

Supplementary Information for Model Predictive Control of Non-interacting Active Brownian Particles

Titus Quah, Kevin J. Modica, James B. Rawlings, and Sho C. Takatori
Department of Chemical Engineering
University of California, Santa Barbara
Santa Barbara, CA 93106

1 Spectral method for orientational discretization

MPC requires a finite-dimensional discrete time system. Here, we explain how we discretize the Smoluchowski equation to obtain a finite-dimensional system. We start with the Smoluchowski equation describing ABPs actuated by an input field $\Omega(\mathbf{x}, \theta, t)$ confined between two plates Eqs. (1) to (4).

$$\frac{\partial P}{\partial t} + \nabla \cdot \mathbf{j} + \frac{\partial}{\partial \theta} j_\theta = 0 \quad (1)$$

$$\mathbf{j} = U_0 \mathbf{q} P - D_T \nabla P \quad (2)$$

$$j_\theta = \Omega P - D_R \frac{\partial}{\partial \theta} P \quad (3)$$

$$\mathbf{n} \cdot \mathbf{j}|_{\text{walls}} = 0 \quad (4)$$

The distribution P and input field Ω are periodic in θ . Thus, we decompose P and Ω into Fourier series, i.e.,

$$P(\mathbf{x}, \theta, t) = \frac{1}{2\pi} \sum_k P_k(\mathbf{x}, t) e^{-ik\theta} \quad (5)$$

$$\Omega(\mathbf{x}, \theta, t) = \frac{1}{2\pi} \sum_k \Omega_k(\mathbf{x}, t) e^{-ik\theta} \quad (6)$$

i is the imaginary number. Note that P_0 is the number density, and the k th orientational moment can be constructed from P_k . For example, for the polar order, $\mathbf{m} = [\text{Re}(P_1) \quad \text{Im}(P_1)]^T$. Details on converting Fourier expansions to Cartesian expansions for orientation moments, e.g., polar order \mathbf{m} , nematic order \mathbf{Q} , is described in te Vrugt and Wittkowski¹. Likewise for the input field, Ω_0 is the isotropic induced angular velocity which can be observed in chiral active matter systems^{2;3}. Ω_1 is the induced angular velocity that orients the particles along the field direction and can be thought of as a magnetic field orienting the particles⁴. Ω_2 is the induced angular velocity that aligns the particles with the field axis which has been done by controlling liquid crystal alignment⁵. Substituting the Fourier series into Eqs. (1) to (4) and taking a Fourier transform in θ yields equations describing the dynamics of the Fourier modes $P_k(\mathbf{x}, t)$.

$$\frac{\partial P_k}{\partial t} + \nabla \cdot \mathbf{j}_k - \frac{ik}{2\pi} \left(\sum_{\hat{k}} P_{k-\hat{k}} \Omega_{\hat{k}} \right) + k^2 D_R P_k = 0 \quad (7)$$

$$\mathbf{n} \cdot \mathbf{j}_k|_{\text{walls}} = 0 \quad (8)$$

$$\mathbf{j}_k = \frac{1}{2} U_0 \begin{bmatrix} (P_{k+1} + P_{k-1}) \\ -i(P_{k+1} - P_{k-1}) \end{bmatrix} - D_T \nabla P_k \quad (9)$$

For this paper, we focus on the input field that orients particles, i.e.,

$$\Omega(\mathbf{x}, \theta, t) = \frac{1}{2\pi} (\Omega_y \cos(\theta) + \Omega_x \sin(\theta)) \quad (10)$$

$$\Omega_1 = \Omega_y + i\Omega_x \quad (11)$$

Note that the orienting field vector $\hat{\mathbf{H}}(\mathbf{x}, t) = [\Omega_x \quad \Omega_y]^T$. We substitute Eq. (10) for Ω in Eq. (7) and obtain

$$\frac{\partial P_k}{\partial t} + \nabla \cdot \mathbf{j}_k - \frac{ik}{2\pi} (P_{k-1} \bar{\Omega}_1 + P_{k+1} \Omega_1) + k^2 D_R P_k = 0 \quad (12)$$

$\bar{\Omega}_1$ is the complex conjugate of Ω_1 . We make a closure approximation by truncating the Fourier series at k_{\max} modes.

