Motion Planning for Mobile Robots, Combining Artificial Potential Field and Velocity Obstacle Algorithms

Norbert Török and Zoltán Gyenes

Department of Control Engineering and Information Technology,
Budapest University of Technology and Economics,
Műegyetem rkp. 3., H-1111 Budapest, Hungary

e-mail: torok.norbert@edu.bme.hu, zgyenes@iit.bme.hu

Abstract: *Effective navigation of mobile robots from the start to the target position requires the ability to safely avoid collisions with both static and dynamic obstacles. To achieve this, motion planning algorithms play a crucial role in devising efficient paths and generating real-time avoidance maneuvers that ensure the robot's safety and the integrity of its surroundings. In this paper, we present a novel reactive motion planning algorithm that integrates the Artificial Potential Field (APF) method and the Velocity Obstacle (VO) method. The APF method is limited by local minimum and local oscillation problems, which our combined approach effectively addresses. We compare our algorithm with the improved APF method and the extended APF method using simulated annealing, evaluating performance through metrics such as path lengths and elapsed time to reach the goal position. Our results demonstrate that the integrated APF-VO algorithm outperforms the other approaches, achieving the best performance in terms of navigation efficiency and obstacle avoidance. The research results present an advancement in motion planning algorithms, offering a robust and efficient solution for mobile robots to navigate complex environments safely and effectively.*

Keywords: Artificial Potential Field, Velocity Obstacle, Simulated Annealing, mobile robot, reactive motion planning

1 Introduction

Thanks to the advancement and widespread adoption of technology, autonomous robots are becoming increasingly common in various fields. Examples include robotic vacuum cleaners in households, automated high-bay warehouses, and delivery robots operating in certain cities [1].

1.1 Motion Planning Algorithms

The primary task of mobile robots is to navigate from a starting position to a target location without collisions. To achieve this, motion planning algorithms are used, which can be classified into two main categories [2].

1.1.1 Global Motion Planning Algorithms

Global motion planning algorithms determine an optimal path based on prior knowledge of the robot's environment. These methods require a map that includes the starting and target positions, as well as all obstacles [3]. Since the path is planned before the robot begins its movement, these algorithms are only suitable for handling stationary obstacles and are not designed to cope with unexpected or moving obstacles. Furthermore, beyond point-to-point navigation, optimization techniques are also crucial for specialized global tasks, such as robot coverage path planning [4].

A commonly used global motion planning algorithm is the Rapidly-exploring Random Tree (RRT), which starts from the robot's initial position and constructs a tree graph from randomly selected points near the root. This process is then cyclically extended from the leaves, mapping out the surrounding space. If the endpoint of one of the tree branches comes sufficiently close to the target position, the robot has found a collision-free path to the goal and only needs to follow the appropriate edges of the tree. However, standard RRT algorithms often produce sub-optimal, non-smooth trajectories and their computational cost increases significantly in highly constrained or complex environments [5-7].

A method like RRT is the Probabilistic Roadmaps (PRM), which randomly selects points in the safe areas of the map and connects them if the edge forms a feasible path for the robot. This results in a graph, but unlike RRT, it is not a tree, rather it is a simple graph. The graph includes both the starting and target positions as nodes, and now, a pathfinding algorithm is used to find a trajectory that the robot can follow. A major limitation of PRM is the well-known "narrow passage problem," where sampling points in tight spaces becomes highly inefficient. Furthermore, adapting the pre-computed roadmap to dynamic environmental changes requires immense computational overhead [8].

1.1.2 Local Motion Planning Algorithms

Local – or reactive – motion planning algorithms do not require prior information to plan the path. In reactive motion planning, the robot decides its moving direction while moving, allowing it to respond to changes in the environment. Over the past few decades, several approaches have been introduced.

One well-known method is Model Predictive Control (MPC), which attempts to predict the movement of obstacles and based on this, optimally control the robot. While effective, the widespread application of MPC is often hindered by its high computational demands and its heavy reliance on the accuracy of the mathematical prediction models, which is challenging in unpredictable dynamic spaces [9].

Another well-known solution is the Dynamic Window Approach (DWA), which controls the robot's movement based on the dynamic characteristics. The algorithm narrows the robot's possible velocity space, to a dynamic window, which contains the velocities that are physically achievable within a given time interval, considering the robot's acceleration, deceleration capabilities, and collision-free movement. Within this window, a cost function is used to decide on the velocity vector. Despite its practical success in obstacle avoidance, DWA is a fundamentally short-sighted method, making it highly susceptible to getting trapped in local minima, especially in complex, U-shaped obstacle configurations [10].

It is also important to note that robust state estimation, such as accurate pose estimation algorithms [11] and localization plays a huge role in the local motion planning algorithms [12].

The aforementioned limitations of existing global and local algorithms – particularly high computational demands, inefficiencies in dynamic environments, and susceptibility to local minima – highlight the ongoing need for more robust reactive solutions. Motivated by these challenges, we introduce the novel method presented in the following sections, which aims to provide safe and efficient navigation by addressing and overcoming these specific limitations.

Among local planning algorithms, the Artificial Potential Field (APF) method has gained significant attention due to its mathematical elegance and real-time performance [17]. However, the literature extensively documents the inherent flaws of traditional APF, most notably the local minimum and local oscillation problems [20]. To mitigate these issues, various structural improvements to the potential field functions have been proposed [19][21][28], alongside active escape mechanisms such as Simulated Annealing [24], Wall-following [30], and Virtual Obstacles [31]. In parallel, the Velocity Obstacle (VO) algorithm and its variants have proven highly effective for navigating safely among static and dynamic obstacles by analyzing the velocity space [13][15]. Despite these advancements, relying solely on APF-based escape mechanisms can still lead to inefficient paths in complex scenarios. Conversely, while advanced VO strategies effectively handle complex obstacle layouts, continuously computing Velocity Obstacles and Reachable Avoidance Velocities for every step is computationally more demanding than simple APF calculations. This gap motivates the development of a streamlined, hybrid approach. In this paper, we aim to merge the computational efficiency and intuitive goal-attraction of an Improved APF with the robust, multi-obstacle maneuvering capabilities of the Maximum Velocity VO (MVVO) strategy, creating an algorithm that specifically balances low computational load with navigation safety.

The key findings of this paper can be summarized as:

- We propose a novel local motion planning approach by combining the Artificial Potential Field (APF) and Velocity Obstacle (VO) methods for enhanced path planning performance.
- The proposed algorithm overcomes common limitations of the APF method, such as local minima and local oscillations.
- We validate the effectiveness of our approach through simulations, comparing it with improved and extended APF techniques.
- Our algorithm demonstrates superior performance in terms of path length and time to reach the goal.
- The algorithm shows promise for future applications in various autonomous navigation domains.

