ACTA IMEKO
ISSN: 2221-870X
September 2021, Volume 10, Number 3, 51 - 60

Uncertain estimation-based motion-planning algorithms for mobile robots

Zoltán Gyenes[1], Emese Gincsainé Szádeczky-Kardoss[1]

[1] *Budapest University of Technology and Economics, Magyar Tudósok Körútja 2, 1117 Budapest, Hungary*

ABSTRACT
Collision-free motion planning for mobile agents is a challenging task, especially when the robot has to move towards a target position in a dynamic environment. The main aim of this paper is to introduce motion-planning algorithms using the changing uncertainties of the sensor-based data of obstacles. Two main algorithms are presented in this work. The first is based on the well-known velocity obstacle motion-planning method. In this method, collision-free motion must be achieved by the algorithm using a cost-function-based optimisation method. The second algorithm is an extension of the often-used artificial potential field. For this study, it is assumed that some of the obstacle data (e.g. the positions of static obstacles) are already known at the beginning of the algorithm (e.g. from a map of the enviroment), but other information (e.g. the velocity vectors of moving obstacles) must be measured using sensors. The algorithms are tested in simulations and compared in different situations.

Section: RESEARCH PAPER

Keywords: Motion planning; mobile robots; cost function; uncertain estimations

Citation: Zoltán Gyenes, Emese Gincsainé Szádeczky-Kardoss, Uncertain estimation-based motion planning algorithms for mobile robots, Acta IMEKO, vol. 10, no. 3, article 9, September 2021, identifier: IMEKO-ACTA-10 (2021)-03-09

Section Editor: Bálint Kiss, Budapest University of Technology and Economics, Hungary

Received January 15, 2021; **In final form** August 9, 2021; **Published** September 2021

Corresponding author: Zoltán Gyenes, e-mail: gyezo12@gmail.com

1. INTRODUCTION

Autonomous driving is a highly frequented research area for mobile robots, cars and drones. Robots have to generate a collision-free motion towards the target position while maintaining safety with respect to the obstacles that occur in the local environment. Motion-planning methods generate both the velocity and the path profiles for the robot using measured information about the velocity vectors and the positions of the obstacles.

Motion-planning algorithms can be divided into two parts. If all the data for the robot's environment are known and available at the start, then global motion-planning algorithms can be used to generate a collision-free path [1], [2]. However, if the robot can only use local sensor-based information about its surrounding dnese and dynamic environment, then reactive motion-planning algorithms can provide an acceptable solution for generating the robot's path and velocity [3], [4].

Using a reactive motion-planning algorithm, generating optimal evasive manoeuvers that can ensure a safe motion for the agent and the environment is an NP-hard problem [5]. The task is more difficult if the uncertainties of the measured data (velocity vectors and positions) are taken into account. In this paper, a novel reactive motion-planning algorithm is presented that can calculate the uncertainties of every obstacle using their velocity vectors and distance from the agent.

The paper is ordered in the following way. Section 2 outlines some often-used reactive motion-planning methods that have been introduced in recent decades. In some algorithms, the uncertainties of the measured data have also been considered. At the end of Section 2, the basics of the velocity obstacle (VO) and artificial potential field (APF) methods are presented. In Section 3, the novel concept for the calculation of obstacle uncertainties is set out. Section 4 then presents the introduced motion-planning algorithms, which can generate a safety motion for the agent taking into account the uncertainties. In Section 5, the simulation results are presented, and the introduced motion-planning methods are compared. In Section 6, the CoppeliaSim simulation environment is discussed, and Section 7 provides a conclusion and sets out plans for future research.

2. PREVIOUS WORK

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

The inevitable collision states method (ICS) calculates all states of the robot where there is no available control command that would result in a collision-free motion between the robot and the environment. The main goal is to ensure that the agent never finds itself in an ICS situation. The algorithm is appropriate not only for static but also for dynamic environments [6]-[8].

The main concept behind the dynamic window method [9], [10] is that the agent selects a velocity vector from the reachable and admissible set of the velocity space. The robot executes a collision-free motion by selecting a velocity vector from the admissible velocity set. At the same time, reachable velocities can be generated using the kinematic and dynamic constraints of the agent.

The admissible gap is a relatively new concept for motion-planning algorithms [11]. If the robot can move through the gap safely using motion control, then it is admissible, obeying the constraints of the agent. This method is also usable in an unknown environment. The gap-based online motion-planning algorithm has also been used with a Lidar sensor by introducing a binary sensing vector (the value of the vector element is equal to 1 if there is an obstacle in that direction) [12].

2.1. Velocity obstacle method

The main concept behind our method is based on the VO method [13]. Using the positions and the velocities of the obstacles and the agent, the VO method generates a collision-free motion for the robot. The VO concept has been used in different methods.

The steps in the VO method are as follows: B_i denotes the different obstacles ($i = 1 \ldots m$, where m represents the number of obstacles), and the agent is A. For every obstacle, a VO_i cone can be generated that constitutes every robot velocity vector that would result in a collision between the agent (A) and the obstacle (B_i) at a future time:

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

where $\mathbf{p}_\mathrm{A}$ and $\mathbf{p}_{\mathrm{B}i}$ are the positions and $\mathbf{v}_\mathrm{A}$ and $\mathbf{v}_{\mathrm{B}i}$ are the velocity vectors of the robot and the obstacle. The velocities of the obstacles and the robot are assumed to be constant until t.

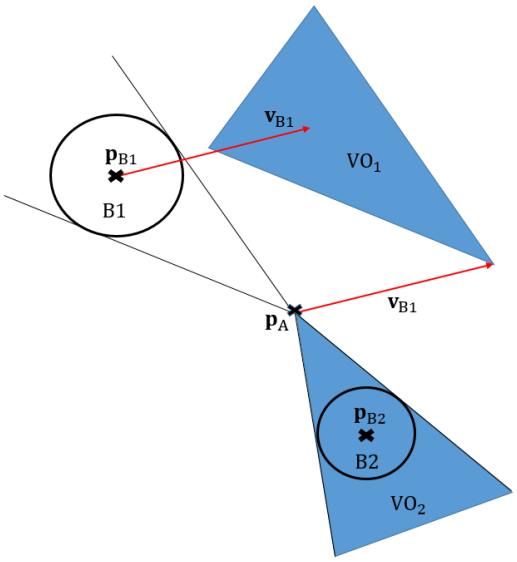

Figure 1. Velocity obstacle method.

