Soft Matter

PAPER

View Article Online
View Journal | View Issue

Cite this: *Soft Matter*, 2024, **20**, 8581

Model predictive control of non-interacting active Brownian particles†

Titus Quah, Kevin J. Modica, James B. Rawlings * and Sho C. Takatori *

Received 27th July 2024,
Accepted 7th October 2024

DOI: 10.1039/d4sm00902a

rsc.li/soft-matter-journal

Active matter systems are strongly driven to assume non-equilibrium distributions owing to their self-propulsion, *e.g.*, flocking and clustering. Controlling the active matter systems' spatiotemporal distributions offers exciting applications such as directed assembly, programmable materials, and microfluidic actuation. However, these applications involve environments with coupled dynamics and complex tasks, making intuitive control strategies insufficient. This necessitates the development of an automatic feedback control framework, where an algorithm determines appropriate actions based on the system's current state. In this work, we control the distribution of active Brownian particles by applying model predictive control (MPC), a model-based control algorithm that predicts future states and optimizes the control inputs to drive the system along a user-defined objective. The MPC model is based on the Smoluchowski equation with a self-propulsive convective term and an actuated spatiotemporal-varying external field that aligns particles with the applied direction, similar to a magnetic field. We apply the MPC framework to control a Brownian dynamics simulation of non-interacting active particles and illustrate the controller capabilities with two objectives: splitting and juggling sub-populations, and polar order flocking control.

1 Introduction

Active matter is made up of units that convert energy from their surroundings into self-directed motion.[1–4] Interactions between these units can lead to complex collective behaviors such as flocking or swarming.[5–8] Active matter systems include dense swarms of micro- or nanoscale robots that can be used for directed assembly,[9,10] programmable materials,[11,12] and drug delivery.[13–15] To these ends, researchers have developed various controllable active matter systems that can be actuated *via* magnetic,[16–18] optical,[19–24] electric,[25,26] or acoustic fields.[27] While these actuated systems offer solutions for manipulating active matter, they also raise a new question: how do we determine the best control actions? In other words, how can we design a control policy that directs the system to achieve a user-specified task?

In experiments and simulations, heuristic feedback policies have been successful in controlling single agents to trace paths, populations to accumulate into desired shapes, active nematic defects to follow trajectories and active nematic flows to maintain user-specified speeds.[25,28–34] However, these policies are limited to simple tasks and require both intuition and trial and error, which would not generalize well to complicated tasks, or systems with interactions between agents. Other approaches include model-free reinforcement learning, which designs a policy by interacting with the system and learning from the results.[35–40] However, this approach is data-intensive and may not be feasible for systems with many degrees of freedom or where data collection is expensive.

The optimal control problem (OCP) provides a framework to design control policies that leverage a system model to predict future states and optimize control inputs to achieve a specific objective. Solving the OCP involves determining the optimal control inputs that minimize a predefined objective function for a given system over a specified time horizon. One approach to solve the OCP is to derive the necessary conditions for optimality. This yields a two-point boundary problem with Euler–Lagrange equations which can be solved to obtain the optimal control inputs for a given initial condition.[41] In active matter systems, these Euler–Lagrange equations have been solved numerically to design optimal open-loop policies to control active nematics, active droplets and motility-induced phase separation.[42–46] However, handling inequality constraints and solving the two-point boundary value problem is computationally expensive and ill-suited for real-time control.[47]

To handle constraints and achieve real-time control, a useful approximate solution to the OCP is model predictive control (MPC), which discretizes the system dynamics and solves a nonlinear optimization problem at every sample time.[48] MPC

Department of Chemical Engineering, University of California, Santa Barbara, Santa Barbara, CA 93106, USA. E-mail: jbraw@ucsb.edu, stakatori@ucsb.edu

† Electronic supplementary information (ESI) available. See DOI: https://doi.org/10.1039/d4sm00902a

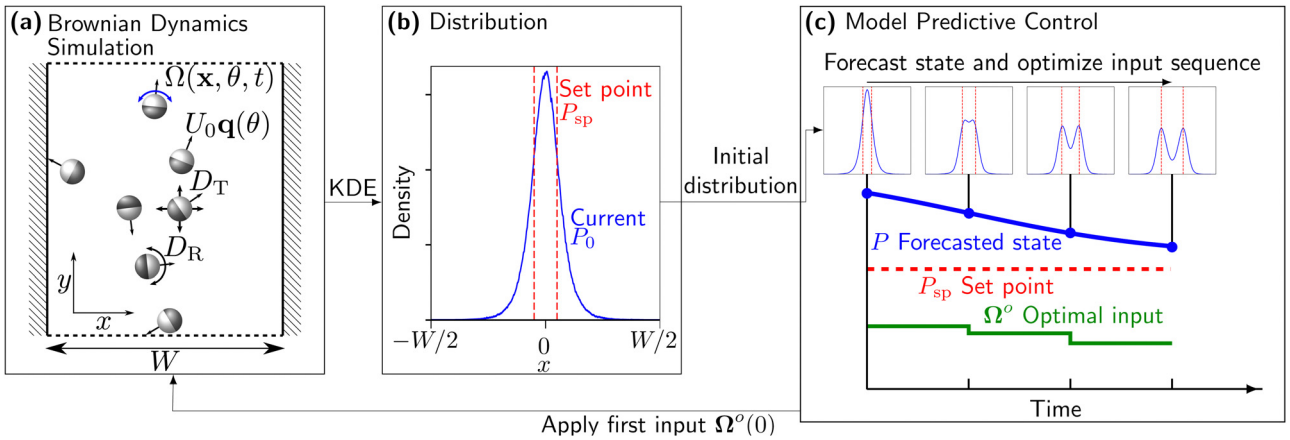

Fig. 1 Feedback control framework for controlling a population of active Brownian particles (ABPs) to split into two equal populations *via* model predictive control (MPC). (a) A Brownian dynamics (BD) simulation of confined ABPs with self-propulsion speed U_0, and translational and rotational diffusion characterized by D_T and D_R, respectively. In addition, the ABPs are actuated by a spatiotemporally varying magnetic-like field $\Omega(\mathbf{x},\theta,T)$ that aligns particles in the direction of the field, *i.e.* the input to the system. (b) From measurements of the ABPs in the BD simulation, we estimate the initial state, *i.e.*, the one particle probability distribution function $P_0(\mathbf{x},\theta,t)$ of ABP positions and orientations using kernel density estimation. (c) Using model predictive control (MPC), we determine the optimal control input sequence $\mathbf{\Omega}^o$ by simultaneously forecasting the time-evolution of the system's probability distribution $P(\mathbf{x},\theta,t)$, starting from the initial distribution $P_0(\mathbf{x},\theta)$, and optimizing the sequence of actuator inputs. The goal is to minimize the distance between the predicted evolution of the system and a desired set point P_{sp}. Finally, we apply the first value of the optimized input sequence $\mathbf{\Omega}^o(0)$ to the BD simulation.

allows for the incorporation of control constraints and can be solved in real-time to perform feedback control which is helpful to handle disturbances and has been applied to a variety of systems such as chemical processes, robotics, and flow control.[47,49,50]

In this work, we apply MPC to control the number density and orientational moments of non-interacting active Brownian particles (ABPs) that are controlled by alignment with a spatiotemporal-varying external field. As outlined by Fig. 1, we demonstrate that MPC on the probability distributions of position and orientation can be used to control a particle-level Brownian dynamics (BD) simulation of ABPs to accomplish various tasks such as population splitting and velocity control.

