Robotica (2023), **41**, pp. 3333–3348
doi:10.1017/S0263574723001005

CAMBRIDGE
UNIVERSITY PRESS

RESEARCH ARTICLE

A divide-and-conquer control strategy with decentralized control barrier function for luggage trolley transportation by collaborative robots

Xuheng Gao[1], Hao Luan[1], Bingyi Xia[1], Ziqi Zhao[1], Jiankun Wang[1,2] and Max Q.-H. Meng[1]

[1]Shenzhen Key Laboratory of Robotics Perception and Intelligence, Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, China and [2]Jiaxing Research Institute, Southern University of Science and Technology, Jiaxing, China

Corresponding authors: Jiankun Wang, Max Q.-H. Meng; Emails: wangjk@sustech.edu.cn, max.meng@ieee.org

Received: 7 March 2023; **Revised:** 28 May 2023; **Accepted:** 24 June 2023; **First published online:** 17 August 2023

Keywords: mobile robots; multi-robot systems; automation; decentralized control; collaborative robots

Abstract

This article focuses on the luggage trolley transportation problem, an essential part of robotic autonomous luggage trolley collection. To efficiently address the nonholonomic constraints derived from the formation of two collaborative robots and a queue of luggage trolleys, we propose a comprehensive framework consisting of a global planning method and a real-time divide-and-conquer control strategy. The popular Hybrid A* algorithm generates a feasible path as the global planner. A model predictive controller is designed to track this path stably and in real time. To maintain the formation so that the whole queue of robots and luggage trolleys does not split, a safety filter that consists of a discrete-time control Lyapunov function and a decentralized control barrier function is implemented in the transportation process. Finally, we conduct real-world experiments to verify the effectiveness of the proposed method on three representative paths, and the results show that our approach can achieve robust performance. The demonstration video can be found at https://www.youtube.com/watch?v=iPiT8BfLIpU.

1. Introduction

Today, intelligent robots are widely used in shopping malls, restaurants, hospitals, and other places to assist people in completing tedious tasks. They can also be used to achieve robotic autonomous luggage trolley collection at airports, which is a major task at all airports [1]. Currently, most airports still manually collect and transfer luggage trolleys to designated areas, which is labor-intensive and relatively inefficient [2]. Using robots to collect and transfer dispersed luggage trolleys can improve work capacity [3]. The collection task can be divided into three sub-tasks: (1) Identifying and collecting dispersed and unoccupied luggage trolleys one by one [4]; (2) Gathering the collected trolleys into a queue; and (3) Two collaborative robots transport the luggage trolley queue to a designated area for continued use by other passengers. The first two sub-tasks have been addressed by many researchers [1,4]. Herein, this article focuses on the last challenging sub-task and proposes a divide-and-conquer control strategy that solves the multiple and complicated constraints separately. In this article, the control system we need to deal with consists of a luggage trolley queue and two collaborative robots, one as the leader and the other as the follower. Figure 1 illustrated a photo of the whole system.

The luggage trolley queue can be regarded as a series of rigid bodies. Transporting a series of rigid bodies is commonly used in object manipulation, especially for a large quantity. However, it is not easy to handle the transportation of luggage trolley queues. The system is subject to nonholonomic constraints because collaborative robots have differential structures. In addition, the relative position of the leading and following robots must be considered to maintain a stable formation. Due to the complexity of the

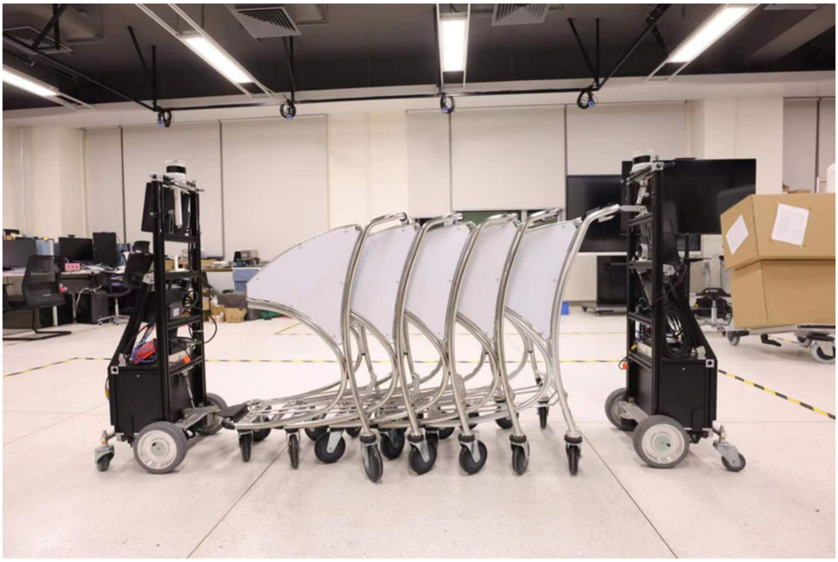


Figure 1. The luggage trolley transportation system.

task, it is usually to calculate the feasible control input by means of optimization, which is widely used in a variety of complicated robot tasks [5–7].

Inspired by the idea of divide-and-conquer, we propose a hierarchical policy to achieve this transportation task in this article. First, a high-level global planner provides a smooth path from initial to goal positions. Second, a low-level controller is designed to track the global path while maintaining the system's formation. Specifically, we use Hybrid A* [8] to generate a global path that is smooth enough and practical for the system to track. After that, we propose a divide-and-conquer control strategy that contains three modules. We use a model predictive controller module (MPC) to generate a set of initial control inputs. MPC can plan the trajectory by minimizing the loss [9]. Subsequently, a safety filter, which consists of a discrete-time control Lyapunov function module (DT-CLF) and a control barrier function module (CBF) in a decentralized form, is employed to optimize the generated control inputs. The safety filter ensures that the control inputs satisfy all formation constraints in the collaborative transportation task. Finally, the filtered control inputs are sent to the actuators to drive the two collaborative robots to move the luggage trolley queue along the global path.

1.1. Related work

Manipulating objects with mobile robots has received a lot of attention recently. A substantial potential application is for object transportation [10]. However, existing challenges include but are not limited to mechanical design, planning algorithms, control strategies, etc. In this article, we pay attention to planning and control parts. Currently, the research works can be divided into two categories depending on the end effector. (1) Integrating robot manipulators on the mobile platform. (2) Designing customized manipulation mechanisms for specific tasks. The former is universal for sophisticated tasks represented by PR2 [11] and Fetch [12]. Refs. [13] and [14] also focuses on the control of collaborative mobile manipulators. However, the kinematic redundancy will complicate planning and control difficulties [15]. Another disadvantage is the high cost, which limits its widespread use. In contrast, the latter exhibits greater flexibility in specific tasks, which has also received extensive attention from academia and industrial community.

Most of the current research focuses on manipulating a single object by mobile robots. Both refs. [4] and [16] employ one mobile robot to collect a single luggage trolley. When dealing with more complex tasks, the functions and capabilities of a single robot are limited. Multi-robots gradually replace the

https://doi.org/10.1017/S0263574723001005 Published online by Cambridge University Press

single robot to achieve more complex work [17]. In our luggage trolley transport task, the connection between any adjacent two trolleys is not perfectly rigid, so the connection is unstable. Relying on push and pull in a single direction will not make the queue move as expected. In addition, a single robot cannot control the direction of the queue of luggage trolleys during transport. In order to solve the above problems, we adopt the mode of two robots, one in front and one in back, lining up with luggage trolleys and cooperating to complete the transportation.

