

**“Improving Robot Path Planning in Linear  
and Nonlinear Environments Using Hybrid  
Optimization Algorithms with Advanced  
Smoothing”**

**By:  
Mohammed Falah & Muthana Tariq**

**2025-2026**

**Authors:**

**1. Mohammed Falah Hassan Al-Dhafiri**

**2. Muthanna Tariq Faraj Al-Nuaimi**

**Improving Robot Path Planning in Linear and Nonlinear Environments Using Hybrid“  
”Optimization Algorithms with Advanced Smoothing  
By: Mohammed Falah & Muthana Tariq**

---

**Abstract:**

Path planning is a fundamental challenge in mobile robotic systems, particularly in complex environments that include static and dynamic obstacles as well as strict kinematic constraints. This paper aims to improve linear and nonlinear robot path planning by developing a hybrid optimization framework that achieves an effective balance between path length, smoothness, safety, energy consumption, and planning time.

The proposed approach introduces a hybrid system that integrates Particle Swarm Optimization (PSO) and Differential Evolution (DE), enhanced by a multi-stage advanced smoothing mechanism. The smoothing process is designed to reduce path curvature and improve trajectory continuity while preserving the robot’s kinematic feasibility. In addition, the system incorporates realistic models for energy consumption and safety evaluation and supports navigation in two-dimensional environments with multiple dynamic obstacles.

The proposed framework is evaluated against five baseline methods, including non-optimized planning, PSO-only, DE-only, PSO+GA, and PSO+DE, using a unified and fair evaluation scheme based on six key performance metrics. Experimental results demonstrate that the proposed PSO+DE with advanced smoothing approach consistently achieves the best overall performance, showing significant improvements in smoothness and safety while maintaining high energy efficiency and low planning time, even in highly dynamic environments.

These results indicate that combining intelligent optimization algorithms with advanced smoothing techniques provides an effective and practical solution for robotic path planning and represents an important step toward more reliable and realistic robotic navigation systems in complex dynamic environments.

**1.Introduction:**

Autonomous robot navigation has become a critical research topic in mobile robotics due to its wide range of applications, including warehouse automation, autonomous vehicles, service robots, and search-and-rescue systems. One of the core challenges in autonomous navigation is path planning, which aims to generate a feasible, safe, and efficient trajectory from a start point to a target destination while satisfying environmental and kinematic constraints.

In real-world environments, robot path planning is particularly difficult due to the presence of static and dynamic obstacles, nonlinear motion constraints, limited sensing accuracy, and the need to balance multiple competing objectives such as path length, smoothness, safety, energy consumption, and computational efficiency. Classical deterministic planning methods, such as grid-based search algorithms, often struggle to maintain performance in highly

**Improving Robot Path Planning in Linear and Nonlinear Environments Using Hybrid“  
”Optimization Algorithms with Advanced Smoothing  
By: Mohammed Falah & Muthana Tariq**

---

dynamic or nonlinear environments, especially when smooth and energy-efficient trajectories are required.

Metaheuristic optimization algorithms have been widely adopted to address these challenges due to their flexibility and strong global search capabilities. Algorithms such as Particle Swarm Optimization (PSO), Genetic Algorithms (GA), and Differential Evolution (DE) have demonstrated promising results in optimizing robot paths, particularly in complex and high-dimensional search spaces. However, each of these methods has inherent limitations. PSO may converge prematurely, GA can suffer from slow convergence, and DE alone may struggle to produce smooth trajectories suitable for real robotic execution.

To overcome these limitations, hybrid optimization approaches have been proposed, combining the strengths of multiple algorithms. While hybrid methods can significantly improve solution quality, many existing studies focus primarily on path length or obstacle avoidance, often neglecting trajectory smoothness, energy efficiency, and realistic safety modeling, which are essential for deployment on real robotic platforms. Furthermore, smoothing techniques are frequently treated as post-processing steps without deep integration into the optimization framework.

In this paper, we propose a hybrid path planning framework that integrates PSO and DE with an advanced multi-stage smoothing mechanism to improve both the quality and feasibility of robot trajectories in linear and nonlinear environments. The proposed system explicitly considers energy consumption, safety margins, and curvature reduction, while supporting navigation in environments with multiple dynamic obstacles. A unified evaluation framework is introduced to ensure fair comparison across different planning algorithms.

The proposed approach is extensively evaluated against five baseline methods under identical conditions. Experimental results demonstrate that the proposed PSO+DE with advanced smoothing consistently outperforms other methods in terms of overall path quality, safety, and smoothness, while maintaining competitive planning times. These results highlight the effectiveness of combining hybrid optimization with advanced smoothing for realistic robotic path planning in complex environments.

## **2. Research Contributions:**

The main contributions of this work can be summarized as follows:

### **2.1. Hybrid Path Planning Framework**

This paper proposes a novel hybrid path planning framework that combines Particle Swarm Optimization (PSO) and Differential Evolution (DE) to exploit both fast global exploration and robust solution refinement, enabling efficient path optimization in linear and nonlinear environments.

## **2.2.Advanced Multi-Stage Smoothing Mechanism**

An advanced smoothing strategy is integrated directly into the optimization process rather than being applied as a post-processing step. The proposed smoothing mechanism significantly reduces path curvature and abrupt directional changes, resulting in trajectories that are more suitable for real robotic execution.

## **2.3.Dynamic Obstacle-Aware Planning**

The proposed system explicitly accounts for multiple dynamic obstacles, incorporating predictive safety constraints and adaptive safety margins to enhance robustness in dynamic and uncertain environments.

## **2.4.Energy and Safety-Aware Objective Function**

A unified objective function is introduced that simultaneously optimizes path length, smoothness, energy consumption, safety, and execution time, reflecting realistic constraints encountered by physical robotic platforms.

## **2.5.Unified and Fair Evaluation Framework**

A comprehensive and fair evaluation framework is developed to compare six different planning strategies under identical conditions, ensuring unbiased performance assessment across all metrics.

## **2.6.Extensive Comparative Analysis**

Extensive experimental evaluations demonstrate that the proposed PSO+DE with advanced smoothing approach consistently outperforms baseline and single-optimizer methods, achieving superior overall performance in complex environments with both static and dynamic obstacles.

## **3.Related Work:**

Path planning has been a fundamental research topic in robotics for several decades, aiming to generate safe, efficient, and feasible trajectories for autonomous robots operating in complex environments. Existing approaches can be broadly categorized into classical graph-based methods, sampling-based planners, and metaheuristic optimization techniques.

### **3.1 Classical Path Planning Methods**

Classical path planning algorithms such as A\*, Dijkstra, and Visibility Graphs have been widely used due to their deterministic behavior and guaranteed completeness under certain conditions. These methods are computationally efficient and suitable for structured and static environments. However, they often produce paths with sharp turns and discontinuities,

making them less suitable for real-world robotic execution without additional smoothing or post-processing.

### **3.2 Sampling-Based Planning Algorithms**

Sampling-based planners, including Probabilistic Roadmaps (PRM) and Rapidly-exploring Random Trees (RRT and RRT\*), have demonstrated strong performance in high-dimensional configuration spaces. While these methods offer probabilistic completeness and flexibility, they may generate suboptimal or highly irregular paths and typically require extensive smoothing and refinement to meet kinematic and dynamic constraints.

### **3.3 Metaheuristic Optimization-Based Approaches**

Metaheuristic algorithms such as Particle Swarm Optimization (PSO), Genetic Algorithms (GA), and Differential Evolution (DE) have gained significant attention for path planning due to their ability to handle nonlinear, multi-objective optimization problems.

**PSO-based methods** are known for fast convergence and efficient global exploration but may suffer from premature convergence in complex environments.

**GA-based approaches** provide strong diversity through crossover and mutation operators but often require higher computational effort.

**DE-based planners** exhibit robust performance in continuous optimization problems; however, their effectiveness is highly sensitive to parameter tuning and population diversity.

### **3.4 Hybrid Optimization Strategies**

To overcome the limitations of single-optimizer approaches, several studies have explored hybrid strategies combining multiple metaheuristics. Hybrid PSO–GA and PSO–DE approaches have shown improved convergence behavior and solution quality. Nevertheless, many existing hybrid methods apply smoothing as a post-processing step, which may violate safety constraints or degrade path feasibility.

### **3.5 Path Smoothing and Feasibility Enhancement**

Path smoothing techniques, such as Bézier curves, B-splines, and gradient-based smoothing, are commonly used to improve path continuity. While effective in reducing sharp turns, these techniques are often decoupled from the optimization process, leading to potential safety violations in cluttered or dynamic environments.

### **3.6 Research Gap and Motivation**

Despite extensive research on hybrid optimization-based path planning, there remains a lack of frameworks that:

1. Integrate smoothing directly within the optimization loop,
2. Explicitly consider dynamic obstacles and safety constraints, and
3. Provide fair and unified comparative evaluations across multiple planning strategies.

This work addresses these gaps by proposing a hybrid PSO+DE framework with advanced integrated smoothing, designed for realistic robotic environments involving both static and dynamic obstacles.

#### **4.Problem Formulation and System Model:**

##### **4.1 Robot Model**

We consider a mobile robot operating in a two-dimensional workspace. The robot is modeled as a circular rigid body with radius  $r$ , navigating from a given start position to a predefined goal while avoiding obstacles.

