Motion planning for mobile robots using uncertain estimations about the environment

Zoltán Gyenes
Department of Control Engineering and Information Technology
Budapest University of Technology and Economics
Budapest, Hungary
gyezo12@gmail.com

Emese Gincsainé Szádeczky-Kardoss
Department of Control Engineering and Information Tech.
Budapest University of Technology and Economics
Budapest, Hungary
szadeczky@iit.bme.hu

***Abstract*—The motion planning for mobile robots is a challenging task even if the agent has to reach the target position in a dense, dynamic environment. In this paper, our goal is to develop a motion planning algorithm using the changing uncertainties of the sensor-based data of the obstacles. The collision-free motion must be ensured by the algorithm using a cost function optimization method. As an assumption, some of the data of the obstacles (e.g. positions of the static obstacles) are already known at the beginning of the planning, and the other information (e.g. velocity vectors of moving obstacles) must be measured using the sensors. The algorithm is tested in simulations.**

***Index Terms*—motion planning, mobile robots, cost function, uncertain estimations**

I. INTRODUCTION

The autonomous driving is a common research area not only for cars but also for mobile robots. The agent has to execute its motion to the target position without a collision with any of the obstacles that occur in the workspace of the robot.

The motion planning algorithms generate both the path and velocity profiles for the agent using the measured information about the positions and velocity vectors of the obstacles.

The motion planning algorithms can be separated into two parts. If all data are available from the obstacles at the beginning, then global motion planning algorithms can be used for generating the appropriate path [1], [2]. On the other hand, if the robot can use only the local sensor-based information about the surrounding dense, dynamic environment, then reactive motion planning algorithms can be used [3], [4].

Using the reactive motion planning algorithm, it is challenging to generate evasive maneuvers that can ensure a safe motion for the agent and its environment. The task is more difficult, if the uncertainties of the measured data (position and velocity vectors) are taken into consideration. In this paper, a novel reactive motion planning method is presented that can calculate the uncertainties for every obstacle using their velocities and distances from the agent.

The paper has the following structure: Section II presents some reactive motion planning algorithms focusing on the papers in that the uncertainties of the measured data were used. In Section III, the basics of the *Velocity Obstacles* method will be presented. Later on, in Section IV, the novel concept of the calculation of the uncertainties of the obstacles will be shown. Then the usage of the cost function will be presented which helps the algorithm to select the appropriate velocity vector at every sampling time. In Section V, the simulated results will be presented. In Section VI, the presented algorithm will be summarized with the opportunities of further development.

II. PRIOR WORKS

In this section, few reactive motion planning algorithms are presented.

The main concept of the *Artificial Potential Field (APF)* method is that the final selected velocity vector of the robot can be calculated with the summation of the attractive and repelling forces. All of the obstacles generate repelling forces and the target position has an attractive force to the robot [5], [6]. One weakness of the algorithm is that sometimes only the local optimum can be found.

The *Inevitable Collision States method (ICS)* generates every states of the agent where there is no control command that would cause a collision-free motion. The main concept is to guarantee that the robot never finds itself in an *ICS* situation. The algorithm is appropriate both in static and dynamic environments [7], [8].

The ϵCCA is an extended version of the *Reciprocal Velocity Obstacles* algorithm [9], using kinodynamic constraints of the agent. The algorithm provides a suitable solution for the multirobot collision avoidance problem in a dense environment. The computational time plays an essential role in this method. The whole environment of the robot is divided into a grid-based map. The agent selects a collision-free velocity vector using both convex and nonconvex optimization methods [10].

The *Probabilistic Velocity Obstacles (PRVO)* algorithm is also an extended version of the *Reciprocal Velocity Obstacles (RVO)* method [9], using probabilistic. The time scaling algorithm and Bayesian decomposition is used. This method provided better performance in traversal times than the existing bound based methods. The algorithm was tested using simulation results [11].

