Dynamic Path Planning and Motion Control of Microrobotic Swarms for Mobile Target Tracking

Qian Zou, *Student Member, IEEE*, Xingzhou Du, *Member, IEEE*, Yuezhen Liu, *Student Member, IEEE*, Hui Chen, *Graduate Student Member, IEEE*, Yibin Wang, *Student Member, IEEE*, and Jiangfan Yu, *Member, IEEE*

***Abstract*— Magnetic field-driven microrobotic swarms have drawn extensive attention, especially in the field of automatic control. Realizing dynamic path planning and motion control of microrobotic swarms for mobile target tracking is one of the important tasks that still remains unsolved. In this paper, we firstly present an enhanced bidirectional rapidly-exploring random tree star (EB-RRT*) algorithm considering the physical size of the swarm to dynamically plan the optimal path for obstacle avoidance. An image-guided motion controller, which consists of a direction controller and a Genetic Algorithm based Linear Quadratic Regulator (GA-LQR) velocity controller, is then proposed to realize mobile target tracking using microrobotic swarms. Targeted bursting algorithm is subsequently developed to meet the requirement of tracking high-speed (*i.e.*, 20 $\mu m/s$) mobile targets. Simulations are performed to validate the proposed methods and obtain the proper ranges of the input parameters for the controllers. Finally, the control effectiveness of mobile target tracking in different conditions and environments is validated by experimental results.**

***Note to Practitioners*—The motivation of this work is to develop an effective control scheme for mobile target tracking using microrobotic swarms. Conventional control schemes mainly focus on the control of single microrobots to reach static targets, and thus the desired path is fixed once planned. In addition, the motion of single monolithic microrobots can be modelled precisely. However, in mobile target tracking using microrobotic swarms, dynamic planning algorithms are demanded to frequently update the desired path. Swarms consisting of millions of micro-agents are also difficult to be modelled due to the complex agent-agent interactions. In this work, an effective control scheme consisting of a dynamic path planner, a motion controller and a targeted bursting unit is developed. Real-time dynamic paths will be planned even though the positions of the swarm and the target change rapidly. The precise control of the swarm direction and velocity are achieved, and moreover, using the targeted bursting algorithm, the swarm can be accelerated to approach mobile targets accurately with higher efficiency. Experimental results validates the proposed tracking strategy in different environments with virtual obstacles. The proposed control scheme paves the way for a better understanding of advanced motion control methods for microrobotic swarms.**

***Index Terms*— Microrobotic swarms, path planning, mobile target tracking, swarm control.**

Manuscript received 20 April 2022; revised 30 June 2022; accepted 13 September 2022. Date of publication 21 September 2022; date of current version 6 October 2023. This article was recommended for publication by Associate Editor X. Liu and Editor J. Yi upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Project 62103347; in part by the Start-Up Funding of The Chinese University of Hong Kong, Shenzhen, under Project UDF01001929; in part by the Shenzhen Institute of Artificial Intelligence and Robotics for Society under Project AC01202101109; and in part by the Shenzhen Science and Technology Program under Grant RCBS20210609103155061. *(Corresponding author: Jiangfan Yu.)*

Qian Zou and Yuezhen Liu are with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen 518172, China.

Xingzhou Du is with the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), Shenzhen 518129, China.

Hui Chen and Yibin Wang are with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen 518172, China, and also with the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), Shenzhen 518129, China.

Jiangfan Yu is with the School of Science and Engineering, School of Medicine, The Chinese University of Hong Kong, Shenzhen 518172, China, and also with the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), Shenzhen 518129, China (e-mail: yujiangfan@cuhk.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TASE.2022.3207289.

Digital Object Identifier 10.1109/TASE.2022.3207289

I. Introduction

MILLI/MICROROBOTS remotely navigated by magnetic fields have attracted extensive attention due to their potential biomedical applications [1], [2], [3], [4], [5], [6], [7], [8], [9]. Different types of milli/microrobots have been investigated, *e.g.*, spherical [10], [11], [12], helical [13], [14], [15], and bio-hybrid milli/microrobots [16], [17], [18]. Meanwhile, milli/microrobots with different locomotion are also been reported, *e.g.*, millirobots with a composited agglutinate magnetic spray are capable of crawling, walking and rolling [19], soft microrobots consisting of photoactive liquid-crystal elastomers can perform translation and rotation [20], and trimer-like microrobots is able to rolling and chiral rotating [21]. Although various kinds of milli/microrobots have been developed, microrobotic swarms are considered as potential candidates to tackle challenges encountered in low-invasive therapies, such as targeted drug delivery and in-situ sensing [22], [23]. Inspired by the living swarm behaviors in nature, various kinds of mirorobotic swarms have been reported, *e.g.*, vortex-like swarms [24], ribbon-like swarms [25], elliptical swarms [26] and tornado-like swarms [27]. Since microrobotic swarms can hardly be equipped with onboard sensors and circuits, closed-loop control of them is significant for realizing navigated locomotion and pattern adaptive reconfiguration, especially in confined environments [26]. Moreover, tracking a mobile target using microrobotic swarms could be attached with further significances. In this case, dynamic path planning and motion control of the swarms are two important steps to realize the purpose.

Path planning is the first step for mobile target tracking. Previously, different path planning methods for microrobots to track static targets have been investigated, *e.g.*, a gradient path planner in a cluttered environment with triangular obstacles [28], an informed rapidly-exploring random tree star (Informed RRT*) path generator in an environment with rectangular obstructions [29], and an obstacle-weighted rapidly-exploring random tree path planning algorithm in a simulated vascular network [30]. The approaches share the similarity that the trajectory is fixed after planning because the targeted position is static. The position of the mobile target changes at each moment, and thus, it is challenging for the path planner to respond to the changes in time. To meet the requirement of tracking mobile targets, dynamic path planning methods are demanded.

Motion control of swarms to precisely tune their moving direction and velocity is another critical issue. Model-based and model-free control are two major categories. Various model-based control approaches are applied to navigate microrobots with higher accuracy. For instance, a novel model predictive controller of the magnetic spore-based microrobot is designed for targeted delivery [31], and an MRI-based control method of a magnetic microcapsule is developed for endovascular navigation [32]. Model-free control could play an important role if the models of microrobots are difficult to obtain. A model-free trajectory tracking control method without considering complicated dynamics models is reported to navigate a two-particle magnetic microrobot, which can guarantee the high accuracy of tracking [33]. To date, to develop effective and efficient control strategies for microrobotic swarms is still yet fully investigated [34], [35].

Challenges exist to realize mobile target tracking using microrobotic swarms, in terms of both dynamic path planning and motion control. Effective dynamic path planning methods need to be further investigated. The simultaneous movement of a microrobotic swarm and a target makes the path planning complex, and thus, high updating frequency of the planned path is difficult to be obtained to adapt the rapid changes of the swarm and target position. Moreover, efficient motion control of microrobotic swarms requires further investigations. The changes of external physical inputs may cause significant changes of agent-agent interactions among the swarms, and thus the stability of swarm patterns will be affected [26]. Meanwhile, the interactions and motion of swarm agents are complex, and the precise mathematical model of swarms is difficult to obtain [36]. Therefore, an effective and efficient strategy considering dynamic path planning and motion control for microrobotic swarms to track a mobile target are required to be investigated.

Herein, this work develops a control scheme of a microrobotic swarm for mobile target tracking. An enhanced bidirectional rapidly-exploring random tree star (EB-RRT*) algorithm is proposed as the path planner to dynamically generate the desire path avoiding obstacles collisions with sufficient updating frequency. By integrating the direction controller and the Genetic Algorithm based Linear Quadratic Regulator (GA-LQR) velocity controller, an image-guided motion controller is developed to control the moving direction and velocity of the swarm with high tracking precision. In order to meet the requirement of tracking a high-speed mobile target, a targeted bursting algorithm is proposed. Simulations and experiments of tracking a low-speed and a high-speed mobile target by the swarm in different environments, *i.e.*, an environment with virtual obstacles and a micromaze with virtual walls, are performed to validate the effectiveness and robustness of the proposed control scheme.

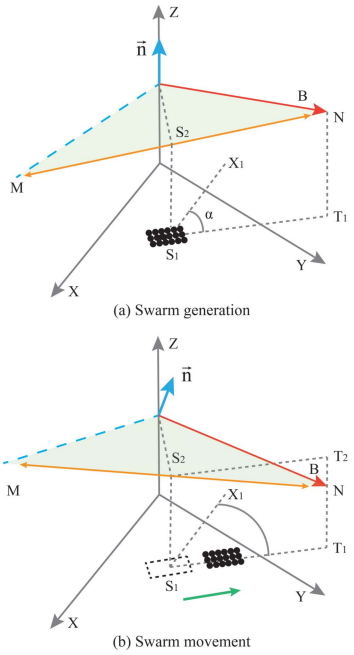


Fig. 1. The schematics of the swarm generation and movement actuated by oscillating magnetic fields. (a) Swarm generation. (b) Swarm movement actuated by an oscillating field with a pitch angle. The green, red and orange arrows denote the movement direction of the swarm, the magnetic field vector and the oscillation of the field, respectively. The normal vector of the magnetic field is indicated by $\vec{n}$, while the green triangle with the blue dotted boundary represents the magnetic field plane. The direction angle and pitch angle of the magnetic field are indicated by α and γ, respectively.

II. Modelling

A. Magnetic Actuation

Previously, ribbon-like swarms have been reported [25]. The schematics of swarm generation and movement triggered by an oscillating magnetic field are demonstrated in Fig. 1. Swarm generation actuated by an oscillating magnetic field is demonstrated in Fig. 1(a). By applying a small pitch angle (*e.g.*, 2-6°), the translational motion of the swarm can be realized, as shown in Fig. 1(b). The pitch angle is represented by γ, which denotes the angle between normal vector $\vec{n}$ of the magnetic field plane and Z axis. In Fig. 1(b), the oscillation of the field is indicated by line MN and its projection in $X-Y$ plane is represented by line S_1T_1. The line S_1X_1 is parallel to X-axis. By tuning the direction angle α, *i.e.*, the angle between line S_1T_1 and line S_1X_1, the swarm moving direction can be changed. The input actuation oscillating magnetic field can be expressed as:

$$B=\begin{bmatrix}B_x\\B_y\\B_z\end{bmatrix}=\begin{bmatrix}Acos(\alpha)cos(\gamma)sin(2\pi ft)-Csin(\alpha)\\Asin(\alpha)cos(\gamma)sin(2\pi ft)+Ccos(\alpha)\\-Asin(\gamma)sin(2\pi ft)\end{bmatrix} \quad (1)$$