2 Physical model and model predictive control framework

2.1 Modeling active Brownian particles (ABPs)

Consider N non-interacting ABPs in two dimensions confined between two parallel plates spaced W units apart as shown in Fig. 1a. Each particle has position $\mathbf{x}_j = [x\ y]^T$ and orientation θ_j where the subscript $j \in [1,N]$ denotes the particle's index. The collection of the particles' position and orientations is denoted the system's state $\mathbf{X}(t) = [\mathbf{x}_j^T\ \theta_j]^T$. The particles self-propel with velocity $U_0\mathbf{q}_j$, where U_0 is the self-propulsive speed and $\mathbf{q}_j$ is the unit vector pointing in the direction of the particle's orientation, *i.e.*, $\mathbf{q}_j = [\cos(\theta_j)\ \sin(\theta_j)]^T$. Additionally, we consider an actuated spatiotemporal-varying external field that aligns the particles with the field direction $\hat{\mathbf{H}}(\mathbf{x},t) = [\Omega_x(\mathbf{x},t)\ \Omega_y(\mathbf{x},t)]^T$ which results in an induced angular velocity of $\Omega_j = \Omega(\mathbf{x}_j,\theta_j,t) = \Omega_x(\mathbf{x}_j,t)\sin(\theta_j) + \Omega_y(\mathbf{x}_j,t)\cos(\theta_j)$. These types of aligning external fields have been used to control the orientation of the particles.[17,51,52] Note that a positive $\Omega_x(\mathbf{x}_j,t)$ aligns the particles in the positive x-direction, while a negative value induces alignment in the opposite direction. Thus, we refer to $\Omega_x(\cdot)$ as the x-field. Similarly, $\Omega_y(\mathbf{x}_j,t)$ governs alignment along the y-axis in an analogous manner and we refer to $\Omega_y(\cdot)$ as the y-field. The particles are subject to fluctuations that yield translational and rotational diffusion D_T and D_R, respectively. The dynamics of the particles are governed by overdamped Langevin dynamics, *i.e.*, Browian dynamics:

$$\frac{d\mathbf{x}_j}{dt} = U_0\mathbf{q}_j - \frac{D_T}{k_BT}\nabla_j V(\mathbf{x}_j) + \mathbf{W}_{T,j}(t) \quad (1)$$

$$\frac{d\theta_j}{dt} = \Omega_j + W_{R,j}(t), \quad (2)$$

where k_BT is the thermal energy scale; $V(\mathbf{x}_j)$ is the potential energy that keeps the particle confined between the plates; $\mathbf{W}_{T,j}(t)$ and $W_{R,j}(t)$ are Wiener processes with white noise statistic $\overline{\mathbf{W}_{T,j}} = \mathbf{0}$, $\overline{\mathbf{W}_{T,j}(0),\mathbf{W}_{T,j}(t)} = 2D_T\delta(t)\mathbf{I}$, $\overline{W_{R,j}} = 0$, and $\overline{W_{R,j}(0),W_{R,j}(t)} = 2D_R\delta(t)$; $\mathbf{I}$ is the identity matrix; $\delta(\cdot)$ is the Dirac delta function. For the wall potentials $V(\mathbf{x}_j)$, we use the Weeks–Chandler–Andersen potential. (1) and (2) are integrated using HOOMD-blue.[53]

There are two time scales: the reorientation time $\tau_R = 1/D_R$ and the time for the actuator to rotate the particles $1/\Omega_j$. There are also three length scales: the plate gap W, the particle's run length $\ell = U_0\tau_R$, and the microscopic diffusion length $\delta = \sqrt{D_T\tau_R}$. Scaling time and length by the reorientation time τ_R and run length ℓ, respectively, yields three nondimensional quantities: the ratio of the plate gap to the run length $w = W/\ell$, the ratio of the diffusivities $(\ell/\delta)^2 = U_0^2\tau_R/D_T$, and the ratio of the actuator rotation time to the thermal

reorientation time $\omega_j = \tau_R \Omega_j$, *i.e.*, the nondimensionalized angular velocity.

Our goal is to control the distribution trajectory of ABPs to dynamically track a user-specified trajectory. Suppose that we consider the example shown in Fig. 1, where we have an initial population of ABPs accumulated at the center, and we would like to design a controller that selects input fields to split the population into two equal groups. Instead of optimizing control inputs to each individual particle, our goal is to apply feedback control on a population-level distribution function.

To these ends, we apply MPC to the Smoluchowski equation and model the probability of the particles' positions and orientation. Following Shaik *et al.*[54] and Saintillan and Shelley,[55] we model the probability distribution $P(\mathbf{x},\theta,t)$ of finding a particle at position $\mathbf{x}$ with orientation θ at time t using the Smoluchowski equation with zero flux boundary conditions at the walls and periodic boundary conditions in y.

$$\frac{\partial P(\mathbf{x},\theta,t)}{\partial t} + \nabla \cdot \mathbf{j} + \frac{\partial}{\partial \theta} j_\theta = 0 \tag{3}$$

$$\mathbf{j} = U_0 \mathbf{q} P(\mathbf{x},\theta,t) - D_T \nabla P(\mathbf{x},\theta,t) \tag{4}$$

$$j_\theta = \Omega(\mathbf{x},\theta,t) P(\mathbf{x},\theta,t) - D_R \frac{\partial}{\partial \theta} P(\mathbf{x},\theta,t) \tag{5}$$

$$\mathbf{n}\cdot\mathbf{j}|_{\text{walls}} = 0 \tag{6}$$

$$P(\mathbf{x},\theta,0) = P_0(\mathbf{x},\theta) \tag{7}$$

$\mathbf{n}$ is the unit vector normal to the wall; $P_0(\mathbf{x},\theta)$ is the initial distribution of the particles. We define the number density $n(\mathbf{x},t) = \int P(\mathbf{x},\theta,t)\mathrm{d}\theta$ and the polar order $\mathbf{m}(\mathbf{x},t) = [\,m_x \quad m_y\,]^T = \int \mathbf{q} P(\mathbf{x},\theta,t)\mathrm{d}\theta$.

