

RESEARCH AND EVALUATION OF THE EFFECTIVENESS OF MOTION PLANNING ALGORITHMS FOR ROBOTS OPERATING IN CROWDED ENVIRONMENTS

NGHIÊN CỨU VÀ ĐÁNH GIÁ HIỆU QUẢ CỦA CÁC THUẬT TOÁN LẬP KẾ HOẠCH CHUYỂN ĐỘNG CHO ROBOT HOẠT ĐỘNG TRONG MÔI TRƯỜNG ĐÔNG ĐÚC

Nghiem Hoang Nam^{1,*}, Nguyen Thi Lan Anh¹, Nguyen Viet Tiep¹,
Pham Tran Quyen Anh¹, Truong Xuan Tung¹

DOI: <https://doi.org/10.57001/huiv5804.2025.318>

ABSTRACT

Navigating robots in crowded environments presents significant challenges, particularly in applications involving service and assistive robots in public spaces such as airports, shopping malls, or train stations. Ensuring safe, efficient, and human-friendly robot navigation is a critical and pressing requirement. This paper analyzes and compares the performance of four prominent navigation algorithm groups: Social Force Model (SFM), Dynamic Window Approach (DWA), Hybrid Reciprocal Velocity Obstacles (HRVO), and Timed Elastic Band (TEB). These algorithms are evaluated in simulated environments using the Pedsim platform, across various scenarios with differing pedestrian densities levels. Evaluation criteria include collision avoidance, trajectory smoothness, task completion time, and the social acceptability of the robot's behavior. Simulation results reveal that each algorithm exhibits distinct advantages and limitations, depending on specific conditions. This study provides a valuable reference for selecting suitable navigation algorithms for mobile robots in complex, crowded environments.

Keywords: Hybrid Reciprocal Velocity Obstacles (HRVO), Timed Elastic Band Local Planner (TEB), Social Force Model (SFM), Dynamic Window Approach (DWA).

TÓM TẮT

Điều hướng robot trong môi trường đông người đặt ra nhiều thách thức, đặc biệt trong các ứng dụng liên quan đến robot dịch vụ và hỗ trợ con người tại những không gian công cộng như sân bay, trung tâm thương mại hoặc nhà ga. Việc bảo đảm robot có thể di chuyển an toàn, hiệu quả và thân thiện với con người là một yêu cầu quan trọng và cấp thiết. Trong bài báo này, chúng tôi tiến hành phân tích và so sánh hiệu suất của bốn nhóm thuật toán điều hướng phổ biến, bao gồm Social Force Model (SFM), Dynamic Window Approach (DWA), Hybrid Reciprocal Velocity Obstacles (HRVO) và Timed Elastic Band (TEB). Các thuật toán được đánh giá trong môi trường mô phỏng sử dụng nền tảng Pedsim, với nhiều kịch bản khác nhau về mật độ người di chuyển. Các tiêu chí so sánh bao gồm khả năng tránh va chạm, độ mượt của quỹ đạo, thời gian hoàn thành nhiệm vụ và mức độ thân thiện xã hội của robot. Kết quả mô phỏng cho thấy mỗi thuật toán có những ưu điểm và hạn chế riêng, phụ thuộc vào từng điều kiện cụ thể. Nghiên cứu này cung cấp cơ sở tham khảo hữu ích cho việc lựa chọn thuật toán điều hướng phù hợp trong các ứng dụng robot di động trong môi trường phức tạp và đông người.

Từ khóa: Hybrid Reciprocal Velocity Obstacles (HRVO), Timed Elastic Band Local Planner (TEB), Social Force Model (SFM), Dynamic Window Approach (DWA).

¹Institute of Control Engineering, Le Quy Don Technical University, Vietnam

*Email: nghiemhoangkhanh@gmail.com

Received: 20/6/2025

Revised: 10/9/2025

Accepted: 28/9/2025

1. INTRODUCTION

Safe and efficient navigation for mobile robots in crowded environments represents a significant challenge

within service robotics research. Real-world settings, such as airport terminals, shopping malls, or train stations, typically feature high pedestrian density, complex

movement behaviors, and implicit social norms, such as maintaining personal space, giving priority to groups, or sustaining consistent walking directions. In these situations, robots must not only avoid collisions with people and objects but also exhibit socially appropriate navigation behaviors, respecting social norms and ensuring human comfort and expectations.

To meet these requirements, local motion planning plays a crucial role by enabling robots to respond flexibly and adaptively to their immediate surroundings in real time. Over recent years, numerous local motion planning algorithms have been proposed and widely applied, notably Social Force Model (SFM), Dynamic Window Approach (DWA), Timed Elastic Band (TEB), and Hybrid Reciprocal Velocity Obstacles (HRVO).

The Social Force Model (SFM) was initially proposed by Helbing and Molnár in 1995, describing goal-directed and collision avoidance forces based on interpersonal distances [1]. Reference [2] further expanded SFM by integrating personalized space mechanisms to minimize collisions in crowds. Reference [3] analyzed trajectory oscillation phenomena frequently encountered in SFM, particularly when applied to robotic platforms. Research [4] demonstrated that integrating SFM with AutoRL enhanced the navigation capability of differential-drive robots in crowded environments. Reference [5] proved that improved SFM with three force components allows efficient robot navigation in densely populated areas with low computational overhead. Additionally, Reference [6] developed SRM, derived from SFM, which performed better in generating socially compliant trajectories.

Despite widespread adoption and positive outcomes, the traditional SFM model exhibits several limitations. Specifically, Reference [4] indicated that traditional SFM lacks adaptability to diverse social contexts and requires manual parameter adjustments to maintain effective navigation across different environments. Reference [5] emphasized the dependency of SFM on sensor data quality; noisy or delayed sensor data can destabilize the robot's trajectory. Additionally, research [7] concluded that without supplementary supporting methods, SFM offers only moderate performance improvements.

