IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 21 February 2023, accepted 14 March 2023, date of publication 22 March 2023, date of current version 30 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3260646

RESEARCH ARTICLE

Toward Safer and Energy Efficient Global Trajectory Planning of Self-Guided Vehicles for Material Handling System in Dynamic Environment

MASSINISSA GRABA[1,2], ALI AMAMOU[1,2], (Member, IEEE), SOUSSO KELOUWANI[1,3], (Senior Member, IEEE), BILEL ALLANI[1,2], LOTFI ZEGHMI[1], KODJO AGBOSSOU[1,2], (Senior Member, IEEE), AND MOHAMMAD MOHAMMADPOUR[1,3]

[1]Institut de Recherche sur l'Hydrogène, Université de Québec à Trois-Rivières, Trois-Rivières, QC G9A 5H7, Canada
[2]Department of Electrical and Computer Engineering, Université de Québec à Trois-Rivières, Trois-Rivières, QC G8Z 4M3, Canada
[3]Department of Mechanical Engineering, Université de Québec à Trois-Rivières, Trois-Rivières, QC G8Z 4M3, Canada

Corresponding author: Massinissa Graba (massinissa.graba@uqtr.ca)

This work was supported in part by the Noovelia Research Chair in Intelligent Navigation of Autonomous Industrial Vehicles and Engineering Research Council of Canada.

ABSTRACT For a sustainable operation of multiple Self-Guided Vehicles (SGVs) in a dynamic manufacturing environment, it is essential to guarantee collision-free and efficient navigation to the autonomous mobile platforms and safety to the surrounding subjects. To prevent from navigation failures, an SGV must avoid conflicts that constrain itself to abruptly brake or stop to avoid collisions. These inefficient conflicts result from unexpected changes in the configuration space or due to nearby unforeseen obstacle. In this paper, a navigation approach is proposed to adapt the global trajectory in order to reduce conflict occurrence while limiting energy consumption of the mobile platform. To generate such trajectory, first the collision risks are characterized using an objective risk perception parameter, the Time-To-Collision TTC, that rely on the kinematics of the egoSGV and the neighboring obstacles. Next, weighted Kernel Density Estimation (wKDE) defines the spatial distribution of conflict severity in configuration space. The defined zones are incorporated as a conflict layer in the global map. Then, a global trajectory planner algorithm is used to weigh between the length cost and conflict cost. Finally, to test the proposed solution, a simulation is performed in a factory-like environment, then an experiment is conducted with a real SGV. In comparison with the state-of-the-art geometrical path planning method, the results show that the proposed approach reduces navigation failures by up to 52%, while reducing the trajectory execution time by around up to 10 %. Also, the smoothness of the executed motion allowed to reduce energy consumption by over 12%.

INDEX TERMS Energy efficiency, global path planning, material handling system, operation safety, self-guided vehicle, time-to-collision.

I. INTRODUCTION

Automated Guided Vehicles (AGVs) have been introduced in industries (factories, warehouses) to improve the intralogistics, but currently mark their presence in public infrastructures such as hospitals, airports, malls…etc. These platforms are used for Material Handling System (MHS) purposes since the mid-twentieth century where their main task is to carry loads to different locations. Nevertheless, these platforms are rigid, and can only operate on a fixed path layout. In addition, this indoor freight solution is less reactive to changes and the paths must be cleared from obstacles, otherwise it will lead to deadlocks [1], [2].

As technologies regarding the autonomous mobile navigation are at their culmination point, extensive works have emerged for autonomous material handling purposes [3].

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Quan.

The Self-Guided Vehicles (SGVs) compared to the classical AGVs, are laser-guided mobile platforms with higher degree of self-guidance that can be effectively deployed in several types of applications [4]. Furthermore, SGV-System is a decentralized paradigm, that unlike its predecessor, the motion decision algorithms are undertaken by the SGVs themselves. Decentralized architecture provides the capability to make the SGVs operation more agile [5], [6]. However, their ability to navigate autonomously in all free spaces relies on the path planning algorithms. Two main path planning algorithms are required for autonomous navigation, the global path planning and the local path planning. The global path design is an offline planner that uses a map of static obstacle. This map is characterised with filled or empty cells representing obstacle space and free space respectively. The global planner draws a geometrically feasible path from actual position to destination point. Next, the second planner is the online local path planning algorithm. It utilises the SGV's perception sensors to generate controlled motion to move along the global path while avoiding obstacles [7]. Therefore, both planned have complementary roles so that the SGV will reach its destination while avoiding collision. However, as modern MHS have reached an unprecedented complexity and size, path planning performances is a key factor in travelling long distances safely and efficiently in a dynamic environment while carrying loads [8]. Moreover, SGVs must deal with continuous changes in the environment. Consequently, from autonomous navigation perspective, these dynamic changes may provoke a confrontation between the operating SGV with new static and dynamic obstacles, and this confrontation is commonly named as conflict [9].

In recent years, research work has paid particular attention to real-time local motion control and obstacle avoidance algorithms [10]. However, state-of-the-art global path criteria are traditionally limited to minimum-distance to the goal. The most popular are Dijkstra and A-star algorithms [5], and kinematics constrains through motion primitives such as Search-based Planning Library (SBPL) [11], [12]. Although Dijkstra provides the optimal path distance, it's main drawback is the path search ability that becomes computationally heavy for large environment as it requires to explore the whole environment A-star was proposed as an improvement to cope with the computation time of Dijkstra by adding a heuristic term that indicates the closeness of each cell to the goal. However, the performances of the optimal global path design drops in unpredictable situation such as the presence of unknown obstacles [13]. In such confliction scenario, the global optimality is lost, and the trajectory needs to be adapted.

As the SGVs operate in a dynamic and busy environment, conflict that is not solved on time will lead to idle situation and even to multi-SGV congestion. Based on the literature review, a vast amount of research works regarding the problem of confliction resolution are proposed. In this paper, we denoted two types of deconfliction approaches, the reactive approaches and proactive approaches.

Reactive deconfliction attempts are triggered subsequently to a conflict event. In [14], a progressive motion planner is used as an intermediate planner between the global and local planners to mitigate the number of possible conflicts in dynamic environment. The latter uses situation awareness module to identify the adequate decision based on traffic scenario map to lead to safer moves. In Robotic Operating System (ROS) navigation stack, recovery behavior module is invoked when navigation fails [15]. These behaviors may vary from passive behaviours such as *Clearing costmap* or *Waiting*, to active behaviors like *Spinning* or *Reversing* motions. Another reactive solution is the *global replanning*. This occurs when the trajectory planned initially is no longer viable, and therefore a global alternative path is required to reach the destination. The sampling-based methods such as Rapidly-exploring Random Tree (RRT) is a very popular replanning method [16]. Improved version of RRT methods such as Online-RRT, Fixed Nodes-RRT, Real Time-RRT and RRT$*$, are proposed to improve the real-time requirement of replanning, but the substitute path is less efficient than the first planned path due to longer distances to travel [17], [18].

In contrast with the above approaches, *proactive conflict avoidance* approaches are anticipative strategies that requires a local coordination between the SGVs when conflict is imminent. The online dynamic obstacle avoidance is the simplest example of this technique, where the mobile platform bypasses an approaching dynamic obstacle [19], [20]. Time Elastic Band (TEB) is an example of a powerful local path planner with obstacle avoidance capability [21]. However, this method is less adequate when two or more SGVs are conflicting each other, because the conflict might just be displaced in adjacent space when both are trying to avoid each other. In [22], intentional exchange through explicit communication is proposed in order to determine "who goes first". In [23] instead, the mobile platforms interpret the approaching vehicle intention through motion legibility. The recent learning-based approaches proposed in [24], [25], and [26] have been studied to address the real-time tuning of the planners' parameters such as speed, acceleration …etc. Even though the global path planned initial is conserved more often, end-to-end learning approaches are data-hungry, requiring hours of training data, either from expert demonstration or trial-and-error tests. Moreover, learning-based methods typically lack safety and explainability, both of which are important properties for SGV operating in environment such as warehouses with infinite number of possible scenarios. Other rule-based and priority-based motion monitor were proposed in literature [27], [28], [29], [30], [31], [32], [33], yet, as far as we know, none of these selected articles have showed the limitation of navigation regarding safety and energy efficiency.

From the state-of-the-art overview, one can state the following: 1) The proposed local conflict avoidance strategies are complex and scenario-oriented. 2) The same conflict resolution attempts are repeated over and over if the same type of conflict is observed in the configuration space. 3) To the extent of our knowledge, no global path algorithm has proposed an anticipative path that avoids or limits the severity of the conflict. Thereby, the scope of this paper involves the recurring collision risk representation through conflict zones on the map, that are considered at the global path level.

The main objective of this paper is to demonstrate that the combination of namely, global path design, environment change, and unknown static and dynamic obstacle have an impact on both safety and energy consumption. Thus, based on the latter assumption, we propose an approach that would represent navigation inefficiency in map locations that are caused by conflicts between the ego-vehicle to new static or dynamic obstacles. Then, depending on the severity of different conflicts, adapting the future trajectories would lead to smoother and safer motion execution also would limit the fast depletion of battery level. The contributions of this present work are described as follows:

- Investigate navigation safety, using TTC the parameter to model the collision risk through the operating space in loaded and unloaded conditions.
- Design an adaptive Conflict Layer (CL) that represents the conflict and behaviour data of local motion executions and generate smoother global trajectories that are less prone to conflicts and replanning.
- Bring forward the metrics regarding the impact of critical area on the battery's State-of-Charge (SOC) and task completion duration.

The remainder of this paper is organized as follows. Section II presents the methodology to attain the contributions. Section III provides the simulation to test the proposed algorithm then experimental tests were performed with a real SGV. Section IV discusses the results, performances, and limitations of the article. Finally, conclusions are dawn in Section V.

II. METHODOLOGY

In this section, we overview the proposed steps to characterise the conflicts that occur between the vehicle with its surrounding environment by use of laser scanners. First, we propose a simplified real-time TTC to estimate the risk of collision of the egoSGV with respect to the dynamic activity. Then a spatial weighted Kernel Density Estimation (wKDE) is defined according to the conflict severity weight measured using minimum TTC at different 2D sample points in a Costmap Layer. The general diagram of the proposed methodology is shown in Fig. 1.

A. MOTION MODEL

The generated motion trajectory is based on a two-wheeled differential drive platform, and since this model allows any turns radius, the footprint of the platform is a circular shape.

