axioms

MDPI

Article

Optimizing Port Multi-AGV Trajectory Planning through Priority Coordination: Enhancing Efficiency and Safety

Yongjun Chen [1], Shuquan Shi [2,*], Zong Chen [3], Tengfei Wang [4,5], Longkun Miao [1] and Huiting Song [1]

[1] Airport College, Binzhou University, Binzhou 256600, China; 277797@whut.edu.cn (Y.C.); jc_mlk@bzu.edu.cn (L.M.)

[2] China Waterborne Transport Research Institute, Beijing 100088, China

[3] School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

[4] School of Transportation and Logistics Engineering, Wuhan University of Technology, Wuhan 430063, China; wangtengfei@whut.edu.cn

[5] State Key Laboratory of Maritime Technology and Safety, Wuhan University of Technology, Wuhan 430063, China

* Correspondence: shishuquan@wti.ac.cn

Abstract: Efficient logistics and transport at the port heavily relies on efficient AGV scheduling and planning for container transshipment. This paper presents a comprehensive approach to address the challenges in AGV path planning and coordination within the domain of intelligent transportation systems. We propose an enhanced graph search method for constructing the global path of a single AGV that mitigates the issues associated with paths closely aligned with obstacle corner points. Moreover, a centralized global planning module is developed to facilitate planning and scheduling. Each individual AGV establishes real-time communication with the upper layers to accurately determine its position at complex intersections. By computing its priority sequence within a coordination circle, the AGV effectively treats the high-priority trajectories of other vehicles as dynamic obstacles for its local trajectory planning. The feasibility of trajectory information is ensured by solving the online real-time Optimal Control Problem (OCP). In the trajectory planning process for a single AGV, we incorporate a linear programming-based obstacle avoidance strategy. This strategy transforms the obstacle avoidance optimization problem into trajectory planning constraints using Karush-Kuhn-Tucker (KKT) conditions. Consequently, seamless and secure AGV movement within the port environment is guaranteed. The global planning module encompasses a global regulatory mechanism that provides each AGV with an initial feasible path. This approach not only facilitates complexity decomposition for large-scale problems, but also maintains path feasibility through continuous real-time communication with the upper layers during AGV travel. A key advantage of our progressive solution lies in its flexibility and scalability. This approach readily accommodates extensions based on the original problem and allows adjustments in the overall problem size in response to varying port cargo throughput, all without requiring a complete system overhaul.

Keywords: port automation; cooperative control; port AGV planning; trajectory optimization; KKT conditions

MSC: 46N10; 37N40; 13P25

check for updates

Citation: Chen, Y.; Shi, S.; Chen, Z.; Wang, T.; Miao, L.; Song, H. Optimizing Port Multi-AGV Trajectory Planning through Priority Coordination: Enhancing Efficiency and Safety. *Axioms* **2023**, *12*, 900. https://doi.org/10.3390/axioms12090900

Academic Editors: Haichuan Yang and Chaofeng Zhang

Received: 25 August 2023
Revised: 18 September 2023
Accepted: 19 September 2023
Published: 21 September 2023

1. Introduction

Ocean container transport plays a pivotal role in facilitating the global movement of goods, with standardized container transport emerging as the predominant method in modern ocean logistics. Modern ports have introduced a plethora of unmanned automated equipment, including automated cranes, unmanned container trucks (AGV), and unmanned cranes. These innovations significantly enhance the efficiency of container loading and unloading processes while reducing labor costs, eliminating uncertainties associated with human involvement in these operations. Automated terminal yard management systems, container tracking systems, and other technologies form the cornerstone of port digitalization and intelligence. Nevertheless, the rapid unloading and transfer of goods within confined areas, such as port terminals, pose significant challenges and act as constraints on the swift turnaround of container ships. This often leads to extensive queues of container ships waiting outside the port for unloading. In this paper, we delve into the efficient transshipment of containers within ports to expedite the turnaround of container ships. Our focus centers on two critical aspects: formulating an optimal global path for individual AGVs and implementing a cooperative priority strategy for multiple AGVs. To achieve optimal path planning, our approach harnesses augmented graph search techniques to ensure that the generated paths maintain a safe distance from obstacle corners. Notably, this approach eliminates the need for employing Voronoi diagrams (as depicted in Figure 1, right), thereby reducing computational overhead. Furthermore, we synchronize the sequence of AGVs entering the coordination zone with their proximity to the destination point, enabling the determination of AGV priorities during local planning within the complex region. This strategic framework proactively mitigates local deadlocks and significantly enhances the overall efficiency of task completion.

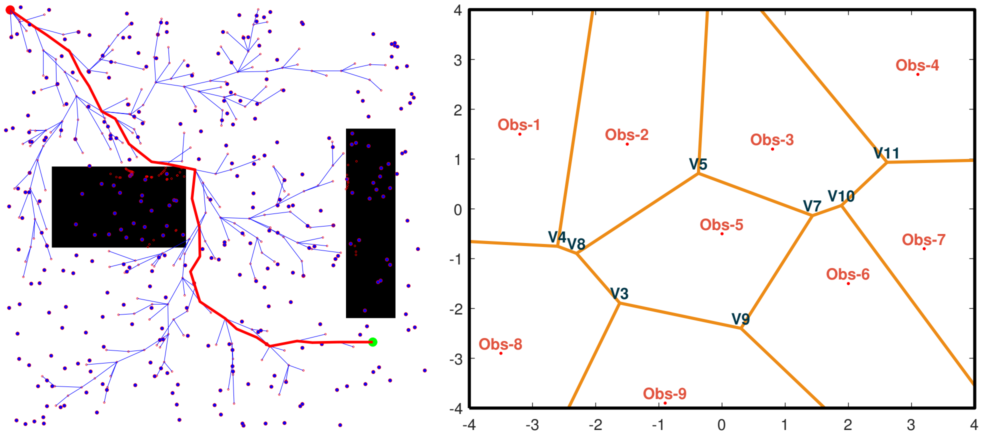


Figure 1. Illustration of the path generated by RRT* (**left**), where the red line represents the path generated by the RRT* algorithm, while the blue lines represent the branches of the exploring random tree. The red points indicate the random sampling points used in the algorithm, (**right**) Voronoi Graph, the yellow lines represent the feasible edges generated by the Voronoi Graph, while the red points indicate the obstacles in the environment.

1.1. Related Work

In the realm of single AGV or vehicle motion planning, the grid map serves as the foundational planning space, given that the environment is known and the locations of static obstacles, such as container yards, remain fixed. Graph search, a prevalent approach, stands out for minimizing planning time and enhancing solution efficiency. A frequently adopted strategy is the bi-level planning approach [1,2], wherein global path planning determines the overarching motion direction, while local optimization refines specific motion behaviors and tracking trajectories. Global path planning employs the A* algorithm [3], grounded in heuristic functions. However, the paths generated by A* may occasionally appear convoluted [4], and navigating around obstacle corners can prove challenging, as depicted in Figure 2. Consequently, post-processing techniques such as obstacle inflation [5] or path smoothing might be requisite for this algorithm. A more sophisticated approach is the Hybrid A* search algorithm [6], which integrates the robot's kinematic attributes into the A* algorithm, yielding smoother paths conducive to robot tracking through the application of Reeds-Shepp (RS) curves. Nevertheless, it does not suitability for omnidirectional robots. In contrast, the Probabilistic Roadmap (PRM) algorithm [7] and the Rapidly Exploring Random Tree (RRT) algorithm [8,9] represent sampling-based techniques adept at sidestepping the complexity imposed by high-dimensional spaces, as demonstrated in Figure 1 (left). Despite their merits, they are not devoid of limitations, including sluggish solution efficiency, paths characterized by multiple turning points, and challenges in yielding optimal solutions. Path planning using polynomial curves [10], Bézier curves, and spline curves [11] contributes to the attainment of smoother paths. Nevertheless, these algorithms falter in autonomously identifying paths within unknown environments from starting points. On extended paths, the concatenation of multiple curve segments becomes requisite. As a result, polynomial curves, Bézier curves, and related techniques predominantly find application in local path planning for robots. Beyond focusing on paths in proximity to the globally designated trajectory, local path planning necessitates accounting for dynamic obstacle avoidance and precise control of robot motion. To this end, numerical optimization [12–14] proves valuable in generating control inputs directly trackable by the robot, via the formulation of an objective function alongside constraints anchored in robot kinematics and initial position. However, numerical optimization entails substantial computational demand, rendering real-time execution for robot tracking intricate. Furthermore, it mandates the administration of the robot's motion through a lower-level tracking algorithm. Moreover, several widely used local planning algorithms, such as the Dynamic Window Approach (DWA) [15,16], come into play. Nevertheless, DWA's drawback lies in intricate parameter tuning, limited efficacy in dynamic obstacle avoidance, and suboptimal path selection. The Time-Elastic Band (TEB) [17,18], by manipulating the path via the treatment of obstacles as external forces given a starting point, yields viable paths. While it excels in dynamic obstacle avoidance, its solution efficiency is compromised due to the necessity for hypergraph construction and graph optimization tools like "g2o". This, in turn, leads to fluctuation-prone control inputs, causing the robot's motion to become unstable. Furthermore, approaches like artificial potential fields [19,20] and swarm intelligence-based algorithms, such as the ant colony algorithm (ACA) [21] and particle swarm optimization (PSO) [22], find extensive employment. Nevertheless, these methods are susceptible to local optima, and hard to find a more appropriate solution.

