ is a positive constant design parameter called *potential steepness factor* and

$$C \triangleq ||\boldsymbol{r}||^2 + \theta^2 \frac{k_w}{k_w + ||\boldsymbol{r}||^2}. \tag{5}$$

Equation (5) describes a unified error in which k_w denotes a positive constant design parameter called *orientation importance factor* that allows to tune the influence of the orientation term on the navigation function depending on the euclidean distance to the reference position. The aggregation of repelling potentials occurs in term β which is defined as:

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

In the case in which robot reaches the boundary of the task space or any obstacle, product (6) evaluates to zero and the navigation potential (4) evaluates to one. This is the largest value of potential in this design. The potential reaches minimum at the target location where it evaluates to zero.

The control algorithm proposed in work [15] is defined as:

$$\boldsymbol{u} \triangleq -\left\{ a \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + b \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \right\} \boldsymbol{B}^\top \sigma \nabla V, \tag{7}$$

where a is a positive constant design parameter called *direct motion factor* and

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

is a varying parameter, where $\bar{b}$ is a positive constant design parameter called *transverse motion factor* and $\boldsymbol{L} \triangleq [\sin\theta, -\cos\theta, 0]^\top$. Furthermore, function $h(g)$ is defined as:

$$h(g) \triangleq g^2 + \epsilon_g \sqrt{g}, \tag{9}$$

where $g \triangleq ||\boldsymbol{B}^\top \nabla V||$ and ϵ_g is a small, positive, constant design parameter.

It is known that according to the Brockett's theorem asymptotic stabilization of system (1) cannot be achieved by the smooth control. Control given by Eq. (7) is discontinuous or at least non-differentiable in the equilibrium points.

The control signal $\boldsymbol{u}$ is a projection of navigation potential gradient, denoted by ∇V, onto directions admissible for the mobile robot. Regardless of number of obstacles, the gradient can be obtained in analytical form as:

$$\begin{aligned} \nabla V = & \tfrac{\nabla C (C^\kappa + \beta)^{\frac{1}{\kappa}}}{(C^\kappa + \beta)^{(\frac{2}{\kappa})}} + \\ & - \tfrac{\frac{C}{\kappa}(C^\kappa + \beta)^{(\frac{1}{\kappa} - 1)}(\kappa C^{\kappa - 1} \nabla C + \nabla \beta)}{(C^\kappa + \beta)^{(\frac{2}{\kappa})}}, \end{aligned} \tag{10}$$

where

$$\nabla C = \begin{bmatrix} \frac{\partial C}{\partial x} & \frac{\partial C}{\partial y} & \frac{\partial C}{\partial \theta} \end{bmatrix} \tag{11}$$

and

$$\nabla \beta = \sum_{i=0}^{N} \left\{ \frac{\partial \beta_i}{\partial \boldsymbol{q}} \prod_{j=0, j \neq i}^{N} \beta_j \right\}. \tag{12}$$

Equations representing specific terms in (11) were presented in work [6].

Since the potential field depends on the robot orientation, it can non intuitively rearrange during the run. This effect can be noticed mostly when the position error is relatively small. In such case, the orientation error becomes an essential factor and small rotation can result in a radical change of potential steepness. Because of this phenomenon, we propose the following, discontinuous gradient normalization:

$$\sigma = \begin{cases} 1 & \text{if} \quad ||\nabla V|| < ||\nabla V||_{\text{thr}},\ V < V_{\text{thr}} \\ ||\nabla V||^{-1} & \text{otherwise} \end{cases}. \tag{13}$$

Application of gradient normalization results in much faster convergence as well as more uniform motion of the robot. In equation (13) symbols $||\nabla V||_{\text{thr}}$ and V_{thr} represent the threshold values for normalization activation. The normalization is deactivated when the robot reaches the neighborhood of the target.

Finally, the control vector is subject to a scaling procedure, which saturates the control signals for the sake of safety:

$$\boldsymbol{u}^* \triangleq \boldsymbol{u}/s, \quad s \triangleq \max\left\{1, \frac{|v|}{v_{\max}}, \frac{|\omega|}{\omega_{\max}}\right\}, \tag{14}$$

where $v_{\max}, \omega_{\max}$ represent the designed limits for control signals.

For both control signals fulfilling conditions $|v| \leq v_{max}$ and $|\omega| \leq \omega_{max}$ stability analysis presented in [15] holds true.

