MOX-Report No. 24/2025

Adjoint-based optimal control of jump-diffusion processes

Bartsch, J.; Borzi, A.; Ciaramella, G.; Reichle, J.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it https://mox.polimi.it

ADJOINT-BASED OPTIMAL CONTROL OF JUMP-DIFFUSION PROCESSES

Abstract. Stochastic differential equations (SDEs) using jump-diffusion processes describe many natural phenomena at the microscopic level. Since they are commonly used to model economic and financial evolutions, the calibration and optimal control of such processes are of interest to many communities and have been the subject of extensive research. In this work, we develop an optimization method working at the microscopic level. This allows us also to reduce computational time since we can parallelize the calculations and do not encounter the so-called curse of dimensionality that occurs when lifting the problem to its macroscopic counterpart using partial differential equations (PDEs). Using a discretize-then-optimize approach, we derive an adjoint process and an optimality system in the Lagrange framework. Then, we apply Monte Carlo methods to solve all the arising equations. We validate our optimization strategy by extensive numerical experiments. We also successfully test a optimization procedure that avoids storing the information of the forward equation.

Jan Bartsch[✉*1], Alfio Borzì[✉2], Gabriele Ciaramella [✉3], and Jan Reichle[✉4]

[1]Institut für Mathematik, Universität Würzburg,
Emil-Fischer-Str. 40, 97074 Würzburg, Germany

[2]Institut für Mathematik, Universität Würzburg,
Emil-Fischer-Str. 30, 97074 Würzburg, Germany

[3]MOX Lab, Dipartimento di Matematica, Politecnico di Milano,
Piazza Leonardo da Vinci 32, 20133 Milano, Italy

[3]Fachbereich für Mathematik und Statistik, Universität Konstanz,
Universitätsstraße 10, 78464 Konstanz, Germany

1. Introduction

Since many physical phenomena are inherently subject to uncertainties and noise their accurate description is often achieved using stochastic differential equations (SDEs). A very broad class of SDEs is the one governed by the so-called jump-diffusion processes. Their applications range from physics [3,18,28] over economics and finance [15,20,36] to medical imaging [22]. The jump-diffusion processes consist of three parts: 1) a deterministic drift, 2) a stochastic diffusion that is usually modeled as a Brownian motion, and 3) a stochastic jump usually realized using a compound Poisson process. Suppose that one wants to model and control the state $\boldsymbol{z}(t) \in \mathbb{R}^d$ of a physical system over a time interval $[0,T]$, $T>0$, then the corresponding SDE is given by

$$\mathrm{d}\boldsymbol{z}(t) = \boldsymbol{a}(\boldsymbol{z}(t), u(\boldsymbol{z},t), t)\ \mathrm{d}t + \boldsymbol{b}(\boldsymbol{z}(t), t)\ \mathrm{d}\boldsymbol{B}(t) + \boldsymbol{c}(\boldsymbol{z}(t^-), t)\ \mathrm{d}\boldsymbol{Y}(t) \qquad t \in (0,T]. \tag{1.1}$$

In (1.1), we denote by $\boldsymbol{B}$ a standard d-dimensional Brownian motion [35, Definition 1.1.13] and by $\boldsymbol{Y}$ a d-dimensional compound Poisson process [35, p. 10]. The function $u(\boldsymbol{z},t)$ is called the control. The equation (1.1) is completed with an initial condition $\mathring{\boldsymbol{z}}$ such that $\boldsymbol{z}(0) = \mathring{\boldsymbol{z}}$.

The optimal control of jump-diffusion processes is of interest to many communities and has therefore led to extensive research over many years; see, e.g., [14, 20, 21, 31, 32] and references therein. An optimal control problem is given by minimizing a certain objective $J(\boldsymbol{z},u)$ subject to the pair $(\boldsymbol{z},u)$ satisfying (1.1) and there are in principle two approaches to solving optimal control problems governed by (1.1). On the one hand, one can lift the problem to the level of

Date: May 7, 2025 (submitted).

2020 *Mathematics Subject Classification.* 93E03, 93D15, 65C05 .

Key words and phrases. Optimization of SDEs, Jump-diffusion process, First-order optimality conditions, Monte Carlo methods, Stochastic gradient methods.

