Prediction, Planning, and Coordination of Thousand-Warehousing-Robot Networks With Motion and Communication Uncertainties

Zhe Liu, Hesheng Wang, *Senior Member, IEEE*, Huanshu Wei, Ming Liu, *Senior Member, IEEE*, and Yun-Hui Liu, *Fellow, IEEE*

***Abstract*— In this article, we focus on resolving the traffic flow prediction, robot path planning, and motion coordination problems in large-scale warehousing robotics systems with thousand-robot networks. The warehousing environment is partitioned into several sectors, and a hierarchical framework is developed, which includes a centralized prediction and planning level and a decentralized local coordination level. In the centralized level, a traffic flow prediction algorithm is first proposed to predict the evolution of the robot density distribution in a future horizon and estimate the future traffic heat value of each sector. Based on this, the sector-level robot path can be generated in the time-expended sector graph by comprehensively considering the traveling distance and the predicted traffic heat value and will be dynamically updated by considering the most recent traffic information. In the coordination level, local cooperative A* algorithm, incorporated with the conflict-based searching strategy, is implemented within each sector to generate conflict-free road-level paths for all the robots in the sector simultaneously, and the rolling planning scheme is utilized in order to immediately react to robot motion uncertainties and communication disconnections. The effectiveness and practical applicability of the proposed approach are validated by large-scale simulations with more than one 1000 robots and real laboratory experiments.**

***Note to Practitioners*—Considering practical situations and requirements in industrial warehouses and automated logistics systems, this article resolves the life-long planning and coordination problems of large-scale robot networks and ensures the practical execution performance in the presence of robot motion uncertainties and temporary communication disconnections. Our main idea is to reduce robot congestions and improve warehouse working efficiency by balancing the traffic flow in the whole environment. To achieve this, we present a traffic flow prediction algorithm to estimate the robot density distribution in a future horizon and take this information into consideration in sector-level path planning. The reliability, scalability, and the real-time performance of the proposed solution are achieved by the presented hierarchical system framework and the dynamic planning scheme. The proposed concept and approach can also be used to coordinate other large-scale systems with multirobot or multi-AGV networks. Simulation and experimental results suggest that the proposed solution is effective and practically applicable, but a saturation phenomenon of the system capacity can be observed under a very heavy workload. In the future, we will investigate the relation between the maximum system capacity and the environment structure and make further efforts to optimize the environment structure and road layout in order to improve the warehouse working efficiency.**

***Index Terms*—Hierarchical planning and coordination, large-scale robot network, traffic prediction, uncertainty.**

I. Introduction

RECENTLY, mobile robots are widely implemented in various logistics applications, such as the automatic parcel sorting in robotic warehouses, pickup and delivery in unmanned storage systems, cargo transportation in autonomous container piers, and the mail service in office buildings [1]–[7]. Replacing human workers with the robotic system contributes to reducing labor costs, improving warehouse working efficiency, and increasing the reliability and scalability of the warehousing system. In order to give full play to the potential and efficiency advantage of the robotic system and split the cost, practicalrobotic warehousing systems usually contain a large number of mobile robots. In such large-scale systems, the real-time path planning and robust motion coordination are two of the most challenging problems, the performance of which will greatly affect the efficiency and reliability of the overall warehousing system. In this article, we focus on the life-long path planning and motion coordination problems of the large-scale robotic networks, aiming for the practical warehousing and logistics applications (such as the system shown in Fig. 1).

At present, lots of algorithms are presented in the literature [1], [4], [6]–[9] to solve the multirobot path planning and motion coordination problems; however, there exists a

Manuscript received April 18, 2020; revised June 10, 2020; accepted July 25, 2020. Date of publication August 19, 2020; date of current version October 6, 2021. This article was recommended for publication by Associate Editor L. Xia and Editor Q. Zhao upon evaluation of the reviewers' comments. This work was supported in part by the Natural Science Foundation of China under Grant U1613218, Grant 61722309, and Grant U1913204; in part by the National Key Research and Development Program of China under Grant 2018YFB1309300; and in part by the Beijing Advanced Innovation Center for Intelligent Robots and Systems under Grant 2019IRS01. *(Corresponding author: Hesheng Wang.)*

Zhe Liu is with the Department of Computer Science and Technology, University of Cambridge, Cambridge CB2 1TN, U.K. (e-mail: zl457@cam.ac.uk).

Hesheng Wang is with the Department of Automation, Institute of Medical Robotics, Key Laboratory of System Control and Information Processing of Ministry of Education, Key Laboratory of Marine Intelligent Equipment and System of Ministry of Education, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the Beijing Advanced Innovation Center for Intelligent Robots and Systems, Beijing Institute of Technology, Beijing 100811, China (e-mail: wanghesheng@sjtu.edu.cn).

Huanshu Wei and Yun-Hui Liu are with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: weihuanshu94@hotmail.com; yhliu@cuhk.edu.hk).

Ming Liu is with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong (e-mail: eelium@ust.hk).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2020.3015110

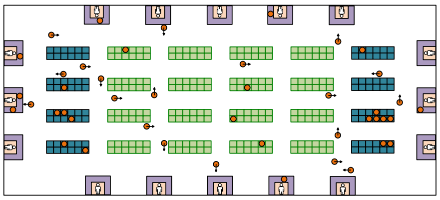


Fig. 1. Example of the robotic warehousing system, where the blue, green, and purple grids represent the robot stations, task pickup stations, and working stations, respectively. Robots are commanded to transport cargos from pickup stations to working stations while avoiding motion conflicts with each other.

dichotomy between the academic researches and industry implementations and several open challenges that have not been well-resolved yet. First, as opposed to the traditional multirobot pathfinding problem [8], the practical robotic warehousing system operates in a life-long manner. How to build a well-constructed environment/a well-formulated problem is of great importance in such life-long planning and coordination instance since directly utilizing a well-established one-shot multirobot planning algorithm in a life-long system may not always obtain the feasible solutions [10]. Second, in large-scale systems with hundreds (or possibly thousands) of robots, the unneglected traffic congestion often leads to the extreme decline of the warehouse operation efficiency and even causes robot deadlocks. Thus, the robot density distribution is expected to be balanced over the whole environment in order to alleviate traffic congestions and avoiding traffic shock waves [7]. How to consider the traffic flow equilibrium in the path planning stage is a very critical issue, especially in large-scale robot networks. Third, in practical systems, robot motion uncertainties (caused by the human–robot interaction, motion control accuracy, or hardware malfunctions) and temporary communication disconnections (caused by bandwidth limitations, route handoff, reestablishment operators, or system malfunctions) are unneglected and also unpredictable [11]. These uncertainties will greatly decrease the system performance and make the coordination problem very challenging. Finally, how to improve the real-time performance and ensure the system scalability are also challenging issues in the large-scale system.

We develop a systematic solution in this article to address the abovementioned issues. Our main contributions include the following.

1) Considering the structure of the practical robotic warehouses, we present a set of criteria to formulate a well-formed planning and coordination problem; based on which, the solvability of the life-long system can be ensured.
2) We present a hierarchical approach to exploit the advantages of both the centralized and decentralized methods. In the centralized level, a time-efficient traffic flow prediction algorithm is presented to estimate the evolution of the robot density distribution in a future horizon. The most recent traffic predictions will be utilized to generate the sector-level path for each robot and dynamically adjust it. These manners aim to ensure the traffic flow equilibrium in the whole environment and guarantee real-time performance. In the decentralized level, the local cooperative A* algorithm is utilized in the reduced spatial–temporal road map within each sector to generate collision-free local paths for all the robots in the sector simultaneously. What is more, the rolling planning strategy is used to improve the system tolerance to uncertainties, and a K-step redundancy mechanism is designed to handle communication failures.
3) To the best of our knowledge, this article is the first to solve the coordination problem of large-scale networks with more than 1000 robots in the presence of motion and communication uncertainties. Large-scale simulations and laboratory experiments are presented to validate the scalability, robustness, and practical applicability of the proposed solution.

A. Related Work

Multirobot path planning and motion coordination issues have been studied actively for more than 20 years. Existing approaches can be mainly classified into the centralized [8] and decentralized methods [12]. Generally speaking, centralized methods mainly put their effort into finding the optimal solution and decreasing the computation complexity, while decentralized methods mainly focus on increasing system flexibility, fulfilling real-time requirements, and resolving local deadlocks.

Centralized methods include the reduction-based algorithm [8] and the A*-based algorithm [6]. The former one transforms the multirobot pathfinding problem into well-solved mathematical programming problems (such as the time-expanded network flow problem [8]) and calculates the optimal planning result by utilizing the existing algorithms (such as the integer linear programming algorithm). However, in most cases, reduction-based algorithms are only effective in solving small-scale problems with the makespan optimization index [9], and the problem transformation processes are also computationally expensive. The decoupled hierarchical cooperative A*-based methods [13] are promising in solving large-scale problems as they have good real-time performances although they cannot guarantee the optimal result. Recently, searching tree structures [9], [11], [14] are utilized to decrease the searching complexity in traditional cooperative A* algorithms and improve the optimality. However, since the computational complexity of these methods is much larger than the decoupled hierarchical cooperative A*, their performances in solving large-scale planning problems have not been demonstrated. In order to solve this problem, in our previous work [15], we develop a hierarchical framework for large-scale robot networks that utilize the searching tree-based cooperative A* [9] to ensure the local motion coordination performance. However, traffic flow prediction information has not been considered, and communication uncertainties have also been neglected. More recently, path diversification-based heuristics [16] are introduced into decoupled cooperative A* algorithm, and traffic balance is considered in order to decrease path conflicts. However, the routing performance in complex environments with hundreds of robots decreases largely, and motion and communication uncertainties have not been considered.

