

# Cooperatively Gathering Scattered Objects with a Tethered Robot Duo

Yao Su<sup>1</sup> Yuhong Jiang<sup>1</sup> Yixin Zhu<sup>1,2</sup> Hangxin Liu<sup>1</sup>

**Abstract**—We devise a cooperative planning framework to generate optimal trajectories for a tethered robot duo, who is tasked to gather scattered objects spread in a large area using a flexible net. Specifically, the proposed planning framework first produces a set of dense waypoints for each robot, serving as the initialization for optimization. Next, we formulate an iterative optimization scheme to generate smooth and collision-free trajectories while ensuring cooperation within the robot duo to efficiently gather objects and properly avoid obstacles. We validate the generated trajectories in simulation and implement them in physical robots using Model Reference Adaptive Controller (MRAC) to handle unknown dynamics of carried payloads. In a series of studies, we find that: (i) a U-shape cost function is effective in planning cooperative robot duo, and (ii) the task efficiency is not always proportional to the tethered net's length. Given an environment configuration, our framework can gauge the optimal net length. To our best knowledge, ours is the first that provides such estimation for tethered robot duo.

**Index Terms**—Cooperative path planning, optimization, tethered WMRs, adaptive control

## I. INTRODUCTION

We consider the task of gathering and carrying scattered objects on the floor by deploying two Wheeled Mobile Robots (WMRs) tethered by a flexible net or rope as a duo. Compared to picking and placing individual objects one by one, this setting is more intriguing and compelling for robots to collect small items spread across a large area autonomously. Fig. 1a showcases two similar tasks in the physical world that share a similar spirit, wherein humans adopt a net to collect scattered objects with high efficiency.

For a tethered robot duo to accomplish this task, the algorithm must resolve two crucial challenges on planning and implementation. First, how to effectively generate **cooperative** trajectories? Despite that cooperative planning affords robots to accomplish more complex and dexterous tasks comparing with a single robot [1], it also introduces additional constraints on robots' behavior once they are physically connected. Specifically, when the connection is rigid (*e.g.*, [2, 3]), the whole system can be modeled as one rigid body with well-defined dynamics for path planning and trajectory tracking. However, planning the robot duo with a non-rigid connection (*e.g.*, a rope or a net) requires special attentions [4, 5]; one has

Manuscript received: September 9, 2021; Revised: November 25, 2021; Accepted: December 22, 2021. This paper was recommended for publication by Editor Lucia Pallottino upon evaluation of the Associate Editor and Reviewers' comments. (Corresponding author: Hangxin Liu)

<sup>1</sup> Beijing Institute for General Artificial Intelligence (BIGAI).

<sup>2</sup> Institute for Artificial Intelligence, Peking University.

Emails: {suyao, jiangyuhong, y, liuhx}@bigai.ai  
Digital Object Identifier (DOI): see top of this page.

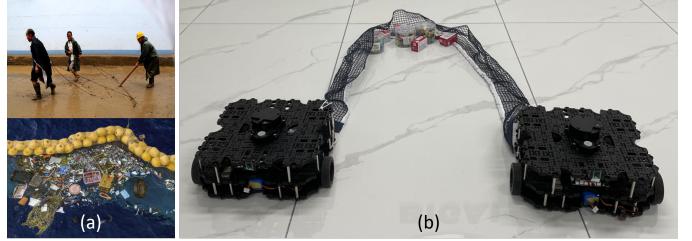


Fig. 1: (a) Two examples of similar setups in daily life, wherein humans use long/flexible structures to gather scattered objects (Source: online). (b) Two wheeled robots are tethered by a net as a duo. Our framework **cooperatively** plans the robot duo's trajectories, indirectly maintaining the net's shape through motions. It also handles the **uncertainty** of payloads using Model Reference Adaptive Controller (MRAC) to properly track the trajectories during operation.

to consider the task goals, obstacle avoidance, and net shape maintenance between each individual of the duo. Second, the tethered robot duo is subjected to **increasing payloads** during operations. In addition to the friction force initially introduced by the net, the dragging force will increase significantly and deteriorate the trajectory tracking of each robot as the task progresses and more objects are gathered. Of note, this challenge is mostly overlooked in prior literature [4–7].

In this paper, we devise an optimization-based cooperation framework for a pair of tethered WMRs (*i.e.*, a duo) to gather and transport objects with a net; see Fig. 1b for the setup. Laying at the core is a trajectory optimization scheme that produces smooth trajectories for the robot duo, which accounts for task goals, obstacle avoidance, and net shape maintenance. A set of waypoints extended from a centerline connecting all the target objects along the robots' initial and final configurations serve as the initialization for the optimization scheme. A MRAC is further implemented on the robot duo to stably track the produced trajectories under unknown increasing payloads. Our framework is verified in both simulation and experiments.

## A. Related Work

Although the field of **cooperative planning** has incubated many successful applications, such as search and rescue [8], exploration [9], object manipulation [10, 11], payload transportation [12, 13], the planning problem of tethered robot duo to date is less explored. Prior work [4, 14] mostly formulates it as a separation problem; the goal is to find trajectories for the robot duo to separate all the objects from obstacles. Of note, prior methods assume an infinite separation distance between the robots, unpractical and unrealistic in many cases. Teshnizi *et al.* [5] relax this assumption and introduce a modified A\* planning algorithm for a tethered robot duo under specific

tether lengths. Nevertheless, it fails to properly account for the obstacle avoidance (*e.g.*, in final configuration) and net shape maintenance, which we fully address in this work.

Gathering objects using the robot duo necessitates maintaining the shape of the flexible tether connecting the robots, though not necessarily precisely. Several setups have showcased the ability to **handle flexible connections** in diverse scenarios among robots. For instance, a dual-arm robot manipulates rope-like objects in clutter [15], three quadcopters connected by a net catch and throw ball [11], multiple robots manipulate deformable bed sheet [16], a visual servoing scheme maintains the tether's shape between two-wheel robots [17, 18], and a pair of tethered quadcopters move an object using the tether [6, 7]. Although the above literature strongly suggests the significance of maintaining tether/net shapes in cooperative planning, directly and precisely controlling its shape is oftentimes unnecessary, resulting in increased complexity in both modeling and control. To tackle this problem, our proposed framework adopts a U-shape cost function to maintain a proper distance between the two tethered robots, such that the net shape is indirectly controlled.

As an essential branch of robust control, **model reference adaptive control** can maintain the platform stability with unmodeled dynamics. It has been implemented on various robot platforms to handle inaccurate physical parameters, unknown payloads, or external disturbance [19]. In particular, MRAC is implemented on a tilt-rotor quadcopter platform to pull an unmodeled cart [20], and WMR adopts MRAC to maintain the platform stability when the physical parameters are inaccurate [21]. Similarly, we implement a MRAC on the tethered robot duo for the object gathering task, since the physical parameters of the objects are unknown, and the drag force increases with more objects collected by the net.