Since we assume the input and initial condition is uniform in y , the solution is also uniform in y . Thus, we focus on the spatial discretization in x . Due to the zero-flux boundary conditions at the walls, we use a Gauss-Legendre pseudospectral method to discretize the spatial domain. Note that other methods such as finite differences or finite elements can be used, but we choose the pseudospectral method for its accuracy and efficiency. We could also choose different basis functions, e.g., a Fourier basis or Chebyshev polynomial basis. Legendre polynomials are chosen to handle both boundary layers at the walls that arise from ABPs accumulating at walls and the interior boundary layers that arise from the actuator orienting particles. With a pseudospectral discretization, we enforce the governing equations or boundary equations at collocation points \mathbf{x}_c we choose to be roots of the Legendre polynomials and the end points. Later, when we perform density estimation, we evaluate the density at the collocation points, which projects the density to the basis functions. Finally, we step forward in time using a numerical ODE solver, namely a third-order orthogonal collocation scheme.

2 Kernel Density Estimation and evaluation at collocation points

To apply feedback control, we require an estimate of the distribution projected to the basis functions. Given the x -positions x_i and the orientations θ_i of the particles, we estimate the k th-mode of the density distribution at the collocation points \mathbf{x}_c using weighted Kernel Density Estimation (KDE).

$$P_k(\mathbf{x}_c) \approx \frac{1}{N} \sum_{i=1}^N K(\mathbf{x}_c - x_i) \exp(ik\theta_i) \quad (13)$$

$$K(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (14)$$

σ is the bandwidth of the kernel. Note that $P(x, \theta)$ can be constructed from $P_k(x)$ using Eq. (5).

3 Model verification

Recall we made a closure approximation by assuming $P_k = 0, \forall |k| \leq k_{\max}$ which introduces model errors. We can verify our model by comparing with theory for unconfined systems. We focus on the nondimensionalized average steady-state velocity component in the direction of the applied field, $\langle u_{\parallel}^* \rangle_P$. This quantity represents the average velocity of the particles along the orienting direction, scaled by the reorientation time τ_R and run length ℓ . In their analytical work, Takatori and Brady⁶ derived $\langle u_{\parallel}^* \rangle_P$ as a function of the nondimensional field strength $|\Omega_1/\tau_R^{-1}|$ for particles in three dimensions. A similar analysis in two dimensions yields the following relation

$$\langle u_{\parallel}^* \rangle = \frac{I_1(|\Omega_1/\tau_R^{-1}|)}{I_0(|\Omega_1/\tau_R^{-1}|)} \quad (15)$$

We compare Eq. (15) to our model by orienting the particles in the unconfined direction y , i.e., we set Ω_y to a constant value and Ω_x to zero. The nondimensional velocity component along the y direction is $\langle \cos(\theta) \rangle$ and

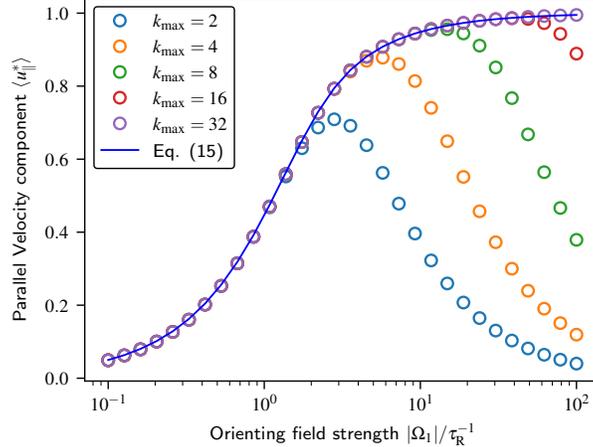


Figure 1: Comparing model and analytical average velocities along orienting direction as a function of orienting field strength