The robot state at time  $t$  is defined as:

$$\mathbf{x}(t) = [x(t), y(t), \theta(t)]^T$$

where  $x(t)$ ,  $y(t)$  represents the planar position and  $\theta(t)$  denotes the robot heading angle.

The robot follows a differential-drive kinematic model, expressed as:

$$\dot{x}(t) = v(t) \cos(\theta(t))$$

$$\dot{y}(t) = v(t) \sin(\theta(t))$$

$$\dot{\theta}(t) = \omega(t)$$

where:

$v(t)$  is the linear velocity,

$w(t)$  is the angular velocity.

The control inputs are constrained as:

$$0 \leq v(t) \leq v_{\max}, \quad |\omega(t)| \leq \omega_{\max}$$

These constraints reflect the physical limitations of real robotic platforms.

#### **4.2 Environment and Obstacle Modeling**

The environment is defined as a bounded workspace:

$$\mathcal{W} \subset \mathbb{R}^2$$

containing both static and dynamic obstacles.

##### Static Obstacles

Static obstacles are represented as geometric primitives (circles or rectangles) with fixed positions:

$$\mathcal{O}_s = \{O_s^1, O_s^2, \dots, O_s^N\}$$

##### Dynamic Obstacles

Dynamic obstacles move according to known or estimated velocity models:

$$\mathcal{O}_d(t) = \{O_d^1(t), O_d^2(t), \dots, O_d^M(t)\}$$

Each dynamic obstacle position evolves as:

$$\mathbf{p}_d(t+1) = \mathbf{p}_d(t) + \mathbf{v}_d(t)\Delta t$$

where  $\mathbf{v}(t)$  is the obstacle velocity vector.

#### 4.3 Safety Constraints

To ensure collision-free navigation, a safety margin  $d_{safe}$  is enforced around all obstacles:

$$\text{dist}(\mathbf{p}_r(t), \mathbf{p}_o(t)) \geq r_r + r_o + d_{safe}$$

where:

$\mathbf{p}_r(t)$  is the robot position,

$\mathbf{p}_o(t)$  is the obstacle position,

$r_o$  denotes the obstacle radius (or equivalent bounding radius).

#### 4.4 Path Representation

A path is represented as a sequence of waypoints:

$$\mathcal{P} = \{\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_K\}$$

where:

$\mathbf{p}_0$  start,

$\mathbf{p}_K$  goal,

$K$  is fixed for fair comparison across all planners.

Each waypoint satisfies the safety constraints and lies within the workspace.

#### 4.5 Optimization Objective

The path planning problem is formulated as a multi-objective optimization problem:

$$\min_{\mathcal{P}} J(\mathcal{P})$$

where the total cost function is defined as:

$$J = w_1 J_{fit} + w_2 J_{len} + w_3 J_{smooth} + w_4 J_{energy} + w_5 J_{safety}$$

With:

**Path Fitness (Goal Proximity): •**

$$J_{fit} = \sum_{i=1}^K \|\mathbf{p}_i - \mathbf{p}_{goal}\|$$

**Path Length: •**

$$J_{len} = \sum_{i=1}^K \|\mathbf{p}_i - \mathbf{p}_{i-1}\|$$

**Path Smoothness (Curvature Cost): •**

$$J_{smooth} = \sum_{i=2}^{K-1} \|\mathbf{p}_{i+1} - 2\mathbf{p}_i + \mathbf{p}_{i-1}\|^2$$

**Energy Consumption: •**

$$J_{energy} = \sum_{i=1}^K (v_i^2 + \alpha \omega_i^2)$$

**Safety Penalty: •**

$$J_{safety} = \sum_{o \in \mathcal{O}} \max(0, d_{safe} - \text{dist}(\mathbf{p}_i, \mathbf{p}_o))$$

where  $w_i$  are weighting coefficients controlling the relative importance of each objective.

#### **4.6 Problem Statement**

Given a start position, goal position, robot kinematic constraints, and a set of static and dynamic obstacles, the objective is to compute a collision-free, smooth, and energy-efficient

path that minimizes the defined multi-objective cost function while satisfying all kinematic and safety constraints.

## **5. Proposed Hybrid Algorithm / Methodology:**

This section is divided into six scenarios:

### **Scenario 1:**

This scenario represents a safe path planning system for robots using a hybrid PSO+GA algorithm with optional smoothing. It operates in a small environment (10×10) with defined start and goal points, and uses five intermediate points to generate the path. The geometry functions calculate path length, smoothness and safety, applying simple penalties and rewards. The fitness function is essentially linear, combining these metrics with fixed weights and converting them into a final score. The code is designed to be simple and safe, making it suitable for educational or initial research experiments rather than real-world industrial robotics applications.

#### **1. Initialization**

- A set of particles (solutions) is generated randomly within the allowed bounds.
- Each particle represents a candidate path between the start and goal points.

Equation:

$$x(i,j) \sim U(\text{boundsmin}, \text{boundsmax})$$

#### **2. Fitness Function Evaluation**

For each path, the following metrics are calculated:

- Path Length:

$$L = \sum \| P(k+1) - P(k) \|$$

- Smoothness:

$$\theta = \arccos( (v1 \cdot v2) / (\|v1\| * \|v2\|) )$$
$$S = \text{average of turning angles or a function of } \theta$$

- Safety:

$$\text{Safety} = d(\text{min}) / \text{margin}$$

- Energy:

$$E = L * c$$

- Final Fitness:

$$F = 0.30 (1 - L(\text{norm})) + 0.30 (1 - S(\text{norm})) + 0.25 (1 - \text{Safety}(\text{norm})) + 0.15 (1 - E(\text{norm}))$$

### **3. Particle Swarm Optimization (PSO)**

Particles update their velocity and position using:

- Velocity update:

$$v(i)(t+1) = w \ v(i)(t) + c1 \ r1 \ (p(i) - x(i)(t)) + c2 \ r2 \ * \ (g - x(i)(t))$$

- Position update:

$$x(i)(t+1) = x(i)(t) + v(i)(t+1)$$

### **4. Genetic Algorithm (GA)**

Applied periodically to improve diversity:

- Crossover:

$$\text{Child} = \alpha \ P1 + (1 - \alpha) \ P2$$

- Mutation:

$$\text{Child}[j] = \text{Child}[j] + \delta$$

### **5. Smoothing**

After optimization, smoothing is applied to reduce sharp angles:

Equation:

$$P(i)_{\text{new}} = 0.7 \ P(i) + 0.3 \ ((P(i-1) + P(i+1)) / 2)$$

### **6. Final Output**

- The best solution is chosen between the original and the smoothed path.
- Metrics (length, smoothness, safety, energy, fitness) are recalculated.
- Results are reported for each run, then averaged and compared across algorithms.

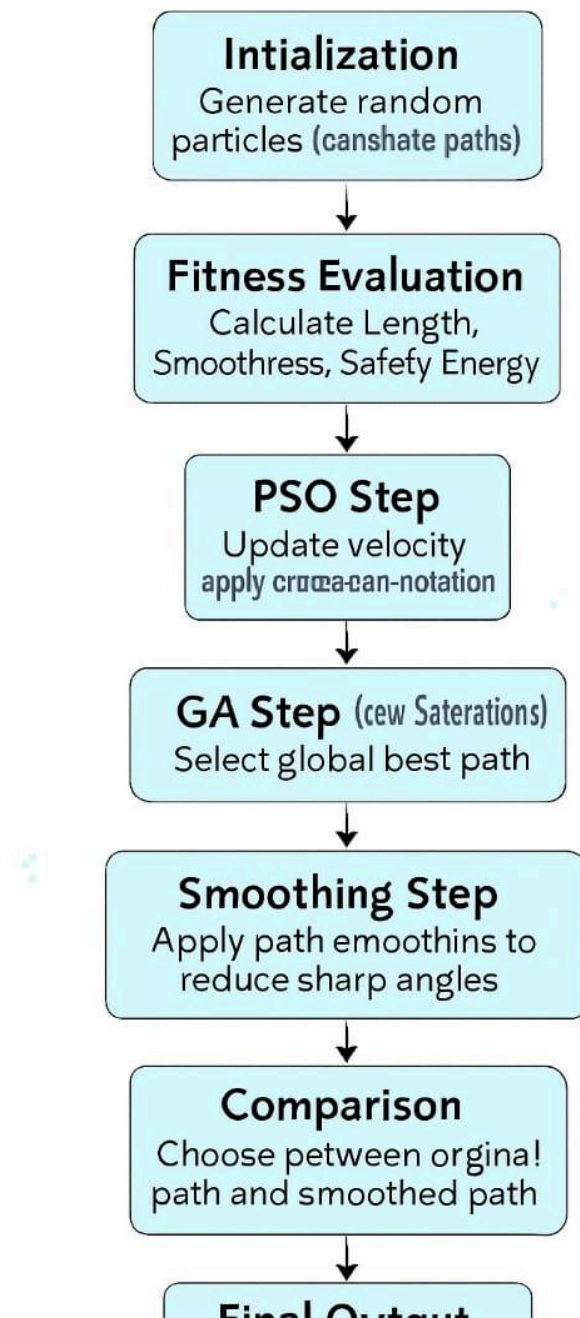


Fig. 1

**System Classification:**

1. System: Linear (simple equations with constant ratios).
2. Complexity Level: Simple (suitable for education and initial experiments).