The Dynamic Window Approach (DWA), proposed by Fox et al., is a real-time local navigation method selecting optimal trajectories based on an objective function [8]. This function incorporates safety, speed, and goal-directed criteria. Reference [9] optimized this function to

handle narrow spaces effectively, while [10] extended DWA for urban autonomous vehicles. References [11-13] enhanced the objective function to reduce collision rates, maintaining trajectory smoothness and efficiency even in environments with high densities of dynamic obstacles.

Limitations of DWA include susceptibility to local minima [8], lack of global planning capabilities [13], and difficulties avoiding fast-moving obstacles [14]. Additionally, DWA performance heavily depends on manually tuning objective function weights [11].

Another prominent method, the Timed Elastic Band (TEB), is recognized for effectively handling dynamic and temporal constraints. It generates smooth trajectories suitable for differential-drive robots [15]. TEB's strength lies in its simple constraint structure, enabling fast and efficient trajectory planning, easily customizable for various robot types and environments [16, 17]. Reference [20] combined TEB with Theta* to enhance global planning effectiveness, achieving shorter and safer paths in complex terrains. Additionally, Reference [18] improved TEB by incorporating dynamic obstacle velocity information (acquired from lidar and Kalman filtering), enabling future position prediction and real-time trajectory adjustments. Reference [19] developed RTEB using A* for replanning upon TEB failure, combined with soft constraint smoothing techniques to enhance adaptability in dense obstacle environments.

However, some studies highlight TEB's limitations. Reference [18] noted TEB's shortcomings in accurately predicting dynamic obstacle movements in dense environments. Reference [19] indicated increased computational time with high obstacle densities, causing planning disruptions without supplementary recovery strategies. Furthermore, Reference [16] underscored TEB's significant dependency on manual parameter tuning, limiting generalizability and scalability.

Lastly, HRVO, an extension of Reciprocal Velocity Obstacles (RVO) [21], aims to mitigate oscillations arising from reciprocal interactions in multi-agent navigation. HRVO enables robots to predict surrounding agents' behaviors, selecting optimal velocity directions for smooth, safe navigation in dynamic environments. Reference [22] demonstrated that combining HRVO with Monte Carlo velocity sampling produces smooth, safe trajectories in human-shared workspaces, even without centralized coordination. However, HRVO's effectiveness diminishes in highly crowded environments or with non-reciprocal agents, necessitating extended variants such

as Emotional RVO (ERVO). Additionally, Reference [23] pointed out minor oscillations in HRVO trajectories with unevenly moving agents. To address this, the Fuzzy Interacting Multiple-Model VO (FIMVO), integrating fuzzy logic, was proposed, enhancing trajectory smoothness and stability but significantly increasing computational costs due to parallel model predictions.

From these studies, each navigation algorithm group (SFM, DWA, TEB, HRVO) shows distinct advantages and limitations, contingent upon specific environmental conditions and application objectives. Given that mobile robots must operate effectively in complex, interactive environments such as smart factories, hospitals, or public spaces selecting appropriate navigation algorithms becomes critical. Therefore, this study aims to create diverse, realistic simulation scenarios to evaluate, analyze, and compare these algorithm groups. Results will provide comprehensive insights, supporting the selection of suitable navigation solutions for modern mobile robot applications.

The remainder of the paper is structured as follows: Section 2 presents the mathematical foundations of the SFM, DWA, TEB, and HRVO algorithms. Experiments and results evaluation are described in Section 3. Section 4 concludes the paper.

2. MATHEMATICAL BASIS OF SFM, DWA, TEB, HRVO ALGORITHMS

This section presents the mathematical foundations and operational principles of four popular local navigation algorithms: SFM, DWA, TEB, and HRVO. Each algorithm is constructed based on different assumptions and models, ranging from social force modeling and velocity-space sampling to real-time trajectory optimization and predictive reciprocal collision avoidance. A detailed analysis of their mathematical formulations not only clarifies the underlying nature of each method but also provides a basis for evaluating their strengths and limitations across complex environmental scenarios.

2.1. Social Forces Model (SFM) Algorithm

SFM was initially proposed to simulate pedestrian movements within crowds using imaginary social forces [1]. In robotic navigation applications, particularly for differential-drive robots, SFM is adapted to represent the robot's motion as a resultant combination of goal-directed forces, forces exerted by other agents, and obstacle avoidance forces.

The total force acting on robot i at time t is expressed as follows:

$$m_i \frac{dv_i}{dt} = F_i(t) = F_{\text{goal}}(t) + \sum_{j=1} F_{ij}(t) + \sum_k F_{ik}^{\text{obs}}(t) \quad (1)$$

where $F_{\text{goal}}(t)$ is the force driving the robot towards its goal, described as follows:

$$F_{\text{goal},i}(t) = \frac{1}{\tau_i} (v_i^0 - v_i(t)) \quad (2)$$

where v_i^0 is the desired speed, $v_i(t)$ is the current velocity, and τ_i is the relaxation time constant. The social force $F_{ij}(t)$ between robot i and agent j is modeled as:

$$F_{ij}(t) = A_i \exp\left(\frac{d_{ij}}{B_i}\right) n_{ij} \quad (3)$$

where A_i và B_i are constants controlling the interaction strength and range, d_{ij} is the distance between robot i and agent j , and n_{ij} là vector đơn vị hướng t is the unit vector from i to j .

Similarly, the repulsive force from obstacles $F_{ij}^{\text{obs}}(t)$ is defined as:

$$F_{ik}^{\text{obs}}(t) = C_i \exp\left(\frac{d_{ik}}{D_i}\right) n_{ik} \quad (4)$$

where C_i and D_i are the strength and influence range coefficients of the obstacle force, d_{ik} is the distance from robot i to obstacle k , and n_{ik} is the unit vector pointing from i to k .

After computing the total force, the resulting acceleration $a_i(t)$ is used to update the linear velocity v and angular velocity ω according to the differential-drive kinematics:

$$v = v + a_x \cdot \Delta t, \quad \omega = \omega + a_\theta \cdot \Delta t \quad (5)$$

where a_x and a_θ are the linear and angular components of the acceleration, respectively, derived from the total force $F_i(t)$.

