Steering Particles via Micro-actuation of Chemical Gradients using Model Predictive Control

Mark N. McDonald,[1] Cameron K. Peterson,[2] and Douglas R. Tree[1]

[1)]*Department of Chemical Engineering, Brigham Young University*

[2)]*Department of Electrical and Computer Engineering, Brigham Young University*

(Dated: 16 September 2022)

Biological systems rely on chemical gradients to direct motion through both chemotaxis and signalling, but synthetic approaches for doing the same are still relatively naive. Consequently, we consider here a novel method for using chemical gradients to manipulate the position and velocity of colloidal particles in a microfluidic device. Specifically, we show that a set of spatially localized chemical reactions that are sufficiently controllable can be used to steer colloidal particles via diffusiophoresis along an arbitrary trajectory. To do so, we develop a control method for steering colloidal particles with chemical gradients using nonlinear model predictive control with a model based on the unsteady Green's function solution of the diffusion equation. We illustrate the effectiveness of our approach using Brownian dynamics simulations that steer single particles along paths that include a circle, square, and figure-eight. We subsequently compare our results with published techniques for steering colloids using electric fields, and we provide an analysis of the physical parameter space where our approach is useful. Based on the above, we conclude that it is theoretically possible to explicitly steer particles via chemical gradients in a microfluidics paradigm.

I. INTRODUCTION

Positioning colloidal particles in precisely defined locations is necessary for micro- and nano-system engineering. For example, microfluidic particle positioning has been used for nanophotonics,[1] tissue engineering,[2,3] bio-hybrid technologies, and man-made systems that incorporate living components.[4] Such precise positioning can be achieved for a small number of particles using a top-down approach by applying external forces or fields to the system. In particular, there has been significant progress on using electric fields to move colloidal particles into a desired formation.[1,5–13]

Driving the motion and assembly of larger numbers of particles usually requires a bottom-up approach that leverages the local interactions between particles. The reigning paradigm for bottom-up assembly employs colloids with static, predetermined local interactions.[14] Such static local interactions mean that the explicit and dynamic positioning of top-down methods is lost.

Unlike these static bottom-up methods, microscale biological systems use internal feedback in the form of chemical signals generated from networks of chemical reactions to drive motion and assembly.[15] Accordingly, researchers have suggested an alternative bottom-up assembly paradigm based on particles that dynamically respond to their local environment to cooperatively form a desired arrangement.[16,17] Importantly, this approach is inherently non-equilibrium, and like biological signalling, it permits behavior that is unattainable at equilibrium. For example, one could create dynamically programmable colloidal assemblies that are self-repairing, have dynamically tunable properties, or that function as nanomachines that move and actuate.

Unfortunately, theories and simulations of non-equilibrium colloidal assembly are considerably less developed than equilibrium methods.[18] While much remains to be understood, a key mechanism for biomimetic non-equilibrium assembly involves directing particle chemotactic motion into desired states using chemical gradients that are generated by a distributed network of chemical reactions. This dynamic, many-body process is complex, and we postulate that the simpler problem of *chemical micro-actuation*—feedback-controlled steering of *individual* colloidal particles using *externally applied* chemical gradients—is an important step towards engineering nonequilibrium colloidal assembly via biomimetic chemical signalling.

In this paper, we develop a method for steering particles using chemical gradients that is based on a well-established approach to electrokinetic micro-actuation.[1,5–10] The specifics of steering colloidal particles with electrokinetic micro-actuation have been explained in detail by Chaudhary

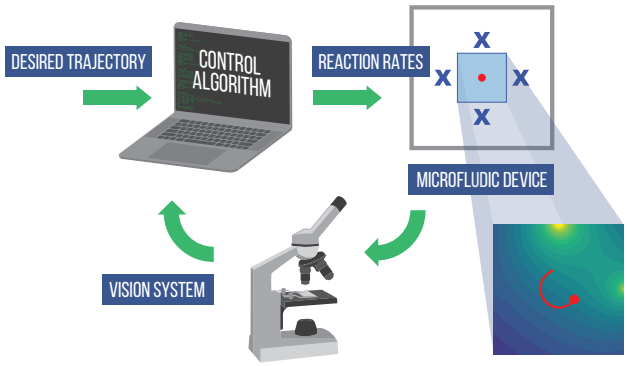

FIG. 1 An example microfluidic device for steering particles via diffusiophoresis. The trajectory (red line) of a colloidal particle (red circle) is controlled using feedback from a vision system that is fed to a computer algorithm that computes reaction rates for chemical actuators. The chemical gradient of the control area is shown by the contour plot in the lower right, where bright yellow indicates a high-concentration area and dark blue a low one.

and Shapiro[5]. The type of microfluidic device they describe contains three main components:

1. a vision system that measures the position of the particle in real time,

2. electrodes arranged around a control region to produce an electric field that exerts an electroosmotic force on the particle, and

3. a controller that uses a minimum norm least-squares algorithm to calculate the voltages to apply to the electrodes.

To steer colloids, position information from the vision system is fed to a controller that calculates the voltages that will move the particles on pre-specified trajectories. This method has been used to steer several particles at a time, both in simulations[5–7] and experimentally.[1,8–10]

In this paper we develop an analogous method for chemical micro-actuation and compare and contrast it with its electrokinetic counterpart. We make use of the phenomenon of diffusiophoresis, where a gradient in the concentration of a chemical solute causes a colloidal particle to move. Similar to the method of Chaudhary and Shapiro, we assume access to an experimental system with the following components:

1. real-time measurement of the position of the particle,

2. highly controllable solute concentration inputs (sources) and a method of removing the solute as it accumulates (sinks or flushing system), and

3. a controller that uses model predictive control to calculate the concentration inputs to apply.

Consequently, in our control method we assume that (1) the position of the particle is always known, (2) chemical species are created by point-sources that can be arbitrarily controlled, (3) species are infinitely soluble, and (4) species can be readily removed at the boundaries of the domain of interest.

In this paper, we focus on an example of such a system shown in Figure 1. In the example, a computer vision algorithm is used to measure the position of a particle using a camera and extract its pixel location. The position information is then fed to a control algorithm containing a simplified model of diffusion and particle motion. This algorithm calculates the required chemical source concentrations to propel the particle along a desired trajectory. We assume that these chemical sources (which we refer to as "probes" by analogy to the electrokinetic case) are localized, easily controlled, and far enough from particles to be considered point-sources. Consistent with our ideas of synthetic biochemical signalling discussed above, we further assume that these sources are chemical reactions whose rates are easily actuated (e.g., electrochemical,[19] photochemical,[20], or temperature-controlled[21] reactions), though our method is not limited by this latter assumption. Finally, to limit solute build-up, we assume that solute is removed at the edges of the microfluidic device via a thin semipermeable membrane with a pure solvent sweep on the exterior. This membrane is placed far from the controllable area, so it does not affect the gradient felt by the colloid.

The primary contribution of this paper is the development and simulation-based testing of control methods for steering a single colloidal particle in this example system. While it has been previously suggested that large concentration gradients can be used to direct the motion of particles through diffusiophoresis,[22–25] to the best of our knowledge, this is the first feedback-controlled method for dynamically steering individual particles via chemical gradients.

To steer a particle using chemical gradients, one requires a controller to calculate the necessary chemical gradients. Due to the inherently slow dynamics of chemical diffusion, a more sophisticated controller is required for chemical micro-actuation than for its electrokinetic analogue.[5] We use a controller that employs nonlinear model predictive control (NMPC), a method that optimally chooses control inputs based on predicted future states of the underlying physical process.[26] NMPC deals with nonlinear models, which is necessary for the non-equilibrium problem we are considering. NMPC has many of the advantages of linear model predictive control, including the ability to handle long delay times, multiple input and output variables, and constraints on the con-

trol inputs. NMPC has previously been used in a microfluidic device to position microchips using dielectrophoresis.[27]

We test our control method using two-dimensional (2D) Brownian dynamics (BD) simulations in which the motion of the colloid is coupled with a solute reaction-diffusion system that exerts a diffusiophoretic force on the colloid. To describe the diffusiophoretic motion, we assume that the velocity of a colloidal particle is directly proportional to the concentration gradient of a non-ionic solute,[28] and we explicitly calculate the concentration field of the solute using a finite difference method. Hydrodynamic effects other than diffusiophoresis and Brownian motion are neglected.

In addition to the above, we also provide an analysis of the viability of steering particles with different system parameters. We include the distance from chemical signals, particle size, diffusiophoretic mobility, and solute diffusivity in this analysis.

With these objectives in mind, the paper proceeds as follows: We first detail our methods for particle steering starting with a review of the methods for electrokinetic control, and then we detail our approach for diffusiophoretic control, including the setup of NMPC. We then demonstrate chemical gradient steering via NMPC using BD simulations, and we compare and contrast the method with particle steering using an electric field. We also provide an analysis of the relevant timescales involved in steering particles using a chemical gradient and produce design rules for choosing viable system parameters. Finally, we provide future direction for this method and offer some conclusions and perspective on future work.

II. METHODS

In this section, we describe our methods for steering particles using both electrokinetic and diffusiophoretic controllers. We begin by describing our BD simulations, and then we describe the control approaches for each method. Finally, we briefly discuss the physical parameters used in the simulations.

A. Brownian dynamics simulations

The simulations are designed assuming a single colloidal particle will be controlled within a 2D region, as shown in Figure 2. Bordering the region are either electrodes or reactive chemical "probes". For the electric field simulations, four electrodes are used, one on each of the North,

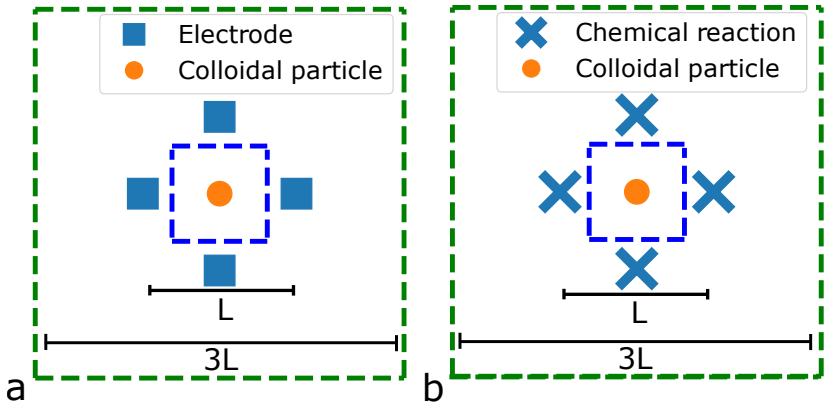

FIG. 2 Domain geometry for (a) electric field and (b) chemical gradient simulations. In (a), electrodes (blue squares) are controlled by changing their voltages with time to move a colloidal particle (orange dot) on a target path. In (b), the electrodes are replaced by chemical reactions (blue x's). In each, the dashed blue line is the area in which the particle is allowed to move (the control region) and the dashed green box is the boundary of the simulated area (the simulation domain). The electrodes or chemical probes are separated by a distance of L, and the simulation domain is a box of side length $3L$.

South, East, and West sides of the control region (shown in Figure 2a). Similarly, for the chemical gradient simulations, four chemical reaction "probes" are placed on each side of the control region (shown in Figure 2b). The electrodes/probes are separated by a distance L, and the control region (the region in which the particle are allowed to move) is the area between the electrodes/probes. The simulation domain extends beyond the control region to a length of $3L$. This makes the boundary conditions of the simulation distant enough that it does not disturb the motion of the particle.

We use Brownian dynamics (BD) to simulate the motion of a colloidal particle moving in an externally applied field. We simulate a single particle (implying there are no interparticle interactions) that feels either electrokinetic or diffusiophoretic forces, with other forces assumed to be zero. With the assumption that the motion of the particle is overdamped, the velocity of the particle is described by[29–31]

$$\dot{\boldsymbol{X}} = \frac{1}{\gamma}(\boldsymbol{F}_{\text{diff}} + \boldsymbol{F}_{\text{elec}}) + \sqrt{2D_c}\boldsymbol{\xi}, \tag{1}$$

where $\boldsymbol{X} = [x_{\text{particle}}, y_{\text{particle}}]^T$ is the 2D position of the colloidal particle, $\boldsymbol{F}_{\text{diff}}$ and $\boldsymbol{F}_{\text{elec}}$ are the diffusiophoretic and electrokinetic forces respectively, and $\boldsymbol{\xi}$ is a Gaussian white noise term defined

as

$$\xi \equiv \lim_{dt \to 0} N(0, dt^{-1}) \tag{2}$$

where $N(0, dt^{-1})$ is a normal distribution with a mean of 0 and a variance of dt^{-1} and dt is the time differential. Additionally, $\gamma = 6\pi\eta R_c$ is the Stokes friction coefficient, and $D_c = k_b T/\gamma$ is the diffusion coefficient of the colloidal particle, which introduces Boltzmann's constant k_b, the system temperature T, the viscosity of the surrounding fluid η, and the radius of the colloidal particle R_c.

