

Obstacles Avoidance Algorithm for Mobile Robots, Using the Potential Fields Method

Vesna Antoska-Knights^{1,*}, Zoran Gacovski², Stojce Deskovski³

¹Faculty of Technology and Technical Sciences Veles 1400, University St. Kliment Ohridski-Bitola, Republic of Macedonia

²European University, Republic of Macedonia

³Technical Faculty, University "St. Clement Ohridski", Republic of Macedonia

Copyright©2017 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

Abstract In this paper - a mobile robot guidance and control has been researched in the environment full of obstacles, by using the potential fields method. The mobile robot has 4-wheels configuration, electric drive on the rear vehicles, and is directed from the front wheels (Ackerman control algorithm). A known environment has been considered, where fixed potentials were assigned to the goal and the obstacles. When the obstacles are unknown - the potential fields have to be applied, as the robot moves and detect new obstacles. A potential field's method was applied with one attraction potential assigned to the goal point and fixed rejection points assigned to the obstacles. It moves successfully within different obstacle configurations (closely spaced obstacles), and it solves the problem with a local minimum occurrence. The simulation results showed small and stable tracking errors along 2 axes.

Keywords Mobile Robots, Guidance and Control, Trajectory Planning, Potential Fields Method, Obstacle Avoidance

1. Introduction

The autonomous robot navigation problem consists of the determining the possible path between two points, an initial and a final (goal) point. The local navigation method should produce an optimal (possibly shortest) path, avoiding the obstacles present in the moving environment. Usually, the obstacles and the target could be fixed or dynamic. The goal of the path planning method is to determine the robot's movement, but to avoid collisions while reaching the destination. Potential fields are a good approach to the path determining problem, [1]. Different authors have undertaken different approaches to calculate appropriate field configurations.

What is the basic idea of potential fields? It is to

calculate attraction and rejection forces within the working environment of the robot to guide it to the target. The robot is treated as an object under the impact of an artificial potential field. Space is represented with an array (or field) of vectors; vectors consist of magnitude (Length of arrow) and direction (Angle of arrow). Generated robot movement is similar to a ball rolling down a hill. Goal generates attractive force, while obstacles are repulsive forces. That is the control law for the robot's motion. A vector space will be used, as a 2D map which will be divided into squares, creating a (x,y) grid. Each element represents a square of space and perceivable objects exert a force field on the surrounding space. Other types of potential fields are: Uniform, Perpendicular and Tangential.

The robot uses functions defining potential fields at its position to calculate component vector, [2]. The entire field doesn't have to be computed - only the portion of field influencing the robot will be computed - for each behaviour of the potential field; the sum of the vectors at the robot's position will get resultant output vector. If the sum of the vectors is zero, that is the local minima; if the robot comes to the local minima, it will stay still. There are issues with combining potential fields; the impact of update rates is that lower update frequency can lead to ragged (irregular) paths. If the robot is treated as a mass object, it cannot be expected to change velocity and direction spontaneously (cannot happen). The task is to find a solution for local minimum problem; if the global minimum is not guaranteed, a different approach than gradient descent should be chosen. The functions should be chosen in such a way that global minimum can be guaranteed and the robot will escape the local minima. "Avoid-past" behaviours can be included - remembering where robot has been and bringing the robot to other places. Numerical techniques, Random walk methods and Navigation functions, i.e. "navigation templates" have been combined by [3], to get the resulting vector that will avoid local minima. They give "avoid" behaviour a preferred direction and insert tangential fields around obstacles.

There are approaches that are able to produce the resultant forces required to avoid a set or several obstacles in a close distance, [4]. Several additional attractions points with flexible position and force magnitude will enable movement around big obstacles, as well as through closely placed obstacles, but the field optimization cost will be enlarged. Wu et al.[5], have successfully applied genetic algorithms, in order to optimise the potential fields with a large number of unknown obstacles and four auxiliary attraction points. Their method is rather fast and can be used for real-time control of a mobile robot.

This paper is organized as follows: in Section 2 we present the kinematic and dynamic model of the 4-wheeled mobile robot; in Section 3 - the Guidance and control method is presented, and in Section 4 - the trajectory planning algorithm using the potential field method. Finally we present the simulation results, our summary conclusions and directions for further work.

2. Mathematical Model of the Mobile Robot

In this section, the robot movement in horizontal plane will be elaborated. A mobile robot with 4 wheels (Ackerman drive) was considered, with fixed rear wheels, and steering front wheels. Fig.2.1 shows the robot geometry while moving in 2D space. The kinematic and dynamic equations for the movement of the mobile robot can be found in [6, 7, 8], and here we'll use these equations without derivation.

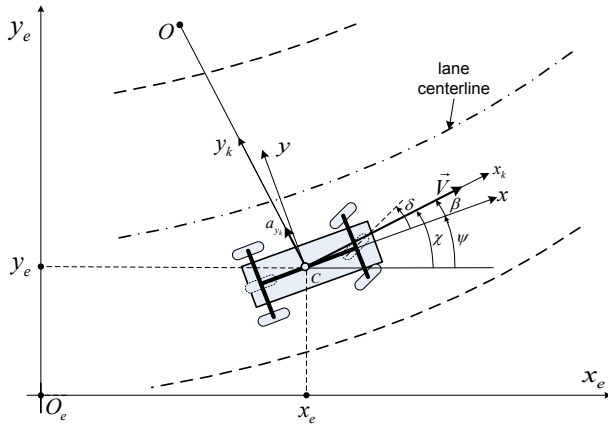


Figure 2.1. Kinematics of lateral vehicle motion

In many cases, when the kinematics and dynamic model of a car-like robot is required - a bicycle model is used, where front and rear vehicles are presented with a single wheel, as in Fig.2.2, [6].

