Electronics and Information Technologies, 2025, 30, 75–88
http://publications.lnu.edu.ua/collections/index.php/electronics/index

UDC: 004.896 + 004.415

BENCHMARKING GAUSS AND LAPLACE ARTIFICIAL POTENTIAL FIELD APPROACHES FOR REAL-TIME OBSTACLE AVOIDANCE IN VIRTUAL SCENARIOS

Ihor Berizka**, Ivan Karbovnyk
Department of Radiophysics and Computer Technologies,
Ivan Franko National University of Lviv,
107 Tarnavskoho St., Lviv 79013, Ukraine

Ihor Berizka, Ivan Karbovnyk. (2025). Benchmarking Gauss and Laplace Artificial Potential Field Approaches for Real-Time Obstacle Avoidance in Virtual Scenarios. *Electronics and Information Technologies*, 30, 75–88. https://doi.org/10.30970/eli.30.6

ABSTRACT

Background. Autonomous mobile robots require robust real-time obstacle avoidance algorithms to navigate dynamic environments safely. The Artificial Potential Field (APF) method remains widely adopted for local path planning due to its computational efficiency and conceptual simplicity. However, conventional implementations suffer from two well-documented limitations: local minima and computational inefficiencies. This study investigates two probabilistic APF variants – a Gaussian formulation (ODG-PF) and a Laplace-based approach to address these limitations

Materials and Methods. A comparative framework was developed using ROS2/Gazebo with TurtleBot3 as target platform. The Gaussian APFM (ODG-PF) and Laplace APFM were mathematically modeled, with key differences in their repulsive force calculations: Gaussian uses squared terms, while Laplace employs absolute values. Both methods were tested in identical static environments with 25 repeated runs (28 steps each). Performance metrics included computational time and path length, analyzed via boxplots, kernel density estimation, and Mann-Whitney U tests ($p<0.05$).

Results and Discussion. The Laplace APFM demonstrated superior efficiency, with 34% faster median execution time (68 μs vs. 104 μs) and tighter interquartile range (28 μs vs. 52 μs). Its unimodal time distribution contrasted with the Gaussian's bimodal pattern, attributed to simpler arithmetic operations. While both methods achieved collision-free navigation, Laplace generated statistically shorter paths ($p=0.0001$), though with marginally higher variability. The Gaussian method's squaring operations introduced computational overhead without navigational benefits.

Conclusion. The Laplace-based APFM outperforms its Gaussian counterpart in computational speed and path optimization, making it ideal for resource-constrained systems. These findings suggest that simpler mathematical formulations can yield superior real-world performance in obstacle avoidance applications. Future work should validate these findings in dynamic environments and explore hybrid implementations with global planners.

Keywords: cyber-physical system, information technologies, obstacle avoidance, mobile robotic platforms, IoT concepts, wheeled mobile platform

INTRODUCTION

In the field of robotics, a significant area of research and development is focused on autonomous mobile robots. These advanced systems are designed to navigate

independently and make context-specific decisions in real time. This capability enables them to operate without human intervention, responding adaptively to dynamic environments based on continuous sensory input.

Autonomous mobile robots are employed across a wide range of applications. Examples include service robots in hospitality environments, such as waiter robots that deliver food and beverages, and transport robots used in industrial settings to move goods efficiently. A particularly prominent example is autonomous vehicles, commonly referred to as self-driving cars. These systems integrate sophisticated sensor arrays and computational algorithms to interpret their surroundings, allowing them to navigate complex traffic scenarios and reach destinations without human control.

These examples underscore the wide-ranging applicability of autonomous mobile robots and emphasize their potential to transform numerous sectors, including hospitality, logistics, and the automotive industry. The continued development and optimization of these systems remain a central focus within the robotics research community, with ongoing efforts directed toward advancing their functional capabilities and broadening their domains of deployment.

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

Path planning is typically categorized into two main types: global and local. Global path planning relies on data from Geographic Information Systems (GIS) in conjunction with global localization techniques. This approach requires the robot to possess a comprehensive, large-scale understanding of its environment, enabling navigation over 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 environment. This form of planning focuses on short-range navigation and dynamic interactions with the surroundings, such as avoiding pedestrians on a sidewalk or maneuvering around other vehicles in traffic. Local path planning is crucial for ensuring safe and responsive behavior in unpredictable and rapidly changing environments.

A wide range of algorithms has been developed to address both global and local path planning challenges. Each of these algorithms presents distinct advantages and limitations, and their selection can have a substantial influence on the overall efficiency, reliability, and safety of autonomous navigation. Comprehensive reviews of these path planning techniques, including their underlying methodologies and application domains, can be found in the literature [1, 2, 3, 4].

Obstacle detection and avoidance constitute a critical component of local path planning algorithms, serving a vital role in ensuring the safety of both the autonomous system and its surrounding environment. This area has been the focus of extensive research over several decades, leading to the development of numerous methodological approaches. Many of these techniques have demonstrated practical effectiveness and have been successfully implemented in real-world applications.

To effectively avoid collisions, a robot must not only detect obstacles but also dynamically recalculate its path and modify its trajectory in real time. Real-time responsiveness is essential for navigating complex and dynamic environments safely and efficiently.

The initial step in obstacle avoidance involves the robot detecting potential obstacles using its onboard sensors. Once an obstacle is identified, the system must generate a new trajectory that enables safe navigation around the object. This alternative path must be computed with minimal latency to ensure that the robot can adjust its motion in real time, thereby preventing collisions and maintaining smooth path.