During the collaborative motion planning of multiple AGVs within a port terminal, each individual AGV can be conceptualized as an independent entity. Examined through this lens, the multi-AGV scheduling quandary can be reformulated into an integer planning problem [23]. However, with the escalation of problem magnitude, the emergence of computational complexity escalation in intricate scenarios becomes a conspicuous challenge. In existing literature [24], the cooperative predicament faced by each vehicle within conflict-prone zones is transformed into discrete scheduling quandaries, effectively partitioning the overarching complex problem. Although this simplification alleviates the problem-solving complexity burden, it doesn't inherently contribute to the enhancement of the overall road network. Intelligent body planning [25,26], on the other hand, envisions the terminal's intelligent body strictly for the purpose of tracking tasks, without factoring in the influence of the port AGV's autonomous tracking. Moreover, this approach bypasses the terminal's decision-making and planning mechanisms. In analogous vein, certain methodologies [27,28] approach the problem as a mixed-integer quadratic programming quandary, invoking formation-based approaches and sequential quadratic programming. Regrettably, these strategies are marred by suboptimal processing efficiency. Furthermore, an alternative avenue involves employing reinforcement learning [29–31], swarm intelligence algorithms [32–34], and game theory as planning mechanisms. However, these approaches grapple with challenges such as intensive computation and marginal impact on the overall solution effectiveness.

1.2. Our Contribution

In the context of global planning, our methodology capitalizes on an enhanced graph search algorithm for the purpose of deriving globally feasible paths. Remarkably, we incorporate a penalty mechanism for corner points of obstacles within the search path. This strategic decision serves a dual purpose: first, it curbs the over-occupancy of the free space beyond what can be achieved with conventional obstacle unfolding approaches. Second, it guarantees that the path, while inherently shorter, maintains a significant separation from the obstacles, effectively remaining close to the central region of the unobstructed space. This judicious approach helps us circumvent the challenges posed by Voronoi paths [35] established via the Voronoi diagram. This specific type of path tends to cling too closely to obstacles, resulting in elongated trajectories, as portrayed in Figure 1 (right). In addition, we introduce a preferential strategy predicated on the coordination circle. With this strategy, distinct priority sequences are assigned to individual AGVs, ensuring both efficiency in overall task completion in the presence of a large number of AGVs and mitigation of deadlock scenarios within locally complex road conditions. This tactical approach serves to optimize large-scale AGV operations while simultaneously addressing the complexities posed by localized environments.

In AGV local planning, we obtain the trajectory information of the underlying trajectory by simplifying the AGV model with Ackerman structure and planning the local trajectory using a simple kinetic model. In the local planning of obstacle avoidance, we adopted the method of obstacle avoidance based on the linear planning approach [36], using the KKT condition, which converts the optimized obstacle avoidance conditions into the constraints for trajectory optimization.

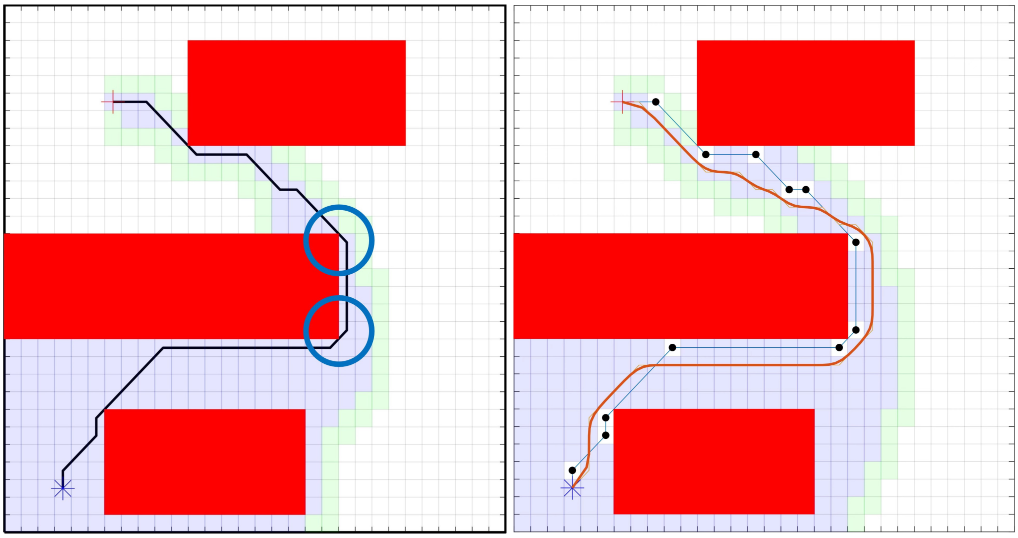


Figure 2. Illustration of the Path that generated by the A* algorithm (**left**) and Augmented Graph Search (**right**). Where the blue star and red cross demonstrate the start and goal point, respectively. and the black line (**left**) and the thin blue line (**right**) are the result of the A* algorithm, the red line is the result of the enhanced graph search, and the black dots are the virtual obstacles added in the enhanced graph search.

2. Problem Formulation

When orchestrating global path planning for the coordinated movement of multiple AGVs within port transportation, the resulting trajectory must skillfully circumvent a spectrum of obstacles. These encompass both static impediments like container yards and dynamic hindrances including other AGVs and pedestrians. Simultaneously, the devised path should endeavor to be the most direct route to the intended destination, all the while ensuring minimal interaction among AGVs. This intricate interplay of factors necessitates the attainment of a finely-balanced solution. Furthermore, the trajectory generated should also prioritize smoothness, aiming to minimize abrupt maneuvers. Upon delving into the lower echelons of local planning, it becomes imperative to factor in the specific kinematic constraints inherent to each robot's geometric attributes. This meticulous consideration aids in curbing tracking errors of the tracking unit, thus elevating the overall path tracking precision. The conceptual underpinning of this paper is encapsulated in the schematic delineated in Figure 3. And the defination of the model in our work is shown in Figure 4.

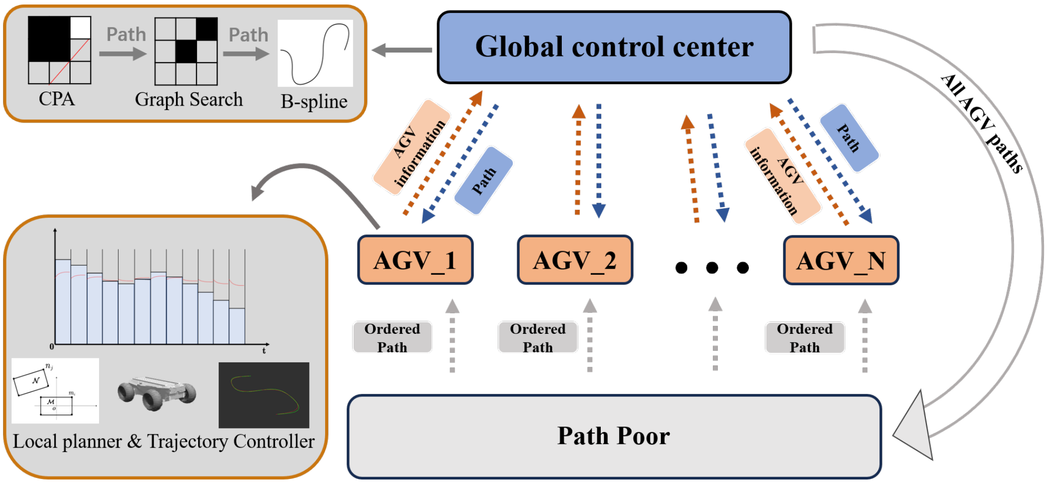


Figure 3. Illustration of Augmented Graph-search-based Motion Planning Framework.

Numerous methodologies have been put forth to facilitate seamless interaction between AGVs or vehicles and their surroundings. This interaction empowers them to discern obstacles and other environmental intricacies, leading to insights into their operational status. To exemplify, in the study by Zhu et al. [37], artificial intelligence techniques are harnessed to discern environmental anomalies. Conversely, Zhang et al. [38] harness the power of deep learning to prognosticate vehicle motion. In our study, we make the assumption that interactions involving dynamical obstacles are confined to local knowledge rather than spanning a global domain. In our proposed multi-AGV planning system for container transfers in port terminals, unmanned operations play a crucial role. In standardized transfer terminals, only AGVs are responsible for moving and transferring containers within the terminals. Therefore, we do not need to extensively consider the impact of non-AGV dynamic obstacles on the operations of these intelligent terminals. However, in our proposed multi-AGV planning coordination system, we can solve the problem of dynamic obstacle avoidance between non-AGVs, such as sudden appearance of pedestrians, animals, vehicles, etc., by obtaining the pose and speed of dynamic obstacles, and using additional prediction modules, these dynamic obstacles can be avoided easily through high-frequency planning and control. From the perspective of the AGVs themselves, dynamic obstacles, such as those detected by on-board lidar and visual cameras, can be effectively identified and tracked. In addition, the position and motion of the robot model can be determined by an array of sensors within the simulation environment. These encompass encoders, GPS systems, and Inertial Measurement Units (IMU).

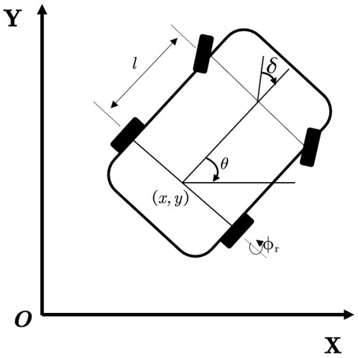


Figure 4. Illustration of pose variables of the Robot.