2.2 Model predictive control

As mentioned before, the OCP considers finding optimal inputs to minimize a user-defined objective function over a finite time horizon. Applied to the Smoluchowski equation, the OCP to find the optimal control inputs starting with initial distribution $P_0(\mathbf{x},\theta)$ is given by

$$\begin{aligned} \min_{\Omega(\cdot)} \quad & \int_0^T L(P(\mathbf{x},\theta,t),\Omega(\mathbf{x},\theta,t))\mathrm{d}t + L_f(P(\mathbf{x},\theta,T)) \\ \text{s.t.} \quad & (3) \text{ to } (7) \\ & \Omega(\mathbf{x},\theta,t) \subseteq \mathbb{W} \quad \forall t \in [0,\ T] \end{aligned} \tag{8}$$

where T is the time horizon; $L(\cdot)$ is a stage cost that penalizes both the deviation of the distribution from the desired values and the control effort; $L_f(\cdot)$ is the terminal cost that only penalizes the final distribution; $\mathbb{W}$ is the set of allowable control inputs. These constraints may arise from the physical limitations of the actuator, *e.g.*, a maximum magnetic field strength.

In MPC, (8) is discretized to obtain a nonlinear optimization problem. Thus, in (3) to (7), we discretize the distribution P and Ω in both space and orientation using a finite number of orthogonal basis functions, yielding a set of ordinary differential equations (ODEs) (see ESI†). The control input $\Omega(\mathbf{x},\theta,T)$ is discretized by representing it as a piecewise constant function over the prediction horizon $K\Delta$, with control values determined at each sample time Δ as illustrated by the green input in Fig. 1c such that $\mathbf{\Omega} = [\Omega(\mathbf{x},\theta,0) \quad \mathbf{\Omega}(\mathbf{x},\theta,\Delta) \quad \cdots \quad \Omega(\mathbf{x},\theta,(K-1)\Delta)]$. Finally, we discretize the system dynamics by applying a numerical ODE solver, such as a Runge–Kutta method, to approximate the evolution of the state variables over discrete time intervals, determined by the sample time Δ. For simplicity, we continue to refer to the discretized system as (3) to (7).

With the discretized system, we can now formulate the MPC problem as illustrated in Fig. 1c. Our goal is to obtain an optimal control sequence $\mathbf{\Omega}^o$ that contains a sequence of input fields that steers the ABPs to track the user-specified distribution trajectory. Given some initial distribution $P_0(\mathbf{x},\theta)$, we solve the following nonlinear optimization problem to find the optimal control sequence $\mathbf{\Omega}^o$:

$$\begin{aligned} \min_{\mathbf{\Omega}} \quad & \sum_{k=0}^{K-1} L(P(\mathbf{x},\theta,k\Delta),\mathbf{\Omega}(k)) + L_f(P(\mathbf{x},\theta,K\Delta)) \\ \text{s.t.} \quad & (3) \text{ to } (7) \\ & \mathbf{\Omega}(k) \subseteq \mathbb{W} \quad \forall k \in [0,\ K-1] \end{aligned} \tag{9}$$

$\mathbf{\Omega}(k)$ is the $(k+1)$ th element of $\mathbf{\Omega}$. This nonlinear optimization problem is solved using IPOPT,[56] an interior point optimizer, at each sample time Δ to obtain the optimal control sequence $\mathbf{\Omega}^o$.

Recall we are controlling a BD simulation using MPC with the Smoluchowski equation as the model and require estimates of the distribution at every sample time for the initial condition $P_0(\mathbf{x},\theta)$ in (9). In the BD simulation, instead of having direct measurements of the distribution $P(\mathbf{x},\theta,t)$, we have access to the positions and orientations of the particles $\mathbf{X}(t)$, *i.e.*, samples of the distribution. Thus, we estimate the initial distribution $P_0(\mathbf{x},\theta)$ from the BD simulations by applying Kernel density estimation (KDE) to the particle positions and orientations, and project to the basis functions (see ESI†). The feedback controller can be summarized into the following steps:

0. Given an initial particle positions and orientations in the BD simulations, $\mathbf{X}(0)$, fix the horizon length K and initialize $k = 0$.

1. Perform KDE and projection on $\mathbf{X}(k\Delta)$ to estimate the distribution $P(\mathbf{x},\theta,k\Delta)$.

2. Initialize the state $P_0(\mathbf{x},\theta) = P(\mathbf{x},\theta,k\Delta)$ and solve (9) for $\mathbf{\Omega}^o$.

3. Apply first value of control sequence $\mathbf{\Omega}^o(0)$

4. Evaluate $\mathbf{X}((k+1)\Delta)$, set $k := k+1$ and return to step 1.

3 Examples

Before we present the examples, let us consider some limiting cases of the system. In the limit of no activity, because the particles are passive, spherical, and noninteracting, they cannot be directed. The actuator only aligns the particles with the

field so without activity, the number density would adopt the typical equilibrium distribution, *i.e.*, a uniform distribution across the domain. In the limit of no actuation, *i.e.* $\Omega = 0$, but some activity, the ABPs would accumulate at the walls due to the self-propulsion and create a boundary layer that scales with the run length ℓ.[57] Therefore, to achieve distributions other than uniform or concentration at the walls, we need the combined effects of both activity and actuation. The actuator biases the particles' orientations which then the activity drives the population to the desired state, *e.g.*, accumulation in the center.

To illustrate the capabilities of our framework, we present two examples of ABP control using MPC on BD simulations. First, we demonstrate the control of the ABP density by splitting the population of ABPs into two groups and juggling the particles back-and forth as shown in Movie S1 (ESI†). Next, we simultaneously maintain two groups of ABPs and specify the density-normalized y-polar order m_y/n to be sinusoidal in x as shown in Movie S2 (ESI†). We provide an additional example in the ESI† demonstrating the advantages of MPC over a heuristic solution. This example involves controlling the density of ABPs to dynamically track a sinusoidal x-position set point. We compare the performance of a heuristically designed controller (shown in Movie S3, ESI†) with that of the MPC solution (shown in Movie S4, ESI†).

