Published by
NIGERIAN SOCIETY OF PHYSICAL SCIENCES
Available online @ https://journal.nsps.org.ng/index.php/jnsps

J. Nig. Soc. Phys. Sci. 7 (2025) 2659

Journal of the Nigerian Society of Physical Sciences

Artificial potential field path length reduction using Kenneth-Nnanna-Saleh algorithm

Kenneth Christopher Ugwoke*, Nnanna Nwojo Agwu, Saleh Abdullahi

Department of Computer Science, Nile University of Nigeria, Plot 681, Cadastral Zone C, Airport Road, Jabi, Abuja, Nigeria

Abstract

The artificial potential field (APF) is one of the famous path planning algorithms. It creates a virtual force field that attracts a robot to the goal or repels it from an obstacle, forcing it to move along the direction of the resultant forces toward the goal. The repulsive force pushes the robot away from the obstacle, causing a large displacement from the straight path, increasing the path length. This paper presents the Kenneth-Nnanna-Saleh (KNS) algorithm that can shorten the length of an APF path by reducing its waypoints. The algorithm takes an APF path that is generated from the problem domain as input, evaluates angles at each point, and compares the angle with a pre-defined threshold angle to remove or retain the point in the resultant KNS path. Simulation environments, each with varying complexity in obstacle arrangement, were designed for various simulations of the proposed algorithm. A Python-based computer simulation program was implemented and used to simulate the KNS, APF, and a similar waypoint reduction algorithm - Ramer-Douglas-Peucker (RDP) and the results were analyzed. The results show that KNS can yield a shorter path than APF and RDP and retain the obstacle avoidance feature of the path. The shortened path maintains the geometry of the APF path and leads to reduced energy cost for the deployment of robots.

DOI:10.46481/jnsps.2025.2659

Keywords: Path planning, Artificial potential field, Obstacle avoidance, Robot

Article History :
Received: 03 February 2025
Received in revised form: 30 May 2025
Accepted for publication: 02 June 2025
Available online: 12 June 2025

Communicated by: O. Akande

1. Introduction

Path planning is finding a feasible, reliable, and efficient collision-free path from a starting position to a predefined goal position through a series of configurations in a given real-world environment within the shortest possible time [1–3]. It involves determining an optimal path for a robot to reach a specific destination while avoiding obstacles and minimizing costs such as time, distance, and energy consumption. Advancements in path planning for modern robots have increased the need to consider other qualities and attributes, such as the smoothness of the path, the number of bends, the sharpness of the bends, the clearance of the path from obstacles, and the jerk of the path [4, 5]. An efficient and reliable obstacle avoidance path must ensure adequate clearance around obstacles to prevent the robot from scratching against them during its journey. The path should avoid being "too close" or "too far" from obstacles [3]. However, increasing clearance often results in a longer path. Therefore, it is essential to identify a path that minimizes bends and displacement while maintaining effective obstacle avoidance, as illustrated in Figure 1.

APF method has a rich history in robotics and has undergone several advances since its inception [1, 6, 7]. The successes of the APF in planning the path for mobile robots have

*Corresponding author Tel. No.: +234-802-501-1670.
Email address: kenchijoke@yahoo.com (Kenneth Christopher Ugwoke)

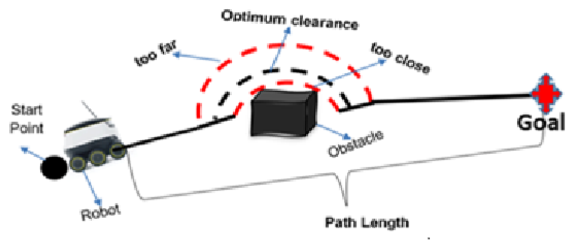

Figure 1: Optimum robotic path with appropriate clearance.

drawn the attention of several researchers with the aim of extending and improving it to provide solutions to its limitations and improve its performance [8–12]. These works have resulted in improved APF techniques capable of handling dynamic obstacle avoidance, addressing local minima and deadlocks, and integrating with other planning and control techniques and desirable features[13–18].

Energy costs are on the rise worldwide, increasing the desire to implement robots that deliver their objectives at a reduced energy cost. A reduced path length will ultimately translate to energy savings if all other factors are constant. This work presents the KNS algorithm that significantly reduces the waypoints of an APF path to yield a path that maintains the original geometry, obstacle avoidance attributes, and shortened path length. The main contributions of this paper are as follows:

1. The KNS algorithm - a novel technique that inputs an APF path and eliminates non-essential waypoints to yield a path that is shorter than the original (inputted APF) path.
2. The KNS algorithm can achieve lower path costs through shorter path lengths and reduced waypoints while avoiding collisions with obstacles.
3. The KNS algorithm takes advantage of all the great features of the APF algorithm, including its simplicity, elegance, easy implementation, use of a simple math model, and good collision avoidance attributes, to speed up the computation time and obtain results in considerably less time.
4. Simulation data, comparison data, and analysis results of the effectiveness of the KNS algorithm generated from complex environmental configurations with different obstacle arrangements.

Section 2 (Related works) presents the APF and RDP algorithms, APF pseudocode, a description of the RDP algorithm, and the RDP flowchart. Section 3 (Materials and methods) presents the KNS algorithms, KNS pseudocode, KNS steps, flowchart, and proposed algorithm description. Section 4 (Path planning simulation and analysis) presents the simulation results in tabular data and charts, a comparative study, and the analysis of the simulation results data. Finally, in Section 5, conclusions are drawn.

1.1. Background

Amongst the several approaches to measure the effectiveness of robot paths, this research adopts three measures: path length, smoothness, and obstacle avoidance. To achieve these measures, this research design followed a three-step process as listed below:

1. Generation of a path using Artificial Potential Field (APF) algorithm
2. Application of the KNS algorithm on the APF path to yield a shorter path.
3. Compare the resultant path from the application of the KNS algorithm with the original APF path and the resultant path using a similar path simplification algorithm – the RDP algorithms

Table 1: Value range for curvature (k) of a robotic path.

