MagTile: A Modular Platform for Controlling Electromagnetically Actuated Agents for Ethorobotics Applications

Akhil Sankar, Cy Westbrook, Mohammad Tuqan, Guosong Yang, *Member, IEEE*, and Daniel Burbano, *Senior Member, IEEE*

***Abstract*—Ethorobotics, the study of animal–robot interactions, has advanced behavioral research by enabling controlled, repeatable experiments through the use of biomimetic robotic agents. Existing platforms, however, often lack flexibility, scalability, and computational tools for in-silico experimentation, limiting their easy deployment across diverse experimental configurations. Here, we introduce MagTile, a modular and scalable platform for controlling ensembles of electromagnetically actuated agents. Our design utilizes standard components and 3-D-printed parts, enabling multiagent actuation without the need for complex onboard electronics. We formulate a mathematical model of MagTile and analyze its convergence properties using Lyapunov stability theory. The model serves as the foundation for establishing a computational framework, facilitating simulation, testing, and optimization of experimental conditions prior to physical deployment, thereby improving efficiency and reducing trial-and-error experimentation. The effectiveness of our platform was validated through a series of illustrative experiments showcasing multiagent coordination, vision-based collision avoidance, and collective behavior using biomimetic fish.**

***Index Terms*—Electromagnetic actuation, ethorobotics, feedback control, multiagent systems (MASs).**

I. Introduction

MULTIAGENT systems (MASs) consist of interacting agents that cooperate to achieve shared objectives through emergent behaviors [1], [2], [3]. They have been applied across a wide range of engineering domains, including cooperative robotics [4], distributed sensing [5], and intelligent transportation systems [6]. Despite their broad applicability, MAS face critical challenges across control, communication, and actuation layers [7]. Designing scalable, decentralized control strategies remains a fundamental difficulty, particularly in the absence of a central coordinator. Common tasks such as multiagent path planning and coordination require efficient distributed solutions, and recent advances have leveraged decentralized control and reinforcement learning to address these problems [8]. In addition, limited bandwidth and unreliable communication demand robust distributed protocols, while mapping high-level coordination goals to physical motion is especially challenging in constrained environments. These challenges are further compounded when MAS are deployed in the physical world, where sensing noise, actuation delays, and dynamic environments introduce additional complexity.

Nature offers a rich source of inspiration for devising solutions that overcome the challenges faced in engineered MASs. Complex collective behaviors observed in animal groups, such as fish schools, bird flocks, and insect swarms, demonstrate scalable, robust coordination without centralized control [9]. These biological systems have inspired a range of bio-inspired control algorithms for MAS, with applications spanning autonomous robotics to distributed sensing, network optimization, and behavioral research [10], [11], [12]. While various MAS platforms have focused on studying collective behavior with robotic swarms [13], [14], [15], [16], [17], a specialized class of systems has been designed specifically to investigate animal–robot interactions, establishing the foundation of the emerging field of ethorobotics.

Ethorobotics is an interdisciplinary area integrating ethology and robotics to study animal behavior via biomimetic robotic agents [18], [19], [20], [21]. These systems enable controlled, repeatable interactions that can help uncover the principles of animal locomotor control giving rise to complex behavior such as collective decision-making, social coordination, and behavioral dynamics [19], [22], [23], [24]. Specifically, by systematically engineering social interactions, we can probe how animal collectives integrate multiple sensory modalities (vision, lateral-line cues, and olfaction) to navigate complex environments. This is particularly important to help address MAS challenges, as in animal groups, information is exchanged through passive channels (motion signatures, spatial configurations, and medium-mediated disturbances) rather than wireless communications. Quantifying these feedback pathways can help understand how local perception-response rules scale into robust, decentralized control strategies.

To translate biological insights into testable mechanisms, we require experimental platforms that provide control, repeatability, and measurable perturbations of the interaction rules. Unlike traditional observation-based methods, robotic agents can be precisely engineered to replicate specific behavioral

Received 14 July 2025; revised 20 October 2025 and 11 January 2026; accepted 28 January 2026. This work was supported by the National Science Foundation under Grant CMMI 2443301. Recommended by Associate Editor S. Santini. *(Corresponding author: Daniel Burbano.)*

The authors are with the Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ 08854 USA (e-mail: asankar@scarletmail.rutgers.edu; cww45@scarletmail.rutgers.edu; mt1298@scarletmail.rutgers.edu; guosong.yang@rutgers.edu; daniel.burbano@rutgers.edu).

Digital Object Identifier 10.1109/TCST.2026.3662320

patterns and morphologies, allowing researchers to systematically isolate and analyze the effect of variables of interest. For instance, Bonnet et al. [25] developed an interspecies biohybrid system where robotic agents facilitated information transfer between honeybees and zebrafish despite being in separate environments. The platform utilized a wheeled mobile robot that controlled a fish lure via magnetic coupling. Honeybees, housed in a silicon oil-coated plexiglass arena, interacted with two stationary bee robots, which provided a binary aggregation choice. Both experimental setups were computer-controlled in a closed-loop system to enable remote interactions. Wheeled robots have also been used to study social behavior and empathy in rodents. Quinn et al. [26] demonstrated that rats could distinguish between a socially responsive robot and an asocial, randomly moving one, preferentially releasing the former from a restrainer. These findings support the growing use of robotic agents in rodent research on social and emotional behavior [27], [28]. Swain et al. [29] proposed a cyber-physical robotic system that utilized real-time feedback control to study fish schooling behavior. A magnetically coupled fish replica, actuated by a wheeled robot beneath the tank, received video-based position feedback from an overhead camera, allowing it to respond to live fish movements. Similar platforms, each utilizing a single robotic agent, have been developed to investigate the role of biomimicry in eliciting social behavior. Landgraf et al. [30] examined the degree of realistic eye features necessary to trigger social responses in guppies. In contrast, more recent work utilized cartesian robotic systems positioned beneath a tank to control a fish replica in order to study how fish size and morphology affect social interaction [31]. This platform has also been used to explore closed-loop strategies informed by mathematical models of fish [32]. Moreover, recent approaches use flexible magnetic skeletons to enable locomotion of a single agent via external magnetic fields [33].

While existing platforms provide a promising approach for studying animal-robot interactions, they still suffer from scalability and flexibility, both of which are critical for accommodating a wide range of experimental designs. In fact, existing solutions based on Cartesian robotic systems cannot be scaled beyond a single agent due to the mechanical constraints of the 2-D gantry structure [31]. While approaches based on magnetic skeletons [33] require animals to swim within strong magnetic fields, which can interfere with their sensory-based navigation and compromise the validity of behavioral observations [34]. In addition, these systems also lack scalability, as they are limited to controlling a single individual. Finally, wheeled robot-based systems can, in principle, control multiple agents; however, they typically involve numerous mechanical components, onboard sensors, and battery systems that require frequent maintenance and recharging. These factors increase system complexity and reduce modularity compared to fixed actuation platforms. Moreover, mobile robots require sufficient space to move freely, which becomes a significant constraint when conducting experiments with live animals such as fish, as the experimental setup must maintain a safe distance between the observation arenas and the mobile robots. To advance ethorobotics research, several technical challenges must be addressed toward developing flexible and scalable platforms.

First, experiments involving live animals require adaptable setups that vary based on the size and shape of the experimental arenas. For example, in fish behavioral studies, different experimental paradigms necessitate specific tank configurations: rectangular (or circular) tanks are often used for novel environment tests [35], while T-shaped tanks are employed for T-maze tasks to assess spatial learning and decision-making [36]. A versatile platform should be capable of adapting to diverse experimental environments without requiring extensive modifications. Second, existing ethorobotics platforms are typically limited to controlling a single robotic agent. They rely on cartesian robots that can only actuate a single agent [31], [32] or would require complex mobile robotic designs [29], making multiagent control challenging. Developing a scalable and flexible platform capable of simultaneously actuating multiple agents would significantly expand the scope of collective behavior studies and interactive multiagent experiments. Third, current platforms lack computational tools for in-silico experimentation, preventing researchers from simulating and optimizing experimental conditions before physical deployment. While physical testing without live animals can help verify platform functionality, it is often time-consuming and impractical for extensive optimization. Computational modeling provides an efficient alternative, allowing us to systematically test and refine control strategies, explore a wide range of conditions, and preemptively identify potential issues. Notably, this approach aligns with the principles of the three Rs (refine, reduce, replace) [37] by minimizing unnecessary physical trials and ensuring that live animal experiments are conducted when the platform is fully validated only.

In this work, we seek to address these technical challenges by introducing a simple yet effective architecture for controlling ensembles of electromagnetically actuated payloads or agents through a scalable tile-based system (here, the terms payload and agent will be used interchangeably). Each tile contains solenoids that generate magnetic forces capable of moving payloads and are designed for flexible interconnection, allowing the tiles to adapt to different surface areas. Our approach circumvents the need for complex onboard control systems or sophisticated mechanical components by leveraging simple mechanical designs and 3-D-printed parts. Our platform enables the concurrent control of multiple agents, offering novel possibilities for multiagent experiments and complementing existing behavioral paradigms based on cartesian and mobile robots [29], [31], [32]. In this article, scalability refers to the ability to increase the number of simultaneously controlled agents and the total actuation workspace area supported by the platform. Flexibility refers to the ability to reconfigure the workspace geometry and surface area by rearranging the modular tile layout.

The contributions of this work are threefold.

1) We introduce a modular architecture enabling the control of multiple agents. Unlike existing platforms that require customized mechanical designs for each agent, our solution relies on standardized off-the-shelf components

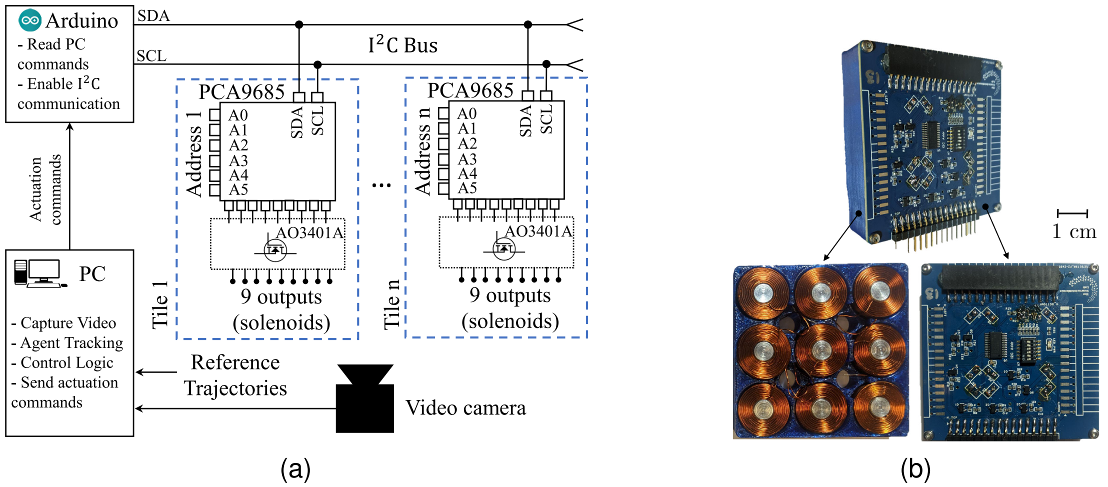


Fig. 1. Illustration of the MagTile design. (a) Overview of the system architecture. (b) Picture of a single tile consisting of nine solenoids and a PCB control board with pinouts to facilitate tile stacking. The enclosure is housed in a 3-D-printed case.

and 3-D-printed parts. This could reduce implementation costs while enabling scalability for large ensembles of agents. In addition, the system design is publicly available, reproducible, and easily customizable for various experimental needs.

2) We develop a mathematical model of the robotic platform by representing the interaction between the solenoid and a disk magnet payload using pairs of fictitious magnetic monopoles with equal magnitude but opposite polarity. This formulation resulted in a nonlinear state-space model, which we analyzed using Lyapunov stability theory [38] to assess the system behavior and performance as a function of physical parameters.

