Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi
Niğde Ömer Halisdemir University Journal of Engineering Sciences

Araştırma makalesi / Research article

www.dergipark.org.tr/tr/pub/ngumuh / www.dergipark.org.tr/en/pub/ngumuh

A local reservation-based conflict avoidance algorithm for bottleneck management in cooperative vehicles: A snow-plow operations example

İş birlikçi araçlarda darboğaz yönetimi için yerel rezervasyon tabanlı çatışma önleme algoritması: Kar küreme operasyonları örneği

Abdullah Karaağaç[1,*]

[1] *Karamanoğlu Mehmetbey Üniversitesi, Teknik Bilimler MYO, Harita ve Kadastro Programı, 70000, Karaman Türkiye*

Abstract

In automated work environments, collaborative mobile robots and autonomous vehicles often operate in spatially constrained and dynamically changing work areas where bottlenecks arise due to narrow passages, temporary obstacles, or shared operational resources. Addressing the scalability and robustness limitations of centralized coordination, this study proposes a decentralized, local interaction-based modeling framework where vehicles adapt their speed, distance, and yielding behaviors to neighboring agents and geometric constraints without relying on global planning or supervisory control. Through these interaction-oriented mechanisms, collective and self-organizing behaviors emerge, providing efficient bottleneck resolution under high arrival density conditions. Simulation-based evaluations in scenarios such as construction or airports demonstrate reduced congestion-induced delays, suppressed stop-start oscillations, and increased efficiency compared to non-cooperative baseline models. Treating bottleneck resolution as a fundamental coordination problem in automated construction systems, this study provides a scalable and robust modeling foundation for collaborative mobile robots operating in constrained industrial environments.

Keywords: Cooperative vehicle coordination, Bottleneck resolution, Autonomous mobile robots, Decentralized control, Constrained operational environments, Snow-plow operations

Öz

Otonomlaştırılmış çalışma ortamlarında, işbirlikçi mobil robotlar ve otonom araçlar genellikle dar geçitler, geçici engeller veya paylaşılan operasyonel kaynaklar nedeniyle darboğazların ortaya çıktığı, mekânsal olarak kısıtlı ve dinamik olarak değişen çalışma bölgelerinde faaliyet göstermektedir. Bu çalışmada, merkezi koordinasyonun ölçeklenebilirlik ve sağlamlık sınırlamalarını ele alan bu çalışma, araçların küresel planlama veya denetleyici kontrole dayanmadan komşu ajanlara ve geometrik kısıtlamalara göre hız, mesafe ve yol verme davranışlarını uyarladığı, merkezi olmayan, yerel etkileşime dayalı bir modelleme çerçevesi önerilmektedir. Bu etkileşim odaklı mekanizmalar aracılığıyla, yüksek varış yoğunluğu koşullarında verimli darboğaz çözümü sağlayan kolektif ve kendi kendine organize olan davranışlar ortaya çıkmaktadır. İnşaat veya havaalanı gibi senaryolarda yapılan simülasyon tabanlı değerlendirmeler, işbirlikçi olmayan temel modellere kıyasla tıkanıklık kaynaklı gecikmelerin azaldığını, dur-kalk salınımlarının bastırıldığını ve verimliliğin arttığını göstermektedir. Darboğaz çözümünü otomatikleştirilmiş inşaat sistemlerinde temel bir koordinasyon problemi olarak ele alan bu çalışma, kısıtlı endüstriyel ortamlarda çalışan işbirlikçi mobil robotlar için ölçeklenebilir ve sağlam bir modelleme temeli sağlamaktadır.

Anahtar kelimeler: İşbirlikçi araç koordinasyonu, Darboğaz çözümü, Otonom mobil robotlar, Merkezi olmayan kontrol, Kısıtlı operasyonel ortamlar, Kar küreme operasyonları

1 Introduction

The increasing adoption of automation and robotics in construction environments has led to the deployment of cooperative mobile robots and autonomous vehicles in spatially constrained and dynamically evolving work zones. Applications such as land transport, site logistics, and autonomous equipment coordination require multiple agents to operate simultaneously within narrow corridors, temporary passageways, and shared operational resources. Under such conditions, bottlenecks naturally emerge and often become the dominant factor limiting system-level efficiency, safety, and operational continuity [1–3].

Bottlenecks represent a fundamental coordination challenge in collective motion systems. When multiple agents attempt to traverse constrained regions, local conflicts, queue formation, stop-and-go oscillations, and even deadlocks may arise, particularly under high-density conditions [3]. In construction settings, these effects are further amplified by temporary obstacles, changing layouts, and heterogeneous agent capabilities, which reduce the effectiveness of static planning and centralized control schemes [4-7]. As a result, robust and scalable coordination strategies that can adapt to local conditions are essential for automated construction operations.

* Sorumlu yazar / Corresponding author, e-posta / e-mail: akaraagac@kmu.edu.tr (A. Karaağaç)
Geliş / Received: 16.04.2026 Kabul / Accepted: 14.06.2026 Yayımlanma / Published: 25.06.2026
doi: 10.28948/ngumuh.1931848

Existing approaches to bottleneck management in industrial and robotic systems have largely focused on explicit coordination mechanisms, including centralized scheduling, reservation-based control, and priority assignment [1, 4, 5]. Such methods have demonstrated effectiveness in structured environments and have provided important guarantees regarding safety and deadlock avoidance [4, 9]. However, their reliance on global supervision or predefined coordination rules may limit scalability and adaptability in construction scenarios characterized by uncertainty, frequent reconfiguration, and limited communication infrastructure [6].

Parallel research in multi-robot systems has emphasized decentralized navigation and collision avoidance in constrained spaces. Distributed planning, optimization-based coordination, and control-theoretic approaches have been proposed to ensure feasibility and safety in narrow passages and conflict zones [8–11]. While these methods successfully address local safety and deadlock prevention, they often treat bottlenecks as constraints to be handled rather than as dynamic interaction zones where collective behavior emerges. Consequently, system-level performance metrics such as throughput, delay, and flow stability are not always explicitly addressed.

Insights from traffic flow and connected vehicle research further highlight the importance of collective behavior in bottleneck scenarios. Studies on work-zone merging, cooperative lane changing, and connected vehicle coordination demonstrate that local interaction rules and information sharing can give rise to self-organized patterns that improve flow stability and bottleneck utilization [12–14]. These findings suggest that bottleneck resolution need not rely solely on centralized control but can instead emerge from cooperative local decision-making. Although rooted in road traffic contexts, the underlying principles of density-dependent interactions and emergent coordination are highly relevant to automated construction and industrial environments.

Despite the relevance of these insights, there remains a lack of unified modeling frameworks that explicitly capture bottleneck resolution as an emergent outcome of cooperative motion in construction-oriented automation systems. Existing studies typically separate traffic flow, multi-robot coordination, and industrial logistics into distinct domains, limiting cross-fertilization of concepts and methods [6]. In particular, the question of how cooperative vehicle groups can self-organize to resolve bottlenecks through purely local interactions, without centralized scheduling or global optimization, remains insufficiently explored.