As shown in Fig. 2 the *Instantaneous Center of Rotation* (*ICR*) is a point about which the nonholonomic platform rotates, the forward kinematics of the differential-drive at time $t' = t + \delta t$ is described by the following equations [5]:

$$ICR = \begin{pmatrix} ICRx \\ ICRy \end{pmatrix} = \begin{pmatrix} x - Rsin\,(\theta) \\ y + Rcos\,(\theta) \end{pmatrix} \tag{1}$$

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} cos\,(\omega\delta t) & -sin\,(\omega\delta t) & 0 \\ sin\,(\omega\delta t) & cos\,(\omega\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICRx \\ y - ICRy \\ \theta \end{bmatrix} + \begin{bmatrix} ICRx \\ ICRy \\ \omega\delta t \end{bmatrix} \tag{2}$$

$$R = \frac{v}{\omega} \tag{3}$$

where,v is the translational velocity, ω is the rotation velocity of the Self-Guided Vehicle, R is the distance from ICR to the center between the actuated wheels, and $(ICRx, ICRy)$ are the coordinates of ICR. For the rest of this paper the index $\{X\}_{sgv}$ is used to refer to the vehicle parameter.

B. OBSTACLE TRACKING

In order to track the static or dynamic unknown obstacles, our approach utilises the standalone *costmap_converter* plugin in Robot Operating System (ROS) [34]. Density-Based Spatial Clustering of Application Method algorithm is used to convert the obstacle cells shapes provided by the SGV's 2D laser scanners into primitive geometrical forms such as points, lines or polygons [34], [35]. This is to reduce the computation power required to represent obstacles in the configuration space.

Unknown objects are prone to causing abrupt conflicts that may results in operation delay or, in worst case scenario, a severe collision. Therefore, by means of this clustering-based method, we were able to track and estimate the position, orientation and velocity of unforeseen obstacles in real-time, see Fig. 3. The Algorithm 13 is used to limit the observation window to track critical obstacles within the vicinity of SGV, while Algorithm 30 estimates the coordinates and velocity of the latter obstacles.

C. TIME-TO-COLLISION CALCULATION

Analysing the motion performances of the SGV is a crucial step into identifying the conflicts. Therefore, to calculate the collision risk of the ego-SGV with its surrounding obstacles we have used the Time-To-Collision (TTC) parameter [36]. TTC is a time-based risk indicator and is defined as the remaining time before a collision occurs between the ego-SGV and any neighboring obstacle if the speed and direction remain fixed [37], [38]. The TTC is calculated from the predicted poses and the surround information of the ego-platform. Laser scan data and TTC calculation are fast enough to give the conflict severity in real-time. Thus, the following steps are proposed to calculate the TTC more accurately for different scenarios as shown in Fig. 4.

Two methods are considered for TTC calculation, and this depends on the value of the angular velocity of the ego-SGV in the cases where the actual value $\omega_{SGV}$is superior or

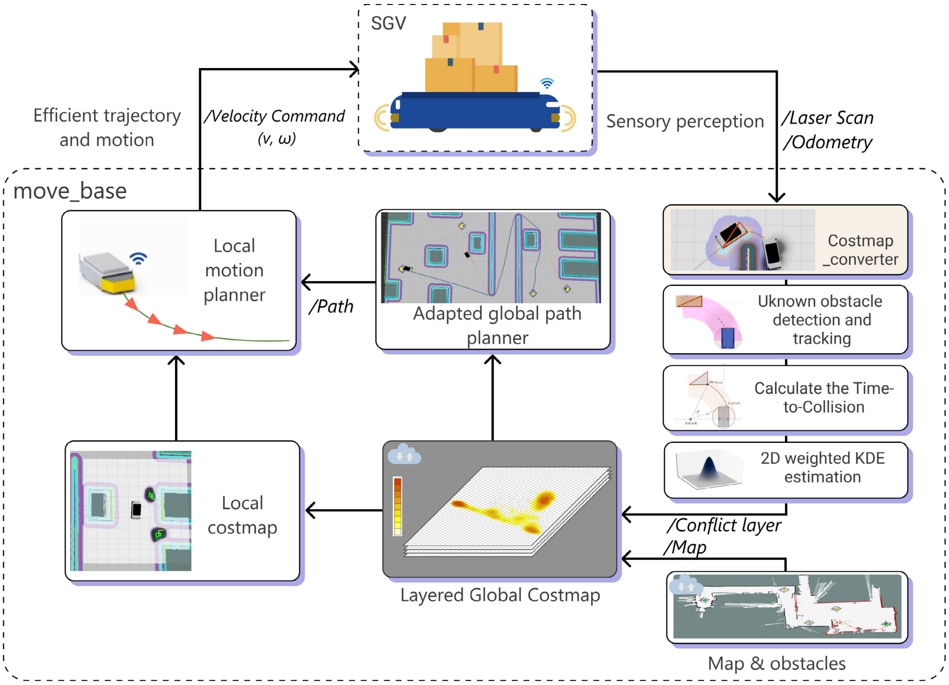


FIGURE 1. The diagram describing the mechanism of proposed method.

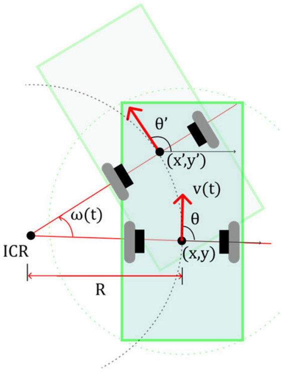


FIGURE 2. Kinematic model of the mobile platform along a circular trajectory.

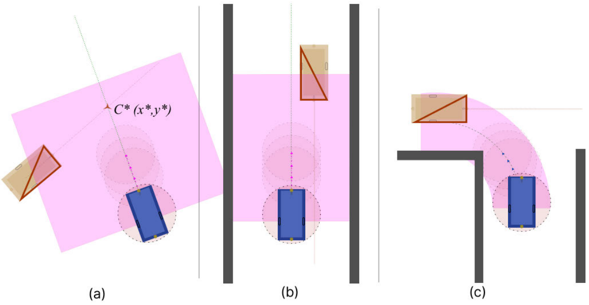


FIGURE 4. TTC estimation for (a) lateral, (b) frontal and (c) turning conflict scenarios corresponding to open, narrow, and blind areas respectively.

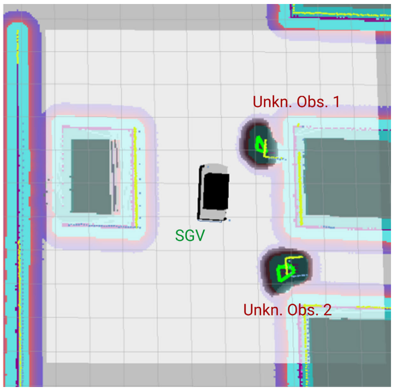


FIGURE 3. Unknown obstacle detection and tracking with a lookahead distance of 5 meters.

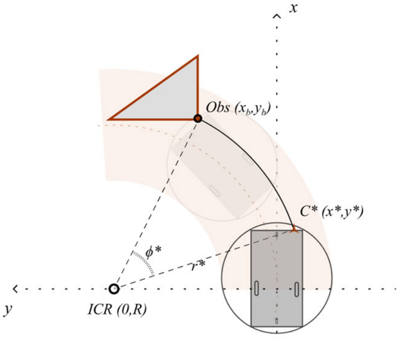


FIGURE 5. The predicted trajectory of the SGV with collision risk with the obstacle.

inferior to the threshold value ε_ω close to zero. The steps for calculating the egoSGV's TTC is summarized in Fig. 6.

Case 1: In the case where, $|\omega_{SGV}| < \varepsilon_\omega$, the mobile platform is moving in a straight line (see Fig. 4 (a) and (b)), hence TTC parameter is calculated by solving the following

Algorithm 1 Dynamic Obstacle's CostMap

```
Result: Defining lookahead window for unknown
        obstacles tracking
Input: Org_cmap: upload from map_server
        the costmap,
       $Upd\_cmap$: global costmap update,
       $Odom = (x_{SGV}, y_{SGV}, \theta_{SGV})$ : SGV
        odometry
       $W$: The window's width
Output: $DO\_CMap$: Dynamic Obstacles' CostMap
Variables: $ROI\_1$: Region Of Intrest of the original
            global costmap,
           $ROI$: Region Of Intrest of the updated
  global costmap,
           $TH$: Threshold of the cost allowed in the
  new costmap,
           $(X_s, Y_s)$ : Top right cell of ROI
$X_s = x_{SGV} + Sign(\theta_{SGV}) \times \frac{W}{2}$;
$Y_s = y_{SGV} + Sign(\theta_{SGV}) \times \frac{W}{2}$;
$ROI\_1 = Copy\,(Org\_CMap, \mathrm{X_s}, \mathrm{Y_s}, \mathrm{W})$
$ROI = Copy\,(Updt\_CMap, \mathrm{X_s}, \mathrm{Y_s}, \mathrm{W})$
for $i \leftarrow 1$ to $W^2$ do
    Cost $= ROI[i] - ROI\_1[i]$;
    if $Cost < TH$ then
        $DO\_CMap[i] \leftarrow 0$;
    else
        $DO\_CMap[i] \leftarrow Cost$;
    end
end
return DO_CMap
```

equation:

$$\left[x_{sgv} + v_{sgv}cos\theta_{sgv} \cdot TTC_i + \frac{1}{2}a_{sgv}cos\theta_{sgv} \cdot TTC_i^2 - \left(x_{obs} + v_{obs}cos\theta_{obs} \cdot TTC_i + \frac{1}{2}a_{obs}cos\theta_{obs} \cdot TTC_i^2\right)\right]^2 + \left[y_{sgv} + v_{sgv}sin\theta_{sgv} \cdot TTC_i + \frac{1}{2}a_{sgv}sin\theta_{sgv} \cdot TTC_i^2 - \left(y_{obs} + v_{obs}sin\theta_{obs} \cdot TTC_i + \frac{1}{2}a_{obs}sin\theta_{obs} \cdot TTC_i^2\right)\right]^2 = \left(r_{sgv} + r_{obs}\right)^2, \tag{4}$$

where TTC_i is the unknown Time-to-collision between the ego-SGV and the dynamic obstacle at i^{th} pose, x_{sgv} and y_{sgv} are the coordinates, v_{sgv} is the translational velocity, a_{sgv} is the acceleration, θ_{sgv} is the orientation, r_{sgv} the footprint radius of the SGV, and r_{obs} is the critical obstacle's footprint. The T_i terms describe the components of the critical obstacle defined in algorithm 2. The smallest real positive root of Eq. (4) is the value of TTC.