The SFM enables the generation of natural trajectories that exhibit social behavior, such as maintaining interpersonal distances and avoiding collisions. However, its performance remains highly dependent on parameter tuning and data quality. Noisy sensor data or inadequate model generalization may hinder robust performance in real-world environments. To address these limitations, recent studies [5, 7] have proposed improved force formulations and learning-based enhancements. Moreover, [8] extends the social capability of SFM for more adaptive robot service navigation

2.2. Dynamic Window Approach (DWA) Algorithm

The Dynamic Window Approach (DWA) is a real-time local planning algorithm proposed by Fox et al., designed to generate feasible trajectories that comply with the robot's kinematic constraints while ensuring collision avoidance in dynamic environments [8]. The algorithm exploits the differential-drive kinematics of the robot to produce curved trajectories, each defined by a pair of linear and angular velocities (v, ω) in the velocity space.

DWA restricts the search space for velocity candidates to a so-called "dynamic window", which is derived based on the current velocity and the robot's acceleration limits. The feasible velocity window is defined as follows:

$$V_d = \{ (v, \omega) \mid v_{min} \leq v \leq v_{max}, \omega_{min} \leq \omega \leq \omega_{max} \} \tag{6}$$

In which the velocity limits depend on the current velocity and the maximum acceleration of the robot.

$$v_{min} = v(t) - a_{lin} \cdot \Delta t, \quad v_{max} = v(t) + a_{ang} \cdot \Delta t \tag{7}$$

$$\omega_{min} = \omega(t) - a_{lin} \cdot \Delta t, \quad \omega_{max} = \omega(t) + a_{ang} \cdot \Delta t \tag{8}$$

Specifically, v : the robot's current linear velocity (m/s), ω : the robot's current angular velocity (rad/s), (v_{min}, v_{max}) : the lower and upper bounds of the linear velocity achievable within a short time interval, $(\omega_{min}, \omega_{max})$: the lower and upper bounds of the angular velocity achievable within that time interval.

The set v_d represents the "dynamic window", which defines the range of velocities that the robot can immediately achieve based on its acceleration and deceleration capabilities over a short time interval (Δt).

$v(t)$ denotes the current linear velocity at time t ; $\omega(t)$ is the current angular velocity; a_{lin} is the maximum linear acceleration of the robot (m/s^2); a_{ang} is the maximum angular acceleration (rad/s^2); and Δt refers to a short prediction time window, typically ranging from 0.1 to 0.5 seconds.

Each trajectory is evaluated using a cost function composed of three main criteria:

$$G(v, \omega) = \alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{clearance}(v, \omega) + \gamma \cdot \text{velocity}(v) \tag{9}$$

In this context, the term $\text{heading}(v, \omega)$ quantifies the alignment toward the goal (typically, a smaller angular deviation is preferred), $\text{clearance}(v, \omega)$ represents the minimum distance to obstacles along the trajectory, and $\text{velocity}(v, \omega)$ favors higher translational velocities to minimize travel time. The weighting coefficients α, β, γ are used to balance the relative importance of each criterion.

Among the set of generated trajectories, the one with the highest cost function value $G(v, \omega)$ is selected. The corresponding velocity pair (v, ω) is then passed to the robot's kinematic controller to execute the optimized motion.

The main advantages of DWA include its simplicity, fast computation, and the ability to ensure emergency stops due to its direct incorporation of dynamic constraints. However, the algorithm is prone to getting stuck in local minima, exhibits limited responsiveness to fast-moving obstacles, and is highly sensitive to the tuning of the weighting parameters α, β, γ .

Subsequent research has extended the original DWA framework by integrating it with global path planning [13], fuzzy logic [11], reinforcement learning [12], and motion prediction techniques to enhance performance in dynamic and densely populated environments.

2.3. Timed Elastic Band (TEB) Algorithm

The Timed Elastic Band (TEB) algorithm is an online local trajectory optimization approach for navigation and control of mobile robots [15]. Accordingly, it is employed to predict the robot's motion trajectory. The core principle of the TEB technique is to generate locally optimal trajectories by minimizing the execution time while simultaneously ensuring obstacle avoidance and adhering to dynamic constraints, such as velocity and acceleration limits. TEB represents an extension of the "Elastic Band" algorithm and has been successfully implemented in the `teb_local_planner` package within the Robot Operating System (ROS) framework. In TEB, an initial trajectory is provided by the global planner and subsequently optimized during runtime. The optimization process aims to minimize the trajectory execution time, maintain a safe distance from obstacles, and satisfy kinodynamic constraints, including the robot's velocity and acceleration bounds.

The trajectory of the "elastic band" is represented as a sequence of N robot poses $\mathbf{S}_k = [x_r^k, y_r^k, \theta_r^k]^T \in \mathbb{R}^2 \times \mathbb{S}^1$ (for $k = 1, 2, \dots, N$), where (x_r^k, y_r^k) denotes the position of the robot and θ_r^k denotes its orientation. However, a limitation of this technique is the absence of temporal information, which results in a trajectory that is not optimized with respect to time.

To overcome this limitation, the Timed Elastic Band (TEB) method extends the original Elastic Band (EB) approach by incorporating temporal information. TEB defines the time sequence for traversing from the start to

the goal as a set of time intervals ΔT_k between consecutive poses, (with $k = 1, 2, \dots, N-1$). Consequently, the trajectory of the autonomous robot is formally defined as follows:

$$B=[s_1, \Delta T_1, s_2, \Delta T_2, \dots, s_{N-1}, \Delta T_{N-1}, s_N]^T \quad (10)$$

To optimize the TEB, it is necessary to simultaneously optimize both the robot's states and the time intervals between consecutive states. The TEB optimization problem is formulated as the minimization of the following sequence:

$$\min_B \sum_{k=1}^{N-1} \Delta T_k^2 \quad (11)$$

When the TEB optimization problem is reformulated as a weighted multi-objective optimization problem, it becomes dependent on the following conditions:

The optimization problem in TEB is subject to several constraints, including: the time interval between two consecutive states, $0 \leq \Delta T_k \leq \Delta T_{\max}$; the safety distance to obstacles, $P_k(s_k) \geq 0$; non-holonomic kinematic constraints, $h_k(s_{k+1}, s_k) = 0$; velocity limits, $v_k(s_{k+1}, s_k, \Delta T_k) \geq 0$; and acceleration limits: $\alpha_k(s_{k+1}, s_k, s_{k-1}, \Delta T_k, \Delta T_{k-1}) \geq 0$.