3) We establish a computational framework for simulating, testing, and optimizing experimental conditions prior to physical deployment. This framework is built upon the mathematical model of the platform and integrates a feedback control strategy to guarantee agents track reference trajectories in real-time. In our formulation, the operational space of the robot is represented as a graph, where vertices denote the locations of the solenoids and edges correspond to their nearest neighbors, forming a grid-like topology. The feedback control algorithm is based on Dijkstra's algorithm, which provides a correction path toward the reference trajectory after a perturbation has occurred. Feedback is implemented through vision-based tracking, where an overhead camera continuously monitors agent positions using computer vision algorithms.

The rest of the article is organized as follows. Section II describes the platform design and overall system architecture. Section III presents the mathematical modeling and analysis. Section IV details the computational implementation and feedback control strategy. Finally, Section V validates the effectiveness of the proposed platform through a series of representative experiments.

II. System Architecture and Design

The MagTile platform is composed of: 1) a modular actuation system; 2) multiagent tracking for vision-based feedback; and 3) computer-based control, as depicted in Fig. 1. Central to this architecture is the electromagnetic actuation system, where solenoids generate magnetic forces to manipulate payloads. The modular design enables flexible interconnection, allowing easy adaptation to different surface areas. An I^{2}C communication protocol was utilized for real-time coordination between tiles through an Arduino host controller. Our design enables concurrent control of each tile, allowing for precise manipulation of agents. Here, we consider the agents to be a group of m disk magnets as they can easily be utilized to actuate biomimetic replicas [25], [29], [30], [31]. Each agent's position is defined in a 2-D coordinate space as $(x_i(t), y_i(t))$ for $t \geq 0$, where $i \in \{1, 2, \ldots, m\}$. These positions are sampled at discrete time intervals T (s), producing a sequence of measurements $(x_i(kT), y_i(kT))$ for $k \in \{0, 1, 2, \ldots, \}$, obtained via the multiagent tracking algorithm. Finally, the feedback controller guarantees the m agents track their reference trajectories, $(x_i^r(kT), y_i^r(kT))$, in real-time, compensating for potential disturbances and avoiding collisions with other agents.

A. Tile Design

Each tile is composed of a 3-D-printed case designed to hold nine solenoids arranged in a 3 × 3 array. The pinouts are designed to support modular assembly, allowing tiles to be daisy-chained in both vertical and horizontal configurations, provided they maintain a uniform orientation. The actuation control is managed by the PCA9685 I^{2}C driver from NXP Semiconductor. This IC provides 16 synchronized 12-bit pulse with modulation (PWM) outputs, with a variable frequency range from 24 to 1526 Hz, and an I^{2}C interface supporting up to 62 unique addresses, configurable via six address bits. The first nine PWM output channels are routed to N-channel MOSFETs, which serve as low-side switches for actuating the nine solenoids mounted on the tile. The tenth output channel is allocated to an upward-facing LED, functioning as a visual diagnostic indicator for testing and debugging. Each tile's I^{2}C address is physically assigned via a six-position DIP switch integrated into the PCB. Additionally, the printed circuit board (PCB) incorporates parallel I/O headers for system integration.

B. Multiagent Tracking

A vision-based tracking system is utilized to obtain the agent's centroid coordinates using an overhead camera. Each agent is uniquely identified by a color. The tracking pipeline is implemented in Python 3, utilizing OpenCV for video acquisition and image processing, NumPy for data manipulation, and Redis for real-time inter-process communication [39], [40]. The video stream is cropped to a predefined region of interest prior to tracking. To identify agents, color segmentation is first applied using predefined upper and lower bounds in the RGB color space. Pixels that fall within these bounds are isolated into binary masks, where white regions correspond to detected agents and black regions represent the background. Due to the sensitivity of certain colors (i.e., red) to lighting variations, detection can instead be performed in the hue, saturation, value (HSV) color space for improved robustness. Following color-based segmentation, contours corresponding to white regions in the binary mask are extracted. To eliminate noise and irrelevant features, a surface-area filter is applied: contours with areas outside predefined bounds are discarded. The remaining contours are treated as valid agent detections. For each detected contour, a bounding box is computed, and its geometric centroid is extracted. These pixel-based centroids are then mapped to real-world Cartesian coordinates using a calibrated pixel-to-centimeter scaling factor.

C. Feedback Control Design

The main goal of the feedback controller is to actuate magnetic payloads and move them along predefined reference trajectories while ensuring collision avoidance and robustness to external perturbations. Specifically, each reference trajectory $(x_i^r(kT), y_i^r(kT))$ is defined as a sequence of solenoid positions that define the intended path for each agent. We model the MagTile workspace as an undirected graph $\mathscr{G} := (\mathcal{S}, \mathcal{E})$, where the vertices $\mathcal{S}$ correspond to solenoids, and the edges $\mathcal{E}$ define the possible transitions between neighboring solenoids. Each node in the graph is connected to its eight nearest neighbors (four cardinal and four diagonal directions), forming a structured grid topology. The edge weights correspond to the Euclidean distance defined by d_{pq} between connected nodes $(p, q) \in \mathcal{E}$, with diagonal movements assigned higher weights than cardinal movements. In particular, the topology of the graph $\mathscr{G}$ is represented by the weighted adjacency matrix $A = [a_{pq}]$, defined as follows [41]:

$$a_{pq} := \begin{cases} d_{pq}, & \text{if } (p,q) \in \mathcal{E} \\ \infty, & \text{otherwise.} \end{cases} \tag{1}$$

To ensure that each agent tracks its reference trajectory $X_i^{\text{ref}}(kT) := (x_i^r(kT), y_i^r(kT))$, the feedback controller operates in a mode-switching framework with three distinct control actions.

1) *Trajectory Tracking:* Following the reference trajectory $X_i^{\text{ref}}(kT)$.
2) *Path Planning:* Compute and follow an intermediate path $X_i^{\text{plan}}(kT)$ when direct tracking is infeasible.
3) *Collision Avoidance:* Adjusting movement to avoid obstacles following $X_i^{\text{obs}}(kT)$.

In the trajectory tracking mode, agents follow a predefined reference path composed of a sequence of solenoids. To do so, agents must remain within the activation region of each solenoid along the path. This activation region is defined as a circular neighborhood of radius r_0, within which the actuation force overcomes friction to move the agent. If an agent deviates from the activation region and no imminent collisions with other agents are detected, a path planning routine is invoked to guide the agent back to its reference trajectory. Finally, to prevent agent interference, a collision avoidance region of radius r_c is defined around each agent. We set $r_\text{c} = r_0$, so that actuation and interference constraints operate on the same spatial scale. When another agent enters this region, the controller switches to a collision avoidance mode. Therefore, the overall control switching logic is given by

$$\begin{cases} X_i^{\text{ref}}(kT), & \text{if } I_i(kT) = 0 \wedge e_i(kT) \le r_0 \\ X_i^{\text{plan}}(kT), & \text{if } I_i(kT) = 0 \wedge e_i(kT) > r_0 \\ X_i^{\text{obs}}(kT), & \text{if } I_i(kT) = 1 \end{cases} \tag{2}$$

where $e_i(kT)$ denotes the instantaneous tracking error

$$e_i(kT) := \sqrt{\left(x_i(kT) - x_i^r(kT)\right)^2 + \left(y_i(kT) - y_i^r(kT)\right)^2}$$

and $I_i(kT) \in \{0, 1\}$ is a binary indicator function that activates obstacle avoidance when Agent i enters the collision avoidance region of any other agent (i.e. $I_i(kT) = 1$). The symbol $\wedge$ denotes the logical AND operation. The trajectories defined by $X_i^{\text{obs}}(kT)$ can be generated using any obstacle avoidance algorithm. In our experiments, we implemented a halting strategy and also the method from [42]. Similarly, the trajectories for $X_i^{\text{plan}}(kT)$ can be determined using any path planning algorithm [43], [44], [45]. In this work, we selected Dijkstra's algorithm [46], which computes the shortest path between two points in the graph $\mathscr{G}$. Notably, recent mathematical efforts have established that Dijkstra's algorithm is universally optimal in terms of execution efficiency, as no alternative algorithm can outperform it when implemented with an appropriately optimized heap [47].

III. Mathematical Modeling and Analysis

A. Model Derivation

1) Dipole Model: The solenoid and the disk magnet payload can be seen as pairs of fictitious magnetic monopoles with equal magnitudes but opposite polarity [48]. Specifically, for the solenoid, we consider two magnetic monopoles $\pm g_s$ separated by a small distance represented by the displacement vector $\boldsymbol{\ell}_s \in \mathbb{R}^3$. Similarly, for the disk magnet, we have magnetic monopoles $\pm g_d$ separated by a small distance $\boldsymbol{\ell}_d \in \mathbb{R}^3$. Throughout the article, the subscripts "s" and "d" denote quantities related to the solenoid and disk magnet, respectively. The displacement vectors $\boldsymbol{\ell}_s$ and $\boldsymbol{\ell}_d$ point from the minus to the plus magnetic charge. This configuration results in the dipole moments $\mathbf{m}_s := g_s\boldsymbol{\ell}_s$ and $\mathbf{m}_d := g_d\boldsymbol{\ell}_d$.

Consider the solenoid and the payload centroids to be located at coordinates $\mathbf{x}_s := [x_s, y_s, z_s]^\top \in \mathbb{R}^3$ and $\mathbf{x}_d := [x_d, y_d, z_d]^\top \in \mathbb{R}^3$, respectively as shown in Fig. 2(a) and (b). Let $\hat{\mathbf{r}}(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^3 := (\mathbf{x} - \mathbf{y})/\|\mathbf{x} - \mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$ be the

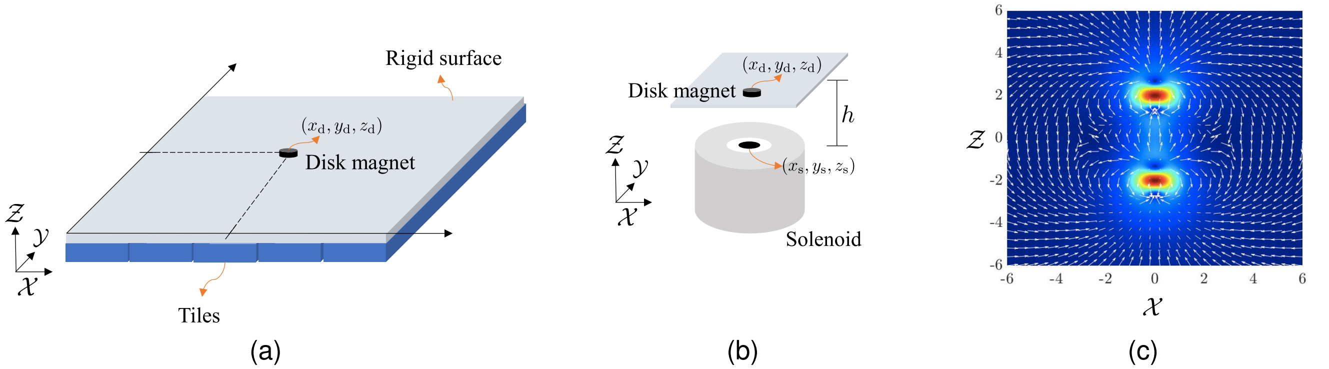


Fig. 2. Mathematical modeling of MagTile. (a) Magtile surface and disk magnet, with the magnet's centroid described in the inertial reference frame $\mathcal{X}-\mathcal{Y}-\mathcal{Z}$. (b) Schematic of a single solenoid, highlighting the separation between the rigid surface and the solenoid. (c) Side view of the magnetic field approximation based on the dipole model. White arrows represent the vector field direction, with red and blue indicating regions of high- and low-field intensity, respectively.

normalized displacement vector, then the magnetic flux density of the solenoid and disk at any point in space $\mathbf{x} \in \mathbb{R}^3$ can be approximated as [48], [49]

$$\mathbf{B} = \frac{\mu_0}{4\pi}\left(\mathbf{B}_0\left(\mathbf{m}_s, \mathbf{x}, \mathbf{x}_d\right) + \mathbf{B}_0\left(\mathbf{m}_d, \mathbf{x}, \mathbf{x}_s\right)\right)$$

where

$$\mathbf{B}_0(\mathbf{m}, \mathbf{x}, \mathbf{y}) := \left(\frac{3(\mathbf{m} \cdot \hat{\mathbf{r}}(\mathbf{x}, \mathbf{y}))\, \hat{\mathbf{r}}(\mathbf{x}, \mathbf{y})) - \mathbf{m}}{\|\mathbf{x} - \mathbf{y}\|^3}\right)$$

and μ_0 is the vacuum permeability constant with unit (N·A^{-2}). This equation provides a far-field approximation of the magnetic field, as illustrated in Fig. 2(c).