The movement of the mobile robot in horizontal plane can be described as a system with 3-degrees of freedom—two translations of the mass centre (C) along x and y axes, and one rotation around z axis. Three coordinate

systems are used: inertial coordinate system - fixed to the ground (global) $E(O_e; x_e, y_e)$, coordinate system fixed to the robot body (local) $B(C; x, y)$ and coordinate system attached to the robot trajectory $K(C; x_k, y_k)$, where x_k is axis - tangent, and y_k is axis normal to the trajectory.

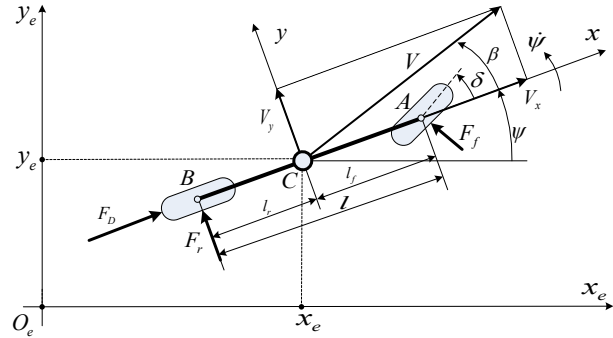


Figure 2.2. Bicycle model of lateral mobile robot dynamics

The kinematic equations of the mobile robots are:

$$\begin{aligned} \dot{x}_e &= V \cos(\psi + \beta) \\ \dot{y}_e &= V \sin(\psi + \beta) \\ \dot{\psi} &= \frac{V \cos \beta}{l_f + l_r} \operatorname{tg} \delta \end{aligned} \quad (2.1)$$

where: V is velocity of centre of gravity (c.g.) of the vehicle which is at point C, ψ is yaw angle (orientation angle with respect to global coordinate x_e), β is vehicle slip angle, $\chi = \psi + \beta$ is the angle of turn of the vehicle, δ is the steering angle of front wheels, l_f and l_r are distances of points A and B from centre of gravity of the vehicle, respectively, $l = l_f + l_r$ is wheelbase of the vehicle.

Dynamic equations can be derived from Fig. 2.2 and they are:

$$\begin{aligned} m(\dot{V}_x - V_y \dot{\psi}) &= -F_f \sin \delta + F_D \\ m(\dot{V}_y + V_x \dot{\psi}) &= F_f \cos \delta + F_r \\ J\ddot{\psi} &= l_f F_f \cos \delta - l_r F \end{aligned} \quad (2.2)$$

where m and J are mass and inertial moment of the vehicle around the mass centre, F_D is driving force on the rear axis - along x axis, F_f and F_r are resultant side forces on the front and the rear wheel. The first two equations in (2.2) define the forces balance along the system axes $B(C; x, y)$, while the third equation in (2.2), defines the moments' balances around the z axis, normal to the plane Cxy .

The obtained non-holonomic constraints of the mobile robot (vehicle) are:

$$\begin{aligned}\dot{\psi} &= \frac{\tan \delta}{l} V_x, \\ V_y &= \dot{\psi} l_r,\end{aligned}\quad (2.3)$$

and their derivations have the form:

$$\begin{aligned}\ddot{\psi} &= \frac{\tan \delta}{l} \dot{V}_x + \frac{V_x}{l(\cos \delta)^2} \dot{\delta}, \\ \dot{V}_y &= \ddot{\psi} l_r,\end{aligned}\quad (2.4)$$

Using the kinematics equations (2.1), dynamic equations (2.2), and non-holonomic connection (2.3) and its derivatives (2.4), applying some math computation, the following equations were obtained - i.e. mathematical model of the mobile robot:

$$\begin{aligned}\dot{x}_e &= (\cos \psi - C_l \tan \delta \sin \psi) V_x \\ \dot{y}_e &= (\sin \psi + C_l \tan \delta \cos \psi) V_x \\ \dot{\psi} &= \frac{\tan \delta}{l} V_x\end{aligned}\quad (2.5)$$

$$\dot{V}_x = \frac{1}{d} [V_x J_{eq} \tan \delta \dot{\delta} + l^2 (\cos \delta)^2 F_D]$$

$$d = (\cos \delta)^2 [C_m + J_{eq} (\tan \delta)^2]$$

where parameters C_l , J_{eq} , C_m are computed with:

$$C_l = \frac{l_r}{l}, \quad J_{eq} = l_r^2 m + J, \quad C_m = l^2 m. \quad (2.6)$$

The driving force F_D is generated by a DC motor which rotates the rear wheels, and can be computed by:

$$\dot{F}_D = -C_{FD} F_D - C_{Vx} V_x + C_{u2} u_2, \quad (2.7)$$

while the parameters can be computed with:

$$\begin{aligned}C_{FD} &= \frac{R_a}{L_a}, \\ C_{Vx} &= \frac{(K_m K_b + R_a b_m) N_w^2}{L_a N_m^2 R_w^2}, \\ C_{u2} &= \frac{K_m N_w}{L_a N_m R_w}\end{aligned}\quad (2.8)$$

where: R_a - resistance in armature circuit, L_a - inductivity in the armature circuit, K_m , K_b , b_m - motor constants, N_w - number of cogs in the wheels' gear, N_m - number of cogs in the motor axis, R_w - wheel radius.

The controller (actuator) for the steering wheels can be described as a 1-st order system:

$$\dot{\delta} = -\frac{1}{\tau_a} \delta + c_a u_1 \quad (2.9)$$

where c_a is reinforcement, and τ_a - time constant of the actuator.

The mobile robot model obtained by the upper equations is displayed in Fig.3.2. Input in the model are the control signals u_1 - for front wheels control, u_2 - for velocity control. The output of the system is the complete state vector:

$$\mathbf{x} = [x_e, y_e, \psi, V_x, F_D, \delta]^T \quad (2.10)$$

and other variables that are dependent on the robot state.

3. Robot Guidance and Control

The robot guidance and control system should provide accurate trajectory following which can be known in advance, or can be computed in real time, [9].