The objective function is expressed as follows:

$$\begin{aligned} B^* &= \operatorname{argmin}_{B \setminus \{s_1, s_N\}} V(B); \\ V(B) &= \sum_{k=1}^{N-1} \left(\Delta T_k^2 + \delta_h \|h_k\|_2^2 + \delta_v \|\min\{\mathbf{0}, v_k\}\|_2^2 \right. \\ &\quad \left. + \delta_o \|\min\{\mathbf{0}, P_{\text{obs}}^k\}\|_2^2 + \delta_a \|\min\{\mathbf{0}, \alpha_k\}\|_2^2 \right) \quad (12) \\ &= \mathbf{w}^T \mathbf{f}(B) \end{aligned}$$

In this context, w^T represents the individual weight, and $f(B)$ encapsulates the individual cost conditions. Finally, the desired designed control command is directly extracted from the optimal control sequence B^* .

2.4. The Hybrid Reciprocal Velocity Obstacles (HRVO) algorithm

The HRVO algorithm, an extension of the Reciprocal Velocity Obstacles (RVO) approach, enables robots or agents to navigate safely in dynamic multi-agent environments by facilitating more effective collision-avoidance velocity selections [21]. While RVO assumes all agents are capable of reciprocal collision avoidance, HRVO mitigates undesirable velocity oscillations and enhances trajectory stability by modifying the velocity obstacle region.

An agent A, with position \bar{p}_A and velocity \bar{v}_A , avoids collisions with agent B, having position \bar{p}_B and velocity \bar{v}_B by selecting a new velocity \bar{v}_{new} that lies outside the HRVO set, defined as:

$$\text{HRVO}_{A|B} = \left\{ \bar{v} \left| \left(\bar{v} - \left(\bar{v} \right)_{AB}^{\text{HRVO}} \right) \in \text{VO}_{A|B} \right. \right\} \quad (13)$$

where $\text{VO}_{A|B}$ represents the velocity obstacle region between A and B, and $\bar{v}_{AB}^{\text{HRVO}}$ is the modified reciprocal velocity.

$$\bar{v}_{AB}^{\text{HRVO}} = \bar{v}_B + \frac{1}{2} (\bar{v}_A - \bar{v}_B) \quad (14)$$

The agent then selects the optimal velocity \bar{v}_{new} that is closest to its preferred velocity \bar{v}_{pref} , such that

$$\bar{v}_{\text{new}} \notin \text{HRVO}_{A|B} :$$

$$\bar{v}_{\text{new}} = \operatorname{argmin}_{\bar{v} \in \text{HRVO}} \|\bar{v} - \bar{v}_{\text{pref}}\| \quad (15)$$

The algorithm is extended to handle multiple agents by considering the union of HRVO regions between the robot and all other surrounding agents. Additionally, the robot's kinematic constraints (e.g., for a differential-drive or omnidirectional robot) are incorporated by restricting the feasible velocity space based on the robot's motion model.

HRVO is an efficient real-time collision avoidance algorithm, particularly well-suited for environments with high agent density. However, it still exhibits several limitations, including a strong dependence on localization accuracy and the lack of comprehensive integration of social constraints in navigation. Recently, variants such as FIMVO [23] and models incorporating social cost maps [22] have been proposed to address these shortcomings.

3. EXPERIMENTS AND EVALUATION

In the context of simulating robot navigation through crowds, selecting a suitable simulation environment plays a crucial role in ensuring the accuracy and effectiveness of algorithm evaluation. Compared to comprehensive physics-based platforms such as Gazebo, Pedsim offers several significant advantages due to its focus on modeling pedestrian (agent) behavior. Specifically, Pedsim is designed to simulate the motion of multiple agents in 2D environments, with built-in capabilities to reproduce key social behaviors such as collision avoidance, group dynamics, personal space preservation, and consistent directional movement.

Additionally, Pedsim features a lightweight architecture that delivers high computational efficiency, enabling the simulation of complex and crowded scenarios in real time without requiring substantial hardware resources. Furthermore, Pedsim integrates seamlessly with the ROS ecosystem, which facilitates the deployment and testing of local motion planning algorithms. While Gazebo provides accurate 3D physical simulation and detailed sensor integration, constructing socially interactive scenarios in densely populated environments using Gazebo is often more complex and resource-intensive.

Therefore, Pedsim is considered an appropriate platform for this study, as our focus is on evaluating the robot's navigational behavior in high-density human environments.

3.1. Scenario Design and Simulation Setup

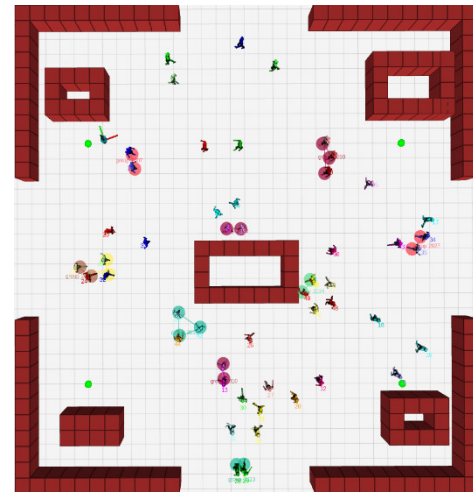
To evaluate the effectiveness of navigation algorithms in crowded environments, we designed three simulation scenarios using Pedsim integrated within the ROS ecosystem.