thus, can be obtained by normalizing the imaginary part of the first orientational moment by the number density $\langle u_{\parallel}^* \rangle = m_y/n$. We solve Eq. (12) for the steady state solution for various orienting field strengths and k_{\max} values. Fig. 1 shows the comparison between Eq. (15) and the model $\langle u_{\parallel}^* \rangle$ at $x = 0$. At low field strengths ($|\Omega_1| < 1\tau_R^{-1}$), all the models' parallel velocity components agree with Eq. (15). The $k_{\max} = 2$ model deviates from Eq. (15) at less than $|\Omega_1| = 2\tau_R^{-1}$ and the $k_{\max} = 4$ starts deviating from Eq. (15) at around $|\Omega_1| = 4\tau_R^{-1}$. We set our system to have a input constraint on the maximum field strength

$$|\Omega_1|_{\infty} \leq \Omega_{\max} = 3\tau_R^{-1} \quad (16)$$

so we remain in the region where the $k_{\max} = 4$ model agrees with the steady state analytical solution.

4 Unlimited actuation

To get a sense of the system's behavior, consider the unlimited actuation case, i.e. $\Omega_{\max} = \infty$. For unlimited actuation, the particles align with the field instantaneously and thus, rotational diffusion is negligible. However, translational diffusion is still present. Consider the density control example where we first gather the particles into the center, and let us put a step function in Ω_x directing particles to the center.

This results in an internal boundary layer about the origin, giving the leading order solution

$$n(x) = \begin{cases} ce^{(\ell/\delta)^2 x} & x < 0 \\ ce^{-(\ell/\delta)^2 x} & x > 0 \end{cases}$$

$$c := \frac{1}{2}(\ell/\delta)^2 \left(1 - e^{-\frac{W}{2}(\ell/\delta)^2}\right)^{-1}$$

For the parameters of interest ($W = 10\ell$, $(\ell/\delta)^2 = 10$), the number density $n(x)$ is shown in Fig. 2.

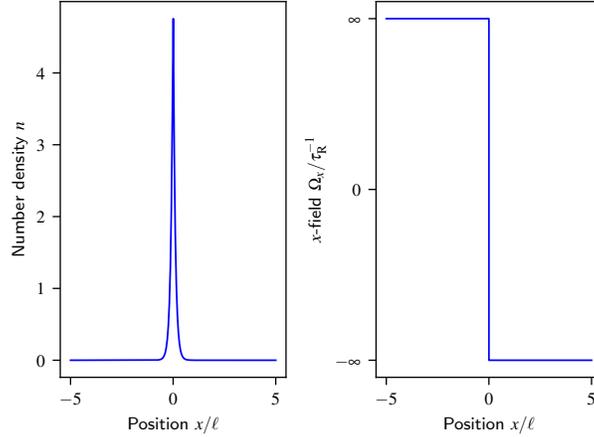


Figure 2: Number density for unlimited actuation.

5 Dynamic set point tracking

To illustrate the capabilities of our framework, we use MPC to steer a BD simulation to track a dynamic set point $x_{\text{sp}}(t)$ where we want the particles to gather. MPC is used to find the input field $\Omega_x(x, t)$ that achieves this. We also compare MPC to a heuristic solution.

Our stage cost is formulated as follows.

$$\begin{aligned}
 L &= L_P + L_\Omega \\
 L_P(P, x_{\text{sp}}(t)) &:= c_1 \langle (x - x_{\text{sp}}(t))^2 \rangle \\
 L_\Omega(\Omega_x(t), \Omega_x(t - \Delta)) &:= c_2 \left| \frac{\partial^2 \Omega_x}{\partial x^2} \right|^2 + c_3 |\Omega_x(t) - \Omega_x(t - \Delta)|^2
 \end{aligned} \tag{17}$$

L_P penalizes the expected square distance between the particles' x -position and the set point. L_Ω penalizes both controller variation in space and time by adding a cost to curvature in x and a cost to the change in input field. c_1 , c_2 , and c_3 are weights that determine the importance of each term. We prioritize set point tracking so we set $c_2, c_3 \ll c_1$.

