

**I. A. Berizka, I. D. Karbovnyk***Ivan Franko Lviv National University, Lviv, Ukraine*

SIMULATION-BASED EVALUATION OF HYPERBOLIC SECANT POTENTIAL FIELD FOR REAL-TIME OBSTACLE AVOIDANCE

Obstacle avoidance is a fundamental capability for autonomous mobile robots, ensuring safe navigation in dynamic and unstructured environments. This paper presents a novel approach to real-time obstacle avoidance based on an Artificial Potential Field Method (APFM) utilizing a hyperbolic secant function. A mathematical formulation of the proposed model is developed and analyzed. To validate the approach, a simulation framework was implemented using ROS 2, the Gazebo simulator, and the TurtleBot3 Burger platform. Extensive simulations were conducted, generating LiDAR sample plots alongside corresponding repulsive, attractive, and total potential field distributions, demonstrating the correctness and effectiveness of the proposed method. Additionally, RViz visualization confirmed the smoothness of the robot's path and continuous heading adjustments over 28 navigation steps. To assess computational efficiency, execution times for evaluating Gaussian and hyperbolic secant functions were measured using C++ implementations with varying compiler optimization flags. The results indicate that the computational cost of the hyperbolic secant function is approximately 2–3 % lower than that of the Gaussian, a negligible difference in practice. The findings support the suitability of the hyperbolic secant-based APFM for real-time obstacle avoidance in robotic applications. The hyperbolic secant function offers a sharper decay near the origin compared to the Gaussian distribution, resulting in stronger immediate repulsive forces when the robot is close to an obstacle and rapidly diminishing influence at greater distances. This property provides improved responsiveness and local obstacle avoidance without introducing excessive long-range effects that could unnecessarily distort the global path. Additionally, the hyperbolic secant's symmetric, heavy-tailed nature maintains smoothness in the potential field, ensuring stable and predictable robot motion while enabling faster, more efficient trajectory adjustments in dynamic environments.

Keywords: artificial potential field method, cyber-physical system, edge computing, information technologies, IoT concepts, obstacle avoidance, robotics.

Introduction

In the field of robotics, autonomous mobile robots represent a major area of research and development. These sophisticated systems are engineered to navigate independently and to make context-specific decisions in real time. This capability allows them to operate without human intervention, dynamically adapting to their environments based on continuous sensory input. Autonomous mobile robots are utilized in a broad range of applications. Examples include service robots in hospitality settings, such as waiter robots that deliver food and beverages, as well as transport robots employed in industrial environments to efficiently move goods. A particularly notable example is autonomous vehicles, commonly known as self-driving cars. These systems integrate advanced sensor technologies and computational algorithms to interpret their surroundings, enabling them to navigate complex traffic environments and reach their destinations without human control.

Such examples highlight the extensive applicability of autonomous mobile robots and underscore their transformative potential across multiple sectors, including hospitality, logistics, and the automotive industry. The ongoing

development and refinement of these systems remain a central focus of robotics research, with continuous efforts aimed at enhancing their capabilities and expanding their range of applications.

One of the core software components in autonomous mobile robots is the set of algorithms dedicated to path planning and obstacle avoidance. These algorithms are fundamental to enabling critical functionalities such as autonomous parking, evasive maneuvers during emergency situations, and, ultimately, full autonomy in navigation and control.

Path planning is generally classified into two main categories: global and local. Global path planning relies on data from Geographic Information Systems (GIS) in combination with global localization methods. This approach necessitates that the robot possesses a comprehensive, large-scale understanding of its environment, enabling navigation across extended distances, such as traversing urban areas or intercity routes.

In contrast, local path planning requires only the robot's relative position and real-time perception of obstacles within its immediate surroundings. This type of planning focuses on short-range navigation and dynamic interaction with the

environment, such as avoiding pedestrians on a sidewalk or maneuvering around other vehicles in traffic. Local path planning is essential for ensuring safe and adaptive behavior in unpredictable and rapidly changing environments.

A diverse array of algorithms has been developed to address the challenges associated with both global and local path planning. Each algorithm offers specific advantages and exhibits particular limitations, and the choice of an appropriate algorithm can significantly impact the efficiency, reliability, and safety of autonomous navigation. Comprehensive surveys detailing these path planning techniques, their underlying methodologies, and their application contexts are available in the existing literature [1, 2, 3, 4].

Obstacle detection and avoidance are critical components of local path planning algorithms, playing an essential role in ensuring the safety of both the autonomous system and its surrounding environment. This field has been the subject of extensive research over several decades, resulting in the development of numerous methodological approaches. Many of these techniques have proven effective and have been successfully deployed in real-world applications.

Effective collision avoidance requires that a robot not only detect obstacles but also dynamically recalculate its path and adjust its trajectory in real time. Real-time responsiveness is crucial for navigating safely and efficiently through complex and dynamic environments.

The process of obstacle avoidance begins with the robot detecting potential obstacles using its onboard sensors. Upon identification of an obstacle, the system must promptly generate an alternative trajectory that enables safe passage around the object. This new path must be computed with minimal latency to allow real-time adjustment of the robot's motion, thereby preventing collisions and maintaining a smooth navigation trajectory.

Moreover, the robot must be capable of adapting to dynamic changes in its environment. For instance, if a new obstacle suddenly appears along its path, the robot must rapidly detect the obstruction, compute an alternative route, and adjust its trajectory in real time to ensure safe and continuous navigation.

The fundamental concept of obstacle avoidance involves moving toward the target point while dynamically adjusting the trajectory whenever obstacles are detected along the current path. A graphical representation of the path planning and obstacle avoidance algorithm is presented in Fig. 1.

