This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3504360

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.0429000

New design of smooth PSO-IPF navigator with kinematic constraints

MAHSA MOHAGHEGH[1,2] HEDIEH JAFARPOURDAVATGAR[3], and Samaneh-Alsadat Saeedinia.[4],

[1]School of Engineering, Computing and Mathematical Sciences, Auckland University of Technology (AUT), Auckland, New Zealand (e-mail: mahsa.mohaghegh@aut.ac.nz)

[2]Faculty of Design and Creative Technologies, AUT, Auckland, New Zealand

[3]Control Engineering Department, Amirkabir University, Tehran, Iran

[4]Complex System Laboratory, Iran University of Science and Technology

Corresponding author: Mahsa Mohaghegh (e-mail: mahsa.mohaghegh@aut.ac.nz).

ABSTRACT Robotic applications across industries demand advanced navigation for safe and smooth movement. Smooth path planning is crucial for mobile robots to ensure stable and efficient navigation, as it minimizes jerky movements and enhances overall performance Achieving this requires smooth collision-free paths. Partial Swarm Optimization (PSO) and Potential Field (PF) are notable path-planning techniques, however, they may struggle to produce smooth paths due to their inherent algorithms, potentially leading to suboptimal robot motion and increased energy consumption. In addition, while PSO efficiently explores solution spaces, it generates long paths and has limited global search. On the contrary, PF methods offer concise paths but struggle with distant targets or obstacles. To address this, we propose Smoothed Partial Swarm Optimization with Improved Potential Field (SPSO-IPF), combining both approaches and it is capable of generating a smooth and safe path. Our research demonstrates SPSO-IPF's superiority, proving its effectiveness in static and dynamic environments compared to a mere PSO or a mere PF approach.

INDEX TERMS Path Planning, Path smoothness, Kinematic Constraint, Collision Avoidance, Swarm Optimization, Improved Potential field.

I. INTRODUCTION

ROBOTIC navigation in dynamic and uncertain environments poses formidable challenges, demanding the development of efficient and adaptable algorithms. Optimization algorithms are indispensable in mobile robot navigation, facilitating the discovery of optimal solutions to intricate problems. These algorithms empower robots to meticulously plan their paths, accurately localize themselves, and dynamically avoid obstacles in real-time, thereby significantly enhancing navigation performance [1].

Among the plethora of optimization methodologies, Particle Swarm Optimization (PSO) and Potential Fields (PF) stand out as widely utilized approaches with extensive applications in mobile robot navigation [2]. PSO and PF algorithms represent two significant methodologies in navigation and optimization tasks, particularly in scenarios characterized by complex constraints and the necessity for energy consumption minimization. Integrating these approaches leverages the strengths of each to provide a robust solution for multifaceted navigation problems [3].

PSO, inspired by social behaviors observed in nature, excels at exploring and exploiting search spaces, making it ideal for finding optimal solutions in dynamic, multi-dimensional environments [4]. However, this method faces challenges in path planning, especially in scenarios with narrow passages or complex environments. These challenges include difficulty in avoiding obstacles, ensuring smooth trajectories, and maintaining computational efficiency. Researchers have proposed innovative solutions to address these issues [5] [6] [7] [10].

Yang et al. introduced an improved PSO (IPSO) to address the limitations of standard PSO, particularly in dynamic environments. This IPSO demonstrated better performance in avoiding premature convergence and finding optimal paths [8]. Zheng et al. developed an artificial potential field-based PSO (apfrPSO) that combines the strengths of both methods, resulting in more effective path planning and obstacle avoidance [9].The Artificial PF (APF) method is widely used for real-time obstacle avoidance and path smoothing. Despite its advantages, classical APF methods are prone to local minima and oscillations near obstacles. Melchiorre et al. proposed enhancements to the APF approach by integrating local optimization techniques, which improved the method's ability to navigate in dynamic environments without getting trapped in local minima [11]. Similarly, other studies delving into the application of PSO and APF optimization algorithms in mobile robot navigation, underscore their pivotal role in path planning, environmental adaptability, and navigation efficiency [12]–[14]. Within these algorithms, particles traverse the search space to ascertain the optimal path, typically defined by a fitness function. However, despite its efficacy in finding optimal paths, PSO often grapples with generating smooth trajectories [15].This lack of smoothness can result in jerky or discontinuous motions, compromising path traversal efficiency and potentially destabilizing the system. In contrast, PF emerges as a probabilistic filtering technique adept at estimating the state of dynamic systems based on noisy sensor measurements [8].

In PF path planning, a single robot or agent moves through the environment based on attractive and repulsive forces. The potential field is usually designed to have a global minimum at the goal, attracting the robot towards the target [17], [18]. However, maintaining smooth paths can be challenging for potential field methods, especially when the robot deviates significantly from the target [19]. This issue arises due to the potential field's inherent tendency to excessively attract the robot towards obstacles, which can lead to discontinuous, non-smooth paths [20]. Moreover, the resampling process in PF can inadvertently introduce variability and diminish path smoothness [8]. To alleviate these concerns, researchers have proposed modifications to the artificial potential field approach, introducing novel sets of potential fields [21]. Techniques such as deterministic annealing [22], wavefront planning [23], and navigation functions [24] can help address these limitations and improve the overall performance and safety of PF-based path planning.

One fundamental hurdle of navigation in dynamic and uncertain environments using PSO and PF lies in balancing exploration and exploitation [16]. While exploration seeks new and potentially better paths, the dominance of this factor can yield erratic and disorganized paths. Conversely, an overemphasis on exploitation may trap the search in local optima, resulting in suboptimal trajectories. Thus, meticulous tuning of this trade-off is pivotal for both path efficiency and smoothness.

Furthermore, the discrete representation of paths in PSO and PF, often grid-based, introduces artifacts and discontinuities [18]. Researchers propose solutions like higher-resolution grids or alternative spatial representations such as curves or splines to mitigate this challenge [18], [25]. However, these approaches may introduce additional computational complexity and trade-offs between smoothness, accuracy, and real-time performance. Moreover, the complexity and dynamic nature of environments significantly impact path smoothness [26], [27]. PSO and PF may struggle in complex environments with numerous obstacles or narrow passages due to limited exploration capabilities. In highly dynamic environments, rapid changes may render generated paths obsolete, necessitating abrupt corrections or replanning.

Integration of PF and PSO techniques emerges as a promising solution for optimal path planning, synergizing their strengths to enhance mobile robot navigation capabilities. This integration offers potential solutions to address the challenges of exploration-exploitation balance, path representation, and environmental dynamics, ultimately facilitating smoother and more efficient navigation [9].

Kinematic constraints play a crucial role in robotic systems, affecting their performance in various ways. These constraints, including joint limits, self-collision avoidance, and torque limits, significantly impact the movement and control of robots [28]–[30]. Understanding and managing these limitations are essential for optimizing the kinematic movement of robots and ensuring their overall effectiveness in various applications.

Considering kinematic constraints in path planning is crucial because it ensures that the planned paths are feasible for the actual movement of vehicles, especially in scenarios where traditional methods treat vehicles as point masses, neglecting constraints like turning radius [31]. Failures in path planning using PSO and Potential Fields can occur when the planned path contains inflection points with large curvatures that exceed the vehicle's performance capabilities, leading to 'planning-failure' [32]. Addressing kinematic constraints helps avoid local minima in PSO potential fields path planning, particularly when the target is near obstacles, making the points unreachable [32].

By integrating kinematic constraints into path planning algorithms, such as the improved artificial potential field method, the feasibility and effectiveness of the planned paths can be significantly enhanced, ultimately improving the overall performance of autonomous vehicles [34]. Previous research has addressed kinematic constraint issues in path planning using Potential Fields and Particle Swarm Optimization (PSO) algorithms. The integration of Artificial Potential Field (APF) methods with kinematic-based control systems has been proposed to overcome the limitations of assuming robots as point masses and neglecting nonholonomic constraints [35], [36].

Our primary contribution is the fusion of PSO with an Improved PF (IPF) methodology, accompanied by the design of an innovative objective function customized for seamless path planning while taking into account the robot's kinematic restrictions on rotational angles and smoothness of the generated path. By explicitly integrating these kinematic constraints into the repulsive and attractive forces of the proposed IPF technique, coupled with the optimization capabilities of PSO, we have propelled the advancement of heuristic approaches in path planning. The original fitness function we have devised accommodates these aspects, ensuring the creation of secure and fluid trajectories in both stationary and changing environments, as evidenced by simulation investigations. While some existing techniques may address smooth path creation through diverse curve-fitting methodologies [36], [37], few have integrated rotational constraints alongside kinematic limitations to enable swift, real-time path planning. Moreover, our strategy employs the IPF approach to tackle the issue of long-range target path planning, a significant drawback of traditional PF methods. Incorporating kinematic constraints like turning radius, angular velocity boundaries, and maximum velocity into the path planning procedure enables our algorithm to produce realistic trajectories that the robot can execute dependably. This, in conjunction with the optimization capabilities of PSO, results in notable enhancements in the safety, fluidity, and effectiveness of the planned trajectories, thus constituting a valuable addition to autonomous navigation research. The integration of kinematic constraints, such as turning radius and angular velocity limits, into the path planning process allows our algorithm to generate feasible trajectories that can be reliably executed by the robot. This, combined with the optimization capabilities of PSO, leads to significant improvements in the safety, smoothness, and efficiency of the planned paths, making it a valuable contribution to the field of autonomous navigation. The paper is organized into several sections to systematically present the research findings. The Materials and Methods section details the proposed approach of combining Smooth PSO (SPSO) with IPF (SPSO-IPF), to enhance path planning smoothness in mobile robots. This section outlines the modifications made to the objective function to prioritize smoothness and describes the experimental setup and parameters used for simulation. The Results section presents the simulation outcomes, showcasing the effectiveness of the SPSO-IPF approach in generating smoother paths compared to traditional PSO and PF solitary methods, and two state-of-the-art hybrid PSO approaches. Simulation results are illustrated and discussed comprehensively.

