A receding horizon control for multi-robot navigation under LiDAR-driven graph updates

Antonello Venturino, Francesco Tedesco, Alessandro Casavola and Giuseppe Franzè

***Abstract*—In the context of Industry 5.0, where environments are subject to unpredictable changes, achieving real-time adaptability and robust collision avoidance is essential for safe navigation and timely task execution. This paper introduces a Robust Grid-Based Receding Horizon Control scheme tailored for multi-robot logistics, designed to address dynamic obstacles and bounded multiplicative disturbances in unicycle-type robots. The method employs a single grid graph, continuously updated in real time using LiDAR data. A distributed robust set-theoretic model predictive control strategy leverages the grid graph to ensure safe and efficient navigation. The proposed approach is validated through realistic simulations in ROS/Gazebo, demonstrating its effectiveness in complex and dynamic scenarios.**

I. INTRODUCTION

Autonomous Mobile Robots (AMRs) are increasingly deployed in industrial logistics to facilitate efficient, resilient, and safe operations, particularly under the emerging Industry 5.0 paradigm. Their ability to navigate autonomously in cluttered, dynamic factory floors can significantly reduce transportation costs and human workload [1]. Recent advances emphasize multi-robot coordination, requiring reliable real-time mapping, collision avoidance, and robust control.

Most multi-robot navigation frameworks rely on either (i) static or semi-static maps that quickly become inaccurate when obstacles move or new ones appear [2], [3], or (ii) per-robot local mapping approaches, which can yield overlapping sub-maps when multiple robots share partially intersecting regions [4]. Planning and navigation in real time is further complicated by uncertain motion models: nonholonomic constraints, such as unicycle or differential-drive kinematics, and disturbance effects (wheel slippage, sensor drift, etc.) can degrade tracking unless suitably accounted for [5]–[7].

A common approach to robustly address these issues is to adopt distributed Model Predictive Control (MPC) strategies, which allow to handle state and input bounds [8], [9]. Recent work exploits set-theoretic methods to guarantee collision avoidance and reference tracking even under bounded disturbances [9]. However, robust MPC strategies require to solve optimization problems online and in order to be able to run in real time, usually the obstacle scenario surrounding the robot is convexified [10]–[13]. This further operation can be unfeasible in real-time high-dynamic environment.

In this paper, a Robust Grid-based Receding Horizon Controller (RGb-RHC) is presented for multi-robot fleets that improves upon previous approaches in two principal ways. First of all, bounded multiplicative noise is explicitly incorporated in the unicycle robot model, reflecting realistic underactuation, wheel slippage, or sensor uncertainties. The proposed controller combines feedback-linearization and set-theoretic methods to handle these disturbances, ensuring robust path tracking. Moreover, instead of maintaining multiple overlapping local maps, as in [4], a single grid graph is adopted and it is updated in real time based on each robot's LiDAR rays. This approach eliminates the need to reconcile sub-maps from different robots, and preserves lightweight updates of the robots' surroundings. As a result, the method attains faster reactivity to moving obstacles and naturally scales with more AMRs contributing LiDAR data.

II. NOTATION AND DEFINITIONS

The symbol $\mathbf{I}_n$ denotes the $n \times n$ identity matrix, $\mathbf{0}_{n\times m}$ an $n \times m$ zero matrix, while $\mathbf{1}_{n\times m}$ an $n \times m$ all-ones matrix. Given the matrix $\mathbf{H} \in \mathbb{R}^{n\times m}$, h_{ij} is its generic entry and $\mathbf{H}^\top$ its transpose matrix. The element-wise multiplication between two vectors $\mathbf{z}_1$ and $\mathbf{z}_2$ is denoted as $\mathbf{z}_1 \odot \mathbf{z}_2$. The symbol $\|\mathbf{z}\| := \sqrt{\mathbf{z}^\top \mathbf{z}}$ is the Euclidean norm of the vector $\mathbf{z}$. The disk of radius ρ, centered at $\mathbf{p}$, is defined as $\mathcal{B}(\mathbf{p}, \rho) := \{\mathbf{z} \in \mathbb{R}^2 : \|\mathbf{z} - \mathbf{p}\| \leq \rho\}$. The symbol $\lfloor \cdot \rfloor$ denotes the floor function.
Consider a discrete-time linear time-invariant system:

$$\mathbf{z}(k+1) = \mathbf{A}\mathbf{z}(k) + \mathbf{B}\mathbf{q}(k) + \mathbf{B}_d\mathbf{d}(k), \qquad (1)$$

where $k \in \mathbb{Z}_+ := \{0, 1, \ldots\}$, $\mathbf{z}(k) \in \mathbb{R}^{n_z}$, $\mathbf{q}(k) \in \mathbb{R}^{n_q}$, and $\mathbf{d}(k) \in \mathcal{D} \subset \mathbb{R}^{n_z}$ are state, input, and noise vectors, respectively. The following constraints are considered:

$$\mathbf{q}(k) \in \mathcal{U}, \quad k \geq 0, \qquad (2)$$

where $\mathcal{U} \subset \mathbb{R}^{n_q}$ and $\mathcal{D}$ are convex and compact sets.

Definition 1 (One-step robust controllable sets)**.** Given the plant (1), it is possible to compute the sets of states j-step robust controllable [14] to χ^0 via the recursions:

$$\chi^j := \{\mathbf{z} \in \mathcal{X} : \exists \mathbf{q} \in \mathcal{U}, \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{q} + \mathbf{B}_d\mathbf{d} \in \chi^{j-1}, \forall \mathbf{d} \in \mathcal{D}\}. \qquad (3)$$

Definition 2 (Grid graph [15], [16])**.** Let $m, n > 0$. Let $\mathcal{V} := \{\nu_{i,j} = (i, j) \in \mathbb{N}^2 : 1 \leq i \leq m, 1 \leq j \leq n\}$ be the set of vertices forming an $m \times n$ grid. A grid graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is an undirected graph where the edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ consists of

---

Giuseppe Franzè is with DIMEG, Università della Calabria, Via Pietro Bucci, Cubo 44-C, Rende (CS), 87036, ITALY. Antonello Venturino, Francesco Tedesco, Alessandro Casavola are with DIMES, Università della Calabria, Via Pietro Bucci, Cubo 41-C, Rende (CS), 87036, ITALY, {`antonello.venturino`, `francesco.tedesco`, `alessando.casavola`, `giuseppe.franze`}`@unical.it`

This work was supported in part by the research project - ID:PE00000014 "SEcurity and RIghts in the CyberSpace - SERICS" granted by the Italian Ministry of University and Research (MUR), and European Union within the NextGenerationEU program and in part by the research project - ID:20222N4C8E "Resilient and Secure Networked Multivehicle Systems in Adversary Environments" granted by the European Union - Next Generation EU, Mission 4, Component 1, CUP H53D23000410006.