The work was partially funded by Deutsche Forschungsgemeinschaft (DFG) within SFB 1432, Project-ID 425217212, and the Young Scholar Fund by of the University of Konstanz.

*Corresponding author: Jan Bartsch.

partial differential equations (PDEs). To this end, one defines a probability density function $f(z,t)$ that contains information on the probability of the state $\boldsymbol{z}$ being in a certain configuration z at a certain timestep t. The PDE that governs the evolution of f is given by the famous Fokker-Planck equation [21] that contains in the case of jump-diffusion processes also an integral part. In this setting, one can apply tools from PDE-constrained optimization [38]. In the case of $\boldsymbol{b} \equiv 0$ and $\boldsymbol{c} \equiv 0$, the first author considered optimal control problems for the corresponding partial differential equation, i.e. the Liouville equation. In [7,8] deterministic numerical methods were applied to solve the arising equations. In [5, 10, 11], the author considered a Monte-Carlo framework to solve the arising equations.

On the other hand, it is possible to work at the microscopic level and directly characterize the control using stochastic differential equations. Here, no probability density functions come into play and hence no PDEs must be solved.

In the PDE-based approach, the following difficulties occur. First, the complexity of the problem grows exponentially with the dimension of the state $\boldsymbol{z} \in \mathbb{R}^d$. This is the so-called *curse-of-dimensionality* from which all PDE-based approaches suffer. Additionally, for jump-diffusion processes the Fokker-Planck equation becomes an integro-differential equation, requiring more advanced solution techniques than in standard PDE optimization.

The problems of the PDE approach can be circumvented to some extent when remaining on the microscopic level. First, an increase in the state dimensions does not lead to an exponential growth of computational complexity. If one additionally supposes that each state component is independent of the others then it is possible to parallelize the computations with respect to the dimensions. Furthermore, the additional presence of the Poisson process does not lead to a significantly different structure of the solution method. In fact, the jump part can be approximated by a piecewise deterministic process for which the jump times have to be sampled according to a Poisson distribution.

The new contribution in this work is that we construct an optimization strategy working directly on the microscopic level. In particular, we do not have to assemble probability density functions that turn out to be a time-consuming bottleneck in hybrid (macroscopic-microscopic) methods [5, 11]. Furthermore, in contrast to [9], we do not consider only Brownian motion, but also jump processes determined by a Poisson process. The structure of our control is chosen in such a way that it realizes a feedback-like control. More specifically, we consider the control u as consisting of a fixed dependence on the state variable given by the smooth function $\boldsymbol{\Phi}(\boldsymbol{z})$ and a square integrable time-dependent function $\boldsymbol{\mu}(t)$. This time-dependent function is considered considered to be our control mechanism that we want to optimize. Hence, the control can be written given as

$$u(\boldsymbol{z},t) = \boldsymbol{\mu}(t)\boldsymbol{\Phi}(\boldsymbol{z}).$$

Furthermore, we use a tracking-type cost functional with a Gaussian-like exponential function. This is motivated by the fact, that we want to have bounded functions with bounded derivatives in the optimality system of our framework later on. Furthermore, this setting is then close to previous works [5, 10].

This work is organized as follows. In Section 2, we formulate and analyze the optimal control problem, introducing the system dynamics and the corresponding cost functional. Assumptions necessary for well-posedness are also discussed. In Section 3, we present the fully discretized problem. Furthermore, we analyze the convergence of discrete approximations to the continuous problem. In Section 4, we derive the optimality system using an adjoint-based approach, including the equations necessary for characterizing optimal solutions. In Section 5, we explain the numerical implementation, including the discretization scheme, the design of shape functions, the handling of jump processes, and the optimization procedure. In Section 6, we present numerical experiments to validate the proposed method. Specific test cases include centering particles, stabilization, and following a time-dependent trajectory with systems of coupled and uncoupled particles. Moreover, we successfully test here an idea that avoids the storing of the forward trajectories. A section of conclusion finishes this work.

