Control of the Differentially-driven Mobile Robot in the Environment with a Non-Convex Star-Shape Obstacle: Simulation and Experiments

Wojciech Kowalczyk, Krzysztof Kozłowski

Poznań University of Technology, Chair of Control and Systems Engineering,
Piotrowo 3A, 60-965 Poznań, Poland
wojciech.kowalczyk@put.poznan.pl, krzysztof.kozlowski@put.poznan.pl

Abstract: The paper presents numerical verification and experimental results for the set-point control of the nonholonomic mobile robot. The task is to move to the goal and reach it with desired orientation avoiding collisions with static obstacles. The obstacles in the task space are modelled using analytic functions. The algorithm is investigated for both convex and non-convex star-shape obstacles.

Keywords: nonholonomic mobile robot; collision avoidance; navigation function; set-point control

1 Introduction

In the mid-1980s Khatib [1] presented a crucial idea of using repulsive and attractive interactions to avoid collision with obstacles and move to a set goal. This approach utilized local artificial potential functions (APF) associated with the obstacles and the goal. Local minima were at a significant disadvantage using this method.

In 1990 Rimon and Koditschek in series of publications [3], [4], [5], [6] proposed a navigation function that is a global artificial potential function without local minima. This method was applicable for a variety of task space configurations: sphere worlds, star worlds and trees of stars. The shapes of the obstacles and also the shape of the task space are described by analytic functions. The free space (task space) remains after all obstacles have been removed from the workspace. It must be emphasized that even in the case of the properly designed navigation function there still remain saddle points and their number is equal to the number of the obstacles.

In 2004 Urakubo [2] proposed an extension of the above approach. In this method nonholonomic constraints of the differentially driven mobile robot are taken into account. Urakubo also introduced methodology based on the time-varying functions such that the robot leaves the saddle point.

In the next years the navigation function was used to control multi-robot systems [8], [9], [10], [11]. In all of them robots are treated as obstacles. Problems of conflicts resolution between agents, limited knowledge of the environment and nonholonomic constraints of mobile platforms are addressed it these papers.

Another approach for collision-free control of nonholonomic mobile robot was proposed in [13]. The algorithm does not require the map of the environment. An iteratively generated path is optimized in free space (the curvature of the path) and replanning is done in the case of an unpredictable drift. The algorithm was verified numerically.

Algorithms that decompose the free space into subsets is another class of methods. In [14] the nonholonomic robot is driven through a sequence of triangular cells. This method was verified experimentally with process noise and both static and dynamic environments with obstacles.

In this paper simulation and experimental results for the algorithm presented in [2] are shown. According to the author's best knowledge this method has not been previously verified experimentally. In comparison to the original publication the effectiveness of the algorithm was investigated not only for a convex but also for non-convex obstacles.

It should be mentioned that local artificial potential functions can also be used to solve complex collision avoidance problems [7] in the case of circular obstacles; however, they must be designed carefully to avoid local minima.

In Section 2 the kinematic model of the robot and control algorithm introduced in [2] are presented. In Section 3 simulation results are shown. Section 4 briefly describes the experimental test-bed and experiments. Finally, the paper finishes off with concluding remarks.

2 Control Algorithm

The model of the differentially driven mobile robot is given by the following equation:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = B \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{1}$$

where x, y and θ are position and orientation coordinates of the robot respectively, $u=[v\,\omega]^T$ is control vector with v denoting linear velocity control and ω - angular velocity control of the mobile platform and

$$B = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}. \tag{2}$$

Control proposed in [2] is given by the following equation:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = -\left(a\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + b\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \right) B^T \nabla V \tag{3}$$

where

$$b = -\bar{b}\,\frac{L^T \nabla V}{h(g)} \tag{4}$$

and ∇V denotes the gradient of the artificial potential function, $L = [\sin\theta\ -\cos\theta\ 0]^T$, $h(g) = g^2 + \varepsilon_g \sqrt{g}$, $g = \left\| B^T \nabla V \right\|$. In the above equations a, $\bar{b}$ and ε_g are positive parameters.