We compare MPC to a heuristic controller that orients particles towards the set point, i.e.,

$$\Omega_x(x, t) = \begin{cases} \Omega_{\text{max}}, & x < x_{\text{sp}}(t) \\ 0, & x = x_{\text{sp}}(t) \\ -\Omega_{\text{max}}, & x > x_{\text{sp}}(t) \end{cases} \tag{18}$$

$$\Omega_y(x, t) = 0 \tag{19}$$

The dynamic set point function is chosen to be the following:

1. The set point starts at $x_{\text{sp}}(t) = 0\ell$.
2. At $t = 15\tau_R$, the set point jumps to $x_{\text{sp}}(t) = -2.5\ell$.
3. The set point follows a cosine function with amplitude $a = 2.5$ and period $b = 2$.
4. The simulation ends at $t = 51\tau_R$.

More formally, the set point is defined as

$$x_{\text{sp}}(t) = \begin{cases} 0, & t < t_s \\ -a \cos((t - t_s)/b), & t_s \leq t \leq t_s + 2\pi b \\ 0, & t_s + 2\pi b \leq t \leq t_f \end{cases} \tag{20}$$

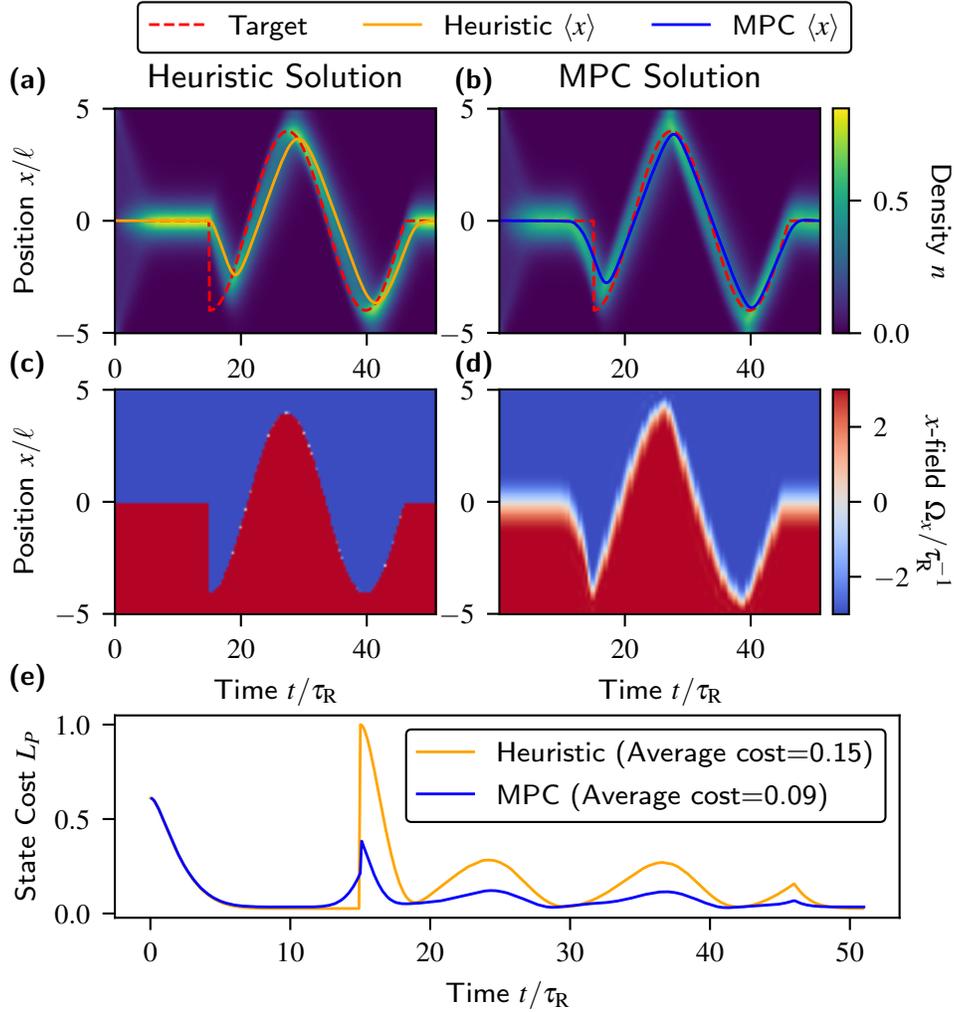


Figure 3: Dynamic set point tracking with a BD simulation using a heuristic policy and MPC. (a) and (b) show the number densities as the particles track the given set point for the heuristic and MPC solutions, respectively. (c) and (d) show the controller inputs to track the set point. (e) shows a comparison between the heuristic and MPC solutions' stage state cost, i.e., how far the system is from the set point along with an integrated state cost.

