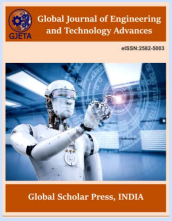
Global Journal of Engineering and Technology Advances
eISSN: 2582-5003
Cross Ref DOI: 10.30574/gjeta
Journal homepage: https://gjeta.com/



(RESEARCH ARTICLE)

Check for updates

Reducing planning time in swarm exploration via quorum sensing and discrete-event scheduling

Fredy Hernán Martínez Sarmiento *

Facultad Tecnológica, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia.

Global Journal of Engineering and Technology Advances, 2025, 24(03), 021-033

Publication history: Received on 22 July 2025; revised on 28 August 2025; accepted on 1 September 2025

Article DOI: https://doi.org/10.30574/gjeta.2025.24.3.0258

Abstract

This paper presents an event-driven framework for multi-agent exploration that couples a lightweight scheduler and blackboard with a Quorum Sensing (QS) coordination layer. Robots emit and sense a diffusive field using two scalar probes and a contact signal, select one of four motion primitives from local gradients, and make fully asynchronous, arrival-triggered decisions without global rounds. The QS field is integrated on a grid with explicit diffusion, decay, and no-flux obstacle boundaries, while motion uses arrival-time timestamps to reduce intermediate events and avoid idle time. Across six layouts (200×300 cells) and swarms of 4–20 agents, the QS policy lowered planning time by about 33% and route imbalance by about 31% relative to an asynchronous non-QS baseline; average path length remained competitive and repeated length fell by roughly 17%. Ablations (no circumnavigation, fixed emissions, single-probe sensing) degraded at least one metric, clarifying the role of each mechanism, and scheduler wall-time scaled nearly linearly up to six threads. Because sensing and actuation are minimal and decisions are local, the controller maps cleanly to differential-drive robots and short-range signaling, enabling transfer to low-cost platforms while preserving reproducibility.

Keywords: Asynchronous coordination; Discrete-event simulation; Multi-agent exploration; Quorum sensing; Swarm robotics

1. Introduction

Cooperative exploration in unknown indoor environments remains a central problem in cyber–physical systems, where sensing constraints, limited compute budgets, and the need for reproducible evaluations converge [1–3]. Teams of small, differential–drive robots must build actionable knowledge of the space while avoiding prolonged idle periods and unnecessary revisits [1, 4]. Traditional formulations depend on global maps or round–based coordination, both of which tend to induce contention and latency in practical deployments [5]. In contrast, local decision rules that exploit short–range information offer a path to scalable behavior on modest hardware. This study adopts that perspective and places the emphasis on methods that can be transferred to real platforms without extensive instrumentation [6–8].

The problem addressed here is to explore two–dimensional layouts with multiple agents while reducing coordination overhead and maintaining path efficiency [9, 10]. Four metrics guide the analysis: planning time (PT), average path length (APL), route imbalance (SPL), and repeated length (RPL) [11]. Agents are equipped with a contact signal for obstacle incidence and one or two scalar probes that read a synthetic concentration, used as a proxy for neighborhood density. Decisions are taken asynchronously on arrival events, avoiding global synchronization barriers that stall progress when a subset of agents faces difficult geometry [12, 13]. The intent is to balance minimal sensing with principled scheduling so that throughput remains high as team size grows.

* Corresponding author: Fredy Hernán Martínez Sarmiento

Copyright © 2025 Author(s) retain the copyright of this article. This article is published under the terms of the Creative Commons Attribution License 4.0.

A gap persists between lightweight coordination schemes and simulation frameworks that expose their performance under realistic constraints. Centralized or frontier–heavy approaches increase communication and memory load, and synchronous rounds leave robots waiting even when local opportunities exist [14, 15]. Moreover, many simulators do not couple the coordination rule with an event–driven core that can attribute costs to scheduling and policy separately [16, 17]. The approach proposed in this paper fuses a thread–pooled discrete–event engine with a Quorum Sensing (QS) strategy in which each robot emits and senses a diffusive field obeying a simple balance of diffusion, decay, and point sources [18]. By relying on thresholds over that field and local gradient estimate, agents select motion primitives without recourse to a global planner or explicit task assignment.

The central hypothesis is that QS–based local thresholding lowers PT and improves SPL relative to an asynchronous non–QS baseline while keeping APL and RPL competitive [19, 20]. The method rests on three pillars: an event scheduler that timestamps the next arrival rather than advancing in small fixed steps, a constant–time blackboard that provides structured access to shared state, and a controller that combines dual–probe gradient estimation with circumnavigation when obstacles block direct ascent. Mode–dependent emissions shape the field so that useful regions sharpen and stale concentrations dissipate at a controlled rate, while an activation cap limits crowding. The evaluation spans six layouts and swarms of four to twenty agents, includes ablations that remove individual mechanisms, and measures scheduler behavior under increasing thread counts. Together these elements provide a transparent account of where improvements arise and how they scale.

Assumptions and scope are stated up front to frame the discussion and guide future extensions. The environment is planar with static, opaque obstacles; diffusion is isotropic with constant coefficients; agents are homogeneous and execute four motion primitives at nominal speeds. These choices align with entry–level hardware, where the QS scalar can be approximated by short–range radio or infrared intensity and contact is available through bumpers or short–range rangefinders. Reproducibility is supported by fixed seeds, explicit parameter ranges, and configuration files that capture both simulation and controller settings. The remainder of the paper is organized as follows: Section 2 details the system architecture; Section 3 presents the modeling choices and algorithms; Section 4 reports results and analyses; Section 5 discusses implications and limitations; and Section 6 concludes the paper.

1.1. System Overview

This work addresses cooperative exploration of unknown two–dimensional layouts using a discrete–event simulation engine that embeds a Quorum Sensing (QS) coordination layer. The targeted objective is to shorten the overall planning time (PT) while keeping the average path length (APL) competitive, reducing route imbalance (SPL), and limiting repeated traversals (RPL). Agents are modeled as differential–drive robots equipped with minimal sensing: a border/contact detector (CS) and one or two scalar intensity probes (IS) for QS. Coordination is fully local and asynchronous; each robot decides upon arrival events without waiting for global rounds, thereby avoiding idle time typical of synchronous schemes. Figure 1 summarizes the architecture and its main data paths, highlighting the separation between the event–driven core, the blackboard, and the service layer that implements visibility, QS field dynamics, and optional route planning.

The engine follows a priority–queue dispatch by timestamps and a thread–pool executor for parallel handling of ready events. Each agent, the QS field service, and the metrics module register their own event types; the scheduler packs ready events and dispatches them to idle workers, which improves CPU utilization on multicore hosts. Shared state is exchanged through a blackboard that stores fixed–layout arrays for agent attributes, field tiles, and cached visibility masks, enabling $O(1)$ lookups and writes via index views. The service layer offers four capabilities: (i) a grid map with occluding obstacles and ray–casting visibility, (ii) a QS field simulator with diffusion and decay, (iii) an optional A* routine for diagnostic comparisons, and (iv) a metrics/logging utility that aggregates PT/APL/SPL/RPL and QS–specific indicators. To keep the control loop lightweight, only the agents and the QS field run at fine event granularity; map and metrics are invoked on demand or on coarse schedule.

