Check for updates

Received: 4 July 2026 | Revised: 19 August 2026 | Accepted: 14 September 2026

DOI: 10.1002/aic.70662

AIChE JOURNAL

RESEARCH ARTICLE

Process Systems Engineering

Online adaptive model-based stochastic control with limited data: A case study with a colloidal self-assembly system

Syeda Simra Shoaib[1] | Henry Hanchey[1] | Wei Sun[1] | Zheyu Jiang[2] | Xun Tang[1]

[1]Cain Department of Chemical Engineering, Louisiana State University, Baton Rouge, Louisiana, USA

[2]School of Chemical Engineering, Oklahoma State University, Stillwater, Oklahoma, USA

Correspondence
Xun Tang, Cain Department of Chemical Engineering, Louisiana State University, Baton Rouge, LA 70803, USA.
Email: xuntang@lsu.edu

Zheyu Jiang, School of Chemical Engineering, Oklahoma State University, Stillwater, OK 74078, USA.
Email: zheyu.jiang@okstate.edu

Funding information
National Science Foundation, Grant/Award Number: 2218077

Abstract

Data-driven optimal control is receiving increasing attention in colloidal self-assembly. While model-based control can suffer from predictive model errors, model-free control such as reinforcement learning can struggle with policy training which can result in waste of materials. Here we proposed an online adaptive model-based stochastic model predictive control framework to address the challenge of limited data for predictive model development. We evaluated its performance on a colloidal self-assembly system using Markov state models as the surrogate physical system. Results showed that online adaptation can substantially improve the prediction accuracy of an initially inaccurate model, enabling stochastic model predictive control to yield satisfactory performance. We further found that the control performance is dependent on objective function design and improved overall model accuracy does not necessarily guarantee better control. We anticipate the framework to also be applicable to other systems, where obtaining large amount of data for model development is challenging.

KEYWORDS
colloidal self-assembly, online adaptive modeling, residual network, stochastic model predictive control, variational recurrent neural network

1 | INTRODUCTION

Optimal control deals with the problem of finding the best control strategies to achieve the desired outcome and has received vast attention in various fields where "optimality" is desirable, such as finding the shortest pathway or time to the target with the lowest cost of the manipulation, etc. One example is the control of colloidal self-assembly with an external electric field, which is a highly stochastic and nonlinear process where particles can assemble into specific configurations with delicately designed manipulation of the applied voltage level.[1–5] However, due to the existence of multiple local energy minima in the system, assemblies can get stuck in undesirable structures that are time-consuming and challenging to get rid of.

To address the tradeoff between speed and quality of the assembled structures in the colloidal system, researchers have developed both model-based and model-free optimal control strategies. Model predictive control (MPC) and its variants are one of the most prevalent model-based optimal control strategies in industrial production. In

© 2026 American Institute of Chemical Engineers.
All rights reserved, including rights for text and data mining and training of artificial intelligence technologies or similar technologies.

an MPC setting, a predictive model is developed to forecast future process dynamics based on feedback from the underlying system, providing the basis for *online* optimization of the control actions to maneuver the system dynamics. While MPC is well-known for its capability of handling constraints, ensuring stability and safety, controlling highly nonlinear and stochastic processes with MPC can suffer from several major challenges. First, significant discrepancy between the predicted and the true process dynamics can undermine the control effectiveness that an accurate predictive model is necessary to ensure the performance. Second, online optimization of the control action can be time-consuming, especially for highly nonlinear and non-convex systems, impeding the practical implementation of the control. Third, typical MPC strategies perform well with deterministic processes but normally struggle with stochastic processes. To deal with these challenges, existing MPC-based colloidal self-assembly control strategies have sought to use simplified models or data-driven models to approximate the complex process dynamics, and to focus on the average behavior with a typical MPC setting or to use stochastic MPC (sMPC) to explicitly account for system stochasticity. Notable works in those frontends include Refs. [6–9], reinforcement learning-based MPC,[10] and stochastic MPC (sMPC),[11] just to name a few. Despite the great promise, these methods still require an accurate predictive model or large datasets to learn the underlying system dynamics to ensure the control efficacy, which can be unfortunately challenging to obtain.

Reinforcement learning (RL), on the other hand, has emerged as an effective alternative to MPC for controlling stochastic systems with limited-to-no prior knowledge over the past several years, as it can learn the control policy by directly interacting with the system, bypassing the need for an explicit model.[12] Recent advancements in applying RL for colloidal self-assembly systems include Refs. [8,13–17]. However, control with RL normally requires a large number of interactions with the system to ensure sufficient sampling for policy convergence, which again could face the same challenge as with MPC in terms of obtaining large data.[18] Furthermore, the slow convergence of the control policy could further result in waste of materials, as demonstrated in Ref. [17], that hundreds of thousands of trials would be needed to learn an effective control policy. For further reading on applying RL for colloidal self-assembly, we refer the readers to review articles.[5,19]

In this study, we focus on the problem of stochastic optimal control with limited data, and we overcome the impact of modeling error due to limited data with an online adaptive sMPC framework as outlined in Figure 1. Specifically, given a stochastic process of interest, for example, an electric field-mediated colloidal self-assembly system (to be described in detail later), we first develop a roughly estimated predictive model, for example, a variational recurrent neural network (VRNN), using limited samples from the system. With this initial predictive model, we then solve the optimal control actions with an sMPC setting to explicitly account for system stochasticity. The controlled outcome from the true system is then fed back to a residual neural network, which performs online correction of the VRNN model prediction by minimizing the difference between the predicted and the true system dynamics. Successive predictions will then come from the online corrected model for improved accuracy, thus leading to improved control performance. With this framework, we can reduce the amount of online data required by typical RL algorithms with a roughly estimated initial model, we can also improve the performance of sMPC with the online model adaptation, mitigating the impact of modeling error due to limited data.

We demonstrate the efficacy of the proposed framework with an electric field-mediated 2-dimensional colloidal self-assembly system, same as in Refs. [17,20,21]. In this system, 300 spherical SiO_2 colloidal particles are suspended in a batch system with four electrodes generating an electric driving force to function as the control inputs to influence particle movement for assembly. In this particular system, we consider compressing force generated by AC fields at specific frequency, with its strength (representable by a dimensionless metric λ) increases as the voltage magnitude increases.[4,22] At different voltage levels, the system could form different morphologies including randomly dispersed fluid state (labeled in dark blue in Figure 2), defective states (labeled in orange in Figure 2) and ordered crystalline states (labeled in red in Figure 2). The assembly dynamics are typically described by Brownian dynamics, as detailed in Ref. [20], as a result of multiple forces in the system, controllable by manipulating the applied voltage levels. Therefore, the specific control problem here is equivalent to designing the optimal manipulation of the voltage levels to drive the assembly to the target state, such as the ordered structure in Figure 2.

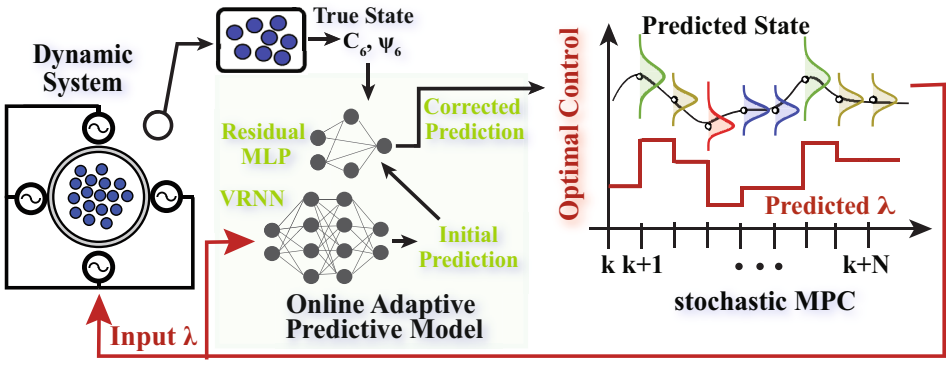


FIGURE 1 The proposed optimal control framework features a residual neural network to iteratively adapt the originally inaccurate model with incoming process measurements for improved accuracy, to ensure the control effectiveness of the stochastic model predictive control.