The navigation function is given by the following equation:

$$V = \frac{C}{C^{\kappa} + \beta}, \tag{5}$$

where $C = \|\hat{r}\|^2 + w\theta^2$, $w = \dfrac{k_w}{k_w + \|\hat{r}\|^2}$; k_w is a positive parameter that allows tuning the weight of orientation when the robot approaches the desired position. As noted in [4] by setting a sufficiently large value of κ it is ensured that (4) is a navigation function i.e. there are no local minima in the APF.

To solve the problem of local minima for the case of star shape obstacles position coordinates of the robot $r = [x\ y]^T$ are transformed to the model sphere world as follows:

$$\hat{r} = \left(1 - \sum s_i(r)\right) r + \sum s_i(r) T_i(r) \tag{5}$$

where M – number of the obstacles, $r = [x\ y]^T$,

$$s_i(r)=\frac{\|r\|^2\prod\beta_j(r)}{\|r\|^2\prod\beta_j(r)+\lambda_s\beta_i(r)} \tag{6}$$

and

$$T_i(r)=\frac{\rho_i(1+\beta_i(r))}{\|r-q_i\|}(r-q_i)+p_i\ , \tag{7}$$

$$T_0(r)=\frac{\rho_0(1-\beta_0(r))}{\|r-q_0\|}(r-q_0)+p_0\ . \tag{8}$$

In the above equations p_0 , p_i represent the centers of the spheres to which original obstacles are transformed ρ_0 , ρ_i are their radiuses, q_0 , q_i are the centers of the stars (points from which all the rays cross the boundary of the obstacle once and only once) and β_0 , β_i are analytic obstacle functions.

The sphere-world obstacle function is a product of the sphere obstacle functions:

$$\hat{\beta}=\prod_{i=0}^{M}\hat{\beta}_i \tag{9}$$

where

$$\hat{\beta}_i(\hat{r})=\|\hat{r}-p_i\|^2-\rho_i^2\ , \tag{10}$$

$$\hat{\beta}_0(\hat{r})=\rho_0^2-\|\hat{r}-p_0\|^2\ . \tag{11}$$

3 Simulations Results

Simulation results were obtained for the task space containing one circular obstacle and one star-shape obstacle. The origin is the desired position and the desired orientation is equal to zero.

As the navigation function (4) is the function of three variables it is not possible to show its graph. In Figs. 1a and 2a the x-y plane cross-sections of V for $\theta=0$ and $\theta=\pi/2$ are shown, respectively. It can be observed that in the first case the global minimum exists in the center of the task space. In the second one the potential grows around the origin not allowing the robot to approach it with incorrect orientation. If the robot is too close it is repelled, but as the orientation converge to the desired value the x-y components of the V drive robot to the origin.

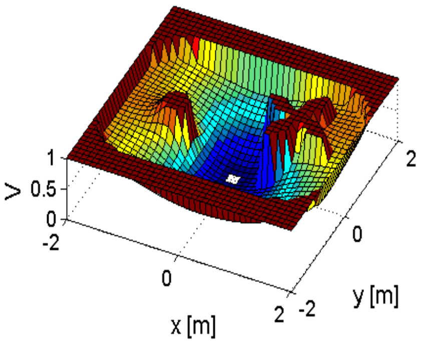

Figure 1a
APF for $\theta = 0$

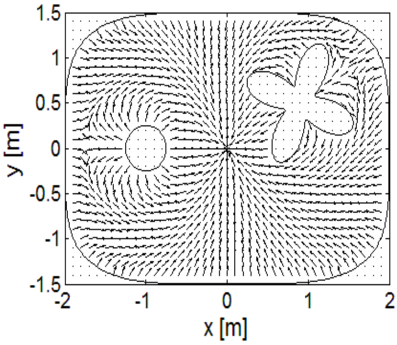

Figure 1b
Control vecor field $-\nabla V$ for $\theta = 0$

In Figs. 1b and 2b x-y plane cross-sections of the $-\nabla V$ for the same robot angles are shown (the length of the vectors was normalized).