Fig.3.1 shows the geometry during the referent trajectory following by the mobile robot. This figure can help us in determining the kinematic equations and the errors in following. The point C_d of the referent trajectory defines the required robot position given with the global coordinates x_{ed}, y_{ed} . The real position is determined by the point C (c.g. - centre of gravity of mobile robot) whose coordinates are x_e, y_e . The error in following $\vec{e} = \vec{R}_d - \vec{R}$ can be represented in the projections in the global $\mathbf{e}_E = [e_{x_e}, e_{y_e}]^T$ and in local coordinates $\mathbf{e}_B = [e_x, e_y]^T$, and the transformation is:

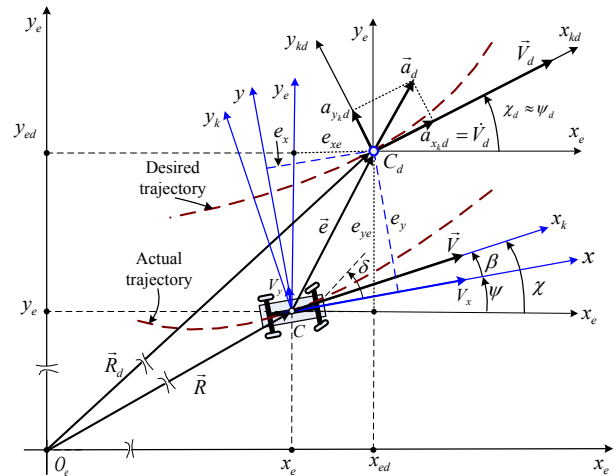


Figure 3.1. Tracking reference (desired) trajectory with a mobile robot

$$\mathbf{e}_B = \mathbf{T}_{BE} \mathbf{e}_E,$$

or

$$[e_x, e_y]^T = \mathbf{T}_{BE} [e_{x_e}, e_{y_e}]^T \quad (3.1)$$

where \mathbf{T}_{BE} is transformation matrix from global E to local

B coordinate system, and is given by:

$$\mathbf{T}_{BE} = \mathbf{T}_{\psi} = \begin{bmatrix} \cos \psi & \sin \psi \\ -\sin \psi & \cos \psi \end{bmatrix} \quad (3.2)$$

One way to generate the referent trajectory, given the accelerations along the trajectory tangent and normal respectively- $a_{x_k d}$, $a_{y_k d}$, is to apply the following equations:

$$\begin{aligned} \dot{V}_d &= a_{x_k d}, \quad V_d(0) = V_{d0} \\ \dot{\psi}_d &= \frac{a_{y_k d}}{V_d}, \quad \psi_d(0) = \psi_{d0} \end{aligned} \quad (3.3)$$

$$\begin{aligned} \dot{x}_{ed} &= V_d \cos \psi, \quad x_{ed}(0) = x_{ed0} \\ \dot{y}_{ed} &= V_d \sin \psi, \quad y_{ed}(0) = y_{ed0} \end{aligned} \quad (3.4)$$

The diagram of the control system is given in Fig.3.2. The 'Mobile robot' block computes the equations (2.5), the block 'Steering wheel' solves (2.9), the block 'Driving force' generates the driving force – (2.7), the block 'Transformation to robot coordinates' computes the errors:

$$\begin{aligned} e_{xe} &= x_{ed} - x_e, \\ e_{ye} &= y_{ed} - y_e, \\ e_{\psi} &= \psi_d - \psi \end{aligned} \quad (3.5)$$

transforming them in local coordinate system with (3.1), the 'Reference trajectory' block computes the referent trajectory. To obtain better guidance accuracy, especially when the trajectory needs fast manoeuvres, together with the feedback controller - a feedforward controller based on the inverse robot model must be applied. The controller (its algorithm) generates control signal u_1 to control the steering vehicles. To generate the signal u_1 - a PID controller has been used. The signal u_2 defines the given velocity of the robot movement along the given trajectory.

Based on the above model and the block diagram shown on Fig.3.2, a simulation model in MATLAB/Simulink was developed, which can be used for testing of the guidance algorithm. The generation of the referent trajectory in an obstacle environment is described in the following section.

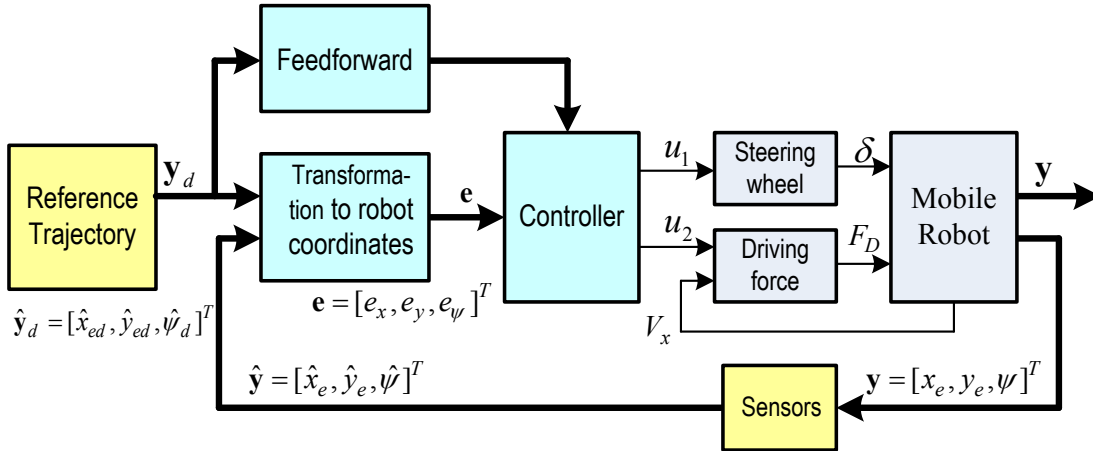


Figure 3.2. Blok diagram of mobile robot guidance and control system

4. Trajectory Planning by Potential Fields

The basic idea of trajectory planning by potential fields is very practical. Robot is presented as a point in the space and the environment is represented as a potential field, [10]. Potential field is an array of vectors which represent the space, and the vectors represent forces. Each vector consists of length of magnitude (m) and angle of arrow or direction (d).