Although [18] realizes transporting multiple objects by six robots, the transported objects are not connected to any robots. Six robots form a cage that surrounds the objects and push them forward. This method does not control the payload well, especially for large objects. Ref. [19] presents a four-robot fleet to transport a 20 kg plate. However, the payload is not caught by any robots. The connection between robots and the payload is unstable, especially during acceleration and deceleration. To solve the instability, we design a specific grasping device to fix the trolley. Ref. [15] realizes multi-robot collaborative handling of a deformable thin sheet. Since the object is deformable, the distance between robots and the angle of each robot does not need to be strictly constrained. Contrary to the flexibility of non-stationary applications and deformable objects, transporting a luggage trolley queue is much more difficult because there is little room for redundancy in the changing distance between two robots. The collaborative robots must clamp the luggage trolley queue to prevent disassembly. Additionally, due to the mechanical connection restrictions of the grasping device and the formation requirements of the system, there should be a maximum limit on the robot's steering angle during the turning process. Therefore, we also need to control the angular constraints of the robot.

A well-known formation is the leader-follower formation. A popular communication type is implicit. The robots acquire information by interactions with the environment and other robots [20]. The authors construct two mobile robots to transport the rod, which is also a rigid body. The authors implement the decentralized framework to improve the robustness. Ref. [21] also makes followers apply force in the leader's direction as a policy. Ref. [22] introduces a consensus-based approach by using only local measurements from each robot. The authors demonstrate that leader-follower formation converges as long as the leader exists. In leader-follower formation, the leader has absolute leadership in the system. Only the leader knows the goal configuration of the task. The follower ensures contact with the object and the system formation only through force/torque measurements. Since the leader has no feedback from the follower, it cannot adjust accordingly when the follower is disturbed [23].

1.2. Our contributions

The contributions of this work are threefold.

1. We propose a divide-and-conquer control strategy to deal with the multiple constraints in luggage trolley transportation.
2. In the control strategy, a decentralized CBF for managing angle constraints is applied.
3. We verify the effectiveness of our system on both the simulation platform and the physical hardware platform.

2. System description

2.1. Notation and problem definition

In this article, we use bold font lowercase letters for vectors, bold font capital letters for matrixes, regular font lowercase letters for scalars, and calligraphy font capital letters for spaces. We call each robot's 2-D coordinates and 1-D yaw angle as state, and the whole system's 2-D coordinates and 1-D yaw angle as pose. The collaborative transportation task has a 6-D state for two robots and a 3-D pose for the system as Fig. 2 shows. Let $\mathcal{X} \subset \mathbb{R}^6$ be the state space of the two robots and $\mathcal{P} \subset \mathbb{R}^3$ be the pose space of the whole system. We use $\mathbf{x} \in \mathcal{X}$ and $\mathbf{p} \in \mathcal{P}$ to represent the state of two robots and the pose of the system, respectively. They can be expressed as

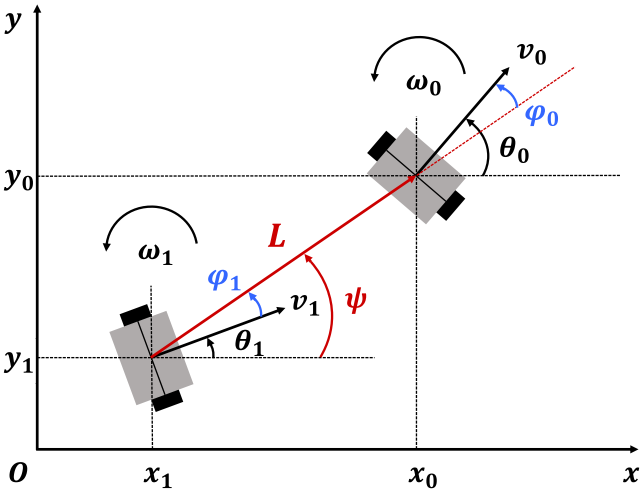


Figure 2. *The kinematic model is with nonholonomic constraints. The system contains two differential robots numbered as* 0 *(leading) and* 1 *(following), respectively. Inter-robot distance (L), system's yaw angle ψ, and robot steering angles relative to the system (φ_0 and φ_1) are also illustrated.*

$$\mathbf{x} = \left[x_0, \quad y_0, \quad \theta_0, \quad x_1, \quad y_1, \quad \theta_1\right]^{\mathrm{T}}, \tag{1a}$$

$$\mathbf{p} = \left[\alpha, \quad \beta, \quad \psi\right]^{\mathrm{T}}, \tag{1b}$$

where

$$\alpha = \frac{x_0 + x_1}{2}, \qquad \beta = \frac{y_0 + y_1}{2}, \tag{2a}$$

$$\psi = \arctan 2\,(y_0 - y_1, x_0 - x_1)\,. \tag{2b}$$

To describe $\mathbf{x}$, we denote k as time step and i, j as robot's IDs specially, namely, $\mathbf{x}_i = \left[x_i, \quad y_i, \quad \theta_i\right]^{\mathrm{T}}$. Let $\mathcal{X}_{\text{obs}} \subset \mathcal{X}$ represent the obstacle space and $\mathcal{X}_{\text{free}} = \mathcal{X} \setminus \mathcal{X}_{\text{obs}}$ be the free space. So does $\mathcal{P}_{\text{obs}} \subset \mathcal{P}$ and $\mathcal{P}_{\text{free}} \subset \mathcal{P}$. $\mathcal{P}_{\text{goal}} \subset \mathcal{P}_{\text{free}}$ is the goal space that can be expressed as a circle $\left\{\mathbf{x} \in \mathcal{X}_{\text{free}} \mid \left\|\mathbf{p} - \mathbf{p}_{\text{goal}}\right\| \leq r\right\}$ with a radius of r. L is to represent the distance between two robots, where $L = [\,(x_0 - x_1)^2 + (y_0 - y_1)^2\,]^{1/2}$. The control input space is $\mathcal{U} \subset \mathbb{R}^4$, which consists of linear velocity v and angular velocity ω for both robots. We use $\mathbf{u} \in \mathcal{U}$ to be the control input of two robots as Eq. (3) shown,

$$\mathbf{u} = \left[v_0, \, w_0, \, v_1, \, w_1\right]^{\mathrm{T}}. \tag{3}$$

The kinematics is with the form of Eq. (4), where k is the discrete time step. The time interval of any adjacent two time steps is assumed as a constant, Δt,

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{B}\mathbf{u}_k \Delta t. \tag{4}$$

Our problem is defined as follows: First, a real-time path planner should solve a feasible global path $\mathbf{g}$:$\mathbb{R} \rightarrow \mathcal{P}_{\text{free}}$ from the initial pose $\mathbf{g}_0 = \mathbf{p}_{\text{init}}$ to the goal $\mathbf{g}_S \in \mathcal{P}_{\text{goal}}$ with step $s \in \{0, 1, 2, \ldots, S\}$. Let $\mathcal{G}$ be the set that contains all the feasible paths. A cost function $J(\mathbf{g})$:$\mathcal{G} \rightarrow \mathbb{R}$, which maps the feasible path $\mathbf{g}$ to a real number, is used to measure the path quality. The optimal path is formulated by

$$\mathbf{g}^* = \arg \min_{\mathbf{g} \in \mathcal{G}} J(\mathbf{g}). \tag{5}$$

Then, a real-time controller will generate a set of control inputs $\mathbf{u}_k \in \mathcal{U}$ for two robots to track the optimal path while holding the distance of two robots to a reference distance ($L \rightarrow L_{\text{ref}}$) and limiting the steering angles of the two robots relative to the queue ($|\varphi_i| \leq \varphi_{i,\max}$). L_{ref} is decided by the number of trolleys in the queue, which can be measured when the queue forms, and $\varphi_{i,\max}$ is the maximum steering angle

https://doi.org/10.1017/S0263574723001005 Published online by Cambridge University Press

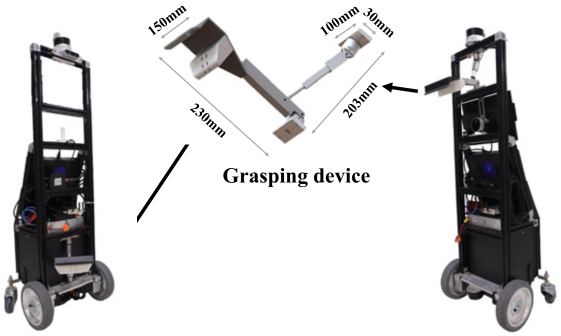


Figure 3. *The physical hardware consists of chassis, grasping device, computing platform, and sensor modules.*

of each robot relative to the system, which can be acquired by experiments. For any feasible path $\mathbf{g} = [\mathbf{g}_0, \mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_S]$, the controller must generate the control inputs to make the system traverse all the intermediate pose $\mathbf{g}_s$ from $\mathbf{g}_0$ to $\mathbf{g}_S$ in turn, where $s \in \{1, 2, 3, \ldots, S-1\}$.