For the examples, we assume the input field and distributions are constant across y and thus, focus on the one dimensional problem in x. We set the ratio of the diffusivities $(\ell/\delta)^2 = 10$ and rigid plates placed at $x = \pm W/2 = \pm 5\ell$, *i.e.*, plate gap of $W = 10\ell$. We set the number of ABPs $N = 10^6$. KDE was done using KDEpy[58] with a bin width of 10^{-3}. The distribution is discretized in orientation with four Fourier modes, and in position, with 40 Legendre collocation points. The controls (both Ω_x and Ω_y) are discretized in position with 20 Legendre collocation points. MPC was done using MPCTools[59] which relies on CasADI.[60] The sample time $\Delta = 0.5\tau_R$. The horizon was set to $K = 14$, *i.e.*, $7\tau_R$. Code is available at **https://github.com/titusswsquah/mpc_abp_2024**.[61]

3.1 Density control

In this example, our objective is to control the density of the ABPs in a user-specified trajectory. In particular, we shall split a population of ABPs into two groups along the x-direction and then juggle the density of particles between the two groups.

For this example, we focus only on $\Omega_x(x,t)$ and fix $\Omega_y(x,t) = 0\tau_R^{-1}$. We initialize the system with a zero field, *i.e*, the system starts at steady state with $\Omega_x(x) = 0\tau_R^{-1}$. Our user-specified set points are as follows:

1. Accumulate particles into center.

2. Split particles into two equal groups centered at $x_+ = 2.5\ell$ and $x_- = -2.5\ell$.

3. Achieve a distribution where the probability of finding a particle with position $x < 0\ell$ is 25% while maintaining the populations centered at $x_\pm$.

4. Repeat step 3, but with the probability of finding a particle with position $x < 0\ell$ is 75%.

5. Return to step 2.

To simulate having limited actuation capabilities, we constrain the input x-field to have a maximum magnitude of $3\tau_R^{-1}$, *i.e.*, $|\Omega_x(\mathbf{x},\theta,t)|_\infty \leq 3\tau_R^{-1}$. $|\cdot|_\infty$ is the maximum norm over the domain.

Let us introduce some notation for the stage cost. $|\cdot|$ denotes the 2-norm over the domain. $\langle\cdot\rangle$ is the expectation of the argument over the entire domain, given distribution P, *i.e.*, $\langle\cdot\rangle = \int\int(\cdot)P(\boldsymbol{x},\theta,t)\mathrm{d}\boldsymbol{x}\mathrm{d}\theta$. Since we only have one distribution, the distribution is implied to be of the particle's positions and orientations. We are sometimes interested in parts of the domain, mainly in position x. Thus, we define $\langle\cdot\rangle_{\mathscr{D}}$ as the expectation of the argument over the domain $\mathscr{D}$. For example, the expected value of the argument for the left half of the region $\langle\cdot\rangle_{x<0} = \int\int\int_{-W/2}^{0}(\cdot)P(\mathbf{x},\theta,t)\mathrm{d}x\mathrm{d}y\mathrm{d}\theta$. Lastly, we denote the probability of finding a particle in the left half, *i.e.*, $\langle 1\rangle_{x<0}$, as $n(x<0)$. Similarly, we denote the probability of finding a particle in the right half as $n(x>0)$.

With these definition, we are ready to define the stage cost $L_1(\cdot)$. We omit position and orientation arguments for clarity.

$$L_1 = L_{P,1} + L_{\Omega,1}$$

$$L_{P,1}(P(t),\mathbf{x}_{\mathrm{sp},1}(t)) := c_1\left(\left\langle(x-x_-(t))^2\right\rangle_{x<0} + \left\langle(x-x_+(t))^2\right\rangle_{x>0}\right) + c_2\left(n(x<0) - r_{\mathrm{sp}}(t)n(x>0)\right)^2$$

$$L_{\Omega,1}(\Omega_x(t),\Omega_x(t-\Delta)) := c_3\left|\frac{\partial^2\Omega_x}{\partial x^2}\right|^2 + c_4|\Omega_x(t) - \Omega_x(t-\Delta)|^2$$

$$\mathbf{x}_{\mathrm{sp},1}(t) := [x_-(t) \quad x_+(t) \quad r_{\mathrm{sp}}(t)]^{\mathrm{T}}$$

The stage cost $L_1(\cdot)$ is the sum of the distribution cost $L_{P,1}$ and the control cost $L_{\Omega,1}$. The distribution cost $L_{P,1}$ consists of three contributions. $\langle(x - x_-(t))^2\rangle_{x<0}$ penalizes the expected square distance of particles to the left of the origin $(x < 0\ell)$ to the left set point $x_+(t)$. Likewise, $\langle(x - x_+(t))^2\rangle_{x>0}$ penalizes the expected square distance of particles to the right of the origin $(x > 0\ell)$ to the right set point $x_+(t)$. The last term in the distribution cost $(n(x<0) - r_{\mathrm{sp}}(t)n(x>0))^2$ penalizes deviations from the left-to-right ratio set point $r_{\mathrm{sp}}(t)$ between probability of finding a particle to the left $n(x<0)$ and right $n(x>0)$ of the origin. This ratio penalty is used to achieve set points like equal-sized groups or 25% of particles to the left.

In a real experiment, rapid control adjustments may strain or damage the actuator. Thus, we include a control cost $L_{\Omega,1}$, aimed at reducing large changes over space and time. The control cost consists of two terms: the first term $\left|\frac{\partial^2\Omega_x}{\partial x^2}\right|^2$ penalizes the curvature of the input field in x, and the second term $|\Omega_x(t) - \Omega_x(t-\Delta)|^2$ penalizes the change in the input field in time. c_1, c_2, c_3, and c_4 are weights that determine the importance of each term. We prioritize the distribution cost