The *Collision Avoidance under Bounded Localization Uncertainty (COCALU)* method [12] introduced convex hull peeling, generating a limitation of the localization error. This method results a better performance than the previously introduced *Multi-robot collision avoidance with localization uncertainty (CALU)* algorithm [13] in consideration of the

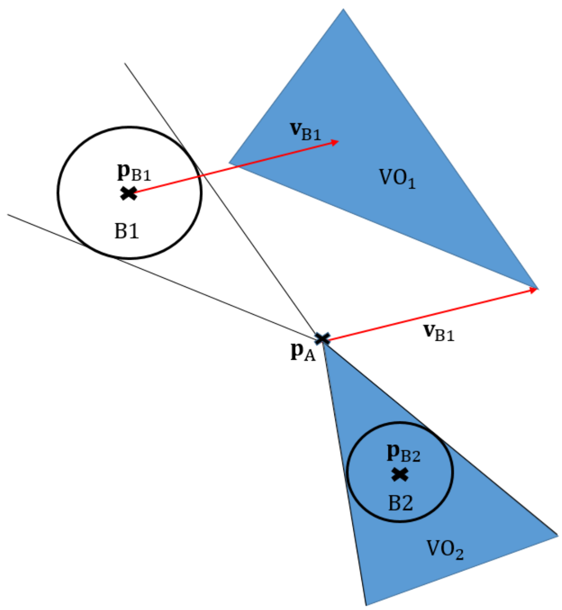

Fig. 1. *Velocity Obstacles* method

tightness of the bound. Particle filter is used for the robot localization problems. In that case, convex polygons are defined as the robot footprints. The algorithm provides that the robot is inside of this convex polygon with a probability of $1-\varepsilon$. A time truncation is also used in this method, hence it supports the velocity selection even in a crowded environment.

III. Velocity Obstacles method

The main concept of our method is based on the *Velocity Obstacles method (VO)* [14].

Using the positions and the velocities of the obstacles and the agent, the *VO* method generates a collision-free motion for the robot.

B_i defines the obstacles ($i = 1...m$ where m represents the quantity of obstacles) and the agent is A.

For every obstacle a VO_i cone can be specified as a cone that consists every velocity vector of the agent that would cause a collision between the robot (A) and the obstacle (B_i) in a future time:

$$VO_i = \{\, \mathbf{v}_{\mathrm{A}} \mid \exists\, t : \mathbf{p}_{\mathrm{A}} + \mathbf{v}_A t \,\cap\, \mathbf{p}_{\mathrm{Bi}} + \mathbf{v}_{\mathrm{Bi}} t \neq 0\} \quad (1)$$

where $\mathbf{v}_{\mathrm{A}}$ and $\mathbf{v}_{\mathrm{Bi}}$ are the velocity vectors and $\mathbf{p}_{\mathrm{A}}$ and $\mathbf{p}_{\mathrm{Bi}}$ are the positions of the agent and the obstacle. As an assumption, the velocities of the agent and the obstacles are unchanged until t.

The whole *VO* can be determined if there are more obstacles as:

$$VO = \cup_{i=1}^{m} VO_i \quad (2)$$

Figure 1 represents an example where a moving obstacle is in position $\mathbf{p}_{\mathrm{B1}}$ and it has velocity $\mathbf{v}_{\mathrm{B1}}$ at the actual time. There is a static obstacle in the workspace too (it is in the position $\mathbf{p}_{\mathrm{B2}}$). The two *VO* areas are depicted with blue color.