## B. Overview

We organize the remainder of the paper as follows. Section II presents the proposed cooperative path planning framework for the tethered robot duo. Section III describes the implementation of MRAC on the robot duo for robust trajectory tracking. Section IV and Section V show the simulation and experiment results with comprehensive evaluations, respectively. Finally, we conclude the paper in Section VI.

## II. PATH PLANNING

This section describes the cooperative path planning framework for the tethered robot duo, assuming the environment and object locations are known.

### A. Problem Setup

Fig. 2 illustrates our problem setup. Given an environment with a set of objects (red squares) and obstacles (black blocks), a robot duo is tasked to navigate from the initial configuration  $P_s = (P_s^1, P_s^2)$  (pink dots) to reach the end configuration  $P_e = (P_e^1, P_e^2)$  while collecting the objects on-the-way and avoiding obstacles. The environment is described as a occupancy grid *Map*, wherein  $Map(i, j) = 0, 1$ , and

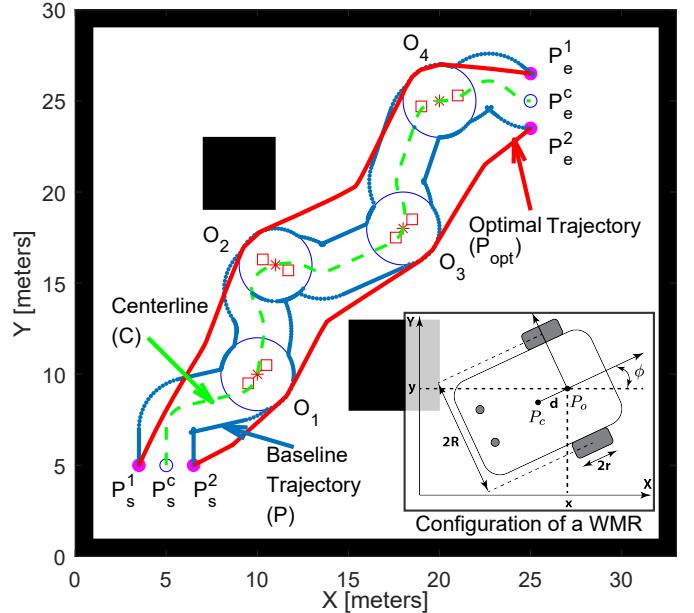


Fig. 2: **The proposed cooperative path planning framework for tethered robot duo.** Our framework generates a **centerline**  $C$  to connect start point  $P_s^c$ , end point  $P_e^c$ , and all object points, together with a set of waypoints (*i.e.*, **baseline trajectory**  $P$ ), who serves as the initialization for the subsequent **trajectory optimization** to find  $P_{opt}$ .

$2$  denote empty grids, grids occupied by obstacles, and grids occupied by objects, respectively. A safety margin  $\gamma$  defined based on the robot's dimension is further added to obstacles for collision avoidance and to the radius of each target object's minimum bounding circle for safety. We also merge the circles that are closely overlapped. As a result, we can have  $M$  circles in total  $O = \{O_1, \dots, O_M\}$ , where  $O_i = [x_i^o, y_i^o, r_i^o]$  includes the center position and radius of each circle. Of note, avoiding obstacles not only requires the robot duo to avoid directly colliding with the obstacle, but further demands the tethered net not to enclose any obstacle at any moment.

We devise a two-stage cooperative path planning framework to generate an optimal trajectory, shown also in Fig. 2. First, our framework generates a set of waypoints (blue dots) from the centerline path (dashed green line) that connects the robots' start and end configurations along the circles that enclose target objects. We call these waypoints **baseline trajectory**; they serve as a good initialization for the subsequent step. Next, our framework performs **trajectory optimization** to produce feasible trajectories (solid red line).

**Assumption:** Our framework has two assumptions: (i) The circles do not overlap with obstacles; namely, the objects cannot be too close to the obstacle such that the robot duo fails to navigate without violating the safety margin. (ii) The size of the circles is smaller than the length of the tethered net; the net constrains the robot duo's motions.

### B. Baseline Trajectory

Let the middle point of the tethered robot duo at the start configuration be  $P_s^c = \frac{1}{2}(P_s^1 + P_s^2)$  and at the end configuration be  $P_e^c = \frac{1}{2}(P_e^1 + P_e^2)$ . First, we adopt a conventional path planning algorithm (hybrid  $A^*$  [22] or  $RRT^*$  [23]) to construct the centerline by connecting  $P_s^c$ , each circle's center

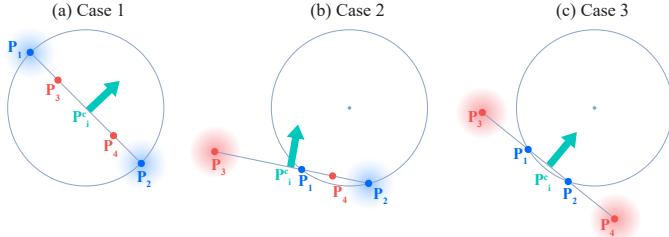


Fig. 3: **The selection of baseline points.** Given the centerline point  $P_i^c$  (green arrow), four candidates of baseline points are proposed; Eqs. (2) and (3) yield  $P_1, P_2$  (blue dots), and Eq. (1) yields  $P_3, P_4$  (red dots). Shaded points indicate the selected baseline points  $P_i^l, P_i^r$ .

point  $O_i$ , and  $P_e^c$ ; see dashed green line in Fig. 2. Formally, the centerline is denoted as  $C = \{P_s^c, \dots, P_e^c\} \in \mathbb{R}^{N \times 3}$ ,  $P_i^c = [x_i^c, y_i^c, \phi_i^c]$ , where  $N$  is the total number of points. Next, we expand this centerline to form robot duo's baseline trajectories,  $P = [P^l, P^r] \in \mathbb{R}^{N \times 6}$ ; see blue curves in Fig. 2.

Specifically, we divide  $C$  into two groups: on or near the circumference of circles  $C_{\text{circle}}$  and others  $C_{\text{free}}$ . For point  $P_i^c \in C_{\text{free}}$ , the baseline trajectory points are given by

$$P_i^l, P_i^r = \begin{bmatrix} x_i^c \pm l \cos(\phi_i^c + \frac{\pi}{2}) \\ y_i^c \pm l \sin(\phi_i^c + \frac{\pi}{2}) \\ \phi_i^c \end{bmatrix}, \quad (1)$$

where  $l$  is the separation distance chosen based on the dimension of the WMR.