Decentralized methods include the local coordination-based algorithm (off-line independent planning and online coordination) and the priority-based algorithm (off-line prioritized cooperative planning). Local coordination-based algorithms depend on well-designed local traffic rules and coordinate robot velocity vectors [17], regenerate local robot routes according to priorities [18], or reschedule the passing time [5], [12] to avoid local conflicts. Priority-based algorithms [1], [18] define a priority value for each robot. The path coordination is achieved in an incremental manner by the continuous replanning conducted on the low-priority robots. Decentralized methods are mostly incomplete, and the traffic jams and robot conflicts cannot be avoided in the crowded environment with a large number of robots. What is more, the high requirements for the robot perception and communication abilities also limit these methods' practical applicabilities. Recently, a hierarchical strategy is presented for the coordination of automated guided vehicles [7], and a probabilistic traffic model is proposed to predict the possibly congested areas [19]. However, the scalability of this method in large-scale problems has not been validated. Furthermore, since the future robot action is modeled by random movements, the proposed traffic predictions are not reliable. Compared with [7] and [19], our traffic flow prediction algorithm is more accurate and time-efficient, and the proposed approach can be used in large-scale planning and coordination problems with high uncertainties.

In recent years, learning-based approaches have attracted wide attention [20], [21], which provides fully decentralized policies and teach robots to reactively plan motions by observing local environmental information while coordinating with each other implicitly. However, learning-based approaches require high computational resources, and the training time can be as long as a few weeks [20]. In addition, the low success rate in crowded environments and the poor generalizability to large robot numbers also greatly limit their practical applicability. A comprehensive comparison can be found in [21], which demonstrates that the recent learning-based approaches still cannot achieve comparable performance with the traditional approaches [9], [13].

The abovementioned methods mostly focus on the one-shot problem, where all the tasks are predefined and each robot needs to deliver its allocated task while avoiding conflicts with other robots, and the system terminates once all robots have arrived at their destinations. However, for a life-long system, we need to well-formulate the problem to ensure the solvability first. What is more, since finding out the optimal solution is very difficult and time expansive in a dynamic system, improving the real-time planning performance and increasing the system reliability are more important. The life-long problem is solved in [10] with a token passing strategy and the cooperative A* algorithm, and recently, the windowed cooperative A* is further utilized with the frequent replanning strategy to solve the large-scale planning problem [23]. In addition, an online cooperative path planning method is presented in [24], which considers the predicted information of the robots and tasks that will appear in the future. However, these approaches have not taken communication and motion uncertainties into consideration. In order to ensure the practical applicability, the hierarchical planning-execution framework is studied by considering kinematic constraints [4] and arbitrary dynamic limits [22]. However, the introduction of the motion orders will greatly affect the system performance in the presence of motion uncertainties, and this framework cannot be used to solve the life-long planning problem of large-scale networks.

II. Well-Formulation of the Life-Long Problem

As shown in Fig. 1, we consider the warehouse environment that contains robot stations (such as the maintenance station or charging station in a warehouse), the task stations (such as the inventory pod area), and the working stations (such as the order packaging area or cargo delivery area). Consider a large group of mobile robots R_n, $n = 1, \ldots, N_r$, in which all the free robots stay in the robot station area and wait for the task assignments. Transportation tasks are online published one by one or in the batch model. Each transportation task has one pickup station and one working station. In real applications, the order management system, the task allocation system, and the robot management system are almost decoupled in order to ensure real-time performance and decrease the mutual influence between each system. Thus, in this article, we assume that there are high-level order management and task allocation systems to publish transportation tasks and assign them to proper free robots; then, we mainly focus on the robot path planning and motion coordination problems. Once a new task is allocated to a free transportation robot, the robot will move from the robot station to the task pickup station to load the cargo and then transport it to one working station along the planned path. In the meantime, the robot needs to avoid motion conflicts with other transportation robots. We consider the discrete environment model in this article and assume that, in time k, the location of robot R_n is $q^n(k)$, and then, two types of location constraints are considered in order to avoid collisions:

1) $\forall R_n, R_g, n \neq g, q^n(k) \neq q^g(k)$;
2) $\forall R_n, R_g, n \neq g, q^n(k+1) \neq q^g(k)$.

The second constraint is designed to avoid conflict in the presence of motion and communication uncertainties.

In the life-long system described earlier, the path planning and motion coordination problem may not always be solvable (a problem is solvable means that one can find out at least one solution to solve the problem) since a free robot that has accomplished its previous task may obstruct the movement of other robots that are assigned new tasks [10]. Robot motion and communication uncertainties in practical applications will further decrease the problem solvability. Thus, in this article, we define the following criteria to ensure the solvability. First, the warehouse environment should fulfill the following criteria.

C1: The number of robot stations is not less than the number of robots.

C2: Between any two stations (including the robot stations, task pickup stations, and working stations), there exists at least one path that traverses no other station.

Second, the assignment of a newly published task should fulfill the following criteria.

C3: For a new task, its corresponding task pickup station and working station should be different from those of all the unaccomplished tasks that have been assigned.

C4: If C3 cannot be fulfilled, the new task should be put into a waiting list to wait for the accomplishment of all the conflicting tasks (i.e., until C3 is fulfilled).

Third, the motion of all the robots should fulfill the following criterion.

C5: Once the assigned task is accomplished, the robot should either return to one of the free robot stations that have not been occupied by any other robot or move away from the working station of its previous task to the task pickup station area to accomplish a new task.

Based on the abovementioned criteria, we have the following result.

Proposition 1: The life-long multirobot path planning and motion coordination problem is always solvable, i.e., the problem is well-formulated, if the abovementioned criteria C1–C5 are fulfilled.

Proof: C1 and C2 ensure that the free robots have enough parking stations, and a free robot that stays in a robot station will not conflict with any other robot. C2–C4 ensure that a robot that is staying in a task pickup station (or a working station) and is preparing to pick up its assigned task (or is waiting to accomplish its assigned task) will not conflict with any other robot. C5 further ensures that a robot that has accomplished its task assigned before will not occupy the working station all the time and affect other robots to accomplish their tasks. The abovementioned analyses ensure that a robot that stays in any station will not conflict with any other robot. Then, in the worst case, the path planning and motion coordination problem can be solved by a priority-based approach, i.e., each robot has a priority value defined by its task assignment order, and the robot with the lower priority will stay in the robot station until all the robots with higher priorities have accomplished their tasks and returned to robot stations. In this way, robots can move one by one without any collision and accomplish their assigned tasks according to the task assignment order; thus, the life-long task planning and motion coordination problem is always solvable. ■

In this section, we present criteria C1–C5 to make sure that any problem generated by the following five criteria is solvable, i.e., we can find out at least one solution to solve this problem. Then, the abovementioned proposition gives the solution in the worst case. Please note that criteria C1–C5 and the abovementioned proposition only provide a sufficient condition to ensure the solvability; however, the actual path of each robot in the proposed system is not planned in this way.

III. Hierarchical Planning and Coordination Framework

A. Environment Model

In this article, the environment map is modeled as a hierarchical form. On the global level, we divide the whole environment into several partitions, and each partition is defined as a sector S_j. Then, at the local level, we build the road map within each sector and define adjacent exits and entrances between every two neighboring sectors. More specifically, we utilize the following environment partitioning criteria in the sector generation.

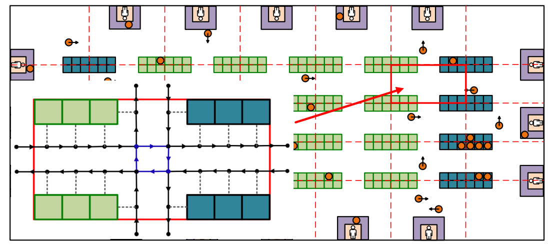


Fig. 2. Example of the road topology map in a sector, where the arrowed lines represent monodirectional road segments, the dotted lines represent the ramps to task pickup stations, and the four blue road segments in the middle of the sector form an intersection.

C6: In the local road map within each sector S_j, there is one and only one intersection.

C7: For two sectors that are neighbors, there is one and only one unidirectional entryway (includes the exit and entrance) from one sector to the other sector, and vice versa.

C8: In the road map within each sector S_j, from each entrance to each exit, there is at least one movable path located within this sector that does not traverse any station (including the robot stations, task pickup stations, and working stations) located in the sector.

The abovementioned environment partitioning criteria result in a sector topology graph G_s, where each vertex in G_s denotes a sector S_j, $j = 1, 2, \ldots, N_s$, and each edge in G_s represents the connection between two neighboring sectors. Although C7 implies that all the edges in G_s are bidirectional, we define two weights $w_{ij}(k) \neq w_{ji}(k)$ to represent the different traveling costs from sector S_i to S_j and from S_j to S_i in time k.