The following equation estimates the intersection point $C^*\,(x^*, y^*)$ of the ego-SGV's path with the obstacle's path when $\theta_{sgv} \neq \theta_{obs} \pm \pi$ [39].

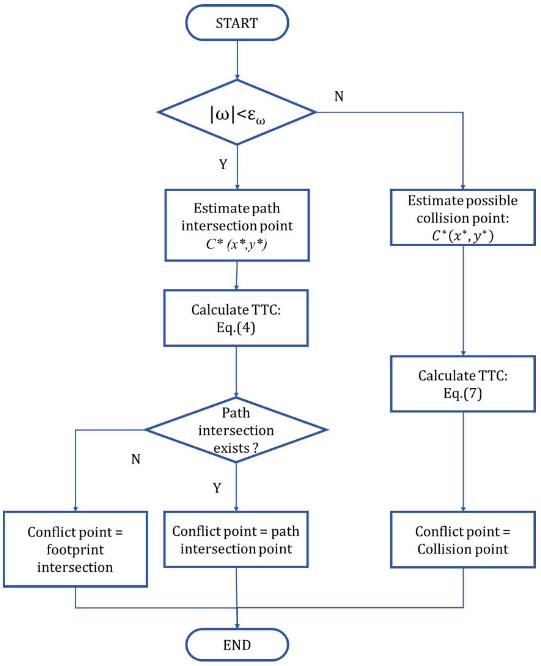

START
$|\omega|<\varepsilon_\omega$
N
Y
Estimate path intersection point $C^*(x^*,y^*)$
Estimate possible collision point: $C^*(x^*,y^*)$
Calculate TTC: Eq.(4)
Calculate TTC: Eq.(7)
Path intersection exists ?
N
Y
Conflict point = footprint intersection
Conflict point = path intersection point
Conflict point = Collision point
END

FIGURE 6. Flowchart for Time-to-Collision calculation.

$$x^* = \frac{\left(y_{sgv} - y_{obs}\right) - \left(x_{sgv}tan\theta_{sgv} - x_{obs}tan\theta_{obs}\right)}{tan\theta_{obs} - tan\theta_{sgv}}$$
$$y^* = \frac{\left(x_{sgv} - x_{obs}\right) - \left(y_{sgv}cot\theta_{sgv} - y_{obs}cot\theta_{obs}\right)}{cos\theta_{obs} - cos\theta_{sgv}} \tag{5}$$

Case 2: If $|\omega_{SGV}| \geq \varepsilon_\omega$ (see Fig. 4 (c)), then the SGV is making a circular motion around *ICR*. In order to determine the exact potential collision point, we first identify if the tracked obstacle polygon is within the planned poses, if so then the minimum arc from obstacle point to the platform (see Fig. 5) is calculated using the formulas in [37] and [40]. Thus, for every pose with a potential collision the TTC is measured as follows:

$$\phi^* = arctan\frac{x_b}{R - y_b} - arctan\frac{x^*}{R - y^*} \tag{6}$$
$$TTC_i = \frac{\phi^*}{\omega_{SGV}} \tag{7}$$

where, $Obs(x_b, y_b)$ is the closest obstacle point to the platform at i^{th} observation, $C^*\,(x^*, y^*)$ is the possible collision point if the kinematics is maintained, and ϕ^* is the angle of the arc between the two latter points with *ICR* as center point. Negative values of TTC_i are ignored because they indicate that the SGV is moving away from the critical obstacle.

The previous procedure is repeated for all unknow obstacle detected by the 2D laser scan, at every detection the lowest value $TTC_i = minTTC$ is used to weight the severity of the conflict.

D. WEIGHTED KERNEL DENSITY ESTIMATION

The conflict estimation module aims at providing a quantification of the conflicts over the configuration space. This is

Algorithm 2 Obstacle Velocity Tracker

```
Result: Tracking the velocity of the unknown
        obstacle.
Input: j_th point of i_th Polygon $Poly_{i,j}$:,
        $t$: Detection time.
Output: $T = \{T_i\}$: vector of n detected obstacles,
        with $Ti = (x_{i\_obs}, y_{i\_obs}, \theta_{i\_obs}, V_{i\_obs})$ as
        parameters of the i_th obstacle.
Variables: $C = \{C_i\}$ : Vector of polygons centroid,
        $last\_C = \{last\_C_{\text{i}}\}$: Vector of polygons
centroid of last detection,
        $last\_t$: last detection time,
        $\Delta x$, $\Delta y$, $\Delta t$: Variation on X, Y and time of
polygon points,
        $V_{x\_obs}$, $V_{y\_obs}$: Velocity on X and Y of
detected obstacle.
last_$C = \emptyset$;
while $True$ do
    for $i \leftarrow 1$ to $n$ do
        if $length\ (Poly_i == 0)$ then
            $C_i \leftarrow (0,0)$;
        else if $length\ (Poly_{\text{i}} == 1)$ then
            $C_{\text{i}} \leftarrow (\text{Poly}_{\text{i},1})$;
        else if $length\ (Poly_{\text{i}} == 2)$ then
            $C_i \leftarrow \left(\frac{Poly_{i,1}+Poly_{i,2}}{2}\right)$;
        else
            $C_i \leftarrow \left(\frac{Poly_{i,1}+Poly_{i,2}+Poly_{i,3}}{3}\right)$;
        end
        if $length(last\_C) > 0$ then
            $\Delta x = C_{i.x} - last\_C_{\text{i.x}}$;
            $\Delta y = C_{i.y} - last\_C_{i.y}$;
            $\Delta t = t - last\_t$;
            $V_{x\_obs} = \frac{\Delta x}{\Delta t}$;
            $V_{y\_obs} = \frac{\Delta y}{\Delta t}$;
            $V_{i\_obs} = \sqrt{Vx\_obs^2 + Vy\_obs^2}$;
            $\theta_{i\_obs} = \arctan(\frac{\Delta x}{\Delta y})$;
            $x_{i\_obs} = last\_C_{i.x}$;
            $y_{i\_obs} = last\_C_{i.y}$;
            $T_i \leftarrow (x_{i\_obs}, y_{i\_obs}, \theta_{i\_obs}, V_{i\_obs})$;
        end
        $T \leftarrow [T_1, T_2, \dots, T_n]$;
    end
    $last\_C \leftarrow C$
    $last\_t \leftarrow t$
    return T
end
```

modelled on a weighted Kernel Density Estimator (wKDE) by collecting the Time-to-Collision parameter of the SGV in different locations. The wKDE is a nonparametric technique that generates density functions based on observation [41]. It can be used in estimation of events distribution involving spatial data. The formulation of the wKDE is shown in Eq. 8:

$$\hat{f}(x, y) = \sum\nolimits_{i=1}^{n} \frac{1}{\pi r^2} w_i K\left(\frac{d_i}{r}\right) \tag{8}$$

where $\hat{f}\ (x, y)$is density estimation at location $C^*(x^*, y^*)$ that corresponds to the location of imminent collision; K is a kernel function with bandwidth or radius r ; d_i is the distance between i^{th} observation from location $C^*(x^*, y^*)$; and w_i is the corresponding severity of each i^{th} observed conflict.

In this paper, Gaussian function is used as a kernel, and it is defined by:

$$K\left(\frac{d_i}{r}\right) = \frac{1}{\sqrt{2\pi}} exp\left(\frac{-d_i^2}{2r^2}\right) \tag{9}$$

where the weight w_i has a value between 0 and 1, and it based on the minimum value of Time-to-Collision parameter TTC_i at i^{th} conflict event. The following equation Eq. 10 defines the weightings:

$$w_i = e^{-TTC_i} \tag{10}$$

This approach allows to be applied for a decentralized fleet of heterogeneous SGVs operating in a shared environment, without stablishing an explicit communication.

The choice of bandwidth r is subjective and it controls the smoothness of the estimation, its selection is crucial in order to avoid tapered estimation when small value, or flattened estimation when higher value is chosen. The most common method to determine the parameter r for spatial data is the rule of thumb [42]. Therefore, by defining σ as the standard distance of the observations, then:

$$r = \sigma * n^{-1/6} \tag{11}$$

Thus, by defining the conflict severity at different location, it is now possible to represent the conflict zones in the global costmap as a layer along with the existing static obstacle.

E. COSTMAP LAYER

The costmap is a two-dimensional informative representation that encodes the configuration space into discrete cells with gray scale value that is used for global and local path planning algorithms to reduce the cost of motion. The layered costmap is proposed by [43] to categorize the configuration space into ordered list of layers in which each layer describes a cost of a specific functionality of the environment. The fundamental layer is the static map that represent the stored map that is build using Simultaneous Localization And Mapping (SLAM) algorithm [44] that represents the walls and static obstacles. Other added layers are provided as plugins in ROS. For instance, an inflation layer is used to impose a buffer distance from different obstacles. The created plugins are then stacked with static map layer to make the master costmap as shown in Fig. 7. One of the objectives of this paper is to avoid collision-risk by SGVs, hence, the generated Conflict Layer (CL) is shared amongst SGVs, so that each mobile platform will plan paths that bypass the latter areas.

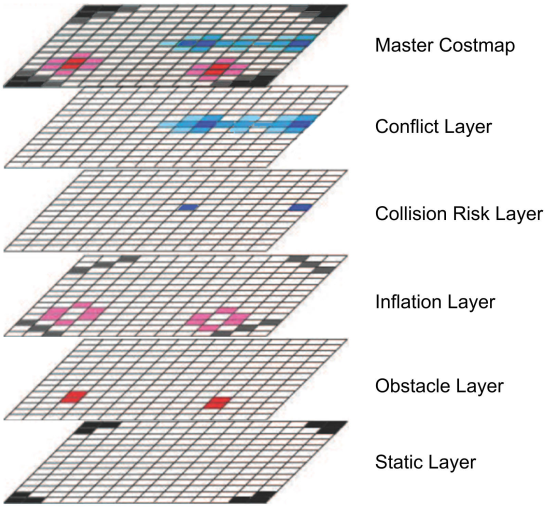


FIGURE 7. **Ordered costmap layers including map of static obstacles and conflict area layer.**

F. GLOBAL PLANNER

In order to generate a global trajectory graph-based search algorithm is used, the cost function consists of a distance cost term and conflict cost term, the function is given as follow:

$$q(n) = \delta_d (g(n) + h(n)) + \delta_c (c(n) + d(n)) \tag{12}$$

where the term $g(n)$ is the distance between starting node to the current node n, and $h(n)$ is a heuristic term that represents distance from node n to the goal. The distance cost term is computed based on the shortest path algorithm namely A-star. The second cost term is computed from the conflict severity index calculated from subsection D, $c(n)$ is the cumulative conflict cost when move through the starting node to the current node n. $d(n)$ is the conflict value of traversing to the next cell $n+1$, considering the conflict severity for each cell as:

$$S(n) = \bar{f}(n)\, \alpha_c (V_i - 1) \tag{13}$$

where $\bar{f}(n)$ is the measured probability density estimated at each cell, V_i is the inscribed cost value (defined as 253), and α_c is the scaling coefficient that maps $S(n)$ between [0, 252] [45]. $\delta_d$and δ_c are parameters that compromises between the distance to travel and the conflict area to avoid. These parameters depend on the navigation scenario and environment. Therefore, in both simulation and real-world tests, the values of $\delta_d$and δ_c are set to 0.9 and 0.6 respectively.