For $\frac{|v|}{v_{max}} > 1$ and $\frac{|\omega|}{\omega_{max}} < \frac{|v|}{v_{max}}$ equations describing evolution of the position coordinates are as follows:

$$\begin{aligned} \dot{x} &= \operatorname{sgn}(v)\, v_{max} \cos\theta \\ \dot{y} &= \operatorname{sgn}(v)\, v_{max} \sin\theta \end{aligned} . \qquad (15)$$

As in this case the following condition is fulfilled: $|\operatorname{sgn}(v) v_{max}| < |v|$, the changes of position coordinates are slower in comparison to the case without scaling.

For $\frac{|\omega|}{\omega_{max}} > 1$ and $\frac{|v|}{v_{max}} < \frac{|\omega|}{\omega_{max}}$ equation describing evolution of the orientation can be obtained by substituting last row of (14) into last row of (1) and integration:

$$\theta = \operatorname{sgn}(\omega)\, \omega_{max}\, t. \qquad (16)$$

The changes of the position coordinates are slower in comparison to the case without scaling as the terms $\frac{\omega_{max}}{|\omega|}$ in equations

$$\begin{aligned} \dot{x} &= \tfrac{\omega_{max}}{|\omega|}\, v \cos\theta \\ \dot{y} &= \tfrac{\omega_{max}}{|\omega|}\, v \sin\theta \end{aligned} \qquad (17)$$

are less then one.

Taking into account that scaling procedure does not affect the linear to angular velocity ratio in the static task space and for set-point control the shape of the resulting robots' path remains unchanged.

5. SADDLE POINT DETECTION

As noted in work [11] all the undesired local minima of navigation function (4) disappear as the parameter κ increases. An algorithm for automatic tuning of analytic navigation functions for sphere worlds was presented in work [3]. The κ parameter must satisfy a lower bound to ensure convergence to the desired value. In this paper navigation functions have been manually tuned to assure convergence.

For sufficiently high value of κ navigation function (4) has a critical point associated with each isolated obstacle, i.e. the saddle point. V has no other critical points other than these points.

Saddle points are unstable equilibrium points. In [14] special control procedure for saddle point avoidance is described. It uses time varying trigonometric function to push the robot away from such configuration.

The detection of a saddle point can be based on analysis of Hessian of the navigation function $H(V)$. The primary necessary condition for the occurrence of saddle point is $\nabla V = \mathbf{0}$ (in practice $||\nabla V|| < \epsilon_\nabla$, where ϵ_∇ is a small positive value). The secondary necessary condition is that at least one of the eigenvalues of $H(V)$ must be negative and at least one must be positive.

Hessian of the navigation function (4) is as given by:

$$H(V) = \frac{\partial(\nabla V)}{\partial \boldsymbol{q}} = \frac{H(C)}{(\beta + C^\kappa)^{1/\kappa}} +$$

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

$$- \frac{2(\nabla\beta + \kappa C^{\kappa-1}\nabla C)^\top \nabla C}{\kappa(\beta + C^\kappa)^{\frac{1}{\kappa}+1}} \qquad (18)$$

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

Where the Hessian of the attraction function is given by:

$$H(C) = \frac{\partial(\nabla C)}{\partial \boldsymbol{q}} = \nabla^\top \nabla C = \begin{bmatrix} \frac{\partial^2 C}{\partial x^2} & \frac{\partial^2 C}{\partial x \partial y} & \frac{\partial^2 C}{\partial x \partial \theta} \\ \frac{\partial^2 C}{\partial y \partial x} & \frac{\partial^2 C}{\partial y^2} & \frac{\partial^2 C}{\partial y \partial \theta} \\ \frac{\partial^2 C}{\partial \theta \partial x} & \frac{\partial^2 C}{\partial \theta \partial y} & \frac{\partial^2 C}{\partial \theta^2} \end{bmatrix} \qquad (19)$$

and the Hessian of the obstacle function is given by:

$$H(\beta) = \frac{\partial(\nabla \beta)}{\partial \boldsymbol{q}} = \qquad (20)$$

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

If the saddle point is detected the avoidance procedure is activated meaning that the control given by Eq. (7) is temporarily replaced by the following one [15]:

$$\boldsymbol{u_a} \triangleq [a_1 \sin(wt) + b_1, a_2 \cos(wt)]^\top \qquad (21)$$

where a_1, a_2, w and b_1 are chosen such that $[b_1, \frac{a_1 a_2}{2w}, 0]^\top$ is parallel to the eigenvector of $H(V)$ corresponding to the negative eigenvalue and $|\frac{a_1}{w}| \ll 1$, $|\frac{a_2}{w}| \ll 1$ and $|b_1| \ll 1$.