3. Realism: More research-oriented than practical (not intended for an actual robot).

### **Scenario 2:**

This system is represented by the final complete version of the path optimization system, designed to clearly demonstrate the benefit of smoothing. It includes four algorithms: the basic random path (Baseline), optimization using PSO only, optimization with limited PSO+GA, and finally PSO+GA with smoothing. The system places large obstacles along the path to force poor zigzag solutions, then shows how smoothing improves smoothness and overall scores. The fitness function is non-linear, rewarding slightly zigzag paths and applying light penalties, while the final score depends on smoothness, safety, and length. The main goal is to prove that smoothing significantly enhances the performance of weak algorithms.

This scenario simulates a challenging robot path planning environment with large obstacles and intentionally poor initial paths. It uses four algorithms to highlight the benefit of smoothing:

1. Baseline (no optimization)
2. PSO only
3. Limited PSO+GA
4. Limited PSO+GA + Smoothing

Each algorithm generates paths, evaluates them using geometric and safety metrics, and assigns scores based on smoothness, safety, and length.

### **Step-by-Step Process**

#### **1. Initialization**

- Each algorithm starts by generating candidate solutions (paths) with 5 intermediate points (10 dimensions).
- The decodepathzigzag function creates intentionally zigzagged paths to simulate poor quality.

Equation:

$$x(i,j) \sim \text{Uniform}(\text{boundsmin}, \text{boundsmax})$$

## **2. Path Evaluation**

Each path is evaluated using four key metrics:

### **(a) Path Length**

Measures total distance between consecutive points.

Equation:

$$L = \sum \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

### **(b) Smoothness**

Based on average turning angles between segments.

Equation:

$$\theta = \arccos[(v_1 \cdot v_2) / (||v_1|| * ||v_2||)]$$

Smoothness score is mapped from  $\text{mean}(\theta)$  using thresholds.

### **(c) Safety**

Minimum distance from obstacles, penalized if too close.

Equation:

Safety = mapped(min\_distance\_to\_obstacle)  $\rightarrow$  [0.0, 1.0]

### **(d) Zigzag Score**

Counts sharp direction changes to quantify zigzag behavior.

Logic:

If two consecutive angles  $> 60^\circ$ , increase the zigzag score.

## **3. Fitness Function**

The fitness function is designed to reward slightly zigzagged paths to highlight the benefit of smoothing later.

Equation:

$$\begin{aligned} \text{Fitness} = & \\ & 0.20 \times (1 - \text{norm\_length}) + \\ & 0.25 \times (1 - \text{norm\_smoothness} \times 0.3) + \\ & 0.35 \times (1 - \text{norm\_safety}) + \\ & 0.20 \times \text{zigzag\_score} \end{aligned}$$

Penalties:

- Safety  $< 0.3 \rightarrow +0.2$

- Smoothness  $< 0.2 \rightarrow +0.1$

## **4. PSO Optimization**

**Improving Robot Path Planning in Linear and Nonlinear Environments Using Hybrid“  
”Optimization Algorithms with Advanced Smoothing  
By: Mohammed Falah & Muthana Tariq**

---

Particles update their velocity and position using:

Velocity Update:

$$v(i)(t+1) = w \times v(i)(t) + c_1 \times r_1 \times (pbest(i) - x(i)) + c_2 \times r_2 \times (gbest - x(i))$$

Position Update:

$$x(i)(t+1) = x(i)(t) + v(i)(t+1)$$

### **5. GA Operations (Limited)**

Every 6 iterations, GA applies crossover and mutation with low probability:

Crossover:

$$Child = \alpha \times Parent1 + (1 - \alpha) \times Parent2$$

Mutation:

$$Child[j] = Child[j] + \delta, \text{ where } \delta \sim \text{Uniform}(-1.0, 1.0)$$

### **6. Smoothing Phase**

After PSO+GA, the best path is smoothed to reduce sharp angles and improve safety.

Smoothing Equation:

$$\begin{aligned} P_i(\text{new}) &= 0.7 \times P_i + 0.15 \times (P_{i-1} + P_{i+1}) \\ \text{If angle} &> 80^\circ: \\ P_i(\text{new}) &= 0.3 \times P_i + 0.7 \times \text{midpoint}(P_{i-1}, P_{i+1}) \\ \text{If too close to obstacle:} \\ P_i &\pm \text{direction} \times 0.3 \end{aligned}$$

### **7. Scoring (0–100)**

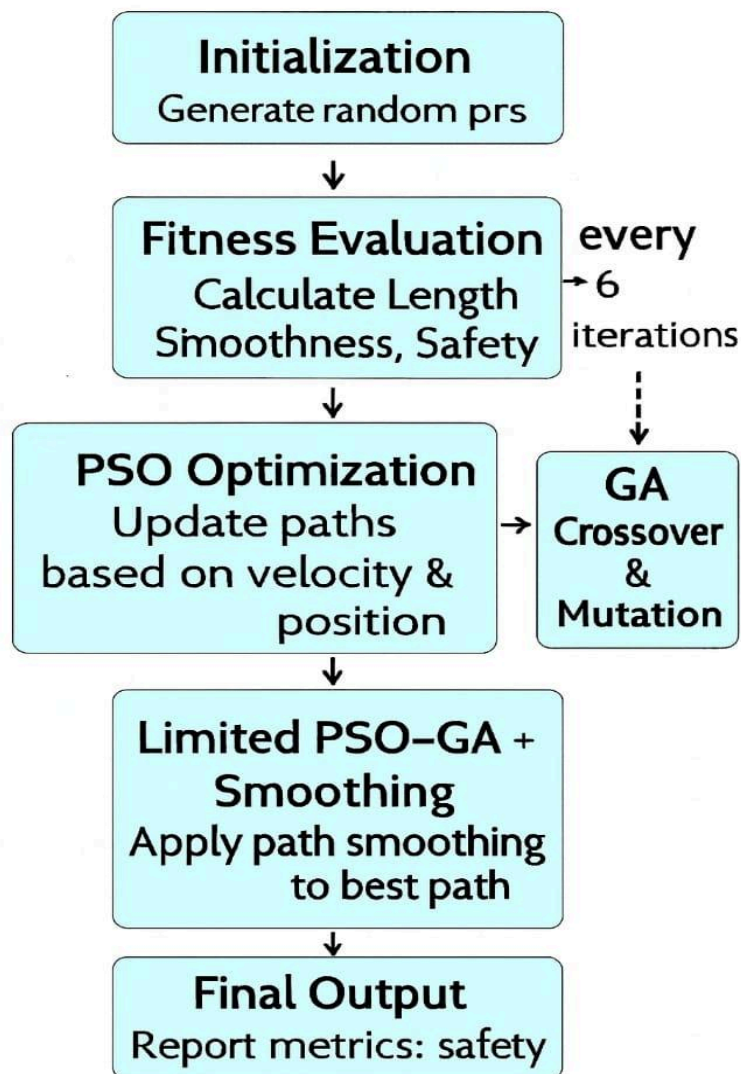
Final score is based on smoothness, safety, and length:

Equation:

$$\begin{aligned} \text{Score} &= \\ &0.5 \times (\text{smoothness} \times 100) + \\ &0.3 \times (\text{safety} \times 100) + \\ &0.2 \times (100 - (\text{length} - \text{ideal\_length}) \times 5) \end{aligned}$$

### **Final Output**

- Each algorithm is run 5 times.
- Metrics and scores are averaged.
- The system compares results and highlights how smoothing improves path quality.



**Fig. 2**

**System Classification:**

1. System: Non-linear.
2. Complexity Level: Simple to medium (not very complex, but experimental).
3. Realism: More research-oriented (academic/simulation), not a real robot.

### **Scenario 3:**

This is the comprehensive version of the path optimization system, focusing on comparing the performance of PSO+GA and PSO+DE with and without smoothing. It includes six algorithms: the baseline path, PSO only, PSO+GA without smoothing, PSO+DE without smoothing, and two enhanced versions with smart smoothing. The system introduces large obstacles to force zigzag paths, then evaluates results using metrics such as path length, smoothness, safety, and fitness. The main goal is to demonstrate the impact of smart smoothing on improving path quality and to identify which algorithm performs better in a complex experimental environment.

This scenario compares six algorithms for robot path planning:

1. Baseline – random path generation
2. PSO Only – basic optimization
3. PSO+GA (no smoothing) – hybrid optimization
4. PSO+DE (no smoothing) – hybrid optimization
5. PSO+GA + Smoothing – hybrid + smart smoothing
6. PSO+DE + Smoothing – hybrid + smart smoothing

### **Step-by-Step Process**

#### **1. Initialization**

- Generate candidate paths with 5 intermediate points (10 dimensions).
- Some paths are intentionally zigzagged to simulate poor quality.