II. MATERIALS AND METHODS

In this section, we present the methodology employed for path planning, focusing on a novel approach integrating Smooth Particle Swarm Optimization (PSO) with an Improved Potential Fields (IPF) method. This approach is designed to address the challenges of path planning for mobile robots while considering kinematic constraints. The integration of PSO and IPF aims to generate smooth and collision-free paths in dynamic environments, ensuring optimal trajectory lengths. The smooth PSO-IPF approach is a result of advancements in optimization techniques, particularly suited for geometric path planning in autonomous and robotics applications. By incorporating kinematic constraints, the proposed approach seeks to improve the efficiency and safety of mobile robot navigation. The following sections outline the key components of the SPSO-IPF approach, detailing its formulation and implementation to achieve robust and effective path-planning solutions under dynamic conditions.

A. PROBLEM PRIORITIES AND CONSTRAINTS

To design an effective path planning algorithm, it is crucial to identify the problem priorities and constraints. In this study, the following problem priorities are considered:

1) **Ensure collision-free movement** from start to goal in a dynamic environment.
2) **Ensure smooth traversal** along the path.
3) **Consider near-optimal path length**.

The constraints for the problem are defined as follows:

1) **Linear velocity** of the robot is restricted to be lower than 0.8 m/s in x-y coordinates.
2) **Angular velocity** is limited to $\pi/6$ rad/s.

While achieving an optimal path length is a priority, global approaches are favored, acknowledging their limitations in ensuring safety in dynamic environments. Therefore, a hybrid approach combining reactive and classic methods is proposed. Specifically, the PF method is integrated with the PSO algorithm to address the identified priorities within the mentioned constraints.

B. ARTIFICIAL POTENTIAL FIELD

The conventional PF method is a widely used approach in path planning, where the robot's environment is modeled as a potential field. The potential field is composed of an attractive force that pulls the robot towards the goal, and a repulsive force that pushes the robot away from obstacles. The total potential field is the sum of these two forces.

The attractive potential field, $U_{att}(\mathbf{q})$, is typically defined as:

$$U_{att}(\mathbf{q}) = \frac{1}{2}\, k_{att}\, d_G^2 \tag{1}$$

where k_{att} is the attractive force gain, and d_G is the distance between the robot position ($\mathbf{q} = [x, y]$), and the goal ($\mathbf{q}_{\text{Goal}}$).

The repulsive potential field, $U_{rep}(\mathbf{q})$, is typically defined as:

$$U_{rep}(\mathbf{q}) = \begin{cases} \frac{1}{2}\, k_{rep} \left(\frac{1}{d_{obs}} - \frac{1}{q_0}\right)^2, & \text{if } d_{obs} \leq q_0 \\ 0, & \text{if } d_{obs} > q_0 \end{cases} \tag{2}$$

where k_{rep} is the repulsive force gain, d_{obs} is the distance between the robot and the nearest obstacle position ($\mathbf{q}_{obs}$), and q_0 is the influence distance of the obstacle.

The total potential field, $U_{total}(\mathbf{q})$, is the sum of the attractive and repulsive potential fields:

$$U_{total}(\mathbf{q}) = U_{att}(\mathbf{q}) + U_{rep}(\mathbf{q}) \tag{3}$$

Conventional PF methods can fail in certain situations. For example, when the robot is at a considerable distance from its target, the attractive force is significantly heightened. This can lead to the robot approaching obstacles too closely when the distance is substantial, prompting [44] to recommend adjustments to the conventional U_{att}. Conversely, when the robot is near its destination, it may also be in proximity to surrounding obstacles. In this scenario, the repulsive force becomes much more dominant than the attractive force, hindering the robot's ability to reach its goal. To tackle this problem, [44] proposed modifications to U_{rep}. To address these issues, we propose the use of a modified version of the APF method, as presented in [21], instead of techniques that integrate traditional PF methods into their navigation approach, such as those described in [9] the attractive and repulsive potentials are modified as follows:

$$U_{att}((\mathbf{q}) = \begin{cases} \frac{1}{2}k_{att}.d_G^2 & d_G \leq q_G \\ q_G.k_{att}.d_G - \frac{1}{2}k_{att}.q_G^2 & d_G > q_G \end{cases} \tag{4}$$

where q_G is the threshold value that defines the distance between the robot and the goal which is compared for the choice between conic and quadratic potential. The attractive force which is the gradient of the function (4) is defined below:

$$F_{att}(\mathbf{q}) = -\nabla U_{att}(\mathbf{q}) = \begin{cases} -k_{att}.(\mathbf{q} - \mathbf{q}_{\text{Goal}}) & d_G \leq q_G \\ -q_G.k_{att}\frac{(\mathbf{q}-\mathbf{q}_{\text{Goal}})}{d_G} & d_G > q_G \end{cases} \tag{5}$$

Correspondingly, the repulsive potential can be described by:

$$U_{rep} = \begin{cases} \frac{1}{2}k_{rep}\left(\frac{1}{d_{obs}} - \frac{1}{d_0}\right)^2 d_G^n & d_{obs} \leq d_0 \\ 0 & d_{obs} > d_0 \end{cases} \tag{6}$$

where d_0 is the threshold value and n is the positive constant. The repulsive force of (6) is calculated as follows:

$$F_{rep} = -\nabla U_{rep}(\mathbf{q}) = \begin{cases} F_{rep1} + F_{rep2} & d_{obs} \leq d_0 \\ 0 & d_{obs} > d_0 \end{cases} \tag{7}$$

$$F_{rep1} = k_{rep}\left(\frac{1}{d_{obs}} - \frac{1}{d_0}\right)\frac{d_G^n}{d_G^2} \tag{8}$$

$$F_{rep2} = -\frac{n}{2}k_{rep}\left(\frac{1}{d_{obs}} - \frac{1}{d_0}\right)^2 d_G^{n-1} \tag{9}$$

While the modified potential field function, defined by equations (4) and (6), can address some of the issues, it still yields non-smooth paths and neglects kinematic constraints. Additionally, it may remain vulnerable to local minima. To mitigate these limitations, especially in dynamic environments, we recommend adopting the concept of force Model scenarios [45] within the potential field framework. Instead of using the potential function from [44], this approach allows for a more conservative strategy. These scenarios, although similar, may differ in their force definitions. We propose accounting for the velocity of obstacles, or at least considering the maximum possible velocity if the actual velocities are not readily measurable. Our study introduces a modification to the parameter d_0, which is one of the key contributions of this paper.

C. IMPROVED ARTIFICIAL POTENTIAL FIELD

To address these limitations, this paper proposes enhancing the potential field-based path planning approach through several key modifications:

1) **Cell Decomposition:** This method is adapted to incorporate angular and linear displacement constraints.
2) **Kinematic Constraints:** Turning radius and angular velocity limits are considered within the IPF framework to ensure the feasibility of the generated paths.

The IPF method aims to overcome the limitations of the conventional potential field method and enhance the Artificial Potential Field (APF) algorithm by incorporating these modifications to generate more realistic and feasible paths for the robot.

To this end, this paper suggests adapting the new distance threshold value, d_0, based on the velocity of the robot and obstacles to handle local minima as a modified version of [21] and [22]. If the obstacle velocity, denoted as $\mathbf{V}_{ob}(k)$, is observable, the paper proposes a new predictive IPF collision avoidance criteria for the repulsive force. This criteria takes into account the relative velocity between the robot and the obstacle, as well as the distance between them. This approach enables more accurate prediction of potential collisions and generates smoother and safer paths for the robot to navigate through cluttered environments.

By incorporating these modifications, the IPF method aims to overcome the limitations of the conventional potential field method and provide a more robust and efficient path planning algorithm for mobile robots. The adaptive distance threshold and predictive collision avoidance criteria enable the robot to navigate more safely and effectively, making it suitable for real-world applications.

The paper outlines the specific details of the new predictive IPF collision avoidance criteria for the repulsive force as follows:

$$d_0 = \begin{cases} \|\dot{\mathbf{q}}\| + \|\dot{\mathbf{q}}_{obs}\| + d_c & \dot{\mathbf{q}}_{obs} \text{ determined} \\ \|\dot{\mathbf{q}}\| + \max\{\|\dot{\mathbf{q}}_{obs}\|\} + d_c & \max\{\|\dot{\mathbf{q}}_{obs}\|\} \text{ determined} \\ 2\max\{\|\dot{\mathbf{q}}\|\} + d_c & \text{otherwise} \end{cases} \tag{10}$$

It is assumed that the velocity of the robot, denoted as v, can be observed and estimated either through the optimal particle velocity approximation (see (16)) or the dynamic equations of the robot. In most environments, it is possible to determine the velocity or at least the maximum velocity of the obstacles. For uncertainty considerations, the boundary of uncertainty is considered as a constant value d_c. Although our simulation studies, like others, may provide complete knowledge of all values, we have intentionally assumed that only the maximum velocity values are known. This decision reflects a more realistic approach to robot navigation in the real world, where unknown values and uncertainties regarding parameters, estimated positions, and velocities are common. By assuming that we only know the maximum velocity in the environment, we can better simulate the challenges faced in practical scenarios. Consequently, we calculated the boundary of uncertainty, d_0, based on this maximum velocity. This approach allows us to address the inherent uncertainties and complexities of real-world navigation, making our findings more applicable to actual robotic applications. The simplifying assumption allows us to focus on the main aspects of the problem without getting bogged down in the complexities of real-time velocity estimation. However, in practical applications, the uncertainty boundary d_c may not be truly constant due to factors such as sensor noise, environmental changes, or robot dynamics. Nevertheless, assuming a constant d_c serves as a reasonable approximation for the purpose.