2.1. Enhanced Graph Search

Here the green arrow donates the start point, the red arrow is the end point, the red line is the path the generated by A*, the black pix represents the obstacles, and the light blue area means the obstacle expansion space. In the A* algorithm, a heuristic function is used as a guide for the search direction. By continuously updating the heuristic with the minimum heuristic value, a feasible path can be found on a shorter path. However, by this kind of approach, the path searched may appear to be close to obstacles, as shown in Figure 2 (left) the blue circle and Figure 5. While the distance between the path and the obstacle can be increased by expanding the obstacle, this approach also compresses the active space further. Therefore, we propose penalizing the inflection point of the path obtained from the A* algorithm. For example, we can increase its weight to make the inflection point more difficult to search or add its position to the CloseList in the A* algorithm directly. This approach can also produce paths away from obstacles, while ensuring shorter paths. Furthermore, compared with Figure 2 (left), it can better solve the problem of avoiding corner points of obstacles in graph search algorithms such as A*, as shown in Figure 2 (right). Based on the path obtained by the augmented graph search algorithm, we smooth the original curve with a B-spline curve and smooth the path by adjusting the weights of the B-spline curve to reduce the curvature at the critical points of the obstacle corners.

In Figure 2 (left) the blue star is the start point, the red cross means the end point, the red rectangles are the obstacles, the black line represents the path that generated by the A*, the light blue area donates the area that searched by A*, the light green area means the edge that the A* algorithm search in the next. In Figure 2 (right) the black points represent the penalized points, the blue line represents the original A* path, and the orange line represents the path generated by the augmented graph search algorithm.

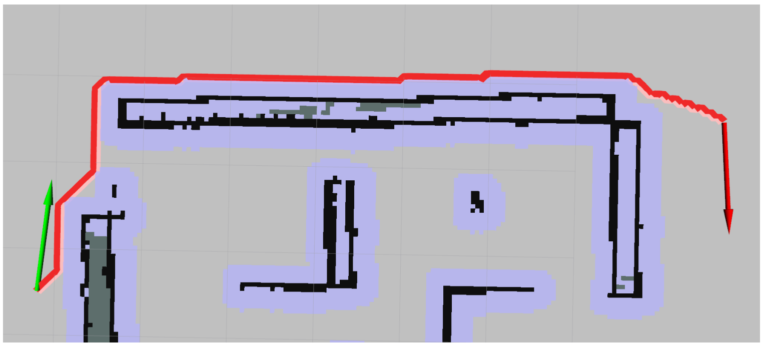


Figure 5. Illustration of the Path that generated by the A* algorithm in Rviz. Where the blue and red arrow demonstrate the start and goal pose.

While paths obtained through graph searches ensure trajectories free from collisions, the subsequent smoothing of these paths can inadvertently introduce new collision points, as exemplified in Figure 6 (left). To address this concern, we propose an adaptive fitting point approach, visually outlined in Figure 6 (right). In this innovative approach, we employ an iterative progressive algorithm for each pair of contiguous fitting points. This algorithm systematically identifies potential collision points along curves connecting these adjacent fitting points. Our collision detection process is guided by the principles articulated in references [39]. Should collisions be identified within the interval delimited by the two fitting points $[t_j, t_{j+1}]$, we strategically introduce a new fitting point precisely at the midpoint of the original path spanning from t_j to t_{j+1}. Subsequently, we re-evaluate for collisions. The inclusion of these supplementary fitting points is governed by the intrinsic properties of the B-spline curves. It mainly affects the immediately adjacent path segments, thus leaving the original collision-free path intact.

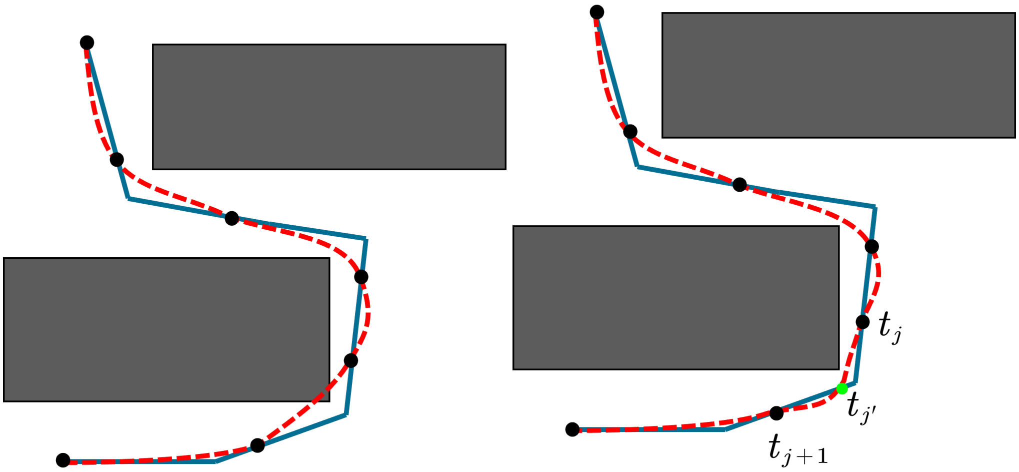


Figure 6. Illustration that collision happens during the interval $[t_j, t_{j+1}]$. We add new fit point $t_{j'} \in [t_j, t_{j+1}]$, and modified spline is collision free in $[t_j, t_{j+1}]$.

In motion planning, the lower level local path planning serves as a direct planning module for the robot to make specific behavioral decisions. In the present work, the global path provides constant guidance from the robot's starting point to the end point, while local planning is based on the global path and utilizes numerical optimization to derive specific trajectory information. The motion control of robots is often limited to a particular speed to ensure safety and avoid collisions between individual robots, while maintaining the cost advantage of logistics. To this end, we use an Ackermann-structured robot cart with front-wheel steering as our simulation model.

2.1.1. Mobile AGV Model

In AGV planning and control, the prevailing approaches often rely on simplified kinematic single-track models. In this context, we have harnessed the dynamical model inherent to our simulations. It is important to note that opting for a more complex model, such as a two-track dynamics model or a high-dimensional dynamics model, amplifies the computational load associated with tracking and planning, thereby hindering real-time local tracking optimization. In order to strike a balance between accuracy and computational efficiency, we have chosen to utilize a low-dimensional, simple dynamic model for our robot. This model is derived from a variation of a reduced Lagrangian approach [40]. The visual representation of our warehouse robot model is depicted in Figure 4. The configuration space of this model is denoted as $Q = SE(2) \times S^1 \times S^1$, comprising local coordinates $\mathbf{q} = [x, y, \theta, \phi_\mathrm{r}, \delta_f]$. Here, $[x, y, \theta]$ represent the robot's position and orientation, ϕ_r denotes the rolling angle of the robot's rear wheels, and δ_f signifies the angle of the front wheel's steering. The vertical mass and inertia of the robot are denoted as m and I_z, while the inertia of the rear wheel is designated as I_r. Our assumption posits that the robot is subject to control inputs $u = [\tau, u^\sigma]$, encompassing the torque τ applied to the rear wheel and the front wheel's steering angle u^σ. The Lagrangian formulation that encapsulates the robot model is formulated as follows:

$$L(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \frac{1}{2}\left(m\left(\dot{x}^2 + \dot{y}^2\right) + I_z\dot{\theta}^2 + I_\mathrm{r}\dot{\phi}_\mathrm{r}^2\right) \tag{1}$$

The nonholonomic constraints can be expressed as:

$$\begin{gathered} -\sin\theta \mathrm{d}x + \cos\theta \mathrm{d}y = 0 \\ \mathrm{con}\theta \mathrm{d}x + \sin\theta \mathrm{d}y = r_w \mathrm{d}\phi_\mathrm{r} \\ \mathrm{d}\theta = \frac{r_w \tan\left(\delta_f\right)}{l}\mathrm{d}\phi_\mathrm{r} \end{gathered} \tag{2}$$

where l is the wheelbase of the model, the r_w is the radius of the robot's wheel.

The Lagrangian (2) and it's nonholonomic constraints (3) are invariant, the continuous equations of the robot model can be expressed as fellows:

$$\begin{aligned} \dot{x} &= r_w \cos\theta\dot{\phi}_\mathrm{r} \\ \dot{y} &= r_w \sin\theta\dot{\phi}_\mathrm{r} \\ \dot{\theta} &= \frac{r_w\sigma}{l}\mathrm{d}\phi_\mathrm{r} \\ \left(I_r + mr_w^2 + \frac{I_z\sigma^2 r_w^2}{l^2}\right)\dot{u}^{\phi_\mathrm{r}} &= \tau - \frac{2I_z r_w^2\sigma\dot{\sigma}\dot{\phi}_\mathrm{r}}{l^2} \\ \dot{\sigma} &= u^\sigma \end{aligned} \tag{3}$$

where the $\sigma = \tan\delta_f$, and the value of I_r is too small, which can be ignored in practice.