S/N	Value Range	Classification	Interpretation
1	$\kappa = 0$	Zero curvature	The path is a straight line
2	$0 < \kappa < 1$	Small curvature	The path has gentle, gradual bends
3	$0.1 \leq \kappa \leq 1$	Moderate curvature	The path has noticeable bends
4	$\kappa > 1$	High curvature	The path has sharp bends or tight turns
5	$\kappa \to \infty$	Infinite curvature	The path has a discontinuity or an extremely sharp turn

1.2. Evaluation of the KNS and comparison with APF and RDP path

To measure the smoothness of the generated paths, the metrics described below were computed and compared for the APF, RDP, and KNS in the various simulation environments.

1. Path Length: The total distance covered along a path. It measures how long a path is from the starting point to the goal position, including all the intermediate points or segments along the pathway. Formula: For points, (x_1, y_1), (x_2, y_2), (x_3, y_3), ..., (x_n, y_n),

$$L = \sum_{k=0}^{n} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}. \tag{1}$$

   A shorter path is preferred because it is assumed to minimize energy consumption if all other variables are constant.
2. Curvature: Measures how sharply a path bends or changes direction. A lower average curvature indicates a smoother path. Table 1 shows the value range for curvature (k).
3. Number of Turns The Number of Turns refers to the total count of directional changes from the start point to the goal point. Fewer turns usually mean a smoother path, which is preferred.

Table 2: Value range for a jerk (J) of a robotic path.

S/N	Value Range	Classification	Interpretation
1	(0 - 1 m/s³)	Low jerk	Very smooth changes in acceleration
2	(1 - 5 m/s³)	Moderate jerk	Moderate changes in acceleration, relatively smooth but noticeable
3	(5 - 20 m/s³)	High jerk	Significant changes in acceleration, potentially discomfort
4	> 20 m/s^3	Very high jerk	An extreme change in acceleration may cause damage

Table 3: Classification and interpretation of collision risk value range for a path.

S/N	Value Range	Interpretation	Action
1	0.0 - 0.9	Very low probability of collision	Continue on the current path with standard monitoring.
2	0.1 - 0.39	Low probability of collision	Maintain the current path but stay vigilant for any changes in the environment.
3	0.4 - 0.69	Moderate probability of collision	Minor adjustments to the path
4	0.7 - 1.00	High probability of collision	Adjust the path to reduce risk

4. Jerk The Jerk refers to the rate of change of acceleration over time. Lower jerk values indicate smoother transitions. Table 2 shows the value range classification for jerks in Robots [18].
5. Collision risk This is the likelihood of the robot colliding with an environmental obstacle. Table 3 shows the value range that guided the interpretation of the outcomes of our simulation results.
6. Safety score This Quantifies how safe a path is for a robot, considering various factors that could lead to collisions, and helps to determine the risk level associated with the path as classified in Table 4 [8, 19].

Table 4: Classification and interpretation of safety score value range of a path.

S/N	Value Range	Classification	Interpretation	Action
1	1 (close to 1)	Very/ highly safe	Minimal risk of collision	Good for robot
2	0.7 – 0.9	Generally safe	May have minor risk	Monitoring may be required
3	0.4 – 0.7	Marginally safe	Noticeable risk	The risk needs to be addressed
4	< 0.4	Unsafe path	Significant risk	The path is not suitable for robots

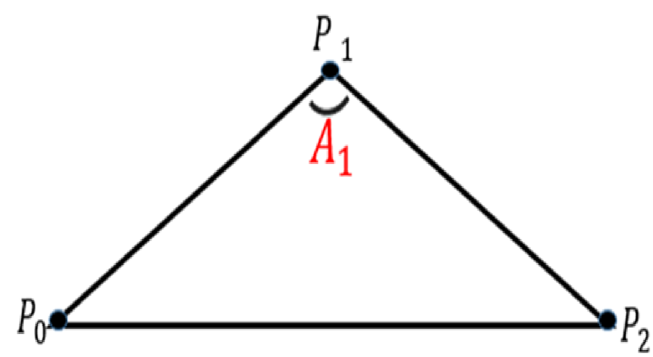

Figure 2: Angle A1 computed from points P_0, P_1 and P_2 .

2. Related works

Solutions to path planning problems have been as old as the existence of mobile robots, dating back more than 100 years during the Second World War, and have been transforming with growing advancements in computing [20]. These traverses from the first mobile robot "Shakey the robot", which was developed in the late 1960s, to recent advanced work on humanoid robots of modern days that have incorporated advanced Artificial Intelligence (machine learning and deep learning). Continuous research and advancement in path planning techniques through the enhancement, improvement, and optimization of existing techniques have resulted in more efficient, precisely controlled, and hybrid methods that employ more efficient computational methodologies [21].

2.1. The APF algorithm

Several approaches generate a reliable and feasible path for a robot. The APF path planning approach creates a virtual potential field where attractive forces guide the robot toward the goal and repulsive forces push it away from obstacles [22, 23]. The robot moves along the resulting force vectors towards the goal while avoiding obstacles. The force of the potential field F_{APP} is the sum of the attractive potential field F_{att} and the repulsive potential field F_{rep}, as shown in Equation (2).

$$F_{APP} = F_{att} + F_{rep}. \tag{2}$$

Consider a potential function U of $R^M \Rightarrow$ R where R^M is the configuration space and R is the field of real numbers. Equation (3) gives the total potential function.

$$U(q) = U_{att}(q) + U_{rep}(q), \tag{3}$$

where U_{att} is the attractive potential that moves the robot toward the goal and U_{rep} is the repulsive potential that moves the robot away from the obstacles. The total potential function is the combination of the attractive and the repulsive potential, and the resultant force is the gradient of the total potential function.

$$F(q) = -\nabla U(q) = -\nabla U_{att}(q) - \nabla U_{rep}(q), \tag{4}$$

where $-\nabla U(q)$, $-\nabla U_{att}(q)$ and $-\nabla U_{rep}(q)$ are the first derivatives of U(q), $U_{att}(q)$ and $U_{rep}(q)$ respectively. Equation (5) depicts the total potential function of an APF.

$$F(q) = F_{att}(q) + F_{rep}(q). \tag{5}$$

Pseudocode 1 gives the iterative operations of the APF algorithm to generate a reliable and feasible path.