unordered pairs of distinct vertices that are adjacent in the grid, i.e., $(\nu_{i,j}, \nu_{k,l}) \in \mathcal{E}$ if and only if $\nu_{i,j} \neq \nu_{k,l}$ and $\max\{|i - k|, |j - l|\} = 1$. $\square$

Definition 3 (Linear map for ellipsoids)**.** Given the ellipsoid: $\Xi := \left\{\mathbf{z} \in \mathbb{R}^n : (\mathbf{z} - \mathbf{c})^\top \mathbf{Q}^{-1}(\mathbf{z} - \mathbf{c}) \leq 1\right\}$, centered at $\mathbf{c} \in \mathbb{R}^n$ and with $\mathbf{Q} = \mathbf{Q}^\top > 0, \mathbf{Q} \in \mathbb{R}^{n\times n}$. Let $\mathbf{M} \in \mathbb{R}^{n\times n}$ be a non-singular matrix and $\mathbf{Q}' = \mathbf{M}\mathbf{Q}\mathbf{M}^\top$, the linear map $\mathbf{z}' = \mathbf{M}\mathbf{z}$ of the ellipsoid Ξ is defined as:

$$\mathbf{M} \circ \Xi := \left\{\mathbf{z}' \in \mathbb{R}^n : (\mathbf{z}' - \mathbf{M}\mathbf{c})^\top (\mathbf{Q}')^{-1}(\mathbf{z}' - \mathbf{M}\mathbf{c}) \leq 1\right\}.$$

Property 1. Given two disks $\mathcal{S}_1 = \mathcal{B}(\mathbf{c}_1, r_1)$ and $\mathcal{S}_2 = \mathcal{B}(\mathbf{c}_2, r_2)$, their Minkowski sum is given by:

$$\mathcal{S}_1 \oplus \mathcal{S}_2 := \mathcal{B}(\mathbf{c}_1 + \mathbf{c}_2, r_1 + r_2).$$

Whereas, assuming $r_1 > r_2$, their Pontryagin difference is given by:

$$\mathcal{S}_1 \ominus \mathcal{S}_2 := \mathcal{B}(\mathbf{c}_1, r_1 - r_2).$$

III. PROBLEM FORMULATION

In this section, the problem of coordinating a fleet of n_r Autonomous Mobile Robots (AMRs) for logistics operations in dynamic planar industrial environments is formalized.

AMR model – Each robot $i \in \{1, \dots, n_r\}$ is described by the continuous-time unicycle kinematics with multiplicative disturbance:

$$\dot{\mathbf{x}}^i(t) = \begin{bmatrix} \dot{p}_x^i(t) \\ \dot{p}_y^i(t) \\ \dot{\theta}^i(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta^i(t)) & 0 \\ \sin(\theta^i(t)) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \left(1 - d_v^i(t)\right) v^i(t) \\ \left(1 - d_\omega^i(t)\right) \omega^i(t) \end{bmatrix}, \quad (4)$$

where $t \in \mathbb{R}, \mathbf{p}^i(t) = \begin{bmatrix} p_x^i(t) & p_y^i(t) \end{bmatrix}^\top \in \mathbb{R}^2$ are the robot's planar coordinates, $\theta^i(t) \in \mathbb{R}$ is the heading angle, and $\mathbf{u}^i(t) = \begin{bmatrix} v^i(t) & \omega^i(t) \end{bmatrix}^\top \in \mathbb{R}^2$ are the linear and angular velocity inputs, subject to:

$$\begin{aligned} \mathcal{U} &= \left\{\mathbf{u}^i \in \mathbb{R}^2 : \mathbf{H}_u \mathbf{u}^i \leq \mathbf{1}_{4\times 1}\right\}, \\ \mathbf{H}_u &= \begin{bmatrix} -v_{\max}^{-1} & v_{\max}^{-1} & 0 & 0 \\ 0 & 0 & \omega_{\max}^{-1} & -\omega_{\max}^{-1} \end{bmatrix}^\top, \end{aligned} \quad (5)$$

with $v_{\max}, \omega_{\max} \in \mathbb{R}_+$ being the maximum admissible linear and angular velocities, respectively. The multiplicative disturbance terms $d_v^i(t)$ and $d_\omega^i(t)$ capture bounded underactuation or wheel slip, with $|d_v^i(t)| \leq \epsilon_v < 1$ and $|d_\omega^i(t)| \leq \epsilon_\omega < 1$. In the sequel, the discretized version of (4) is considered.

On-board sensors – Each AMR^i uses a 2D Light Detection And Ranging (LiDAR) sensor which provides, at each time k, a set of rays $\{\rho_l^i(k)\}_{l=1}^{n_L}$ with bearings $\{\psi_l\}_{l=1}^{n_L}$ in the robot reference frame. The LiDAR detects objects in the range $\rho_{\min} \leq \rho \leq \rho_{\max}$, it has a 360-degrees field of view and resolution $\delta_\psi \in \mathbb{R}_+$, with $n_L = 360/\delta_\psi$. The l-th LiDAR's reading is transformed into the global reference frame by the following transformation:

$$\mathbf{y}_l^i(k) = \mathbf{p}^i(k) + \rho_l^i(k) \begin{bmatrix} \cos(\theta^i(k)) & -\sin(\theta^i(k)) \\ \sin(\theta^i(k)) & \cos(\theta^i(k)) \end{bmatrix} \begin{bmatrix} \cos(\psi_l) \\ \sin(\psi_l) \end{bmatrix}. \quad (6)$$

It is assumed that the state vector $\mathbf{x}^i(k)$ is fully accessible at each time step k since it possible to leverage well-established indoor localization methods (see, e.g., [17], [18]).

Topology – The fleet operates within a time-varying grid topology, where the set of neighbors $\mathcal{N}^i(k)$ for each robot $i = 1, \dots, n_r$, is defined as:

$$\mathcal{N}^i(k) := \left\{j \neq i : \left\|\mathbf{p}^i(k) - \mathbf{p}^j(k)\right\| < R_C\right\}, \quad (7)$$

where $R_C \in \mathbb{R}_+$ represents the communication radius, specifying the maximum allowable distance for data exchange between robots.