The following equations and parameters are valid for the goal:

(x_G, y_G) denotes the position of the goal, (x_O, y_O) denotes the position of the obstacle, d is the radius of the goal, $V = [x, y]^T$ gives the (x, y) coordinate position of the point (robot). Distance (d) between the goal and the robot is:

$D = \sqrt{(x_G - x)^2 + (y_G - y)^2}$, and $D = \sqrt{(x_O - x)^2 + (y_O - y)^2}$ is the distance between the obstacle and the robot,

$\theta = \tan^{-1} \left(\frac{y_G - y}{x_G - x} \right)$ - angle between the robot and the goal,

α represents the strength of the attraction potential of the goal.

Programming a single potential field - can be done by a repulsive field with linear drop-off:

$$V_{direction} = 180^0$$

$$V_{magnitude} = \begin{cases} D-d & \text{for } d \leq D \\ D & \text{for } d > D \end{cases}, \quad (4.1)$$

where D is the max range of the field's effect.

To generate potential field function $U(q)$ for the robot - we will start with the force acting on a robot at point q : $F(q) = -\nabla U(q)$, and environment is represented by potential function: $U(x, y)$. Force is proportional to the gradient of the potential function:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = -\nabla U(x, y) \quad (4.2)$$

Environment is sum of the potential of fields, [10]:

$$U(q) = U_{att}(q) + U_{rep}(q) \quad (4.3)$$

where, $U_{att}(q)$ is attracting (goal) and $U_{rep}(q)$ is repulsing (obstacle) fields.

The gradient of the potential function must be differentiable:

$$\nabla U = \begin{bmatrix} \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \end{bmatrix} \quad (4.4)$$

From above equations, artificial force field acting on a robot $F(q)$, as the gradient of the potential field is given by:

$$F(q) = -\nabla U(q)$$

$$F(q) = F_{att}(q) + F_{rep}(q) = -\nabla U_{att}(q) - \nabla U_{rep}(q) \quad (4.5)$$

Converting to robot control, the robot velocity is set to (v_x, v_y) , proportional to the force $F(q)$ generated by the field, the force field drives the robot to the goal and the robot model is derived in section 3.

Attraction potential field is defined with: $U_{att} = \frac{1}{2} \xi_{att} d^2$ where $d = \|q - q_{goal}\|$; q is the current position of the robot and q_{goal} is the position of the attraction point; ξ_{att} is adjustable positive scaling factor (constant).

In the attractive potential field these are the main functions: linear function of distance, quadratic function of distance and mix of the both.

Linear function of the distance is:

$$U_{att}(q) = \xi_{att} \|q - q_{goal}\|,$$

$$F_{att}(q) = -\nabla U_{att}(q) = -\xi_{att} \frac{(q - q_{goal})}{\|q - q_{goal}\|} \quad (4.6)$$

where ξ_{att} is a positive scaling factor, and $q - q_{goal}$ is the distance.

- Quadratic function of distance is given by:

$$U_{att}(q) = \xi_{att} \frac{1}{2} \|q - q_{goal}\|^2 \quad (4.7)$$

Attracting force will converge linearly towards 0 (goal):

$$F_{att}(q) = -\nabla U_{att}(q) = -\xi_{att} \|q - q_{goal}\| \nabla \|q - q_{goal}\| \quad (4.8)$$

$$= -\xi_{att} (q - q_{goal})$$

Repulsive potential field will generate different barriers around the obstacles: strong, if the obstacle is near, or zero - if the obstacle is far away:

$$U_{rep}(q) = \begin{cases} \frac{1}{2} \xi_{rep} \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0} \right) & \text{if } \rho(q) \leq \rho_0 \\ 0 & \text{if } \rho(q) \geq \rho_0 \end{cases} \quad (4.9)$$

where $\rho(q)$ is the minimal distance to the obstacle from q ; ρ_0 is distance of influence of obstacle.

Field is larger or equal to zero and leans to infinity, as q approaches the obstacle:

$$F_{rep}(q) = -\nabla U_{rep}(q)$$

$$= \begin{cases} \xi_{rep} \left(\frac{1}{\rho(q)} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2(q)} \frac{q - q_{obstacle}}{\rho(q)} & \text{if } \rho(q) \leq \rho_0 \\ 0 & \text{if } \rho(q) \geq \rho_0 \end{cases} \quad (4.10)$$

