

RESEARCH ARTICLE OPEN ACCESS

Model Predictive Control for Magnetically Actuated Cellbots

Mehdi Kermanshah¹  | Logan E. Beaver²  | Max Sokolich³  | Fatma Ceren Kirmizitas³  | Ahmad Ahmad⁴  | Roberto Tron¹  | Ron Weiss⁵  | Sambeeta Das³  | Calin Belta⁶

¹Department of Mechanical Engineering, Boston University, Boston, Massachusetts, USA | ²Department of Mechanical and Aerospace Engineering, Old Dominion University, Norfolk, Virginia, USA | ³Department of Mechanical Engineering, University of Delaware, Newark, Delaware, USA | ⁴Department of System Engineering, Boston University, Boston, Massachusetts, USA | ⁵Department of Biological Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA | ⁶Department of Electrical and Computer Engineering, University of Maryland, College Park, Maryland, USA

Correspondence: Mehdi Kermanshah (mker@bu.edu)

Received: 9 December 2025 | **Revised:** 16 August 2026 | **Accepted:** 20 August 2026

Keywords: Gaussian process | magnetic actuation | medical application | microrobots | model predictive controller

ABSTRACT

This article presents a control strategy for, which are magnetically actuated, micron-scale robots embedded into living cells. Our approach combines model predictive control, which provides low computational complexity and adaptability to changing environments, with Gaussian Processes (GPs), which are used as disturbance estimators for precise path tracking. To address the challenges posed by unmodeled dynamics, we integrate data-driven modeling with model-based control to accurately track desired trajectories using relatively small data. To the best of our knowledge, this is the first work to integrate data-driven modeling with model-based control for magnetically actuated cellbots. The GP effectively learns and predicts unmodeled disturbances, providing uncertainty bounds as well. The experimental results show that our controllers perform very well, even in environments cluttered with cells.

1 | Introduction

The use of micron-scale robots (μ bots) for medical applications is an active area of research. Recent works have demonstrated their potential in drug delivery [1, 2], biopsy [3], microsurgery [4], and cellular manipulation [5–8]. Among these, magnetically actuated μ bots are particularly promising due to their biocompatibility and the established use of magnetic fields in clinical settings [9]. Precise motion control for magnetically actuated μ bots in the presence of noise and disturbances, especially in cluttered environments with obstacles, such as biological cells, is a significant challenge [10].

Cellbots are made by embedding magnetic μ bots into living cells (see Figure 1a). Cellbots have been shown to be particularly useful for cell delivery and manipulation [11] and hold the potential to significantly advance the field of organoid development [12]. However, due to the variability of numbers and positions of μ bots

inside the cells, controlling cellbots is a very challenging problem—this is our focus in this article.

Unlike rigid μ bots, whose physical properties are consistent across units, each cellbot is unique due to the stochastic nature of bead endocytosis. The number, size, and spatial arrangement of internalized μ bots vary from cell to cell, resulting in different dynamic behaviors. This means that a model trained for one cellbot cannot be directly reused for another, and the control framework must be able to rapidly adapt to each individual unit using limited data. Furthermore, the biological environment introduces disturbances such as cell membrane deformation, cytoplasmic drag, and interactions with neighboring cells that are absent in standard μ bot settings.

Several model-based approaches have been developed for path tracking using μ bots [13]. Examples include linear and non-linear control methods, involving disturbance observers such

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2026 The Author(s). *Advanced Robotics Research* published by Wiley-VCH GmbH.

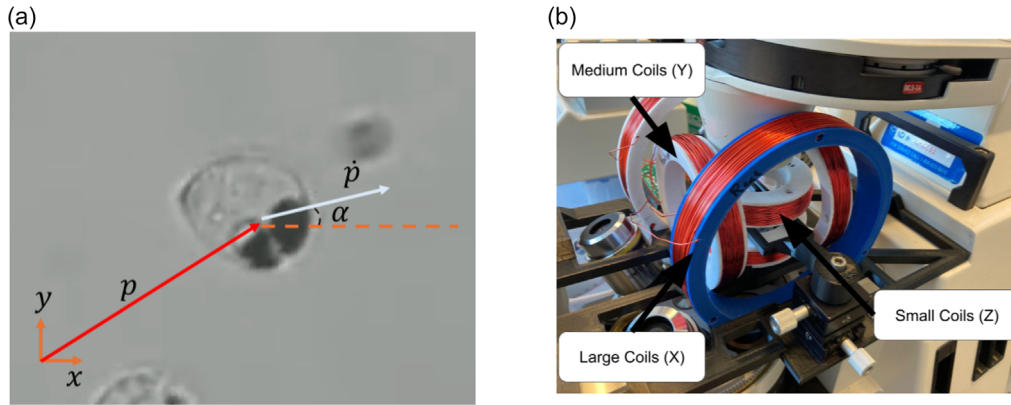


FIGURE 1 | (a) Microscope photo of a cellbot, showing μ bots (black disks) inside a cell. (b) The experimental setup used for cellbot control.

as high-gain and extended state observers [14]. Robust control techniques have also been employed to handle uncertainties in the system dynamics [15, 16]. model predictive control (MPC) has been implemented to accurately track desired paths while minimizing control effort and ensuring adherence to output constraints [13, 17–19]. However, the performance of MPC depends heavily on the accuracy of the system model and the estimation of disturbances acting on the μ bot. Traditional disturbance observers have been combined with MPC [18, 20] to improve model accuracy. These approaches are limited in their ability to capture more complex dynamics. Alternatively, data-driven methods, such as radial basis function (RBF) neural networks [16], have been used to model disturbances and unknown dynamics. In our previous study [21], we demonstrated that the Gaussian Process (GP) is a more suitable choice as a disturbance estimator.

The works mentioned above apply solely to μ bots and are not suited to deal with the more complex dynamics of cellbots. Related studies exploring μ bots embedded within living cells [22, 23] utilize different robot designs [24, 25] and actuation mechanisms [26]. An earlier study used a rolling magnetic robot within a cell. However, it was limited to simple scenarios where the cellbot reached a target location in a straight line at a constant speed [27]. Among works that combine data-driven modeling and model-based control, the authors of [28] propose a framework that combines extreme learning machine and RBF as disturbance estimator with sliding mode control for magnetically actuated helical microswimmers. Although their approach achieves accurate path tracking, it is not directly applicable to our setting as it requires more training data than we can collect for each individual cellbot.