The sequence of obstacle detection and trajectory adjustment constitutes an iterative process. The robot continuously monitors its environment for potential obstacles and performs real-time path corrections as necessary until it successfully reaches the designated target location. This iterative procedure highlights the dynamic nature of autonomous navigation and emphasizes the critical importance of robust obstacle detection and avoidance mechanisms in ensuring the safe and efficient operation of mobile robotic systems.

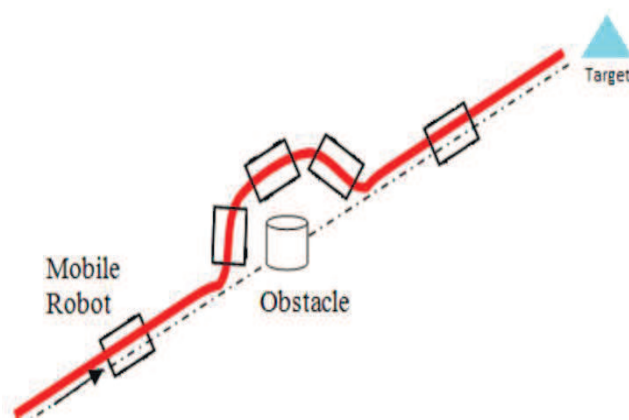


Fig. 1. Obstacle avoidance problem consists of two subtasks: computing motion control that avoids collisions with the obstacles gathered by the sensors and driving the robot towards target location

Object of research: The study focuses on real-time obstacle avoidance algorithms for autonomous mobile robots.

Subject of research: It specifically examines the Hyperbolic Secant-based Artificial Potential Field Method (APFM) and its simulation in ROS 2/Gazebo.

Purpose of research: The goal is to develop a faster, more efficient path-planning solution while creating a standardized testing framework for future research.

Analysis of recent research and publications. The Artificial Potential Field Method (APFM) is a well-established technique in robotics, particularly in the domains of path planning and obstacle avoidance. This approach was first introduced by Khatib in 1984 [5].

In the APF method, a virtual potential field is constructed wherein the target location generates an attractive force, while obstacles in the environment produce repulsive forces. The robot, or autonomous agent, is guided by the resultant force derived from the combination of these influences. It moves toward the target while simultaneously avoiding obstacles under the effect of these virtual forces [5]. The mathematical representations of these forces are provided in equations (1)–(3).

$$\vec{f}_{total} = \vec{f}_{attr} + \vec{f}_{rep} \quad (1)$$

$$\vec{f}_{attr} = k_{attr} * \left(\frac{\vec{r}_{goal} - \vec{r}}{|\vec{r}_{goal} - \vec{r}|} \right) \quad (2)$$

$$\vec{f}_{rep} = \begin{cases} -k_{rep} * \sum_{i=1}^n \left(\frac{1}{d_i} - \frac{1}{d_{max}} \right) * \vec{s}_i, & \text{if } d_i < d_{max} \\ 0 & \end{cases} \quad (3)$$

Despite its advantages, the traditional Artificial Potential Field (APF) method exhibits several limitations. One notable issue is the occurrence of local minima, where the robot becomes trapped in a position that is not the target due to the balancing of attractive and repulsive forces. Additionally, the

classical APF approach may result in unreachable targets and the generation of inefficient paths [6].

To address these challenges, several enhanced versions of the APF method have been proposed. One such modification, presented in [7], incorporates probabilistic elements into the classical framework. The ODG-PF method was specifically developed to improve obstacle detection and calculate the likelihood of collision with detected obstacles. This approach introduces novel formulations for both the attractive and repulsive fields, as well as a new strategy for direction determination.

The authors conducted extensive simulations and experimental validations, comparing the ODG-PF method against other potential field-based obstacle avoidance techniques. The results demonstrated that the ODG-PF approach outperformed alternative methods in most evaluated scenarios.

Within this algorithm, a key component is the computation of a repulsive field – a vector field representing the force exerted to push the robot away from nearby obstacles.

In [8], a more detailed review of the ODG-PF method was provided, along with proposals for further research in this direction. In particular, a mathematical model of the Artificial Potential Field Method (APFM) was introduced, utilizing the Laplace function to represent the repulsive field.

In this paper, we propose the use of the hyperbolic secant function to model the repulsive force within the artificial potential field framework.

$$f_k(\theta_i) = A_k \cdot \operatorname{sech}\left(\frac{\pi}{2} * \frac{\theta_k - \theta_i}{\sigma_k}\right), \quad (4)$$

where θ_k corresponds to the central angle of the k_{th} obstacle, σ_k is half of angle occupied by the k_{th} obstacle.

In the context of a sensor with a resolution of 1 degree, the index can be interpreted as a discrete representation of angular measurements within the 0 to 360-degree range. Each Hyperbolic Secant function, corresponding to a specific angle, contributes to the construction of the overall repulsive field, thereby forming a vector field that actively directs the robot away from detected obstacles.

A key parameter in this formulation is the coefficient A_k , which is carefully adjusted to ensure that the Hyperbolic Secant function sufficiently covers the extent of the obstacle. This adjustment guarantees an accurate representation of the repulsive forces in the proximity of obstacles. The derivation and appropriate selection of this coefficient represent a critical component of the algorithm and merit detailed examination.

$$A_k = \frac{\bar{d}_k}{\operatorname{sech}\left(\frac{\pi}{2}\right)}, \quad (5)$$

where $\bar{d}_k = d_{\max} - d_k$, $d_{\max}$ is sensor range distance.