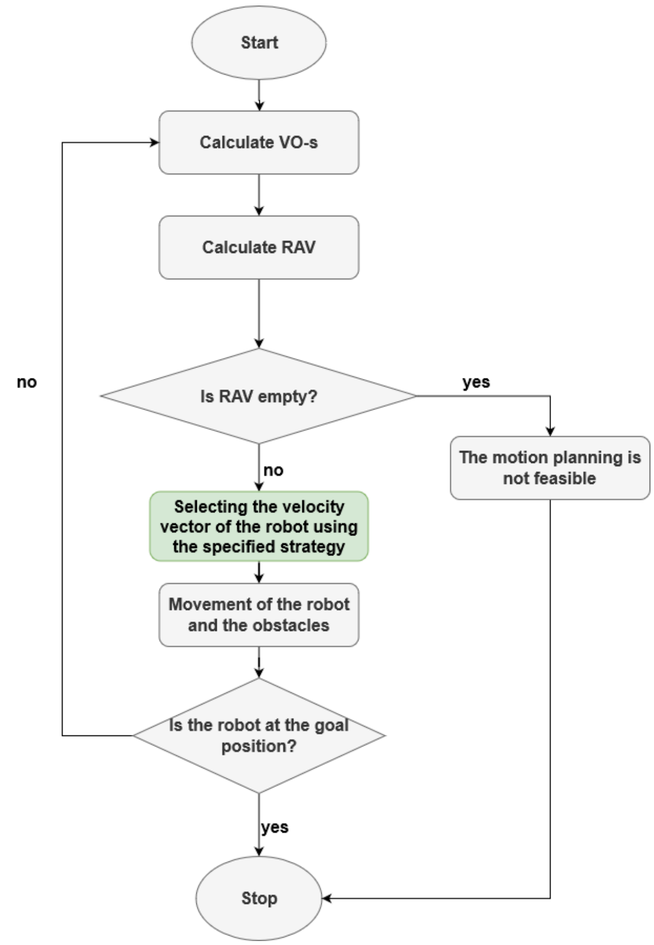

Fig. 2. *Steps of the whole algorithm*

The *Reachable Velocities (RV)* can be determined that consist every $\mathbf{v}_{\mathrm{A}}$ velocity vector of the robot that is reachable considering the actual previously selected velocity vector. After the subtraction of the *VO* from the *RV* the *Reachable Avoidance Velocities (RAV)* can be received.

Every step of the motion planning algorithm can be seen in Figure 2. The main difference between the algorithms is that how the velocity vector of the robot will be selected from the *RAV*.

IV. Cost function based velocity selection under changing uncertainties

In this section a cost function based velocity selection method will be presented that can consider the changing uncertainties of the obstacles.

A. Precheck algorithm

Only those VO_i of the obstacles must be considered during the calculation process of the RAV set that fulfill the precheck algorithm. Two cases of the obstacles will not be considered:

- those obstacles that are far from the agent
- those obstacles that will cross the paths of the robot in the far future.

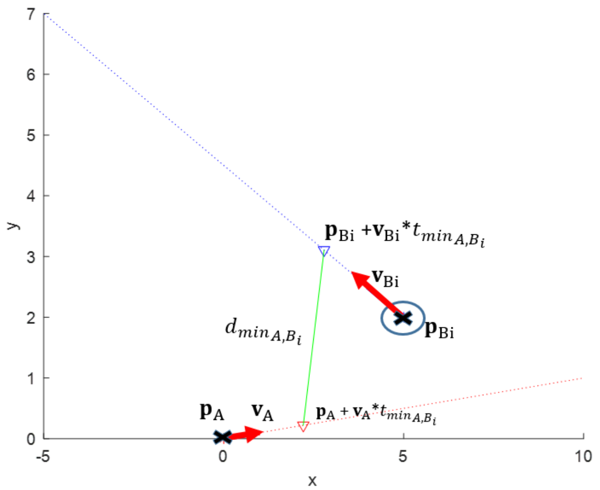

Fig. 3. *Precheck algorithm*

For all obstacles, the minimum time and distance must be calculated when the agent and the obstacle are closest to each other during their motion.

$$t_{min_{A,Bi}} = \frac{-(\mathbf{p}_{\mathrm{A}} - \mathbf{p}_{\mathrm{Bi}})(\mathbf{v}_{\mathrm{A}} - \mathbf{v}_{\mathrm{Bi}})}{||\mathbf{v}_{\mathrm{A}} - \mathbf{v}_{\mathrm{Bi}}||}, \quad (3)$$

where $t_{min_{A,Bi}}$ presents the time interval when the robot and the obstacle will be nearest to each other. If the value of this parameter is a negative number then it was in the past. $||.||$ represents the secondary norm.

The minimal distance can be calculated:

$$d_{min_{A,Bi}} = ||(\mathbf{p}_{\mathrm{A}} + \mathbf{v}_{\mathrm{A}} t_{min_{A,Bi}}) - (\mathbf{p}_{\mathrm{Bi}} + \mathbf{v}_{\mathrm{Bi}} t_{min_{A,Bi}})||, \quad (4)$$

So only those obstacles must be considered that fulfill the next equation:

$$\begin{gathered} 0 < t_{min_{A,Bi}} < 2 * T_{precheck} \quad \text{AND} \\ d_{min_{A,Bi}} < \mathrm{v}_{\max} * T_{precheck} \end{gathered} \quad (5)$$

where $\mathrm{v}_{\max}$ means the maximum velocity that the robot can reach and $T_{precheck}$ is a parameter of the algorithm that must be tuned. Our experiments showed that if the value of the $T_{precheck}$ parameter is too small, it generates not a smooth path for the agent.

The precheck algorithm is illustrated in Figure 3.

B. Calculation of changing uncertainties

In our prior work, all of the uncertainties of the obstacles were constant during the algorithm [15]. Now they will be changed considering the actual distances of the obstacles from the robot, the magnitudes and the changes of the velocity vectors of the obstacles.