G. ENERGY MODEL

SGVs are battery powered vehicle whose energy autonomy is yet a factor that limits the self-sufficiency of the MHS system. Although these mobile platforms are required to process continuously for several hours, battery source remains one of the main constraints for operation lifespan. Therefore, the energy model is used to evaluate the energy consumption of the proposed trajectory design. The energy consumed by the mobile platform measured through the energy model validated in [46]. The total energy E_{SGV} has four main terms represented by Eq.14 to Eq.19.

$$E_{SGV} = E_{DC} + E_F + E_K + E_E \tag{14}$$

E_{DC}, is the energy consumed by the DC motors defined as:

$$E_{DC} = \int \left((I_a^r)^2 R_a^r + (I_a^l)^2 R_a^l \right) dt \tag{15}$$

where I_a^r and I_a^l are the armature currents, and R_a^r and R_a^l are the armature resistance of the left and right DC motors respectively.

The energy dissipated through friction is represented by E_F:

$$E_F = \int \mu m g \left((V_r(t) - L w_r(t)) + (V_l(t) - L w_l(t)) \right) dt \tag{16}$$

where μ is the coefficient of rolling friction, m is the robot mass, g is the gravity, V_r and w_r are the linear and angular velocities of the right wheel, V_l and w_l are the linear and angular velocities of the left wheel, and L is the axle length of the vehicle.

The kinetic energy E_K is due to motion is expressed as:

$$E_K = \frac{1}{2}(m v_{sgv}^2 + I \omega_{sgv}^2) \tag{17}$$

$$v_{sgv} = \frac{V_r + V_l}{2} \tag{18}$$

where v_{sgv} and ω_{sgv} (defined in equation (3)) are the linear and angular velocities of the vehicle, m is the mass, and I is the moment of inertia of the vehicle.

The energy consumed by the embedded circuitry is represented by E_E, is given by the following equation:

$$E_E = \int (I_{elec} V_{elec}) dt \tag{19}$$

where I_{elec} and V_{elec} are, respectively, the flowing current at the battery and the supply voltage.

Equations (14) to (19) show that the main terms that affect the total energy are the linear and angular velocities and the two motor armature currents that depends on the accelerations. Therefore, a trajectory that has lower accelerations phase will have a direct impact on the amount of energy consumed by the vehicle.

The following sections present tests of the proposed methodology in both, simulation, and real environment.

III. SIMULATION

A. SIMULATION TESTS

This section describes the simulation procedure to evaluate the proposed approach in open-source ROS-Gazebo simulation environment. Two model-based SGVs are deployed, SGV1 and SGV2, in a warehouse-like 3D indoor environment, with four dynamic objects moving in the configuration space, two walking pedestrians and two forklifts, as shown in Fig. 8. In addition, the environment consists of narrow area such a corridor and a blind zone such as corners, to produce

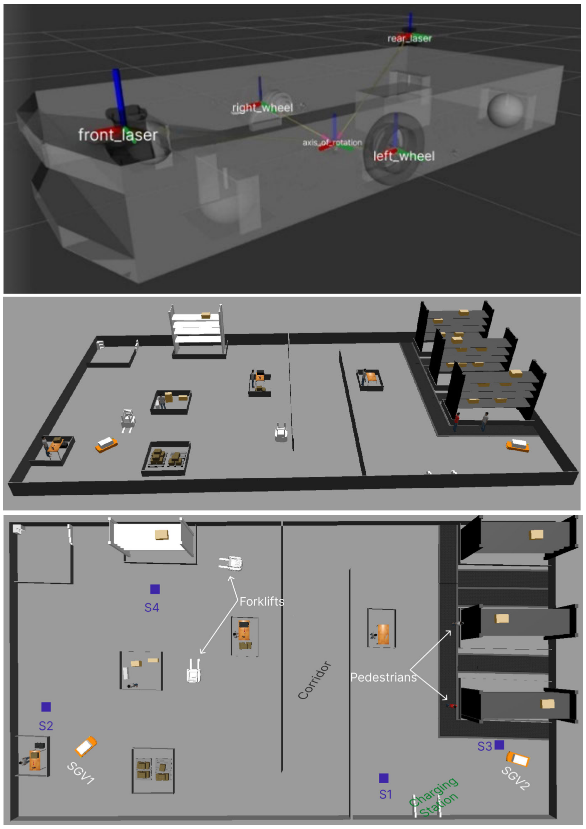


FIGURE 8. The 3D model of the SGV and the warehouse environment in Gazebo.

conflict scenarios as illustrated in Fig. 4. The platform is equipped with laser scanner in both simulation and experimental tests, in addition the map is constructed with the SLAM algorithm [44] prior to the navigation tests.

ROS is used to implement the algorithms with the navigation stack framework. The two-stage planning approach is used, the defined global planner searches the geometrical path to the next station considering the known static obstacles. Then, Time Elastic Band (TEB) is used as local planner to generate controlled motion to follow the planned global trajectory, with translational velocity and angular velocity (v, ω) as control signals respectively [47].

The TEB is tuned in such a way to stay away from obstacles while moving along the global path. The autonomous platforms speeds are set to 1.0 m/s for the linear speed and 0.8 rad/s for the angular speed, whereas the maximum linear and angular accelerations are set to 0.5 m/s^2 and 0.6 rad/s^2 respectively.

Both SGV1 and SGV2 share the configuration space simultaneously. For sake of safety, 0.5 m inflation layer is set from all known static obstacles. SGV1 is allocated to go from station S1→S2→S3→S4→S1 repeatedly, while SGV2 is assigned to S4→S1→S2→S3→S4 repeatedly as well. We define when all stations are visited as completed cycle. In all simulation, the platforms are transporting a constant 50 Kg package to simulate a real-world mission.

In addition to the fact that the SGVs confront each other, the SGVs must avoid the randomly deployed dynamic obstacles in the environment, that is the forklifts and the pedestrians. However, for sake of realistic scenario, the pedestrians are walking in the dedicated zone that is not specified in the costmap, and the forklifts are operating randomly. The velocities of the forklifts and the pedestrians are set to 0.8 and 0.5 m/s, respectively.

The computational platform used to proceed to the tests is core i7, 9rg generation CPU with 16 GB of RAM, and to simulate the environment in Gazebo GTX Geforce 1660 Ti Graphical Processing Unit is used. We have implemented the lookahead window described in Algorithm 13 and the obstacle tracker in Algorithm 30 as ROS nodes that are executed with real-time navigation. However, the calculation of the TTC and the weighted Kernel Density Estimator in Matlab. After processing data, a conflict layer is generated with conflict severity indices for every cell and sent to ROS AS a message to be considered in the navigation stack. The online bidirectional communication between ROS and Matlab is set prior to the tests. Although the laser scanners may provide data at 20 Hz, the node publishing the obstacle position can have an update frequency of 5 Hz, which corresponds to 0.2s between every measurement, therefore makes the calculation critical TTC rigorous.

The simulation is conducted for 7 hours, and we have recorded collision-risks with the TTC index. The minimum observed value of TTC parameter is used in order to estimate the wKDE and then generate the conflict costmap layer for each our of the simulation.

The performance metrics used to support the contribution are, the average consumed energy per cycle, the average path execution time per cycle, the lowest Time-to-Collision measured, and failures to deconflict in percentage. We define failure when the vehicle is stuck in a particular position or takes more than the required time to arrive to the next station.

B. SIMULATION RESULTS

The simulation is set to run the SGVs for several hours in a MHS warehouse. Fig. 9 shows the conflict heatmap generated by the TTC-based weighted KDE. The six-hour evolution of the estimation is shown through H1 to H6 of the same figure. As shown in Fig. 9 the estimation is remain mostly unchanged after H4 in which most of the critical conflicts are defined in the configuration space. The generated global path for every hour is represented in Fig. 10. Table 1. shows the evolution performances of SGV1 and SGV2. Here we compare the navigation performances at H6 with CL, with initial configuration at H0. The average execution time per rotation of the proposed implementation has been decreased by up to 10% for SGV1 and 8.1% for SGV2. In addition, the minimum estimated TTC has increased by 3 times for both vehicles. This reduces the collision risk, and makes the navigation within dynamic obstacle, such as unforeseen pedestrians, more secure and less prone to severe conflict. Moreover, with Conflict Layer added to the master

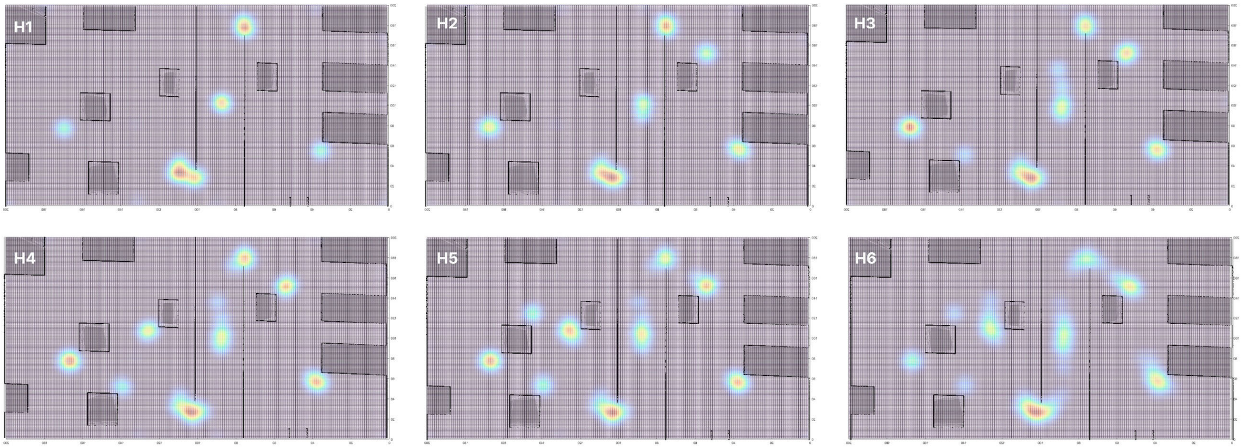


FIGURE 9. The evolution of the conflict estimation with time, (H1) to (H6) corresponding to first hour to sixth hour.