For point  $P_i^c \in C_{\text{circle}}$ , we first calculate the line in the direction perpendicular to its orientation  $\phi_i^c$  with

$$y = kx + b, \quad (2)$$

where  $k = \tan(\phi_i^c + \frac{\pi}{2})$ , and  $b = y_i^c - kx_i^c$ . Next, we can solve two intersection points,  $P_1$  and  $P_2$ , by combining Eq. (2) with the related circle equation of  $O_j$ ,

$$(x - x_j^o)^2 + (y - y_j^o)^2 = r_j^{o2}. \quad (3)$$

Another two points obtained by Eq. (1) are denoted as  $P_3$  and  $P_4$ . We choose  $P_i^l, P_i^r$  from these four candidates by selecting the two outer ones, as shown in Fig. 3.

### C. Trajectory Optimization

With the estimated baseline trajectory in Section II-B as the initialization, we further devise an iterative optimization framework to find the optimal trajectory for the tethered duo. Compared with optimizing the whole trajectory at once, our iterative optimization framework helps improve computing efficiency and avoid local minimum [24, 25]. At each iteration, the trajectory of one mobile robot, denoted as  $P^1$ , is optimized, whereas the trajectory of the other robot, denoted as  $P^2$ , is treated as a known constant. Algorithm 1 summarizes this framework, where  $\text{opt}$  stands for the nonlinear optimization solver with defined cost functions and constraints.

#### 1) Cost Functions:

**Distance Cost:** The distance cost is calculated as the weighted sum of the Euclidean distance between two consec-

utive points and the orientation difference:

$$\begin{aligned} J_{\text{dist}}(i) &= t_1 \sqrt{\Delta x_i^2 + \Delta y_i^2} + t_2 |\Delta \phi_i|, \\ \Delta x_i &= P^1(i, 1) - P^1(i + 1, 1), \\ \Delta y_i &= P^1(i, 2) - P^1(i + 1, 2), \\ \Delta \phi_i &= P^1(i, 3) - P^1(i + 1, 3), \end{aligned} \quad (4)$$

where  $t_1$  and  $t_2$  are two non-negative parameters.

**Obstacle Cost:** We define obstacle cost in two forms: the cost of physical obstacles and the cost of objects as obstacles. The cost of physical obstacles is defined as

$$J_{\text{obs},1}(i) = \begin{cases} K_1, & \text{Map}(P^1(i, 1), P^1(i, 2)) \geq 1 \\ 0, & \text{Else} \end{cases}, \quad (5)$$

where  $K_1$  is a positive parameter to be selected.

The cost of objects as obstacles treats circles  $O_j$  as obstacles; the cost is designed similarly to the potential field method [26]:

$$J_{\text{obs},2}(i, j) = \begin{cases} \frac{1}{2} K_2 \left( \frac{1}{D(i, j)} - \frac{1}{r_j^o} \right)^2, & D(i, j) \leq r_j^o \\ 0, & D(i, j) > r_j^o \end{cases}, \quad (6)$$

where  $D(i, j)$  denotes the Euclidean distance between the trajectory point  $P_i^1$  and the object circle center  $O_j$ , and  $K_2$  is a positive parameter to penalize the distance.

As such, the combined obstacle cost of point  $P_i^1$  is defined as

$$J_{\text{obs}}(i) = J_{\text{obs},1}(i) + \sum_{j=1}^M J_{\text{obs},2}(i, j). \quad (7)$$

**Expansion Cost:** The cost function for expansion is designed as a U-shape cost function [27]:

$$J_e(i) = \begin{cases} k_{d1} \tan^2(\gamma_1 d + \gamma_2), & d_{\min} \leq d \leq d_{\text{rest}} \\ k_{d2}(d - d_{\text{rest}})^2 - \frac{(d - d_{\text{rest}})^2}{(d - d_{\max})^2}, & d_{\text{rest}} < d \leq d_{\max} \\ K_3, & d < d_{\min} \text{ or } d > d_{\max} \end{cases} \quad (8)$$

where  $\gamma_1 = \frac{\pi}{2(d_{\text{rest}} - d_{\min})}$ , and  $\gamma_2 = -\gamma_1 d_{\text{rest}}$ .  $d$  is the Euclidean distance between trajectory points  $P_i^1$  and  $P_i^2$ , representing the expansion of the tethered net.  $k_{d1}$ ,  $k_{d2}$ , and  $K_3$  are gain parameters to tune the shape of the above cost function.  $d_{\max}$ ,  $d_{\min}$ , and  $d_{\text{rest}}$  denote the minimum, maximum, and the ideal expansion of the tethered net, selected according to various hardware setup and task scenarios.

The reason for having this expansion cost is to maintain the U-shape of the tethered net. Conversely, the shape of the net would make it infeasible to collect objects when the two WMRs are too close or too far. In this paper, we set  $d_{\min} = 0.1d_{\max}$  and  $d_{\text{rest}} = \frac{2}{\pi}d_{\max}$ —a half-circle shape. Of note, if  $d_{\max}$  is not specified, we can treat it as a random variable to estimate the optimal net length in certain task scenarios.

**Smoothness Cost:** The smoothness cost is defined as the sum of squared accelerations along the trajectory:

$$J_s = \sum_{j=1}^3 b_j P^1(:, j)^T Q P^1(:, j), \quad (9)$$

**Algorithm 1:** Trajectory Optimization Algorithm

---

**Data:**  $N_{\text{total}}, P, O, \text{Map}, t_{1-2}, K_{1-3}, k_{d1}, k_{d2}$   
 $Q, b_{1-3}, a_{1-4}, \Delta L_{\max}, \Delta \phi_{\max}$

**Result:**  $P_{\text{opt}}, d_{\max}$

```

step ← 1,  $d_{\max} \leftarrow N \Delta L_{\max}$ ;           // Initialization
while step ≤  $N_{\text{total}}$  do
    if mod(step, 2) = 1 then // Optimize left half
         $P_s \leftarrow P(1, 1 : 3)$ ,  $P_e \leftarrow P(N, 1 : 3)$ ;
        // Specify start and end points
         $P_{\text{ini}}^1 \leftarrow P(2 : N-1, 1 : 3)$ ;
        // Traj to be optimized
         $P_{\text{ini}}^2 \leftarrow P(2 : N-1, 4 : 6)$ ;
        // Traj of another half as constant
         $[P_{\text{opt}}^1, d_{\max}] \leftarrow \text{opt}(P_{\text{ini}}^1, P_{\text{ini}}^2, P_s, P_e, \text{Map}, O, d_{\max})$ ;
        // Optimization process
         $P(2 : N-1, 1 : 3) \leftarrow P_{\text{opt}}^1$ ;
        // Update left half trajectory
    else // Optimize right half
         $P_s \leftarrow P(1, 4 : 6)$ ,  $P_e \leftarrow P(N, 4 : 6)$ ;
         $P_{\text{ini}}^1 \leftarrow P(2 : N-1, 4 : 6)$ ;
         $P_{\text{ini}}^2 \leftarrow P(2 : N-1, 1 : 3)$ ;
         $[P_{\text{opt}}^2, d_{\max}] \leftarrow \text{opt}(P_{\text{ini}}^1, P_{\text{ini}}^2, P_s, P_e, \text{Map}, O, d_{\max})$ ;
         $P(2 : N-1, 4 : 6) \leftarrow P_{\text{opt}}^2$ ;
    end
    step ← step + 1;           // Update steps
end
 $P_{\text{opt}} \leftarrow P$ ;      // Output optimized trajectory