2. Formulation of the optimal control problem

We consider a finite time horizon $[0,T]$ with $T>0$ and the a system of $N\in\mathbb{N}$ particles in the phase space $\mathbb{R}^{d_z}$ consisting of velocity and position space. Hence, we have the whole state space $\mathbb{R}^d$ with $d \coloneqq N\,d_z$. In the following, we consider a two-dimensional phase space, i.e. $d_z=2$. We appoint each particle with position and velocity, i.e. $\boldsymbol{z}_j=(\boldsymbol{x}_j,\boldsymbol{v}_j)$ for $j\in[N]$, where we use the notation $[N]\coloneqq\{1,\dots,N\}$.

Our goal is to design a control field capable of driving the mean of an initial configuration $\mathring{\boldsymbol{z}}$ of particles to a desired configuration or follow a desired trajectory $\bar{\boldsymbol{z}}_{\mathrm{d}}(t)$ on average. For this purpose, we define the structure of the deterministic coefficient as:

$$\boldsymbol{a}(\boldsymbol{z}(t),u(\boldsymbol{z},t),t)=\mathrm{e}\otimes\begin{pmatrix}0\\u(\boldsymbol{z},t)\end{pmatrix}+H\boldsymbol{z},\qquad H\in\mathbb{R}^{N\,d_z\times N\,d_z}, \tag{2.1}$$

where we define $\mathrm{e}=(1,\dots,1)\in\mathbb{R}^{N\,d_z}$ and denote by $\otimes$ the standard Kronecker product, i.e.

$$\mathrm{e}\otimes\begin{pmatrix}0\\u(\boldsymbol{z},t)\end{pmatrix}=\Big(0,u(\boldsymbol{z}_1,t),0,u(\boldsymbol{z}_2,t),\dots,0,u(\boldsymbol{z}_N,t)\Big)^\top\in\mathbb{R}^{N\,d_z}. \tag{2.2}$$

The symbol $^\top$ denotes the transpose. By the shape of the coefficient of the deterministic part given in (2.1), the control can be interpreted as force acting in the velocity component. The matrix H accounts for the dynamic of the particles in the uncontrolled case. For example, it can model Hook's law.

To measure the effectiveness of our control mechanism, we consider the cost functional

$$J(\boldsymbol{z},u)=\mathbb{E}\left[\int_0^T\frac{1}{N}\sum_{j=1}^N\mathcal{J}(\boldsymbol{z}_j,t)\,\mathrm{d}t+\frac{\alpha}{2}\|u\|^2_{L^2(\mathbb{R}^d\times[0,T])}\right], \tag{2.3}$$

where $\alpha>0$ is the control weight and states the relative importance of the cost of the control in the optimization problem. The symbol $\mathbb{E}$ denotes the expectation value with respect to the stochastic variable. On the functional $\mathcal{J}$ in (2.3), we impose the following

Assumption 2.1.
1) The functional $\mathcal{J}:\mathbb{R}^d\to\mathbb{R}$ is lower semicontinuous and bounded from below.
2) The functional $\mathcal{J}$ is continuously differentiable, i.e. $\mathcal{J}\in C^1(\mathbb{R}^d\times[0,T])$.

We consider the following form of the control for some $L\in\mathbb{N}$ that realizes a separation in phase space and time

$$u:\mathbb{R}^d\times[0,T]\to\mathbb{R},\qquad u(\boldsymbol{z}(t),t)=\sum_{\ell=1}^L\boldsymbol{\mu}_\ell(t)\,\boldsymbol{\phi}_\ell(\boldsymbol{z}(t)), \tag{2.4}$$

with $\boldsymbol{\mu}_\ell:[0,T]\to\mathbb{R}$ for $\ell\in[L]$. In this work, the shape functions $\boldsymbol{\phi}_\ell:\mathbb{R}^d\to\mathbb{R}$ in phase space are given and $\boldsymbol{\mu}_\ell$ are the optimization variables. We assume the following structure of $\boldsymbol{\phi}_\ell$ with given shape functions $\boldsymbol{\phi}^x_\ell$ and $\boldsymbol{\phi}^v_\ell$ in position and velocity, respectively,

$$\boldsymbol{\phi}_\ell(\boldsymbol{z}(t))\coloneqq\boldsymbol{\phi}^x_\ell(x(t))\,\boldsymbol{\phi}^v_\ell(v(t)).$$