TABLE 1. The performances of simulation tests of the two SGVs.

Simulation		Average execution time / cycle (s)		Minimum TTC (s)		Fail to deconflict (%)		Average energy consumption / cycle (KJ)		
Time	*Algo.*	*SGV1*	*SGV2*	*SGV1*	*SGV2*	*SGV1*	*SGV2*	*SGV1*	*SGV2*	*Saved (%)*
H0	A*+TEB	162.1	166.9	0.32	0.39	23.2	24.6	31.762	32.156	-
H1	A*+CL+ TEB	161.575	164.1	0.49	0.55	21.1	25.3	31.446	30.941	2.4
H2		159.625	156.3	0.55	0.59	19.7	22.5	28.273	29.965	8.9
H3		151.15	161.4	0.61	0.76	19.0	18.7	29.204	29.274	8.5
H4		145.85	159.8	0.91	0.88	15.9	14.8	27.990	28.747	11.2
H5		146.7	155.6	0.97	1.10	13.7	14.1	28.580	27.291	12.6
H6		147.025	153.3	1.02	1.08	14.5	13.9	27.846	28.916	11.2

TABLE 2. The characteristics of the self-guided vehicle.

Parameter	Value
Maximum speed	1.0 m/s
Vehicle weight	90 Kg
Payload	100Kg
Battery type	40 Ah LiFePO4
Battery lifetime	7h to 9h
Dimensions (l x w x h)	165 x 76 x 23 cm

costmap, we have observed that the capacity of the SGVs to go from station to another within reasonable time has considerably improved. Another performance aspect is the ability to reach its destination on time, hence for both SGVs, 40-45 % previously persisting conflicts are now avoided. The last column of the same table shows the average consumed energy. The total consumption drops by up to 12.6% for both vehicles.

IV. EXPERIMENT

A. EXPERIMENTAL VALIDATION

The validation tests of path planning and execution with the proposed CL-based global costmap are made using a real industrial Self-Guided Vehicle shown in Fig. 11, with the characteristics detailed in Table 2. The tests were conducted in the warehouse of the Hydrogen Research Institute. Although the experimental environment differs from the simulation one, the navigation scenarios are of the same nature, the SGV operates in narrow and blind areas. Also, it is to mention the presence of a pedestrian and an autonomous robot (TurtleBot 3) in the warehouse as dynamic obstacles.

The only exteroceptive sensor used for localization and mapping is the 2D 360-degree laser scanners with a maximum range of 40 m as shown in Fig. 11. Two computers are used, one is embedded with the platform that contains all developed algorithms along with the navigation stack and the second PC is used for remote control to lunch and abort the navigation missions. The proposed methodology is implemented with plugins in the navigation stack that are coded using C++ programming language. TEB is used as a local motion

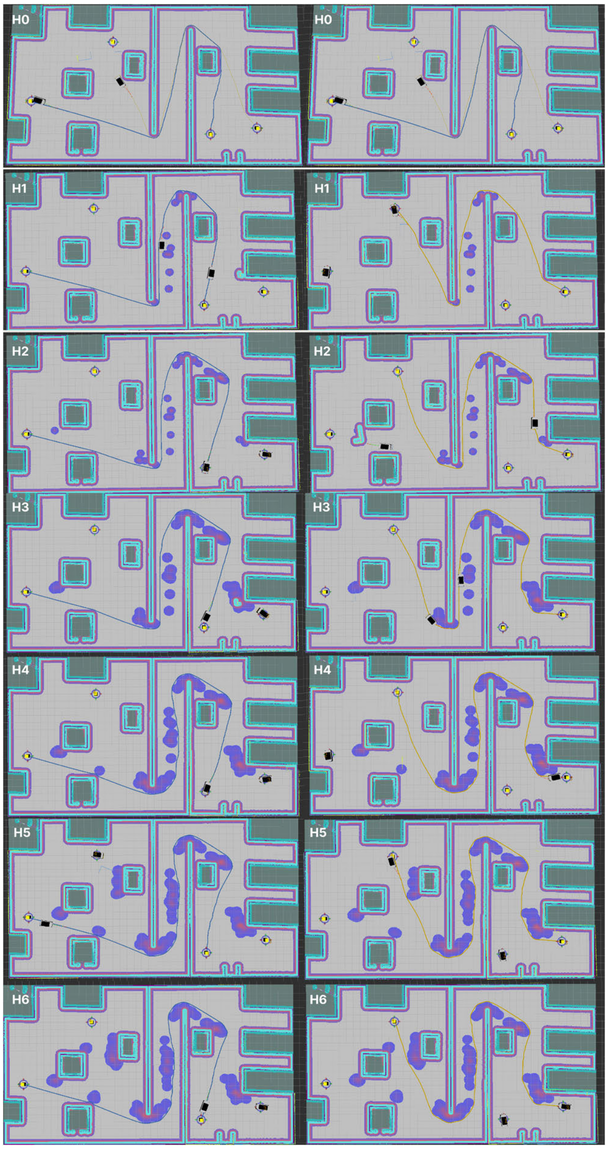


FIGURE 10. The hourly global trajectory sample of SGV1 (left column) and SGV2 (right column) in blue and yellow lines respectively.

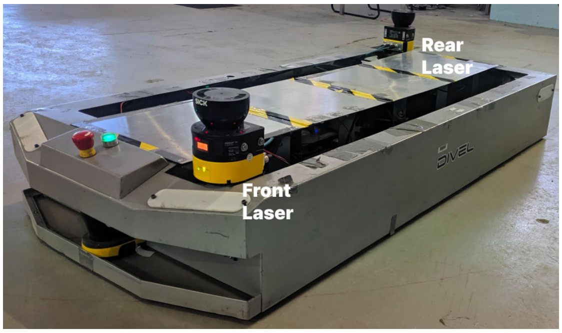


FIGURE 11. The Self-Guided Vehicle with front and rear laser scanners used for experimental validation.

executor during which the parameters of the motion remain unchanged. Collisions due to localization uncertainty are excluded throughout the simulation and experimental tests.

The aim of the experimental test is to evaluate the proposed method with a real-world collision-risk between the SGV and the moving obstacles. Another aspect of experiment is the energy consumption, the battery level evolution is observed with and without the built CL. To do so, the battery state-of-charge (SOC) is computed after each cycle l with the following forumula:

$$SOC_l = SOC_{l-1} - \sum_{k=1}^{n} \frac{I_k \Delta T}{C_{bat}} \tag{20}$$

where SOC_{l-1} is the estimated battery SOC after completing cycle $l-1$, ΔT is sampling time, C_{bat} is the battery nominal capacity, I_k is the current provided by the battery at the k$^{\text{th}}$ sample and it is defined as:

$$I_k = I_a^r + I_a^l + I_{elec.} \tag{21}$$

where I_a^r and I_a^l are the right and left motor traction respectively and $I_{elec.}$ is the current required by the instrumentation. The sampling time ΔT is set to 0.08s.

As presented in Fig.12, the mobile platform moves from station S1 to S3 through S2, after S3 the vehicle goes back to S1 to complete a cycle. Two 8 hours continuous tests are considered, the first has no load the second has a 100 Kg carrying. A fully charged battery are used at the start of both tests. In other words, the initial and final SOC is 100% and 30 %. The reason is that below 30% SOC the vehicle is less responsive to obstacle, this might even result in collision [5].

B. EXPERIMENTAL RESULTS

To model the Conflict Layer, two unforeseen dynamic obstacles are present in the navigation space (TurtleBot 3 detected in location 1 and 4, and pedestrian in location 2 and 5), and a new static obstacle appears in the corridor located in 3 and 6 shown in Fig 12. The initial paths generated between stations are shown in Fig. 12 (a) with green line, whereas the resulting CL with the adapted trajectories are shown in Fig. 12 (b). The difference between the two cases from safety aspects are illustrated in captures 1 through 6 of the same figure. The blind corner scenarios in 1 and 2 reveal that the SGV has nearly collided the unforeseen obstacles. In addition, in the case where the SGV was able to deconflict, high dynamic maneuvers and reverse motion were performed by the SGV. However, in images 4 and 5, the same obstacles were detected in advance by the mobile platform. Thus, the local planner was able to adjust its motion by initiating an obstacle avoidance strategy with minimum deceleration. In scenario shown in images 3 and 6, a new fixed obstacle appeared in the corridor. Although, the presence of the obstacle is detected earlier (see image 3), the first attempts to avoid the latter wasn't successful due to the lack of space to generate collision-free poses. Nevertheless, when the obstacle is represented in the CL of the global costmap, the SGV avoids the obstacle through global path (see image 6).