In this article, we propose a framework for precise path tracking of magnetically actuated cellbots. We build on our previous work on data-driven μ bot modeling [29] by developing MPC schemes and estimators for dynamic disturbances using GP. A limitation of our previous approach is the initial data collection phase, which poses a collision risk in highly cluttered environments due to model inaccuracies. In this article, by incorporating disturbance estimation into the MPC framework, we combine the advantages of both data-driven and model-based methods for accurate path tracking. GPs are particularly well suited for this task due to their ability to learn quickly from small datasets,

allowing for individualized retraining of models for each cellbot. Additionally, the confidence intervals provided by GPs can be further utilized to ensure system safety under dynamic uncertainty. We conduct experiments with cellbots to demonstrate the effectiveness of our approach in real-world biological scenarios. While similar approaches have been used for larger-scale robotic systems, to the best of our knowledge, this is the first approach to utilize MPC enhanced with data-driven modeling for cellbot control.

The remainder of this article is organized as follows. In Section 2, we provide a brief overview of GPs. The problem formulation and an outline of our approach are given in Section 3. Our technical solution is presented in Section 4. Section 5 shows the experimental results with a cellbot. Finally, in Section 6, we conclude with discussions and potential directions for future work.

2 | Mathematical Preliminaries

In this section, we provide some background on GPs. Following [30], a GP is a stochastic process that generalizes the concept of a Gaussian distribution to a function space. Formally, a GP is an infinite collection of random variables, any finite number of which have a joint Gaussian distribution. A GP $y(\mathbf{x})$ is completely described by its *mean* $m(\mathbf{x})$ and *covariance* $k(\mathbf{x}, \mathbf{x}')$ functions. We denote a GP using the compact form

$$y(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \quad (1)$$

In this article, the feature vector $\mathbf{x}$ corresponds to the state and control variables of the cellbot (described in Section 3), while the function $y(\mathbf{x})$ captures the unmodeled dynamics and stochastic disturbances acting on the cellbot. We consider a zero-mean ($m(\mathbf{x})=0$) GP with noiseless observations, which is not restrictive in general [30].

The shape of the GP is completely defined by the covariance function $k(\mathbf{x}, \mathbf{x}')$, which measures the distance between two features. Furthermore, we can construct a covariance matrix K , where each entry $k_{ij}=k(\mathbf{x}_i, \mathbf{x}_j)$ and i, j index the observations. Finally, by construction, we can predict the probability distribution of any finite number of disturbances $\mathbf{y}_*$ from predicted

values of $\mathbf{x}_*$ given finitely many noiseless observations of $\mathbf{x}$ and $\mathbf{y}(\mathbf{x})$, i.e.,

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix} \sim \mathcal{N} \left(\mathbf{0}, \begin{bmatrix} K(\mathbf{x}, \mathbf{x}) & K(\mathbf{x}, \mathbf{x}_*) \\ K(\mathbf{x}_*, \mathbf{x}) & K(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix} \right) \quad (2)$$

The distribution of $\mathbf{y}_*$ conditioned on our previous observations and $\mathbf{x}_*$ is

$$\mathbb{P}(\mathbf{y}_* | \mathbf{x}_*, \mathbf{x}, \mathbf{y}) = \mathcal{N}(K(\mathbf{x}_*, \mathbf{x})K(\mathbf{x}, \mathbf{x})^{-1}\mathbf{y}, K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, \mathbf{x})K(\mathbf{x}, \mathbf{x})^{-1}K(\mathbf{x}, \mathbf{x}_*)) \quad (3)$$

which is a probability distribution over the unmodeled dynamics and disturbances $\mathbf{y}_*$ at a predicted value of $\mathbf{x}_*$ based on historical observations of $\mathbf{y}, \mathbf{x}$.

3 | Problem Formulation and Approach

This work focuses on steering a cellbot to track a desired trajectory by manipulating a magnetic field using an MPC framework. The experimental setup is shown in Figure 1b (see Section 5.1 for details). An example of a cellbot is shown in Figure 1a. A cellbot is created by injecting magnetically actuated μ bots into the cell. These robots are controlled entirely through an external magnetic field. The velocity magnitude of each μ bot is determined by the oscillation frequency of the magnetic field, denoted by f , and its heading (direction of motion) is determined by the orientation of the magnetic field, denoted by α . We make the following assumption:

Assumption 1. *The cells are mechanically coupled with the inserted μ bots.*

Under the above assumption, a cellbot behaves as a rigid body with a heading, which is determined by the arrangement of the μ bots inside the cell.

Based on our experimental results and earlier studies [21], we also make the following assumption:

Assumption 2. *Cellbots align instantaneously and perfectly with the magnetic field.*

The embedded μ bots are polystyrene microspheres with a thin, half-coated nickel shell. The nickel layer is a thin film deposited on only one hemisphere. Note that the total nickel content is well below the 3%–10% weight percent threshold required for a nickel–polystyrene composite to exhibit ferromagnetic behavior. Consequently, the particles are paramagnetic and magnetize only in the presence of an applied field. Reported saturation magnetization values for nickel-coated polystyrene particles range from approximately 1.8 to 8.9 emu/g depending on shell thickness and chemistry [31, 32]. Since the μ bots carry no remanent magnetization, their magnetic moment aligns with the external field direction at each instant.

Based on the above assumptions, we model the cellbot as a unicycle, in which the heading angle can be instantaneously controlled, and which is subject to a generalized disturbance [17]

$$\dot{\mathbf{p}} = a_0 f(t) \begin{bmatrix} \sin(\alpha(t)) \\ \cos(\alpha(t)) \end{bmatrix} + \mathbf{D}(t) \quad (4)$$

where $\mathbf{p} \in \mathbb{R}^2$, $a_0 \in \mathbb{R}_{\geq 0}$, denote the position of the center of the cellbot and effective radius of rotation, respectively. The term $\mathbf{D} \in \mathbb{R}^2$ models the disturbance, capturing unmodeled dynamics such as Brownian motion, which is prevalent at the microscale. Essentially, $a_0 f$ determines the forward speed and α sets the direction. In (4), the control inputs are f and α . Using Euler integration, a discrete-time version of the dynamics can be written as

$$\mathbf{p}_{t+1} = \mathbf{p}_t + \left(a_0 f_t \begin{bmatrix} \sin \alpha_t \\ \cos \alpha_t \end{bmatrix} + \mathbf{D}_t \right) \Delta t \quad (5)$$

where $\mathbf{p}_t$, f_t , α_t , and Δt denote the position, frequency, heading angle, and time-step size, respectively.