The uncertainties can be calculated from the probabilities of the previously introduced parameters. The main idea is that the measured information has a higher reliability if the obstacles are closer to the agent. First, let calculate the obstacle distance part:

$$P_{dist_i} = \begin{cases} 1 - \frac{dist_{OR_i}}{\mathrm{v}_{\max} * T_u} & \text{if} \quad dist_{OR_i} < \mathrm{v}_{\max} * T_u \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where P_{dist_i} means distance based probability term, $dist_{OR_i}$ is actual distance between the obstacle and the agent, and T_u is the uncertainty time parameter.

The magnitude of the velocity of the obstacle plays also a role in the calculation of the uncertainties. The smaller the velocity of the obstacle is the higher reliability of the information of the obstacle is available :

$$P_{MV_i} = \begin{cases} 1 - \frac{||\mathbf{v}_{\mathrm{Bi}}||}{\mathrm{v}_{\max}} & \text{if} \quad ||\mathbf{v}_{\mathrm{Bi}}|| < \mathrm{v}_{\max} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where P_{MV_i} is the velocity based probability term, where $||\mathbf{v}_{\mathrm{Bi}}||$ means the actual magnitude of the velocity of the obstacle.

The change of the velocity of the obstacle also influences the uncertainties. The change of the velocity of the obstacle can be calculated:

$$CV_i = ||\mathbf{v}_{\mathrm{Bi,new}} - \mathbf{v}_{\mathrm{Bi,old}}|| \quad (8)$$

where CV_i means the change of the velocity of the obstacle, $\mathbf{v}_{\mathrm{Bi,new}}$ the actual velocity of the obstacle, and $\mathbf{v}_{\mathrm{Bi,old}}$ is previous velocity of the obstacle.

$$P_{CV_i} = \begin{cases} 1 - \frac{CV_i}{\mathbf{v}_{\max}} & \text{if} \quad CV_i < \mathrm{v}_{\max} \\ 0 & \text{otherwise} \end{cases}, \quad (9)$$

where P_{CV_i} is the probability term depending on the change of the velocity of the obstacle .

The probability can be calculated:

$$P_i = \frac{P_{dist_i} + P_{MV_i} + P_{CV_i}}{3} \quad (10)$$

From the calculated probability the parameter of the uncertainty can be calculated:

$$\alpha_i = 1 - P_i, \quad (11)$$

where this α_i uncertainty parameter must be calculated for every obstacle ($i = 1...m$, if there are m obstacles in the workspace).

C. Cost function

In our previous work, a method was defined called *Safety Velocity Obstacles (SVO)* method [16]. In this method, a cost function was used where different parts influenced the motion planning algorithm (speed, safety). This method was extended with the changing uncertainty parameter and a heading parameter that provides information about the orientation of the agent to the goal position.

In every step the nearest distance is calculated between the investigated velocities and the VO cone:

$$D_S(\mathbf{v}_{\mathrm{A}}) = \min_{\mathbf{v}_{VO} \in VO} dist(\mathbf{v}_A, \mathbf{v}_{VO}), \quad (12)$$

where $D_S(\mathbf{v}_\mathrm{A})$ means the minimal distance and $\mathbf{v}_{VO}$ is the closest point on the VO cone.

$D_S(\mathbf{v}_\mathrm{A})$ can be maximalized as:

$$D_S(\mathbf{v}_\mathrm{A}) = \begin{cases} D_S(\mathbf{v}_\mathrm{A}) & \text{if} \quad D_S(\mathbf{v}_\mathrm{A}) < D_{max} \\ D_{max} & \text{otherwise} \end{cases} \tag{13}$$

where D_{max} is a given value of the algorithm.

A normalization must be generated into the [0, 1] interval using $D_S(\mathbf{v}_\mathrm{A})$. The $C_S(\mathbf{v}_\mathrm{A}, VO)$ can be defined that will be used in the cost function:

$$C_S(\mathbf{v}_\mathrm{A}, VO) = 1 - \frac{D_S(\mathbf{v}_\mathrm{A})}{D_{max}} \tag{14}$$

$C_G(\mathbf{v}_\mathrm{A})$ will be also a part of the cost function

$$C_G(\mathbf{v}_\mathrm{A}) = \frac{||\mathbf{p}_A + \mathbf{v}_A T_s - \mathbf{p}_{goal}||}{||\mathbf{p}_A(0) - \mathbf{p}_{goal}||} \tag{15}$$

where $\mathbf{p}_{goal}$ is the target position, $\mathbf{p}_A(0)$ is the first position of the agent from where it executed the motion. $C_G(\mathbf{v}_\mathrm{A})$ denoted how far the robot will be from the goal if it will use the selected velocity, after that, it has to be divided by the distance of the first position and the desired position. T_s is the sampling time.