We define $\boldsymbol{\phi}\coloneqq(\boldsymbol{\phi}_1,\dots,\boldsymbol{\phi}_L)$ and $\boldsymbol{\mu}=(\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_L)$. Then, we can write (2.4) as

$$u(\boldsymbol{z}(t))=\boldsymbol{\mu}(t)^\top\boldsymbol{\phi}(\boldsymbol{z}(t)). \tag{2.5}$$

We define

$$\Psi(\boldsymbol{z},\boldsymbol{\mu})\coloneqq\mathrm{e}\otimes\begin{pmatrix}0\\\boldsymbol{\mu}^\top\boldsymbol{\phi}(\boldsymbol{z})\end{pmatrix}. \tag{2.6}$$

With this, we can write the coefficient $\boldsymbol{a}$ in (2.1) as

$$\boldsymbol{a}(\boldsymbol{z},u,t)=\Psi(\boldsymbol{z},\boldsymbol{\mu})+H\boldsymbol{z}. \tag{2.7}$$

For the well-posedness of the state equation (1.1), we need the following assumptions (cf. [37] and [25]).

Assumption 2.2. The coefficients $\boldsymbol{a} : \mathbb{R}^d \times \mathbb{R} \times [0,T] \to \mathbb{R}^d$, $\boldsymbol{b} : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$, $\boldsymbol{c} : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ in (1.1) fulfill

1) (At most quadratic growth) There exists a constant $L > 0$, such that
$$|\boldsymbol{a}(\boldsymbol{z},u,t)|^2 + |\boldsymbol{b}(\boldsymbol{z},t)|^2 + |\boldsymbol{c}(\boldsymbol{z},t)|^2 \leq L(1+|\boldsymbol{z}|^2).$$
2) (Lipschitz continuity) For an arbitrary $R > 0$ there exists a constant $C_R > 0$ such that for all $|\boldsymbol{z}_1| \leq R$ and $|\boldsymbol{z}_2| \leq R$
$$|\boldsymbol{a}(\boldsymbol{z}_1,u,t) - \boldsymbol{a}(\boldsymbol{z}_2,u,t)|^2 + |\boldsymbol{b}(\boldsymbol{z}_1,t) - \boldsymbol{b}(\boldsymbol{z}_2,t)|^2 + |\boldsymbol{c}(\boldsymbol{z}_1,t) - \boldsymbol{c}(\boldsymbol{z}_2,t)|^2 \leq C_R|\boldsymbol{z}_1 - \boldsymbol{z}_2|^2$$
for all $t \in [0,T]$ and $u \in \mathbb{R}$.

We can now formulate our optimal control problem:

$$\min_{(\boldsymbol{z},\boldsymbol{\mu})} \mathbb{E}\left[\int_0^T \frac{1}{N}\sum_{j=1}^N \mathcal{J}(\boldsymbol{z}_j(t),t)\,\mathrm{d}t + \frac{\alpha}{2}\|\boldsymbol{\mu}(\cdot)^\top \boldsymbol{\phi}(\boldsymbol{z}_j(\cdot))\|^2_{L^2([0,T])}\right] =: j(\boldsymbol{z},\boldsymbol{\mu}) \tag{2.8a}$$

$$\text{s.t.} \begin{cases} \mathrm{d}\boldsymbol{z}(t) = \Big(\Psi(\boldsymbol{z}(t),\boldsymbol{\mu}(t)) + H\boldsymbol{z}(t)\Big)\,\mathrm{d}t + \boldsymbol{b}(\boldsymbol{z}(t),t)\,\mathrm{d}\boldsymbol{B}(t) \\ \qquad\qquad + \boldsymbol{c}(\boldsymbol{z}(t^-),t)\,\mathrm{d}\boldsymbol{Y}(t) \quad t \in (0,T], \\ \boldsymbol{z}(0) = \mathring{\boldsymbol{z}}. \end{cases} \tag{2.8b}$$

We state in the following theorem the existence and uniqueness of solutions to (1.1). Notice that $\boldsymbol{a}$ in the structure of (2.7) fulfills Assumption 2.2 for suitable $\boldsymbol{b}$, $\boldsymbol{c}$.