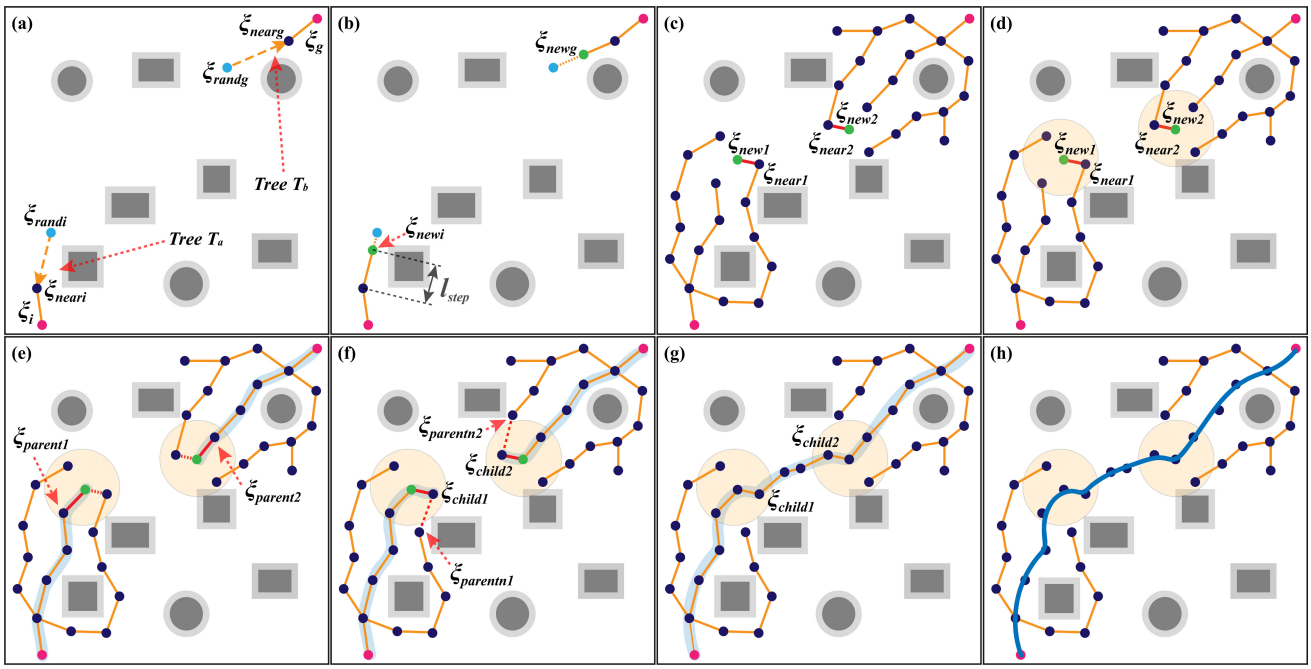


Fig. 2. The schematic illustration of the EB-RRT* dynamic path planning algorithm. (a) Random nodes sampling. The node ξ_{randi} and node ξ_{randg} are randomly sampled from two separate trees T_a and T_b, respectively. The nearest nodes of ξ_{randi} and ξ_{randg} are marked by ξ_{neari} and ξ_{nearg}, respectively. (b) Steering procedure. New nodes ξ_{newi} and ξ_{newg} are sampled after the process of steering, while l_{step} is the predefined step length. (c) Node extension. The current nodes of the two trees are represented by ξ_{new1} and ξ_{new2}, while their nearest nodes are ξ_{near1} and ξ_{near2}. (d) Definition of neighboring regions. The neighboring areas are marked by the orange circles whose center are the new nodes, *e.g.*, ξ_{new1}. (e) New parent node creation. $\xi_{parent1}$ and $\xi_{parent2}$ are the new parent nodes. (f) Rewiring procedure. The new child nodes are denoted by ξ_{child1} and ξ_{child2}, while their previous parent nodes are denoted by $\xi_{parentn1}$ and $\xi_{parentn2}$. (g) The connection of the two trees. (h) Path smoothing. The pink dots denote the starting nodes ξ_i and ξ_g, and the current nodes are represented by the green dots. The dark blue dots are the general nodes. The grey regions are the obstacles, and the surrounding light-grey regions are the collision buffer layers.

where B_x, B_y, B_z denote the corresponding component of the magnetic field in X, Y, Z axis, respectively.

In order to fully control the swarm motion, three critical parameters shall be precisely tuned, *i.e.*, oscillating frequency $f(t)$, direction angle $\alpha(t)$ and pitch angle $\gamma(t)$.

B. Analytical Model

It is challenging to establish an accurate analytical model for the swarm due to the large numbers of swarm agents and the complex interaction among them, *e.g.*, magnetic and hydrodynamic interactions. Herein, we apply the model:

$$\begin{cases} \dot{C}_x(t) = c_i g[f(t), \gamma(t)] cos(\alpha(t)) \\ \dot{C}_y(t) = c_i g[f(t), \gamma(t)] sin(\alpha(t)) \end{cases} \tag{2}$$

where $C[C_x(t), C_y(t)]$ is the position of the swarm *i.e.*, the centroid of the swarm contour, $\dot{C}_x(t)$ and $\dot{C}_y(t)$ are the derivatives of $C_x(t)$ and $C_y(t)$, respectively. The resultant factor $g[f(t), \gamma(t)]$ considering magnetic field frequency $f(t)$ and pitch angle $\gamma(t)$ affects the swarm velocity. As previously reported [36], the swarm velocity is mainly determined by the pitch angle $\gamma(t)$, and has a minor correlation with the input frequency $f(t)$. The linear function considering the above factors can thus be expressed as:

$$\begin{cases} \dot{C}_x(t) = c_v \gamma(t) cos(\alpha(t)) \\ \dot{C}_y(t) = c_v \gamma(t) sin(\alpha(t)) \end{cases} \tag{3}$$

where c_v is a positive constant that can be calibrated by experiments.

III. Path Planning and Motion Control

A. Dynamic Path Planning Algorithm

1) EB-RRT Path Planning Algorithm:* Herein, we apply an enhanced bidirectional rapidly-exploring random tree star (EB-RRT*) algorithm as the path planner and its schematics is shown in Fig. 2. Without losing any generality, tree T_a is taken as an example, and the dynamic path planning procedure is described as following steps.

As shown in Fig. 2(a), EB-RRT* starts from the initial node ξ_i and a random node ξ_{randi} is sampled from the free space χ_{free}. It is noted that, node ξ_{randi} will not be sampled from the region of obstacles. Based on ***Algorithm 1***, the nearest node ξ_{neari} of the random node ξ_{randi} with the shortest distance is selected by **NearestNode**, and meanwhile, a new node ξ_{newi} is chosen by the **Steer** function in ***Algorithm 1***. In Fig. 2(b), because $|\overrightarrow{\xi_{neari}\xi_{randi}}|$, *i.e.*, the norm of vector $\overrightarrow{\xi_{neari}\xi_{randi}}$ (a vector connecting node ξ_{neari} and node ξ_{randi}), is larger than the predefined step length l_{step}, node ξ_{newi} is selected as the new node. However, if $|\overrightarrow{\xi_{neari}\xi_{randi}}|$ is smaller than the step length l_{step}, the new node will be node ξ_{randi}. The proposed algorithm then utilizes **CollisionCheck** in ***Algorithm 1*** to check whether there is any collision between the vector $\overrightarrow{\xi_{neari}\xi_{newi}}$, *i.e.*, the vector connecting node ξ_{neari}

Algorithm 1 EB-RRT* Algorithm

```
CollisionBufferLayer();
T_a ← InitTree();
T_b ← InitTree();
T_a ← InsertNode(ξ_init, T_a);
T_b ← InsertNode(ξ_goal, T_b);
for k = 1 to K do:
    ξ_rand ← Sample(k);
    ξ_near ← NearestNode(ξ_rand, T_a);
    ξ_new ← Steer(ξ_rand, ξ_near);
    if(CollisionCheck(ξ_near, ξ_new) then:
        ξ_na ← NearArea(ξ_new, T_a, r);
        ξ_parent ← ChooseParent(ξ_na, ξ_near, ξ_new);
        T_a ← InsertNode(ξ_parent, ξ_new, T_a);
        T_a ← Rewire(ξ_na, x_parent, ξ_new, T_a);
        ξ'_near ← NearestNode(ξ_new, T_b);
        ξ'_new ← Steer(ξ_new, ξ'_near);
        if(CollisionCheck(ξ'_near, ξ'_new) then:
            ξ'_na ← NearArea(ξ'_new, T_b, r);
            ξ'_parent ← ChooseParent(ξ'_na, ξ'_near, ξ'_new);
            T_b ← InsertNode(ξ'_parent, ξ'_new, T_b);
            T_b ← Rewire(ξ'_na, x'_parent, ξ'_new, T_b);
if |T_a| < |T_b| then:
    Swap(T_a, T_b)
    PathSmoothing(T_a, T_b)
return T_a, T_b
```

and node ξ_{newi}, and obstacles. If collision is detected, the node ξ_{newi} will be discarded. A successful extension of tree T_a is shown in Fig. 2(c). By using the **NearArea** function in ***Algorithm 1***, a circle is generated, with its center locating in node ξ_{new1} and its radius of $r_{na} = \varepsilon(\frac{log_\kappa}{\kappa})^{\frac{1}{d}}$ [37], where $\epsilon \in [50, 60]$ is an independent constant, κ is the number of nodes in tree T_a, and d is the dimension of the space, *i.e.*, $d = 2$. The circle is regarded as the neighboring region of node ξ_{new1}, which is shown by the orange circle in Fig. 2(d). Once the neighboring area is generated, the new parent node can be determined by **ChooseParent** function in ***Algorithm 1***, which is represented by:

$$\begin{aligned} Cost_p^i &= \sum_{j=1}^{J_i-1} Dist(\xi_p^j, \xi_p^{j+1}) + Dist(\xi_p^{J_i}, \xi_{new1}) \\ \xi_{parent1} &\leftarrow min\{Cost_p^1, \ldots, Cost_p^i, \ldots, Cost_p^I\} \end{aligned} \tag{4}$$

where $i \in [1, I]$, $j \in [1, J_i]$, $J_i \in [J_1, J_I]$, p denotes the factors relevant to the **ChooseParent** function, ξ_p^j is the j-th node, $Cost_p^i$ is the i-th distance cost function, $\xi_{parent1}$ is the new parent node, and particularly, $\xi_p^1 = \xi_i$. As shown in Fig. 2(e), the connection of node ξ_{new1} and node ξ_{near1} (red dotted line) will be discarded. Subsequently, node $\xi_{parent1}$ inside the circular boundary is chosen as the new parent node and the red line denotes the new connection. After the new parent node is selected, the rewiring process is performed based on **Rewire** function in ***Algorithm 1***, which is demonstrated in Fig. 2(f). The mathematical expression of **Rewire** is shown as:

$$\begin{aligned} Cost_r^e &= Cost_p^P + Dist(\xi_{new1}, \xi_p^e) \\ &= \sum_{j=1}^{J_P-1} Dist(\xi_p^j, \xi_p^{j+1}) + Dist(\xi_p^{J_P}, \xi_{new1}) \\ &\quad + Dist(\xi_{new1}, \xi_p^e) \\ \xi_{child1} &\leftarrow min\{Cost_r^1, \ldots, Cost_r^e, \ldots, Cost_r^{J_I}\} \end{aligned} \tag{5}$$

where $j \in [1, J_P]$, $e \in [J_1, J_P) \cup (J_P, J_I]$, r denotes the factors relevant to **Rewire** function, ξ_p^e is e-th node of the remaining nodes in neighboring region, $Cost_p^e$ is the e-th distance cost function, ξ_{child1} is the new child node, and particularly, $\xi_p^{J_P} = \xi_{parent1}$. In this process, the green node ξ_{new1} is regarded as the new parent node. The current node ξ_{new1} selects node ξ_{near1} as its new child node since the Euclidean distance between them is the shortest. This leads to the disconnection between ξ_{near1} and its previous parent node $\xi_{parentn1}$, which is represented by the red dotted line in Fig. 2(f). Therefore, node ξ_{near1} is replaced by node ξ_{child1}. The other tree T_b will be formed starting from the goal node ξ_g, with the same processes shown in Fig. 2(a) - 2(f). Finally, two trees are connected with each other and the whole path is marked by the blue region shown in Fig. 2(g). The static planned path σ_p consists p_N pairs of nodes, with their positions represented by:

$$\begin{aligned} \sigma_p = \{&(x_c, y_c), (x_{p_1}, y_{p_1}), \ldots, (x_{p_n}, y_{p_n}), \ldots, \\ &(x_{p_N}, y_{p_N}), (x_r, y_r)\} \end{aligned} \tag{6}$$

where $p_n \in [p_1, p_N]$, (x_{p_n}, y_{p_n}) is the n-th point of the static planned path, $C(x_c, y_c)$ and $R(x_r, y_r)$ represent the positions of the swarm and the mobile target, respectively. However, since the swarm and the target are moving during the target tracking process, the dynamic updating of the planned path σ_p^t is demanded and it is defined as:

$$\begin{aligned} \sigma_p^t = \{&(x_c^t, y_c^t)|t, (x_{p_1}^t, y_{p_1}^t)|t, \ldots, (x_{p_n}^t, y_{p_n}^t)|t, \ldots, \\ &(x_{p_N}^t, y_{p_N}^t)|t, (x_r^t, y_r^t)|t\} \end{aligned} \tag{7}$$

where $t \in [t_1, t_T]$, $p_n \in [p_1, p_N]$, $(x_{p_n}^t, y_{p_n}^t)$ is the n-th point of the dynamic planned path at moment t, $C^t(c_x^t, c_y^t)$ and $R^t(r_x^t, r_y^t)$ are the swarm and target position at moment t, respectively. Because swarms may lose dynamic stability upon sudden shift of moving direction, the generated path shall be smoothened. The smoothed static path σ_s is shown in Fig. 2(h).