15475905, 0, Downloaded from https://aiche.onlinelibrary.wiley.com/doi/10.1002/aic.70662 by Oklahoma State University, Wiley Online Library on [25/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

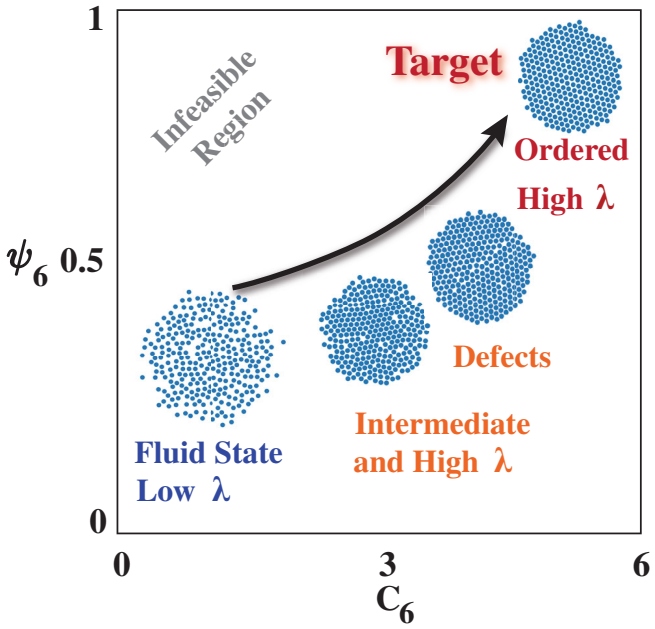


FIGURE 2 Different colloidal assembly configurations in the order parameter $C_6 - \psi_6$ space. Low C_6 and ψ_6 states represent fluid-like states, low ψ_6 high C_6 states correspond to defective states, high C_6 and ψ_6 states are the target ordered states. Note that states with a low C_6 but high ψ_6 are not feasible, due to the definitions of the order parameters (see Section 4).

In previous studies, we investigated this system with MPC using a simplified Langevin equation as the predictive model,[9] Dynamic Programming with Markov state models,[20] and RL with Brownian Dynamics simulations.[17] While these studies proved the feasibility of optimal control of the system to the target ordered state, we have previously left the data acquisition challenge untreated. Here, referencing to those successes and also to enable a direct comparison, we consider the same four discretized voltage levels, represented by the dimensionless metric $\lambda = 0.2, 0.9, 2.0, 19.7$ as the control inputs, and use order parameters $C_6 \in [0, 6]$ and $\psi_6 \in [0, 1]$ to quantify the assembly state as feedback. The detailed experimental setup and definitions of the order parameters are given in Ref. [20]. Furthermore, for computational effectiveness, we use an accurate Markov state model (aMSM, trained with more than 10,000 Brownian Dynamics simulations) as the true process to be controlled, instead of real experiments or the detailed Brownian Dynamics simulations which can take hours to solve for one simulation. To evaluate the performance of the proposed control framework, we compare the following three scenarios:

1. Using the aMSM as the predictive model to calculate the control action to control the process, that is, aMSM. This represents the ideal case where the true process model is used for prediction and is used to verify the applicability of the sMPC algorithm in controlling the colloidal system, as well as to set the performance upper bound reference.
2. Using the inaccurate initial predictive VRNN model developed with limited data without online adaptation (trained with only 419 Brownian Dynamics simulations) to calculate the control action to control the aMSM. This shows how model inaccuracy affects the control performance and is also used as a benchmark to evaluate the benefits of the adaptive modeling.
3. Using the proposed online adaptive predictive model to calculate the control action to control the aMSM.

The detailed descriptions on λ, the order parameters, the development of the aMSM, the inaccurate initial VRNN model, and the online corrective residual neural network are given in Section 4.

2 | RESULTS

We first demonstrate the efficacy of the proposed framework by controlling the system to the ordered state starting from an initial fluid state within 1000s of process time, which was previously proven sufficient for accomplishing the task and corresponds to 10 transition steps in the aMSM as the Markov transition time is 100 s.

In all the aforementioned three scenarios, at each time step, 100 independent simulations using the corresponding predictive model from the current state are generated to calculate the mean predicted states for optimal control action calculation. The first control action is then applied to the aMSM, and the one-step-forward aMSM prediction is then used as the new state in the predictive model for future prediction and control action calculation. This aMSM predicted state is also used by the residual neural network to learn the correction factors that can mitigate the VRNN prediction error in the proposed online adaptive model-based control framework. This process is iterated until reaching the end of the 10 transition steps, concluding a single rollout of 1000s simulation. If the ψ_6 value obtained at the end of the rollout is ≥ 0.95 (as we have formulated a terminal reward sMPC problem as detailed in Section 4), this controlled trial is denoted as a success. Since the proposed online adaptive model-based control framework requires iterative updates of the residual neural network with the newly collected transition pairs from the aMSM, we allowed the control to continue up to 200 rollouts, that is, 200 independent simulations of 1000s trials, to ensure convergence of the model and to investigate how many rollouts are needed for model convergence if ever. For a fair comparison, performance of control using the uncorrected VRNN model and the aMSM as the predictive models is also analyzed based on 200 rollouts controlled results.

Figure 3 summarizes the 200 controlled rollouts for each of the three control settings. The thicker plots are the averages and the shaded areas showing the span of the states visited by the individual rollouts. Starting from the same initial fluid state $C_6 = 2.10, \psi_6 = 0.05$, control using predictions from the aMSM (red) showed the best performance as expected, since it used the same model as the underlying process, landing in the most upper right state in the $\psi_6 - C_6$ state space with an overall success rate of 94%, that is, 188 out of 200 rollouts have reached the target state within 10 control steps. Control with the online adaptive predictive model (purple) showed the second best performance with a success rate of 85%,

15475905, 0, Downloaded from https://aiche.onlinelibrary.wiley.com/doi/10.1002/aic.70662 by Oklahoma State University, Wiley Online Library on [25/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

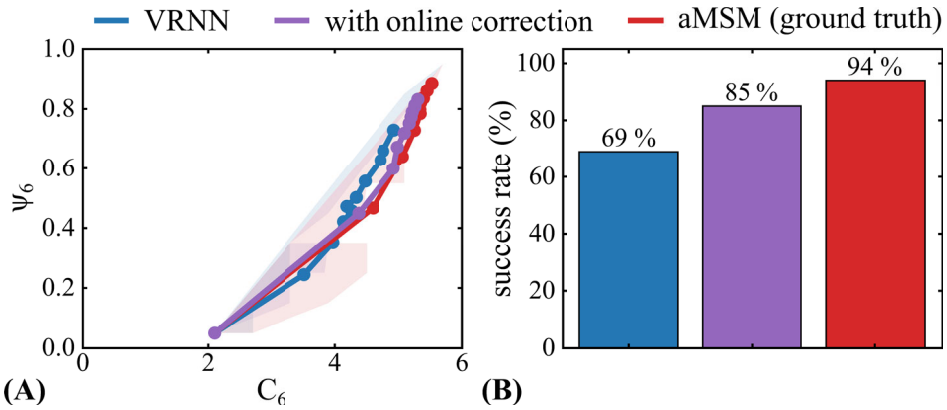


FIGURE 3 Comparison of 200 rollouts controlled results using uncorrected VRNN (blue), online adaptive model (purple), and aMSM (red) as the predictive model in the sMPC framework, showing substantial benefits of the proposed online adaptive model-based control framework. (A) controlled rollouts in the state space. (B) cumulative success rates of the three control settings.

demonstrating a substantial improvement over control using the uncorrected VRNN model (blue) which showed a success rate of only 69%. These observations confirmed that the proposed online adaptive model-based sMPC is a viable approach to controlling the system for achieving the target state and is effective in overcoming modeling errors to ensure control performance.

Investigating the model prediction accuracy in terms of absolute difference between the mean one-step-forward prediction from the predictive model and the aMSM, starting from the same state and under the same control action, reveals that online adapting the model can significantly improve the model accuracy (purple) as shown in Figure 4A,B. Without online correction, prediction errors are sustained at a relatively high level (blue), whereas with correction, prediction errors were substantially mitigated (purple). Despite the obvious prediction error in the online adapted model, the sMPC framework seemed to have been able to handle this moderate modeling error, yielding satisfactory controlled results as shown in Figure 3. Note that the prediction with the predictive model is calculated by averaging 100 independent simulations, whereas the *true* prediction with the aMSM is computed directly from the Markov state transition matrices. As a result, due to the stochasticity, even when the aMSM is used as the predictive model (red), there exists a discrepancy between the predicted and the ground truth prediction.