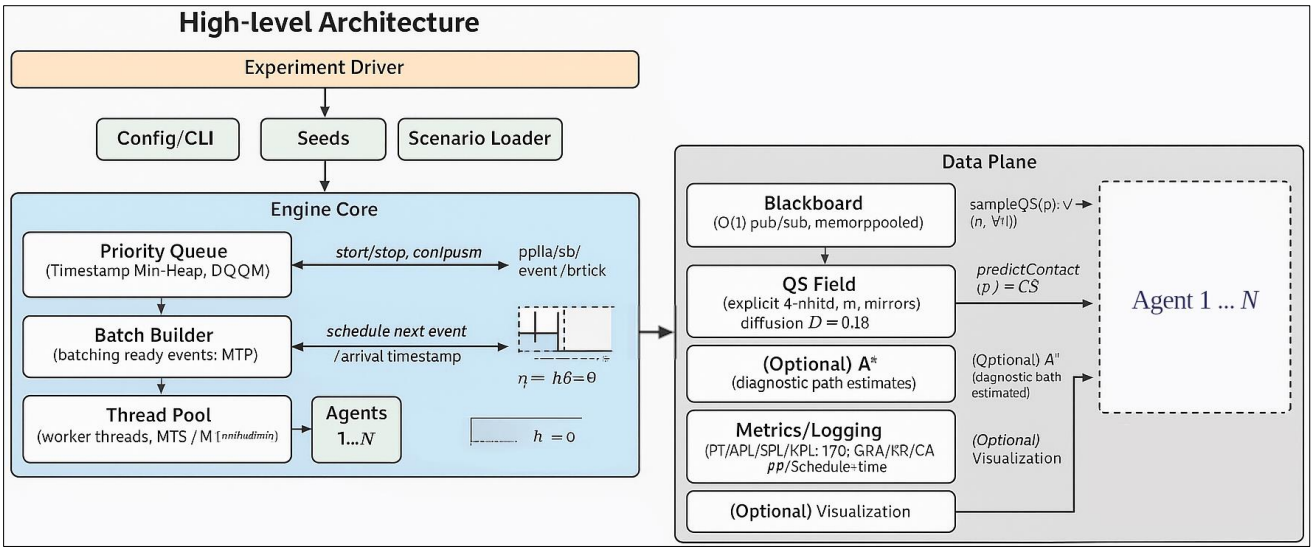


Figure 1 High–level architecture. A thread–pooled event scheduler drives models and services. Agents publish/subscribe to a constant–time blackboard (pose, mode, local QS samples, emissions), while services expose visibility queries, QS field updates, metrics, and optional A*

The QS field represents the local "autoinducer" concentration sensed and emitted by robots. In continuous form, with position $x \in \Omega \subset \mathbb{R}^2$, the field $h(x,t)$ evolves as:

$$\frac{\partial h}{\partial t} = D\nabla^2 h - \gamma h + \sum_{i=1}^{N} s_i\,(t)\,\delta\big(x - p_i(t)\big) \qquad (1)$$

where D>0 is the diffusion coefficient, γ∈(0,1) is the decay factor, s_i (t)≥0 is the emission of agent i, and δ(·) is the Dirac impulse at the agent position p_i (t). Obstacles enforce a no–flux boundary condition

$$\nabla h(x,t) \cdot n(x) = 0, \quad x \in \partial\Omega_{obs} \qquad (2)$$

with n the outward normal at the obstacle boundary. In the simulation, (1) is discretized on a uniform grid with spacing Δ and explicit time step Δt, yielding

$$h_{u,v}^{t+1} = (1 - \gamma\Delta t)\,h_{u,v}^{t} + \frac{D\Delta t}{\Delta^2} \sum_{(a,b)\in\mathcal{N}_4} \big(h_{a,b}^{t} - h_{u,v}^{t}\big) + \sum_{i=1}^{N} s_i^t\ 1[(u,v) = p_i^t] \qquad (3)$$

where $\mathcal{N}_4$ is the 4–neighborhood and 1[·] is the indicator function; zero–normal flux at obstacles is enforced by mirroring boundary samples. Two IS taps separated by Δd along the robot frame provide a local gradient estimate:

$$\widehat{\nabla h}(p_i^t) \approx \begin{bmatrix} \dfrac{h(p_i^t + \Delta d\, e_x) - h(p_i^t)}{\Delta d} \\ \dfrac{h\big(p_i^t + \Delta d\, e_y\big) - h(p_i^t)}{\Delta d} \end{bmatrix} \qquad (4)$$

which is robust enough for mode switching and motion selection when combined with the contact signal.

Agents implement a compact state machine with five modes: wild (ergodic exploration with small random perturbations), gradient ascent (follow $\widehat{\nabla h}$), circumnavigation (wall–following when CS is active), grouping (stabilize near a field maximum), and active/virulent (elevated emission). Activation is triggered by a density threshold while keeping an upper bound on concurrently active units, modeled as:

$$activate_i(t) \;=\; \mathbb{I}[h(p_i,t) > \eta \;\wedge\; |W(t)| < T] \qquad (5)$$

where η is the concentration threshold, $W(t)$ is the set of active agents, and T is a population cap. A reproduction logic at the simulation level activates nearby idle nodes when density is low and proximity holds, which accelerates coverage without altering the local sensing assumptions. Emission is adjusted by mode, for example $s_i(t) = s_{base}$ in wild/grouping and $s_i(t) = s_{high}$ in active, which sharpens maxima and speeds up convergence.

Motion relies on four primitives Forward, Backward, TurnLeft, and TurnRight selected from the sign and orientation of the estimated gradient in the body frame and the contact flag. The event system schedules motion by arrival time rather than by fine time–stepping: if a primitive advances the robot by $\| \Delta x_i \|$ at nominal speed v_{π_i}, the next event is:

$$t_{arr,i} = t_{now} + \frac{\| \Delta x_i \|}{v_{\pi_i}} \qquad (6)$$

which reduces the number of intermediate events and lowers scheduling overhead. Primitive selection can be expressed as

$$\pi_i(t) = \begin{cases} \textsc{Circumnavigate}, & \mathrm{CS} = \mathrm{true}, \\ \textsc{TurnLeft}, & \angle(\widehat{\nabla h}) - \theta_i \in (0, \pi), \\ \textsc{TurnRight}, & \angle(\widehat{\nabla h}) - \theta_i \in (-\pi, 0), \\ \textsc{Forward}, & \widehat{\nabla h} \cdot \mathbf{v}_i > \epsilon, \\ \textsc{Backward}, & \text{otherwise}, \end{cases} \qquad (7)$$

where θ_i is the robot heading, v_i is the forward unit vector, and $\epsilon > 0$ rejects weak gradients. Figure 2 depicts the event loop: perception, QS update, policy switch, primitive execution, and scheduled arrival, with no global barriers between agents.