```

---

where  $Q = A^T A$ , and  $A$  is the finite differencing matrix [28],  $b_j$ s are weighting gains.

Taking together, the total cost along the whole trajectory is defined by the aforementioned costs:

$$J_{\text{total}} = a_1 J_s + \sum_{i=1}^N a_2 J_{\text{dist}}(i) + a_3 J_{\text{obs}}(i) + a_4 J_e(i) \quad (10)$$

where  $a_i$ s are weighting gains for costs.

## 2) Constraints:

**Maximum Velocity Constraint:**

$$\begin{aligned} 0 &\leq \sqrt{\Delta x_i^2 + \Delta y_i^2} \leq \Delta L_{\max}, \\ -\Delta \phi_{\max} &\leq \Delta \phi_i \leq \Delta \phi_{\max}, \end{aligned} \quad (11)$$

where  $\Delta L_{\max}$  is the maximum travel distance at each step, and  $\Delta \phi_{\max}$  is the maximum turn angle at each step.

**Object and Obstacle Constraint:** We select consecutive points on trajectory  $P^1$  and  $P^2$  (*i.e.*,  $P_i^1, P_{i+1}^1, P_i^2, P_{i+1}^2$ ) to build a quadrilateral; hence, we can have  $N-1$  quadrilaterals. To ensure collecting all the objects, the centers of all circles  $O$  must be covered by these quadrilaterals. Moreover, to prevent the path from enclosing any obstacles inside, these quadrilaterals must not overlap with any obstacles.

**III. ROBUST TRAJECTORY TRACKING CONTROL**

We adopt a decentralized framework to design the trajectory tracking controller for individual WMR. In our proposed setup, the drag force from the tethered net and gathered objects are unknown. Hence, the conventional model-based controller is not robust enough, and a more advanced MRAC is implemented to improve the robustness of the trajectory tracking control.

**A. Dynamics of Individual WMR**

The configuration of the WMR is defined by  $q = [x, y, \phi]$ , shown in the lower right corner of Fig. 2. Its dynamics can be described by [29]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} = B(q)\tau - A(q)^T \lambda, \text{ where} \quad (12)$$

$$M(q) = \begin{bmatrix} m & 0 & md \sin \phi \\ 0 & m & -md \cos \phi \\ md \sin \phi & -md \cos \phi & J \end{bmatrix}, \quad (13)$$

$$C(q, \dot{q}) = \begin{bmatrix} 0 & 0 & md\dot{\phi} \cos \phi \\ 0 & 0 & -md\dot{\phi} \sin \phi \\ 0 & 0 & 0 \end{bmatrix}, B(q) = \frac{1}{r} \begin{bmatrix} \cos \phi & \cos \phi \\ \sin \phi & \sin \phi \\ R & -R \end{bmatrix},$$

$$A(q) = [-\sin \phi \ \cos \phi \ -d], \lambda = -m(\dot{x} \cos \phi + \dot{y} \sin \phi)\dot{\phi},$$

where  $m$  is the WMR's mass,  $J$  its rotational inertia,  $2R$  the distance between two driving wheels,  $r$  the radius of each wheel,  $d$  the distance from the coordinate origin  $P_0$  to the CoM  $P_c$ ,  $M(q)$  the inertia matrix,  $C(q, \dot{q})$  the Coriolis and Centrifugal force matrix,  $B(q)$  the input transformation matrix,  $\tau$  the torque vector for two wheels,  $A(q)$  matrix associated with the non-holonomic constraints (*i.e.*,  $A(q)\dot{q} = 0$ ), and  $\lambda$  the vector of constraint forces.

Selecting  $S(q)$  as a basis of  $A(q)$  nullspace (*i.e.*,  $S(q)^T A(q)^T = 0$ ),

$$S(q) = \begin{bmatrix} \cos \phi & -d \sin \phi \\ \sin \phi & d \cos \phi \\ 0 & 1 \end{bmatrix}, \quad (14)$$

we can rewrite the WMR's kinematic equation with the velocity vector,

$$\dot{q} = S(q)v. \quad (15)$$

$v = [\nu, \omega]^T$ , and  $\nu$  and  $\omega$  are the WMRs' linear and angular velocity. Taking the derivative of Eq. (15), we have

$$\ddot{q} = \dot{S}(q)v + S(q)\dot{v}. \quad (16)$$

Substituting Eqs. (15) and (16) into Eq. (12) and multiplying  $S^T$  at each side, we have

$$\bar{M}\dot{v} + \bar{C}v = \bar{\tau}, \quad (17)$$

where  $\bar{M} = S(q)^T M(q) S(q)$ ,  $\bar{\tau} = \bar{B}\tau$ ,  $\bar{B} = S(q)^T B(q)$ , and  $\bar{C} = S(q)M(q)\dot{S}(q) + S(q)^T C(q, \dot{q})S(q)$ .

**B. Model-Based Control of Individual WMR**

We design a basic model-based control based on Eq. (17),

$$\tau = \bar{B}^\dagger(\bar{M}\dot{v}^d + \bar{C}v^d), \quad (18)$$

where  $v^d$  is the desired velocity vector, designed to track the reference trajectory [29]:

$$v^d = \begin{bmatrix} \nu^d \\ \omega^d \end{bmatrix} = \begin{bmatrix} \nu^r \text{cose}_\phi + k_1(e_x + d(1 - \text{cose}_\phi)) \\ \omega^r + k_2\nu^r(e_y - dsine_\phi + k_3\nu^r \text{sine}_\phi) \end{bmatrix}, \quad (19)$$

where  $\nu^r$  and  $\omega^r$  are reference linear and angular velocity,  $k_1$ ,  $k_2$ , and  $k_3$  are positive parameters, and  $e_x$ ,  $e_y$ , and  $e_\phi$  are errors between the reference trajectory and the real trajectory in  $x$ ,  $y$ , and  $\phi$ , respectively [29],

$$e_x = x^r - x, e_y = y^r - y, e_\phi = \phi^r - \phi. \quad (20)$$

Of note, although this controller has proven to work well on WMRs given accurate physical parameters, it fails in our setting as the tethered net will collect multiple objects with unknown physic parameters (mass, inertia, *etc.*). As such, we further design an adaptive control law; see the next section.