2) Collision Buffer Layers: Using the aforementioned method, even though the generated path avoids all obstacles, a swarm may still collide with the obstacles if their physical sizes are ignored. Therefore, the factor is taken into account by adding collision buffer layers surrounding the obstacles accordingly. By detecting the real-time shape and dimension of the swarm, the thickness of the collision buffer can be correspondingly tuned. For different shapes of obstacles, the collision buffer layer can be determined by:

$$\delta_k = c_k \frac{l_s^t}{2} \tag{8}$$

where l_s^t represents the length of the detected swarm at moment t, δ_k denotes the thickness of the collision buffer layer

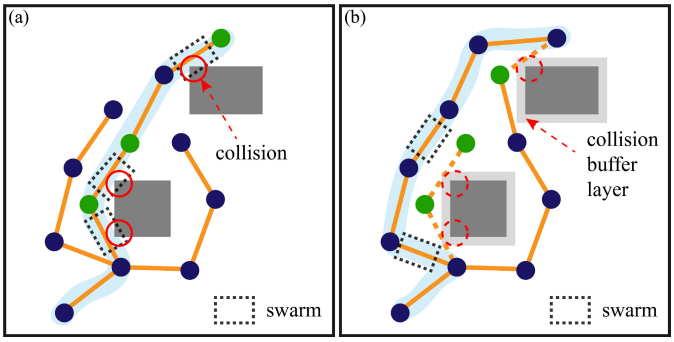


Fig. 3. The design of collision buffer layers. (a) The planned path without considering the collision buffer layers. The grey square and rectangle represent the obstacles. (b) The planned path considering the collision buffer layer. The light grey areas surrounding the grey obstacles are the collision buffer layers. The red circles indicate the regions of the collision between the swarm and obstacles, while the red dotted circles show that there is no collision by considering the collision buffer layers. The dotted orange lines denote the disconnection between nodes, whereas the orange lines are the connections between nodes. The dark blue dots are the general nodes belonging to the tree. The green dots are effective in (a) but invalid in (b). The final planned path is covered by the blue area. The dotted rectangles represent the swarms.

surrounding the k-th obstacle, and c_k is the secure constant to compensate the experimental error, including detection error. The range of the secure constants calibrated by preliminary experiments are [1.5, 2].

The comparison between the cases without and with collision buffer layers is shown in Fig. 3. If no buffer layers are created, as shown in Fig. 3(a), collision between the swarm and obstacles will occur since the swarm has a physical size. By adding collision buffer layers, the updated planned path can guarantee that the swarm will not collide with the obstacles, as shown in Fig. 3(b).

3) Path Smoothing: In order to improve the smoothness of the planned path, B-spline is applied and the smoothed static path σ_s represented by the blue curve in Fig. 2(h) is expressed as [38]:

$$\sigma_s = \sum_{n=1}^{N} \Phi_{n,m}(u)\sigma_{p_n} \tag{9}$$

where σ_{p_n} denotes n-th point of the planned path σ_p, $\Phi_{n,p}(u)$ represents the n-th B-spline basis function of degree m and u is a normalized curve knot. The first order and higher orders basis functions can be defined by Eq. 10 and Eq. 11, respectively [39]:

$$\Phi_{n,0} = \begin{cases} 1 & , \quad u \in [\hat{u}_n, \hat{u}_{n+1}) \\ 0 & , \quad else \end{cases} \tag{10}$$

$$\Phi_{n,m}(u) = \frac{u-\hat{u}_n}{\hat{u}_{n+m}-\hat{u}_n}\Phi_{n,m-1}(u) + \frac{\hat{u}_{n+m+1}-u}{\hat{u}_{n+m+1}-\hat{u}_{n+1}}\Phi_{n+1,m-1}(u) \tag{11}$$

where $\hat{u}_i$ is the knot vector of i-th knot u_i. Subsequently, the smoothed dynamic path σ_s^t at moment t is then represented by:

$$\sigma_s^t = \sum_{n=1}^{N} \Phi_{n,m}(u)\sigma_{p_n}^t \tag{12}$$

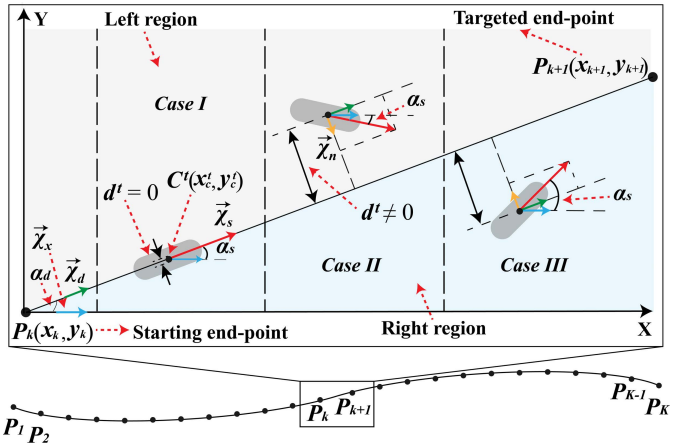


Fig. 4. The schematic diagram of swarm direction control. Three cases are presented, *i.e.*, the swarm locates on the line, in the left and right regions of the line. The rounded rectangles denote the swarm. The starting end-point and targeted end-point of the line segment $P_k P_{k+1}$ are represented by P_k and P_{k+1}, respectively. The blue arrow $\vec{\chi}_x$ is a unit vector parallel to X axis, while the green arrow $\vec{\chi}_d$ is a unit vector parallel to line segment $P_k P_{k+1}$. The angle between line segment $P_k P_{k+1}$ and X axis is denoted by α_d. The red arrow $\vec{\chi}_s$ denotes the unit vector of the swarm moving direction. The direction angle between $\vec{\chi}_s$ and $\vec{\chi}_x$ is denoted by α_s. The orange arrow $\vec{\chi}_n$ denotes the unit vector that is perpendicular to $\vec{\chi}_d$. The distance between the swarm center and the current path segment $P_k P_{k+1}$ at the moment t is denoted by d^t.

where $\sigma_{p_n}^t$ denotes n-th point of the dynamic planned path σ_p^t at moment t. The smoothed dynamic path guarantees the stability of the swarm during the target tracking process.

B. Image-Guided Motion Control

In order to precisely control the motion of the swarm, an image-guided motion controller is proposed, which controls the moving direction and velocity of a swarm.

1) Direction Control: The method to determine the moving direction of a swarm is shown in Fig. 4. The planned path generated by EB-RRT* is a sequence of key points, and each two adjacent points form a path segment. The procedure of target tracking is an iterative process. Each iteration ends if the distance $d_{C,P_{k+1}}^t$ between the swarm center C^t and the targeted end-point, *e.g.*, P_{k+1} in Fig. 4, of the current path segment is smaller than a predefined threshold ϵ, where ϵ is a real constant close to zero, *e.g.*, 5% body length of the swarm. When the swarm lies on the desired path segment, as shown in **Case I** of Fig. 4, the unit vector of swarm velocity $\vec{\chi}_s$ is equal to the unit vector of path segment $\vec{\chi}_d$, which ensures the low distance error during the target tracking process. In contrast, in **Case II** and **Case III**, the swarm is deviated from the desired path segment, and therefore, the unit vector of swarm velocity $\vec{\chi}_s$ is required to be modified to minimize the distance error d^t ($d^t \to 0$).

Given three points $P_k(x_k, y_k)$, $P_{k+1}(x_{k+1}, y_{k+1})$ and $C^t(x_c^t, y_c^t)$, the dynamic orientation coefficient c_{ori} can be obtained:

$$c_{ori} = det\begin{bmatrix} x_c^t - x_k & x_c^t - x_{k+1} \\ y_c^t - y_k & y_c^t - y_{k+1} \end{bmatrix} \tag{13}$$

Therefore, the relative position of the swarm center C^t and the desired path segment $P_k P_{k+1}$ can be determined:

Algorithm 2 Genetic Algorithm Based Linear-Quadratic Regulator (GA-LQR) Velocity Control

input : $a\ LQR\ specification\ (A, B, Q, R)$;
output : $an\ optimal\ feedback\ U^*(k)$;
backward pass:
  for $k = 1$ **to** K **do:**
    $min\ J \leftarrow \sum_{k=o}^{\infty}[X^T(k)QX(k) + U^T(k)RU(k)]$;
    $Q_g, R_g \leftarrow GA\ optimization$;
    $H_k \leftarrow Q_g + A^T H_{k+1} A - [A^T H_{k+1} B] \cdot [R_g + B^T H_{k+1} B]^{-1}[B^T H_{k+1} A]$;
    $L_k \leftarrow [R_g + B^T H_{k+1} B]^{-1} B^T H_{k+1} A$;
forward pass:
  for $k = 1$ **to** K **do:**
    $U_k^* \leftarrow -L_k X_k$
    $X_{k+1} \leftarrow AX_k + BU_k$
return U_k^*

Algorithm 3 Genetic Algorithm

input : population I_k; size K; initialize, $k = 0$;
output : best population I_b;
while $k \leq k_{desire}$ **do:**
  Use $p_s(I_k)$ to select parents I_1, I_2;
  Inject crossover sites with probability $p_c(I_k)$;
  Perform mutation under probability $p_m(I_k)$;
  Evaluate individuals with $F(I_k)$;
  Generate offspring;
  if $any\ doubles$ **or** $misses$ **then:**
    Eliminate doubles;
    Eliminate misses;
  end
  $k = k + 1$;
end
return best population I_b

- If $c_{ori} < 0$, the swarm locates in the left region.
- If $c_{ori} = 0$, the swarm lies on the line segment $P_k P_{k+1}$.
- If $c_{ori} > 0$, the swarm locates in the right region.

Accordingly, the angle α_o between vector $\vec{\chi}_s$ and vector $\vec{\chi}_d$ can be determined, which is expressed as:

$$\alpha_o = arccos\left(\frac{h\vec{\chi}_d \cdot ld^t \vec{\chi}_n}{|h\vec{\chi}_d||ld^t \vec{\chi}_n|}\right) \tag{14}$$

where h is a stable constant to maintain the low distance error when the swarm is close to the path segment, ld^t is a dynamic coefficient to guarantee a quick converge of distance error, l is the scale factor to ensure that vector $\vec{\chi}_d$ and vector $\vec{\chi}_n$ have the same order of magnitude. If $h/ld^t > 1$, the distance error tends to converge slowly but the swarm move smoothly. On the contrary, a quicker converge of distance error can be achieved if $h/ld^t < 1$. In this case, the moving direction of the swarm will be suddenly changed, which weakens the stability of the swarm. A suitable h can maintain the balance between the converge speed and the tracking accuracy. Finally, the direction vector $\vec{\chi}_s$ is modified using the following formula:

$$\vec{\chi}_s = \begin{cases} Rot(Z, \alpha_d - \alpha_o)\vec{\chi}_d, & c_{ori} < 0 \\ \vec{\chi}_d, & c_{ori} = 0 \\ Rot(Z, \alpha_o - \alpha_d)\vec{\chi}_d, & c_{ori} > 0 \end{cases} \tag{15}$$

where $\alpha_d = arccos\left(\frac{\vec{\chi}_d \cdot \vec{\chi}_x}{|\vec{\chi}_d||\vec{\chi}_x|}\right)$ is the angle between vector $\vec{\chi}_d$ and vector $\vec{\chi}_x$, and $Rot(\bullet)$ represents the rotation matrix of vector $\vec{\chi}_d$ around Z axis.

Instead of the unit direction vector $\vec{\chi}_s$, the angle α_s between vector $\vec{\chi}_s$ and vector $\vec{\chi}_x$ will be used in experiments to tune the moving direction of the swarm, which is described as:

$$\alpha_s = \begin{cases} arccos\left(\frac{\vec{\chi}_x \cdot Rot(Z, \alpha_d - \alpha_o)\vec{\chi}_d}{|\vec{\chi}_x||Rot(Z, \alpha_d - \alpha_o)\vec{\chi}_d|}\right), & c_{ori} < 0 \\ arccos\left(\frac{\vec{\chi}_x \cdot \vec{\chi}_d}{|\vec{\chi}_x||\vec{\chi}_d|}\right), & c_{ori} = 0 \\ arccos\left(\frac{\vec{\chi}_x \cdot Rot(Z, \alpha_o - \alpha_d)\vec{\chi}_d}{|\vec{\chi}_x||Rot(Z, \alpha_o - \alpha_d)\vec{\chi}_d|}\right), & c_{ori} > 0 \end{cases} \tag{16}$$

Therefore, the moving direction of the swarm can be determined by Eq. 16.