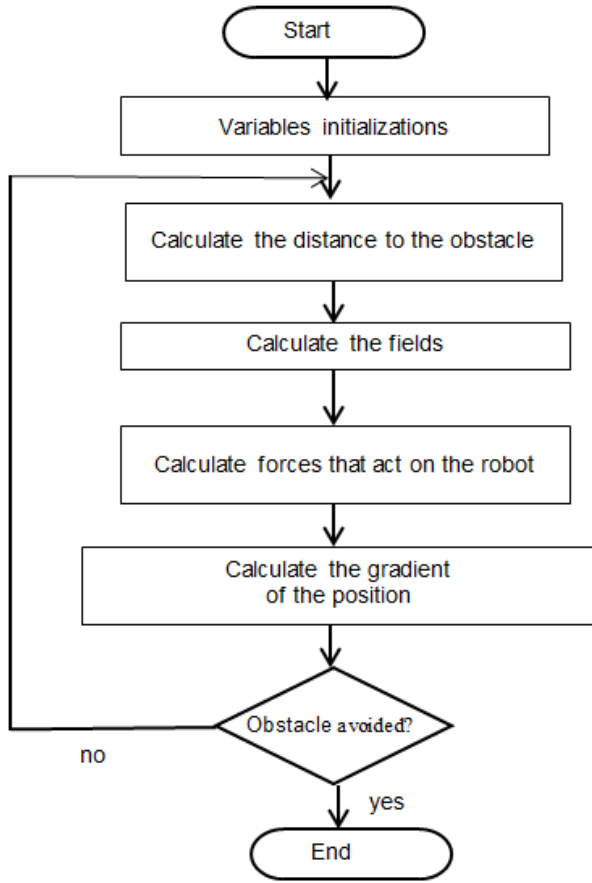


Figure 4.1. Block diagram for calculation of the potential fields

This formula is popular because it is elegant and simple. However, there could be oscillations and local minima situations with some obstacle positions, and possible problems include: trap situations due to local minima, oscillations in tight corridors, or difficulty of moving between closely spaced obstacles. Some other potential fields algorithms have been introduced in the past in order to solve these problems. Above equations for rejection forces can be modified in a case with non-reachable target when it is near the obstacles, since when the robot approaches the goal near an obstacle, the attraction force decreases and becomes drastically smaller than the increasing rejection force. The modified rejection potential takes the form of:

$$U_{rep} = \begin{cases} \frac{1}{2} \xi_{rep} \left(\frac{1}{d_{(q)}} - \frac{1}{d_0} \right)^2 |q - q_{goal}|^n & d_{(q)} \leq d_0 \\ 0 & d_{(q)} \geq d_0 \end{cases} \quad (4.11)$$

For potential field path planning harmonic potentials have been used, since the robot is moving in an environment with fixed obstacles, ensuring that there are no local minima. The derived harmonic potential of the point charge q is:

$$U(r) = \begin{cases} \frac{q}{r^{n-2}} & n=1,3,4,\dots \\ q \ln \frac{1}{r} & n=2 \end{cases} \quad (4.12)$$

where $r = \sqrt{\sum_n x_n^2}$, $x = (x_1, x_2, \dots, x_n)$ and the gradient $-\nabla U$ is described by:

$$-\nabla U(r) = \begin{cases} (n-2) \frac{q}{r^{n-1}} \vec{e}_r & n=1,3,4,\dots \\ \frac{q}{r} \vec{e}_r & n=2 \end{cases} \quad (4.13)$$

Where \vec{e}_r denotes a unit vector in radial direction.

5. Simulation Results

In accordance with the block diagram in Fig.3.2 and equations that describe each module, we developed the simulation model of the mobile robot in Matlab/Simulink. The parameters of the mobile robot which are used in the simulation are: $l = 0.2540$ [m] - distance between rear wheel and front wheel; $b = 0.1651$ [m] - space between wheels; $H = 0.0762$ [m] - height; $D = 0.0635$ [m] - diameter of wheels; $F = 0.0318$ [m] - width of rear wheels; $R = 0.0508$ [m] - width of rear wheels; $l_r = 0.0889$ [m] - distance from CG to rear axle; $l_f = 0.1651$ [m] - distance from CG to front axle; $R_w = 0.0318$ [m] - wheel radius; $m = 1.4175$ [kg] - mass of the mobilniot robot; $J = 0.0594$ [kg*m²] - inertial moment of the robot; $c_a = 1$ gain and $\tau_a = 0.01$ [s] - time constant of the steering wheels actuator. Electromotor parameters: $R_a = 1.9$ [Ω] - resistance in armature circuit; $L_a = 1.0640e-04$ [H] - inductivity of the armature circuit; $N_w = 81$ - number of cogs in the motor gear; $N_m = 21$ - number of cogs in the motor gear; $b_m = 3.3970e-05$ [Nms/rad], $K_m = 0.0680$ [Nm/A], $K_b = 0.0680$ [Vs/rad] - motor constants; $J_m = 0.002$ [Nms²] - moment of inertia of the motor.

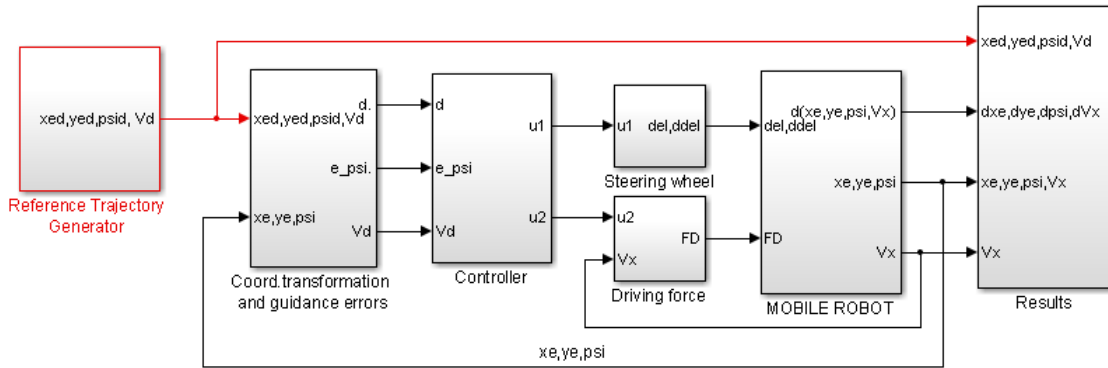


Figure 5.1. Mobile robot control system model - developed in Simulink

The first experiment for robot’s guidance and control is based on a case with a circled reference trajectory with a given radius R_d and given constant velocity along the circle V_d . For reference trajectory generation the equations (3.3) and (3.4) were applied. In the simulation we’ve used the initial values: $V_d(0) = 1\text{ m/s}$, $\psi_d(0) = 0\text{ rad}$, $x_{ed}(0) = 0\text{ m}$, $y_{ed}(0) = 0\text{ m}$. The tangent acceleration of the vehicle is $a_{x_k,d} = 0\text{ m/s}^2$, and the normal (centripetal) acceleration for $R_d = 10\text{ m}$ and $V_d = 1\text{ m/s}$ is $a_{y_k,d} = V_d^2 / R_d = 0.1\text{ m/s}^2$. The initial conditions for the mobile robot are: $x_e(0) = 0\text{ m}$, $y_e(0) = 0\text{ m}$, $\psi(0) = 0\text{ rad}$, $V_x(0) = 0.5\text{ m/s}$, $F_D(0) = 0\text{ N}$, $\delta(0) = 0\text{ rad}$. Simulation results for the robot’s motion along circular trajectory are given in Fig.5.2 to Fig.5.7.