Theorem 2.3. *Let Assumption 2.2 hold. Then the equation (1.1) has a unique solution whose almost all sample functions are continuous from the right.*

For the proof, we refer to [37, Theorem 3.14]. For further information on the functions that are continuous from the right (so-called cadlag functions) $\mathcal{D}([0,T])$, we refer the reader to, e.g., [13]. Using Theorem 2.3, we can introduce the *control-to-state map* $\mathcal{S}$:

$$\mathcal{S} : L^2(0,T)^L \to \mathcal{D}([0,T]) \tag{2.9}$$

that associates to every $\boldsymbol{\mu} \in L^2(0,T)^L$ the corresponding solution of (2.8b).

Using the control-to-state map, we can introduce the *reduced objective function* $\widehat{j}(\boldsymbol{\mu})$ as

$$\widehat{j}(\boldsymbol{\mu}) \coloneqq j(\mathcal{S}(\boldsymbol{\mu}),\boldsymbol{\mu}). \tag{2.10}$$

We now discuss the existence of optimal controls in our setting. It is in general not possible to prove existence of optimal controls for general SDE in the strong sense. On the contrary, even for SDEs without jumps one has to consider so-called *relaxed controls* for which the probability space can not be fixed a-priori [16, 40]. In [1], the authors deal with the question of existence of optimal stochastic control for a quite general form of stochastic optimal controls problems with a state equation of the form (1.1). However, it is possible to characterize an optimal control by certain maximum principles. In Section 4, we use the Lagrange framework to characterize an optimal control function.

3. Formulation and analysis of the fully discrete problem

In this work, we apply the discretize-before-optimize approach. Hence, we show in the following how to discretize the continuous optimal control problem (2.8).

To approximate the expectation value $\mathbb{E}$ in (2.8), we choose $M \gg 1$ realizations for the Brownian motion and the Poisson process.

For each realization $m \in [M]$ and each particle $j \in [N]$, we calculate the N_j^m jump times $\{T_{j,k}^m\}_{k=1}^{N_j^m}$. Furthermore, we create a uniform grid in time with N_t^u subintervals of size $\Delta\tau = {}^T\!/_{N_t^u}$ as

$$[0,T] = \cup_{\kappa=0}^{N_t-1}[\tau^\kappa, \tau^{\kappa+1}], \qquad \tau^k = \kappa\,\Delta\tau. \tag{3.1}$$

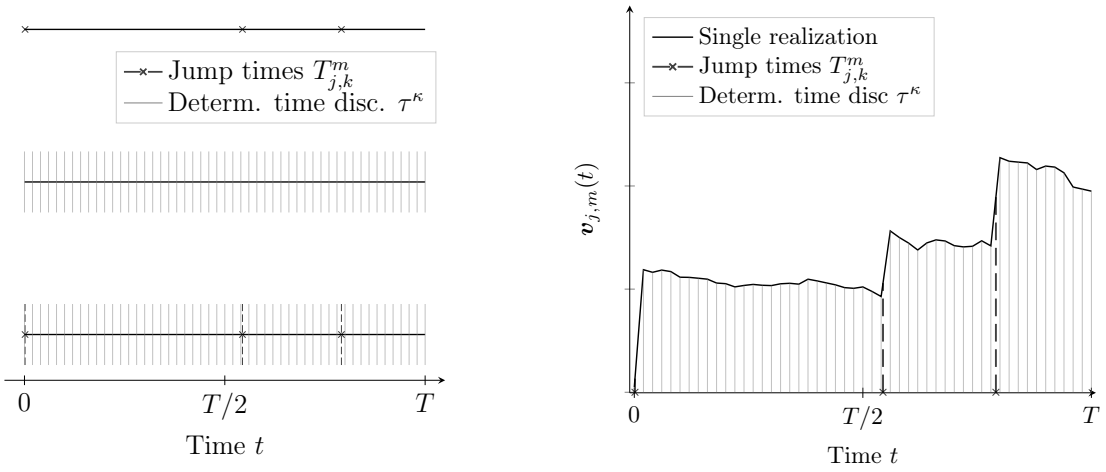


FIGURE 3.1. Different discretization of the time interval $[0,T]$ and exemplary trajectory. Gray: Deterministic splitting in N_t^u intervals; Black dashed: Stochastic splitting of the time interval into N_j jump times, different for every realization and particle. (a) Visualization of stochastic and deterministic parts of the time discretization. (b) Visualization of an exemplary SDE trajectory of a single particle in a single realization together with the corresponding discretization.