Based on the specified priorities and criteria for the navigation problem, the minimization of the potential field described in equation (3) satisfies two criteria: minimizing the path length and avoiding collisions. To also address the criterion of path smoothness in conjunction with the Improved Potential Field strategy, this paper suggests the following objective function:

$$f(\mathbf{q}) = U(\mathbf{q}) + (\dot{\theta} - \omega_{\max})^2 + (\dot{\mathbf{q}} - \dot{\mathbf{q}}_{\max})^2 \tag{11}$$

Where $\dot{\theta}q$ is the angular velocity, ω_{max} and $\dot{q}_{max}$ are respectively the maximum angular velocity and linear velocity of the robot, defined by kinematic constraints.

The objective function (11) is paramount for achieving a balance between speed and smoothness in the generated path. It accomplishes this by imposing penalties on deviations from the optimal linear and angular velocities, thereby effectively steering the trajectory towards smoother paths. The parameter w in equation (20) assumes a pivotal role in this methodology by modulating the velocity updates within the framework of PSO. By adjusting the velocity in accordance with the term $(1-\frac{d\theta}{dq})$, wherein an elevated value of $\frac{d\theta}{dq}$ results in diminished velocity, the system fosters smaller and more controlled steps toward the next position. This modulation not only mitigates abrupt directional shifts but also promotes gradual modifications, ultimately contributing to the enhancement of the overall smoothness of the generated pathway.

This study proposes to reformulate the angular velocity based on the linear velocity and the robot's movement. To this aim, consider fig1, which indicates the relationship between the angle of movement and the displacement in the x and y directions. The figure illustrates how the robot's motion can be decomposed into linear and angular components, which can be used to derive a novel expression for the angular velocity.

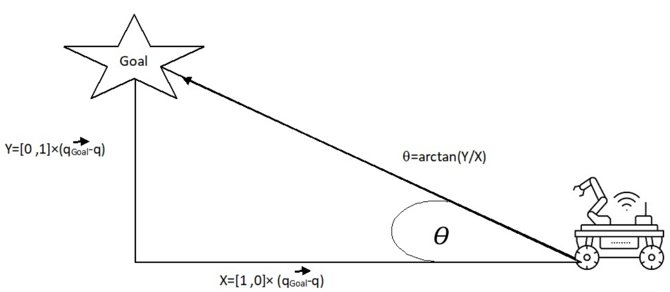


FIGURE 1: Relationship between displacement and angle of movement

By analyzing the geometric relationships shown in the figure, and considering the kinematic constraints of the robot, this study aims to develop a more comprehensive model for the angular velocity $\dot{\theta}_q$. This reformulation will provide a better understanding of the robot's motion and enable more efficient control strategies. The proposed approach leverages the insights gained from the visual representation in fig1 to derive a refined expression for the angular velocity, which can be used to improve the overall navigation and control of the robotic system.

In this regard, let $(q_{Goal} - q) = [x_d, y_d]$ be a vector in the Cartesian plane. The angle can be defined as the inverse tangent of the ratio of the y-component and the x-component $(\beta = \frac{y_d}{x_d})$ of the vector:

$$\theta = \arctan(\beta) \tag{12}$$

The derivative of θ with respect to t can be calculated using the chain rule:

$$\frac{d\theta}{d\mathbf{q}} = \frac{1}{1+\beta^2} \cdot \frac{1}{\|\mathbf{q}_{\text{Goal}} - \mathbf{q}\|^2} \begin{bmatrix} -y_d & x_d \end{bmatrix} \tag{13}$$

This derivative can be expressed in vector form as:

$$\frac{d\theta}{d\mathbf{q}} = \frac{1}{\|\mathbf{q}_{\text{Goal}} - \mathbf{q}\|^2} \begin{bmatrix} -y_d & x_d \end{bmatrix} \tag{14}$$

where $\|\mathbf{q}_{\text{Goal}} - \mathbf{q}\|$ represents the magnitude of the vector $\mathbf{q}_{\text{Goal}} - \mathbf{q}$, i.e., d_G.

This equation is calculated based on considering a restriction on $\arctan(q_{goal} - q)$ which is calculated as follows:

$$\dot{\theta} = \frac{d\theta}{d\mathbf{q}} \cdot \frac{d\mathbf{q}}{dt} = \frac{(\dot{\mathbf{q}}_{\text{goal}} - \dot{\mathbf{q}}) \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} (\mathbf{q}_{\text{goal}} - \mathbf{q})^T}{d_G^2 + \varepsilon_0} \tag{15}$$

Where ε_0 is a small positive constant which is considered with the aim of avoiding the singularity condition.

1) PSO Algorithm

The PSO algorithm is used to optimize the total potential field, which serves as the fitness function. PSO is chosen for its ability to efficiently explore the search space and find near-optimal solutions [39]. The velocity and position of an ith particle can be updated iteratively in the kth iteration as follows:

$$\mathbf{v}_i^{k+1} = w \cdot \mathbf{v}_i^k + c_1 r_1 (\mathbf{q}_{p_i}^k - \mathbf{q}_i{}^k) + c_2 r_2 (\mathbf{q}_g^k - \mathbf{q}_i{}^k) \tag{16}$$

$$\mathbf{q}_i{}^{k+1} = \mathbf{q}_i{}^k + \mathbf{v}_i^{k+1} \tag{17}$$

where $\mathbf{v}_i^k$ is the velocity of particle i at iteration k, $\mathbf{q}_i{}^k$ is the ith particle at kth iteration, and w is the inertia weight. c_1 and c_2 are the coefficients of particle acceleration, mostly selected in the range of [0,2], and r_1 and r_2 are random numbers uniformly distributed between 0 and 1.$\mathbf{q}_{p_i}^k$ is the best-known position of particle i, and $\mathbf{q}_g^k$ is the global best position known by the swarm. $\mathbf{q}_{p_i}^k$ and $\mathbf{q}_g^k$ are respectively determined as follows:

$$\mathbf{q}_{p_i}(i,t) \quad = \quad \arg\min_{k=1,\ldots,t} f(\mathbf{q}_{k,i}), \quad i \quad \in \quad \{1,\ldots,N_p\} \tag{18}$$

$$\mathbf{q}_g(t) = \underset{k=1,\ldots,t,\, i\in\{1,\ldots,N_p\}}{\arg\min} f(\mathbf{q}_{k,i}) \tag{19}$$

where N_p is the number of particles and $f(\mathbf{q})$ is the fitness function, defined at (11).
Since the velocity of particles must be adjusted according to the robot's kinematic constraints, it is essential to generate a path using PSO while adhering to these constraints. To effectively incorporate the objective function defined in (11), the variables and constant values in (16) are established based on this objective function.
To this aim and to address the issue of getting trapped in local minima and generating smooth paths, we have utilized several techniques. Firstly, it is considered specific values for r_1c_1 and r_2c_2 that may exceed the commonly used range to speed up the convergence of PSO. This approach allows us to explore the impact of these coefficients on the optimization process and the generated path.
Additionally, we have utilized the velocity direction of repulsive and attractive forces calculated in the potential field to guide the particles toward the global minimum. By considering the direction of these forces, we can push the particles away from local minima and towards the desired goal. Furthermore, we have incorporated a negative feedback of angular velocity in the particle velocity to generate a smoother path.
To further enhance the smoothness of the generated path and ensure that the robot stays within the maximum allowed velocity, we have included the term $(\dot{\theta} - \omega)^2$ in the potential field cost function. This term penalizes deviations from the desired angular velocity, encouraging the robot to maintain a consistent velocity throughout the navigation process.
By combining these techniques, we have effectively addressed the challenges of local minima and generated smooth, consistent paths for the robot to navigate in dynamic environments.
The equation (20) is derived from the concept that the velocity difference can be obtained from the difference in force, which is in turn derived from kinetic energy, expressed as $K_m = \frac{1}{2}m\mathbf{v}^2$, and the force, given by $F_m = m\frac{d\mathbf{v}}{dt}$.
Accordingly, the parameters are defined as follows:

$$w = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \frac{d\theta}{d\mathbf{q}} \tag{20}$$

The equation (20) is derived from the concept that the velocity difference can be obtained from the difference in force, which is in turn derived from kinetic energy, expressed as $K_m = \frac{1}{2}m\mathbf{v}^2$, and the force, given by $F_m = m\frac{d\mathbf{v}}{dt}$. Accordingly, other parameters are updated based on the following laws:

$$c_1 r_1 = k_{rep}\left(\frac{1}{d_{obs}} - \frac{1}{d_0}\right) \cdot \frac{1}{d_{obs}} \cdot \mathrm{step}(d_0 - d_{obs}) \tag{21}$$

$$c_2 r_2 = k_{att}\left(1 + \left(\frac{q_G}{d_G} - 1\right) \cdot \mathrm{step}(d_G - q_G)\right) - k_{rep}\left(\frac{1}{d_{obs}} - \frac{1}{d_0}\right)^2 \tag{22}$$

where step(.) is step function, defined as follows:

$$\mathrm{step}(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases} \tag{23}$$

2) Methodology

The problem definition and methodology are illustrated in the accompanying flowchart (Figure 2). This flowchart provides a detailed, step-by-step depiction of the proposed hybrid approach, showcasing the integration of the IPF method and PSO. Additionally, it highlights how kinematic constraints are incorporated throughout the process, ensuring a comprehensive understanding of the approach's structure and functionality.