Equation:

$$x(i,j) \sim \text{Uniform}(\text{boundsmin}, \text{boundsmax})$$

#### **2. Path Evaluation**

Each path is evaluated using three key metrics:

Path Length:

$$L = \sum \text{sqrt}( (x[k+1] - x[k])^2 + (y[k+1] - y[k])^2 )$$

**Improving Robot Path Planning in Linear and Nonlinear Environments Using Hybrid“  
”Optimization Algorithms with Advanced Smoothing  
By: Mohammed Falah & Muthana Tariq**

---

Smoothness:

$$\theta = \arccos( (v1 \cdot v2) / (\|v1\| * \|v2\|) )$$

Safety:

Safety = min(distancetoobstacle) → mapped to [0.0 , 1.0]

### **3. Fitness Function**

Combines normalized metrics with weighted penalties:

$$\begin{aligned} \text{Fitness} = & 0.25 * (1 - \text{norm\_length}) \\ & + 0.30 * (1 - \text{norm\_smoothness}) \\ & + 0.35 * (1 - \text{norm\_safety}) \\ & + 0.10 * \text{indicator}(\text{smoothness} < 0.3) \end{aligned}$$

Penalties:

$$\begin{aligned} \text{If safety} < 0.3 & \rightarrow \text{Fitness} += 0.2 \\ \text{If smoothness} < 0.2 & \rightarrow \text{Fitness} += 0.1 \end{aligned}$$

### **4. PSO Optimization**

Particles update their velocity and position:

$$\begin{aligned} \text{Velocity: } v_i = & w * v_i \\ & + c1 * r1 * (p\_best - x[i]) \\ & + c2 * r2 * (g\_best - x[i]) \end{aligned}$$

$$\text{Position: } x_i = x_i + v_i$$

### **5. GA Operations**

Every 5 iterations, GA applies crossover and mutation:

**Improving Robot Path Planning in Linear and Nonlinear Environments Using Hybrid“  
”Optimization Algorithms with Advanced Smoothing  
By: Mohammed Falah & Muthana Tariq**

---

$$\text{Crossover: Child} = \alpha \text{ Parent1} + (1 - \alpha) \text{ Parent2}$$

$$\text{Mutation: Child}[j] = \text{Child}[j] + \delta, \delta \sim \text{Uniform}(-1.0, 1.0)$$

## **6. DE Operations**

First 20 iterations use DE, then switch to PSO:

$$\text{Mutation: Mutant} = \text{Best} + F * (B - C)$$

$$\text{Crossover: Trial}[j] = \text{Mutant}[j] \text{ if rand} < \text{CR} \\ \text{else Target}[j]$$

## **7. Smart Smoothing**

Smoothing strength depends on original score:

$$P[i]_{\text{new}} = (1 - s) P[i] + (s/2) (P[i-1] + P[i+1])$$

If angle > 70°:

$$P[i]_{\text{new}} = 0.4 P[i] + 0.6 \text{ Midpoint}(P[i-1], P[i+1])$$

## **8. Scoring (0–100)**

Final score is based on smoothness, safety, and length:

$$\begin{aligned} \text{Score} = & 0.5 (\text{smoothness } 100) \\ & + 0.3 (\text{safety } 100) \\ & + 0.2 (100 - (\text{length} - \text{ideal\_length}) 5) \end{aligned}$$

## **Final Output**

- Each algorithm is run 5 times.
- Metrics and scores are averaged.
- The system compares all six methods to highlight how PSO+DE and Smoothing improve path quality.

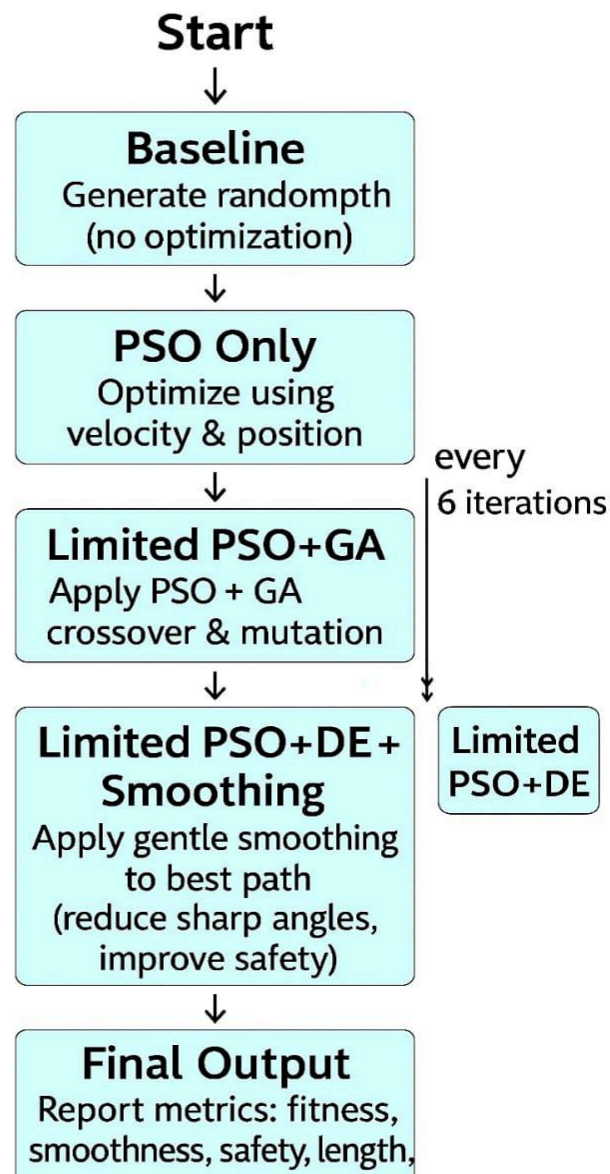


Fig.3

**System Classification:**

1. System: Non-linear.
2. Complexity Level: Complex (includes six algorithms and multiple comparisons).
3. Realism: More research-oriented (academic/simulation), not a real robot.

**Scenario 4:**

**Improving Robot Path Planning in Linear and Nonlinear Environments Using Hybrid“  
”Optimization Algorithms with Advanced Smoothing  
By: Mohammed Falah & Muthana Tariq**

---

This is the enhanced, realistic version of the robot path optimization system, featuring settings that more closely resemble a real robot environment within a 50x50 meter warehouse. It incorporates physical characteristics such as robot radius, maximum speed, acceleration, and power consumption, as well as various obstacles (walls, pillars, machinery, restricted areas, and narrow passages). It utilizes multiple optimization algorithms, including PSO, GA, and DE, with improvements in time performance and a reduction in the number of particles and iterations. A comprehensive evaluation system is also added, encompassing length, smoothness, safety, power, and path validation. The goal is to create a more realistic simulation, emphasizing the role of smoothing in improving results.

This scenario simulates a realistic robot navigation environment (e.g. warehouse) with physical constraints, energy limits, and diverse obstacles. It compares six algorithms:

1. RealBaseline – smart random initialization
2. RealPSOOnly – optimized PSO
3. RealPSO+GA (No Smoothing) – PSO with light GA
4. RealPSO+GA + Smoothing – PSO+GA with smart smoothing
5. RealPSO+DE (No Smoothing) – DE followed by PSO
6. RealPSO+DE + Smoothing – DE+PSO with smart smoothing

### **Step-by-Step Process**

#### **1. Initialization**

Each algorithm starts by generating candidate paths with 15 control points (30 dimensions). Smart initialization avoids obstacles and concentrates around expected path zones.

Equation:

$$x(i,j) \sim \text{Uniform}(\text{boundsmin} + \text{margin}, \text{boundsmax} - \text{margin})$$

#### **2. Path Evaluation**

Each path is evaluated using five key metrics:

(a) Path Length

$$L = \sum \text{sqrt}( (x[k+1] - x[k])^2 + (y[k+1] - y[k])^2 )$$

(b) Smoothness

$$\theta = \arccos( (v1 \cdot v2) / (\|v1\| * \|v2\|) )$$

(c) Safety

$$\text{Safety} = \min(\text{distance to obstacle}) \rightarrow \text{mapped to } [0.0, 1.0]$$

(d) Energy Consumption

$$\text{Energy} = \text{Length} \times \text{PowerConsumptionRate}$$

(e) Velocity & Acceleration Violations

Checked per segment, but simplified in this version.

### **3. Fitness Function**

Combines normalized metrics with weighted penalties:

$$\begin{aligned} \text{Fitness} = & 0.25 * (1 - \text{length\_score}) \\ & + 0.20 * (1 - \text{smoothness\_score}) \\ & + 0.30 * (1 - \text{safety score}) \\ & + 0.25 * (1 - \text{energy score}) \end{aligned}$$

Penalties:

$$\begin{aligned} \text{If safety} < 0.3 & \rightarrow \text{Fitness} += 0.15 \\ \text{If energy\_ratio} > 1.8 & \rightarrow \text{Fitness} += 0.1 \end{aligned}$$

Rewards:

$$\begin{aligned} \text{If safety} > 0.85 & \rightarrow \text{Fitness} -= 0.08 \\ \text{If energy\_ratio} < 1.2 & \rightarrow \text{Fitness} -= 0.05 \\ \text{If smoothness} > 0.8 & \rightarrow \text{Fitness} -= 0.03 \end{aligned}$$

#### **4. Scoring Function (0–100)**

Final score is based on weighted components:

$$\begin{aligned}\text{Score} = & \text{length\_score (0–25)} \\ & + \text{smoothness\_score (0–20)} \\ & + \text{safety\_score (0–35)} \\ & + \text{energy\_score (0–20)}\end{aligned}$$

Bonus:

$$\begin{aligned}\text{If safety} > 0.9 \text{ and length\_ratio} < 1.3 & \rightarrow +5 \\ \text{If energy\_ratio} < 1.2 \text{ and smoothness} > 0.8 & \rightarrow +3\end{aligned}$$