Motivated by this gap, this study focuses on bottleneck problems in airport snow cleaning with ploughing trucks. A decentralized modeling framework is introduced to capture collective vehicle behavior under cooperative motion, emphasizing how local interaction rules give rise to emergent bottleneck resolution. By framing bottlenecks not merely as constraints but as interaction-driven phenomena, the proposed approach aims to provide a scalable and adaptable foundation for cooperative automation in constrained airport settings.

2 Literature Review

The coordination of multiple vehicles or mobile robots in spatially constrained environments has been widely investigated in the contexts of industrial automation, multi-robot systems, and, on the other hand, traffic flow theory. A central challenge shared across these domains is the formation of bottlenecks, where limited spatial capacity, merging flows, or shared operational resources lead to congestion, deadlocks, and performance degradation [1–3]. Modeling how agents behave collectively under these kinds of constraints therefore remains a fundamental challenge This challenge becomes particularly evident in industrial and construction-oriented environments, where autonomous mobile robots (AMRs) and automated guided vehicles (AGVs) often need to move through narrow corridors, temporary work zones, and shared logistics routes.

Existing studies have mostly approached coordination in such settings through centralized scheduling frameworks, reservation-based control strategies, or predefined priority schemes [4, 5].

Scalable traffic management and decentralized AGV control frameworks have demonstrated deadlock-free operation and improved throughput in large-scale industrial fleets [1, 4]. However, these approaches often depend on explicit coordination protocols or global planning structures, which may limit flexibility and robustness in dynamically changing construction environments [6, 7]. From a robotics perspective, decentralized multi-robot navigation in constrained spaces has been extensively studied with a focus on collision avoidance, safety guarantees, and deadlock prevention. Graph-based planning, optimization-based coordination, and control-theoretic approaches have been proposed to ensure feasibility and liveness in narrow passages and conflict zones [5, 8–11]. In particular, distributed deadlock-avoidance strategies highlight the importance of local interaction rules in maintaining system-wide operability under spatial constraints [9]. While these methods provide strong safety and feasibility guarantees, collective flow efficiency and emergent performance indicators such as throughput, delay, and stop-and-go behavior are often treated as secondary considerations.

Complementary insights emerge from traffic flow and connected vehicle literature, where bottlenecks, merges, and lane reductions are recognized as fundamental sources of instability and capacity drop. Studies on work-zone merging and cooperative lane-changing demonstrate that local coordination and information sharing can significantly improve throughput and flow stability under constrained geometries [2, 12, 13]. Moreover, self-organizing and cooperative vehicle strategies have been shown to suppress stop-and-go oscillations and enhance bottleneck utilization without centralized control [12, 14]. Although these studies are largely grounded in road traffic scenarios, the core mechanisms of density-dependent interactions and emergent coordination translate naturally to automated construction and industrial environments.

Despite this progress, an important gap still exists in understanding how cooperative vehicle or robot groups resolve bottlenecks as a self-organized outcome of local interactions, rather than through explicit scheduling, reservation policies, or global optimization. Existing approaches predominantly frame bottleneck handling as a control or planning problem [4, 5, 9], while fewer studies explicitly address collective behavior and flow-level performance emerging from decentralized cooperative motion [14]. In the context of automated construction environments—characterized by temporary constraints, evolving layouts, and heterogeneous agents, there is a need for scalable modelling frameworks that capture emergent bottleneck resolution driven by local decision-making.

This study addresses this gap by focusing on bottleneck scenarios as a representative coordination challenge and by proposing a decentralized modeling framework in which collective vehicle behavior emerges from cooperative local interactions. By emphasizing emergent dynamics rather than explicit centralized control, the proposed approach complements existing robotics and industrial automation methods and provides a unified perspective on bottleneck resolution in constrained automated construction environments.

3 Methodology

3.1 Problem definition

This study considers a cooperative fleet of autonomous service vehicles operating in constrained construction, industrial, and airport-like environments. In such environments, vehicles are often required to move along predefined service paths while sharing narrow corridors, temporary access routes, transition zones, or limited-capacity operational areas. These restricted regions create bottlenecks, where only one vehicle can safely pass at a given time.

Airport snow-plow operations are used in this study as a representative application case. During snow removal, multiple cleaning vehicles may approach the same narrow transition area or service corridor with different arrival times, speeds, and operational priorities. If these vehicles are not coordinated, they may form queues, stop unnecessarily, create unstable stop-and-go motion, or reduce the utilization efficiency of the constrained section.

The bottleneck is modelled as a single-directional critical section with unit capacity. This means that only one vehicle is allowed to occupy the bottleneck at any time. These areas are labelled and demonstrated in Figure 1. Vehicles approaching the bottleneck are assumed to have basic sensing and communication capabilities, allowing them to detect nearby vehicles and exchange limited local information. The main objective is to regulate vehicle entry into the bottleneck in a way that ensures safety, avoids deadlock, reduces waiting time, and maintains stable traffic flow.

The proposed method does not rely on centralized global scheduling or full-route optimization. Instead, bottleneck access is managed through a local reservation-based decision process. Vehicles approaching the bottleneck submit passage requests, and a temporary access token is assigned to one vehicle at a time according to a priority score. Vehicles without access permission wait safely in the holding zone until the critical section becomes available.