Fig. 5.2 shows the robot movement along circular trajectory with $R_d = 10\text{ m}$. Fig.5.3 shows the course angle change during the motion along the reference trajectory, as well as the error of following.

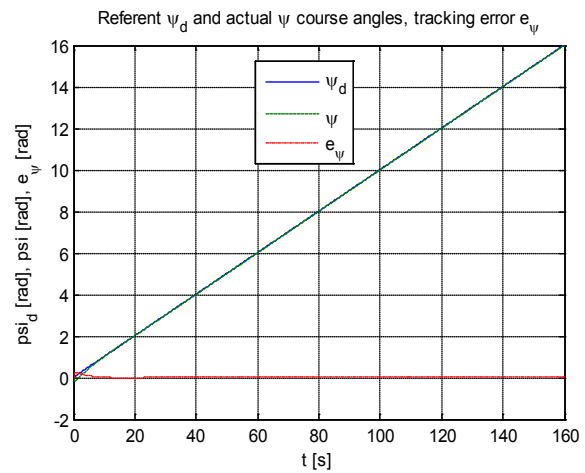


Figure 5.3. Robot course angle following during the movement along circular referent trajectory

Figures 5.4, 5.5 represent the errors of trajectory following along the x_k, y_k axes attached to the robot trajectory $K(C; x_k, y_k)$.

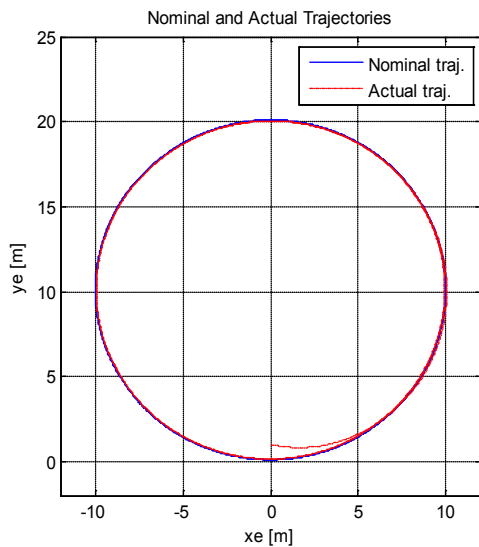


Figure 5.2. Mobile robot movement along circled referent trajectory

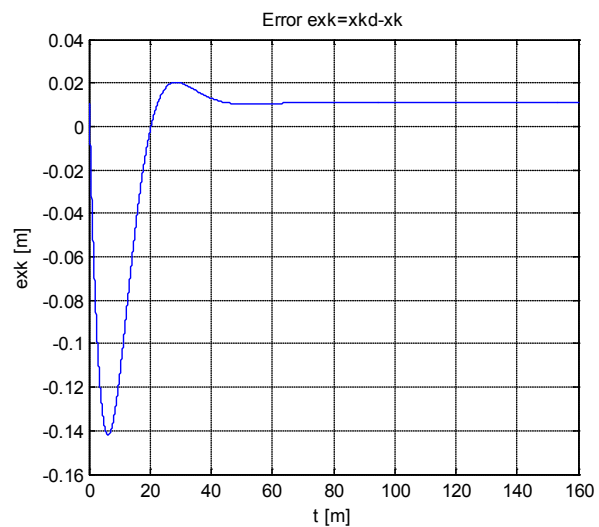


Figure 5.4. Tracking error along x_k axis

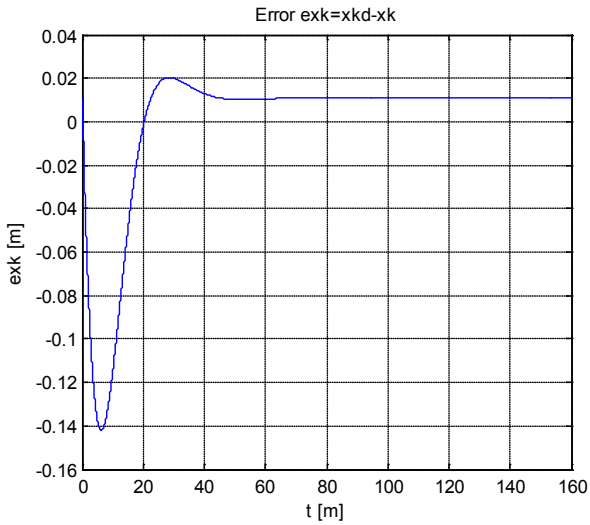


Figure 5.5. Tracking error along y_k axis.

Fig. 5.6, 5.7 show the change of the nominal and actual coordinates of the robot relatively to the global inertial (coordinate) system $C(O, x_e, y_e)$, as well as the errors in following.

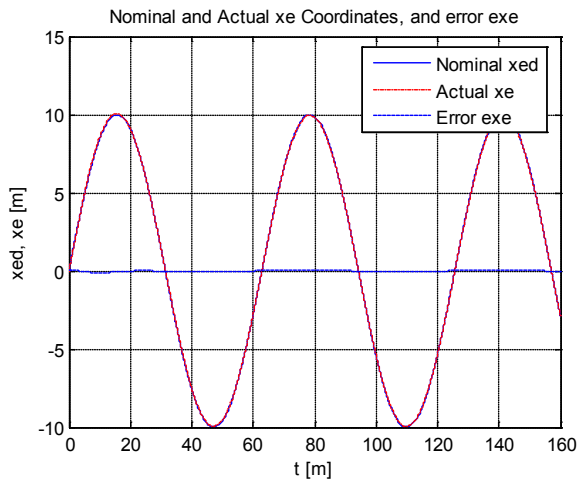


Figure 5.6. Nominal and actual coordinates and tracking error along x_e axis.