Additionally, the robot must be able to adapt to dynamic changes in its environment. For example, if a new obstacle unexpectedly occurs in its path, the robot needs to rapidly

detect the obstacle, compute an alternative route, and adjust its trajectory in real time to ensure safe navigation.

This demonstrates the complexity inherent in autonomous mobile robots and highlights the critical importance of ongoing research in this domain. The development and optimization of robust obstacle detection and avoidance algorithms remain central to robotics research, with the objective of improving the safety, reliability, and efficiency in terms of path length, computational time and resources of autonomous systems.

The obstacle avoidance problem can be framed as follows: A robot is situated within an unknown environment. The robot is expected to reach a specified target location or following a goal direction, all while navigating around any obstacles that may occur on its path. More detailed overview of the problem is presented in [1, 2].

MATERIALS AND METHODS

The Artificial Potential Field Method (APFM) is a well-established approach in robotics, widely applied in path planning and obstacle avoidance tasks. Originally proposed by Khatib [5], the method models the robot's environment as a virtual potential field, where the target location produces an attractive force, while obstacles produce repulsive forces. The robot, treated as a particle under the influence of these virtual forces, is guided toward the goal while being repelled from surrounding obstacles. Its motion is determined by the resultant force vector calculated as the superposition of these attractive and repulsive components. The mathematical formulation of these forces is represented by Eqs. (1)-(3).

$$\mathbf{f}_{\text{total}} = \mathbf{f}_{\text{rep}} + \mathbf{f}_{\text{attr}}, \tag{1}$$

$$\mathbf{f}_{\text{attr}} = k_{\text{attr}} \cdot \frac{\mathbf{r}_{\text{goal}} - \mathbf{r}}{\left|\mathbf{r}_{\text{goal}} - \mathbf{r}\right|}, \tag{2}$$

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

where $\mathbf{s}_i = (\mathbf{r} - \mathbf{o}_i)/|\mathbf{r} - \mathbf{o}_i|$, $\mathbf{r}_{\text{goal}}$ is the position vector of the goal point and $\mathbf{r}$ is the position vector of the vehicle, $\mathbf{o}_i$ – the position vector of each obstacle.

Despite its widespread use, the traditional Artificial Potential Field method (APFM) has several limitations. One of the primary issues is the occurrence of local minima, where the robot may become trapped in a position that is not the target, as the attractive force towards the goal and the repulsive forces from obstacles cancel each other out. Additionally, the method can result in situations where the target becomes unreachable, or the robot follows inefficient paths due to suboptimal force interactions [2, 6]. To overcome these issues, improved versions of the APF method have been proposed. One of such modifications of classic APFM is based on probabilities. In the following subsections, we examine these modifications within the mathematical models of the method in detail.

Mathematical model of Gauss APFM

The Obstacle-Dependent Gaussian Potential Field (ODG-PF) method was developed and implemented to facilitate obstacle detection and assess the probability of collision and is presented in detail in [7]. This study introduces a novel approach to calculating attractive and repulsive fields, as well as an innovative direction decision strategy. Comprehensive simulations and experimental evaluations were conducted, comparing the ODG-PF method with other potential field-based obstacle avoidance techniques. The results demonstrate that the ODG-PF method outperforms existing approaches in the majority of tested scenarios.

During the development of the mathematical model, the authors introduce a pair of values (θ, d), where θ represents the measurement angle in degrees, and d denotes the distance to an object in meters. To obtain data in this format, the authors propose using a movable ultrasonic distance sensor. Alternatively, a one-dimensional LiDAR can be employed to scan the working environment.

At the initial stage, it is necessary to define a threshold distance d_{thr}. All objects located closer to the robot than d_{thr} are considered as obstacles. This parameter can significantly influence the performance of the algorithm and therefore requires empirical tuning. Obstacles are defined using additional parameters represented as a pair $(\theta_{\text{start}}, \theta_{\text{end}})$, where θ_{start} is the angle at which the obstacle is first detected, and θ_{end} is the angle at which the detection of the same obstacle ends. This approach enables obstacle identification from one-dimensional LiDAR input data. In addition to angular boundaries, it is necessary to compute supplementary parameters for each obstacle: d_k the average distance to the k-*th* obstacle; $\Phi_k = \theta_{\text{end}} - \theta_{\text{start}}$ is the angular width occupied by the obstacle. Additionally, the robot is modeled as a square with side length w_{robot} in meters.

Although in many experiments or computer simulations the dimensions of the robot are often neglected - treating it as a point mass to simplify simulations and mathematical models - considering the robot's physical dimensions is essential for improving the model's practicality and aligning it more closely with real-world conditions. While this increases the complexity of the model to some extent, it also enhances its practical value. The authors propose a hybrid approach: the robot's dimensions are taken into account, but they are incorporated into the model by modifying the perceived size of surrounding obstacles using specific formulas. In this way, obstacles are effectively enlarged from the robot's perspective, while the robot itself continues to be modeled as a point mass.

To account for the dimensions of the robot, it is necessary to recalculate the angle according to the following equation:

$$\varphi_k = 2\sigma_k = 2\text{atan}\left[d_k \cdot \tan\left(\frac{\Phi_k}{2}\right) + \frac{w_{\text{robot}}}{2}, d_k\right] \tag{4}$$

The next step involves calculating the repulsive force exerted by each obstacle according to Eq. (5).

$$f_k(\theta_i) = A_k \cdot \exp\left[-\frac{(\theta_k - \theta_i)^2}{2\sigma_k^2}\right], \tag{5}$$

where θ_k denotes the central angle of the obstacle, and σ_k represents half of the angular width occupied by the obstacle.