Given the necessity to account for the influence of each individual obstacle, the overall repulsive field is computed

as the superposition of all repulsive fields generated by the obstacles. Consequently, the resulting function is expressed as a dependence on the angular variable θ .

$$f_{rep}(\theta_i) = \sum_{k=i}^n A_k \cdot \operatorname{sech}\left(\frac{\pi}{2} * \frac{\theta_k - \theta_i}{\sigma_k}\right) \quad (6)$$

The next stage involves the computation of the attractive field, as defined by equation (7). This field represents the force that draws the robot toward the desired direction of motion. When combined with the repulsive field, the attractive field shapes the robot's trajectory, ensuring obstacle avoidance while maintaining progress toward the target location.

$$f_{attr}(\theta_i) = \gamma |\theta_{goal} - \theta_i| \quad (7)$$

$$f_{total}(\theta_i) = f_{attr}(\theta_i) + f_{rep}(\theta_i) \quad (8)$$

$$\theta_{dir} = \arg \min(f_{total}) \quad (9)$$

Parameter γ is selected experimentally and is set to 6. Equation (8) represents total field produced by the system, and the safe direction of robot movement is determined using (9).

Evaluation framework. Numerous software platforms, often referred to as middleware, have been developed to introduce modular and adaptable features that simplify the construction of robotic systems. Over time, some of these middleware platforms have evolved into comprehensive ecosystems, offering a wealth of utilities, algorithms, and sample applications. Among them, the Robot Operating System (ROS 1) stands out for its profound impact on the robotics industry.

ROS 1 gained widespread popularity through the efforts of the robotics incubator Willow Garage. While the system achieved significant success in creating a high-quality and high-performance framework, certain aspects – such as security, network topology optimization, and system uptime – were not primary focuses. Despite these limitations, ROS 1 has left a considerable legacy across nearly every sector involving intelligent machines. Its commercial success was propelled by flagship projects that enabled autonomous navigation, simulation, visualization, control, and more. However, as commercial applications matured into full-scale products, the foundational limitations of ROS 1 as a research-oriented platform became increasingly apparent. The demand for improved security, greater reliability in heterogeneous environments, and robust support for large-scale embedded systems necessitated a new architectural approach. Consequently, many companies developed workarounds on top of ROS 1 to build production-ready applications.

Recognizing these limitations, the robotics community initiated the development of ROS 2. Built from the ground up, ROS 2 addresses critical requirements for modern robotic systems, including enhanced security, real-time communication, system reliability, and scalability across distributed and resource-constrained platforms. ROS 2 adopts the Data Distribution Service (DDS) middleware standard, providing high-performance and flexible communication essential for

complex robotic deployments [9, 10]. Its improved modularity, cross-platform support, and emphasis on production-grade robustness make ROS 2 the framework of choice for both academic research and industrial robotics applications.

The use of ROS, and particularly ROS 2, in the implementation of robotic systems offers numerous advantages. It accelerates the development process through a rich ecosystem of open-source tools, libraries, and algorithms. It also facilitates the integration of heterogeneous subsystems and ensures interoperability between components, which is crucial for building reliable and scalable robotic platforms. Furthermore, the active global ROS community ensures continuous support, innovation, and validation across a broad range of robotic domains.

The integration of ROS 2 as a unified middleware platform for deploying practical outcomes brings significant advancements towards sim-to-real experiences, enabling researchers to test their concepts beyond controlled laboratory environments. The mathematical model of the newly proposed Hyperbolic Secant Artificial Potential Field Method (APFM) was implemented as a shared C++ library, facilitating its seamless integration with other software components, including its incorporation into the ROS 2 ecosystem.

In order to preform simulation, we decided to choose Gazebo simulator. Gazebo represents an optimal simulation platform for integration with ROS 2 due to its high-fidelity dynamic modeling, accurate sensor emulation, and native compatibility with the ROS ecosystem. As the reference simulator for ROS-based applications, it enables physically realistic simulation of robotic systems, including multi-body dynamics, environmental interactions, and sensor data generation (e. g., depth cameras, LiDAR, inertial measurement units). The Gazebo-ROS 2 interface provides bi-directional communication through dedicated plugins, ensuring temporal synchronization between simulated and real-world control paradigms. This integration facilitates robust validation of perception, planning, and control algorithms prior to physical deployment, significantly reducing development cycles. Furthermore, Gazebo's support for complex scenarios – including multi-agent systems and dynamic environments – combined with ROS 2's distributed computational architecture, offers a scalable framework for advanced robotic research and system verification. More detailed review of Gazebo and its alternatives are presented in [11].

The next stage involved selecting a ROS-enabled wheeled mobile robot platform. We conducted an analysis of existing solutions and determined that the TurtleBot series is among the most used platforms for research in the domain of mobile wheeled robots. A study presented in [11] demonstrates that the TurtleBot series is utilized in approximately 20 % of related academic publications, confirming its suitability and convenience for scientific research. Furthermore, the TurtleBot platforms offer native integration with ROS 2 and Gazebo, significantly simplifying the setup process. Comprehensive documentation, simulation models,

and software packages are readily available through the official website and associated GitHub repositories, further supporting ease of deployment.

TurtleBot3, in particular, represents an optimal choice for academic and experimental purposes due to its modularity, affordability, and extensive community support. It features scalable hardware configurations, enabling researchers to customize the robot according to specific research needs. The platform is lightweight yet powerful, equipped with a suite of sensors necessary for implementing and testing autonomous navigation algorithms, including LIDAR, IMU, and wheel encoders. Its seamless compatibility with ROS 2 facilitates rapid development, testing, and integration of complex robotic applications, making TurtleBot3 an ideal candidate for validating the proposed obstacle avoidance algorithms within a realistic yet controllable experimental environment.

Research results and their discussion

To evaluate proposed mathematical model of Hyperbolic Secant APFM we used virtual room with obstacles which is presented on Fig. 2. Parameters and its values needed for Hyperbolic Secant APFM are presented in table 1.