Scenarios I and II emulate a typical indoor space in a shopping mall, as illustrated in Fig. 1(a), with an area of 30m × 30m. Scenario III represents a more open layout resembling public spaces such as train stations or airport terminals, as shown in Fig. 1(b). Each environment includes both static and dynamic obstacles. Static obstacles consist of walls, tables, and shelves, reflecting realistic architectural features. For dynamic obstacles, we deploy 40, 80, and 150 agents moving freely at an average speed of 0.5m/s in Scenarios I, II, and III, respectively. Some of these agents are grouped into clusters of 2 to 5 individuals to simulate typical social interactions in dense human environments.

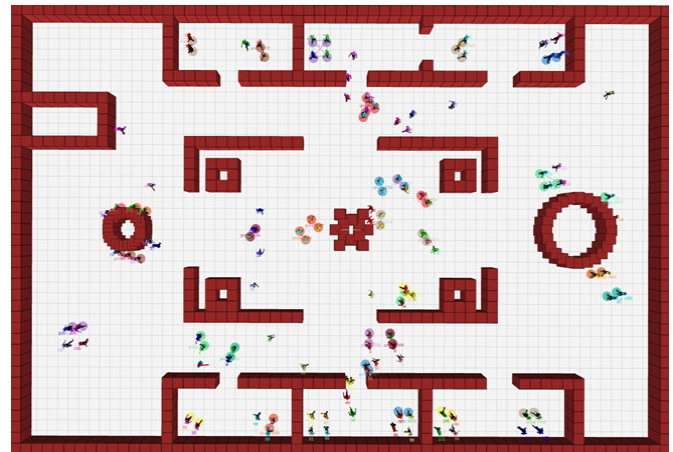
The robot is tasked with navigating to predefined goal points, aiming to achieve safe and efficient motion while adhering to socially acceptable behaviors. These scenarios are designed to comprehensively assess the performance of various navigation algorithms under conditions closely resembling real-world crowded settings.

To quantitatively evaluate robot navigation performance under the selected algorithms, the following metrics are employed: (i) Total travel distance - calculated as the sum of trajectory lengths from start to goal; (ii) Average velocity - defined as the ratio of total distance traveled to execution time; (iii) Task completion

time - measured from the time the robot starts moving until it reaches its destination. Additionally, two social interaction metrics, the Social Individual Index (SII) and Social Group Index (SGI), adopted from [24], are used to assess the robot's ability to navigate harmoniously among humans in crowded environments.



a)



b)

Fig.1. Simulated environment for Scenarios I and II (a), and for Scenario III (b)

Table 1. Experimental parameters

Parameter	Value	Parameter	Value	Parameter	Value
v_{max}	1.0 [m/s]	a_{max}	0.1 [m/s ²]	ω_{max}	0.5 [rad/s]
r_r	0.25 [m]	r_h	0.25 [m]	d_p	0.5 [m]
d_c	0.9 [m]	T_p	0.54	T_c	0.14

The parameters used in the experiments and the SII and SGI indicators are listed in Table 1, selected according to the personal space definitions from Hall's proxemics model [25].

Specifically, the radius of both human agents r_h and the robot r_r is set to 0.25m, resulting in a physical safety

distance d_p of 0.5m between the robot and a person. The psychological safety distance d_c is set to 0.9m. Psychological safety is considered preserved when the SII value remains below the threshold $T_c = 0.14$ (i.e., the relative distance between the robot and a human is greater than 0.9m). In contrast, physical safety is considered violated indicating a collision risk if the SII value exceeds the physical threshold $T_p = 0.54$ (i.e., the distance between the robot and a human falls below 0.5m).

For the SFM algorithm, the social force parameters are adjusted so that pedestrians maintain an average interpersonal distance of 0.5 to 0.6m. In the case of DWA and TEB, the robot utilizes a costmap with a safety buffer of 0.3m around each agent; TEB additionally uses a straight-line global path connecting the start and goal positions. For HRVO, the robot updates the position and velocity of agents every 0.1 seconds, assuming constant velocities over short time intervals. To gather sufficient experimental data, simulations were executed in a looped manner until more than 100,000 samples were collected, with each sample corresponding to one simulation timestep.