Initial Navigable Graph – Using an *a priori* known 2D layout (e.g., a static floorplan or CAD model), the navigable domain can be discretized into cells of size $R \times R$. Let $\mathbf{p}$ denote a generic planar point. The operator:

$$\mathcal{I}_\nu(\mathbf{p}) := \left\lfloor \mathbf{p} R^{-1} \right\rfloor \quad (8)$$

maps the vector $\mathbf{p}$ to a cell $\nu_{i,j} \in \mathbb{N}^2$. A cell $\nu_{i,j} = \mathcal{I}_\nu(\mathbf{p})$ is considered obstacle-free if no obstacles are present within a radius R from the point $\mathbf{p}$, i.e., the disk $\mathcal{B}(\mathbf{p}, R)$ contains no obstacles. Let $\mathcal{V}_0 \subseteq \mathbb{N}^2$ represent the set of indices $\nu_{i,j}$ for all obstacle-free cells, and $\mathcal{E}_0 \subseteq \mathcal{V}_0 \times \mathcal{V}_0$ define the 8-connected adjacency relations [4]. The resulting initial grid graph is thus given by $\mathcal{G}_0 = (\mathcal{V}_0, \mathcal{E}_0)$.

Logistic tasks – At each time step k, the set of targets $\{\mathbf{g}^i(k)\}_{i=1}^{n_r}$, with $\mathbf{g}^i(k) \in \mathbb{R}^2$, is determined based on criteria aimed at balancing workload and optimizing energy consumption [19], and then communicated to the n_r vehicles.

Path planning – At each time step k, a path $\boldsymbol{\pi}^i(k) = \{\mathbf{w}_0^i \equiv \mathbf{p}^i(k), \dots, \mathbf{w}_s^i, \dots, \mathbf{w}_f^i \equiv \mathbf{g}^i(k)\}$ is computed, i.e., a sequence of way-points that connect the current position $\mathbf{p}^i(k)$ to the target $\mathbf{g}^i(k)$. The path is derived from the grid graph [20].

The problem to solve can be stated as follows.

Problem (point-to-point Navigation and Obstacle aVoidance in dynamic indUStrial environments (NOVUS)). Given an initial grid graph $\mathcal{G}_0$, a fleet of unicycle robots $\{\mathrm{AMR}^i\}_{i=1}^{n_r}$, described as in (4), and the current paths $\{\boldsymbol{\pi}^i(k)\}_{i=1}^{n_r}$, determine a robust state-feedback control laws complying with (5):

$$\mathbf{u}^i(k) = h\left(\mathbf{x}^i(k), \boldsymbol{\pi}^i(k)\right), \quad i = 1, \dots, n_r, \quad (9)$$

such that the regulated state trajectories are *Uniformly Ultimate Bounded* (UUB) [14] and there exist a finite time step $\bar{k} > 0$ and a constant $C \in \mathbb{R}_+$ such that:

$$\left\|\mathbf{p}^i(k) - \mathbf{g}^i(k)\right\| \leq C, \quad \forall k > \bar{k}, \quad (10)$$

$$\mathbf{p}^i(k) \notin \mathcal{B}(\mathbf{p}^j(k), 0.5R), \forall j \in \mathcal{N}^i(k), \forall k, \quad \text{Anti-collision,} \quad (11)$$

$$\mathcal{I}_p\left(\mathcal{B}\left(\mathbf{p}^i(k), R\right)\right) \in \mathcal{V}(k), \forall k, \quad \text{Obstacle avoidance.} \quad (12)$$

IV. REAL-TIME GRAPH UPDATING VIA LIDAR MEASUREMENTS

During online operations, each robot i acquires LiDAR ranges $\{\rho_\ell^i(k)\}_{\ell=1}^{n_L}$ at bearings $\{\psi_\ell\}_{\ell=1}^{n_L}$. These ranges are used to: (i) add any newly discovered free cells and (ii) remove any cells observed to contain obstacles to the current vertex set $\mathcal{V}(k)$, thereby to the edges in $\mathcal{E}(k)$ as well.

In the sequel, a matrix-based structure is adopted to represent the grid graph $\mathcal{G}(k)$, rather than storing explicit vertex $\mathcal{V}(k)$ and edge $\mathcal{E}(k)$ sets. Specifically, let $\mathbf{M}(k) \in \{0,1\}^{n\times m}$

a matrix where each entry $\mathbf{M}_{\nu_{i,j}}(k)$ indicates whether the cell $\nu_{i,j}$ in the discretized plane is a valid vertex. If $\mathbf{M}_{\nu_{i,j}}(k) = 1$, the vertex $\nu_{i,j}$ exists, i.e., it is obstacle-free; otherwise, it is absent, i.e., there is an obstacle. By definition, two vertices $\nu_{i,j}$ and $\nu_{k,l}$ are considered adjacent if:

1) $\mathbf{M}_{\nu_{i,j}}(k) = \mathbf{M}_{\nu_{k,l}}(k) = 1$ (i.e., both vertices exist), and
2) $\max(|i-k|, |j-l|) = 1$ (i.e., they lie in neighboring cells).

The matrix-based structure of the grid graph introduces some benefits. In particular, adding or removing vertices operations are simply set operations, i.e., $\mathbf{M}_{\nu_{i,j}}(k) \leftarrow 1$ and $\mathbf{M}_{\nu_{i,j}}(k) \leftarrow 0$, respectively. Moreover, there is no need to store an explicit set of edges $\mathcal{E}$. Instead, adjacency is determined "on the fly" by checking whether any of the eight neighboring entries $\mathbf{M}_{\nu_{i\pm1,\,j\pm1}}(k)$ are set to 1. This eliminates the need to maintain or reconstruct adjacency lists after each update. The grid graph enjoys the following property.

Property 2. For any two adjacent vertices $\nu_{i,j}$ and $\nu_{k,l}$, the distance between them is $\sqrt{2}R$ if they are diagonally adjacent, and R otherwise.

A. Grid graph update

The grid graph $\mathbf{M}(k)$ is updated, at each time step k, according to the following lines. For each ray l of AMR^i, define the continuous line-of-sight segment as:

$$\mathcal{L}_l^i(k) = \left\{\mathbf{p}^i(k) + \alpha\left(\mathbf{y}_l^i(k) - \mathbf{p}^i(k)\right) : 0 \le \alpha \le 1\right\}.$$

This continuous segment is converted into a set of grid vertices via a function $\Gamma(\mathcal{L}_l^i(k)) \subseteq \mathbb{N}^2$ which collects exactly those discrete cells $\nu_{u,v} = \mathcal{I}_\nu(\mathbf{p}_\alpha)$ whose corresponding point $\mathbf{p}_\alpha$ intersects the segment $\mathcal{L}_l^i(k)$:

$$\Gamma(\mathcal{L}_l^i(k)) = \left\{\nu_{u,v} \in \mathbb{N}^2 : \nu_{u,v} = \mathcal{I}_\nu(\mathbf{p}), \mathbf{p} \in \mathcal{L}_l^i(k)\right\}. \tag{13}$$

Equivalently, Γ acts like a Bresenham line-drawing algorithm in the $\mathbb{N}^2$ index space [21].