Algorithm 1 Pseudocode 1 APF algorithm

1. Input: Start point (S), Goal point (G), Obstacles (O), Max iterations (Max_Iter), Step size (Step_Size).
2. Output: Path from S to G.
3. Initialize:.
   (a) Current position (P) = S.
   (b) Path = [S]
   (c) Iteration = 0
4. while Iteration ¡ Max_Iter and P != G:
   (a) Attractive force F_{rep} = G - P
   (b) Repulsive forces F_{rep} = 0
   (c) for each obstacle (o) in O:
      i. if P is within influence range of o:
         A. F_{rep} += Repulsive_Force(P, o)
   (d) Total force $F_{total} = F_{att} + F_{rep}$
   (e) P = P + Step_Size * NormalizeF_{total}
   (f) Path.append(P)
   (g) Iteration += 1
5. Return Path

2.2. Ramer-Douglas-Peucker algorithm (RDP)

The RDP algorithm effectively reduces the number of points in a curve while preserving its essential shape. The RDP algorithm was independently proposed by Urs Ramer, David Douglas, and Thomas Peucker in the 1970s with some improvements that kept the corner selection procedure simple and also

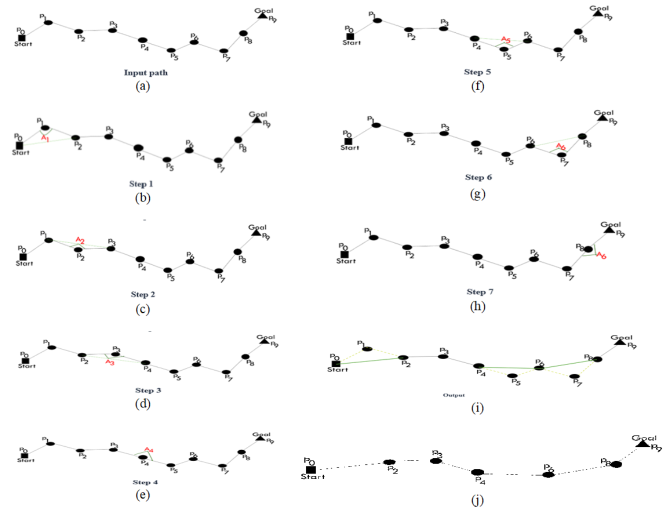

Figure 3: KNS-path algorithm steps.

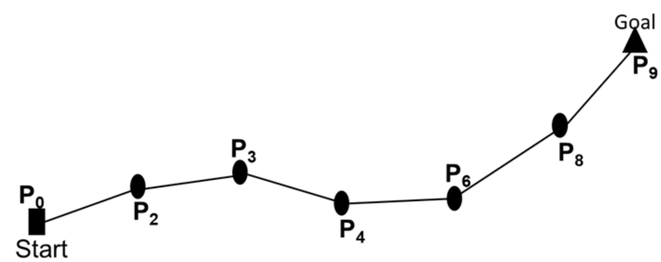

Figure 4: Adjusted path with optimum clearance and reduced path.

Table 5: Results of average performance evaluation of the APF, RDP, and KNS paths.

S/No	Measures (Average)	APF	RDP	KNS
1	Number of turns	26.9167	6.4167	7.0833
2	Collision Risk	0.0581	0.0505	0.0525
3	Clearance	3.3718	3.2905	3.4574
4	Jerk	0.0614	2.4638	0.5116
5	Curvature	0.2279	0.0844	0.3037
6	Safety Score	1.0000	1.0000	1.0000

selected corners in a more accurate pattern [24–26]. The goal is to simplify the representation of a curve by retaining the critical points and removing the points that contribute less to its overall geometry.

2.3. How RDP simplification works

The RDP takes a set of points representing a path as input, processes it through the steps of its algorithm, and outputs a simplified set of points that approximate the original path. The following steps represent the complete RDP process.

1. Select the first and last points in the input set; these are always part of the simplified path.

Table 6: Simulation environments, SE01 to SE11, configurations (q_{start} and q_{goal}) and obstacle layouts (obstacles O_i).

Environment	Start (Q_s) (X_s,Y_s)	Goal (Q_g) (X_g,Y_g)	Obstacles Points O_i = (X_i, Y_i): O_x= obstacle x position list [m], O_y = obstacle y list [m]
SE01	(0.0, 10.0)	(30.0, 30.0)	ox = [15.0, 5.0, 20.0, 25.0] oy = [25.0, 15.0, 26.0, 25.0]
SE02	(30.0, 30.0)	(0.0, 10.0)	ox = [15.0, 5.0, 8.0, 20.0, 25.0] oy = [25.0, 15.0, 15.0, 26.0, 25.0]
SE03	(0.0, 15.0)	(30.0, 26.0)	ox = [15.0, 5.0, 8.0, 20.0, 22.0, 25.0] oy = [25.0, 15.0, 15.0, 26.0, 26.0, 25.0]
SE04	(30.0, 26.0)	(0.0, 15.0)	ox = [15.0, 5.0, 8.0, 20.0, 22.0, 25.0, 27.0] oy = [25.0, 15.0, 15.0, 26.0, 26.0, 25.0, 25.0]
SE05	(0.0, 10.0)	(30.0, 26.0)	ox = [15.0, 5.0, 8.0, 16.0, 20.0, 22.0, 26.0] oy = [25.0, 15.0, 15.0, 17.0, 26.0, 26.0, 23.0]
SE06	(30.0, 26.0)	(0.0, 10.0)	ox = [15.0, 5.0, 8.0, 16.0, 20.0, 22.0, 26.0] oy = [25.0, 15.0, 15.0, 17.0, 26.0, 26.0, 23.0]
SE07	(0.0, 10.0)	(40.0, 36.0)	ox = [15.0, 5.0, 8.0, 16.0, 20.0, 22.0, 26.0, 27.0, 30.0, 35.0, 37.0, 39.0, 40.0] oy = [25.0, 15.0, 15.0, 17.0, 26.0, 26.0, 23.0, 25.0, 31.0, 32.0, 32.0, 32.0, 33.0]
SE08	(40.0, 36.0)	(0.0, 10.0)	ox = [15.0, 5.0, 8.0, 16.0, 20.0, 22.0, 26.0, 27.0, 30.0, 35.0, 37.0, 39.0, 40.0] oy = [25.0, 15.0, 15.0, 17.0, 26.0, 26.0, 23.0, 25.0, 31.0, 32.0, 32.0, 32.0, 33.0]
SE09	(0.0, 10.0)	(80.0, 76.0)	ox = [15.0, 5.0, 3.0, 20.0, 22.0, 40.0, 46.0, 62.0, 67.0, 73.0, 78.0, 30.0, 55.0] oy = [25.0, 15.0, 15.0, 26.0, 25.0, 36.0, 45.0, 58.0, 69.0, 71.0, 70.0, 23.0, 46.0]
SE10	(80.0, 76.0)	(0.0, 10.0)	ox = [15.0, 5.0, 3.0, 20.0, 22.0, 40.0, 46.0, 62.0, 67.0, 73.0, 78.0, 30.0, 55.0] oy = [25.0, 15.0, 15.0, 26.0, 25.0, 36.0, 45.0, 58.0, 69.0, 71.0, 70.0, 23.0, 46.0]
SE11	(40.0, 0.0)	(50.0, 76.0)	ox = [42.0, 43.0, 43.0, 30.0, 50.0, 50.0, 50.0, 47.0, 52.0, 47.0, 47.0, 48.0, 78.0, 55.0] oy = [2.0, 5.0, 10.0, 23.0, 21.0, 25.0, 50.0, 30.0, 55.0, 58.0, 64.0, 72.0, 70.0, 46.0]
SE12	(50.0, 76.0)	(40.0, 0.0)	ox = [42.0, 43.0, 43.0, 30.0, 50.0, 50.0, 50.0, 47.0, 52.0, 47.0, 47.0, 48.0, 78.0, 55.0] oy = [2.0, 5.0, 10.0, 23.0, 21.0, 25.0, 50.0, 30.0, 55.0, 58.0, 64.0, 72.0, 70.0, 46.0]