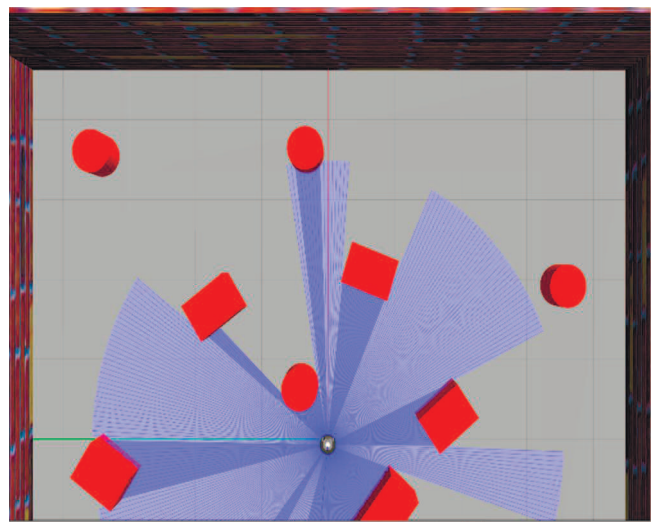


Fig. 2. Workplace configuration in Gazebo simulator /
Конфігурація робочого середовища
у симуляторі Gazebo

Fig. 2 presents the detailed workspace configuration utilized within the Gazebo simulation environment for evaluating the proposed mathematical model of the obstacle avoidance method. In this experimental setup, the TurtleBot3 Burger robot is initially positioned at the center of the room, visually represented as a small cylinder marked with a white dot. The physical parameters of the TurtleBot3 Burger, including dimensions, mass, and sensor specifications, were derived from the official documentation and incorporated into the simulation model to ensure accuracy and realism. The surrounding environment consists of a square room enclosed by walls and populated with multiple obstacles of varying geometric shapes, specifically cylindrical and

rectangular forms, thereby providing a diverse and challenging setting for testing obstacle avoidance performance.

Additionally, the blue lines radiating from the center of the robot represent the one-dimensional (1D) LiDAR path rays. The LiDAR sensor resolution is configured to 1° , ensuring high angular precision. In the present study, it is assumed that a single LiDAR scan corresponds to a full rotation of the sensor, thereby capturing a complete 360° view of the environment.

Tab. 1. Hyperbolic Secant APFM parameters and corresponding values

Parameter	VALUE
Threshold distance	1 m
Robot diameter	0.2 m
LiDAR max range	6 m
LiDAR resolution	$-179-180$, step 1°
Gamma	6
Goal direction	0°
Robot linear speed	0.1 m/s

Based on the specified configuration and parameter values, the robot is expected to move forward toward the wall while actively avoiding collisions with any obstacles encountered along its path. A single iteration of the robot's navigation algorithm is presented in the pseudocode depicted in Fig. 3.

Algorithm 1 General Robot Navigation Procedure Based on APFM

```

1: while the target location is not reached do
2:   Step 1: Environment Scanning
3:   Acquire 1D LiDAR sensor data
4:   Identify obstacles within the sensitivity zone
5:   Enlarge detected obstacles based on robot dimensions
6:   Step 2: Force Computation
7:   Compute repulsive forces using Eq. (6)
8:   Compute attractive force toward the target using Eq. (8)
9:   Determine the resultant force vector using Eq. (9)
10:  Step 3: Safe Direction Evaluation
11:  Calculate the safe movement angle according to Eq. (10)
12:  if the safe direction differs from the target direction then
13:    Rotate the robot to align with the computed safe angle
14:    Move forward for a predefined time interval (e.g., 1 second)
15:    Reorient the robot back toward the original target direction
16:  else
17:    Continue moving forward for the same time interval
18:  end if
19: end while

```

Fig. 3. Pseudocode of general robot navigation algorithm based APFM family algorithms

Next, we proceed to examine in detail the first three steps of the proposed algorithm. At the initial timestamp t_1 (i. e., the first iteration), the robot is positioned at the center of the room, as illustrated in Fig. 2. The corresponding 1D LiDAR scan data is presented in Fig. 4.

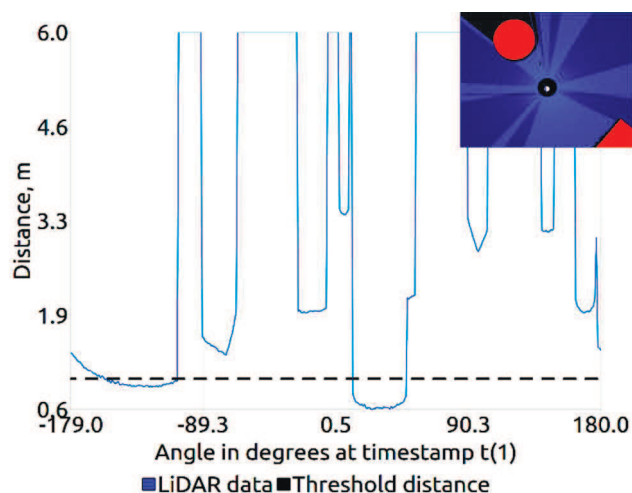


Fig. 4. 1D LiDAR scan at the timestamp t_1 .

Solid curve shows data from one complete LiDAR scan. Dashed line indicates threshold distance. Objects closer to the robot than threshold distance are marked as obstacles. Top-right corner picture depicts the actual position of the robot in simulator environment

Objects detected at distances smaller than this threshold are classified as obstacles. Analysis of the LiDAR scan reveals two distinct continuous regions falling below the threshold, located approximately at angular intervals of $[-170^\circ, -100^\circ]$ and $[+10^\circ, +45^\circ]$. Based on this observation, the Artificial Potential Field Method (APFM) is expected to recognize two obstacles within these angular sectors. The robot's actual position within the simulation environment is shown in the top-right inset of Fig. 4. It can be observed that the obstacles identified by the APFM, depicted in red, correspond accurately to the angular locations inferred from the LiDAR scan, confirming the correctness of the obstacle detection stage.