a and b are the amplitude and period, respectively, of oscillating portion of the dynamic set point. t_s is the time when the oscillation starts. t_f is the time when the simulation stops. We choose the period b such that the set point's maximum velocity is faster than the average translational velocity with a saturated controller, i.e., $\Omega_x(x) = \Omega_{\max}$. The set point is illustrated with the dashed red line in Fig. 3a. Note that the maximum velocity occurs when $x_{\text{sp}}(t) = 0\ell$. Thus, the controller will need to anticipate the limitation of the actuator and orient the particles in front of the set point.

The heuristic control results are shown in Fig. 3(a-b) and MPC results are shown in Fig. 3(c-d). The colors in Fig. 3(a, c) show the particles number density evolution $P_0(x, t)$ throughout the simulation as a function of space and time; The dotted line shows the dynamic set point $x_{\text{sp}}(t)$ and the solid lines show the expected value of the particle position $\langle x \rangle$. Fig. 3(b, d) colors show the actuator $\Omega_x(x, t)$ as a function of space and time; Fig. 3e compares the two controllers state cost throughout the simulation and reports the average state cost $(1/t_f) \int_0^{t_f} L_P(t) dt$. We do not compare the control costs since the heuristic controller has undefined curvature at x_{sp} .

The distributions in Fig. 3(a, c) show both controller track the set point. At the start, the particles are at steady state with $\Omega_x = 0$ and both controllers orient the particles to the center, which can be seen by the probability density starting off on the edges at $x = \pm 5\ell$ and accumulating at $x = 0$. At $t = 15\tau_R$, the set point changes to the cosine function and the particles' probability distribution functions follow the set point until the end of the simulation. However, Fig. 3a shows the heuristic controller's expected position $\langle x \rangle$ lags behind the set point, especially where the set point changes the fastest, e.g., $t = 15\tau_R$ when the set point jumps and $x_{sp}(t) = 0\ell$ when the set point speed is at its maximum. Fig. 3b shows that MPC anticipates changes in the set point and optimizes the actuator to keep the particles expected position close to the set point. A good example occurs at $t = 15\tau_R$ where the set point jumps from $x_{sp}(t) = 0\ell$ to $x_{sp}(t) = -2.5\ell$. As defined by the heuristic policy, $\Omega_x(x, t)$ only reacts to the set point change, whereas MPC anticipates the change in the set point and begins to move the particles to $x = -2.5\ell$ before the set point change arrives. Before the set point change, MPC incurs a higher cost for departing from the set point. However, after the set point change, MPC does not suffer from as large a jump in the stage cost compared to the heuristic solution.

Additionally, when the set point speed is at its maximum, the MPC controller keeps the expected particle position closer to the set point compared to the heuristic policy. While the set point is moving slow, i.e., when $\frac{\partial}{\partial t}x_{sp}(t) \sim 0$, MPC orients particle such that $\langle x \rangle$ is ahead of the set point. Then, when the actuator cannot match the needed velocity to match the set point velocity due to the input constraints, the set point $x_{sp}(t)$ catches up to expected position $\langle x \rangle$ and the state cost remains relatively smaller compared to the heuristic controller. The heuristic controller input, as shown in Fig. 3c, as prescribed, has a step at the set point. As seen in Fig. 3c, the MPC actuator's changes in space and time appear to be reasonable and not overly aggressive, given the task. From Fig. 3e, we can see the advantages of using MPC versus the heuristic controller. At first, both controllers have the same state stage cost. Near $t = 15\tau_R$, MPC incurs a higher state cost to move the particles to forecasted set point while the heuristic controller continues keeping particles in the center. Thus, when the set point changes from $x_{sp}(t) = 0\ell$ to $x_{sp}(t) = -2.5\ell$, the heuristic controller has the particles at the center and suffers from a large state cost. We can also see the result of MPC anticipating where the set point is going as MPC maintains the state cost lower compared to the heuristic solution. As a result, MPC has a lower average state cost. This example demonstrates the benefits of MPC compared to using a heuristic controller.