2) GA-LQR Velocity Control: In order to reduce the distance error during the locomotion of the swarm, a Linear Quadratic Regulator velocity controller with Genetic Algorithm (GA-LQR) is developed. The discrete-time state space equation of the system can be represented by:

$$\begin{cases} X_{k+1} &= AX_k + BU_k \\ Y_k &= CX_k \end{cases} \tag{17}$$

where X is the system state, U is the control feedback, Y is the system output,

$$A = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},\ B = \begin{bmatrix} c_v & 0 \\ 0 & c_v \end{bmatrix},\ C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{18}$$

and c_v is a positive constant that can be calibrated by experiments. As shown in ***Algorithm 2***, the GA-LQR velocity control is applied for generating the optimal linear feedback $U_k = -L_k X_k + X_d$ (*i.e.*, the control input of the magnetic system), where X_d is the desired state. The optimal feedback U_k is further used to minimize the quadratic cost function, which is expressed as:

$$J = \sum_{k=0}^{K}(X_k^T Q X_k + U_k^T R U_k) \tag{19}$$

where K is the time horizon, $Q > 0$ and $R > 0$ are weighting matrices. Because X_d serves as the external input of the GA-LQR control system and it has negligible effect on the system stability [40], the simplified feedback $U_k = -L_k X_k$ is used in our experiments. The input U_k of the GA-LQR control loop is relevant to the feedback gain L_k, as shown in ***Algorithm 2***, and U_k will be further affected by the weighting matrices Q and R. The system with a large value of Q, *e.g.*, $Q \in [1000, 5000]$, can ensure a smaller distance error, but the swarm could be unstable during locomotion. On the contrary, the system with the $R \in [1000, 2000]$ may reduce the sudden change of the swarm velocity, which ensures the stability of a swarm. However, the higher value of R may cause low tracking accuracy. Therefore, in order to find the optimal values of Q and R, the Genetic Algorithm in ***Algorithm 3*** is introduced to tackle the optimization problem. The specific ranges are defined: $Q \in (0, Q_l]$, $R \in [R_s, R_l]$. With K numbers, an initial population of chromosomes which

contains k gene bit is randomly chosen from the set $\Omega = \omega_1, \omega_2, \ldots, \omega_x$. The fitness function is described as follows:

$$F(I_k) = X_k^T Q X_k + U_k^T R U_k \tag{20}$$

where $k \in [1, K]$ and K is the maximum number of iterations. The function $F(I_k)$ is used to evaluate the fitness of each individual in the population. The individuals with suitable fitness values are stochastically selected from the current population to form a new generation, whereas the other individuals will be discarded. In this case, the selection probability of each chromosome I_k is inversely proportional to their fitness, which is defined as [41]:

$$p_s(I_k) = \frac{F(I_k)}{\sum_{k=1}^{K} F(I_k)} \tag{21}$$

where $k \in [1, K]$ and K is the maximum number of iterations. Therefore, the chromosomes with a low fitness value are more likely to be selected for the offspring. The crossover sites of genes are chosen and exchanged among the independent gens. The adaptive process promises a higher crossover probability in the parent generation, which can be expressed as [34]:

$$p_c(I_k) = \begin{cases} p_c(I_1), \ if \ k = 1 \\ \frac{k}{K}[p_c(I_1) - \frac{\sum_{u=1}^{k} p_c(I_u)}{k}], \ if \ k > 1 \end{cases} \tag{22}$$

where $k \in [1, K]$ and $p_c(I_1)$ is a constant, *e.g.*, 0.01. After crossover between individual samples, the mutation process occurs. Replacing individuals of population on random gene segments with certain mutation probability is the key operation of mutation. The procedure can be analytically expressed as [35]:

$$p_m(I_k) = \begin{cases} p_m(I_1), \ if \ k = 1 \\ \frac{F(I_k) - F_{mean}(I_k)}{F_{max}(I_k) - F_{mean}(I_k)}, \ if \ k > 1 \end{cases} \tag{23}$$

where $k \in [1, K]$ and $p_m(I_1)$ is a constant, *e.g.*, 0.01, $F_{mean}(I_k)$ and $F_{min}(I_k)$ are the mean and minimum fitness value, respectively. The process of selection, crossover and mutation is repeated until the maximum number of iterations is reached. After the optimization process using the Genetic Algorithm, the optimal Q_g and R_g are selected. By substituting Q_g and R_g into the previous equations, *i.e.*, Eq. 17 and Eq. 19, the solution of the discrete-time algebraic Riccati equation of GA-LQR is defined as:

$$H_k = Q_g + A^T H_{k+1} A - [A^T H_{k+1} B] \cdot [R_g + B^T H_{k+1} B]^{-1} [B^T H_{k+1} A] \tag{24}$$

where the transient matrix is H. Therefore, the feedback gain L_k and optimal control input U_k^* in ***Algorithm 2*** are represented by:

$$\begin{cases} L_k &= [R_g + B^T H_{k+1} B]^{-1} [B^T H_{k+1} A] \\ U_k^* &= -L_k X_k \end{cases} \tag{25}$$

Subsequently, the discrete-time input parameters of the magnetic field are described as:

$$\begin{cases} \gamma_k &= sel(\sqrt{(U_{k_x}^*)^2 + (U_{k_y}^*)^2}, \ \gamma_{max}) \\ \alpha_k &= \alpha_s \end{cases} \tag{26}$$

where $sel(a, b)$ is a selection function [33], which returns the smaller value between a and b:

$$sel(a, b) = \begin{cases} a \ , & a \le b \\ b \ , & a > b \end{cases} \tag{27}$$

C. Targeted Bursting Algorithm

In order to improve the tracking accuracy, the GA-LQR velocity control method will cause the decrease of velocity when the swarm is close to the targeted end-point of each path segment, which may cause the failure of tracking a high-speed mobile target. Herein, we further modify the control scheme by tailoring the targeted bursting algorithm. Considering the morphology of the target, we define a circular region surrounding it, *i.e.*, the bursting region, with its radius of $r_{ba} = c_{bu} r_T$, where r_T is the radius of the circular target, and $c_{bu} \in [1.2, 1.5]$ is a constant to enlarge the circular region of the mobile target. The bursting region and the circular target share the same center. When the swarm reaches the bursting region, the path σ_s^t is replaced by a new one σ_{sa}^t, which is expressed as:

$$\sigma_{sa}^t = \begin{cases} \{(x_c^t, y_c^t) \,|\, t), (x_r^t, y_r^t) \,|\, t)\}, & 0 < d_{C,R}^t \le \frac{r_{ba}}{2} \\ \{(x_c^t, y_c^t) \,|\, t, (x_m^t, y_m^t) \,|\, t, (x_r^t, y_r^t) \,|\, t\}, & \frac{r_{ba}}{2} < d_{C,R}^t \le r_{ba} \end{cases} \tag{28}$$

where $t \in [t_1, t_G]$, $d_{C,R}^t$ is the distance between the swarm $C^t(x_c^t, y_c^t)$ and the mobile target $R^t(x_r^t, y_r^t)$ at the moment t, $x_m^t = \frac{(x_c^t + x_r^t)}{2}$ and $y_m^t = \frac{(y_c^t + y_r^t)}{2}$. Moreover, during the bursting process, the pitch angle γ_k generated by the image-guided motion controller is replaced by the largest pitch angle γ_{max} that the swarm can still maintain its stability.

The overall control diagram is presented in Fig. 5. It consists of the dynamic path planning unit, the imaging-guided motion control unit and the targeted bursting unit. The external environmental conditions are firstly identified by the dynamic path planning unit. With a known environment, the positions of the target R^t and the obstacles $O_{k(k=1,\ldots,n,\ldots,N)}$ are obtained accordingly, as the input of the entire control system. The positions are then input to the EB-RRT* path planner to generate a smoothed dynamic path σ_s^t, which serves as the input of the image-guided motion controller. The real-time position of the detected swarm is considered as the feedback signal of the motion controller. Subsequently, the distance error d^t is obtained and the direction controller is activated to control the moving direction of the swarm. Meanwhile, the distance $d_{C,P_{k+1}}^t$ between the swarm and the targeted end-point is also obtained for the GA-LQR velocity controller, which leads to the desire pitch angle and velocity of the swarm. With desire direction angle and pitch angle, the oscillating field is then generated by the 3D Helmholtz coil system to actuate the swarm. In order to meet the requirement of tracking a

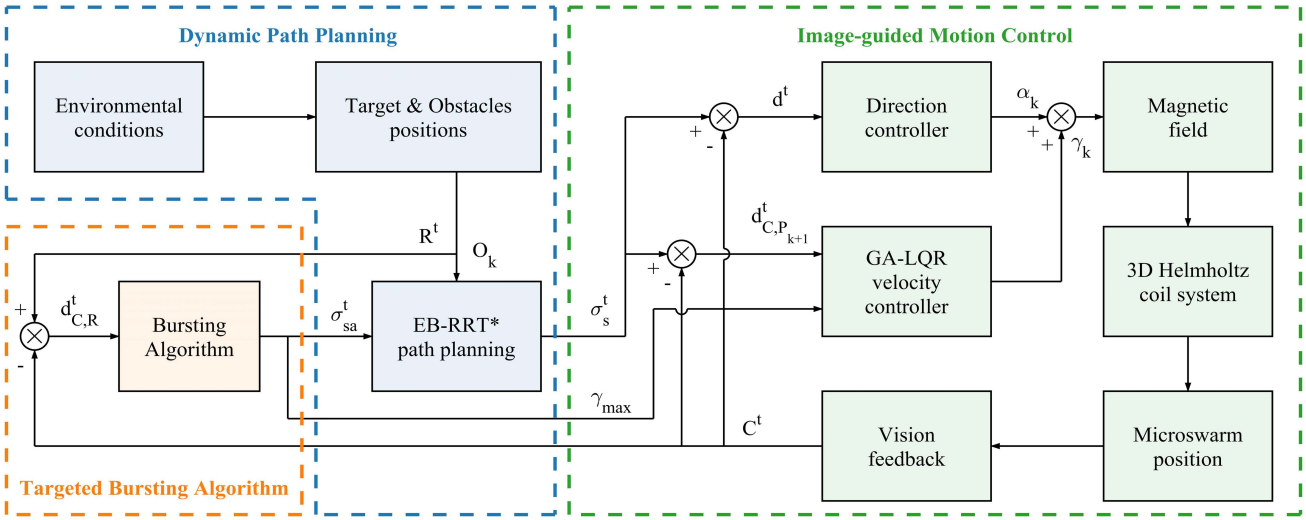


Fig. 5. The system diagram consisting of the dynamic path planning unit, the image-guided motion control unit and the targeted bursting unit. The positions of the swarm and the target are represented by C^t and R^t, respectively. The position of k-th obstacle is denoted by O_k. The distance between the swarm and the line segment $P_k P_{k+1}$, *i.e.*, the distance error, at moment t is d^t. The distance between the swarm and the target at moment t is $d^t_{C,R}$. The distance between the swarm and the targeted end-point at moment t is $d_{C,P^t_{k+1}}$. The paths generated by the EB-RRT* path planning algorithm and the bursting algorithm are represented by σ^t_s and σ^t_{sa}, respectively. The desire direction angle generated by the direction controller is indicated by α_k, while the desire pitch angle generated by the GA-LQR velocity controller is denoted by γ_k. The largest pitch angle that the swarm can still maintain its stability is labelled by γ_{max}.

high-speed mobile target by the swarm, the targeted bursting algorithm is designed. Similarly, the position of the detected swarm serves as the feedback signal of the bursting process.

IV. Simulation

Simulations are performed to testify the effectiveness of the EB-RRT* path planning, the motion control and the targeted bursting algorithm by tracking mobile targets, which will bounce when contacting with obstacles and walls.

A. Formation of Trees

The simulation results of the planned paths and branch configurations are shown in Fig. 6. The results when the distance between the swarm and the target is small are shown in Fig. 6(a) and 6(b), respectively. In these two cases, the simulated swarms are close to the targets in an environment with obstacles and a micromaze. The simulation results show that paths are generated with few branches, and only a small region is searched by the path planner, indicating the high efficiency of the path planner in complex environments. On the contrary, if the starting position has a large distance to the target position in an environment with obstacles and a micromaze, well-planed paths that avoid collision from obstacles and walls can still be generated by the path planner through extending the branches, which are demonstrated in Fig. 6(c) and 6(d), respectively. The simulation results indicate the considerable robustness of the EB-RRT* path planner.