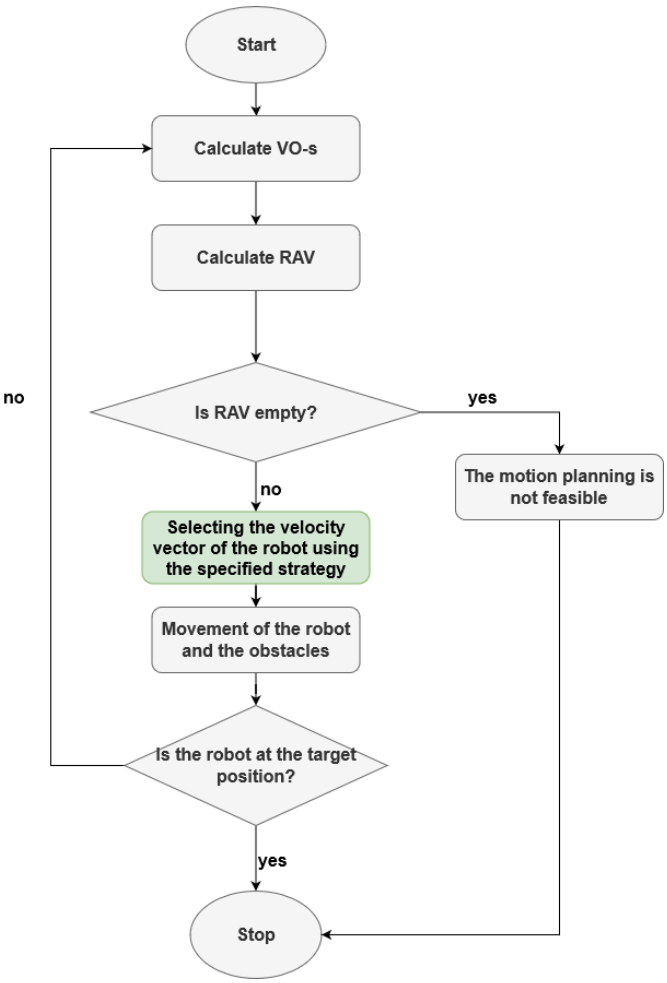

Figure 2. Steps of the whole VO algorithm.

If there are more obstacles, then the whole VO set can be generated:

$$VO = \cup_{i=1}^{n} \quad VO_i\,. \quad (2)$$

Figure 1 provides an example in which a moving obstacle is in position $\mathbf{p}_{\mathrm{B1}}$ and has velocity $\mathbf{v}_{\mathrm{B1}}$ at the actual time step. There is also a static obstacle in the workspace of the agent ($\mathbf{p}_{\mathrm{B2}}$ represents its position). The two VO areas are depicted in blue.

Reachable velocities (RV) can be defined as the velocity area that constitutes every $\mathbf{v}_\mathrm{A}$ velocity of the agent that is reachable considering the previously selected velocity vector and the motion capabilities of the robot. Reachable avoidance velocities (RAV) can be received by subtracting the VO from the RV set.

Figure 2 represents the steps of the motion-planning algorithm. The main difference between the algorithms is the method for selecting the robot's velocity vector from the RAV set.

The ϵCCA is an extended version of the reciprocal velocity obstacle (RVO) algorithm [14], which uses the kinodynamic constraints of the robot. The method generates an appropriate solution for the multi-robot collision avoidance problem in a complex environment. The computational time plays an important role in this algorithm. The whole environment of the agent is divided into a grid-based map. The agent selects a collision-free velocity vector using both convex and nonconvex optimisation algorithms [15].

The probabilistic velocity obstacle method is also an extended version of the RVO method [14], which uses the time-scaling method and Bayesian decomposition. This method demonstrates better performance in terms of traversal times than the existing bound-based methods. The algorithm was tested using simulation results [16].

The collision avoidance under bounded localisation uncertainty method [17] introduced convex hull peeling, resulting in a limitation in the localisation error. This method results in a better performance than the previously introduced multi-robot collision avoidance with localisation uncertainty method [18] with respect to the tightness of the bound. Particle filter is used for robot localisation problems. In this case, convex polygons are generated as the robot footprints. The algorithm ensures that the robot is inside this convex polygon with a probability of $1-\varepsilon$. A time truncation is also used in the algorithm because it supports the velocity selection even in a crowded, complex environment.

The directive circle (DC) method is an extended version of the VO method [19], [20]. In this algorithm, the velocity of the robot is selected from DC, which is calculated using the maximum velocity of the agent for the radius of the DC. Ensuring the kinematic constraints of the agent, the best solution is selected from the DC in the optimal direction for the target position. Using the DC method, the local minima situations are preserved.

All the presented reactive motion-planning algorithms assume a complete set of information on the position and the velocity vectors of the obstacles that occur in the robot's workspace. The main advantage of our introduced method is that the uncertainty of the measured sensor information can be taken into consideration, and the novel motion-planning algorithm can generate collision-free target reaching for the agent even when the data is inaccurate.

2.2. Artificial potential field method

The APF method is an often-used reactive motion-planning algorithm. The main concept is to calculate the summation of the attractive (between the robot and the target) and repelling (between the agent and the obstacles) forces [21], [22]. One weakness of this algorithm is that sometimes only the local optimum can be found. The algorithm has also been developed for unmanned arial vehicles [23], while human–robot interaction was also simulated using the APF motion-planning method by using the motion characteristics of household animals [24].

The steps of the APF method are as follows:

During motion planning, in every sampling time step, $\mathbf{A_r}$ force will influence the motion of the agent. The $\mathbf{A_r}$ force depends on the repelling ($\mathbf{F_{ar_i}}$) and the attractive ($\mathbf{F_{rc}}$) forces.

The closer the robot is to the obstacle, the larger the volume of the repelling force. The repelling force can be calculated by

$$\mathbf{F_{ar_i}} = \frac{\eta\sqrt{\frac{1}{D_{\mathrm{ra}_i}} + \frac{1}{D_{\mathrm{ra}_{\max}}}}}{D_{\mathrm{ra}_i}^2}\mathbf{AR_i}\,, \tag{3}$$

where D_{ra_i} denotes the distance between the robot and the obstacle, $\mathbf{AR_i}$ is the vector between the obstacle and the agent, η is a specific parameter that identifies the role of the repelling force in the motion-planning algorithm and $D_{\mathrm{ra}_{\max}}$ is the largest distance that should be considered in the motion-planning algorithm, which can be calculated as

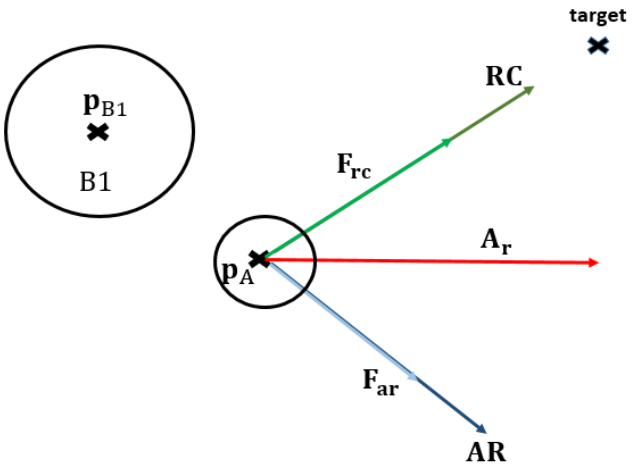

Figure 3. APF method.

$$D_{\mathrm{ra}_{\max}} = v_{\max}\,T_s\,, \tag{4}$$

where $v_{\max}$ is the maximum velocity of the robot and T_s denotes the sampling time.

The attractive force can be calculated as

$$\mathbf{F_{rc}} = \xi\,\mathbf{RC}\,, \tag{5}$$

where $\mathbf{RC}$ is the vector between the robot and the target and ξ is the parameter of the attractive force (depending on the usage of the algorithm).

The force that influences the motion of the robot can be calculated (if there is one obstacle in the workspace) with the sum of the repelling and the attractive forces:

$$\mathbf{A_r} = \sum_{i=1}^{m}\mathbf{F_{ar}}_i + \mathbf{F_{rc}}\,. \tag{6}$$

Figure 3 illustrates the different forces presented. There is one obstacle in the workspace (*B1*) with the position $\mathbf{p_{B1}}$. The agent is at the $\mathbf{p_A}$ position at this point, and the summation of the forces can be checked.

If the mass of the agent is known, then the acceleration can be calculated using Newton's second law:

$$\mathbf{a} = \frac{\mathbf{A_r}}{m}\,, \tag{7}$$

where m is the mass of the robot.