6 Stage cost for splitting particles

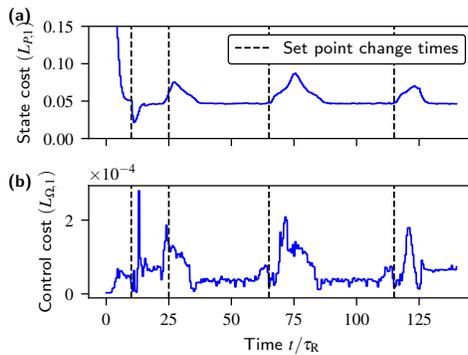


Figure 4: Stage cost for splitting particles. Dashed black lines denote times when the set point changes. (a) shows the state cost L_{P_1} and (b) shows the control cost L_{Ω_1}

To split the particles into two groups, we use the following stage cost

$$\begin{aligned}
 L_1 &= L_{P,1} + L_{\Omega,1} \\
 L_{P,1}(P(t), \mathbf{x}_{\text{sp},1}(t)) &:= c_1 \left(\left\langle (x - x_-(t))^2 \right\rangle_{x < 0} + \left\langle (x - x_+(t))^2 \right\rangle_{x > 0} \right) \\
 &\quad + c_2 (n(x < 0) - r_{\text{sp}}(t)n(x > 0))^2 \\
 L_{\Omega,1}(\Omega_x(t), \Omega_x(t - \Delta)) &:= c_3 \left| \frac{\partial^2 \Omega_x}{\partial x^2} \right|^2 + c_4 |\Omega_x(t) - \Omega_x(t - \Delta)|^2 \\
 \mathbf{x}_{\text{sp},1}(t) &:= [x_-(t) \quad x_+(t) \quad r_{\text{sp}}(t)]^T
 \end{aligned}$$

The stage cost still penalizes the expected square distance between the particles' and the set points, but now we have two set points $x_-(t)$ and $x_+(t)$. Thus, for particles with position $x < 0$, we penalize the expected square distance between the particles and x_- and for particles with position $x > 0$, we penalize the expected square distance between the particles and x_+ . We also penalize the ratio of particles in the two groups using the ratio set point $r_{\text{sp}}(t)$ that we want the particles to split into. The control cost $L_{\Omega,1} = L_{\Omega}$. c_1 , c_2 , c_3 , and c_4 are again the weights. c_4 is set to be larger than c_1 to prioritize the particles maintaining the desired ratios. c_2 and c_3 are kept small to prioritize the state cost. Fig. 4 shows the state and control stage costs.

As shown in Fig. 4a, the state cost $L_{P,1}$ initially drops as the particles are gathered to the center. At $t = 10\tau_R$, the set point changes to split the particles into two groups which causes the state cost to dip then increase back to a steady value. Subsequently, every time there is a ratio set point change, the state cost increases before the particles reach the desired ratio and the state cost stabilizes. The control cost $L_{\Omega,1}$ in Fig. 4b shows the control actions are not penalized very heavily. The control cost related to the curvature of the input field is small and most of the variation in the control cost is due to the change in the input field. Thus, every set point change, the control cost increases as the controller needs to change the input field.

7 Stage cost for orientational moment control

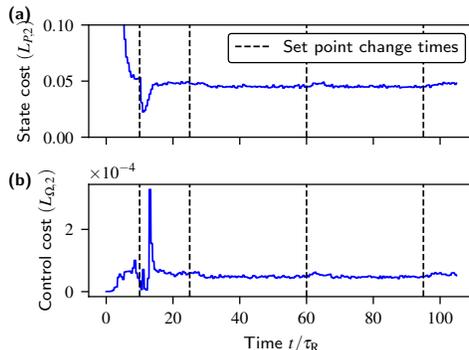


Figure 5: Stage cost for orientational moment control. Dashed black lines denote times when the set point changes. (a) shows the state stage cost L_P and (b) shows the control stage cost L_{Ω}