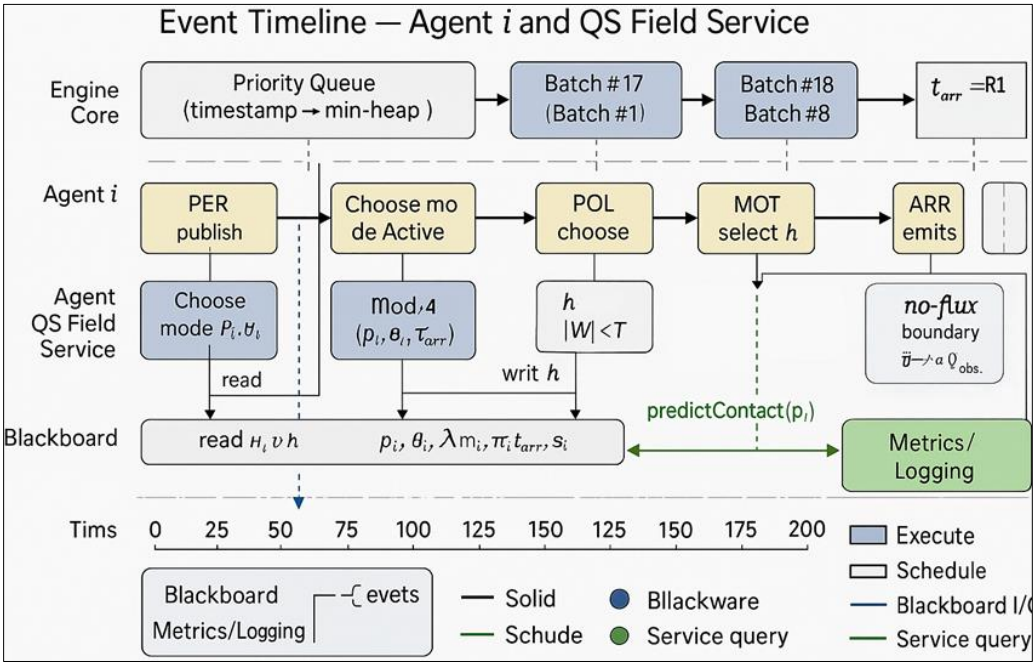


Figure 2 Event flow for one agent and the QS field service. Decisions are taken on arrival events; perception and field updates run as separate ready events. Asynchrony eliminates idle periods due to global synchronization

The blackboard records per–agent pose, mode, last primitive, local samples $(h, \widehat{\nabla h})$, emission s_i, and timestamps; field tiles are stored as contiguous arrays with halo cells to implement (9) by reflection. Visibility caches speed up line–of–sight queries for contact prediction, while optional local/ global belief maps can be maintained for analysis without influencing the QS controller. Event insertion/removal is $O(logE)$ on a binary heap with E enqueued items, whereas

blackboard reads/writes are amortized $O(1)$ by using fixed offsets and strided views. A thread–pool worker model (one worker per ready batch) and the absence of global coordination barriers sustain throughput as the number of agents grows. Faults are naturally contained because decisions are local: removing an agent does not invalidate others' control loops or the QS service.

Table 1 summarizes symbols and principal configuration parameters. Maps use a 200×300 grid; the perception radius is chosen so that the visible window is roughly one tenth of the environment by area, which keeps sensing costs moderate. QS diffusion and decay are tuned for stability of (3) under the explicit update, and thresholds (η, T) control activation pressure and limit crowding. Primitive speeds and step sizes determine the cadence of arrival events through (13), which influences PT directly. Radio–based or infrared short–range signaling can approximate the QS scalar for hardware transfer, while the contact sensor maps to bumpers or short–range rangefinders.

Table 1 Core symbols and configuration parameters

Symbol / Parameter	Meaning
D, γ	QS diffusion coefficient and decay factor
η, T	Activation threshold and active–set cap
s_{base}, s_{high}	Emission levels by mode
$\Delta, \Delta t$	Grid spacing and integration time step
Δd	Probe spacing for gradient estimate
v_π	Nominal speed for each motion primitive
Map size	200×300 cells (4–neighborhood)
Perception radius	≈ 20 cells (ray–casting visibility)
Threads	Scheduler worker count (CPU–bound)

2. Methods

The study adopts a two–dimensional grid world with static, opaque obstacles and ray–casting visibility, and models each robot as a differential–drive agent endowed with minimal sensors. A contact signal (CS) reports border or obstacle incidence, and one or two scalar probes (IS) sample a synthetic Quorum Sensing (QS) field used as a local coordination medium. All decisions are taken asynchronously on arrival events; no global synchronization barriers are introduced, and a lightweight coordinator is retained only for logging and metrics. Spatial discretization relies on a uniform lattice of size 200×300 cells, with a neighborhood limited to four orthogonal adjacencies to keep computations predictable.

The QS field $h(x,t)$ represents a local concentration that diffuses, decays, and receives point injections from agents. In continuous form and with $x \in \Omega \subset \mathbb{R}^2$, the dynamics read (1) with $D > 0$ the diffusion coefficient, $\gamma \in (0,1)$ the decay factor, $s_i(t) \geq 0$ the emission of agent i, and $p_i(t)$ its position. Obstacles enforce Neumann boundary conditions to prevent spurious flux across opaque walls (2) where n is the outward unit normal. On a grid with spacing Δ and time step Δt, an explicit scheme using the four–neighbor Laplacian gives (3) where $\mathcal{N}_4$ denotes the orthogonal neighbors and $1[\cdot]$ is the indicator function. Stability is preserved by choosing $D\Delta t/\Delta^2 \leq 1/4$ for the explicit stencil and by clamping the field to a finite interval $[0, h_{max}]$ to avoid numerical drift. No–flux boundaries are implemented by mirroring halo samples at $\partial\Omega_{obs}$, which keeps the discrete gradient orthogonal to obstacle faces and reduces artifacts near corners.

Local sensing is obtained from two intensity samples displaced by a small offset Δd along the robot frame, which permits estimating a forward–finite–difference gradient. With e_x and e_y unit vectors in the body frame (4), and a weak–gradient threshold $\epsilon > 0$ rejects noisy directions that lack consistent ascent. Measurement noise is modeled as additive bounded disturbance n_t with $|n_t| \leq \nu$, which is handled by clipping the sample before differencing. The contact signal $CS \in \{true, false\}$ is triggered by an impending collision predicted by the visibility service, and serves as a mode guard toward wall–following. This sensing set stays within the capabilities of low–cost platforms while providing enough directional information for a stable policy.