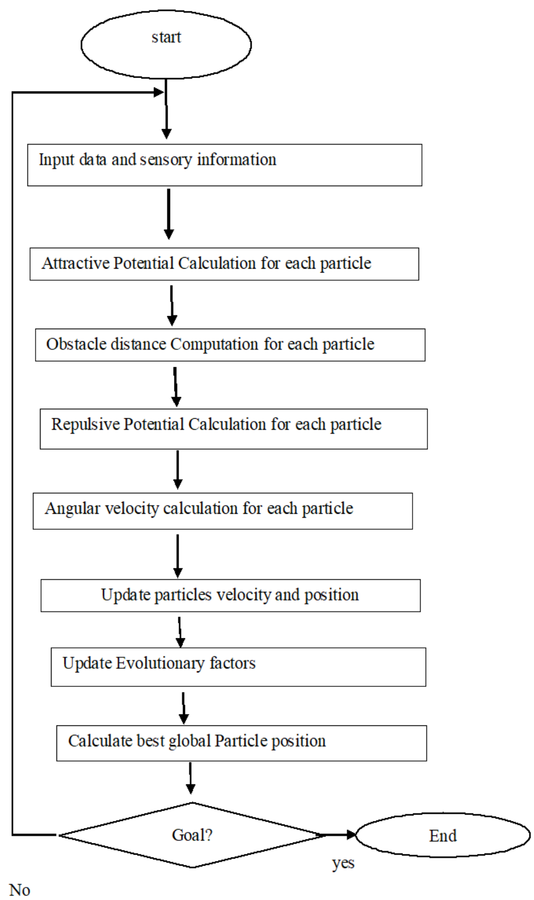


FIGURE 2: Flow chart of path planning method of PSO integrated by PF

By combining the strengths of the IPF method and the smooth PSO algorithm (SPSO-IPF), while incorporating kinematic constraints, the proposed approach aims to generate safe, smooth, and near-optimal paths for the robot in dynamic environments, within the specified velocity constraints. The algorithm of the proposed SPSO-IPF is described below:

Algorithm 1 Path Planning Method of proposed SPSO-IPF

Step 1: Initialization

Determine Lower bound L_b and Upper bound U_b of particles' positions, based on kinematic constraints
for each Particle $i = 1, \ldots, N_p$ **do**
  Calculate d_0 (see (10))
  Initialize the particles' position $\mathbf{q}_i^0$ and uniformly in the range of $[L_b, U_b]$
  Initialize velocity uniformly $V_i(0)$, according to the kinematic constraints
end for

Step 2: Repeat until the minimum objective value is achieved

while true **do**
  Calculate $c_1 r_1$, $c_2 r_2$, and w (see (21) to (22))
  Update the velocity and position of each particle (see (16) and (17))
  Update $\mathbf{q}_{p_i}^k$ and $\mathbf{q}_g^k$ (see (18) and (19))
  Update time epoch
end while

Step 3: Return the best-found solution at time epoch t

III. RESULTS

To assess the performance of our proposed method, SPSO-IPF, we conducted a comparative analysis with three other path planning strategies: Cubic Spline PSO [39], Bezier curve PSO [19], and apfrPSO [9]. We chose to compare our approach with Cubic Spline PSO and Bezier curve PSO because they are both smooth variants of the PSO algorithm, which aligns with the goals of our SPSO-IPF method. Additionally, we included apfrPSO, which combines PSO with a Potential Field approach, to provide a fair comparison with similar state-of-the-art techniques.

Table 1 summarizes the parameter settings used across the various algorithms evaluated in our experiments, including Cubic Spline PSO, Bezier Curve PSO, apfrPSO, and our proposed method, SPSO-IPF.

TABLE 1: Parameter settings for the evaluated algorithms.

Parameter	Value
Max iterations	300
Population size	200
Initial Cognitive coefficient (c_1)	2
Initial Social coefficient (c_2)	1
Initial Inertia weight (w)	0.8
Damping factor of w	1
Resetting parameter	25
Attractive force coefficient (k_{att})	0.5
Repulsive force coefficient (k_{rep})	1
Repulsion radius	0.5
q_G	0.5
d_c	0.01
n	2

The comparison was performed in four distinct static environments to ensure a comprehensive evaluation. In this study, four distinct environments are designed to rigorously evaluate the performance of our algorithm in navigating complex obstacle configurations. Each environment is constructed with specific characteristics to challenge traditional PSO and potential field methods. For instance, Environment 1 features five obstacles of varying sizes, strategically placed to create a mix of open and constrained spaces. The obstacles, with radii ranging from 5 to 12 units, are positioned to test the algorithm's ability to navigate through tight spaces while maintaining a clear path from the starting point at [5, 5] to the goal at [95, 95].

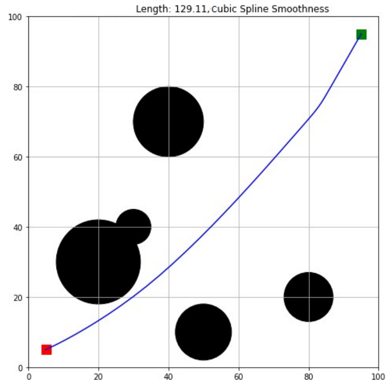
In contrast, Environment 2 introduces additional complexity with seven obstacles, including larger ones with radii up to 15 units. This arrangement creates potential bottlenecks, particularly near the goal, challenging the algorithm to find efficient paths while avoiding collisions.

Environment 3 is the most complex, featuring a total of 13 obstacles that vary in size and placement, designed to create a highly cluttered space. This requires the algorithm to demonstrate its capability to navigate through densely populated areas effectively.

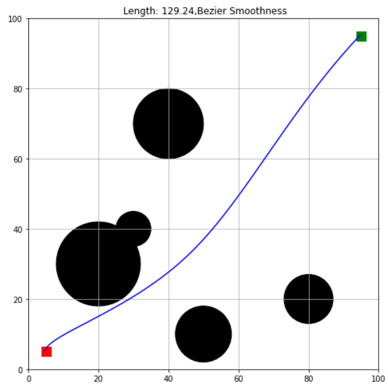
Finally, Environment 4, while similar to Environment 1 in terms of the number of obstacles, presents different placements and sizes to further evaluate how well the algorithm adapts to varying configurations. By designing these environments with diverse obstacle characteristics, we aimed to create scenarios that push the limits of conventional navigation algorithms, allowing for a comprehensive assessment of our proposed method's effectiveness in overcoming challenges associated with local minima and collision avoidance.

The best path generation of these methods, has been achieved by 10 runs with three iterations is illustrated in Figures 3 to 6 for the different environmental examples. The results of this comparison are summarized in Table 2, highlighting the performance differences among these methods.

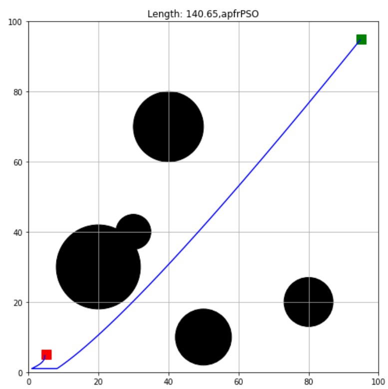
To provide a comprehensive evaluation of the algorithms' performance, we conducted multiple execution runs to account for the stochastic nature of the methods. Initially, we reported the averaged results from three iteration runs for each algorithm.



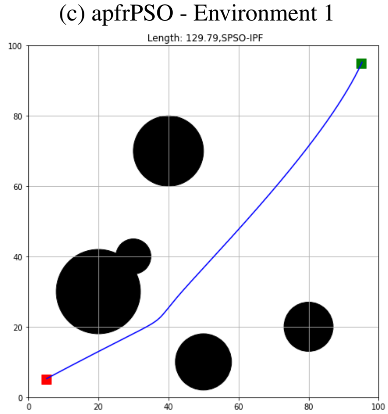
(a) Cubic-Spline PSO - Environment 1



(b) Bezier Curve PSO - Environment 1

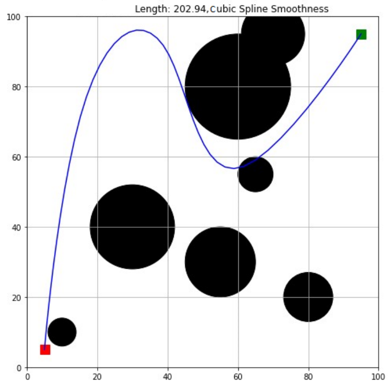


(c) apfrPSO - Environment 1

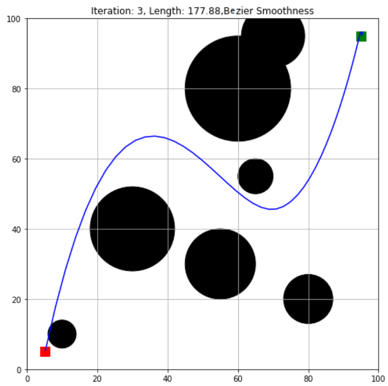


(d) SPSO-IPF - Environment 1

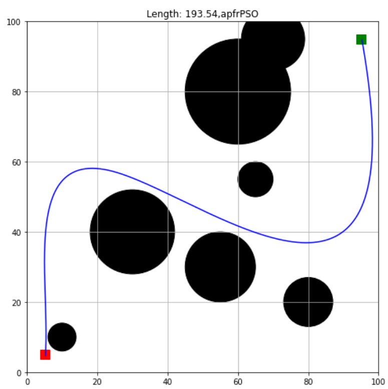
FIGURE 3: Generated Path Comparison in Environment 1



