Mobile Robot Guidance in Presence of Obstacles, using the Potential Fields Method

Vesna Antoska-Knights[1], Zoran Gacovski[2], Stojce Deskovski[3]

Abstract – **In this paper we have researched and simulated the mobile robot guidance and control in the environment full of obstacles, by using the potential fields method. We have considers a known environment where fixed potentials were assigned to the goal and the obstacles. We have applied a potential field's method with one attraction potential assigned to the goal point and fixed repulsion 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.**

Keywords – **Mobile Robot Guidance, Obstacle avoidance, Potential Field Method.**

I. 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 result in optimal (possibly shortest) path, avoiding the obstacles present in the working 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 moves but to avoid collisions while reaching the objective. Potential fields are elegant solutions to the path finding problem [5]. From different authors, different approaches are taken to calculate appropriate field configurations.

The robot uses functions defining potential fields at its position to calculate component vector [6]. The entire field doesn't have to be computed - only the portion of field affecting robot will be computed for each behavior 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 reaches local minima, it will just sit still. We can have issues with combining potential fields; the impact of update rates is that lower update rates can lead to "jagged" paths. If the robot is treated as a mass object, it cannot be expected to change velocity and direction instantaneously (cannot happen). We should find a solution for local minimum problem; if the global minimum is not guaranteed, we'll need to apply something else than gradient descent. 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" behaviors can be included – remembering where robot has been and bringing the robot to other places. Numerical techniques, Random walk methods and Navigation functions [7], i.e. "navigation templates" have been combined to get the resulting vector that will avoid local minima. They give "avoid" behaviour a preferred direction and insert tangential fields around obstacles.

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.

II. Mathematical Model of the Robot

In this section we will elaborate the robot movement in horizontal plane. We have considered a mobile robot with 4 wheels (Ackerman drive), 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 [1,2,10,11], and here we'll use these equations without derivation.

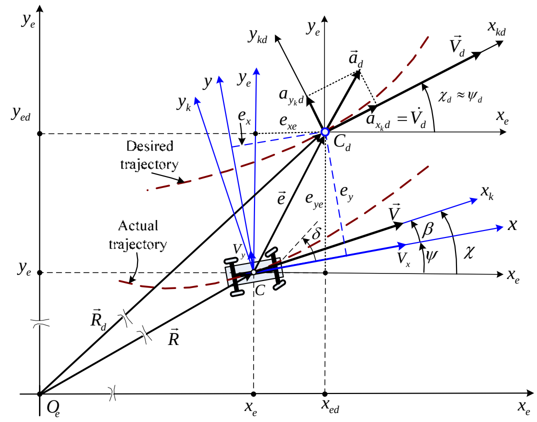


Fig.2.1 Kinematics of lateral vehicle motion: desired and actual trajectories of the mobile robot.

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 [1].

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 center (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

[1]Vesna Antoska-Knights is with Faculty of Technol. and Technic. Sciences – Veles, MK, E-mail: vesna_antoska@gmail.com.
[2]Zoran Gacovski is with the European University-Skopje, MK, E-mail: zgacovski@yahoo.com.
[3]Stojce Deskovski is with Technical Faculty-Bitola, MK, E-mail: stojce.deskovski@gmail.com.

robot trajectory $K(C; x_k, y_k)$, where x_k is axis - tangent, and y_k is axis normal to the trajectory.

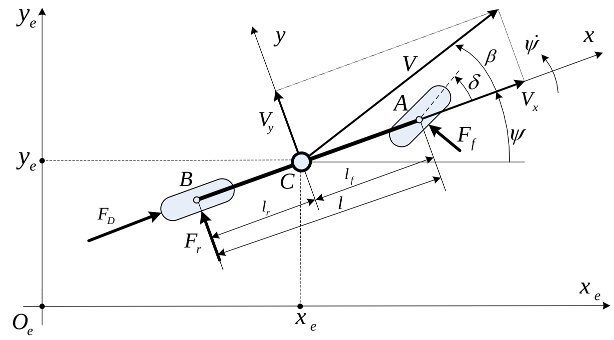


Fig.2.2 Bicycle model of lateral mobile robot dynamics

The kinematic equations of the mobile robots are [1]:

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

where: V is velocity of center 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 $l = l_f + l_r$. 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 c.g. of the vehicle, respectively.

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

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

where m and J are mass and inertial moment of the vehicle around the mass center, 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.

Complete model of mobile robot includes kinematic (2.1) and dynamic (2.2) equations, equations of non-holonomic constraints, model of the steering wheels actuator, and model of DC motor for driving force F_D generation.

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:

$\mathbf{x} = [x_e,\ y_e,\ \psi,\ V_x,\ F_D,\ \delta]^T$ and other variables that are dependent on the robot state.