The changes in the velocity vector can be calculated if the force and the sampling time are known:

$$\Delta\mathbf{v} = \mathbf{a}\,T_s\,. \tag{8}$$

So the actual velocity can be calculated using the previous velocity and the change in velocity:

$$\mathbf{v_{new}} = \mathbf{v_{prev}} + \Delta\mathbf{v}\,. \tag{9}$$

3. UNCERTAINTY CALCULATION USING MEASUREMENT DATA

In previous studies, all the uncertainties of the obstacles were constant throughout the algorithm [25]. In the present study, they will be adjusted using the changes in the velocity vectors of the obstacles, the actual distances of the obstacles from the robot and the magnitudes of the obstacles' velocity vectors.

The uncertainties can be calculated from the probabilities of the previously introduced parameters. The main concept behind this method is that the measured information has lower reliability if the obstacles are far from the robot. First, the obstacle distance is generated:

$$P_{\mathrm{dist}_i} = \begin{cases} 1 - \dfrac{\mathrm{dist}_{OR_i}}{v_{\max} \cdot T_{\mathrm{u}}} & \text{if} \quad \mathrm{dist}_{OR_i} < \mathbf{v}_{\max} \cdot T_{\mathrm{u}} \\ 0 & \text{otherwise} \end{cases}, \tag{10}$$

where T_{u} is the uncertainty time parameter, P_{dist_i} is the distance-based probability term and dist_{OR_i} is the actual distance between the robot and the obstacle B_i.

The magnitude of the velocity vector of the obstacle also plays a significant role in generating the uncertainties; the higher the velocity of the obstacle, the smaller the reliability of the available information on the obstacle:

$$\mathrm{P}_{\mathrm{MV}_i} = \begin{cases} 1 - \dfrac{||\mathbf{v}_{\mathrm{Bi}}||}{v_{\max}} & if \quad ||\mathbf{v}_{\mathrm{Bi}}|| < v_{\max} \\ 0 & \text{otherwise} \end{cases}, \tag{11}$$

where $||\mathbf{v}_{\mathrm{Bi}}||$ refers to the actual magnitude of the velocity of the obstacle B_i ($||.||$ is the secondary norm (Euclidean distance)) and $\mathrm{P}_{\mathrm{MV}_i}$ is the velocity-based probability term.

The change in the obstacle's velocity vector also influences the volume of the uncertainties. The changes in the velocity of the obstacle can be calculated for each obstacle:

$$CV_i = ||\mathbf{v}_{Bi,\mathrm{new}} - \mathbf{v}_{Bi,\mathrm{old}}||, \tag{12}$$

where $\mathbf{v}_{Bi,\mathrm{new}}$ is the actual velocity of the obstacle, $\mathbf{v}_{Bi,\mathrm{old}}$ is the previous velocity of the obstacle and CV_i denotes the change in the obstacle's velocity:

$$P_{\mathrm{CV}_i} = \begin{cases} 1 - \dfrac{CV_i}{2\, v_{\max}} & \text{if} \quad CV_i < v_{\max} \\ 0 & \text{otherwise} \end{cases}, \tag{13}$$

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

The probability for obstacle B_i can be generated as

$$P_i = \frac{P_{\mathrm{dist}_i} + P_{\mathrm{MV}_i} + P_{\mathrm{CV}_i}}{3}. \tag{14}$$

The uncertainty parameter can be calculated from the calculated probability:

$$\alpha_i = 1 - P_i, \tag{15}$$

where this α_i uncertainty parameter must be calculated for every obstacle ($i = 1 \ldots m$, if there are m obstacles in the environment; if $\alpha_i = 0$, then there is no measurement uncertainty).

4. VELOCITY SELECTION BASED ON MOTION-PLANNING ALGORITHMS

4.1. Precheck algorithm

The agent has to consider only those obstacles that fulfil the precheck algorithm during the motion-planning algorithm. Two obstacle situations are not used:

• obstacles that will cross the path of the agent in the distant future,

• obstacles that are at a considerable distance from the agent.

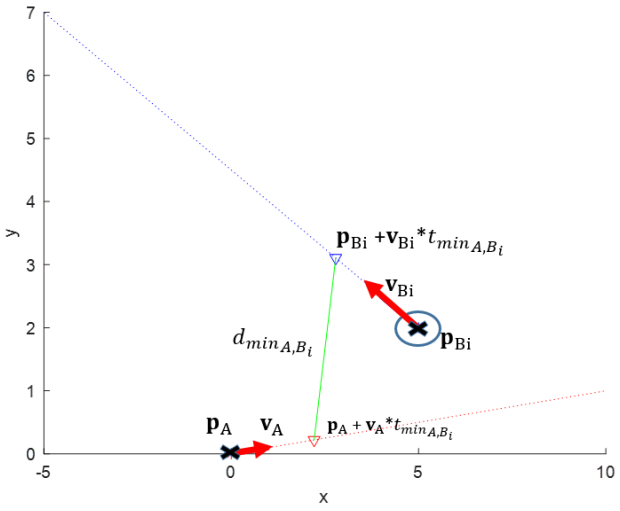

Figure 4. Precheck algorithm.

For all obstacles, the minimum distance and time must be calculated for the point at which the agent and the obstacle are closest to each other during their motion:

$$t_{\mathrm{min}_{\mathrm{A,B}i}} = \frac{-(\mathbf{p}_{\mathrm{A}} - \mathbf{p}_{\mathrm{B}i})(\mathbf{v}_{\mathrm{A}} - \mathbf{v}_{\mathrm{B}i})}{||\mathbf{v}_{\mathrm{A}} - \mathbf{v}_{\mathrm{B}i}||}, \tag{16}$$

where $t_{\mathrm{min}_{\mathrm{A,B}i}}$ presents the time interval for the point at which the agent and the obstacle are closest to each other. The nearest point is in the past if the value of this parameter is negative.

The minimal distance can be calculated as follows:

$$d_{\mathrm{min}_{\mathrm{A,B}i}} = \left|\left|\left(\mathbf{p}_{\mathrm{A}} + \mathbf{v}_{\mathrm{A}} t_{\mathrm{min}_{\mathrm{A,B}i}}\right) - \left(\mathbf{p}_{\mathrm{B}i} + \mathbf{v}_{\mathrm{B}i} t_{\mathrm{min}_{\mathrm{A,B}i}}\right)\right|\right|. \tag{17}$$

So, only those obstacles that fulfill the following inequalities must be considered:

$$0 < t_{\mathrm{min}_{\mathrm{A,B}i}} < 2 \cdot T_{\mathrm{precheck}} \text{ AND } d_{\mathrm{min}_{\mathrm{A,B}i}} < v_{\max} \cdot T_{\mathrm{precheck}}, \tag{18}$$

where $v_{\max}$ denotes the maximum velocity of the agent and T_{precheck} is a parameter of the algorithm that must be tuned. The experiments in this study demonstrate that if the value of the T_{precheck} parameter is too small, the generated path is not smooth enough.

The precheck algorithm is illustrated in Figure 4. When there is a moving obstacle in the robot's workspace, the minimal distance and time point can be calculated when the obstacle and the agent are closest to each other.

4.2. Cost-function-based velocity selection using the extended VO method

The safety velocity obstacle method has been defined in a previous study [26]. In this method, a cost function was used when different aspects influenced the motion-planning method (safety, speed). This algorithm is extended with a heading parameter, which provides information on the orientation of the agent in relation to the target position, and the method is also extended with the changing uncertainty parameter.