The data were obtained for the following values of the parameters: $a = 0.25$, $\bar{b} = 1.25$, $\varepsilon_g = 10^{-6}$, $\kappa = 3$, $k_w = 0.1$, $\lambda_s = 10^6$.

In Fig. 3 the robot path in x-y plane is shown. The dashed line represents simulation results and solid line experimental data (their comparison will be discussed in the next section). The initial coordinates of the robot were as follows $[x \quad y \quad \theta]^T = [-1.01 \quad 0.92 \quad -2.92]^T$.

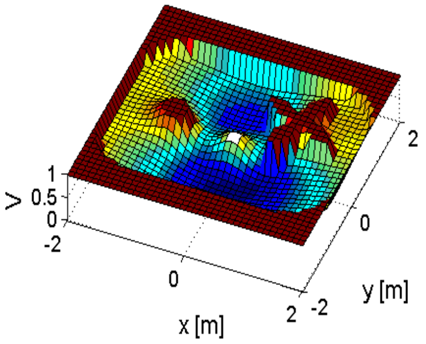

Figure 2a
APF for $\theta = \pi / 2$

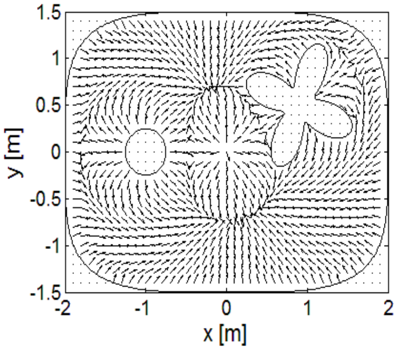

Figure 2b
Control vector field $-\nabla V$ for $\theta = \pi / 2$

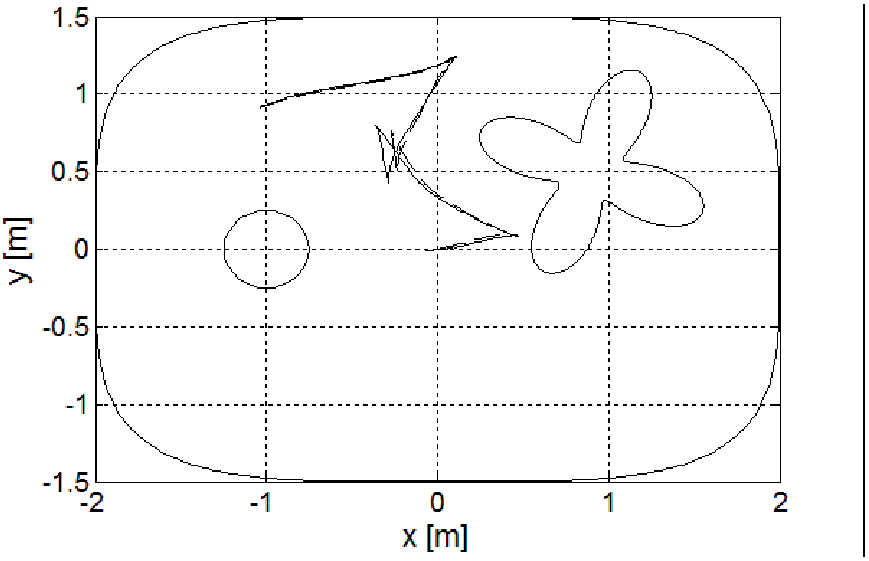

Figure 3
Robot motion in (x,y) plane; dashed line – simulation, solid line – experiment

In Fig. 4a time graphs of the position errors (e_x - solid line, e_y - dashed line) and orientation errors (e_θ - dot line) are shown. They converge to zero reaching small values in about 30 seconds.