Fig. 5 illustrates the force distributions at an initial time step. The repulsive force, represented by blue scatter circles, exhibits two distinct peaks near -120° and $+20^\circ$, corresponding to the directions where obstacles are detected by the LiDAR sensor. The attractive force, shown as a dashed black line, follows a symmetric V-shaped profile with a minimum at 0° , promoting motion toward the target. The total force, indicated by green scatter triangles, combines the effects of the repulsive and attractive fields. The minimum of the total force is slightly shifted to approximately -15° , implying a necessary deviation from the direct path to ensure safe navigation around obstacles. The broader distribution of the repulsive peaks contributes to an increased safety margin by enlarging the obstacle avoidance zones. The broader shape of the repulsive force peaks contributes to an increased safety margin, effectively enlarging the obstacle avoidance zone and enhancing the overall robustness of the navigation strategy.

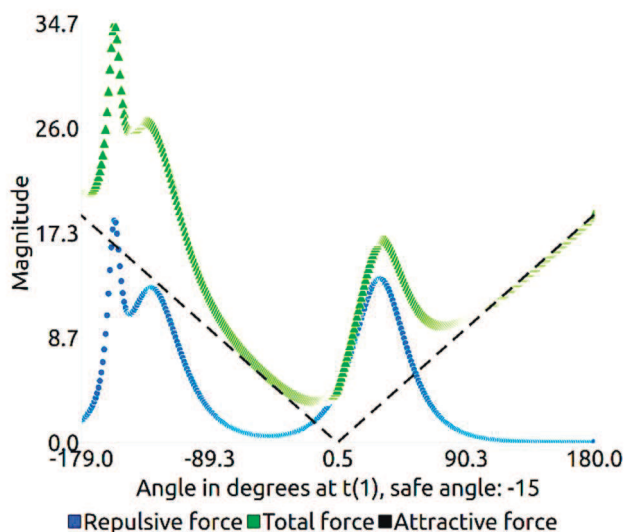


Fig. 5. Calculated repulsive, attractive and total forces at the timestamp t_1

As outlined in the navigation algorithm presented in Fig. 3, the robot initially performs a rotation toward the calculated safe directional angle. This adjustment ensures that its heading is reoriented to avoid the detected obstacles effectively. Subsequently, the robot advances by moving forward for a duration of one second, thereby making incremental progress toward the target location. After completing the forward movement, the robot executes a corrective rotation to realign its heading with the original target direction, thereby resuming its intended trajectory.

This sequence of operations is iterative, continuing systematically until the robot either successfully reaches the designated target location or encounters a local minimum, where further progress becomes infeasible due to environmental constraints.

Under the current experimental setup, the robot successfully navigated the environment and reached the target in 28 discrete steps. To further validate the reliability and performance of the proposed method, a detailed analysis of two more specific iterations is proposed. The first analysis will focus on an intermediate location between the starting point and the target destination, specifically at timestamp t_{16} . The second analysis will examine the final approach to the target location, recorded at timestamp t_{28} situated adjacent to a wall.

In both cases, a comprehensive evaluation of the forces acting on the robot during navigation will be performed. Finally, the overall trajectory of the robot, visualized using the RViz tool, will be thoroughly reviewed to assess the smoothness of the proposed obstacle avoidance approach.

Analyzing the data presented in Fig. 6, a single continuous region below the threshold line is observed, approximately spanning the angular range of angles $[-110^\circ, -45^\circ]$. Consequently, the APFM is expected to detect a single obstacle. Furthermore, it is evident that the position of the obstacle within the simulation environment corresponds precisely to the angular locations identified in the LiDAR scan.

Additionally, it is important to highlight that the detected obstacle is located in close proximity to the robot, at an approximate distance of 0.3 meters from the side of the TurtleBot3. This close range implies that, during the obstacle enlargement step, a wider peak will be generated in the force calculation process, significantly influencing the resulting repulsive force profile.

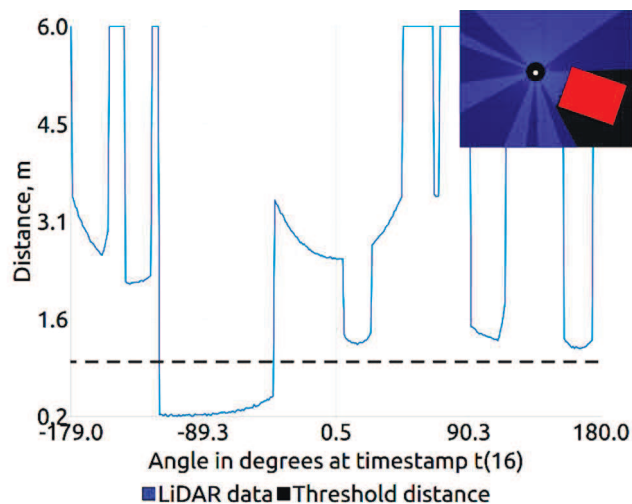


Fig. 6. 1D LiDAR scan at the timestamp t_{16} . Solid curve shows data from one complete LiDAR scan. Dashed line indicates threshold distance. Objects closer to the robot than threshold distance are marked as obstacles. Top-right corner picture depicts the actual position of the robot in simulator environment

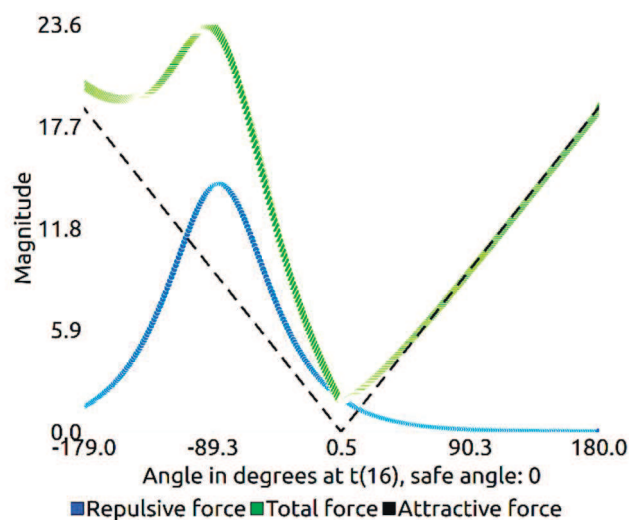


Fig. 7. Calculated repulsive, attractive and total forces at the timestamp t_{16}