2.2. Preliminary work

We have independently developed two autonomous robots for luggage trolley queue transportation as shown in Fig. 3, which integrate moving chassis, grasping device, computing platform, control, and sensing modules. The two robots have similar structures and electron devices except for the position of grasping device, which is exclusively designed to secure the front and back of luggage trolleys, respectively. Similar robots can ensure the transportation of payloads.

2.3. Localization

In this article, our robot formation is an approximate rigid body structure. A slight error in distance control (e.g., > 5 cm) will cause the formation to fall apart. However, traditional localization methods, including but not limited to Lidar-based, such as AMCL [24], and vision-based, such as ORB-SLAM2 [25] and its variants, cannot directly achieve the required accuracy. In addition, the localization uncertainty of a single robot will cause a more significant error in the distance measurement of robot formation. Therefore, in order to reduce the impact of localization problems in our system, we temporarily use Motion Capture System to provide an accurate positioning to verify the effectiveness of the control strategy.

3. Planning and control strategy

In this section, we will describe the system framework, as shown in Fig. 4, which consists of the global planning algorithm and the divide-and-conquer control strategy with a novel decentralized CBF. We call our robots the leader and follower for convenient description.

3.1. Global planning

Considering that this task deals with a two-robot system with nonholonomic constraints that needs to maintain system formation, traditional path planning algorithms such as basic A* [26] cannot handle the problems with nonholonomic constraints. We have also verified through simulation experiments that these methods are unsuitable for our objective. Therefore, we choose Hybrid A* [8] algorithm, which considers the actual motion constraints of nonholonomic systems, as our global planning algorithm. Compared with basic A*, Hybrid A* has the following characteristics.

https://doi.org/10.1017/S0263574723001005 Published online by Cambridge University Press

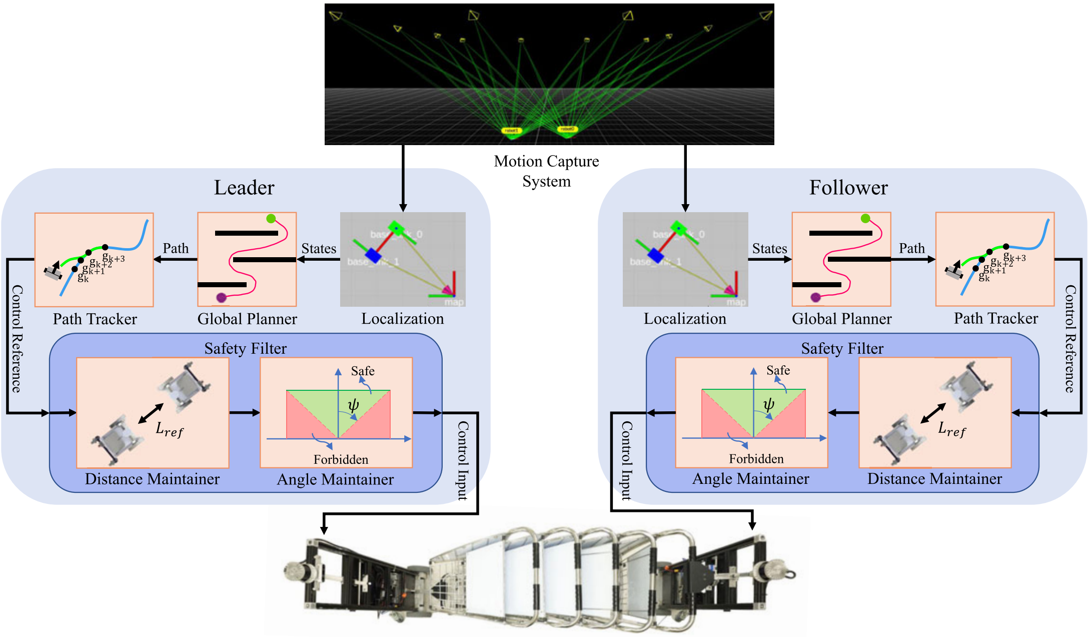


Figure 4. *The framework of our system. The two collaborative robots run the programs with the same modules simultaneously. The location information is all from the Motion Capture System. The photo of two collaborative robots and the luggage trolley queue is from the top view.*

1. Hybrid A* is a heuristic search algorithm in the continuous coordinate system, and can ensure that the generated trajectory satisfies the vehicle nonholonomic constraint.
2. Hybrid A* takes the kinematic characteristics of the vehicle into account by constrained heuristics.
3. In order to improve the search speed and precision, Hybrid A* uses Reeds-Shepp model [27] to augment the search quality.
4. Hybrid A* uses the Voronoi field as the potential field to handle the navigability of narrow openings.
5. An optimization is used to smooth the final path.

Although the kinematics of our system is a little different from a vehicle, the path generated by Hybrid A* is still practical for our system by experimental verification. The path quality and comparison with basic A* will be illustrated in Section 4.1.1.

3.2. Control strategy

After obtaining the optimal path from the global planner, the next step is to achieve how to effectively track the path by two robots with the luggage trolley queue. On this basis, we also need to ensure the maintenance of system formation. We adopt a divide-and-conquer control strategy with the combination of three parts:

1. MPC: track the path generated by the global planner.
2. DT-CLF: maintain the distance between two robots, namely, to maintain the formation.
3. CBF: limit the steering angles of two robots, also to hold the system formation.

In this article, we call the latter two modules together as the safety filter. The functions of the three modules are basically decoupled from each other. Compared with integrating them in a single controller, this divide-and-conquer way has two advantages: higher solution efficiency and larger solution probability. We have tried to fuse them in one optimization problem and found the result unsatisfactory. Sometimes, it needed to add slacks to reduce constraints for solving a feasible solution. Finally, we adopt the mode of splitting into three modules. All nonlinear programs are formulated in CasADi [28] and solved by IPOPT [29].

3.2.1. MPC

Considering the two robots as a whole, we adopt the discrete model. The kinematic model is shown in Fig. 2 and Eq. (6),

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{B}\mathbf{u}_\mathbf{k}\Delta t, \tag{6a}$$

$$\mathbf{x}_k = \begin{bmatrix} x_{0,k}, & y_{0,k}, & \theta_{0,k}, & x_{1,k}, & y_{1,k}, & \theta_{1,k} \end{bmatrix}^{\mathrm{T}}, \tag{6b}$$

$$\mathbf{B} = \begin{bmatrix} \cos\left(\theta_{0,k}\right) & 0 & 0 & 0 \\ \sin\left(\theta_{0,k}\right) & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\left(\theta_{1,k}\right) & 0 \\ 0 & 0 & \sin\left(\theta_{1,k}\right) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{6c}$$

$$\mathbf{u}_\mathbf{k} = \begin{bmatrix} v_{0,k}, & w_{0,k}, & v_{1,k}, & w_{1,k} \end{bmatrix}^{\mathrm{T}}. \tag{6d}$$

In this phase, we pay more attention to make the system's center track the global plan during movement. The constraints on formation control are not strict. Only the last predictive state MPC is required to maintain the distance as L_{ref}. For other states, there is no distance constraint. This not only gives the system a tendency to maintain distance requirements but also avoids problems such as the increase in computing time and infeasible solutions caused by strong constraints. This tendency also ensures that the solution of MPC will not make the two-robot system deviate too far away from the formation, which alleviates the pressure of solving the following modules to a certain extent.