2.1.2. Obstacle Avoidance

In the optimization process of local trajectory generation, it is important to consider, among other things, obstacle avoidance for dynamic obstacles. Common collision detection algorithms such as the Gilbert-Johnson-Keerthi (GJK) algorithm [41], Expanding Polytope Algorithm (EPA) algorithm [42], and area method have limitations such as nonlinearity and nonfineness, making it difficult to achieve optimal results in the numerical optimization process. This presents an obstacle to the solution of nonlinear programming. To address this, we use a linear programming-based collision-free avoidance constraint, which is integrated into the lower-level optimization model using Karush-Kuhn-Tucker (KKT) conditions [43].

We assume that the warehouse robot, obstacles, etc., can be enclosed by a convex polygon border, as shown in Figure 7. For two convex polygons $\mathcal{M} = [m_1, \ldots, m_{tm}]$ and $\mathcal{N} = [n_1, \ldots, n_{tn}]$, where$\{m_i\}_{i=1}^{tm}$ and $\{n_j\}_{j=1}^{tn}$ are the vertices of the convex polygons $\mathcal{M}$ and $\mathcal{N}$, respectively. The collision-free property of arbitrary convex polygons can be constructed as follows:

$$\begin{aligned} J(\mathcal{M},\mathcal{N}) \ &= \ \min \sum_{\mathrm{k}=1}^{p+2} \mathcal{Z}_\mathrm{k} \\ s.t.\ \mathcal{M}A - \mathcal{N}B + \mathcal{Z} &= 0 \\ \sum_{i=1}^{tm} a_i + \mathcal{Z}_{p+1} \ &= 1 \\ \sum_{j=1}^{tn} b_j + \mathcal{Z}_{p+2} \ &= 1 \\ a_i \geqslant 0, b_j \geqslant 0, z_p &\geqslant 0 \end{aligned} \tag{4}$$

where $a_i \in A$, $b_i \in B$.

Especially,

$$J > 0 \Longleftrightarrow \mathcal{M} \cap \mathcal{N} = \oslash \tag{5}$$

When the origin of the coordinate system is inside $\mathcal{M}(O \in \mathcal{M})$, the Equation (2) can be simplified as fellows,

$$\begin{aligned} J(\mathcal{M},\mathcal{N}) &= \min 1 + \varepsilon^T P \\ s.t.\ \mathcal{Q}P = \mathcal{T},\ 1 + \varepsilon^T P &\geqslant 0 \\ P &\geqslant 0 \end{aligned} \tag{6}$$

where $\mathcal{Q} = \begin{bmatrix} \mathcal{M} & \mathcal{N} \\ 1 & 0 \end{bmatrix}, \mathcal{T} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, P = \begin{bmatrix} A \\ B \end{bmatrix}, \varepsilon = \begin{bmatrix} 0 & -1 \end{bmatrix}$

Where $\mathcal{M}$ and $\mathcal{N}$ are each composed of multiple convex polyhedra, the non-touch constraint is presented as follows:

$$J = \min_{A_i \in A, B_j \in B} J\left(A_i, B_j\right) \geqslant \gamma > 0 \tag{7}$$

where $A = \{A_1, A_2, \cdots A_u\}$, $B = \{B_1, B_2, \cdots B_v\}$, and γ is the relaxation factor. Figure 7 shows an example of two convex polygons: rectangle $\mathcal{M}$ and rectangle $\mathcal{N}$ that are collision-free.

Equation (2) or (4) serves as a constraint in local planning and is presented in an optimal form. In order to integrate it into higher level optimization, the KKT optimally conditions are applied. However, the optimal problem with KKT conditions is difficult to solve, as the strict equation constraints often lead to infeasible or local optimal points.

To improve the efficiency of optimization, the KKT conditions are combined with a relaxation condition [44] and an equilibrium is applied. The Equation (4) with the relaxation KKT condition can be expressed as follows:

$$\begin{aligned} \left\| -\varepsilon - \kappa + \mathcal{Q}^{\mathrm{T}} \lambda \right\| &\leqslant \sqrt{\epsilon} \\ \kappa^{\mathrm{T}} P &\leqslant \epsilon \end{aligned} \tag{8}$$

where the κ and λ are the Lagrange multipliers, the ϵ is the margin parameter.

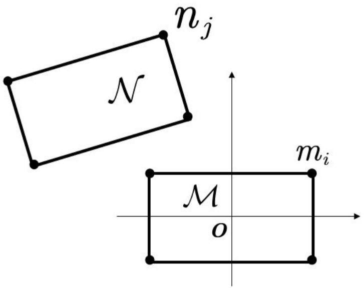


Figure 7. Illustration of collision avoidance of two convex polygons $\mathcal{M}$ and $\mathcal{N}$.

2.1.3. Numerical Method

When tracking the route provided by global planning, the local planner must provide a feasible trajectory from the local initial position to the local terminal position based on the global path. In the continuous-time domain, the objective function comprises two parts—Mayer type and Lagrange type [45]. In this paper, we do not consider the effect of the final state position and orientation on the objective function. Instead, we incorporate the soft constraint in the form of weight values to limit it in the objective function. We formulate the objective function using a time-optimal integration of the acceleration in the continuous-time domain. Thus, the local programming problem can be formulated as follows:

$$\begin{aligned} &\min_{t, q(t), u(t)} \omega_1 t_f + \omega_2 r\left(q(t_0), q\left(t_f\right)\right) + \omega_3 \int_{t_0}^{t_f} g(u(t)) dt \\ &subjectto \quad \begin{cases} \dot{q}(t) = f(t, q(t), u(t)) \mathcal{J}(q(t)) < 0 \\ \mathcal{J}(q(t)) < 0 \\ \mathcal{S}(t_0, q(t_0)) = 0 \\ q^l \leqslant q(t) \leqslant q^u \\ u^l \leqslant u(t) \leqslant u^u \end{cases} \end{aligned} \tag{9}$$

Here, $\{\omega_i\}_{i=1}^3$ are the weighting factors, and $r(\cdot)$ is the error function of the end-state and global path. The function $g(\cdot)$ represents the amount of change of the control quantity, while the first constraint represents the system kinematics constraints. The second denotes the collision-free constraint, and thrid represents the initial restrictions of the system. Additionally, the last two constraints are the state and control limits of the system.

The above NLP problems can be solved using various software, such as CasADi [46] and CppAD [47], which are based on nonlinear programming solvers such as IPOPT [48] and SNOPT [49].

In this study, we have adopted the multiple shooting method [50] as the primary approach to tackle the optimal control problem. This technique revolves around finding a problem at a particular time point, effectively splitting the time span into a large number of smaller intervals. Within each of these intervals, both state and control variables are treated as independent free entities. This transforms the system of ordinary differential equations (ODE) into a substantial nonlinear equation system, which can subsequently be addressed using a nonlinear planning solver. To ensure continuity between state trajectories, matching constraints are introduced. The continuity of the state variables is refined meticulously by successive iterations of the algorithm. This iterative process is instrumental in driving the objective function towards its optimal value. It is essential to acknowledge that the accuracy of the solution of the problem is profoundly affected by the chosen size of the solution interval. While a smaller interval may enhance the accuracy of the solution, it may also escalate the complexity associated with solving the problem. Striking the right balance between solution accuracy and computational complexity is crucial when employing this approach.

3. Single AGV Simulation

In this part, we utilized numerical simulations to evaluate the performance of the single local planner. The objective function and its constraints are shown in Equation (6), and the model parameters are presented in Table 1. The local trajectory optimization problems are solved using open-source software CasADi with the nonlinear programming solver IPOPT on Ubuntu 20.04. Additionally, we present experiments in the simulation environment Gazebo under ROS.

Table 1. Parameters of Simulation Model.

Symbol	Description	Value	Unit
l	wheelbase	0.65	m
r	wheel radium	0.330	m
l_w	axle track	0.605	m
m	weight	70	kg
v_{max}	max speed	1.5	m/s
R_{max}	minimum turning radius	1.6	m
I_z	moment of inertia in Z axis	51.64	$\text{kg} \cdot \text{m}^2$

3.1. Numerical Simulation

In Figure 8, the initial and terminal poses of the warehouse robot model is $[x_0, y_0, \theta_0] = [-5, 1.05, 0]$ and $[x_f, y_f, \theta_f] = [5, 1.05, 0]$. The obstacle is a rectangle, which is formed by $(-2,-2), (2,-2), (2,2), (-2,2)$. In Figure 9, the initial and terminal poses of the model is $[x_0, y_0, \theta_0] = [-5, 1.25, 0]$ and $[x_f, y_f, \theta_f] = [5, 1.25, 0]$, the obstacle is a circle, which radius is $\sqrt{2}$, the center is $(0,0)$. Two scenarios involving obstacle avoidance are simulated numerically: rectangular and circular obstacle avoidance with the J-function ensuring collision-free trajectories. The details are presented below:

In Figure 10, the above figures show the states and control values changing with time t, avoid rectangle obstacle shown in Figure 8 and circle obstacle shown in Figure 9 are presented in Figure 10a,b. Here, θ, u^σ, σ use the left ordinate and τ uses the right ordinate.