2) Potential Energy of the Solenoid-Disk System: Given that the disk is always separated from the solenoid by a plate at a constant distance along the $\mathcal{Z}$-axis, the potential energy of the two dipoles with moments $\mathbf{m_s}$ and $\mathbf{m_d}$ is given by [50]

$$U = \frac{3\mu_0\left(\mathbf{m}_s \cdot \mathbf{m}_d - \left(\mathbf{m}_s \cdot \hat{\mathbf{r}}\left(\mathbf{x}_s, \mathbf{x}_d\right)\right)\left(\mathbf{m}_d \cdot \hat{\mathbf{r}}\left(\mathbf{x}_s, \mathbf{x}_d\right)\right)\right)}{4\pi\|\mathbf{x_s} - \mathbf{x_d}\|^3}. \tag{3}$$

Because the magnetic moment is primarily aligned about the $\mathcal{Z}$-axis, we have $\mathbf{m}_s = [0, 0, \alpha_s]^\top$ and $\mathbf{m}_d = [0, 0, \alpha_d]^\top$, with α_s and α_d being two positive constant parameters representing the magnitude of each moment, respectively. Substituting these vectors into (3) and after some algebraic manipulations yields

$$U = -\frac{3\mu_0\alpha_s\alpha_d}{4\pi r^5}\left(-\left(x_s - x_d\right)^2 - \left(y_s - y_d\right)^2\right)$$

where $r := \|\mathbf{x}_s - \mathbf{x}_d\|$.

3) State-Space Model: The disk magnet is a rigid body whose centroid (x_d, y_d) movements are constrained to a plane and can be described in a global inertial frame $\mathcal{X} - \mathcal{Y}$ [51]. Assuming all n_s solenoids are identical, we propose to describe the time evolution of the centroid dynamics of the agent by solving the following pair of second-order differential equations:

$$m_d\ddot{x}_d(t) + d_d\dot{x}_d(t) = \sum_{p=1}^{n_s} F_x\left(x_d, y_d, x_p, y_p\right)\delta_i(t) \tag{4a}$$

$$m_d\ddot{y}_d(t) + d_d\dot{y}_d(t) = \sum_{p=1}^{n_s} F_y\left(x_d, y_d, x_p, y_p\right)\delta_i(t). \tag{4b}$$

Here, d_d represents a damping coefficient due to dry friction between the disk and the surface, while m_d is the disk mass. The terms $F_x(x_d, y_d, x_p, y_p)$ and $F_y(x_d, y_d, x_p, y_p)$ represent the magnetic forces in the $\mathcal{X}$ and $\mathcal{Y}$ directions, respectively, due to the pth solenoid centered at (x_p, y_p). The $\delta_p(t)$ function is an activation term that captures the region of attraction of the solenoid and is given by

$$\delta_p(t) = \begin{cases} 1, & \text{if } r_p(t) < r_0 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $r_p(t) := \sqrt{(x_d(t) - x_p)^2 + (y_d(t) - y_p)^2 + h^2}$ and $h := z_d - z_p$ for all $p \in \{1, \ldots, n_s\}$.

The magnetic force is given by $[F_x, F_y] = -\nabla U$ [49], [50], where U is the potential energy function given in (3). Given that h is maintained constant, we can establish

$$F_x = \frac{3\mu_0\alpha_s\alpha_d}{4\pi r_p^5(t)}\left(1 - \frac{5h^2}{r_p^2(t)}\right)\left(x_d(t) - x_p\right)$$

$$F_y = \frac{3\mu_0\alpha_s\alpha_d}{4\pi r_p^5(t)}\left(1 - \frac{5h^2}{r_p^2(t)}\right)\left(y_d(t) - y_p\right).$$

Next, letting $v_x(t) = \dot{x}_d(t)$ and $v_y(t) = \dot{y}_d(t)$, equations in (4) can be rewritten in state-space form as

$$\dot{x}_d(t) = v_x(t) \tag{6a}$$

$$\dot{y}_d(t) = v_y(t) \tag{6b}$$

$$\dot{v}_x(t) = -\beta v_x(t) + \alpha\sum_{p=1}^{n_s} G\left(r_p(t)\right)\frac{x_d(t) - x_p}{r_p^5(t)}\delta_p(t) \tag{6c}$$

$$\dot{v}_y(t) = -\beta v_y(t) + \alpha\sum_{p=1}^{n_s} G\left(r_p(t)\right)\frac{y_d(t) - y_p}{r_p^5(t)}\delta_p(t) \tag{6d}$$

where $\alpha := (3\mu_0 a_s a_d)/(4\pi m_d)$ is a positive constant representing magnetic coupling strength, $\beta := d_d/m_d$ is a rescaled damping coefficient, and $G(r_p(t)) := 1 - 5h^2/r_p^2(t)$.

B. System Analysis

Theorem 3.1: Consider the system in (6), and assume that only one solenoid is activated at each time. Let (x_s, y_s, z_s) be the location of the active solenoid and $\tau > 0$ be an arbitrary scaling constant with unit (s). If

$$r_0 < \sqrt{5}h, \quad \alpha > \frac{r_0^7}{\tau^2\left(5h^2 - r_0^2\right)}, \quad \beta > 0 \tag{7}$$

then the desired equilibrium point $(x_s, y_s, 0, 0)$ is asymptotically stable, and the region of attraction contains the set

$$\Omega := \left\{\left(x_d, y_d, v_x, v_y\right) \mid \Delta_x^2 + \Delta_y^2 + \tau^2\left(v_x^2 + v_y^2\right) \le r_0^2 - h^2\right\}$$

where $\Delta_x := x_d - x_s$ and $\Delta_y := y_d - y_s$. Moreover, the function

$$V := \frac{1}{2}\left(\Delta_x^2 + \Delta_y^2 + \tau^2\left(v_x^2 + v_y^2\right)\right)$$

is a Lyapunov function that satisfies

$$V(t) \le V(0)\, e^{-\lambda t} \quad \forall t \ge 0$$

for all trajectories starting within Ω, with a convergence rate

$$\lambda \ge \frac{\tau\alpha\left(5h^2 - r_0^2\right)}{r_0^7} - \frac{1}{\tau}.$$

Proof: Clearly, the compact set Ω contains the desired equilibrium point $(x_s, y_s, 0, 0)$. All other equilibrium points of (6) belong to the set $\{(x_d, y_d, 0, 0) \mid \Delta_x^2 + \Delta_y^2 = 4h^2\}$ and thus cannot lie within Ω, since the first inequality in (7) implies $r_0^2 - h^2 < 4\,h^2$.

Differentiating V along the trajectories of (6) yields (note that $\delta_p = 0$ for all $p \neq s$):

$$\begin{aligned}\dot{V} &\le \Delta_x v_x + \Delta_y v_y - \tau^2\beta\left(v_x^2 + v_y^2\right)\\ &\quad + \tau^2\alpha\left(1 - \frac{5h^2}{r^2}\right)\frac{\Delta_x v_x + \Delta_y v_y}{r^5}\\ &\le \left(\Delta_x v_x + \Delta_y v_y\right)\left(1 + \frac{\tau^2\alpha}{r^5}\left(1 - \frac{5h^2}{r^2}\right)\right)\\ &\quad - \tau^2\beta\left(v_x^2 + v_y^2\right)\end{aligned}$$

where $r := \sqrt{\Delta_x^2 + \Delta_y^2 + h^2}$ satisfies $r \le r_0 < \sqrt{5}h$ within Ω by the first inequality in (7). The last term $-\tau^2\beta(v_x^2 + v_y^2) \le 0$ as $\beta, \tau > 0$. Moreover, Young's inequality implies

$$\Delta_x v_x + \Delta_y v_y \le \frac{\Delta_x^2 + \Delta_y^2}{2\tau} + \frac{\tau\left(v_x^2 + v_y^2\right)}{2} = \frac{1}{\tau}V.$$

Substituting these inequalities into the Lyapunov derivative yields

$$\dot{V} \le \frac{1}{\tau}\left(1 + \frac{\tau^2\alpha}{r^5}\left(1 - \frac{5h^2}{r^2}\right)\right)V =: \kappa V. \tag{8}$$

Then, $\dot{V} < 0$ within Ω, except at the equilibrium point, if $\kappa < 0$. As $r \le r_0 < \sqrt{5}h$, this is equivalent to

$$\alpha > \frac{r^7}{\tau^2\left(5h^2 - r^2\right)} \tag{9}$$

which is ensured by the second inequality in (7) as the right-hand side is increasing in r.

Finally, combining (8) with Gronwall's lemma shows that V is exponentially converging within Ω, with a rate

$$\lambda \ge -\kappa = \frac{\tau\alpha\left(5h^2 - r^2\right)}{r^7} - \frac{1}{\tau} \ge \frac{\tau\alpha\left(5h^2 - r_0^2\right)}{r_0^7} - \frac{1}{\tau}$$

where the last inequality holds as the right-hand side is decreasing in r and $r \le r_0 < \sqrt{5}h$. ■

Remark 3.1: Theorem 3.1 establishes sufficient conditions on the model parameters α, β, and h to ensure that the disk magnet converges to the solenoid's centroid after entering the compact set Ω. Intuitively, as the surface-to-solenoid distance h increases, a stronger magnetic coupling α is required for stabilization. Additionally, the free parameter τ can be adjusted to reduce conservativeness in the stability bound. This parameter regulates the allowable initial speed of the disk upon entering Ω: a higher τ permits larger initial velocities, allowing a weaker coupling α for stabilization, whereas a lower τ restricts the initial speed, requiring a strong magnetic coupling.

Theorem 3.2: Consider the system in (6), with an active solenoid located at (x_s, y_s, z_s). The transient behavior of the payload in the vicinity of the equilibrium point $(x_s, y_s, 0, 0)$ is characterized by the largest eigenvalue of the Jacobian matrix, given by

$$\lambda_{\max}(\alpha, \beta, h) = -\frac{\beta}{2} + \frac{1}{2}\sqrt{\beta^2 - \frac{16\alpha}{h^5}}. \tag{10}$$

Proof: Linearizing the system in (6) around the equilibrium point $(x_s, y_s, 0, 0)$ yields the following Jacobian matrix:

$$J = \begin{bmatrix} 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ \alpha\frac{\partial f_1}{\partial x_d} & \alpha\frac{\partial f_1}{\partial y_d} & -\beta & 0\\ \alpha\frac{\partial f_2}{\partial x_d} & \alpha\frac{\partial f_2}{\partial y_d} & 0 & -\beta \end{bmatrix}$$

where

$$\begin{aligned}\frac{\partial f_1}{\partial x_d} &= \left(\frac{10h^2\Delta_x}{r^4}\right)\frac{\Delta_x}{r^5} + \left(1 - \frac{5h^2}{r^2}\right)\left(\frac{r^2 - 5\Delta_x^2}{r^7}\right)\\ \frac{\partial f_1}{\partial y_d} &= \left(\frac{10h^2\Delta_x}{r^4}\right)\frac{\Delta_y}{r^5} + \left(1 - \frac{5h^2}{r^2}\right)\left(-\frac{5\Delta_x\Delta_y}{r^7}\right)\\ \frac{\partial f_2}{\partial x_d} &= \left(\frac{10h^2\Delta_y}{r^4}\right)\frac{\Delta_x}{r^5} + \left(1 - \frac{5h^2}{r^2}\right)\left(-\frac{5\Delta_y\Delta_x}{r^7}\right)\\ \frac{\partial f_2}{\partial y_d} &= \left(\frac{10h^2\Delta_y}{r^4}\right)\frac{\Delta_y}{r^5} + \left(1 - \frac{5h^2}{r^2}\right)\left(\frac{r^2 - 5\Delta_y^2}{r^7}\right).\end{aligned}$$

The eigenvalues of J evaluated at the equilibrium point are given by $\lambda = -\beta/2 \pm \sqrt{\beta^2 - 16\alpha/h^5}/2$. As $\alpha, \beta, h > 0$, all eigenvalues have negative real parts, and thus the equilibrium is a sink. Finally, transient behavior can be characterized by the largest eigenvalue of J given by (10). ■

Remark 3.2: Theorem 3.2 provides insights for experimental characterization and tuning. Specifically, the system is underdamped and will exhibit decaying oscillations for $\beta^2 h^5 < 16\alpha$. As the distance h increases, $\lambda_{\max}$ decreases, causing the system to exhibit more oscillatory behavior. In contrast, no transient oscillations will occur for $\beta^2 h^5 > 16\alpha$.