The tracking can be interpreted as an optimization problem. The objective is to minimize the weighted sum of the square of tracking error, $\|\mathbf{p}_N - \mathbf{g}_m\|^2_{\mathbf{Q}_{\mathrm{MPC}}}$, and the square of control inputs $\sum_{k=0}^{N-1} \|\mathbf{u}_k\|^2_{\mathbf{R}_{\mathrm{MPC}}}$, where N is the predictive horizon of MPC, $\mathbf{p}_N$ is the last pose of MPC prediction, $\mathbf{g}_m$ is the tracked pose on the global path, $\mathbf{Q}_{\mathrm{MPC}}$ and $\mathbf{R}_{\mathrm{MPC}}$ are the positive definite weight matrixes. For selecting $\mathbf{g}_m$, first the pose $\mathbf{g}_o$, which is the nearest to the system's current pose, is chosen. Then, $\mathbf{g}_m = \mathbf{g}_{o+s}$ can be calculated by a certain step s. The Euclidean distance between $\mathbf{g}_m$ and $\mathbf{p}_N$ is calculated. Minimizing control inputs to the objective is to control the system by a set of control inputs that are as small as possible. The relevant hard constraints are added to limit the maximum value of linear velocity v and angular velocity ω. Additionally, as mentioned before, we want MPC to express a tendency that holds the system's formation while avoiding too much pressure on the solver in this phase, so we just make a hard constraint that the last state prediction must maintain the reference distance strictly, which means $L_N = L_{\mathrm{ref}}$, where L_N and L_{ref} are the inter-robot distance of the N-th state and reference inter-robot distance respectively. The final formulation of MPC is

$$\min_{\mathbf{u}} \quad \|\mathbf{p}_N - \mathbf{g}_m\|^2_{\mathbf{Q}_{\mathrm{MPC}}} + \sum_{k=0}^{N-1} \|\mathbf{u}_k\|^2_{\mathbf{R}_{\mathrm{MPC}}}, \tag{7a}$$

$$\text{s.t.} \quad -\mathbf{u}_{k,\max} \leq \mathbf{u}_k \leq \mathbf{u}_{k,\max}, \tag{7b}$$

$$\mathbf{u}_k \in \mathcal{U}, \tag{7c}$$

https://doi.org/10.1017/S0263574723001005 Published online by Cambridge University Press

$$L_N = L_{\text{ref}}, \tag{7d}$$

$$k \in \{0, 1, 2, \ldots, N-1\}, \tag{7e}$$

where $\mathbf{u}_k \in \mathbf{U} = \left[\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{N-1}\right]$. Angle constraints are not taken into account in this module. The predicted control input is $\mathbf{U} \subset \mathbb{R}^{4\times N}$. We call $\mathbf{u}_0$ as $\mathbf{u}_{\text{MPC}}$ and send it to the next module.

3.2.2. DT-CLF

The previous step calculates the control input mainly for tracking the global path, while the constraints that guarantee the system's formation are ignored. Although we add a distance constraint on the last predictive state, it only maintains a converging tendency, which is insufficient in practical application. Stronger guarantees on the stabilization of the inter-robot distance are wanted. We leverage DT-CLF to meet the distance requirement.

Definition 1(DT-CLF, based on ref. [30]). *A map $V{:}\mathcal{X} \to \mathbb{R}$ is an exponential control Lyapunov function for the discrete-time control system of Eq. (4) if there exist positive constants κ_1, κ_2, κ_3 and a control input $\mathbf{u}_k{:}\mathcal{X} \to \mathbb{U}, \forall \mathbf{x}_k \in \mathcal{X}$ such that*

$$\kappa_1 \left\|\mathbf{x}_k\right\|^2 \leq V\left(\mathbf{x}_k\right) \leq \kappa_2 \left\|\mathbf{x}_k\right\|^2, \tag{8}$$

and

$$\Delta V\left(\mathbf{x}_k, \mathbf{u}_k\right) + \kappa_3 V\left(\mathbf{x}_k\right) \leq 0, \tag{9}$$

where $\Delta V\left(\mathbf{x}_k, \mathbf{u}_k\right) \triangleq V\left(\mathbf{x}_{k+1}\right) - V\left(\mathbf{x}_k\right)$ with $\mathbf{x}_{k+1}$ following Eq. (4).

We focus on the error between the inter-robot distance and its desired value:

$$\begin{aligned} e_k &\triangleq \mathbf{x}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{D}\mathbf{x} - L_{\text{ref}}^2 \\ &= \left(x_{0,k} - x_{1,k}\right)^2 + \left(y_{0,k} - y_{1,k}\right)^2 - L_{\text{ref}}^2, \end{aligned} \tag{10}$$

with

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \end{bmatrix}. \tag{11}$$

Then, we design the candidate DT-CLF for distance stabilization as

$$V(e_k) \triangleq \left(e_k\right)^2. \tag{12}$$

Obviously, Eq. (12) satisfies Eq. (8) by $0 < \kappa_1 < 1$ and $\kappa_2 > 1$. To render Eq. (12) as a valid DT-CLF, we pose the following CLF constraint for our dual-robot system Eq. (6) to ensure the origin is exponentially stable:

$$\Delta V\left(e_k, \mathbf{u}_k\right) + cV\left(e_k\right) \leq 0, \tag{13}$$

where $\mathbf{x}_{k+1}$ (associated with e_{k+1}) follows the robots' kinematics Eq. (6).

We expect to modify $\mathbf{u}_{\text{MPC}}$ as slightly as possible while holding the DT-CLF constraints to enable both path tracking and distance maintenance. Therefore, the optimization in this phase is formulated as

$$\min_{\mathbf{u}} \left\|\mathbf{u} - \mathbf{u}_{\text{MPC}}\right\|_{\mathbf{R}_{\text{CLF}}}^2, \tag{14a}$$

$$\text{s.t.} \quad \Delta V(e, \mathbf{u}) + cV(e) \leq 0, \tag{14b}$$

$$-\mathbf{u}_{\max} \leq \mathbf{u} \leq \mathbf{u}_{\max}, \tag{14c}$$

$$\mathbf{u} \in \mathcal{U}. \tag{14d}$$

We call the optimization result as $\mathbf{u}_{\text{CLF}}$ and sent it to the next module.

https://doi.org/10.1017/S0263574723001005 Published online by Cambridge University Press

3.2.3. CBF

Considering the above-mentioned mechanical designs of the robot, there exists a physical limit for the angle between the grasping device and the robot chassis. Respecting this angle constraint is essential for the system's safe and successful operation. For practical consideration while without loss of theoretical guarantees, we use CBF and formulate a quadratic programming (CBF-QP) [31,32] to ensure the system's safety. In this subsection, we shall first construct the CBFs for our system and then pose the optimization problem. In addition, we present a distributed derivation toward the optimization problem for the sake of scalability.

Definition 2(CBF, based on ref. [31]). *Let $\mathcal{C} \subset \mathcal{X} \subset \mathbb{R}^n$ be the superlevel set of a continuously differentiable function $h : \mathcal{X} \to \mathbb{R}$, then h is a control barrier function (CBF) if it satisfies*

$$\dot{h}(\mathbf{x}, \mathbf{u}) \geq -\gamma h(\mathbf{x}) \tag{15}$$

for a constant $\gamma > 0$ and for all $\mathbf{x} \in \mathcal{X}$.