III. Guidance and Control System

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

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. – center of gravity of mobile robot) whose coordinates are x_e, y_e. The error in following 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:

$$\mathbf{e}_B = \mathbf{T}_{BE}\mathbf{e}_B, \text{ or } [e_x, e_y]^T = \mathbf{T}_{BE}[e_{x_e}, e_{y_e}]^T,$$

$$\mathbf{T}_{BE} = \mathbf{T}_\psi = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \qquad (3.1)$$

where $\mathbf{T}_{BE}$ is transformation matrix from global E to local B coordinate system.

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:

$$\dot{V}_d = a_{x_k d},\ V_d(0) = V_{d0},\ \dot{\psi}_d = \frac{a_{y_k d}}{V_d},\ \psi_d(0) = \psi_{d0} \qquad (3.2)$$

$$\dot{x}_{ed} = V_d\cos\psi,\ x_{ed}(0) = x_{ed0},\ \dot{y}_{ed} = V_d\sin\psi,\ y_{ed}(0) = y_{ed0} \qquad (3.3)$$

The diagram of the control system is given in Fig.3.1. The „Mobile robot" block computes the equations, the block „Steering wheel" generates steering angle of front wheels, the block „Driving force" generates the driving force F_D, the block „Transformation to robot coordinates"computes the errors:

$$e_{xe} = x_{ed} - x_e,\ e_{ye} = y_{ed} - y_e,\ e_\psi = \psi_d - \psi\ , \qquad (3.4)$$

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 maneuvers, together with the feedback controller - we must apply a feedforward controller based on the inverse robot model. The controller (its algorithm) generates control signal u_1 to control the steering vehicles. To generate the signal u_1 we have used a PID controller. 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.3.1, we have developed simulation model in Matlab/Simulink 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.

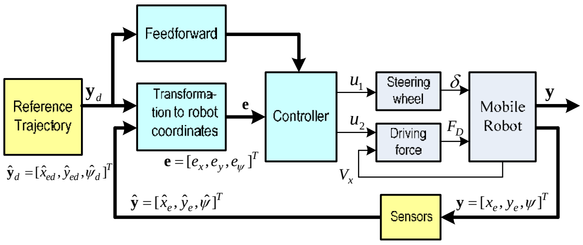


Fig.3.1 Blok diagram of mobile robot guidance and control system

IV. Trajectory Generation by Potential Fields Method

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

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

$$V_{magnitude} = \begin{cases} \frac{D-d}{D} & for\ d \le D \\ 0 & for\ d > D \end{cases}$$, where D is the max range of the field's effect. (4.1)

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:

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

Environment is sum of the potential of fields:

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

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

Gradient of the potential function must be differentiable:

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

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

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

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

Converting to robot control, we set the robot velocity (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.

Functions of attractive potential field will be: linear function of distance, quadratic function of distance and combination of the both. Linear function of distance is:

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

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

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

- Quadratic function of distance is given by:

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

Attracting force converges linearly towards 0 (goal):

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

Repulsive Potential Field should generate a barrier around all the obstacles: strong, if close to the obstacle, or no influence - if far from the obstacle:

$$U_{rep}(q) = \begin{cases} \frac{1}{2}\xi_{rep}\left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right) & if\ \rho(q) \le \rho_0 \\ 0 & f\ \rho(q) \ge \rho_0 \end{cases} \qquad (4.9)$$

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

Field is positive or zero and tends to infinity, as q gets closer to the obstacle:

$$F_{rep}(q) = -\nabla U_{rep}(q) = \begin{cases} \xi_{rep}\left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right)\frac{1}{\rho^2(q)}\frac{q - q_{obstacle}}{\rho(q)} & if\ \rho(q) \le \rho_0 \\ 0 & f\ \rho(q) \ge \rho_0 \end{cases}$$
(4.10)

For potential field path planning we have used harmonic potentials, since the robot is moving in an environment with fixed obstacles, ensuring that there are no local minima.

V. Simulation Results

In accordance with block diagram in Fig.3.1 and equations that describe each module we developed the simulation model of the mobile robot in Matlab/Simulink.

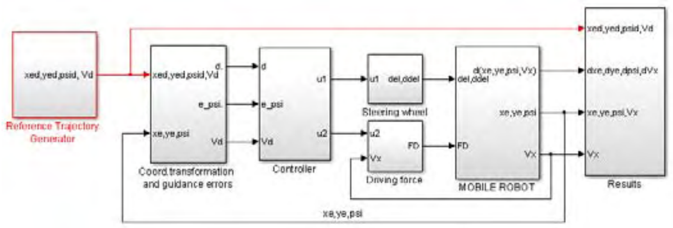


Fig. 5.1. Mobile robot control system model - developed in Simulink

Referent trajectory in an obstacle environment is generated in the block Reference Trajectory Generator using potential field method. Fig. 5.2 shows how the mobile robot follows this referent trajectory. Obstacles in Fig.5.2 are presented by circles.