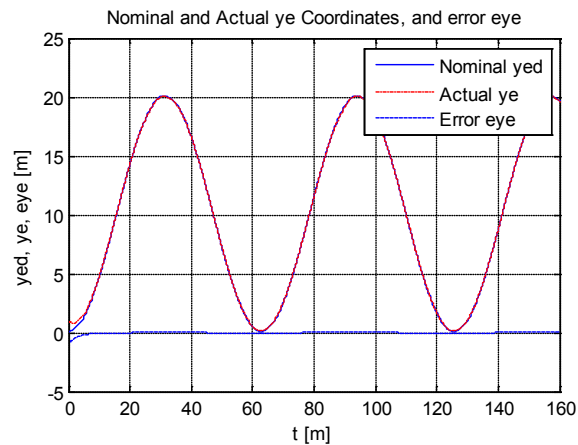


Figure 5.7. Nominal and actual coordinates and tracking error along y_e axis.

In the next experiment we have simulated the robot motion in the obstacle environment. Referent trajectory in an obstacle environment is generated in the block Reference Trajectory Generator using potential field method. Fig. 5.8 shows how the mobile robot follows this referent trajectory. The obstacles in Fig.5.8 are presented by circles.

Fig 5.9 represents the robot course angle following during the movement along the referent trajectory.

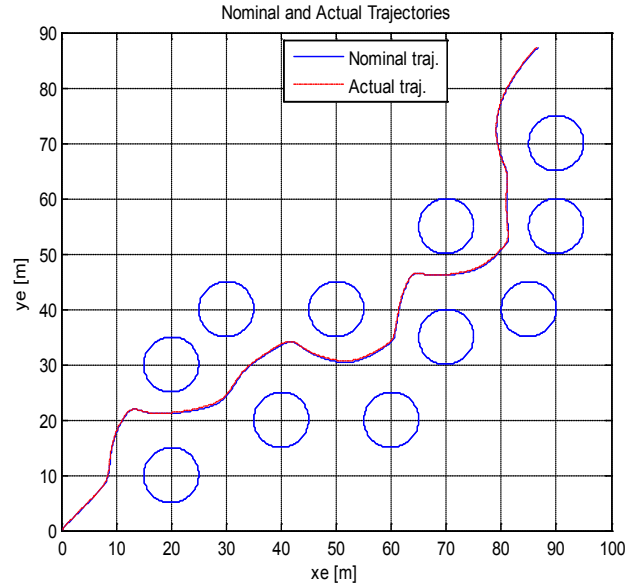
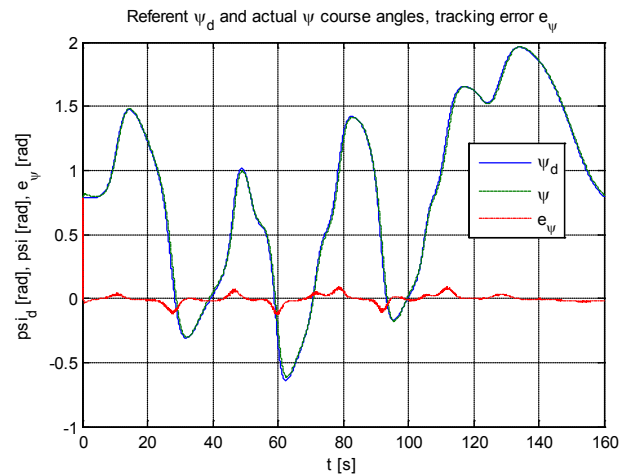


Figure 5.8. Referent trajectory following



ψ_d - nominal angle, ψ - real angle, e_ψ tracking error

Figure 5.9. Robot course angle following during the movement along referent trajectory