For each particle j and each realization m, we the consider a (possible) different time discretization generated by gridpoints which are given by the union of the N_t^u deterministic splitting and the N_j^m stochastic jump points. More specifically, the gridpoints are given by

$$\{T^m_{j,1},\dots,T^m_{j,N_j^m}\} \cup \{\tau^0,\dots,\tau^{N_t^u}\} =: \{t^0,\dots,t^{N_t^{j,m}}\}. \tag{3.2}$$

Now, we consider the discretization of the time interval into subintervals as

$$[0,T] = \cup_{\kappa=0}^{N_t^{j,m}-1}[t^\kappa,t^{\kappa+1}], \qquad \Delta t^\kappa := t^{\kappa+1}-t^\kappa, \tag{3.3}$$

Using the small time-intervals Δt^κ, we generate the samples of Brownian motion $\Delta \boldsymbol{B}^\kappa_{j,m} \sim \mathcal{N}(0,\Delta t^\kappa)$. In Figure 3.1(a), we visualize the different time discretizations in our work for a single particle and a single realization. In Figure 3.1(b), we plot an exemplary trajectory corresponding to the time discretization given in Figure 3.1(a).

However, for convenience of the implementation, we want to approximate the control to be constant in equidistant intervals of a given length. For this, we use the partition of the time interval $[0,T]$ into the $N_t^u > 1$, equally-spaced subintervals given in (3.1). Within one subinterval, we assume the control to be constant. In particular, we assume that we have the continuous in time interpolation of the control given by

$$\boldsymbol{\mu}^h(t) := \sum_{\kappa=0}^{N_t^u-1} \boldsymbol{\mu}^\kappa \chi_{[t^\kappa,t^{\kappa+1}]}(t), \qquad t\in[0,T]. \tag{3.4}$$

The fully discrete problem is then given by

(3.5a)

$$\min_{(\boldsymbol{z},\boldsymbol{\mu})} \frac{1}{M}\sum_{m=1}^{M}\frac{1}{N}\sum_{j=1}^{N}\sum_{k=1}^{N_j^m}\sum_{\kappa=N_t^{j,m,k}}^{N_t^{j,m,k+1}} \mathcal{J}(\boldsymbol{z}^\kappa_{j,m},t^\kappa) + \frac{\alpha}{2}\|\boldsymbol{\mu}(t^\kappa)^\top \boldsymbol{\phi}(\boldsymbol{z}^\kappa_{j,m})\|_2^2 \Delta t^k$$

(3.5b)

$$\text{s.t.} \quad \begin{cases} \boldsymbol{z}_{j,m}^{\kappa+1} = \boldsymbol{z}_{j,m}^{\kappa} + \Delta t\Big(\Psi(\boldsymbol{z}_{j,m}^{\kappa}, \boldsymbol{\mu}(t^\kappa)) + H\boldsymbol{z}_{j,m}^{\kappa}\Big) \\ \qquad\qquad\qquad\qquad + \boldsymbol{b}(\boldsymbol{z}_{j,m}^{\kappa}, t^\kappa)\Delta \boldsymbol{B}_{j,m}^{\kappa} \qquad \text{for } \kappa \in [N_t^{j,m,k-1}, N_t^{j,m,k} - 1] \\ \boldsymbol{z}_{j,m}^{N_t^{j,m,k}} = \boldsymbol{c}(\boldsymbol{z}_{j,m}^{N_t^{j,m,k}-1}) \\ \boldsymbol{z}_{j,m}^{0}(0) = \mathring{\boldsymbol{z}}_{j,m}. \end{cases}$$

We define the continuous in time representation of a numerical solution for the j-th particle in the m-th realization as a piecewise constant function. For this, we set $h = (M, N_t^u)$ and

$$\boldsymbol{z}_{j,m}^{h}(t) = \sum_{\kappa=0}^{N_t^{j,m}-1} \boldsymbol{z}_{j,m}^{\kappa} \chi_{[t^\kappa, t^{\kappa+1}]}(t), \qquad t \in [0,T]. \tag{3.6}$$

With this definition, we can evaluate $\boldsymbol{z}_{j,m}^{h}$ at arbitrary times, in particular at the time discretizations points of the other particles or realizations. We set $\boldsymbol{z}^h = \{\boldsymbol{z}_{j,m}^{h}\}_{m,j=1}^{M,N}$.