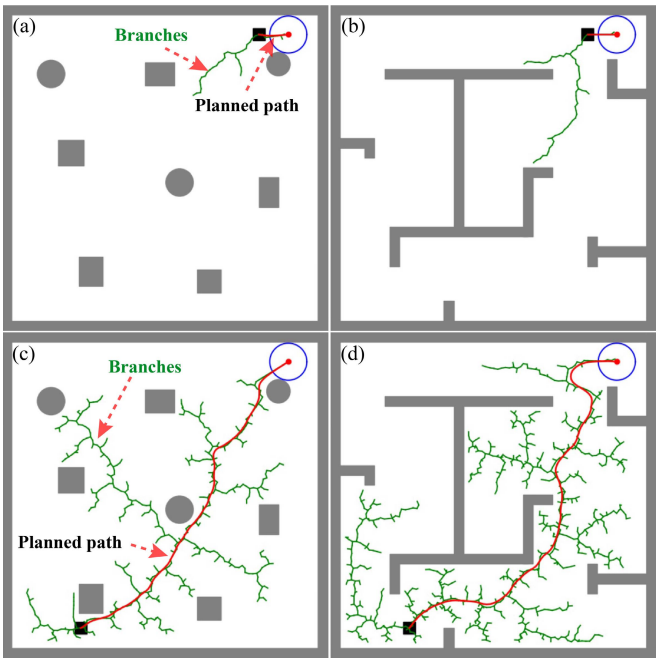


Fig. 6. The formation of trees with different external conditions. (a) The tree configuration when the simulated swarm is close to the target in an environment with obstacles. (b) The tree configuration when the simulated swarm is close to the target in a micromaze. (c) The tree configuration when the simulated swarm has a large distance to the target in an environment with obstacles. (d) The tree configuration when the simulated swarm has a large distance to the target in a micromaze. The black squares denote the simulated swarms while the dark blue circles represent the targets. The tree branches are marked by green curves, and the final paths are outlined with a red curve.

B. Dynamic Path Planning and Mobile Target Tracking

We firstly conduct the simulation of EB-RRT* dynamic path planning in the environment with virtual circular, square and rectangular obstacles, and the results are shown in Fig. 7. A low-speed (*i.e.*, 12 $\mu m/s$) mobile target is applied in Fig. 7(a). In the simulation, the swarm has a translational velocity of 16 $\mu m/s$. The swarm is actuated to approach

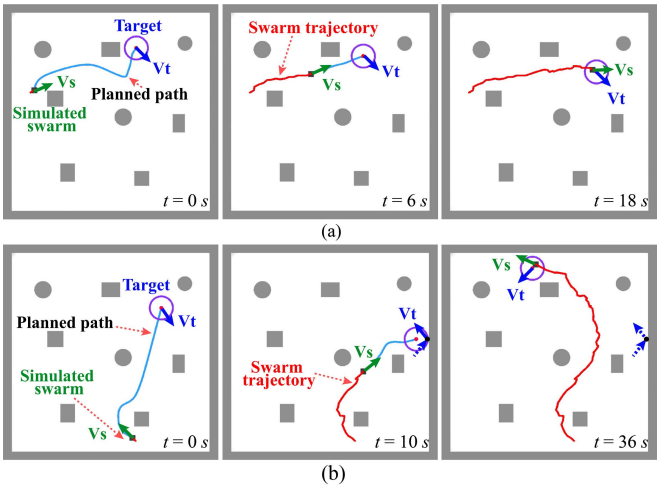


Fig. 7. The simulation results of mobile target tracking. (a) The tracking of a low-speed target with EB-RRT* path planning algorithm in an environment with obstacles. (b) The tracking of a high-speed target with EB-RRT* path planning algorithm in an environment with obstacles. The velocities of the swarm and the target are represented by V_s and V_t, respectively. The blue curve is the dynamic planned path and the red curve shows the trajectory of the swarm. The dotted blue arrows show the previous bounces, and the black dots are the bouncing points.

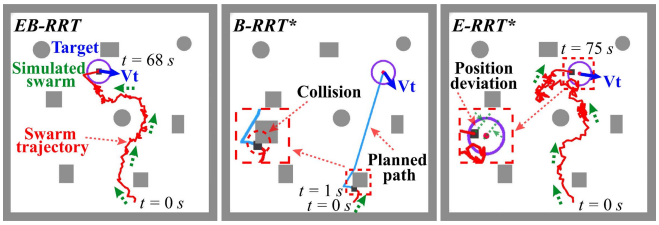


Fig. 8. The simulated results of tracking high-speed targets with EB-RRT, B-RRT* and E-RRT* path planning algorithms in an environment with obstacles. The velocity of the target is represented by V_t. The blue curve is the dynamic planned path and the red curve shows the trajectory of the swarm. The dotted green arrows show the moving direction of the simulated swarm. The enlarged insets provide clear observations of collision and position deviation in experiments.

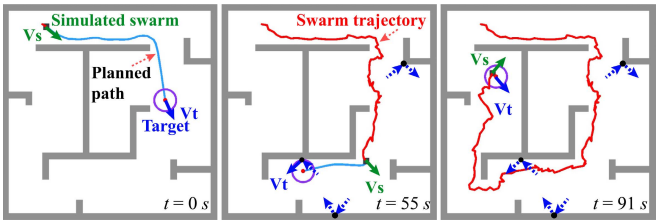


Fig. 9. The simulation results of tracking a high-speed target with EB-RRT* path planning algorithm in a micromaze.The velocities of the swarm and the target are represented by V_s and V_t, respectively. The blue curve is the dynamic planned path and the red curve shows the trajectory of the swarm. The dotted blue arrow pairs show the previous bounces and the black dots are the bouncing points.

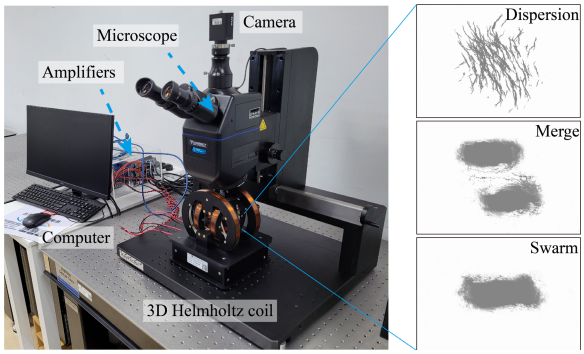


Fig. 10. The three-axial Helmholtz electromagnetic coil setup. The zoomed-in insets show different states of the magnetite nanoparticles, *i.e.*, dispersion, merge and swarm.

and subsequently track the target at $t = 6\ s$ and $t = 18\ s$. It is noted that, in this case, the bouncing target is tracked by the swarm before it contacts the boundary or obstacles, and thus, there is no sudden shift of the target motion direction. Furthermore, in order to testify the effectiveness of the dynamic path planning algorithm in a condition that the target encounters several bounces with the obstacles or the boundaries, a high-speed (*i.e.*, 20 $\mu m/s$) mobile target is applied. In this case, the moving direction of the target changes frequently, making the target trajectory complex. Based on the simulation results shown in Fig. 7(b), it can be observed that, the EB-RRT* algorithm can still generate the dynamic path without obstacle collision. Meanwhile, the motion control and bursting algorithm are also validated in this case. When the simulated swarm reaches the predefined bursting region, the targeted bursting process is triggered. During the process, the swarm is accelerated to track the mobile target with a re-planned path, and the maximum velocity of the swarm that the swarm can still maintain its stability is applied. As a result, the simulated swarm can track the high-speed mobile target.

For comparisons, the simulation results of tracking high-speed mobile targets using three other path planning algorithms, *i.e.*, enhanced bidirectional rapidly-exploring random tree (EB-RRT), bidirectional rapidly-exploring random tree star (B-RRT*) and enhanced rapidly-exploring random tree star (E-RRT*) algorithm, are shown in Fig. 8. Although the EB-RRT algorithm generates non-optimal paths, the mobile target can still be successfully tracked. However, the swarm moves longer distance to reach the target, and thus, it results in a longer time period (*i.e.*, 68 s) to complete the tracking process, which is longer than that using the EB-RRT* algorithm (*i.e.*, 36 s). When the B-RRT* algorithm is applied, the physical size of the swarm is ignored, and there are no collision buffer layers surrounding the obstacles. Therefore, the collision between the simulated swarm and the obstacle occurs. When the E-RRT* algorithm is applied, the swarm cannot reach the current position of the mobile target. The E-RRT* algorithm takes more time in generating the planned path, since only one tree will be deployed. As a result, the planned path can only connect the swarm and the previous position of the mobile target. The algorithm fails to adapt to the rapid changes of the swarm and target position. The results show that the EB-RRT* algorithm can effectively and efficiently generate the desired paths to avoid the collision with obstacles during the mobile target tracking.

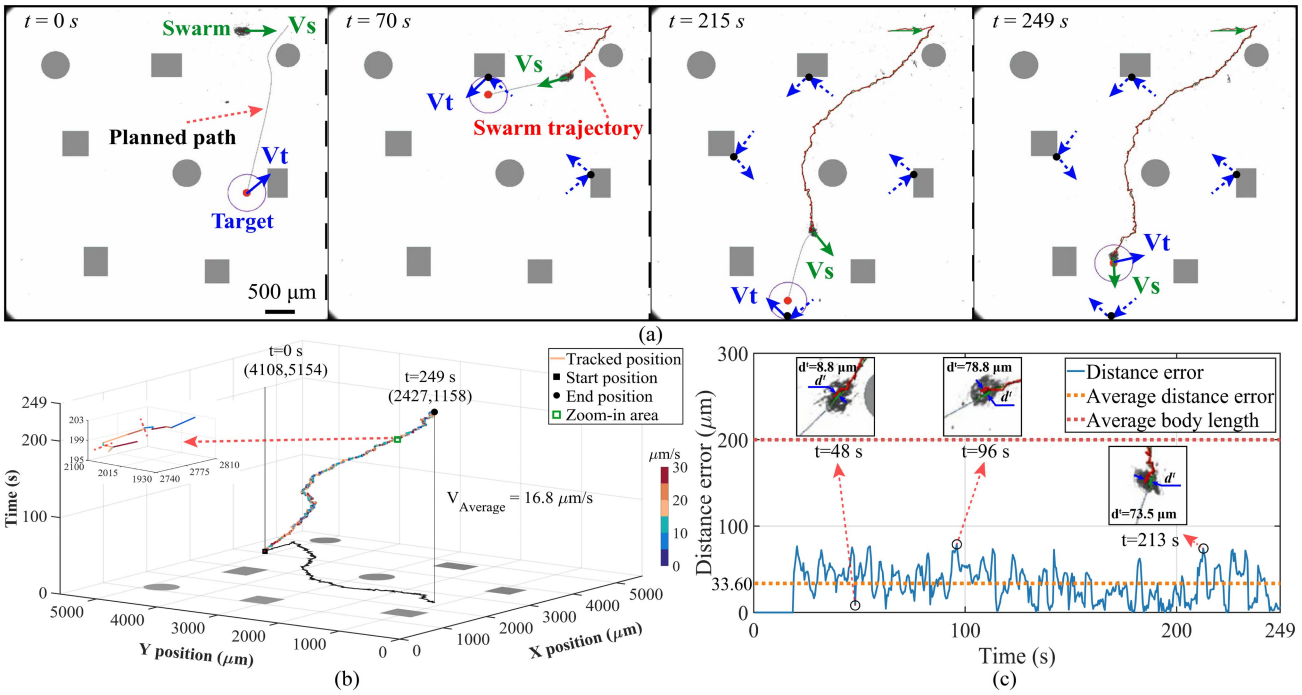


Fig. 11. The experimental results of tracking a low-speed mobile target in an environment with obstacles. (a) The tracking results for the swarm. The velocities of the swarm and the target are represented by V_s and V_t, respectively. The dotted blue arrow pairs show the previous bounces and the black dots are the bouncing points. (b) The plot of trajectory over time. Different colors indicate the speed distribution of the swarm over time. The grey areas represent the obstacles and the black path denotes the 2D trajectory of the swarm. The starting position and terminal position are marked by black square and circle, respectively. The red dotted lines separate the iterations. (c) The distance error of the target tracking. The enlarged insets provide clear observations of distance error in experiments. The average body length of the swarm is 200 μm. The scale bar is 500 μm.

C. Mobile Target Tracking in a Micromaze

A micromaze is deployed to further testify the proposed control scheme. As shown in Fig. 9, in the highly constrained environment, the moving trajectory of the mobile target is irregular due to the bouncing on the wall of the micromaze. In this case, the simulated swarm is still capable of tracking the high-speed mobile target, which further validates the effectiveness of the proposed method in constrained environments. The simulations indicate that the proposed control scheme is effective for tracking both the low-speed target and the high-speed mobile target using the swarm in different environments, *i.e.*, an environment with obstacles and a micromaze.