We first formulate the CBFs for the system concerning the angle constraints. As shown in Fig. 2, we denote the angle between the luggage trolley queue and each robot as φ_i for $i \in \{0, 1\}$. The safe operating zones of these angles are $-\varphi_{i,\max} \leq \varphi_i \leq \varphi_{i,\max}$, $i \in \{0, 1\}$. For each robot, we choose the safety value function h_i: $\mathcal{X} \to \mathbb{R}$ as

$$h_i(\mathbf{x}) \triangleq \frac{1}{2} \left(\varphi_{i,\max}^2 - \varphi_i^2\right). \tag{16}$$

Accordingly, the safe set for the entire system reads as the intersection of the zero-superlevel sets of the continuously differentiable functions h_i for $i \in \{0, 1\}$:

$$\mathcal{C} \triangleq \bigcap_{i\in\{0,1\}} \{\mathbf{x} \in \mathcal{X}_{\text{free}} : h_i(\mathbf{x}) \geq 0\}. \tag{17}$$

Without loss of generality, we take robot i as an example. To make h_i a valid CBF, we shall ensure the following:

$$\dot{h}_i + \gamma_i h_i \geq 0, \tag{18}$$

where γ_i is a positive constant. Differentiating h_i with respect to time, we get

$$\begin{aligned} \dot{h}_i &= -\varphi_i \dot{\varphi}_i \\ &= -\varphi_i \left(\dot{\psi} - \dot{\theta}_i\right) \\ &= -\varphi_i \left(\frac{\partial \psi}{\partial \mathbf{x}_i}^{\mathrm{T}} \dot{\mathbf{x}}_i + \frac{\partial \psi}{\partial \mathbf{x}_j}^{\mathrm{T}} \dot{\mathbf{x}}_j - \omega_i \right). \end{aligned} \tag{19}$$

where the first partial is associated with robot i's state and movements and the second relates to the other robot j's movements. Naturally, the angle between the robot i and the luggage trolley queue is also up to the other robot for that the other robot is simultaneously manipulating the queue as well. The first partial in the parenthesis of Eq. (19) is

$$\frac{\partial \psi}{\partial \mathbf{x}_i} = \left[\frac{y_j - y_i}{L^2}, \frac{x_i - x_j}{L^2}, 0 \right]^{\mathrm{T}}. \tag{20}$$

https://doi.org/10.1017/S0263574723001005 Published online by Cambridge University Press

Further combining the differential-drive model of each robot yields

$$\begin{aligned}\frac{\partial\psi}{\partial\mathbf{x}_i}^{\mathrm{T}}\dot{\mathbf{x}}_i &= \left[\frac{y_j-y_i}{L^2},\ \frac{x_i-x_j}{L^2},\ 0\right]\begin{bmatrix} v_i\cos\theta_i\\ v_i\sin\theta_i\\ \omega_i\end{bmatrix}\\ &= \frac{v_i}{L^2}\left(\left(y_j-y_i\right)\cos\theta_i+\left(x_i-x_j\right)\sin\theta_i\right)\\ &= \frac{v_i}{L^2}\cdot\overrightarrow{L}^{-\perp}\cdot\overrightarrow{\theta_i}\\ &= \frac{v_i}{L}\cdot\overrightarrow{\psi}^{-\perp}\cdot\overrightarrow{\theta_i}, \qquad (21)\end{aligned}$$

where $\overrightarrow{L}=L\left[\cos(\psi),\sin(\psi)\quad\right]^{\mathrm{T}}$, $\overrightarrow{\theta_i}=\left[\cos(\theta_i),\sin(\theta_i)\quad\right]^{\mathrm{T}}$, $\overrightarrow{\psi}=\left[\cos(\psi),\sin(\psi)\quad\right]^{\mathrm{T}}$, $z^{-\perp}$ denotes rotating a planar vector z by $-\frac{\pi}{2}$ rad, and $z^{\perp}$ means a $\frac{\pi}{2}$ rotation. In a similar spirit, the second term in the parenthesis of Eq. (19) is derived as

$$\frac{\partial\psi}{\partial\mathbf{x}_j}^{\mathrm{T}}\dot{\mathbf{x}}_j=\frac{v_j}{L}\cdot\overrightarrow{\psi}^{\perp}\cdot\overrightarrow{\theta_j}. \qquad (22)$$

Taking Eqs. (21) and (22) back into Eq. (19), we have

$$\dot{h}_i=\frac{\varphi_i}{L}\begin{bmatrix}-\overrightarrow{\psi}^{-\perp}\cdot\overrightarrow{\theta_i}\\ L\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}v_i\\ \omega_i\end{bmatrix}+\frac{\varphi_i}{L}\begin{bmatrix}-\overrightarrow{\psi}^{\perp}\cdot\overrightarrow{\theta_j}\\ 0\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}v_j\\ \omega_j\end{bmatrix}. \qquad (23)$$

Now, we have the expression of $\dot{h}_i$. Substituting Eq. (23) into Eq. (18), we have

$$\frac{\varphi_i}{L}\begin{bmatrix}-\overrightarrow{\psi}^{-\perp}\cdot\overrightarrow{\theta_i}\\ L\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}v_i\\ \omega_i\end{bmatrix}+\frac{\varphi_i}{L}\begin{bmatrix}-\overrightarrow{\psi}^{\perp}\cdot\overrightarrow{\theta_j}\\ 0\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}v_j\\ \omega_j\end{bmatrix}+\gamma_i h_i\geq 0. \qquad (24)$$

For notation’s brevity, we further simplify Eq. (23) by defining $\mathbf{A}_i\triangleq\left[\mathbf{A}_{ii}^{\mathrm{T}},\ \mathbf{A}_{ij}^{\mathrm{T}}\right]$, wherein

$$\mathbf{A}_{ii}\triangleq\frac{\varphi_i}{L}\begin{bmatrix}\overrightarrow{\psi}^{-\perp}\cdot\overrightarrow{\theta_i}\\ -L\end{bmatrix}, \qquad (25a)$$

$$\mathbf{A}_{ij}\triangleq\frac{\varphi_i}{L}\begin{bmatrix}\overrightarrow{\psi}^{\perp}\cdot\overrightarrow{\theta_j}\\ 0\end{bmatrix}. \qquad (25b)$$

Then, we can write condition Eq. (24) as a linear constraint with respect to control

$$\mathbf{A}_i\mathbf{u}\leq\gamma_i h_i. \qquad (26)$$

Finally, by applying the CBF constraints of all robots, we can formulate the filter as an optimization problem, resulting in a CBF-QP:

$$\min_{\mathbf{u}}\quad \|\mathbf{u}-\mathbf{u}_{\mathrm{CLF}}\|^2_{\mathbf{R}_{\mathrm{CBF}}}, \qquad (27a)$$

$$\text{s.t.}\quad \mathbf{A}_i\mathbf{u}\leq\gamma_i h_i,\quad \forall i\neq j, \qquad (27b)$$

$$-\mathbf{u}_{\max}\leq\mathbf{u}\leq\mathbf{u}_{\max}, \qquad (27c)$$

$$\mathbf{u}\in\mathcal{U}, \qquad (27d)$$

where $\mathbf{u}_{\mathrm{CLF}}$ is the filtered control input provided in Section 3.2.2.

Observing Eq. (27), one can see that the optimization is in a centralized sense because it involves all of the robots’ states and controls. This optimization formulation scales poorly with respect to the number of robots and thus is not a good choice for real-time execution. Inspired by ref. [33], we shall resolve this problem by splitting the constraints so the CBF-QP can be solved distributedly by each robot without compromising the theoretical guarantees.

https://doi.org/10.1017/S0263574723001005 Published online by Cambridge University Press

Here we define $\mathbf{u}_i \in \mathcal{U}_i$ for each robot analogically, where $\mathcal{U}_i \subset \mathbb{R}^2$ is the control input space for each robot. Splitting Eq. (27b), for each coupled pair of robots i and j (there is only one in our case), we have

$$\mathbf{A}_{ii}^{\mathrm{T}}\mathbf{u}_i + \mathbf{A}_{ij}^{\mathrm{T}}\mathbf{u}_j \leq \gamma_i h_i. \tag{28}$$

Similarly, for CBF of robot j, we have the analogous separation with

$$\mathbf{A}_{ji}^{\mathrm{T}}\mathbf{u}_i + \mathbf{A}_{jj}^{\mathrm{T}}\mathbf{u}_j \leq \gamma_j h_j. \tag{29}$$