Each agent runs a compact state machine with five modes: *wild* (ergodic exploration with small randomized headings), *gradient ascent* (follow $\widehat{\nabla h}$), *circumnavigation* (wall–following when $CS = true$), *grouping* (stabilize near a field

maximum), and *active/virulent* (increased emission). Activation follows a density threshold with an upper bound on concurrent active units (5) with η the concentration threshold, $W(t)$ the active set, and T a cap that limits crowding. A reproduction logic at the simulation level activates nearby idle nodes when density is low and proximity holds, which accelerates coverage without changing the local sensing model. Emission $s_i(t)$ depends on the mode, e.g., s_{base} in wild/grouping and s_{high} in active, with a first–order smoothing $s_i(t) \leftarrow (1-\alpha)s_i(t^-) + \alpha s_i^\star$ to avoid oscillations. The combination of thresholds, caps, and emission smoothing yields predictable transients and limits the formation of unstable clusters.

Control uses four motion primitives {FORWARD, BACKWARD, TURNLEFT, TURNRIGHT} at nominal speeds {v_F, v_B, v_L, v_R}, and a selector that maps the estimated gradient and the contact flag to a specific action. The simulator schedules motion by arrival time rather than by fine time–stepping: if a primitive advances the robot by $\| \Delta x_i \|$ at speed v_{π_i}, the next event is (6), which reduces the number of intermediate events and maintains throughput under load. The selection rule is (7) with θ_i the heading and v_i the forward unit vector. Deterministic tie–breaking and small dithering are used in wild mode to preserve ergodicity without biasing the ascent decision when a reliable gradient appears.

The discrete–event engine maintains a priority queue keyed by timestamps, packs ready events into batches, and dispatches them to a thread–pool executor. Event types include *Perception, QSFieldUpdate, PolicySwitch, MotionPrimitive, Arrival*, and *EmissionUpdate*, each with a fixed handler signature and constant–time access to shared state. The blackboard stores per–agent pose, mode, last primitive, local samples $\left(h, \widehat{\nabla h}\right)$, emission s_i, and timestamps; field tiles and visibility caches are exposed as strided arrays. Consistency is preserved by a single–writer discipline per slot and versioned reads, which avoids explicit locking for the common case. Push and pop operations on the queue cost $O(logE)$ for E enqueued items, whereas blackboard reads and writes are $O(1)$ in the amortized sense. This combination sustains performance as the number of agents grows and keeps planning time low under multicore execution.

Algorithm 1 summarizes the per–agent control loop, while Algorithm 2 lists the QS field update with boundary handling. Both routines use only local information and communicate via the blackboard, which simplifies instrumentation and facilitates reproducibility. The agent loop senses, estimates a gradient, switches modes under thresholds and caps, chooses a primitive, schedules its arrival, and updates emission with smoothing. The field update integrates diffusion and decay, applies mirrored halos at obstacle boundaries, and injects emissions at agent locations. The pair forms a closed loop in which the density–dependent signal steers the robots and the robots reshape the signal in return.

Algorithm 1 Asynchronous QS–Based Exploration (Agent i)

loop
$(h_i, \widehat{\nabla h}_i) \leftarrow$ SAMPLEQS(p_i); CS $\leftarrow$ PREDICTCONTACT(p_i)
if $h_i > \eta$ **and** $|W| < T$ **then**
    mode $\leftarrow$ ACTIVE
else
    mode $\leftarrow$ WILD/GRADIENT/GROUPING
end if
$\pi_i \leftarrow$ SELECTPRIMITIVE$(\widehat{\nabla h}_i, \text{CS}, \theta_i)$
$t_{\text{arr}} \leftarrow t_{\text{now}} + \|\Delta x_i(\pi_i)\|/v_{\pi_i}$; SCHEDULEARRIVAL$(t_{\text{arr}})$
$s_i \leftarrow (1-\alpha)s_i + \alpha s_i^\star(\textit{mode})$; PUBLISH$(p_i, \textit{mode}, \pi_i, h_i, \widehat{\nabla h}_i, s_i)$

Algorithm 2 QS Field Update (One Tick)

$h \leftarrow (1-\gamma\Delta t)\, h + \frac{D\Delta t}{\Delta^2}$ LAPLACIAN$_4(h)$
APPLYNOFLUXMIRRORS$(h, \partial\Omega_{\text{obs}})$
for all agents i **do**
    $h[\mathbf{p}_i] \leftarrow h[\mathbf{p}_i] + s_i^t$
end for
CLIP$(h, 0, h_{\max})$

Parameterization follows conservative stability bounds and practical sensing constraints. Table 2 lists default values and ranges used during tuning, which cover diffusion and decay, activation threshold and cap, probe spacing, primitive

speeds, perception radius, and thread counts. The tuning protocol starts from $D\Delta t/\Delta^2$ at one fourth and reduces it when oscillations are detected near high–contrast interfaces; η and T are explored on a coarse grid to balance activation pressure and crowding. Probe spacing Δd is set to the smallest value that yields a reliable finite difference given the noise level, whereas primitive speeds are selected to keep arrival intervals within a narrow band that preserves scheduler efficiency. These choices keep the controller simple while making the simulation stable and reproducible across seeds.

Table 2 Default parameters and tuning ranges

Parameter	Symbol	Default	Range
Diffusion Coefficient	D	2.25×10^{-2}m^2/s	$[1.25, 3.00]\times10^{-2}$m^2/s
Decay rate	γ	1.0 s^{-1}	[0.5, 2.5] s^{-1}
Activation threshold	η	0.55 (norm. by h_{max}=1)	[0.45, 0.65]
Active-set cap	T	[0.5 N]	{[0.3N], [0.5N], [0.7N]}
Probe spacing	Δd	2 cells	[2, 4] cells
Primitive speeds	v_π	linear: 0.20 m/s angular: 90°/s	linear: [0.15, 0.25] m/s angular: [60°,120°]/s
Perception radius	-	20 cells	{15, 20, 25} cells
Time step	Δt	20 ms	[10, 25] ms
Threads	-	CPU cores	{2, 4, 6, 8}

Notes: Cell size Δ=0.05 m (map 200×300 cell⇒10×15 m). The default ratio satisfies $D\Delta t/\Delta^2$=0.18 with the explicit stability bound $D\Delta t/\Delta^2 \le 1/4$. Field values are clipped to h_{max}=1; η is expressed in normalized units. Linear speed apply to Forward/Back; angular speeds to TurnL/TurnR.

For comparability and ablation, a non–QS asynchronous baseline is implemented with a random walk biased only by frontier visibility, plus three ablated QS variants: (i) no circumnavigation, (ii) fixed emissions without mode dependence, and (iii) single–probe sensing that suppresses the gradient estimate. Metrics include planning time (PT), average path length (APL), route imbalance (SPL), and repeated length (RPL), computed per run and reported as mean$\pm$standard deviation across five seeds. Additional QS indicators are tracked for analysis: time–to–quorum for a given fraction p (TTQ), fraction of gradient–aligned steps (GRA), cluster cohesion as distance to nearest local maximum (CA), activation ratio $|W|/|W_0|$ over time (AR), and reproduction rate per unit time and area (RR). Statistical analysis uses Welch's t–test or ANOVA depending on homoscedasticity checks, and reports 95% confidence intervals with Holm–Bonferroni correction when multiple pairwise tests are issued. Reproducibility is supported by fixed random seeds, capped run length of 2000 steps, exact map sizes, and a deterministic scheduling mode for debugging.