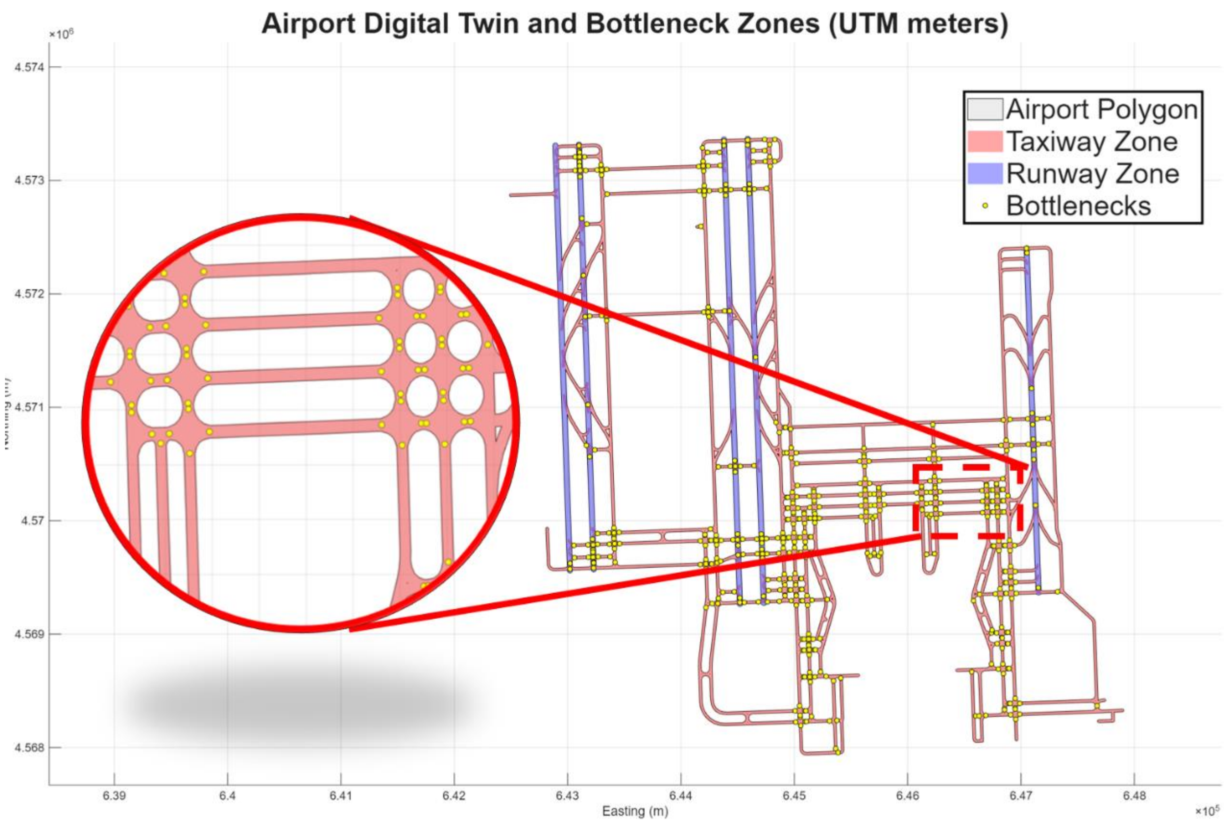

Figure 1. Airport digital twin and bottleneck zones (Yellow dot areas)

3.2 The bottleneck zone structure

The bottleneck area is divided into three operational zones: the approach zone, the holding zone, and the critical section.

The Approach Zone (AZ) is the area before the bottleneck where vehicles detect the upcoming constrained region and start the coordination process. When a vehicle enters this zone and reaches the reservation trigger distance, it estimates its expected time of arrival and sends a request to access the bottleneck.

The Holding Zone (HZ) is a safe waiting area located before the critical section. Vehicles that do not receive permission to enter the bottleneck reduce their speed or stop in this zone while maintaining a safe distance from other vehicles. This zone prevents vehicles from blocking upstream traffic and allows the fleet to form an orderly passing sequence.

The Critical Section (CS) represents the bottleneck itself. It is modeled as a shared resource with unit capacity. Therefore, only the vehicle holding the active reservation token is permitted to enter the critical section. After the authorized vehicle exits the critical section, the token is released and reassigned to the next eligible vehicle.

This three-zone structure provides a simple and implementable operational framework. It separates early coordination, safe waiting, and exclusive passage into distinct stages, making the bottleneck management process easier to model and control.

3.3 Local reservation-based coordination framework

The proposed coordination framework combines a local reservation mechanism with smooth motion regulation. The reservation mechanism determines which vehicle is allowed to enter the bottleneck, while the motion regulation component ensures safe deceleration, stopping, and re-acceleration in the approach and holding zones.

When a vehicle approaches the bottleneck, it calculates whether a reservation request must be issued. This decision is based on the minimum required coordination distance, which depends on vehicle speed, braking capability, communication delay, and safety margin. If the vehicle is within this trigger distance, it becomes a requester for the critical section.

For each requesting vehicle, a priority score is calculated using three local factors: estimated time of arrival, accumulated waiting time, and task-related priority. The vehicle with the lowest priority score is selected as the next token holder. Once the token is assigned, the selected vehicle is allowed to enter the critical section, while all other vehicles remain in the holding zone.

The token is released only after the authorized vehicle exits the critical section. This rule ensures that no two vehicles occupy the bottleneck simultaneously. Since the decision process is based only on local information, the proposed framework remains scalable and suitable for dynamic field conditions where centralized planning may not be practical.

3.4 Decision model and mathematical formulation

Let the cooperative fleet consist of N autonomous vehicles indexed by $i \in \{1, \ldots, N\}$. Each vehicle is described by its longitudinal position $x_i(t)$, velocity $v_i(t)$, and distance $d_i(t)$ to the bottleneck entrance. The bottleneck is modeled as a single-directional critical section (CS) with unit capacity, such that at most one vehicle can occupy the CS at any time.

Access to the critical section is regulated through a discrete reservation token $T(t)$, defined as Equation (1).

$$T(t) \in \{0,1\} \tag{1}$$

where $T(t) = 0$ means that the critical section is free, and $T(t) = i$ means that vehicle i currently has permission to enter or occupy the critical section. The identity of the vehicle holding the access token is defined separately as given in Equation (2).

$$\mathrm{h}(t) \in \{0,1,2, \ldots, N\} \tag{2}$$

where $h(t) = 0$means that no vehicle currently holds the token, and $h(t) = j$means that vehicle $j$has permission to enter or pass through the critical section. This separation prevents ambiguity between the occupancy state of the bottleneck and the identity of the authorized vehicle.

A vehicle i becomes a requester if it is located in the approach zone and its distance to the critical section is smaller than or equal to the required coordination distance is defined as Equation (3).

$$di(t) \leq d_{req,i}(t) \tag{3}$$

The estimated time of arrival of vehicle (ET) i is calculated as Equation (4):

$$\mathrm{ET\ A}_i(\mathrm{t}) = \frac{d_i(t)}{\max\left(v_i(t), \varepsilon\right)} \tag{4}$$

where $\varepsilon > 0$ is a small constant preventing numerical singularities at low speeds. Based on local information, a priority score is assigned to each requesting vehicle as given in Equation (5).

$$S_i(t) = w_1 \,\mathrm{ET\ A}_i(t) + w_2\, W_i(t) + w_3\, P_i, \tag{5}$$

where $W_i$denotes the accumulated waiting time in the holding zone, $P_i$represents task-related priority, and $w_1, w_2, w_3$are weighting coefficients. The vehicle with the minimum score $S_i$is granted access to the critical section, while all others remain in the holding zone.

To ensure safe operation, reservation requests must be issued before a minimum coordination distance d_{req}, defined as Equation (6).

$$d_{\mathrm{req,i}}(t) = \frac{v_i(t)^2}{2a_{\max}} + v_i(t)\tau_{\mathrm{comm}} + d_{\mathrm{safe}} \tag{6}$$

where $a_{\max}$ is the maximum achievable deceleration, τ_{comm} denotes communication latency, and d_{safe} is a safety margin. This constraint ensures that vehicles can decelerate smoothly and stop within the holding zone if access to the critical section is not granted.

If the critical section is free, the next token holder is selected according to Equation (7).

$$j = arg \min_{i \in R(t)} S_i(t) \tag{7}$$

where $R(t)$ is the set of requesting vehicles. The token is then assigned as $T(t) = j$.

Once the authorized vehicle exits the critical section, the reservation token is released ($T(t) = 0$), and the decision process is repeated. Although access control is discrete, vehicle motion within the approach and holding zones follows continuous speed regulation, enabling smooth deceleration and re-acceleration. Also, vehicle motion is updated in discrete time and is formulated as Equations (8) and Equation (9).

$$v_i(t + \Delta t) = max(0, v_i(t) + a_i(t)\Delta t) \tag{8}$$

$$x_i(t + \Delta t) = x_i(t) + v_i(t)\Delta t \tag{9}$$

where Δt is the simulation time step. Vehicles holding the token are allowed to pass through the critical section under the maximum critical-section speed limit. Vehicles without a token apply holding behavior and maintain safe spacing in the holding zone.