Table 7: Planning results for (path length in meters) for APF, RDP and KNS improvements on APF with corresponding path differences in meters and percentage reduction due to each improvement.

Environment	APF path length (m)	RDP path length (m)	KNS path length (m)	(RDP - APF) (m)	(KNS - APF) (m)	(RDP - APF) %	(KNS - APF) %
SE01	40.82	39.94	38.72	0.88	2.10	2.16	5.14
SE02	39.99	39.32	38.75	0.67	1.24	1.67	3.10
SE03	38.70	36.78	35.88	1.92	2.82	4.96	7.28
SE04	37.46	37.04	36.38	0.42	1.08	1.12	2.88
SE05	40.70	38.08	37.14	2.62	3.56	6.43	8.74
SE06	39.58	38.64	37.51	0.94	2.07	2.37	5.22
SE07	61.72	58.90	57.19	2.82	4.53	4.56	7.33
SE08	55.31	53.78	51.87	1.53	3.44	2.75	6.21
SE09	115.53	112.64	110.29	2.89	5.24	2.66	4.54
SE10	112.51	110.60	109.73	1.91	2.78	1.70	2.47
SE11	92.57	87.90	84.90	4.67	7.67	5.04	8.28
SE12	87.18	83.50	81.94	3.68	5.34	4.23	6.01

2. Calculate the perpendicular distance from each intermediate point to the line formed by the first and last points.
3. Identify the point with the maximum perpendicular distance d_{max}.
4. If d_{max} is greater than a user-defined threshold (epsilon):
   (a) Split the curve at the point with maximum distance, creating two sub-curves.
   (b) Recursively apply the RDP algorithm to each sub-curve.
5. If d_{max} is less than or equal to epsilon:

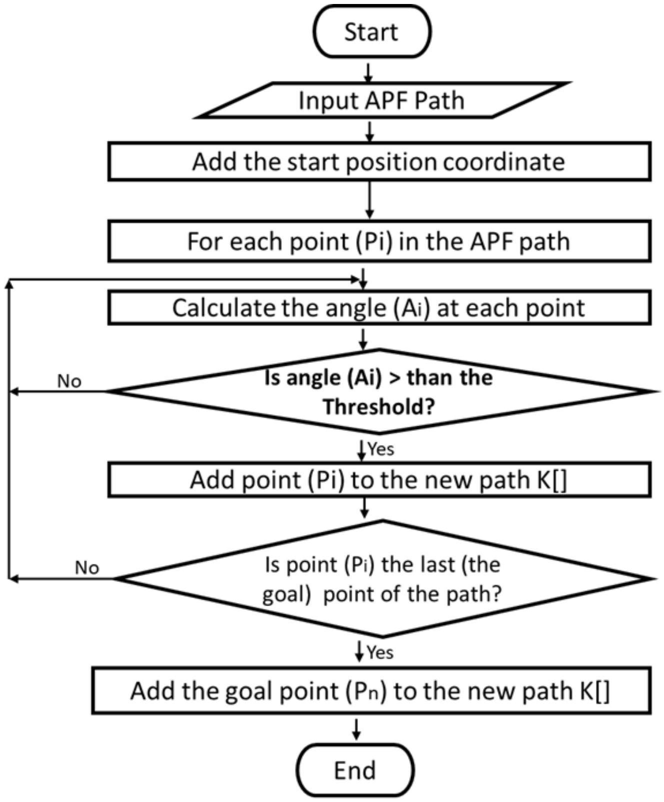

Figure 5: Flowchart of the KNS-path algorithm.

(a) The line segment between the first and last points sufficiently represents the original curve

6. End

2.4. Gap

The APF, RDP, and all the improvements researches on APF in the past and present time have focused on the notable drawbacks of APF such as local minima problem,unreachable target,poor adaptability, oscillation near obstacle, difficulty in narrow passages, etc. with the increasing energy cost worldwide,there is increasing need to generate robotic paths that are as short as possible.

3. Material and methods

As more obstacles are introduced into an APF path, the avoidance attribute of the APF algorithm (powered by the repulsive force) causes the path to bend, resulting in curves with varying angles. The generated path takes different shapes with varying degrees of bends proportionate to the obstacle's position on the robot's path in the environment.

3.1. KNS algorithm

Given a path G, as shown in equation (6),

$$G = (P_1, P_2, P_3, \dots, P_n), \tag{6}$$

where G is an APF-generated path, P_i is the set of points (x_i , y_i) coordinates that make up the path. Let be the resultant path of the KNS algorithm on G. The first point is automatically added to (G_k to maintain the geometry of the path G. Iteratively, the angle at the next point P_i, (i= 1, 2, 3, ..., n) is evaluated and compared with a predefined angle threshold θ_t. If the angle of the point θ_i is less than or equal to the defined angle threshold θ_i, the point is eliminated in the path; else, the point is added to the path G_k.