6. EXPERIMENTAL SETUP

Experimental verification of the method was conducted on a mobile platform equipped with 2D obstacles detector. The experiments were performed in a laboratory environment which did not pose any significant form of obstructions related to the control tasks.

6.1 Mobile platform

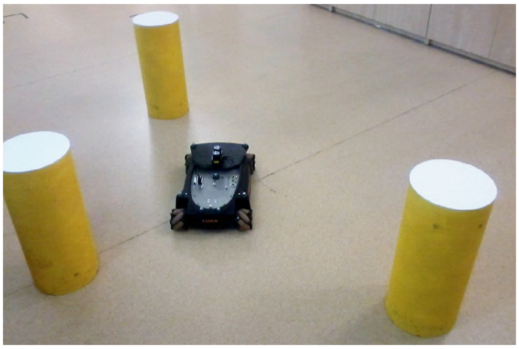

Fig. 1. Experimental setup showing Kuka youBot platform and three obstacles used during runs.

The platform used for experiments was Kuka youBot (Fig. 1). Although it is a four wheeled, omnidirectional mobile robot, its transverse velocity was restricted to zero and its characteristic point [1] was placed at its geometrical center, thus emulating a behavior of a differentially driven robot.

The omnidirectional wheels, robot inertia (20 kg) as well as other kinds of disparity between idealized robot model

[1] A point which represents the robot in the control task.

described in Sec. 2 and the one used for experiments posed additional requirement regarding robustness on the control design. The following experiments have shown that this requirement was fulfilled.

The software used for the operations was designed in Robot Operating System (ROS). The mobile platform had an onboard computer with Ubuntu Linux version 16.04 and ROS version Kinetic Kame. The human operator's computer worked with Ubuntu Linux version 14.04 and ROS version Jade Turtle. Both computers communicated via WiFi network. All of computations related to control design were performed on the youBot computer with a rate of 50 Hz.

Position and orientation of the platform was obtained from OptiTrack motion capture system which consisted of ten cameras working at a rate of 100 Hz. For this purpose, the mobile platform was equipped with seven infra-red reflecting markers. Because the network was prone to packet loss, a state estimator in form of two Kalman filters (one for orientation and one for position) was applied to smooth out the data in case of such events.

6.2 Environment perception

The mobile platform was equipped with two Hokuyo laser range finders (LRF). Both devices were placed back-to-back to scan environment in 360°. The appropriate procedures extracted obstacles in the form of line segments and circles from the merged scans. For the needs of considered control design only circular obstacles were taken into account. Although both LRFs worked at a rate of 10 Hz, application of Kalman filtering on detected obstacles provided supersampling which resulted in output rate of 50 Hz.

The low working rate of LRFs is a significant source of errors in radius and position of detected obstacles. While the robot is moving (in particular: rotating) some of the ranges do not get measured correctly, which leads to false geometric view of the environment. An exemplary graphs of radius and position variations for a single obstacle, taken during one of the following experiments, are presented in Figs. 2 and 3. The position variation is represented by L_2 norm of position x and y coordinates (distance from the origin). One must note that due to local nature of LRFs measurements (resulting from limited range and occlusions) perception of material objects existing in environment using these devices can be limited depending on robot-environment configuration (e.g. after reaching certain distance, one obstacle is discovered behind another). This fact can result in a non-stationary and discontinuous characteristic of a quasi-static environment, which can pose a significant impact on potential based control designs. Most tangible effects of such phenomena are discontinuities in control signals, which could be suppressed by proper time-based smoothing. This subject, however, is not the aim of this work.

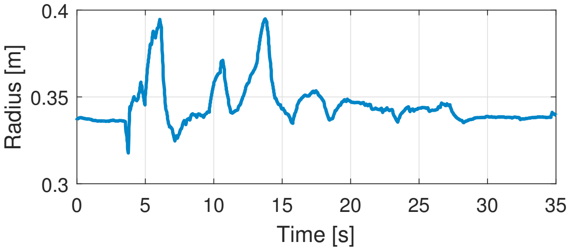

Fig. 2. Variation of obstacle radius in time ($r_{\min} = 0.32$ m, $r_{\max} = 0.39$ m, $r_{\text{avg}} = 0.34$ m).