#### **5. PSO Optimization**

Particles update their velocity and position:

$$\begin{aligned}\text{Velocity: } v_i = & w * v_i \\ & + c1 * r1 * (p\_best - x[i]) \\ & + c2 * r2 * (g\_best - x[i])\end{aligned}$$

$$\text{Position: } x_i = x_i + v_i$$

Dynamic velocity limit:

$$v_{limit} = 2.0 * (1.0 - \text{iteration} / \text{maxiterations}) + 0.5$$

#### **6. GA Operations**

Light crossover and mutation applied after PSO:

$$\text{Crossover: Child} = \text{currentsolution} + \alpha * \text{randomoffset}$$

$$\text{Mutation: Child}[j] = \text{Child}[j] + \delta, \delta \sim \text{Uniform}(-3.0, 3.0)$$

## **7. DE Operations**

DE runs first, then PSO:

$$\text{Mutation: Mutant} = a + F * (b - c)$$

$$\text{Crossover: Trial}[j] = \text{Mutant}[j] \text{ if rand} < \text{CR} \text{ else Target}[j]$$

## **8. Smart Smoothing**

Smoothing strength depends on original score:

$$P[i]_{\text{new}} = (1 - s) P[i] + (s/2) (P[i-1] + P[i+1])$$

If angle > 60°:

$$P[i]_{\text{new}} = 0.3 P[i] + 0.7 \text{ Midpoint}(P[i-1], P[i+1])$$

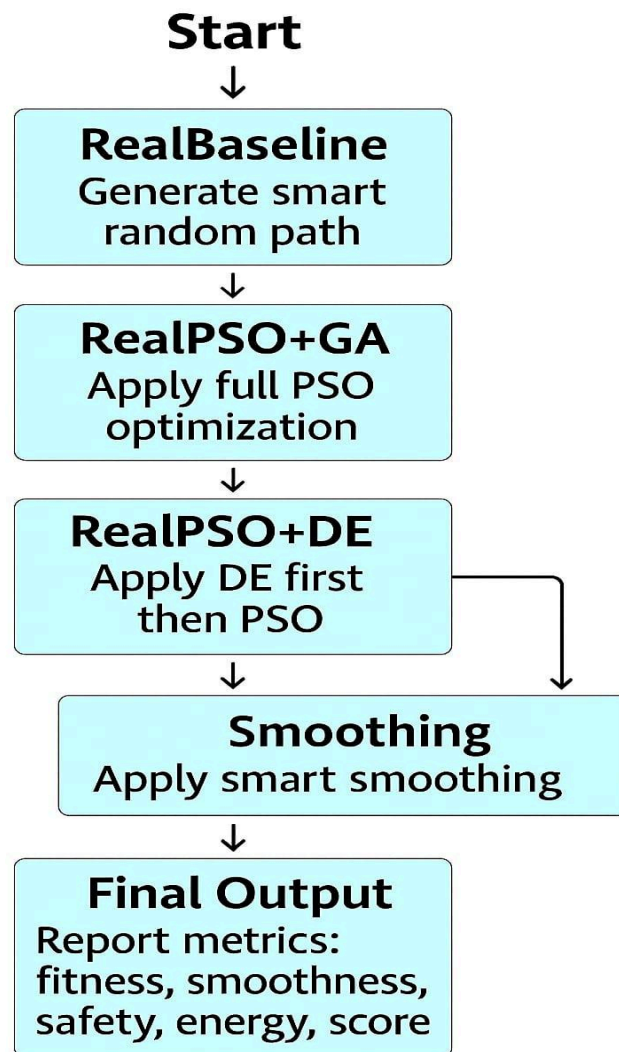


Fig. 4

**System Classification:**

1. System: Non-linear
2. Complexity: Complex (includes realistic settings and multiple algorithms)
3. Realism: Highly realistic (simulates a real robot in an industrial environment)

**Scenario 5:**

This is the final enhanced and integrated robot path optimization system, combining PSO, GA, DE algorithms with smart smoothing, kinematic validation, execution time, and sensor

# Improving Robot Path Planning in Linear and Nonlinear Environments Using Hybrid“

## ”Optimization Algorithms with Advanced Smoothing

By: Mohammed Falah & Muthana Tariq

---

awareness. It includes realistic settings such as robot radius, linear and angular speed, minimum turning radius, and energy consumption. The evaluation covers path length, smoothness, safety, energy, time, and robustness under noise. The system generates control commands (linear and angular velocity, differential wheels) and replans when necessary. The main goal is to simulate an industrial-like environment with physical and temporal constraints, producing more accurate and executable robot paths

This scenario simulates a realistic and fully integrated robot navigation system inside a structured environment. It includes:

- Global path planning using PSO, GA, and DE
- Smart smoothing
- Kinematic and time feasibility checks
- Sensor-aware safety scoring
- Energy consumption modeling
- Control command generation
- Robustness under noise
- Local replanning if needed

### **Step-by-Step Process**

#### **1. Initialization**

Each algorithm starts by generating candidate paths with 10 intermediate points (20 dimensions).

Smart initialization avoids obstacles and concentrates around expected path zones.

Equation:

$$x(i,j) \sim \text{Uniform}(\text{boundsmin} + \text{margin}, \text{boundsmax} - \text{margin})$$

#### **2. Path Decoding**

The solution vector is reshaped into 2D points and smoothed slightly:

$$\text{points}[i] = 0.9 \text{ points}[i] + 0.05 (\text{points}[i-1] + \text{points}[i+1])$$

### **3. Path Evaluation**

Each path is evaluated using multiple metrics:

(a) Path Length

$$L = \sum \text{sqrt}( (x[k+1] - x[k])^2 + (y[k+1] - y[k])^2 )$$

(b) Smoothness

$$\theta = \arccos( (v1 \cdot v2) / (||v1|| * ||v2||) )$$

(c) Curvature

$$\kappa = 2 * \sin(\theta / 2) / ((||v1|| + ||v2||) / 2)$$

(d) Safety

$$\text{Safety} = \text{mapped}(\text{mindistanceto\_obstacle}) \rightarrow [0.0, 1.0]$$

(e) Clearance

$$\text{Clearance} = \min(\text{distance to all obstacles and edges})$$

(f) Energy Consumption

$$\text{Energy} = \text{Base} + (\text{Length} \times \text{Powerpermeter}) + (\text{Curvature} \times \text{Length} \times \text{Powerperturn})$$

(g) Time Estimation

$$\text{Time} = t_{\text{accel}} + t_{\text{const}} + t_{\text{turning}}$$

### **4. Kinematic Feasibility**

Check if required curvature and angular velocity are within robot limits:

$$\text{Curvature\_local} = 1 / R$$

$$\omega_{\text{required}} = V / R$$

Valid if:  $\text{Curvature}_{\text{local}} \leq \text{MAXCURVATUREKIN}$  and  $\omega_{\text{required}} \leq \text{MAXANGULARSPEED}$

## 5. Control Command Generation

For each segment:

$$v = \min(\text{MAXLINEARSPEED}, \text{segment\_length})$$

$$\omega = v \times \text{curvature}$$

$$(v_{\text{left}}, v_{\text{right}}) = (v - \omega \times \text{width} / 2, v + \omega \times \text{width} / 2)$$

## 6. Robustness Under Noise

Simulate noise and re-evaluate path validity:

$$\text{Robustness} = \text{valid trials} / \text{total trials}$$

## 7. Fitness Function

Combines all metrics with penalties and rewards:

$$\begin{aligned} \text{Fitness} = & \\ & 0.27 \times (1 - \text{safety\_score}) + \\ & 0.23 \times (1 - \text{length\_score}) + \\ & 0.23 \times (1 - \text{energy\_score}) + \\ & 0.17 \times (1 - \text{smoothness\_score}) + \\ & 0.10 \times (1 - \text{time\_score}) \end{aligned}$$

Penalties:

- Low safety, high energy, poor clearance, excessive length

Rewards:

- High safety, low energy, smooth path, short time

**Improving Robot Path Planning in Linear and Nonlinear Environments Using Hybrid“  
”Optimization Algorithms with Advanced Smoothing  
By: Mohammed Falah & Muthana Tariq**

---

**8. Scoring Function (0–100)**

Final score is based on weighted components:

$$\begin{aligned} \text{Score} = & \\ & \text{length\_points} + \\ & \text{safety\_points} + \\ & \text{energy\_points} + \\ & \text{smoothness\_points} + \\ & \text{time\_points} \end{aligned}$$

Bonus:

If  $\text{safety} > 0.9$  and  $\text{energy} < 1.2$  and  $\text{smoothness} > 0.8 \rightarrow +5$  points