The organization of this paper is as follows: Section 2 presents the background of the Velocity Obstacle motion planning method. Section 3 shows the basics of the Artificial Potential Field (APF) local motion planning algorithm and presents its limitation during different motion planning scenarios. In Section 4, we introduce a novel algorithm which combines the APF and VO method. After that, the introduced algorithm is compared with the previous methods in Section 5. Finally, Section 6 the main conclusions of the introduced approach, are summarized.

2 Background of the Velocity Obstacle Algorithm

The Velocity Obstacle is a first-order reactive motion planning algorithm, meaning it does not integrate velocities to provide positions as a function of time. The method was first introduced by Paolo Fiorini and Zvi Shiller in 1998 [13]. As with most motion planning algorithms, three main objects are distinguished in this method: the robot, the obstacle, and the goal. In this case, there is only one robot. There can be multiple obstacles, which can be either static or dynamic. The goal is a point, and in the current application, there is only one goal with a fixed position.

The listed entities are located within a workspace, which is a two-dimensional Euclidean space. The robot and the obstacles are defined by their current position and velocity vector, and these are assumed to be known. In terms of kinematics, the algorithm assumes omnidirectional, holonomic mobile robots capable of moving in any direction within the plane, this assumption is maintained throughout the paper.

Furthermore, both the robot and the obstacles are assumed to be disk-shaped objects throughout this work. This will not limit the applicability of the method, as any polygon can be constructed as a union of circles [14]. However, it is practical because the circular robot can be treated as a point if the radii of the obstacles are increased by the robot's original radius [15].

2.1 Collision Cone and Velocity Obstacle

The essence of the VO method is that it divides the space of velocity vectors available to the robot into two parts: one consists of all points where choosing a velocity vector would lead to a collision with an obstacle if neither changes its velocity vector, while the other is its complement, containing points that correspond to velocity vectors ensuring collision-free motion. The division is facilitated by Collision Cones, which are defined as follows:

$$CC_{\hat{R},\hat{O}} = \{\boldsymbol{v}_{\hat{R},\hat{O}} \mid \lambda_{\hat{R},\hat{O}} \cap \hat{O} \neq \emptyset\} \tag{1}$$

where $\boldsymbol{v}_{\hat{R},\hat{O}} = \boldsymbol{v}_{\hat{R}} - \boldsymbol{v}_{\hat{O}}$ is the $\hat{R}$ robot's relative velocity vector in proportion to the $\hat{O}$ obstacle, and $\lambda_{\hat{R},\hat{O}}$ is the line fitted to $\boldsymbol{v}_{\hat{R},\hat{O}}$.

2.1.1 Static obstacle

In the case of a static obstacle, the Collision Cone corresponds to the infinitely extended cone defined by the two tangents drawn from the robot's point to the obstacle. In this case $\boldsymbol{v}_{\hat{R},\hat{O}} = \boldsymbol{v}_{\hat{R}}$, meaning the relative velocity vector is equal to the robot's velocity vector since the obstacle is stationary $\boldsymbol{v}_{\hat{O}} = \emptyset$. This implies that if the robot's velocity vector is within the Collision Cone, i.e., the velocity vector points toward the obstacle, the robot will collide with the obstacle as long as it remains static.

2.1.2 Moving Obstacle

In the case of moving obstacles $\boldsymbol{v}_{\hat{R},\hat{O}} = \boldsymbol{v}_{\hat{R}} - \boldsymbol{v}_{\hat{O}}$, meaning the relative velocity vector is not equal to the robot's velocity vector, since $\boldsymbol{v}_{\hat{O}} \neq \emptyset$. It would be more practical when dealing with multiple obstacles, not to check each obstacle's relative velocity vector and its corresponding Collision Cone separately. Instead, by knowing the robot's single velocity vector and the dangerous and safe regions, we could determine whether $\boldsymbol{v}_R$ is appropriate or not.

To achieve this, we introduce the Velocity Obstacle:

$$VO_{\hat{R},\hat{O}} = CC_{\hat{R},\hat{O}} \oplus \boldsymbol{v}_{\hat{O}} \tag{2}$$

where $\oplus$ is the Minkowski addition operator [16]. The expression means that we add the $\boldsymbol{v}_{\hat{O}}$ vector to every element of the set $CC_{\hat{R},\hat{O}}$, i.e., we translate the Collision Cone associated with the given obstacle by the obstacle's velocity vector. Naturally, for static obstacles, the Collision Cone and the Velocity Obstacle are the same. A collision occurs in the future if the robot's velocity vector points in the Velocity Obstacle set.

2.2 Dangerous and safe regions

The union of the sets of Velocity Obstacles associated with static and moving obstacles gives us the portion of the workspace from which, by choosing a velocity vector, the robot will collide with some obstacle at a future moment. However, if we choose a velocity vector from outside this region, the robot will move safely. The dangerous region of the workspace is:

$$VO_{\hat{R}} = \cup_{i=1}^{m} VO_{\hat{R},\hat{O}_i} \tag{3}$$

where m represents the number of obstacles. The safe region is the complement of the dangerous region, i.e., $\overline{VO_{\hat{R}}}$. This is illustrated in Figure 1.

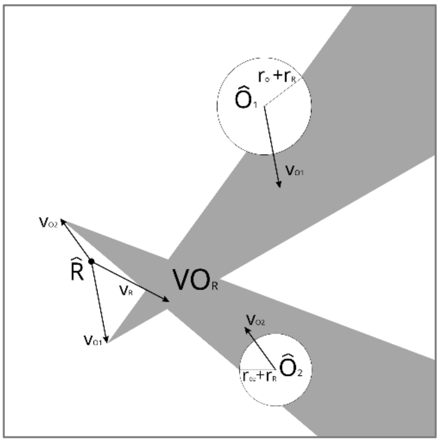

Figure 1: Dangerous (grey) and safe (white) regions

2.3 Collision Avoidance Strategies

Collision avoidance strategies consider various aspects to select desirable velocity vectors, such as algorithms that prioritize the fastest travel or the safest maneuvering between obstacles. However, what all these strategies have in common is that they must ensure the robot's safe movement until it reaches the goal.

Before discussing the strategies, we need to introduce the Reachable Velocities (RV) set, which contains the points that the robot's velocity vector can take. If we subtract the $VO_{\hat{R}}$ from the Reachable Velocities, we obtain the Reachable Avoidance Velocities (RAV) set, which contains the available velocity vectors that will not lead to a collision (Figure 2).

$$RAV = RV \backslash VO_{\hat{R}} \tag{4}$$

If the strategies select the most suitable velocity vector from the RAV set based on their algorithm, they automatically guarantee collision-free motion.

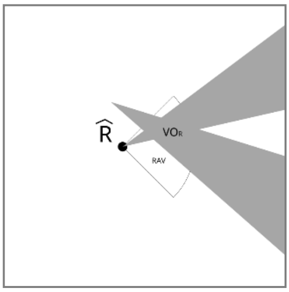

Figure 2: The available velocity vectors that result in safe movement
(RAV, the white sections of the circular sector)

2.3.1 To goal (TG)

TG is one of the simplest VO strategies. Its principle is that it selects the highest speed vector from those within the RAV, along the line drawn from the robot to the goal. Its drawback is that it is only applicable in limited cases, since if there is a static obstacle along the line or if the line does not intersect the RAV, the robot will never reach its goal.