At every time step, the nearest distance is calculated between the VO cone and the investigated velocities:

$$D_{\mathrm{S}}(\mathbf{v}_A, VO_i) = \min\left\{ \min_{v_{VO} \in VO} ||\boldsymbol{v}_A - \boldsymbol{v}_{VO}||, D_{max} \right\}, \tag{19}$$

where $D_{\max}$ is the maximum distance that should be considered and $\mathbf{v}_{\mathrm{VO}}$ is the nearest point on the VO cone.

The $D_{\mathrm{S}}(\mathbf{v}_{\mathrm{A}})$ value must be normalised into the interval of [0,1]. The $C_{\mathrm{S}}(\mathbf{v}_{\mathrm{A}})$ can then be calculated, which will be used in the cost function later:

$$C_{\mathrm{S}}(\mathbf{v}_{\mathrm{A}}, VO_i) = 1 - \frac{D_{\mathrm{S}}(\mathbf{v}_{\mathrm{A}}, VO_i)}{D_{\max}}. \tag{20}$$

$C_{\mathrm{G}}(\mathbf{v}_{\mathrm{A}})$ will also form part of the cost function:

$$C_{\mathrm{G}}(\mathbf{v}_{\mathrm{A}}) = \frac{\left|\left|\mathbf{p}_{\mathrm{A}} + \mathbf{v}_{\mathrm{A}} T_{\mathrm{s}} - \mathbf{p}_{\mathrm{goal}}\right|\right|}{\left|\left|\mathbf{p}_{\mathrm{A}}(0) - \mathbf{p}_{\mathrm{goal}}\right|\right|}, \tag{21}$$

where T_{s} is the sampling time, $\mathbf{p}_{\mathrm{A}}(0)$ is the first position of the robot at the beginning of the motion and $\mathbf{p}_{\mathrm{goal}}$ is the position of the target. $C_{\mathrm{G}}(\mathbf{v}_{\mathrm{A}})$ denotes how far the robot will be from the target if it uses the selected velocity; subsequently, it has to be divided by the distance of the first position and the desired position.

In this novel algorithm, the prior method is extended by changing $\alpha_i(t)$ parameters (calculated at every time step) for the different obstacles with respect to the reliability of the obstacles' velocity and position information.

The orientation of the agent can also play a role in the cost function. The heading parameter of the cost function can be calculated as follows:

$$C_{\mathrm{h}}(\mathbf{v}_{\mathrm{A}}) = \frac{|angleRG - angleIV(\mathbf{v}_{\mathrm{A}})|}{\pi}, \tag{22}$$

where $angleRG$ refers to the angle of the vector from the robot position to the target position and $angleIV(\mathbf{v}_{\mathrm{A}})$ denotes the angle of the investigated velocity vector of the agent. Using the difference in these angles, the heading parameter can be calculated (angles are defined in the global coordinate system).

The whole cost function can be determined using different parameters:

$$\begin{aligned} \mathrm{Cost}(\mathbf{v}_A) = &\sum_{i=1}^{m} \alpha_i(t)\, C_{\mathrm{S}}(\mathbf{v}_{\mathrm{A}}, VO_i) + \beta_{\mathrm{d}}\, C_{\mathrm{G}}(\mathbf{v}_{\mathrm{A}}) \\ &+ \beta_{\mathrm{h}}\, C_{\mathrm{h}}(\mathbf{v}_{\mathrm{A}}), \end{aligned} \tag{23}$$

where β_{d} is the distance parameter, β_{h} is the heading parameter and $\alpha_i(t)$ denotes the actual calculated uncertainty parameter of an obstacle.

This velocity vector is selected for the agent, which has minimal cost value. The different parameters of the cost function have a significant impact on the velocity selection, as will be presented in Section 5.

4.3. Velocity selection based on the extended APF method

The APF method can be extended using $\alpha_i(t)$ and β_{d} parameters, which were introduced in (15) and (23). The repelling forces must be calculated for every obstacle. The constant η parameter must be substituted with the changing uncertainty parameter, which has a value for every obstacle:

$$\mathbf{F}_{\mathbf{ar_i}} = \frac{\alpha_i(t)\sqrt{\dfrac{1}{D_{\mathrm{ra}_i}} + \dfrac{1}{D_{\mathrm{ra}_{\max}}}}}{D^2_{\mathrm{ra}_i}} \mathbf{AR_i}, \tag{24}$$

where the notations are the same (with the extension of the *i* parameter, which refers to the *i-th* obstacle) as introduced in (3).

The attractive force, $\mathbf{F_{rc}}$, can be calculated in the same way as in (5) by using the β_{d} parameter instead of ξ:

$$\mathbf{F_{rc}} = \beta_{\mathrm{d}}\, \mathbf{RC}. \tag{25}$$

The final force that influences the movement of the agent can be calculated as the addition of the attractive force and the summation of the repelling forces, as presented in (6). After calculating the force that influences the actual movement of the agent, the selectable velocity vector can be calculated using (7), (8) and (9).

5. SIMULATION RESULTS

In this section, the simulation results are discussed based on the changing uncertainties.

5.1. Two static obstacles

In the first example, there are two static obstacles in the workspace of the agent. Initially, using the introduced cost-function-based VO method, the velocity vector that is exactly in the middle of the two obstacles is selected because the two obstacles have the same uncertainties. This situation is presented in Figure 5, in which the VOs are presented as grey areas, the blue circle is the selected velocity vector, the target is depicted by a black x, each of the agent's velocity vectors identified through the motion-planning algorithm is represented by a red x and the robot is the red circle.

The changes and the magnitude of the velocities of the obstacles do not influence the calculation of the uncertainties because there are two static obstacles in the workspace. So, in this case, only the distances between the obstacles and the agent have an impact on the calculation. The velocity vector between the two obstacles is selected, and the distances between the agent and the two obstacles are the same during the motion, resulting in the same uncertainties for both obstacles, as presented in

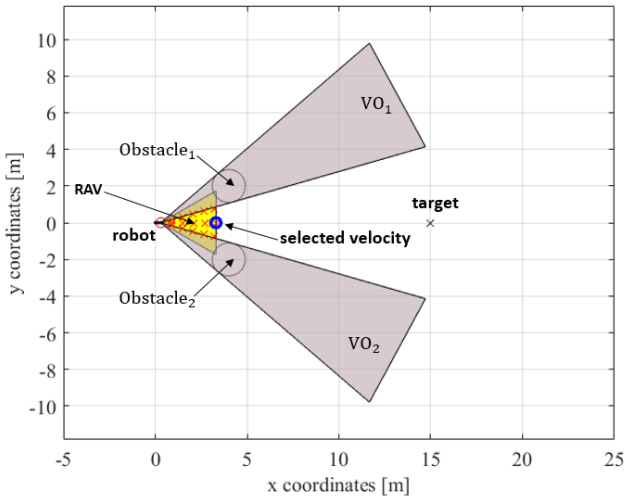

Figure 5. First example: Velocity selection based on the extended VO method.