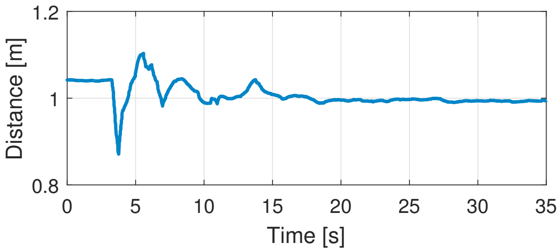

Fig. 3. Variation of position in time ($d_{\min} = 0.87$ m, $d_{\max} = 1.10$ m, $d_{\text{avg}} = 1.00$ m).

7. EXPERIMENTAL RESULTS

Experiments conducted in order to show the efficacy of the method were performed for three cases and in three variants. In each case the robot task was to reach the origin of the global coordinate frame while avoiding collisions with three obstacles of radius $r = 0.3$ m placed in locations: $o_1 = (-1, 0)$, $o_2 = (0, -1)$ and $o_3 = (1, 0)$ in meters. The radius r included 0.2 m of margin added to true obstacles radius. In the first case, the robot started from initial coordinates $\boldsymbol{q}(0) = [2, 1.5, 0]^\top$, in the second case: $\boldsymbol{q}(0) = [0, 1.5, \pi/2]^\top$, and in the third case: $\boldsymbol{q}(0) = [-2, 1.5, \pi]^\top$. The initial velocity of the robot was always zero.

For each case three variants were tested. Firstly the behavior of the robot with full knowledge of the environment was simulated. Secondly, the control design was applied to real (physical) robot but also assuming full knowledge of the environment. Lastly, the behavior of the real robot was tested with online obstacle detection. Full knowledge of the environment meant that the exact positions and radii of obstacles were provided a priori to the controller. The initial conditions of the real robot state for second and third variants were firstly stabilized with omni directional youBot controller.

The simulated robot was affected by an artificial inertia of time constant equal 0.2 s applied to control signals, to imitate execution of real robot drives. All of design parameters were identical for all cases and variants. The list of parameters and their values is shown in Tab. 1.

The following subsections present the comparison of variants for each case. In the legends of figures, variant one was called *Simulation*, variant two: *Fixed obstacles*, and variant three: *Detected obstacles* (detection technique was described in [5]). One can note minor offsets in time between compared signals. This effect is the result of different delays between beginning of data recording and moment of controller launch. Although it could be corrected, it was left for better readability.

Table 1. Values of design parameters used for the experiments

Symbol	Name	Value
a	direct motion factor	0.5
$\bar{b}$	transverse motion factor	2.5
k_w	orientation importance	0.1
ϵ	convergence parameter	0.0001
κ	potential steepness	3.0
$\lvert\lvert\nabla V\rvert\rvert_{\min}$	potential gradient threshold	0.05
$V_{\min}$	potential threshold	0.07
ρ_0	world radius	5.0
$v_{\max}$	linear control saturation	0.5
$\omega_{\max}$	angular control saturation	3.0

7.1 Case I: $\boldsymbol{q}(0) = [2, 1.5, 0]^\top$

The first case was the simplest one. The platform orientation was in the beginning aligned with desired one, hence the robot did not have to reorient itself significantly. This is an important factor, because the navigation function seems to be sensitive to angular changes. The simulation provided us with decent looking curve of the the path taken during the run (Fig. 4).Resulting path seems to be directly applicable to car-like kinematic system but this remark needs further investigation.

The second variant, which involved experiment with a real robot, provided nearly identical results, which can be hardly distinguished by looking at the paths taken by the simulated and real robot.The path executed by robot with on-line obstacle detection was slightly more conservative because the detected obstacles appeared to be closer and had subtly larger radius. Any form of noise or erratic changes in detected obstacles state did not pose a significant impact on the robot behavior during the run.

The chart of navigation potential (Fig. 5) shows a non monotonic decay at about 10th and 13th second of the run (for the simulated variant). The small spikes of potential are the effect of applied system dynamics, which forbids an instantaneous stop during parallel parking maneuvers.