2.3.2 Maximum velocity (MV)

The MV strategy is an enhancement of the TG strategy, where we no longer examine the velocity vectors along a single line. Instead, the segment connecting the robot and the goal is rotated at the robot's position by a given angle in both directions, creating an area within which the fastest velocity vector is selected from the RAV.

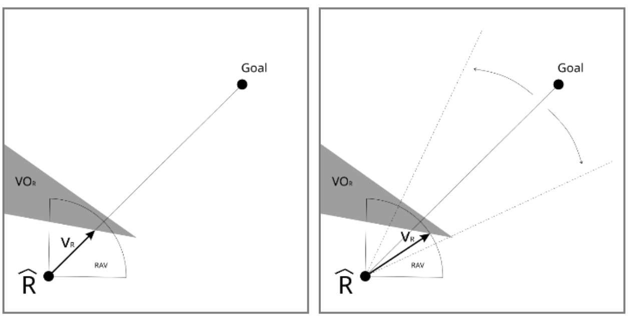

Figure 3: The TG (left) and MV (right) strategies

3 Introduction to the Artificial Potential Field Algorithm

Like the Velocity Obstacle algorithm, in the Artificial Potential Field method, the robot, the goal, and the static or dynamic obstacles are placed within a two-dimensional workspace. For simplicity and to allow a fair comparison of the two methods, we also assume that the positions and velocity vectors of the obstacles and the goal are known. Additionally, we treat the robot as a point.

The APF algorithm was first introduced in Oussama Khatib's 1986 paper [17], and since then, the method has become a widely used approach for solving trajectory planning tasks of mobile robots. It is favored for its simplicity, mathematical elegance, low computational cost, and good real-time performance. The popularity of the method is evident, as many researchers used and extended the original method in the recent years [19][24].

3.1 Electric field

The elegance of the Artificial Potential Field (APF) method lies in its use of electrostatics – originally unrelated to robotics – where motion planning mimics interactions between electric charges, with like charges repelling and opposites attracting, guiding paths via analogous force fields [29].

The electric scalar potential, or potential field, is a scalar field that assigns the value of the electric potential at each point in space, which corresponds to the electric potential at that particular point:

$$\Phi(\boldsymbol{p}) = \int_{\boldsymbol{p}}^{\boldsymbol{p_0}} \boldsymbol{E}\, d\boldsymbol{l} \tag{5}$$

$\boldsymbol{E}$ is the electric field, the position vector $\boldsymbol{p}$ pointing to the examined point, and $\boldsymbol{p_0}$ is a fixed point where the electric potential energy is considered to be zero. However, the potential field can also be determined from the electric field, not just the other way around:

$$\boldsymbol{E} = -grad[\Phi(\boldsymbol{p})] \tag{6}$$

The vector field can be calculated as the negative gradient of the scalar field [18].

The basic idea of APF is to treat the workspace of the agent as an electric field, where the goal is a negative charge, the obstacles are all positive charges, and the robot is a test charge that does not influence the field. In other words, the goal exerts an attractive force on the robot, while the obstacles exert a repulsive force.

3.2 Traditional Artificial Potential Field

The APF field is formed as the sum of two potential fields. One field exerts a repulsive force on the robot to keep it away from obstacles, while the other exerts an attractive force to guide the robot toward the goal. This is shown in Figure 4.

The potential field influencing the robot's movement is obtained as the sum of the repulsive and attractive fields:

$$\Phi(\boldsymbol{p}_{\hat{R}}) = \Phi_{att}(\boldsymbol{p}_{\hat{R}}) + \Phi_{rep}(\boldsymbol{p}_{\hat{R}}) \tag{7}$$

where $\Phi(\boldsymbol{p}_{\hat{R}})$ is the resulting potential field, i.e., the APF, $\Phi_{att}(\boldsymbol{p}_{\hat{R}})$ is the attractive potential field, $\Phi_{rep}(\boldsymbol{p}_{\hat{R}})$ is the repulsive potential field, and $\boldsymbol{p}_{\hat{R}}$ coordinates of the robot's position. The electric field can be obtained as the negative gradient of the scalar potential:

$$\begin{aligned}\boldsymbol{F}(\boldsymbol{p}_{\hat{R}}) &= -grad[\Phi(\boldsymbol{p}_{\hat{R}})] \\ &= -grad[\Phi_{att}(\boldsymbol{p}_{\hat{R}})] - grad[\Phi_{rep}(\boldsymbol{p}_{\hat{R}})] \\ &= \boldsymbol{F}_{att}(\boldsymbol{p}_{\hat{R}}) + \boldsymbol{F}_{rep}(\boldsymbol{p}_{\hat{R}})\end{aligned} \tag{8}$$

where $\boldsymbol{F}(\boldsymbol{p}_{\hat{R}})$ is the resulting force, $\boldsymbol{F}_{att}(\boldsymbol{p}_{\hat{R}})$ is the attractive force, $\boldsymbol{F}_{rep}(\boldsymbol{p}_{\hat{R}})$ is the repulsive force [19].

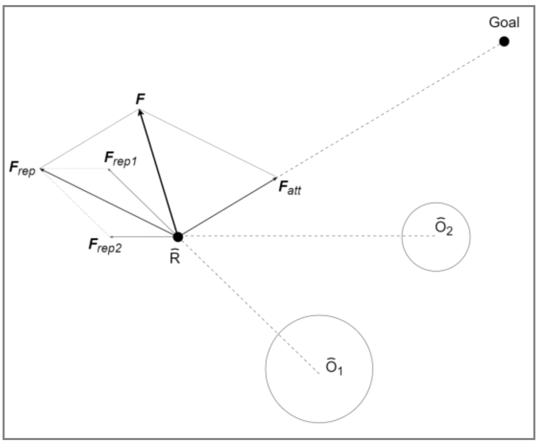

Figure 4: Illustration of the forces effecting on the robot

3.2.1 Attractive potential field and force

The traditional APF refers to the method implemented with the equations defined by Khatib [17]. Accordingly, the attractive potential field that moves the robot towards the goal is:

$$\Phi_{att}(\boldsymbol{p}_{\hat{R}}) = \frac{1}{2} * \xi * |\boldsymbol{p}_{\hat{R}} - \boldsymbol{p}_{\hat{G}}| \tag{9}$$

where ξ is the positive constant characteristic of attraction, $\boldsymbol{p}_{\hat{R}}$ is the position vector of robot, $\boldsymbol{p}_{\hat{G}}$ is the position vector of the goal, thus $|\boldsymbol{p}_{\hat{R}} - \boldsymbol{p}_{\hat{G}}|$ expressing the Euclidean distance between the robot and the goal.