The coefficient A_k is selected such that the Gaussian function fully encompasses the obstacle and is computed according to Eq. (6).

$$A_k = \tilde{d}_k \cdot \exp(0.5), \tag{6}$$

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

As with other artificial potential field-based methods, the repulsive force f_k represents the field generated by the k-*th* obstacle. Thus, the overall repulsive field is computed as the sum of the repulsive forces from all individual obstacles according to the Eq. (7).

$$f_{\text{rep}}(\theta_i) = \sum_{k=1}^{n} A_k \cdot \exp\left[-\frac{(\theta_k - \theta_i)^2}{2\sigma_k^2}\right] \tag{7}$$

The next step is the computation of the attractive field using Eq. (8). This field represents the force that draws the robot toward the specified direction θ_{goal}. Consequently, the resulting field is calculated according to Eq. (9) and determines the safe movement direction for the robot at the current iteration (Eq. (10)). The safe movement direction is computed as the argument at which the total potential field function reaches its minimum. This is a significant modification, as it simplifies the computation compared to the classical approach. Parameter γ is selected experimentally and is set to $\gamma = 0.06$

$$f_{\text{attr}}(\theta_i) = \gamma \cdot |\theta_{\text{goal}} - \theta_i|, \tag{8}$$

$$f_{\text{total}}(\theta_i) = f_{\text{attr}}(\theta_i) + f_{\text{rep}}(\theta_i), \tag{9}$$

$$\theta_{\text{dir}} = \arg\min(f_{\text{total}}). \tag{10}$$

In the traditional method, the movement direction is determined using the arctangent between two vectors, which increases computational complexity - particularly for resource-constrained systems. Another advantage of this approach is its compatibility with data obtained solely from ultrasonic distance sensors or one-dimensional LiDAR sensors. This choice of data format considerably reduces computational load, especially when compared to algorithms that rely on obstacle detection and segmentation from images (i.e., computer vision and image processing techniques). It is also worth noting that the authors conducted experiments in both static and dynamic environments and reported that the algorithm requires no additional adjustments or modifications to operate effectively in the presence of moving obstacles.

Mathematical model of Laplace APFM

Following the analysis of the work presented in [7], we identified opportunities to extend this line of research by exploring alternative probability density functions for modeling the repulsive force. To maintain symmetry in the initial modeling phase, we selected the Laplace distribution due to its simplicity and analytical tractability. In our subsequent work [8], we proposed a mathematical formulation of the repulsive field based on the Laplace function, providing a foundation for further investigation into its applicability and performance in robot navigation tasks.

All preliminary steps described for the artificial potential field method (APFM) based on the Gaussian function remain unchanged. Specifically, the environment is scanned and obstacles are detected, followed by their expansion to account for safety margins. The primary modification lies in the calculation of the repulsive force. By substituting the Gaussian function with the Laplace function, we derived the Eq. (11) to describe the repulsive force within the proposed framework.

$$f_k(\theta_i) = A_k \cdot \exp\left(-\frac{\sqrt{2} \cdot |\theta_k - \theta_i|}{\sigma_k}\right), \tag{11}$$

where θ_k denotes the central angle of the obstacle, and σ_k represents half of the angular width occupied by the obstacle.

Upon analysis of Eq. (11), it becomes evident that this formulation offers computational advantages, as it eliminates the need to square the value inside the exponential term – instead relying solely on the absolute value, which simplifies the calculation.

The coefficient A_k serves the same purpose in shaping the repulsive field as in the Gaussian variation. It is adjusted such that the Laplace function adequately covers the spatial extent of the obstacle, ensuring a precise representation of the repulsive force in its proximity, and is calculated using Eq. (12). The derivation of this coefficient is based on the

so-called sigma rule, which has been adapted for the Laplace probability density function (PDF). This approach allows for the systematic tuning of the function's spread in relation to the dimensions of the obstacle.

$$A_k = \tilde{d}_k \cdot \exp(\sqrt{2}), \tag{12}$$

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

To account for the influence of each obstacle, the total obstacle field is determined by summing the repulsive fields produced by all obstacles. Consequently, the resulting function becomes a function of the angle θ_i

$$f_{\text{rep}}(\theta_i) = \sum_{k=1}^{n} A_k \cdot \exp\left(-\frac{\sqrt{2} \cdot |\theta_k - \theta_i|}{\sigma_k}\right). \tag{13}$$

The subsequent stage involves the computation of the attractive field, which follows the same formulation as in the Gaussian-based modification. This field represents the force that draws the robot toward the target direction of movement. When combined with the repulsive field, the attractive field contributes to shaping the robot's trajectory, enabling it to avoid obstacles while steadily progressing toward the goal. The second distinction at this stage lies in the necessity to experimentally adjust the coefficient γ for the Laplace-based formulation. Parameter γ is selected experimentally and is set to $\gamma = 6.36$. Total field produced by the system is calculated using Eq. (9), and the safe direction of robot movement is determined using Eq. (10).

Evaluation framework

In robotics research and development, the use of standardized software frameworks and simulation environments is essential for efficient algorithm design, validation, and deployment. One of the most prominent and widely adopted frameworks is the Robot Operating System (ROS), which provides a comprehensive set of tools, libraries, and conventions for developing modular and scalable robotic applications. ROS enables seamless communication between different software components through a publish-subscribe architecture and supports integration with a wide variety of sensors, actuators, and control algorithms. Its flexibility and strong community support have made it a de facto standard in both academic and industrial robotics.