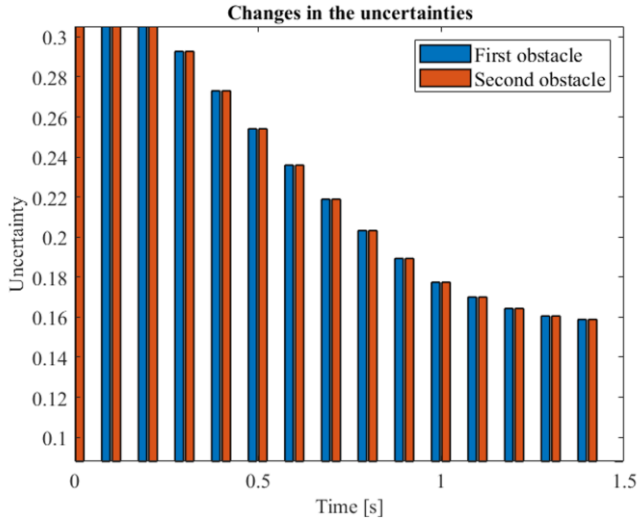

Figure 6. First example: Two static obstacles; changing uncertainties during motion. The uncertainties are the same for both obstacles.

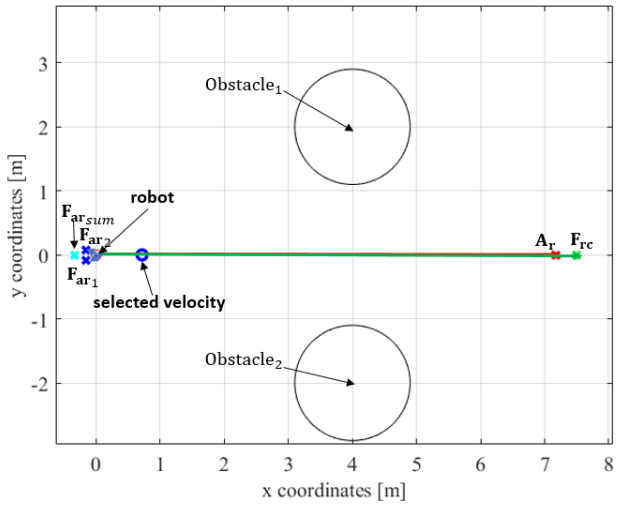

Figure 8. First example: Extended APF method.

Figure 6 (at each time step, the uncertainties for the obstacles can be seen next to each other).

This example was also tested using both the original APF method and the extended APF method.

Using the original APF method that was introduced in Section 2, the agent cannot reach the target position. This is because at the beginning of the motion, the APF method results in a velocity vector that causes a motion in the opposite direction from the target position, as can be seen in Figure 7 (the values of the constant parameters were $\eta = 2$ and $\xi = 0.1$). $\mathbf{F_{ar_1}}$ and $\mathbf{F_{ar_2}}$ represent the repelling forces for the different obstacles, and $\mathbf{F_{ar_{sum}}}$ is the summation of the repelling forces (the other notations are the same as those introduced in previous sections). Eventually, the summation of the forces will become a force in the direction of the target position. The agent will then move towards the target position. This sequence is repeated, resulting in an oscillation without reaching the target position.

The example was also tested with the extended APF method, introduced in Section 4.3. In this case, the reactive motion-planning method can generate a collision-free motion towards the target position. The different forces and the selected velocity vector can be seen in Figure 8. Using this method, the obstacles' uncertainties change in the same way as seen in Figure 6 because the agent executes its motion along the same path between the two obstacles.

5.2. One moving and one static obstacle

In this example, the first obstacle is a moving obstacle, and the second obstacle is a static obstacle. If the agent is at a considerable distance, it can select a velocity vector in line with the target position. After that, if it gets closer to the obstacles, it selects a velocity vector that results in a manoeuvre next to the static obstacle because the corresponding probability is higher. The results of the velocity selection, in this case, are depicted in Figure 9. The path of the robot is presented as a black line.

In this example, the uncertainties of the obstacles are not the same as in the previous example because the static obstacle has a smaller uncertainty throughout the motion, as presented in Figure 10. It can be seen that in the first step, the difference between the obstacles' uncertainties is not significant. This is because the moving obstacle has a small magnitude of velocity and the distances between the obstacles and the robot are the same at the first step.

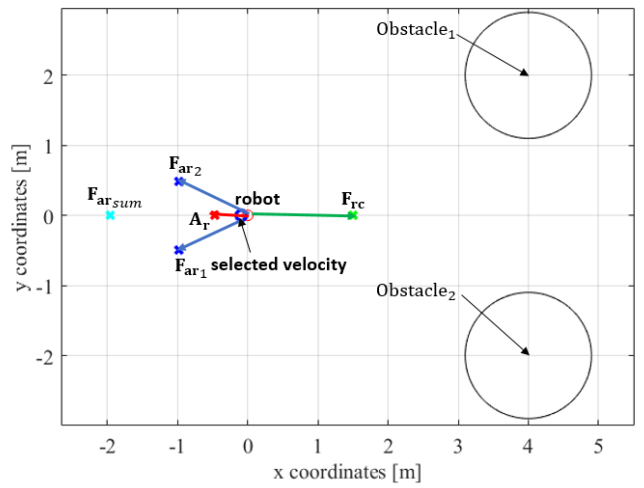

Figure 7. First example: Original APF method.

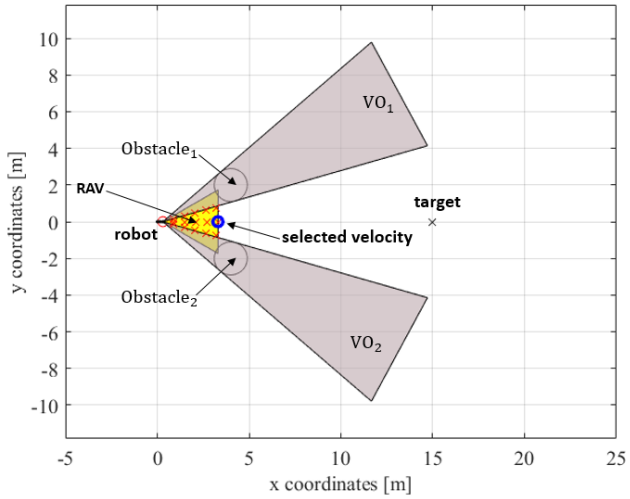

Figure 9. Second example: Velocity selection based on the extended VO method.

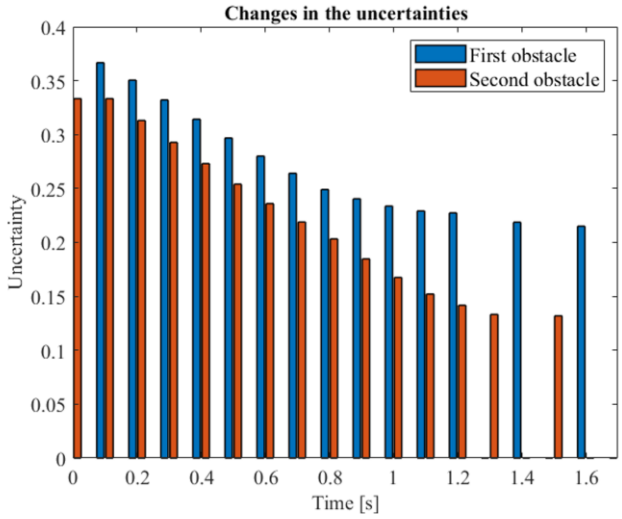

Figure 10. Second example: One moving (first) and one static (second) obstacle; changing uncertainties during motion using the extended VO method. The moving obstacle has a higher uncertainty parameter.