3.2. Scenario I: Moving in an environment with a density of 40 people

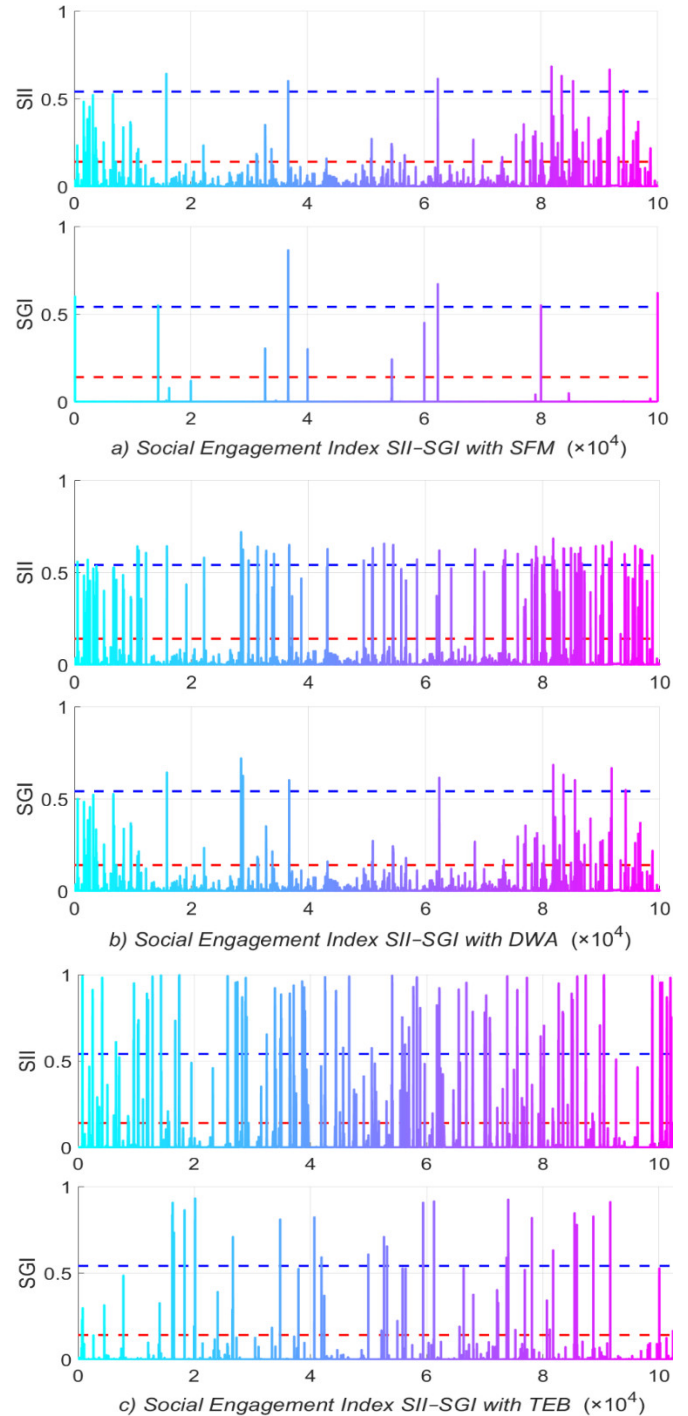
The robot was navigated within a laboratory setting, as depicted in Fig. 1a, with four distinct algorithms implemented independently using identical parameter configurations (Table 1). All algorithms achieved a 100% task completion rate. However, the results presented in Table 2 reveal performance disparities: TEB demonstrated the shortest travel distance and minimum completion time, whereas HRVO achieved the highest average velocity (0.72m/s).

Table 2. Quantitative results - Scenario I (40 people, 100,000 samples)

Algorithm	Success Rate	Distance (m)	Time (s)	Speed (m/s)
SFM	100%	3962	3910	0.65
DWA	100%	3875	3892	0.61
TEB	100%	3860	3866	0.62
HRVO	100%	4012	3981	0.72

Social interaction capabilities were evaluated using the Social Individual Index (SII) and Social Group Index (SGI), as depicted in Fig. 2. The results indicate that HRVO (Fig. 2d) consistently maintained both indices at low and stable levels (averaging approximately 0.12 and 0.08, respectively), demonstrating effective spatial distribution

in dynamic environments. Conversely, DWA and TEB (Figs. 2b and 2c) exhibited multiple peaks exceeding thresholds of 0.2 - 0.5, reflecting instances of overly close individual approaches or intrusions into human groups. While SFM (Fig. 2c) generated more stable motion, it still showed some phases of close proximity to human groups, leading to a slight increase in the SGI. These findings suggest that HRVO possesses higher potential for deployment in environments demanding a significant degree of social interaction.



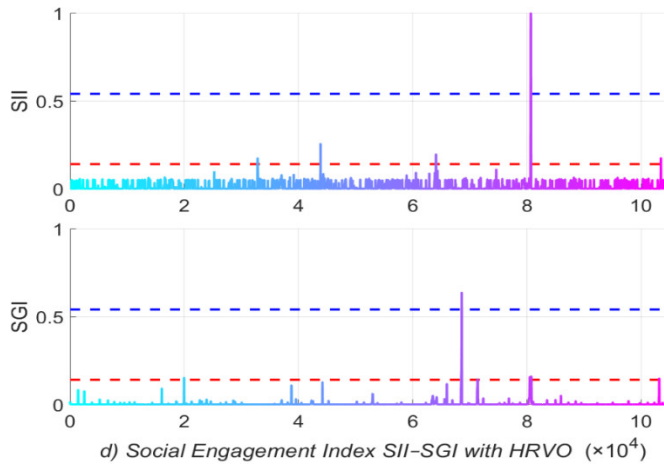


Fig. 2. Comparison of SII and SGI indices of SFM and HRVO in Scenario I

3.3. Scenario II: Moving in an environment with a density of 80 people

The second scenario was conducted in the same environment as the first (Fig. 1a), but with an increased human presence of 80 individuals to simulate a rush hour setting. The evaluation results, presented in Table 3, indicate that only HRVO and SFM maintain stable navigation capabilities, achieving a 100% successful movement rate. Conversely, the success rates for TEB and DWA were notably low in these experiments. The primary reason for this limited performance is the high human density, which creates blockages that prevent TEB and DWA from formulating feasible plans and consistently leads them into local minima.

Table 3. Quantitative results - Scenario II (80 people, 100,000 samples)

Algorithm	Success Rate	Distance (m)	Time (s)	Speed (m/s)
HRVO	100%	4635.12	6952	0.68
SFM	100%	4471.52	6703	0.64
TEB	0%	–	–	–
DWA	0%	–	–	–

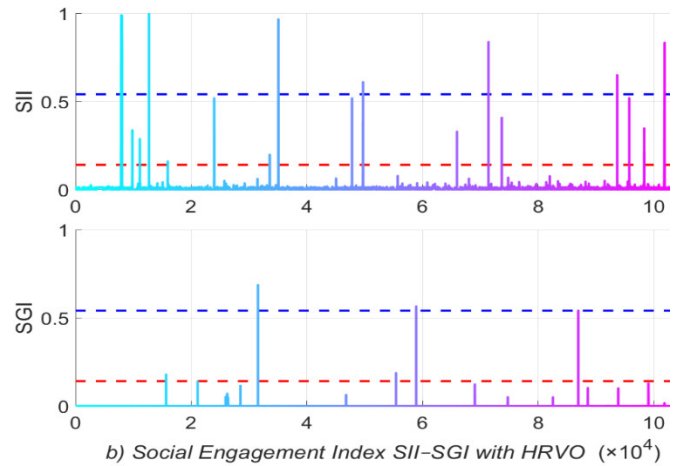
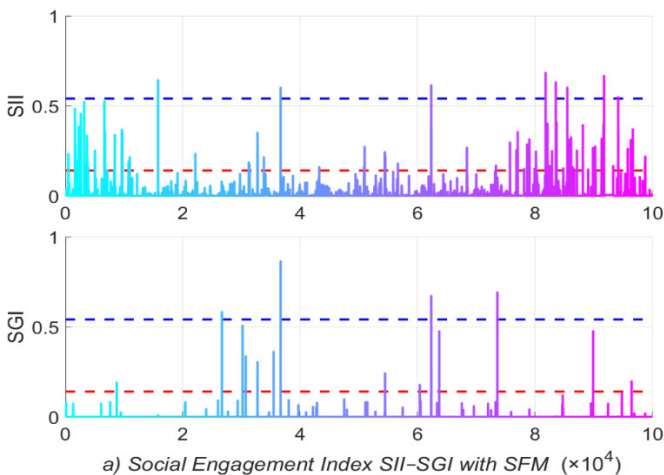