So in the introduced method, the prior method is extended by using changing $\alpha_i(t)$ parameters (they are calculated in every sampling time) for the different obstacles in consideration of the reliability of the velocity and position information of the obstacles.

The orientation of the robot plays also a role in the cost function. The heading parameter of the cost function can be calculated:

$$C_h(\mathbf{v}_\mathrm{A}) = \frac{|angleRG - angleIV(\mathbf{v}_\mathrm{A})|}{\pi}, \tag{16}$$

where $angleRG$ means the angle of the vector from the robot position to the goal position and $angleIV(\mathbf{v}_\mathrm{A})$ has a meaning of the angle of the investigated velocity vector of the agent. The heading parameter can be calculated using the difference of these angles.

The whole cost function can be determined using the previously introduced parameters:

$$Cost(\mathbf{v}_\mathrm{A}) = \sum_{i=1}^{m} \alpha_i(t)\; C_S(\mathbf{v}_\mathrm{A}, VO_i) + \beta_d\; C_G(\mathbf{v}_\mathrm{A}) + \beta_h\; C_h(\mathbf{v}_\mathrm{A}) \tag{17}$$

where $\alpha_i(t)$ means the actual calculated uncertainty parameter of an obstacle β_d is the distance parameter, and β_h is the heading parameter.

The different parameters of the cost function have a huge impact on the velocity selection, as it will be presented in Section V.

V. Simulation results

In this section some results of the simulation of the motion planning algorithm will be presented considering the changing uncertainties.

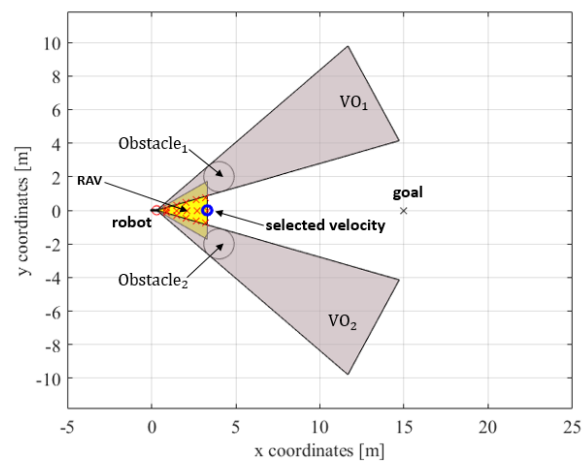

Fig. 4. *First example: Velocity selection*

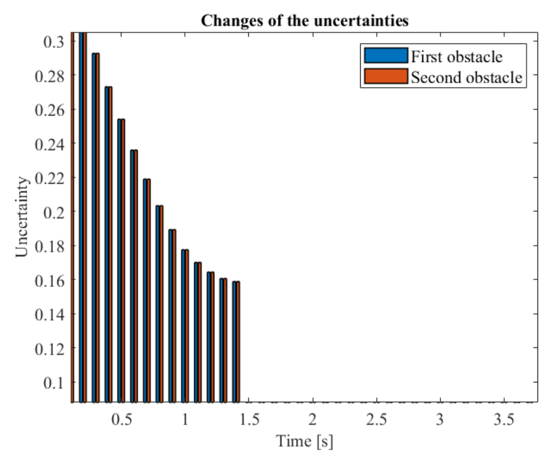

Fig. 5. *First example: two static obstacles; changing uncertainties during the motion*

A. Two static obstacles

In the first example, there are two static obstacles. In that case, using the introduced cost function, the algorithm will select a velocity vector for the robot that is exactly in the middle of the two obstacles because the two obstacles have the same uncertainties. This situation is presented in Figure 4, where the *Velocity Obstacles* are presented with the grey areas, the robot is the red circle, blue circle means the selected velocity vector and the goal is depicted by black x.

The magnitudes and the changes of the velocities of the obstacles do not influence the calculation process of the uncertainties because there are two static obstacles. So in that case only the distances between the robot and the obstacles has an impact on the calculation. Always the velocity vector between the two obstacles will be selected, so the distances between the agent and the two obstacles will be the same during the whole motion resulting the same uncertainties for both obstacles as it is presented in Figure 5.