Let $\mathcal{V}_{\mathbf{y}_l^i}(k)$ be a set of cells surrounding the detected obstacle at measurement $\mathbf{y}_l^i(k)$, defined as:

$$\mathcal{V}_{\mathbf{y}_l^i}(k) = \begin{cases} \mathcal{B}(\mathcal{I}_\nu(\mathbf{y}_l^i(k)), R_F), & \text{if } \rho_l^i(k) < \infty, \\ \emptyset, & \text{otherwise}, \end{cases} \tag{14}$$

where $R_F > 0$ is a given *inflation* radius. Moreover, in order to distinguish obstacles from other AMRs, the following neighborhood cell set is defined:

$$\begin{aligned}\mathcal{A}_{\mathcal{N}}^i(k) = \big\{\nu \in \mathbb{N}^2 &: \nu = \mathcal{I}_\nu(\mathcal{B}(\mathbf{p}^j(k), R)), j \in \mathcal{N}^i(k), \\ &\left\|\mathbf{p}^i(k) - \mathbf{p}^j(k)\right\| \le \rho_{\max}\big\}.\end{aligned} \tag{15}$$

Hence, the set containing only obstacles cells can be determined by:

$$\mathcal{R}^i(k) = \left(\bigcup_{l=1}^{n_L} \mathcal{V}_{\mathbf{y}_l^i}(k)\right) \setminus \mathcal{A}_{\mathcal{N}}^i(k), \tag{16}$$

while the set of free cells is:

$$\mathcal{A}^i(k) = \left(\bigcup_{l=1}^{n_L} \Gamma\left(\mathcal{L}_l^i(k)\right)\right) \setminus \mathcal{R}^i(k). \tag{17}$$

Finally, the grid graph $\mathbf{M}(k)$ is updated as:

$$\mathbf{M}_{\nu_{i,j}}(k) = \begin{cases} 1, & \forall\, \nu_{i,j} \in \mathcal{A}^i(k), \\ 0, & \forall\, \nu_{i,j} \in \mathcal{R}^i(k). \end{cases} \tag{18}$$

In other words, cells newly detected free are added to the graph, while cells associated with obstacles or within the inflated zone are removed from the graph.

Remark 1. The update in (18) ensures that the grid graph $\mathbf{M}(k)$ dynamically reflects the current free-space geometry based on real-time data from the robots' LiDAR beams. This approach eliminates the need to build or merge local sub-maps.

Because new obstacles might emerge in previously safe regions, the set of vertices can shrink over time, temporarily removing known paths. Any global path planning procedure (e.g., A*, Dijkstra, or conflict-based search) can then run on $\mathbf{M}(k)$ at fixed intervals to compute a way-point sequence $\boldsymbol{\pi}^i(k) = \{\mathbf{w}_0^i \equiv \mathbf{p}^i(k), \ldots, \mathbf{w}_s^i, \ldots, \mathbf{w}_f^i \equiv \mathbf{g}^i(k)\}$. Then, a local planner selects the current way-point $\mathbf{w}_{\text{curr}}^i(k) \in \boldsymbol{\pi}^i(k)$, which the associated grid cell is guaranteed to be feasible and safely distanced from obstacles, see [4, Prop. 1].

V. ROBUST GRID-BASED RECEDING HORIZON CONTROLLER

In this section, the proposed Robust Grid-Based Receding Horizon Controller (RGb-RHC) is proposed. It drives each AMR^i to follow a desired way-point sequence $\boldsymbol{\pi}^i(k)$ extracted from the grid graph $\mathbf{M}(k)$.

A common technique for feedback linearization of a unicycle is to introduce a *virtual point* on the robot's main axis that is offset from the center by a constant distance $b > 0$, see [22]. Let

$$\mathbf{z}^i(t) = \begin{bmatrix} p_x^i(t) \\ p_y^i(t) \end{bmatrix} + b \begin{bmatrix} \cos\left(\theta^i(t)\right) \\ \sin\left(\theta^i(t)\right) \end{bmatrix}. \tag{19}$$

Thus the origin of the coordinate frame is shifted along the direction of travel. To understand the dynamics of $\mathbf{z}^i(t)$, we take its time derivative. From (19), we get

$$\dot{\mathbf{z}}^i(t) = \begin{bmatrix} \dot{p}_x^i(t) \\ \dot{p}_y^i(t) \end{bmatrix} + b \begin{bmatrix} -\sin\left(\theta^i(t)\right)\dot{\theta}^i(t) \\ \cos\left(\theta^i(t)\right)\dot{\theta}^i(t) \end{bmatrix}. \tag{20}$$

Recalling the unicycle equations (4), we substitute $\dot{p}_x^i(t)$, $\dot{p}_y^i(t)$, and $\dot{\theta}^i(t)$ into (20):

$$\dot{\mathbf{z}}^i(t) = \underbrace{\begin{bmatrix} \cos\left(\theta^i(t)\right) & -b\sin\left(\theta^i(t)\right) \\ \sin\left(\theta^i(t)\right) & b\cos\left(\theta^i(t)\right) \end{bmatrix}}_{\mathbf{J}(\theta^i(t))} \begin{bmatrix} \left(1 - d_v^i(t)\right)v^i(t) \\ \left(1 - d_\omega^i(t)\right)\omega^i(t) \end{bmatrix},$$

where $\mathbf{J}\left(\theta^i(t)\right)$ is the *geometric Jacobian* and it is always invertible for $b \neq 0$. Moreover, we can rewrite

$$\begin{bmatrix} \left(1 - d_v^i(t)\right)v^i(t) \\ \left(1 - d_\omega^i(t)\right)\omega^i(t) \end{bmatrix} = \begin{bmatrix} 1 - d_v^i(t) & 0 \\ 0 & 1 - d_\omega^i(t) \end{bmatrix} \mathbf{u}^i(t).$$

Hence

$$\dot{\mathbf{z}}^i(t) = \mathbf{J}\left(\theta^i(t)\right) \begin{bmatrix} 1 - d_v^i(t) & 0 \\ 0 & 1 - d_\omega^i(t) \end{bmatrix} \mathbf{u}^i(t). \tag{21}$$

Then, notice that

$$\begin{bmatrix} 1 - d_v^i(t) & 0 \\ 0 & 1 - d_\omega^i(t) \end{bmatrix} \mathbf{u}^i(t) = \mathbf{u}^i(t) - \mathbf{d}^i(t) \odot \mathbf{u}^i(t). \tag{22}$$

As a result, by combining (21) and (22) we get

$$\begin{aligned}\dot{\mathbf{z}}^i(t) &= \mathbf{J}\left(\theta^i(t)\right)\left(\mathbf{u}^i(t) - \mathbf{d}^i(t)\odot\mathbf{u}^i(t)\right) &(23)\\ &= \underbrace{\mathbf{J}\left(\theta^i(t)\right)\mathbf{u}^i(t)}_{\mathbf{q}^i(t)\ \text{nominal}} - \underbrace{\mathbf{J}\left(\theta^i(t)\right)\left(\mathbf{d}^i(t)\odot\mathbf{u}^i(t)\right)}_{\mathbf{d}_z^i(t)}. &(24)\end{aligned}$$