Fig. 3. Comparison of SII and SGI indices of SFM and HRVO in Scenario II

Table 3 reveals that SFM held an advantage in terms of travel distance and completion time in high-density scenarios, while HRVO demonstrated superior travel velocity. TEB and DWA frequently became trapped in confined areas, failing to find valid paths through clusters of people blocking their way. This led to significantly increased computation times and an inability to move.

Furthermore, Fig. 3b illustrates that HRVO's Social Individual Index (SII) and Social Group Index (SGI) remained stable around 0.17, indicating consistent and effective social navigation. In contrast, SFM (Fig. 3a) exhibited considerable fluctuations and peaks exceeding acceptable thresholds, suggesting less consistent social interaction. TEB and DWA's social interaction metrics are not represented due to an insufficient number of successful trials to generate a stable distribution.

3.4. Scenario III: Robot Navigation in a Shopping Mall Environment

To further assess the social navigation capabilities of the algorithms, we developed a third scenario simulating the main hall of a shopping mall. In this scenario, the robot is tasked with navigating through a sequence of fixed waypoints under conditions of high pedestrian density. Social interaction metrics, including the Social Individual Index (SII) and Social Group Index (SGI), continue to be employed to quantitatively evaluate the quality of the robot's navigation.

The results, illustrated in Fig. 4, reveal a clear distinction between the two algorithms. As shown in Fig. 4b, the HRVO algorithm maintains a stable SII—mostly below the 0.14 threshold and keeps SGI values under 0.54, indicating effective social distancing from both individuals and groups. In contrast, the SFM algorithm (Fig. 4a) exhibits frequent spikes, with SII reaching its

maximum value (1.0) and SGI exceeding the 0.54 threshold during navigation. This suggests that the robot often approaches too closely or even moves through dense crowds when guided by SFM.

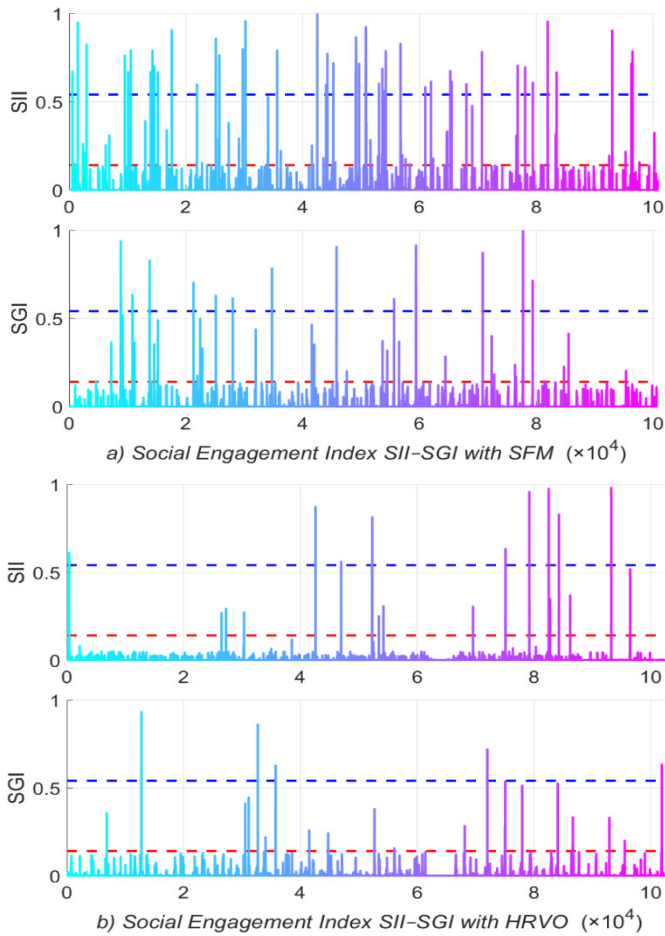


Fig. 4. Comparison of SII and SGI metrics between the SFM and HRVO algorithms in Scenario III

Through the evaluation results of the three scenarios, we see the three scenarios indicate that in low-density environments (with 40 agents), all four algorithms achieved high task completion rates, with negligible differences in travel distance and time. However, as the crowd density increased to 100 and 150 agents, performance differences became more pronounced. DWA exhibited fast response and algorithmic simplicity but was prone to local minima, often leading to fragmented paths or the robot getting stuck in crowded regions. TEB produced moderately smooth trajectories but frequently failed in constrained spaces where maneuverability was limited. SFM demonstrated strong compliance with social behavior norms, enabling the robot to avoid humans in a natural manner. Nevertheless, it lacked decisiveness when encountering blocked paths, resulting in suboptimal navigation under high-density

conditions. In contrast, HRVO consistently outperformed the others by effectively avoiding collisions and maintaining stable trajectories, even in densely populated environments.

4. CONCLUSION

This paper presented and evaluated four local navigation algorithms SFM, DWA, TEB, and HRVO across three simulation scenarios with increasing agent density (40, 80, and 150 agents). Experimental results demonstrate that each algorithm exhibits distinct strengths and limitations depending on the density and interaction complexity of the environment.