Within each sector, we build a local road map that satisfies the environment partitioning criteria C6–C8. Then, a whole road network can be generated by combining all the local road maps. The roads in all the sectors are divided into segments, and a road topology graph G_x is generated to describe the whole road network, where each vertex in G_x represents a place x_l, $l = 1, 2, \ldots, N_x$, each edge in G_x represents that there is a road segment from place x_l to x_p that does not travel through any other place, and the edge weight v_{lp} is set to the distance of the road segment. In order to improve the traffic speed and reduce robot congestions, all the road segments are defined as monodirectional roads, i.e., G_x is a directed graph. What is more, an index matrix $\Phi_{N_s \times N_x}$ is defined as follows: if $x_l \in S_j$, $\Phi_{jl} = 1$; otherwise, $\Phi_{jl} = 0$. Fig. 2 shows an example of the local road map in a sector.

B. Hierarchical Framework and System Structure

We present a hierarchical planning and coordination framework, as shown in Fig. 3, which consists of a centralized sector-level traffic flow prediction and dynamic sector path planning system, and a decentralized local road path planning and motion coordination system. The detailed algorithms implemented are shown in Fig. 4. In the high level, a traffic flow prediction algorithm is presented to estimate the robot

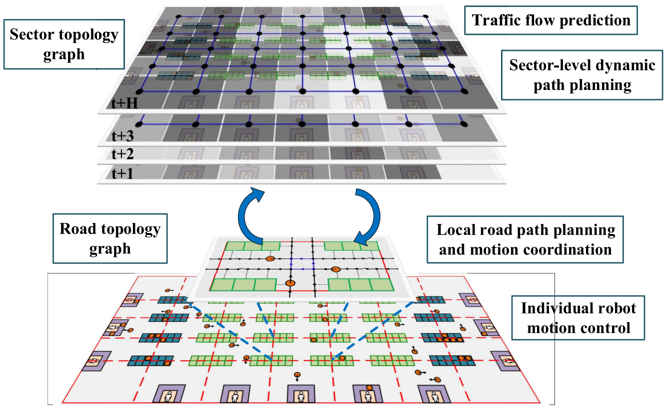


Fig. 3. Proposed hierarchical framework.

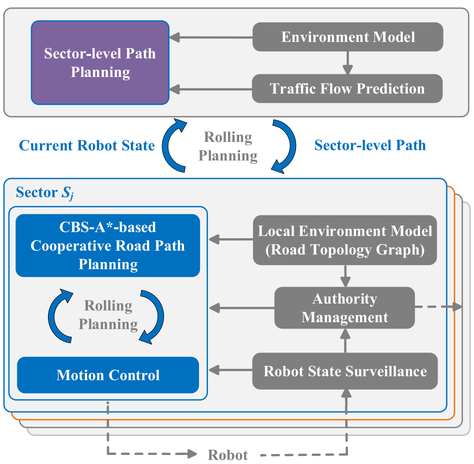


Fig. 4. System structure of the proposed approach.

density distribution in a future horizon. In the sector topology graph, the traffic prediction information will be considered in the traveling cost through a sector in order to estimate the practical traveling time. Then, the sector path of each robot will be planned on the time-expended sector graph, and a dynamic planning strategy will be utilized in order to take into account the most recent traffic information. These manners contribute to achieve the traffic flow equilibrium in the whole environment, improve the optimization performance of the robot sector path planning, and ensure the real-time performance in the centralized high level, simultaneously. In the low level, local cooperative A* algorithm in the local spatial–temporal road topology graph is utilized within each sector, and the conflict-based searching strategy (CBS) is implemented to increase the searching efficiency. Local road path planning and motion coordination are executed within each sector periodically on a reduced spatial horizon to immediately react to robot state changes in the sector and to improve the algorithm tolerance to robot motion delays and communication failures. What is more, an entryway reservation system is implemented to coordinate every two neighboring sectors and manage the authority to avoid any robot collision caused in sector switchings during robot movements.

IV. Traffic Flow Prediction and Sector-Level Dynamic Path Planning

In this article, our main idea is to improve the warehouse working efficiency and reduce robot congestions by balancing the traffic flow over all the sectors in the whole environment. Thus, we need to first predict the evolution of the traffic in a future horizon; based on which, the sector-level robot path can be generated and dynamically updated by comprehensively considering the traveling distance and the traffic information.

A. Traffic Flow Prediction

Given the current task assignments and planed robot paths, the evolution of the traffic in a future horizon can be predicted by utilizing a probabilistic model. To achieve this, we need to first predict the position probability distribution of each moving robot and estimate the number of robots in each sector at each time step. Based on which, the robot motion delay probability (due to local motion coordination reasons) in each sector at each time step can be estimated. Then, by repeating the abovementioned two steps, the robot density distribution over sectors in a future horizon and the evolution of the traffic flow can be predicted.

Since the road map has been partitioned into segments, without loss of generality, we assume that each road segment has a uniform traveling distance and can be tracked in a unit time. Then, robot movement along its planed path can be modeled in a discrete manner, i.e., at each time step, each robot can either travel through a road segment on its path (if no conflict occurs) or stay in the place where it is (if a conflict exists and the robot should coordinate its motion with others). Let $Q^n = [q_1^n, q_2^n, \ldots, q_i^n \ldots]$ be the road path of robot n, and the predicted probability that robot n is located in place q_i^n at time k can be modeled as

$$P_{q,i}^n(k) = P_{q,i}^n(k-1)\rho_{q,i}^n(k) + P_{q,i-1}^n(k-1)\left[1-\rho_{q,i}^n(k)\right] \tag{1}$$

where $k = 2, \ldots, H$ and H is the prediction horizon. The motion delay probability $\rho_{q,i}^n(k)$ represents the probability that robot n stays in its previous place, and if $q_i^n \in S_j$, we set $\rho_{q,i}^n(k) = \rho_j(k)$, where $\rho_j(k)$ denotes a uniform motion delay probability of all the robots in sector S_j at time k. Then, for a given place x_l in the road map, the predicted probability that place x_l is occupied by a robot at time k can be calculated as

$$P_{x,l}(k) = \sum_{n \mid q_i^n = x_l} P_{q,i}^n(k). \tag{2}$$

Define an occupancy vector $O(k) = [O_1(k), \ldots, O_{N_x}(k)]^T$, where $O_l(k) = 1$ if $P_{x,l}(k) > \epsilon$; otherwise, $O_l(k) = 0$, ϵ is an empirical threshold. Then, the number of robots in each sector can be estimated as

$$M(k) = \Phi O(k) \tag{3}$$

where $M(k) = [M_1(k), \ldots, M_{N_s}(k)]^T$, and $M_j(k)$ represents the number of robots in sector S_j at time k. Then, we define the motion delay probability of sector S_j in time step $k+1$ as

$$\rho_j(k+1) = \frac{M_j(k)}{C_j} \tag{4}$$

where C_j is the maximum allowable robot number of sector S_j, which can be defined to be linearly proportional to the size of movable area in sector S_j. In this article, we set $C_j = \Phi_j \mathbf{1}_{N_x}$, where $\Phi_j = [\Phi_{j,1}, \ldots, \Phi_{j,N_x}]$ is the jth row of Φ.

With the initial location information of each robot, we can directly count the number of robots in each sector and calculate the initial motion delay probability $\rho_j(1)$. Then, by repeating (1)–(4) from $k = 2$ to $k = H$, the robot density distribution over sectors in a future horizon H and the evolution of the traffic flow can be predicted. Please note that the robots that stay in robot stations without any assigned task will not be considered in the abovementioned traffic flow predictions.

The computational complexity of the abovementioned traffic flow prediction method can be estimated as $O(H^2 N_r)$ since, in each iteration k from $k = 2$ to $k = H$, for each robot n, we need to calculate the probabilities that robot n is located in place $q_1^1, q_1^2, \ldots, q_1^{\min\{k,|Q^n|\}}$. As a consequence, the upper bound of the complexity can be estimated as $O(H^2 N_r)$.

B. Traffic Flow Prediction-Based Sector-Level Path Planning

Based on the proposed traffic flow prediction method, we can predict the robot number $M_j(k)$ in each sector S_j in a future horizon. We define the traffic heat value $h_j(k)$ of each sector S_j as

$$h_j(k) = \frac{M_j(k)}{C_j - M_j(k)}. \tag{5}$$

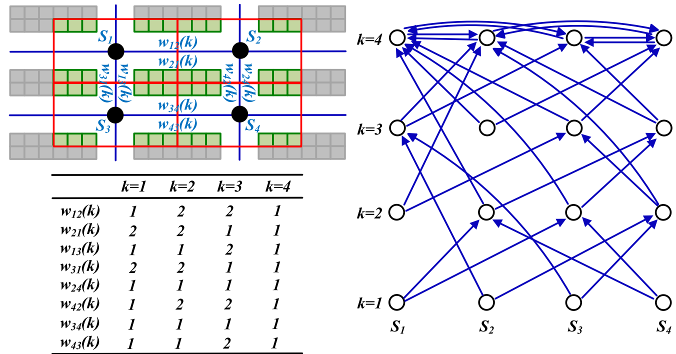
The predicted traffic heat values of all the sectors constitute a dynamic traffic heat map in the future horizon H. The traffic heat map can be considered as an additional cost in the sector-level path planning, in order to balance the traffic flow in the whole environment. To achieve this, the edge weight $w_{ij}(k)$ in sector topology graph G_s defined in Section III-A is set to

$$w_{ij}(k) = \begin{cases} \gamma_{ij} \times [1 + h_j(k)], & k \le H \\ \gamma_{ij} \times [1 + h_j(H)], & k > H \end{cases} \tag{6}$$

where γ_{ij} represent the traveling distance from sector S_i to sector S_j. In this article, without loss of generality, we set γ_{ij} to the shortest road path length from the center of S_i to the center of S_j. The edge weight $w_{ij}(k)$ represents the traveling cost from sector S_i to S_j in the future time k, which can be regarded as an estimated traveling time from sector S_i to S_j under the predicted traffic condition in time k.