Problem 1. *Given a cellbot with dynamics described by Equation (4), a reference trajectory described as a finite set of waypoints $\mathbf{r}_{0:T-1} = \{\mathbf{r}_0, \mathbf{r}_1, \dots, \mathbf{r}_{T-1}\}$, where $\mathbf{r}_t \in \mathbb{R}^2$ denotes the waypoint at (discrete) time t , $t = 0, \dots, T-1$ for a given time horizon T . Our goal is to minimize the following cost function*

$$\begin{aligned} \min_{f_{0:T-1}, \alpha_{0:T-1}, \mathbf{p}_{0:T}} \quad & \sum_{t=0}^{T-1} (\gamma^t (\mathbf{p}_t - \mathbf{r}_t)^T Q (\mathbf{p}_t - \mathbf{r}_t) + R f_t^2) \\ \text{subject to} \quad & 0 \leq f_t \leq f_{\max}, \\ & \mathbf{p}_{t+1} = \mathbf{p}_t + \left(a_0 f_t [\sin \alpha_t, \cos \alpha_t]^T + \mathbf{D}_t \right) \Delta t \end{aligned} \quad (6)$$

In other words, we aim for the robot to follow $\mathbf{r}_t$ while regulating the frequency and adhering to the dynamics in (5).

In (6), $Q \in \mathbb{R}^{2 \times 2}$ and $R \in \mathbb{R}$ denote the weighting matrix and positive scalar, respectively, that balance tracking accuracy against frequency regulation. The term $f_{\max}$ specifies the maximum allowable frequency. We introduce a discount factor γ to reduce the influence of future predictions on current actions as these predictions become increasingly uncertain due to system stochasticity. Finally, in (6), we penalize f to encourage path tracking at lower frequencies, thereby avoiding the excitation of high-frequency disturbances.

We propose the following assumption to simplify the disturbance estimation:

Assumption 3. *The environmental disturbance $\mathbf{D}$ and any error in our model of the true dynamics (4) are isotropic, i.e., they are not a function of the cellbot's position $\mathbf{p}$.*

This assumption implies that the disturbance depends only on the magnitude and direction of the cellbots' velocity, suggesting that the dominant disturbance terms are drag forces. Our results in this article, along with our earlier work [29], support this assumption. While it is possible to model the disturbance as a function of additional variables (e.g., position), doing so would significantly increase the state-space dimension and require more extensive data collection. We model the disturbance as a function of frequency and heading angle (i.e., $\mathbf{D}(\alpha, f)$).

Our proposed approach is shown in Figure 2. First, during the initial learning phase, we collect data consisting of the pairs

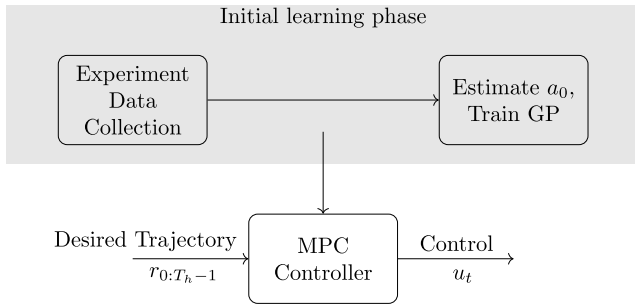


FIGURE 2 | Overview of the control framework: Initially, experimental data are collected offline to estimate $\hat{a}_0$ and train the GP for $\hat{\mathbf{D}}$. These are then used in an MPC framework to model the system dynamics.

$(\dot{\mathbf{p}}, f, \alpha)$. Using these data, we estimate a_0 in the dynamics model (4) through linear regression. Next, we utilize the GP to learn the error between the unicycle model and the collected data. MPC is then adopted to derive the control action, minimizing the cost function (6) over the time horizon T using the GP-enhanced model.

To validate our approach, we consider two case studies:

1. **Obstacle-free environment:** We evaluate the algorithm's performance in an open space by tracking a Lemniscate of Bernoulli trajectory.
2. **Cluttered environment:** We evaluate the performance in a more realistic setting by considering a cluttered environment with real cells. A motion planner is used to generate an obstacle-free path tracked by our controller.

4 | Model Predictive Control With Gaussian Process Disturbance Estimator

4.1 | Disturbance Estimation With Gaussian Process

The system dynamics (4) is nonlinear in α , which makes the above optimization problem nonconvex. To reduce computational complexity, we transform the system from polar to Cartesian coordinates using $\mathbf{u} = [u_x, u_y]^T = [f \cos(\alpha), f \sin(\alpha)]^T$

$$\dot{\mathbf{p}} = a_0 \begin{bmatrix} u_x \\ u_y \end{bmatrix} + \mathbf{D}(\alpha, f) = a_0 \mathbf{u} + \mathbf{D}(\alpha, f) \quad (7)$$

Note that in (7), $\mathbf{u}$ can be mapped to f and α as

$$f = \sqrt{\mathbf{u}^T \mathbf{u}}, \quad \alpha = \text{atan2}(u_y, u_x) \quad (8)$$

According to Assumption 2, the coordinate transformation does not introduce any kinematic constraints (like the minimum turning radius) since the electrical system for varying the magnetic field responds instantaneously.

In this section, we describe our approach for estimating $\mathbf{D}$ and a_0 . We train the GP during an initial learning phase, where the cellbot is given a sequence of control inputs, either from a human operator or a predefined path. The learning phase occurs in the same environment and with the same robot as the experiment;

thus, the training and testing environments are consistent. Position data ($\mathcal{P} = \{[p_x(t_k), p_y(t_k)]^T\}$) and control action data ($\mathcal{X} = \{(\alpha(t_k), f(t_k))\}$) are recorded at discrete intervals. We derive the cellbot velocity ($\mathcal{V} = \{[v_x(t_k), v_y(t_k)]^T\}$) by numerical differentiation and low-pass filtering of $\mathcal{P}$. Based on the dynamics, we do a linear regression separately for the x and y axes

$$\begin{aligned} v_x &= \hat{a}_{0,x} u_x + \hat{\mathbf{D}}_{c,x}, \\ v_y &= \hat{a}_{0,y} u_y + \hat{\mathbf{D}}_{c,y} \end{aligned} \quad (9)$$

Note that in (9), we assumed that $\hat{\mathbf{D}}$ is constant and $\mathbf{D}_c = [D_{c,x}, D_{c,y}]^T$ can be interpreted as the mean disturbance value, as estimated by the linear regression model. We use linear regression to estimate a_0 by $\hat{a}_0$ and utilize the GP to estimate the model mismatch. Based on (9), we estimate a_0 by

$$\hat{a}_0 = \frac{1}{\sqrt{2}} \sqrt{\hat{a}_{0,x}^2 + \hat{a}_{0,y}^2} \quad (10)$$

Based on this estimation, we can rewrite the dynamics as follows

$$\dot{\mathbf{p}} = \hat{a}_0 \mathbf{u} + \mathbf{v}_e \quad (11)$$

We estimate $\mathbf{v}_e$, the model mismatch, from data using a GP. The model mismatch $\mathbf{v}_e$ consists of two components: The first is the model uncertainty $\mathbf{D}$ in (4), and the second is the error in estimating a_0 . Consequently, compared to the original dynamics, the model mismatch is expressed as $\mathbf{v}_e = (a_0 - \hat{a}_0) \mathbf{u} + \mathbf{D}$. The GP essentially maps $\mathcal{X}$ to the observed error velocities denoted by $\mathcal{V} = \{(\mathbf{v}(t_k) - \hat{a}_0 \mathbf{u}(t_k))\}$.