The attractive force vector field is obtained as the negative gradient of the scalar field:

$$\boldsymbol{F}_{att}(\boldsymbol{p}_{\hat{R}}) = -grad[\Phi_{att}(\boldsymbol{p}_{\hat{R}})] = -\xi * (\boldsymbol{p}_{\hat{R}} - \boldsymbol{p}_{\hat{G}}) \tag{10}$$

where the expression $(\boldsymbol{p}_{\hat{R}} - \boldsymbol{p}_{\hat{G}})$ now represents the difference between the two vectors, not a distance, meaning that the farther the robot is from the target, the stronger the attractive force exerted on the robot. This property can lead to collisions if the robot is too far from the target. The issue will be addressed by the Improved APF with its revised attractive force equation.

3.2.2 Repulsive potential field and force

To ensure collision-free motion planning for the robot, a repulsive potential field and force are necessary, which "push" the robot away from obstacles. The equation for the repulsive potential field for a given obstacle is:

$$\Phi_{rep}(\boldsymbol{p}_{\hat{R}}) = \begin{cases} \frac{1}{2} * \eta * \left(\frac{1}{d(\hat{R},\hat{O})} - \frac{1}{d_{max}}\right)^2, & d(\hat{R},\hat{G}) \le d_{max} \\ 0, & d(\hat{R},\hat{G}) > d_{max} \end{cases} \tag{11}$$

where η is the positive constant characteristic of repulsion, $d(\hat{R},\hat{O})$ is the shortest distance between the robot and the given obstacle, where $\hat{R}$ represents the robot's position, $\hat{O}$ is the position of the obstacle, and d_{max} is the influence distance, which depends on the robot's maximum speed and deceleration capability, $\hat{G}$ is the target position. If the robot is outside the obstacle's influence distance, it is not considered hazardous for the robot.

The repulsive force vector field is obtained as the negative gradient of the scalar field:

$$\boldsymbol{F}_{rep}(\boldsymbol{p}_{\hat{R}}) = -grad\left[\Phi_{rep}(\boldsymbol{p}_{\hat{R}})\right] = \begin{cases} \eta * \left(\frac{1}{d(\hat{R},\hat{O})} - \frac{1}{d_{max}}\right) * \frac{1}{d(\hat{R},\hat{O})^2} * \frac{\partial d(\hat{R},\hat{O})}{\partial x}, & d(\hat{R},\hat{G}) \le d_{max} \\ 0, & d(\hat{R},\hat{G}) > d_{max} \end{cases} \tag{12}$$

3.2.3 Problems arising in traditional APF

The Traditional APF method is effective for solving real-time motion planning tasks, however there are cases where the robot gets "trapped" and cannot reach the goal position. This section presents these cases [20].

Gravity imbalance problem

The Gravity Imbalance Problem arises from the phenomenon observed in the attractive force formula of the Traditional APF (3.2.1). If the robot is too far from the goal and an obstacle is located between them, it is possible that the attractive force becomes so dominant in the resultant force that the robot collides with the obstacle. This is illustrated in Figure 5, where the small white circle represents the robot's next position, which in this case, would lead to a collision.

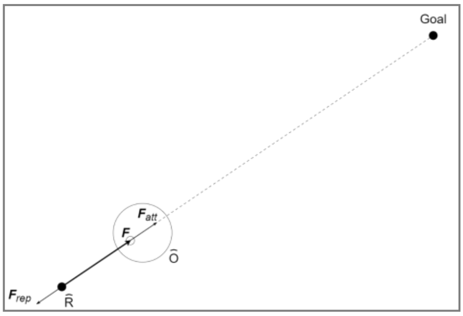

Figure 5: Gravity imbalance problem

Local minimum problem

If the robot reaches a position during its movement - different from the goal - from which it cannot escape, it has fallen into a local minimum. This can occur when the resultant repulsive forces from obstacles and the attractive force vector are collinear, meaning they are parallel, pointing in opposite directions, and have equal

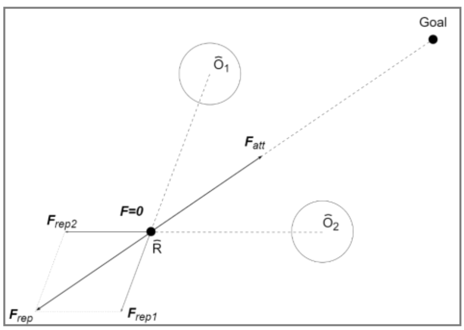

Figure 6: Local minimum problem

magnitudes (Figure 6). In this case, the attractive and repulsive forces cancel each other out, preventing the robot from moving. This problem is addressed by the APF method enhanced with Simulated Annealing.

Special local minimum problems that also occur in the APF method enhanced with Simulated Annealing:

- The robot cannot pass between closely placed obstacles (a narrow corridor).
- If the goal is very close to an obstacle, the robot may not be able to reach it.

Local oscillation problem

The local oscillation problem is similar to the local minimum problem and can be traced back to it with appropriately chosen ξ and η constants. Oscillation occurs when the resultant repulsive force vectors and the attractive force vector are parallel (and remain so), but their magnitudes are not equal. In this case, at one moment, the attractive force is stronger, pulling the robot closer to the obstacle. However, as the robot moves closer, the repulsive force increases and becomes dominant, pushing the robot back to its initial position (or sufficiently close to it). Figure 7 illustrates a robot trapped in oscillation, where $\hat{R}_n$ represents the robot's current position and $\hat{R}_{n+1}$ indicates its next position. The left and right cases occur alternately.

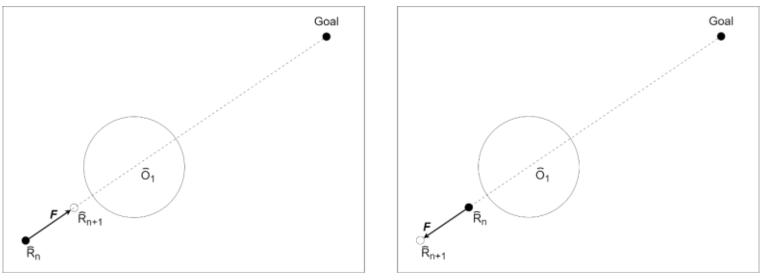

Figure 7: Local oscillation problem

3.3 Improved Artificial Potential Field

The Improved APF is introduced to eliminate the Gravity imbalance problem by modifying the attractive force formula, thereby providing a more stable foundation for further developments.