Then, the sector-level path planning problem can be transferred into finding the path with the smallest traveling cost in the sector topology graph G_s, which has time-varying edge weight $w_{ij}(k)$. This is the shortest path planning problem in the dynamic graph. We introduce the concept of finite time expanded network (FTEN) to solve this problem [25], [26].

The FTEN $\mathcal{G}_s$ is a directed and time-invariant graph, which is formulated by expanding the corresponding dynamic graph G_s in the discrete time dimension. More specifically, we first make a separate copy of all the vertices in G_s for every time k. Each vertex in FTEN $\mathcal{G}_s$ represents a time-vertex pair (j, k) in G_s, where $j = 1, \ldots, N_s$ represents each sector S_j and $k = 1, 2, \ldots, H$ is the time step. Since the traffic prediction horizon is set to H, the vertices copied in $k = H$ also represent all time-vertex pairs with $k > H$. Then, for every edge $i \to j$



	k=1	k=2	k=3	k=4
$w_{12}(k)$	1	2	2	1
$w_{21}(k)$	2	2	1	1
$w_{13}(k)$	1	1	2	1
$w_{31}(k)$	2	2	1	1
$w_{24}(k)$	1	1	1	1
$w_{42}(k)$	1	2	2	1
$w_{34}(k)$	1	1	1	1
$w_{43}(k)$	1	1	2	1

Fig. 5. Example of the FTEN. Top left: the original dynamic graph with four sectors. Bottom left: the dynamic weights with the prediction horizon $H = 4$. Right: the corresponding FTEN.

in G_s and every time $k = 1, \ldots, H$, we add an edge in FTEN $\mathcal{G}_s$ from the time-vertex pair (i, k) to the time-vertex pair $(j, \min\{H, k + w_{ij}(k)\})$ with the edge weight $w_{ij}(k)$. Fig. 5 shows an example of the FTEN.

Based on the abovementioned definition, we can find that a sector path in the dynamic sector topology graph G_s corresponds to a sector path in the FTEN $\mathcal{G}_s$ with the same departure time, the same reaching time, and the same traveling cost. Thus, the shortest path planning problem in a dynamic graph is equivalent to the shortest path planning problem in the corresponding FTEN. Since FTEN $\mathcal{G}_s$ is a time-invariant graph, any existing graph-based path planning algorithm can be utilized to solve the problem. Furthermore, since, in Section III-A, we have defined all the road segments as monodirectional roads, and we do not allow the overtaking behavior in the robot control, the first-in–first-out (FIFO) property [25] of each edge both in G_s and $\mathcal{G}_s$ can be satisfied, i.e.,

$$k + w_{ij}(k) \le k + 1 + w_{ij}(k+1) \quad \forall k = 1, \ldots, H. \tag{7}$$

Based on which, the shortest path planning problem from the start sector S_s to any sector S_j can be formulated as

$$K_j = \begin{cases} \min\limits_i \ (K_i + w_{ij}(K_i)) & \forall i | (S_i, S_j) \in G_s, \ j \ne s \\ 1, & j = s \end{cases} \tag{8}$$

where K_j represents the earliest reaching time at sector S_j.

From any sector S_j to the destination sector S_e, we also have the following results.

Proposition 2: Consider a static graph $\underline{G}_s$, which has the same vertices and edges with G_s, but in that the weight is defined as $\underline{w}_{ij} = \gamma_{ij}$. For any sector S_j, its traveling cost $d_{je}(k)$ (with the departure time k) to the destination sector S_e satisfies

$$d_{je}(k) \ge \underline{d}_{je} \tag{9}$$

where $\underline{d}_{je}$ represents the minimum traveling cost from S_j to S_e in the static graph $\underline{G}_s$.

Proof: Assume that the sector paths with the minimum traveling cost in the static graph $\underline{G}_s$ and the dynamic graph G_s are $\underline{U}$ and U, respectively. Since $\underline{G}_s$ has the same vertices and edges with G_s, and U is also a feasible sector path in $\underline{U}$, this implies that $\underline{d}_{je} \le \underline{d}_{je}^U$, where $\underline{d}_{je}^U$ is the traveling cost

Algorithm 1 Sector Path Planning With Traffic Prediction

Input: The sector topology graph G_s with $w_{ij}(k)$ updated by the current traffic prediction results; The start sector S_s and destination sector S_e for a robot R_n.
Sector-level
Output: path U^n for the robot R_n.

Generate the FTEN $\mathscr{G}_s$ from G_s;
Set $K_s = 1$, $F_s = \underline{d}_{se} + 1$;
$\forall j \neq s$, Set $K_j = \infty$, $F_j = \infty$;
Set $Open = \{s\}$, $Close = \emptyset$;
while $i \neq e$ **do**
  $i = \arg\min_{j \in Open} F_j$;
  $Open = Open \backslash \{i\}$, $Close = Close \cup \{i\}$;
  for Each j that there is an edge from the pair $(i, \min\{K_i, H\})$ to $(j, \min\{\min\{K_i, H\} + w_{ij}(K_i), H\})$ in $\mathscr{G}_s$ **do**
    if $K_i + w_{ij}(K_i) + \underline{d}_{je} < F_j$ **then**
      Set $K_j = K_i + w_{ij}(K_i)$, $F_j = K_j + \underline{d}_{je}$;
      if $j \notin Open$ **then**
        $Open = Open \cup \{j\}$
      end if
    end if
  end for
end while
return The corresponding shortest path U^n;

of U in $\underline{G}_s$. By the definition in (6), we have $w_{ij}(k) \geq \gamma_{ij}$, which implies that $\underline{d}^U_{je} \leq d_{je}(k)$. Then, we have $d_{je}(k) \geq \underline{d}_{je}$; this proposition is proven. ■

Based on the abovementioned proposition and (8), we utilize the A*-based algorithm on FTEN $\mathscr{G}_s$ to generate the sector-level path U^n for each robot R_n, and the details are shown in Algorithm 1. From criteria C7 and C8 in Section III-A, we can find that, based on the planned sector-level path U^n, a rough prediction of the corresponding road path Q^n can be obtained by directly combining the shortest road path between every two consecutive sectors in U^n. Please note that in order to ensure the optimality of A* and increase its searching efficiency, we need to provide a heuristic method to estimate the traveling cost between any two sectors in the FTEN $\mathscr{G}_s$. Thus, the abovementioned proposition presents a heuristic method to estimate the traveling cost by using $\underline{G}_s$, and the traveling cost in $\underline{G}_s$ provides a good lower-bound of the traveling cost in the FTEN $\mathscr{G}_s$.

C. Dynamic Planning Strategy

The proposed traffic flow prediction method only considers the robot motion delay probability that results from local motion coordinations. However, the unpredictable uncertainties considered in this article, including the temporary communication failures and other robot motion delays caused by human–robot interactions or hardware malfunctions, will increase the robot motion delay probability and decrease the accuracy of the traffic flow prediction results. Furthermore, the longer prediction horizon H will lead to less reliable prediction results.

In order to solve these problems, we introduce the roll planning strategy to replan the sector-level path of each robot dynamically. More specifically, when a robot R_n enters a new sector, count the current robot number M_j in each sector S_j and the number Γ of sectors with $M_j > \alpha C_j$. If $\Gamma > \beta N_s$, predict the traffic flow in the future horizon H and replan the rest sector-level path of the robot R_n by using Algorithm 1 under the current traffic flow prediction results. α and β are empirical threshold parameters. The abovementioned strategy implies that we only repredict the traffic flow and replan the robot sector path when the number of crowded sectors (i.e., the sectors with $M_j > \alpha C_j$) exceeds a certain threshold βN_s. On the one hand, when the problem scale is very large (i.e., the sector number N_s and the robot number N_r are very large), traffic flow prediction will be computationally expensive, and the resulting large FTEN will also lead to a long path planning time. Repredicting the traffic flow and replaning the robot sector path frequently cannot ensure real-time performance. On the other hand, the unpredictable motion and communication uncertainties will greatly reduce the traffic prediction accuracy; the robot sector-level path should be reoptimized by repredicting the traffic flow under the most recent traffic information. Defining the threshold βN_s contributes to achieve a tradeoff between the real-time performance and the optimization performance.

D. Discussion

Balancing the traffic flow in the whole environment contributes to reducing the probability of robot collisions and congestions, increasing transportation flow capacities, and improving warehouse working efficiency. Only considering the shortest path in the planning process cannot ensure the expected optimization performance since the robot congestions (especially in crossroads or trunk roads) cannot be neglected and will greatly affect the real performance of the previous planning results. Thus, we present the traffic flow prediction-based dynamic sector path planning method to achieve the traffic flow equilibrium purpose in the whole environment.