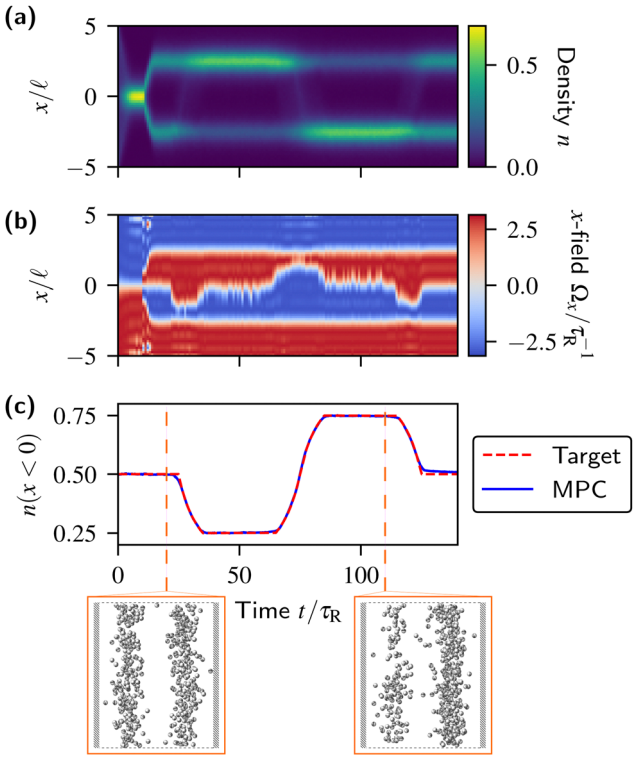

Fig. 2 MPC is used to split the distribution of ABPs into two groups and juggle particles back-and-forth in a user-specified manner. (a) Number density and (b) MPC input (external field) to split and juggle the particles. Positive (red) x-field Ω_x aligns particles in the positive x-direction, and negative (blue) x-field aligns particles in the negative x-direction. The maximum x-field magnitude is constrained to be less than $3\tau_R^{-1}$. (c) The fraction of particles in the left half of the simulation box $n(x < 0)$ as the particles are juggled back-and-forth, for the target (dashed red) and MPC results (solid blue). Insets are snapshots of a subset of particles in a Brownian dynamic simulation at times $t = 20\tau_R$ and $t = 100\tau_R$. The $t = 20\tau_R$ inset shows the particles split into two even groups and the $t = 100\tau_R$ inset shows the particles in two groups with more particles in the x_+ group.

so $c_1 = 6.4 \times 10^3$, $c_2 = 6.4 \times 10^4$, $c_3 = 5 \times 10^{-5}$, and $c_4 = 10^{-3}$. The terminal cost L_f is chosen to be the same as $L_{P,1}$.

A movie of the process is shown in Movie S1 (ESI†). As seen in Fig. 2a, we start with an initial condition of $\Omega_x(x,0) = 0\tau_R^{-1}$. First, the particles accumulate at $x = 0\ell$. At $t = 15\tau_R$, we change the position set points to split into two groups and see the group at $x = 0\ell$ split into two even groups centered at the two set points $x_- = -2.5\ell$ and $x_+ = 2.5\ell$. At around $t = 25\tau_R$, we see particles transferred from the x_- group to the x_+ group until $n(x < 0) = 0.25$. In Fig. 2a, this corresponds the brighter band at $x = 2.5\ell$ and a darker band at $x = -2.5\ell$. At $t = 65\tau_R$, the opposite happens and particles are transferred from the x_+ to the x_- group until $n(x < 0) = 0.75$. Finally, particles are transferred until we return to both groups having roughly equal amounts of particles. The ratios are shown to dynamically track the chosen set point in Fig. 2c.

Fig. 2b shows the feedback control input field. We choose the input field to be a Legendre polynomial in x, similar to how we discretize the distribution. Recall these are only the first values of the optimal control sequence $\mathbf{\Omega}^o$ obtained by solving the K-stage optimization problem stated in (9). Note that areas where the input field change from positive to negative are regions where the particles accumulate as the particles are oriented towards those regions. For example, at $t = 0\tau_R$, the controller gathers the particles at the $x = 0\ell$ by changing the input from positive (red) to negative (blue) at $x = 0\ell$. Conversely, changing the input from negative to positive would orient the particles away from that location. At $t = 15\tau_R$, the controller splits the population by orienting the particles away from the center and towards the two set points. One can imagine the input field creating an effective double well potential with two attractive wells at $x_\pm$ and the hill located at $x = 0\ell$. The wells are where the input field changes from positive to negative, and the hill is where the input field changes from negative to positive.

To transfer particles from one group to the other while maintaining the populations centered at $x_\pm$, the controller keeps the well locations at $x_\pm$ and only moves the hill location. Thus, at $t = 25\tau_R$, when particles need to be transferred from x_- to x_+, the hill is moved to $x < 0\ell$ which expands the region orienting particles to x_+. Once the ratio is achieved, the controller moves the hill to a slightly negative x as to maintain the unequal ratio. At $t = 65\tau_R$, the particle transfer is reversed, and the set transfer rate is faster, as shown by the slope of the set point in Fig. 2c. Thus, the hill is moved to give a larger region orienting particles to x_-, then returns to a slightly positive x.

3.2 Density and polar order control

So far we have demonstrated control on number density only. In this example, we control both the number density and the polar order in the y-direction, m_y. Note the polar order, normalized by the number density $\mathbf{m}/n$, is the dimensionless average velocity of the particles, with both time and length scaled by the reorientation time τ_R and the run length ℓ, respectively. We refer to the y-component of the nondimensional average velocity as the y^*-velocity. We wish to have two groups of particles centered at $x_\pm$ and the y^*-velocity to fit a sine wave as shown with the dotted red line in Fig. 2c. For this example, we require both the x and y components of the input field to manipulate the number density and the y-polar order m_y. Additionally, we constrain the input field to have a maximum magnitude of $3\tau_R^{-1}$, *i.e.*, $|\Omega|_\infty = \left|\sqrt{\Omega_x^2 + \Omega_y^2}\right|_\infty \leq 3\tau_R^{-1}$.

To control the y^*-velocity to fit to a sine wave and split the particles into two groups, we use the same stage cost for splitting particles, with the addition of a distribution cost for deviation of the y^*-velocity from the set point and a control cost for the y-component of the input field.

$$L_2 = L_{P,2} + L_{\Omega,2}$$

$$L_{P,2}\left(P(t), \mathbf{x}_{\mathrm{sp},2}(t)\right) := L_{P,1}\left(P(t), \begin{bmatrix} x_-(t) & x_+(t) & 1 \end{bmatrix}^{\mathrm{T}}\right) + c_5\left|\frac{m_y(x)}{n(x)} - v_{\mathrm{sp}}(t)\sin(\pi x/W)\right|^2$$

$$L_{\Omega,2}\left(\Omega_x(t),\Omega_x(t-\Delta),\Omega_y(t-\Delta)\right) := c_6\left(\left|\frac{\partial^2\Omega_x}{\partial x^2}\right|^2+\left|\frac{\partial^2\Omega_y}{\partial x^2}\right|^2\right) + c_7\left(\left|\Omega_x(t)-\Omega_x(t-\Delta)\right|^2 + \left|\Omega_y(t)-\Omega_y(t-\Delta)\right|^2\right)$$

$$\mathbf{x}_{\mathrm{sp},2}(t) := \begin{bmatrix} x_-(t) & x_+(t) & v_{\mathrm{sp}}(t)\end{bmatrix}^{\mathrm{T}}$$

The left-to-right ratio r_{sp} is set to unity to have equal sized groups. We again prioritize the distribution cost so we set $c_1 = 6.4\times10^3$, $c_2 = 6.4\times10^4$, $c_5 = 6.4\times10^6$, $c_6 = c_7 = 10^{-4}$.