The algorithm governing local planning is outlined in Algorithm 1. The local planner for a singular AGV embarked on an integrated loop, meticulously traversing each step of the local planning process. The sequence commences with the publication of the pose information for each AGV to the central coordinate (i.e., the global planner). Subsequently, the algorithm evaluates whether the AGV in question is entering the local coordination circle. This is accomplished by monitoring the IN_{Circle} signal. If the AGV enters the local coordination circle, the local planner calculates its priority within the circle. In addition, it acquires trajectory points pertaining to AGVs with higher priority and includes them as obstacles within its own local trajectory planning. In the subsequent phase, the algorithm orchestrates the derivation of the local trajectory for the current AGV by solving an Optimal Control Problem (OCP). Once the local trajectory is acquired, the control signal is dispatched to the AGV and the trajectory of the current AGV is added to the Path pool. This marks the conclusion of the current iteration, and the algorithm proceeds to update the pose and weight (w_{eig}) of the current AGV before embarking on the subsequent loop iteration. This cyclic process continues iteratively, with each AGV undergoing the same sequence of operations. The algorithm thus facilitates the seamless coordination of AGVs within the local environment, ensuring dynamic adaptation and adherence to priority-based trajectory planning.

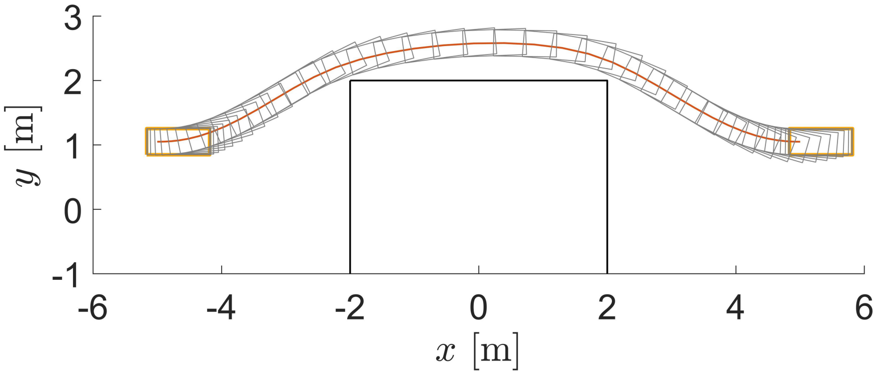


Figure 8. Illustration of numerical simulation of obstacle avoidance using J-function under rectangular obstacle.

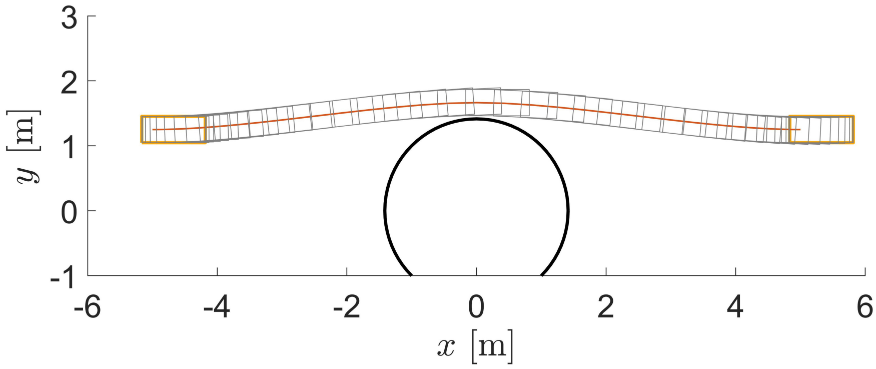
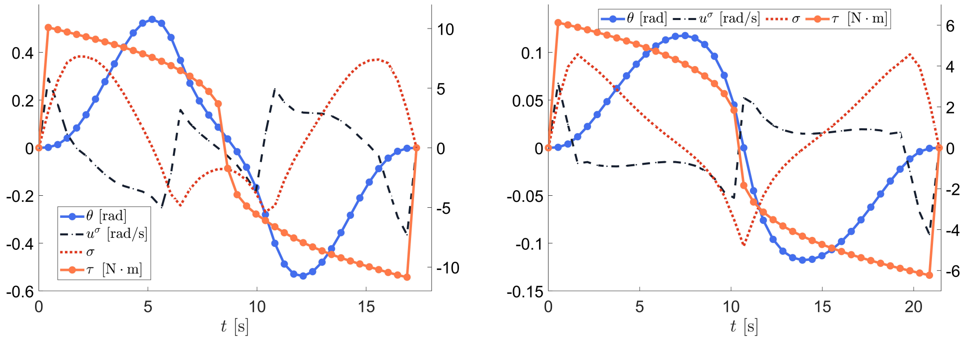


Figure 9. Illustration of numerical simulation of obstacle avoidance using J-function under circle obstacles.



(**a**) (**b**)

Figure 10. Variation curve of heading angle θ with time s with J obstacle avoidance. (**a**) Rectangle collision avoidance. (**b**) Circle collision avoidance.

Algorithm 1: Single AGV local planning

```
Data: AGV position $pose$, signal of whether in coordination circle $In_{Circle}$, vehicle
      and environments constraints $c$, global path of $i_t h$ AGV $p_{g,i}$ (where $_g$
      represent the global, $_i$ denote the number of AGVs.), grid-based map $\mathcal{F}^{\mathcal{G}}$,
      function to get the sequence of other AGVs in the coordination circle $f_{obs}(\cdot)$,
      function of getting obstacles position by $O_{seq}$ $f_{getobs}(\cdot)$, function of getting
      the local trajectory $f_{OCP}(\cdot)$
Result: Trajectory $T_l$
while $True$ do
    Publish $pose$ and $w_{eig}$ to Center
    if $In_{Circle}$ then
        $p_{order} \leftarrow Pri(p_{g,i})$ ;   // get the order of priority in current AGV.
        $O_{seq} \leftarrow f_{obs}(p_{order}, pose, \mathcal{F}^{\mathcal{G}})$ ;   // get other AGV number in Circle.
        $Obs \leftarrow f_{getobs}(O_{seq}, pose, \mathcal{F}^{\mathcal{G}})$;   // get obstacles position.
    end
    $T_l \leftarrow f_{OCP}(pose, p_{g,i}, obs, \mathcal{F}^{\mathcal{G}}, c)$;   // get the tracking trajectory by
     solving OCP.
    $AGV \leftarrow C_{unit}(T_l)$ ;   // Control AGV movement by control unit.
    $Path\ Pool \leftarrow T_l$;
    Update $pose$ and $w_{eig}$;
    $sleep()$;
end
// Algorithm ends here
```

3.2. Single AGV Gazebo Simulation

We conducted further experiments in the Gazebo simulation environment, utilizing both the global and local planners. The overall process is shown in Figure 11, while the tracking trajectories and global planning paths are displayed in Figure 12.

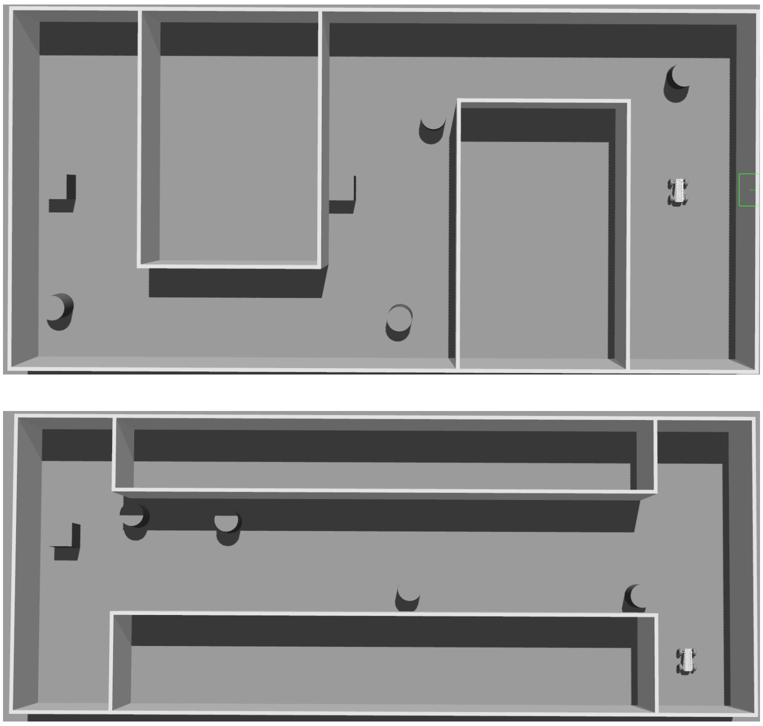


Figure 11. Simulation in S and L environments was conducted under Gazebo. The static obstacle is a wall, and cuboids and spheres were randomly added as dynamic obstacles.

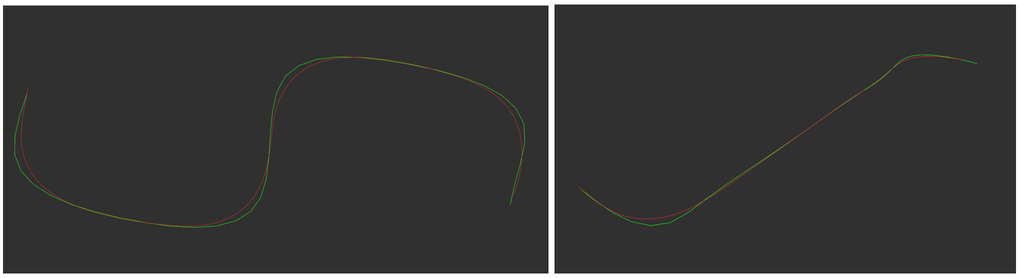


Figure 12. Simulation in S and L environments using Rviz, where the green lines represent the global paths and the red lines represent the actual tracking trajectories produced by the warehouse model.