Given three points P_0, P_1, and P_2 as shown in Figure 2 as vertexes, the angle on the vertex P_1, that is the angle between the vector from P_0 to P_1 and the vector form P_1 to P_2. To evaluate the angle at P_1, three points are considered, the prior point P_0 and P_1 itself, and the next point at P_2 on the path is considered.

For P_i(r_i, θ_i), consider the points (P_{i-1}, P_i, P_{i_1}). The angle on the vertex P_i, that is the angle between the vector P_0 to P_1 and P_1 to P_2. The length of the vertex is given in equation (7).

$$\overrightarrow{P_0P_1} = P_0 - P_1,$$
$$\overrightarrow{P_1P_2} = P_1 - P_2. \tag{7}$$

The scalar product (the dot product) has the property depicted in equation (8).

$$\overrightarrow{P_0P_1} \cdot \overrightarrow{P_1P_2} = ||\overrightarrow{P_0P_1}||||\overrightarrow{P_1P_2}|| \cos\theta, \tag{8}$$

where $|| * || \; || * ||$ measures the length and θ is the angle between the two vectors is given in equation (9).

$$\theta = \arccos\left(\frac{\overrightarrow{P_0P_1} \cdot \overrightarrow{P_1P_2}}{||\overrightarrow{P_0P_1}||||\overrightarrow{P_1P_2}||}\right), \tag{9}$$

where $\arccos\theta$ = Inverse cosine function = $\cos^{-1}\theta$. Thus, generalizing using Figure 10, to evaluate the angle at P_i, three points are used: the prior point P_{i-1}, point P_i itself and the next point P_{i+1} on the path are considered. that is, for P_i (r_i, θ_i), consider points (P_{i-1}, P_i, P_{i+1}).

Therefore, given three points, P_{i-1}, P_i and P_{i+1} as vertexes, to evaluate angle at point P_i, three point: the prior point P_{i-1}, point P_i itself and next point P_{i+1} are considered. Equation (10) is the function for the computation of angle θ

$$\theta = \arccos\left(\frac{\overrightarrow{P_{i-1}P_i} \cdot \overrightarrow{P_iP_{i+1}}}{||\overrightarrow{P_{i-1}P_i}||||\overrightarrow{P_iP_{i+1}}||}\right), \tag{10}$$

where $i = 1, 2, 3, \dots, n$. Finally, the goal point P_n is added to the new path G_k to form the full path.

3.2. KNS algorithm methodology

The KNS-path algorithm was implemented using the Python development and simulation environment. The resultant path was compared with paths generated with other algorithms with similar objectives and the data results were analysed. Pseudocode 2 shows the internal working techniques of the KNS algorithm.

Figure 3 shows the steps of the KNS algorithm. It takes as input an APF path into the KNS algorithm through steps 1 to 7 and the final (shortened) path – the output also depicted in Figure 4 as an adjusted path with a shorter path length and acceptable clearance. Figure 5 illustrates the flow chart of the KNS algorithm.

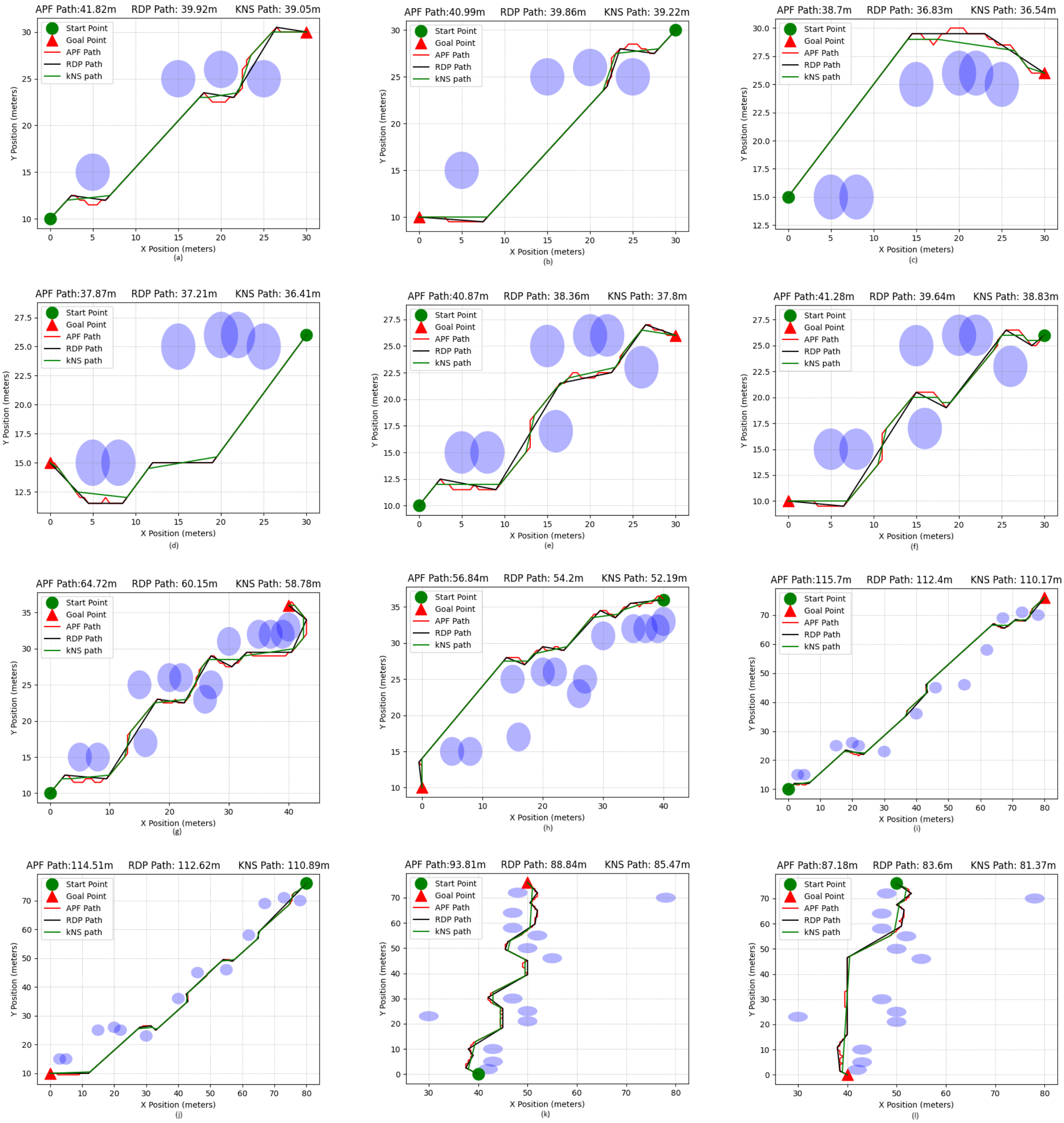

Figure 6: Robot path for different simulation environments presented in Table 6. Each map shows the outcome the APF, RDP, and K-path with their corresponding path length in meters.