Dynamically planning the robot path in the sector level aims to ensure real-time performance and solve the problems caused by manipulation uncertainties and hardware malfunctions. In existing approaches, the cooperative A* algorithm in the spatial–temporal road graph [9], [13] and the time-window-based algorithms [12] are commonly used to plan a collision-free path for each robot. However, these approaches require that each robot should follow its predefined time schedule strictly. In practical applications, this requirement usually cannot be ensured since the human manipulation time in task loading and unloading processes, the movement time from one place to another place, and the waiting time in working stations usually cannot be accurately predicted or controlled. What is more, directly planning (and replanning) the path of each robot in the spatial–temporal road graph will be very computationally expansive, especially in large-scale environments with a large number of robots. Thus, we only plan the sector-path of each robot and leave the motion coordination problem to the local planner of each sector.

V. Decentralized Local Motion Coordination

Road-level path planning and robot motion coordination are accomplished locally within each sector S_j. This aims to decrease the coordination complexity, react to robot delays and failures immediately, and sure the solvability of the whole system. More specifically, the objective in the local planner of each sector S_j is to plan local road paths for all the robots that travel through the sector S_j and, in the meantime, coordinate their motions to avoid conflicts.

A. Road-Level Cooperative Path Planning

In Section III-A, we have defined the road topology graph G_x; based on which, we can obtain the local road topology graph G_x^j that describes the local road topology within the sector S_j. Based on the local road topology graph G_x^j, the cooperative A* algorithm [9], [13] is utilized in each sector S_j to generate a collision-free local road path (within sector S_j) for each robot located in sector S_j. In the cooperative A* algorithm, the path of each robot is planned in a time-expanded road graph generated from G_x^j simultaneously; thus, the robot collision can be avoided by considering the location constraints presented in Section II. In order to ensure the real-time performance, we utilize the CBS in the spatial–temporal space [9] to speed up the convergence procedure of the cooperative A* algorithm. Please note that introducing the CBS will not affect the optimality of the cooperative A* algorithm.

Each time, a new robot may move into sector S_j, or a robot that is located in sector S_j previously may move out from sector S_j. Thus, the robots located in each sector usually change continuously. In order to solve this problem, we introduce the online roll planning strategy to continuously replan the local road path of each robot in each sector S_j, i.e., at each time step, the local planner of each sector plans the local road path for each robot in the sector from its current locations to the exit corresponding to its next sector in the sector path, but each robot only moves one step. Introducing the rolling planning structure also contributes to solve the motion delay problem caused by uncertainties mentioned in Section IV-D since the location constraints presented in Section II have taken the motion uncertainties into consideration, and the path of all the robots will be regenerated frequently.

In order to avoid any robot conflict caused in sector switchings during robot movements, an entryway reservation system is implemented to manage the authority. More specifically, if a robot has arrived at the exit of its current sector S_j, the robot should first apply for the authority to enter the next sector S_{j+1}. If the entrance of S_j has not been occupied by any other robot, the entrance will be reserved for the robot and will not delegate to anyone else. After the robot has entered the sector S_{j+1} and moved away from the entrance, the entrance will be released again. Note that the movement delays caused by the abovementioned authority application procedures will also be regarded as motion uncertainties in this article and will be solved by the proposed rolling planning strategy.

B. Strategies for Resolving Communication Failures

The communication connection between robots and the local planner/controller of each sector may be temporarily disabled due to bandwidth limitations, route handoff and reestablishment operators, or hardware malfunctions. By implementing the heart-single-based method [27], the state of each robot in the sector can be monitored, and communication failures can be detected in real time.

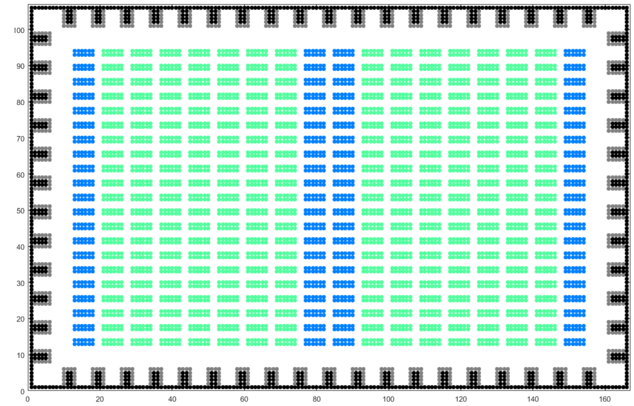


Fig. 6. Environment map in simulations. The blue, green, and gray grids denote the robot station, the task pickup station, and the working station, respectively. The white area is the moveable region of each robot, and the black grids represent the human activity areas.

In this article, we design a K-step redundancy mechanism to solve the robot collision problem caused by communication failures: if a communication failure occurs, the failed robot will continue to follow its road-level path planned in the previous time step while waiting for the reestablishment of the communication connection. In order to ensure the safety, if the communication cannot be reconstructed immediately, the failed robot will move up to K-steps and then stops to wait for the communication reconstruction. What is more, if the failed robot reaches the exit of its current sector, it will also stop to ensure safety. In the meantime, the local planner/controller in the sector will estimate the maximum motion range (the K-step range) of the failed robot and close the corresponding area to prevent other robots from entering, and command the robots that are currently located in that area to stop immediately. Since, in the individual robot motion control system, each robot can detect collisions by its onboard sensors (such as range sensors or impact sensors), the safety can be ensured. Finally, once the communication connection is reconstructed, the local planner/controller in the sector will replan the road-level path for all the influenced robots.

The proposed K-step redundancy mechanism aims to reduce the impact of communication failure on traffic efficiency while ensuring safety. Since most of the communication failures in practical applications are temporary failures and can be recovered in a short time (during the K-step movements of the failed robot), the impact can be eliminated, and the system performance can be maintained.

VI. Large-Scale Simulations With 1000-Robot Networks

Simulations are conducted in MATLAB platform by using a computer with 2.2-GHz Core i7-8750 CPU 16-GB RAM. The environment utilized in simulations is shown in Fig. 6.

TABLE I

UNCERTAINTY LEVEL DEFINITION

	Motion Delay Prob. f_m	Comm. Recovery Prob. f_c
Level 1	0.5%	35%
Level 2	1%	30%
Level 3	1.5%	25%
Level 4	2%	20%
Level 5	2.5%	15%
Level 6	3%	10%

The whole environment is discretized into a mesh of $N_x = 166 \times 106$ four-connecting grids, and the size of any station is set to one grid, which can only be occupied by one robot in each step. In this map, there are totally 1008 robot stations, 744 working stations, and 3528 task pickup stations. The whole environment is partitioned into $N_s = 472$ sectors by following criteria C6–C8 proposed in Section III-A. According to criteria C1–C5 and Proposition 1, this environment enables the test of up to 744 running tasks at the same time and can be used to conduct large-scale simulations with up to 1008 robots. Thus, 1008 robots and 3000 online published transportation tasks are considered in simulations, where each task is assigned to the nearest free robot. The Mahalanobis distance is utilized to compute the distance between two grids (places) or two-sector centers. The parameters used in simulations are set to $\epsilon = 0.5$, $H = 40$, $\alpha = 0.3$, $\beta = 0.3$, and $K = 3$. At each time step, we add a motion delay probability f_m to each moving robot. The delayed robots will stay in their current locations for one time step, i.e., ignore the local road path planned by its current sector planner. In order to simulate temporary communication failures, one of the normal robots will be disconnected, and the previously disconnected robots will recover their communication connections with the probability of f_c. As shown in Table I, the uncertainties introduced into simulations are divided into six levels. Note that at the higher level, both the motion uncertainties and the communication uncertainties are larger. All the evaluation indices considered in simulations are defined in Table II. Simulations under each condition are repeated for five tests, and both the mean value and the standard deviation of each evaluation index defined in Table II are recorded.

We first evaluate the proposed approach under different TaskPubRate from 1 to 7. The uncertainty condition is set to Level 1 in order to eliminate the effect of large uncertainties. An obvious saturation phenomenon of each evaluation index can be seen in the results shown in Fig. 7. This implies that the system capacity has an upper bound, and the algorithm performance tends to be saturated under a very heavy workload. The possible reasons are listed as follows. First, the environment capacity is determined by the map size and environment structure. In order to provide enough space for robot motion and coordination, the number of road segments and stations cannot be very large. Second, according to Proposition 1, the number of running tasks at the same time has an upper bound that is related to the number of task pickup and working stations. If criterion C3 in Section II cannot be fulfilled, the release of new tasks will be delayed. Third, a large number of moving robots will increase traffic congestions and uncertainties, which will further lead to system performance degradations. From the indices MakespanT, AveTaskWaitT, AveTaskAccomT, and AveAveHeatVal, we can find that the saturated capacity of the system can be estimated as TaskPubRate = 4 or 5. After this bound, the system performance tends to be stable. The saturated computation time in each simulation is lower than 1.7 s. Since the computing resource in practical warehousing systems is typically much higher than the computer used in simulations, the real-time performance of the proposed approach can be ensured.

As an example, Fig. 8 shows the dynamic task accomplishment states in one simulation with TaskPubRate = 5 and the uncertainty Level 1. We can find that the system performance is very stable, and there is no strong fluctuation appeared.