As a result, bottleneck resolution emerges from local cooperative interactions rather than explicit global scheduling.

3.5 Simulation setup and performance metrics

The proposed coordination framework is evaluated using a discrete-time numerical simulation. The simulation represents a single-directional bottleneck scenario inspired by airport snow-plow operations, while the underlying coordination structure is intended to be applicable to constrained construction and industrial environments.

Vehicles move along predefined service paths toward a unit-capacity bottleneck. All vehicles are assumed to have identical geometric and kinematic limits. However, their initial positions, arrival times, and task priorities may vary across scenarios. The proposed cooperative strategy is compared with a non-cooperative baseline strategy. In the baseline case, vehicles rely only on local collision avoidance and do not use explicit reservation-based access control.

Several operational load conditions are tested by changing the arrival spacing between successive vehicles. Shorter arrival spacing represents heavier traffic demand and creates stronger bottleneck pressure. Each scenario is repeated multiple times with different initial arrival patterns to improve the consistency of the results.

3.6 Performance metrics

The performance of the proposed local reservation-based coordination method is evaluated using five metrics: throughput, average delay, maximum delay, stop-and-go index, and minimum safety margin. These metrics are selected to assess efficiency, fairness, flow stability, and safety under different bottleneck load conditions. Let N_p denote the number of vehicles that successfully pass through the bottleneck during the simulation period, T_{sim} denote the total simulation duration, and N denote the number of vehicles in a given scenario.

Throughput measures how efficiently the bottleneck is utilized. It is defined as the number of vehicles successfully passing through the critical section per unit time in Equation (10).

$$Q = \frac{N_p}{T_{sim}} \tag{10}$$

where Q is the throughput value. A higher throughput indicates that the bottleneck is used more efficiently.

Average delay represents the mean waiting time experienced by vehicles in the holding zone before entering the critical section. For each vehicle i, the holding-zone delay is denoted by D_i. The average delay is calculated as Equation (11).

$$\bar{D} = \frac{1}{N}\sum_{i=1}^{N} D_i \tag{11}$$

This metric reflects the overall time cost caused by bottleneck coordination. Lower average delay indicates a more efficient access management strategy.

Maximum delay represents the longest waiting time observed among all vehicles is given in Equation (12).

$$D_{max} = \max_{i \in \{1,2,\ldots,N\}} D_i \tag{12}$$

This metric is used to evaluate fairness and to determine whether any individual vehicle experiences excessive waiting. A lower maximum delay indicates that the access mechanism prevents extreme waiting conditions and supports a more balanced vehicle sequence.

The stop-and-go index is used to measure the magnitude of speed fluctuations near the bottleneck. It is calculated using the standard deviation of vehicle speeds within the bottleneck influence area as given in Equation (13).

$$SGI = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\frac{1}{K}\sum_{k=1}^{K}(v_i(k) - \bar{v}_i)^2} \tag{13}$$

where K is the number of sampled time steps in the bottleneck influence area, $v_i(k)$ is the speed of vehicle i at time step k, and $\bar{v}_i$ is the mean speed of vehicle i over the same interval. A lower stop-and-go index indicates smoother vehicle motion and better flow stability.

Minimum safety margin is defined as the smallest observed inter-vehicle distance during the simulation in Equation (14).

$$d_{min} = \min_{t} \min_{i \neq j} d_{ij}(t) \quad (14)$$

where $d_{ij}(t)$denotes the distance between vehicles $i$and $j$at time t. This metric is used to confirm that the coordination strategy maintains safe separation between vehicles. In the simulation, the observed minimum safety margin is compared with the predefined safety distance d_{safe}. A feasible and safe coordination result should satisfy is given in Equation (15).

$$d_{min} \geq d_{safe} \quad (15)$$

By comparing these metrics for the proposed cooperative reservation-based method and the non-cooperative baseline, the simulation evaluates whether local reservation-based coordination improves bottleneck performance under different operational load conditions. In the application section, the reported values are presented as representative MATLAB simulation results and are given as mean ± standard deviation over repeated runs.

4 Application and simulation-based evaluation

4.1 Simulation environment: Airport snow-plow operation scenario

To evaluate the proposed local reservation-based coordination framework, a MATLAB-based discrete-time simulation environment was developed using airport snow-plow operations as a representative application case. Although the proposed method is intended for general constrained construction and industrial environments, airport snow removal provides a suitable test scenario because multiple service vehicles must operate under strict safety, time, and continuity requirements.

The simulated environment consists of a predefined service route including a single-directional bottleneck. This bottleneck represents a narrow service corridor, restricted transition area, or limited-capacity operational passage that may occur during airport ground operations. As illustrated in Figure 2, cleaning vehicles move along assigned routes and approach the bottleneck from the same direction. The bottleneck region is divided into three operational zones: the Approach Zone (AZ), the Holding Zone (HZ), and the Critical Section (CS).

The Approach Zone is the area where vehicles detect the upcoming bottleneck and initiate the coordination process as demonstrated in Figure 3. The Holding Zone is used as a safe waiting area for vehicles that do not receive access permission. The Critical Section is modeled as a unit-capacity constrained region, meaning that only one vehicle is allowed to occupy it at any time. This structure allows the simulation to represent both the geometric restriction of the bottleneck and the operational logic required for safe vehicle sequencing.

In the MATLAB simulation, all vehicles have identical geometric and kinematic limits. However, their initial positions, arrival times, and task priorities are varied to create different operational load conditions. This setup makes it possible to evaluate how the proposed coordination method behaves under both low-density and high-density arrival patterns.

4.2 Algorithm implementation and scenario design

The proposed reservation-based algorithm was implemented in MATLAB as a discrete-time numerical simulation. At each simulation step, vehicles update their positions, velocities, waiting states, reservation requests, and token status according to the local coordination logic described in the methodology section. MATLAB was selected because it allows transparent implementation of the vehicle update rules, parameter control, scenario repetition, and graphical evaluation of time–space (as shown in Figure 4) and delay-based outputs.

The main algorithmic variables used in the application are consistent with the definitions given in the methodology section. R(t) denotes the set of vehicles requesting access to the critical section at time t. T(t) is the binary reservation state of the critical section, where T(t)=0 indicates that the critical section is free and T(t)=1 indicates that it is reserved or occupied. The identity of the token-holding vehicle is represented by h(t). Here, h(t)=0 means that no vehicle currently holds the token, whereas h(t)=j means that vehicle j has permission to enter or pass through the critical section.

The simulation was designed to test the proposed method under different operational densities. The number of vehicles was varied between 5 and 20, while the arrival spacing between successive vehicles was changed between 2 s and 10 s. Shorter arrival spacing values represent high-density conditions and create stronger bottleneck pressure. Longer arrival spacing values represent lighter operational conditions.

Each scenario was repeated 30 times using different initial arrival patterns. Since the purpose of this application is to demonstrate the operational behavior of the proposed framework rather than to report field-calibrated airport data, the numerical values are presented as representative simulation results under the selected parameter settings. The reported values represent the average behavior over repeated MATLAB simulation runs.