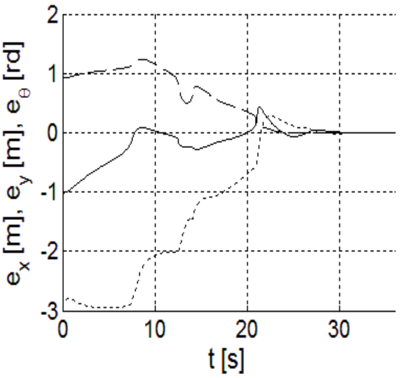

Figure 4a
State errors – simulation

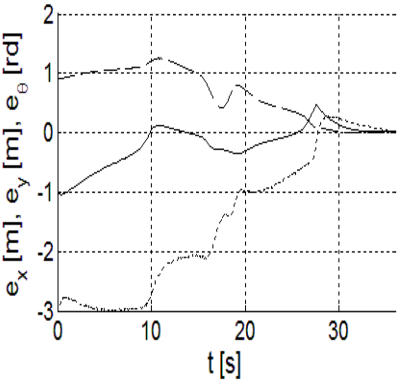

Figure 4b
State errors – experiment

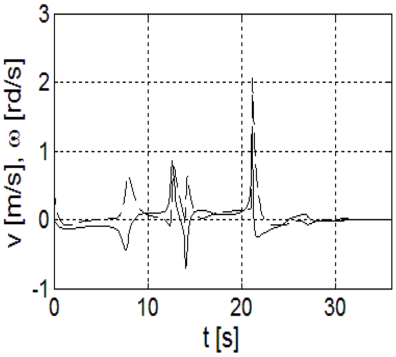

Figure 5a
Platform controls - simulation

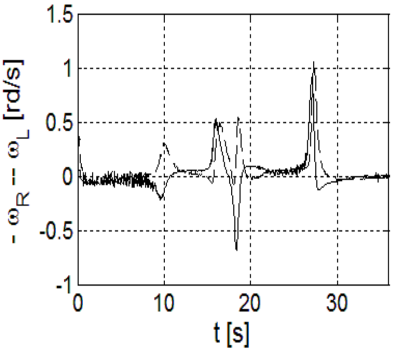

Figure 5b
Platform controls – experiment

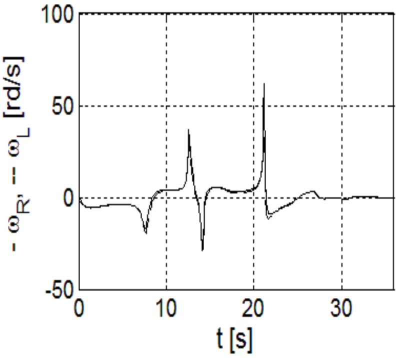

Figure 6a
Wheel controls - simulation

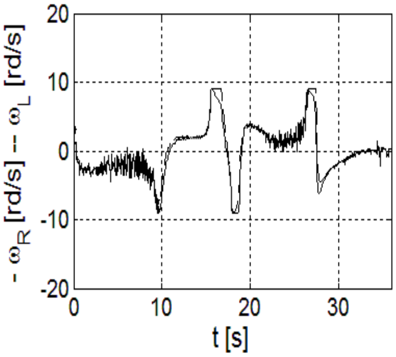

Figure 6b
Wheel controls – experiment

In Fig. 5a linear (solid line) and angular (dashed line) controls for the robot are presented.

Fig. 6a presents wheel control velocities (ω_R - solid line, ω_L - dashed line). There are peaks of the large values that cannot be achieved in the physical system. In the experiments presented in the next section these values are limited to achievable values.

In Fig. 7 robots path in x-y plane for a more complex case is shown. To reach the desired coordinates robot has to bypass the "leaf" of the obstacles. For this case local APF control algorithm usually fails trapping the robot in the local minimum. The initial coordinates of the robot were as follows $[x \quad y \quad \theta]^T = [1.22 \quad 0.65 \quad -0.51]^T$.