7.2 Case II: $\boldsymbol{q}(0) = [0, 1.5, \pi/2]^\top$

In the second case initial position of the robot was much closer to the goal position in comparison to previously presented one but the orientation error in the beginning was 90°. In all three variants the robots started with saturated linear velocity due to potential gradient normalization and performed parallel parking maneuvering in the end. What is interesting is an abrupt jerk occurring at about 5th second of the run. It cannot be seen directly from the path taken by the robots (Fig. 6) but it was evident while looking at the moving platform. A possible explanation of this phenomena can be the growing importance of orientation in the navigation potential, as the robot gets closer to the origin according to Eq. (5). The dynamics preventing the robot to stop immediately, pushed it onto a navigation potential of great steepness. This resulted in the spikes of control signals. Both linear and angular velocities get saturated when robot is located far form the target position.

7.3 Case III: $\boldsymbol{q}(0) = [-2, 1.5, \pi]^\top$

The third case was the most demanding because the robot started with orientation opposite to the goal. The reorientation process covered many maneuvers (Fig. 8) but in general had a smooth character (Fig. 9).Additional tests in simulations showed that by reduction of k_w parameter one could obtain a path with less maneuvers but at the cost of much greater settling time.

8. CONCLUSION

This work showed via simulation and experimental verification that the NF control approach can be exploited as a satisfactory method of performing parallel parking task. With appropriate normalization of potential gradient the automatic motion of the robot is human-acceptable even in the case of noisy perception of the environment. The normalization procedure needs further investigation for smooth transition in the neighborhood of the target.

Applying appropriate margin to the obstacles radii keeps the robot at a safe distance and it appears that the relative change in radii of about 20% does not significantly change the trajectory. Investigation of robot behavior in crowded and non-static environment also needs to be carried out.

Resulting robots paths in simulation and experiments were quite similar. It confirms that numerical simulation tests can be used for preliminary verification of this kind of control algorithm.

We can conclude, that the effectiveness of the NF algorithm for this kind of task was confirmed.

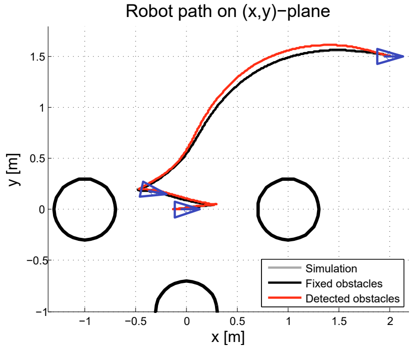

Fig. 4. Case I: Paths taken by the robots.

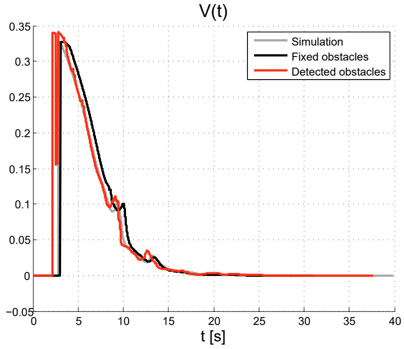

Fig. 5. Case I: Navigation potential V.

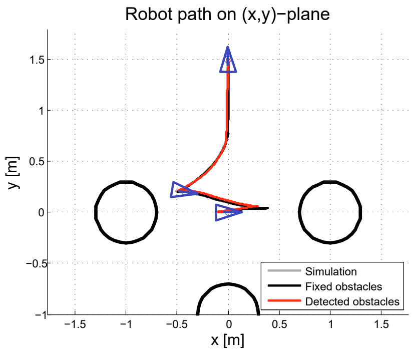

Fig. 6. Case II: Paths taken by the robots.

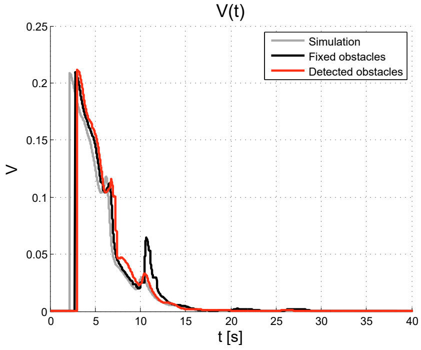

Fig. 7. Case II: Navigation potential V.