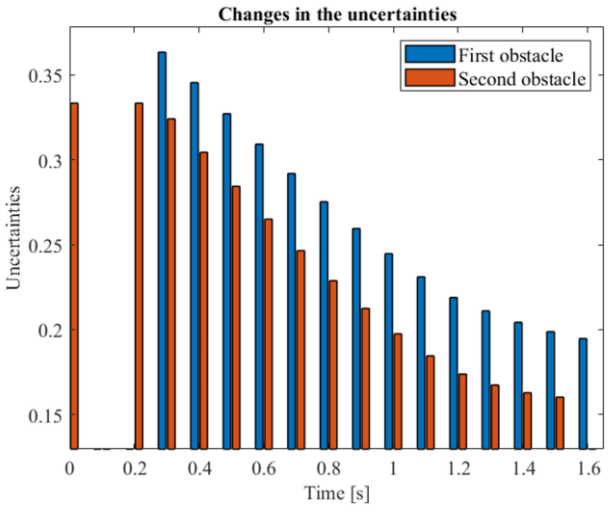

Figure 12. Second example: One moving (first) and one static (second) obstacle; changing uncertainties during motion using the extended APF method. The moving obstacle has a higher uncertainty parameter.

This example was also simulated with the extended APF method. Because the first obstacle has a nonzero velocity vector, this obstacle has a higher uncertainty when using the motion-planning algorithm. However, because the magnitude of the velocity vector is not a large value compared with the summation of the forces, the agent executes its motion almost directly in line with the target position, as presented in Figure 11. So, in this case, there is a difference between the result of the extended VO method and that of the extended APF method, but both of them can result in a collision-free motion between the agent and the environment, and using both methods, the agent can reach the target position. The uncertainties of the obstacles can also be calculated during the motion of the agent with the extended APF method, although the result will be slightly different from the result of the extended VO method.

5.3. Three obstacles in front of each other

In the next example, there are three obstacles in front of each other with different velocities (the first obstacle is static, and the others are moving). Figure 13 shows the velocity selection of the VO method next to the first obstacle, and Figure 14 presents the velocity selection next to the second obstacle. It can be seen that a further velocity component is selected at the second obstacle because it has a higher velocity.

The higher the velocity of the obstacle, the bigger the uncertainty for the obstacle, as depicted in Figure 15, in which the uncertainty parameters are presented as aspects of the three obstacles. After passing the obstacle, the uncertainty is reduced. This figure shows that not all the obstacles need to be considered throughout the motion, only those that influence the motion of the robot and which have fulfilled the precheck algorithm at the time of sampling.

The β_h parameter plays a significant role in the cost-function-based velocity selection as a factor in the target-reaching strategy. In the previous examples, the value of β_h was 0.3. If this parameter has a higher value, it has a larger impact on the motion than the uncertainties of the obstacles, as presented in Figure 16. In this case (β_h = 0.6), the agent executes the motion as close to the obstacle as the collision-free motion-planning algorithm allows.

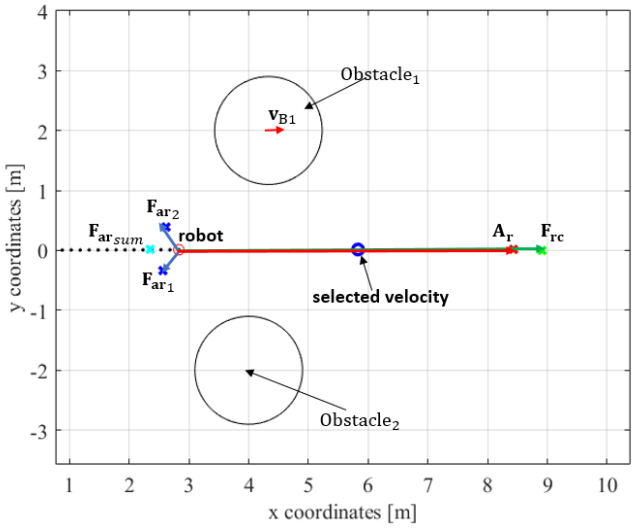

Figure 11. Second example: Velocity selection based on the extended APF method.

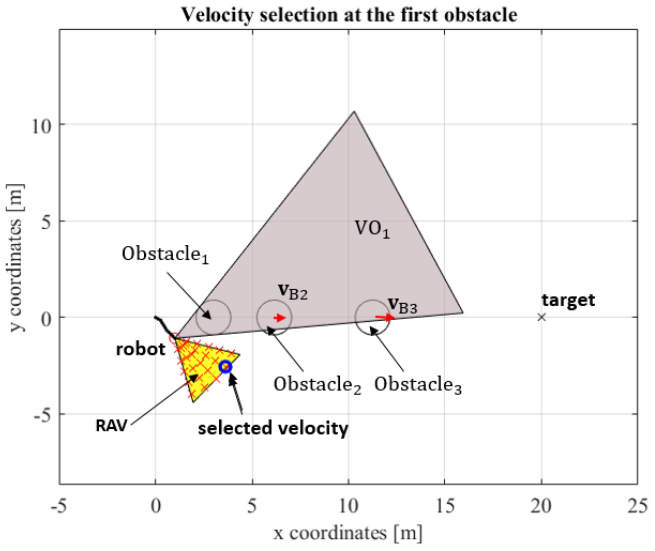

Figure 13. Third example: Velocity selection at the first obstacle using the extended VO method.

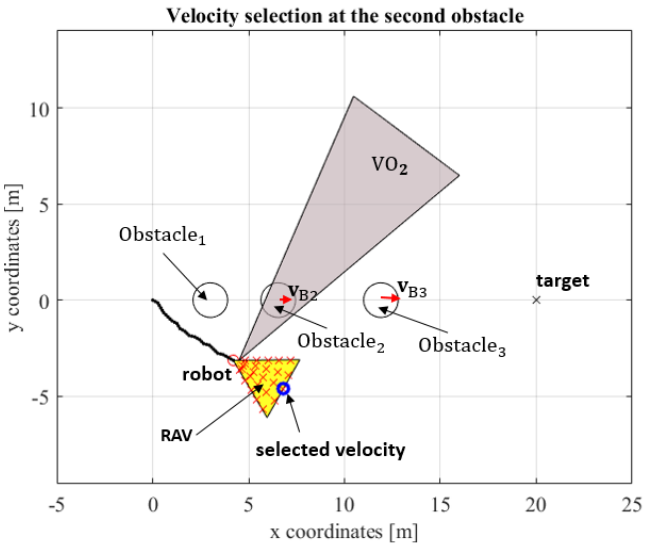

Figure 14. Third example: Velocity selection at the second obstacle using the extended VO method.

The values of the parameters depend on the usage of the algorithm; different parameter values generate different results using the collision-free motion-planning algorithm. However, it is not possible to find a solution that takes into account every aspect of the motion-planning problem. A sub-optimal solution must therefore always be calculated.

This example can also be tested using the extended APF method. Only the first obstacle should be considered for the motion-planning algorithm because (18) is the appropriate equation for the first obstacle (the second and third obstacles are at a distance from the agent). In this case, the algorithm selects a velocity vector for the agent that results in a movement in the exact direction of the first obstacle (because the summation of the forces is in line with the movement of the agent). So, after a few steps, the agent reaches and collides with the first obstacle. In this example, the extended APF method cannot guarantee that the robot will reach the target collision free.

5.4. Standard VO method and the novel motion-planning method

The introduced novel motion-planning algorithm was also compared with the original VO method because the basic concept of the motion-planning algorithm is based on this algorithm.