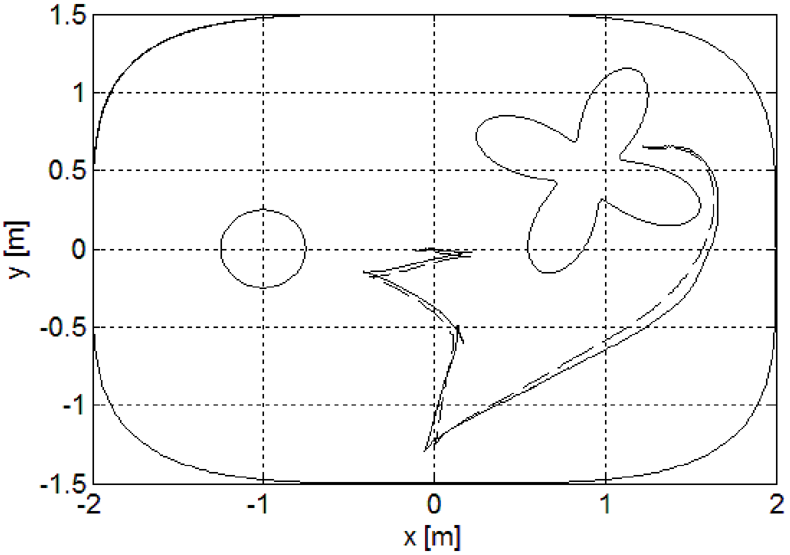

Figure 7
Robot motion in (x,y) plane; dashed line – simulation, solid line - experiment

In Fig. 8a position and orientation errors are presented, while Fig. 9a shows platform control signals. In Fig. 10a wheel control graphs are plotted.