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

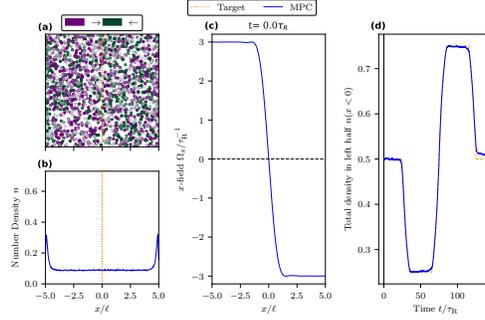
$$\begin{aligned}
L_2 &= L_{P,2} + L_{\Omega,2} \\
L_{P,2}(P(t), \mathbf{x}_{\text{sp},2}(t)) &:= c_5 \left(\left\langle (x - x_-(t))^2 \right\rangle_{x < 0} + \left\langle (x - x_+(t))^2 \right\rangle_{x > 0} \right) \\
&\quad + c_6 (n(x < 0) - n(x > 0))^2 \\
&\quad + c_7 \left| \frac{m_y(x)}{n(x)} - v_{\text{sp}}(t) \sin(\pi x/W) \right|^2 \\
L_{\Omega,2}(\Omega_x(t), \Omega_x(t - \Delta), \Omega_y(t), \Omega_y(t - \Delta)) &:= c_8 \left(\left| \frac{\partial^2 \Omega_x}{\partial x^2} \right|^2 + \left| \frac{\partial^2 \Omega_y}{\partial x^2} \right|^2 \right) \\
&\quad + c_9 \left(|\Omega_x(t) - \Omega_x(t - \Delta)|^2 + |\Omega_y(t) - \Omega_y(t - \Delta)|^2 \right) \\
\mathbf{x}_{\text{sp},2}(t) &:= [x_-(t) \quad x_+(t) \quad v_{\text{sp}}(t)]^T
\end{aligned}$$

The stage cost now penalizes the expected deviation of the y^* -velocity from the set point function $v_{\text{sp}}(t) \sin(\pi x/W)$. In addition, we penalize the curvature and change in time of the y -component of the input field. In Movie S2, we show that we

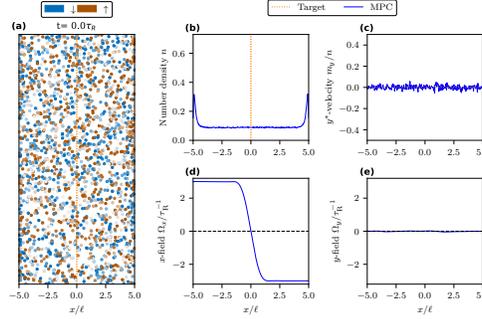
1. Accumulate particles into center and initialize $v_{\text{sp}} = 0$, i.e., no polar order in y .
2. Split particles into two equal groups centered at x_+ and x_- .
3. Set $v_{\text{sp}} = 0.4$ so the y^* -velocity fits to $0.4 \sin(\pi x/w)$.
4. Repeat step 3, but with a $v_{\text{sp}} = -0.4$.
5. Set $v_{\text{sp}} = 0$.

Fig. 5 shows the state and control stage costs. Up to $t = 25\tau_R$, the costs are identical to the splitting particles example. At $t = 25\tau_R$, the set point changes to $v_{\text{sp}} = 0.4$ and the state cost slightly decreases as the particles' velocity matches the set point function. Furthermore, the controller is able to concentrate the particles slightly more at x_- and x_+ using Ω_y . Before applying Ω_y , at $x = x_{\pm}$, $\Omega_x, \Omega_y = 0$. Thus, there is no input field at the set points to suppress rotational diffusion. When Ω_y is used to match the y^* -velocity to the set point function, the rotational diffusion is suppressed which results in the slight concentration of particles at x_{\pm} , slightly decreasing the state cost. At $t = 60\tau_R$, the set point changes to $v_{\text{sp}} = -0.4$ and the state cost increases temporarily before returning to the same value. At $t = 95\tau_R$, the particle velocity set point is 0, and we return to two groups of particles with little polar order in y .