4. Path planning simulation and results

The simulation environment for the proposed planning path was performed in twelve (12) different environmental configurations as indicated in Equation (11).

$$F(q_i) = Q_i(X_i, Y_i, O_i), \tag{11}$$

where $i = 1, 2, 3, \ldots, 12$.

In all simulations, the path planning algorithm begins from the initial position q_i, which is the point (x_0, y_0). As path generation progresses to the goal point, $F(q_0)$ takes into consideration the positions of all obstacles $O_i = (O_x. \ O_y)$ to generate effective avoidance. The resultant path $F(q_i) = ([x_0, y_0], [x_1, y_1], ..., [x_n, y_n])$ with starting point $= [x_0, y_0]$ and goal position $= [x_n, y_n]$ for the APF algorithm. The APF path

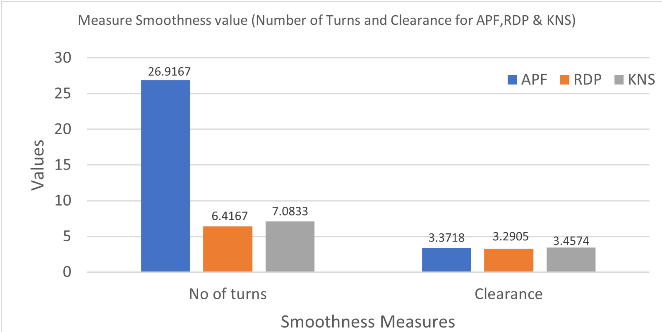

Figure 7: Measure of smoothness value (number of turns and clearance for APF, RDP & KNS).

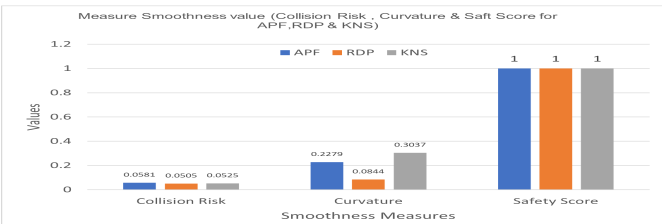

Figure 8: Measure of smoothness value (collision risk, curvature and saft score for APF, RDP & KNS).

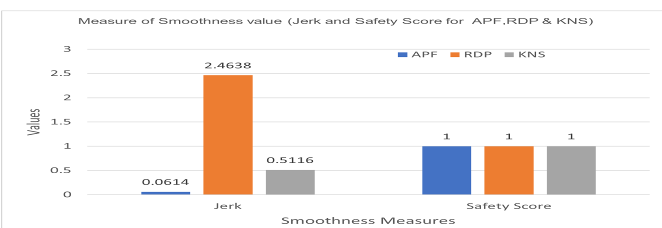

Figure 9: Measure of smoothness value (jerk and safety score for APF, RDP & KNS).

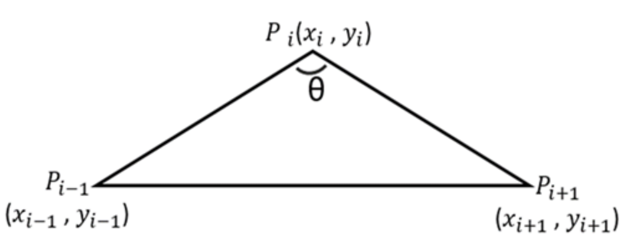

Figure 10: Angle θ computed from points P_{i-1}, P_i and P_{i+1} .

F(q_i) is fed independently as input into the RDP and KNS path reduction algorithms to generate the corresponding improved path for each of the algorithms in each simulation. All simulation environments were configured to evaluate the length of the path, the differences in the path and the percentage of path improvement relative to the original APF path. The simulation environments SE01, SE02... SE12 as shown in Table 6.

In Table 6, the different configurations of the twelve simulation environments are shown with varying complexity of the

Algorithm 2 Pseudocode 2 KNS-path algorithm

1. Start
2. $path_{in} \longleftarrow path\_Matrix_{original}$
3. $path_{out} \longleftarrow []$
4. t $\longleftarrow$ $Amgle\_threshold$
5. i $\longleftarrow$ 0
6. k $\longleftarrow$ len (P_{in}) - 1
7. $path_{out}$. append($path_{in}$ (i))
8. for i in range (1,k):
   (a) $Angle_i$ $\longleftarrow$ angle between $path_{in}$(i-1), $path_{in}$(i) and $path_{in}$(i+1)
   (b) if ($Angle_i \leq$ t)
      i. $path_{out}$. append($path_{in}$ (i))
   (c) End If
9. next i
10. $path_{out}$. append($path_{in}$ (k))
11. End

obstacle arrangement. The environments are labeled SE01 to SE12, representing simulation environment 01 to simulation environment 12, respectively. For each of the environment in the, we defined the start point (Q_s), the goal point (Q_g), and the positions of obstacles (Q_i) in each environment. The Start and goal positions provides coordinates (Q_s) (X_s,Y_s) and (Q_g) (X_g,Y_g) for each simulation environment. The obstacle position in each environment is depicted by Ox and Oy, which are arrays of x and y coordinates of each obstacle in various simulation environments.

Table 6 shows the length in meters of the generated paths by the three algorithms (APF, RDP, and KNS in each of the simulation environments SE01 to SE12. Additionally, it shows the absolute and percentage reduction in length achieved by RDP and KNS algorithms compared with the original path generated by APF. The algorithms (RDP and KNS) recorded reductions in path length in all the simulation environments, but KNS recorded a higher reduction in all the environments.