In parallel, simulation environments play a critical role in the development pipeline by offering safe, controlled, and reproducible testing scenarios. Among these, Gazebo stands out as one of the most robust and feature-rich robotic simulators. It provides realistic physics-based modeling of robot dynamics, environmental interactions, and sensor feedback, making it well-suited for prototyping and validating complex robotic behaviors without the risks or costs associated with physical testing. When used in conjunction with ROS, Gazebo allows for rapid iteration and debugging of robotic systems in a simulated 3D environment, closely mirroring real-world conditions.

These software tools not only streamline the development process but also enable researchers to perform extensive benchmarking and testing under varied environmental conditions, which is crucial for ensuring the reliability and robustness of robotic algorithms, especially those involved in autonomous navigation and obstacle avoidance. A more detailed review of the rationale behind the selection of ROS 2, Gazebo, and the TurtleBot3 platform is provided in [9].

Motivated by these capabilities, in [9] we introduced a dedicated testing framework built upon ROS and Gazebo, specifically designed to evaluate obstacle avoidance strategies. This framework utilizes the TurtleBot3 Burger robot model – a widely used

platform in the ROS ecosystem known for its affordability, modularity, and compatibility with both simulation and real-world deployment. The framework allows for the systematic assessment of obstacle avoidance algorithms in simulated environments that closely approximate the conditions encountered in real-world scenarios. By leveraging this simulation infrastructure, we aim to validate the correctness and performance of our proposed mathematical model prior to its application on physical robotic systems.

The proposed framework creates a virtual indoor environment populated with static obstacles, simulating common navigation challenges typically encountered by mobile robots. Within this simulated environment, the TurtleBot3 Burger robot is located in the center of the room and is tasked with navigating autonomously while avoiding collisions with the surrounding obstacles. Virtual room with obstacles in Gazebo simulator is presented on Fig. 1. The primary objective is to evaluate the robot's ability to traverse space safely and efficiently using the implemented obstacle avoidance algorithms.

In addition to the core simulation setup, the framework includes auxiliary software components for debugging, performance monitoring, and statistical analysis. These tools enable visualization and logging of key data such as LiDAR readings, computed potential fields, and the robot's steering angle at each time step, which are essential for understanding the robot's perception and control behavior. This facilitates the identification of performance issues and the refinement of navigational algorithms. The framework also supports the generation of statistical visualizations, including box plots, kernel density estimates (KDEs), and histograms for specified data. Furthermore, it performs non-parametric Mann–Whitney tests and computes statistical indicators such as the mean, median, standard deviation, min, max, IQR and p-values. The results of this analysis are presented in the following sections.

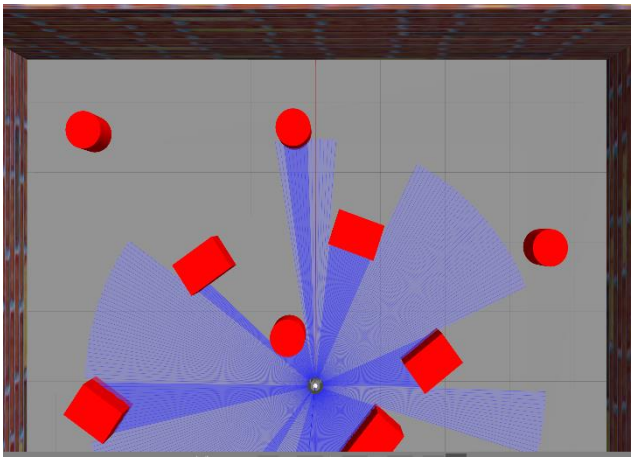

Fig. 1. View of virtual room with obstacles in Gazebo simulator. Robot is located in the center of the room. Obstacles are colored in red.

RESULTS AND DISCUSSION

In our previous work [9], we demonstrated that the Artificial Potential Field Method (APFM) based on the Laplace distribution generates smooth paths and enables reliable obstacle avoidance in environments with static obstacles, achieving collision-free navigation. In the present study, we extend that analysis by performing a comparative statistical evaluation of the Gaussian-based and Laplace-based APFM approaches. Specifically, we compare the total path lengths produced by both algorithms under identical environmental conditions. Additionally, we analyze the computational efficiency of each method.

The angle toward the goal in all the algorithms in this study is fixed and set to $\theta_{\text{goal}} = 0°$. That is to say, there is no goal position but the fixed angle and the purpose of

the obstacle avoidance algorithms in this paper is to do their best to keep the fixed heading without colliding with anything.

The tests were performed on a virtual machine running Ubuntu 20.04, equipped with 8 GB of RAM and 8 virtual CPU cores. The virtual environment was hosted on a physical machine featuring an AMD Ryzen 7 3700X processor, 32 GB of RAM, a Samsung 970 EVO Plus NVMe M.2 SSD, and an NVIDIA RTX 3070 GPU.

Statistical analysis of computational time between Gauss and Laplace APFMs

To assess the performance of two modified versions of the APFM navigation algorithm – Laplace and Gauss – a comprehensive comparative analysis was conducted. Each algorithm was executed 25 times, with each run comprising 28 discrete steps, as shown in Fig. 2, where the highlighted steps correspond to those for which execution time was measured. The primary evaluation metric was the total computational time required per highlighted steps, measured in microseconds. This procedure resulted in a dataset of 700 samples for each method. The statistical outcomes derived from this analysis are presented in Table 1.