3.3.1 Attractive force

In several articles, the improved attractive force is defined as follows [19][21]:

$$\mathbf{F}_{att}(\boldsymbol{p}_{\hat{R}}) = \begin{cases} -\xi * (\boldsymbol{p}_{\hat{R}} - \boldsymbol{p}_{\hat{G}}), & d(\hat{R},\hat{G}) \leq d \\ -\xi * d * \frac{(\boldsymbol{p}_{\hat{R}} - \boldsymbol{p}_{\hat{G}})}{d(\hat{R},\hat{G})}, & d(\hat{R},\hat{G}) > d \end{cases} \tag{13}$$

where d represents the radius of an imaginary circle around the goal, beyond which the magnitude of the attractive force is no longer influenced by the distance between the robot and the goal.

The attractive force can be simplified, we used the following form in our algorithm:

$$\mathbf{F}_{att}(\boldsymbol{p}_{\hat{R}}) = -\xi * \frac{(\boldsymbol{p}_{\hat{R}} - \boldsymbol{p}_{\hat{G}})}{|\boldsymbol{p}_{\hat{R}} - \boldsymbol{p}_{\hat{G}}|} \tag{14}$$

thus, the magnitude of the attractive force is the same at every point in the field, and this can be represented by the variable ξ [20].

3.3.2 Repulsive force

In the case of the Traditional APF, the definition of the repulsive force is relatively complex, and determining d_{max} can also be problematic. Therefore, instead of the original expression (3.2.2), we introduce a simpler yet effective formula for the repulsive force.

When selecting the function for the repulsive force, among several options, we can choose the following equation:

$$\boldsymbol{F}_{rep}(\boldsymbol{p}_{\hat{R}}) = \frac{1}{e^{|\boldsymbol{p}_{\hat{R}} - \boldsymbol{p}_{\hat{O}}| - r_{\hat{O}}}} * \frac{\boldsymbol{p}_{\hat{R}} - \boldsymbol{p}_{\hat{O}}}{|\boldsymbol{p}_{\hat{R}} - \boldsymbol{p}_{\hat{O}}|} \tag{15}$$

function, where $|\boldsymbol{p}_{\hat{R}} - \boldsymbol{p}_{\hat{O}}|$ is the distance between the robot and the center of the obstacle, $r_{\hat{O}}$ is the radius of the given obstacle, and $\frac{\boldsymbol{p}_{\hat{R}} - \boldsymbol{p}_{\hat{O}}}{|\boldsymbol{p}_{\hat{R}} - \boldsymbol{p}_{\hat{O}}|}$ is the unit vector pointing from the obstacle toward the robot. The new formula has the appropriate slope, and instead of starting its effect at the boundary of the obstacle, it begins to exert its influence at a sufficient distance before the boundary, ensuring collision-free navigation for the robot. The new definition of the repulsive force was created by transforming the logistic function. The original form of the logistic function is:

$$f(x) = \frac{k}{1 + \alpha * e^{-b*(x - x_0)}} \tag{16}$$

where $k > 0$ is the smallest upper bound of the function along the y-axis, b determines the slope of the curve, and x_0 is the x value of the function's midpoint. α is a user-specific parameter that modifies the shape of the logistic function. In practical applications, $\alpha = 1$ is usually chosen to optimize the behavior of the system [22][23].

3.4 Improved Artificial Potential Field with Simulated Annealing

There are several methods to solve the local minimum problem outlined above. One of them is the Wallfollowing method, which attempts to escape from the minimum position by moving along the edge of a nearby obstacle until it exits the area [24], [30]. Another creative solution, the Virtual Obstacle Concept, creates a virtual obstacle – one that doesn't actually exist in reality – at the local minimum position, effectively pushing the robot out of the "trap" [31].

Among the various options, we used the Improved APF enhanced with Simulated Annealing (SA) to solve the local minimum problem. The main role of the following section is to present this method.

3.4.1 Simulated annealing

Simulated Annealing (SA) is a widely used metaheuristic method for solving optimization problems. The SA algorithm was first introduced by S. Kirkpatrick in 1983 [32], though the foundations of the method can be traced back to 1953 [33]. There are also other chemical reactions where motion planning is used [34].

The method of simulated annealing is based on the analogy of the cooling process known in metallurgy, during which a material is slowly cooled to reduce the number of dislocations and, consequently, the internal stresses, making it easier to process. As the temperature decreases, the mobility of the molecules also diminishes, and in the process, they arrange themselves into the lowest energy state, i.e., a crystal lattice [35].

Simulated Annealing (SA) searches for a global optimum in a space defined by a scalar-valued function by iteratively sampling nearby points. Better solutions are accepted directly, while worse ones are accepted with a decreasing probability, enabling escape from local minima. This probability diminishes as the algorithm's temperature parameter gradually decreases, balancing exploration and convergence [36-[38].

3.4.2 Application of Simulated Annealing with the Improved APF Method

The algorithm can begin when it is detected that the robot has entered a local minimum or is experiencing oscillation. The former can be detected by examining the distance between the robot's previous and current positions. If this distance is close to zero, the robot is almost not moving, indicating that it has entered a local minimum. The oscillation problem can be detected by examining the velocity vectors required to reach the previous and current positions. The robot will begin to oscillate if the two velocity vectors are parallel and point in opposite directions.

When setting the initial conditions, it is worth considering that starting from a higher temperature and using a slower cooling rate increases the likelihood of the algorithm finding a solution, but this also results in a longer runtime. To determine the cost of a given point, we have introduced the following function:

$$C(\boldsymbol{p}) = \left| \left| \mathbf{F}_{avg} \right| - \left| \boldsymbol{F}(\boldsymbol{p}) \right| \right| \tag{17}$$

where $\left|\mathbf{F}_{avg}\right|$ is the average magnitude (length) of the force vectors occurring at every point in the workspace, and $|\boldsymbol{F}(\boldsymbol{p})|$ is the magnitude of the force acting at a given $\boldsymbol{p}$ point.

The algorithm can accept a point with a higher cost as the next point, and the probability of this event can be expressed as follows:

$$P(\boldsymbol{p}_i) = \frac{1}{1+e^{\frac{C(\boldsymbol{p}_i)-C(\boldsymbol{p}_{i-1})}{T}}} \tag{18}$$

where $C(\boldsymbol{p}_i)$ is the cost of the possible new position, $C(\boldsymbol{p}_{i-1})$ is the cost of the previous position, and T is the current temperature. This probability is only examined if $C(\boldsymbol{p}_i) > C(\boldsymbol{p}_{i-1})$, and furthermore $T > 0$, so $0 < P(\boldsymbol{p}_i) \leq \frac{1}{2}$ [38]. This stochastic acceptance allows the robot to effectively "climb out" of "traps" (local minima) and continue its search toward the global optimum. The reduction of the temperature value is managed as follows:

$$T_i = \alpha * T_{i-1} \tag{19}$$

where T_i is the new temperature, T_{i-1} is the previous temperature value, and α defines the cooling rate. The value of α should be chosen such that the rate of temperature decrease is neither too fast nor too slow. [24]

The introduced algorithm will revert to the standard APF method from the simulated annealing algorithm when $C(\boldsymbol{p}_i) < \varepsilon$ come true, where ε is a constant that indicates how close $C(\boldsymbol{p}_i)$ needs to approach the ideal state. The optimum is represented by $C(\boldsymbol{p}_i) = 0$.