The proposed cooperative reservation-based strategy was compared with a non-cooperative baseline. In the baseline strategy, vehicles do not use explicit bottleneck reservation. Instead, they rely only on local collision avoidance and react to the bottleneck when they encounter a potential conflict. In contrast, the proposed method regulates access before vehicles reach the critical section, allowing the fleet to form a more orderly passing sequence. The performance metrics defined in the previous section were used for the comparison. The main simulation parameters are summarized in Table 1.

Uncoordinated profiles shown in Figure 5 further illustrate the effect of cooperative coordination. Without explicit bottleneck access control, vehicles experience abrupt deceleration and acceleration near the constrained region. The proposed approach significantly smooths these profiles by enforcing early reservation decisions and continuous speed adaptation.

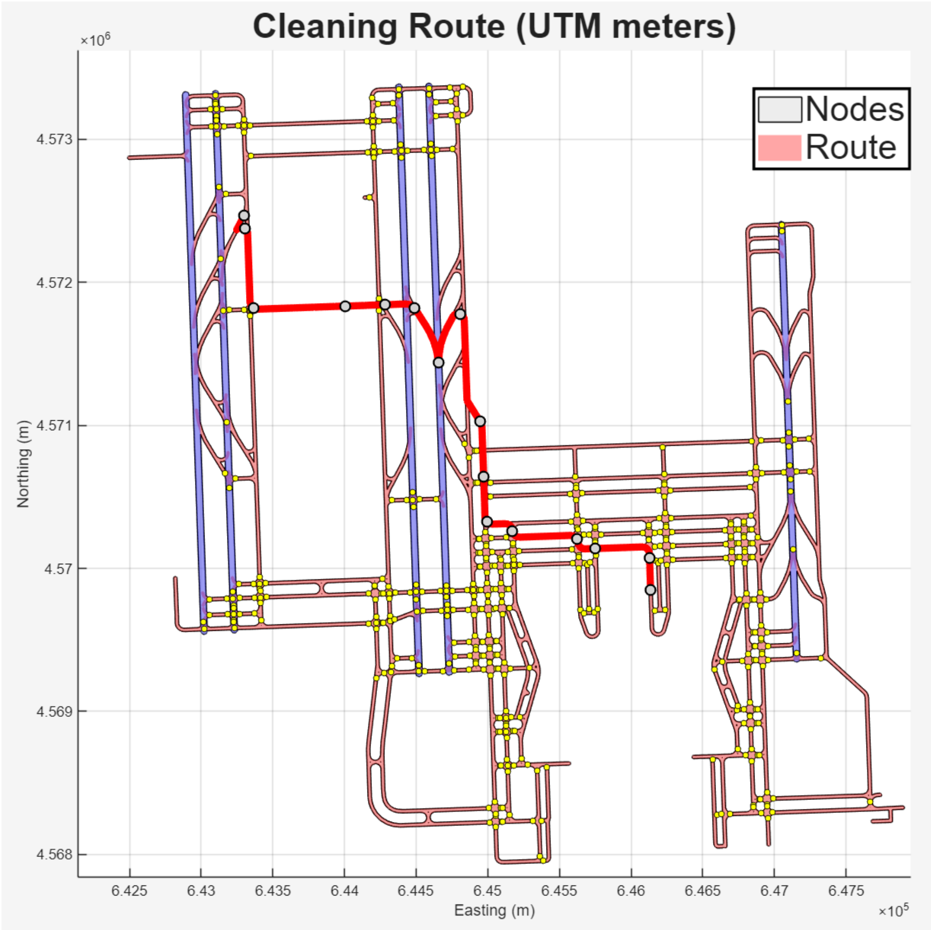

Figure 2. Assigned cleaning route

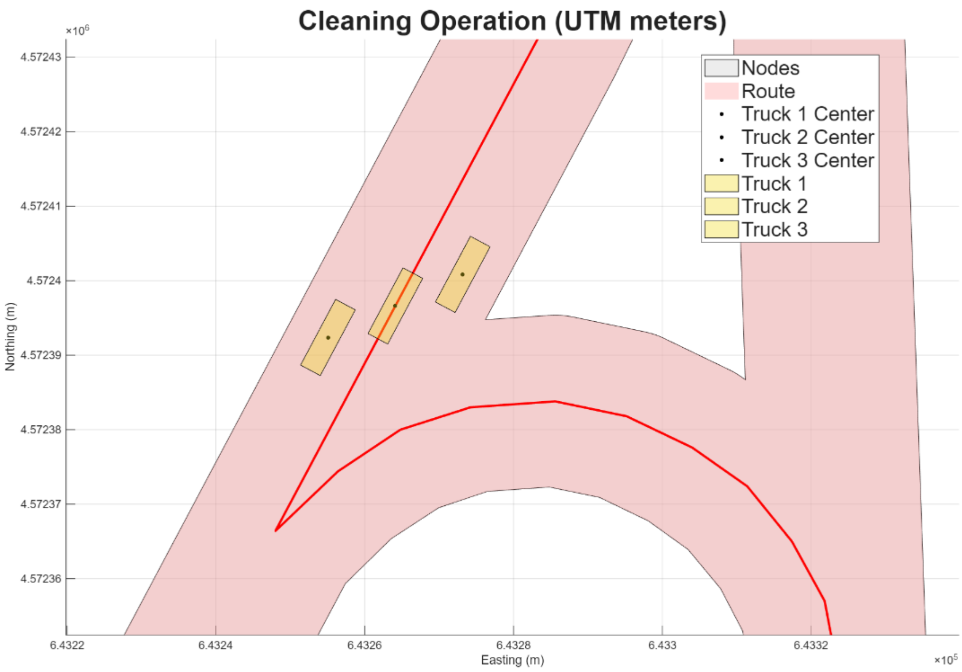

Figure 3. Cleaning operation approaching a bottleneck zone in simulation

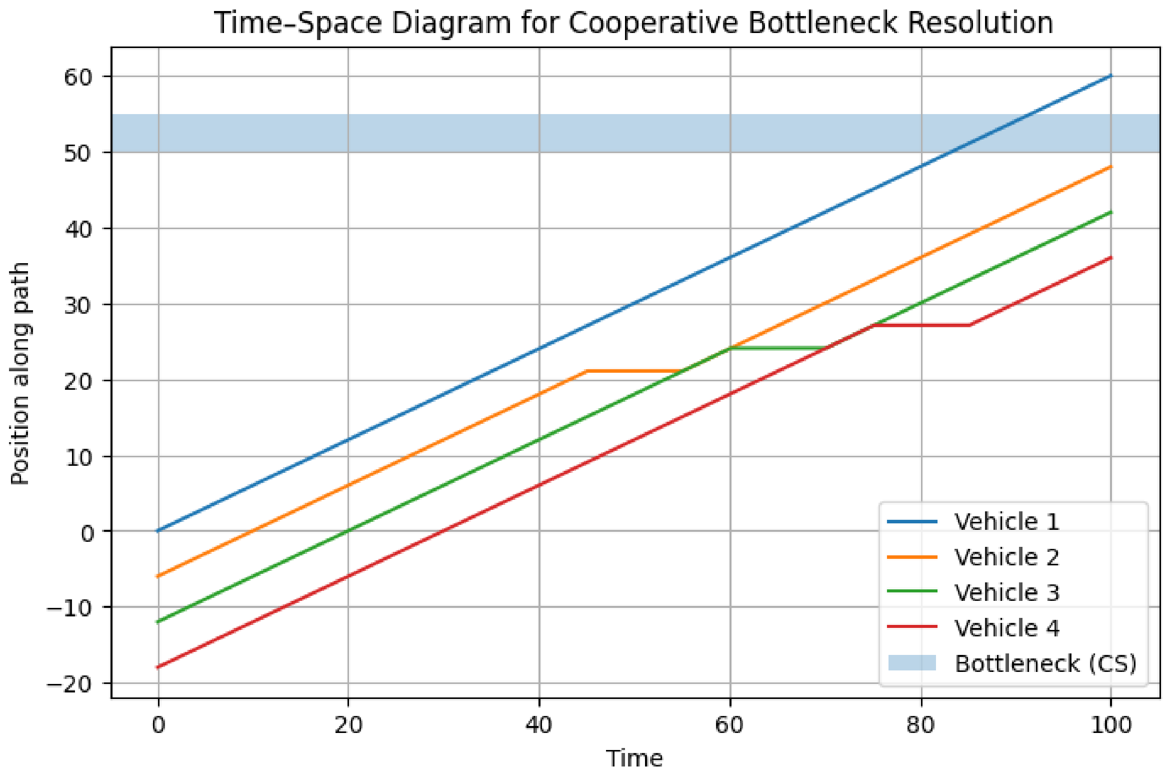

Figure 4. Time-Space diagram

Table 1. Simulation parameters used in the numerical experiments

Parameter	Symbol	Value / Range	Description
Number of vehicles	N	5–20 vehicles	Fleet size approaching the bottleneck
Simulation time step	(Δt)	0.1 s	Discrete update interval
Maximum vehicle speed	v_{max}	8 m/s	Nominal upper speed limit
Critical section speed limit	$v_{CS,max}$	4 m/s	Maximum speed inside the bottleneck
Maximum deceleration	a_{max}	2.5 m/s²	Comfortable braking limit
Communication/decision delay	τ_{comm}	0.2 s	Local communication and processing delay
Minimum safety distance	d_{safe}	5 m	Required minimum inter-vehicle spacing
Approach zone length	L_{AZ}	80 m	Distance before the holding zone
Holding zone length	L_{HZ}	30 m	Safe waiting region before the bottleneck
Critical section length	L_{CS}	25 m	Length of the constrained passage