Fig. 3 summarizes the results of the regulation problem where we fix the set point $\mathbf{x}_{\mathrm{sp},2}(t) = [-2.5\ell \quad 2.5\ell \quad 0.4]^{\mathrm{T}}$. A movie of the process is shown in Movie S2 (ESI†). Fig. 3a shows the trajectory of two particles over the simulation with arrows pointing the direction θ_i and colors to denote $\sin(\theta_i)$. The color is orange for $\sin(\theta_i) = 1$, blue when $\sin(\theta_i) = -1$ and white for $\sin(\theta_i) = 0$. The walls are in the horizontal axis. As illustrated by the two particle trajectories in Fig. 3a, the particles in the x_- have a negative y^*-velocity and the particles in the x_+ group have a positive y^*-velocity.

Fig. 3(b)–(e) show the averaged quantities over the simulation with shaded regions marking the third standard deviation. Fig. 3b contains the number density and shows the particles are split into two groups with a profile that is similar to the particle splitting example. The variation of the number density as shown by the standard deviation is minimal. Fig. 3c shows the expected y^*-velocity whose set point is a sine wave. The expected y^*-velocity meets the set point sine wave as shown by the blue and red lines. The variation of the velocity is large where the number density is low since less number density corresponds to fewer samples which results in a larger variation in the expected velocity.

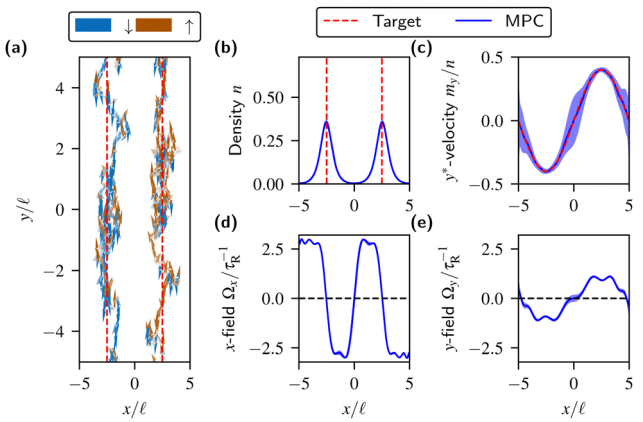

Fig. 3 MPC is used to control both the ABP density and the y^*-velocity, *i.e.*, the y component of the dimensionless average velocity, with time and length scaled by reorientation time τ_R and run length ℓ, respectively. We specify a sinusoidal y^*-velocity while simultaneously splitting the particles into two equal groups. (a) Trajectory of two selected particles (blue markers moving down, orange markers moving up), (b) number density for the target (dashed red) and averaged MPC results (solid blue) throughout the simulation, (c) y^*-velocity for the target (dashed red) and the averaged MPC results (solid blue), (d) x and (e) y components of the field needed to achieve both population splitting and y^*-velocity control. Positive x-field Ω_x aligns particles in the positive x-direction, and negative x-field aligns particles in the negative x-direction. y-Field Ω_y governs alignment in the y-direction in a similar manner. The maximum magnitude of the field $\left|\sqrt{\Omega_x^2+\Omega_y^2}\right|_\infty$ is constrained to be less than $3\tau_R^{-1}$. Blue shades in (b)–(e) mark the third standard deviation throughout the simulation.

Fig. 3d shows the x component of the input field Ω_x. The input field is similar to the particle splitting example, but is different near $x_\pm$. Because the maximum speeds are also located at $x_\pm$, the controller needs to use Ω_y to match the set point. Consequently, because of the maximum magnitude constraint on the input, the controller uses a smaller magnitude of Ω_x near $x_\pm$.

Fig. 3e shows the y component of the input field $\Omega_y(x,t)$ that orients particles along the y axis. $\Omega_y(x,t)$ has an interesting shape. One would guess to obtain a sine velocity as depicted in Fig. 3c, a sine input should be applied to Ω_y. While the controller's $\Omega_y(x,t)$ input somewhat looks like a sine curve, there are more wiggles than expected. This is due to the effective diffusion of particles. This is most clear at the walls where one might expect zero input. Instead, the input field is positive at $x = -5\ell$ and negative at $x = 5\ell$. This is to counteract the particles on the interior that are oriented in the desired direction, but drift into the boundary regions where the desired y^*-velocity is zero. Similarly, a larger field near the walls is required to turn particles that drift from the zero y^*-velocity region. Effective diffusion can also reduce the control needed, *e.g.*, near $x = x_\pm$. Since the particles nearby are traveling with the desired y^*-velocity, effective diffusion reduces the magnitude of control required at $x = x_\pm$.

4 Discussion

In this work, we show that MPC can be used with the Smoluchowski model for feedback control of ABP particle simulations with constraints on the input field. By modeling the probability distribution of the particles and estimating the distribution with KDE, we can control the number density and orientational moments of the particles to accomplish tasks such as population splitting and fitting the velocity to a sine wave.

This work demonstrates that with MPC, precise feedback control over particle distributions and dynamics is achievable. However, to deploy MPC on real-world systems like light-activated particles or nematics[16,20–22,62] in complex environments,[3,63–66] further advancements are necessary. One key area for future research is deploying MPC with models that account for realistic effects like multibody and hydrodynamic interactions between particles and obstacles or other actuation mechanisms, *e.g.*, speed control.[1,2,67,68] In this work, the MPC model (the Smoluchowski equation) accurately describes the dynamics of the true system (the distribution dynamics of non-interacting ABPs). However, in future work, we will consider interacting active agents where exact models do not exist.

Thankfully, an exact MPC model is not necessary for acceptable control in practice. In fact, process industries often use linear models with MPC to maintain nonlinear processes at set points.[49] This does not mean accurate models are useless; depending on the system's complexity or the task at hand, a more accurate forecast may be necessary. In situations demanding higher model accuracy, we can utilize machine learning techniques to enhance precision.