Based on performance metrics including path length, task completion time, and social interaction indices (SII and SGI), HRVO and SFM emerged as the most effective methods in crowded environments. HRVO, in particular, offered a favorable balance between navigation efficiency, human avoidance capability, and social distancing adherence. Building upon these findings, future work will focus on integrating the trajectory optimization strengths of TEB with the socially-aware, reactive collision avoidance mechanism of HRVO. This hybrid approach aims to produce smoother and shorter paths while maintaining safe distances from nearby humans even in highly congested areas.

In addition, we plan to enhance the robot's social behavior model by incorporating dynamic constraints such as traffic flow priorities and recognition of group interaction dynamics. These improvements are expected to generate more natural and socially acceptable motion patterns, contributing to the robot's seamless integration into public spaces.

REFERENCES

- [1]. D. Helbing, P. Molnár, "Social force model for pedestrian dynamics," *Physical Review E*, 51, 5, 4282-4286, 1995.
- [2]. M. Moussaïd, D. Helbing, G. Theraulaz, "How simple rules determine pedestrian behavior and crowd disasters," *Proc. Natl. Acad. Sci. U.S.A.*, 108, 17, 6884-6888, 2011.
- [3]. C. Rudloff, A. Seer, N. Matyus, "Validation of a social force model of pedestrian movement with real life data," *Transportation Research Procedia*, 2, 724-732, 2015.
- [4]. Y. Fan, et al., "Social Robot Navigation Tasks: Combining Machine Learning Techniques and Social Force Model," *Frontiers in Robotics and AI*, 2022.

- [5]. Y. Li, et al., "A Social Force Model for Multi-Agent Systems With Application to Robots Traversal in Cluttered Environments," *IEEE Access*, 12, 2024.
- [6]. P. Wang, et al., "Mobile Robot Navigation for Human-Robot Social Interaction Using Social Force Model," in **IEEE Access**, vol. 11, 2023.
- [7]. Z. Chen et al., "Effects of a Social Force Model Reward in Robot Navigation Based on Deep Reinforcement Learning," *Robotics and Autonomous Systems*, 2023.
- [8]. D. Fox, W. Burgard, S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics and Automation Magazine*, 4, 1, 23-33, 1997.
- [9]. D. Seder and I. Petrovic', "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," in **Proc. ISIE**, 2007.
- [10]. H. Zhang, Z. Liu, "A Modified Dynamic Window Approach for Motion Planning of Autonomous Vehicles," *Journal of Intelligent Robotic Systems*, 91, 313-324, 2018.
- [11]. M. Kumar. S. Singh, "Improved Dynamic Window Approach Based on Energy Consumption Management and Fuzzy Logic Control for Local Path Planning of Mobile Robots," in *Proc. ICCNT*, 2020.
- [12]. Q. Li, H. Wang, "An Improved Dynamic Window Approach Based on Reinforcement Learning for the Trajectory Planning of Automated Guided Vehicles," *Sensors*, 21, 5, 2021.
- [13]. Y. Ren, H. Wang, "A Global Path Planning Algorithm Based on the Feature Map," in *Proc. ICCSIT*, 2010.
- [14]. X. Wang, Z. Liu, "Dynamic Obstacle Avoidance Control Based on a Novel Dynamic Window Approach for Agricultural Robots," *Applied Sciences*, 12, 6, 2022.
- [15]. C. Rosmann, W. Feiten, T. Wosch, F. Hoffmann, and T. Bertram, "Trajectory modification considering dynamic constraints of autonomous robots," in *7th German Conference on Robotics*, 1{6}, 2012.
- [16]. J. Wu, et al., "An Improved Timed Elastic Band (TEB) Algorithm of Autonomous Ground Vehicle (AGV) in Complex Environment," *Sensors*, 21, 24, 2021.
- [17]. J. S. Smith, et al., "egoTEB: Egocentric, Perception Space Navigation Using Timed-Elastic-Bands," in *Proc. ICRA*, 2020.
- [18]. X. Ou, et al., "Local Path Planner for Mobile Robot Considering Future Positions of Obstacles," *Processes*, 12, 984, 2024.
- [19]. G. Kulathunga, et al., "Resilient Timed Elastic Band Planner for Collision-Free Navigation in Unknown Environments," *arXiv preprint arXiv:2412.03174*, 2024.
- [20]. J. Sun, et al., "Path Planning Algorithm for a Wheel-Legged Robot Based on the Theta* and Timed Elastic Band -Algorithms," *Symmetry*, 15, 1091, 2023.
- [21]. J. Snape, J. Van den Berg, S. Guy, D. Manocha, "The hybrid reciprocal velocity obstacle," *IEEE Transactions on Robotics*, 27, 4, 696-706, 2011.
- [22]. J. Douthwaite, et al., "Decentralized Human-Aware Navigation in Shared Human-Robot Workspaces Based on Velocity Obstacles," *IEEE Transactions on Robotics*, 40, 2024.
- [23]. Y. Liu, et al., "A Fuzzy Interacting Multiple-Model Velocity Obstacle (FIMVO) Approach," *Proc. ROBIO*, 2023.
- [24]. Truong, Xuan-Tung, Trung-Dung Ngo, "To approach humans?: A unified framework for approaching pose prediction and socially aware robot navigation," *IEEE Transactions on Cognitive and Developmental Systems*, 10, 3: 557-572, 2017.
- [25]. E. T. Hall, *The Hidden Dimension: Man's Use of Space in Public and Private*. The Bodley Head, 1966.

THÔNG TIN TÁC GIẢ

**Nghiêm Hoàng Nam, Nguyễn Thị Lan Anh, Nguyễn Việt Tiệp,
Phạm Trần Quyền Anh, Trương Xuân Tùng**

Viện Tên lửa và Kỹ thuật điều khiển, Học viện Kỹ thuật Quân sự