Consequently

$$\dot{\mathbf{z}}^i(t) = \mathbf{q}^i(t) - \mathbf{d}_z^i(t). \tag{25}$$

At this point, observe that $\mathbf{J}\left(\theta^i(t)\right)$ has the inverse

$$\mathbf{J}^{-1}\left(\theta^i(t)\right) = \begin{bmatrix}\cos\left(\theta^i(t)\right) & \sin\left(\theta^i(t)\right)\\ -\frac{\sin\left(\theta^i(t)\right)}{b} & \frac{\cos\left(\theta^i(t)\right)}{b}\end{bmatrix}. \tag{26}$$

Hence, one can consider the original control $\mathbf{u}^i(t)$ as

$$\mathbf{u}^i(t) = \mathbf{J}^{-1}\left(\theta^i(t)\right)\mathbf{q}^i(t). \tag{27}$$

For completeness, we also wish to express $\dot{\theta}^i(t)$ in terms of $\mathbf{z}^i(t)$. From the unicycle model (4), we have:

$$\dot{\theta}^i(t) = \left(1 - d_\omega^i(t)\right)\omega^i(t).$$

Using again the relationship (27)

$$\omega^i(t) = \left[\mathbf{J}^{-1}\left(\theta^i(t)\right)\mathbf{q}^i(t)\right]_2,$$

i.e. the second component of $\mathbf{J}^{-1}\left(\theta^i(t)\right)\mathbf{q}^i(t)$. Equivalently, one has

$$\dot{\theta}^i(t) = \begin{bmatrix}-\frac{\sin\left(\theta^i(t)\right)}{b} & \frac{\cos\left(\theta^i(t)\right)}{b}\end{bmatrix}\dot{\mathbf{z}}^i(t), \tag{28}$$

since the geometry of $\mathbf{J}(\theta^i)$ implies that projecting $\dot{\mathbf{z}}^i(t)$ onto the robot's angular direction yields precisely $\omega^i(t)$, up to the factor $1 - d_\omega^i(t)$. Putting these pieces together, we obtain the equivalent representation of the unicycle model (4) given by (25) and (28). Notice that in the ideal (undisturbed) scenario, i.e. $d_v^i(t) = 0$ and $d_\omega^i(t) = 0$, then $\dot{\mathbf{z}}^i(t) = \mathbf{q}^i(t)$. This shows the classical result that, under nominal conditions, we can make $\mathbf{z}^i(t)$ behave like a simple integrator in two dimensions driven by $\mathbf{q}^i(t)$. Therefore, a feedback controller for $\mathbf{q}^i(t)$ will make the point $\mathbf{z}^i(t)$ track any reference trajectory, even with discontinuous tangents to the path [22]. Conversely, when $d_v^i(t)$ and $d_\omega^i(t)$ are nonzero, $\mathbf{z}^i(t)$ behaves like a single integrator driven by $\mathbf{q}^i(t)$, with the disturbance $-\mathbf{d}_z^i(t)$. In these transformed coordinates, the multiplicative noise is bounded within the following set:

$$\begin{aligned}\mathcal{D}_z := \big\{\mathbf{d}_z^i \in \mathbb{R}^2 : \exists \mathbf{u} \in \mathcal{U}, \theta \in [0, 2\pi), \mathbf{d} \in \mathcal{D},\\ \text{s.t. } \mathbf{d}_z^i = \mathbf{J}(\theta)\left(\mathbf{d}\odot\mathbf{u}\right)\big\}.\end{aligned} \tag{29}$$

The set $\mathcal{D}_z$ can be outer-approximated by:

$$\bar{\mathcal{D}} = \left\{\mathbf{d}_z \in \mathbb{R}^2 : \mathbf{d}_z^\top\mathbf{d}_z \le r_d^2\right\}, r_d = \frac{\epsilon_{v,\max} + b\epsilon_{\omega,\max}}{\sqrt{2}}, \tag{30}$$

since $|d_v v| \le \epsilon_v v_{\max} = \epsilon_{v,\max}$ and $|d_\omega \omega| \le \epsilon_\omega \omega_{\max} = \epsilon_{\omega,\max}$. Finally, the continuous-time model is discretized using a forward Euler method with a sampling time T_s, resulting in the following system matrices: $\mathbf{A} = \mathbf{I}_2, \mathbf{B} = \mathbf{B}_d = T_s\mathbf{I}_2$. A limitation of this approach lies in the emergence of state-dependent input constraints:

$$\mathcal{U}(\theta) = \left\{\mathbf{q} \in \mathbb{R}^2 : \mathbf{H}(\theta)\mathbf{q} \le \mathbf{1}_{4\times 1}\right\}, \mathbf{H}(\theta) = \mathbf{H}_u\mathbf{J}^{-1}(\theta). \tag{31}$$

This computational challenge can be addressed by applying the worst-case approach proposed in [9]. Specifically, the input constraint set $\underline{\mathcal{U}} := \bigcap_{\theta=0}^{2\pi}\mathcal{U}(\theta)$ replaces (31), and is given by:

$$\underline{\mathcal{U}} := \left\{\mathbf{q} \in \mathbb{R}^2 : \mathbf{q}^\top\mathbf{q} \le r_q^2\right\}, r_q = \frac{b v_{\max}\omega_{\max}}{\sqrt{b^2\omega_{\max}^2 + v_{\max}^2}}. \tag{32}$$

Based on the grid based planar representation $\mathbf{M}(k)$, the admissible travel directions θ for each robot are defined by the set $\Theta := \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\}$, corresponding to movements in the *horizontal*, *45-degree diagonal*, *vertical*, and *135-degree diagonal* directions. Giving $\eta \in \mathbb{Z}_+$, the robot's position $\mathbf{p}$ can be expressed as a function of the distance $r \in \mathbb{R}_+$ traveled along each direction, as follows:

$$\mathbf{p}(r,\theta) := \begin{cases}\begin{bmatrix}r & 0\end{bmatrix}^\top, & \theta = \eta\pi,\\ \begin{bmatrix}r & r\end{bmatrix}^\top, & \theta = \left(\eta + \frac{1}{4}\right)\pi,\\ \begin{bmatrix}0 & r\end{bmatrix}^\top, & \theta = \left(\eta + \frac{1}{2}\right)\pi,\\ \begin{bmatrix}-r & r\end{bmatrix}^\top, & \theta = \left(\eta + \frac{3}{4}\right)\pi.\end{cases} \tag{33}$$