Additionally, our current model assumes access to higher-order orientation moments, *i.e.*, the full state, which is difficult to measure in practice. Therefore, it is crucial to develop estimators[69] or models that require only density measurements like active model B.[70] Lastly, while MPC is capable of controlling systems in real-time, our current implementation is still too slow for experimental usage since for each sample time Δ, we require five minutes of compute time. However, this can be greatly accelerated using C++ code generation and approximations to the MPC solution, all available in the acados package.[71,72]

Author contributions

Conceptualization: T. Q., J. B. R., S. C. T.; data curation: T. Q.; formal analysis: T. Q.; funding acquisition: J. B. R., S. C. T.; investigation: T. Q.; methodology: T. Q, K. J. M.; project administration: J. B. R., S. C. T.; software: T. Q.; resources: J. B. R., S. C. T.; supervision: J. B. R., S. C. T.; visualization: T. Q.; writing – original draft: T. Q.; writing – review & editing: T. Q, K. J. M., J. B. R., S. C. T.

Data availability

Code is available at **https://github.com/titusswsquah/mpc_abp_2024**.[61]

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

We would like to thank Steven J. Kuntz and Sachit G. Nagella for helpful discussions. T. Q. is supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. 2139319. S. C. T. is supported by the National Science Foundation under Grant No. 2150686 and the Packard Fellowship in Science and Engineering. Use was made of computational facilities purchased with funds from the National Science Foundation (CNS-1725797) and administered by the Center for Scientific Computing (CSC). The CSC is supported by the California NanoSystems Institute and the Materials Research Science and Engineering Center (MRSEC; NSF DMR 2308708) at UC Santa Barbara.

Notes and references

1 M. C. Marchetti, J. F. Joanny, S. Ramaswamy, T. B. Liverpool, J. Prost, M. Rao and R. A. Simha, *Rev. Mod. Phys.*, 2013, **85**, 1143–1189.
2 H. Chaté, *Annu. Rev. Condens. Matter Phys.*, 2020, **11**, 189–212.
3 C. Bechinger, R. Di Leonardo, H. Löwen, C. Reichhardt, G. Volpe and G. Volpe, *Rev. Mod. Phys.*, 2016, **88**, 045006.
4 R. A. Simha and S. Ramaswamy, *Phys. Rev. Lett.*, 2002, **89**, 058101.
5 A. Sciortino and A. R. Bausch, *Proc. Natl. Acad. Sci. U. S. A.*, 2021, **118**, e2017047118.
6 T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen and O. Shochet, *Phys. Rev. Lett.*, 1995, **75**, 1226.
7 S. Thampi and J. Yeomans, *Eur. Phys. J.-Spec. Top.*, 2016, **225**, 651–662.
8 J. Tailleur and M. E. Cates, *Phys. Rev. Lett.*, 2008, **100**, 218103.
9 D. Grober, I. Palaia, M. C. Uçar, E. Hannezo, A. Šarić and J. Palacci, *Nat. Phys.*, 2023, 1–9.
10 S. Ramananarivo, E. Ducrot and J. Palacci, *Nat. Commun.*, 2019, **10**, 1–8.
11 R. Zhang, S. A. Redford, P. V. Ruijgrok, N. Kumar, A. Mozaffari, S. Zemsky, A. R. Dinner, V. Vitelli, Z. Bryant, M. L. Gardel and J. J. de Pablo, *Nat. Mater.*, 2021, **20**, 875–882.
12 A. M. Tayar, F. Caballero, T. Anderberg, O. A. Saleh, M. Cristina Marchetti and Z. Dogic, *Nat. Mater.*, 2023, **22**, 1401–1408.
13 H. Xie, M. Sun, X. Fan, Z. Lin, W. Chen, L. Wang, L. Dong and Q. He, *Sci. Robot.*, 2019, **4**, eaav8006.
14 H. Joh and D. E. Fan, *Adv. Mater.*, 2021, **33**, 2101965.
15 M. Akter, J. Keya, K. Kayano, A. Kabir, D. Inoue, H. Hess, K. Sada, A. Kuzuya, H. Asanuma and A. Kakugo, *Sci. Robot.*, 2022, **7**, eabm0677.
16 P. Guillamat, J. Ignés-Mullol and F. Sagués, *Proc. Natl. Acad. Sci. U. S. A.*, 2016, **113**, 5498–5502.
17 J. Palacci, S. Sacanna, A. P. Steinberg, D. J. Pine and P. M. Chaikin, *Science*, 2013, **339**, 936–940.
18 M. A. Fernandez-Rodriguez, F. Grillo, L. Alvarez, M. Rathlef, I. Buttinoni, G. Volpe and L. Isa, *Nat. Commun.*, 2020, **11**, 1–10.
19 S. Palagi, D. P. Singh and P. Fischer, *Adv. Opt. Mater.*, 2019, **7**, 1900370.
20 I. Buttinoni, G. Volpe, F. Kümmel, G. Volpe and C. Bechinger, *J. Phys.: Condens. Matter*, 2012, **24**, 284129.
21 G. Frangipane, D. Dell'Arciprete, S. Petracchini, C. Maggi, F. Saglimbeni, S. Bianchi, G. Vizsnyiczai, M. L. Bernardini and R. Di Leonardo, *eLife*, 2018, **7**, e36608.
22 L. M. Lemma, M. Varghese, T. D. Ross, M. Thomson, A. Baskaran and Z. Dogic, *PNAS Nexus*, 2023, **2**, pgad130.
23 J. Arlt, V. A. Martinez, A. Dawson, T. Pilizota and W. C. K. Poon, *Nat. Commun.*, 2018, **9**, 768.
24 J. Arlt, V. A. Martinez, A. Dawson, T. Pilizota and W. C. K. Poon, *Nat. Commun.*, 2019, **10**, 2321.