We implement this approach similarly to [21], utilizing a linear combination of the RBF kernel and a constant-scaled white noise kernel, which effectively captures the velocity error. Our kernel function can be formally expressed as

$$K(\mathbf{x}, \mathbf{x}') = C \left(\exp \left(- \frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma} \right) + \eta \right) \quad (12)$$

where σ and C are length scale and constant scale hyperparameters, and η is drawn from a normal distribution with zero mean and variance as another hyperparameter.

4.2 | Model Predictive Control

To generate trajectories, we discretize the dynamics using Euler's method

$$\mathbf{p}_{t+1} = \mathbf{p}_t + \hat{a}_0 \Delta t \mathbf{u}_t + \hat{\mathbf{v}}_e \Delta t \quad (13)$$

where $\mathbf{p}_t$, $\mathbf{u}_t$, $\hat{\mathbf{v}}_e$, and Δt are the position, control, estimation of the model mismatch by the GP, and time step, respectively.

By rewriting (6) for the discretized system, the control action at each time step is derived from the following optimization problem

$$\begin{aligned} \min_{\mathbf{u}_{0:T_h-1}, \mathbf{p}_{0:T_h-1}} \quad & \sum_{t=0}^{T_h-1} [\gamma^t (\mathbf{p}_t - \mathbf{r}_t)^T Q (\mathbf{p}_t - \mathbf{r}_t) + \mathbf{u}_t^T R \mathbf{u}_t] \\ \text{subject to} \quad & \\ \mathbf{p}_t = & \mathbf{p}_{t-1} + \hat{a}_0 \Delta t \mathbf{u}_t + \hat{\mathbf{v}}_e \Delta t, \quad t \in [0, T_h - 1] \\ \mathbf{u}_{\min} \leq & \mathbf{u}_t \leq \mathbf{u}_{\max}, \quad t \in [0, T_h - 1] \end{aligned} \quad (14)$$

We estimate the disturbance using GP based on the current values of f and α , and keep the estimated disturbance $\hat{\mathbf{v}}_e$ constant throughout the predictive horizon. This approach simplifies the optimization into a quadratic problem. However, this simplification reduces the accuracy of long-term prediction. The second constraint captures the limit on f . We use this to define the range within which the GP is trained to estimate disturbances.

5 | Results

5.1 | Experimental Setup

To form cellbots, we embed rolling microrobots [29] in Chinese Hamster Ovary (CHO) cells. The cell culture components are purchased from GIBCO. CHO cells are cultured in Dulbecco's Modified Essential Medium-F12 (DMEM-F12) with 10% fetal bovine serum, 1% penicillin/streptomycin, 5% CO₂, and incubated at 37 °C. After washing with Dulbecco's phosphate-buffered saline and trypsinizing to detach, cells from the third passage are used. CHO cells are seeded into a 35-mm Petri dish at a density of 1×10^5 cells/dish and incubated for 24 h. Microrobots are dispersed in the culture media and added to the Petri dish. After incubation, cells are washed, trypsinized, and centrifuged to collect the cellbots. To prevent the cellbots from sticking to the surface, they are suspended in a 0.1% sodium dodecyl sulfate solution instead of DI water. Although this increases viscosity and reduces rolling speed, it significantly reduces surface adhesion.

In order to generate the magnetic fields necessary to actuate the cellbot, six Helmholtz coils are designed and arranged in parallel pairs, as shown in Figure 1b. The coils are mounted on a Zeiss Axiovert 100 inverted microscope. To power the coils, we use an Arduino Mega microcontroller connected to a Jetson Xavier NX single board A custom tracking and control program is written in Python to read incoming images from a FLIR BFS-U3-28S5M-C USB 3.1 Blackfly S Monochrome Camera. The continuous stream of images is analyzed in Python's OpenCV library, which is used to extract position and velocity data for the cellbot.

These action/input commands are converted to a 3D rotating magnetic field. Because current is proportional to the magnetic field generated from electromagnets, the magnetic field $\mathbf{B} = [B_x, B_y, B_z]^T$ is mapped to the heading, attitude, and frequency signals via

$$\mathbf{B} = \begin{bmatrix} \cos(\gamma) \cos(\alpha) \cos(2\pi f t) + \sin(\alpha) \sin(2\pi f t) \\ -\cos(\gamma) \sin(\alpha) \cos(2\pi f t) + \cos(\alpha) \sin(2\pi f t) \\ \sin(\gamma) \cos(2\pi f t) \end{bmatrix}$$

where $\gamma \in [-\pi, \pi]$ is a fixed attitude. The angles and coordinate system are shown in Figure 3.

5.2 | Experiment

First, we collect data during an initial training phase to learn the disturbance experienced by the robot in the environment. We collect experimental data within the frequency range $0 \leq f \leq 5$ Hz with increments of 1 Hz, and angular range $0 \leq \alpha \leq 2\pi$ with increments of 1°. We call this circular training dataset since when

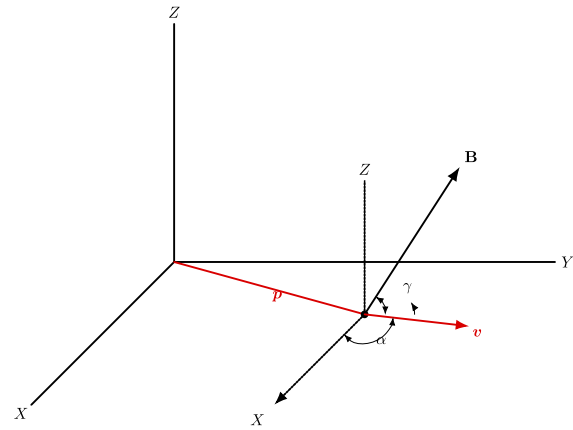


FIGURE 3 | Schematic illustrating the magnetic field $\mathbf{B}$ with angles α and γ in XYZ coordinates.

the frequency is fixed, the robot moves in a circular path, and the radius of the circle changes with the frequency. The videos of the experiments, along with Supporting Information, are available online.¹ Note that we perform training and testing in the same environment. Thus, we do not consider issues that may arise from policy transfer or environmental inconsistency.