Figure 4C,D presents the evolution of the residual neural network learned correction factors over each transition step and rollout. Correction factors for ψ_6 seemed to have converged faster than that of C_6, indicating a rollout number of as few as about 40 would be sufficient (Figure 4D), whereas it would require at least 120 rollouts for C_6 correction factors to converge (Figure 4C). Within each rollout, corrections for both order parameters favored up-corrections (red color) at early stages but shifted to down-corrections (blue color) in later steps, suggesting that the predictive model tends to overestimate the prediction at the beginning and then counteracts this by down-correcting the predictions. This could be a reason why at early steps prediction errors from the corrected model are higher than that from the uncorrected model in Figure 4B. Investigating the absolute prediction errors by models obtained at different rollouts showed that, overall, prediction error decreased as training rollouts increased, but more rollouts did not guarantee a better model as the error fluctuated over the training process (Figure 4E,F). According to Figure 4E,F, the lowest rolling average-processed absolute ψ_6 prediction error was with model obtained with 60 rollouts (with mean$\left|\psi_6^{\text{true}} - \psi_6^{\text{pred}}\right| = 0.12$), showing a slightly lower value as compared to that with model obtained with 180 rollouts (with mean$\left|\psi_6^{\text{true}} - \psi_6^{\text{pred}}\right| = 0.17$). Note that, the heatmaps in Figure 4C,D are smoothed by taking a rolling average of every 11 consecutive rollouts for demonstration purposes, and the dots in Figure 4E,F are prediction errors at each rollout, whereas the linear plot represents the average prediction error for the corresponding model. As the model was only updated every 20 rollouts, the same model was used for the successive 20 rollouts, but due to the use of different control inputs and the process stochasticity, the prediction error of the same model would be different (the dotted plot).

To understand if the Rollout 60 model would indeed outperform the model obtained at the end of the training, that is, Rollout 180 model, we then included this model as a comparison in the following analysis.

We first sought to understand the differences among the correction factors from the two models by projecting the correction factors into the $\psi_6 - C_6$ state space in Figure 5. Note that Figure 5A,D is the initial correction factors, which are randomly selected and bear no physical meaning, presented here for completeness. Comparing Figure 5B,C, as well as Figure 5E,F reveals that, both Rollout 60 and 180 models tend to feature a down-correction of ψ_6 in most states (E and F), whereas Rollout 60 model features mainly up-corrections of C_6 (B) while Rollout 180 model features mainly down-corrections of C_6 (C). This could be that with more samples, Rollout 180 model tends to favor a more balanced prediction accuracy over a wider range of states, whereas the Rollout 60 model is less generalizable.

We then sought to understand the model accuracy under other initial conditions. To investigate this, we randomly selected three additional non-absorbing initial states and conducted 100 independent simulations with 100 independently and randomly generated input trajectories, using the uncorrected VRNN model, the Rollout 60 model, the Rollout 180 model, and the aMSM. Comparing the 100 rollouts-averaged predictions from the four models in Figure 6 reveals that Rollout 180 model (purple) yielded the closest ψ_6 predictions to the aMSM predictions (red) in all three initial conditions, followed by

15475905, 0, Downloaded from https://aiche.onlinelibrary.wiley.com/doi/10.1002/aic.70662 by Oklahoma State University, Wiley Online Library on [25/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

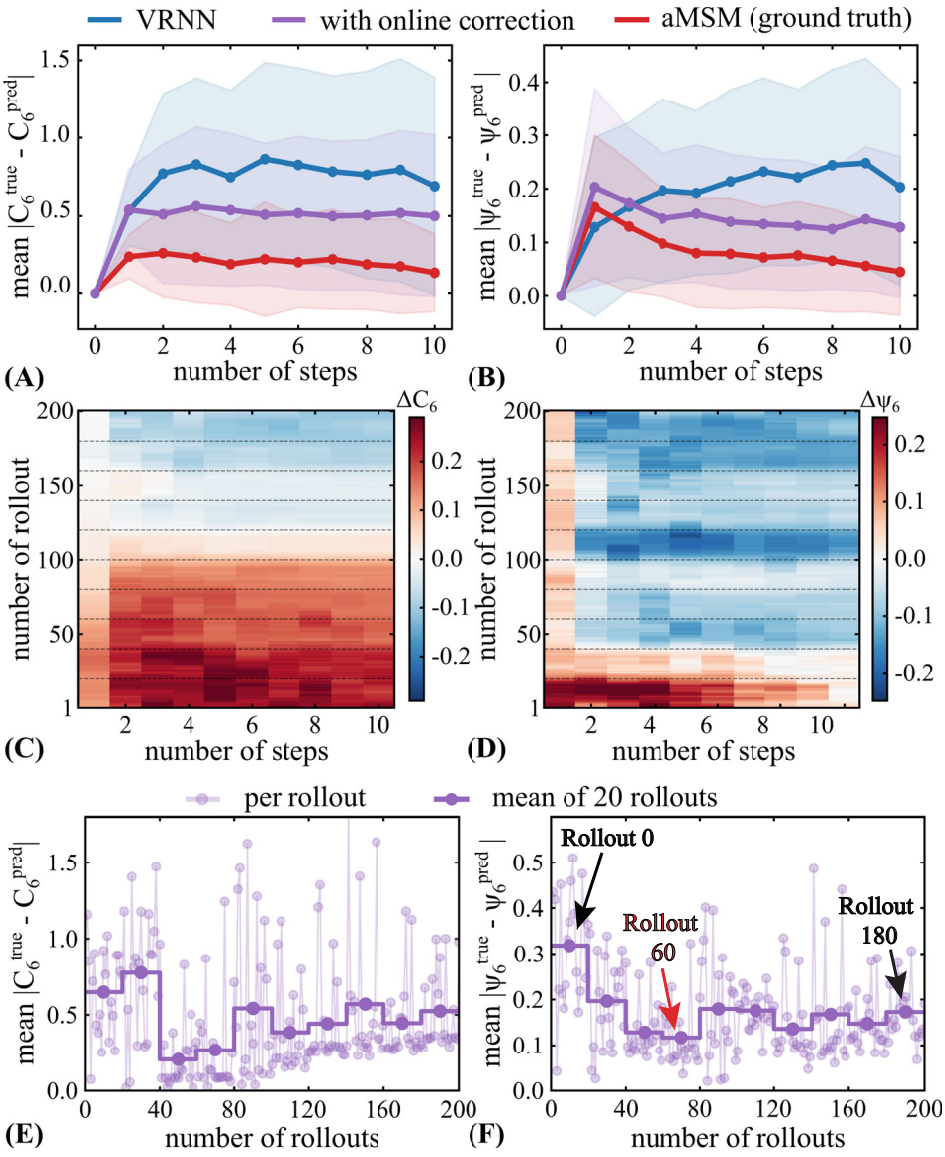


FIGURE 4 Modeling error evaluation showing online model adaptation substantially improves model accuracy. (A) and (B) Absolute mean error of order parameters predicted from the predictive model and the aMSM model at each transition step, over 200 rollouts. (C) and (D) Correction factors from the residual neural network over rollouts and transition steps for the two order parameters. (E) and (F) Absolute mean prediction error of the online adapted model over rollouts.

the Rollout 60 model (green), whereas the uncorrected VRNN model yielded the least accurate predictions, confirming the reliability of the model learned with the online adaptation algorithm. This confirms the above hypothesis that Rollout 180 model has advantages over the Rollout 60 model in terms of generalizability, due to additional training rollouts. However, all three non-aMSM models showed evidently different C_6 predictions as compared to the aMSM predictions, but this could be a visual artifact due to the difference in the scale of the two order parameters (max $C_6 = 6$ vs. max $\psi_6 = 1$).

We then proceeded with control using the four models, and the 100 simulations of controlled results are summarized in Figure 7. According to the results, while control with prediction using the aMSM still achieved the best performance with success rates of 95%, 97%, and 88% in systems starting from fluid (Figure 7A, with $C_6 = 1.5, \psi_6 = 0.05$), moderate defective (Figure 7B, with $C_6 = 4.5, \psi_6 = 0.25$), and close-to-perfect state (Figure 7C, with $C_6 = 5.7, \psi_6 = 0.9$) respectively, control using the two online adapted models also showed satisfactory performance with slightly lower success rates, and showed significant improvements from control with the uncorrected VRNN model. Comparing the performance between the two corrected models showed no significant difference except in the system started with the moderate defective state (Figure 7B, with $C_6 = 4.5, \psi_6 = 0.25$), where Rollout 60 model-based control yielded a success rate of 92% (green dot in Figure 7D), whereas Rollout 180 model-based control yielded a success rate of 86% (purple dot in Figure 7D). However, overall, both corrected models provided comparably satisfactory performances in the tested scenarios, confirming the applicability of the learned model from one initial state to other initial states in the sMPC framework. The detailed success rates are summarized in Table 1.