The electrokinetic force $\boldsymbol{F}_{\text{elec}}$ is a combination of the electrophoretic force $\boldsymbol{F}_{\text{ep}}$ on the particle and the force imparted by electroosmotic flows $\boldsymbol{F}_{\text{eo}}$. In general, a particle can be electrokinetically steered using electroosmotic flows if it is neutral or a combination of electrophoresis and electroosmosis if it is charged.[8] However, electroosmosis complicates the comparison between electrokinetic steering and diffusiophoretic steering. Accordingly, since our primary purpose is to examine particle steering via diffusiophoresis, we will limit our analysis to the case where electroosmotic flows are suppressed ($\boldsymbol{F}_{\text{eo}} = 0$), and a charged particle is steered solely using electrophoresis ($\boldsymbol{F}_{\text{elec}} = \boldsymbol{F}_{\text{ep}}$).

The electrophoretic force is given by[32]

$$\boldsymbol{F}_{\text{elec}}(\boldsymbol{X}, t) = \gamma \mu_e \boldsymbol{E}(\boldsymbol{X}, t) = -\gamma \mu_e \nabla \Phi(\boldsymbol{X}, t), \tag{3}$$

where μ_e is the electrophoretic mobility, $\boldsymbol{E}$ is the electric field, and Φ is the electric potential. As emphasized in Eq. 3, the electric potential is a function of position and time, and is determined by the voltages $\phi_i(t)$ of the i electrodes. We calculate the electric potential by solving Laplace's equation

$$\nabla^2 \Phi(x, y, t) = 0 \tag{4}$$

with boundary conditions $\Phi(\boldsymbol{X}_{\text{probe},i}, t) = \phi_i(t)$ at the electrode locations $\boldsymbol{X}_{\text{probe},i}$ and Φ = 0 at the edge of the computational domain as indicated in Figure 2. We discretize the domain using a 2D finite difference methood, and solve Laplace's equation using numerical relaxation via the Gauss-Seidel algorithm.

The diffusiophoretic force for a dilute, non-ionic solute is given by[28]

$$\boldsymbol{F}_{\text{diff}}(\boldsymbol{X}, t) = \gamma \mu_d \nabla C(\boldsymbol{X}, t). \tag{5}$$

where C is the concentration of the solute and μ_d is the diffusiophoretic mobility. The sign and the magnitude of the mobility depend on interactions between the particle, the solute, and the solvent,[33–35] and the units for the mobility are $m^2/(M\ s)$, where M means moles per liter.

Analogous to the need to know the electric potential gradient in the case of electrophoresis, one must determine the concentration gradient near the colloid to compute the diffusiophoretic force. Assuming the diffusion coefficient of the solute is constant (which is appropriate for a dilute solution) and that there is no coupling between hydrodynamics and diffusion (i.e. that there are no Marangoni effects or density-induced flows and that the colloid velocity is slow relative to diffusion), we determine the concentration field using a reaction-diffusion equation

$$\frac{\partial C(x,y,t)}{\partial t} = D_s \nabla^2 C(x,y,t) + G(x,y,t), \tag{6}$$

where D_s is the diffusion coefficient of the solute and $G(x,y,t)$ is the reaction rate of a local time-varying chemical reaction caused by reactive "probes" located at various points in the simulation domain as shown in Figure 2. We used Dirichlet conditions $C = 0$ at the boundary of the computational domain. As previously discussed, the zero concentration boundary acts as a concentration sink that keeps the average concentration from monotonically increasing with time. We calculate the concentration field using a finite difference method by discretizing Equation (6) in two dimensions using a Forward Time Centered Space (FTCS) scheme on a regular grid. We chose a time step of $\Delta t_{\text{sim}} \leq 0.5(\Delta x)^2/4D_s$ to satisfy the Neumann stability criterion for the FTCS method,[36] where Δx is the distance between grid nodes.

B. Electrokinetic micro-actuation control algorithm

Our electric field micro-actuation control loop includes the following four steps: (1) A vision system measures the position of a particle relative to a set of electrodes arranged regularly around the edge of the device. (2) Using the position information, a dynamic model is used to relate the motion of the particle to the voltages of the electrodes. (3) A user-defined reference velocity is chosen and then modified with a feedback term to correct for Brownian motion and other inaccuracies in sensing, actuation, and modelling. (4) Electrode voltages are chosen by minimizing the difference between the dynamic model velocity and the reference velocity. These steps are explained in detail in the following paragraphs.

In step (1), we assume that a vision system is able to measure the position of a colloidal particle in real time. The proposed vision system would include a camera attached to a microscope, along with any image processing algorithms necessary to extract the x and y position of the particle relative to the electrodes.

In step (2), we model the dynamics of a particle that moves with the drift velocity due to the electrophoretic force only, neglecting any other forces or Brownian motion. Accordingly, the velocity of a particle experiencing electrophoresis becomes

$$\hat{\boldsymbol{v}}_{\text{elec}} = \frac{\boldsymbol{F}_{\text{elec}}}{\gamma} = -\mu_e \nabla \Phi(\boldsymbol{X}, t) \tag{7}$$

where we have substituted Equation 3 for $\boldsymbol{F}_{\text{elec}}$. What remains is to relate the electric potential Φ to the voltages of the electrodes, with the voltage of electrode i given by ϕ_i. As mentioned in previously, this can be done using a numerical solution to the Laplace equation. However, we found it more convenient to instead use the Charge Simulation Method[37–39] to relate the electrode voltage ϕ_i to an effective charge $q_{\text{eff},i}$ at the electrode location. There are two primary advantages to this approach. (1) We can make use of an analytical relation between the effective charge and the potential that simplifies the calculation of the electric field. (2) There is a useful analogy that we wish to highlight between the effective charge $q_{\text{eff},i}$ of an electrode in electrokinetic micro-actuation and the reaction rate at a point in space in diffusiophoretic micro-actuation.

We now derive the dynamic model used in the control algorithm by using the Charge Simulation Method to relate the velocity of the particle to the effective charge of each electrode. To represent the electric potential in a 2D electric field, line charges perpendicular to the 2D plane may be used.[39] For an effective line charge located at $\boldsymbol{X}_{\text{probe},i}$, we define the vector from the charge to the particle as $\boldsymbol{r}_i = \boldsymbol{X} - \boldsymbol{X}_{\text{probe},i}$ with magnitude r_i. The electric potential for such a line charge as a function of distance is given by[37,38]

$$\Phi_i = \frac{q_{\text{eff},i}}{2\pi\varepsilon} \ln\left(\frac{R_{z,i}}{r_i}\right), \tag{8}$$

where $q_{\text{eff},i}$ is the effective charge per unit length, ε is the electric permittivity of the medium, and $R_{z,i}$ is the distance between $\boldsymbol{X}_{\text{probe},i}$ and the nearest location of zero reference potential, which occurs at the boundary. The voltage ϕ_i of an electrode is the electric potential a particle would feel at a distance of one electrode radius $R_{\text{electrode}}$ from the effective charge. Thus it is possible to

convert between $q_{\text{eff},i}$ and ϕ_i using the relationship

$$\phi_i = \frac{q_{\text{eff},i}}{2\pi\varepsilon} \ln\left(\frac{R_{z,i}}{R_{\text{electrode}}}\right). \tag{9}$$

The motion of the particle depends on the gradient of the electric potential. Taking the gradient of Equation 8 with respect to $\boldsymbol{X}$ gives

$$\nabla\Phi_i = -\frac{q_{\text{eff},i}}{2\pi\varepsilon r_i^2}\boldsymbol{r}_i. \tag{10}$$

The total field is a linear superposition of the fields produced by the i individual charges. After summing up the contributions from Equation (10) of each of N_{probe} electrodes and substituting the result into Equation (7), the model velocity of the colloidal particle in an electric field is then

$$\hat{\boldsymbol{v}}_{\text{elec}}(t) = \frac{-\mu_e}{2\pi\varepsilon}\sum_{i=0}^{N_{\text{probe}}}\frac{q_{\text{eff},i}(t)}{r_i^2}\boldsymbol{r}_i. \tag{11}$$

This is the dynamic model used for our control algorithm. Note that we have neglected to include image charges to meet the boundary conditions that were present in the simulation. This was done to reduce the computational complexity of the dynamic model. Neglecting to include image charges introduces an error to the dynamic model due to boundary effects, but we found the error to be small.

Now that we have developed a dynamic model, we can continue to step (3). In step (3), we define a reference trajectory and modify it using a feedback term to correct for thermal noise and sensing and modelling inaccuracies. First, we pre-define a reference trajectory $\boldsymbol{x}_{\text{ref}}(t) = [x_{\text{ref}}(t), y_{\text{ref}}(t)]^T$ for the colloidal particle to follow. We also take the derivative of the pre-defined trajectory, $\dot{\boldsymbol{x}}_{\text{ref}}(t)$, which is the open-loop velocity that our controller must produce. However, to track the trajectory in the presence of noise, measurement errors, and process errors, we add a feedback term $k_{\text{elec}}(\boldsymbol{X} - \boldsymbol{x}_{\text{ref}})$, or the difference between the measured position and the reference position, to ensure the particle actually follows the desired trajectory. Thus, we define

$$\boldsymbol{v}_{\text{des}} \equiv \dot{\boldsymbol{x}}_{\text{ref}} - k_{\text{elec}}(\boldsymbol{X} - \boldsymbol{x}_{\text{ref}}) \tag{12}$$

as the desired velocity to produce using the optimization algorithm.

In step (4), we minimize the difference between the predicted velocity $\hat{\boldsymbol{v}}_{\text{elec}}$ from Equation (11) and the desired velocity $\boldsymbol{v}_{\text{des}}$ from Equation (12). To solve this optimization problem, we used a nonlinear optimization algorithm to calculate the electrode voltages by minimizing the cost function

$$J_{\text{elec}}(\boldsymbol{u}_{\text{elec}}) = w_{\text{elec},1}\frac{||\hat{\boldsymbol{v}}_{\text{elec}} - \boldsymbol{v}_{\text{des}}||^2}{v_0^2} + w_{\text{elec},2}\frac{||\boldsymbol{u}_{\text{elec}}||^2}{q_{\text{scale}}^2}, \tag{13}$$

where v_0 is a velocity scale that will be presented in the Results section, q_{scale} is a scale for the electric charge derived in the Appendix, and $\boldsymbol{u}_{\text{elec}}$ is the vector of control inputs defined as

$$\boldsymbol{u}_{\text{elec}} = \left[q_{\text{eff},1}, q_{\text{eff},2}, \ldots, q_{\text{eff},N_{probe}}\right]^T. \tag{14}$$

The weights $w_{\text{elec},1}$ and $w_{\text{elec},2}$ are tuning parameters of the method. The optimization was implemented using in Python using the `least_squares` function from the `scipy.optimize`[40] library, which uses the "Trust Region Reflective" algorithm. The voltages were constrained to be positive to make a better comparison with our chemical controller results. The simulations were performed with a variety of initial positions, domain sizes, and paths. A subset of these simulations will be presented in the Results section.

There are two major differences between our methods and the methods for electric field steering used previously by Chaudhary and Shapiro.[5] First, in step (2), they used a finite element solution to the Laplace equation to create a dynamic model instead of using the Charge Simulation Method. Our choice to use the Charge Simulation Method added a small amount of modeling error, but it also made the dynamic model less expensive to compute and will provide a clearer analogy with our chemical gradient controller. Second, in step (4), instead of using nonlinear optimization, they used a minimum norm least-squares algorithm to calculate the voltages of the electrodes in a linearized system of equations, and they dealt with ill-conditioning using a singular value decomposition. While their minimum norm solution is simpler and faster to compute, we used nonlinear optimization to be consistent with our chemical controller, which cannot be computed using a minimum norm solution.