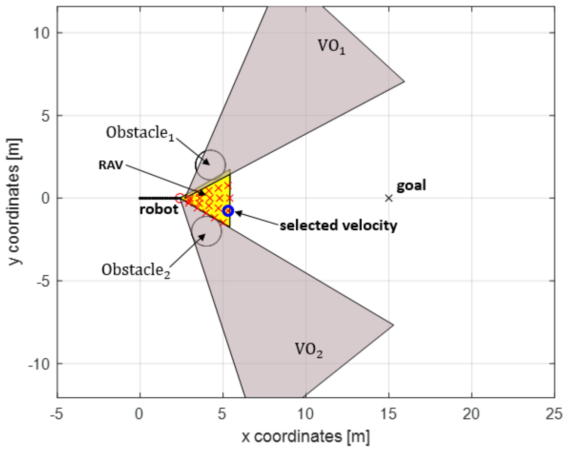

Fig. 6. *Second example: Velocity selection*

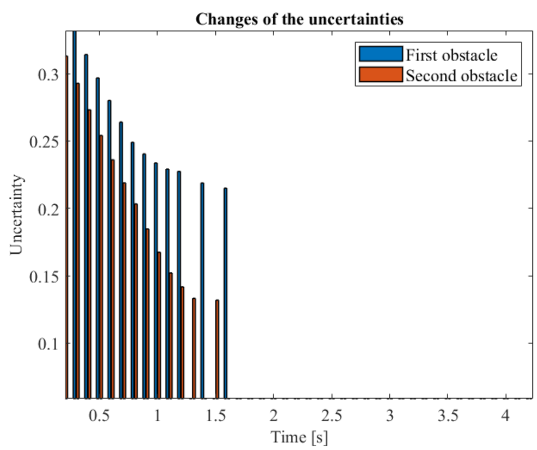

Fig. 7. *Second example: one static and one moving obstacles; changing uncertainties during the motion*

B. One static and one moving obstacles

In this example, the first obstacle is a moving obstacle and the second obstacle is a static obstacle. If the agent is far from the obstacles then it has the opportunity to select the velocity in the line to the goal position. After that if it reaches the obstacles it selects a velocity vector that results a maneuver near to the static obstacle because its uncertainty is smaller. The results of the velocity selection in this case is depicted in Figure 6.

In that example, the uncertainties of the obstacles are not the same as in the previous case, the static obstacle has a smaller uncertainty during the whole motion as it is presented in Figure 7. It can be detected that in the first step there is not a huge distance between the uncertainties of the obstacles. It is generated because the moving obstacle has a small magnitude of the velocity and the distances between the obstacles and the robot are the same at the first step.