Further separating terms with respect to different robots' controls yields

$$\mathbf{A}_{ii}^{\mathrm{T}}\mathbf{u}_i \leq \frac{\eta_{ii}}{\eta_{ii}+\eta_{ij}}\gamma_i h_i, \tag{30a}$$

$$\mathbf{A}_{ji}^{\mathrm{T}}\mathbf{u}_i \leq \frac{\eta_{ji}}{\eta_{ji}+\eta_{jj}}\gamma_j h_j, \tag{30b}$$

for robot i, and

$$\mathbf{A}_{ij}^{\mathrm{T}}\mathbf{u}_j \leq \frac{\eta_{ij}}{\eta_{ij}+\eta_{ii}}\gamma_i h_i, \tag{31a}$$

$$\mathbf{A}_{jj}^{\mathrm{T}}\mathbf{u}_j \leq \frac{\eta_{jj}}{\eta_{jj}+\eta_{ji}}\gamma_j h_j, \tag{31b}$$

for robot j, where η_{**} are positive parameters. One can always verify that adding up terms from both sides of Eqs. (30a) and (31a) yields Eq. (28), and adding Eqs. (30b) and (31b) yields Eq. (29). For simplicity, we denote

$$\gamma_{ii} \triangleq \frac{\eta_{ii}}{\eta_{ii}+\eta_{ij}}\gamma_i, \tag{32a}$$

$$\gamma_{ji} \triangleq \frac{\eta_{ji}}{\eta_{ji}+\eta_{jj}}\gamma_j. \tag{32b}$$

Therefore, the decentralized CBF-QP for each robot to solve will bes

$$\begin{aligned}
\min_{\mathbf{u}_i} \quad & \left\|\mathbf{u}_i - \mathbf{u}_{\mathrm{CLF},i}\right\|^2_{\mathbf{R}_{\mathrm{CBF},i}}, && (33a)\\
\text{s.t.} \quad & A_{ii}\mathbf{u}_i \leq \gamma_{ii} h_i, && (33b)\\
& A_{ji}\mathbf{u}_i \leq \gamma_{ji} h_j, && (33c)\\
& -\mathbf{u}_{i,\max} \leq \mathbf{u}_i \leq \mathbf{u}_{i,\max}, && (33d)\\
& \mathbf{u}_i \in \mathcal{U}_i. && (33e)
\end{aligned}$$

After being filtered by DT-CLF and CBF, the control input commands $\mathbf{u}_{\mathrm{safe}}$ will be ultimately sent to the corresponding actuators.

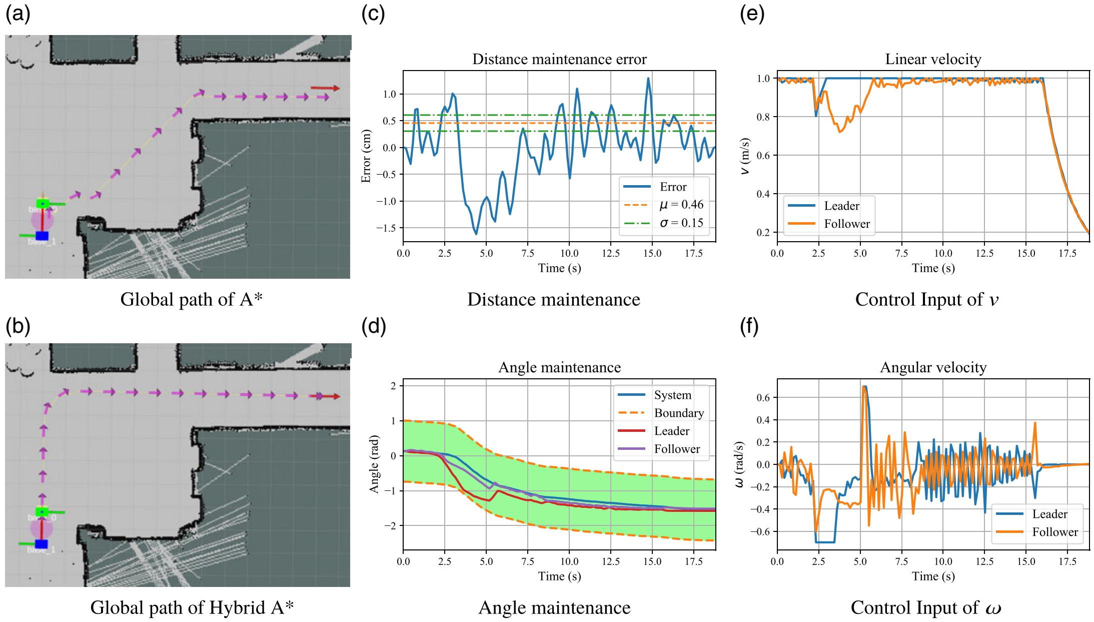
4. Experiments

In this section, the experiment results of the proposed control strategy as well as the high-level global planner are shown. The experiments consist of two parts. The first part is to verify the smoothness of the path solved by the global planner. The second part is to verify the formation maintenance of inter-robot distance and angles by control strategy while tracking the global path. The former is completed on the simulation platform, while the latter is completed on both the simulation and physical hardware platforms. For the physical experiments, we design three representative path shapes constructed by different initial and goal positions: straight line, right-angle curve, and U-shape curve. These shapes are the dominating path conditions that the robots mainly deal with in real-world scenarios.

https://doi.org/10.1017/S0263574723001005 Published online by Cambridge University Press

Table I. *The value of pivotal parameters.*

Parameter	Description	Value
N	MPC predictive horizon	10
c	DT-CLF convergence rate	0.5
γ_i	CBF convergence rate	5
$\varphi_{i,max}$	Maximum steering angle for safety	50°



Global path of A*

Distance maintenance

Control Input of v

Global path of Hybrid A*

Angle maintenance

Control Input of ω

Figure 5. *Simulation experiment results. For distance maintenance, the mean and standard deviation are indicated. For angle maintenance, the safe region is shown with the boundary and green shadow.*

In order to facilitate the readers who have interests to reproduce our work, Table I provides the value of pivotal parameters and the corresponding description. Because the two robots share the same parameters, for brevity, we use robot i to represent.

4.1. Simulation experiments

4.1.1. Planning

We use Hybrid A* to plan a path that conforms to the nonholonomic constraints of the system. Compared with A*, the path is much smoother and with better performability. We use a right-angle curve for demonstration. The visualization of path quality is illustrated in Fig. 5(a) and (b). Obviously, Hybrid A* provides a more viable path. Figure 5(b) is the path that we use in simulation experiments. The computing time is also acceptable, which is under 100 ms on our computing platform. For the physical experiments, the leading robot must face back to the following one to grasp the luggage trolley queue as shown in Fig. 1. We also adjust the frame of the leading robot in simulation correspondingly.

4.1.2. Control strategy

Figure 5(c) and (d) demonstrate the distance and angle maintenance of the control strategy. The distance between the two robots remains obviously stable near the reference, and the error ($<$ 2 cm) is acceptable. The mean and standard deviation of distance maintenance error are calculated by the absolute value of

https://doi.org/10.1017/S0263574723001005 Published online by Cambridge University Press

each sample in case the positive errors counteract the negative errors. In the following parts, the processes of distance maintenance error are the same. The steering angles of the leader and follower remain in the safe region consistently. The computing time of the safety filter is 21 ms, which is comparable to the MPC computing time of 22 ms. That is also what we are looking for. Figure 5(e) and (f) show the ultimate control inputs.

4.2. Physical hardware experiments

We implement the proposed planning algorithm and control strategy on physical hardware robots. The robot-mounted computing platform is NUC with Intel i7-1165G7 CPU and 16-GB RAM. Each transported luggage trolley is with 4 omnidirectional wheels. The weight is 15 kg and the length is 790 mm for each. We put five luggage trolleys into a queue for physical experiments. The bond of adjacent trolleys is just by the trolley-mounted fastener without any other additional connection. The overall length of our system is more than 2 m. For activation, we use warm-up, which accelerates the robots to the specified control input in two seconds. The experiment results are shown in Fig. 6 and the detailed experimental design is shown as follows.

4.2.1. Straight line

The system moves along a straight line shown in Fig. 6(a), which is the easiest and the most common path. The proposed method achieves a distance maintenance error of less than 2.5 cm as Fig. 6(b) shows. Figure 6(c) angle constraints bring almost no impact in this case. Only the distance constraints will influence system formation. Figure 6(d) and (e) illustrate the control inputs of this case.