### C. Adaptive Control with Unknown Payload

Since MRAC laws guarantee the asymptotic convergence of the trajectory tracking error in the presence of parametric and matched uncertainties [19], we implement an MRAC on WMRs to improve its robustness under unknown payload.

Specifically, we rewrite the matrices in Eq. (17) with unknown dynamics,

$$\begin{aligned}\bar{M} &= \bar{M}_k + \bar{M}_u^* = (\bar{M}_k - I) + (I + \bar{M}_u^*) \\ &= (\bar{M}_k - I) + \bar{M}_u, \\ \bar{C} &= \bar{C}_k + \bar{C}_u,\end{aligned}\quad (21)$$

where  $k$  and  $u$  denote the known and unknown dynamics, respectively. Combining Eq. (21) with Eq. (17), we have

$$\dot{X}(t) = A_r X(t) + B_r u(t), \quad (22)$$

where  $X = v$ ,  $A_r = -\bar{M}_u^{-1} \bar{C}_u$ , and  $B_r = \bar{M}_u^{-1}$ . The input  $\bar{\tau}$  can be recovered by

$$\bar{\tau} = u + (\bar{M}_k - I)\dot{v}(t) + \bar{C}_k v(t), \quad (23)$$

and the reference model is given as

$$\dot{X}_m(t) = A_m X_m(t) + B_m u_m(t), \quad (24)$$

where  $X_m = v^d$ ,  $A_m = -K$ ,  $K$  is a Hurwitz matrix, and  $B_m = I$ . The reference input  $u_m$  can be calculated by

$$u_m = B_m^{-1}(\dot{X}_m - A_m X_m) = \dot{v}^d + Kv^d. \quad (25)$$

And the adaptive law is designed as

$$\begin{aligned}u &= \bar{u}_a + \bar{u}_k, \\ \bar{u}_k &= -KX + u_m, \\ \bar{u}_a &= -\Delta_X X + \Delta_m u_m,\end{aligned}\quad (26)$$

where  $\bar{u}_k$  and  $\bar{u}_a$  are the linear and adaptive feedback component, respectively.  $\Delta_X$  and  $\Delta_m$  can be calculated with adaptive gains and state errors; please refer to Canigur *et al.* [21] for more implementation details. Of note, the motor torque saturation and the adaptive gains of MRAC jointly determine the maximum payload can be collected while the stability of tracking controller is still maintained. Larger adaptive gains can increase the response speed but may lead to oscillations and higher overshoot [30].

## IV. SIMULATION

In this section, we evaluate our cooperative planning framework in simulation based on Gazebo, a 3D dynamic environment simulator. Our results demonstrate that (i) the proposed framework produces optimal trajectories for the tethered robots, and (ii) our U-shape cost function design effectively maintains the tethered net during the tasks.

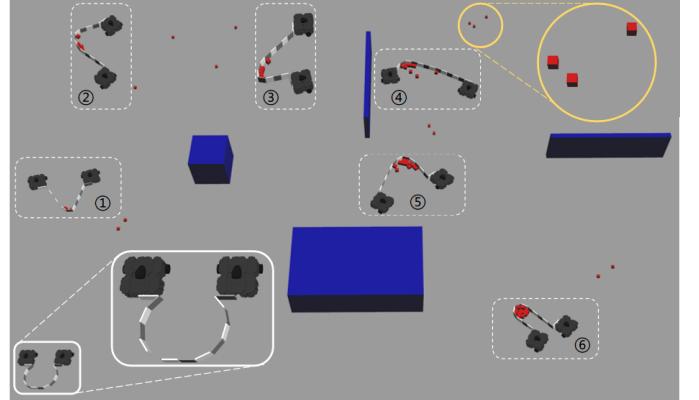


Fig. 4: **Simulation Environment.** The tethered robot duo is tasked to gather 13 objects (small red cubes), navigating from the bottom left to bottom right (white solid boxes). The dashed boxes depict some key moments when the robots gather objects with the flexible net.

### A. Simulation Setup

For the tethered robot platform, we utilize two TurtleBot3 Waffle Pi robots and tether them with a flexible “net” by connecting several 0.1-meter-long thin cuboid links with passive revolute joints; the length of the net can be easily modified by inserting or removing cuboid links. With such a design, the tethered net would deform in accord to gathered objects’ weights, and drag forces could be introduced to robots in an unspecified direction, crucial in evaluating our MRAC implementation. To enable torque command to the robots in simulation, we use *JointEffortController* in *ros\_control* to command desired torques to TurtleBot3’s wheels instead of using the built-in controller with velocity control only.

Fig. 4 illustrates the simulated environment with several keyframes of a task performance. The tethered robot duo starts from the bottom left to gather red movable cubes while avoiding blue obstacles at various locations. Table I tabulates some essential physical and software properties of the simulation setup. We set the parameters involved in the trajectory planning as following:  $l=0.3$  m,  $N_{\text{total}}=8$ ,  $t_1=300$ ,  $t_2=10$ ,  $K_1=K_2=K_3=1e7$ ,  $k_{d1}=20$ ,  $k_{d2}=15$ ,  $b_1=b_2=110$ ,  $b_3=1$ ,  $a_1=a_2=a_3=a_4=1$ ,  $\Delta L_{\max}=0.05$  m, and  $\Delta \phi_{\max}=0.1$  rad. We utilize the nonlinear optimization solver *fmincon* in Matlab to solve the optimal trajectory with the interior point method.