Fig. 7 presents the force profiles at the later time instance. In this case, the repulsive force shows a single pronounced peak centered around -90° , indicating an isolated obstacle in that direction. The attractive force maintains its V-shaped configuration with its minimum at 0° , corresponding to the goal direction. The total force also attains its minimum at 0° ,

suggesting that no significant obstacle interferes with the direct path, and the robot can proceed straight toward the target without requiring a heading adjustment. This situation demonstrates the APFM's capacity to maintain goal-directed motion when environmental conditions permit.

Finally, Fig. 8 presents the LiDAR data sample captured during the robot's final step, at the point where it reached the wall.

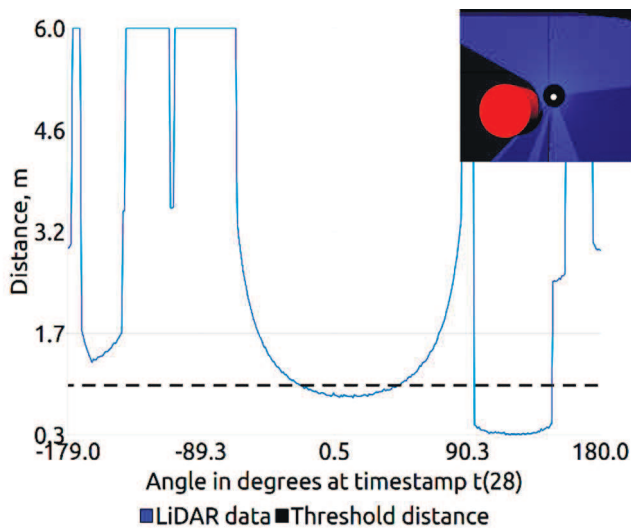


Fig. 8. 1D LiDAR scan at the timestamp t_{28} . Solid curve shows data from one complete LiDAR scan. Dashed line indicates threshold distance. Objects closer to the robot than threshold distance are marked as obstacles. Top-right corner picture depicts the actual position of the robot in simulator environment

Analyzing the data presented in Fig. 8, two continuous regions below the threshold line are observed, approximately within the angular ranges $[-10^\circ, 60^\circ]$ and $[110^\circ, 170^\circ]$. Accordingly, the APFM is expected to detect both obstacles. Upon further examination of the plot, it becomes evident that the first detected obstacle corresponds to a wall. This conclusion is drawn based on the geometric characteristics observed in the LiDAR scan, where the obstacle exhibits a smooth, circular-like shape – a signature feature typically associated with walls due to their uniform and continuous surfaces.

The second continuous region identified in the LiDAR data corresponds to another obstacle. Its angular location suggests that it is positioned slightly behind the robot. This inference is further supported by the visual representation of the simulation environment shown in the top-right corner of Fig. 8, where a wall is clearly positioned in front of the robot, and an additional obstacle is situated slightly to the rear.

Fig. 9 captures the force field structure at a subsequent stage. The repulsive force displays two significant peaks around 90° and 150° , marking the presence of multiple obstacles along the forward path. The attractive force retains its canonical V -shaped profile, while the total force reveals a minimum shifted to approximately -44° .

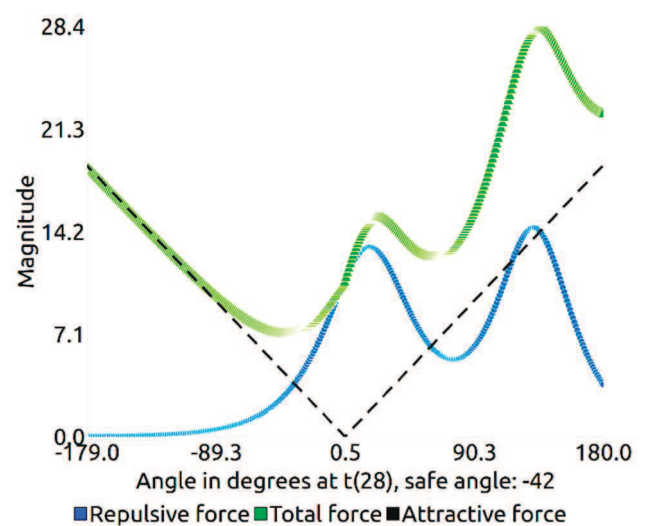


Fig. 9. Calculated repulsive, attractive and total forces at the timestamp t_{28}

This deviation indicates that the robot must adjust its heading significantly to the left to circumvent the obstacles safely. The resulting total force distribution highlights the dynamic balance between obstacle avoidance and goal attraction, enabling robust and adaptive navigation. However, since the robot has already reached the target location, this calculated safe direction is disregarded. This decision reflects the algorithm's design to prioritize the final destination once it is achieved, rather than continuing to adjust navigation based on force computations.

Fig. 10 presents a trajectory visualization generated in RViz, illustrating the robot's navigation path.