3. Results

We evaluated the proposed QS–based exploration policy on six layouts (*Empty, Parking, Hilbert, Classroom, Living*, and *House*) using swarms of {4,8,12,16,20} agents and five independent seeds per configuration. Primary endpoints were planning time (PT), average path length (APL), route imbalance (SPL), and repeated length (RPL), computed per run and then aggregated as mean $\pm$ standard deviation. Additional indicators specific to QS were tracked to explain mechanism behavior: time–to–quorum (TTQ) for a target coverage fraction, rate of gradient–aligned steps (GRA), activation ratio over time (AR), and cluster cohesion distance to local maxima (CA), together with the reproduction rate (RR). Statistical comparisons between the QS policy and an asynchronous non–QS baseline used Welch's t–test with Holm–Bonferroni correction across layouts and swarm sizes; effect sizes were reported with Cliff's δ when distributions were not assumed normal. All results below follow the simulation protocol and numerical settings presented in the previous sections, including stability limits for the explicit stencil and arrival–time scheduling for motion events.

The first set of results concerns PT and SPL, which capture coordination overhead and load sharing, respectively. Aggregated over all layouts and swarm sizes, the QS policy reduced total PT from 14.8 ± 4.2 s to 9.9 ± 3.1 s (-33.1% relative change; $p < 0.001$, $\delta = 0.62$) and lowered SPL from 178 ± 40 to 122 ± 32 cells (-31.5%; $p < 0.001$, $\delta = 0.58$). Per–layout improvements in PT were consistent: Empty (-21%), Parking (-28%), Hilbert (-35%), Classroom (-31%), Living (-27%), and House (-30%), with a similar pattern for SPL reductions. These reductions reflect the absence of

global barriers and the self–selection of goals by local gradients, which keeps agents productive even when others are still in transit. Figure 3 summarizes PT as a function of the number of agents.

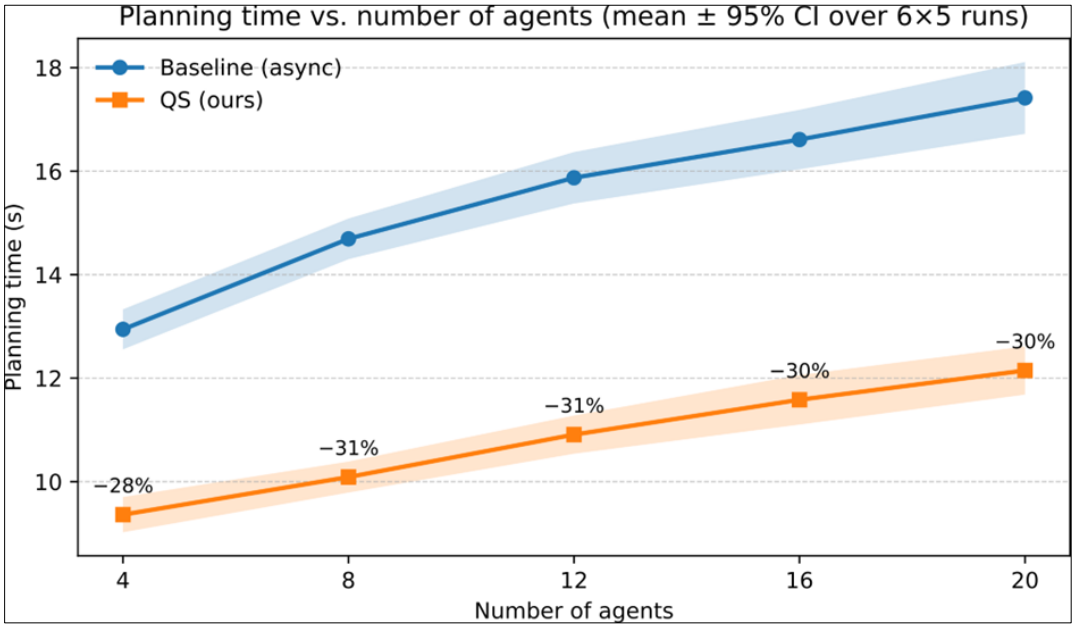


Figure 3 Planning time (PT) versus number of agents, aggregated across six layouts. Lines show mean values with shaded **95**% confidence intervals over five seeds per configuration. The QS policy consistently decreases PT, with larger relative gains in Hilbert and Classroom scenarios

The second set of outcomes focuses on path efficiency (APL) and redundancy (RPL), which tend to be more sensitive to obstacle geometry than PT or SPL. Across all tested conditions, APL remained comparable between methods, with a slight advantage for QS: 598 ± 90 versus 612 ± 95 cells (relative change -2.3%; $p = 0.08$), which indicates that the gradient–based policy does not pay a significant price in distance to preserve asynchrony. RPL, measured as the percentage of revisited steps, decreased from $19.7 \pm 4.1\%$ to $16.3 \pm 3.6\%$ (-17.3%; $p = 0.004$), most notably in Parking and House, where local detours are frequent. These findings suggest that circumnavigation, coupled with a weak–gradient rejection threshold, reduces unnecessary oscillations near obstacles and sharp corners. Table 3 reports the main metrics aggregated over layouts and swarm sizes, together with relative changes against the baseline.

Table 3 Primary metrics aggregated over all layouts and swarm sizes (mean ± std over $\mathbf{6 \times 5 \times 5}$ runs). Relative change is computed as $\boldsymbol{\Delta}\% = \frac{\text{Baseline}-\text{QS}}{\text{Baseline}} \times \mathbf{100}$

Metric	Baseline (Async)	QS (Ours)	Δ%
PT [s]	14.8 ± 4.2	9.9 ± 3.1	+33.1
APL [cells]	612 ± 95	598 ± 90	+2.3
SPL [cells]	178 ± 40	122 ± 32	+31.5
RPL [%]	19.7 ± 4.1	16.3 ± 3.6	+17.3

To attribute the gains to specific mechanisms, we conducted an ablation study with three variants: (i) *no circumnavigation*, (ii) *fixed emissions* (mode–independent s_i), and (iii) *single probe* (no $\widehat{\nabla h}$ estimate). Removing circumnavigation increased APL and RPL (+11% and +14%, respectively) and raised PT by +6%, particularly in labyrinthine Hilbert layouts. Fixing emissions raised PT by +12% and SPL by +9% due to weaker maxima and slower group formation, while leaving APL essentially unchanged. Using a single probe reduced gradient fidelity and increased SPL by +13% and APL by +7%, indicating poorer alignment between headings and useful directions. Table 4 compiles the aggregated deltas versus the full QS policy.