C. Diffusiophoretic micro-actuation control algorithm

Similar to the case of electrokinetic micro-actuation, we use optimal control with diffusiophoretic micro-actuation to minimize the difference between the diffusiophoretic velocity $\hat{\boldsymbol{v}}_{\text{diff}}$

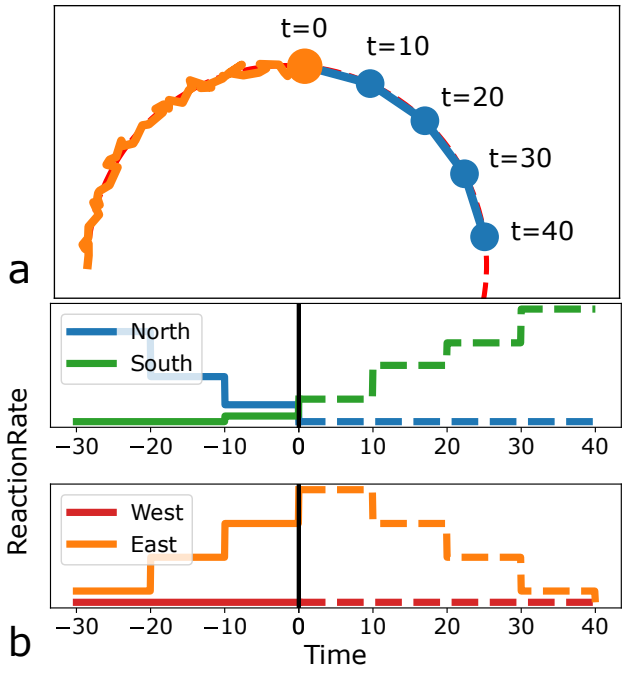

FIG. 3 An illustration of how model predictive control steers a colloid on a circular trajectory. (a) The colloid (orange dot) is steered on a reference trajectory (red dashed line) using predictions of its future position (blue dots) with N_{pred}=4. In this example, the particle has already been moving for $t = 50$ (arbitrary units), and the four future points are predicted at a fixed $\Delta t_{\text{control}} = 10$. (b) Past (solid) and predicted (dashed) optimal inputs (i.e., reaction rates) for the four probes located North, South, East and West of the particle.

from a dynamic model developed in the following paragraphs and the desired velocity $\boldsymbol{v}_{\text{des}}$ from Equation (12). However, unlike the near-instantaneous changes in electric potentials, changes in concentration are due to diffusive transport processes, which take time to propagate through the solution. This leads to a significant delay between the chemical signal from the source and the motion of the particle, necessitating a more sophisticated control strategy.

To deal with this delay, we use nonlinear model predictive control (NMPC). In NMPC one specifies a cost function J that includes information about the current state and model predictions of the future states of the particle.[26,41,42] One then uses a nonlinear optimization algorithm to minimize this cost function and derive control commands for the concentration sources. The model prediction is necessarily limited to a finite future time, called the event or prediction horizon, resulting in an optimization over a set number of future steps N_{pred}. The choice of the size of the prediction horizon results in a tradeoff between speed and accuracy: a larger prediction horizon may give better performance, but it also increases the computational cost of the optimization problem.[41]

Figure 3 shows an example of how NMPC can be used to steer a colloid. Panel (a) shows the trajectory $\boldsymbol{X}(t)$ and dynamic model prediction $\hat{\boldsymbol{x}}_{\text{diff}}(t)$ of a colloid as it is steered along a circular reference trajectory $\boldsymbol{x}_{\text{ref}}(t)$. Panel (b) shows the optimal inputs (i.e., reaction rates) for four probes that surround the particle, obtained by using an optimization algorithm to minimize the cost function J. The key aspect of NMPC demonstrated here is that the input reaction rates in part (b) are determined such that the dynamic model prediction in part (a) lies along the target trajectory.

Given this overview, we now describe the specifics of this method, starting with the cost function. We would like a cost function that, when minimized, drives the particle to the reference trajectory while simultaneously minimizing the amount of source concentration needed to actuate motion. A cost function based on the commonly used concept of a squared error[26] that accomplishes this goal is given by

$$J_{\text{diff}}\left(\{\boldsymbol{u}_k\}\right) = \sum_{k=1}^{N_{\text{pred}}+1} w_{\text{diff},1} \frac{||\hat{\boldsymbol{v}}_{\text{diff},k} - \boldsymbol{v}_{\text{des},k}||^2}{v_0^2} + \sum_{k=1}^{N_{\text{pred}}+1} w_{\text{diff},2} \frac{||\boldsymbol{u}_k||^2}{g_{\text{scale}}^2}, \tag{15}$$

where $\hat{\boldsymbol{v}}_{\text{diff},k}$ is the velocity predicted by the dynamic model for k time steps into the future, $\boldsymbol{v}_{\text{des},k}$ is the desired velocity for time step k, given in Equation (16), v_0 is a velocity scale that will be presented in the Results section, g_{scale} is a reaction rate scale derived in the Appendix, and $\boldsymbol{u}_k = \left[u_{1k}, u_{2k}, \ldots, u_{N_{probe}k}\right]^T$ is a vector of N_{probe} reaction rates at time step k. The weights $w_{\text{diff},1}$ and $w_{\text{diff},2}$ are tuning parameters of the method.[26] Generating these tuning parameters is a key step in developing optimal control of the system and will be discussed below. As desired, the cost function has two terms: the first drives the dynamic model to the desired trajectory and the second minimizes the concentration needed to actuate the system.

For the desired velocity $\boldsymbol{v}_{\text{des},k}$, we modify Equation (12) to be a function of the time step k, giving

$$\boldsymbol{v}_{\text{des},k} = \dot{\boldsymbol{x}}_{\text{ref}}(t_k) - k_{\text{diff}}\left(\hat{\boldsymbol{x}}_{\text{diff},k} - \boldsymbol{x}_{\text{ref}}(t_k)\right). \tag{16}$$

The main modification we have made is that we now use $\hat{\boldsymbol{x}}_{\text{diff},k}$, the position predicted from the dynamic model for time step k, instead of the measured position $\boldsymbol{X}$. Since the initial prediction is set equal to the measured position, or $\hat{\boldsymbol{x}}_{\text{diff},1} = \boldsymbol{X}$, this equation reduces to Equation (12) when $k = 1$.

In addition to specifying a cost function, our NMPC method requires a dynamic model for

the motion of the particle. Rather than use a computationally expensive numerical solution (e.g., with finite differences), we develop here a method based on the Green's function for the diffusion equation that is analogous to the Charge Simulation Method for the Laplace equation described previously. As we will see, a primary advantage of this method is that the concentration gradient ∇C only needs to be evaluated at the colloid location, rather than finding the solution to the unsteady diffusion problem in all space. However, the efficiency of this approach is attenuated by the history dependence of the Green's function, which causes the method to become increasingly costly with time.

For the dynamic model, we assume that the only forces acting on the particle are diffusiophoretic forces

$$\hat{\boldsymbol{v}}_{\text{diff}} = \frac{\boldsymbol{F}_{\text{diff}}}{\gamma} = \mu_d \nabla C(\boldsymbol{X},t), \tag{17}$$

neglecting thermal noise. The concentration $C(\boldsymbol{X},t)$ is again given by the reaction-diffusion equation provided in Equation (6), where the reaction rate term $G(x,y,t)$ is a function of both time and space.

In an infinite domain, Equation (6) has an analytical Green's function solution,[43]

$$C(\boldsymbol{X},t) = \int_0^t \int \frac{1}{[4\pi D_s(t-\tau)]^{n/2}} e^{\frac{-||\boldsymbol{X}-\boldsymbol{\sigma}||^2}{4D_s(t-\tau)}} G(\boldsymbol{\sigma},\tau) d^n\boldsymbol{\sigma} d\tau + \int \frac{1}{(4\pi D_s t)^{n/2}} e^{\frac{-||\boldsymbol{X}-\boldsymbol{\sigma}||^2}{4D_s t}} C(\boldsymbol{X},0) d^n\boldsymbol{\sigma}, \tag{18}$$

where n is the number of spatial dimensions, and $\boldsymbol{\sigma}$ and τ are dummy integration variables for space and time respectively. As previously mentioned, boundary correction terms can be added using the method of images, but we chose to neglect these corrections in favor of a simpler and faster dynamic model and anticipate that feedback control will correct for this model simplification.

We are interested in the case of a 2D domain, which allows us to specify that $n = 2$. We also assume a solution initially devoid of solute, i.e., $C(\boldsymbol{X},t) = 0$, meaning the second integral in Equation (18) disappears. We further assume that there are N_{probe} solute sources (chemical control probes) at discrete points in space $\boldsymbol{X}_{\text{probe},i}$, so that

$$G(\boldsymbol{\sigma},\tau) = \sum_{i=1}^{N_{\text{probe}}} g_i(\tau)\delta(\boldsymbol{\sigma} - \boldsymbol{X}_{\text{probe},i}), \tag{19}$$

where $g_i(\tau)$ is the reaction rate of source i in units of moles per time per unit length and δ is the Dirac delta function. Defining $\boldsymbol{r_i} \equiv \boldsymbol{X} - \boldsymbol{X}_{\text{probe},i}$ to be the vector between the particle and probe i with magnitude r_i, simplifies Equation (18) to

$$C(\boldsymbol{X},t) = \sum_{i=1}^{N_{\text{probe}}} \int_0^t \frac{1}{4\pi D_s(t-\tau)} e^{\frac{-r_i{}^2}{4D_s(t-\tau)}} g_i(\tau) d\tau. \tag{20}$$

Finally, taking the gradient with respect to $\boldsymbol{X}$ and multiplying by μ_d, gives the diffusiophoretic force needed for the dynamic model in Equation (17), yielding

$$\hat{\boldsymbol{v}}_{\text{diff}} = \mu_d \sum_{i=1}^{N_{\text{probe}}} \int_0^t \frac{-\boldsymbol{r}_i}{8\pi D_s^2(t-\tau)^2} e^{\frac{-r_i{}^2}{4D_s(t-\tau)}} g_i(\tau) d\tau. \tag{21}$$

Equation 21 is a valid model for NMPC, but the presence of the integral makes it slow for use in real-time control settings. Accordingly, we make two additional simplifications that permit us to evaluate the integral. First, we assume that the distance from the particle to the probe does not change significantly within a single time step. That is, we replace the vector $\boldsymbol{r}_i$ in Equation (21) with the most recently measured probe-particle vector $\boldsymbol{r}_i^*$. Since $\boldsymbol{r}_i^*$ is not a function of time, it can be pulled out of the integral, giving

$$\hat{\boldsymbol{v}}_{\text{diff}} = \sum_{i=1}^{N_{\text{probe}}} \frac{-\mu_d \boldsymbol{r}_i^*}{8\pi D_s^2} \int_0^t \frac{1}{(t-\tau)^2} e^{\frac{-r_i{}^{*2}}{4D_s(t-\tau)}} g_i(\tau) d\tau, \tag{22}$$

where r_i^* is the magnitude of $\boldsymbol{r}_i^*$.

Second, we assume a form for $g(t)$ that allows us to evaluate the integral analytically. The functions $g_i(t)$ correspond to the control inputs, and we now assume they are composed of discrete steps

$$g_i(t) = \sum_{j=1}^{N_t+1} u_{ij} \left[H(t-t_j) - H\left(t-t_{j+1}\right) \right], \tag{23}$$

where u_{ij} is a constant control input (controlled reaction rate at the probe location) for probe i applied at time step j, with units of moles per time per unit length, and N_t is the number of time steps elapsed since the beginning of the simulation. Additionally, Equation 23 includes the Heaviside function

$$H(t') = \begin{cases} 1 & t' > 0 \\ 0 & t' \leq 0 \end{cases}, \tag{24}$$

which depends on t_j and $t_{j+1} = t_j + \Delta t_{\text{control}}$ so that the control impulse lasts for the length of a single time step between control loops $\Delta t_{\text{control}}$. Using Equation 23, we can evaluate the integral in Equation (22) using Laplace transforms and the convolution theorem. This procedure, which is detailed in the Supporting Information, yields,

$$\hat{\boldsymbol{v}}_{\text{diff}} = \frac{-\mu_d}{2\pi D_s} \sum_{i=1}^{N_{\text{probe}}} \left[\frac{\boldsymbol{r}_i^*}{r_i^{*2}} \sum_{j=1}^{N_t+1} u_{ij} \Delta h_j(t) \right] \tag{25}$$

with

$$\Delta h_j(t) \equiv h(t, t_j) - h(t, t_{j+1}) \tag{26}$$

and

$$h(t,\tau) = e^{\frac{-r_i^{*2}}{4D_s(t-\tau)}} H(t-\tau). \tag{27}$$

Equation (25) is a nonlinear time-series model that gives the predicted velocity of a particle as a function of its position, time, and the history of control inputs u_{ij}. Each input u_{ij} in Equation (25) is multiplied by the nonlinear coefficient function $\Delta h_j(t)$. Since this function plays a key role in the control algorithm, it is useful to examine it more closely. The variable $\Delta h_j(t)$ is the time-dependent dimensionless response of the particle velocity from a single impulse lasting for time $\Delta t_{\text{control}}$ from a single concentration probe. Thus, a reaction rate impulse applied to a probe at time $t = 0$ and lasting for $\Delta t_{\text{control}}$ seconds will have an impact on the particle with a magnitude that is spread over time according to this coefficient function.