Arrival spacing	$\boldsymbol{\Delta t}_{arr}$	2–10 s	Time interval between approaching vehicles
Number of repetitions	—	30 runs	Repeated simulation runs per scenario
ETA weight	w_1	1.0	Weight of estimated arrival time
Waiting-time weight	w_1	0.5	Weight of accumulated waiting time
Task-priority weight	w_1	0.3	Weight of operational priority
Numerical stability constant	ε	0.1 m/s	Avoids division by zero in ETA calculation

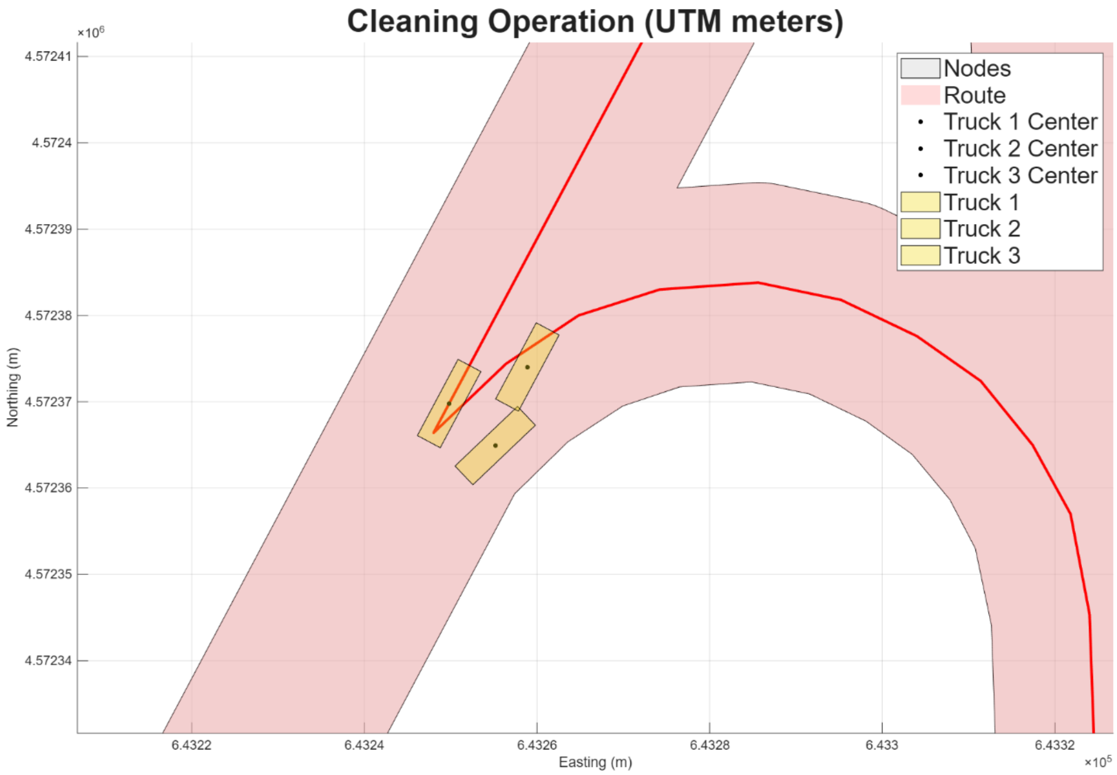

Figure 5. Uncoordinated congestion in a bottleneck

4.3 Quantitative performance comparison

To support the visual simulation results with explicit numerical evidence, the proposed cooperative reservation-based strategy was compared with the non-cooperative baseline under increasing operational load. It should be noted that the numerical values reported in this section are representative MATLAB simulation results generated under the selected parameter settings. They are used to compare the relative behavior of the proposed cooperative coordination strategy and the non-cooperative baseline. Therefore, they should not be interpreted as field-measured airport operation data, but as controlled simulation outputs supporting the algorithmic evaluation of the proposed method.

Table 2 presents representative simulation results obtained from repeated MATLAB runs under the selected parameter settings. The values are reported as mean ± standard deviation over 30 runs.

The representative simulation results indicate that the proposed local reservation-based strategy can improve bottleneck performance compared with the non-cooperative baseline under the selected parameter settings. In the representative scenario, the average holding-zone delay decreases from 18.6 s to 10.2 s, corresponding to an improvement of approximately 45.2%. The maximum delay is also reduced from 34.8 s to 18.9 s, indicating that the proposed method can improve fairness by preventing excessive waiting for individual vehicles.