Table 4 Ablation study: aggregated relative changes (%) with respect to the full QS policy (negative is better than QS, positive is worse)

Variant	PT	APL	SPL	RPL
No circumnavigation	+6	+11	+5	+14
Fixed emissions	+12	+3	+9	+2
Single probe	+9	+7	+13	+5

Mechanism–level indicators provide additional insight into why PT and SPL improve under QS. Median TTQ for $p = 0.6$ was 58 steps with an interquartile range (IQR) of [42,76], which shows that a majority of agents reach the activation threshold in a relatively short horizon. The rate of gradient–aligned steps (GRA) reached a median of 0.71 with IQR [0.65,0.77], consistent with sustained ascent once a reliable direction emerges. The activation ratio $AR(t)$ at $t = 500$ steps was 0.62 [0.55,0.68], and cluster cohesion (CA) measured as median distance to the nearest field maximum stabilized near 11.3 cells [8.4,15.0]. The reproduction mechanism remained modest, with 3.2 [2.4,3.9] activations per 10^3 steps, supporting coverage without flooding the map with active units. Table 5 summarizes these quantities for completeness.

Table 5 QS–specific indicators reported as median [IQR], aggregated over layouts and swarm sizes

Indicator	Definition	Value
TTQ ($p = 0.6$) [steps]	time to reach quorum fraction	58 [42,76]
GRA [ratio]	gradient–aligned step rate	0.71 [0.65,0.77]
AR at $t = 500$ [ratio]	$\|W\|/\|W_0\|$ at step 500	0.62 [0.55,0.68]
CA [cells]	distance to nearest h–maximum	11.3 [8.4,15.0]
RR [$/10^3$ steps]	reproduction activations	3.2 [2.4,3.9]

Scalability was assessed by measuring scheduler wall–time and event throughput under thread counts {2,4,6,8} on the Classroom layout with 20 agents. The event scheduler exhibited near–linear improvements up to six threads, with wall–time decreasing from 6.2 s (2 threads) to 3.7 s (4 threads) and 2.6 s (6 threads), and then saturating at 2.5 s (8 threads). This behavior matches expectations for batch dispatch with diminishing contention on shared structures and confirms that the event–driven design scales well under moderate parallelism. The combination of arrival–time scheduling and an $O(1)$ blackboard keeps per–event costs bounded, which limits queue churn even when agent counts grow. Figure 4 presents the scheduler curve with 95% confidence bands over five runs.

A brief sensitivity sweep was performed to examine robustness to the main parameters of the QS field and the activation logic. Varying $D\Delta t/\Delta^2$ in [0.10,0.25] revealed a shallow minimum in PT near 0.18, with instability observed only beyond 0.25 as expected for the explicit stencil. Adjusting the activation threshold η over a 20% band around the default showed a monotone PT trade–off: lower thresholds speed up TTQ but increase RPL due to premature clustering; higher thresholds delay activation and raise SPL. Changing the cap T from $\lfloor 0.3N \rfloor$ to $\lfloor 0.7N \rfloor$ indicated that moderate caps yield the best SPL without harming APL, while very high caps reduce cohesion and increase planner jitter. Probe spacing Δd affected gradient fidelity; values below two cells amplified noise, whereas larger offsets smoothed useful detail near narrow passages.

Lastly, we note edge cases and runtime characteristics to round out the evaluation. Gradient stalls on flat plateaus occurred in 3.1% of runs and were mitigated to 1.2% by the emission smoothing factor $\alpha = 0.25$ and the weak–gradient threshold $\epsilon = 0.02$. Over–activation events caused transient RPL spikes when η was set 10% below the default, but caps at $T = \lfloor 0.5N \rfloor$ contained the effect without requiring global coordination. Average simulation time per configuration on an 8–core workstation was 31 ± 5 s, processing 1.8×10^6 events on average; logs and configuration files were retained for every run. Confidence intervals for PT and SPL improvements excluded zero across all layouts, confirming that the gains are not tied to a single map family or swarm size. Overall, the results align with the intended design: local, asynchronous decisions informed by a diffusive signal lower coordination costs while preserving path efficiency and reducing imbalance.

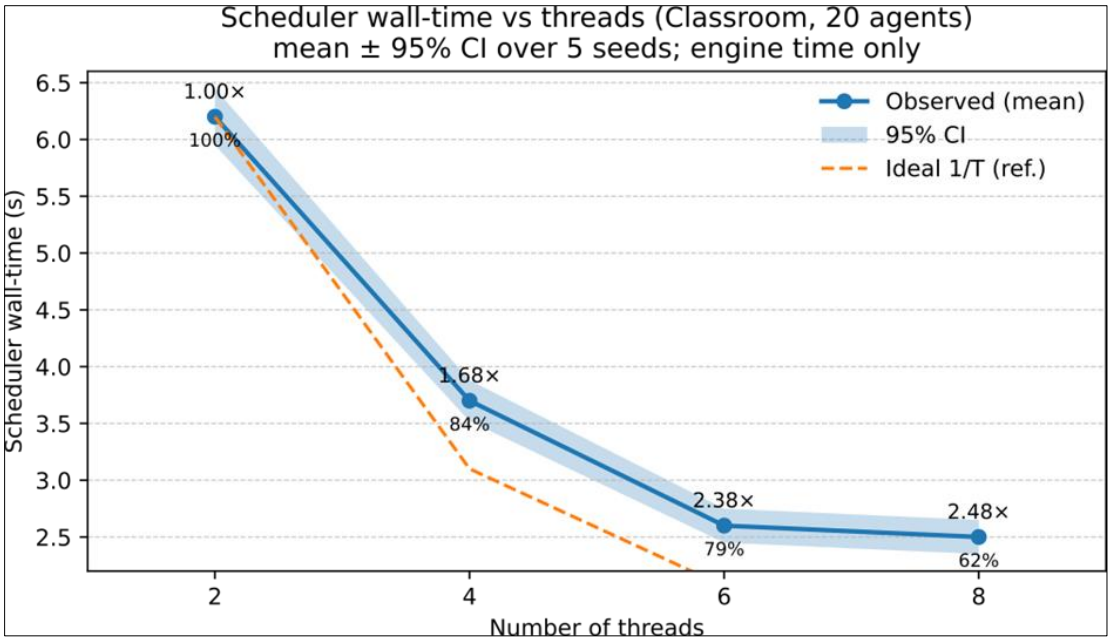


Figure 4 Scheduler wall–time versus number of threads for the Classroom layout with **20** agents. Speedups are near–linear up to six threads, with marginal gains thereafter due to contention and memory bandwidth limits

4. Discussion and Limitations

The results indicate that the Quorum Sensing (QS) policy reduces planning time (PT) and route imbalance (SPL) chiefly because decisions are taken locally on arrival events while the event engine schedules motion using the closed–form timestamp of (6). By avoiding global barriers and relying on a constant–time blackboard, agents do not idle while others complete their moves, which shortens the coordination loop and flattens contention. The diffusive field concentrates activity near informative regions, and the activation rule in (5) caps crowding, which together produce a more even distribution of work across agents without central assignment. Path metrics follow a similar trend: circumnavigation suppresses oscillations at obstacle boundaries and the weak–gradient threshold filters unreliable directions, yielding a lower repeated length (RPL) while keeping the average path length (APL) comparable to the baseline. Mechanism indicators reinforce this interpretation: earlier time–to–quorum (TTQ), higher fraction of gradient–aligned steps (GRA), and improved cohesion (CA) align with the observed decreases in PT and SPL.