In the context of execution time analysis for robotic algorithms, the Interquartile Range (IQR) method for outlier detection and removal was used due to its robustness and distribution-independent nature. Execution time measurements may occasionally include anomalous values caused by transient system states, background processes, or irregular hardware scheduling events - none of which reflect the true performance of the algorithm. The IQR method effectively identifies and excludes these statistical outliers by focusing on the central 50% of the data and filtering out values that lie beyond 1.5 times the interquartile range from the first and third quartiles. This ensures that the resulting statistical analysis reflects the typical behavior of the algorithm, providing more reliable and interpretable performance metrics. Given that our data may not follow a perfectly normal distribution, this non-parametric approach is especially well-suited for maintaining analytical integrity.

Following the outlier filtering step based on the interquartile range (IQR) method, the Gaussian dataset retained 637 samples, while the Laplace dataset retained 654 samples.

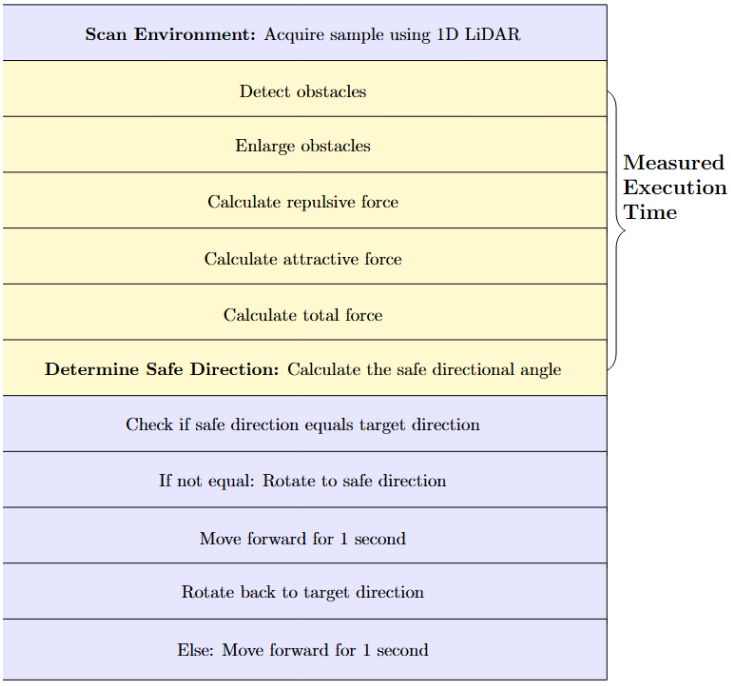

Fig. 2. Pseudocode of general robot navigation algorithm based APFM family algorithms. Steps for which execution time was measured are highlighted.

Table 1. **The calculated statistical values of computational time for Laplace and Gauss APFMs**

APFM variation	Statistical values of computational time, us					
	Mean	Median	Std	Min	Max	IQR
Laplace	76.690	68.000	26.453	26.000	159.000	28.000
Gauss	107.419	104.000	44.909	34.000	237.000	52.000

The relatively small number of excluded data points indicates a high degree of consistency in the measurements. The resulting datasets are therefore more representative of the typical performance of each method, enhancing the reliability and interpretability of the subsequent statistical analysis.

Fig. 3 presents a boxplot comparison of the execution times for the Laplace and Gauss variants of the APFM navigation algorithm. The median execution time for the Laplace algorithm is significantly lower than that of the Gauss variant, indicating superior computational efficiency. Additionally, Laplace exhibits a narrower interquartile range (IQR), suggesting more consistent performance across runs. In contrast, the Gauss algorithm shows greater variability, with a wider IQR and a higher number of outliers, some exceeding 200 microseconds. These results indicate that the Laplace-based implementation not only executes faster on average but also provides more stable timing behavior which might be crucial for real-time applications.

Fig. 4 presents execution time histograms for the Laplace and Gauss variants of the APFM navigation algorithm. The Laplace histogram (left) demonstrates a sharply peaked distribution centered around 65–70 microseconds, with the majority of samples clustered within a relatively narrow range. This indicates both high efficiency and temporal consistency. A small number of outliers are present beyond 100 microseconds, but they remain infrequent.

In contrast, the Gauss histogram (right) displays a broader and more dispersed distribution, with a secondary concentration around 120 microseconds and a notable tail extending beyond 200 microseconds. The wider spread and presence of multiple local peaks suggest greater variability and occasional latency spikes in execution time. These findings align with the results shown in the corresponding boxplot (Fig. 4), confirming the Laplace variant's advantage in both average performance and stability.

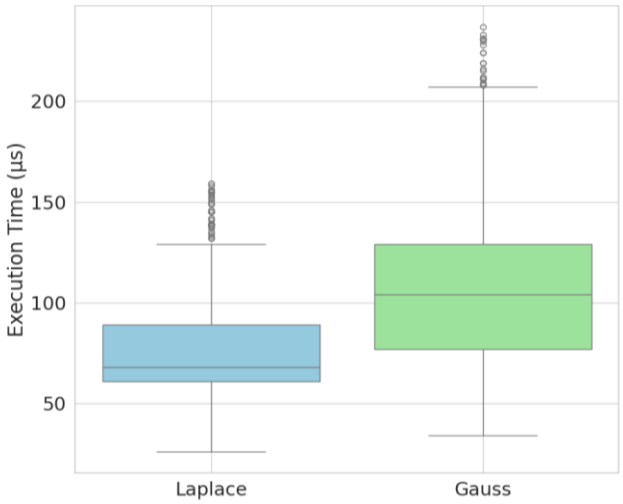

Fig. 3. Boxplot visualization of the experiment results for computational time of Laplace and Gauss modifications.