4. Multi AGVs Global Planning

During the port container transportation, we use the single AGV local planning Algorithm 1 as a local planner for multi-AGV cooperative planning and control in the port, and by using the global cooperative control central, the AGVs at the intersection are processed in a synchronous and coordinated manner. The algorithm details are shown in Algorithm 2.

In the initial phase, the global planner performs an exhaustive evaluation of all AGVs along with their respective starting points. Subsequently, an initial global path denoted as P_{init} is allocated to each AGV through graph-based searching. Building upon this foundation, a static obstacle-avoidant path P is derived via Enhanced graph searching, ensuring its deviation from obstacle corners. Finally, the acquired path P undergoes smoothing through B-spline curve techniques, resulting in a more refined trajectory. Once all port AGVs initiate motion, cyclic position detection is performed for each AGV. This process determines whether or not an AGV enters the coordination circle. After the circle entry, the AGV priority is established based on their individual weight coefficients, corresponding to their order of circle entry. This information informs the update of the global path details associated with each AGV. In the context of this study, the priority calculation adheres to the following approach:

$$\omega = c_1 O_{rder} + c_2 w_{eig} \tag{10}$$

where ω is the final evaluation value for prioritization by the coordination circle, O_{rder} and w_{eig} are the values of the order of entering the coordination circle and the distance of this AGV from the end point, respectively, and c_1 and c_2 correspond to their weight coefficients.

The path sorting function $f_{o,order}(\cdot)$ operates by ranking in descending order based on the ω value. This strategy, even when disregarding the order of circle entry, plays a crucial role in handling the intersection of trajectories between any two AGVs. If a situation arises where the planned trajectories of two AGVs intersect, the AGV further away from its destination is given higher priority. As a result, it is given precedence in its local planning phase. Moreover, its projected trajectory points are incorporated as dynamic obstacle data in the planning process of the AGV with lower priority. This strategic integration not only enhances the overall mission efficiency, but also guarantees the successful mission completion of AGVs. In addition, this prioritization strategy safeguards against potential deadlocks at intricate intersections among multiple AGVs. Moreover, the inclusion of the O_{rder} component from circle entry ($Order_i$) in Equation (10) effectively reduces the time individual AGVs spend within the circle, subsequently curtailing time overhead at complex intersections.

Besides, we designed two scenarios to demonstrate the global coordinate center algorithm. All simulations are constructed in ROS with Rviz, record the simulation information and replay in Matlab, the details are follows.

Algorithm 2: Global coordinate circle center

```
Data: The number of AGV $N$, grid-based map $\mathcal{F}^{\mathcal{G}}$, initial and goal position $P_{*,s}$,
      $P_{*,g}$, AGVs' position $pose$, Path weight $w_{eig}$, graph search function $\mathcal{G}_{\mathcal{F}}(\cdot)$,
      Enhanced graph search function $\mathcal{G}^*_{\mathcal{F}}(\cdot)$, B-spline function $\mathcal{S}_B(\cdot)$, function of
      judging whether AGV in coordinate circle $f_{dispatch}(\cdot)$, path sorting function
      $f_{o,order}(\cdot)$.
Result: Sequence of path $\boldsymbol{S}$.
for $i \leftarrow 0$ to $N$ do
    $P_{init} \leftarrow \mathcal{G}_{\mathcal{F}}(\mathcal{F}^{\mathcal{G}}, P_{*,s}, P_{*,g})$;          // Get Enhanced global path.
    $P \leftarrow \mathcal{G}^*_{\mathcal{F}}(P_{init}, \mathcal{F}^{\mathcal{G}}, P_{*,s}, P_{*,g})$ ;      // Get initial global path by graph
     search.
    $P_g \leftarrow \mathcal{S}_B(S_i)$
    $\boldsymbol{S} \leftarrow S_i$
end
while $True$ do
    // Cyclic detection of each AGV.
    for $i \leftarrow 0$ to $N$ do
        $Circle_i \leftarrow f_{dispatch}(pose_i, \mathcal{F}^{\mathcal{G}}, )$;        // Judge whether current AGV in
         coordinate circle.
        if $Circle_i$ then
            $prio \leftarrow f_{p,order}(pose_i, \mathcal{F}^{\mathcal{G}}, w_{eig})$;   // Get priority of current AGV.
            Update $\boldsymbol{S}$;
        end
    end
end
$Pathpool \leftarrow \boldsymbol{S}$
// Algorithm ends here.
```

4.1. Port Overview Scenes

The Overview of the port is shown in Figure 13, And the trajectories that all the AGVs tracking are shown in Figure 14. Where the green dots are the start points of AGV, the light brown points demonstrate the current position of AGVs and the red dots represent the goal position.

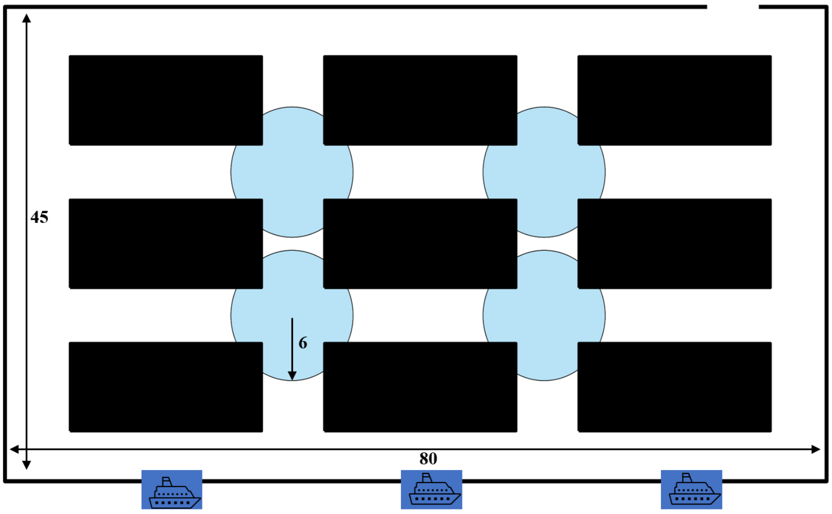


Figure 13. Port overview scenes. Where the black block demonstrate the container yard, the shadow blue circle is the coordinate circle, the dark blue block represents the container ship berth, the gap in the upper right corner is the exit from the port area.

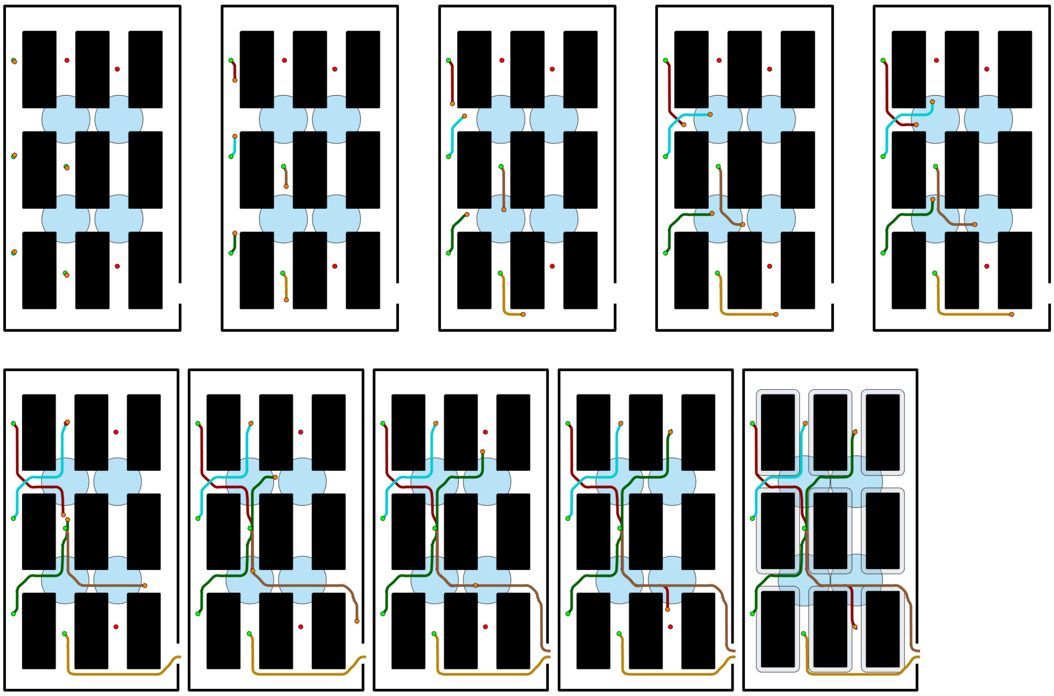
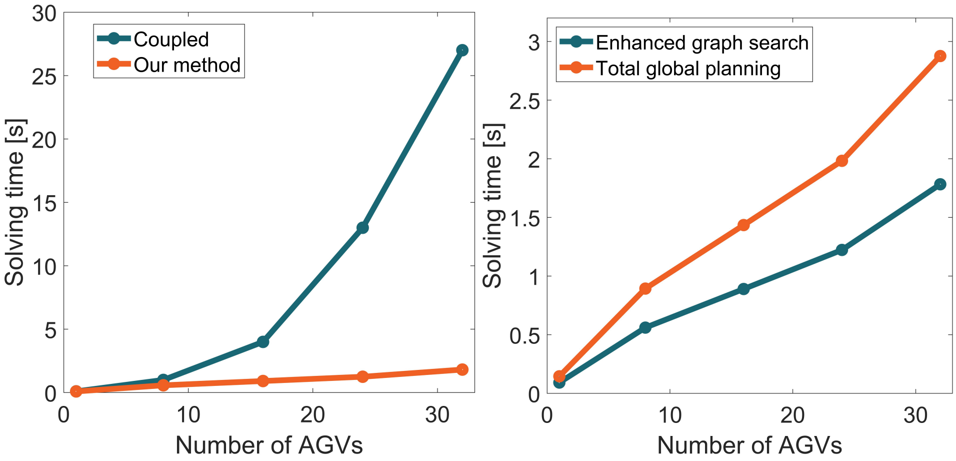


Figure 14. Multi AGVs coordinate transportation in port scenes.