4.2.2. Right-angle curve

The system turns a right-angle curve as shown in Fig. 6(f). Distance constraints are also dominated because the rotation angle is relatively small. Distance error is under 1.5 cm as presented in Fig. 6(g). Figure 6(h) shows that the steering angles of two robots are also limited in the safe region obviously. Figure 6(i) and (j) illustrate the control inputs of this case.

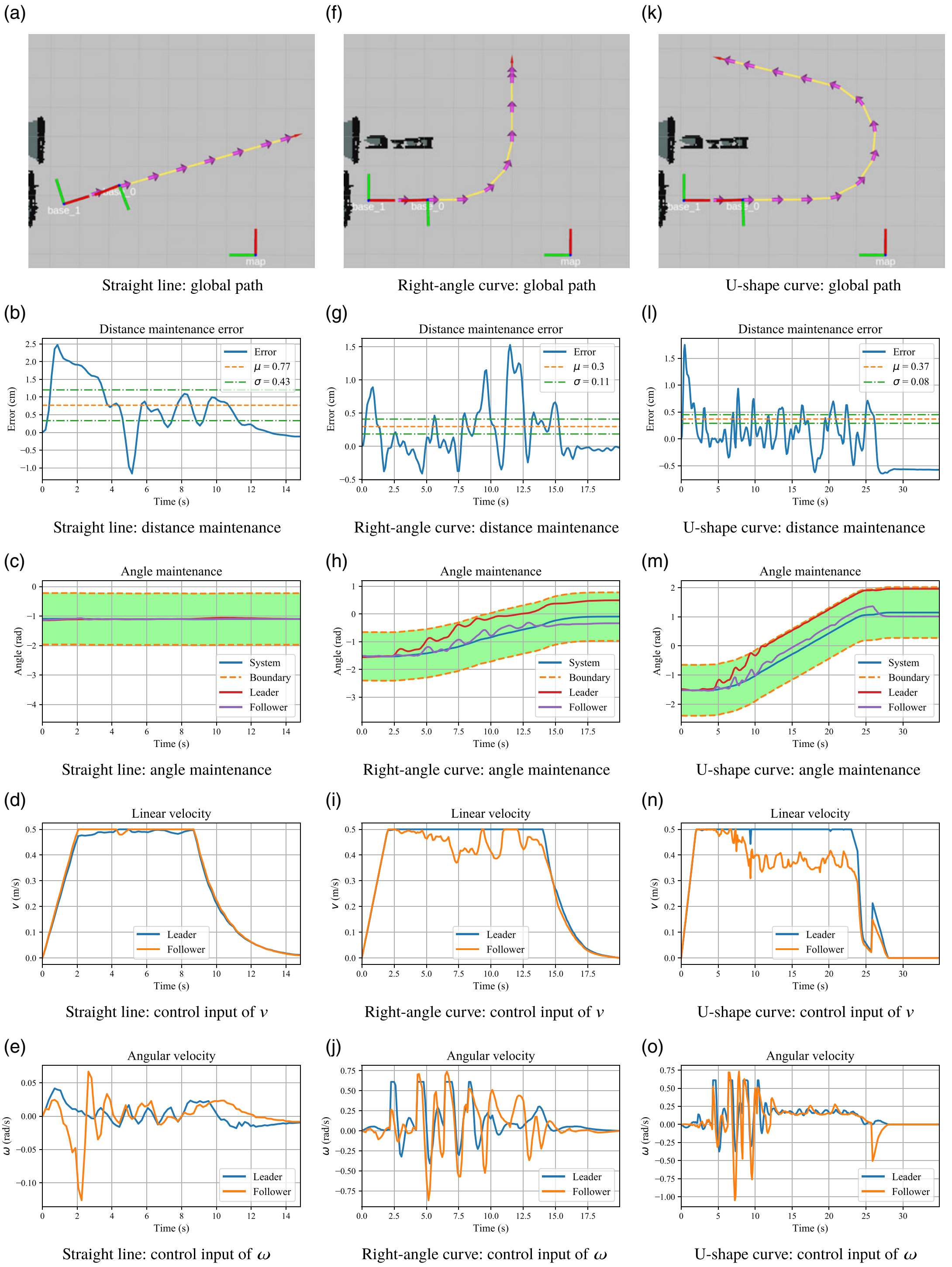
4.2.3. U-shape curve

The system turns a U-shape curve with about 150° shown in Fig. 6(k), which is the most difficult. Due to the nonholonomic constraints and mechanical limitations, the system will deconstruct if the steering angles exceed the maximum values. According to experiment measurement, the maximum steering angle is about 60° for both robots. In view of the safety margin, we set the maximum value as 50° in CBF. We have tried that if CBF is removed, the system will fall apart when tracking this U-shape path. Figure 6(l) demonstrates the distance maintenance error is under 1.8 cm. Attractively, the robot steering angles are still located in the safe region as shown in Fig. 6(m), although the path curve is very sharp compared with the system length. The leader's steering angle presses close to the boundary because it wants to turn. However, CBF guarantees that it will never exceed the safe region and further guarantees the system's formation. Figure 6(n) and (o) illustrate the control inputs of this case.

4.3. Summary

Both the simulation and physical hardware experiments show the effectiveness of the proposed method. The collaborative robots can drive the luggage trolley queue to move from the initial pose to the goal region via a smooth path. While moving, the collaborative robots can maintain the system's formation. The distance maintenance error is less than 2.5 cm for all the conditions. The steering angles for the robots are always maintained in the safe region. The errors and computing times are all acceptable.

https://doi.org/10.1017/S0263574723001005 Published online by Cambridge University Press



Straight line: global path

Right-angle curve: global path

U-shape curve: global path

Straight line: distance maintenance

Right-angle curve: distance maintenance

U-shape curve: distance maintenance

Straight line: angle maintenance

Right-angle curve: angle maintenance

U-shape curve: angle maintenance

Straight line: control input of v

Right-angle curve: control input of v

U-shape curve: control input of v

Straight line: control input of ω

Right-angle curve: control input of ω

U-shape curve: control input of ω

Figure 6. *Physical experiment results : (a)-(e) are for the straight line, (f)-(j) are for the right-angle curve, and (k)-(o) are for the U-shape curve. For each case, the global path, distance, and angle maintenance effectiveness, ultimate control inputs of v and ω are provided. For distance maintenance and angle maintenance, the processes are the same as Fig. 5.*

5. Conclusion and future work

This article proposes a divide-and-conquer control strategy for transporting a luggage trolley queue. For the multi constraints in this problem, we decouple and solve them one by one. Before the controller, we use Hybrid A* to generate a global path that considers the nonholonomic constraints. Then, we use a real-time MPC to track the global path. Followed by a safety filter, which consists of DT-CLF and CBF, the controller will output a $\mathbf{u}_{\text{safe}}$ that can drive the system move to the designated goal region while holding the distance and angle constraints. We verified the effectiveness of our method through both the simulation platform and the physical hardware platform.

Subject to the precise localization of the Motion Capture System, we cannot do physical hardware experiments in a larger environment. Therefore, we will focus on state estimation to enhance the localization performance of Lidar and extend our experimental scenarios to more complex environments in future work. Additionally, avoiding pedestrians and obstacles in dynamic environments is also the next stage of our work.

Author contributions. Xuheng Gao and Hao Luan conceived and designed this work. Hao Luan set up the simulation platform. Xuheng Gao and Bingyi Xia conducted the physical hardware experiments. Ziqi Zhao designed the hardware. Xuheng Gao and Hao Luan wrote this article. Jiankun Wang and Max Q.-H. Meng provided guidance and supervision.

Financial support. This research is supported by ShenZhen Key Laboratory of Robotic Perception and Intelligence (ZDSYS20200810171800001), National Natural Science Foundation of China Grant #62103181, and Shenzhen Outstanding Scientific and Technological Innovation Talents Training Project under Grant RCBS20221008093305007.