V. Experimental Setup

The experiments are conducted in a three-axial Helmholtz electromagnetic coil setup, which mainly includes an optical microscope (Model PS888, SEIWA Optical CO.,LTD.), an sCMOS camera (Model GS3-U3-41C6C-C, Teledyne FLIR LLC.) and a host computer, as shown in Fig. 10. The control signals generated by the host computer is amplified by servoamplifiers (Model JSP-180-10, Analogic Corporation), and current with corresponding amplitude will be input to the coils, which generates on-demand magnetic fields. The strength and frequency of the magnetic field are maintained as 10 mT and 10 Hz, and the framerate of the camera is 10 frame per second (fps). In the experiments, a piece of silicon wafer is used as the substrate. The upward polished surface of the silicon wafer can enhance the imaging contrast during observation, and a clear vision feedback for the closed-loop control system can be provided.

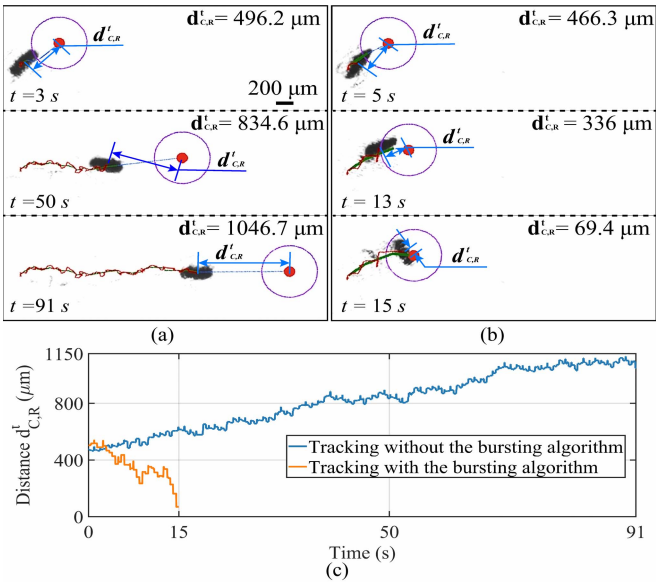


Fig. 12. The validation of the targeted bursting algorithm. (a) The results of tracking a high-speed mobile target without applying the bursting algorithm. (b) The results of tracking a high-speed mobile target with applying the bursting algorithm. (c) The results of distance $d'_{C,R}$ between the swarm center and the target. The scale bar is 200 μm.

Magnetite nanoparticles with an average diameter of 50 nm are dispersed in deionized (DI) water. One drop of nanoparticle suspension (2 μL, 1 mg/ml) is added into a tank. The tank is

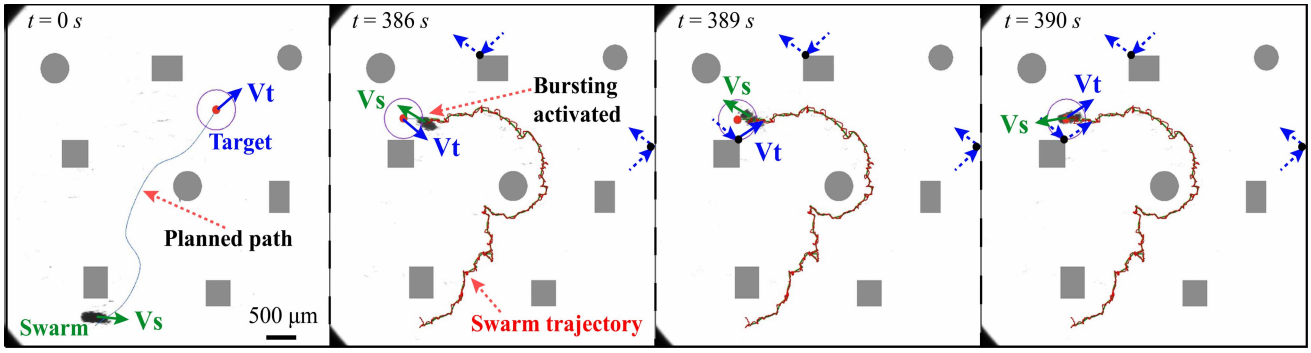


Fig. 13. The experimental results of tracking a high-speed mobile target with applying the bursting algorithm using EB-RRT* path planning algorithm in an environment with obstacles. The velocities of the swarm and the target are represented by V_s and V_t, respectively. The dotted blue arrow pairs show the previous bounces and the black dots are the bouncing points. The scale bar is 500 μm.

filled with DI water with addition of 20 μL Tween 20 solution. The particles are gathered into clusters by applying a magnetic field gradient, and then, the particle clusters are put into the workspace of the magnetic coils for further actuation and control.

VI. Experimental Results

A. Mobile Target Tracking With Obstacle Avoidance

The proposed control scheme is further applied for tracking mobile targets in an environment with virtual obstacles. The circular targets are mobile circles with bouncing boundary condition (*i.e.*, they will bounce when contacting with obstacles and walls).

1) Low-Speed Mobile Target Tracking: The results of low-speed (*i.e.*, 12 $\mu m/s$) mobile target tracking are shown in Fig. 11(a). Even though the moving direction of the mobile target may change suddenly due to the bounces, the EB-RRT* dynamic path planning algorithm with sufficient updating frequency (*i.e.*, 1.7-2.4 Hz) is capable of generating desire path avoiding collision with the obstacles, as shown by the blue curves. The bouncing points are labelled by the black dots.

After the validation of dynamic path planning, the effectiveness of the proposed image-guided motion control method, which combines the dynamic path planning unit, the direction controller and the GA-LQR velocity controller, for both velocity and direction control of the swarm is also testified. As demonstrated in Fig. 11(a), the initial positions of the swarm and the mobile target are shown at $t = 0\ s$. Due to the obstacles and the bouncing boundary condition, the moving trajectory of the target is complex and difficult to predict, and it is also blinded for the controller. For instance, at $t = 70\ s$ and $t = 117\ s$, the mobile target contacts rectangular obstacles, which causes the sudden change of the moving direction of the target. After several bouncing processes, the mobile target approaches the bottom boundary at $t = 215\ s$. The swarm is continuously actuated to track the target by following the dynamically planned trajectories avoiding collision with obstacles. Finally, at $t = 249\ s$, the mobile target can be successfully tracked. It can be noted that the trajectory of the swarm (*i.e.*, the red curve) well matches the planned path towards the mobile target (*i.e.*, the green curve), as shown in Fig. 11(a), indicating a small distance error. The locomotion

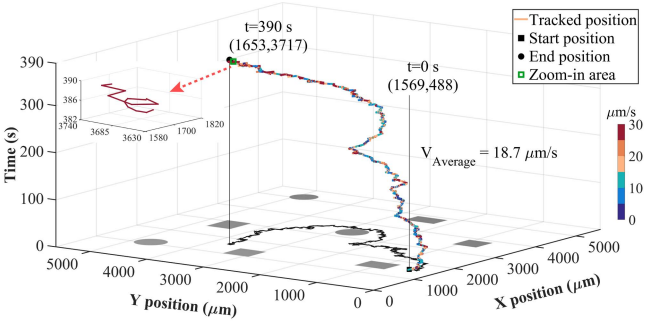


Fig. 14. The plots of trajectory over time using GA-LQR velocity control algorithm. Different colors indicate the speed distribution of the swarm over time. The grey areas represent the obstacles and the black path denotes the 2D trajectory of the swarm. The starting position and terminal position are marked by red square and circle, respectively.

trajectory over time is shown in Fig. 11(b). The initial position of the swarm is (4108, 5154) μm and the average velocity of the swarm is 16.8 $\mu m/s$. As shown in the inset, the color changes from yellow to blue in one iteration and the change of the color is periodic across the tracking process, indicating that the velocity of the swarm decreases when approaching targeted end-point in each iteration. Therefore, it is indicated that the GA-LQR velocity controller can well control the velocity of the swarm. The distance error is demonstrated in Fig. 11(c). The highest distance error (*i.e.*, 78.8 μm) is 39% of the average body length of the swarm (*i.e.*, 200 μm), and the average distance error (*i.e.*, 33.6 μm) is 17% of the swarm body length. The amplitude oscillation of distance error in Fig. 11(c) keeps remaining within a reasonable range. The results indicate that the proposed image-guided motion control method with the GA-LQR velocity controller can ensure the high accuracy of tracking.

2) High-Speed Mobile Target Tracking: As discussed in *Section III Part C*, the swarm velocity decreases when it is close to the targeted end-point in each iteration. In order to overcome the drawback and meet the requirements of tracking a high-speed (*i.e.*, 20 $\mu m/s$) mobile target, the tracking control scheme with the targeted bursting algorithm is demanded. The control scheme without the bursting algorithm, in which the other units of the control scheme (*i.e.*, the EB-RRT* path planner, the direction controller and the GA-LQR velocity

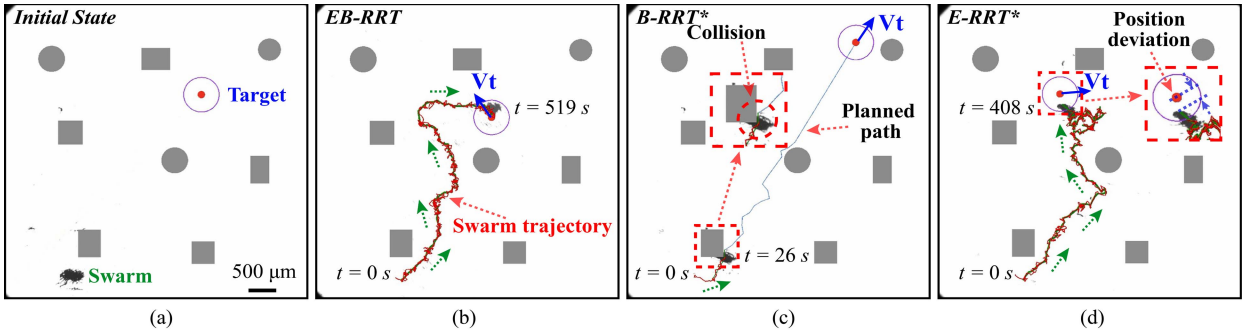


Fig. 15. The experimental results of tracking a high-speed mobile target using different path planning algorithms in an environment with obstacles. (a) The initial positions of the swarm and the target. (b)-(d) The tracking results using EB-RRT, B-RRT* and E-RRT* algorithms, respectively. The velocity of the target is represented by V_t. The dotted green arrows show the moving direction of the simulated swarm. The enlarged insets provide clear observations of collision and position deviation in experiments. The scale bar is 500 μm.

controller) are maintained equally, is also applied for comparisons. The effectiveness of the bursting algorithm is firstly validated by tracking a high-speed mobile target in a free space, as shown in Fig. 12. Swarms maneuvered by the control scheme without and with the bursting algorithm are deployed to track high-speed mobile targets. The mobile targets are released in a free environment. The experimental conditions, such as the initial position of the swarm and the target, the initial velocity of the swarm and the target, and the locations of the obstacles, are kept approximately the same. In Fig. 12(a), the distance $d^t_{C,R}$ between the swarm and the target increases because the swarm velocity decreases when the swarm is actuated close to the targeted end-point in each iteration. Therefore, the swarm cannot reach the target in this case. On the contrary, by activating the bursting unit, the swarm can successfully track the high-speed mobile target within a short time period, as shown in Fig. 12(b). The bursting process is triggered when the swarm reaches the bursting region of the mobile target. The comparison between these two cases is shown in Fig. 12(c). The distance between the swarm and the target keeps increasing with time if the bursting algorithm is not applied, while the distance decreases rapidly if the bursting algorithm is applied, indicating the effectiveness of the bursting algorithm.