TABLE II

SIMULATION INDEX DEFINITION

Index	Definition
TaskPubRate	the number of tasks published at each time step
MakespanT	the minimum time step needed to accomplish all the tasks
AveCalT	the average computing time at each time step
AveMoDelayN	the average number of robots with motion delays at each time step
MaxMoDelayN	the maximum number of robots with motion delays in all the time steps
AveCommLostN	the average number of robots with communication failures at each time step
MaxCommLostN	the maximum number of robots with communication failures in all the time steps
MaxCommLostT	the longest disconnection time duration of the robot in all the time steps
AveTaskWaitT	the average time step needed from the publishing time to the pickup time of each task
AveTaskAccomT	the average time step needed from the publishing time to the accomplish time of each task
MaxMaxHeatVal	the maximum heat-value of all sectors in all the time steps
AveMaxHeatVal	the average of the maximum heat-value of all the sectors at each time step
AveAveHeatVal	the average of the average heat-value of each sector at each time step

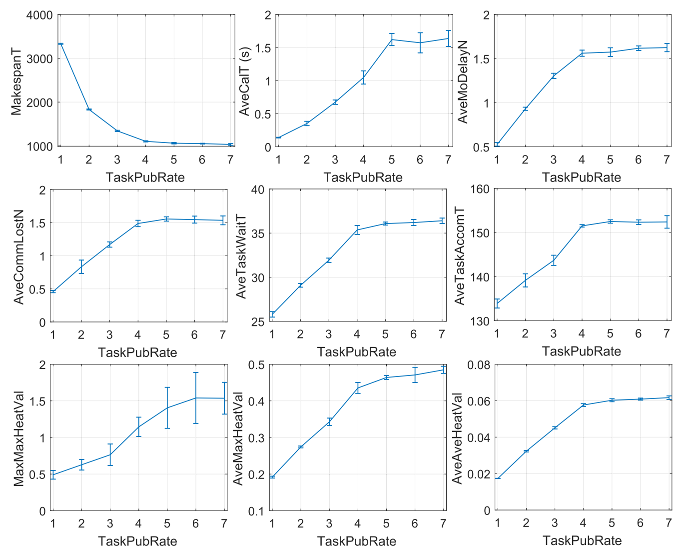


Fig. 7. Simulation results under a different TaskPubRate.

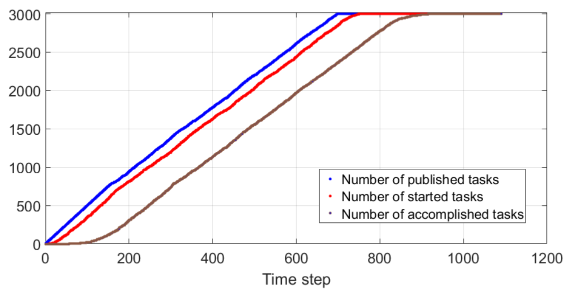
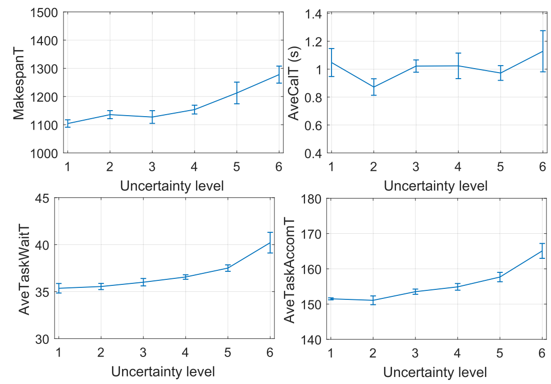


Fig. 8. Example of the dynamic task accomplishment states.

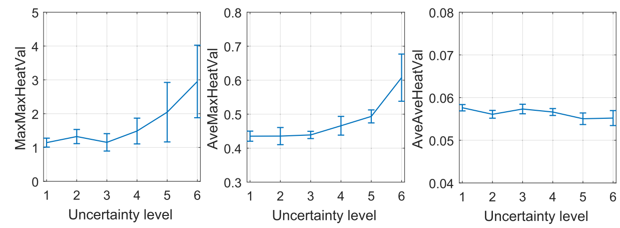
We set TaskPubRate = 4 to further evaluate the proposed approach under different uncertainty levels defined in Table I. As shown in Fig. 9, MakespanT, AveTaskWaitT, and AveTaskAccomT increase slightly with the increasing uncertainties and the average traffic heat value of each sector remains in the acceptable value interval across all the uncertainty levels, and these imply that the traffic jams and robot deadlocks are avoided even in the presence of large motion and communication uncertainties. Furthermore, the algorithm computing time remains lower than 1.3 s under different uncertainty levels, i.e., the real-time performance of the proposed system will not be affected by large uncertainties. Please note that, in the abovementioned simulations, the maximum number of delayed robots is up to 24, the maximum number of robots with communication disconnections is up to 18, and the longest disconnection duration is up to 71 time steps, which are large enough to simulation the worst situations in practical robotic warehouses. Due to the proposed approach, there is no robot collision occurred in simulations, and all the tasks are accomplished successfully. These simulation results validate the effectiveness of the proposed approach in large-scale problems with large uncertainties.

In addition, we compare the proposed approach with [7] in a small environment with a size of 78×50 (similar to the map structure shown in Fig. 6). In [7], since the distributed negotiation mechanism is utilized to achieve local coordinations, the robot deadlocks and traffic jams cannot be avoided completely in large-scale problems. In order to ensure the success rate of [7], when robot deadlock occurs, we remove the blocked robots from the simulation for 20 s and then reintroduce them in the same positions (please note that we do not introduce this operation in our approach). We consider the one-shot problem and set the robot number from 70 to 150 with different uncertainty levels. From the comparison results in Figs. 10 and 11, we can find the following.

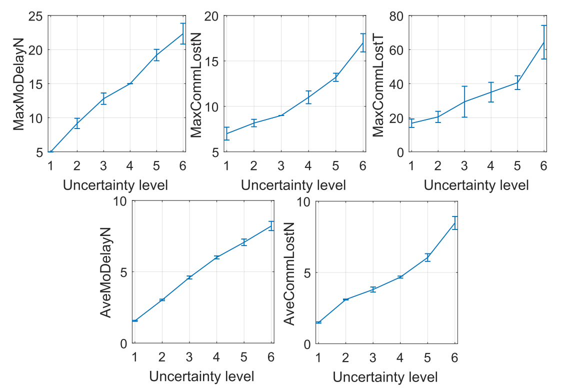
1) The performance of the proposed approach is similar to [7] in the case with the smallest robot number and lowest uncertainty level; as, in this case, the robot density is very low, the motion conflicts can be easily solved, and the effect of the traffic heat distribution on the robot motion can be neglected, so both approaches can handle the path planning and motion coordination tasks easily.
2) With the increasing of the robot number and uncertainty level, the performance of [7] decreases largely since more robot conflicts occur and the robot congestion problem cannot be well handled. Especially, in the case with 150 robots, the average task accomplish time of [7] is much higher than ours.
3) Our heat values are much smaller than [7]; this validates that the traffic flow can be balanced, and robot congestions can be reduced in our simulations.



(a)



(b)



(c)

Fig. 9. Simulation results under different uncertainty levels. (a) Time indices. (b) Heat-value indices. (c) Motion and communication uncertainties indices.

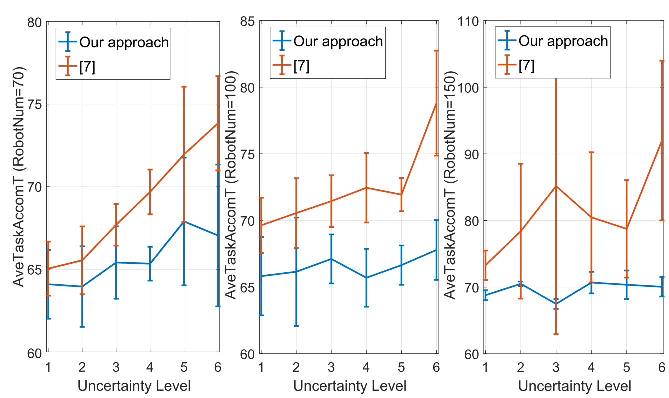


Fig. 10. Comparison results: the average task accomplish time.

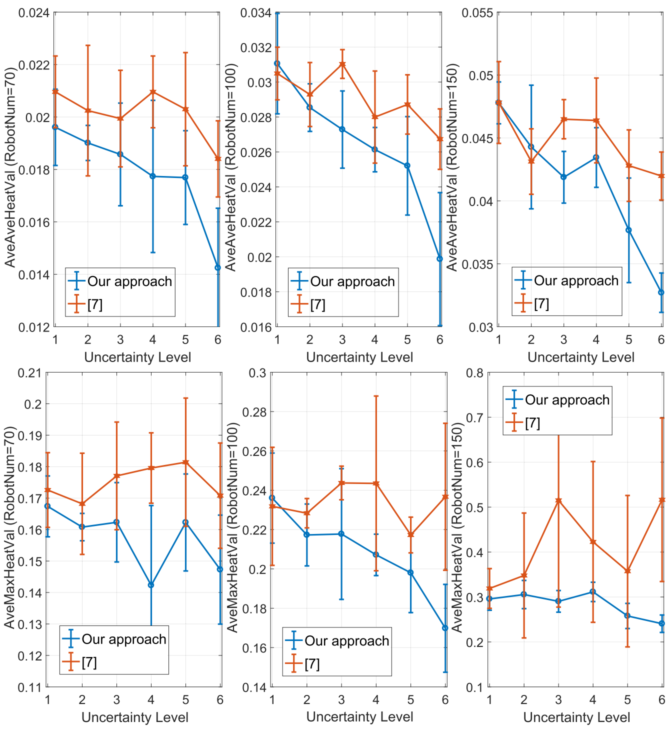


Fig. 11. Comparison results: heat values of each sector.