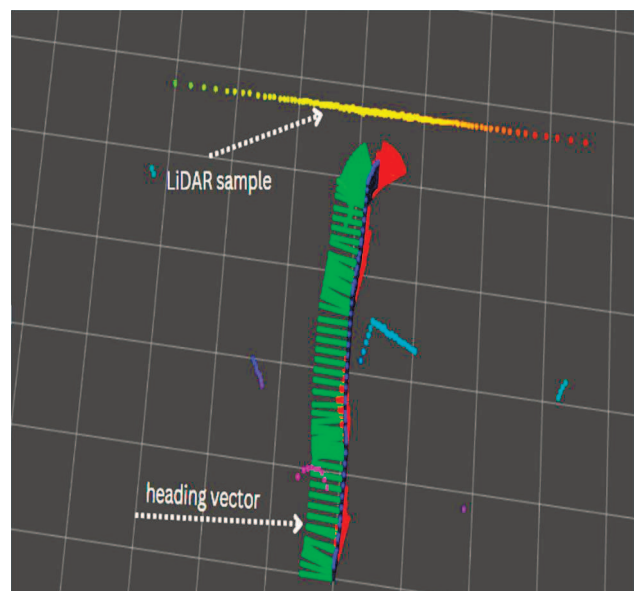


Fig. 10. Traversed path visualization in RViz

The RViz visualization demonstrates the operational efficacy of the implemented navigation system, showing three key elements of the obstacle avoidance process. Green vectors indicate real-time directional corrections where the calculated safe navigation angle diverges from the target

trajectory due to detected obstacles, reflecting the system's dynamic response to environmental constraints. Red markers represent LiDAR-measured obstacle positions, forming a point cloud that quantifies the robot's perceptual field and correlates with the repulsive potential field generated by our APF implementation. The traversed path exhibits smooth deviation patterns when encountering obstacles, confirming proper repulsive force application, while minor oscillatory artifacts suggest some room for parameter optimization. This visualization provides empirical validation of the algorithm's capacity to maintain forward progress while executing collision-free path modifications. The results confirm the real-world applicability of Hyperbolic Secant APFM modification in dynamic environments.

Computational efficiency. The computational performance of Hyperbolic Secant and Gaussian repulsive force models was evaluated through systematic benchmarking under controlled virtualized conditions. The host system featured an Intel Core i7-6700HQ mobile processor (4C/8T, 2.6–3.5 GHz) with 16GB DDR4-2666 RAM and Samsung 960 Pro NVMe SSD, running Windows 10 x64. Testing was conducted on an Ubuntu 22.04 LTS x64 virtual machine configured with 4 vCPUs and 5376MB RAM through VirtualBox, using g++ 11.4.0 (Ubuntu 11.4.0-1ubuntu1~22.04) with standard libstdc++-11 libraries.

The benchmark protocol executed 10^7 iterations of each force model ($\mu = 0, \sigma = 1$) while maintaining rigorous experimental controls. Volatile accumulators prevented compiler optimization removal, and incremental input adjustments ($\Delta x = 10^{-6}$) mitigated caching artifacts.

As demonstrated in Table 1, the Hyperbolic Secant implementation exhibited minimal performance penalties when compiled with architecture-specific optimizations. Under standard -O3 optimization, the sech model showed a 12.4 % time increase relative to the Gaussian baseline (410 ms versus 365 ms). This overhead reduced to just 2.3 % (347 ms versus 339 ms) when enabling -march=native. Contrary to expectations, the addition of -ffast-math flags increased the relative slowdown to 14.4 %, suggesting interactions between aggressive floating-point optimizations and the virtualized execution environment.

Table. 1. Computational performance comparison (10^7 iterations)

Optimization configuration	Gaussian, ms	Sech, ms
-O3	365 ± 2.1	410 ± 3.7 (+12.4 %)
-O3 -march= native	339 ± 1.8	347 ± 2.2 (+2.3 %)
-O3 -march= native-ffast-math	322 ± 1.5	368 ± 2.9 (+14.4 %)

These results collectively indicate that while Hyperbolic Secant models incur measurable but minor computational overhead compared to Gaussian formulations, proper compiler targeting can reduce this penalty to negligible levels (2.3 %). The unexpected performance degradation under –

ffast-math optimization warrants particular consideration for virtualized development environments, suggesting that standard optimization approaches may require validation when deployed in such configurations.

Conclusion

The Hyperbolic-Secant APFM represents a meaningful advancement in obstacle avoidance algorithms, offering distinct advantages in path planning quality. The intrinsic properties of the Hyperbolic Secant function – particularly its smooth gradient transitions and responsive behavior to nearby obstacles – enable more natural navigation patterns compared to Gaussian-based methods. This manifests in visibly smoother trajectories that require fewer abrupt corrections, especially when dealing with sharp-edged obstacles or complex environmental features. The method's enhanced sensitivity to proximal obstacles allows for earlier and more gradual course adjustments, reducing unnecessary path oscillations. These improvements emerge from the mathematical formulation itself rather than relying on secondary systems or sensors. Future developments could focus on optimizing the parameter selection process and investigating the method's behavior in more varied environmental conditions, potentially expanding its applicability to different robotic platforms and operational scenarios. The approach maintains the computational efficiency needed for real-time operation while delivering acceptable navigation performance.

References

- [1] Berizka I. (2024). Path planning and obstacle avoidance methods for autonomous mobile robots, ISSN 2224-087X. *Electronics and information technologies*, Issue 28, 123–142, <https://doi.org/10.30970/eli.28.11>.
- [2] Katona, K.; Neamah, H. A.; Korondi, P. (2024). Obstacle Avoidance and Path Planning Methods for Autonomous Navigation of Mobile Robot. *Sensors*, 24, 3573. <https://doi.org/10.3390/s24113573>
- [3] Debnath, D., Vanegas, F., Sandino, J., Hawary, A. F., & Gonzalez, F. (2024). A Re-view of UAV Path-Planning Algorithms and Obstacle Avoidance Methods for Re-mote Sensing Applications. *Remote Sensing*, 16(21), 4019. <https://doi.org/10.3390/rs16214019>
- [4] Hongbo Liu, Shuai Zhang, Xiaodong Yang, Overview of Path Planning Algorithms, Recent Patents on Engineering; vol. 18, issue 7. e280823220445. <https://doi.org/10.2174/1872212118666230828150857>.
- [5] Khatib O. (1986). Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *The International Journal of Robotics Research*. 5(1):90–98. <https://doi.org/10.1177/027836498600500106>.
- [6] Xiaojing Fan, Yinjing Guo, Hui Liu, Bowen Wei, Wenhong Lyu (2020 Apr). Improved Artificial Potential Field Method Applied for AUV Path Planning. *Mathematical Problems in Engineering*. Mathematical Problems in Engineering. [Online]. Available: <https://doi.org/10.1155/2020/6523158>
- [7] Jang-Ho Cho, Dong-Sung Pae, Myo-Taeg Lim, Tae-Koo Kang (2018 Aug). A Real-Time Obstacle Avoidance Method for