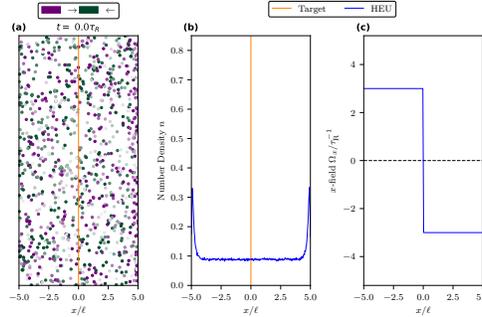
8 Movie descriptions



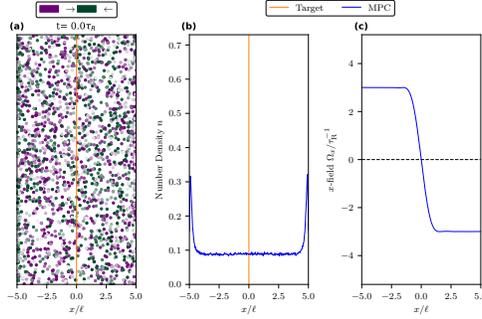
Movie S1: MPC for splitting population and juggling densities. (a) shows a subset of the particles in the BD simulation. The particles are colored by their orientation as indicated by the legend. The target positions are marked by the orange lines. (b) shows the KDE number density in blue; (c) shows the input field Ω_x ; (d) shows the achieved fraction of particles with position $x < 0$ in blue and the target fraction in orange. The black horizontal line indicates the time of the simulation.



Movie S2: MPC for splitting population and controlling y^* -velocity. (a) shows a subset of the particles in the BD simulation. The particles are colored by their orientation as indicated by the legend. The target positions are marked by the orange lines. (b) shows the KDE number density in blue; (c) shows the KDE y^* -velocity in blue along with the target y^* -velocity in orange; (d) shows the input field Ω_x ; (e) shows the input field Ω_y .



Movie S3: Heuristic control for dynamic set point tracking. (a) shows a subset of the particles in the BD simulation. The particles are colored by their orientation as indicated by the legend. The target position is marked by the orange line. (b) shows the KDE number density in blue. (c) shows the input field Ω_x from Eq. (18).



Movie S4: MPC for dynamic set point tracking. (a) shows a subset of the particles in the BD simulation. The particles are colored by their orientation as indicated by the legend. The target position is marked by the orange line. (b) shows the KDE number density in blue. (c) shows the input field Ω_x .

References

- [1] Michael te Vrugt and Raphael Wittkowski. Relations between angular and Cartesian orientational expansions. *AIP Adv.*, 10(3):035106, 2020. ISSN 2158-3226. doi: 10.1063/1.5141367.
- [2] Tomoyuki Mano, Jean Baptiste Delfau, Junichiro Iwasawa, and Masaki Sano. Optimal run-And-Tumble-based transportation of a Janus particle with active steering. *Proc. Natl. Acad. Sci. USA*, 114(13):E2580–E2589, Mar 2017. doi: 10.1073/PNAS.1616013114/-/DCSUPPLEMENTAL.
- [3] Lorenzo Caprini and Umberto Marini Bettolo Marconi. Active chiral particles under confinement: surface currents and bulk accumulation phenomena. *Soft Matt.*, 15(12):2627–2637, 2019. doi: 10.1039/C8SM02492H.
- [4] Jeremie Palacci, Stefano Sacanna, Asher Preska Steinberg, David J. Pine, and Paul M. Chaikin. Living crystals of light-activated colloidal surfers. *Science*, 339(6122):936–940, Feb 2013.
- [5] Pau Guillamat, Jordi Ignés-Mullol, and Francesc Sagués. Control of active liquid crystals with a magnetic field. *Proc. Natl. Acad. Sci. USA*, 113(20):5498–5502, 2016. doi: 10.1073/pnas.1600339113.
- [6] Sho C Takatori and John F Brady. Swim stress, motion, and deformation of active matter: effect of an external field. *Soft Matt.*, 10(47):9433–9445, 2014.