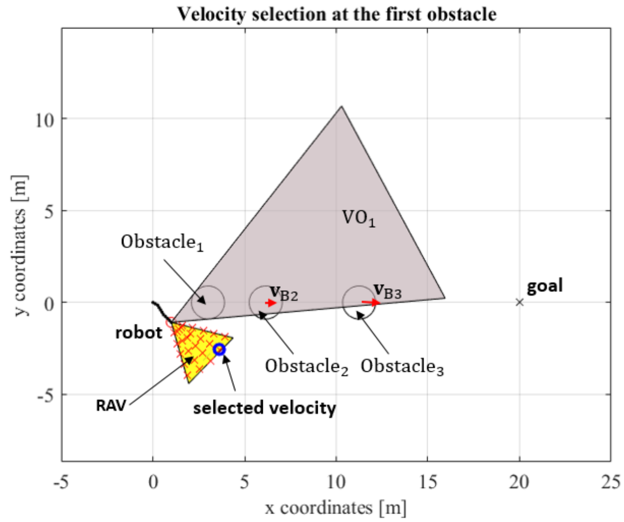

Fig. 8. *Third example: Velocity selection at the first obstacle*

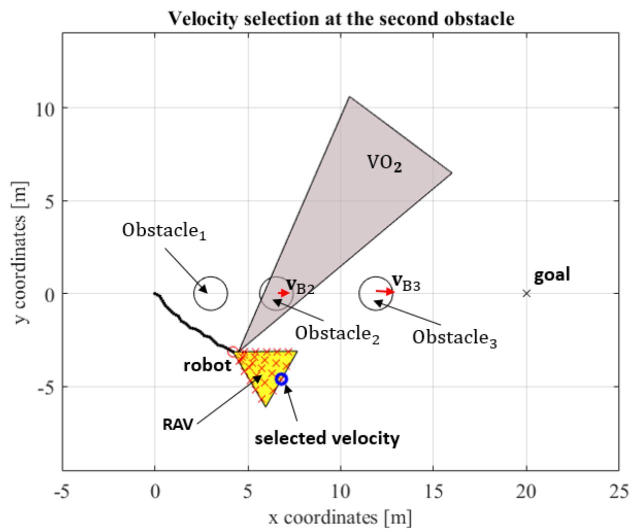

Fig. 9. *Third example: Velocity selection at the second obstacle*

C. Three obstacles in front of each other

In the following example, there are three obstacles in front of each other with different velocities (the first obstacle is a static obstacle, and the others have even higher velocity vectors). Figure 8 represents the velocity selection at the first obstacle and Figure 9 shows the velocity selection at the second obstacle. It can be detected that a further velocity vector will be selected at the second obstacle, hence it has a higher velocity vector.

The bigger the velocity of the obstacle is, the higher the uncertainty is for the obstacle as it is represented in Figure 10. After passing the obstacle the uncertainty will be smaller.

The β_h parameter plays a significant role in the cost function at the aspect of the target reaching strategy. If this parameter has a bigger value, it has a higher impact on the motion than

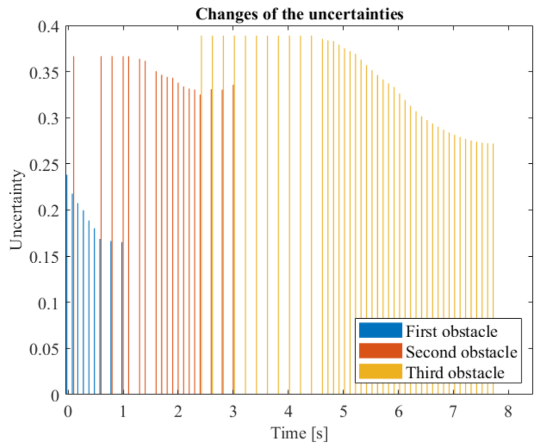

Fig. 10. *Third example: three obstacles in front of each other with different velocities; changing uncertainties during the motion*

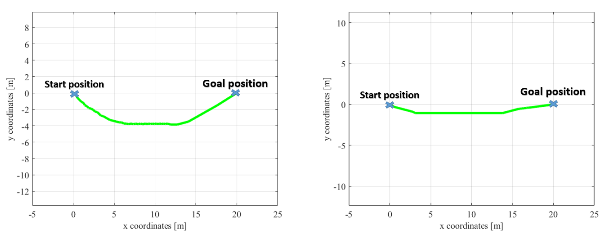

Fig. 11. *The resulted paths of the motion of the robot with different heading parameter, in the first example $\beta_h = 0.3$, in the second example $\beta_h = 0.6$*

the uncertainties of the obstacles, as presented in Figure 11. In that case ($\beta_h = 0.6$), the agent moves as close to the obstacles as the collision-free motion planning algorithm allows it.

So the values of the parameters depend on the usage of the algorithm. Different values of parameter generate different results in the collision-free motion planning algorithm. But it has to take into consideration that there will never be a solution that can fulfill every aspect of the motion planning problem. A sub-optimal solution can be generated.

VI. Conclusion

In this paper, a novel motion planning algorithm was introduced, using the basics of the *Velocity Obstacles* method. In this algorithm, the collision-free motion planning for the mobile robot can be generated after the calculation of the changing uncertainties of the obstacles. The uncertainties depend on the distances between the robot and the obstacles, the magnitudes of the velocity vectors of the obstacles, and the changes of the velocities of the obstacles. The selected velocity vector for the robot can be determined by using a cost function method.

In the future, the calculation of the uncertainties could be submitted by using Particle filter using the measured and estimated values of the positions and velocities of the obstacles. As a future plan, we would like to test this algorithm on a real robotic system.

Acknowledgment

SUPPORTED BY THE ÚNKP-20-3 NEW NATIONAL EXCELLENCE PROGRAM OF THE MINISTRY FOR INNOVATION AND TECHNOLOGY FROM THE SOURCE OF THE NATIONAL RESEARCH, DEVELOPMENT AND INNOVATION FUND and by the EFOP-3.6.2-16-2016-00014 project - financed by the Ministry of Human Capacities of Hungary.