This coefficient function is shown as a function of time in Figure 4 for two different values of the probe distance r_i^* and selected values of other parameters, for the purpose of illustrating two important points. First, the relationship between an input and the resulting velocity is affected by the distance between the probe and the particle. Consider a single input that lasts for $\Delta t_{\text{control}}$ seconds and then turns off. If the particle and the probe are close together (blue line in Figure 4), the particle will accelerate quickly, and then rapidly decelerate once the input is turned off. If the particle is farther away from the probe (green dotted line in Figure 4), the velocity will increase more slowly, reach a lower maximum speed, and decay less quickly to zero than the previous case. The slower and weaker response of the second case is a consequence of both the additional time it takes for the solute to diffuse to the colloid and the lower concentration of the solute due to additional dispersion over space.

The second lesson of Figure 4 is that even though a control input affects the velocity of the par-

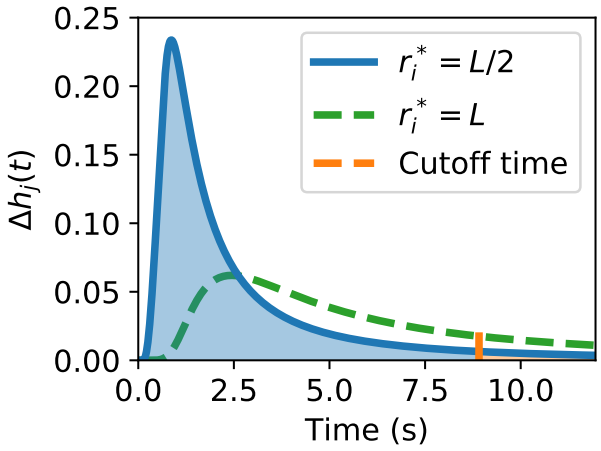

FIG. 4 The nonlinear coefficient function $\Delta h_j(t)$ as a function of time with parameters $t_j = 0$, $\Delta t_{\text{control}} = 0.5$ s, and D_s and L as given in Table I. Coefficient function $\Delta h_j(t)$ is given for both an average particle-probe distance $r_i^* = L/2$ (blue line) and for a large particle-probe distance $r_i^* = L$ (green dotted line). The vertical orange dashed line shows the impact of implementing a cutoff by setting the coefficient function to zero after n_{cutoff} time steps have passed. The portion of the coefficient function captured by the dynamic model when the particle is at $L/2$ is shaded in blue, and the portion neglected by the cutoff is shaded in orange.

ticle long after it has turned off, the effects decay to zero as t gets large. Consequently, introducing a cutoff to $\Delta h_j(t)$ could result in a significant savings of computational time while introducing only small errors to the model. Indeed, the reduction in computational time can be quite large because: (1) the sum in Equation (25) contains an increasing (and unbounded) number of terms as t increases, and (2) $\Delta h_j(t)$ is a nonlinear function of both space and time that must be re-evaluated at each time step. As such, without a cutoff, the time it takes to compute the sum in Equation (25) diverges as the simulation proceeds. We avoid this problem by evaluating a fixed number n_{cutoff} of terms in the relevant sum, giving

$$\hat{\boldsymbol{v}}_{\text{diff}} = \frac{-\mu_d}{2\pi D_s} \sum_{i=1}^{N_{\text{probe}}} \left[\frac{\boldsymbol{r}_i^*}{r_i^{*2}} \sum_{j=N_t-n_{\text{cutoff}}}^{N_t+1} u_{ij} \Delta h_j(t) \right] \tag{28}$$

when n_{cutoff} timesteps have passed. As expected, implementing this cutoff results in an optimization problem that is solved in a consistent and much smaller amount of time. Further discussion of the choice of n_{cutoff} and its impact on particle control will be discussed below.

With Equation (28) as our dynamic model, we implemented NMPC using the nonlinear optimization algorithm `least_squares`, from the scipy.optimize[40] library, to minimize the cost func-

tion in Equation (15). This process minimizes the difference between the velocity predicted by the dynamic model and the reference velocity from Equation (16). For this work, we use evenly spaced prediction steps with a spacing equal to the time between control loops, $\Delta t_{\text{control}}$. Once the optimization algorithm has calculated the optimal inputs over the prediction horizon, the controller executes the first of the calculated inputs only. At the next time step, the position of the particle is measured again and the optimization is repeated. This iterative process allows the controller to benefit from more frequent feedback, which can reduce the effects of noise and errors in the dynamic model.

Since chemical reactions generally cannot be negative, we constrained the solver to only allow non-negative inputs. Consequently, the x-direction can be fully controlled using the East and West probes, even with the lower bound, because the probes act from different directions. The same is true for the y-direction and the North and South probes. In addition to the lower bound, an upper bound was set to avoid unrealistically high input values and to help the optimization algorithm converge. Details relating to the performance of the control algorithm will be discussed in the Results section.

D. Simulation parameters

Table I provides the parameters used for both simulations with a time-varying electric potential and those with a time-varying chemical gradient. In both simulations, we assume a room-temperature water-like solvent which sets the values of the viscosity and relative permittivity, and we assume the solute has a diffusion coefficient equivalent to the self-diffusion coefficient of water. We consider a particle with a radius of 1 μm, which is on the smaller end of what can be measured by optical microscopy. We use a value of the electrophoretic mobility that is typical of a polystyrene bead,[8,44] and set the diffusiophoretic mobility to a value predicted by Anderson[28] for a typical non-ionic solute. Simulation specific parameters include: the spacing of the finite difference grid with 150×150 nodes $\Delta x = 3L/150$, and the radius of the electrode which was set equal to Δx.

We programmed both the simulation and control algorithm using Python. Not having attempted extensive optimizations, simulations of 600 seconds (with $\Delta t_{\text{sim}} = 0.0008$ seconds chosen for numerical stability) take approximately 25 minutes to complete on a standard desktop computer.

TABLE I Simulation Parameters.

Symbol	Explanation	Value
L	Length of control region	200 μm
R_c	Particle size	1 μm
T	Temperature	298 K
η	Fluid viscosity	0.89 mPa s
ε	Electric permittivity	6.94×10^{-10}
μ_e	Electrophoretic mobility	$2.0\times10^{-8}\mathrm{m}^2/(\mathrm{V\,s})$
D_s	Solute diffusion coefficient	2.3×10^{-9} m^2/s
μ_d	Diffusiophoretic mobility	2.0×10^{-10} $\mathrm{m}^2/(\mathrm{M\,s})$

TABLE II Electric Controller Tuning Parameters.

Symbol	Explanation	Value/expression
k_{elec}	Feedback constant	0.1 s^{-1}
$w_{\mathrm{elec},1}$	Velocity error cost function weight	400
$w_{\mathrm{elec},2}$	Electrode cost function weight	1
ub_{elec}	Upper bound on optimization	$1.3q_{\mathrm{scale}}$

E. Controller tuning

The tuning parameters for the electric field controller and the chemical gradient controller are presented here. The tuning parameters for the electric field controller are the feedback constant k_{elec}, the cost function weights $w_{\mathrm{elec},1}$ and $w_{\mathrm{elec},2}$, and the upper bound on the optimization algorithm ub_{elec}. The values of these parameters are shown in Table II. The tuning parameters for the chemical controller are the feedback constant k_{diff}, the cost function weights $w_{\mathrm{diff},1}$ and $w_{\mathrm{diff},2}$, the upper bound on the optimization algorithm ub_{diff}, the number of terms n_{cutoff} evaluated in the time series model, and the prediction horizon N_{pred}. The values, or expression for choosing the values of these parameters, are shown in Table III. These parameters will be discussed in further detail in the following paragraphs.

The feedback constants k_{elec} and k_{diff} are chosen to avoid excessive control effort. When the feedback constant is large, the controller uses more effort to correct errors in the position of the particle. This can make the inputs hug the upper bounds ub_{diff} and ub_{elec} respectively, which is a sign of excessive control effort. We chose a value of $k_{\mathrm{elec}} = k_{\mathrm{diff}} = 0.1$ as a balance between correcting position errors and minimizing control effort.

Cost function weights $w_{\mathrm{elec},1}$, $w_{\mathrm{elec},2}$, $w_{\mathrm{diff},1}$ and $w_{\mathrm{diff},2}$ determine the relative cost of deviating

TABLE III Chemical Controller Tuning Parameters.

Symbol	Explanation	Value/expression
k_{diff}	Feedback constant	0.1 s^{-1}
$w_{\text{diff},1}$	Velocity error cost function weight	400
$w_{\text{diff},2}$	Reaction rate cost function weight	1
ub_{diff}	Upper bound on optimization	$1.3g_{\text{scale}}$
n_{cutoff}	Number of terms in model	30% max decay time
N_{pred}	Prediction horizon	Between 1 and 5

from the reference velocity compared to increasing the magnitude of the reaction rates that drive control. For the following simulations, we use $w_{\text{elec},1} = w_{\text{diff},1} = 400$ and $w_{\text{elec},2} = w_{\text{diff},2} = 1$, giving a relative cost of $400:1$, meaning the controller greatly prioritizes creating the correct velocity over minimizing the control effort.

The upper bounds ub_{elec} and ub_{diff} of the optimization algorithm provide a maximum limit on the electrode voltage and reaction rate the probes can use to drive control. We will show in the Results section that a charge of q_{scale} or a reaction rate of g_{scale} gives a particle velocity on the order of v_0, where q_{scale} and g_{scale} are defined in the Appendix. Accordingly, we set the upper bound of the controller to be slightly larger than q_{scale} and g_{scale} to allow it to correct for Brownian motion. In the simulations shown below, we set $ub_{\text{elec}} = 1.3q_{\text{scale}}$ and $ub_{\text{diff}} = 1.3g_{\text{scale}}$.

As discussed previously, the performance of the chemical control algorithm is strongly dependent on n_{cutoff}, the number of terms retained in the time series model, Equation (28). Recall from Figure 4 that the coefficient function $\Delta h_j(t)$ depends on the probe distance r_i^* and decays at long times. The longest decay time occurs when the particle is farthest from the probe at a distance of $r_i^* = L$. Accordingly, we choose a cutoff time, t_{cutoff}, when the coefficient function has decayed to 30% of its peak value by solving the following implicit equation

$$\Delta h_0(t_{\text{cutoff}}) = 0.3 \max\left[\Delta h_0(t_{\text{cutoff}})\right] \tag{29}$$

where $t_0 = 0$, $t_1 = \Delta t_{\text{control}}$, and h is evaluated at $r_i^* = L$. This maximum decay time is a function of constant parameters, so it only needs to be calculated once at the beginning of a simulation. The parameter n_{cutoff} is then readily determined via the time step size using

$$n_{\text{cutoff}} = \text{ceil}(t_{\text{cutoff}}/\Delta t_{\text{control}}) \tag{30}$$

The choice of 30% was made based on trial and error as a balance between accuracy and speed. Note that, while the cutoff time is determined as 30% of the longest time to decay to zero, on average the error will be much smaller than 30% because the particle is on average much closer to the probe.

Finally, we choose a prediction horizon $N_{\text{pred}} \in [1,5]$, which is large enough to ensure the dynamic model can capture solute diffusion but small enough to avoid excessive computational cost. Note that the choice of N_{pred} is sensitive to model parameters, and criteria influencing this decision are discussed in greater detail in the Results section.

III. RESULTS

In this section, we present simulation results showing our ability to steer colloidal particles. These simulations demonstrate the use of concentration gradients to control colloidal particles as described in the Methods section. They also highlight that a chemical gradient controller can achieve a level of performance similar to that of an electric field controller.

We begin by showing an example of successfully steering a colloid on a circular trajectory, first using electric fields and then using chemical gradients, and analyze the similarities and differences between these two methods. Then we demonstrate the capability of the chemical gradient controller to create trajectories other than circles. Finally, we use dimensional analysis of the relevant time scales to provide heuristics of when steering a colloid using chemical gradients is effective.

A. Electric field and chemical gradient comparison

This subsection shows our controllers steering colloids to a circular trajectory and compares the behavior of the electric and chemical controllers. We use the parameters from Table I, a time step of $\Delta t_{\text{control}} = 1$ second, and an end time of 600 seconds.