One advantage of MPC over many alternative control strategies is its ability to explicitly incorporate state and input constraints into the control formulation. In this study, however, the control objective was designed to drive the system toward a single terminal target state rather than to constrain the system to specific regions of the order-parameter space or to explicitly avoid undesirable states. This

15475905, 0, Downloaded from https://aiche.onlinelibrary.wiley.com/doi/10.1002/aic.70662 by Oklahoma State University, Wiley Online Library on [25/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

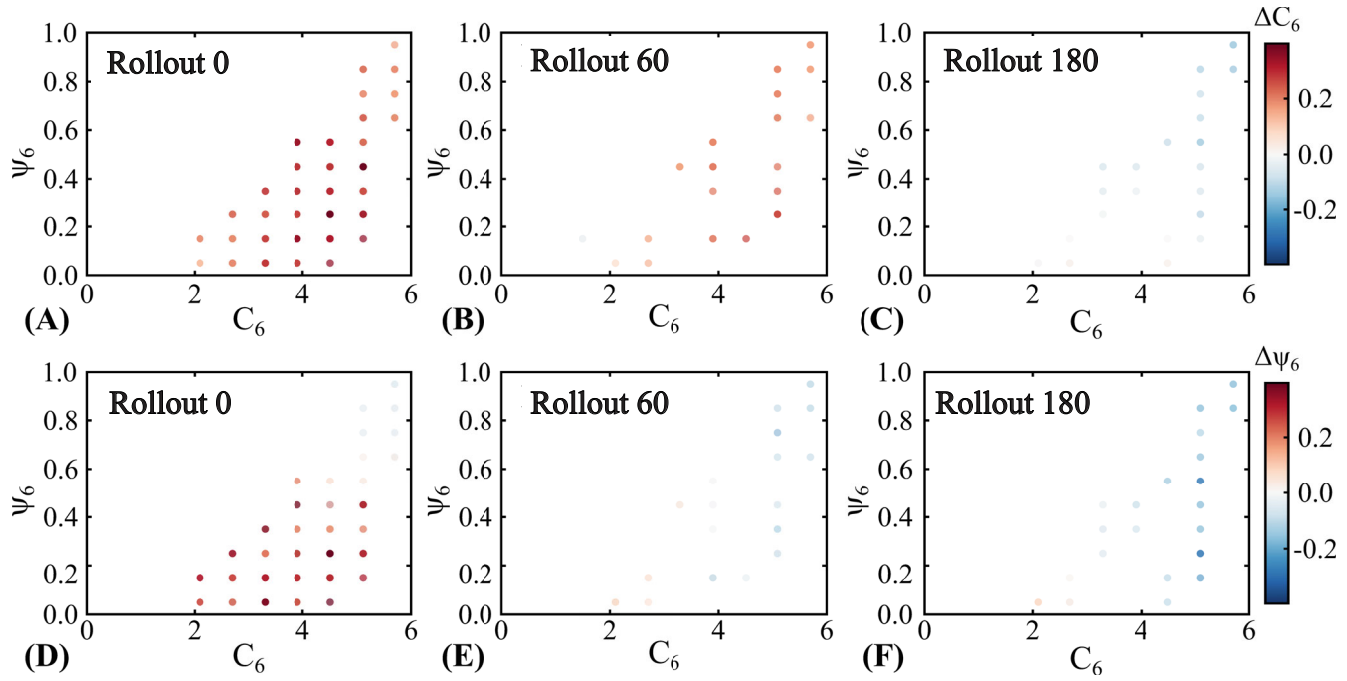


FIGURE 5 Evolution of residual model correction factors in the $\psi_6 - C_6$ state space over rollouts, showing the shifting from up-corrections to down-corrections as rollout number increases.

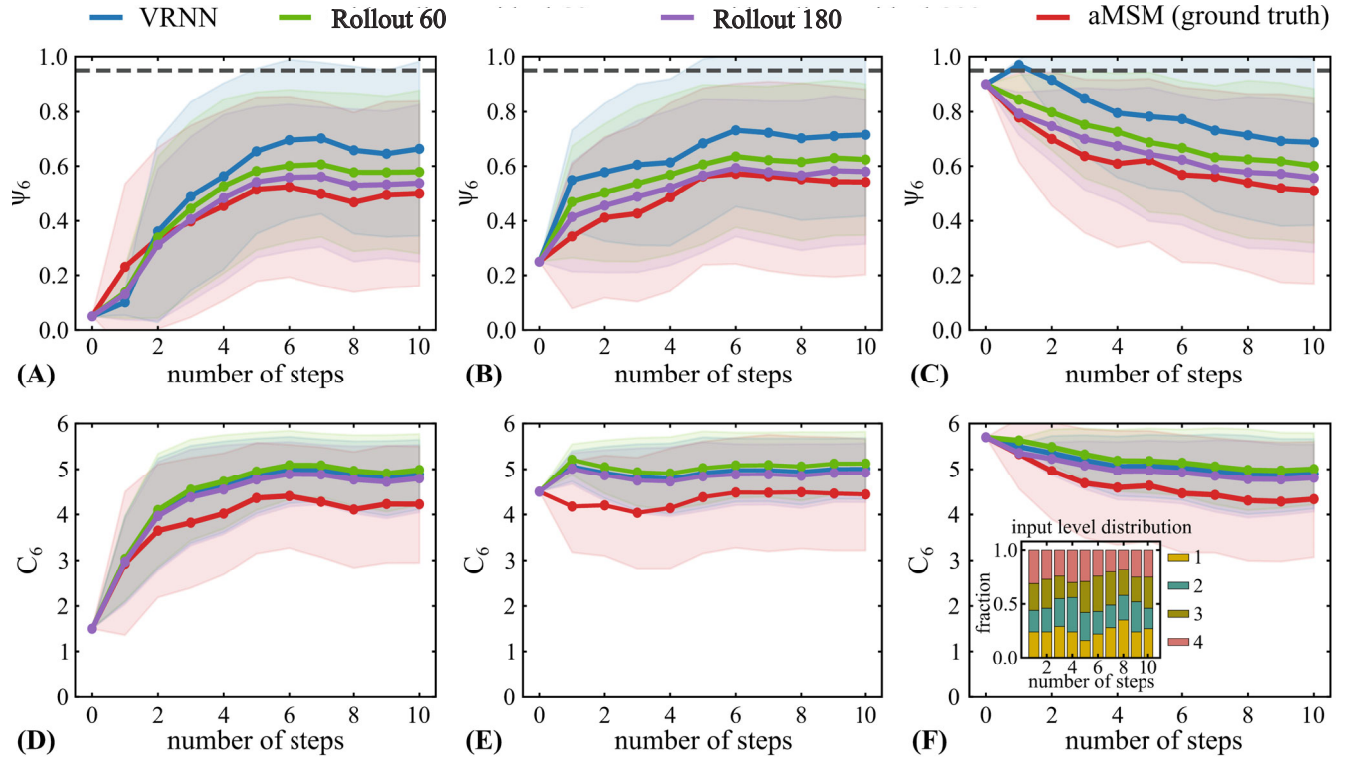


FIGURE 6 Uncontrolled model predictions starting from three new initial conditions, showing the reliability of the online adapted models. Solid plots are 100 rollouts-averaged predictions with shaded areas representing the spanning of all simulations. (A) and (D) are for initial fluid state with $C_6 = 1.5, \psi_6 = 0.05$; (B) and (E) are for initial moderate defective state with $C_6 = 4.5, \psi_6 = 0.25$; (C) and (F) are for initial close-to-perfect state with $C_6 = 5.7, \psi_6 = 0.9$. Inset figure in (F) summarizes the portion of each input levels applied at each transition steps in all the 100 randomly generated control input trajectories.