An example of motion profiles prior and after modeling the CL are plotted in Fig. 13. The execution time to complete

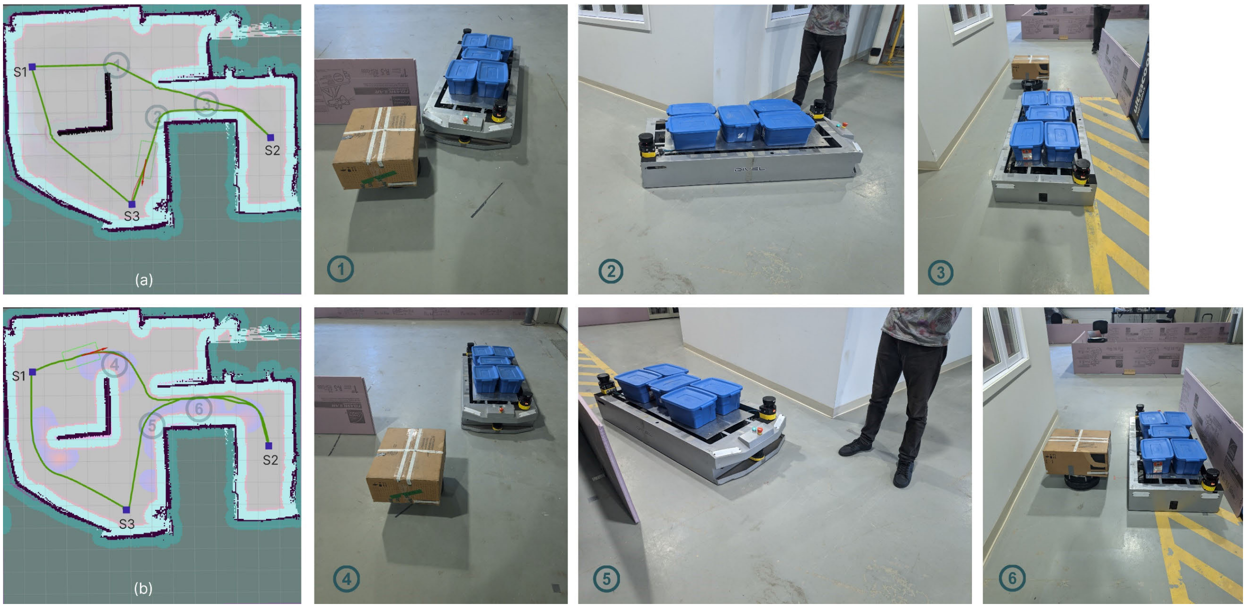


FIGURE 12. The real-world tests for static costmap a (1, 2, 3), and static costmap with Conflict Layer b (4, 5, 6).

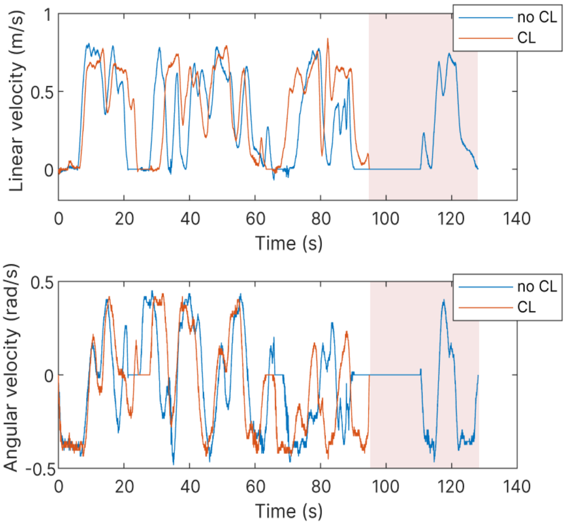


FIGURE 13. The translational and angular velocities profiles of the SGV completing a cycle in both cases.

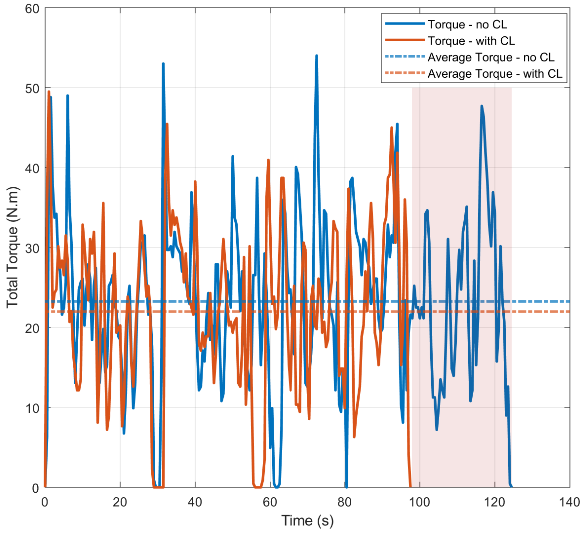
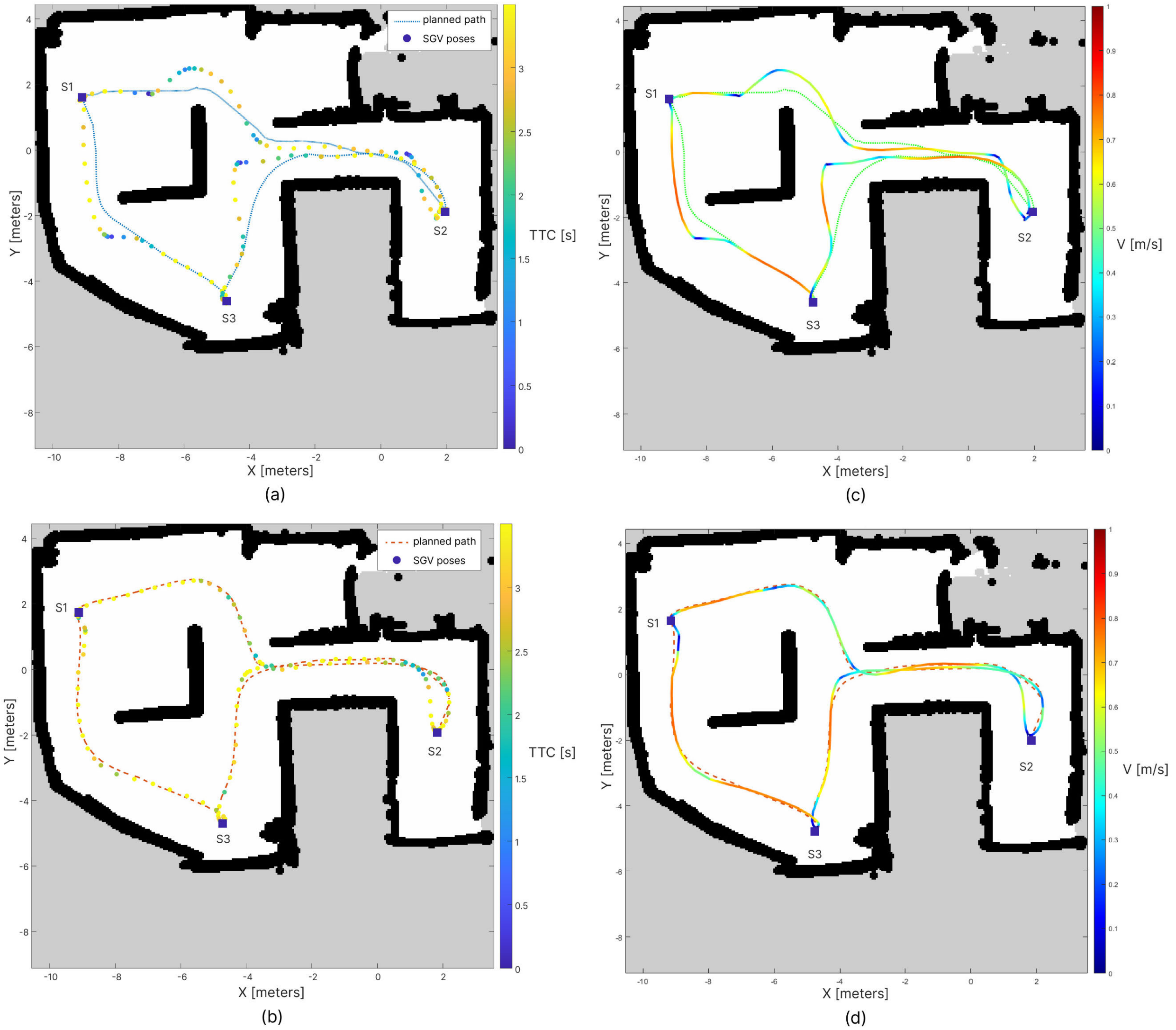


FIGURE 14. The total required torque of the SGV completing a cycle in both cases for a 100 Kg load.

a whole cycle in the second case has reached up to 24.8 % lesser than in the first. This is because a conflict forced the platform to stop, rotate and accelerate to reach the nominal speed in the first case. The corresponding torque required by the SGV is represented in Fig.14. In Fig.15, the poses of the executed trajectories and the corresponding velocity are presented. The results obtained in Fig.15 (a) and (b) show that the poses of the vehicle with CL follow better the global path represented with dashed orange line. In addition, higher velocity is maintained, this means that less brakings occur (see Fig. 15 (c) and (d)). The TTC distribution in Fig.16 shows the severity of collision-risk at different poses. The minimum TTC value went from 0.42 to 0.95 seconds with the CL, that is an increase of 125 %. This makes the navigation much more secure to the vehicle and to other critical obstacles such as pedestrian. The power consumption for both cases is presented in Fig. 17, where Fig. 17 (a) shows the histograms of power that is consumed before and after considering the CL, with 0 Kg pay load. While in Fig.17 (b), the power required to carry a 100Kg payload for both cases. The SOC estimation reveals that the energy consumed initially is higher than when considering the conflict locations. This intensifies when higher loads are carried due to repeated acceleration and rotation. For the same task workload, Fig. 18 shows that the SGV has saved up to 12.2% SOC with 100Kg payload. Furthermore, the failure to clear from the conflicts is improved, thus 19 out of 100 planned trajectories failed to



(a)

(c)

(b)

(d)

FIGURE 15. The executed trajectory of the SGV with the corresponding TTC at every pose, without CL (a) and with CL (b). The velocity profile for each case is represented in (c) and (d) respectively.

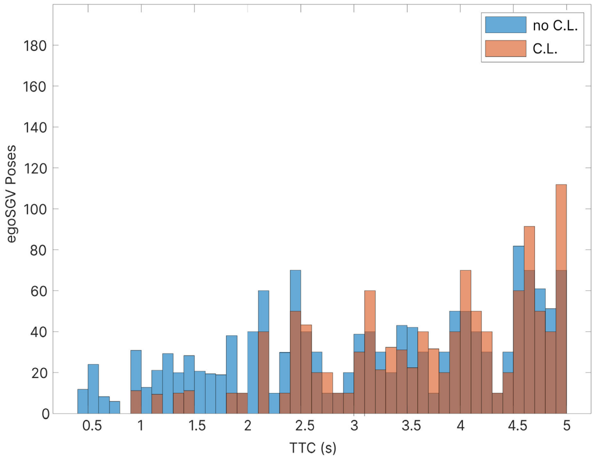


FIGURE 16. The TTC (<5s) distribution at different poses for both cases.

deconflict initially, whereas, when the global trajectory was adapted, the failure drops to 9 failures.