We use a circular reference trajectory in our analysis as a base case because a circular trajectory is a simple path that demonstrates the need for continuous control. For this trajectory, we chose a circular path of radius $r_d = 30\mu$m and centered at (x_t, y_t), with the particle speed a constant

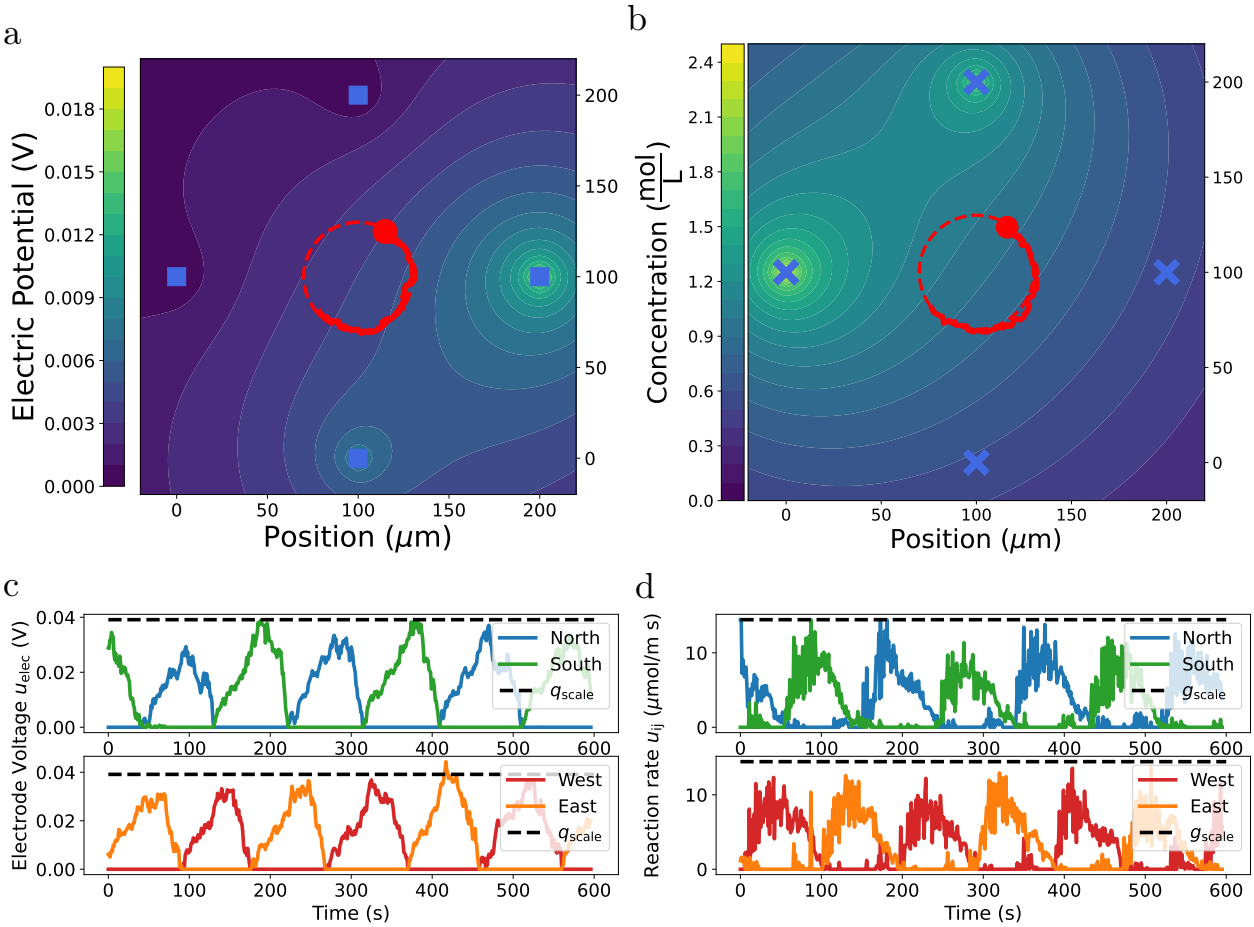

FIG. 5 The trajectory and control inputs for both the electric field controller and the chemical gradient controller. (a) An electric field simulation, in which the particle (red dot) is steered to follow a circular trajectory (dashed red line), where the actual path of the particle is shown as a solid red line and the locations of the four electrodes are represented by blue x's. (b) The voltages of each of the four probes are shown as a function of time. (c) A chemical simulation, where the four chemical probes are shown as blue x's. (d) The chemical reaction rates are shown as a function of time.

$v_0 = 1\mu$m/s. The equation for the circular reference trajectory is then

$$x_{\text{ref}}(t) = x_t + r_d \cos(v_0 t / r_d) \tag{31}$$

$$y_{\text{ref}}(t) = y_t + r_d \sin(v_0 t / r_d) \tag{32}$$

with derivatives

$$\dot{x}_{\text{ref}}(t) = -v_0 \sin(v_0 t / r_d) \tag{33}$$

$$\dot{y}_{\text{ref}}(t) = v_0 \cos(v_0 t / r_d) \tag{34}$$

The simulated trajectories for both the electric field controller and the chemical gradient controller are shown in Figure 5. Videos S1 and S2 are also provided in the Supporting Information showing animations of the corresponding trajectories. Figure 5a shows results using the electric field controller, with the particle being steered around a circular reference trajectory by four electric probes that change the electric potential with time. The simulated trajectory for the chemical gradient controller is shown in Figure 5b, where the particle is steered around a circle by four local chemical reactions that change the solute concentration with time. In both the electrokinetically and diffusiophoretically steered cases, the particle closely follows the reference trajectory with perturbations from the path due to Brownian motion.

Figure 5c gives the control values (electrode voltages) for the electric field controller as a function of time. The control values are approximately sinusoidal, which can be explained with a brief analysis of a 1D controller. The x-direction reference velocity is

$$\dot{x}_{\text{ref}}(t) = -v_0 \sin(v_0 t / r_d). \tag{35}$$

To produce this velocity using a single probe, we substitute this reference velocity into Equation (11), take the magnitude, and rearrange to get

$$q_{\text{eff}}(t) = \frac{2\pi\varepsilon r_i v_0}{\mu_e} \sin(v_0 t / r_d). \tag{36}$$

$q_{\text{eff}}(t)$ is approximately sinusoidal as long as the distance r_i between the probe and particle is approximately constant. This occurs when the radius of the circular trajectory is small compared to the distance between the trajectory and the probe, $r_d \ll L$.

The control values in Figure 5c are truncated sine waves because we constrain the electrode voltages to be positive. Consequently, for motion in the x-direction, a positive voltage on the West electrode create positive velocities and a positive voltage on the East electrode creates negative velocities. The same analysis can be applied to the y-direction and the North and South electrodes. Moreover, if we assume the particle is never farther than a distance L from the electrode, then the maximum charge input will be

$$q_{\text{scale}} = \frac{2\pi\varepsilon L v_0}{\mu_e}, \tag{37}$$

where q_{scale} is the scale for electric charge derived in the Appendix. The charge input q_{scale} can be transformed to a voltage through Equation (9). In Figure 5c, the transformed q_{scale} gives a useful

benchmark for the maximum electrode voltages needed, though it is not exact because of modeling error and Brownian noise.

Figure 5d shows the control values, or reaction rate of each of the chemical probes, for the chemical gradient controller as a function of time. If we assume the diffusion of the solute is much faster than the motion of the particle, we find that the chemical controller becomes analogous to the electric field controller when we replace q_{scale} with g_{scale}. For conciseness, this derivation has been placed in the Appendix. From this analysis, the approximate maximum reaction rate is

$$g_{\text{scale}} = \frac{2\pi D_s L v_0}{\mu_d}. \tag{38}$$

Similar to the electric field controller, the chemical reaction inputs behave like a truncated sine wave with a peak of approximately g_{scale}. This pleasing result provides additional support for our intuition that control with electric fields or chemical gradients are analogous in appropriate conditions.

The electric field controller shows similar performance to Chaudhary and Shapiro in its ability to track the trajectory and deal with Brownian noise. Visual inspection of trajectories also suggests that the chemical gradient controller, with the choice of parameters given in Table I, performs similarly to the electric field controller. Interestingly, the chemical controller only requires N_{pred}=1 for this simulation. A larger prediction horizon is needed when the diffusion coefficient is small or when the distance between the probes and the particle is large. Acceptable values for the prediction horizon under different choices of parameters are discussed in more detail below.

There are several important observations to be made comparing and contrasting the results of the two controllers. First, note that in Figure 5a the particle moves toward lower electric potential, but Figure 5b the particle moves towards higher solute concentration. This is due to the definition of the mobility in each case: the electrophoretic mobility from Equation (3) is defined as the proportionality constant between the electrophoretic velocity and the electric field (negative electric potential gradient), meaning that a positive electrophoretic mobility causes a particle to move down the electric potential gradient. By constrast, the relationship for the diffusiophoretic mobility in Equation (5) does not have the extra negative sign, meaning a positive diffusiophoretic mobility causes the particle to move towards higher concentrations. The sign of the electrophoretic mobility depends on the charge of the particle (with a positive mobility corresponding to a particle with a positive surface charge), so changing the sign of the surface charge will change the sign

of the mobility. In rare cases, the sign on the diffusiophoretic mobility can also change, and the particle will move down the solute gradient instead.[33]

Second, in Figure 5c-d, note the similarities and differences in the electrode voltage and chemical reaction rate inputs. As previously mentioned, both plots can be approximated as truncated sine waves and look strikingly similar, with a couple of notable exceptions. Because of the differing directions of mobility mentioned in the previous paragraph, the North/South input values for the electric field controller (Figure 5c) are reversed in the chemical gradient controller (Figure 5d), as are the East/West inputs. Additionally, the chemical reaction rate inputs are noisier than those of the electric field controller. This latter effect is due to the need to correct for additional Brownian motion when there is a time delay due to solute diffusion.

Third, note the magnitude of the scale bars for electric potential compared to those for concentration. The voltages required for the electric field controller are very small (around 0.02 V). This is because the electrodes are placed close to the particle (0.1 mm compared to 7 mm used by Armani et al.[8]), and because the particle is moving slowly (1 μm/s compared to the maximum of 5 μm/s in Ref. 8). By contrast, the concentrations in the chemical gradient simulations are very large and reach over 3 molar. This violates our assumption of a dilute gradient and is large enough that it could cause solubility issues.

The large concentration required is a significant issue, and we will now discuss ways to mitigate this issue based on our analysis of the chemical gradient controller given above. One way to deal with large concentrations is to incorporate concentration sinks—for example by flushing the solution at the device boundary. Bringing the boundary closer to the control region reduces concentration build-up, but it also introduces modeling errors into our method, because boundary effects are not included in the dynamic model. Another way to decrease the concentration is to reduce the reaction rate that is necessary to induce particle motion. Equation (38) suggests that there are four parameters that can be modified to do so:

1. Select a colloid-solute system with a larger diffusiophoretic mobility, μ_d.

2. Miniaturize the system by decreasing the spacing, L, between the probes and decreasing the average distance between the particle and the probes.

3. Decrease the speed, v_0.

4. Decrease the solute diffusion coefficient, D_s.

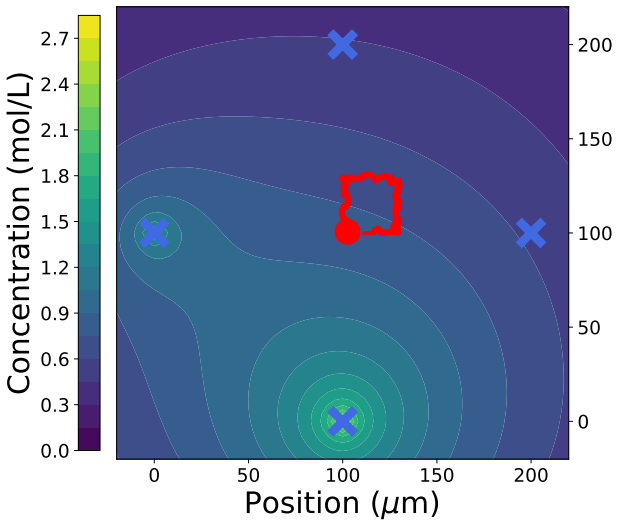

FIG. 6 A particle (red dot) steered on a square-shaped path (red dashed line).

The above choices are not without potential negative consequences. For example, decreasing v_0 too far can lead to problems due to excessive Brownian motion, and decreasing D_s increases the time delay, making the control problem more difficult. A more thorough examination of these trade-offs is discussed below.

B. Other trajectories

The control methods described here are not limited to simple circular trajectories. Accordingly, we demonstrate the flexibility of our method by presenting simulated particle trajectories that follow a square path and a figure-eight. Again, the method is valid for an arbitrary choice of paths and is not limited to these specific choices.

Figure 6 shows a simulated particle moving on a square trajectory. An animation of the time-progression of this trajectory is also available as Video S3 in the Supporting Information. The square had a side length of $r_d = 30\mu m$, with the particle starting in the center of the control region and traveling counter-clockwise at a speed of $v_0 = 1\mu m/s$. The controller is able to accurately produce the desired path, based on visual inspection. However, the corners of the square are slightly rounded off due to a combination of Brownian motion with a limit to the resolution of the shape caused by the 1 second controller time step. As we discuss in the next section, sharp corners require a smaller controller time step to produce accurately.