The throughput increases from 0.42 veh/s to 0.56 veh/s. This improvement occurs because the proposed method assigns bottleneck access before vehicles reach the critical section, reducing unnecessary braking and delayed re-acceleration. The stop-and-go index decreases from 0.78 to 0.41, which suggests that the cooperative strategy produces smoother motion near the bottleneck. The minimum safety margin remains above the predefined 5 m safety threshold in both strategies. However, the proposed method maintains this safety margin with more stable and predictable vehicle behavior.

4.4 Interpretation of simulation results

Figure 4 presents the time–space diagram of vehicle movement near the bottleneck. In the non-cooperative case, vehicles accumulate upstream of the critical section and show irregular waiting and re-acceleration patterns. This behavior occurs because vehicles react to local conflicts only after reaching the constrained region. In contrast, the proposed method creates a more ordered passing sequence by assigning access permission before vehicles enter the critical section.

Figure 5 illustrates the congestion behavior observed under the non-cooperative baseline. Abrupt deceleration and repeated acceleration are visible near the bottleneck region. These stop-and-go patterns increase delay and reduce the stability of the vehicle flow. The proposed reservation-based method reduces these oscillations by moving the decision point upstream into the approach and holding zones.

Figure 6 reports the average and maximum holding-zone delays under different operational load conditions. Error bars indicate one standard deviation over 30 repeated MATLAB simulation runs. As the arrival spacing between vehicles decreases, the non-cooperative baseline produces longer delays and greater variability. In contrast, the proposed cooperative method maintains lower average and maximum delays across the tested load levels. This result indicates that the method is more robust under high-density bottleneck conditions.

The numerical comparison and the visual simulation outputs support the same conclusion. The proposed local reservation-based coordination framework improves bottleneck utilization, reduces unnecessary waiting, suppresses stop-and-go behavior, and preserves safe vehicle separation in the representative simulation setting. These findings suggest that the method provides a practical balance between efficiency, fairness, flow stability, and safety in constrained operational environments.

4.5 Statistical consistency of the results

The repeated MATLAB simulation runs provide a simple statistical basis for comparing the two coordination strategies. The lower standard deviation values observed under the proposed method indicate that the cooperative reservation mechanism produces more consistent bottleneck behavior. All algorithm structure and pseudo-code are given in Algorithm 1. In particular, the standard deviation of average delay decreases from 3.4 s in the baseline case to 2.1 s in the proposed method. Similarly, the standard deviation of maximum delay decreases from 4.7 s to 3.2 s.

These representative results suggest that, under the selected simulation settings, the proposed method not only improves mean performance but also reduces variability across repeated runs. This is important for airport snow-plow and similar service operations, where predictable vehicle behavior is as important as average efficiency. A coordination strategy that produces lower delay variability can support safer and more reliable field operation planning.

5 Discussion

The presented results demonstrate that effective bottleneck resolution can emerge from local cooperative interactions without reliance on centralized scheduling or global trajectory planning. By treating the bottleneck as a shared resource and regulating access through a lightweight reservation mechanism, vehicles dynamically reorganize into an efficient passing sequence.

Importantly, the empirical example confirms that the proposed framework is well suited for operational environments such as airports, where spatial constraints, safety requirements, and dynamic task execution are dominant factors. The simplicity of the coordination logic and its direct correspondence to implementable code make the approach attractive for real-world deployment of cooperative robotic fleets.

Table 2. Representative MATLAB simulation results for the baseline and proposed methods

Performance metric	Non-cooperative baseline	Proposed cooperative method	Improvement
Average holding-zone delay	18.6 ± 3.4 s	10.2 ± 2.1 s	45.2% reduction
Maximum holding-zone delay	34.8 ± 4.7 s	18.9 ± 3.2 s	45.7% reduction
Throughput	0.42 ± 0.04 veh/s	0.56 ± 0.03 veh/s	33.3% increase
Stop-and-go index	0.78 ± 0.09	0.41 ± 0.06	47.4% reduction
Minimum safety margin	5.1 ± 0.3 m	5.4 ± 0.2 m	Maintained above limit

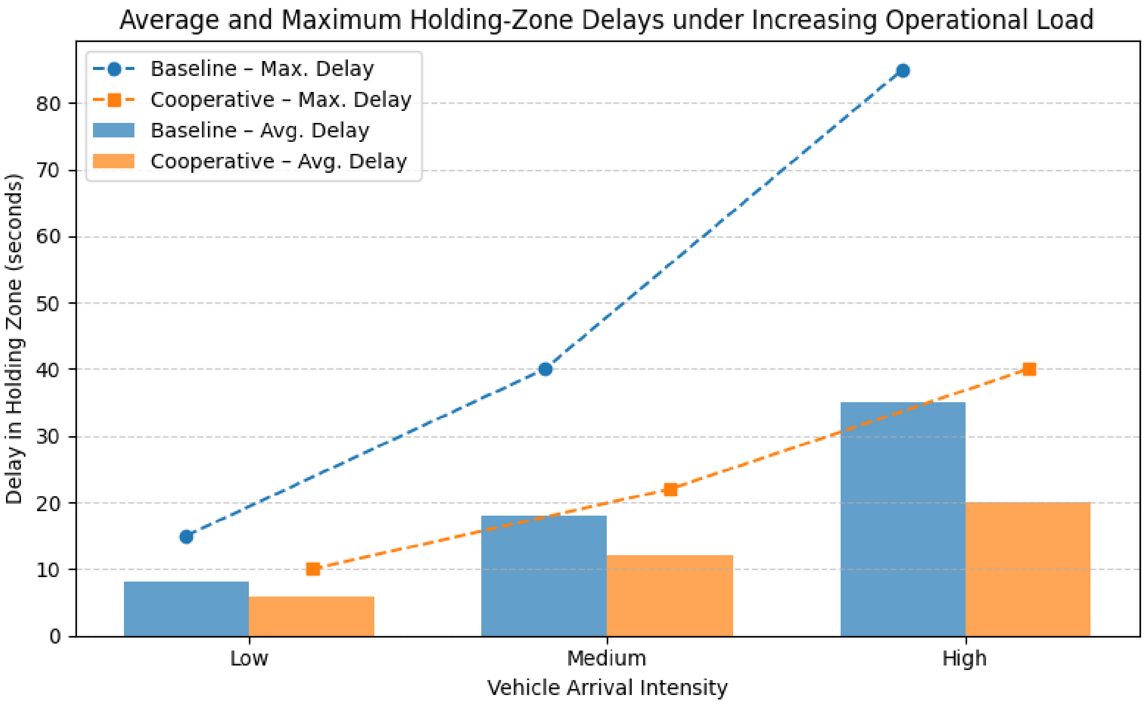

Figure 6. Average and maximum holding-zone delays under increasing operational load