4 Introduction of the IAPF-MVVO Algorithm

The IAPF-MVVO algorithm combines the IAPF and MVVO methods, and its flowchart is shown in Figure 8. The algorithm initially uses IAPF, as it is a simpler method that requires less computational capacity. In this phase, the robot calculates the repulsive and attractive forces acting on it, based on which it can easily determine the resultant force. The robot's position is then updated, and we check if the goal has been reached, if a collision with an obstacle occurred, and if the robot has entered a local minimum.

If the robot is "trapped," we can assume that it is surrounded by a complex obstacle layout, from which it would be difficult – or impossible – to escape using just the simple IAPF or IAPF-SA. In this case, the algorithm switches to the MVVO method, which is particularly efficient at maneuvering among multiple obstacles. The necessary variables for the method are then set, and the Velocity Obstacle (VO) for the robot-obstacle pairs is calculated. Following this, the Reachable Velocity (RV) and Reachable Avoidance Velocity (RAV) are determined, and the robot's velocity vector is set.

The overall design and tuning of the proposed motion planning strategy were strictly guided by a set of essential performance specifications. The primary criteria for successful navigation were guaranteeing collision-free movement and ensuring a stuck-free execution, meaning the robot must be able to complete its entire motion toward the target without becoming permanently trapped in a local minimum. To fulfil these specifications, the hybrid strategy was designed to intervene and switch to the MVVO method precisely when the simple IAPF method fails the stuck-free criterion.

Furthermore, while trajectory planning and tracking control challenges in autonomous vehicles can be systematically specified using appropriate optimization problems to establish useful benchmarks [25], the proposed IAPF-MVVO approach addresses these performance specifications deterministically. By structurally combining reactive methods, it eliminates the need for solving complex numerical optimization models during runtime, thereby significantly minimizing computational overhead.

In the following section, we will compare this approach with the IAPF and IAPF-SA algorithms. Figure 8: The flowchart of the IAPF-MVVO algorithm

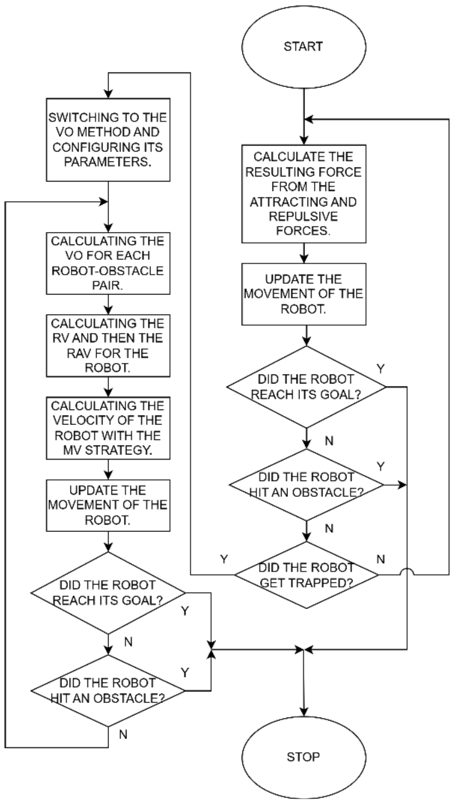

Figure 8: The flowchart of the IAPF-MVVO algorithm

5 Comparison of IAPF, IAPF-SA and IAPF-MVVO approaches, using simulations

In order to understand the efficiency and behavior of motion planning algorithms, it is essential to systematically compare the approaches in different simulation environments. In this chapter, we compare the various versions of the Improved Artificial Potential Field algorithm with the newly introduced Improved Artificial Potential Field – Maximum Velocity Obstacle algorithm. The aim of the analysis is to identify the strengths and limitations of the approaches. During the simulations, both static and dynamic obstacles are placed in the workspace.

The robot's physical limitations enforce a strict maximum velocity limit on its reachable velocities (Section 2.3). The trajectories are directly determined by the potential forces from Section 3 (Eq. 14, 15) during normal navigation, and by the IAPF-MVVO switching logic (Section 4) using the velocity obstacle formulations whenever a local minimum trap is detected.

The specific parameters of the path planning algorithms (such as the attractive and repulsive field constants, or the cooling rate for simulated annealing) were determined empirically. Through repeated complex simulation trials, these values were tuned to achieve optimal performance specifications, effectively balancing the path length and the elapsed time required to reach the goal.

Beyond empirical parameter tuning, the broader literature emphasizes several advanced methodologies that produce exceptional control and performance results in robotics. For instance, fuzzy logic control has proven highly effective in handling system uncertainties, demonstrating success in specialized fields such as telesurgical applications [39], as well as in computing shortest paths for mobile robots navigating static and dynamic indoor environments [40]. Furthermore, optimization-driven control designs show significant potential in managing complex, nonlinear mechanisms, such as two-wheeled unstable transporter systems [41]. While the current study relies on empirical settings for the IAPF-MVVO algorithm, integrating such fuzzy logic or optimization-driven frameworks represents a promising direction for future parameter synthesis and robust robotic control.

5.1 Two static obstacles

In the first test case, we used a layout where the robot is guaranteed to fall into a local minimum, allowing us to observe the different trajectories taken by the IAPF-SA and IAPF-MVVO algorithms after getting "trapped."

In Figure 9, the red circles represent obstacles, the blue circle indicates the goal, and the green points show the movement of the robot using the IAPF, the cyan points with SA, and the pink points with MVVO. The figures show that with the simple IAPF, the robot gets stuck and cannot move forward. In contrast, both the IAPF-SA and IAPF-MVVO reach the goal, but the latter algorithm achieves this in about one-third of the time compared to the former, providing a better solution in the path planning task. This is because the IAPF-MVVO algorithm allows the robot to take a much shorter path to the goal. The last figure illustrates the distance between the robot and the nearest obstacle. The IAPF-MVVO algorithm enables the robot to approach the given obstacle the closest.