The set-theoretic RHC controller introduced in [23] is adapted here to align with the grid-based framework outlined in the preceding sections. In particular, the offline phase involves calculating a terminal region along with the corresponding stabilizing state feedback law:

$$\mathbf{q}_\theta(k) = \mathbf{K}_\theta\mathbf{z}(k), \tag{34}$$

for each $\theta \in \Theta$. This is achieved by solving the following semi-definite programming problem:

$$\begin{aligned}&[\mathbf{Q}^*, \mathbf{Y}^*] = \arg\min_{\gamma,\mathbf{Q},\mathbf{Y}} \gamma &(35a)\\ &\text{s.t.}\quad \mathbf{Q} = \mathbf{Q}^\top \succ 0 &(35b)\\ &\begin{bmatrix}1 & \mathbf{p}^\top(r_d,\theta)\\ \mathbf{p}(r_d,\theta) & \mathbf{Q}\end{bmatrix} \succeq 0 &(35c)\\ &\begin{bmatrix}\mathbf{Q} & * & * & *\\ \mathbf{AQ}+\mathbf{BY} & \mathbf{Q} & * & *\\ \mathbf{Q}_q^{\frac{1}{2}}\mathbf{Q} & \mathbf{O}_2 & \gamma\mathbf{I}_2 & *\\ \mathbf{Q}_z^{\frac{1}{2}}\mathbf{Y} & \mathbf{O}_2 & \mathbf{O}_2 & \gamma\mathbf{I}_2\end{bmatrix} \succeq 0 &(35d)\\ &\begin{bmatrix}r_q^2\mathbf{I}_2 & \mathbf{Y}\\ \mathbf{Y}^\top & \mathbf{Q}\end{bmatrix} \succeq 0 &(35e)\end{aligned}$$

where $\mathbf{Q}_q$ and $\mathbf{Q}_z$ are state and input weighting matrices, respectively. Then one obtains:

$$\mathbf{K}_\theta = \mathbf{Y}^*\left(\mathbf{Q}^*\right)^{-1}, \tag{36}$$

$$\chi_\theta^0 = \left\{\mathbf{z} \in \mathbb{R}^2 : \mathbf{z}^\top\mathbf{z} \le r_0^2\right\}, \tag{37}$$

with $r_0^2 = \bar{\sigma}(\mathbf{Q}^*)$ being the maximum singular value of $\mathbf{Q}^*$. The sequence of one-step robust controllable sets $\{\chi_\theta^j\}_{j=0}^{j_n}$, defined under the set-membership condition $\mathbf{p}(R,\theta) \in \chi^{j_n}, \forall\theta \in \Theta$, is derived based on the following result.

Proposition 1. *Consider the system* (1) *with* $\mathbf{A} = \mathbf{I}_2$, $\mathbf{B} = T_s\mathbf{I}_2$*, and* $\mathbf{B}_d = T_s\mathbf{I}_2$*, where the input and disturbance are bounded by the sets defined in* (32) *and* (30)*, respectively. Given the terminal region* (37)*, the sequence of one-step robust controllable sets* $\{\chi_\theta^j\}_{j=0}^{j_n}$ *can be computed as follows:*

$$\chi_\theta^j = \left\{\mathbf{z} \in \mathbb{R}^2 : \mathbf{z}^\top\mathbf{Q}_j^{-1}\mathbf{z} \le 1, \mathbf{Q}_j = r_{j_\theta}^2\mathbf{I}_2\right\}, \tag{38}$$

$$r_{j_\theta} = r_{j_\theta - 1} - T_s r_d + T_s r_q. \tag{39}$$

Proof. Let χ_θ^j denote the generic j-th one-step robust controllable set, defined as:

$$\begin{aligned}\chi_\theta^j &= \left\{\mathbf{z} \in \mathbb{R}^2 : \exists \mathbf{q} \in \underline{\mathcal{U}} \text{ s.t. } \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{q} + \mathbf{B}_d\mathbf{d} \in \chi_\theta^{j-1}, \forall\, \mathbf{d} \in \bar{\mathcal{D}}\right\}\\ &= \mathbf{A}^{-1} \circ \left[\left(\chi_\theta^{j-1} \ominus \mathbf{B}_d \circ \bar{\mathcal{D}}\right) \oplus (-\mathbf{B}) \circ \underline{\mathcal{U}}\right]. \end{aligned} \tag{40}$$

By exploiting $\mathbf{A} = \mathbf{I}_2$, $\mathbf{B} = T_s\mathbf{I}_2$, and $\mathbf{B}_d = T_s\mathbf{I}_2$ the last l.h.s. of (40) can be rewritten as:

$$\chi_\theta^j = \left(\chi_\theta^{j-1} \ominus (T_s\mathbf{I}_2) \circ \bar{\mathcal{D}}\right) \oplus (-T_s\mathbf{I}_2) \circ \underline{\mathcal{U}}, \tag{41}$$

where $(-T_s\mathbf{I}_2) \circ \underline{\mathcal{U}}$ is a disk with radius $T_s r_q$ centered at the origin. Similarly, for $(T_s\mathbf{I}_2) \circ \bar{\mathcal{D}}$. Hence, by applying Property 1, it follows that $r_{j_\theta} = r_{j_\theta - 1} - T_s r_d + T_s r_q$. □

As proven in [4, Prop. 2], the sequence $\{\chi_\theta^j\}_{j=0}^{j_n}, \forall\theta \in \Theta$ enjoys translation invariance. Hence, during the on-line operations, given the difference between previous and current waypoints, i.e., $\Delta\mathbf{w}^i(k) = \mathbf{w}^i_{\text{curr}}(k) - \mathbf{w}^i_{\text{prev}}(k)$, the direction of travel can be computed by:

$$\theta_g^i(k) = \tan^{-1}\left(\Delta w_y^i(k)\left(\Delta w_x^i(k)\right)^{-1}\right), \tag{42}$$

while, the current set-membership index $j_\theta^i(k)$ is determined as follows:

$$j_\theta^i(k) = \arg\min_{j_\theta}\left\{j_\theta : \left(\mathbf{p}^i(k) - \mathbf{w}^i_{\text{curr}}(k)\right) \in \chi_\theta^j\right\}. \tag{43}$$

Then, the control action

$$\mathbf{u}^i(k) = \mathbf{J}^{-1}(\theta^i(t))\mathbf{q}^*(k) \tag{44}$$

is obtained by solving the following optimization:

$$\mathbf{q}^*(k) = \arg\min_{\gamma,\mathbf{q}} \gamma \tag{45a}$$

$$\text{s.t.} \quad 0 < \gamma \leq 1 \tag{45b}$$

$$\mathbf{H}(\theta^i(t))\mathbf{q} \leq \mathbf{1}_{2\times 1} \tag{45c}$$

$$\begin{bmatrix} \gamma & * \\ \mathbf{A}\mathbf{p}^i(t) + \mathbf{B}\mathbf{q} - \mathbf{w}^i_{\text{curr}}(k) & r^2_{j_\theta^i(k)-1}\mathbf{I}_2 \end{bmatrix} \succeq 0 \tag{45d}$$

Consequently, the next proposition outlines the key properties of the above developments.