- Autonomous Vehicles Using an Obstacle-Dependent Gaussian Potential Field. *Journal of Advanced Transportation* [Online]. Available: <https://doi.org/10.1155/2018/5041401J>.
- [8] Berizka, I. and Karbovnyk, I. (2024). Mathematical model of modified real-time obstacle avoidance method based on laplace artificial potential field. *Applied Problems of Computer Science, Security and Mathematics*, 3 (Sep. 2024), 12–22, <https://apcssm.vnu.edu.ua/index.php/Journalone/article/view/123>.
- [9] Pardo-Castellote G. (2003). OMG Data-Distribution Service: architectural overview. In *International Conference on Distributed Computing Systems Workshops*, 200–206.
- [10] William Woodall. ROS on DDS. https://design.ros2.org/articles/ros_on_dds.html, accessed April 27, 2025
- [11] Yoo, T.; Choi, B. W. (2024). Interactive Path Editing and Simulation System for Motion Planning and Control of a Collaborative Robot. *Electronics*, 13, 2857. <https://doi.org/10.3390/electronics13142857>

I. А. Берізка, І. Д. Карбовник

Львівський національний університет імені Івана Франка, Львів, Україна

МОДЕЛЮВАННЯ ТА ОЦІНЮВАННЯ МЕТОДУ ШТУЧНИХ ПОТЕНЦІАЛЬНИХ ПОЛІВ НА ОСНОВІ ГІПЕРБОЛІЧНОГО СЕКАНСА ДЛЯ УНИКНЕННЯ ПЕРЕШКОД У РЕЖИМІ РЕАЛЬНОГО ЧАСУ

Уникнення перешкод є фундаментальною здатністю автономних мобільних роботів, що забезпечує безпечну навігацію у динамічних та неструктурованих середовищах. У цій роботі запропоновано новий підхід до уникнення перешкод у режимі реального часу на основі методу штучних потенціальних полів (Artificial Potential Field Method, APFM), який використовує функцію гіперболічного секанса. Розроблено та проаналізовано математичну модель запропонованого методу. Для перевірки ефективності підходу реалізовано віртуальне середовище на основі ROS 2, симулятора Gazebo та платформи TurtleBot3 Burger. Здійснено моделювання, у межах якого побудовано графіки вибірок даних із LiDAR сенсора разом із відповідними модулями відштовхувального, притягувального та загального потенціальних полів, що підтвердило правильність і дієвість запропонованого підходу. Додатково візуалізація в RViz продемонструвала плавність траєкторії руху робота та безперервні коригування курсу протягом 28 навігаційних кроків. Для оцінювання обчислювальної ефективності виміряно час виконання обчислень функцій Гауса та гіперболічного секанса за допомогою реалізацій на C++ із різними параметрами оптимізації компілятора. Результати показали, що обчислювальний час гіперболічного секанса приблизно на 2–3 % нижчий порівняно з функцією Гауса, що на практиці є незначною різницею на рівні похибки вимірювань. Отримані результати підтверджують доцільність використання APFM на основі гіперболічного секанса для уникнення перешкод у режимі реального часу у робототехнічних системах.

Функція гіперболічного секанса має стрімкіший спад поблизу нуля порівняно із функцією Гауса, що забезпечує сильніші миттєві відштовхувальні сили під час наближення робота до перешкоди та швидке зменшення впливу на великих відстанях. Така властивість підвищує чутливість системи та локальне уникнення перешкод без створення надмірних довгострокових ефектів, які могли б спотворити глобальну траєкторію. Крім того, симетричність і важкі “хвости” розподілу гіперболічного секанса сприяють збереженню плавності потенціального поля, забезпечуючи стабільний та передбачуваний рух робота, а також швидше й ефективніше коригування траєкторії в динамічних умовах.

Ключові слова: метод штучних потенціальних полів, кіберфізична система, обчислення на кінцевих пристроях, інформаційні технології, концепції IoT рішень, уникнення перешкод, робототехніка.

Інформація про авторів:

Берізка Ігор Андрійович, аспірант, асистент кафедри радіофізики та комп'ютерних технологій. Email: ihor.berizka@lnu.edu.ua; <https://orcid.org/0009-0007-1111-1493>

Карбовник Іван Дмитрович, д-р фіз.-мат. наук, професор, завідувач кафедри радіофізики та комп'ютерних технологій. Email: ivan.karbovnyk@lnu.edu.ua; <https://orcid.org/0000-0002-3697-4902>

Цитування за ДСТУ: Берізка І. А., Карбовник І. Д. Моделювання та оцінювання методу штучних потенціальних полів на основі гіперболічного секансу для уникнення перешкод у режимі реального часу. *Український журнал інформаційних технологій*. 2025, т. 7, № 1. С. 160–168.

Citation APA: Berizka, I. A., & Karbovnyk, I. D. (2025). Simulation-based evaluation of hyperbolic secant potential field for real-time obstacle avoidance. *Ukrainian Journal of Information Technology*, 7(1), 160–168. <https://doi.org/10.23939/ujit2025.01.160>