We train two separate GPs to estimate the model mismatch in the x and y directions, both using the kernel function defined in (12). To evaluate the effectiveness of our approach, we compared the GP-enhanced model against the actual collected data and a baseline linear model without the GP disturbance estimator, as shown in Figure 4. The GP models effectively captured disturbance behavior, leading to a significant enhancement in the accuracy of the linear model.

In our experiments, we set $Q = I$, $R = 0.01 \times I$, $\Delta t = 0.1$, and a prediction horizon $T_h = 4$, where I denotes the identity matrix. The desired trajectory for this case study is a Lemniscate of Bernoulli path, as shown in Figure 5a. The desired and actual trajectories are represented by gray and red curves, respectively. Our approach successfully tracks the desired trajectory with high accuracy. In the same figure, we compare our method with a naive MPC approach that does not utilize the GP disturbance estimator. As shown, our approach tracks the desired trajectory more smoothly and more accurately. It achieves a 14% reduction in the 2-norm mean tracking error and a 28% reduction in the maximum deviation (2-norm error) from the desired trajectory.

We also plot our model's predictions in Figure 6a. In some areas, the GP disturbance prediction closely aligns with the linear model. This indicates that the MPC controller drives the system into α and f pairs that were not observed in the training data due to sampling only integer frequencies in the circular training dataset.

5.3 | Iterative Learning

To improve tracking accuracy, we collect additional data (pairs of $(\alpha, f, \hat{p})$) through an iterative process [33]. We use MPC with a GP disturbance estimator to track the desired trajectory, which in our case study is the Lemniscate of Bernoulli. All data points not observed in the initial circular dataset are then augmented

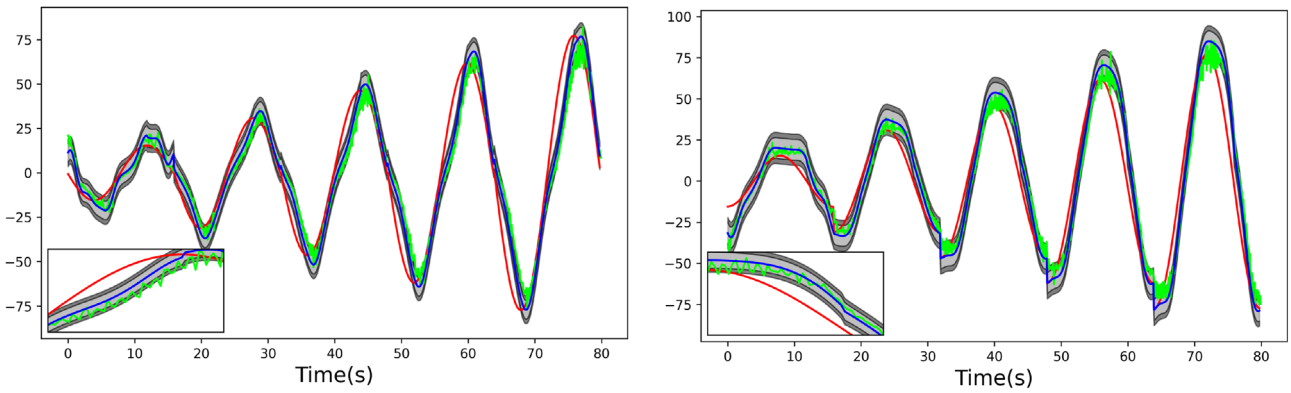


FIGURE 4 | Comparison between the learned model, the experimental data, and the linear model with GP disturbance. The left and right panels correspond to the x - and y -directional velocity components, respectively. The green curves represent the experimentally measured velocities (in $\mu\text{m/s}$). The red and blue curves correspond to the linear model with the GP disturbance estimator and the GP estimator, respectively. The light and dark gray-shaded regions indicate the uncertainty bounds corresponding to one (65%) and two (95%) standard deviations around the model mean. The inset plots provide a magnified view of the time interval between 40 and 45 s, presenting a more detailed comparison of the model performance against the experimental data within this range.

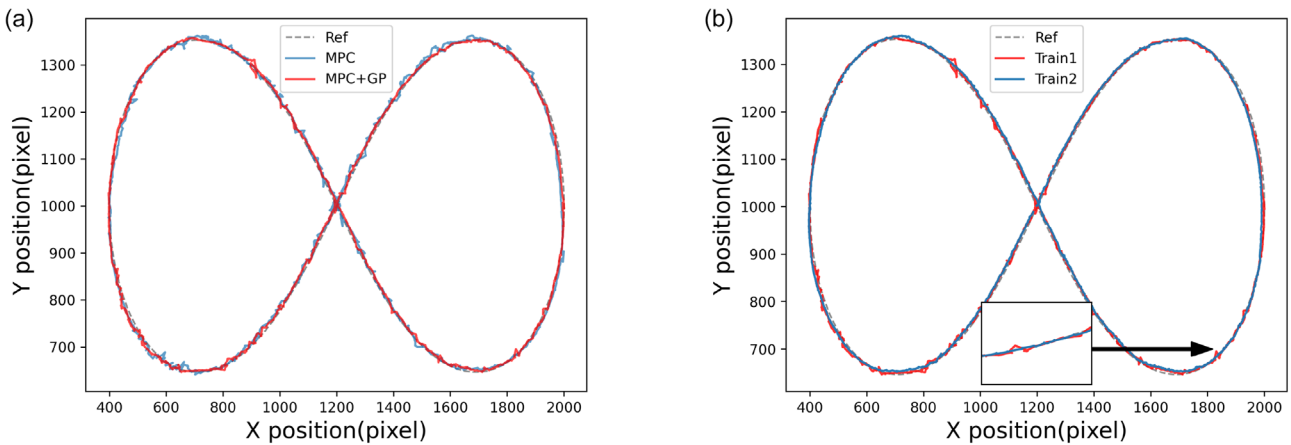


FIGURE 5 | Trajectories after circular (a) and iterative (b) training. In (b), the inset zoom in on the lower left shows a closer look at the differences in the trajectories.

in the new training dataset. This process can be repeated to explore the control space more effectively. We found that performing this process just once significantly improves both our model's performance and tracking accuracy, as shown in Figure 6b. Note that circular training alone provides sufficient accuracy for navigation in the environment. However, incorporating this iterative process further improves accuracy, enabling the cellbots to more reliably track the desired trajectory.