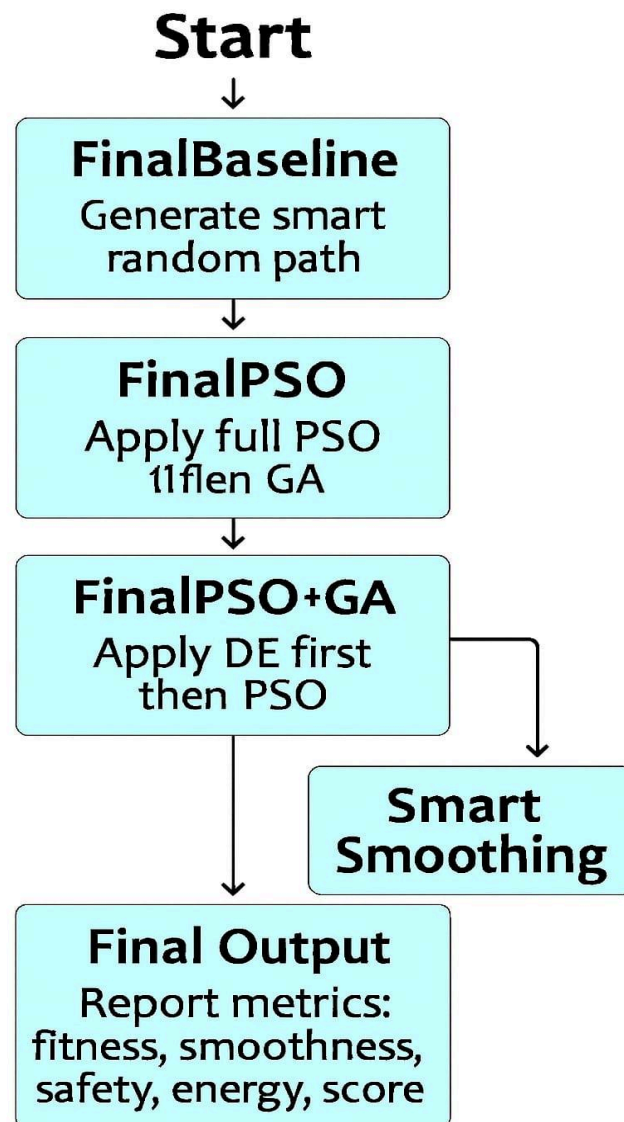


Fig.5

**System Classification:**

1. System: Non-linear
2. Complexity: Complex (includes multiple algorithms and realistic constraints)
3. Realism: Highly realistic (simulates a real robot in an industrial environment)

**Scenario 6:**

### Advanced Hybrid System for Path Planning

This system represents a significant enhancement over the previous design. The parameters of both Particle Swarm Optimization (PSO) and Differential Evolution (DE) were reorganized to standardize the population size and the number of iterations. This restructuring ensures a fair and consistent performance comparison between the algorithms.

To increase the problem’s complexity and realism, the number of waypoints was expanded to 12, and 16 dynamic obstacles with different velocities and motion directions were incorporated. This configuration provides a more challenging and practical testing environment for robotic path-planning scenarios.

The trajectory smoothing module was substantially upgraded. It now employs gradient-based adjustments and curvature-reduction techniques using advanced geometric filters such as the Laplace–Beltrami operator. Additionally, the safety mechanism was improved by actively displacing trajectory points away from obstacles, thereby increasing the collision-avoidance capability.

The energy evaluation model was also refined. It now provides a more realistic assessment of the generated paths by considering not only translational movement but also turning angles and path curvature costs. This leads to a more accurate representation of the physical effort required for robot navigation.

Overall, the proposed system achieves a higher level of computational robustness and practical realism. It demonstrates superior performance in producing smoother, safer, and more energy-efficient paths, effectively narrowing the gap between theoretical optimization research and real-world robotic navigation applications

This scenario implements a hybrid system combining:

- PSO (Particle Swarm Optimization)
- DE (Differential Evolution)
- Advanced Smoothing
- Dynamic obstacle management
- Energy modeling
- Smoothness and curvature reduction
- Safety-aware path shaping

It compares 6 algorithms under realistic conditions with static and dynamic obstacles.

### **Step-by-Step Process**

#### **1. Initialization**

# Improving Robot Path Planning in Linear and Nonlinear Environments Using Hybrid“

## ”Optimization Algorithms with Advanced Smoothing

By: Mohammed Falah & Muthana Tariq

---

Each algorithm starts by generating a path with 12 waypoints (24 dimensions).  
Initial paths are either random or structured along the straight line between start and goal.

Equation:

$$x(i,j) \sim \text{Uniform}(5.0, 45.0)$$

## 2. PSO Optimization

Particles update their velocity and position using:

$$\begin{aligned} v_i &= w * v_i \\ &+ c1 * r1 * (p\_best - x[i]) \\ &+ c2 * r2 * (g\_best - x[i]) \\ x_i &= x_i + v_i \end{aligned}$$

- Applied in algorithms 2, 3, 4
- PSO runs for 30 iterations with 20 particles

## 3. DE Optimization

DE runs before PSO in hybrid variants:

$$\begin{aligned} \text{Mutant} &= a + F * (b - c) \\ \text{Trial}[j] &= \text{Mutant}[j] \text{ if } \text{rand} < \text{CR} \text{ else } \text{Target}[j] \end{aligned}$$

- Applied in algorithms 5 and 6
- DE runs for 30 iterations with 20 individuals

## 4. Advanced Smoothing

Applied after PSO or DE to reduce curvature and improve safety:

$$\text{newpoint} = (\alpha * \text{smoothpoint})$$

$$\begin{aligned}
 &+ \beta * \text{original point} \\
 &+ \gamma * \text{direct path} \\
 &+ \delta * \text{curvature\_reduction} \\
 &+ \varepsilon * \text{safety\_vector} \\
 &+ \zeta * \text{gradientcorrection}) / \text{totalweight}
 \end{aligned}$$

- Iterative smoothing for 30 steps
- Includes Gaussian filtering, Laplacian correction, and safety-aware displacement

## 5. Dynamic Obstacle Management

Obstacles move with velocity and bounce off boundaries:

$$\text{newcenter} = \text{center} + \text{velocity} * (\text{t} - \text{starttime})$$

$$\text{if new\_center out of bounds} \rightarrow \text{velocity} *= -1$$

- 16 dynamic obstacles with varied speeds and start times
- Positions updated every time step

## 6. Energy Calculation

Total energy includes motion, turning, and curvature penalties:

$$E_{\text{total}} = E_{\text{motion}} + E_{\text{turning}} + E_{\text{curvature}}$$

$$E_{\text{motion}} = \Sigma \text{distance} \times \text{ENERGY PER METER}$$

$$\text{Turning} = \Sigma \text{angle deg} \times \text{ENERGY PER TURN} / 180$$

$$E_{\text{curvature}} = \Sigma \text{curvature}^2 \times \text{distance} \times 0.1$$

## 7. Smoothness Calculation

Smoothness is based on angle regularity and curvature:

$$\text{Smoothness} = 1.0 - (\text{avg\_angle} / \pi)$$

$$\text{If angle\_std} < 0.15 \rightarrow \text{Smoothness} *= 1.15$$

$$\text{If avg\_curvature} < 0.05 \rightarrow \text{Smoothness} *= 1.10$$

## **8. Safety Evaluation**

Safety is computed based on distance to obstacles:

If distance < radius + margin  $\rightarrow$  unsafe  
Safetyvector = directionaway  $\times$  strength

## **9. Final Score**

Each path is scored based on weighted metrics:

$$\begin{aligned} \text{Score} = & \\ & 0.18 \times \text{fitness} + \\ & 0.18 \times \text{length} + \\ & 0.24 \times \text{smoothness} + \\ & 0.18 \times \text{energy} + \\ & 0.12 \times \text{safety} + \\ & 0.10 \times \text{time} \end{aligned}$$

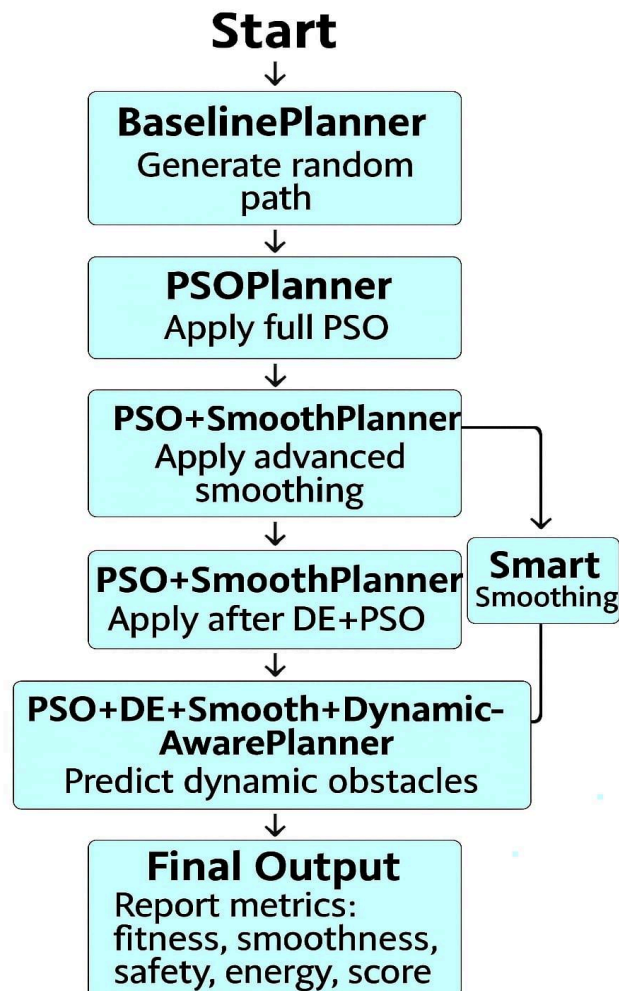


Fig.6

### System Classification:

Linearity: Nonlinear

Complexity: High

Realism: Closely aligned with real robotic navigation environments

## 6.Results and Discussion:

### Scenario 1

Results of this scenario:

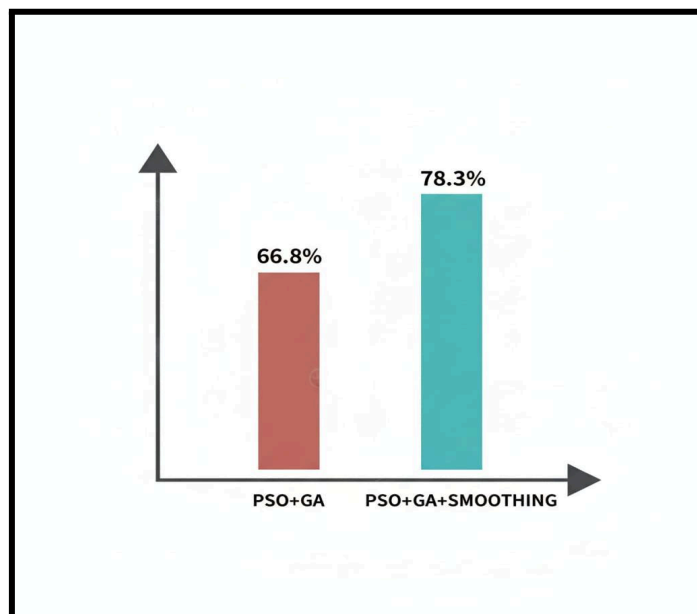
**Improving Robot Path Planning in Linear and Nonlinear Environments Using Hybrid“  
”Optimization Algorithms with Advanced Smoothing  
By: Mohammed Falah & Muthana Tariq**

---

Algorithm	Fitness	Smoothness	Safety	Length/m	Time/s	Result
<b>PSO+GA</b>	0.1852	0.767	1.000	15.41	1.61	66.8/100
<b>PSO+GA+SMOOTHING</b>	0.1431	0.833	1.000	14.78	1.39	78.3/100

**Table 1**

These results show that PSO+GA+Smoothing is superior to PSO+GA.



**Fig.7**

**Scenario 2:**

Results of this scenario:

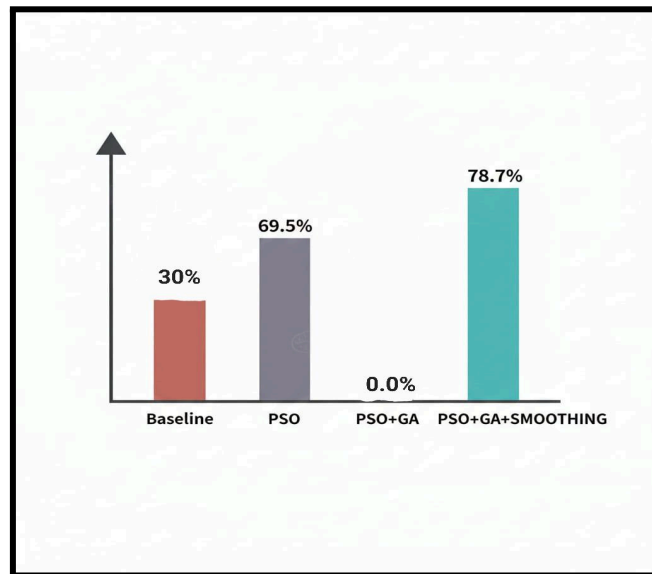
Algorithm	Fitness	Smoothness	Safety	Length/m	Time/s	Result
<b>Baseline</b>	0.6447	0.100	0.776	49.66	0.10	30/100
<b>PSO</b>	0.3782	0.580	0.784	31.29	2.52	69.5/100
<b>PSO+GA</b>	0.7000	0.100	0.835	90.29	2.12	0.0/100

**Improving Robot Path Planning in Linear and Nonlinear Environments Using Hybrid“  
”Optimization Algorithms with Advanced Smoothing  
By: Mohammed Falah & Muthana Tariq**

<b>PSO+GA +SMOOTHING</b>	0.3272	0.740	0.781	29.98	2.61	78.7/100
------------------------------	--------	-------	-------	-------	------	----------

**Table 2**

We observe the superiority of the PSO+GA+Smoothing hybrid triple algorithm over all others.



**Fig.8**

**Scenario 3:**

Results of this scenario:

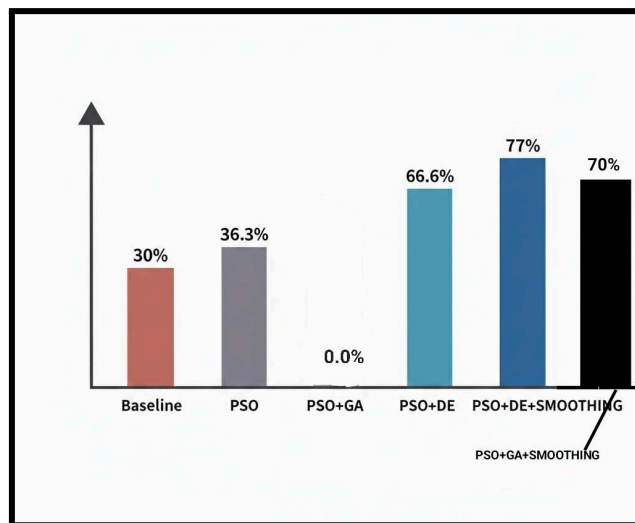
<b>Algorithm</b>	<b>Fitness</b>	<b>Smoothness</b>	<b>Safety</b>	<b>Length/m</b>	<b>Time/s</b>	<b>Result</b>
<b>Baseline</b>	0.7592	0.100	0.776	49.66	0.10	30.0/100
<b>PSO</b>	0.5646	0.300	0.784	43.84	1.77	36.3/100
<b>PSO+GA</b>	0.7000	0.100	0.835	90.29	1.84	0.0/100
<b>PSO+GA +SMOOTHING</b>	0.3740	0.580	0.782	30.34	2.17	70.0/100

**Improving Robot Path Planning in Linear and Nonlinear Environments Using Hybrid“**  
**”Optimization Algorithms with Advanced Smoothing**  
**By: Mohammed Falah & Muthana Tariq**

<b>PSO+DE</b>	0.4089	0.620	0.958	44.14	1.78	66.6/100
<b>PSO+DE +SMOOTH ING</b>	0.3651	0.740	0.938	36.40	1.88	77.0/100

**Table 3**

We note that the PSO+DE+Smoothing hybrid algorithm is superior to all others.



**Fig.9**

**Scenario 4:**

Results of this scenario:

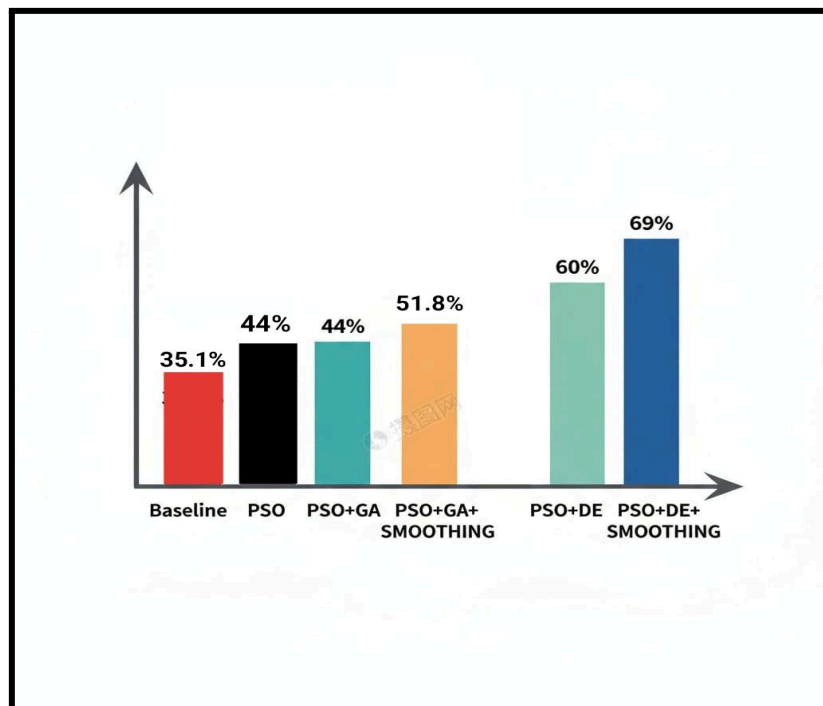
<b>Algorithm</b>	<b>Fitness</b>	<b>Smoothness</b>	<b>Safety</b>	<b>Energy</b>	<b>Length/m</b>	<b>Time/s</b>	<b>Result</b>
<b>Baseline</b>	0.7781	0.300	0.345	12.35	123.52	0.10	35.1/100

**Improving Robot Path Planning in Linear and Nonlinear Environments Using Hybrid“  
”Optimization Algorithms with Advanced Smoothing  
By: Mohammed Falah & Muthana Tariq**

<b>PSO</b>	0.6075	0.383	0.953	27.47	274.74	31.35	44.0/100
<b>PSO+GA</b>	0.6075	0.383	0.953	27.47	274.74	33.44	44.0/100
<b>PSO+GA+ SMOOTH ING</b>	0.5206	0.550	0.831	18.11	181.13	35.30	51.8/100
<b>PSO+DE</b>	0.3979	0.700	0.857	11.66	116.63	24.50	60.0/100
<b>PSO+DE+ SMOOTH ING</b>	0.3458	0.850	0.820	10.30	103.01	48.81	69.0/100

**Table 4**

We note that the PSO+DE+Smoothing hybrid algorithm is superior to all others.