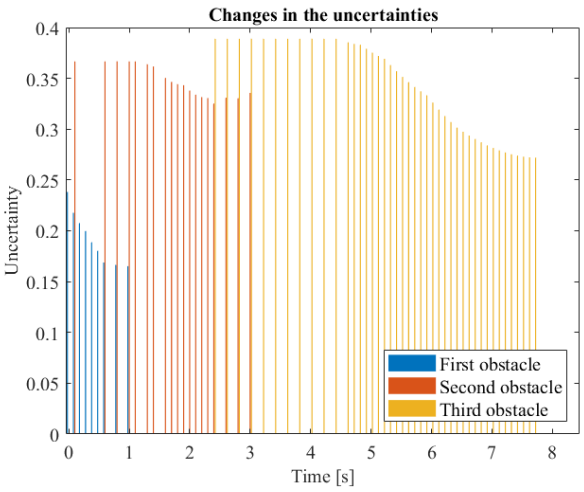

Figure 15. Third example: Three obstacles in front of each other with different velocities; changing uncertainties during motion using the extended VO method.

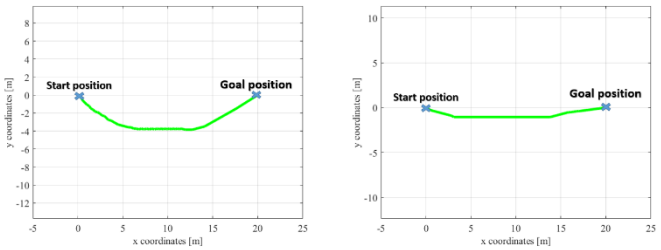

Figure 16. The resulting motion paths of the robot with different heading parameters; in the first example β_h = 0.3, in the second example β_h = 0.6.

The comparison used the example of two moving obstacles in the robot's workspace. One of them has a changing velocity vector that results in a higher uncertainty in the motion of the robot. Figure 17 shows the final path of the robot using the different motion-planning algorithms. It can be seen that using the standard VO method, which provides the fastest target-reaching concept, the agent executes a tangential motion next to the first obstacle (this is also presented in a video [27]). However, if the uncertainties in the measurement data are also considered, the target-reaching method will be solved, generating a path that is relatively far from the first obstacle (with changing velocities). The motion of the robot is also presented in a video [28]. Figure 18 represents the distances between the agent and the obstacles using the different motion-planning algorithms. As has already been mentioned, when using the standard VO method, there is a time point at which the distance between the robot and the obstacle is zero. In the case of the novel motion-planning algorithm, the uncertainties can be taken into account, so the agent can move safely towards the target position. However, if there is even a tiny measurement or system noise in the process, the tangential movement will immediately cause a collision. So, if this occurs, it is better to use the novel motion-planning algorithm, which generates a collision-free motion for the agent in every situation.

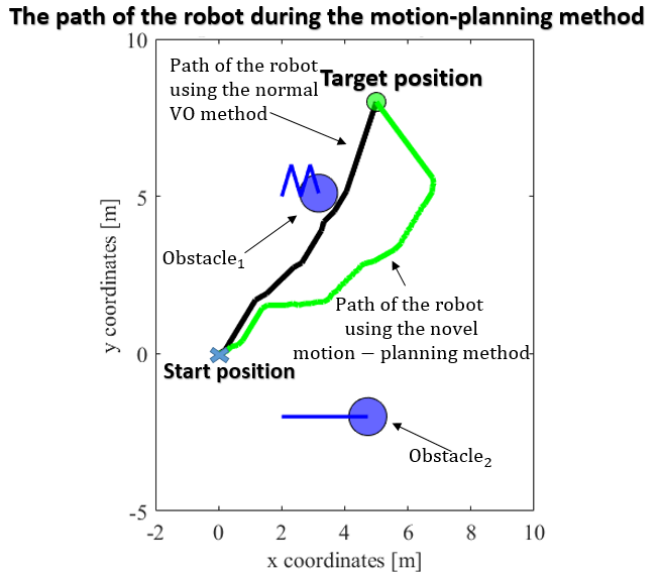

Figure 17. Final paths of the robot using the normal VO method and the novel motion-planning algorithm.

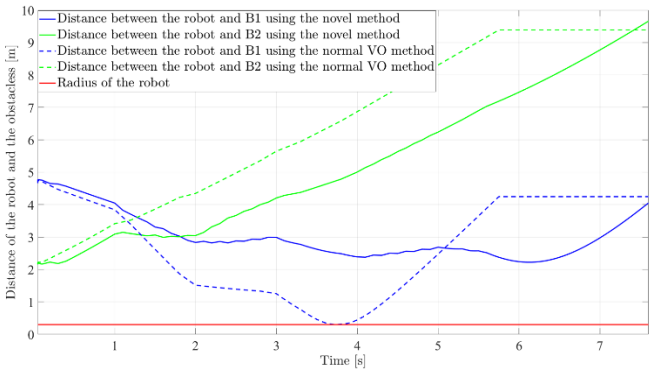

Figure 18. Distances between the robot and the obstacles using the different motion-planning algorithms.

6. COPPELIASIM SIMULATION ENVIRONMENT

CoppeliaSim (VREP version) is suitable for testing robotic arms as well as holonomic and non-holonomic mobile robots using reactive motion-planning algorithms. Different types of obstacles can occur in the workspace of the robot, and there are a wide range of obstacles that can be used in the simulation environment

The results of the introduced methods were tested in the CoppeliaSim simulation environment, as presented in [29].

The agent is an omnidirectional robot (blue). This type of mobile robot is often used because it can execute its motion in any direction from an actual position. In the example in Figure 19, there are two static obstacles (grey cylinders) in the workspace of the agent. The main goal of the robot is to reach the target position without colliding with the two obstacles, as presented in Section 5.1.

7. CONCLUSIONS

In this paper, novel motion-planning methods were introduced using the basics of the VO and the APF methods. The mobile robot was able to execute collision-free motion planning after calculating the changing uncertainties of the obstacles. These uncertainties depend on the magnitudes of the velocity vectors of the obstacles, the distances between the obstacles and the robot, and the changes in the obstacles' velocities.

The VO-based method can generate collision-free motion using a cost-function-based optimisation method.

The basic APF method was also extended by using the uncertainty and distance parameters in the algorithm. The extended APF method can generate a better solution than the original APF method, but there are some situations in which it cannot provide a target-reaching solution. In these cases, the cost-function-based VO method was able to guarantee that the target was reached. The parameters for the APF method could also be calculated in another way, thus solving the local minima problem [30], [31].

The introduced algorithm could be implemented in a real robotic system using an omnidirectional mobile robot. The state estimation of the obstacles that occur in the workspace of the robot could be solved using an extended particle filter algorithm. In this case, the position and the velocity vectors of the obstacles could be estimated for every sampling time [32]. To achieve this, a Lidar sensor can be used.

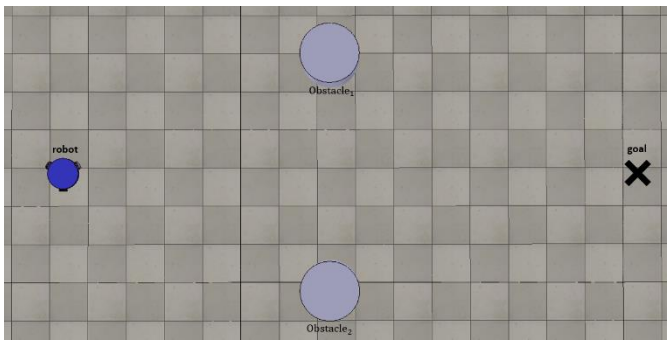

Figure 19. CoppeliaSim simulation environment [29].