To further testify the effectiveness of the bursting algorithm, the experiments of tracking the high-speed (*i.e.*, 20 $\mu m/s$) mobile target are subsequently conducted in an environment with obstacles. By applying the proposed bursting algorithm, the control results are shown in Fig. 13. The swarm is initially actuated to follow the planned path to track the mobile target. In this case, the bursting region surrounding the mobile target is activated, and the bursting behavior is triggered when the swarm reaches the bursting region at $t = 386$ s. The path towards the target is re-planned, and the swarm is accelerated due to the increased input pitch angle of the magnetic field. Therefore, the target can be tracked within a short time, *i.e.*, 390 s. The locomotion trajectory over time using the proposed control scheme is shown in Fig. 14. The initial position of the swarm is (1569, 488) μm is selected and the average velocity of the swarm is 18.7 $\mu m/s$. By observing the enlarged inset, the yellowish color dominates when the bursting process is triggered, *i.e.*, the latter part of the trajectory, which indicates an increased velocity of the swarm. The results indicate that

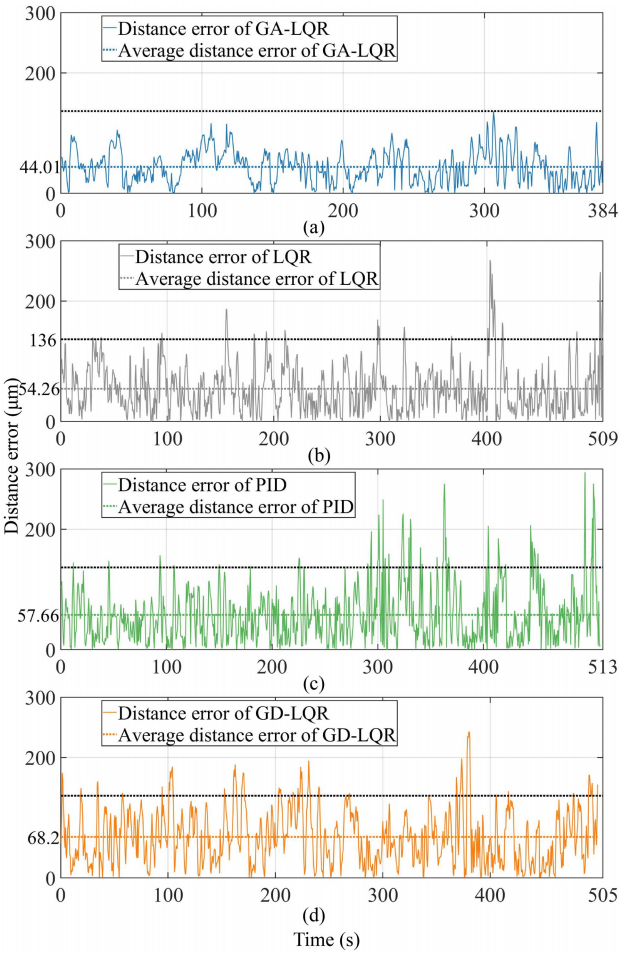


Fig. 16. The distance error of the mobile target tracking using different velocity control algorithms. (a)-(d) Distance errors using GA-LQR, LQR, PID and GD-LQR velocity control algorithms, respectively. The average body length of the swarm is 200 μm.

the tracking control scheme is effective and has high tracking efficiency.

In order to testify the performance of the EB-RRT* algorithm, three other path planning algorithms, *i.e.*, enhanced bidirectional rapidly-exploring random tree (EB-RRT), bidirectional rapidly-exploring random tree star (B-RRT*) and enhanced rapidly-exploring random tree star (E-RRT*) algorithm, are performed for comparisons. In this case, the other

units of the control scheme, *i.e.*, the direction controller, the GA-LQR velocity controller and the bursting algorithm, are equally employed. The control results are shown in Fig. 15. The experimental conditions, *i.e.*, the initial positions of the swarm and the target, and the locations of the obstacles, are kept equally, as shown in Fig. 15(a). As demonstrated in Fig. 15(b), the mobile target can be successfully tracked when the EB-RRT algorithm is applied. However, it causes a longer time period (*i.e.*, 519 s) compared to that cost in the trail using the proposed EB-RRT* algorithm (*i.e.*, 390 s). The B-RRT* algorithm causes collision between the swarm and a rectangular obstacle, as presented in Fig. 15(c). In this case, the physical size of the swarm is ignored, indicating that there are no collision buffer layers surrounding the obstacles. The tracking result using the E-RRT* algorithm is shown in Fig. 15(d). The swarm can only reach the previous positions of the mobile target. The experimental results have a good agreement with the simulation results shown in Fig. 8, and it reveals that the EB-RRT* algorithm can ensure the shortest time cost during the mobile target tracking.

After the verification of the bursting algorithm and the EB-RRT* path planning algorithm, the performance of the Genetic Algorithm is validated and the other units of the control scheme, *i.e.*, the EB-RRT* path planner, the direction controller and the bursting algorithm, are applied equally. The experimental results are shown in Fig. 16(a) and 16(b). As shown in Fig. 16(a), the proposed GA-LQR algorithm is applied and the Genetic Algorithm is used to select optimal values of the weighting matrices (*i.e.*, Q and R) to obtain the desired velocity. The average distance error is 44.01 μm, which is 22% of the swarm body length (*i.e.*, 200 μm). Meanwhile, the oscillation amplitude of the distance error is maintained in a relatively small range, *i.e.*, 0-136 μm. The experimental results using the LQR velocity controller is demonstrated in Fig. 16(b). In this case, Q and R are determined by experiments. The highest distance error increases to 268 μm and the average distances reaches 54.26 μm. The results reveal that the proposed Genetic Algorithm can ensure the high accuracy of tracking.

By comparing the tracking performance with two other velocity control algorithms, *i.e.*, PID and Gradient Descent based Linear Quadratic Regulator (GD-LQR) velocity control algorithm, the GA-LQR velocity control algorithm is testified. In this case, only the velocity control module is different whereas the other parts of the control scheme, *i.e.*, the EB-RRT* path planner, the direction controller and the bursting algorithm, are maintained equally. The GD-LQR velocity uses the position difference as the input and it relies on the Gradient Descent (GD) algorithm to find the optimal values of Q and R to obtain the desired velocity [42]. The distance errors using the two other velocity control algorithms are plotted in Fig. 16(c) and 16(d). Using the PID velocity controller leads to an average distance error of 57.66 μm. The highest distance error significantly increases to 294 μm, which is 2.16 times higher than that using GA-LQR algorithm (*i.e.*, 136 μm), as demonstrated in Fig. 16(c). When the GD-LQR is applied, as shown in Fig. 16(d), the highest distance error reaches 242 μm. In this case, the average distance error is 68.2 μm, and it is the highest among the three trails. The oscillation amplitudes of the distance error in Fig. 16(c) and 16(d) maintain in an unsatisfactory range and may exceed the highest distance error in Fig. 16(a), *i.e.*, 136μm. The results indicate that the high accuracy of tracking can be guaranteed if the GA-LQR velocity control algorithm is applied.

As a result, the EB-RRT* path planning algorithm, the proposed image-guided motion control algorithm consisting of the direction controller and the GA-LQR velocity controller,

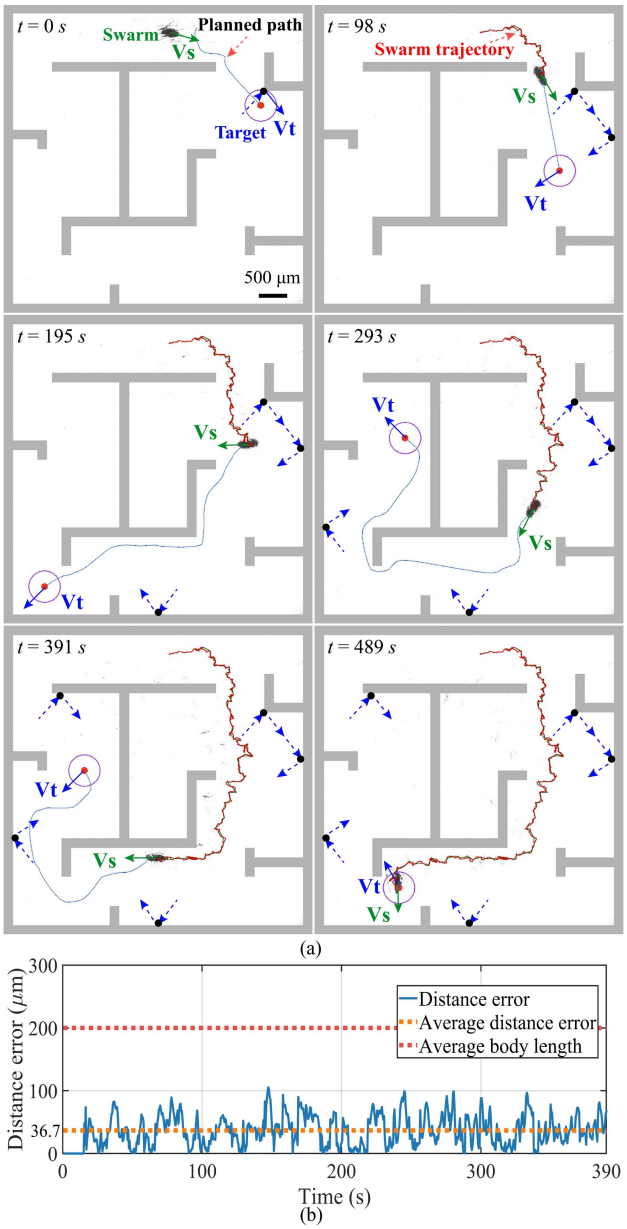


Fig. 17. The experimental results of tracking a high-speed mobile target in a micromaze. (a) The trajectory tracking results of the swarm. The velocities of the swarm and the target are represented by V_s and V_t, respectively. The blue curve is the dynamic planned path and the red curve shows the trajectory of the swarm. The dotted blue arrow pairs show the previous bounces and the black dots are the bouncing points. (b) The distance error of the target tracking. The average body length of the swarm is 200 μm. The scale bar is 500 μm.

and the bursting algorithm coordinate to realize the efficient path planning and the accurate mobile target tracking.

B. Mobile Target Tracking in a Micromaze

Compared with the environment with obstacles, the target will encounter a higher chance of bouncing in a highly confined environment, *e.g.*, a micromaze with virtual walls, leading to a more complicated trajectory of the target. Herein, the experimental results of a swarm tracking a high-speed target in a micromaze is shown in Fig. 17(a). The swarm can follow the path avoiding from the collision with the walls of the micromaze, and finally track the mobile target. The real-time and average distance error are plotted in Fig. 17(b). The average distance error is 36.7 μm, which is significantly lower than the body length of the swarm ($\approx$ 200 μm). The results indicate that the tracking control scheme with the bursting algorithm is effective for tracking a high-speed dynamic target in a micromaze with the high tracking accuracy.

VII. Conclusion

In this paper, we proposed an effective and efficient swarm control scheme for mobile target tracking. The control scheme combines an EB-RRT* path planning unit, an image-guided motion control unit and a targeted bursting process. An EB- RRT* based dynamic path planner is firstly designed to iteratively plan the real-time path avoiding obstacles in complex environments. The image-guided motion control algorithm including the direction control algorithm and the GA-LQR based velocity control algorithm is then proposed to control the moving direction and velocity of swarms, which can guarantee the small distance error during the tracking process. In order to meet the requirement of tracking high-speed mobile targets, a targeted bursting algorithm is implemented. Finally, the tracking control scheme is validated by simulation and experimental results to track low-speed and high-speed mobile targets in an environment with obstacles and a micromaze, respectively. This work provides a prototypical paradigm for other types of swarms to realize dynamic motion control tasks, and also new chances to better understand the field.

The proposed control scheme has a high compatibility with additional modules to meet requirement of complex conditions, *e.g.*, being actuated in a curved surface and encountering external disturbances in the environments. If the swarm is actuated in a curved surface, the velocity of the swarm may be affected. In this case, by tuning the pitch angle or adding a velocity adjustment module, the distance error can be maintained in satisfactory range. External disturbances, *e.g.*, adhesion, friction and fluid flows in the environment, may also have a significant impact on the moving direction and the velocity of the swarm. The proposed direction control algorithm is capable of correcting the angle of deviation and a disturbance observer can be employed for compensate the error of velocity. Moreover, if the swarm encounters physical obstacles, the stability and pattern can still be maintained when it is guided to approach the obstacles [25]. Therefore, the moving direction and the velocity of the swarm are controllable with the proposed control scheme. In addition, different medical imaging techniques, *e.g.*, Ultrasound Doppler Imaging, Fluorescence Imaging and Magnetic Resonance Imaging, could be potentially employed for *in vivo* detection of the swarm and the mobile target.

Acknowledgment

The authors would like to thank J. H. Law (University of Toronto) for providing technical support and fruitful discussions.

References

[1] J. Li, B. Esteban-Fernández de Ávila, W. Gao, L. Zhang, and J. Wang, "Micro/nanorobots for biomedicine: Delivery, surgery, sensing, and detoxification," *Sci. Robot.*, vol. 2, no. 4, Mar. 2017.