In Figure 5b, the trajectories after the first and iterative training are shown in red and blue, respectively. The snapshots of the real system are shown in Figure 7. After iterative training, our approach tracks the desired trajectory more accurately, with reduced deviation. To evaluate the performance of the GP, we compare the mean absolute error (MAE) of the learned model with that of the linear model, as shown in Table 1. After iterative training, the GP reduces the modeling error by approximately 60%. Additionally, we present the output of our learned model with confidence intervals alongside the actual velocity in Figure 6a,b, corresponding to the first training and after iterative training, respectively. After the final training, the learned model closely predicts the observed velocity. The actual velocity is

within one standard deviation 56% of the time and within two standard deviations 86% of the time. Based on this, we can conclude that the Gaussian distribution is a close approximation for D , but it tends to under-approximate (e.g., 86% vs. 96% for ± 2 standard deviations). These results demonstrate the effectiveness of the training dataset and indicate that the disturbances experienced by the cellbots during testing closely resemble those encountered in the training data.

To demonstrate the effectiveness of our approach in real biomedical scenarios, we evaluated its performance in a cluttered environment with other CHO cells acting as obstacles. We use RRT* [34] for path planning to generate a reference trajectory, which our controller then follows. The positions of the obstacle cells and the cellbot are extracted using OpenCV. As shown in Figure 8, our controller accurately tracks the reference trajectory with a mean error of $6\mu\text{m}$ while successfully avoiding obstacles.

Although the confidence bounds provided by GPs have not yet been integrated into our framework, it offers a potential avenue to be combined in robust MPC frameworks such as robust [35] or tube-based [36] MPC.

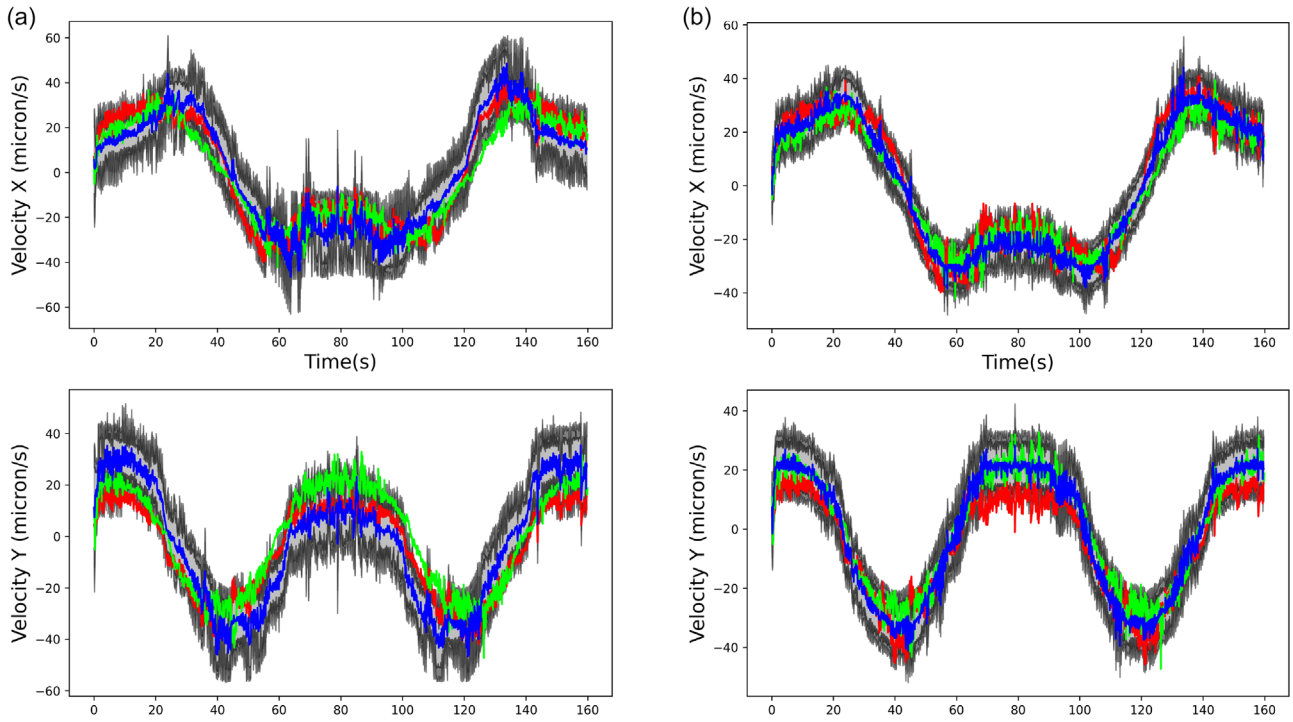


FIGURE 6 | Model predictions after the circular (a) and iterative training (b), respectively. The top and bottom graphs correspond to the x and y directions, respectively. Iterative training noticeably enhances the model's accuracy. We used the same color coding as in Figure 4.

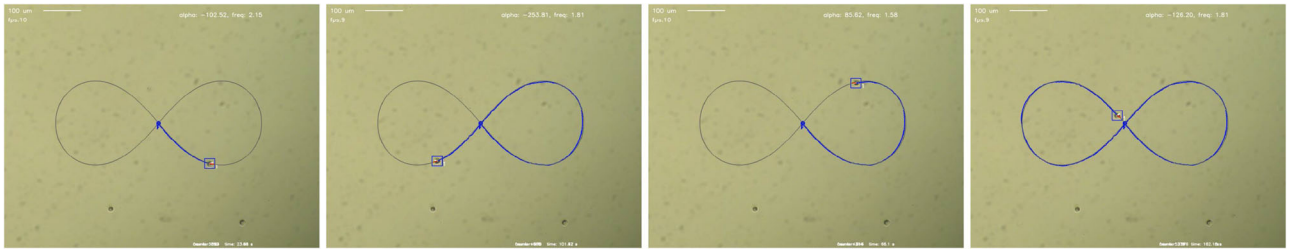


FIGURE 7 | Snapshots from an experiment showing the cellbot tracking a Lemniscate of Bernoulli trajectory. The square highlights the cellbot suspended in the aqueous solution. The history of the cellbot's movement is overlaid. The blue and gray curves represent the actual and desired trajectories, respectively.

TABLE 1 | MAEs for the cellbot experiments and comparison of improvements relative to the baseline (MPC with GP disturbance estimator). The numbers in parentheses indicate the percentage improvement over the baseline.

Axis	MPC (baseline)	Circular training	Iterative training
x axis	$9.64 \mu\text{m}\cdot\text{s}^{-1}$	$7.93 \mu\text{m}\cdot\text{s}^{-1}$ (18%)	$3.84 \mu\text{m}\cdot\text{s}^{-1}$ (60%)
y axis	$10.18 \mu\text{m}\cdot\text{s}^{-1}$	$10.07 \mu\text{m}\cdot\text{s}^{-1}$ (1%)	$3.93 \mu\text{m}\cdot\text{s}^{-1}$ (61%)

5.4 | Controller Robustness to Cellbot Configuration Variability

Since the endocytosis of μ bots is inherently stochastic, each cellbot differs in the number and arrangement of internalized μ bots, which in turn affects both a_0 and $D(a, f)$ in our model (4). Our framework addresses this by treating each cellbot as an individual system: The linear regression in Equations (9)–(11) estimates

$\hat{a}_0$, and the GP learns $\hat{\mathbf{v}}_e$ from a short circular training sequence, absorbing configuration-specific dynamics into the learned model. The sample efficiency of GPs makes this per-cellbot calibration practical, unlike methods requiring large training datasets for deep neural networks [28].