IV. Computational Implementation and Validation

A. Model Calibration

We begin by discretizing equations in (6) using the forward difference method. Given that only one solenoid is activated at a time and that all solenoids are identical in design and

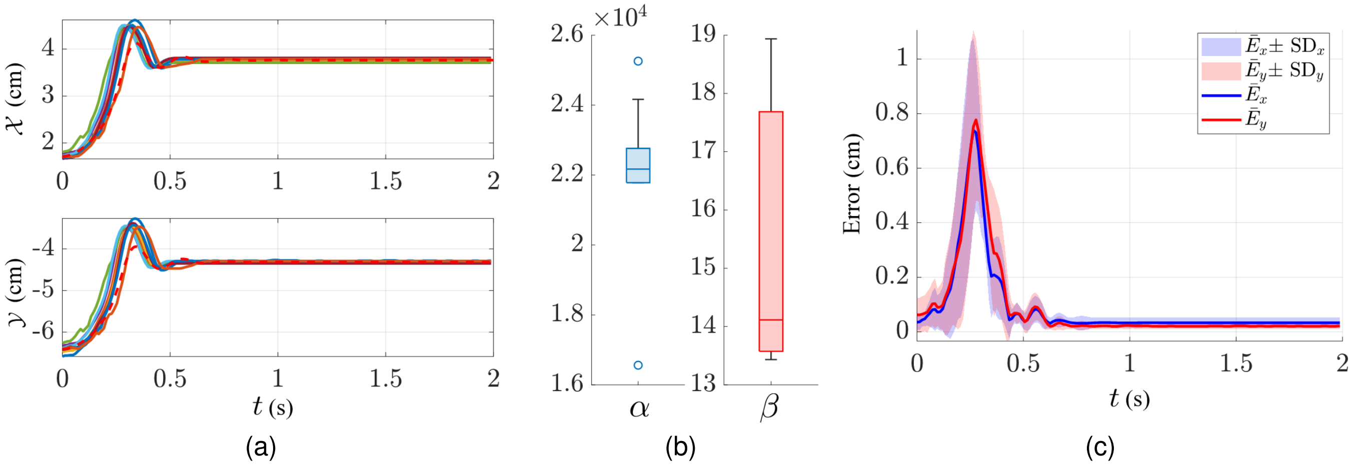


Fig. 3. Model calibration and validation. (a) Time series of experimental data showing a payload being attracted to a single solenoid. Differently colored solid lines represent nine different trials starting from the same initial condition. The red dashed line represents the numerical solution of our model using calibrated parameters. (b) Box plots of the calibrated parameters after solving the optimization problem in (12). Each box plot reports the mean, first, and third quartiles (blue/red boxes). (c) Absolute errors between the numerical solution and experimental data for the $\mathcal{X}$ and $\mathcal{Y}$ components. The dark blue and red lines indicate the mean error across trials for each axis, $\bar{E}_x$ and $\bar{E}_y$, respectively. The lighter blue and red shaded regions represent the corresponding standard deviations.

placement relative to the workspace, we focus on a single solenoid located at (x_s, y_s, z_s), yielding

$$\begin{aligned}
x_d\left((k+1)\,\widetilde{T}\right) &= x_d\left(k\widetilde{T}\right) + v_x\left(k\widetilde{T}\right)\widetilde{T} && \text{(11a)}\\
y_d\left((k+1)\,\widetilde{T}\right) &= y_d\left(k\widetilde{T}\right) + v_y\left(k\widetilde{T}\right)\widetilde{T} && \text{(11b)}\\
v_x\left((k+1)\,\widetilde{T}\right) &= v_x\left(k\widetilde{T}\right) - \beta v_x\left(k\widetilde{T}\right)\widetilde{T}\\
&\quad + \alpha\left(1-\frac{5h^2}{r_s^2\left(k\widetilde{T}\right)}\right)\frac{x_d\left(k\widetilde{T}\right)-x_s}{r_s^5\left(k\widetilde{T}\right)}\widetilde{T} && \text{(11c)}\\
v_y\left((k+1)\,\widetilde{T}\right) &= v_y\left(k\widetilde{T}\right) - \beta v_y\left(k\widetilde{T}\right)\widetilde{T}\\
&\quad + \alpha\left(1-\frac{5h^2}{r_s^2\left(k\widetilde{T}\right)}\right)\frac{y_d\left(k\widetilde{T}\right)-y_s}{r_s^5\left(k\widetilde{T}\right)}\widetilde{T} && \text{(11d)}
\end{aligned}$$

where $\widetilde{T}$ (s) is the sampling period and $k \in \{0, 1, \ldots, N_s - 1\}$ with N_s being the total number of data samples collected. Define $Y_k := [x_d(k\widetilde{T}), y_d(k\widetilde{T}), v_x(k\widetilde{T}), v_y(k\widetilde{T})]^\top$ as the vector of sampled data, and $G_k(\theta)$ as the nonlinear vector function on the right-hand side of the equation set (11) with $h = 2.485$ cm. This value was established by measuring the distance from the solenoid to a plexiglass surface, which was utilized in the experiments (see Section V). The function $G_k(\theta)$ encapsulates the unknown model parameters $\theta := [\alpha, \beta]^\top$, that can be estimated via

$$\widehat{\theta} = \arg\min_{\theta\in\Theta} \sum_{k=0}^{N_s-1} \left\| Y_{k+1} - G_k(\theta) \right\|^2. \tag{12}$$

We collected data from a single solenoid starting from the same initial position in space using a video camera (GS3-U3-41C6NIR-C1, Teledyne FLIR LL) at 83 frames/s so that $\widetilde{T} = 0.012$ s. We conducted ten trials and tracked the position of the disk-magnet using the algorithm described in Section II-B. The data was postprocessed using a moving average filter with a window of three samples. The time series $\{x_d(k\widetilde{T})\}_{k=0}^{N_s-1}$ and $\{y_d(k\widetilde{T})\}_{k=0}^{N_s-1}$ for the ten different trials are shown in Fig. 3(a), represented by the multicolored solid lines.

Next, we used this data to solve the optimization problem in (12) via the Nonlinear Optimization Toolbox in MATLAB. Specifically, we utilized the Sequential Quadratic Programming (SQP) algorithm [52] within the `fmincon` function. For each trial, we obtained a pair of calibrated parameters. The results are shown in Fig. 3(b), where the mean values across trials are $\bar{\alpha} = 22042.797$ and $\bar{\beta} = 15.549$.

B. Model Validation

We numerically solved the system in (6) utilizing the discretized equations in (11) with $\alpha = \bar{\alpha}$ and $\beta = \bar{\beta}$. The results are depicted in Fig. 3(a) as the red dashed line. Note that the simulated system accurately reproduces the time response, with a maximum overshoot at approximately 0.3 s. The absolute error between the simulated trajectory and the actual data of the ten trials is shown in Fig. 3(c), separately for the $\mathcal{X}$ and $\mathcal{Y}$ components. Note that, on average, the error remains below 0.8 cm. To verify system stability, we apply Theorem 3.1 using the calibrated constants $\bar{\alpha}$ and $\bar{\beta}$ while selecting $\tau = 1$. Empirical observations indicated that $r_0 = 3.05$ cm. Note that $r_0 < \sqrt{5}h = 5.556$ and $\bar{\beta} > 0$. Moreover, $\bar{\alpha} > r_0^7/(5h^2 - r_0^2) = 113.808$, which confirms all condition in (7) are satisfied. Consequently, convergence to the equilibrium point $(x_s, y_s, 0, 0)$ is guaranteed with an exponential rate of $\lambda \geq 192.682$. In addition, Theorem 3.2 implies that the system will exhibit decaying oscillations as $\beta^2 h^5 = 22910.033 < 16\alpha = 352684.748$. This theoretical prediction is consistent with the experimental observations in Fig. 3(a), where the system converges within the region defined by r_0 and exhibits decaying oscillations toward the equilibrium.

C. Model Simulation With Feedback Control

Using the calibrated model parameters derived from experimental data, we numerically solve the nonlinear system

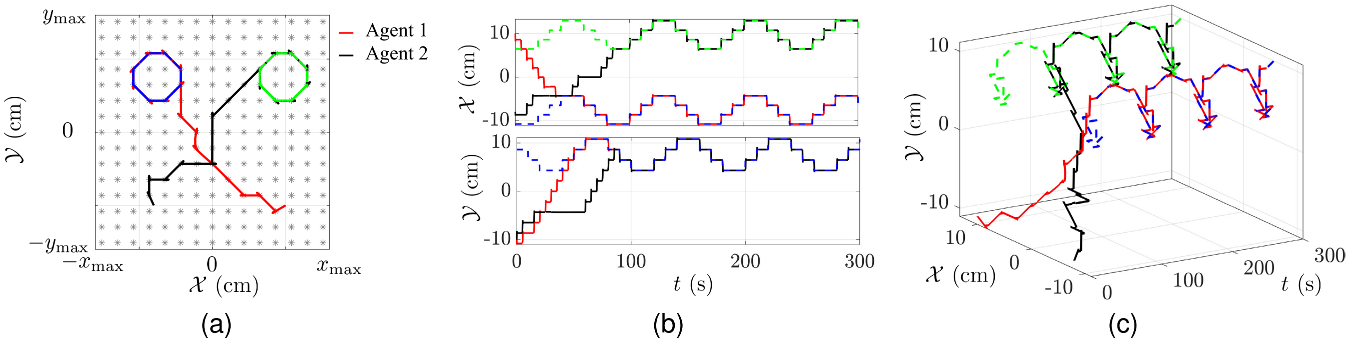


Fig. 4. Computational results. (a) Two agents tracking their respective reference trajectories, shown as blue and green dashed lines, starting from different initial conditions. Gray asterisks * indicate solenoid locations. (b) and (c) 2-D and 3-D representations of the time series depicting the agents' trajectories as they track their references.

in (6) via the forward difference method. We note that more advanced numerical techniques, such as Runge–Kutta methods [53], could be used as well.