TABLE I: Physical and Software Properties used in Simulation

Group	Parameter	Value
Robot	Mass $m$	1.43 kg
	Moment $J$	0.146 kg · m <sup>2</sup>
	$R$	0.144 m
	$r$	0.033 m
	$d$	0.020 m
Net	Mass of each cuboid link	2 g
	Size of each cuboid link $d \times w \times h$	1 × 10 × 8 cm <sup>3</sup>
	Damping of each cuboid joint	1e - 3
	Friction of each cuboid joint	$\mu_1 = \mu_2 = 0.2$
Others	Mass of object cube $m_i$	25 g
	Size of object cube	5 × 5 × 5 cm <sup>3</sup>
	Trajectory tracking controller rate	100 Hz
	Motor torque controller rate	500 Hz

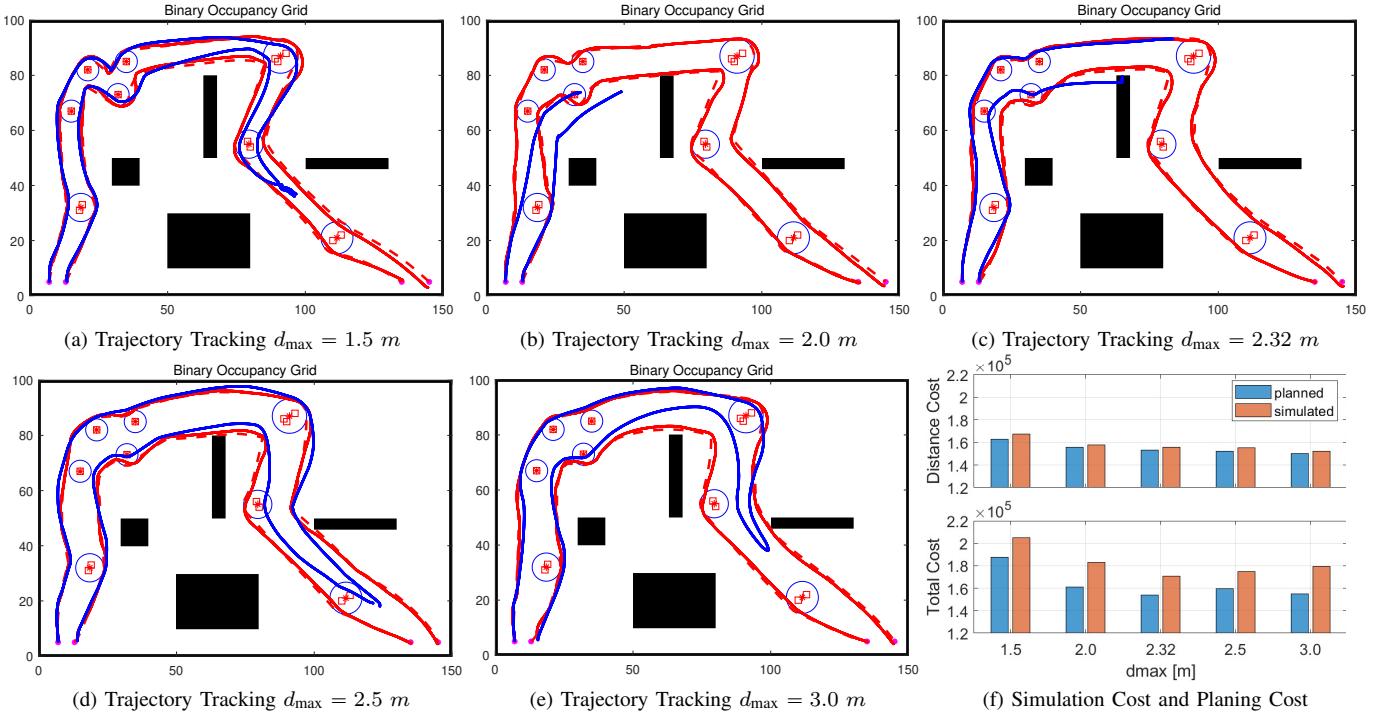


Fig. 5: The schematic of the simulated environment constructed in Gazebo. (a–e) With various lengths of the tethered net, our cooperative planning framework produces reference trajectory (dashed red lines). Tracked trajectories by the proposed MRAC controller and conventional model-based controller are shown in solid red and blue lines, respectively; the MRAC controller can robustly track trajectories, whereas a conventional controller constantly fails as the task progresses. (f) Distance cost and total cost of the optimized trajectories in the planning and the simulated execution phases. Our framework can also produce the optimal  $d_{\max} = 2.32 \text{ m}$  for a given configuration.

### B. Trajectory Tracking

Fig. 5 shows the simulation results with various lengths of the tethered net  $d_{\max}$ . Given a configuration during the optimization, our framework can further estimate the optimal length of the tethered net in terms of defined cost function  $J_{\text{total}}$  by treating  $d_{\max}$  as a variable in Algorithm 1. If  $d_{\max}$  is already specified, Algorithm 1 can be easily modified by treating  $d_{\max}$  as a constant instead of a variable. For instance, among five cases with various maximum net lengths shown in Fig. 5, our framework estimates  $d_{\max} = 2.32 \text{ m}$  is the optimal net length. In these simulations, we implement both MRAC and a conventional model-based controller. Specifically, the MRAC can maintain the platform stability along the whole trajectory while successfully gathering and transporting all the objects to the endpoint. In contrast, the conventional model-based controller is unstable with more objects collected, resulting in collisions between the two robots or between the robot and objects/obstacles.

For each of the above cases, the distance cost  $J_{\text{dist}}$  and total cost  $J_{\text{total}}$  are first obtained with recorded robot trajectory. Next, these costs are compared with planned costs; the results are shown in Fig. 5f. In particular, if we only care about  $J_{\text{dist}}$ , both  $d_{\max} = 2.5 \text{ m}$  and  $d_{\max} = 3.0 \text{ m}$  are better. However, it is difficult to maintain the shape of the tethered net under these two settings. As such, our optimization framework deems they are not the optimal solution. In comparison, although  $d_{\max} = 2.32 \text{ m}$  results in a slightly higher  $J_{\text{dist}}$ , this length is considered as the optimal one due to the lowest  $J_{\text{total}}$ .

This simulation result demonstrates the tracking performance of the proposed MRAC controller to handle increasing

payloads. It also suggests that a longer tether between robots does not necessarily correspond to better efficiency. This finding may impact prior arts wherein the length is seldom considered [4, 14] or fixed [5, 6, 17, 18, 31].

### C. Maintaining Net's Shape

We design a U-shape cost function in Eq. (8) to penalize the tethered robot duo for being too close/far, so that the curvature of the net is indirectly controlled to embrace new objects and carry gathered objects. To demonstrate the efficacy of the U-shape design, we compare with two baseline cost functions:

- 1) **No separation constraint included.** This strategy does not take separation distance into consideration [4]; referred to as Baseline 1 in Fig. 6.
- 2) **Desired separation distance  $d_{\text{rest}}$  as hard constraint.** This strategy maintains the desired separation distance along the path [6, 17, 18, 31]; referred to as Baseline 2 in Fig. 6.