15475905, 0, Downloaded from https://aiche.onlinelibrary.wiley.com/doi/10.1002/aic.70662 by Oklahoma State University, Wiley Online Library on [25/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

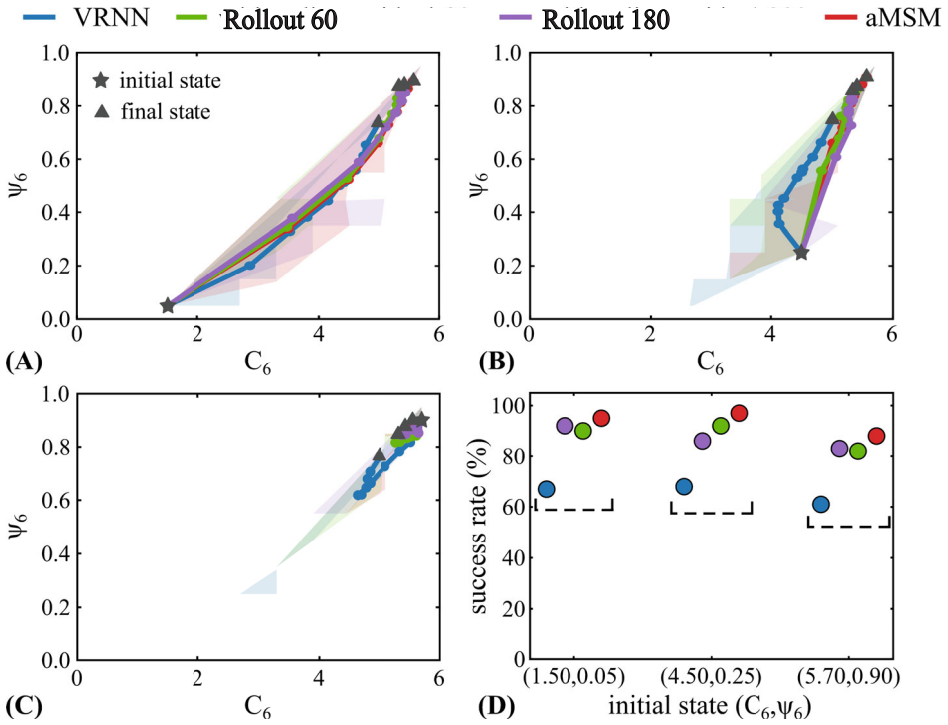


FIGURE 7 One hundred controlled results using the sMPC framework using the uncorrected VRNN (blue), Rollout 60 (green), Rollout 180 (purple), aMSM (red) as the predictive models for three new initial conditions. Thick plots are 100 rollouts averages; shaded areas present the total spanning of the dynamics. (A)–(C) Controlled state evolution starting from fluid (with $C_6 = 1.5, \psi_6 = 0.05$), moderate defective (with $C_6 = 4.5, \psi_6 = 0.25$), and close-to-perfect state (with $C_6 = 5.7, \psi_6 = 0.9$) states. (D) Success rates of the four control settings for each initial state.

TABLE 1 Success rates of controlled simulations starting from different initial conditions using the uncorrected model, aMSM, and the two corrected models.

Initial state (C_6, ψ_6)	Uncorrected VRNN (%)	aMSM (%)	Rollout 60 (%)	Rollout 180 (%)
(1.50, 0.05)	67	95.0	90	92
(4.50, 0.25)	68	97	92	86
(5.70, 0.90)	61	88	82	83

was primarily due to the difficulty of identifying such regions in the order-parameter space, and if these regions can be characterized, they could be incorporated directly as state constraints within the MPC formulation. As a related demonstration, we considered two additional formulations of the objective function: one that maximizes the immediate reward ($\max_{a_t} \mathbb{E}[\psi_6(t+1)]$, Figure 8A,B), and another that tracks a predefined reference trajectory ($\min_{a_{t:t+H-1}} \mathbb{E}\left\| \begin{matrix} C_6(t) - C_6^{\text{ref}}(t) \\ \psi_6(t) - \psi_6^{\text{ref}}(t) \end{matrix} \right\|_2$, Figure 8C,D). The reference trajectory was defined as the average controlled trajectory obtained using the aMSM-based controller. As in the terminal-reward case, the online correction model was trained from scratch over 200 rollouts for each objective. Among the resulting models, the one with the lowest prediction error was selected as the best online-corrected model (purple curves in Figure 8) for subsequent control evaluation.

The 200 controlled results with the immediate reward objective function in Figure 8A,B demonstrated that while the aMSM-based control still yielded the best control performance followed by the corrected model-based control, the VRNN-based control also gave a reasonably good performance comparable to the corrected model-based control. This phenomenon suggests that the design of the objective function plays a significant role in determining the control performance. The 200 controlled results of tracking the reference pathway (red) in Figure 8C,D demonstrated that control with the online corrected model can lead to better tracking of the reference (purple), as compared to control with the original VRNN model (blue). Note that Figure 8C shows the averaged trajectory in the state space, whereas Figure 8D summarizes the pair-wise absolute mean difference between the controlled and the reference trajectories. Altogether, these findings suggest the feasibility of incorporating state constraints in the proposed control framework, and that while model accuracy is important, the design of the objective function also plays a significant role in determining the control performance.

To assess the generalizability of the framework to models with substantially higher state-space resolution, we then considered a previously developed Markov state model for the same physical system where the same order parameter state space was evenly discretized into 6000 states. For clarity, we label this model as aMSM6000 and the previously discussed 100-state model as aMSM100 in the following analysis. Following the same workflow used for aMSM100, we first performed online adaptation under a terminal reward objective function over 200 rollouts to identify the corrected model with the lowest prediction mean squared error, and then evaluated the control performance in terms of achieving the target ordered state.

Evaluating the model prediction accuracy by conducting simulations with 1000 randomly generated input trajectories revealed similar observations as in the 100-state MSM model system (Figure 9A,B),

15475905, 0, Downloaded from https://aiche.onlinelibrary.wiley.com/doi/10.1002/aic.70662 by Oklahoma State University, Wiley Online Library on [25/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

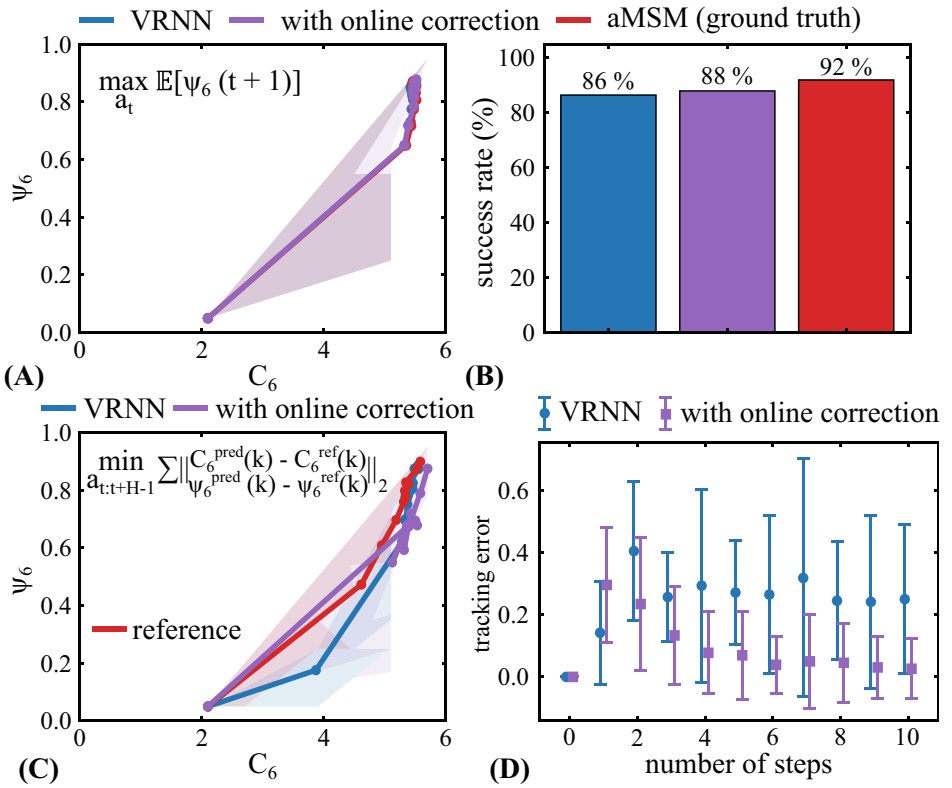


FIGURE 8 Evaluation of control with different objective functions. (A) 200 rollouts of controlled trajectories in the state space starting from $C_6 = 2.1, \psi_6 = 0.05$ with an objective function of maximizing the expected immediate next step ψ_6 value, using the 100-state aMSM as the true environment. (B) Success rates of the control for each of the three cases. (C) 200 rollouts of controlled trajectories in the state space starting from $C_6 = 2.1, \psi_6 = 0.05$ with a reference pathway tracking objective function, indicated by the red plot, using the 100-state aMSM as the true environment. (D) Absolute mean error between the controlled and the reference pathway at each time step (blue for VRNN-based control, purple for online corrected model-based control).