To empirically support this claim, we analyze data collected from three different cellbots used throughout the course of this study. These cellbots are prepared in independent experiments and exhibit visibly different μ bot loadings under the microscope. For each cellbot, we execute the same training protocol described. Table 2 reports the estimated $\hat{a}_0$ and the resulting trajectory tracking error of Lemniscate of Bernoulli for each cellbot. Despite substantial variation in $\hat{a}_0$ across cellbots, the GP-augmented MPC achieves consistent post-training tracking errors in all cases. This indicates that the framework reliably adapts to the specific dynamics of each cellbot, and that repeatability is achieved not by uniformity of the underlying robots, but by the per-cellbot calibration built into the pipeline.



FIGURE 8 | Experimental results for a cellbot moving in an environment cluttered with other cells. The red curve represents the reference trajectory generated by the RRT* algorithm, while the blue curve shows the actual trajectory generated by our controller.

TABLE 2 | Controller performance across cellbots with different μ bot configurations. Despite the variation in $\hat{a}_0$, the GP-augmented MPC achieves consistent tracking accuracy after per-cellbot calibration.

Cellbot	Approx. # μ bots	$\hat{a}_0$, $\mu\text{m}\cdot\text{s}^{-1}\cdot\text{Hz}^{-1}$	Tracking error, μm
C1	1	11.49	3.23
C2	4	16.84	3.18
C3	3	13.63	2.42

We further note that as the number of internalized μ bots increases or their geometry becomes less symmetric, the variance of the disturbance D increases and may become dependent on internal states that are difficult to measure (e.g., the relative orientation and position of each individual μ bot). Our experiments included cellbots up to four embedded μ bots, and the GP-augmented MPC successfully handled these variations.

6 | Conclusion

This article presents a control framework for cellbots that integrates MPC with GPs for disturbance estimation. Our approach addresses the challenges of controlling a cellbot in the presence of disturbances. By leveraging GP, we obtain accurate disturbance estimates with a relatively small amount of data. The effectiveness of GP allows us to retrain the model for different cellbots without requiring significant computing power. This approach combines the strengths of data-driven modeling with the robustness of model-based control, offering a practical solution for the precise and reliable control of microrobots. Our experimental results demonstrate that the proposed framework effectively mitigates disturbances and improves path tracking accuracy.

Future work will involve using the GP-provided confidence intervals in planning algorithms with tube-based MPC to

enhance robot safety. A key direction is extending our hybrid optimization-and-learning framework to the multi-robot setting, where a shared global magnetic field must coordinate the motion of a team of μ bots. Multi-robot coordination would significantly broaden the applicability of our approach to biomedically relevant tasks such as collective cell delivery and parallel manipulation within organoid cultures.

Acknowledgments

This work was partially supported by the NSF under grants 2219101 and 2422282 and by the AFOSR under grant FA9550-23-1-0529.

Funding

This work was supported by the National Science Foundation (2219101 and 2422282) and Air Force Office of Scientific Research (FA9550-23-1-0529).

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Endnote

¹ <https://sites.google.com/bu.edu/mpcgp>

References

1. M. Sitti, H. Ceylan, W. Hu, et al., "Biomedical Applications of Untethered Mobile Milli/Microrobots," *Proceedings of the IEEE* 103, no. 2 (2015): 205–224, <https://doi.org/10.1109/JPROC.2014.2385105>.
2. J. Troccaz and R. Bogue, "The Development of Medical Microrobots: A Review of Progress," *Industrial Robot: An International Journal* 35 (2008): 294, <https://doi.org/10.1108/01439910810876373>.
3. C. Bárcena, A. K. Sra, and J. Gao, "Applications of Magnetic Nanoparticles in Biomedicine," in *Nanoscale Magnetic Materials and Applications* (Springer US, 2009), 591–626.
4. S. Guo and Q. Pan, "Mechanism and Control of a Novel Type Microrobot for Biomedical Application," in *Proceedings 2007 IEEE International Conference on Robotics and Automation (IEEE, 2007)*, 187–192.
5. M. S. Sakar, E. B. Steager, A. Cowley, V. Kumar, and G. J. Pappas, "Wireless Manipulation of Single Cells Using Magnetic Microtransporters," in *2011 IEEE International Conference on Robotics and Automation (IEEE, 2011)*, 2668–2673.
6. E. W. H. Jager, O. Inganäs, and I. S. Lundström, "Microrobots for Micrometer-Size Objects in Aqueous Media: Potential Tools for Single-Cell Manipulation," *Science* 288, no. 5475 (2000): 2335–2338.
7. S. Kim, F. Qiu, S. Kim, et al., "Fabrication and Characterization of Magnetic Microrobots for Three-Dimensional Cell Culture and Targeted Transportation," *Advanced Materials* 25, no. 41 (2013): 5863–5868, <https://doi.org/10.1002/adma.v25.41>.
8. E. B. Steager, M. Selman Sakar, C. Magee, M. Kennedy, A. Cowley, and V. Kumar, "Automated Biomanipulation of Single Cells Using Magnetic Microrobots," *International Journal of Robotics Research* 32, no. 3 (2013): 346–359, <https://doi.org/10.1177/0278364912472381>.