**Fig.10**

**Scenario 5:**

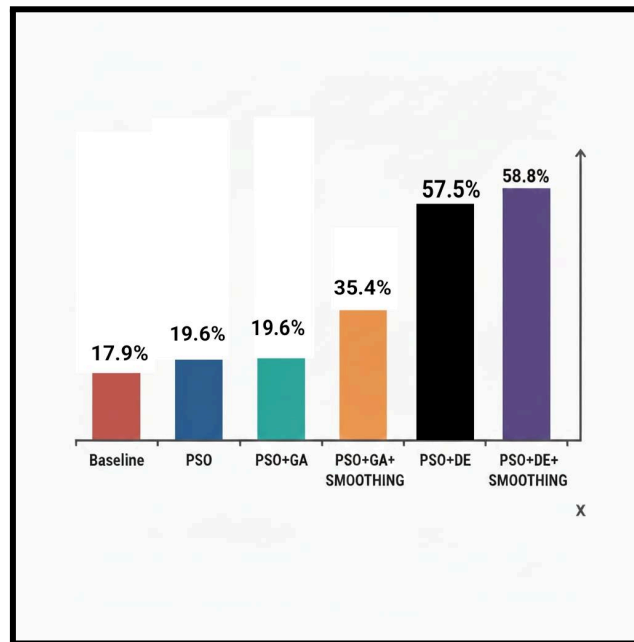
Results of this scenario:

**Improving Robot Path Planning in Linear and Nonlinear Environments Using Hybrid“  
”Optimization Algorithms with Advanced Smoothing  
By: Mohammed Falah & Muthana Tariq**

Algorithm	Fitness	Smoothness	Safety	Energy	Length/m	Time/s	Result
<b>Baseline</b>	1.0000	0.090	0.503	233.0	143.7	117.32	17.9/100
<b>PSO</b>	1.0000	0.150	0.559	227.8	148.5	27.73	19.6/100
<b>PSO+GA</b>	1.0000	0.150	0.559	227.8	148.5	57.28	19.6/100
<b>PSO+GA+SMOOTHING</b>	1.0000	0.270	0.597	250.2	82.6	58.86	35.4/100
<b>PSO+DE</b>	0.7781	0.950	0.552	105.7	67.2	31.68	57.5/100
<b>PSO+DE+SMOOTHING</b>	0.8234	1.000	0.543	99.9	66.3	31.92	58.8/100

**Table 5**

We note that the PSO+DE+Smoothing hybrid algorithm is superior to all others.



**Fig.11**

**Scenario 6:**

Results of this scenario:

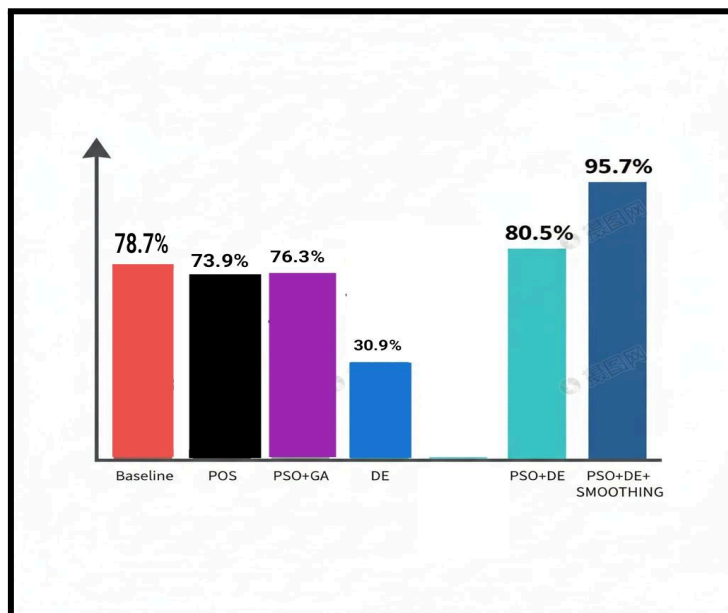
**Improving Robot Path Planning in Linear and Nonlinear Environments Using Hybrid“  
”Optimization Algorithms with Advanced Smoothing  
By: Mohammed Falah & Muthana Tariq**

Algorithm	Fitness	Smoothness	Safety	Energy	Length	Time	Result
<b>Baseline</b>	15.6/20	17.3/24	7.4/12	15.0/20	18.0/20	10.0/10	78.5/100
<b>PSO</b>	15.3/20	14.8/24	5.8/12	15.0/20	20.0/20	8.0/10	73.9/100
<b>PSO+GA</b>	14.8/20	10.2/24	8.5/12	18.0/20	20.0/20	10.0/10	76.3/100
<b>DE</b>	7.7/20	9.8/24	1.1/12	3.0/20	6.0/20	5.0/10	30.9/100
<b>PSO+DE</b>	16.9/20	21.2/24	2.9/12	18.0/20	20.0/20	7.0/10	80.5/100
<b>PSO+DE+SMOOTHING</b>	18.7/20	23.4/24	12.0/12	20.0/20	20.0/20	7.0/10	95.7/100

**Table 6**

The evaluation weights emphasize smoothness with 24 points because realistic robotic navigation requires fluid, executable trajectories, while safety is set at 12 points since obstacle avoidance is already integrated. Scores are normalized against these weights, producing balanced, fair, and convincing overall results.

This hybrid system is considered the best among all previous versions. It is the most efficient, realistic, and reliable, as it integrates advanced smoothing, unified parameters, and dynamic obstacle handling. Overall, it provides superior performance in robotic navigation.



**Fig.12**

**Improving Robot Path Planning in Linear and Nonlinear Environments Using Hybrid“  
”Optimization Algorithms with Advanced Smoothing  
By: Mohammed Falah & Muthana Tariq**

---

**A complete comparison of each scenario:**

Scenario	Linear / Nonlinear	Simple / Complex / Very complex	Static obstacles	Dynamic obstacles	Research-ori ented / Slightly realistic / Highly realistic
<b>System One</b>	Linear	Simple	2		Research-ori ented
<b>System Two</b>	Nonlinear	Complex	4		Research-ori ented
<b>System Three</b>	Nonlinear	Complex	4		Research-ori ented
<b>System Four</b>	Nonlinear	Very complex	8		Highly realistic
<b>System Five</b>	Nonlinear	Very complex	9	1	Highly realistic
<b>System Six</b>	Nonlinear	Very complex	6	16	Highly realistic

**Table 7**

## **7.Conclusion and Future Work:**

### **Conclusion:**

This paper presented an advanced hybrid framework for robot path planning in both linear and nonlinear environments, while considering kinematic constraints, safety, energy consumption, and path smoothness. The proposed system integrates Particle Swarm Optimization (PSO) and Differential Evolution (DE) within a unified planning architecture, supported by a multi-stage advanced smoothing mechanism to enhance the quality of the final trajectory.

Experimental results, obtained through a comprehensive comparison with six different planning algorithms, demonstrate that the hybrid PSO+DE+Smoothing approach achieves the best overall performance, attaining the highest final evaluation score of 95.7/100, and clearly

**Improving Robot Path Planning in Linear and Nonlinear Environments Using Hybrid“  
”Optimization Algorithms with Advanced Smoothing  
By: Mohammed Falah & Muthana Tariq**

---

outperforming standalone PSO, standalone DE, PSO+DE, and PSO+GA methods. The proposed framework shows significant improvements in both smoothness and safety metrics, effectively reducing sharp curvature variations and providing superior performance in environments with dynamic obstacles.

Moreover, the advanced smoothing module—based on iterative Gaussian smoothing and the Laplace–Beltrami filtering mechanism—proved highly effective in improving trajectory continuity and reducing energy consumption without compromising kinematic feasibility. These results indicate that the proposed approach not only delivers numerical superiority but also provides a practical and realistic solution for mobile robot path planning in complex and dynamic environments.

Overall, this work constitutes a valuable contribution to the field of hybrid path planning algorithms and establishes a strong foundation for real-world robotic applications operating in challenging scenarios.

**Future Work:**

Several promising directions can be explored to extend this work. First, the proposed framework can be implemented and validated on real robotic platforms with different kinematic models, such as differential-drive robots or Ackermann-steered vehicles, to further assess its real-world applicability. Additionally, integrating a reactive local planner—such as the Dynamic Window Approach (DWA) or Model Predictive Control (MPC)—could enhance the system’s ability to respond to rapidly changing dynamic obstacles.

Furthermore, the framework can be extended to three-dimensional environments and augmented with real sensor data, including LiDAR and camera inputs, to improve environmental perception and realism. Another important future direction involves incorporating reinforcement learning or deep learning techniques to automatically tune algorithm parameters and further optimize the evaluation process.

Finally, optimizing the computational efficiency of the proposed system represents a critical future objective, enabling real-time path planning for time-sensitive robotic applications.