Proposition 2. *Let the one-step robust controllable sequence* $\{\chi_\theta^j\}_{j=0}^{j_n}$ *be non-empty with* $\mathbf{p}(R,\theta) \in \chi_\theta^{j_n}, \forall\theta \in \Theta$. *Given the path* $\boldsymbol{\pi}^i(k)$, *the proposed RGb-RHC algorithm ensures that each AMR*^{i} *reaches a bounded distance from the target* $\mathbf{g}^i(k) \in \boldsymbol{\pi}^i(k)$ *while satisfying the Problem* ***NOVUS*** *requirements and the constraints prescribed by* (5).

Proof. First, anti-collision and obstacle avoidance requirements are locally guaranteed in virtue of [4, Prop. 2]. On the other hand, the sequence $\{\chi_\theta^j\}_{j=0}^{j_n}$ complies with $\mathbf{p}(R,\theta) \in \chi_\theta^{j_n}, \forall\theta \in \Theta$, therefore it is ensured that (43) is well-defined. The recursive feasibility is guaranteed by Property 2. In fact, the sequence of one-step robust controllable sets $\{\chi_\theta^j\}_{j=0}^{j_n}$ ensures that at each time step k at least an admissible solution for the optimization problem (45) always exists. Moreover, once the local planner switches to the next waypoint $\mathbf{w}^i_{\text{curr}}(k + j_n)$, in at most j_n time steps, the feasibility is again ensured because $\left\|\mathbf{w}^i_{\text{curr}}(k) - \mathbf{w}^i_{\text{curr}}(k + j_n)\right\| \leq \left\|\mathbf{p}(R, \theta_g^i(k + j_n))\right\| \leq \sqrt{2}R$. The same applies, when at a certain time step $\bar{k}$, the AMR^i's position $\mathbf{p}^i(k)$ is close to the last way-point $\mathbf{w}^i_{\text{curr}}(\bar{k}) = \mathbf{g}^i(k) \in \boldsymbol{\pi}^i(k)$, guarantying that $\left\|\mathbf{g}^i(k) - \mathbf{p}^i(k)\right\| \leq \sqrt{2}R, \forall k > \bar{k}$. □

VI. SIMULATIONS

This section presents a simulation-based evaluation of the proposed RGb-RHC for multi-robot logistics tasks. The setup is realized in a ROS 1 Noetic environment coupled with the Gazebo simulator which code is available at https://doi.org/10.5281/zenodo.14217678. A fleet of three AMRs operates in a 3D-reconstructed environment of a small-to-medium enterprise based in Calabria, Italy, which manufactures building components from galvanized and stainless steel coils. Each robot has maximum linear speed $v_{\max} = 0.26\,\mathrm{m\,s^{-1}}$ and maximum angular speed $\omega_{\max} = 1.82\,\mathrm{rad\,s^{-1}}$. The LiDAR sensor simulates 360° coverage with $n_L = 360$ beams, each with $\rho_{\min} = 0.12\,\mathrm{m}$ and $\rho_{\max} = 3.5\,\mathrm{m}$. A static 2D floor map of size $32.5 \times 32.5\,\mathrm{m}$ is discretized at resolution $R = 5\,\mathrm{cm}$. The disturbances bounds have been estimated to be $\epsilon_v \approx 0.1$ and $\epsilon_\omega \approx 0.25$. Moreover, $R_F = 25\,\mathrm{cm}$ and initials poses of the AMRs: $\mathbf{x}^1(0) = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^\top$, $\mathbf{x}^2(0) = \begin{bmatrix}0 & 1 & 0\end{bmatrix}^\top$, and $\mathbf{x}^3(0) = \begin{bmatrix}0 & 2 & 0\end{bmatrix}^\top$.

The simulation comprises a Monte-Carlo experiment with 100 trials, each lasting 300 sec. At each time step, a scheduler assigns target positions $\mathbf{g}^i(k)$ to each robot, selecting them from the predefined zones dedicated to specific tasks within the industrial environment: $(A) = \begin{bmatrix}-1.95 & -9.5\end{bmatrix}^\top$, $(B) = \begin{bmatrix}-8 & 8\end{bmatrix}^\top$, and $(C) = \begin{bmatrix}-4.5 & 11\end{bmatrix}^\top$.

Fig. 1 provides a comprehensive overview of the experiment. Notably, the three AMRs successfully navigate the time-varying obstacle scenario, avoiding collisions with obstacles across all trials.

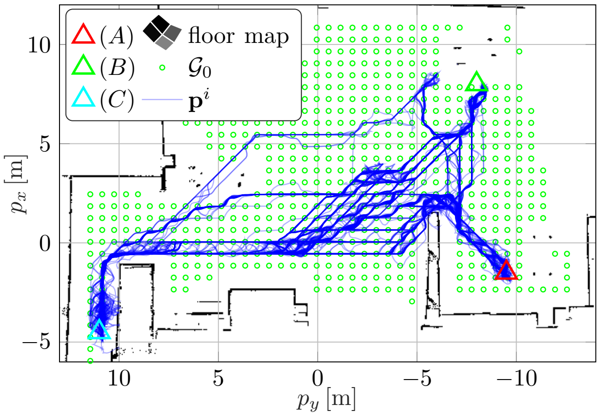

Fig. 1. Experiment overview: *a priori* floor map (black squares); initial graph $\mathcal{G}_0$ (green circles); AMRs state trajectories (continuous lines); logistics zones (A), (B), and (C) (triangles).

Furthermore, Fig. 2 illustrates the planar distances $D^{i,j}(k) = \left\|\mathbf{p}^i(k) - \mathbf{p}^j(k)\right\|$ between the three robots. At each time step k, these distances comply with the constraints defined in (11), demonstrating the effectiveness of the proposed strategy in preventing collisions among the vehicles.

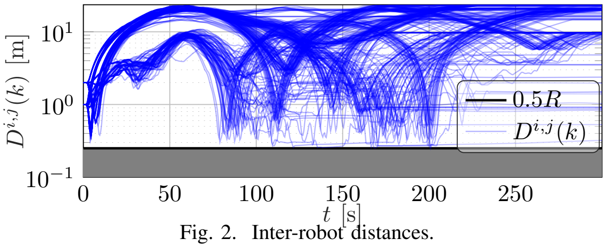

Fig. 2. Inter-robot distances.