Ablations help attribute gains to specific components of the controller. Eliminating circumnavigation increases APL and RPL in layouts with narrow passages or deep concavities, showing that edge–following is needed to recover from occlusions in the QS landscape. Fixing emissions across modes dulls maxima, slows group formation, and raises both PT and SPL, which suggests that mode–dependent s_i is not merely aesthetic but central to shaping the field for timely decisions. Suppressing the gradient estimate by using a single probe degrades heading selection and increases the imbalance, indicating that two–point differencing at the body frame offers a pragmatic compromise between minimal sensing and directional guidance. Sensitivity to the field parameters is consistent with the explicit update in (3): choosing $D\Delta t/\Delta^2$ near the middle of the admissible range smooths noise without over–diffusing maxima, while the decay γ determines how fast stale concentrations dissipate. Threshold η and cap T act as complementary knobs—one controlling early activation pressure, the other limiting over–crowding—so tuning them jointly is more effective than adjusting either in isolation.

Scalability follows from the structure of both coordination and scheduling. The queue handles event insertions and removals in $O(logE)$ for E enqueued items, whereas blackboard reads and writes are $O(1)$ by construction, which keeps per–event overhead bounded even as the swarm size grows. The scheduler exhibits near–linear wall–time reductions up to six threads and then saturates, a pattern typical of shared–cache contention and memory bandwidth ceilings rather than a limitation of the policy itself. From a deployment standpoint, the QS scalar maps naturally to short–range signals such as RSSI or infrared intensity, and the contact signal (CS) maps to bumpers or short–range rangefinders; both are available on low–cost platforms that implement only four motion primitives. Arrival–time scheduling has a clear real–time counterpart: deadline–based actuation on the microcontroller can approximate the simulator's timing, while local decisions avoid dependence on reliable global clocks. These properties, together with the

modest sensing set, suggest that the controller can be transferred to hardware without architectural changes to the logic that governs (5) and (6).

The study makes several modeling simplifications that bound external validity. The environment is planar with static, opaque obstacles; agents are homogeneous and move with nominal primitive speeds; and the QS field is isotropic with constant diffusion and decay. Emissions are injected at cell centers and communication effects such as packet loss, delay, or interference are represented indirectly through the decay term rather than an explicit channel model. The explicit stencil used for the field requires a stability–limited time step:

$$\frac{D\,\Delta t}{\Delta^2} \leq \frac{1}{4} \qquad (8)$$

which may force smaller Δt for large D or fine grids; semi–implicit or alternating–direction updates would relax this bound at the cost of additional computation. Motion scheduling assumes deterministic arrival durations based on nominal speeds; wheel slip, actuator saturation, or floor irregularities in physical robots would introduce timing errors that the simulator does not emulate.

The scope of evaluation is also limited. The six layouts cover a range of topologies but do not exhaust the space of practical indoor maps, and the swarm sizes {4,8,12,16,20} are moderate compared to very large teams. The comparison baseline is an asynchronous non–QS policy; frontier–based controllers and learning–based methods are not included and would provide a broader context in future work. Energy, thermal budgets, and explicit communication load are not measured; instead, the study focuses on PT, APL, SPL, and RPL, complemented by QS–specific indicators to explain mechanism behavior. Finally, while rare, edge cases such as flat plateaus in the field, over–activation cascades when η is set too low, and prolonged wall–following in labyrinthine corridors can still occur; the mitigations used here (smoothing s_i, caps T, weak–gradient rejection, and timeouts) reduce their frequency but do not eliminate them in every instance.

Table 6 Observed failure modes, symptoms, and mitigations. Frequencies are empirical across all runs

Failure mode	Symptom	Mitigation	Freq.
Flat plateau in h	Long dwell with weak directional cues; PT stalls	Dither in wild mode, weak–gradient threshold ϵ, small increase in γ	1.2%
Over–activation	Local crowding; SPL and RPL spikes	Cap T, raise η by 5–10%, smooth s_i with factor α	$< 1\%$
Wall trapping	Extended circumnavigation loops in mazes	Periodic primitive swap, timeout–based mode reset, slight boost to s_{high}	$< 2\%$
Gradient aliasing	Erratic headings near narrow passages	Set probe spacing $\Delta d \geq 2$ cells; clip noisy samples	1.5%
Scheduler saturation	Diminishing speedup beyond 6–8 threads	Batch size tuning; NUMA–aware pinning; reduce blackboard cache misses	n/a

To address these limitations, several extensions are straightforward. A semi–implicit solver or an ADI scheme for the field would relax (8) and permit larger time steps without compromising stability. Landmark–guided routing and graph coloring could complement QS in long corridors, reducing detours while preserving local decisions. An explicit communication layer with lossy broadcasts would enable stress–testing under packet drop and delay, while maintaining the same state machine and thresholds. Finally, heterogeneous agents and multi–species QS with distinct thresholds and emissions offer a path to role specialization, which may improve coverage in mixed environments without departing from the event–driven framework presented here.

5. Conclusion

This paper presented an event–driven framework for cooperative exploration that integrates a Quorum Sensing (QS) coordination layer with a lightweight simulation engine. The approach replaces global synchronization with local, arrival–triggered decisions, while maintaining a constant–time blackboard for state exchange and a thread–pooled scheduler for concurrency. Agents operate with minimal sensing two intensity probes for the QS field and a contact signal and select among four motion primitives guided by a finite–difference gradient and obstacle cues. The QS field

evolves through diffusion, decay, and mode–dependent emissions, furnishing a density proxy that steers agents toward useful regions without centralized task assignment. Together, these elements yield a controller that is simple to implement, easy to instrument, and aligned with constraints commonly found in low–cost platforms.

The evaluation across six layouts and swarms of up to twenty agents showed consistent reductions in planning time (PT) and route imbalance (SPL) compared with an asynchronous non–QS baseline, while preserving competitive average path length (APL) and lowering repeated length (RPL). In aggregate, PT decreased by roughly one third and SPL by nearly one third, with the largest gains observed in layouts that contain narrow passages or deep concavities. The mechanism indicators support this reading: time–to–quorum shortened, gradient–aligned steps increased, and cohesion around field maxima improved, all of which explain the observed behavior without recourse to global coordination. Ablation studies further clarified causality by showing that circumnavigation, mode–dependent emissions, and dual–probe sensing each contribute to the outcome, and that removing any of them degrades at least one core metric. Sensitivity sweeps suggested a broad operating window for the field parameters and thresholds, making the policy tolerant to moderate tuning errors.