V. DISCUSSION

Global trajectory design is playing a key role in enhancing the MHS's safety and energy efficiency. Traditionally, the maximum velocity, the maximum acceleration and the travelled distance are the parameters associated to energy optimisation techniques of autonomous mobile platform. Nevertheless, it has been demonstrated throughout this article that although the travelled distance increases, energy, and time spent by the mobile platform to reach the next station drops. Furthermore, as accelerations are the part during which power requirements are the highest, the average power consumption is decreasing as conflict zones are being avoided, this is particularly significant when higher payloads are carried. In addition, the TTC is less critical, therefore it insures a safer navigation in the warehouse. The energy losses and safety of navigation conflicts, to the best of our knowledge, is not clearly described in literature, hence the contribution of this paper was to provide a navigation architecture that considers persistent conflicts in the global costmap layer,

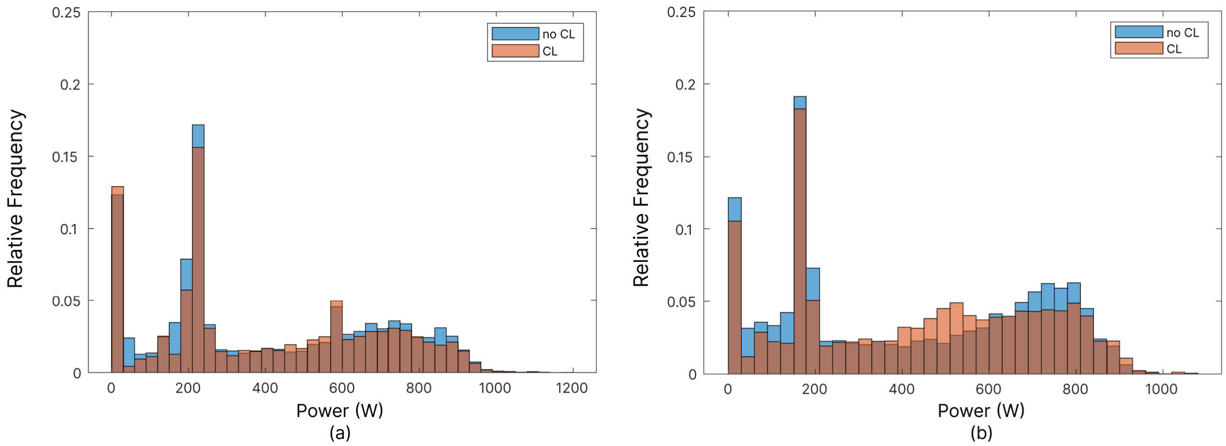


FIGURE 17. (a) The power consumption distribution for 0 kg carried load. (b) The power consumption distribution for 100 kg carried load.

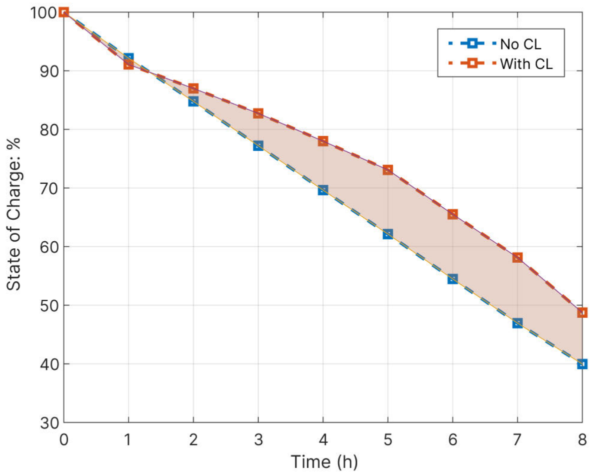


FIGURE 18. The SOC depletion for 8h test with and without CL, for a 100 Kg load.

so that they will be avoided through future global trajectory design.

As online conflict resolution techniques require high computation efforts in real-time and may suffer from repeatability. The proposed method compared to previous methods has the capacity to process big amount of data that is time consuming and that cannot be processed online. In addition, considering navigation conflict at offline global level enables the capacity of updating the costmap and adapting the global trajectory of the SGV. Moreover, theses the data can be shared with other mobile platforms. In terms of energy consumption, as SGV fleet operates in shared indoor space, it is most likely that these vehicles avoid each other with motions generated by the local planner. Nevertheless, highly dynamic motions due to acceleration and rotation are very demanding in terms of power and battery effort (contributing to faster battery depletion), mainly in the case where heavy payload are transported. Therefore, as an assumption made previously in this paper, limiting these efforts improves the navigation safety and the long-term energy consumption of the overall fleet system.

VI. CONCLUSION

In material handling applications, SGVs avoiding each other, and an SGV avoiding a dynamic obstacle are two different scenarios as local motion coordination must be defined among the heterogenous mobile platforms in the former case. To improve navigation smoothness, we have proposed a method based on collision-risk stochasticity that is then merged to the global geometric map. Then, a global trajectory design is proposed to minimise between distance cost and conflict cost. To evaluate the proposed work, tests were considered in both simulation and real-world experiment with different types of dynamic objects. The results have demonstrated a significant increase of the critical TTC parameter by 300% from the initial value, making the navigation in dynamic environment safer and prevents navigation failure by up to 52%. Moreover, for the same amount of workload, our results show that the smoothness of the executed trajectory reduces the energy consumption required by the SGV to complete 8 hours tasks by over 12%, compared to the state-of-the-art path planning techniques. The energy-efficiency will be significant when pallets with higher payload are transported through long distances. Another aspect of the proposed application is the execution time. Solving conflicts are often computationally complex and time-consuming. Besides, we showed that avoiding persistent conflicts fasten the task execution by around 10%.

APPENDIX

The video of the experimental results is available at: https://youtu.be/djfFhJBIkjY.

ACKNOWLEDGMENT

The authors would like to thank Alaeddine Kraiem for the valuable technical support.

REFERENCES

[1] M. Qi, X. Li, X. Yan, and C. Zhang, "On the evaluation of AGVS-based warehouse operation performance," *Simul. Model. Pract. Theory*, vol. 87, pp. 379–394, Sep. 2018.

[2] 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," *Eur. J. Oper. Res.*, vol. 294, no. 2, pp. 405–426, Oct. 2021.

[3] M. B. Alatise and G. P. Hancke, "A review on challenges of autonomous mobile robot and sensor fusion methods," *IEEE Access*, vol. 8, pp. 39830–39846, 2020.

[4] E. A. Oyekanlu, A. C. Smith, W. P. Thomas, G. Mulroy, D. Hitesh, M. Ramsey, D. J. Kuhn, J. D. Mcghinnis, S. C. Buonavita, N. A. Looper, M. Ng, A. Ng'oma, W. Liu, P. G. Mcbride, M. G. Shultz, C. Cerasi, and D. Sun, "A review of recent advances in automated guided vehicle technologies: Integration challenges and research areas for 5G-based smart manufacturing applications," *IEEE Access*, vol. 8, pp. 202312–202353, 2020.

[5] M. Graba, S. Kelouwani, L. Zeghmi, A. Amamou, K. Agbossou, and M. Mohammadpour, "Investigating the impact of energy source level on the self-guided vehicle system performances, in the industry 4.0 context," *Sustainability*, vol. 12, no. 20, p. 8541, Oct. 2020.

[6] M. De Ryck, M. Versteyhe, and F. Debrouwere, "Automated guided vehicle systems, state-of-the-art control algorithms and techniques," *J. Manuf. Syst.*, vol. 54, pp. 152–173, Jan. 2020.

[7] B. K. Patle, G. Babu L, A. Pandey, D. R. K. Parhi, and A. Jagadeesh, "A review: On path planning strategies for navigation of mobile robot," *Defence Technol.*, vol. 15, no. 4, pp. 582–606, Aug. 2019.

[8] M. Mohammadpour, S. Kelouwani, M.-A. Gaudreau, B. Allani, L. Zeghmi, A. Amamou, and M. Graba, "Energy-efficient local path planning of a self-guided vehicle by considering the load position," *IEEE Access*, vol. 10, pp. 112669–112685, 2022.

[9] S. Huang, R. S. H. Teo, and K. K. Tan, "Collision avoidance of multi unmanned aerial vehicles: A review," *Annu. Rev. Control*, vol. 48, pp. 147–164, Jan. 2019.

[10] Y. Han, Y. Cheng, and G. Xu, "Trajectory tracking control of AGV based on sliding mode control with the improved reaching law," *IEEE Access*, vol. 7, pp. 20748–20755, 2019.

[11] N. Limpert, S. Schiffer, and A. Ferrein, "A local planner for Ackermann-driven vehicles in ROS SBPL," in *Proc. Pattern Recognit. Assoc. South Afr. Robot. Mechatronics Int. Conf. (PRASA-RobMech)*, Nov. 2015, pp. 172–177.

[12] G. Tang, C. Tang, C. Claramunt, X. Hu, and P. Zhou, "Geometric A-star algorithm: An improved A-star algorithm for AGV path planning in a port environment," *IEEE Access*, vol. 9, pp. 59196–59210, 2021.

[13] R. Szczepanski, T. Tarczewski, and K. Erwinski, "Energy efficient local path planning algorithm based on predictive artificial potential field," *IEEE Access*, vol. 10, pp. 39729–39742, 2022.

[14] M. Rodrigues, A. McGordon, G. Gest, and J. Marco, "Adaptive tactical behaviour planner for autonomous ground vehicle," in *Proc. UKACC 11th Int. Conf. Control (CONTROL)*, Aug. 2016, pp. 1–8.

[15] A. Saphala and P. I. Tanaya, "Implementation and reconfiguration of robot operating system on human follower transporter robot," *Commun. Inf. Technol. J.*, vol. 9, no. 2, pp. 59–65, 2015.

[16] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access*, vol. 2, pp. 56–77, 2014.

[17] B. Chandler and M. A. Goodrich, "Online RRT* and online FMT*: Rapid replanning with dynamic cost," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 6313–6318.

[18] B. Tong, Q. Liu, and C. Dai, "A RRT*FN based path replanning algorithm," in *Proc. IEEE 4th Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Dec. 2019, pp. 1435–1445.

[19] A. Chakravarthy and D. Ghose, "Obstacle avoidance in a dynamic environment: A collision cone approach," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 28, no. 5, pp. 562–574, Sep. 1998.

[20] H. Dong, C.-Y. Weng, C. Guo, H. Yu, and I.-M. Chen, "Real-time avoidance strategy of dynamic obstacles via half model-free detection and tracking with 2D LiDAR for mobile robots," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 4, pp. 2215–2225, Aug. 2021.