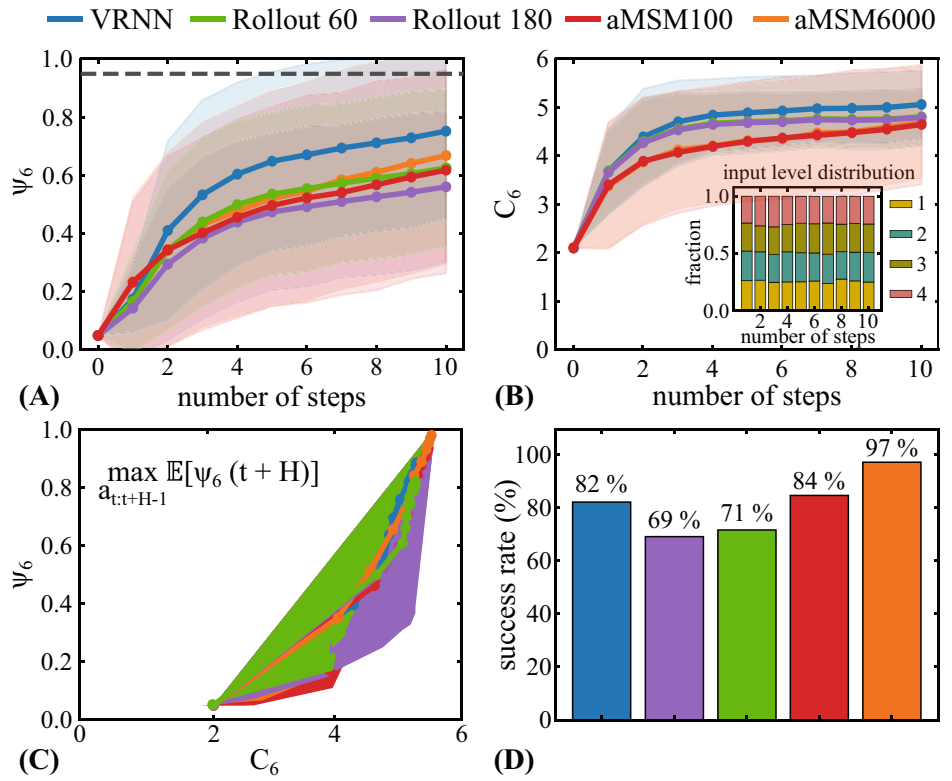


FIGURE 9 Framework evaluation with a 6000-state Markov model (i.e., aMSM6000). (A) and (B) Uncontrolled model predictions from the initial condition $C_6 = 2.1, \psi_6 = 0.05$, with 1000 randomly generated input trajectories. Solid lines represent the mean of 1000 rollouts, and shaded regions indicate the range across all rollouts. Inset figure in (B) shows the fraction of each control input sampled at each step. (C) Two hundred controlled trajectories in the state space using different predictive models: VRNN (blue), aMSM100 (red), aMSM6000 (orange), Rollout 180 (purple), and Rollout 60 (green), with aMSM6000 as the process to be controlled. (D) Success rates across the 200 controlled rollouts for the five sMPC configurations.

that while aMSM100 model provided the best approximation to the aMSM6000 simulations, online corrected models can improve the overall prediction accuracy over the original VRNN model, especially in terms of the ψ_6 (orange plot is for aMSM6000 model, red is for aMSM100, purple is for Rollout 180 model, green is for Rollout 60 model, and blue is for the original VRNN model prediction). This confirms that the proposed framework can improve the model accuracy. However, the 200 controlled results with the terminal reward-

15475905, 0, Downloaded from https://aiche.onlinelibrary.wiley.com/doi/10.1002/aic.70662 by Oklahoma State University, Wiley Online Library on [25/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

based objective function in Figure 9C,D demonstrated that, while the aMSM-based control (orange) yielded the best performance in terms of achieving the ordered target state, followed by the aMSM100-based control, VRNN-based control also yielded a similar success rate to the aMSM100-based control, evidently outperforming that of the two online corrected model-based control.

Several reasons could have contributed to this. First, although the 100-state Markov model achieves higher overall prediction accuracy, the continuous-state VRNN retains finer-grained state information that may be more relevant for control action selection. Consequently, despite its lower global prediction accuracy, the VRNN can still yield comparable control performance when applied to the higher-resolution aMSM6000 system. This may also explain why the VRNN performs less effectively when controlling aMSM100 (Figure 3), that the coarse discretization smooths fine-scale dynamics in aMSM100 model, whereas the VRNN, trained with limited continuous-state data, may not accurately reproduce the aggregated transition behavior of the coarser state representation. Second, MPC performance depends primarily on the accuracy of control-relevant dynamics rather than on overall state-transition accuracy. Thus, although the online-corrected Rollout 60 and Rollout 180 models provide more accurate global approximations of the aMSM6000 dynamics, the VRNN may better preserve the local, action-dependent dynamics that determine optimal control decisions. Examination of the VRNN training data further showed that, accidentally, the sampled trajectories substantially covered the system's optimal control pathway (Figure S1), which may have enhanced its accuracy in the most decision-relevant regions of the state space. This observation is consistent with prior studies showing that greater overall model accuracy does not necessarily lead to superior closed-loop control performance.[23–25] These results further suggest that model selection for control application should consider control-relevant metrics, such as accuracy along optimal trajectories, action-dependent transition fidelity, etc., in addition to global state prediction accuracy.

3 | CONCLUSION AND DISCUSSION

In this work, we presented an online model adaptation approach to iteratively update model accuracy of a stochastic, nonlinear model to address the challenge of obtaining large data for data-driven model development. We then coupled the online model adaptation algorithm with stochastic model predictive control to control an electric field-mediated stochastic colloidal self-assembly process. With the results on the 100-state Markov state model, we demonstrated that online adaptation is a powerful way of improving model accuracy and holds great generalizability to other conditions, despite the fact that the adapted model was learned with samples generated with the optimal control actions, which could be biased. The high success rates of achieving the target states further confirmed the applicability of stochastic MPC as a reliable alternative to reinforcement learning and offline optimal control such as dynamic programming in controlling the stochastic process while avoiding waste of materials during the initial control period. Evaluation using the 6000-state Markov model further showed that, although the online adaptation algorithm improved the overall model prediction accuracy, this improvement did not necessarily lead to a better closed-loop control performance. This finding highlights the limitation of relying solely on global state-transition accuracy for model selection and suggests that additional control-relevant metrics, such as action-ranking fidelity etc., should also be considered, especially when models are intended for closed-loop control.

Moving forward, to improve the model accuracy with greater generalizability, perturbations in the control actions could be considered to explore regions outside of the optimal control actions to cover a wider region of the state space. Advanced sampling methods such as Bayesian sampling based on the estimated modeling error can also be incorporated to accelerate exploration and model improvement. Future work should also focus on the design principles of objective functions, as well as model training and selection criteria for reliable performance in closed-loop control. We anticipate the proposed stochastic MPC with the online predictive model adaptation to hold great promise for stochastic systems where obtaining large data for model development or control policy learning is challenging.

4 | METHODS

4.1 | Colloidal self-assembly-related metrics

1. *Control input* λ: the control input, the dimensionless field strength representation λ is calculated as: $\lambda = \frac{\pi\varepsilon_m a^3 \left(f_{CM}\frac{V_{app.}}{\sqrt{8}d_g}\right)^2}{kT}$, where $a = 1430\,\text{nm}$, is particle radius, $\varepsilon_m = 78$, is the medium dielectric permittivity, $f_{CM} = -0.4667$, is the Clausius-Mosotti factor for the electric field, $d_g = 100\,\mu\text{m}$, is the electrode gap, k is the Boltzmann's constant and T is temperature. $V_{app} = \alpha V_{max}$, where $\alpha = [0.1, 0.2, 0.3, 0.95]$ and $V_{max} = 2.0128\,\text{V}$ in this work, same as in previous studies.[17,20,26]
2. *Order Parameters*: ψ_6 captures assembly bond orientational and is calculated according to: $\psi_6 = \left|\frac{1}{N}\sum_{j=1}^{N}\psi_{6,j}\right|$, with $\psi_{6,j} = \frac{1}{N_{Cj}}\sum_{k=1}^{N_{Cj}} e^{i6\theta_{jk}}$, where $N = 300$ is the total number of particles in our system, $\psi_{6,j}$ is the local 6-fold bond orientation order of particle j, with N_{Cj} being the number of neighbors within the first radius of gyration $g(r)$ peak of particle j, and θ_{jk} is the angle between particle j and each neighboring particle k with an arbitrary reference direction. C_6 captures the compactness of the assembly and is defined as: $C_6 = \frac{1}{N}\sum_{j=1}^{N}\frac{C_{6j}}{\langle C_6\rangle_{HEX}}$, with $C_{6j} = \sum_{k=1}^{N_{Cj}}\begin{bmatrix}1, \chi_{6,jk} \geq 0.32\\ 0, \chi_{6,jk} < 0.32\end{bmatrix}$, where $\chi_{6,jk} = \frac{\left|Re\left[\psi_{6,j}\psi_{6,k}^*\right]\right|}{\left|\psi_{6,j}\psi_{6,k}^*\right|}$, is the connectivity between crystalline particles, and $\langle C_6\rangle_{HEX}$ is the C_6 value for 2-dimensional hexagonal close packed particles defined as: $\langle C_6\rangle_{HEX} = N^{-1}6 \times (3S^2 + S)$, with $S = -\frac{1}{2} + \left[\frac{1}{3}(N-1) + \frac{1}{4}\right]^{1/2}$.