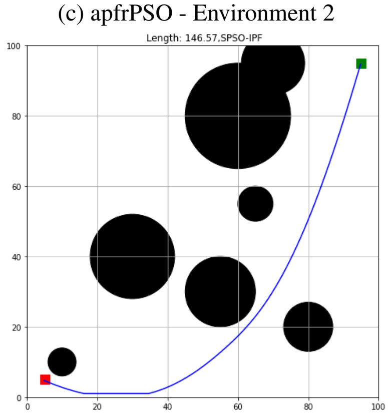
(a) Cubic-Spline PSO - Environment 2



(b) Bezier Curve PSO - Environment 2



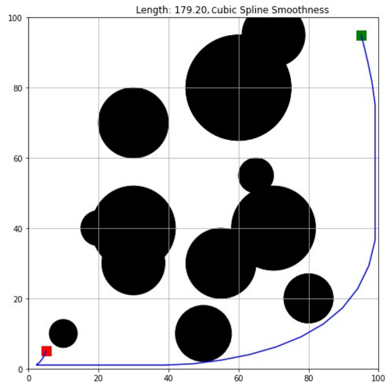
(c) apfrPSO - Environment 2



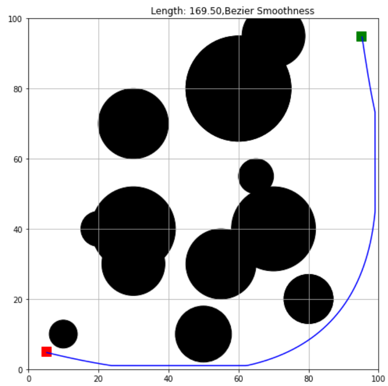
(d) SPSO-IPF - Environment 2

FIGURE 4: Generated Path Comparison in Environment 2

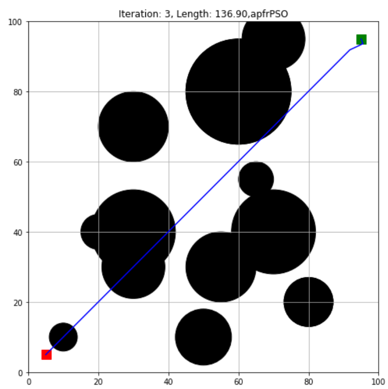
This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3504360



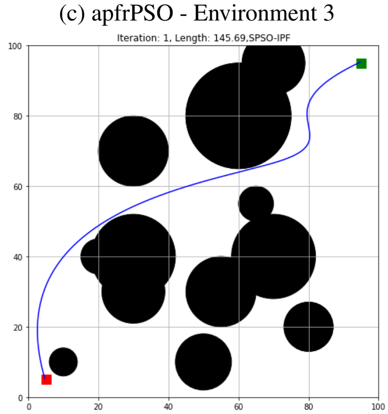
(a) Cubic-Spline PSO - Environment3



(b) Bezier Curve PSO - Environment 3

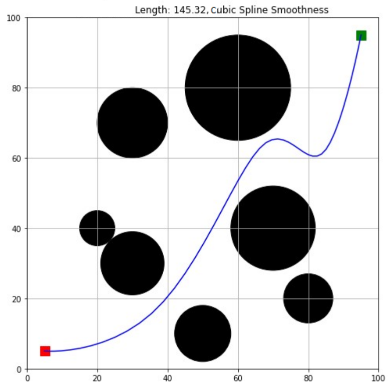


(c) apfrPSO - Environment 3

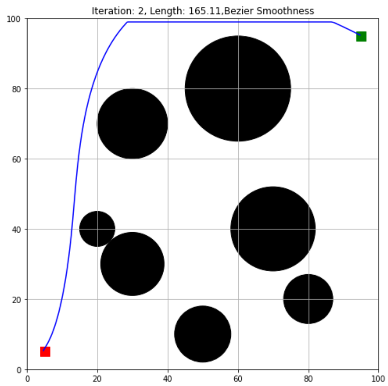


(d) SPSO-IPF - Environment 3

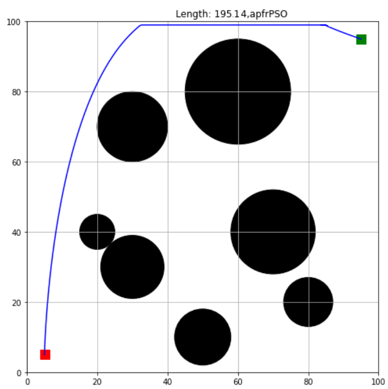
FIGURE 5: Generated Path Comparison in Environment 3



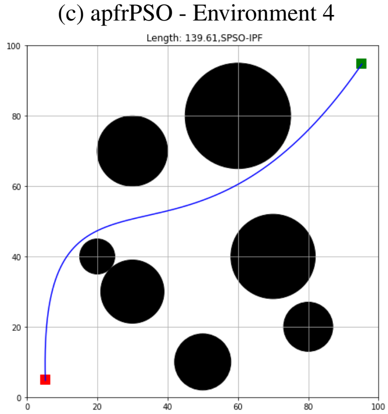
(a) Cubic-Spline PSO - Environment4



(b) Bezier Curve PSO - Environment 4



(c) apfrPSO - Environment 4



(d) SPSO-IPF - Environment 4

FIGURE 6: Generated Path Comparison in Environment 4

However, our experiments involved at least ten executions for each algorithm, with each execution consisting of three iterations. This approach enables us to capture a more accurate representation of the performance variability across different runs. To enhance the clarity and robustness of our results, we have included the mean and standard deviation of key parameters such as path length, time, and speed peaks per meter (SP) as a measure of smoothness criterion [43], for these ten executions in 2.

The T-test statistical analysis of the smoothness of paths generated by the algorithms, presented in 2 shows that our proposed algorithm performs significantly better than the Bezier Curve PSO and APFR-PSO algorithms, with p-values of 0.027531568 and 0.019730091, respectively. This suggests that our algorithm may achieve superior smoothness or efficiency in certain scenarios. Conversely, the lack of significance when compared to the Cubic Spline PSO algorithm (p-value of 0.345805076) indicates that our proposed algorithm and the Cubic Spline PSO algorithm have similar smoothness performance characteristics. This similarity could be attributed to the specific environmental conditions under which the algorithms are evaluated. For instance, in environments with fewer obstacles, both algorithms may perform similarly well.

As shown in Table 2, SPSO-IPF effectively balances shorter path lengths with collision avoidance, while also providing reasonable execution times and competitive smooth path. This positions SPSO-IPF as a compelling option for path planning in autonomous systems when compared to the other methods assessed in this study.Figures 3 to 6 depict the four unique environmental scenarios discussed in the comparative analysis shown in Table 2.

A significant observation is that the SPSO-IPF method effectively balances path length, safety, and smoothness.

A summary of the metrics comparing these methods, provided in Table 2, highlights the superiority of the SPSO-IPF method in more complex environments, particularly regarding the generation of safe and shortest paths. As shown in Table 2, among the methods that produce collision-free paths in environments 1 to 4, the SPSO-IPF method averagely yields the shortest paths, with no collisions occurring during ten test runs for each environment across three iterations.Although the Bezier Curve PSO method produced a slightly shorter average path in Environment 2, its higher standard deviation indicates less reliability compared to the paths generated by SPSO-IPF. Importantly, all methods were tested under the same conditions to ensure a fair evaluation.

In contrast, both the Cubic Spline PSO method failed to generate collision-free paths in Environment 2, and apfrPSO also encountered difficulties in Environment 3. All processes for the Cubic Spline PSO, Bezier Curve PSO, apfrPSO, and SPSO-IPF were conducted using 200 particles, with the time cost calculated for a single iteration for each method throughout one epoch. Notably, Cubic Spline PSO exhibited the lowest time cost, making it the fastest approach, while the proposed SPSO-IPF method ranked third in terms of time cost, demonstrating moderate efficiency.

Figures 7 and 8 illustrate the path generation performance of the SPSO-IPF method in dynamic environment. In real-time applications, where only one run per epoch is typically considered, the focus is primarily on the collision avoidance criterion. Consequently, the criteria for smoothness and path length optimization may not significantly influence the generated trajectory. This underscores the method's effectiveness in ensuring safety by prioritizing obstacle avoidance, even if it compromises the smoothness and optimality of the path length.

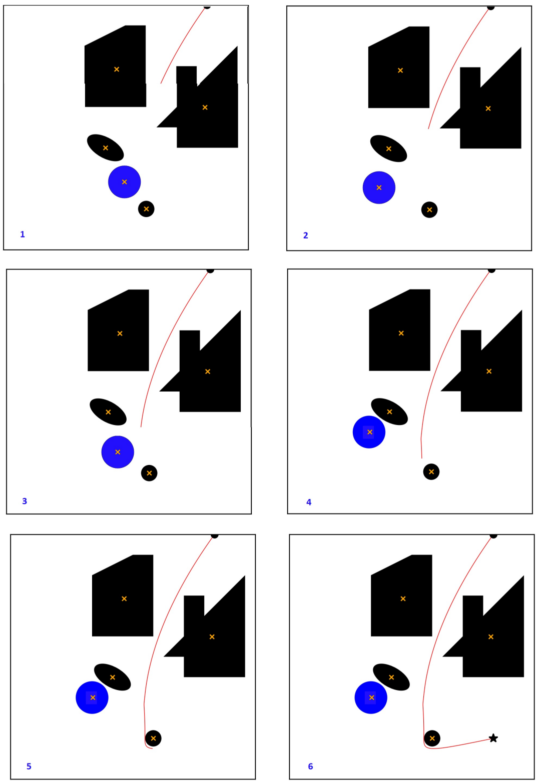


FIGURE 7: Performance of the SPSO-IPF algorithm in a dynamic environment. The moving obstacle is shown by a green circle, the path previously traversed by the obstacle is depicted by a light green tube, the generated path is shown by a red line, and the star represents the target position.