Figs. 6b, 6d and 6f quantitatively compare the distance cost of each cost design with various net lengths in three scenarios, and Figs. 6a, 6c and 6e show the corresponding reference trajectories produced by our framework using the 3 types of expansion cost designs with a specific net length in each scenario. Although Baseline 1 yields the shortest travel distance without including any constraint on separation distance, it may require unrealistic net length. Baseline 2 keeps a specified separation distance at all time, which may find no solution for a small  $d_{\max}$  or cause unnecessarily long travel distance for a large  $d_{\max}$ . In comparison, the proposed U-shape cost function produces feasible solutions for various  $d_{\max}$  while properly maintaining efficient travel distances.

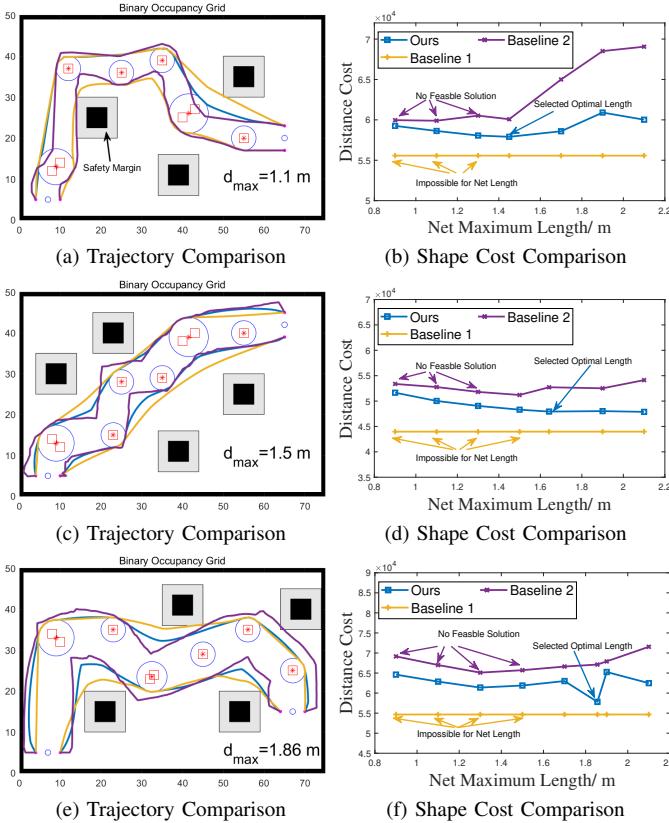


Fig. 6: Planned trajectories with various expansion cost function designs in three scenarios; each row corresponds to one scenario.

## V. EXPERIMENT

Section IV has evaluated the proposed cooperative planning framework and the MRAC controller design in simulation. This section further validates our proposal with physical implementation and experiment in the physical environment.

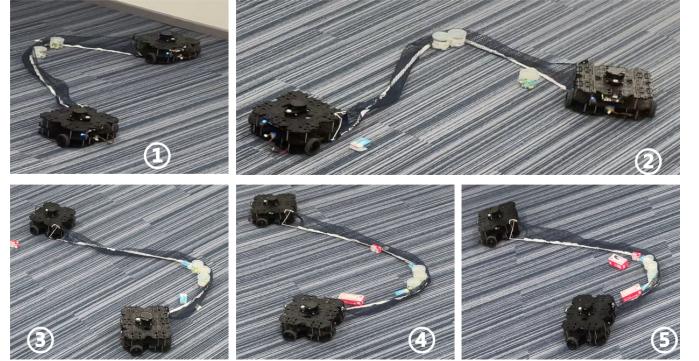
### A. Experiment Setup

The setup of our proposed tethered robot duo is shown in Fig. 1b, wherein a net is connected to two TurtleBot3 Waffle Pi robots at each end. Each TurtleBot is equipped with a Raspberry Pi 3B+ running Robot Operating System (ROS) as the main PC, an OpenCR 1.0 board with an IMU module and a microprocessor for low-level motor control, an LDS-01 Lidar for localization, and two Dynamixel XM430 motors with a maximum torque of  $3 \text{ N} \cdot \text{m}$  for wheel actuation. The path planning component runs on a desktop (AMD Ryzen9 5950X CPU, 64.00 GB RAM) and takes an average of 180.3 s to produce the optimal trajectory with  $N = 200$  in Algorithm 1.

To enable torque command, we modify the built-in low-level controller on OpenCR board that only takes velocity command inputs such that it receives torque command inputs. The Gmapping Simultaneous Localization and Mapping (SLAM) algorithm is utilized to localize the WMR in experiment and outputs  $q$  and  $v$  as feedback. Fig. 7a shows the experimental environment, a replication of the simulated environment studied in Fig. 6a; eight objects are placed on the floor with various weights ranging from 15 g to 50 g.



(a) **Experimental environment**, wherein eight objects are placed on the floor. The robot duo is tasked to gather them with a tethered net while avoiding obstacles during its navigation.



(b) Five keyframes of gathering objects captured with varying weights during the robot duo's task execution.

Fig. 7: Experiments with MRAC implementation in a robot duo in the physical environment.

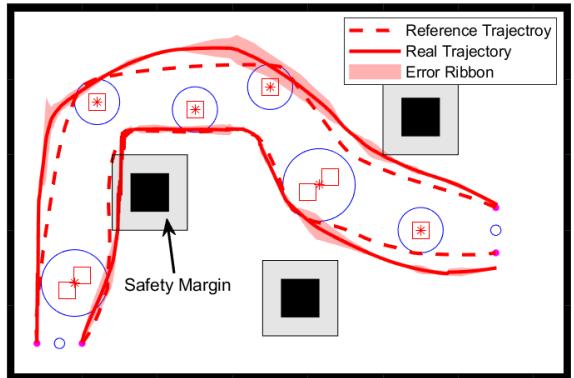


Fig. 8: **Reference and tracked trajectories of the tethered robot duo.** The red shadows indicate the variations of five executions.

### B. Experiment Results

Our experiment's net length is 1.45 m, the optimal length estimated by the path planning according to this environment configuration. Fig. 8 plots the reference and actual trajectories with error ribbon acquired from five trials. Together with the simulation results, our experiments have demonstrated (i) the implemented MRAC controller robustly tracks the planned trajectories, and (ii) the tethered robot duo, using the proposed cooperative planning framework, successfully gathers all the objects along the path and carries them to the end position.

## VI. CONCLUSION

This paper investigated an interesting task of gathering and transporting scattered objects with a tethered robot duo. We addressed it by proposing a two-stage cooperative planning framework based on trajectory optimization and implementing it with MRAC. In planning, we designed a U-shape cost function and incorporated other constraints to produce trajectories, capable of indirectly maintaining the flexible net's shape w.r.t. the distance between two robots. In the implementation, we demonstrated that the MRAC could robustly handle the increased payload with unknown dynamics as more objects were carried. As an extra feature, our planning framework can also estimate the most efficient length given an environment configuration, which led to a crucial extension to existing work in tethered robots that assumed an infinite or fixed tether length. Both the simulation and experiment results have demonstrated the efficacy of the proposed framework and the necessity of MRAC implementation. In the future work, we plan to (i) extend this framework to tethered Unmanned Surface Vehicle (USV) and drones for collaborative tasks, (ii) incorporate a perception module when the environment configuration is not fully available, and (iii) enable effective re-planning for the robot duo to account for new observations.