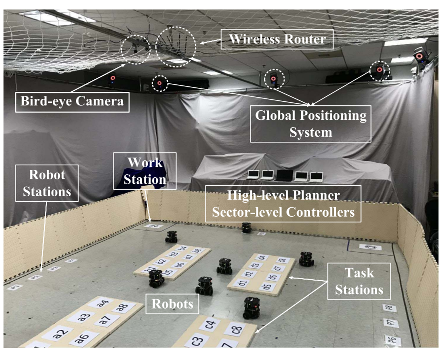


Fig. 12. Experiment system.

VII. Laboratorial Experiments

In order to validate the practical applicability of the proposed approach, we conduct laboratory experiments with seven TurtleBot robots. The experiment system is shown in Fig. 12; we develop an indoor global positioning system with ten VICON cameras to localize each robot in the environment and build a wireless communication network to achieve real-time Wi-Fi communications between the centralized sector path planner, the controller of each sector, and the TurtleBot robots. In experiments, the traffic flow prediction and sector-level dynamic path planning are conducted on a laptop with Core i7-6500U 2.59-GHz CPU and 8-GB RAM, and the local road path planning and motion coordination are conducted on laptops with 1.6-GHz CPU and 1-GB RAM.

As shown in Fig. 13, the experiment environment is divided into 21 sectors, and the moveable region in each sector has been discretized into several four-connecting grids (the gird size is 0.3 m × 0.3 m). One grid corresponds to one place x_l in Section III-A and can only be occupied by one robot at each time step (the robot size is about 0.14 m × 0.18 m).

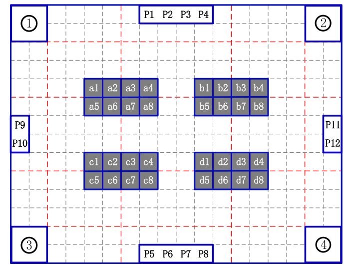


Fig. 13. Experiment map. The environment size is 5.4 m × 4.2 m, including 32 task pickup stations distributed over four task pickup station areas *a*–*d* (in each area, there are eight task pickup stations); 12 robot stations $p_1 - p_{12}$; and 16 working stations distributed over four working station areas ①–④ (in each area, there are four working stations).

TABLE III

Experimental Results

	K_i^{pub}	X_i^{pick}	X_i^{work}	R_i	K_i^{assign}	K_i^{start}	K_i^{finish}
T_1	$2s$	d_8	4	R_5	$2s$	$12s$	$24s$
T_2	$2s$	a_7	2	R_7	$2s$	$8s$	$37s$
T_3	$2s$	c_2	3	R_1	$2s$	$18s$	$35s$
T_4	$2s$	a_2	1	R_3	$2s$	$17s$	$27s$
T_5	$2s$	d_3	1	R_4	$2s$	$7s$	$40s$
T_6	$8s$	b_4	4	R_2	$8s$	$21s$	$59s$
T_7	$8s$	d_6	4	R_6	$8s$	$21s$	$35s$
T_8	$14s$	b_8	2	R_5	$24s$	$42s$	$57s$
T_9	$14s$	a_5	1	R_3	$27s$	$43s$	$64s$
T_{10}	$20s$	a_5	3	R_1	$35s$	$52s$	$81s$
T_{11}	$20s$	b_2	2	R_6	$35s$	$62s$	$83s$
T_{12}	$26s$	a_2	1	R_7	$37s$	$64s$	$80s$

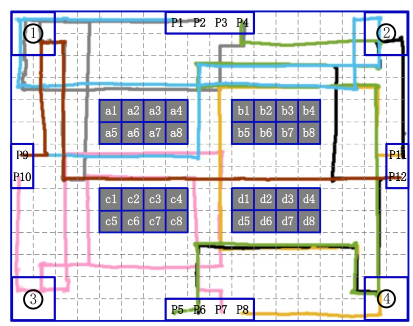


Fig. 14. Real robot trajectories recorded by the global positioning system.

All the stations are of the same size as the grids. Note that temporary communication blocks and motion inaccuracies are inherently existed in the experiment system. As shown in Table III, 12 transportation tasks are online published with the random-generated pickup and working stations. K_i^{pub}, X_i^{pick}, and X_i^{work} in Table III show the publishing time, task pickup station, and working station of each task T_i, respectively. Initial locations of the seven robots are randomly generated in the 12 robot stations, where R_1 is located in p_7, R_2 is located in p_{11}, R_3 is located in p_4, R_4 is located in p_{12}, R_5 is located in p_6, R_6 is located in p_5, and R_7 is located in p_9. In experiments, all the transportation tasks are assigned to the nearest free robot.

The last four columns in Table III show the experimental results, where R_i represents the robot that is assigned to accomplish the transportation task T_i, and K_i^{assign}, K_i^{start},

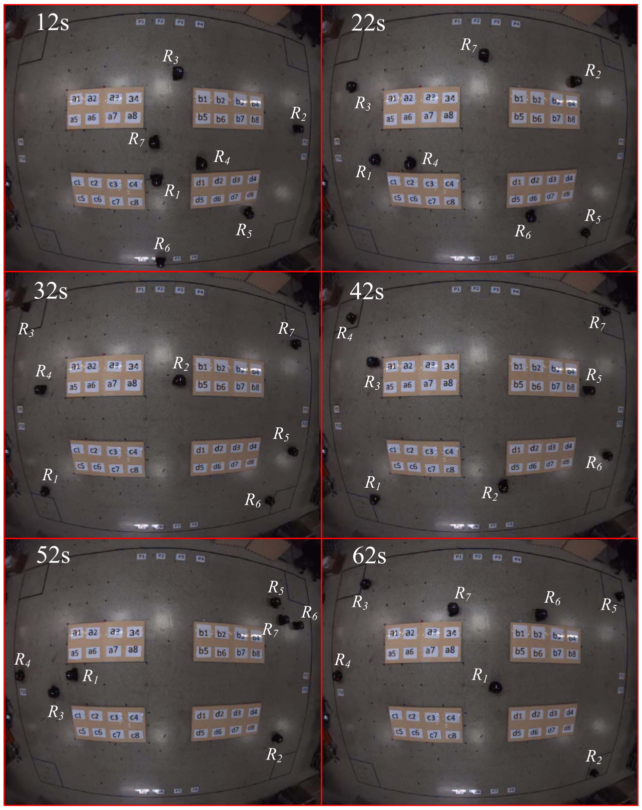


Fig. 15. Real experiment scenarios recorded by the bird-eye camera.

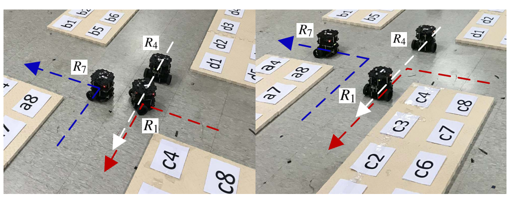


Fig. 16. Experiment scenarios in 14 (left) and 17 s (right), where the motions of three robots are coordinated in an intersection to avoid collisions.

and K_i^{finish} record the real-time instants when the task T_i is assigned to the robot R_i, picked up by the robot R_i in station X_i^{pick}, and accomplished in working station X_i^{work}, respectively. Results in Table III show that each transportation task is immediately allocated to the nearest free robot once published or allocated to the earliest free robot when the robot has accomplished its previous assigned task. All the transportation tasks are accomplished within 83 s, and no robot motion conflict occurs.

Fig. 14 shows the plots of the real robot trajectories recorded by the global indoor positioning system. We can find that robots can track their road-level paths with acceptable navigation accuracy. Fig. 15 shows six real scenarios in the experiments form 12 to 62 s. Fig. 16 shows two consecutive scenarios as an example to show our motion coordination performance. The abovementioned experimental results validate the effectiveness and practical applicability of the proposed approach. More details of our simulations and experiments can be found in the video https://youtu.be/Ztzq78plsIM.

VIII. Conclusion

In this article, we propose a probabilistic traffic flow prediction algorithm and a hierarchical framework to resolve the life-long path planning and motion coordination problem of 1000-robot networks. Large-scale simulations with robot motion uncertainties and communication failures validate the effectiveness and robustness of the proposed approach, and comprehensive comparisons show that our approach outperforms existing work. In addition, real experiments with a group of TurtleBot robots validate the practical applicability of the proposed approach. In future work, we will take the environment changes into consideration and also introduce the learning-based approaches to achieve a fully decentralized system without global information. In addition, we will introduce a local FTEN within each sector to further improve the traffic prediction accuracy.

References

[1] M. Cap, P. Novak, A. Kleiner, and M. Selecky, "Prioritized planning algorithms for trajectory coordination of multiple mobile robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 835–849, Jul. 2015.