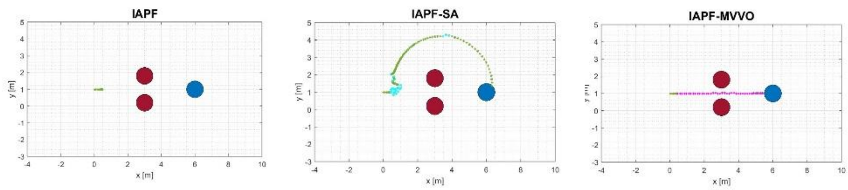

Figure 9: Simulation with two static obstacles

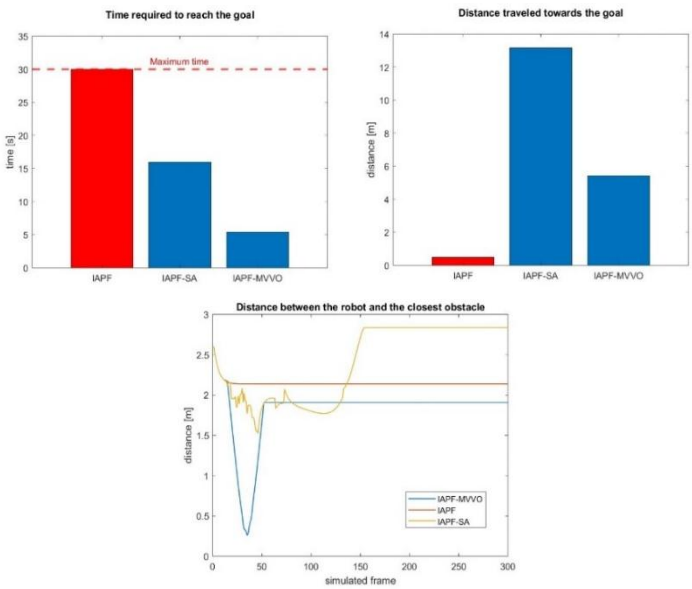

Figure 10: The obtained data; time required to reach the goal (left), distance travelled towards the goal (right), distance between the robot and the closest obstacle (in the middle). If the bar is red, the robot has failed to reach the goal.

5.2 Two static and one moving obstacle

In this arrangement, the robot does not reach a local minimum, as it can still reach the goal using the IAPF method. However, during its movement, it slows down so much at one point that both the IAPF-SA and IAPF-MVVO algorithms classify it as stuck, and thus, the SA and MVVO methods are applied during its motion. In this case, the IAPF-SA trajectory was the longest, and therefore, it reached the goal the slowest. The simple IAPF and IAPF-MVVO performed similarly.

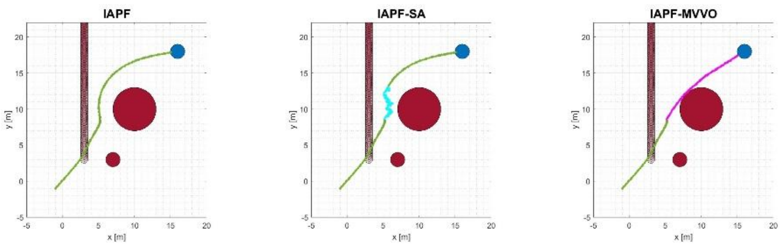

Figure 11: Simulation with two static and one moving obstacle

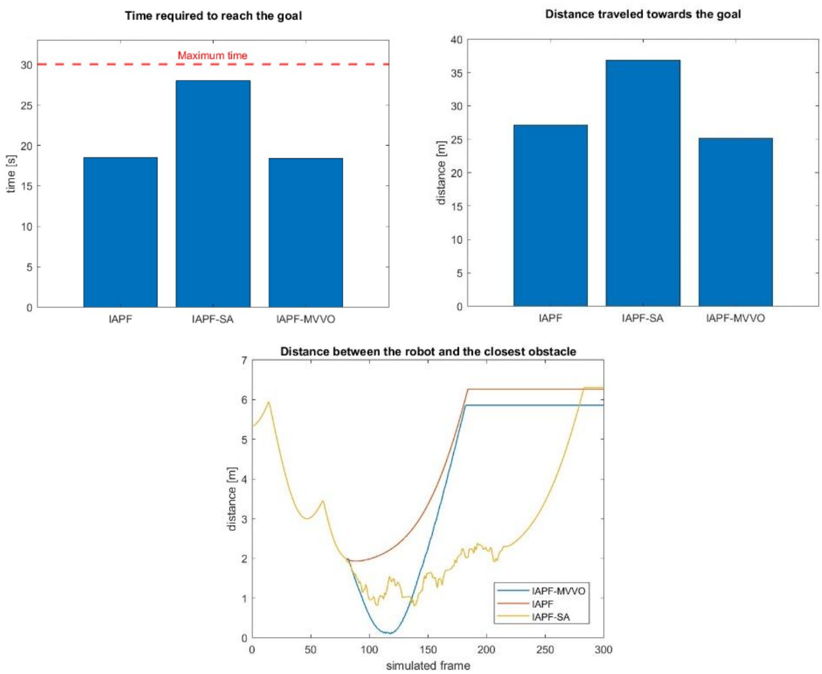

Figure 12: The obtained data; time required to reach the goal (left), distance travelled towards the goal (right), Distance between the robot and the closest obstacle (in the middle). If the bar is red, the robot has failed to reach the goal.

6 Conclusions

This article dealt with the collision-free motion planning problem, for mobile robots, with particular attention to reactive algorithms. Two main methods, the Velocity Obstacle and the Improved Artificial Potential Field algorithms, were presented. Using the combination of these methods, we introduced the IAPF-MVVO algorithm. During the research, the effectiveness of the motion planning algorithms was tested under various obstacle arrangements, using both static and dynamic obstacles.

For a more comprehensive comparison, not only the trajectories of the algorithms were presented, but also, the path length travelled by the robot was shown. Additionally, the time taken to reach the goal, and the distance between the robot and the nearest obstacle were also presented.

Based on the results, the IAPF-MVVO algorithm proved better performance, especially in complex obstacle arrangements.

Future research directions include applying and further developing the proposed algorithm for various robotic applications.

Acknowledgement

Supported by the EKÖP-26-4-II University Research Scholarship Program (EKÖP) of the Ministry of Culture and Innovation from the source of the National Research, Development and Innovation Fund.

References

[1] Telex: Self-driving cargo delivery robots will be tested in Debrecen, https://telex.hu/techtud/2022/01/19/onvezeto-aruszallito-robot-debrecen-teszteles-katalan muszaki-egyetem (2024. May)

[2] YU, Qiang; ZHOU, Jun. A review of global and local path planning algorithms for mobile robots. In: *2024 8th International conference on robotics, control and automation (ICRCA)*. IEEE, 2024. p. 84-90.

[3] KARADENIZ, Ahmet Mehmet; HAJDU, Csaba; KOCZY, Laszlo T. Mobile robot environment representation through fuzzy signatures-integrated quadtrees. *Rom. J. Inf. Sci. Technol*, 2025, 28: 103-116.

[4] HORVATH, Erno; POZNA, Claudiu; PRECUP, Radu-Emil. Robot coverage path planning based on iterative structured orientation. *Acta Polytechnica Hungarica*, 2018, 15.2: 231-249.

[5] The RRT Page: https://lavalle.pl/rrt/ (2025. May)

[6] KUFFNER, James J.; LAVALLE, Steven M. RRT-connect: An efficient approach to single-query path planning. In: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065). IEEE, 2000. p. 995-1001.