Algorithm 1 Control Algorithm

```
while $s \le T_m/T$ do                                  ▷$T_m$: total simulation time
                                    ▷$T$: time between two samples (observations)
    $t_s \leftarrow (s-1)T_s + 1$
                               ▷Compute first simulation step $t_s$ in the current sample
                               ▷$T_s$: number of simulation steps between two samples
    $Sample \leftarrow (x_d(t_s), y_d(t_s))$
                                                               ▷Observe disk location
    $Ref, uRef \leftarrow$ COLLISIONAVOIDANCE($Sample$, $A$)
                                       ▷Input reference location. $A$: adjacency matrix
    $e \leftarrow$ NORM($Sample, Ref$)
                                                               ▷Compute tracking error
    if $e \le r_0$ then                                ▷$r_0$: solenoid activation radius
        $u(t_s, t_s + T_s - 1) \leftarrow uRef$
                                               ▷Choose tracking control if close enough
    else
        $Node, uNode \leftarrow$ CLOSESTSOLENOID($Sample$)
                                                    ▷Compute closest solenoid to disk
        $Path, uPath \leftarrow$ SHORTESTPATH($Node, Ref$)
                                      ▷Compute control for shortest path to reference
        if NORM($Sample, Path(2)$) $\le r_0$ then
            $u(t_s, t_s + T_s/2 - 1) \leftarrow uPath(2)$
            $u(t_s + T_s/2, t_s + T_s - 1) \leftarrow uPath(3)$
                               ▷Go to 2nd and 3rd solenoids in path if close enough
        else
            $u(t_s, t_s + T_s/2 - 1) \leftarrow uNode$
            $u(t_s + T_s/2, t_s + T_s - 1) \leftarrow uPath(2)$
                                     ▷Go to 1st and 2nd solenoids in path otherwise
        end if
    end if
end while
```

Additionally, we implemented the feedback controller described in Section II-C following the operational logic detailed in Algorithm 1. At each time step, the system samples the current positions of the agents, loads the reference trajectories, and computes the control inputs that modulate the magnetic forces. The selection of control action depends on the instantaneous tracking error. If the agent's current position lies within the activation radius r_0 of the next solenoid on its reference trajectory X_i^{ref}, the controller proceeds with direct trajectory tracking. If the tracking error exceeds r_0 and no nearby agents are detected within the collision avoidance radius, the controller switches to the path planning mode. In this case, the function SHORTESTPATH(Node, Ref) is invoked, where Node denotes the agent's current solenoid and Ref is the next target solenoid on the reference trajectory. This routine utilizes the adjacency matrix A of the solenoid connectivity graph $\mathscr{G}$ to compute the shortest path via Dijkstra's algorithm, returning an intermediate trajectory X_i^{plan} that steers the agent back to the reference. If nearby agents are detected, the controller enters the collision avoidance mode. The function COLLISIONAVOIDANCE(Sample, A) is called, which considers the current agent positions and the adjacency matrix. For simplicity, we implement a prioritized halting strategy: lower priority agents temporarily pause to allow higher priority agents to proceed, thus preventing collisions. To ensure safe path computation during this mode, edges in A that lead to imminent collisions (e.g., solenoids within the conflict region of another agent) are temporarily assigned infinite weight. We emphasize that different collision avoidance strategies could be implemented, as illustrated in the experimental results, where we showcase an alternative approach based on line-of-sight and time-to-collision as well. Additionally, we provide an implementation of the Rapidly exploring random tree method (see the Data Availability section).

We simulate a scenario where two agents each tracks a circular reference trajectory [see Fig. 4(a)], starting from positions that lead to intersecting paths. This example illustrates collision avoidance as each agent navigates toward its respective target in opposite directions. As shown in Fig. 4(b) and (c), the agents successfully converge to their reference trajectories while avoiding collisions. Specifically, Agent 2 halts while Agent 1 moves outside the collision avoidance region.

V. Experimental Results

A. Experimental Apparatus

To conduct the experiments, we utilized the setup shown in Fig. 5. It consists of a 5 × 5 MagTile array, comprising

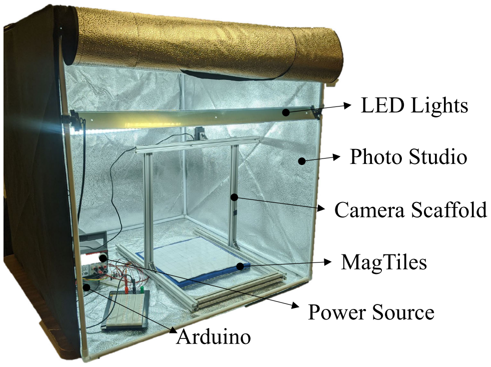


Fig. 5. Overview of the experimental apparatus.

225 solenoids arranged in a 15 × 15 grid (with an inter-solenoid spacing of 21.59 mm), a programmable power supply (Keithley 2231A-30-3), a photo studio with LED lighting to ensure consistent and uniform illumination, and a fixed scaffold supporting an overhead camera (Logitech C920 HD Pro Webcam) for real-time multiagent tracking.

B. Case Study 1: Illustrative Examples

We start by considering a scenario in which two agents must track reference trajectories given by concentric semicircles, as shown in Fig. 6(a). The black and orange curves denote the reference paths, while the larger dots indicate the agents' initial positions. As the system evolves, the agents asymptotically converge to their designated trajectories, as depicted in Fig. 6(d), with the corresponding tracking errors, $E_i(t)$, shown in Fig. 6(g). Here, $E_i(t) \; : \; = \|X_i^{\text{ref}}(t) - X_i(t)\|$ with $t = kT$, $X_i^{\text{ref}}(t) = (x_i^r(t), y_i^r(t))$ and $X_i(t) = (x_i(t), y_i(t))$. The abrupt change observed at $t = 58.78$ s results from a transient measurement error, where the tracking algorithm failed to detect the agent in a single frame. Additional quantitative results from the computational analysis of tracking errors are provided in the Data Availability section. Next, we modify the reference trajectories to two adjacent circles, depicted in Fig. 6(b). The agents are initialized such that their paths intersect over time, as their respective targets lie in opposite directions relative to their starting positions. The results in Fig. 6(e) and (h) confirm that the agents successfully reach their targets, even under external disturbances. Specifically, at $t \approx 50$ s, one agent was manually perturbed off its trajectory. Despite this displacement, the agent was able to converge back to the reference trajectory. Note that the tracking error exhibits an initial transient during convergence, a short increase following the perturbation, and subsequently decays as the agents return to their reference trajectories.

C. Case Study 2: Vision-Based Collision Avoidance

We also implement a collision avoidance strategy based on line-of-sight and time-to-collision, originally formulated for continuous-time robot interactions [42]. The scenario consists of two agents: Agent 1, a noncooperative entity that may be either static or dynamic, and Agent 2, which actively navigates the workspace while adjusting its steering angle to avoid collisions. Specifically, the future position of Agent 2 is determined based on its current state and the collision avoidance control law given by

$$\begin{cases} \text{no change,} & \text{if } \dot{\psi}_{\text{ca}} \leq \psi_{\min} \\ \text{diagonal-left position,} & \text{if } \psi_{\min} < \dot{\psi}_{\text{ca}} < \psi_{\max} \\ \text{perpendicular-left position,} & \text{if } \dot{\psi}_{\text{ca}} \geq \psi_{\max} \end{cases} \tag{13}$$

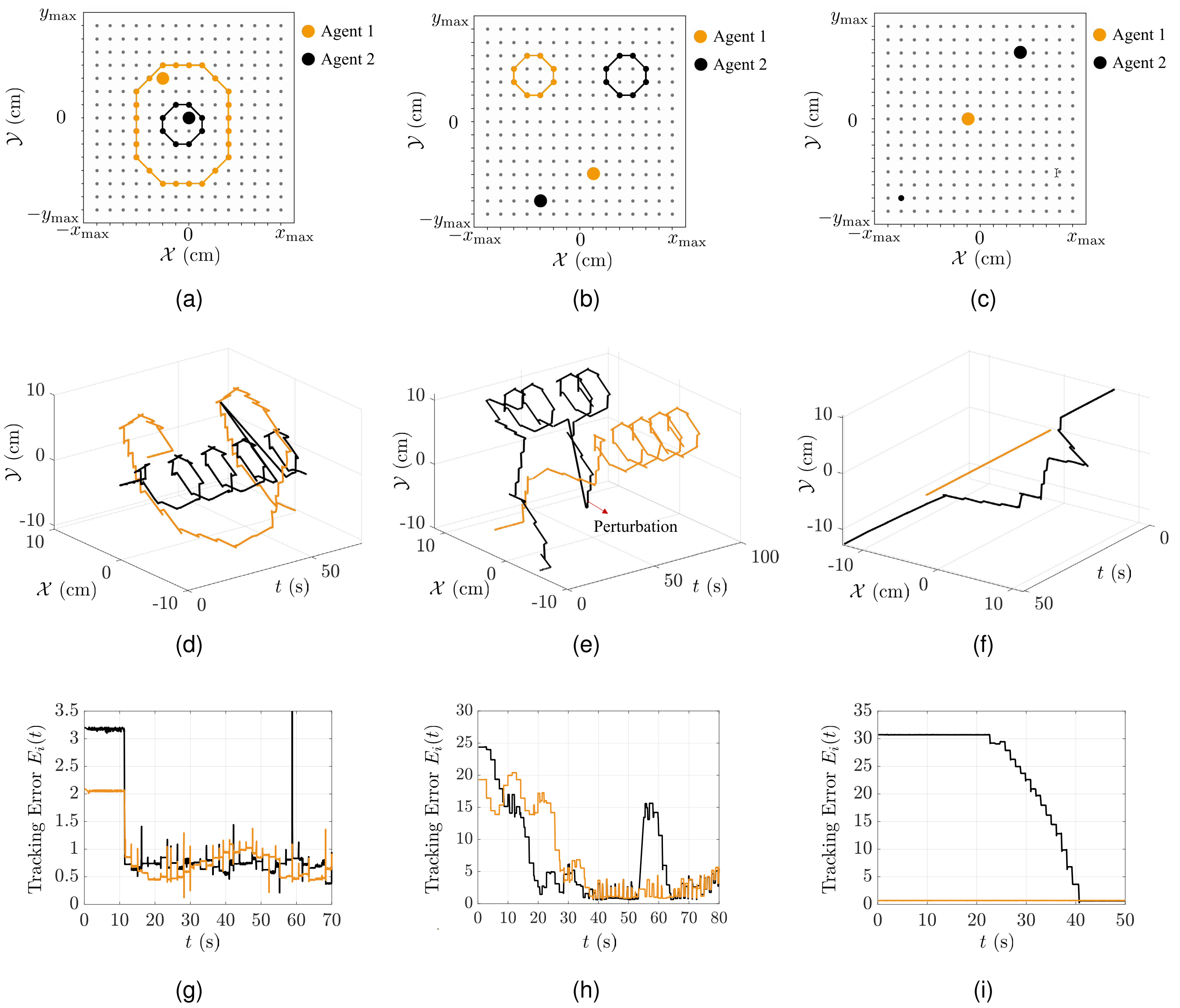
where $\psi_{\min}$ (rad) and $\psi_{\max}$ (rad) are discretization thresholds that define the steering response. The collision avoidance steering angle rate, $\dot{\psi}_{\text{ca}}$ (rad/s) is computed as

$$\dot{\psi}_{\text{ca}}(kT) \;=\; \kappa\,\eta\,(kT)\; e^{-k_\tau \tau(kT)}. \tag{14}$$

Here, κ (s^{-1}) and k_τ (s^{-1}) are tuning parameters that regulate the magnitude of steering adjustments and the sensitivity of the response based on proximity to Agent 1. The variable τ (s) is the time-to-collision defined as $\tau(kT) \; : \; = \rho(kT)/\dot{\rho}(kT)$ with ρ (cm) being the Euclidean distance between the agents. The variable η (rad) quantifies the alignment of the Agent 1 heading vector with the centroid of the obstacle (Agent 2) and is given by $\eta(kT) \; : \; = (\pi/2) + \lambda(kT) - \psi(kT)$, where λ (rad) is the angle between the heading angle and the horizontal axis ($\mathcal{X}$-axis) of the inertial coordinate frame, and ψ (rad) denotes the heading angle. The parameters were heuristically set to $\psi_{\min} = 0.5$ rad, $\psi_{\max} = 0.9$ rad, and $\kappa = k_\tau = 1$ s^{-1}, based on experimental results that demonstrated effective performance. The corresponding results are shown in Fig. 6(c), (f), and (i), where Agent 1 remains static while Agent 2 navigates the workspace, avoiding the obstacle to reach its target on the opposite side. Initially, Agent 2 exhibits a large tracking error as it deviates from the nominal path to execute the avoidance maneuver. Once the obstacle is cleared, the agent resumes its trajectory toward the target, and the tracking error progressively decreases until convergence is achieved.