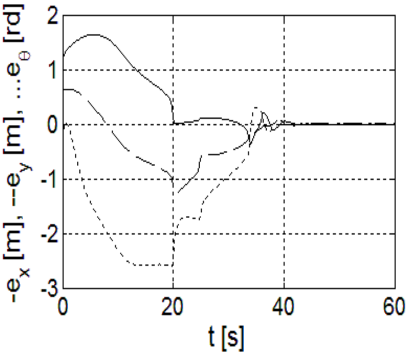

Figure 8a
State errors – simulation

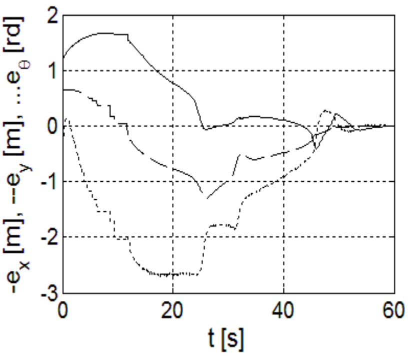

Figure 8b
State errors – experiment

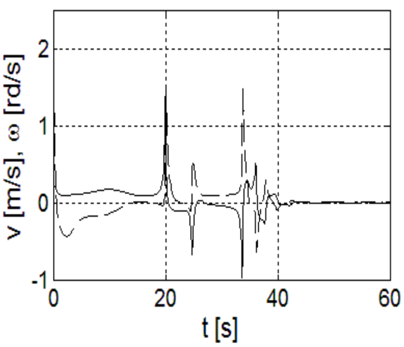

Figure 9a
Platform controls - simulation

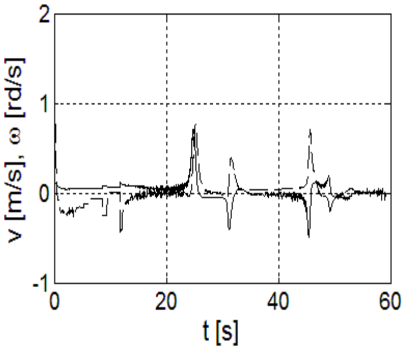

Figure 9b
Platform controls - experiment

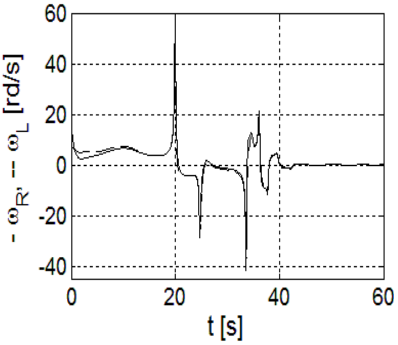

Figure 10a
Wheel controls - simulation

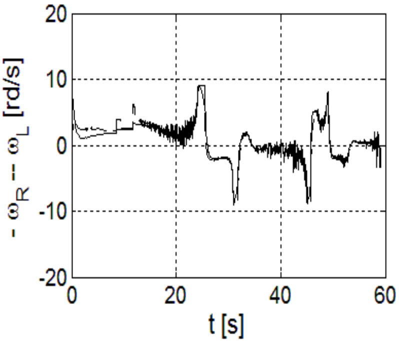

Figure 10b
Wheel controls - experiment

4 Experiments

4.1 Experimental Test-Bed

The algorithm was verified experimentally using the MTracker robot (Fig. 11). The MTracker is a differentially driven mobile robot. Its diameter is 170 mm and the height (in common configuration) – 65 mm. The linear velocity can reach the value of 1 m/s. The low level motion controller is implemented with the signal processor TMS 320F28335 150 MHz. Presented results were obtained using a robot expanded with Intel Atom 1,6 GHz PC board and Wi-Fi link used to obtain localization data from the external vision system. The robot was equipped with an LED marker for reliable and fast visual recognition. More details about the test-bed including robot configuration, communication and localization system can be found in [12].

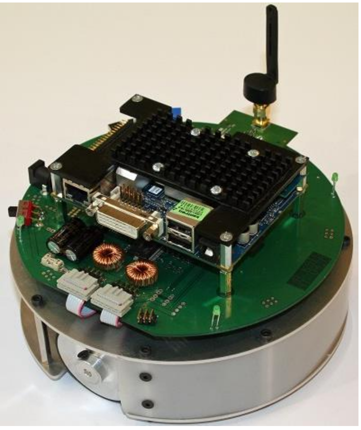

Figure 11
MTracker robot used for experiments

4.2 Experiments

In Fig. 3 path of the robot in x-y plane is presented. The experimental data are represented with a solid line. As can be observed in the figure both simulation and experimental results are very similar. In this experiment robot moves in open areas of the environment but it is required to change the orientation. As it approaches the origin the direction of the motion is changed four times to drive the orientation according to the control rule.

In Fig. 4b position and orientation errors are shown. They reach near zero values in 35 seconds.

In Fig. 5b platform control signals are shown. There are peaks that are not achievable in the real system. In Fig. 6b wheel controls are presented. They are limited ($\omega_{RL\ max} = 9rd/s$) and scaled to keep the robot on the path. Limitation and scaling of the wheel velocities is the best solution of the large control signal problem as it is not affecting the mobile platform direction vector (it affects only its evolution in time). This approach separates a high level controller; whose primary objective is to avoid collisions, form the problems connected with the physical actuator limitations.

In Fig. 7 both simulation and experimental results for the star bypassing case are shown. The solid line represents the experimental path of the robot in the x-y plane and dashed line represents the simulation data. As can be observed in the figures in simulation the robot changed the direction of motion three times. In the experiment one more motion direction change was performed. Which was due to the disturbances and/or delays in the positioning system, however both graphs are very similar.

In Fig. 8b position and orientation errors are presented. They reach the near zero value in about 60 seconds (in simulation 40 seconds).

In Fig. 9b and 10b control signals for the platform and wheels are shown respectively. As in the previous experiment wheel velocities were limited and scaled.

Conclusions

Simulation and experimental results for navigation function control were presented. The non-convex star shape obstacle makes presented cases nontrivial. As shown in the attached figures the nonholonomic mobile robot reaches the desired position and orientation avoiding collisions with both convex and non-convex obstacles.