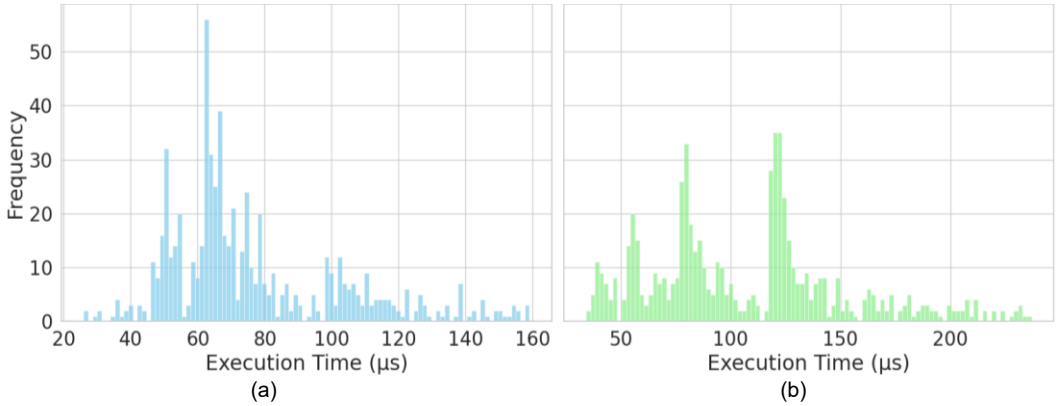

Fig. 4. Histograms of computational time for Laplace (a) and Gauss (b) algorithms. The number of bins is 100.

The Kernel Density Estimation (KDE) plot shown in Fig. 5 further illustrates the distribution of execution times for both algorithms. The Laplace variant demonstrates a unimodal distribution that is narrow and centered around lower execution times, indicating consistent and efficient performance. In contrast, the Gauss variant exhibits a clear bimodal distribution, suggesting the presence of two dominant execution time regimes. This phenomenon may be attributed to the underlying mathematical operations used in each algorithm. Specifically, the Laplace variant employs the absolute value function $abs(value)$, which is a simple and fast operation. Conversely, the Gauss variant utilizes the square function $pow(value, 2)$ under the exponent part, which can be more computationally demanding, particularly when implemented via a generic power function rather than optimized multiplication. This squaring operation introduces a nonlinear amplification of input values, leading to greater variability in computational load depending on the magnitude of the input. As a result, certain steps in the Gauss variant may trigger longer execution paths, contributing to the observed bimodal behavior in its execution time distribution.

For statistical testing we used a nonparametric Mann-Whitney U test, which produced $p\text{-}value = 0.0$. Since $p\text{-}value < 0.05$, the difference is considered statistically significant.

Statistical analysis of path lengths between Gauss and Laplace APFMs

To evaluate the performance of Laplace and Gauss modifications of the APFM algorithms, we have compared the total path length traversed by the mobile robot during 25 repeated runs for each algorithm. The primary metric of interest was the total distance covered in each trial. Calculated statistical values are presented in Table 2.

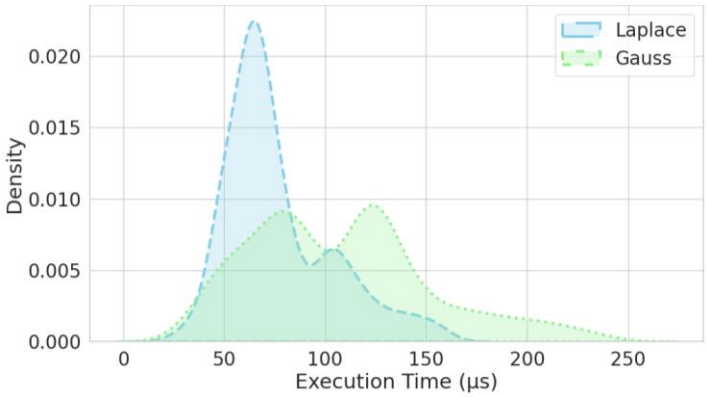

Fig. 5. Kernel density estimation of the same experiment results.

Table 2. **The calculated statistical values of path length for Laplace and Gauss APFMs**

APFM variation	Statistical values of path length, m					
	Mean	Median	Std	Min	Max	IQR
Laplace	3.993	4.000	0.052	3.840	4.060	0.070
Gauss	4.045	4.060	0.035	3.970	4.090	0.040

Due to the relatively small size of the dataset, preprocessing via the interquartile range (IQR) method was not employed. This decision was further supported by the absence of any significant interference between the path length data and potential system load, making such preprocessing unnecessary.

Visualization of the data through boxplots, histograms and kernel density estimation (KDE) are presented on Fig. 6. Left box on Fig. 6a represents data for Laplace modification, right box – Gauss modification. The black line inside each box represents the median value of the distribution, while the whiskers denote the minimum and maximum values that are not considered outliers. The boxplot reveals that the Laplace variant tends to produce slightly shorter path lengths compared to the Gauss modification, as indicated by the lower median and the overall downward shift of the Laplace box. The Laplace distribution also exhibits a wider box along the vertical axis, indicating a greater spread in the middle 50% of the data. This observation is consistent with the higher standard deviation reported in Table 2. Additionally, an outlier is present in the Laplace group, further emphasizing the increased dispersion. In contrast, the Gauss modification shows a narrower box and a lower standard deviation, reflecting more consistent, though slightly longer, path lengths.