Finally, for sake of clarity, Fig. 3 shows only the command inputs $v^1(k)$ and $\omega^1(k)$ computed by the proposed RGb-RHC for a single trial of AMR^1, alongside the corresponding actual velocities $\bar{v}^1(k)$ and $\bar{\omega}^1(k)$. Notably, the strategy adapts online to fully exploit the maximum allowable limits in (5), even though the one-step controllable sets are computed offline using the conservative constraints in (32). Moreover, the presence of multiplicative disturbances does not compromise the ability of the proposed RGb-RHC to reject perturbations once the robot settles at its cruising velocity, even though brief overshoots may occur when velocity commands change abruptly.

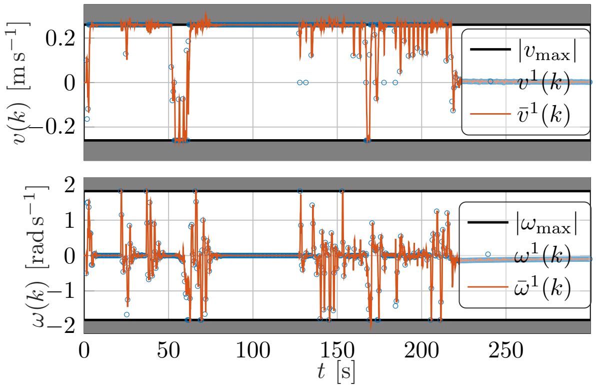

Fig. 3. Command inputs $v^1(k)$ and $\omega^1(k)$ (blue dots) and actual velocities $\bar{v}^1(k)$ and $\bar{\omega}^1(k)$ (orange dotted lines). Black solid lines account for the bounds $|v_{\max}|$ and $|\omega_{\max}|$, respectively.

VII. CONCLUSIONS

In this work, we proposed a novel Robust Grid-Based Receding Horizon Controller tailored for multi-robot logistics under Industry 5.0 requirements. The main novelties can be summarized as follows: 1) the introduction of bounded multiplicative disturbances is formally addressed to ensure satisfactorily tracking capabilities even in the presence of wheel slippage or underactuation; 2) the grid graph is only updated using information provided by the LiDAR rays in a distributed fashion. Simulation results using ROS Noetic and Gazebo demonstrate that the proposed controller allows each robot to accurately track assigned way-points, avoid collisions, and respond swiftly to changes in obstacle configurations.

REFERENCES

[1] A. Gunasekaran and E. W. T. Ngai, "Expert systems and artificial intelligence in the 21st century logistics and supply chain management," *Expert Systems With Applications*, vol. 41, no. 1, pp. 1–4, 2014.

[2] M. Mohanan and A. Salgoankar, "A survey of robotic motion planning in dynamic environments," *Robotics and Autonomous Systems*, vol. 100, pp. 171–185, 2018.

[3] A. Venturino and W. Lucia, "A flexible distributed control strategy for teams of vehicles moving within severe obstacle scenarios," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019, pp. 941–946.

[4] A. Venturino, L. Filice, and G. Franzè, "Grid-based receding horizon control for unicycle robots under logistic operations," in *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2024, pp. 1–7.

[5] S.-H. Chung, "Applications of smart technologies in logistics and transport: A review," *Transportation Research Part E: Logistics and Transportation Review*, vol. 153, p. 102455, 2021.

[6] N. D. Wallace, H. Kong, A. Hill, and S. Sukkarieh, "Receding horizon estimation and control with structured noise blocking for mobile robot slip compensation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1169–1175.

[7] P. D. Christofides, R. Scattolini, D. M. De La Pena, and J. Liu, "Distributed model predictive control: A tutorial review and future research directions," *Computers & Chemical Engineering*, vol. 51, pp. 21–41, 2013.

[8] P. Wang and B. Ding, "Distributed RHC for tracking and formation of nonholonomic multi-vehicle systems," *IEEE Trans. Autom. Control*, vol. 59, no. 6, pp. 1439–1453, 2014.

[9] C. Tiriolo, G. Franzè, and W. Lucia, "A receding horizon trajectory tracking strategy for input-constrained differential-drive robots via feedback linearization," *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 3, pp. 1460–1467, 2023.

[10] M. Aizat, N. Qistina, and W. Rahiman, "A comprehensive review of recent advances in automated guided vehicle technologies: Dynamic obstacle avoidance in complex environment toward autonomous capability," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–25, 2024.

[11] Z. Dong, X. Xu, X. Zhang, X. Zhou, X. Li, and X. Liu, "Real-time motion planning based on MPC with obstacle constraint convexification for autonomous ground vehicles," in *2020 3rd International Conference on Unmanned Systems (ICUS)*, 2020, pp. 1035–1041.

[12] R. Tallamraju, S. Rajappa, M. J. Black, K. Karlapalem, and A. Ahmad, "Decentralized MPC based obstacle avoidance for multi-robot target tracking scenarios," in *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2018, pp. 1–8.

[13] C. Bali and A. Richards, "Robot navigation using convex model predictive control and approximate operating region optimization," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 2171–2176.

[14] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*. Birkhäuser, Boston, 2015.

[15] O. Favaron, G. H. Fricke, D. Pritikin, and J. Puech, "Irredundance and domination in kings graphs," *Discrete Mathematics*, vol. 262, no. 1, pp. 131–147, 2003.

[16] M. X. Punithan and S.-W. Seo, "King's graph-based neighbor-vehicle mapping framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1313–1330, 2013.

[17] A. Venturino, C. Stoica Maniu, S. Bertrand, T. Alamo, and E. F. Camacho, "Multi-vehicle localization by distributed MHE over a sensor network with sporadic measurements: Further developments and experimental results," *Control Engineering Practice*, vol. 132, p. 105410, 2023.

[18] H. Chen, D. Sun, J. Yang, and J. Chen, "Localization for multirobot formations in indoor environment," *IEEE/ASME Trans. Mechatronics*, vol. 15, no. 4, pp. 561–574, 2010.

[19] G. Fragapane, R. de Koster, F. Sgarbossa, and J. O. Strandhagen, "Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda," *European Journal of Operational Research*, vol. 294, no. 2, pp. 405–426, 2021.

[20] A. Venturino, L. Filice, G. Mezzatesta, , F. Tedesco, and G. Franzè, "Coordination of fleets of autonomous vehicles for logistics operations in industrial environments: a grid based receding horizon control approach," *TechRxiv*, December 2024, preprint, doi: 10.36227/techrxiv.173386516.60845320.

[21] P. Koopman, "Bresenham line-drawing algorithm," *Forth Dimensions*, vol. 8, no. 6, pp. 12–16, 1987.

[22] A. De Luca, G. Oriolo, and M. Vendittelli, *Control of Wheeled Mobile Robots: An Experimental Overview*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, ch. 8, pp. 181–226.

[23] D. Angeli, A. Casavola, G. Franzè, and E. Mosca, "An ellipsoidal off-line MPC scheme for uncertain polytopic discrete-time systems," *Automatica*, vol. 44, no. 12, pp. 3113–3119, 2008.