15475905, 0, Downloaded from https://aiche.onlinelibrary.wiley.com/doi/10.1002/aic.70662 by Oklahoma State University, Wiley Online Library on [25/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

For the detailed description on the physics of the experimental system, we refer the readers to Ref. [20].

4.2 | sMPC formulation

The sMPC problem at time step t is formulated as the maximization of the expected final value of the order parameter ψ_6 over the remaining control horizon:

$$\max_{a_{t:t+H-1}} \mathbb{E}[\psi_6(t+H)]$$

where $H = T - t$ is the shrinking prediction horizon, starting with $T = 10$, as we have a 1000s simulation process. The expectation is taken with respect to the stochastic evolution of the system model used inside the controller. At each control step, the optimization is solved in a receding-horizon fashion, that is, only the first action of the best sequence found is applied to the true system, and the problem is resolved at the next step. In our setting, the control horizon has the same length as the prediction horizon.

4.3 | Cross-entropy method for optimal control action calculation

Because the action space is discrete and the horizon is short, we employ the cross-entropy method (CEM), a population-based stochastic optimization algorithm, to solve the finite-horizon optimal control problem.[27,28] In CEM, the method iteratively refines a sampling distribution over action sequences to concentrate on high-performing candidates. In our implementation, an action sequence is a vector $\mathbf{a} = [a_t, a_{t+1}, \ldots, a_{t+H-1}]^{\mathsf{T}}$ of length H, where each element is drawn from a categorical distribution (as the inputs are discrete). The algorithm proceeds as the following:

1. *Initialization*: Set the initial policy $\boldsymbol{p}^{(0)}$ to a uniform distribution over the four actions for each step, that is, $\boldsymbol{p}_s^{(0)} = \left[\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right]$ for all steps $s = 0, \ldots, H-1$.
2. *Sampling*: At iteration k, generate $K = 100$ candidate action sequences by independently sampling from the current distribution $\boldsymbol{p}^{(k)}$.
3. *Evaluation*: For each candidate sequence $\mathbf{a}^{(j)}$, the expected terminal ψ_6 is estimated by performing $M = 10$ independent Monte-Carlo rollouts with the internal stochastic model. The score is the sample mean of the final ψ_6.
4. *Elite selection*: Select the $K_{\text{elite}} = 10$ sequences with the highest scores.
5. *Distribution update*: For each step s and action a, update the probability as

$$\boldsymbol{p}_s^{(k+1)}(a) = \frac{1}{K_{\text{elite}}} \sum_{j \in \mathcal{E}} \mathbf{1}\left\{a_s^{(j)} = a\right\} + \epsilon$$

where $\mathcal{E}$ is the set of elite indices and $\epsilon = 0.01$ is a small smoothing constant to prevent premature convergence.

6. *Iteration*: Repeat steps 2–5 for $n_{\text{iter}} = 5$ iterations.

The best sequence found across all iterations is retained, and its first action is applied to the system. This algorithm requires no gradient information and scales well with the small action space and horizon used here.

4.4 | Predictive model development

1. *Accurate Markov State Model (aMSM)*: In this study, we used the same 100-state Markov state models as in Ref. [21] as the underlying process to be controlled, that is, the true system, named as aMSM. Markov state model is a widely used stochastic modeling approach which simulates the system dynamics via transition probabilities. In this study, four discrete-state Markov state models were developed with one for each of the four input levels. There are in total 100 discrete states, achieved by discretizing the two order parameters into 10 evenly spaced intervals within its own physical bounds. Due to the discretization, the discretized order parameters are then bounded by $C_6 \in [0.30, 5.70]$, $\psi_6 \in [0.05, 0.95]$. To develop the Markov state model, more than 10,000 trials of 1000s Brownian Dynamics simulations initiated from different states were collected and used to calculate the transition probabilities from one discrete state to the others. Same as in previous studies,[20,21,26] we used a transition time of 100s in calculating the transition probability. The detailed description of the construction and application of Markov state models for colloidal self-assembly is given in Ref. [21]. The 6000-state Markov state model was developed with the same dataset used for the 100-state Markov model, with the only difference being the state discretization where C_6 was discretized into 120 evenly spaced intervals and ψ_6 into 50 evenly spaced intervals.
2. *Variational Recurrent Neural Network (VRNN)*: VRNN is an extension of variational autoencoder into a recurrent framework for modeling high-dimensional variabilities in sequential data, introduced by Chung et al.[29] In the VRNN, the stochastic prediction function $f_{\text{VRNN}}(x, a)$ samples a next state from the Gaussian distribution parameterized by the model, with the deterministic mean given by the prior-mean output and the variance decoded from the network. The output is confined to the upper and lower bounds of the two order parameters observed in the Markov state model with $C_6 \in [0.30, 5.70]$, $\psi_6 \in [0.05, 0.95]$.
   In this study, we developed a VRNN to approximate the transition dynamics as a conditional Gaussian distribution $p(x_{t+1}|x_t, a_t)$. Its internal structure consists of four sub-networks:
   - Two feature extractors $\phi_x : \mathbb{R}^2 \to \mathbb{R}^{64}$ and $\phi_u : \mathbb{R} \to \mathbb{R}^{64}$ (each with two ReLU layers) that embed the current state and the scalar control input.

15475905, 0, Downloaded from https://aiche.onlinelibrary.wiley.com/doi/10.1002/aic.70662 by Oklahoma State University, Wiley Online Library on [25/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

- A GRU cell with hidden dimension $d_h = 64$ that maintains a deterministic recurrent state h_t, giving the model memory over past observations.
- A prior network that maps the current state embedding, action embedding, and previous hidden state h_{t-1} to the parameters of a Gaussian latent distribution $p(z_t|x_t, a_t, h_{t-1})$ with diagonal covariance.
- A decoder that takes the sampled latent variable z_t (after passing through a feature extractor ϕ_z), together with the state and action embeddings and the hidden state, and outputs the mean μ_{t+1} and log-variance $\log\sigma^2_{t+1}$ of a Gaussian distribution over the next state. The mean is passed through a sigmoid function to ensure normalized outputs in $[0, 1]$, matching the min-max scaling of the data.

Stochasticity is introduced at two levels. First, the latent variable z_t is drawn from the prior Gaussian, injecting randomness into the transition. Second, the decoder outputs a variance that models the aleatoric uncertainty of the Brownian motion. A next-state sample is obtained via the reparameterization as:

$$x_{t+1} = \mu_{t+1} + \sigma_{t+1} \odot \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I)$$

This noise model is intrinsic to the VRNN and is preserved when the residual correction is added: the corrected model uses the same decoded variance, so the residual network only adjusts the mean prediction while the stochastic spread remains unchanged.

Different from the aMSM, to mimic the situation where limited data are obtainable for model development, instead of 10,000 Brownian Dynamics simulations as used for the aMSM development, we only collected 419 trials of 1000s BD simulations for the initial inaccurate VRNN model development.