D. Case Study 3: Fish Collective Behavior

Motivated by the study of fish collective behavior through ethorobotics [25], [29], [30], [31], we explore a scenario with a different type of payload than the disks used in the previous examples. Specifically, we designed 3-D-printed zebrafish with a magnetic core and introduced them into a transparent acrylic tank filled with water. The tank was placed atop the MagTile platform. We evaluated the reference trajectory shown in Fig. 7(a) under two scenarios: 1) open-loop control [see Fig. 7(b)], in which agents follow their predefined trajectories without feedback and 2) closed-loop control [see Fig. 7(c)], in which agents incorporate real-time feedback from the tracking system to compensate for external disturbances, using the control logic defined in (2). The corresponding tracking errors are reported in Fig. 7(d) and (e), respectively. In the open-loop condition, the agents generally follow their reference trajectories; however, occasional actuation delays may cause the agent to miss one or more control steps. When this occurs, the affected agent must wait until the next update of the reference loop to resume its trajectory, resulting in transient tracking deviations, as illustrated for Agent 2 in Fig. 7(d). Additionally, following a strong manual perturbation at $t = 65.19$ s, the open-loop controller



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 6. Experimental results. Reference and time-series trajectories of two agents for different scenarios. (a) and (d) Agents following two concentric circular trajectories. (b) and (e) Agents traversing two circular trajectories from opposite initial positions. A manual perturbation is applied at t = 53.49 s, displacing one agent from its trajectory. Halting collision avoidance is used in (a) and (d) and (b) and (e). (c) and (f) Illustration of the collision avoidance strategy in (13). (g)–(i) Tracking errors of the two agents (yellow: agent 1, black: agent 2) over time for the three corresponding scenarios.

fails to maintain accurate trajectory tracking for both agents. In contrast, under closed-loop control, the agents rapidly correct such deviations and maintain accurate trajectory tracking even in the presence of disturbances. In addition, we observed that the median speed of the fish in open-loop was 7.01 cm/s, whereas in closed-loop, it was reduced to 3.02 cm/s.

Finally, to demonstrate the potential of MagTile to replicate naturalistic biological locomotion, we used real 2-D trajectories from a single zebrafish, as reported in [54]. These overhead trajectories, recorded in a circular tank [see Fig. 8(a)], were discretized and mapped onto the MagTile workspace [see Fig. 8(b)]. Experimental results are reported in Fig. 8(c), showing the motion of the robotic replica in 3-D space, and in Fig. 8(d) and (e), which illustrate the corresponding time series along the $\mathcal{X}$ and $\mathcal{Y}$ axes, respectively. The discretization errors, defined as the absolute difference between the biological trajectory and the actual system output for each spatial component, are shown in Fig. 8(f). These results demonstrate the system's capability to reproduce biologically realistic trajectories while also revealing its current limitations. In particular, the presence of water and the added mass of the payload introduce hydrodynamic resistance that is not yet accounted for in the current design or mathematical model. During smooth segments of the fish trajectory, tracking accuracy remains high; however, during abrupt maneuvers or sharp turns, transient deviations occur due to the finite actuation speed of the platform. Future work will seek to enhance actuation speed, designing robust feedback controllers to improve trajectory fidelity, and extending mathematical models to incorporate hydrodynamic effects, which will enable more accurate simulation, testing, and optimization of actuation performance prior to physical implementation.

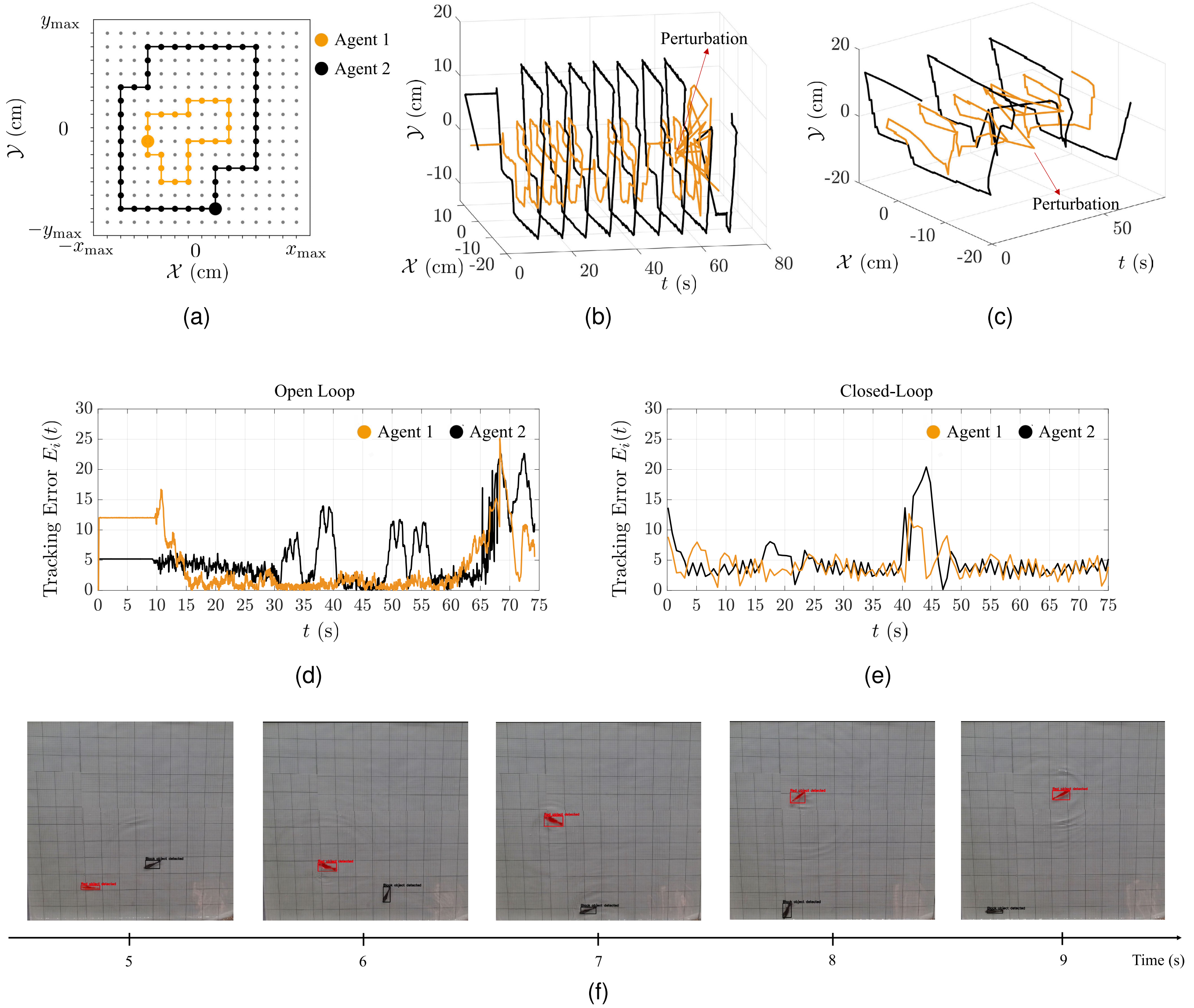


Fig. 7. Experimental results using a 3-D-printed fish replica with a magnetic core as the actuated payload. (a) Reference trajectories. Time-series trajectories of two fish replicas in (b) open-loop configuration and (c) closed-loop configuration, with manual perturbations at t = 65.19 s and t = 40.43 s, respectively. (d) and (e) Tracking errors of the two agents (yellow: agent 1; black: agent 2) over time for the open-loop and closed-loop configurations. (f) Sequential snapshots of the two fish replicas tracking their reference trajectories.

E. Computational Experiment of Formation Control

We consider a scenario involving a larger number of agents. Specifically, we address the formation control problem for ten agents operating within an expanded workspace composed of 30 × 30 solenoids. At the beginning of the simulation, agents are randomly distributed across the workspace and are required to reach a target formation arranged in a circular pattern, indicated by the blue circles in the top panel of Fig. 9. Once convergence to this formation is achieved, the reference configuration changes to a rectangular formation, shown as red circles in the same figure. To quantify performance, we compute the tracking error for each agent, $E_i(t)$. As shown in the middle panel of Fig. 9, $E_i(t)$ is initially large but gradually decreases to zero as agents reach the first formation, increases again after the formation pattern switches, and finally converges back to zero as the new rectangular formation is achieved. In addition, we record the number of collision avoidance maneuvers (halting strategy), denoted as CA(t), triggered during the experiment (bottom panel of Fig. 9). This variable increases when agents' paths intersect while moving toward their targets, reflecting active avoidance behaviors, and returns to zero once all agents have reached their assigned positions.

F. Computational Experiment With Controlled and Uncontrolled Agents

In biological experiments, live animals are uncontrolled agents as they move autonomously and could exhibit random motion. MagTile can accommodate the presence of such uncontrolled agents, provided their positions are available in real-time. In fact, controlled agents can treat uncontrolled

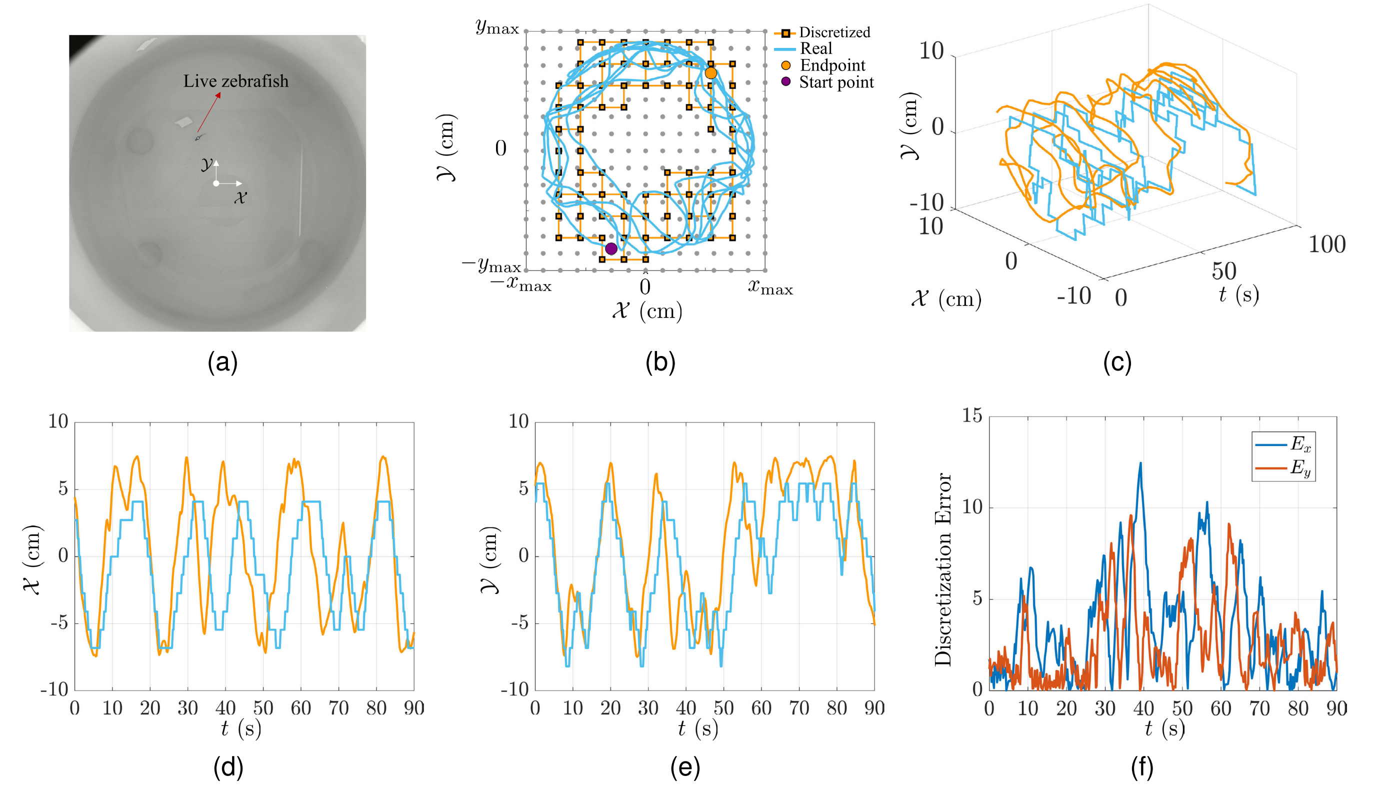


Fig. 8. Experimental results using real 2-D trajectories from a single zebrafish. (a) Circular experimental arena with live zebrafish (*danio rerio*). (b) Real fish trajectory and discretized trajectory in MagTile workspace. (c) Time-series of the fish replica trajectory and reference. (d) and (e) Time evolution of the replica trajectory along the $\mathcal{X}$- and $\mathcal{Y}$-axes, respectively. (f) Tracking error for each spatial component over time.