[21] C. Rosmann, F. Hoffmann, and T. Bertram, "Kinodynamic trajectory optimization and control for car-like robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 5681–5686.

[22] A. Rathi, G. Rohith, and M. Vadali, "Dynamic prioritization for conflict-free path planning of multi-robot systems," 2021, *arXiv:2101.01978*.

[23] B. Capelli, C. Secchi, and L. Sabattini, "Communication through motion: Legibility of multi-robot systems," in *Proc. Int. Symp. Multi-Robot Multi-Agent Syst. (MRS)*, Aug. 2019, pp. 126–132.

[24] X. Xiao, B. Liu, G. Warnell, J. Fink, and P. Stone, "APPLD: Adaptive planner parameter learning from demonstration," 2020, *arXiv:2004.00116*.

[25] Z. Xu, G. Dhamankar, A. Nair, X. Xiao, G. Warnell, B. Liu, Z. Wang, and P. Stone, "APPLR: Adaptive planner parameter learning from reinforcement," 2020, *arXiv:2011.00397*.

[26] Z. Wang, X. Xiao, G. Warnell, and P. Stone, "APPLE: Adaptive planner parameter learning from evaluative feedback," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7744–7749, Oct. 2021.

[27] C. Street, S. Putz, M. Muhlig, N. Hawes, and B. Lacerda, "Congestion-aware policy synthesis for multirobot systems," *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 262–280, Feb. 2022.

[28] M. Shahriari and M. Biglarbegian, "Conflict resolution of cluttered multi-robot systems using metaheuristic optimization algorithms," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2015, pp. 210–215.

[29] P. Scerri, S. Owens, B. Yu, and K. Sycara, "A decentralized approach to space deconfliction," in *Proc. 10th Int. Conf. Inf. Fusion*, Jul. 2007, pp. 1–8.

[30] W. Wu, S. Bhattacharya, and A. Prorok, "Multi-robot path deconfliction through prioritization by path prospects," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 9809–9815.

[31] R. Luna and K. E. Bekris, "Network-guided multi-robot path planning in discrete representations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 4596–4602.

[32] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *Int. J. Adv. Robot. Syst.*, vol. 10, no. 12, p. 399, 2013.

[33] L. S. Marcolino and L. Chaimowicz, "Traffic control for a swarm of robots: Avoiding group conflicts," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 1949–1954.

[34] *costmap_converter*. [Online]. Available: http://wiki.ros.org/costmap_converter

[35] C. Rosmann, W. Feiten, T. Wösch, F. Hoffmann, and T. Bertram, "Efficient trajectory optimization using a sparse model," in *Proc. Eur. Conf. Mobile Robots*, Sep. 2013, pp. 138–143.

[36] M. Shahriari and M. Biglarbegian, "Toward safer navigation of heterogeneous mobile robots in distributed scheme: A novel time-to-collision-based method," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 9302–9315, Sep. 2022.

[37] M. Bosnak and I. Skrjanc, "Efficient time-to-collision estimation for a braking supervision system with LiDAR," in *Proc. 3rd IEEE Int. Conf. Cybern. (CYBCONF)*, Jun. 2017, pp. 1–6.

[38] J. Hou, G. F. List, and X. Guo, "New algorithms for computing the time-to-collision in freeway traffic simulation models," *Comput. Intell. Neurosci.*, vol. 2014, pp. 1–8, Jan. 2014.

[39] F. Jiménez, J. E. Naranjo, and F. García, "An improved method to calculate the time-to-collision of two vehicles," *Int. J. Intell. Syst.*, vol. 11, no. 1, pp. 34–42, 2013.

[40] *Circle-Circle Intersection*. [Online]. Available: https://mathworld.wolfram.com/Circle-CircleIntersection.html

[41] B. Qin and F. Xiao, "A non-parametric method to determine basic probability assignment based on kernel density estimation," *IEEE Access*, vol. 6, pp. 73509–73519, 2018.

[42] L. Liu and D. A. Shell, "An efficient distributed topo-geometric spatial density estimation method for multi-robot systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 828–833.

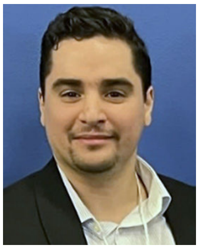
[43] D. V. Lu, D. Hershberger, and W. D. Smart, "Layered costmaps for context-sensitive navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 709–715.

[44] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 108–117, Sep. 2006.

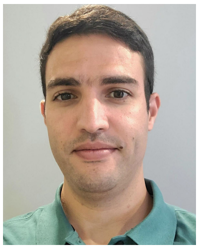
[45] *costmap_2d*. [Online]. Available: http://wiki.ros.org/costmap_2d

[46] A. Ghobadpour, A. Amamou, S. Kelouwani, N. Zioui, and L. Zeghmi, "Impact of powertrain components size and degradation level on the energy management of a hybrid industrial self-guided vehicle," *Energies*, vol. 13, no. 19, p. 5041, Sep. 2020.

[47] C. Rosmann, F. Hoffmann, and T. Bertram, "Timed-elastic-bands for time-optimal point-to-point nonlinear model predictive control," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2015, pp. 3352–3357.

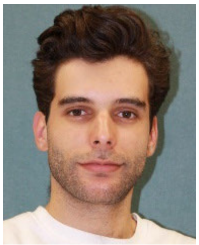


MASSINISSA GRABA received the B.Sc. degree in control and automation engineering from the Institute of Electrical and Electronic Engineering (IEEE), University of Boumerdes, Algeria, in 2015, and the dual M.Sc. degree in smart aerospace and autonomous systems from Université de Paris-Saclay, France, and Poznan University of Technology, Poland, in 2018. He is currently pursuing the Ph.D. degree in electrical and computer engineering with Université du Québec à Trois-Rivières, Canada. His work primarily focuses on safe and energy-efficient trajectory planning of self-guided industrial vehicles in the context of sustainable and smart manufacturing.



ALI AMAMOU (Member, IEEE) received the B.S. degree in industrial computing and automatic science from the National Institute of Applied Sciences and Technology, Tunis, Tunisia, in 2013, the M.S. degree in embedded systems science from Arts et Métiers ParisTech University, France, in 2014, and the Ph.D. degree in energy and thermal management of electric vehicles in cold weather conditions from Université du Québec à Trois-Rivieres, QC, Canada, in 2018. He is currently a Postdoctoral Research Fellow with the Hydrogen Research Institute. His research interests include the optimization of energy systems for stationary and mobile applications, hybridization of energy sources for vehicular application, and eco-energy navigation of autonomous electric vehicle.

SOUSSO KELOUWANI (Senior Member, IEEE) received the Ph.D. degree in robotics systems from Ecole Polytechnique de Montreal, in 2011. He completed his Postdoctoral Internship on fuel cell hybrid electric vehicles with Université du Québec à Trois-Rivières (UQTR), in 2012. He developed expertise in the optimization and intelligent control of vehicular applications. He has been a Full Professor in mechatronics with the Department of Mechanical Engineering, since 2017. He holds four patents in U.S. and Canada. He has published more than 100 scientific articles. His research interests include optimizing energy systems for vehicle applications, advanced driver assistance techniques, and intelligent vehicle navigation taking into account Canadian climatic conditions. He has been a member of the Hydrogen Research Institute. He is a holder of the Canada Research Chair in Energy Optimization of Intelligent Transport Systems and a holder of the Noovelia Research Chair in Intelligent Navigation of Autonomous Industrial Vehicles. He was the Co-President and the President of the Technical Committee of the IEEE International Conferences on Vehicular Power and Propulsion in Chicago, USA, in 2018, and Hanoi, Vietnam, in 2019. He is the winner of the Canada General Governor Gold Medal, in 2003, and a member of the Order of Engineers of Quebec. In 2019, his team received the First Innovation Prize in partnership with DIVEL, awarded by the Association des Manufacturiers de la Mauricie et Center-du-Québec for the development of an autonomous and natural navigation system. In 2017, he received the Environment Prize from the Gala des Grands Prix d'excellence en transport and the Association québécoise du Transport (AQTr), for the development of hydrogen range extenders for electric vehicles.



BILEL ALLANI received the B.S. degree in industrial and automatic computing from the National Institute of Applied Sciences and Technologies, Tunisia, in 2020. He is currently pursuing the master's degree in electrical and computer engineering with Université du Québec à Trois-Rivières (UQTR), QC, Canada. His research interests include multiple mobile robot systems and charging strategies.

LOTFI ZEGHMI received the B.S. degree in control systems from the University of Science and Technology Houari Boumediene (USTHB), Algiers, Algeria, in 2017, and the master's degree in electrical engineering from Université du Québec à Trois-Rivières (UQTR), QC, Canada, in 2020. He is currently a Research Assistant with UQTR. His research interests include mobile robotics and control systems.

KODJO AGBOSSOU (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronic measurements from Université de Nancy I, France, in 1987, 1989, and 1992, respectively. He was the Head of the Engineering School, Université du Québec à Trois-Rivières (UQTR) (2011–2017), where he was the Head of the Department of Electrical and Computer Engineering (2007–2011) and also the Director of graduate studies in electrical engineering (2002–2004). He was a Postdoctoral Researcher (1993–1994) with the Electrical Engineering Department, UQTR, and a Lecturer (1997–1998) with the Electrical Engineering Department. He is currently a Hydro-Québec Research Chair holder with the Transactive Management of Power and Energy in the Residential Sector, and the Chair of the Smart Energy Research and Innovation Laboratory, UQTR. He is the author of more than 325 publications and has four patents and two patents pending. His current research interests include renewable energy, the use of hydrogen, home demand side management, integration of energy production, storage and electrical energy generation systems, connection of electrical vehicle to the grid, and control and measurements. He is a member of the Hydrogen Research Institute and Research Group "GREI," UQTR. Since 2015, he has been the Sub-Committee Chair of Home and Building Energy Management of Smart Grid Technical Committee and IEEE Industrial Electronics Society (IES).

MOHAMMAD MOHAMMADPOUR received the B.S. degree in mechanical engineering, in 2011, and the master's degree in aerospace engineering from the Amirkabir University of Technology, Tehran, Iran, in 2014. He is currently pursuing the Ph.D. degree with the Department of Mechanical Engineering, Université du Québec à Trois-Rivières (UQTR), QC, Canada. His research interests include mobile robotics, motion planning, machine learning, dynamic, and control.