List of notations

x, y - position coordinates of the robot

θ - orientation of the robot

v - linear velocity control

ω - angular velocity control

B - zero space of the Pfaffian matrix

a, $\bar{b}$, ε_g, κ, k_w, λ_s - positive constant parameters

V - navigation function

∇V - gradient of the navigation function

b - is defined by Eq. (4) ; is a weighting coefficient of the skew symmetric matrix

L - Pfaffian matrix

g - norm of the product zero space matrix and gradient of the navigation function

$h(g)$ - nonlinear function of g; used as coefficient in Eq, (4)

C, w - positive coefficients used to define Eq. (5)

r - robot position

$\hat{r}$ - robot position in the auxiliary sphere world

β - product of the all obstacle functions

β_i - obstacle function associated with the *i*-th obstacle

s_i, T_i - auxiliary variable for star-to-sphere transformation

p_i - center of the *i*-th sphere

ρ_i - radius of the *i*-th sphere,

q_i - center of the *i*-th star,

M - number of the internal obstacles,

e_x , e_y , e_θ - coordinate errors,

ω_R , ω_L - wheel velocities,

$\omega_{RL\,max}$ - wheel velocity limit.

Acknowledgement

This work was supported by NCBiR grant PBS1/A3/8/2012 (RobREx).

References

[1] O. Khatib, Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, The International Journal of Robotics Research, Vol. 5, No. 1, pp. 90-98, 1986

[2] 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 (IROS), Vol. 3, pp. 2428-2433, 2004

[3] 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, Vol. 3, pp. 1791-1796, 1988

[4] E. Rimon and D. Koditschek, Exact Robot Navigation using Artificial Potential Functions, IEEE Transactions on Robotics and Automation, Vol. 8, No. 5, pp. 501-518, 1992

[5] E. Rimon and D. E. Koditschek, The Constructions of Analytic Diffeomorphisms for Exact Robot Navigation on Star Worlds, Transaction of the American Mathematical Society, Vol. 327, pp. 71-116, 1991

[6] E. Rimon and D. E. Koditschek, Exact Robot Navigation using Artificial Potential Fields, IEEE Transactions on Robotics and Automation, Vol. 8, No. 5, pp. 501-518, 1992

[7] W. Kowalczyk, K. Kozlowski, and J. K. Tar, Trajectory Tracking for Multiple Unicycles in the Environment with Obstacles, International Conference on Robotics in Alpe-Adria-Danube Region (RAAD), pp. 451-456, 2010

[8] I. Filippidis and K. J. Kyriakopoulos, Adjustable Navigation Functions for Unknown Sphere Worlds, IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), pp. 4276-4281, 2011

[9] D. V. Dimarogonasa, S. G. Loizoua, K. J. Kyriakopoulos and M. M. Zavlanos, A Feedback Stabilization and Collision Avoidance Scheme for Multiple Independent Non-Point Agents, Automatica, Vol. 42, No. 2, pp. 229-243, 2005

[10] 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, Vol. 83, pp. 189-202, 2013

[11] G. Roussos, K. J. Kyriakopoulos, Completely Decentralised Navigation of Multiple Unicycle Agents with Prioritisation and Fault Tolerance, IEEE Conference on Decision and Control (CDC), pp. 1372-1377, 2010

[12] K. Kozlowski, W. Kowalczyk, B. Krysiak, M. Kielczewski, T. Jedwabny, Modular Architecture of the Multi-Robot System for Teleoperation and Formation Control Purposes," in Robot Motion and Control (RoMoCo), 2013 9th Workshop on, pp. 19-24, 3-5 July 2013

[13] M. K. Weir, M. P. Bott, High Quality Goal Connection for Nonholonomic Obstacle Navigation Allowing for Drift using Dynamic Potential Fields, in Robotics and Automation (ICRA), 2010 IEEE International Conference on, pp. 3221-3226, 3-7 May 2010

[14] V. Kallem, A. T. Komoroski, V. Kumar, Sequential Composition for Navigating a Nonholonomic Cart in the Presence of Obstacles, in Robotics, IEEE Transactions on, Vol. 27, No. 6, pp. 1152-1159, Dec. 2011