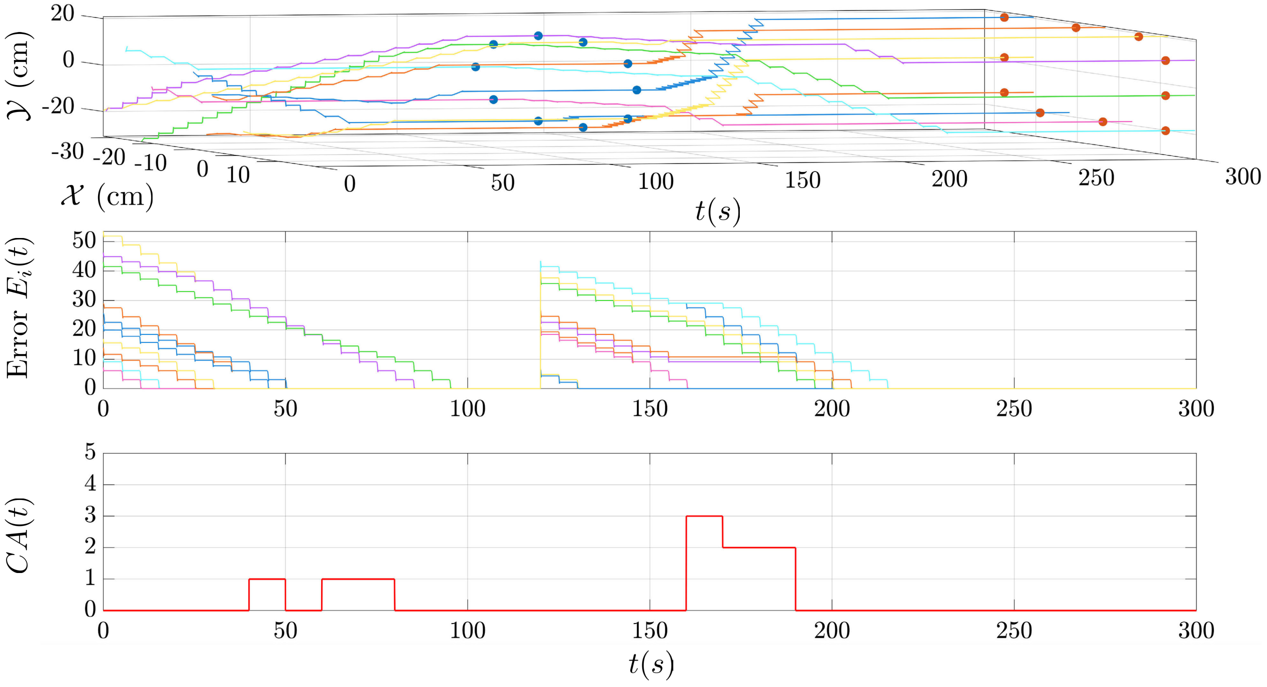


Fig. 9. Computational experiment on formation control with ten agents. Agents are initially placed at random positions and are required to reach a circular formation (blue circles). Once convergence is achieved, the reference configuration switches to a rectangular formation (red circles) (top). Time evolution of the individual tracking errors $E_i(t)$ (middle). Number of collision avoidance maneuvers CA(t) recorded during the experiment (bottom).

agents as moving obstacles and execute collision-avoidance maneuvers through feedback.

To illustrate this, we extend the simulation scenario of Fig. 4 to a mixed case consisting of one controlled agent

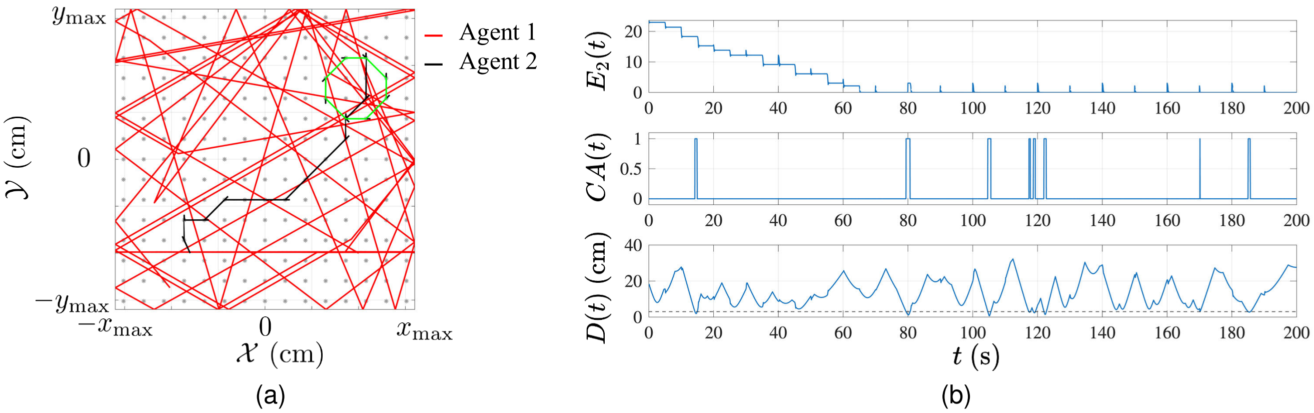


Fig. 10. Computational experiment of controlled and uncontrolled agents. Agent 2 is actuated by MagTile and tracks a prescribed reference trajectory, while Agent 1 is unactuated and moves randomly over the space. (a) Trajectories of both agents. (b) Tracking error $E_2(t)$ of agent 2, number of collision-avoidance maneuvers CA(t), and Euclidean distance $D(t)$ among agents recorded during the experiment.

and one uncontrolled agent sharing the same workspace. In this example, Agent 2 remains actuated by MagTile and continues to track its reference trajectory, while Agent 1 is unactuated and moves freely according to standard discretized unicycle kinematics. To emulate an unpredictable agent, the speed and heading of Agent 1 are randomly perturbed every two sampling periods. That is, at each update, the forward speed and heading rate are drawn from Gaussian distributions (with bounds enforced to maintain positive speeds), and then held constant until the next update. In addition, we impose hard boundary constraints so that when Agent 1 reaches the workspace boundary, its direction is reflected, ensuring the agent remains contained within the MagTile workspace. The resulting simulations are shown in Fig. 10. Note that the uncontrolled agent moves randomly within the workspace, while the controlled agent continues to pursue its reference and triggers the avoidance maneuver whenever the two agents come into close proximity.

VI. Conclusion

Ethorobotics, the study of animal–robot interactions, has emerged as a powerful framework for investigating collective behavior and social dynamics [1], [2], [3]. Existing platforms, however, often exhibit limited scalability, restricted adaptability, and a lack of computational tools for in-silico experimentation [29], [31], [32], constraining their applicability across diverse experimental configurations.

Motivated by these fundamental limitations, this work introduced MagTile, a modular and scalable platform for the control of electromagnetically actuated agents. MagTile integrates a modular actuation system, vision-based multiagent tracking, and computer-based control. Central to the platform's architecture is a tile-based electromagnetic actuation system, which allows flexible reconfiguration, crucial for covering different experimental areas.

We developed a mathematical model of MagTile to characterize the interaction between solenoids and disk magnet payloads, utilizing a fictitious magnetic monopole representation. The resulting nonlinear state-space model was analyzed using Lyapunov stability theory, providing theoretical insights into system behavior and performance. Building on this model, we established a computational framework for simulating, testing, and optimizing experimental conditions prior to physical deployment, minimizing trial-and-error experimentation. The platform incorporates a feedback control strategy based on Dijkstra's algorithm that guarantees real-time trajectory tracking. To validate the system's effectiveness, we conducted a series of illustrative experiments demonstrating multiagent coordination, where agents successfully followed prescribed trajectories despite disturbances. Additionally, we implemented and validated an alternative collision avoidance strategy based on line-of-sight and time-to-collision. Finally, we showcased MagTile's potential for ethorobotics applications by actuating a pair of 3-D-printed fish replicas, illustrating its utility for animal-robot interaction studies.

Our proposed platform is not free of limitations that call for future endeavors. First, the current system operates by activating a single solenoid at a time per agent, resulting in discrete stepwise transitions between tiles. In addition, the communication protocol imposes a hardware limit of approximately 60 tiles, as the driver chips cannot support more than 60 unique addresses. Future work will investigate multisolenoid activation with time-varying PWM control to achieve smoother transitions and more continuous motion. Moreover, we plan to design new circuit boards that support higher voltage operation, real-time power regulation, and an optimized communication protocol, thereby increasing the number of addressable tiles and enhancing overall system scalability. Second, there is a noticeable disparity in agent speed between open-loop and closed-loop actuation, primarily due to the latency introduced by vision-based tracking and control algorithms. Future improvements will focus on reducing computational delays, enhancing real-time responsiveness, and optimizing both control and tracking algorithms. Third, animal behavior is inherently stochastic [55], [56], [57], raising the challenge of tracking complex trajectories. This will require the development of advanced control strategies (possibly using model predictive control [58]), capable of real-time trajectory prediction and adaptive path planning. Fourth, this study does not evaluate fully heterogeneous scenarios in which multiple uncontrolled agents (live animals) interact simultaneously with controlled agents in the same workspace. In principle, uncontrolled agents can be treated as moving obstacles. If their positions are available in real time, the controlled agents can execute collision-avoidance maneuvers (see Section V-F). However, enabling and validating general heterogeneous environments with multiple live subjects would require a robust real-time perception layer capable of tracking several individuals under occlusions and intermittent detections. Developing and integrating this capability into MagTile is an important direction for future work.

Although there are several avenues of future research, MagTile has the potential to further advance studies on ethorobotics and bioinspired MASs, paving the way for new discoveries in robotics and collective behavior.

Data Availability

The circuit designs, computational model code, and datasets used in this study are available at https://github.com/swarmintelligencelab/MagTile/tree/main

References

[1] J. Shamma, *Cooperative Control of Distributed Multi-Agent Systems*. Hoboken, NJ, USA: Wiley, 2008.

[2] F. Bullo, J. Cortés, and S. Martinez, *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*. Princeton, NJ, USA: Princeton Univ. Press, 2009.

[3] F. Chen and W. Ren, "On the control of multi-agent systems: A survey," *Found. Trends Syst. Control*, vol. 6, no. 4, pp. 339–499, 2019.

[4] E. Tuci, M. H. M. Alkilabi, and O. Akanyeti, "Cooperative object transport in multi-robot systems: A review of the state-of-the-art," *Frontiers Robot. AI*, vol. 5, p. 59, May 2018.

[5] M. A. Mahmood, W. K. G. Seah, and I. Welch, "Reliability in wireless sensor networks: A survey and challenges ahead," *Comput. Netw.*, vol. 79, pp. 166–187, Mar. 2015.

[6] D. Angulo-Garcia and D. Burbano, "Advection dynamics in traffic networks: Modeling, analysis, and optimization," *IEEE Trans. Netw. Sci. Eng.*, vol. 12, no. 5, pp. 3611–3624, Sep. 2025.

[7] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28573–28593, 2018.

[8] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2017, pp. 66–83.

[9] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. New York, NY, USA: Oxford Univ. Press, 1999.

[10] K. M. Passino, *Biomimicry for Optimization, Control, and Automation*. Cham, Switzerland: Springer, 2005.

[11] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, 2006.

[12] J. Werfel, K. Petersen, and R. Nagpal, "Designing collective behavior in a termite-inspired robot construction team," *Science*, vol. 343, no. 6172, pp. 754–758, Feb. 2014.

[13] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 3293–3298.

[14] D. Pickem et al., "The robotarium: A remotely accessible swarm robotics research testbed," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 1699–1706.

[15] H. Xie et al., "Reconfigurable magnetic microrobot swarm: Multimode transformation, locomotion, and manipulation," *Sci. Robot.*, vol. 4, no. 28, p. 8006, Mar. 2019.