9. M. Salehizadeh and E. Diller, "Three-Dimensional Independent Control of Multiple Magnetic Microrobots via Inter-Agent Forces," *The International Journal of Robotics Research* 39, no. 12 (2020): 1377–1396, <https://doi.org/10.1177/0278364920933655>.
10. Z. Yang and L. Zhang, "Magnetic Actuation Systems for Miniature Robots: A Review," *Advanced Intelligent Systems* 2 (2020): 2000082, <https://doi.org/10.1002/aisy.v2.9>.
11. Y. Dai, L. Jia, L. Wang, et al., "Magnetically Actuated Cell-Robot System: Precise Control, Manipulation, and Multimode Conversion," *Small* 18, no. 15 (2022): 2105414, <https://doi.org/10.1002/sml.v18.15>.
12. P. Kumar, K. B. Mirza, K. Choudhury, M. Cucchiari, H. Madry, and P. Shukla, "Tissue Regeneration Through Cyber-Physical Systems and Microbots," *Advanced Functional Materials* 31, no. 31 (2021): 2009663, <https://doi.org/10.1002/adfm.v31.31>.
13. J. Jiang, Z. Yang, A. Ferreira, and L. Zhang, "Control and Autonomy of Microrobots: Recent Progress and Perspective," *Advanced Intelligent Systems* 4, no. 5 (2022): 2100279, <https://doi.org/10.1002/aisy.v4.5>.
14. L. Yang, Q. Wang, and L. Zhang, "Model-Free Trajectory Tracking Control of Two-Particle Magnetic Microrobot," *IEEE Transactions on Nanotechnology* 17, no. 4 (2018): 697–700, <https://doi.org/10.1109/TNANO.2018.2815978>.
15. H. Marino, C. Bergeles, and B. J. Nelson, "Robust Electromagnetic Control of Microrobots Under Force and Localization Uncertainties," *IEEE Transactions on Automation Science and Engineering* 11, no. 1 (2014): 310–316, <https://doi.org/10.1109/TASE.2013.2265135>.
16. Y. Liu, H. Wang, and Q. Fan, "Adaptive Learning and Sliding Mode Control for a Magnetic Microrobot Precision Tracking With Uncertainties," *IEEE Robotics and Automation Letters* 8 (2023): 7767–7774, <https://doi.org/10.1109/LRA.2023.3322094>.
17. 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 Transactions on Automation Science and Engineering* 17, no. 1 (2020): 490–501, <https://doi.org/10.1109/TASE.8856>.
18. Z. Yang, L. Yang, and L. Zhang, "Autonomous Navigation of Magnetic Microrobots in a Large Workspace Using Mobile-Coil System," *IEEE/ASME Transactions on Mechatronics* 26, no. 6 (2021): 3163–3174, <https://doi.org/10.1109/TMECH.2021.3054927>.
19. R. Pieters, S. Lombriser, A. Alvarez-Aguirre, and B. J. Nelson, "Model Predictive Control of a Magnetically Guided Rolling Microrobot," *IEEE Robotics and Automation Letters* 1, no. 1 (2016): 455–460, <https://doi.org/10.1109/LRA.2016.2521407>.
20. K. Belharet, D. Folio, and A. Ferreira, "Mri-Based Microrobotic System for the Propulsion and Navigation of Ferromagnetic Microcapsules," *Minimally Invasive Therapy & Allied Technologies* 19, no. 3 (2010): 157–169, <https://doi.org/10.3109/13645706.2010.481402>.
21. L. E. Beaver, B. Wu, S. Das, and A. A. Malikopoulos, "A First-Order Approach to Model Simultaneous Control of Multiple Microrobots," in 2022 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS) (IEEE, 2022), 1–7.
22. B. Chen, H. Sun, J. Zhang, et al., "Cell-Based Micro/Nano-Robots for Biomedical Applications: A Review," *Small* 20, no. 1 (2024): 2304607, <https://doi.org/10.1002/sml.v20.1>.
23. Y. Chen, D. Chen, S. Liang, et al., "Recent Advances in Field-Controlled Micro–Nano Manipulations and Micro–Nano Robots," *Advanced Intelligent Systems* 4, no. 3 (2022): 2100116, <https://doi.org/10.1002/aisy.v4.3>.
24. S. Lee, S. Lee, S. Kim, et al., "Fabrication and Characterization of a Magnetic Drilling Actuator for Navigation in a Three-Dimensional Phantom Vascular Network," *Scientific Reports* 8, no. 1 (2018): 3691, <https://doi.org/10.1038/s41598-018-22110-5>.
25. X. Fan, M. Sun, L. Sun, and H. Xie, "Ferrofluid Droplets as Liquid Microrobots With Multiple Deformabilities," *Advanced Functional Materials* 30, no. 24 (2020): 2000138, <https://doi.org/10.1002/adfm.v30.24>.
26. T. Gwisai, N. Mirkhani, M. G. Christiansen, T. T. Nguyen, V. Ling, and S. Schuerle, "Magnetic Torque-Driven Living Microrobots for Increased Tumor Infiltration," *Science Robotics* 7, no. 71 (2022): eabo0665, <https://doi.org/10.1126/scirobotics.abo0665>.
27. Y. Feng, L. Feng, Y. Dai, et al., "A Novel and Controllable Cell-Based Microrobot in Real Vascular Network for Target Tumor Therapy," in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2020), 2828–2833.
28. J. Liu, X. Wu, C. Huang, et al., "3-D Autonomous Manipulation System of Helical Microswimmers With Online Compensation Update," *IEEE Transactions on Automation Science and Engineering* 18, no. 3 (2021): 1380–1391, <https://doi.org/10.1109/TASE.2020.3006131>.
29. L. E. Beaver, M. Sokolich, S. Alsalehi, R. Weiss, S. Das, and C. Belta, "Learning a Tracking Controller for Rolling μ Bots," *IEEE Robotics and Automation Letters* 9, no. 2 (2024): 1819–1826, <https://doi.org/10.1109/LRA.2024.3350968>.
30. M. Seeger, "Gaussian Processes for Machine Learning," *International Journal of Neural Systems* 14, no. 02 (2004): 69–106.
31. W. Li, T. Qiu, L. Wang, et al., "Preparation and Electromagnetic Properties of Core/Shell Polystyrene@ Polypyrrole@ Nickel Composite Microspheres," *ACS Applied Materials & Interfaces* 5, no. 3 (2013): 883–891, <https://doi.org/10.1021/am302551d>.
32. W. Zhao, Q. Zhang, J. Zhang, J. Gu, and F. Guo, "Preparation and Characterization of Core-Shell PS/Ni Composite Microspheres," *Polymer Composites* 30, no. 8 (2009): 1098–1105, <https://doi.org/10.1002/pc.v30.8>.
33. N. Wagener, C.-A. Cheng, J. Sacks, and B. Boots, "An Online Learning Approach to Model Predictive Control," arXiv preprint arXiv:1902.08967, 2019.
34. S. Karaman and E. Frazzoli, "Sampling-Based Algorithms for Optimal Motion Planning," *International Journal of Robotics Research* 30, no. 7 (2011): 846–894, <https://doi.org/10.1177/0278364911406761>.
35. A. Bemporad and M. Morari, "Robust Model Predictive Control: A Survey," in *Robustness in Identification and Control* (Springer, 2007), 207–226.
36. D. Limón, I. Alvarado, T. Alamo, and E. F. Camacho, "Robust Tube-Based MPC for Tracking of Constrained Linear Systems With Additive Disturbances," *Journal of Process Control* 20, no. 3 (2010): 248–260, <https://doi.org/10.1016/j.jprocont.2009.11.007>.

Supporting Information

Additional supporting information can be found online in the Supporting Information section.