Competing interests. The authors declare no Competing interest exist.

Ethical approval. Not applicable.

References

[1] J. Wang and M. Q.-H. Meng, "Real-time decision making and path planning for robotic autonomous luggage trolley collection at airports," *IEEE Trans. Syst. Man Cybern. Syst.* **52**(4), 2174–2183 (2022).

[2] J. Wang and M. Q.-H. Meng. Path Planning for Nonholonomic Multiple Mobile Robot System with Applications to Robotic Autonomous Luggage Trolley Collection at Airports. **In:** *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2020) pp. 2726–2733.

[3] J. Pan, X. Mai, C. Wang, Z. Min, J. Wang, H. Cheng, T. Li, E. Lyu, L. Liu and M. Q.-H. Meng, "A searching space constrained partial to full registration approach with applications in airport trolley deployment robot," *IEEE Sens. J.* **21**(10), 11946–11960 (2021).

[4] A. Xiao, H. Luan, Z. Zhao, Y. Hong, J. Zhao, W. Chen, J. Wang and M. Q.-H. Meng. Robotic Autonomous Trolley Collection with Progressive Perception and Nonlinear Model Predictive Control. **In:** *2022 IEEE International Conference on Robotics and Automation (ICRA)* (2022) pp. 4480–4486.

[5] S. Gilroy, D. Lau, L. Yang, E. Izaguirre, K. Biermayer, A. Xiao, M. Sun, A. Agrawal, J. Zeng, Z. Li and K. Sreenath. Autonomous Navigation for Quadrupedal Robots with Optimized Jumping Through Constrained Obstacles. **In:** *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, IEEE (2021) pp. 2132–2139.

[6] Z. Jian, Z. Lu, X. Zhou, B. Lan, A. Xiao, X. Wang and B. Liang. Putn: A Plane-fitting Based Uneven Terrain Navigation Framework. **In:** *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE (2022) pp. 7160–7166.

[7] A. Xiao, W. Tong, L. Yang, J. Zeng, Z. Li and K. Sreenath. Robotic Guide Dog: Leading a Human with Leash-Guided Hybrid Physical Interaction. **In:** *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE (2021) pp. 11470–11476.

[8] D. Dolgov, S. Thrun, M. Montemerlo and J. Diebel, "Practical search techniques in path planning for autonomous driving," *Ann. Arbor* **1001**(48105), 18–80 (2008).

[9] Y. Hu, M. Cui, J. Duan, W. Liu, D. Huang, A. Knoll and G. Chen, "Model predictive optimization for imitation learning from demonstrations," *Rob. Auton. Syst.* **163**, 104381 (2023).

[10] X. An, C. Wu, Y. Lin, M. Lin, T. Yoshinaga and Y. Ji, "Multi-robot systems and cooperative object transport: Communications, platforms, and challenges," *IEEE Open J. Comput. Soc.* **4**, 23–36 (2023).

[11] J. Bohren, R. B. Rusu, E. G. Jones, E. Marder-Eppstein, C. Pantofaru, M. Wise, L. Mösenlechner, W. Meeussen and S. Holzer. Towards Autonomous Robotic Butlers: Lessons Learned with the PR2. **In:** *2011 IEEE International Conference on Robotics and Automation* (2011) pp. 5568–5575.

[12] M.Wise, M.Ferguson, D.King, E.Diehr and D. Dymesich, Fetch & Freight: Standard Platforms for Service Robot Applications. **In:** *Workshop on autonomous mobile service robots* (2016) pp. 1–6.

https://doi.org/10.1017/S0263574723001005 Published online by Cambridge University Press

[13] D. Sirintuna, A. Giammarino and A. Ajoudani. Human-Robot Collaborative Carrying of Objects with Unknown Deformation Characteristics. **In:** *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2022) pp. 10681–10687.

[14] H. Zhang, Q. Sheng, J. Hu, X. Sheng, Z. Xiong and X. Zhu, "Cooperative transportation with mobile manipulator: A capability map-based framework for physical human-robot collaboration," *IEEE/ASME Trans. Mech.* **27**(6), 4396–4405 (2022).

[15] J. Hu, W. Liu, H. Zhang, J. Yi and Z. Xiong, "Multi-robot object transport motion planning with a deformable sheet," *IEEE Rob. Autom. Lett.* **7**(4), 9350–9357 (2022).

[16] C. Wang, X. Mai, D. Ho, T. Liu, C. Li, J. Pan and M. Q.-H. Meng, "Coarse-to-fine visual object catching strategy applied in autonomous airport baggage trolley collection," *IEEE Sens. J.* **21**(10), 11844–11857 (2021).

[17] C.-F. Juang, C.-H. Lu and C.-A. Huang, "Navigation of three cooperative object-transportation robots using a multistage evolutionary fuzzy control approach," *IEEE Trans. Cybern.* **52**(5), 3606–3619 (2022).

[18] J. Hu, P. Bhowmick and A. Lanzon, "Group coordinated control of networked mobile robots with applications to object transportation," *IEEE Trans. Veh. Technol.* **70**(8), 8269–8274 (2021).

[19] D. Koung, O. Kermorgant, I. Fantoni and L. Belouaer, "Cooperative multi-robot object transportation system based on hierarchical quadratic programming," *IEEE Rob. Autom. Lett.* **6**(4), 6466–6472 (2021).

[20] A. Tsiamis, C. P. Bechlioulis, G. C. Karras and K. J. Kyriakopoulos. Decentralized Object Transportation by Two Nonholonomic Mobile Robots Exploiting Only Implicit Communication. **In:** *2015 IEEE International Conference on Robotics and Automation (ICRA)* (2015) pp. 171–176.

[21] Z. Wang and M. Schwager. Kinematic Multi-robot Manipulation with no Communication Using Force Feedback. **In:** *2016 IEEE International Conference on Robotics and Automation (ICRA)* (2016) pp. 427–432.

[22] Z. Wang and M. Schwager. Multi-Robot Manipulation with no Communication Using Only Local Measurements. **In:** *2015 54th IEEE Conference on Decision and Control (CDC)* (2015) pp. 380–385.

[23] B. Hichri, L. Adouane, J. C. Fauroux, Y. Mezouar and I. Doroftei, "Flexible co-manipulation and transportation with mobile multi-robot system," *Assem. Autom.* **39**(3), 422–431 (2019).

[24] F. Dellaert, D. Fox, W. Burgard and S. Thrun. Monte Carlo Localization for Mobile Robots. **In:** *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat No.99CH36288C)*, vol. 2 (1999) pp. 1322–1328.

[25] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source slam system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Rob.* **33**(5), 1255–1262 (2017).

[26] P. E. Hart, N. J. Nilsson and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.* **4**(2), 100–107 (1968).

[27] J. Reeds and L. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pac. J. Math.* **145**(2), 367–393 (1990).

[28] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings and M. Diehl, "Casadi: A software framework for nonlinear optimization and optimal control," *Math. Program. Comput.* **11**, 1–36 (2018).

[29] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.* **106**(1), 25–57 (2006).

[30] A. Agrawal and K. Sreenath, "Discrete Control Barrier Functions for Safety-critical Control of Discrete Systems with Application to Bipedal Robot Navigation," **In:** *Robotics: Science and Systems 2017.* vol. 13, Cambridge, MA, USA.

[31] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath and P. Tabuada. Control Barrier Functions: Theory and Applications. **In:** *2019 18th European control conference (ECC)*, IEEE (2019) pp. 3420–3431.

[32] A. D. Ames, X. Xu, J. W. Grizzle and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Autom. Control* **62**(8), 3861–3876 (2016).

[33] L. Wang, A. D. Ames and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Trans. Rob.* **33**(3), 661–674 (2017).

Cite this article: X. Gao, H. Luan, B. Xia, Z. Zhao, J. Wang and M. Q.-H. Meng (2023). "A divide-and-conquer control strategy with decentralized control barrier function for luggage trolley transportation by collaborative robots", Robotica **41**, 3333–3348. https://doi.org/10.1017/S0263574723001005