Additional indices that give more comfort on the reliability and smoothness of APF, RDP, and KNS paths are estimated and shown in Table 5. Both RDP and KNS recorded more than half (50%) reduction in the number of turns of the APF path. The three algorithms recorded very low collision risk, indicating that the chances of the path colliding with an obstacle are very low. Collision Risk of a path measures the likelihood of a path leading to a collision with obstacles or other moving entities. Additionally, the clearance measure of the three algorithms indicates that the paths have adequate space from the obstacles. Clearance of a path measures the minimum distance between the path and obstacles in the environment. The APF had the best jerk value, followed by the KNS algorithm. Jerk measures the rate of change of acceleration over time. If acceleration changes smoothly, the jerk is low; if acceleration changes suddenly, the jerk is high. On the other hand, the RDP had the least curvature (curvature measures how much a path deviates from being a straight line).

The results in Table 7 show that the KNS paths greatly improved path length, with as much as 8.74% reduction in SE05 and 7.67 meters absolute length reduction in SE11. The paths are shown in Figure 6 (a-l), which are the resultant paths for simulation environments SE01-SE12, respectively.

The graphical plot of the APF, RDP, and KNS paths generated in the twelve simulation environments is shown in Figure 6(a-l). Each of the graphs shows the different obstacle arrangements, the plot, and the path length of the three algorithms (APF, RDP, and KNS).

The obstacles are represented by the circular shapes in the graph. The plots of the various simulation environments show that the paths of the KNS algorithm avoided all the obstacles and there was no collision at any point despite the reduced waypoints that led to a shortened path. on closer observation, the graphs show that the KNS and RDP have smoother paths, which lead to shortened path length. Additionally, the plotted chart showed the path lengths of the three algorithms in all the simulations. The KNS path was the shortest in all the instances, which implies a higher reduction in path length both in absolute and percentage terms.

4.1. Analysis of results

The outcome of applying the KNS and RDP algorithms on the APF path is presented in this section. The respective path reductions in each simulation environments were observed. The two algorithms - KNS and RDP take an APF path as input, act on it using different techniques, and output a path with a shorter length as presented in the simulation results, which is depicted in Figures 6(a-l). The new paths ensure that the geometry of the path and the obstacle avoidance feature of the path were preserved [19, 25, 27]. Table 7 shows the original APF path length and the resultant RDP and KNS paths. It depicts the absolute and percentage path reduction of RDP and KNS algorithms. Both algorithms recorded path reductions in all simulation environments, with the highest impact in simulation environment SE05. The KNS algorithm recorded a significant reduction of up to 3.56 meters in absolute length, which translates to 8.74% reduction in length.

Table 5, Figures 7, 8 and 9 show the results of the average performance evaluation of the APF, RDP, and KNS paths. The KNS recorded the lowest average number of turns, indicating a higher smoothness measure than other algorithms that recorded a higher number of turns. Similarly, the collision risk measures of the three algorithms are very low, which signifies high obstacle avoidance and low risk of colliding with obstacles. The clearance results of the three algorithms reaffirms the low risk of the robot to colliding with obstacles. The safety score one (1) the algorithms strongly indicates how safe the three paths are for a robot. Finally, the low Jerk measures recorded by the three algorithms signify that the paths have very smooth changes in acceleration, which will not harm the robot and its components.

5. Conclusions

In this paper, the KNS algorithm was implemented and the simulation results (of the APF, RDP, and KNS path lengths) were compared and analyzed. Firstly, the effect of the large clearances on APF path length due to the effects of repulsive forces from obstacles in the environment was highlighted. Then, the pseudocode and the flowchart of the KNS algorithm were designed and developed. The capability of a similar way point reduction algorithms - RDP, was identified and applied to the APF path. Finally, the APF, RDP, and KNS algorithms were implemented in a Python-based computer simulation environment. Compared to APF path planning algorithms and the RDP curve smoothing technique, the KNS algorithm showed an impressive result, and the comparative resultant path Figure 6(a - l), were the effective and reliable path to move a robot from the start configuration to the goal position. Further evaluation of the KNS path showed very acceptable values of other path performance measures such as clearance, jerk, curvature, collision risk, and safety score. These results show that the KNS can significantly reduce the length of an APF path and maintain obstacle avoidance features.

Future work would be a further study of a standardized method to fix the reduction of the depth of the curve or bend around the obstacle. Another future work would be to continue to improve the path clearance to define the minimum space between the obstacle and the path relative to the speed and size of the obstacle. Furthermore, as deep reinforcement learning (DRL) and other artificial intelligence algorithms are significantly advancing many fields [28, 29], DRL applications can be examined in robot path planning to shorten path length and allow robots to navigate complex environments more efficiently and effectively.

Data availability

The data presented in this study are available upon request from the corresponding author.

Acknowledgment

The authors would like to thank Adekunle A. Adeleke of the Department of Mechanical Engineering and the Departments of Computer Science, Nile University of Nigeria staff for enabling a fit environment to carry out this work.

References

[1] H. Qin, S. Shao, T. Wang, X. Yu, Y. Jiang & Z. Cao, "Review of autonomous path planning algorithms for mobile robots", Drones **7** (2023) 211. https://doi.org/10.3390/drones7030211.

[2] Y. Benjmaa & D. Duvivier, "A review of path planning algorithms", in *Intelligent systems design and applications*, A. Abraham, A. Bajaj, T. Hanne, P. Siarry & K. Ma (Eds.), Springer Nature Switzerland, Cham, 2024, pp. 119–130. https://doi.org/10.1007/978-3-031-64850-2_11.

[3] A. N. A. Rafai, N. Adzhar & N. I. Jaini, "A review on path planning and obstacle avoidance algorithms for autonomous mobile robots", J. Robot. **2022** (2022) 2538220. https://doi.org/10.1155/2022/2538220.

[4] S. Dian, J. Zhong, B. Guo, J. Liu & R. Guo, "A smooth path planning method for mobile robot using a BES-incorporated modified QPSO algorithm", Expert Syst. Appl. **208** (2022) 118256. https://doi.org/10.1016/j.eswa.2022.118256.