Algorithm 1: Formulation (Pseudo-Code) of simulation process

```
Inputs:
  Vehicles i = 1,...,N with states x_i, v_i, a_i, d_i
  Parameters: Δt, a_max, τ_comm, d_safe, ε, v_CS,max
  Weights: w1, w2, w3
  Zones: Approach Zone (AZ), Holding Zone (HZ), Critical
Section (CS)

State variables:
  T ∈ {0,1}        // binary reservation state of the critical
section
  h ∈ {0,1,...,N}   // identity of the token-holding vehicle
  W_i              // accumulated waiting time of vehicle i
  R(t)             // requester set at time t

Initialize:
  T ← 0
  h ← 0
  W_i ← 0 for all vehicles

For each simulation step t do:
  1. Request generation
    R(t) ← ∅
    For each vehicle i do:
      If vehicle i is in AZ and d_i ≤ d_req,i then:
        Compute ETA_i:
        ETA_i = d_i / max(v_i, ε)
        Compute priority score:
        S_i = w1*ETA_i - w2*W_i - w3*P_i
        Add vehicle i to requester set R(t)
      End if
  2. Token assignment
    If T = 0 and R(t) is not empty then:
      Select the vehicle with the minimum priority score:
       j = argmin S_i, for all i ∈ R(t)
      Reserve the critical section:
      T ← 1
      Assign token holder:
      h ← j
    End if
  3. Motion regulation
    For each vehicle i do:
      If i = h then:
        Allow vehicle i to enter or pass through CS
        Apply critical-section speed limit:
        v_i ≤ v_CS,max
        If vehicle i exits CS then:
          Release critical-section reservation:
          T ← 0
          Clear token holder:
          h ← 0
        End if

      Else:
        Apply holding behavior in HZ
        Maintain safe distance from the preceding vehicle
        If vehicle i is waiting in HZ then:
          W_i ← W_i + Δt
        End if
      End if
    End for

  4. Vehicle state update
    For each vehicle i do:
      Update velocity:
      v_i(t+Δt) = max(0, v_i(t) + a_i(t)Δt)
      Update position:
      x_i(t+Δt) = x_i(t) + v_i(t)Δt
      Update distance to the critical section:
      d_i(t+Δt)
    End for
End loop
```

6 Conclusion

This study investigated bottleneck resolution as a fundamental coordination challenge for cooperative vehicle fleets operating in constrained airport and construction-like environments. Focusing on single-directional bottlenecks encountered by autonomous cleaning vehicles, a decentralized hybrid coordination framework was proposed, combining local reservation-based access control with continuous motion regulation. By treating the bottleneck as a shared resource and relying solely on local interactions, the approach enables safe, deadlock-free, and scalable coordination without centralized scheduling or global trajectory planning.

Simulation results and an empirical airport-inspired example demonstrated that the proposed framework induces orderly vehicle passing behavior, suppresses stop-and-go oscillations, and significantly improves bottleneck red to a non-cooperative baseline. The benefits become increasingly pronounced under high arrival intensity, highlighting the robustness of the approach under constrained and competitive operating conditions. Importantly, these performance gains are achieved without compromising safety or requiring complex coordination infrastructure.

The simplicity and implementability of the proposed decision logic make the framework well suited for real-world deployment in automated construction and airport operations, where spatial constraints, dynamic layouts, and operational uncertainty are dominant. Future work will extend the framework to bidirectional bottlenecks, heterogeneous vehicle capabilities, and field-level validation using real operational data, further advancing cooperative automation in constrained industrial environments.

Conflict of interest

The authors declare that there is no conflict of interest.

Similarity Rate (iThenticate): 5%

References

[1] I. Draganjac, T. Petrović, D. Miklić, Z. Kovačić and J. Z. Oršulić, Highly scalable traffic management of

autonomous industrial transportation systems, Robotics and Computer-Integrated Manufacturing, 63, 101915, 2020. https://doi.org/10.1016/j.rcim.2019.101915

[2] M. Algomaiah and Z. Li, Exploring work zone late merge strategies with and without enabling connected vehicles technologies, Transportation Research Interdisciplinary Perspectives, 9, 100316, 2021. https://doi.org/10.1016/j.trip.2021.100316

[3] Y. Zhou, H. Hu, Y. Liu, S. W. Lin and Z. Ding, A distributed method to avoid higher-order deadlocks in multi-robot systems, Automatica, 112, 108706, 2020. https://doi.org/10.1016/j.automatica.2019.108706

[4] I. Draganjac, D. Miklić, Z. Kovačić, G. Vasiljević and S. Bogdan, Decentralized control of multi-AGV systems in autonomous warehousing applications, IEEE Transactions On Automation Science And Engineering, 13(4), 1433-1447, 2016. https://doi.org/10.1109/TASE.2016.2603781

[5] V. Digani, M. A. Hsieh, L. Sabattini and C. Secchi, Coordination of multiple AGVs: a quadratic optimization method, Autonomous Robots, 43(3), 539-555, 2019. https://doi.org/10.1007/s10514-018-9730-9

[6] G. Fragapane, D. Ivanov, M. Peron, F. Sgarbossa and J. O. Strandhagen, Increasing flexibility and productivity in Industry 4.0 production networks with autonomous mobile robots and smart intralogistics, Annals of Operations Research, 308(1), 125-143, 2022. https://doi.org/10.1007/s10479-020-03526-7

[7] N. Boysen, D. Briskorn and S. Emde, Sequencing of picking orders in mobile rack warehouses, European Journal of Operational Research, 259(1), 293–307, 2017. https://doi.org/10.1016/j.ejor.2016.09.046

[8] M. Čáp, J. Vokřínek and A. Kleiner, Complete Decentralized Method for On-Line Multi-Robot Trajectory Planning in Valid Infrastructures, International Conference on Automated Planning and Scheduling, 2015. https://doi.org/10.48550/arXiv.1501.07704

[9] R., Chandra, V. Zinage, E. Bekolas, P. Stone and J. Biswas, Deadlock-free, safe, and decentralized multi-robot navigation in constrained environments using control barrier functions, Autonomous Robots, 49(2), 12, 2025. https://doi.org/10.1007/s10514-025-10194-8

[10] L. Luo, N. Chakraborty and K. Sycara, Distributed Algorithms for Multirobot Task Assignment With Task Deadline Constraints, IEEE Transactions on Automation Science and Engineering, 12(3), 876-888, 2015. https://doi.org/10.1109/TASE.2015.2438032

[11] W. Hönig, T. K. Kumar, L. Cohen, H. Ma, H. Xu, N. Ayanian and S. Koenig, Multi-Agent Path Finding with Kinematic Constraints, Proceedings of the International Conference on Automated Planning and Scheduling, pp. 477-485, 2016. https://doi.org/10.1609/icaps.v26i1.13796

[12] S. Heshami and L. Kattan, Towards self-organizing connected and automated vehicles: A coalitional game theory approach for cooperative lane-changing decisions, Transportation Research Part C: Emerging Technologies, 166, 104789, 2024. https://doi.org/10.1016/j.trc.2024.104789

[13] Y. Fang, H. Min, X. Wu, W. Wang, X. Zhao and G. Mao, On-ramp merging strategies of connected and automated vehicles considering communication delay, IEEE Transactions on Intelligent Transportation Systems, 23(9), 15298-15312, 2022. https://doi.org/10.1109/TITS.2022.3140219

[14] A. Talebpour and H. S. Mahmassani, Influence of connected and autonomous vehicles on traffic flow stability and throughput, Transportation Research Part C: Emerging Technologies, 71, 143–163, 2016. https://doi.org/10.1016/j.trc.2016.07.007