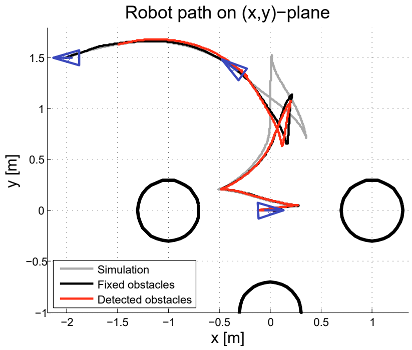

Fig. 8. Case III: Paths taken by the robots.

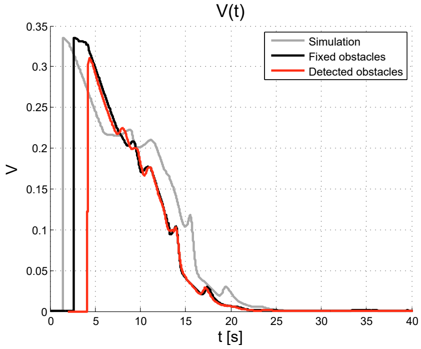

Fig. 9. Case III: Navigation potential V.

REFERENCES

[1] A. Bicchi, G. Casalino, C. Santilli, Planning shortest bounded-curvature paths for a class of nonholonomic vehicles among obstacles, IEEE International Conference on Robotics and Automation, Nagoya, 1995, Vol. 2, pp. 1349–1354.

[2] D.V. Dimarogonasa, S.G. Loizoua, K.J. Kyriakopoulos and M.M. Zavlanosb, A feedback stabilization and collision avoidance scheme for multiple independent non-point agents, Automatica, 2005, Vol. 42, No. 2, pp. 229–243.

[3] I. Filippidis and K.J. Kyriakopoulos, Adjustable navigation functions for unknown sphere worlds, IEEE Conference on Decision and Control and European Control Conference, 2011, pp. 4276–4281.

[4] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, The International Journal of Robotics Research, 1986, Vol 5, No. 1, pp. 90–98.

[5] W. Kowalczyk, M. Przybyla, K. Kozlowski, Control of a mobile robot and collision avoidance using navigation function - experimental verification, International Workshop on Robot Motion and Control, 2015, pp. 148–152.

[6] W. Kowalczyk, M. Przybyla, K. Kozlowski, Saddle point detection of the navigation function in nonholonomic mobile robot control, International Conference on Methods and Models in Automation and Robotics, 2016, pp. 936–941.

[7] I. Paromtchik, C. Laugier, Autonomous parallel parking of a nonholonomic vehicle, IEEE Intelligent Vehicles Symposium, 1996, pp. 13–18.

[8] G.G. Rigatos, Parallel parking control of autonomous vehicles using Extended Kalman and Particle Filtering, International Workshop on Robotics for Risky Interventions and Environmental Surveillance, 2010.

[9] E. Rimon and D.E. Koditschek, Exact robot navigation using cost functions: the case of distinct spherical boundaries, IEEE International Conference on Robotics and Automation, 1988, Vol. 3, pp. 1791–1796.

[10] E. Rimon and D. E. Koditschek, The construction of analytic diffeomorphisms for exact robot navigation on star worlds, Transactions of the American Mathematical Society, 1991, Vol. 327, pp. 71–116.

[11] E. Rimon and D. Koditschek, Exact robot navigation using artificial potential functions, IEEE Transactions on Robotics and Automation, 1992, Vol. 8, No 5, pp. 501–518.

[12] G. Roussos, K.J. Kyriakopoulos, Completely decentralised navigation of multiple unicycle agents with prioritisation and fault tolerance, IEEE Conference on Decision and Control, 2010, pp. 1372–1377.

[13] G. Roussos, K.J. Kyriakopoulos, Decentralized and prioritized navigation and collision avoidance for multiple mobile robots, Distributed Autonomous Robotic Systems - Springer Tracts in Advanced Robotics, 2013, Vol. 83, pp. 189–202.

[14] T. Urakubo, K. Okuma and Y. Tada, Feedback control of a two wheeled mobile robot with obstacle avoidance using potential functions, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004, Vol. 3, pp. 2428–2433.

[15] T. Urakubo, Feedback stabilization of a nonholonomic system with potential fields: application to a two-wheeled mobile robot among obstacles, Nonlinear Dynamics, 2015, Vol. 81, Issue 3, pp. 1475–1487.