Figure 9 illustrates how the proposed method can outperform non-smooth PSOPF [42], both in terms of path length and generating a smooth path within a complex environment.The mean and standard deviation on the number of speed peaks indicates that the proposed SPSO-IPF with 0.9 ± 0.4 outperforms non smooth PSO with 1.16 ± 0.6.

However, it is essential to recognize that the computational cost of the proposed method may increase with the complexity of the environment, which poses a practical concern. As mentioned earlier, finite-time convergence guarantees that the algorithm will find a path within a bounded number of iterations or a specified time limit [41]. While this is a

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3504360

TABLE 2: Performance Metrics of Different Methods Across Environments

Method and Metrics	Env1				Env2			
	Length	Collision	Time(s)	SP	Length	Collision	Time(s)	SP
Cubic Spline PSO [39]	130.9 ± 1.062	No	0.30	1.167 ± 0.11	183.93 ± 27.11	Yes	0.31	1.125 ± 0.0625
Bezier Curve PSO [19]*	133.25 ± 1.78	No	0.33	0.875 ± 0.05	169.89 ± 7.2	No	0.36	0.857 ± 0.346
apfrPSO [9] *	132.24 ± 8.2	No	0.27	1.083 ± 0.062	176.03 ± 6.14	No	0.31	1.303 ± 0.157
SPSO-IPF	129.96 ± 1.04	No	0.31	0.93 ± 0.187	169.95 ± 6.089	No	0.39	1 ± 0.25
Method and Metrics	Env3				Env4			
	Length	Collision	Time(s)	SP	Length	Collision	Time(s)	SP
Cubic Spline PSO [39]	185.84 ± 10.28	No	0.38	1.05 ± 0.04	146.62 ± 1.73	No	0.36	1.12 ± 0.368
Bezier Curve PSO [19]*	169.85 ± 1.7	No	0.49	1.25 ± 0.231	145.46 ± 4.68	No	0.41	1.125 ± 0.347
apfrPSO [9] *	159.33 ± 11.2	Yes	0.47	1.267 ± 0.187	146.19 ± 0.96	No	0.34	1.2 ± 0.08
SPSO-IPF	162.73 ± 6.75	No	0.43	0.875 ± 0.0625	142.31 ± 2.05	No	0.33	1.1 ± 0.06

* indicates SP metric t-test statistical significance at $P < 0.05$.

desirable property, the computational cost associated with achieving finite-time convergence in complex environments should be carefully considered. Investigating techniques to maintain finite-time convergence while reducing computational overhead could be an area for future research.
Furthermore, in dynamic environments with high velocities, the time cost of execution becomes critical. If the vector sum of the velocities of both the robot and the nearest obstacle exceeds the path planning speed, the effectiveness of the path planning may be compromised. This situation could lead to potential failures in collision avoidance, especially if the controller and actuator speeds are not sufficiently responsive. Addressing this challenge requires a holistic approach that considers the interplay between path planning, control, and actuation in high-speed dynamic environments.
Furthermore, regarding the issue of locating the target or start position within a non-convex obstacle, Figure 8, which illustrates step-by-step path planning in a dynamic environment, shows that in step 7, the obstacle collides with the target, and the target position is located at the same position as the obstacle. In this situation, collision avoidance becomes the most critical criterion, and the path planning process remains in the loop. If this condition persists, the algorithm may fail to find a feasible path. However, when generating paths in static environments (Figure 9), we observed that if the process is offline or in a static environment, the start position can be located within a non-convex obstacle, and the algorithm is still capable of generating a safe path. This demonstrates that the proposed method can handle such scenarios in static environments.
Additionally, while considering the kinematics of the problem is crucial, the role of the controller for autonomous vehicles or mobile robots should not be overlooked. In scenarios where obstacles are in close proximity, tracking control errors may lead to collisions.
Future research should prioritize the investigation of tracking controller designs and their effects on the path planning of mobile robots, as well as the evaluation of their performance across diverse scenarios. This exploration has the potential to yield valuable insights into enhancing the robustness and safety of autonomous navigation systems. Furthermore, the emphasis on evaluating all methods under consistent conditions strengthens the validity and reliability of the performance metrics presented.
In addition, in multi-robot systems or decentralized path planning, robots need to communicate their positions and planned paths to coordinate and avoid collisions. Using discrete communication [40], where information is exchanged at specific time intervals rather than continuously, can help reduce communication overhead and complexity.

Figures 10 and 11 showcase additional screenshots of dynamic scenarios. In Figure 10, all obstacles are in motion, creating a fully dynamic environment. Conversely, Figure 11 highlights one dynamic obstacle marked in blue, while the remaining obstacles are static. Figure 12 presents a scenario akin to that in Figure 11 but includes a comparative analysis of different methods' performances in complex environments. As illustrated in Figure 12, the Cubic Spline PSO method collides with an obstacle, whereas both the Bezier Curve PSO and apfrPSO methods produce paths that are nearly identical and dangerously close to the obstacles, posing a high risk of collision. In contrast, the SPSO-IPF method generates a longer yet significantly safer path, demonstrating its effectiveness in navigating complex scenarios.

IV. CONCLUSION

This paper presents an innovative approach known as path-smoothed PSO-IPF, which not only incorporates kinematic constraints but also enhances collision avoidance criteria within the potential field objective function. It underscores the significance of considering various factors, such as obstacle avoidance, robot speed, and environmental dynamics, when formulating optimal path-planning strategies. While the criticality of path smoothness is emphasized, particularly in static environments, the study acknowledges its potential decline in importance in dynamic settings near the target.
Nonetheless, the proposed technique demonstrates promising outcomes by generating secure, concise, and notably smooth paths in both static and dynamic scenarios. Furthermore, this exploration provides valuable insights into the development of effective navigation systems for autonomous robots.

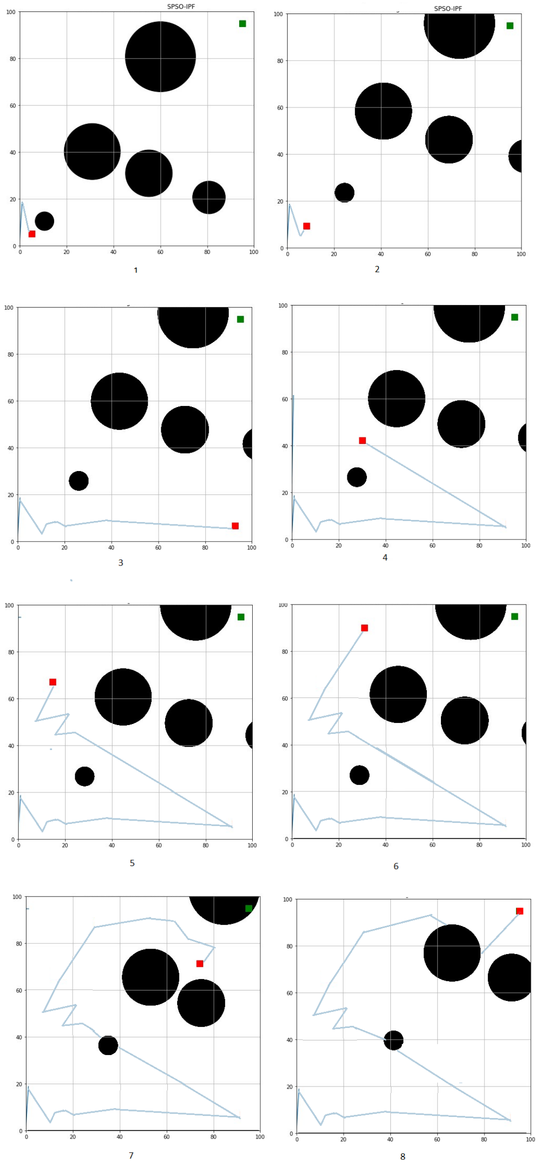


FIGURE 8: Performance of the SPSO-IPF algorithm in a dynamic environment. The moving obstacle are shown by black circles, the generated path is shown by a blue line, and the red square exhibits the robot position and the green square represents the target position.

By prioritizing problem constraints and adopting a hybrid approach that combines reactive and classical techniques, it offers a robust framework for creating effective path-planning solutions. This approach addresses the multifaceted challenges of navigation, paving the way for enhancing the autonomy and adaptability of robotic systems in diverse real-world environments.

In conclusion, the findings of this study have significant practical implications for the implementation of autonomous navigation systems. By integrating advanced methodologies such as Particle Swarm Optimization (PSO) and refining

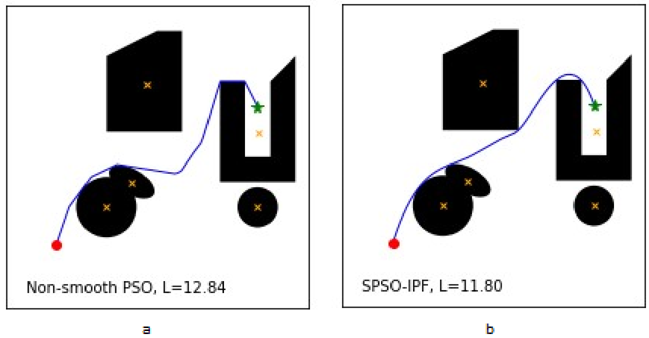


FIGURE 9: Path planning comparison between two methods: (a) non-smooth PSO-PF [42] and (b) the proposed SPSO-IPF. The obstacles are static.