During the intra-port transportation, we evaluate the performance of our proposed method, the comparison of our method and group coupled method is shown in Figure 15a. While the coupled approach will compute all the exercisable trajectories before the AGV tracks them, our proposed approach can start the action with an initial feasible solution and move closer to the optimal solution in a continuous motion without additional time waiting for the computation. The total global calculation time and Enhanced graph searched time are shown in Figure 15b.



(**a**) (**b**)

Figure 15. Comparison of our proposed method and coupled method [51] and the total time in global planning. (**a**) Comparison of our method and coupled method. (**b**) Total planning and Enhanced graph search time.

We work to bring these paths closer to global optimality through the combined efforts of the local trajectory planner, controller, and coordination circle during the travel process. In [51], any addition or removal of an intelligent AGV body necessitates considering teaming and overall problem scheduling. Moreover, any change in the scenario requires reconfiguring the problem. In contrast, the semi-decoupled approach we have adopted can quickly adapt to changes in the number of AGVs and can be readily deployed in various environments, offering flexibility and efficiency.

4.2. Local Intersections Scenes

During the double intersections scenes that shown in Figure 16, five different port AGV vehicles entered the inner part of the coordinate circle below at the same time, and through the algorithms we employed above, the AGVs were given different planning priorities, allowing for smooth and trouble-free passage even in complex traffic environments.

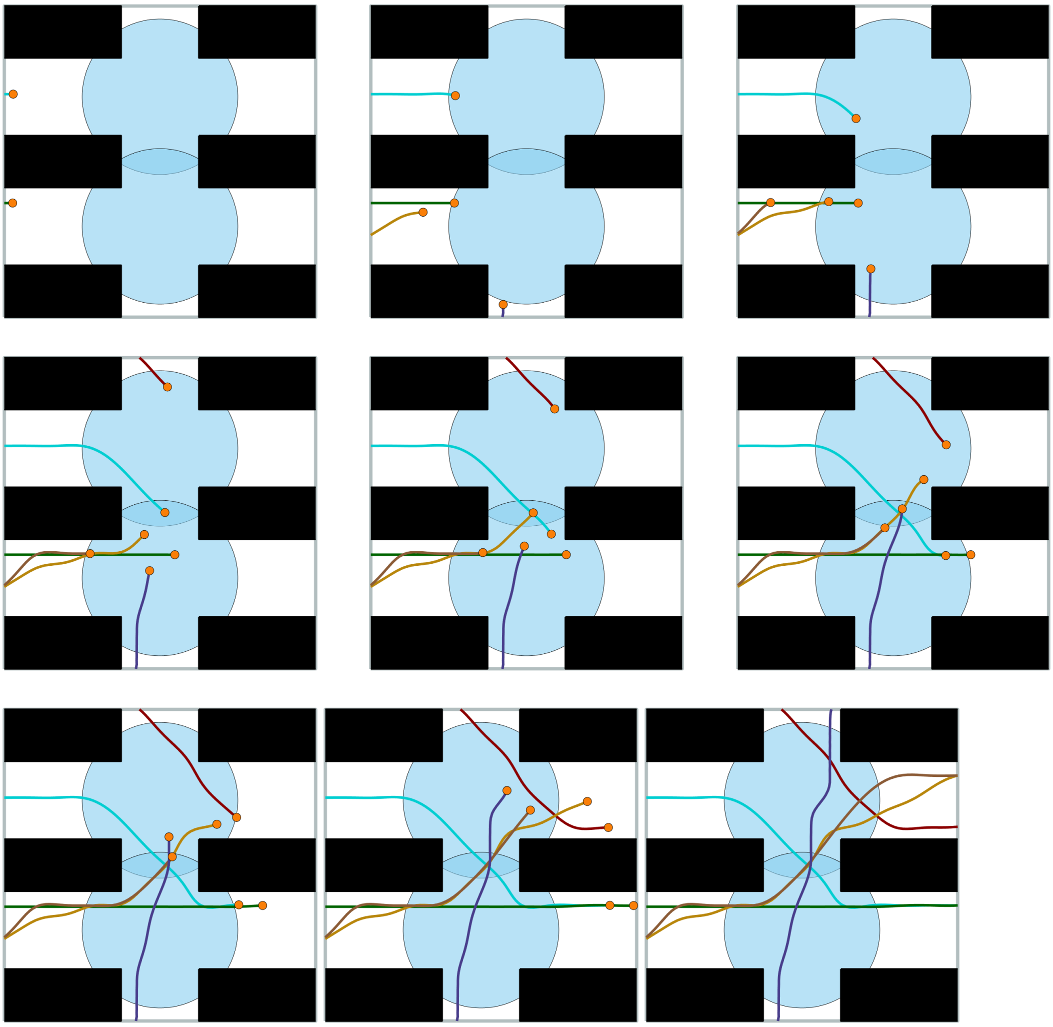


Figure 16. Multi AGVs coordinate transportation in double intersections scenes.

5. Conclusions and Future Work

In this work, our primary focus is on addressing the challenge of fast container movement within an intelligent port terminal. The goal is to increase the efficiency of container loading and unloading, ultimately leading to faster turnaround times for container ships. We achieve this by carefully planning the efficient flow of AGV vehicles within the port terminals through a combination of fast global planning for multiple AGV systems, local planning at the onboard terminals, and the introduction of a coordination circle. To overcome the inherent challenges associated with graph search paths that are too close to obstacles, we introduce a novel enhanced graph search approach within the global motion planning framework. This approach involves imposing a penalty or treating corner points as obstacles, effectively allowing us to avoid these issues and ignore basic graph search paths that might lead too close to obstacles. We anchor our approach in the A* search algorithm or other graph search algorithms, ensuring enhanced spirituality in path lengths. In addition, we incorporate the concept of a global planning-based coordination circle into our methodology. This approach greatly simplifies AGV navigation within complex traffic junctions by prioritizing movement within the coordination zone. In the context of local planning, we expedite the derivation of feasible tracking trajectories by quickly solving nonlinear programming (NLP) problems through a linear programming-based obstacle avoidance technique. Our simulation experiments that combine global planning-based coordination with local planning demonstrate robust performance in intra-port transportation scenarios. We also incorporate a global planning module with global regulation to provide AGVs with initial feasible paths. This module consistently offers local planning solutions by continuously solving Optimal Control Problems (OCP) during AGV travel. Real-time communication between the global planning module and AGVs allows for dynamic prioritization at complex traffic junctions. This prioritization takes into account the proximity to the destination and the order of entry into the coordination zone, effectively reducing the likelihood of deadlock scenarios and ensuring overall efficiency in container transport operations within the port environment. In our future work, we plan to apply multi-AGV systems in real-world ports and factories and consider the impact of forked devices [52] on the planning and control of AGV carriers during various maneuvering operations, such as high-speed travel. In addition, we aim to develop accurate models for vehicle maneuver stability to better meet the practical needs of plant and industrial settings.

Author Contributions: Conceptualization, Y.C., S.S. and Z.C.; methodology, Z.C., S.S. and Y.C.; software, L.M., H.S. and T.W.; validation, Y.C., T.W. and L.M.; formal analysis, Y.C. and S.S.; investigation, Y.C., S.S., T.W., Z.C. and L.M.; resources, Y.C. and S.S.; data curation, L.M., T.W., H.S. and Z.C.; writing—original draft preparation, Y.C.; writing—review and editing, S.S.; visualization, L.M. and T.W.; supervision, Y.C. and S.S.; project administration, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China (grant number 2021YFB1600400), and the National Natural Science Foundation of China (grant number 52101403).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Latombe, J.C. *Robot Motion Planning*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 124.
2. LaValle, S. *Planning Algorithms*; Cambridge University Press: Cambridge, UK, 2006.
3. Stocco, D.M. A StAR search: Implications in controlling steroidogenesis. *Biol. Reprod.* **1997**, *56*, 328–336. [CrossRef] [PubMed]
4. Tseng, F.H.; Liang, T.T.; Lee, C.H.; Der Chou, L.; Chao, H.C. A star search algorithm for civil UAV path planning with 3G communication. In Proceedings of the 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kitakyushu, Japan, 27–29 August 2014; pp. 942–945.
5. Deits, R.; Tedrake, R. Computing large convex regions of obstacle-free space through semidefinite programming. In Proceedings of the Algorithmic Foundations of Robotics XI: Selected Contributions of the Eleventh International Workshop on the Algorithmic Foundations of Robotics, Istanbul, Turkey, 3–5 August 2015; pp. 109–124.
6. Dolgov, D.; Thrun, S.; Montemerlo, M.; Diebel, J. Practical search techniques in path planning for autonomous driving. *Ann Arbor* **2008**, *1001*, 18–80.
7. Kavraki, L.E.; Svestka, P.; Latombe, J.C.; Overmars, M.H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.* **1996**, *12*, 566–580. [CrossRef]
8. Kuffner, J.J.; LaValle, S.M. RRT-connect: An efficient approach to single-query path planning. In Proceedings of the 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), San Francisco, CA, USA, 24–28 April 2000; Volume 2, pp. 995–1001.
9. Zhang, L.; Shi, X.; Yi, Y.; Tang, L.; Peng, J.; Zou, J. Mobile Robot Path Planning Algorithm Based on RRT_Connect. *Electronics* **2023**, *12*, 2456. [CrossRef]
10. Farouki, R.; Koenig, T.; Tarabanis, K.; Korein, J.; Batchelder, J. Path planning with offset curves for layered fabrication processes. *J. Manuf. Syst.* **1995**, *14*, 355–368. [CrossRef]
11. Qian, L.; Xu, X.; Zeng, Y.; Li, X.; Sun, Z.; Song, H. Synchronous maneuver searching and trajectory planning for autonomous vehicles in dynamic traffic environments. *IEEE Intell. Transp. Syst. Mag.* **2020**, *14*, 57–73. [CrossRef]
12. Werling, M.; Kammel, S.; Ziegler, J.; Gröll, L. Optimal trajectories for time-critical street scenarios using discretized terminal manifolds. *Int. J. Robot. Res.* **2012**, *31*, 346–359. [CrossRef]
13. Brito, B.; Floor, B.; Ferranti, L.; Alonso-Mora, J. Model predictive contouring control for collision avoidance in unstructured dynamic environments. *IEEE Robot. Autom. Lett.* **2019**, *4*, 4459–4466. [CrossRef]
14. Lin, X.; Zhang, J.; Shen, J.; Fernandez, G.; Hong, D.W. Optimization based motion planning for multi-limbed vertical climbing robots. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), The Venetian Macao, Macau, 4–8 November 2019; pp. 1918–1925.