Figure 7 shows a simulated particle moving in a figure-eight trajectory described by the refer-

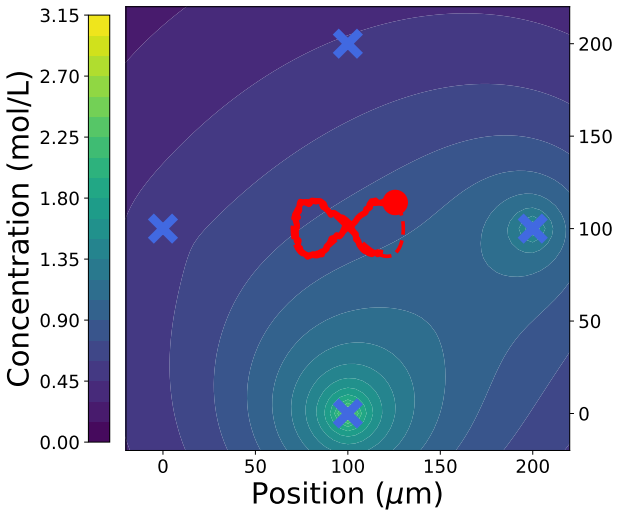

FIG. 7 A particle (red dot) is steered on a figure-eight path (red dashed line).

ence trajectory

$$x_{\text{ref}}(t) = x_t + r_d \sin(v_0 t / r_d) \tag{39}$$

$$y_{\text{ref}}(t) = y_t + r_d \sin(v_0 t / r_d) \cos(v_0 t / r_d), \tag{40}$$

where the shape is centered at (x_t, y_t) and the parameters have values $r_d = 30\mu m$ and $v_0 = 1\mu m/s$. As was the case for the other simulated particle trajectories, an animation of this one is available as Video S4 in the Supporting Information. Like the square trajectory, visual inspection of particle motion indicates that the controller is able to accurately produce the desired path.

Based on the simulations we have performed, paths of almost any shape can be achieved. However, there are some limitations. A particle cannot be moved too close to the probes without violating the assumptions inherent to the method. Also, sharp corners, frequent changes of direction, and small length-scale details of a trajectory are difficult to produce. These difficulties are discussed more rigorously in the following section.

C. Characteristic time scales

Controlling particle trajectories via chemical gradients works well for the combination of the Brownian dynamics model parameters described in Table I and the NMPC algorithm parameters described in Table III. However, certain relationships exist between some of these parameters that must be satisfied in order to accurately reproduce a desired trajectory. In this section, we analyze

TABLE IV Timescales relevant for obtaining a desired trajectory and the value of the timescale using the model and algorithm parameters described above.

Symbol	Time Scale	Value (s)
$\Delta t_{\text{control}}$	Control interval duration	1.00
t_{s}	Diffusion time of solute	1.09
T	Timescale of directed motion	30.0
t_{c}	Diffusion time of particle	917

these relationships in terms of the inherent timescales of the system and use these timescales to postulate rational design rules for parameter selection. The relevant timescales of the system are presented in Table IV. They include the characteristic diffusion time of the solute t_{s}, the characteristic diffusion time of the particle through Brownian motion t_{c}, the timescale of the directed motion following the reference trajectory T, and the control interval of the model predictive controller $\Delta t_{\text{control}}$.

In order to specify design rules, we first need to relate the timescales in Table IV to parameters in the physical model and the control algorithm. The diffusion time of the solute and of the colloid can both be quantified using the relation[45]

$$t = \frac{x_{\text{RMS}}^2}{4D}, \tag{41}$$

where D is a diffusion coefficient and x_{RMS}^2 is the mean-squared displacement of the solute or colloidal particle. The solute diffusion time t_s is a measure of the average delay between the action of controlling the reaction rate at the chemical source and the effect of this input on the particle motion. Accordingly, a useful length scale is the average distance between a probe and the collodial particle, in this case $L/2$. Using this scale as the RMS distance for the solute to travel, the solute diffusion time is

$$t_s = \frac{L^2}{16D_s}. \tag{42}$$

The diffusion time of the colloid t_c is a measure of how long it takes Brownian motion to disrupt the desired trajectory. Assuming (without a loss of generality) that the reference trajectory is a circle with a radius r_d, we care about deviations from the desired trajectory with the length scale αr_d where $\alpha \leq 1$. So, for example, if we want to consider Brownian perturbations that are $10\times$ smaller than the reference trajectory, then we set $\alpha = 0.1$. Using this length scale gives a

particle diffusion time,

$$t_c = \frac{\alpha^2 r_d^2}{4D_c}, \tag{43}$$

where D_c is the diffusion coefficient of the particle defined in the Methods section. For the following analysis, we will set $\alpha = 1$, but we wish to highlight that the timescale t_c has a squared dependence on the desired precision α, meaning a $10\times$ change in precision requires a change in t_c of $100\times$.

The timescale of directed particle motion T quantifies how long the particle moves before changing direction significantly. The motion of a particle moving on a trajectory with a length scale x and a velocity scale v has a timescale x/v. Again, without a loss of generality, we assume a circular trajectory, giving the timescale

$$T = \frac{r_d}{v_0}. \tag{44}$$

Note that for a trajectory that has tight turns, it is appropriate to substitute the minimum radius of curvature for r_d in the analysis.

The control interval duration $\Delta t_{\text{control}}$ is the time step over which the model predictive controller operates. Every $\Delta t_{\text{control}}$ seconds, the position of the particle is measured, reaction rates are calculated by the optimization algorithm, and new reaction rates are applied. Unlike the above parameters, $\Delta t_{\text{control}}$ is not determined by the properties of the colloidal solution. Rather, the control interval is an algorithmic choice bounded by physical limitations. A small value of $\Delta t_{\text{control}}$ is desirable as it gives more frequent feedback and a more responsive controller. However, the control interval must be long enough to encompass particle position measurement, device information transmission, convergence of the optimization algorithm, and actuation of the reaction rates. Practically then, $\Delta t_{\text{control}}$ will be the minimum allowable value given computer vision specifications, reaction chemistry, and the computational speed of the optimization algorithm.

There are four important relationships between these time scales—design rules—that must be satisfied in order to control a particle using chemical gradients via NMPC. If these relationships are not met, then either the inputs required to produce the trajectory get unreasonably large or the controller is not able to accurately produce the trajectory at all. These design rules are described in the following paragraphs and are summarized in Table V.

First, $\Delta t_{\text{control}}$ must be small enough that the resulting path is smooth. At each time step (assum-

TABLE V Design rules for selecting physical parameters.

Rule	Time scales	Physical parameters
1	$\Delta t_{\text{control}} \ll T$	$\Delta t_{\text{control}} \ll \frac{r_d}{v_0}$
2	$t_{\text{s}} < N_{\text{pred}} \Delta t_{\text{control}}$	$\frac{L^2}{16 D_s} < N_{\text{pred}} \Delta t_{\text{control}}$
3	$t_{\text{s}} < T$	$\frac{L^2}{16 D_s} < \frac{r_d}{v_0}$
4	$T \ll t_{\text{c}}$	$\frac{r_d}{v_0} \ll \frac{\alpha^2 r_d^2}{4 D_c}$

ing the particle is moving on the open loop trajectory), the particle travels a distance of $v_0 \Delta t_{\text{control}}$. So to create a smooth path, $v_0 \Delta t_{\text{control}}$ must be small compared to the length scale of the trajectory, r_d. In other words,

$$\Delta t_{\text{control}} \ll \frac{r_d}{v_0} \tag{45}$$

or equivalently

$$\Delta t_{\text{control}} \ll T. \tag{46}$$

When this rule is not met, the particle cannot follow the reference trajectory accurately, and the path of the colloid looks rough in the sense that a polygon looks rough compared to a circle.

Second, the prediction horizon $N_{\text{pred}} \Delta t_{\text{control}}$ should be at least as large as the solute diffusion time t_{s} to enable the NMPC model to adequately capture the behavior of the system. Accordingly, the design rule here is,

$$t_{\text{s}} < N_{\text{pred}} \Delta t_{\text{control}}. \tag{47}$$

The delay between when solute is released from the probe and when its effects reach the particle is on the order of t_{s}. If the prediction horizon is too small, the predictive controller cannot anticipate this delay. This is illustrated in Figure 8, where the coefficient function $\Delta h_j(t)$ from Equation 26 is shown along with t_{s} and $N_{\text{pred}} \Delta t_{\text{control}}$. When calculating a future input, the controller only sees the portion of the coefficient function highlighted in blue.[46] A prediction horizon that is too short will lead to a "short-sighted" controller that sacrifices future performance for immediate performance. This results in very large oscillations in the reaction rates that can cause the particle to move off the trajectory or an ill-behaved optimization problem that the algorithm is not able to solve.

The third design rule is

$$t_{\text{s}} < T, \tag{48}$$

which allows controller inputs to act independently and not interfere with one another. Recall

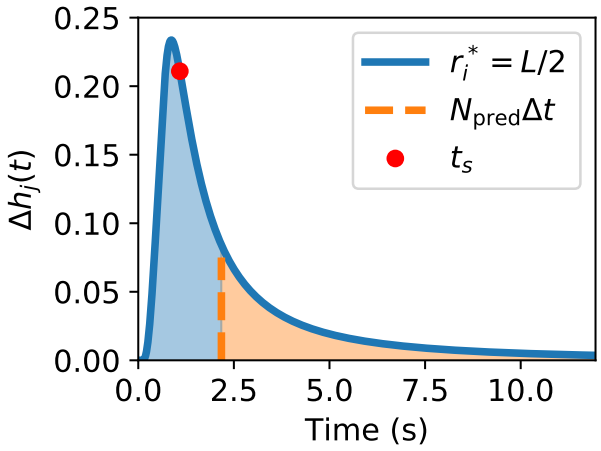

FIG. 8 The coefficient function $\Delta h_j(t)$ from Equation 26 is shown in blue, along with t_s (red dot) and $N_{\text{pred}}\Delta t_{\text{control}}$ (orange dashed line). The portion of $\Delta h_j(t)$ captured by the NMPC controller when calculating a future input is shaded in blue, and the remainder is shaded in orange.

that the motion timescale T quantifies how long the particle moves before significantly changing direction. If controller inputs are slow to decay, as characterized by t_s, then they can act against the desired motion. This can lead to unrealistically large control inputs as the controller tries to compensate. For example, when we tested a case with t_s=14.49 and T = 15.0, the controller inputs reached $2g_{\text{scale}}$, over twice the value predicted by our earlier analysis. In a simulation with t_s=29.0 and T = 15.0, the inputs reached $10g_{\text{scale}}$, over 10 times the value predicted. In some situations, large reaction rates may be acceptable as long as the concentration of the solute does not get too large, but such high reaction rates may be difficult to produce in practice.

The fourth and final design rule

$$T \ll t_c \tag{49}$$

ensures that the directed motion of the particle can overcome random Brownian forces. The particle moves a distance of r_d by directed motion in time T, and Brownian motion induces a perturbation of distance αr_d from the desired path in time t_c. Accordingly, directed motion will dominate Brownian motion for a given value of α if Equation (49) is satisfied.

The design rules in Equations (46)–(49) share four common degrees of freedom that must be selected to ensure the four rules are met. We select $\Delta t_{\text{control}}$ to be small enough to meet Rule 1 and N_{pred} large enough to meet Rule 2, with the understanding that these parameters must be tested to check that the device is able to make measurements and the controller is able to solve the optimization problem in real time. Then, two of the following variables must be selected and

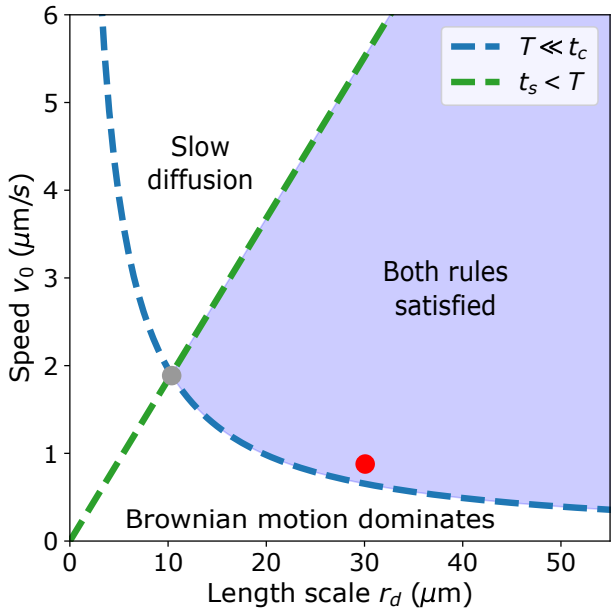

FIG. 9 Plot of the parameter space in terms of speed v_0 and length scale r_d of a desired trajectory for a fixed set of system parameters: L, D_s, and D_c (given in Table I). The region of the parameter space that satisfies the design rules are shaded in blue, while other combinations result in either too much Brownian motion or too slow of diffusion. The design rules are defined using $t_c = 20T$ and $T = 5t_s$. The parameters chosen for the simulations given in the Results section above are represented by a red dot. The minimum length scale predicted by this theory is shown as a gray dot.

balanced to meet the other two rules: r_d, v_0, L, D_s, or D_c. The remaining variables may be chosen freely.