[16] M. Itani, T. Chen, T. Yoshioka, and S. Gollakota, "Creating speech zones with self-distributing acoustic swarms," *Nature Commun.*, vol. 14, no. 1, p. 5684, Sep. 2023.

[17] S. Ceron, G. Gardi, K. Petersen, and M. Sitti, "Programmable self-organization of heterogeneous microrobot collectives," *Proc. Nat. Acad. Sci. USA*, vol. 120, no. 24, Jun. 2023, Art. no. e2221913120.

[18] B. Webb, "What does robotics offer animal behaviour?," *Animal Behaviour*, vol. 60, no. 5, pp. 545–558, Nov. 2000.

[19] J. Krause, A. F. T. Winfield, and J.-L. Deneubourg, "Interactive robots in experimental biology," *Trends Ecology Evol.*, vol. 26, no. 7, pp. 369–375, Jul. 2011.

[20] A. Frohnwieser, J. C. Murray, T. W. Pike, and A. Wilkinson, "Using robots to understand animal cognition," *J. Experim. Anal. Behav.*, vol. 105, no. 1, pp. 14–22, Jan. 2016.

[21] J. Abdai and A. Miklosi, *An Introduction to Ethorobotics: Robotics and the Study of Animal Behaviour*. New York, NY, USA: Taylor & Francis, 2024.

[22] B. A. Klein, J. Stein, and R. C. Taylor, "Robots in the service of animal behavior," *Communicative Integrative Biol.*, vol. 5, no. 5, pp. 466–472, Sep. 2012.

[23] S. Butail, N. Abaid, S. Macrì, and M. Porfiri, "Fish–robot interactions: Robot fish in animal behavioral studies," in *Robot Fish: Bio-Inspired Fishlike Underwater Robots*. Cham, Switzerland: Springer, 2015, pp. 359–377.

[24] D. Romano, E. Donati, G. Benelli, and C. Stefanini, "A review on animal–robot interaction: From bio-hybrid organisms to mixed societies," *Biol. Cybern.*, vol. 113, no. 3, pp. 201–225, Jun. 2019.

[25] F. Bonnet et al., "Robots mediating interactions between animals for interspecies collective behaviors," *Sci. Robot.*, vol. 4, no. 28, p. 7897, Mar. 2019.

[26] L. K. Quinn et al., "When rats rescue robots," *Animal Behav. Cognition*, vol. 5, no. 4, pp. 368–379, Nov. 2018.

[27] R. Siddall, "Ethorobotic rats for rodent behavioral research: Design considerations," *Frontiers Behav. Neurosci.*, vol. 17, Dec. 2023, Art. no. 1281494.

[28] R. d'Isa, "Robotic animals as new tools in rodent neuroscience research: Proposed applications of zooinspired robots for mouse behavioral testing," *Frontiers Behav. Neurosci.*, vol. 19, Feb. 2025, Art. no. 1545352.

[29] D. T. Swain, I. D. Couzin, and N. Ehrich Leonard, "Real-time feedback-controlled robotic fish for behavioral experiments with fish schools," *Proc. IEEE*, vol. 100, no. 1, pp. 150–163, Jan. 2012.

[30] T. Landgraf, D. Bierbach, H. Nguyen, N. Muggelberg, P. Romanczuk, and J. Krause, "RoboFish: Increased acceptance of interactive robotic fish with realistic eyes and natural motion patterns by live trinidadian guppies," *Bioinspiration Biomimetics*, vol. 11, no. 1, Jan. 2016, Art. no. 015001.

[31] M. Karakaya, S. Macrì, and M. Porfiri, "Behavioral teleporting of individual ethograms onto inanimate robots: Experiments on social interactions in live zebrafish," *iScience*, vol. 23, no. 8, Aug. 2020, Art. no. 101418.

[32] P. De Lellis, E. Cadolini, A. Croce, Y. Yang, M. d. Bernardo, and M. Porfiri, "Model-based feedback control of live zebrafish behavior via interaction with a robotic replica," *IEEE Trans. Robot.*, vol. 36, no. 1, pp. 28–41, Feb. 2020.

[33] C. Huang, Z. Lai, L. Zhang, X. Wu, and T. Xu, "A magnetically controlled soft miniature robotic fish with a flexible skeleton inspired by zebrafish," *Bioinspiration Biomimetics*, vol. 16, no. 6, Nov. 2021, Art. no. 065004.

[34] A. Myklatun et al., "Zebrafish and medaka offer insights into the neurobehavioral correlates of vertebrate magnetoreception," *Nature Commun.*, vol. 9, no. 1, p. 802, Feb. 2018.

[35] R. E. Blaser and D. B. Rosemberg, "Measures of anxiety in zebrafish (Danio rerio): Dissociation of black/white preference and novel tank test," *PLoS ONE*, vol. 7, no. 5, May 2012, Art. no. e36931.

[36] R. Benvenutti, M. Marcon, M. Gallas-Lopes, A. J. de Mello, A. P. Herrmann, and A. Piato, "Swimming in the maze: An overview of maze apparatuses and protocols to assess zebrafish behavior," *Neurosci. Biobehavioral Rev.*, vol. 127, pp. 761–778, Aug. 2021.

[37] W. M. S. Russell, R. L. Burch, and C. W. Hume, *The Principles of Humane Experimental Technique*, vol. 238. London, U.K.: Methuen, 1959.

[38] H. K. Khalil, *Nonlinear Systems*, 3rd ed., Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[39] OpenCV Team. (2024). *OpenCV: Open Source Computer Vision Library, Version 4.8.0*. [Online]. Available: https://opencv.org/

[40] P. S. Foundation. (2024). *Python Programming Language*. Accessed: Dec. 31, 2024. [Online]. Available: https://www.python.org

[41] G. Chen, X. Wang, and X. Li, *Introduction to Complex Networks: Models, Structures and Dynamics*. Beijing, China: Higher Education Press, 2012.

[42] T. Marinho, M. Amrouche, V. Cichella, D. Stipanovic, and N. Hovakimyan, “Guaranteed collision avoidance based on line-of-sight angle and time-to-collision,” in *Proc. Annu. Amer. Control Conf. (ACC)*, Jun. 2018, pp. 4305–4310.

[43] J. Yu and S. M. LaValle, “Multi-agent path planning and network flow,” in *Proc. Algorithmic Found. Robot. 10th Workshop Algorithmic Found. Robot.* Cham, Switzerland: Springer, 2013, pp. 157–173.

[44] V. R. Desaraju and J. P. How, “Decentralized path planning for multi-agent teams with complex constraints,” *Auto. Robots*, vol. 32, no. 4, pp. 385–403, May 2012.

[45] P. Tucnik, T. Nachazel, P. Cech, and V. Bures, “Comparative analysis of selected path-planning approaches in large-scale multi-agent-based environments,” *Expert Syst. Appl.*, vol. 113, pp. 415–427, Dec. 2018.

[46] E. W. Dijkstra, “A note on two problems in Connexion with graphs,” in *Edsger Wybe Dijkstra: His Life, Work, and Legacy*. New York, NY, USA: ACM, 2022, pp. 287–290.

[47] B. Haeupler, R. Hladík, V. Rozhoň, R. E. Tarjan, and J. Tětek, “Universal optimality of Dijkstra via beyond-worst-case heaps,” in *Proc. IEEE 65th Annu. Symp. Found. Comput. Sci. (FOCS)*, Oct. 2024, pp. 2099–2130.

[48] T. L. Chow, *Introduction to Electromagn. Theory: A Modern Perspective*. Burlington, MA, USA: Jones & Bartlett Learning, 2006.

[49] D. J. Griffiths, *Introduction to Electrodynamics*. Cambridge, U.K.: Cambridge Univ. Press, 2023.

[50] T. H. Boyer, “The force on a magnetic dipole,” *Amer. J. Phys.*, vol. 56, no. 8, pp. 688–692, Aug. 1988.

[51] J. Wittenburg, *Dynamics of Systems of Rigid Bodies*. Cham, Switzerland: Springer-Verlag, 2013.

[52] J. Nocedal and S. Wright, *Numerical Optimization*. Cham, Switzerland: Springer, 2006.

[53] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*. Hoboken, NJ, USA: Wiley, 2016.

[54] D. Liu and D. Burbano, “Collective intermittent exploration in fish schools is mediated by visual cues,” *Roy. Soc. Open Sci.*, vol. 12, no. 6, Jun. 2025, Art. no. 250033.

[55] D. Noble, “The role of stochasticity in biological communication processes,” *Prog. Biophys. Mol. Biol.*, vol. 162, pp. 122–128, Jul. 2021.

[56] J. Gautrais et al., “Analyzing fish movement as a persistent turning Walker,” *J. Math. Biol.*, vol. 58, no. 3, pp. 429–445, Mar. 2009.

[57] D. Liu, M. Tuqan, and D. Burbano, “Learning to hunt: A data-driven stochastic feedback control model of predator–prey interactions,” *J. Theor. Biol.*, vol. 599, Feb. 2025, Art. no. 112021.

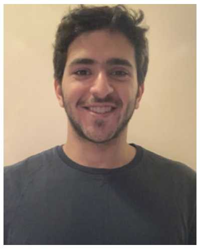
[58] B. Kouvaritakis and M. Cannon, *Model Predictive Control*. Cham, Switzerland: Springer, 2016.

Akhil Sankar is currently pursuing the M.S. degree with the Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ, USA, specializing in machine learning.

From January to December 2023, he worked as a Research Assistant with the Swarm Intelligence Laboratory, Rutgers University. His research interests include control systems and machine learning in the context of autonomous systems.

Cy Westbrook is currently pursuing the B.S. degree in electrical engineering and computer science with the School of Engineering, Rutgers University, Piscataway, NJ, USA.

He has been working as a Research Assistant at the Swarm Intelligence Laboratory, Swarm Intelligence, since the Fall of 2023, developing hardware and software tools to enable methods the laboratory is pursuing. His interests include embedded systems, power conversion, audio, wireless technology, and user interfaces.



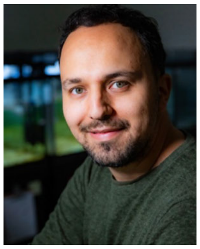
Mohammad Tuqan is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ, USA.

He is a Researcher at the Swarm Intelligence Laboratory, Rutgers University. His research focuses on control and network theory. He is particularly interested in multiagent systems, particularly in the context of collective and social behavior.

Guosong Yang (Member, IEEE) received the B.Eng. degree in electronic engineering from The Hong Kong University of Science and Technology, Hong Kong, in 2011, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Illinois from Urbana–Champaign, Urbana, IL, USA, in 2013 and 2017, respectively.

He was a Post-Doctoral Scholar at the University of California, Santa Barbara, Santa Barbara, CA, USA, from 2017 to 2022, and a Post-Doctoral Research Associate at the University of Illinois Urbana–Champaign, in 2022. He is currently an Assistant Professor at the Department of Electrical and Computer Engineering at Rutgers University, Piscataway, NJ, USA. His research interests span systems and control theory, optimization, and game theory, with emphasis on switched and hybrid systems, information-constrained control, nonconvex optimization, game-theoretic learning, and applications to cyber-physical systems and system resilience and security.

Dr. Yang has won the ACM SIGBED HSCC Best Paper Award in 2019, for his work.



Daniel Burbano (Senior Member, IEEE) received the B.Sc. degree in electronic engineering and the M.Sc. degree in industrial automation from the National University of Colombia, Manizales, Colombia, in 2010 and 2012, respectively, and the Ph.D. degree in control engineering from the University of Naples Federico II, Naples, Italy, in 2015.

From 2017 to 2018, he was a Post-Doctoral Researcher at Northwestern University, Evanston, IL, USA, working on network inference. From 2019 to 2022, he was a Faculty Fellow at New York University, New York, NY, USA, working on animal behavior. He is currently an Assistant Professor at the Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ, USA. His research interests encompass the analysis and control of complex systems and networks, with a focus on data-driven modeling, network inference, and collective decision-making in animal groups.

Dr. Burbano is a recipient of the NSF CAREER Award. He serves as a Senior Associate Editor for IEEE Transactions on Circuits and Systems II (TCAS II) and as an Editor for *PLoS Complex Systems*.