15. Fox, D.; Burgard, W.; Thrun, S. The dynamic window approach to collision avoidance. *IEEE Robot. Autom. Mag.* **1997**, *4*, 23–33. [CrossRef]
16. Teso-Fz-Betoño, D.; Zulueta, E.; Fernandez-Gamiz, U.; Saenz-Aguirre, A.; Martinez, R. Predictive dynamic window approach development with artificial neural fuzzy inference improvement. *Electronics* **2019**, *8*, 935. [CrossRef]
17. Rösmann, C.; Feiten, W.; Wösch, T.; Hoffmann, F.; Bertram, T. Efficient trajectory optimization using a sparse model. In Proceedings of the 2013 European Conference on Mobile Robots, Barcelona, Spain, 25–27 September 2013; pp. 138–143.
18. Rösmann, C.; Feiten, W.; Wösch, T.; Hoffmann, F.; Bertram, T. Trajectory modification considering dynamic constraints of autonomous robots. In Proceedings of the ROBOTIK 2012: 7th German Conference on Robotics, Munich, Germany, 21–22 May 2012; pp. 1–6.
19. Choset, H. *Robotic Motion Planning: Potential Functions*; Robotics Institute, Carnegie Mellon University: Pittsburgh, PA, USA, **2010**.
20. Ji, Y.; Ni, L.; Zhao, C.; Lei, C.; Du, Y.; Wang, W. TriPField: A 3D potential field model and its applications to local path planning of autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 3541–3554 . [CrossRef]
21. Dorigo, M.; Di Caro, G. Ant colony optimization: A new meta-heuristic. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 6–9 July 1999; Volume 2, pp. 1470–1477.
22. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
23. Ashtiani, F.; Fayazi, S.A.; Vahidi, A. Multi-intersection traffic management for autonomous vehicles via distributed mixed integer linear programming. In Proceedings of the 2018 Annual American Control Conference (ACC), Milwaukee, WI, USA, 27–29 June 2018; pp. 6341–6346.
24. Mahbub, A.I.; Malikopoulos, A.A.; Zhao, L. Decentralized optimal coordination of connected and automated vehicles for multiple traffic scenarios. *Automatica* **2020**, *117*, 108958. [CrossRef]
25. Sharon, G.; Stern, R.; Felner, A.; Sturtevant, N.R. Conflict-based search for optimal multi-agent pathfinding. *Artif. Intell.* **2015**, *219*, 40–66. [CrossRef]
26. Li, J.; Ruml, W.; Koenig, S. Eecbs: A bounded-suboptimal search for multi-agent path finding. In Proceedings of the the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 2–9 February 2021; Volume 35, pp. 12353–12362.
27. Mellinger, D.; Kushleyev, A.; Kumar, V. Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, St Paul, MN, USA, 14–18 May 2012; pp. 477–483.
28. Augugliaro, F.; Schoellig, A.P.; D'Andrea, R. Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 1917–1922.
29. Lee, D.; Kim, J.; Cho, K.; Sung, Y. Advanced double layered multi-agent Systems based on A3C in real-time path planning. *Electronics* **2021**, *10*, 2762. [CrossRef]
30. Males, L.; Sumic, D.; Rosic, M. Applications of Multi-Agent Systems in Unmanned Surface Vessels. *Electronics* **2022**, *11*, 3182. [CrossRef]
31. Qiu, H. Multi-agent navigation based on deep reinforcement learning and traditional pathfinding algorithm. *arXiv* **2020**, arXiv:2012.09134.
32. Teodorovic, D. Transport modeling by multi-agent systems: A swarm intelligence approach. *Transp. Plan. Technol.* **2003**, *26*, 289–312. [CrossRef]
33. Wu, J.P.; Peng, Z.H.; Chen, J. 3D multi-constraint route planning for UAV low-altitude penetration based on multi-agent genetic algorithm. *IFAC Proc. Vol.* **2011**, *44*, 11821–11826. [CrossRef]
34. Nowé, A.; Vrancx, P.; De Hauwere, Y.M. Game theory and multi-agent reinforcement learning. In *Reinforcement Learning: State-of-the-Art*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 441–470.
35. Bhattacharya, P.; Gavrilova, M.L. Voronoi diagram in optimal path planning. In Proceedings of the 4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD 2007), Pontypridd, UK, 9–11 July 2007; pp. 38–47.
36. You-lun, X.; Han, D. General Criterion And Control Strategy Of Collision-free Movement For Manipulators. In Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics, Beijing and Shenyang, China, 8–12 August 1988; Volume 2, pp. 783–786.
37. Zhu, Y.; Tang, H. Automatic Damage Detection and Diagnosis for Hydraulic Structures Using Drones and Artificial Intelligence Techniques. *Remote Sens.* **2023**, *15*, 615. [CrossRef]
38. Zhang, M.; Taimuri, G.; Zhang, J.; Hirdaris, S. A deep learning method for the prediction of 6-DoF ship motions in real conditions. *Proc. Inst. Mech. Eng. Part J. Eng. Marit. Environ.* **2023**, 14750902231157852 . [CrossRef]
39. Pan, J.; Zhang, L.; Manocha, D. Collision-free and smooth trajectory computation in cluttered environments. *Int. J. Robot. Res.* **2012**, *31*, 1155–1175. [CrossRef]
40. Kobilarov, M. Discrete Geometric Motion Control of Autonomous Vehicles. Ph.D. Thesis, University of Southern California, Los Angeles, CA, USA , 2008.
41. Glass, K.; Colbaugh, R.; Lim, D.; Seraji, H. Real-time collision avoidance for redundant manipulators. *IEEE Trans. Robot. Autom.* **1995**, *11*, 448–457. [CrossRef]

42. Van Den Bergen, G. *Collision Detection in Interactive 3D Environments*; CRC Press: Boca Raton, FL, USA, 2003.
43. Boyd, S.; Boyd, S.P.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.
44. Dutta, J.; Deb, K.; Tulshyan, R.; Arora, R. Approximate KKT points and a proximity measure for termination. *J. Glob. Optim.* **2013**, *56*, 1463–1499. [CrossRef]
45. Betts, J.T. Survey of numerical methods for trajectory optimization. *J. Guid. Control. Dyn.* **1998**, *21*, 193–207. [CrossRef]
46. Andersson, J.A.; Gillis, J.; Horn, G.; Rawlings, J.B.; Diehl, M. CasADi: A software framework for nonlinear optimization and optimal control. *Math. Program. Comput.* **2019**, *11*, 1–36. [CrossRef]
47. Bell, B.M. CppAD: A package for C++ algorithmic differentiation. *Comput. Infrastruct. Oper. Res.* **2012**, *57* .
48. Wächter, A.; Biegler, L.T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.* **2006**, *106*, 25–57. [CrossRef]
49. Andrei, N.; Andrei, N. A SQP algorithm for large-scale constrained optimization: SNOPT. In *Continuous Nonlinear Optimization for Engineering Applications in GAMS Technology*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 317–330.
50. Morrison, D.D.; Riley, J.D.; Zancanaro, J.F. Multiple shooting method for two-point boundary value problems. *Commun. ACM* **1962**, *5*, 613–614. [CrossRef]
51. Xu, H.; Feng, S.; Zhang, Y.; Li, L. A grouping-based cooperative driving strategy for CAVs merging problems. *IEEE Trans. Veh. Technol.* **2019**, *68*, 6125–6136. [CrossRef]
52. Rinchi, M.; Pugi, L.; Bartolini, F.; Gozzi, L. Design of control system to prevent forklift capsize. *Int. J. Veh. Syst. Model. Test.* **2010**, *5*, 35–58. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.