Left histogram on Fig. 6b represents data for Laplace modification, right histogram – Gauss modification. The histograms illustrate the distribution of path lengths generated by the Laplace and Gauss modifications. The Laplace histogram shows a broader spread with a visible concentration of values around a slightly lower distance range, supporting the observation from boxplots that it tends to generate shorter paths. Additionally, the wider spread across bins suggests greater variability. The Gauss histogram, in contrast, displays

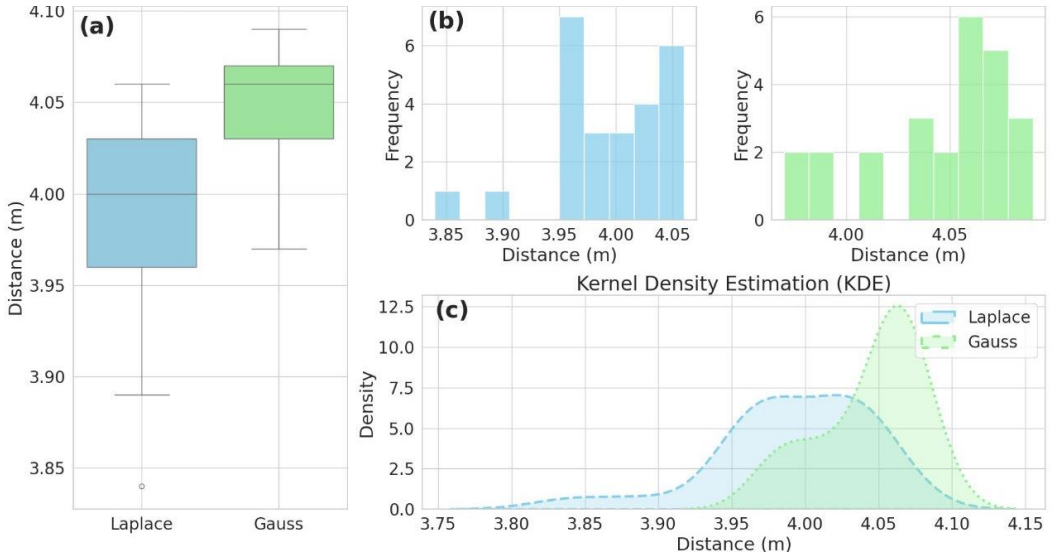

Fig. 6. Boxplot visualization of the experiment results (a); Histograms of path length for each method. Number of bins is set to 10 (b); Kernel density estimation of the same experiment results (c).

a more concentrated distribution centered around higher distance values, with most paths falling within a narrower range.

The Fig. 6c represents kernel density estimation plot, which shows the distribution of path lengths generated by the Laplace and Gauss modifications. Dashed curve corresponds to the Laplace modification, dotted curve – Gauss modification. The height of each curve at a given distance value represents the relative likelihood of that distance occurring. The Laplace curve appears shifted slightly to the left, indicating that it tends to produce shorter path lengths on average. The shape of the curve is also broader and less peaked, reflecting higher variability in path lengths - consistent with the boxplot and standard deviation data. On the other hand, the Gauss curve is more concentrated around its peak, suggesting a more stable and consistent distribution of path lengths, although the average value is slightly higher than that of the Laplace method. Overall, the KDE plot complements the histogram and boxplot by providing a smooth, continuous representation of the data distributions, highlighting both central tendency and spread.

In summary, the observations from the plots in Fig. 6 indicate that the Laplace-based approach tends to produce shorter path lengths on average, albeit with greater variability. In contrast, the Gauss-based method demonstrates more consistent performance, though at the expense of slightly longer average path lengths.

For statistical testing we used a nonparametric Mann-Whitney U test, which produced $p\text{-}value = 0.0001$. Since $p\text{-}value < 0.05$, the difference is considered statistically significant. This suggests that the Laplace algorithm tends to produce shorter paths lengths compared to the Gauss algorithm.

CONCLUSION

This study conducted a comprehensive comparison of Gaussian and Laplace-based Artificial Potential Field Methods (APFMs) for real-time obstacle avoidance in autonomous mobile robots. Through extensive simulations in a Gazebo-ROS environment using the TurtleBot3 platform, the Laplace-based APFM demonstrated notable advantages in computational efficiency and path optimization over its Gaussian counterpart.

The Laplace method achieved significantly faster execution times, with a median of 68 microseconds compared to 104 microseconds for the Gaussian approach. This performance boost stems from its mathematical simplicity, relying on absolute value calculations rather than the squaring operations required by the Gaussian method. Further analysis via Kernel Density Estimation (KDE) revealed that the Laplace variant exhibited a stable, unimodal distribution of execution times, while the Gaussian method displayed a bimodal distribution with occasional latency spikes, indicating less predictable performance.

In terms of path planning, both methods successfully facilitated collision-free navigation. However, the Laplace-based APFM generated statistically shorter paths ($p - value < 0.05$), albeit with slightly higher variability. This suggests that the Laplace method may offer greater agility in dynamic environments, though further testing is needed to confirm its robustness against moving obstacles. The computational efficiency of the Laplace approach makes it particularly well-suited for resource-constrained robotic systems, where real-time responsiveness is critical.

Despite these promising results, certain limitations need consideration. The study focused on static environments, leaving open questions about performance in scenarios with dynamic obstacles.