We define a discretized version of the objective taking into account the time discretization of the control

$$j^h(\boldsymbol{z}^h, \boldsymbol{\mu}^h) = \frac{1}{M}\sum_{m=1}^{M}\frac{\Delta t}{N}\sum_{j=1}^{N}\sum_{\kappa=1}^{N_t^u} \mathcal{J}(\boldsymbol{z}_{j,m}^{h}(t^\kappa), t^\kappa) + \frac{\alpha}{2}|\boldsymbol{\phi}(\boldsymbol{z}_{j,m}^{h}(t^\kappa))^\top \boldsymbol{\mu}^h(t^\kappa)|^2. \tag{3.7}$$

We have that the discrete solution $\boldsymbol{z}^h$ defined as the piecewise constant approximation (cf. (3.6)) converges to the true solution $\boldsymbol{z}$ in the following sense:

Theorem 3.1. *Let Assumption 2.2 hold. Then there exists for any $\boldsymbol{\mu} \in L^2(0,T)^N$ a constant $C > 0$ and $N_t^* \in \mathbb{N}$ such that for all $N_t \geq N_t^*$*

$$\mathbb{E}\sup_{t\in[0,T]}[|\boldsymbol{z}^h(t) - \boldsymbol{z}(t)|^2] \leq C\Delta t(1 + \mathbb{E}[|\boldsymbol{z}(0)|^2]). \tag{3.8}$$

For the proof, see [25, Theorem 2.4]. For further reading about discretization of jump-diffusion processes, we refer to [26, Chapter 16] as well as [25] and [35].

Using the result of Theorem 3.1, we can introduce the reduced functional

$$\widehat{j}^h(\boldsymbol{\mu}^h) = j(\mathcal{S}^h(\boldsymbol{\mu}^h), \boldsymbol{\mu}^h), \tag{3.9}$$

where $\mathcal{S}^h(\boldsymbol{\mu}^h)$ is the solution of (3.5b) using the piecewise constant $\boldsymbol{\mu}^h \in L(0,T)^L$ given $\{\mu^\kappa\}_{\kappa=0}^{N_t^u}$ (cf. (3.4)). The following theorem states that convergence of the value of the discrete functional to the value of the continuous one in the case of convergence of the discrete controls to a continuous one. Recall that we set $h = (M, N_t^u)$. When writing $h \to \infty$, we let both components tend to infinity at the same time.

Theorem 3.2. *Let Assumptions 2.1 and 2.2 hold. Assume that we have given a sequence $(\boldsymbol{\mu}^h)_h$ and a $\boldsymbol{\mu} \in L^2(0,T)^L$ with $\boldsymbol{\mu}^h \to \boldsymbol{\mu}$ for $h \to \infty$. Then we have*

$$|\widehat{j}^h(\boldsymbol{\mu}^h) - \widehat{j}(\boldsymbol{\mu})| \to 0, \qquad \text{for } h \to \infty. \tag{3.10}$$

Proof. Recall the definition of $\hat{j}$ in (2.10). We calculate

$$|\widehat{j}^h(\boldsymbol{\mu}^h) - \widehat{j}(\boldsymbol{\mu})| = |\widehat{j}^h(\boldsymbol{\mu}^h) - \widehat{j}^h(\boldsymbol{\mu})| + |\widehat{j}(\boldsymbol{\mu}) - \widehat{j}^h(\boldsymbol{\mu})|. \tag{3.11}$$

The first summand on the right-hand side in (3.11) converges to zero due to the continuity of $\widehat{j}^h$. The continuity of $\widehat{j}^h$ follows because it is a composition of continuous functions since $\mathcal{J}$ and the absolute value are continuous. The second summand converges to zero due to convergence of quadratures of integral in time [2, Chapter 5], and approximation properties of the discrete expectation value that is known as the *(Weak) Law of Large Numbers* [23, Theorem 8.2]. □

4. Optimality system

In this section, we derive the optimality system corresponding to (3.5). In the following, we consider only a single realization and omit to write the dependence of all quantities to m for the sake of better readability. An additional reason is that we only need one realization of each path of the N particles in each optimization iteration in our optimization procedure below.