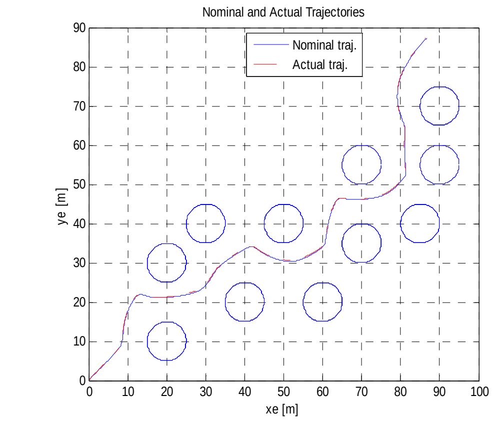
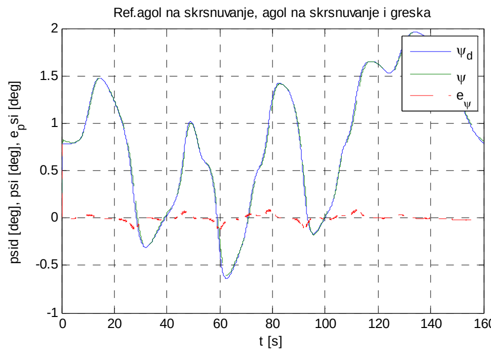


Fig.5.2. Referent trajectory following



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

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

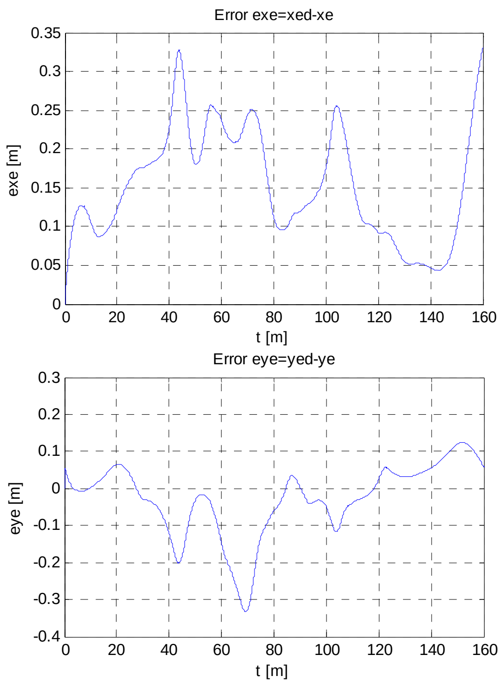


Fig.5.4. a) Tracking error along x axis, b) Tracking error along y ax.

VI. Conclusion

In this paper we have researched and simulated the mobile robot guidance and control 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 modeled as a 3-DOF system (three degrees of freedom).

Our method uses functions defining potential fields at its position to calculate component vector. Only the portion of field affecting robot was computed for each behavior 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 impact of update rates is that lower update rates can lead to "jagged" paths. 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. "Avoid-past" behaviors were included – remembering where robot has been and bringing the robot to other places. They gave "avoid" behaviour a preferred direction and inserted tangential fields around obstacles.

References

[1] Rajesh Rajamani, Vehicle Dynamics and Control, Second Edition, *Springer - New York*, Dordrecht, Heidelberg, London, 2011.

[2] Spyros G. Tzafestas, Introduction to Mobile Robot Control, *Elsevier*, 32 Jamestown Road, London NW1 7BY, 2014.

[3] Eric N Moret, Dynamic Modeling and Control of a Car-Like Robot, *Master Thesis, Virginia Polytechnic Institute and State University*, 2003.

[4] K. Kozłowski, D. Pazderski, I.Rudas, J.Tar, Modeling and control of a 4-wheel skid-steering mobile robot: From theory to practice, Polish-*Hungarian Bilateral Technology Co-operation Project and statuary grant* No. DS 93/121/04 of Poznan University of Technology.

[5] 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.

[6] 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.

[7] Rimon Elon, Koditschek Daniel (1992). Exact Robot Navigation using Artificial Potential Functions. *IEEE Trans. on Robotics and Automation*, pp. 501-517, October (1992).

[8] Kun-Hsiang Wu, Chin-Hsing Chen, and Juing-Ming Ko, Path planning and prototype design of an AGV. *Mathematical and Computer Modelling*, 30**,** 147-167 (1999).

[9] Borenstein J. and Koren Y. (1991), The vector field histogram-fast obstacle avoidance for mobile robots, *IEEE Transactions on Robotics and Automation 7*, 278-288.