[2] A. Ham, "Drone-based material transfer system in a robotic mobile fulfillment center," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 2, pp. 957–965, Apr. 2020.

[3] Z. Liu, H. Wang, W. Chen, J. Yu, and J. Chen, "An incidental delivery based method for resolving multirobot pairwised transportation problems," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1852–1866, Jul. 2016.

[4] H. Ma *et al.*, "Overview: A hierarchical framework for plan generation and execution in multirobot systems," *IEEE Intell. Syst.*, vol. 32, no. 6, pp. 6–12, Nov. 2017.

[5] R. Tai, J. Wang, and W. Chen, "A prioritized planning algorithm of trajectory coordination based on time windows for multiple AGVs with delay disturbance," *Assem. Autom.*, vol. 39, no. 5, pp. 753–768, Nov. 2019.

[6] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," *AI Mag.*, vol. 29, no. 1, pp. 9–19, 2008.

[7] V. Digani, L. Sabattini, C. Secchi, and C. Fantuzzi, "Ensemble coordination approach in multi-AGV systems applied to industrial warehouses," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 922–934, Jul. 2015.

[8] J. Yu and S. M. LaValle, "Optimal multirobot path planning on graphs: Complete algorithms and effective heuristics," *IEEE Trans. Robot.*, vol. 32, no. 5, pp. 1163–1177, Oct. 2016.

[9] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant, "Conflict-based search for optimal multi-agent pathfinding," *Artif. Intell.*, vol. 219, pp. 40–66, Feb. 2015.

[10] H. Ma, J. Li, T. K. S. Kumar, and S. Koenig, "Lifelong multi-agent path finding for online pickup and delivery tasks," in *Proc. 16th Int. Conf. Auto. Agents Multiagent Syst.*, 2017, pp. 1–9.

[11] H. Ma, T. K. S. Kumar, and S. Koenig, "Multi-agent path finding with delay probabilities," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 3605–3612.

[12] N. Smolic-Rocak, S. Bogdan, Z. Kovacic, and T. Petrovic, "Time windows based dynamic routing in multi-AGV systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 1, pp. 151–155, Jan. 2010.

[13] H. Roozbehani and R. D'Andrea, "Adaptive highways on a grid," in *Robotics Research*. Berlin, Germany: Springer-Verlag, 2011, pp. 661–680.

[14] G. Sharon, R. Stern, M. Goldenberg, and A. Felner, "The increasing cost tree search for optimal multi-agent pathfinding," *Artif. Intell.*, vol. 195, pp. 470–495, Feb. 2013.

[15] Z. Liu, S. Zhou, H. Wang, Y. Shen, H. Li, and Y.-H. Liu, "A hierarchical framework for coordinating large-scale robot networks," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 6672–6677.

[16] S. D. Han and J. Yu, "DDM: Fast near-optimal multi-robot path planning using diversified-path and optimal sub-problem solution database heuristics," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1350–1357, Apr. 2020.

[17] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research*. Berlin, Germany: Springer-Verlag, 2011, pp. 3–19.

[18] P. Velagapudi, K. Sycara, and P. Scerri, "Decentralized prioritized planning in large multirobot teams," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 4603–4609.
[19] V. Digani, L. Sabattini, and C. Secchi, "A probabilistic Eulerian traffic model for the coordination of multiple AGVs in automatic warehouses," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 26–32, Jan. 2016.
[20] G. Sartoretti *et al.*, "PRIMAL: Pathfinding via reinforcement and imitation multi-agent learning," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2378–2385, Jul. 2019.
[21] B. Wang, Z. Liu, Q. Li, and A. Prorok, "Mobile robot path planning in dynamic environments through globally guided reinforcement learning," 2020, *arXiv:2005.05420*. [Online]. Available: https://arxiv.org/abs/2005.05420
[22] W. Honig, S. Kiesel, A. Tinka, J. W. Durham, and N. Ayanian, "Persistent and robust execution of MAPF schedules in warehouses," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1125–1131, Apr. 2019.
[23] J. Li, A. Tinka, S. Kiesel, J. W. Durham, T. K. Satish Kumar, and S. Koenig, "Lifelong multi-agent path finding in large-scale warehouses," 2020, *arXiv:2005.07371*. [Online]. Available: http://arxiv.org/abs/2005.07371
[24] J. Švancara, M. Vlk, R. Stern, D. Atzmon, and R. Bartak, "Online multi-agent path finding," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 7732–7739.
[25] I. Chabini and S. Lan, "Adaptations of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 1, pp. 60–74, Mar. 2002.
[26] M. Ferrati and L. Pallottino, "A time expanded network based algorithm for safe and efficient distributed multi-agent coordination," in *Proc. 52nd IEEE Conf. Decis. Control*, Dec. 2013, pp. 2805–2810.
[27] Z. Liu, W. Chen, H. Wang, Y.-H. Liu, Y. Shen, and X. Fu, "A self-repairing algorithm with optimal repair path for maintaining motion synchronization of mobile robot network," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 3, pp. 815–828, Mar. 2020.

Zhe Liu received the Ph.D. degree in control technology and control engineering from Shanghai Jiao Tong University, Shanghai, China, in 2016.

From 2017 to 2020, he was a Post-Doctoral Fellow with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong. He is currently a Research Associate with the Department of Computer Science and Technology, University of Cambridge, Cambridge, U.K. His current research interests include multirobot cooperation and autonomous driving systems.

Hesheng Wang (Senior Member, IEEE) received the B.Eng. degree in electrical engineering from the Harbin Institute of Technology, Harbin, China, in 2002, and the M.Phil. and Ph.D. degrees in automation and computer-aided engineering from The Chinese University of Hong Kong, Hong Kong, in 2004 and 2007, respectively.

He was a Post-Doctoral Fellow and a Research Assistant with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, from 2007 to 2009. He is currently a Professor with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China. His current research interests include visual servoing, service robot, adaptive robot control, and autonomous driving.

Dr. Wang was the General Chair of the IEEE International Conference on Real-time Computing and Robotics (RCAR) 2016 and the Program Chair of the IEEE International Conference on Robotics and Biomimetics (ROBIO) 2014 and the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM) 2019. He has served as an Associate Editor for the IEEE TRANSACTIONS ON ROBOTICS from 2015 to 2019. He is also an Associate Editor of *Assembly Automation* and the *International Journal of Humanoid Robotics* and a Technical Editor of the IEEE/ASME TRANSACTIONS ON MECHATRONICS.

Huanshu Wei received the B.S. degree in aerospace engineering from the Beijing Institute of Technology, Beijing, China, in 2012, and the M.Sc. degree in aeronautical engineering from Beihang University, Beijing, in 2019.

He is currently a Research Assistant with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong. His current research interests include multirobot cooperation, dynamic control systems, and Simultaneous Localization and Mapping (SLAM).

Ming Liu (Senior Member, IEEE) received the B.A. degree in automation from Tongji University, Shanghai, China, in 2005, and the Ph.D. degree from the Department of Mechanical Engineering and Process Engineering, ETH Zürich, Zürich, Switzerland, in 2013.

He was a Visiting Scholar with Erlangen Nurnberg University, Erlangen, Germany, and the Fraunhofer Institute IISB, Erlangen. He is currently an Assistant Professor with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong. His current research interests include autonomous mapping, visual navigation, topological mapping, and environment modeling.

Prof. Liu was a recipient of the Best Student Paper Award from the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI) 2012, the Best Paper in Information Award from the IEEE International Conference on Information and Automation (ICIA) 2013, the Best RoboCup Paper from the IEEE International Conference on Intelligent Robots and Systems (IROS) 2013, the Best Conference Paper Award from the IEEE International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER) 2015, the Best Student Paper Award from the IEEE International Conference on Advanced Robotics (ICAR) 2017, and the Best Paper in Automation Award from the IEEE ICIA 2017. He received the Innovation Contest Chunhui Cup Winning Awards in 2012 and 2013. He received the Wu Weijun AI Award in 2016 and the IROS Toshio Fukuda Young Professional Award in 2018. He is on the Editorial Board of the IEEE ROBOTICS AND AUTOMATION LETTERS.

Yun-Hui Liu (Fellow, IEEE) received the B.Eng. degree from the Beijing Institute of Technology, Beijing, China, in 1985, the M.Eng. degree from Osaka University, Suita, Japan, in 1989, and the Ph.D. degree from The University of Tokyo, Tokyo, Japan, in 1992.

After working at the Electrotechnical Laboratory, Tsukuba Science City, Japan, as a Research Scientist, he joined The Chinese University of Hong Kong, Hong Kong, in 1995, where he is currently a Professor with the Department of Mechanical and Automation Engineering and the Director of the CUHK T Stone Robotics Institute. He is also an Adjunct Professor with the State Key Lab of Robotics Technology and System, Harbin Institute of Technology, Harbin, China. His research interests include visual servoing, medical robotics, mobile robots, and machine intelligence.

Dr. Liu received numerous research awards from international journals and international conferences in robotics and automation and government agencies. He was listed in the Highly Cited Authors (Engineering) by Thomson Reuters in 2013. He has served as the General Chair of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. He has served as an Associate Editor for the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION. He is also the Editor-in-Chief of *Robotics and Biomimetics*.