## VII. ACKNOWLEDGEMENT

The authors would like to thank Bo Dai, Ziyuan Jiao, Aoyang Qin and Zhen Chen at BIGAI for their constructive discussions and help on experiments.

## REFERENCES

- [1] M. M. Gulzar, S. T. H. Rizvi, M. Y. Javed, U. Munir, and H. Asif, "Multi-agent cooperative control consensus: A comparative review," *Electronics*, vol. 7, no. 2, p. 22, 2018.
- [2] A. Yamashita, T. Arai, J. Ota, and H. Asama, "Motion planning of multiple mobile robots for cooperative manipulation and transportation," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 2, pp. 223–237, 2003.
- [3] S. Heshmati-Alamdar, G. C. Karras, and K. J. Kyriakopoulos, "A predictive control approach for cooperative transportation by multiple underwater vehicle manipulator systems," *IEEE Transactions on Control Systems Technology*, 2021.
- [4] S. Bhattacharya, S. Kim, H. Heidarsson, G. S. Sukhatme, and V. Kumar, "A topological approach to using cables to separate and manipulate sets of objects," *International Journal of Robotics Research (IJRR)*, vol. 34, no. 6, pp. 799–815, 2015.
- [5] R. H. Teshnizi and D. A. Shell, "Motion planning for a pair of tethered robots," *Autonomous Robots*, pp. 1–15, 2021.
- [6] D. S. D'antonio, G. A. Cardona, and D. Saldaña, "The catenary robot: Design and control of a cable propelled by two quadrotors," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 2, pp. 3857–3863, 2021.
- [7] G. A. Cardona, D. S. D'antonio, C.-I. Vasile, and D. Saldana, "Non-prehensile manipulation of cuboid objects using a catenary robot," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2021.
- [8] T. Furukawa, F. Bourgault, B. Lavis, and H. F. Durrant-Whyte, "Recursive bayesian search-and-tracking using coordinated uavs for lost targets," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2006.
- [9] I. I. Hussein and D. M. Stipanovic, "Effective coverage control for mobile sensor networks with guaranteed collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 4, pp. 642–657, 2007.
- [10] L. Yan, T. Stouraitis, and S. Vijayakumar, "Decentralized ability-aware adaptive control for multi-robot collaborative manipulation," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 2, pp. 2311–2318, 2021.
- [11] R. Ritz, M. W. Müller, M. Hehn, and R. D'Andrea, "Cooperative quadcopter ball throwing and catching," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2012.
- [12] X. Zhang, F. Zhang, P. Huang, J. Gao, H. Yu, C. Pei, and Y. Zhang, "Self-triggered based coordinate control with low communication for tethered multi-uav collaborative transportation," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 2, pp. 1559–1566, 2021.
- [13] J. Fink, N. Michael, S. Kim, and V. Kumar, "Planning and control for cooperative manipulation and transportation with aerial robots," *International Journal of Robotics Research (IJRR)*, vol. 30, no. 3, pp. 324–334, 2011.
- [14] S. Kim, S. Bhattacharya, H. K. Heidarsson, G. S. Sukhatme, and V. Kumar, "A topological approach to using cables to separate and manipulate sets of objects," in *Proceedings of Robotics: Science and Systems (RSS)*, 2013.
- [15] P. Mitrano, D. McConachie, and D. Berenson, "Learning where to trust unreliable models in an unstructured world for deformable object manipulation," *Science Robotics*, vol. 6, no. 54, 2021.
- [16] J. Alonso-Mora, R. Knepper, R. Siegwart, and D. Rus, "Local motion planning for collaborative multi-robot manipulation of deformable objects," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2015.
- [17] M. Laranjeira, C. Dune, and V. Hugel, "Catenary-based visual servoing for tethered robots," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2017.
- [18] M. Laranjeira, C. Dune, and V. Hugel, "Catenary-based visual servoing for tether shape control between underwater vehicles," *Ocean Engineering*, vol. 200, p. 107018, 2020.
- [19] E. Lavretsky and K. A. Wise, "Robust adaptive control," in *Robust and adaptive control*, pp. 317–353, Springer, 2013.
- [20] R. B. Anderson, J. A. Marshall, and A. L'Afflitto, "Constrained robust model reference adaptive control of a tilt-rotor quadcopter pulling an unmodeled cart," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 1, pp. 39–54, 2020.
- [21] E. Canigur and M. Ozkan, "Model reference adaptive control of a nonholonomic wheeled mobile robot for trajectory tracking," in *International Symposium on Innovations in Intelligent Systems and Applications*, 2012.
- [22] J. Petereit, T. Emter, C. W. Frey, T. Kopfstedt, and A. Beutel, "Application of hybrid a\* to an autonomous mobile robot for path planning in unstructured outdoor environments," in *German Conference on Robotics*, 2012.
- [23] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed rrt\*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2014.
- [24] C. L. Byrne, "Lecture notes on iterative optimization algorithms," *Department of Mathematical Sciences, University of Massachusetts Lowell*, 2014.
- [25] C. T. Kelley, *Iterative methods for optimization*. SIAM, 1999.
- [26] M. C. Lee and M. G. Park, "Artificial potential field based path planning for mobile robots using a virtual obstacle concept," in *International Conference on Advanced Intelligent Mechatronics*, 2003.
- [27] M. Ryll, H. H. Bülthoff, and P. R. Giordano, "A novel overactuated quadrotor unmanned aerial vehicle: Modeling, control, and experimental validation," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 540–556, 2014.
- [28] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "Stomp: Stochastic trajectory optimization for motion planning," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2011.
- [29] L. Li, T. Wang, Y. Xia, and N. Zhou, "Trajectory tracking control for wheeled mobile robots based on nonlinear disturbance observer with extended kalman filter," *Journal of the Franklin Institute*, vol. 357, no. 13, pp. 8491–8507, 2020.
- [30] J. E. Stellet, "Influence of adaptation gain and reference model parameters on system performance for model reference adaptive control," *World Academy of science, engineering and technology*, vol. 60, pp. 1768–1773, 2011.
- [31] K. A. Talke, M. De Oliveira, and T. Bewley, "Catenary tether shape analysis for a uav-usv team," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2018.