[5] K. S. Suresh, R. Venkatesan & S. Venugopal, "Mobile robot path planning using multi-objective genetic algorithm in industrial automation", Soft Comput. **26** (2022) 7387. https://doi.org/10.1007/s00500-022-07300-8.

[6] D. Han, Q. Yu, H. Jiang, Y. Chen, X. Zhu & L. Wang, "Three-dimensional path planning for post-disaster rescue UAV by integrating improved grey wolf optimizer and artificial potential field method", Appl. Sci. **14** (2024) 4461. https://doi.org/10.3390/app14114461.

[7] L. Liu, X. Wang, X. Yang, H. Liu, J. Li & P. Wang, "Path planning techniques for mobile robots: review and prospect", Expert Syst. Appl. **227** (2023) 120254. https://doi.org/10.1016/j.eswa.2023.120254.

[8] A. S. Abdel-Rahman, S. Zahran, B. E. Elnaghi & S. F. Nafea, "Enhanced hybrid path planning algorithm based on APF and A-star", Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. **48** (2023) 867. https://doi.org/10.5194/isprs-archives-XLVIII-1-W2-2023-867-2023.

[9] J. Yang, H. Zhang & P. Ning, "Path planning and trajectory optimization based on improved APF and multi-target", IEEE Access **11** (2023) 139121. https://doi.org/10.1109/ACCESS.2023.3338683.

[10] M. Wakchaure, B. K. Patle & A. K. Mahindrakar, "Artificial intelligence in the life sciences application of AI techniques and robotics in agriculture: a review", Artif. Intell. Life Sci. **3** (2023) 100057. https://doi.org/10.1016/j.ailsci.2023.100057.

[11] G. Huang & Q. Ma, "Research on path planning algorithm of autonomous vehicles based on improved RRT algorithm", Int. J. Intell. Transp. Syst. Res. **20** (2022) 170. https://doi.org/10.1007/s13177-021-00281-2.

[12] F. Method, "Research on autonomous vehicle path planning based on improved A* algorithm and improved DWA algorithm", Sensors **24** (2024) 3899. https://doi.org/10.3390/s24123899.

[13] L. Cao, L. Wang, Y. Liu & S. Yan, "3D trajectory planning based on the rapidly-exploring random tree–connect and artificial potential fields method for unmanned aerial vehicles", Int. J. Adv. Robot. Syst. **19** (2022) 1. https://doi.org/10.1177/17298806221118867.

[14] J. Fan, X. Chen & X. Liang, "UAV trajectory planning based on bi-directional APF-RRT* algorithm with goal-biased", Expert Syst. Appl. **213** (2023) 119137. https://doi.org/10.1016/j.eswa.2022.119137.

[15] M. Li, B. Li, Z. Qi, J. Li & J. Wu, "Optimized APF-ACO algorithm for ship collision avoidance and path planning", J. Mar. Sci. Eng. **11** (2023) 1177. https://doi.org/10.3390/jmse11061177.

[16] J. Luo, Z. Wang & K. Pan, "Reliable path planning algorithm based on improved artificial potential field method", IEEE Access **10** (2022) 108276. https://doi.org/10.1109/ACCESS.2022.3212741.

[17] L. Liu & B. Wang, "Research on path-planning algorithm integrating optimization a-star algorithm and artificial potential field method", Electronics **11** (2022) 3660. https://doi.org/10.3390/electronics11223660.

[18] G. Hao, Q. Lv, Z. Huang, H. Zhao & W. Chen, "UAV path planning based on improved artificial potential", Aerospace **10** (2023) 562. https://doi.org/10.3390/aerospace10060562.

[19] X. Lv, W. Li & J. Wang, "Safety-field-based path planning algorithm of lane changing for autonomous vehicles", Int. J. Control Autom. Syst. **20** (2022) 564. https://doi.org/10.1007/s12555-020-0942-3.

[20] R. Raj & A. Kos, "A comprehensive study of mobile robot: history, developments, applications, and future research perspectives", Appl. Sci. **12** (2022) 6951. https://doi.org/10.3390/app12146951.

[21] A. A. Nippun Kumaar & S. Kochuvila, "Mobile service robot path planning using deep reinforcement learning", IEEE Access **11** (2023) 100083. https://doi.org/10.1109/ACCESS.2023.3311519.

[22] M. Jones, S. Djahel & K. Welsh, "Path-planning for unmanned aerial vehicles with environment complexity considerations: a survey", ACM Comput. Surv. **55** (2023) 1. https://doi.org/10.1145/3570723.

[23] H. Han, Z. Xi, J. Cheng & M. Lv, "Obstacle avoidance based on deep reinforcement learning and artificial potential field", in *2023 9th International Conference on Control, Automation and Robotics (ICCAR)*, 2023, pp. 215–220. https://doi.org/10.1109/ICCAR57134.2023.10151771.

[24] A. Gültekin, S. Diri & Y. Becerikli, "Simplified and smoothed rapidly-exploring random tree algorithm for robot path planning", Teh. Vjesn. **30** (2023) 891. https://doi.org/10.17559/TV-20221015080721.

[25] B. Dergisi, "Effective solutions for common problems of artificial potential field based path planning algorithms for mobile robots", Bujse **15** (2022) 105. https://doi.org/10.20854/bujse.1214752.

[26] J. M. Jayoma, G. P. Navales, F. A. M. Bustamante & L. R. Tradio, "Enhanced Ramer-Douglas-Peucker algorithm for efficient 2D line simplification", in *2023 IEEE 15th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, 2023, pp. 1–6. https://doi.org/10.1109/HNICEM60674.2023.10589058.

[27] B. Dergisi, "Effective solutions for common problems of artificial potential field based path planning algorithms for mobile robots mobil robotlar için yapay potansiyel alan tabanli yol planlama mixed integer linear programming approach artificial potential fields vec", Bujse **15** (2022) 105. https://doi.org/10.20854/bujse.1214752.

[28] M. Kara, "Evaluation of popular path planning algorithms", Int. J. Electron. Telecommun. **70** (2024) 13. https://doi.org/10.24425/ijet.2023.147699.

[29] R. Raza, A. T. Ab Ghani & L. Abdullah, "Extension of hesitant fuzzy weight geometric (HFWG)-VIKOR method under hesitant fuzzy information", J. Niger. Soc. Phys. Sci. **6** (2024) 2157. https://doi.org/10.46481/jnsps.2024.2157.