Figure 9 shows an example of the acceptable range of parameters v_0 (speed of the desired trajectory) and r_d (characteristic size of the desired trajectory) using Rules 3 and 4 when L, D_s, and D_c are set according to Table I. Speeds that are too large or trajectories that are too small can violate Rule 3, where diffusion is too slow for the controller to actuate. Alternatively, speeds or trajectories that are too small can violate Rule 4, and Brownian motion will dominate the controlled motion. The combination of these rules predicts a smallest possible trajectory, which occurs near 10 μm in Figure 9. Note, however, that these rules are "fuzzy" because they depend on inequalities and asymptotic relationships. As such, these design rules should be seen as guiding principles rather than strict limits.

Figure 10 shows an example of how changing the size of the particle R_c affects the allowable trajectories. Since R_c is inversely proportional to D_c, a smaller size will increase the colloid diffusivity and therefore increase the magnitude of Brownian motion. Figure 10 shows that decreasing

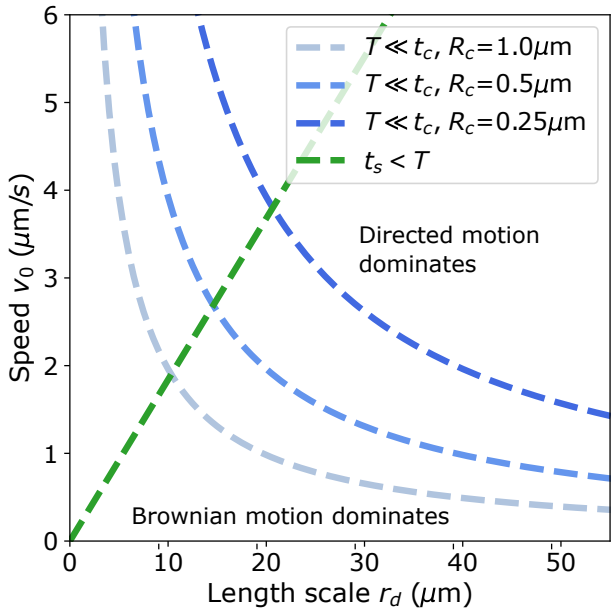

FIG. 10 Plot of the parameter space in terms of speed v_0 and length scale r_d of a desired trajectory for three values of the paramter R_c. The region of parameter space that satisfies the design rules is located below the green line and above the blue line for a given value or R_c.

R_c shifts the $T \ll t_c$ curve up towards larger speeds and trajectory length scales, constricting the allowable region. A smaller R_c leads to larger Brownian fluctuations, which means that a larger v_0 and/or r_d must be chosen for the directed motion to overcome random motion. A similar shift occurs when decreasing the viscosity of the fluid medium or increasing the temperature.

IV. CONCLUSION AND FUTURE WORK

We have successfully demonstrated a method for steering a colloidal particle along an arbitrary user-defined trajectory using chemical gradients by applying model predictive control to external concentration sources. To the best of our knowledge, this is the first demonstration of an algorithm for top-down control of particle trajectories using chemical sources. We tested the method using Brownian dynamics simulations that assume diffusiophoresis is the sole actuatable force, and the controller is able to steer a colloid along micrometer scale trajectories even in the presence of significant Brownian noise. In this respect, the method compares favorably with methods that have used electrokinetic actuation mechanisms to manipulate particle trajectories.

Additionally, we have developed design rules for particle steering based on an analysis of the characteristic time scales present in system. These rules are based on fundamental principles of

control theory and colloidal physics: (1) controller parameters should ensure that the trajectory is smooth; (2) the prediction horizon should capture most of the future behavior of the solute; (3) solvent diffusion (chemical signalling) should be faster than directed motion; and (4) directed motion should dominate Brownian fluctuations. These principles are expressed mathematically as Equations (46)–(49), and provide a recipe for determining limits of e.g., the reference trajectory size and speed once the physical parameters of the system have been determined.

Along with these successes, there are several avenues for future improvements. The most glaring weakness of the method is the potential need for large solute concentrations. For example, the trajectories in Figures 5, 6, 7 require maximum concentrations as large as 2M-3M, likely beyond the solubility limits of many common solutes. Even if diffuisophoretic solutions are available with such solubilities, the physical model we describe herein assumes dilute solutions and should be modified to account for effects of large concentrations. Using the aforementioned design rules as a guide, lowering the maximum concentration can be achieved in cases where the particle has a large diffusiophoretic mobility or the solute has a small diffusion coefficient. Ionic solutes could potentially satisfy both of these criteria,[35] making them a promising candidate for investigation. Notably, our models here are restricted to non-ionic solutes, and we leave the development of similar methods for ionic solutes for future work.

The models and methods we develop here also neglect potentially significant hydrodynamic effects. Large concentration gradients may induce density-driven convection, and diffuisoosmotic flows at device walls may also be important at lower concentrations.[47] An explicit hydrodynamics model could potentially be a helpful addition to the controller. Alternatively, advanced control methods such as moving horizon estimation could be used to account for unknown effects or uncertain parameters.[48] Noting that errors from hydrodynamic effects or others sources are already at least partially compensated for by the feedback term in the controller, this latter approach may prove especially fruitful.

Additionally, in order to implement our method in an experimental system, the optimization problem must be solvable in real time. We consider this problem beyond the scope of the present paper. However, despite the apparent complexity of the model and method, speedy numerical implementations are realistic with low-level languages such as Fortran or C++.

Finally, an important future direction is using this method to steer multiple particles. Methods for doing so will need to include interparticle interactions and hydrodynamic interactions. Also, each additional particle will add N_{pred} decision variables, potentially making it difficult to solve

the optimization problem in real time.

With these potential improvements in mind, the chemical gradient controller we have developed provides a critical step towards developing microfluidic systems that can manipulate particles through chemical micro-actuation. Such a method can provide new and orthogonal degrees of freedom beyond electrokinetic methods to steer particles, and provides a foundation for the future study of biomimetic self-assembly using complex chemical gradients and signalling mechanisms.

APPENDIX: CHARACTERISTIC SCALES

In this appendix, we first perform a brief dimensional analysis of the equations of motion for an electrophoretic particle to derive the characteristic scale for electric charge. We then use dimensional analysis for a diffusiophoretic particle to derive a characteristic scale for reaction rate and show that, under a pseudo-steady state assumption, there is a direct analogy between electric fields and chemical gradients.

Non-dimensionalizaion of electric charge

We begin by restating the equations for the electrophoretic particle in an electric potential modeled by the charge simulation method. This has the equations

$$\frac{\partial \boldsymbol{X}}{\partial t} = \mu_e \nabla \Phi + \sqrt{2D_c}\boldsymbol{\xi} \tag{50}$$

and

$$\nabla \Phi = -\frac{q_i(t)}{2\pi\varepsilon r^2}\boldsymbol{r} \tag{51}$$

for a particle at position $\boldsymbol{r}$ and a single charge located at the origin. We may simplify by substituting Equation 51 into Equation 50, giving

$$\frac{\partial \boldsymbol{X}}{\partial t} = -\frac{\mu_e q}{2\pi\varepsilon r^2}\boldsymbol{r} + \sqrt{2D_c}\boldsymbol{\xi}. \tag{52}$$

We assume that the characteristic length of electric potential gradients are on the order of L, the distance between electrodes, and that the characteristic speed of the particle set by the control algorithm is v_0. We seek to derive the characteristic electric charge q^* through dimensional

analysis.

Using the characteristic scales, we define dimensionless variables

$$\tilde{t} = \frac{t v_0}{L}, \tilde{\boldsymbol{X}} = \frac{\boldsymbol{X}}{L}, \tilde{q} = \frac{q}{q^*},$$
$$\tilde{\boldsymbol{\xi}} = \sqrt{\frac{L}{v_0}}\boldsymbol{\xi}, \tilde{r} = \frac{r}{L}.$$

Substituting these equations into Equation 52 gives

$$\frac{\partial \tilde{\boldsymbol{X}}}{\partial \tilde{t}} = \frac{-\mu_e q^* \tilde{q}(\tilde{t})}{2\pi v_0 \varepsilon L \tilde{r}^2}\tilde{\boldsymbol{r}} + \sqrt{\frac{2D_c}{v_0 L}}\tilde{\boldsymbol{\xi}}. \tag{53}$$

Setting the first dimensionless group to one gives us

$$q^* = \frac{\varepsilon v_0 L}{\mu_e}. \tag{54}$$

This leaves the equation of motion as

$$\frac{\partial \tilde{\boldsymbol{X}}}{\partial \tilde{t}} = \frac{\tilde{q}(\tilde{t})}{2\pi \tilde{r}^2}\tilde{\boldsymbol{r}}. \tag{55}$$

Finally, it is convenient to eliminate the factor of 2π. Thus we define the charge scale

$$q_{\text{scale}} = 2\pi q^* = \frac{2\pi\varepsilon v_0 L}{\mu_e}. \tag{56}$$

Non-dimensionalization of chemical reactions and pseudo-steady state

Now we will perform a dimensional analysis of the equations of motion for a diffusiophoretic particle. To do so, we restate the model for a diffusiophoretic particle coupled to a reaction-diffusion equation

$$\frac{\partial \boldsymbol{X}}{\partial t} = \mu_d \nabla C + \sqrt{2D_c}\boldsymbol{\xi} \tag{57}$$
$$\frac{\partial C}{\partial t} = D_s \nabla^2 C + g(t)\delta(\boldsymbol{r} - \boldsymbol{r}_0) \tag{58}$$

assuming in the diffusion equation that the chemical source term is a point source located at $\boldsymbol{r}_0$. We assume that the characteristic length of concentration gradients are on the order of L, the distance between probes, and that the characteristic speed of the particle set by the control algorithm is v_0. We seek to use dimensional analysis to find the characteristic concentration c^* and the characteristic reaction rate g^*.

Using the characteristic scales, we define dimensionless variables

$$\tilde{t} = \frac{tv_0}{L}, \tilde{\boldsymbol{X}} = \frac{\boldsymbol{X}}{L}, \tilde{C} = \frac{C}{C^*}, \tilde{g} = \frac{g}{g^*},$$
$$\tilde{\nabla} = L\nabla, \tilde{\boldsymbol{\xi}} = \sqrt{\frac{L}{v_0}}\boldsymbol{\xi}, \tilde{\delta} = L^d\delta$$

where d is the number of spatial dimensions, which in our present case is 2. Substituting these equations into Equations (57) and (58) gives

$$\frac{\partial \tilde{\boldsymbol{X}}}{\partial \tilde{t}} = \frac{\mu_d c^*}{Lv_0}\tilde{\nabla}\tilde{C} + \sqrt{\frac{2D_c}{v_0 L}}\tilde{\boldsymbol{\xi}} \tag{59}$$

$$\frac{v_0 L}{D_s}\frac{\partial \tilde{C}}{\partial \tilde{t}} = \tilde{\nabla}^2\tilde{C} + \frac{g^*}{D_s c^*}\tilde{g}(\tilde{t})\tilde{\delta}. \tag{60}$$

Setting the first dimensionless group to one in Equation (59) allows us to identify the characteristic concentration

$$c^* = \frac{v_0 L}{\mu_d}, \tag{61}$$

and the second dimensionless group contains the colloid Péclet number

$$\mathrm{Pe_c} = \frac{v_0 L}{D_c}. \tag{62}$$

The first dimensionless group in Equation (60) is a type of solute Péclet number with respect to colloid motion

$$\mathrm{Pe_s} = \frac{v_0 L}{D_s}, \tag{63}$$

and setting the second group to one identifies the characteristic reaction rate scale

$$g^* = D_s c^* = \frac{D_s v_0 L}{\mu_d}. \tag{64}$$

If colloid motion is fast relative to both solute diffusion and colloid diffusion, the solute con-

centration is at a pseudo-steady state and one may solve for the particle velocity. In this case, the solute Péclet number is small, reducing Equation (60) to the Poisson equation

$$\tilde{\nabla}^2 \tilde{C} = -\tilde{g}(\tilde{t})\tilde{\delta} \tag{65}$$

and the colloid Péclet number is small, reducing Equation (59) to the drift velocity

$$\frac{\partial \tilde{\boldsymbol{X}}}{\partial \tilde{t}} = \tilde{\nabla}\tilde{C} \tag{66}$$

If $\tilde{g}$ changes slowly, then the solution to Equation (65) in 2D is well known[43]

$$\tilde{C} = \frac{-\tilde{g}(\tilde{t})}{2\pi}\ln(\tilde{r}) \tag{67}$$

to within an arbitrary constant with $\tilde{r} = ||\tilde{\boldsymbol{r}}||$. Substituting this into Equation (66) gives the drift velocity

$$\frac{\partial \tilde{\boldsymbol{X}}}{\partial \tilde{t}} = -\frac{\tilde{g}(\tilde{t})}{2\pi\tilde{r}^2}\tilde{\boldsymbol{r}}. \tag{68}$$

This equation is directly analogous to Equation 55, but with $\tilde{q}(\tilde{t})$ replaced with $\tilde{g}(\tilde{t})$. Thus, when the pseudo-steady-state assumption is met, we expect the behavior of the chemical controller to resemble the behavior of the electric controller.