25 T. Mano, J. B. Delfau, J. Iwasawa and M. Sano, *Proc. Natl. Acad. Sci. U. S. A.*, 2017, **114**, E2580–E2589.
26 A. F. Demirörs, M. T. Akan, E. Poloni and A. R. Studart, *Soft Matter*, 2018, **14**, 4741–4749.
27 S. C. Takatori, R. De Dier, J. Vermant and J. F. Brady, *Nat. Commun.*, 2016, **7**, 1–7.
28 M. Fränzl and F. Cichos, *Sci. Rep.*, 2020, **10**, 12571.
29 N. Pellicciotta, M. Paoluzzi, D. Buonomo, G. Frangipane, L. Angelani and R. Di Leonardo, *Nat. Commun.*, 2023, **14**, 4191.
30 H. Massana-Cid, C. Maggi, G. Frangipane and R. Di Leonardo, *Nat. Commun.*, 2022, **13**, 2740.
31 M. Baldovin, D. Guéry-Odelin and E. Trizac, *Phys. Rev. Lett.*, 2023, **131**, 118302.
32 N. Koumakis, A. T. Brown, J. Arlt, S. E. Griffiths, V. A. Martinez and W. C. K. Poon, *Soft Matter*, 2019, **15**, 7026–7032.
33 S. Shankar, L. V. D. Scharrer, M. J. Bowick and M. C. Marchetti, *Proc. Natl. Acad. Sci. U. S. A.*, 2024, **121**, e2400933121.
34 K. Nishiyama, J. Berezney, M. M. Norton, A. Aggarwal, S. Ghosh, M. F. Hagan, Z. Dogic and S. Fraden, *Closed-loop control of active nematic flows*, 2024, preprint, arXiv:2408.14414 [cond-mat, physics:physics], DOI: **10.1073/pnas.2400933121**.
35 Y. Yang, M. A. Bevan and B. Li, *Adv. Theory Simul.*, 2020, **3**, 2000034.
36 M. Durve, F. Peruani and A. Celani, *Phys. Rev. E*, 2020, **102**, 012601.
37 S. Colabrese, K. Gustavsson, A. Celani and L. Biferale, *Phys. Rev. Lett.*, 2017, **118**, 158004.
38 M. J. Falk, V. Alizadehyazdi, H. Jaeger and A. Murugan, *Phys. Rev. Res.*, 2021, **3**, 033291.
39 S. Chennakesavalu and G. M. Rotskoff, *J. Chem. Phys.*, 2021, **155**, 194114.
40 S. Muiños-Landin, A. Fischer, V. Holubec and F. Cichos, *Sci. Robot.*, 2021, **6**, eabd9285.
41 D. P. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific, Belmont, MA, 1995, vol. 1.
42 M. M. Norton, P. Grover, M. F. Hagan and S. Fraden, *Phys. Rev. Lett.*, 2020, **125**, 178005.
43 C. Sinigaglia, F. Braghin and M. Serra, *Phys. Rev. Lett.*, 2024, **132**, 218302.
44 L. K. Davis, K. Proesmans and E. Fodor, *Phys. Rev. X*, 2024, **14**, 011012.
45 S. Shankar, V. Raju and L. Mahadevan, *Proc. Natl. Acad. Sci. U. S. A.*, 2022, **119**, e2121985119.
46 S. Ghosh, A. Baskaran and M. F. Hagan, *Achieving designed texture and flows in bulk active nematics using optimal control theory*, 2024, preprint, arXiv:2408.14596 [cond-mat.soft], DOI: **10.1073/pnas.2121985119**.
47 A. Shenoy, C. V. Rao and C. M. Schroeder, *Proc. Natl. Acad. Sci. U. S. A.*, 2016, **113**, 3976–3981.
48 J. B. Rawlings, D. Q. Mayne and M. M. Diehl, *Model Predictive Control: Theory, Design, and Computation*, Nob Hill Publishing, Santa Barbara, CA, 2nd edn, 2020.
49 S. J. Qin and T. A. Badgwell, *Control Eng. Pract.*, 2003, **11**, 733–764.
50 S. Yu, M. Hirche, Y. Huang, H. Chen and F. Allgöwer, *Auton. Intell. Syst.*, 2021, **1**, 4.
51 R. Dreyfus, J. Baudry, M. L. Roper, M. Fermigier, H. A. Stone and J. Bibette, *Nature*, 2005, **437**, 862–865.
52 S. Klumpp, C. T. Lefèvre, M. Bennet and D. Faivre, *Phys. Rep.*, 2019, **789**, 1–54.
53 J. A. Anderson, J. Glaser and S. C. Glotzer, *Comput. Mater. Sci.*, 2020, **173**, 109363.
54 V. A. Shaik, Z. Peng, J. F. Brady and G. J. Elfring, *Soft Matt.*, 2023, **19**, 1384–1392.
55 D. Saintillan and M. J. Shelley, *C. R. Phys.*, 2013, **14**, 497–517.
56 A. Wächter and L. T. Biegler, *Math. Prog.*, 2006, **106**, 25–57.
57 B. Ezhilan, R. Alonso-Matilla and D. Saintillan, *J. Fluid Mech.*, 2015, **781**, R4.
58 T. Odland, *tommyod/KDEpy: Kernel Density Estimation in Python*, 2018.
59 M. J. Risbeck and J. B. Rawlings, *MPCTools: Nonlinear model predictive control tools for CasADi (Python inferface)*, 2015, **https://bitbucket.org/rawlings-group/mpc-tools-casadi**.
60 J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings and M. Diehl, *Math. Prog. Comp.*, 2019, **11**, 1–36.
61 T. Quah, K. J. Modica, J. B. Rawlings and S. C. Takatori, *Code for "Model Predictive Control of Non-interacting Active Brownian Particles"*, 2024, DOI: **10.5281/zenodo.13017195**.
62 Z. Zarei, J. Berezney, A. Hensley, L. Lemma, N. Senbil, Z. Dogic and S. Fraden, *Soft Matter*, 2023, **19**, 6691–6699.
63 Z. You, A. Baskaran and M. C. Marchetti, *Proc. Natl. Acad. Sci. U. S. A.*, 2020, **117**, 19767–19772.
64 F. Brauns and M. C. Marchetti, *Phys. Rev. X*, 2024, **14**, 021014.
65 K. J. Modica, Y. Xi and S. C. Takatori, *Front. Phys.*, 2022, **10**, 869175.
66 J. M. Barakat, K. J. Modica, L. Lu, S. Anujarerat, K. H. Choi and S. C. Takatori, *ACS Appl. Nano Mater.*, 2024, **7**, 12142–12152.
67 M. t Vrugt, J. Bickmann and R. Wittkowski, *J. Phys.: Condens. Matter*, 2023, **35**, 313001.
68 K. J. Modica, A. K. Omar and S. C. Takatori, *Soft Matter*, 2023, **19**, 1890–1899.
69 P. N. Tran, S. Ray, L. Lemma, Y. Li, R. Sweeney, A. Baskaran, Z. Dogic, P. Hong and M. F. Hagan, *Deep-learning Optical Flow Outperforms PIV in Obtaining Velocity Fields from Active Nematics*, 2024, preprint, arXiv:2404.15497v2 [cond-mat.soft], DOI: **10.1039/D2SM01421A**.
70 R. Wittkowski, A. Tiribocchi, J. Stenhammar, R. J. Allen, D. Marenduzzo and M. E. Cates, *Nat. Commun.*, 2014, **5**, 4351.
71 M. Diehl, H. G. Bock and J. P. Schloder, *SIAM J. Control Opt.*, 2005, **43**, 1714–1736.
72 R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen and M. Diehl, *Math. Prog. Comp.*, 2022, **14**, 147–183.