In summary, the Laplace-based APFM emerges as a compelling alternative for real-time obstacle avoidance, offering a balance between computational efficiency and navigational effectiveness. Its advantages in speed and path optimization position it as a viable solution for autonomous systems operating in dynamic or resource-constrained environments. Future work should investigate its adaptability in more complex scenarios and explore integrations with complementary navigation algorithms.

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that the research was performed in the absence of any potential conflict of interest.

ACKNOWLEDGMENTS AND FUNDING SOURCES

The authors received no financial support for the research, authorship, and/or publication of this article.

AUTHOR CONTRIBUTIONS

Conceptualization, [I. B.]; methodology, [I. B.]; validation, [I. B.]; formal analysis, [I. B.].; investigation, [I. B.]; resources, [I. B.]; data curation, [I. B.]; writing – original draft preparation, [I. B.]; writing – review and editing, [I. K.]; visualization, [I. B., I. K.] supervision, [I. K.]; project administration, [I. K.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Katona, K.; Neamah, H.A.; Korondi, P. Obstacle Avoidance and Path Planning Methods for Autonomous Navigation of Mobile Robot. Sensors 2024, 24, 3573. https://doi.org/10.3390/s24113573

[2] Berizka I, Path planning and obstacle avoidance methods for autonomous mobile robots, ISSN 2224-087X. Electronics and information technologies. 2024. Issue 28. P. 123–142, https://doi.org/10.30970/eli.28.11

[3] Debnath, D., Vanegas, F., Sandino, J., Hawary, A. F., & Gonzalez, F. (2024). A Review of UAV Path-Planning Algorithms and Obstacle Avoidance Methods for Remote 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; Volume 18, Issue 7, Year 2024, e280823220445. https://doi.org/10.2174/1872212118666230828150857.

[5] Khatib O. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. The International Journal of Robotics Research. 1986;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/10.1155/2018/5041401.

[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] Berizka I, Karbovnyk I, Computational Evaluation of Laplace Artificial Potential Field Methods for Real-Time Obstacle Avoidance in Gazebo, ISSN: 2524-0382 (print), 2707-0069 (online). 2025. Advances in Cyber-Physical Systems (ACPS). Volume 10, Number 1. P. 1-9, https://doi.org/10.23939/acps2025.01.001.

ВІРТУАЛЬНЕ ПОРІВНЯННЯ ФУНКЦІЙ ГАУСА ТА ЛАПЛАСА У МЕТОДАХ ШТУЧНИХ ПОТЕНЦІЙНИХ ПОЛІВ ДЛЯ УНИКНЕННЯ ПЕРЕШКОД У РЕАЛЬНОМУ ЧАСІ

Ігор Берізка**, Іван Карбовник
Львівський національний університет імені Івана Франка,
кафедра радіофізики та комп'ютерних технологій,
вул. ген. Тарнавського 107, Львів, 79017,

АНОТАЦІЯ

Вступ. Автономні мобільні роботи потребують надійних алгоритмів уникнення перешкод у реальному часі для безпечної навігації в динамічному середовищі. Метод на основі штучних потенційних полів (APFM) широко використовується для локального планування траєкторії, однак його традиційні реалізації мають недоліки у вигляді локальних мінімумів та обчислювальної неефективності. У цьому дослідженні розглянуто дві ймовірнісні модифікації APFM – на основі розподілів Гауса та Лапласа – з метою подолання цих обмежень.

Матеріали та методи. Розроблено експериментальну платформу з використанням ROS/Gazebo для симуляцій під цільову апаратну платформуTurtleBot3. Алгоритми штучних потенціальних полів із використанням функції Гауса (ODG-PF) та Лапласа були математично змодельовані, причому основною відмінністю є метод обчислення відштовхувальної сили: у гаусівській версії використовуються квадрати значень, а у версії Лапласа – абсолютні значення. Обидва методи тестувалися в однакових статичних середовищах, по 25 запусків кожного (по 28 кроків на запуск). Оцінювання продуктивності включало час обчислення у мікросекундах (мкс) та довжину траєкторії у метрах (м), які аналізувалися за допомогою boxplot-графіків, оцінки щільності розподілу ядра (KDE) та критерію Манна-Уітні ($p<0.05$).

Результати. Метод Лапласа показав вищу ефективність, забезпечивши на 34% швидший медіанний час виконання (68 мкс у порівнянні з 104 мкс) та вужчий міжквартильний розмах (28 мкс у порівнянні з 52 мкс). Модифікація із використанням функції Лапласа згенерувала унімодальний розподіл часу виконання (рис. 6), який контрастує із бімодальним розподілом Гауссової моделі, що пояснюється простішими арифметичними операціями. Обидва методи забезпечили навігацію без зіткнень, однак метод із використанням функції Лапласа сформував статистично коротші траєкторії ($p=0.0001$), хоч і з трохи більшою варіативністю. Квадратичні операції методу Гауса створили додаткове обчислювальне навантаження без покращення навігаційних характеристик.

Висновки. Метод штучних потенціальних полів на основі розподілу Лапласа переважає варіант на основі Гауса за швидкістю обчислень та оптимальністю траєкторій, що робить його придатним для систем з обмеженими ресурсами. Подальші дослідження мають підтвердити ці результати в умовах динамічного середовища, а також дослідити гібридні реалізації з глобальними планувальниками.

Ключові слова: кібер-фізична система, інформаційні технології, уникнення перешкод, мобільна робототехнічна платформа, концепції IoT рішень, колісна мобільна платформа

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
02 May, 2025	09 June, 2025	13 June, 2025	26 June, 2025