[7] DONG, Yiqun; CAMCI, Efe; KAYACAN, Erdal. Faster RRT-based nonholonomic path planning in 2D building environments using skeleton-constrained path biasing. *Journal of Intelligent & Robotic Systems*, 2018, 89.3: 387-401.

[8] KAVRAKI, Lydia E.; KOLOUNTZAKIS, Mihail N.; LATOMBE, J.-C. Analysis of probabilistic roadmaps for path planning. *IEEE Transactions on Robotics and automation*, 1998, 14.1: 166-171.

[9] LIU, Chang, et al. Path planning for autonomous vehicles using model predictive control. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017. p. 174-179.

[10] FOX, Dieter; BURGARD, Wolfram; THRUN, Sebastian. The dynamic window approach to collision avoidance. IEEE Robotics & Automation Magazine, 1997, 4.1: 23-33.

[11] POZNA, Claudiu; PRECUP, Radu-Emil; FÖLDESI, Péter. A novel pose estimation algorithm for robotic navigation. *Robotics and Autonomous Systems*, 2015, 63: 10-21.

[12] STEFANONI, Massimo, et al. A Comparison of Neural Networks and Fuzzy Inference Systems for the Identification of Magnetic Disturbances in Mobile Robot Localization. *Acta Polytechnica Hungarica*, 2025, 22.1: 239-264.

[13] FIORINI, *Paolo*; SHILLER, Zvi. Motion planning in dynamic environments using velocity obstacles. The international journal of robotics research, 1998, 17.7: 760-772.

[14] O'ROURKE, Joseph; BADLER, Norman. Decomposition of three-dimensional objects into spheres. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1979, 3: 295-305.

[15] GYENES, Zoltán; SZÁDECZKI-KARDOS, Emese Gincsainé. Novel motion planning method for mobile robots using velocity obstacle. *Acta Polytechnica Hungarica*, 2020, 9: 221-240.

[16] Wikipedia: Minkowski addition, https://en.wikipedia.org/wiki/Minkowski_addition (2024. May)

[17] KHATIB, Oussama. Real-time obstacle avoidance for manipulators and mobile robots. The international journal of robotics research, 1986, 5.1: 90-98.

[18] Dr. Zombory László: Electromagnetic fields. Budapest, Műszaki Kiadó, 2008.

[19] LI, Guanghui, et al. Effective improved artificial potential field-based regression search method for autonomous mobile robot path planning. International Journal of Mechatronics and Automation, 2013, 3.3: 141-170.

[20] QIXIN, Cao; YANWEN, Huang; JINGLIANG, Zhou. An evolutionary artificial potential field algorithm for dynamic path planning of mobile robot. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2006. p. 3331-3336.

[21] HE, Naifeng, et al. Dynamic path planning of mobile robot based on artificial potential field. In: 2020 International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI). IEEE, 2020. p. 259-264.

[22] Wikipedia: Logistic function, https://en.wikipedia.org/wiki/Logistic_function (2024. november)

[23] LÁSZLÓ, Hunyadi. The logistic function and the logistic distribution. Statisztikai szemle, 2004, 10.11: 991-1011.

[24] ZHU, Qidan; YAN, Yongjie; XING, Zhuoyi. Robot path planning based on artificial potential field approach with simulated annealing. In: Sixth international conference on intelligent systems design and applications. IEEE, 2006. p. 622-627.

[25] PRECUP, Radu-Emil, et al. Problem setting for trajectory planning and cruise control of a connected autonomous electric bus in intersection scenarios with human-driven vehicles to optimize energy, comfort and tracking. *Rom. J. Inf. Sci. Technol*, 2025, 28: 299-312.

[26] VAŠČÁK, Ján. Navigation of mobile robots using potential fields and computational intelligence means. *Acta Polytechnica Hungarica*, 2007, 4.1: 63-74.

[27] KOVÁCS, Bence, et al. A novel potential field method for path planning of mobile robots by adapting animal motion attributes. Robotics and Autonomous Systems, 2016, 82: 24-34.

[28] ROSTAMI, Seyyed Mohammad Hosseini, et al. Obstacle avoidance of mobile robots using modified artificial potential field algorithm. EURASIP Journal on Wireless Communications and Networking, 2019, 2019.1: 1-19.

[29] SIDDIQUI, Rymsha. Path planning using potential field algorithm. Medium. July, 2018., https://medium.com/@rymshasiddiqui/path-planning-using-potential-field-algorithm a30ad12bdb08 (2024. November)

[30] BORENSTEIN, Johann; KOREN, Yoram. Real-time obstacle avoidance for fast mobile robots. IEEE Transactions on systems, Man, and Cybernetics, 1989, 19.5: 1179-1187.

[31] PARK, Min Gyu; LEE, Min Cheol. A new technique to escape local minimum in artificial potential field based path planning. KSME international journal, 2003, 17: 1876-1885.

[32] KIRKPATRICK, Scott; GELATT JR, C. Daniel; VECCHI, Mario P. Optimization by simulated annealing. science, 1983, 220.4598: 671-680.

[33] METROPOLIS, Nicholas, et al. Equation of state calculations by fast computing machines. The journal of chemical physics, 1953, 21.6: 1087-1092.

[34] VIRÁGH, Eszter; DREXLER, Dániel András; KISS, Bálint. Motion planning and modeling for isothermal parallel chemical reactions. In: *2023 IEEE 17th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. IEEE, 2023. p. 91-96.

[35] ODZIEMCZYK, Waldemar. Application of simulated annealing algorithm for 3D coordinate transformation problem solution. Open Geosciences, 2020, 12.1: 491-502.

[36] RUTENBAR, Rob A. Simulated annealing algorithms: An overview. *IEEE Circuits and Devices magazine*, 2002, 5.1: 19-26.

[37] HENDERSON, Darrall; JACOBSON, Sheldon H.; JOHNSON, Alan W. The theory and practice of simulated annealing. In: *Handbook of metaheuristics*. Boston, MA: Springer US, 2003. p. 287-319.

[38] HABLIK, Róbert; SZABÓ, Martin. Application of Simulated Annealing algorithm for solving flexible job shop scheduling problems. Multidisciplinary Sciences, 2020, 10.4: 404-410.

[39] PRECUP, Radu-Emil, et al. Fuzzy control solution for telesurgical applications. *Applied and Computational Mathematics*, 2012, 11.3: 378-397.

[40] HENTOUT, Abdelfetah; MAOUDJ, Abderraouf; KOUIDER, Ahmed. Shortest path planning and efficient fuzzy logic control of mobile robots in indoor static and dynamic environments. *Sci. Technol*, 2024, 27.1: 21-36.

[41] SZEDLAK-STINEAN, Alexandra-Iulia; PRECUP, Radu-Emil; ARI, Zeynep-Melisa. Optimization-driven control design for the nonlinear two-wheeled unstable transporter system. *Proceedings of the Romanian Academy, Series A: Mathematics, Physics, Technical Sciences, Information Science*, 2025, 6.3: 323-330.