References

[1] Panov, S., Koceski, S. (2015). Metaheuristic global path planning algorithm for mobile robots. International Journal of Reasoning-Based Intelligent Systems, 7(1/2), 35. https://doi.org/10.1504/ijris.2015.070910

[2] Hsu, P. M., Lin, C. L., Yang, M. Y. (2014). On the complete coverage path planning for mobile robots. Journal of Intelligent and Robotic Systems: Theory and Applications, 74(3–4), 945–963. https://doi.org/10.1007/s10846-013-9856-0

[3] Masehian, E., Katebi, Y. (2014). Sensor-based motion planning of wheeled mobile robots in unknown dynamic environments. Journal of Intelligent and Robotic Systems: Theory and Applications, 74(3–4), 893–914. https://doi.org/10.1007/s10846-013-9837-3

[4] Mohanan, M. G., Salgoankar, A. (2018). A survey of robotic motion planning in dynamic environments. Robotics and Autonomous Systems, 100, 171–185. https://doi.org/10.1016/j.robot.2017.10.011

[5] Kovács, B., Szayer, G., Tajti, F., Burdelis, M., Korondi, P. (2016). A novel potential field method for path planning of mobile robots by adapting animal motion attributes. Robotics and Autonomous Systems, 82, 24–34. https://doi.org/10.1016/j.robot.2016.04.007

[6] Malone, N., Chiang, H. T., Lesser, K., Oishi, M., Tapia, L. (2017). Hybrid Dynamic Moving Obstacle Avoidance Using a Stochastic Reachable Set-Based Potential Field. IEEE Transactions on Robotics, 33(5), 1124–1138. https://doi.org/10.1109/TRO.2017.2705034s

[7] Fraichard, T., Asama, H. (2004). Inevitable collision states - A step towards safer robots. Advanced Robotics, 18(10), 1001–1024. https://doi.org/10.1163/1568553042674662

[8] Martinez-Gomez, L., Fraichard, T. (2009). Collision avoidance in dynamic environments: An ICS-based solution and its comparative evaluation. Proceedings - IEEE International Conference on Robotics and Automation, 100–105. https://doi.org/10.1109/ROBOT.2009.5152536

[9] Van Berg, J. Den, Lin, M., Manocha, D. (2008). Reciprocal velocity obstacles for real-time multi-agent navigation. Proceedings - IEEE International Conference on Robotics and Automation, 1928–1935. https://doi.org/10.1109/ROBOT.2008.4543489

[10] Alonso-Mora, J., Beardsley, P., Siegwart, R. (2018). Cooperative Collision Avoidance for Nonholonomic Robots. IEEE Transactions on Robotics, 34(2), 404–420. https://doi.org/10.1109/TRO.2018.2793890

[11] Gopalakrishnan, B., Singh, A. K., Kaushik, M., Krishna, K. M., Manocha, D. (2017). PRVO: Probabilistic Reciprocal Velocity Obstacle for multi robot navigation under uncertainty. IEEE International Conference on Intelligent Robots and Systems, 2017-Septe, 1089–1096. https://doi.org/10.1109/IROS.2017.8202279

[12] Claes, D., Hennes, D., Tuyls, K., Meeussen, W. (2012). Collision avoidance under bounded localization uncertainty. IEEE International Conference on Intelligent Robots and Systems, 1192–1198. https://doi.org/10.1109/IROS.2012.6386125

[13] Hennes, D., Claes, D., Meeussen, W., Tuyls, K. (2012). Multi-robot collision avoidance with localization uncertainty. 11th International Conference on Autonomous Agents and Multiagent Systems 2012, AAMAS 2012: Innovative Applications Track, 2, 672–679.

[14] Fiorini, P., Shiller, Z. (1998). Motion planning in dynamic environments using velocity obstacles. International Journal of Robotics Research, 17(7), 760–772. https://doi.org/10.1177/027836499801700706

[15] Zoltan Gyenes; Emese Gincsaine Szadeckzy-Kardoss (2020). Rule-based velocity selection for mobile robots under uncertainties. Proceeding- INES 2020 • 24th International Conference on Intelligent Engineering Systems • July 8-10, 2020 • Reykjavík, Iceland, 127–132.

[16] Zoltan Gyenes; Emese Gincsaine Szadeckzy-Kardoss (2018) Motion planning for mobile robots using the Safety Velocity Obstacles method. *In 19th International Carpathian Control Conference*, pp. 389–394, Szilvásvárad, Hungary, 2018.