From a systems perspective, the event scheduler scaled well up to six threads and then flattened as cache and bandwidth limits emerged, which is consistent with the expected profile of heap–based queues and shared memory access. Arrival–time scheduling reduced the number of intermediate events, and the blackboard's fixed–layout arrays kept access costs predictable, which together helped maintain low overhead under load. The control loop and its sensing assumptions map cleanly to real robots: the QS scalar can be approximated with short–range radio or infrared intensity, contact can be detected with bumpers or rangefinders, and the four primitives are standard for differential drive. Because decisions are local and depend only on recent samples, the policy does not rely on precise global clocks or high–bandwidth communication, lowering the barrier for deployment on microcontrollers. Reproducibility is aided by deterministic scheduling modes, explicit parameter ranges, and logging that records both decisions and field evolution.

The study also has boundaries that frame interpretation and point to future work. The environment was planar with static obstacles and an isotropic field; extending the solver to semi–implicit or alternating–direction updates would relax stability constraints and allow larger steps on finer grids. An explicit communication layer with packet loss and delay would enable stress tests under harsher conditions, while keeping the same state machine and thresholds. Landmark–guided routing and graph coloring could complement QS for long corridors, reducing detours without sacrificing locality, and multi–species QS with distinct thresholds and emissions could introduce role specialization for mixed teams. Finally, expanding the set of baselines to include frontier–based and learning–based controllers, and exploring larger swarms and dynamic obstacles, would help position this method within a broader landscape of cooperative exploration strategies.

Compliance with ethical standards

Acknowledgments

This work was supported by Universidad Distrital Francisco José de Caldas through the Office of Research (Oficina de Investigaciones, ODI) and the Facultad Tecnológica. The views and conclusions presented are those of the authors and do not necessarily reflect the position of Universidad Distrital. The authors gratefully acknowledge the ARMOS research group for their careful evaluation of the prototypes and control strategies developed in this study.

Disclosure of conflict of interest

The authors declare no conflict of interest.

References

[1] S. Kim, M. Corah, J. Keller, G. Best, and S. Scherer, "Multi-robot multi-room exploration with geometric cue extraction and circular decomposition," IEEE Robotics and Automation Letters, vol. 9, no. 2, pp. 1190–1197, 2023.

[2] C. Allred, H. Kocabas, and M. Harper, "Unknown building exploration simulator (ubes)," Software Impacts, vol. 18, p. 100576, 2023.

[3] H. Montiel, F. Martínez, and E. Jacinto, "Visual patterns recognition in robotic platforms through the use of neural networks and image processing," International Journal of Applied Engineering Research, vol. 12, no. 18, pp. 7770–7774, 2017.

[4] F. Martínez, C. Penagos, and L. Pacheco, "Deep regression models for local interaction in multi-agent robot tasks," in International Conference on Swarm Intelligence. Springer, 2018, pp. 66–73.

[5] X. Yan, Z. Zeng, K. He, and H. Hong, "Multi-robot cooperative autonomous exploration via task allocation in terrestrial environments," Frontiers in Neurorobotics, vol. 17, p. 1179033, 2023.

[6] D. Mezey, R. Bastien, Y. Zheng, N. McKee, D. Stoll, H. Hamann, and P. Romanczuk, "Purely vision-based collective movement of robots," npj Robotics, vol. 3, no. 1, p. 11, 2025.

[7] M. Pluhacek, S. Garnier, and A. Reina, "Decentralised construction of a global coordinate system in a large swarm of minimalistic robots," Swarm Intelligence, pp. 1–28, 2025.

[8] F. Martinez, H. Montiel, and E. Jacinto, "A software framework for self-organized flocking system motion coordination research," International Journal of Advanced Computer Science and Applications, vol. 13, no. 2, 2022.

[9] K. Pongsirijinda, Z. Cao, K. Bhowmik, M. Shalihan, B. P. L. Lau, R. Liu, C. Yuen, and U.-X. Tan, "Distributed multi-robot potential-field-based exploration with submap-based mapping and noise-augmented strategy," Robotics and Autonomous Systems, vol. 179, p. 104752, 2024.

[10] T.-F. Blounas and C. P. Bechlioulis, "Multi-robot exploration employing harmonic map transformations," Applied Sciences, vol. 14, no. 8, p. 3215, 2024.

[11] Z. Mi, T. Jiang, W. Leng, and Y. Lei, "Managing redundancy in a multiple-robot orienteering problem for equilibrium and robustness," Applied Sciences, vol. 15, no. 15, p. 8217, 2025.

[12] B. Şenbaşlar and G. S. Sukhatme, "Dream: Decentralized real-time asynchronous probabilistic trajectory planning for collision-free multi-robot navigation in cluttered environments," IEEE Transactions on Robotics, 2024.

[13] B. Şenbaşlar, W. Hönig, and N. Ayanian, "Rlss: real-time, decentralized, cooperative, networkless multi-robot trajectory planning using linear spatial separations," Autonomous Robots, vol. 47, no. 7, pp. 921–946, 2023.

[14] E. Latif and R. Parasuraman, "Communication-efficient multi-robot exploration using coverage-biased distributed q-learning," IEEE Robotics and Automation Letters, vol. 9, no. 3, pp. 2622–2629, 2024.

[15] J. Gielis, A. Shankar, and A. Prorok, "A critical review of communications in multi-robot systems," Current robotics reports, vol. 3, no. 4, pp. 213–225, 2022.

[16] J. M. Flores Gonzalez, E. Coronado, and N. Yamanobe, "Ros-compatible robotics simulators for industry 4.0 and industry 5.0: A systematic review of trends and technologies," Applied Sciences, vol. 15, no. 15, p. 8637, 2025.

[17] P. Thalamy, B. Piranda, A. Naz, and J. Bourgeois, "Visiblesim: A behavioral simulation framework for lattice modular robots," Robotics and Autonomous Systems, vol. 147, p. 103913, 2022.

[18] F. H. Martínez and J. A. Delgado, "Hardware emulation of bacterial quorum sensing," in International Conference on Intelligent Computing. Springer, 2010, pp. 329–336.

[19] K. Dimitrov and V. Hristov, "Robustness and scalability of incomplete virtual pheromone maps for stigmergic collective exploration," Processes, vol. 12, no. 10, p. 2122, 2024.

[20] V. P. Tran, M. A. Garratt, K. Kasmarik, and S. G. Anavatti, "Dynamic frontier-led swarming: Multi-robot repeated coverage in dynamic environments," IEEE/CAA Journal of Automatica Sinica, vol. 10, no. 3, pp. 646–661, 2023.