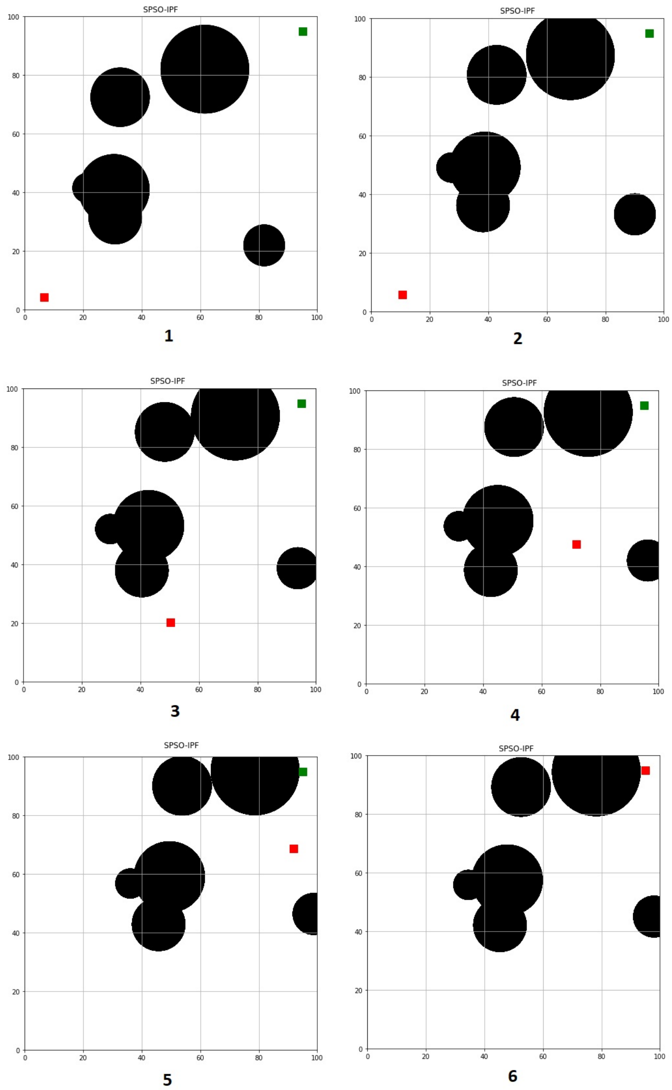


FIGURE 10: Screenshots of the third Dynamic Environment Scenario- The red Square is robot position, and the green Square indicates target

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3504360

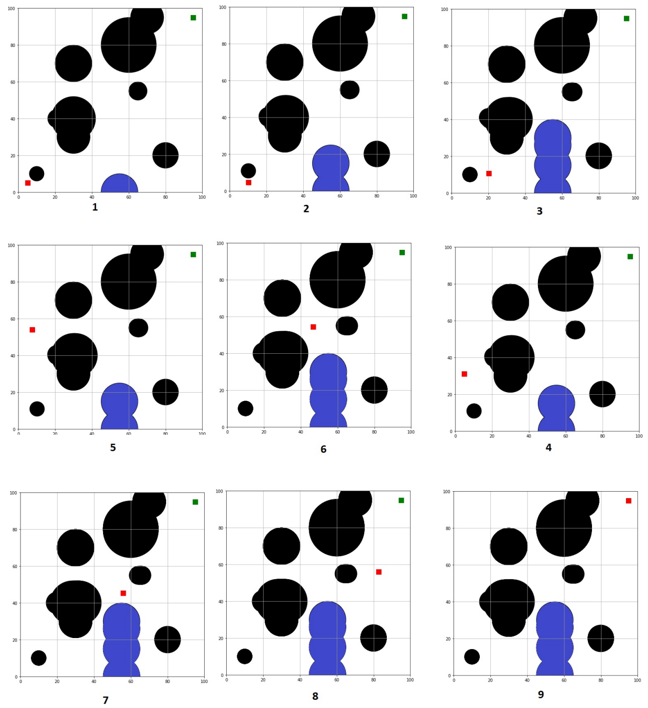


FIGURE 11: Screenshots of the fourth Dynamic and static Environment Scenario- The blue obstacle moves up and down- The red Square is robot position, and the green Square indicates target

potential field techniques while considering kinematic constraints, this research sets the stage for developing robust and adaptable robotic platforms capable of navigating complex environments with precision and efficiency.
However, it is crucial to recognize that the computational cost of the proposed method may increase with the complexity of the environment, which poses a practical concern. Finite-time convergence ensures that the algorithm will find a path within a bounded number of iterations or a specified time limit. While this property is desirable, the computational cost associated with achieving finite-time convergence in complex environments should be carefully considered. Future research should investigate techniques to maintain finite-time convergence while reducing computational overhead.

V. ACKNOWLEDGMENT

The authors express their gratitude and appreciation to Auckland University of Science and Technology (AUT) and its sponsors for their invaluable support and contribution to our research. We are grateful for the opportunity to conduct our research with the generous sponsorship provided by AUT, which has been instrumental in facilitating our project's success.

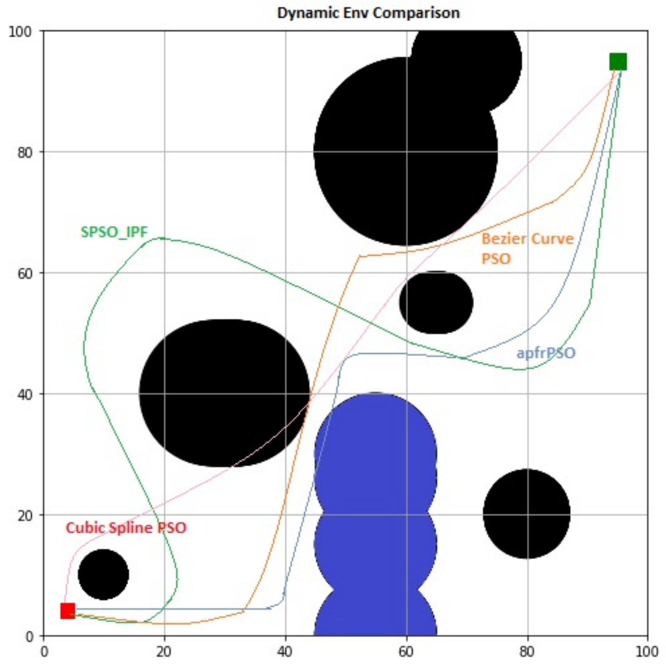


FIGURE 12: Comparison of generated paths by 4 different methods in a Dynamic and static Environment Scenario- The blue obstacle moves up and down- The red Square is robot position, and the green Square indicates target

REFERENCES

[1] M. Mohaghegh, S.-A. Saeedinia, and Z. Roozbehi, "Optimal predictive neuro-navigator design for mobile robot navigation with moving obstacles," Frontiers in Robotics and AI, vol. 10, 2023.
[2] A. Pandey, S. Pandey, and D. Parhi, "Mobile robot navigation and obstacle avoidance techniques: A review," Int Rob Auto J, vol. 2, no. 3, p. 00022, 2017.
[3] J. Xin, S. Li, J. Sheng, Y. Zhang, and Y. Cui, "Application of improved particle swarm optimization for navigation of unmanned surface vehicles," Sensors, vol. 19, no. 14, p. 3096, 2019.
[4] A. G. Gad, "Particle swarm optimization algorithm and its applications: a systematic review," Archives of computational methods in engineering, vol. 29, no. 5, pp. 2531-2561, 2022.
[5] C. Huang, X. Zhou, X. Ran, J. Wang, H. Chen, and W. Deng, "Adaptive cylinder vector particle swarm optimization with differential evolution for UAV path planning," Engineering Applications of Artificial Intelligence, vol. 121, p. 105942, 2023.
[6] Q. Chai, Y. Wang, Y. He, C. Xu, and Z. Hong, "Improved PRM path planning in narrow passages based on PSO," in 2022 IEEE International Conference on Mechatronics and Automation (ICMA), 2022: IEEE, pp. 41-46.
[7] Q. Meng, K. Chen, and Q. Qu, "PPSwarm: Multi-UAV Path Planning Based on Hybrid PSO in Complex Scenarios," Drones, vol. 8, no. 5, p. 192, 2024.
[8] Z. Yang, N. Li, Y. Zhang, and J. Li, "Mobile Robot Path Planning Based on Improved Particle Swarm Optimization and Improved Dynamic Window Approach," Journal of Robotics, vol. 2023, no. 1, p. 6619841, 2023.
[9] L. Zheng, W. Yu, G. Li, G. Qin, and Y. Luo, "Particle Swarm Algorithm Path-Planning Method for Mobile Robots Based on Artificial Potential Fields," Sensors, vol. 23, no. 13, p. 6082, 2023.
[10] R. K. Mandava, S. Bondada, and P. R. Vundavilli, "An optimized path planning for the mobile robot using potential field method and PSO algorithm," in Soft Computing for Problem Solving: SocProS 2017, Volume 2, 2019: Springer, pp. 139-150.
[11] M. Melchiorre, L. Salamina, L. S. Scimmi, S. Mauro, and S. Pastorelli, "Experiments on the Artificial Potential Field with Local Attractors for Mobile Robot Navigation," Robotics, vol. 12, no. 3, p. 81, 2023.
[12] M. S. Alvarez-Alvarado, F. E. Alban-Chacón, E. A. Lamilla-Rubio, C.

D. Rodríguez-Gallegos, and W. Velásquez, "Three novel quantum-inspired swarm optimization algorithms using different bounded potential fields," Scientific Reports, vol. 11, no. 1, pp. 1-22, 2021. .