ACKNOWLEDGEMENT

This paper was supported by the ÚNKP-20-3 new National Excellence Programme of the Ministry for Innovation and Technology from the National Research, Development and Innovation Fund, and the research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 Institution Excellence Sub-Programme, Grant No. BME-IE-MIFM) based on the charter issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

REFERENCES

[1] S. Panov, S. Koceski, Metaheuristic global path planning algorithm for mobile robots, Int. J. of Reasoning-Based Intelligent Systems 7 (2015), p. 35.
DOI: 10.1504/ijris.2015.070910

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

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

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

[5] P. Raja, S. Pugazhenthi, Optimal path planning of mobile robots: a review, Int. J. of Physical Sciences 7 (9), February 2012, pp. 1314-1320.
DOI: 10.5897/IJPS11.1745

[6] S. Petti, T. Fraichard, Safe motion planning in dynamic environments, Proc. of the IEEE RSJ Int. Conf. Intell. Robot. Syst., Edmonton, Canada, 2-6 August 2005, pp. 2210-2215.

[7] T. Fraichard, H. Asama, Inevitable collision states - a step towards safer robots, Advanced Robotics 18 (2004), pp. 1001-1024.
DOI: 10.1163/1568553042674662

[8] L. Martinez-Gomez, T. Fraichard, Collision avoidance in dynamic environments: an ICS-based solution and its comparative evaluation, Proc. of the IEEE Int. Conf. on Robotics and Automation, Kobe, Japan, 12 – 17 May 2009, pp. 100-105.
DOI: 10.1109/ROBOT.2009.5152536

[9] D. Fox, W. Burgard, S. Thrun, The dynamic window approach to collision avoidance, IEEE Robot. Autom. Mag. 4 (1997), pp. 23-33.
DOI: 10.1109/100.580977

[10] M. Seder, I. Petrovic, Dynamic window based approach to mobile robot motion control in the presence of moving obstacles, Proc. of the Int. Conf. on Robotics and Automation, Rome, Italy, 10-14 April 2007, pp. 1986-1991.
DOI: 10.1109/ROBOT.2007.363613

[11] M. Mujahed, D. Fischer, B. Mertsching, Admissible gap navigation: a new collision avoidance approach, Robotics and Autonomous Systems 103 (2018), pp. 93-110.
DOI: 10.1016/j.robot.2018.02.008

[12] N. Hacene, B. Mendil, Autonomous navigation and obstacle avoidance for a wheeled mobile robots: a hybrid approach, Int. J. of Computer Applications 81 (2013), pp. 34-37. Online [Accessed 20 September 2021]
https://research.ijcaonline.org/volume81/number7/pxc3892285.pdf

[13] P. Fiorini, Z. Shiller, Motion planning in dynamic environments using velocity obstacles, Int. J. of Robotics Research 17 (1998), pp. 760-772.
DOI: 10.1177/027836499801700706

[14] J. van den Berg, M. Lin, D. Manocha, Reciprocal velocity obstacles for real-time multi-agent navigation, Proc. of the IEEE Int. Conf. on Robotics and Automation, Pasadena, USA, 19-23 May 2008, pp. 1928-1935.
DOI: 10.1109/ROBOT.2008.4543489

[15] J. Alonso-Mora, P. Beardsley, R.. Siegwart, Cooperative collision avoidance for nonholonomic robots, IEEE Transactions on Robotics 34 (2018), pp. 404-420.
DOI: 10.1109/TRO.2018.2793890

[16] B. Gopalakrishnan, A. K. Singh, M. Kaushik, K. M. Krishna, D. Manocha, PRVO: probabilistic reciprocal velocity obstacle for multi robot navigation under uncertainty, Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems, Vancouver, Canada, 24 – 28 September 2017, pp. 1089-1096.
DOI: 10.1109/IROS.2017.8202279

[17] D. Claes, D. Hennes, K. Tuyls, W. Meeussen, Collision avoidance under bounded localization uncertainty, Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems, Vilamoura, Portugal, 7 – 12 October 2012, pp. 1192-1198.
DOI: 10.1109/IROS.2012.6386125

[18] D. Hennes, D. Claes, W. Meeussen, K. Tuyls, Multi-robot collision avoidance with localization uncertainty, Proc. of the 11th Int. Conf. on Autonomous Agents and Multiagent Systems, AAMAS 2012: Innovative Applications Track, Valencia, Spain, 4-8 June 2012, pp. 672-679.

[19] E. Masehian, Y. Katebi, Robot motion planning in dynamic environments with moving obstacles and target, Int. J. of Mechanical Systems Science and Engineering 1 (2007), pp. 107-112. Online [Accessed 20 September 2021]
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.193.6430&rep=rep1&type=pdf

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

[21] A. Masoud, A harmonic potential approach for simultaneous planning and control of a generic UAV platform, J. Intell. Robot. Syst. 65 (2012), pp. 153-173.
DOI: 10.1007/s10846-011-9570-8

[22] N. Malone, H. T. Chiang, K. Lesser, M. Oishi, L. Tapia, Hybrid dynamic moving obstacle avoidance using a stochastic reachable set-based potential field, IEEE Transactions on Robotics 33 (2017), pp. 1124-1138.
DOI: 10.1109/TRO.2017.2705034s

[23] H. Chiang, N. Malone, K. Lesser, M. Oishi, L. Tapia, Path-guided artificial potential fields with stochastic reachable sets for motion planning in highly dynamic environments, Proc. of the IEEE Int. Conf. on Robotics and Automation, Seattle, USA, 26-30 May 2015, pp. 2347-2354.
DOI: 10.1109/ICRA.2015.7139511

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

[25] Z. Gyenes, E. G. Szadeckzy-Kardoss, Rule-based velocity selection for mobile robots under uncertainties, Proc. of the 24th Int. Conf. on Intelligent Engineering Systems, Reykjavík, Iceland, 8-10 July 2020, pp. 127-132.
DOI: 10.1109/INES49302.2020.9147191

[26] Z. Gyenes, E. G. Szadeckzy-Kardoss, Motion planning for mobile robots using the safety velocity obstacles method, Proc. of the 19th International Carpathian Control Conference, Szilvásvárad, Hungary, 28-31 May 2018, pp. 389-394.
DOI: 10.1109/CarpathianCC.2018.8473397

[27] Akta IMEKO standard VO, Online [Accessed 17th September 2021]
https://youtu.be/Jp6m3ngWjpk

[28] Akta Imeko Uncertainties, Online [Accessed 17th September 2021]
https://youtu.be/Mxz1OM3bzYs

[29] Omnirobot moves between two static obstacles, Online [Accessed 17th September 2021]
https://www.youtube.com/watch?v=HJcNTdKS6-o&feature=youtu.be

[30] M. G. Park, M. C. Lee, A new technique to escape local minimum in artificial potential field based path planning, KSME Int. J. 17 (2003), pp. 1876-1885.
DOI: 10.1007/BF02982426

[31] G. Guerra, D. Efimov, G. Zheng, W. Perruquetti, Avoiding local minima in the potential field method using input-to-state stability, Control Engineering Practice 55 (2016), pp.174-184.
DOI: 10.1016/j.conengprac.2016.07.008

[32] Z. Gyenes, E. G. Szadeckzy-Kardoss, Particle filter-based perception method for obstacles in dynamic environment of a mobile robot, Proc. of the 25th IEEE International Conference on Methods and Models in Automation and Robotics, Miedzyzdroje, Poland, 23-26 August 2021, p. 6 (in press).