3. *Residual Neural Network for Online Model Adaptation*: to improve the prediction accuracy of the VRNN model, we developed a multi-layer feed-forward residual correction network g_θ to learn the correction factors for the prediction. With the residual network, we then obtain the one-step prediction as: $\widehat{x}_{t+1} = \bar{f}_{\text{VRNN}}(x_t, a_t) + g_\theta(x_t, \mathbf{1}_{a_t}) + \varepsilon_t$, where $\bar{f}_{\text{VRNN}}$ is the deterministic next-state mean of the VRNN and $\varepsilon_t \sim N(0, \sum_t)$ is the intrinsic noise sampled from the VRNN's decoded covariance at time t. The correction network g_θ has two hidden layers of 64 ReLU units and is initialized with Xavier uniform weights and zero biases.[30]

   The correction network g_θ is randomly initialized before the first control rollout and updated every 20 controlled rollouts with predictions collected from the aMSM. A replay buffer stores the $20\,\text{rollouts} \times 10\,\text{steps} = 200$ most recent transition tuples $\{(x_t, a_t, x^{\text{true}}_{t+1})\}$ obtained from interaction with the aMSM. After every 20 rollouts, the residual network is updated for 5 epochs with the Adam optimizer (learning rate 10^{-4}, batch size 32) and used for the successive 20 rollouts. At each time step, discrepancy between the true next state and the VRNN predicted deterministic mean, $\delta = x^{\text{true}}_{t+1} - \bar{f}_{\text{VRNN}}(x_t, a_t)$, is calculated and used to formulate the loss function with respect to the network's prediction $\widehat{\delta} = g_\theta(x_t, a_t)$ and this target, as:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left[\left(\delta_{C_6,i} - \widehat{\delta}_{C_6,i}\right)^2 + \lambda\left(\delta_{\psi_6,i} - \widehat{\delta}_{\psi_6,i}\right)^2\right]$$

where $\lambda = \sigma^2_{C_6}/\sigma^2_{\psi_6}$ is the inverse variance ratio estimated from the buffer, balancing two order parameters, which differ substantially in scale, contribute equally to the optimization and the ψ_6 component, crucial for the control objective, is not overshadowed. The buffer is cleared after each update, so the network adapts to the latest data distribution.

AUTHOR CONTRIBUTIONS

Wei Sun: Methodology; writing – review and editing. **Henry Hanchey:** Investigation. **Syeda Simra Shoaib:** Methodology; data curation; formal analysis; investigation; visualization; writing – original draft; writing – review and editing. **Xun Tang:** Conceptualization; methodology; formal analysis; supervision; funding acquisition; visualization; writing – original draft; writing – review and editing. **Zheyu Jiang:** Conceptualization; writing – review and editing; methodology; supervision.

FUNDING INFORMATION

S.S., H.H., and X.T. are supported by NSF 2218077.

DATA AVAILABILITY STATEMENT

The core codes and data used to generate Figures 3–9 are provided at GitHub: https://github.com/xtang38/Shoaib_etal_sMPC_2026. Key data for Figures 3, 6–9 are also included in the Supplementary Information document.

ORCID

Syeda Simra Shoaib https://orcid.org/0009-0008-4827-1017
Henry Hanchey https://orcid.org/0009-0001-2420-1654
Zheyu Jiang https://orcid.org/0000-0003-4747-0539
Xun Tang https://orcid.org/0000-0003-0317-9176

REFERENCES

1. Guo Q, Arnoux C, Palmer R. Guided assembly of colloidal particles on patterned substrates. *Langmuir*. 2001;17(22):7150-7155.
2. Ristenpart WD, Aksay IA, Saville DA. Assembly of colloidal aggregates by electrohydrodynamic flow: kinetic experiments and scaling analysis. *Phys Rev E*. 2004;69(2):21405.
3. Yethiraj A. Tunable colloids: control of colloidal phase transitions with tunable interactions. *Soft Matter*. 2007;3(9):1099-1115.
4. Juarez JJ, Bevan MA. Interactions and microstructures in electric field mediated colloidal assembly. *J Chem Phys*. 2009;131(13):134704.
5. Tang X, Grover MA. Control of microparticle assembly. *Annu Rev Control Robot Auton Syst*. 2022;5:491-514.
6. Nodozi I, O'Leary J, Mesbah A, Halder A. A Physics-informed Deep Learning Approach for Minimum Effort Stochastic Control of Colloidal Self-Assembly. In 2023 American Control Conference (ACC). 2023.
7. Wu Z, Christofides PD, Wu W, et al. A tutorial review of machine learning-based model predictive control methods. *Rev Chem Eng*. 2025;41(4):359-400.
8. O'Leary J, Khare MM, Mesbah A. Novelty search for neuroevolutionary reinforcement learning of deceptive systems: an application to control of colloidal self-assembly. In 2023 American Control Conference (ACC). 2023.

15475905, 0, Downloaded from https://aiche.onlinelibrary.wiley.com/doi/10.1002/aic.70662 by Oklahoma State University, Wiley Online Library on [25/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

9. Tang X, Xue Y, Grover AM. Colloidal self-assembly with model predictive control. American Control Conference. 2013.
10. Lin M, Sun Z, Xia Y, Zhang J. Reinforcement learning-based model predictive control for discrete-time systems. *IEEE Trans Neural Networks Learn Syst*. 2024;35(3):3312-3324.
11. Mesbah A. Stochastic model predictive control: an overview and perspectives for future research. *IEEE Control Syst Mag*. 2016;36(6):30-44.
12. Sutton RS, Barto AG. *Reinforcement Learning: An Introduction*. MIT Press; 2018.
13. Zhang J, Yang J, Zhang Y, Bevan MA. Controlling colloidal crystals via morphing energy landscapes and reinforcement learning. *Sci Adv*. 2020;6(48):eabd6716.
14. Lieu UT, Yoshinaga N. Dynamic control of self-assembly of quasicrystalline structures through reinforcement learning. *Soft Matter*. 2025;21(3):514-525.
15. Dijkstra M, Luijten E. From predictive modelling to machine learning and reverse engineering of colloidal self-assembly. *Nat Mater*. 2021;20(6):762-773.
16. Whitelam S, Tamblyn I. Learning to grow: control of material self-assembly using evolutionary reinforcement learning. *Phys Rev E*. 2020;101(5):52604.
17. Lizano-Villalobos A, Ma F, Tang W, Sun W, Tang X. Machine learning-based optimal control for colloidal self-assembly. *AIChE J*. 2026;72(7):e70389.
18. Devarakonda VS, Sun W, Tang X, Tian Y. Recent advances in reinforcement learning for chemical process control. *Processes*. 2025;13:1791. doi:10.3390/pr13061791
19. McDonald MN, Zhu Q, Paxton WF, Peterson CK, Tree DR. Active control of equilibrium, near-equilibrium, and far-from-equilibrium colloidal systems. *Soft Matter*. 2023;19(9):1675-1694.
20. Tang X, Rupp B, Yang Y, Edwards TD, Grover MA, Bevan MA. Optimal feedback controlled assembly of perfect crystals. *ACS Nano*. 2016;10(7):6791-6798.
21. Tang X, Bevan MA, Grover MA. The construction and application of Markov state models for colloidal self-assembly process control. *Mol Syst Des Eng*. 2017;2(1):78-88.
22. Beltran-Villegas DJ, Bevan MA. Free energy landscapes for colloidal crystal assembly. *Soft Matter*. 2011;7(7):3280-3285.
23. Airaldi F, De Schutter B, Dabiri A. Reinforcement learning with model predictive control for highway ramp metering. *IEEE Trans Intell Transp Syst*. 2025;26(5):5988-6004.
24. Sivaranjani S, Shi Y, Atanasov N, et al. Control-oriented system identification: classical, learning, and physics-informed approaches. *Annu Rev Control*. 2026;62:101067.
25. Piga D, Forgione M, Formentin S, Bemporad A. Performance-oriented model learning for data-driven MPC design. *IEEE Control Syst Lett*. 2019;3(3):577-582.
26. Tang X, Zhang J, Bevan MA, Grover MA. A comparison of open-loop and closed-loop strategies in colloidal self-assembly. *J Process Control*. 2017;60:141-151.
27. Rubinstein R. The cross-entropy method for combinatorial and continuous optimization. *Methodol Comput Appl Probab*. 1999;1(2):127-190.
28. Botev ZI, Kroese DP, Rubinstein RY, L'Ecuyer P. Chapter 3: The cross-entropy method for optimization. In: Rao CR, Govindaraju V, eds. *Handbook of Statistics*. Elsevier; 2013:35-59.
29. Chung J, Kastner K, Dinh L, Goel K, Courville A, Bengio Y. *A Recurrent Latent Variable Model for Sequential Data. In NIPS*. Curran Associates, Inc; 2015.
30. Glorot X, Bengio Y. *Understanding the Difficulty of Training Deep Feedforward Neural Networks*. PMLR; 2010.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Shoaib SS, Hanchey H, Sun W, Jiang Z, Tang X. Online adaptive model-based stochastic control with limited data: A case study with a colloidal self-assembly system. *AIChE J*. 2026;e70662. doi:10.1002/aic.70662

15475905, 0, Downloaded from https://aiche.onlinelibrary.wiley.com/doi/10.1002/aic.70662 by Oklahoma State University, Wiley Online Library on [25/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License