[13] N. e. I. Bouaziz, O. Mechali, K. L. Besseghieur, and N. Achour, "Trajectory Planning for Autonomous Formation of Wheeled Mobile Robots via Modified Artificial Potential Field and Improved PSO Algorithm," Unmanned Systems, pp. 1-20, 2024.

[14] R. M. J. A. Souza, G. V. Lima, A. S. Morais, L. C. Oliveira-Lopes, D. C. Ramos, and F. L. Tofoli, "Modified artificial potential field for the path planning of aircraft swarms in three-dimensional environments," Sensors, vol. 22, no. 4, p. 1558, 2022.

[15] S. A. Saeedinia and M. Tale Masouleh, "The synergy of the multi-modal MPC and Q-learning approach for the navigation of a three-wheeled omnidirectional robot based on the dynamic model with obstacle collision avoidance purposes," Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, vol. 236, no. 17, pp. 9716-9729, 2022.

[16] X. Fan, Y. Guo, H. Liu, B. Wei, and W. Lyu, "Improved artificial potential field method applied for AUV path planning," Mathematical Problems in Engineering, vol. 2020, pp. 1-21, 2020.

[17] P. Gevargiz, "Control and Path Planning of Multi-Agent Systems: Artificial Potential Field Algorithm," California State University, Northridge, 2020.

[18] T. Mao and H. Deng, "Path Planning of Slender Tensegrities Based on the Artificial Potential Field Method," AIAA Journal, vol. 61, no. 5, pp. 2255-2265, 2023.

[19] L. Xu, M. Cao, and B. Song, "A new approach to smooth path planning of mobile robot based on quartic Bezier transition curve and improved PSO algorithm," Neurocomputing, vol. 473, pp. 98-106, 2022.

[20] R. Kala and K. Warwick, "Motion planning of autonomous vehicles on a dual carriageway without speed lanes," Electronics, vol. 4, no. 1, pp. 59-81, 2015.

[21] J. Song, C. Hao, and J. Su, "Path planning for unmanned surface vehicle based on predictive artificial potential field," International Journal of Advanced Robotic Systems, vol. 17, no. 2, p. 1729881420918461, 2020

[22] Z. Wu, J. Dai, B. Jiang, and H. R. Karimi, "Robot path planning based on artificial potential field with deterministic annealing," ISA transactions, vol. 138, pp. 74-87, 2023.

[23] A. Lazarowska, "Comparison of discrete artificial potential field algorithm and wave-front algorithm for autonomous ship trajectory planning," Ieee Access, vol. 8, pp. 221013-221026, 2020.

[24] N. Shvalb and S. Hacohen, "Motion in potential field and navigation function," Autonomous Mobile Robots and Multi-Robot Systems: Motion-Planning, Communication, and Swarming, pp. 87-107, 2019.

[25] J. Song, C. Hao, and J. Su, "Path planning for unmanned surface vehicle based on predictive artificial potential field," International Journal of Advanced Robotic Systems, vol. 17, no. 2, p. 1729881420918461, 2020

[26] P. Raja and S. Pugazhenthi, "Path planning for mobile robots in dynamic environments using particle swarm optimization," in 2009 International Conference on Advances in Recent Technologies in Communication and Computing, 2009: IEEE, pp. 401-405.

[27] F. A. Raheem and U. I. Hameed, "Interactive heuristic D* path planning solution based on PSO for two-link robotic arm in dynamic environment," World Journal of Engineering and Technology, vol. 7, no. 1, pp. 80-99, 2018.

[28] N. Van Dijk, N. Van de Wouw, H. Nijmeijer, and W. Pancras, "Path-constrained motion planning for robotics based on kinematic constraints," in International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2007, vol. 48094, pp. 1071-1080.

[29] C. Chun-Ta and H.-V. Pham, "Trajectory planning in parallel kinematic manipulators using a constrained multi-objective evolutionary algorithm," Nonlinear Dynamics, vol. 67, no. 2, pp. 1669-1681, 2012.

[30] S. H. Hwang, S. Kang, C. Y. Lee, and B. K. Min, "Tool Path Planning Using the Relationship between Cornering Error and Kinematic Constraint Values," Journal of the Korean Society for Precision Engineering, vol. 39, no. 12, pp. 899-904, 2022.

[31] Z. Liu, "Research on Path Planning Problem of Smart Car Considering Kinematic Constraints," Highlights in Science, Engineering and Technology, vol. 46, pp. 311-320, 2023.

[32] J. Fan et al., "Second path planning for unmanned surface vehicle considering the constraint of motion performance," Journal of Marine Science and Engineering, vol. 7, no. 4, p. 104, 2019.

[33] R. M. J. A. Souza, G. V. Lima, A. S. Morais, L. C. Oliveira-Lopes, D. C. Ramos, and F. L. Tofoli, "Modified artificial potential field for the path planning of aircraft swarms in three-dimensional environments," Sensors, vol. 22, no. 4, p. 1558, 2022.

[34] S. BaniHani, M. R. M. Hayajneh, A. Al-Jarrah, and S. Mutawe, "New control approaches for trajectory tracking and motion planning of unmanned tracked robot," Advances in Electrical and Electronic Engineering, vol. 19, no. 1, pp. 42-56, 2021.

[35] H. Jafarpourdavatgar and S. A. Saeedinia, "Overview of Optimization Techniques in Geometric Path Planning for Mobile Robots with a new smooth PSO-IPF design," Authorea Preprints, 2023.

[36] B. Song, Z. Wang, and L. Zou, "An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve," Applied soft computing, vol. 100, p. 106960, 2021.

[37] F. A. Raheem and M. I. Abdulkareem, "Development of a* algorithm for robot path planning based on modified probabilistic roadmap and artificial potential field," J. Eng. Sci. Technol, vol. 15, no. 5, pp. 3034-3054, 2020.

[38] B. Tao and J.-H. Kim, "Mobile robot path planning based on bi-population particle swarm optimization with random perturbation strategy," Journal of King Saud University-Computer and Information Sciences, vol. 36, no. 2, p. 101974, 2024.

[39] J. Lian, W. Yu, K. Xiao, and W. Liu, "Cubic spline interpolation-based robot path planning using a chaotic adaptive particle swarm optimization algorithm," Mathematical problems in engineering, vol. 2020, no. 1, p. 1849240, 2020.

[40] B. Zhou, B. Huang, Y. Su, and C. Zhu, "Interleaved periodic event-triggered communications-based distributed formation control for cooperative unmanned surface vessels," IEEE Transactions on Neural Networks and Learning Systems, 2024.

[41] H. Peng, B. Huang, M. Jin, C. Zhu, and J. Zhuang, "Distributed finite-time bearing-based formation control for underactuated surface vessels with Levant differentiator," ISA transactions, vol. 147, pp. 239-251, 2024.

[42] F. A. Raheem and M. M. Badr, "Development of Modified path planning algorithm using artificial potential field (APF) based on PSO for factors optimization," American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS), vol. 37, no. 1, pp. 316-328, 2017.

[43] S. Balasubramanian, A. Melendez-Calderon, A. Roby-Brami, and E. Burdet, "On the analysis of movement smoothness," Journal of neuroengineering and rehabilitation, vol. 12, pp. 1-11, 2015.

[44] J. Guo, Y. Gao, and G. Cui, "Path planning of mobile robot based on improved potential field," Information Technology Journal, vol. 12, no. 11, p. 2188, 2013.

[45] Ó. Gil, A. Garrell, and A. Sanfeliu, "Social robot navigation tasks: Combining machine learning techniques and social force model," Sensors, vol. 21, no. 21, p. 7087, 2021.

VI. BIOGRAPHIES

MAHSA MOHAGHEGH Mahsa Mohaghegh is an Iranian-born New Zealand computer engineer specializing in artificial intelligence and natural language processing. She is a Senior Lecturer and Director of Women in Tech at Auckland University of Technology's School of Engineering, Computer and Mathematical Sciences. Mohaghegh completed her Bachelor's degree in computer engineering and her Master's degree in computer architecture before earning her doctorate in computer engineering from Massey University in 2013. She has been involved with Google's Computer Science for High Schools program since 2013 and runs workshops in Auckland. Mohaghegh founded She Sharp, a women's networking group aimed at encouraging girls and young women to engage with digital industries. She has received several awards, including the Emerging Leader category in the 2013 Westpac Women of Influence Awards, the 2019 YWCA Equal Pay Champion Award, and the 2020 Massey University Distinguished Alumni Award. Mohaghegh is a well-recognized leader in AI and machine learning and is committed to promoting diversity and inclusion in the tech industry.

HEDIEH JAFARPOURDAVATGAR Hedieh Jafarpourdavatgar, born on April 16, 1989, in Iran, is a dedicated engineer with a strong interest in optimization and control systems. She earned her Bachelor's degree in Electronic Engineering from the Islamic Azad University and completed her Master's degree in Control Engineering at Amirkabir University of Tehran (Polytechnic of Tehran). Throughout her academic and professional journey, Hedieh has consistently demonstrated her commitment to tackling complex challenges in the field of control engineering, contributing to advancements in the industry.

SAMANEH ALSADAT SAEEDINIA Samaneh Alsadat Saeedinia, born on March 29, 1989, in Qazvin, Iran, is a control engineer, holding a Bachelor of Science degree from Imam Khomeini International University and a Master of Science degree from Iran University of Science and Technology (IUST) (GPA: A). With a strong academic foundation, she is currently completed her Ph.D. in control engineering at IUST. Samaneh Alsadat's research expertise encompasses a wide range of applications in control theory and AI, demonstrated through her roles as a Research Assistant and Project Manager.