Finally, it is again convenient to use a reaction rate scale that will eliminate the factor of 2π. Thus we define

$$g_{\text{scale}} = 2\pi g^* = \frac{2\pi D_s v_0 L}{\mu_d}. \tag{69}$$

CONFLICTS OF INTEREST

There are no conflicts to declare.

ACKNOWLEDGEMENTS

We gratefully acknowledge financial support from Brigham Young University and the Simmons Research Endowment at Brigham Young University. We also acknowledge computational resources from the BYU Office of Research Computing.

REFERENCES

[1]C. Ropp, Z. Cummins, S. Nah, S. Qin, J. H. Seog, S. B. Lee, J. T. Fourkas, B. Shapiro, and E. Waks, "Fabrication of nanoassemblies using flow control," Nano Lett. **13**, 3936–3941 (2013).

[2]L. Y. Wu, D. Di Carlo, and L. P. Lee, "Microfluidic self-assembly of tumor spheroids for anti-cancer drug discovery," Biomed. Microdevices **10**, 197–202 (2008).

[3]M. Mirbagheri, V. Adibnia, B. R. Hughes, S. D. Waldman, X. Banquy, and D. K. Hwang, "Advanced cell culture platforms: a growing quest for emulating natural tissues," Mater. Horiz. **6**, 45–71 (2019).

[4]M. Filippi, T. Buchner, O. Yasa, S. Weirich, and R. K. Katzschmann, "Microfluidic Tissue Engineering and Bio-actuation," Adv. Mater. , 2108427 (2022).

[5]S. Chaudhary and B. Shapiro, "Arbitrary steering of multiple particles independently in an electro-osmotically driven microfluidic system," IEEE Trans. Control Syst. Technol. **14**, 669–680 (2006).

[6]M. Armani, S. Chaudhary, R. Probst, and B. Shapiro, "Using feedback control and microfluidics to steer individual particles," Proc. IEEE Int. Conf. Micro Electro Mech. Syst. (MEMS) , 855–858 (2005).

[7]R. Probst and B. Shapiro, "Three-dimensional electrokinetic tweezing: device design, modeling, and control algorithms," J. Micromech. Microeng. **21**, 027004 (2011).

[8]M. D. Armani, S. V. Chaudhary, R. Probst, and B. Shapiro, "Using feedback control of microflows to independently steer multiple particles," J. Microelectromech. Syst. **15**, 945–956 (2006).

[9]P. P. Mathai, A. J. Berglund, J. Alexander Liddle, and B. A. Shapiro, "Simultaneous positioning and orientation of a single nano-object by flow control: theory and simulations," New J. Phys. **13**, 013027 (2011).

[10]C. Ropp, R. Probst, Z. Cummins, R. Kumar, A. J. Berglund, S. R. Raghavan, E. Waks, and B. Shapiro, "Manipulating quantum dots to nanometer precision by control of flow," Nano Lett. **10**, 2525–2530 (2010).

[11]J. J. Juárez and M. A. Bevan, "Feedback controlled colloidal self-assembly," Adv. Funct. Mater. **22**, 3833–3839 (2012).

[12]M. A. Grover, D. J. Griffin, X. Tang, Y. Kim, and R. W. Rousseau, "Optimal feedback control of batch self-assembly processes using dynamic programming," J. Process Control **88**, 32–42

(2020).

[13]R. Xie and X. Y. Liu, "Electrically directed on-chip reversible patterning of two-dimensional tunable colloidal structures," Adv. Funct. Mater. **18**, 802–809 (2008).

[14]M. R. Jones, N. C. Seeman, and C. A. Mirkin, "Programmable materials and the nature of the DNA bond," Science **347**, 1260901 (2015).

[15]S. Kunche, H. Yan, A. L. Calof, J. S. Lowengrub, and A. D. Lander, "Feedback, Lineages and Self-Organizing Morphogenesis," PLoS Comput. Biol. **12**, e1004814 (2016).

[16]J. A. Paulson, A. Mesbah, X. Zhu, M. C. Molaro, and R. D. Braatz, "Control of self-assembly in micro- and nano-scale systems," J. Process Control **27**, 38–49 (2015).

[17]C. Kaspar, B. J. Ravoo, W. G. van der Wiel, S. V. Wegner, and W. H. Pernice, "The rise of intelligent matter," Nature **594**, 345–355 (2021).

[18]B. G. Van Ravensteijn, I. K. Voets, W. K. Kegel, and R. Eelkema, "Out-of-Equilibrium Colloidal Assembly Driven by Chemical Reaction Networks," Langmuir **36**, 10639–10656 (2020).

[19]S. Joshi and D. Marla, "Electrochemical Micromachining," in *Comprehensive Materials Processing*, Vol. 11 (Elsevier, 2014) pp. 373–403.

[20]R. Kosloff, S. Rice, P. Gaspard, S. Tersigni, and D. Tannor, "Wavepacket dancing: Achieving chemical selectivity by shaping light pulses," Chem. Phys. **139**, 201–220 (1989).

[21]C. Y. Jin, Z. Li, R. S. Williams, K.-C. Lee, and I. Park, "Localized Temperature and Chemical Reaction Control in Nanoscale Space by Nanowire Array," Nano Lett. **11**, 4818–4825 (2011).

[22]L. R. Lechlitner and O. Annunziata, "Macromolecule Diffusiophoresis Induced by Concentration Gradients of Aqueous Osmolytes," Langmuir **34**, 9525–9531 (2018).

[23]B. M. Alessio, S. Shim, E. Mintah, A. Gupta, and H. A. Stone, "Diffusiophoresis and diffusioosmosis in tandem: Two-dimensional particle motion in the presence of multiple electrolytes," Phys. Rev. Fluid **6**, 054201 (2021).

[24]K. Lee, J. Lee, D. Ha, M. Kim, and T. Kim, "Low-electric-potential-assisted diffusiophoresis for continuous separation of nanoparticles on a chip," Lab Chip **20**, 2735–2747 (2020).

[25]S. Shim, "Diffusiophoresis, Diffusioosmosis, and Microfluidics: Surface-Flow-Driven Phenomena in the Presence of Flow," Chem. Rev. **122**, 6986–7009 (2022).

[26]E. Camacho, C. Bordons, and C. Alba, *Model Predictive Control*, Advanced Textbooks in Control and Signal Processing (Springer London, 2004).

[27]I. Matei, S. Nelaturi, E. M. Chow, J. P. Lu, J. A. Bert, and L. S. Crawford, "Micro-Scale Chiplets Position Control," J. Microelectromech. Syst. **28**, 643–655 (2019).

[28]J. Anderson, "Colloid Transport By Interfacial Forces," Annu. Rev. Fluid Mech. **21**, 61–99 (1989).

[29]K. D. Dorfman, D. Gupta, A. Jain, A. Muralidhar, and D. R. Tree, "Hydrodynamics of DNA confined in nanoslits and nanochannels," Eur. Phys. J. Spec. Top. **223**, 3179–3200 (2014).

[30]H. C. Öttinger, *Stochastic Processes in Polymeric Fluids* (Springer Berlin Heidelberg, 1996).

[31]D. T. Gillespie, "The mathematics of Brownian motion and Johnson noise," Am. J. Phys. **64**, 225–240 (1996).

[32]R. F. Probstein, *Phisiochemical Hydrodynamics, An Introduction*, Vol. 1 (John Wiley & Sons, Inc., Hoboken, NJ, USA, 1994) pp. 211–236.

[33]J. T. Ault, P. B. Warren, S. Shin, and H. A. Stone, "Diffusiophoresis in one-dimensional solute gradients," Soft Matter **13**, 9015–9023 (2017).

[34]S. Shin, E. Um, B. Sabass, J. T. Ault, M. Rahimi, P. B. Warren, and H. A. Stone, "Size-dependent control of colloid transport via solute gradients in dead-end channels," Proc. Natl. Acad. Sci. **113**, 257–261 (2016).

[35]J. S. Paustian, C. D. Angulo, R. Nery-Azevedo, N. Shi, A. I. Abdel-Fattah, and T. M. Squires, "Direct measurements of colloidal solvophoresis under imposed solvent and solute gradients," Langmuir **31**, 4402–4410 (2015).

[36]A. Greenbaum and T. P. Chartier, *Numerical Methods: Design, Analysis, and Computer Implementation of Algorithms* (Princeton University Press, USA, 2012).

[37]N. H. Malik, "A Review of the Charge Simulation Method and its Applications," IEEE Trans. Electr. Insul. **24**, 3–20 (1989).

[38]M. M. A. Salama and R. Hackam, "Voltage and Electric Field Distributions and Discharge Inception Voltage in Insulated Conductors," IEEE Power Eng. Rev. **PER-4**, 25–26 (1984).

[39]H. Singer, H. Steinbigler, and P. Weiss, "A Charge Simulation Method for the Calculation of High Voltage Fields," IEEE Trans. Power App. Syst. **PAS-93**, 1660–1668 (1974).

[40]P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, A. Vijaykumar, A. P. Bardelli, A. Rothberg, A. Hilboll, A. Kloeckner, A. Scopatz, A. Lee, A. Rokem, C. N. Woods, C. Fulton, C. Masson, C. Häggström, C. Fitzgerald, D. A. Nicholson, D. R. Hagen, D. V. Pasech-

nik, E. Olivetti, E. Martin, E. Wieser, F. Silva, F. Lenders, F. Wilhelm, G. Young, G. A. Price, G. L. Ingold, G. E. Allen, G. R. Lee, H. Audren, I. Probst, J. P. Dietrich, J. Silterra, J. T. Webber, J. Slavič, J. Nothman, J. Buchner, J. Kulick, J. L. Schönberger, J. V. de Miranda Cardoso, J. Reimer, J. Harrington, J. L. C. Rodríguez, J. Nunez-Iglesias, J. Kuczynski, K. Tritz, M. Thoma, M. Newville, M. Kümmerer, M. Bolingbroke, M. Tartre, M. Pak, N. J. Smith, N. Nowaczyk, N. Shebanov, O. Pavlyk, P. A. Brodtkorb, P. Lee, R. T. McGibbon, R. Feldbauer, S. Lewis, S. Tygier, S. Sievert, S. Vigna, S. Peterson, S. More, T. Pudlik, T. Oshima, T. J. Pingel, T. P. Robitaille, T. Spura, T. R. Jones, T. Cera, T. Leslie, T. Zito, T. Krauss, U. Upadhyay, Y. O. Halchenko, and Y. Vázquez-Baeza, "SciPy 1.0: fundamental algorithms for scientific computing in Python," Nat. Methods **17**, 261–272 (2020).

[41]J. Pannek and K. Worthmann, "Reducing the prediction horizon in NMPC: An algorithm based approach," IFAC Proc. Vol. **44**, 7969–7974 (2011).

[42]S. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," Control Eng. Pract. **11**, 733–764 (2003).

[43]R. Haberman, *Applied Partial Differential Equations: with Fourier Series and Boundary Value Problems (4th ed.)* (Prentice Hall, 2004).

[44]C. Zukoski and D. Saville, "An experimental test of electrokinetic theory using measurements of electrophoretic mobility and electrical conductivity," J. Colloid Interface Sci. **107**, 322–333 (1985).

[45]C. Gardiner, *Stochastic methods: a handbook for the natural and social sciences* (Berlin: Springer, 2009).

[46]The orange shaded region is modeled only for past inputs, which are static and cheap to compute, and not for future inputs, which are variable and therefore expensive.

[47]I. Williams, S. Lee, A. Apriceno, R. P. Sear, and G. Battaglia, "Diffusioosmotic and convective flows induced by a nonelectrolyte concentration gradient," Proc. Natl. Acad. Sci. **117**, 25263–25271 (2020).

[48]E. L. Haseltine and J. B. Rawlings, "Critical Evaluation of Extended Kalman Filtering and Moving-Horizon Estimation," Ind. Eng. Chem. Res. **44**, 2451–2460 (2005).