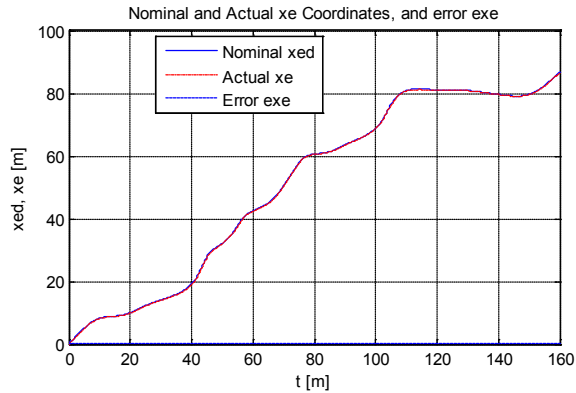


Figure 5.10. Trajectory following along x axis

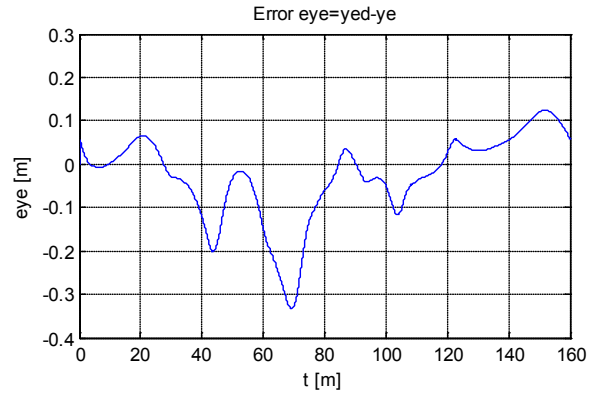


Figure 5.13. Tracking error along y_e axis

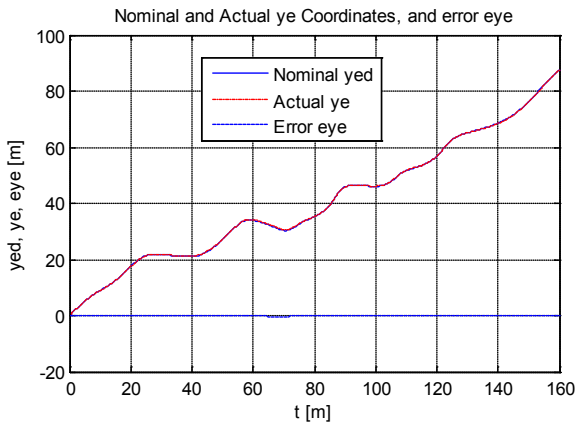


Figure 5.11. Trajectory following along y axis

Fig.5.10, and Fig.5.11 present trajectory following along x, and y axes, respectively. Tracking errors along x_e and y_e axes are given in Fig.5.12 and Fig.5.13 respectively.

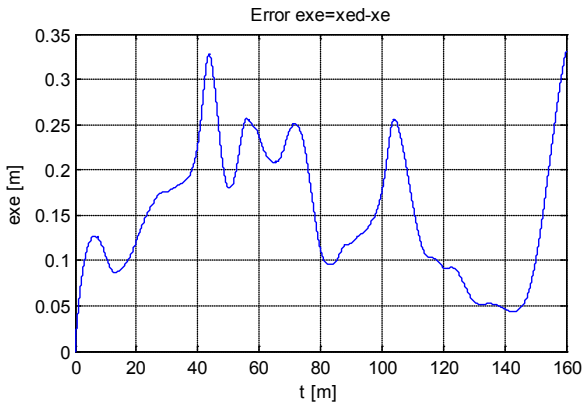


Figure 5.12. Tracking error along x_e axis

6. Conclusions

In this paper - the mobile robot guidance and control was researched - in the environment full of obstacles, by using the potential field's method. The mobile robot has 4-wheels configuration, electric drive on the rear vehicles, and is directed from the front wheels (Ackerman control algorithm). We have simulated a movement in a horizontal (2D) plane and the robot is modelled as a 3-DOF system (three degrees of freedom).

Our method uses functions defining potential fields at its position to calculate component vector. Only a part of the field that affects the robot is computed for each behaviour of the potential field; the sum of the vectors at the robot's position gave the resultant output vector. We did not have issues with combining potential fields; the influence of the low update rates are the broken (irregular) paths.

The contribution of this work (in comparison to other authors) is that we have simulated real 4-wheel robot with complete kinematics and dynamics, unlike existing works, which consider single mass point. The robot was treated as a mass object, it could not to change velocity and direction instantaneously (cannot happen). We found a solution for local minimum problem; if the global minimum is not guaranteed, we have chosen the functions in such a way that global minimum can be guaranteed and the robot will escape the local minima. Previous behaviours that should be avoided are included – thus we track where robot has been and we lead the robot to other places. These “avoid” behaviours will define desired direction and will insert tangential fields around the obstacles.

The tracking errors along both axes remain low (close to zero) and the time to stabilize is in terms of seconds (comparable to previous works, [11]).

REFERENCES

- [1] Shimoda S., Kuroda Y., Iagnemma K., (2005), Potential field navigation of high speed unmanned ground vehicles on uneven terrain, *IEEE Int. Conf. on Robotics and Automation*, Barcelona, Spain, pp.2828-2833.
- [2] Khatib O. (1990), Real-time obstacle avoidance for manipulators and mobile robots, In *Autonomous Robot Vehicles*, (Edited by I.J. Cox and G.T. Wilfong), pp. 396-404, Springer-Verlag.
- [3] Rimon E., Koditschek D. (1992). Exact Robot Navigation using Artificial Potential Functions. *IEEE Trans. on Robotics and Automation*, pp. 501-517, October (1992).
- [4] Borenstein J. and Koren Y. (1991), The vector field histogram-fast obstacle avoidance for mobile robots, *IEEE Transactions on Robotics and Automation* 7, pp. 278-288.
- [5] Wu K.H., Chen C.H., and Ko J.M, (1999). Path planning and prototype design of an AGV. *Mathematical and Computer Modelling*, vol.30, pp.147-167.
- [6] Rajamani R. (2011). Vehicle Dynamics and Control, Second Edition, *Springer - New York*, Dordrecht, Heidelberg, London.
- [7] Siegwart R., I. R. Nourbakhsh, D. Scaramuzza, (2011). Introduction to Autonomous Mobile Robots, Second Edition, *A Bradford Book The MIT Press Cambridge*, Massachusetts London, England.
- [8] Tzafestas G. S. (2014). Introduction to Mobile Robot Control, *Elsevier*, 32 Jamestown Road, London NW1 7BY.
- [9] Cook G., (2011). Mobile Robots: Navigation, Control and Remote Sensing, *A John Wiley & Sons, Inc.*, Publication.
- [10] Robotin R., Lazea G., Herle S. (2004). Hybrid Goal Acquisition System for Pioneer 2 Mobile robot. *Journal of Control Engineering and Applied Informatics (CEAI)*, vol. 6, no. 2, pp. 51-54.
- [11] Kozłowski K., D. Pazderski, I.Rudas, J.Tar, (2004). Modeling and control of a 4-wheel skid-steering mobile robot: From theory to practice, *Polish-Hungarian Bilateral Technology Co-operation Project and statutory grant* No. DS 93/121/04 of Poznan University of Technology.
- [12] Moret, E. N. (2003). Dynamic Modeling and Control of a Car-Like Robot, *Master Thesis, Virginia Polytechnic Institute and State University*.
- [13] Robotin R., Lazea G., Dobra P. (2010). Mobile Robots Path Planning with Heuristic Search. *Journal of Control Engineering and Applied Informatics (CEAI)*, vol. 12, no. 4, pp. 18-23.