[2] B. J. Nelson, I. K. Kaliakatsos, and J. J. Abbott, "Microrobots for minimally invasive medicine," *Annu. Rev. Biomed. Eng.*, vol. 12, pp. 55–85, Aug. 2010.

[3] M. Sitti *et al.*, "Biomedical applications of untethered mobile milli/microrobots," *Proc. IEEE*, vol. 103, no. 2, pp. 205–224, Feb. 2015.

[4] H. Zhou, C. C. Mayorga-Martinez, S. Pané, L. Zhang, and M. Pumera, "Magnetically driven micro and nanorobots," *Chem. Rev.*, vol. 121, no. 8, pp. 4999–5041, Apr. 2021.

[5] Z. Wu, Y. Chen, D. Mukasa, O. S. Pak, and W. Gao, "Medical micro/nanorobots in complex media," *Chem. Soc. Rev.*, vol. 49, no. 22, pp. 8088–8112, 2020.

[6] B. Wang, K. Kostarelos, B. J. Nelson, and L. Zhang, "Trends in micro-/nanorobotics: Materials development, actuation, localization, and system integration for biomedical applications," *Adv. Mater.*, vol. 33, no. 4, 2021, Art. no. 2002047.

[7] C. Hu, S. Pané, and B. J. Nelson, "Soft micro-and nanorobotics," *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 1, pp. 53–75, May 2018.

[8] Y. Hu, "Self-assembly of DNA molecules: Towards DNA nanorobots for biomedical applications," *Cyborg Bionic Syst.*, vol. 2021, pp. 1–3, Oct. 2021.

[9] H. Wang, J. Kan, X. Zhang, C. Gu, and Z. Yang, "Pt/CNT micro-nanorobots driven by glucose catalytic decomposition," *Cyborg Bionic Syst.*, vol. 2021, pp. 1–8, Aug. 2021.

[10] D. Jin, J. Yu, K. Yuan, and L. Zhang, "Mimicking the structure and function of ant bridges in a reconfigurable microswarm for electronic applications," *ACS Nano*, vol. 13, no. 5, pp. 5999–6007, May 2019.

[11] K. Villa, L. Krejčová, F. Novotný, Z. Heger, Z. Sofer, and M. Pumera, "Cooperative multifunctional self-propelled paramagnetic microrobots with chemical handles for cell manipulation and drug delivery," *Adv. Funct. Mater.*, vol. 28, no. 43, Oct. 2018, Art. no. 1804343.

[12] T. O. Tasci, P. S. Herson, K. B. Neeves, and D. W. M. Marr, "Surface-enabled propulsion and control of colloidal microwheels," *Nature Commun.*, vol. 7, no. 1, pp. 1–6, Apr. 2016.

[13] X. Wang *et al.*, "3D printed enzymatically biodegradable soft helical microswimmers," *Adv. Funct. Mater.*, vol. 28, no. 45, Nov. 2018, Art. no. 1804107.

[14] M.-J. Tang *et al.*, "Controllable microfluidic fabrication of magnetic hybrid microswimmers with hollow helical structures," *Ind. Eng. Chem. Res.*, vol. 57, no. 29, pp. 9430–9438, Jul. 2018.

[15] J. Li *et al.*, "Biomimetic platelet-camouflaged nanorobots for binding and isolation of biological threats," *Adv. Mater.*, vol. 30, no. 2, 2018, Art. no. 1704800.

[16] H. Xu, M. Medina-Sánchez, V. Magdanz, L. Schwarz, F. Hebenstreit, and O. G. Schmidt, "Sperm-hybrid micromotor for targeted drug delivery," *ACS Nano*, vol. 12, no. 1, pp. 327–337, Jan. 2018.

[17] V. Magdanz *et al.*, "IRONSperm: Sperm-templated soft magnetic microrobots," *Sci. Adv.*, vol. 6, no. 28, Jul. 2020, Art. no. eaba5855.

[18] Y. Zhang *et al.*, "Real-time tracking of fluorescent magnetic spore–based microrobots for remote detection of *C. diff* toxins," *Sci. Adv.*, vol. 5, no. 1, 2019, Art. no. eaau9650.

[19] X. Yang *et al.*, "An agglutinate magnetic spray transforms inanimate objects into millirobots for biomedical applications," *Sci. Robot.*, vol. 5, no. 48, Nov. 2020, Art. no. eabc8191.

[20] S. Palagi *et al.*, "Structured light enables biomimetic swimming and versatile locomotion of photoresponsive soft microrobots," *Nature Mater.*, vol. 15, no. 6, pp. 647–653, 2016.

[21] S. Yu *et al.*, "Trimer-like microrobots with multimodal locomotion and reconfigurable capabilities," *Mater. Today Adv.*, vol. 14, Jun. 2022, Art. no. 100231.
[22] Z. Wu *et al.*, "A swarm of slippery micropropellers penetrates the vitreous body of the eye," *Sci. Adv.*, vol. 4, no. 11, Nov. 2018, Art. no. eaat4388.
[23] A. Servant, F. Qiu, M. Mazza, K. Kostarelos, and B. J. Nelson, "Controlled *in vivo* swimming of a swarm of bacteria-like microrobotic flagella," *Adv. Mater.*, vol. 27, no. 19, pp. 2981–2988, 2015.
[24] J. Yu, L. Yang, and L. Zhang, "Pattern generation and motion control of a vortex-like paramagnetic nanoparticle swarm," *Int. J. Robot. Res.*, vol. 37, no. 8, pp. 912–930, Jul. 2018.
[25] J. Yu, B. Wang, X. Du, Q. Wang, and L. Zhang, "Ultra-extensible ribbon-like magnetic microswarm," *Nature Commun.*, vol. 9, no. 1, pp. 1–9, Dec. 2018.
[26] J. Yu, L. Yang, X. Du, H. Chen, T. Xu, and L. Zhang, "Adaptive pattern and motion control of magnetic microrobotic swarms," *IEEE Trans. Robot.*, vol. 38, no. 3, pp. 1552–1570, Jun. 2022.
[27] F. Ji, D. Jin, B. Wang, and L. Zhang, "Light-driven hovering of a magnetic microswarm in fluid," *ACS Nano*, vol. 14, no. 6, pp. 6990–6998, Jun. 2020.
[28] H. Kim, U. K. Cheang, L. W. Rogowski, and M. J. Kim, "Motion planning of particle based microrobots for static obstacle avoidance," *J. Micro-Bio Robot.*, vol. 14, nos. 1–2, pp. 41–49, Jun. 2018.
[29] J. Liu, T. Xu, S. X. Yang, and X. Wu, "Navigation and visual feedback control for magnetically driven helical miniature swimmers," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 477–487, Jan. 2020.
[30] L. Huang, L. Rogowski, M. J. Kim, and A. T. Becker, "Path planning and aggregation for a microrobot swarm in vascular networks using a global input," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 414–420.
[31] L. Yang, Y. Zhang, Q. Wang, K.-F. Chan, and L. Zhang, "Automated control of magnetic spore-based microrobot using fluorescence imaging for targeted delivery with cellular resolution," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 1, pp. 490–501, Sep. 2019.
[32] K. Belharet, D. Folio, and A. Ferreira, "Endovascular navigation of a ferromagnetic microrobot using MRI-based predictive control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 2804–2809.
[33] L. Yang, Q. Wang, and L. Zhang, "Model-free trajectory tracking control of two-particle magnetic microrobot," *IEEE Trans. Nanotechnol.*, vol. 17, no. 4, pp. 697–700, Jul. 2018.
[34] X. Du, J. Yu, D. Jin, P. W. Y. Chiu, and L. Zhang, "Independent pattern formation of nanorod and nanoparticle swarms under an oscillating field," *ACS Nano*, vol. 15, no. 3, pp. 4429–4439, Mar. 2021.
[35] Q. Zou, Y. Wang, and J. Yu, "Colloidal microrobotic swarms," in *Field-Driven Micro and Nanorobots for Biology and Medicine*. Cham, Switzerland: Springer, 2022, pp. 179–209.
[36] L. Yang, J. Yu, and L. Zhang, "Statistics-based automated control for a swarm of paramagnetic nanoparticles in 2-D space," *IEEE Trans. Robot.*, vol. 36, no. 1, pp. 254–270, Feb. 2020.
[37] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.
[38] M. Elbanhawi, M. Simic, and R. N. Jazar, "Continuous path smoothing for car-like robots using B-spline curves," *J. Intell. Robotic Syst.*, vol. 80, no. 1, pp. 23–56, 2015.
[39] I. Noreen, "Collision free smooth path for mobile robots in cluttered environment using an economical clamped cubic B-spline," *Symmetry*, vol. 12, no. 9, p. 1567, Sep. 2020.
[40] R. M. Murray *et al.*, "Optimization-based control," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. DRAFT v2.0b, 2009, pp. 111–128.
[41] S. Sivanandam and S. Deepa, "Genetic algorithms," in *Introduction to Genetic Algorithms*. Heidelberg, Germany: Springer, 2008, pp. 15–37.
[42] E. Todorov, "Optimal control theory," in *Bayesian Brain: Probabilistic Approaches to Neural Coding*. Cambridge, MA, USA: MIT Press, 2006, pp. 268–298.

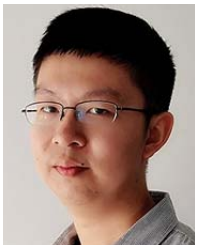
Qian Zou (Student Member, IEEE) received the B.Eng. degree in mechanical engineering from Hainan University, Haikou, China, and the joint M.Sc. degree in sensor and imaging system from the University of Glasgow, Glasgow, U.K., and The University of Edinburgh, Edinburgh, U.K. He is currently pursuing the Ph.D. degree with the Intelligent Microrobotics Laboratory, The Chinese University of Hong Kong, Shenzhen.

His research interests include path planning and tracking control of magnetic microrobots.

Xingzhou Du (Member, IEEE) received the B.Eng. degree in automation science and technology from Xi'an Jiaotong University, Xi'an, China, in 2017, and the Ph.D. degree in biomedical engineering from The Chinese University of Hong Kong (CUHK), Hong Kong, in 2021.

He is currently a Research Associate Scientist with the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), Shenzhen, China. His research interests include the development of electromagnetic systems and the control methods for magnetic microrobots.

Dr. Du was a recipient of the Toshio Fukuda Best Paper Award in Mechatronics at the 2020 IEEE International Conference on Advanced Robotics and Mechatronics (ARM 2020).



Yuezhen Liu (Student Member, IEEE) received the B.Eng. degree in automation from the Harbin Institute of Technology, Shenzhen, China, in 2021. He is currently pursuing the Ph.D. degree in computer and information engineering with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen.

His research interests include vision-based intelligent control and biomedical applications of microrobots.

Hui Chen (Graduate Student Member, IEEE) received the B.Eng. and M.S. degrees in ocean engineering from Harbin Engineering University, Harbin, China, in 2018 and 2021, respectively. He is currently pursuing the Ph.D. degree with the Department of Computer and Information Engineering, The Chinese University of Hong Kong, Shenzhen.

His research interests include intelligent control and biomedical applications of micro-nanorobotic swarms.

Yibin Wang (Student Member, IEEE) received the B.Eng. degree in materials processing from the Harbin Institute of Technology, Weihai, China, in 2018, and the M.Sc. degree in materials science from Carnegie Mellon University, Pittsburgh, PA, USA, in 2019. He is currently pursuing the Ph.D. degree with The Chinese University of Hong Kong, Shenzhen.

His research interests include magnetically controlled soft robotics.

Jiangfan Yu (Member, IEEE) received the B.Eng. degree in automation from the Harbin Institute of Technology, Harbin, China, in 2014, and the Ph.D. degree in mechanical and automation engineering from The Chinese University of Hong Kong (CUHK), Hong Kong, in 2018.

He worked as a Post-Doctoral Fellow with CUHK from 2018 to 2019, and the University of Torontos from 2019 to 2020. He is currently an Assistant Professor with the School of Science and Engineering, CUHK, Shenzhen. He is also the Deputy Director of the Micro-Nanorobotics Center, Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS). His research interests include micro-nanorobotics and medical robotics.

Dr. Yu was awarded with the IEEE/ASME TRANSACTIONS ON MECHATRONICS Best Paper Finalist in 2020 and *Nature Communications* Top 50 Physics Articles in 2018. He is an Associate Editor of the IEEE ROBOTICS AND AUTOMATION LETTERS, and a Guest Editor of *Micromachines*, *Frontiers in Robotics and AI*, and *Journal of Healthcare Engineering*.