We define the following Lagrange functional $\mathcal{L}$, introducing the adjoint processes $\boldsymbol{r}$ and $\boldsymbol{\lambda}$, as

$$\mathcal{L}(\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{r},\boldsymbol{\lambda}) \coloneqq \frac{1}{N}\sum_{j=1}^{N}\sum_{k=0}^{N_t^j-1}\mathcal{L}^{j,k}(\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{r},\boldsymbol{\lambda}), \tag{4.1}$$

with

$$\begin{aligned}\mathcal{L}^{j,k}(\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{r},\boldsymbol{\lambda}) \coloneqq &\sum_{\kappa=N_t^{j,k}}^{N_t^{j,k+1}} \mathcal{J}(\boldsymbol{z}_j^\kappa,t^\kappa)\Delta t^\kappa + \frac{\alpha}{2}|\boldsymbol{\phi}(\boldsymbol{z}_j^\kappa)^\top\boldsymbol{\mu}^\kappa|^2\Delta t^\kappa \\ &- \sum_{\kappa=N_t^{j,k}}^{N_t^{j,k+1}-1}\left(\boldsymbol{z}_j^{\kappa+1} - \boldsymbol{z}_j^\kappa\boldsymbol{a}(\boldsymbol{z}_j^\kappa,\boldsymbol{\mu}^\kappa,t^\kappa)\Delta t^\kappa + \boldsymbol{b}(\boldsymbol{z}_j^\kappa,t^\kappa)\Delta\boldsymbol{B}_j^\kappa\right)^\top\boldsymbol{r}_j^{\kappa+1} \\ &+ \left(\boldsymbol{z}_j^{N_t^{j,k}} - \boldsymbol{c}(\boldsymbol{z}_j^{N_t^{j,k}-1})\right)^\top\boldsymbol{\lambda}_j^{N_t^{j,k+1}}.\end{aligned} \tag{4.2}$$

The optimality system consists of the partial derivatives of the Lagrange functional with respect to its variables set to zero.

Calculating the gradient with respect to $\boldsymbol{z}$, we obtain that $\nabla_{\boldsymbol{z}_j^\kappa}\mathcal{L}\,\delta\boldsymbol{z}_j^\kappa = 0$ must hold for all admissible directions $\delta\boldsymbol{z}_j^\kappa$ with $j\in[N]$ and $\kappa = N_t^{j,k},\dots,N_t^{j,k+1}-1$. This is equivalent to

$$\boldsymbol{r}_j^\kappa - \boldsymbol{r}_j^{\kappa+1} - \nabla_{\boldsymbol{z}}\boldsymbol{a}(\boldsymbol{z}_j^\kappa,\boldsymbol{\mu}^\kappa,t^\kappa)^\top\boldsymbol{r}_j^{\kappa+1} - \nabla_{\boldsymbol{z}}\boldsymbol{b}(\boldsymbol{z}_j^\kappa,t^\kappa)^\top\boldsymbol{r}_j^{\kappa+1} + \nabla_{\boldsymbol{z}}j^h(\boldsymbol{z}_j^\kappa,\boldsymbol{\mu}) = 0. \tag{4.3}$$

We also have to deal with the initial conditions. We have that $\nabla_{\boldsymbol{z}_j^{N_t^j}}\mathcal{L}\,\delta\boldsymbol{z}_j^{N_t^j} = 0$ and $\nabla_{\boldsymbol{z}_j^{N_t^{j,k}-1}}\mathcal{L}\,\delta\boldsymbol{z}_j^{N_t^{j,k}-1} = 0$ are equivalent to

$$\delta\boldsymbol{z}_j^{N_t^{j,k}}\left(\boldsymbol{\lambda}_j^{N_t^{j,k+1}} - \boldsymbol{r}_j^{N_t^{j,k}}\right) = 0$$

and

$$\delta\boldsymbol{z}_j^{N_t^{j,k}-1}\left(\boldsymbol{r}_j^{N_t^{j,k}-1} - \nabla_{\boldsymbol{z}}\boldsymbol{c}(\boldsymbol{z}_j^{N_t^{j,k}-1})^\top\boldsymbol{\lambda}_j^{N_t^{j,k+1}}\right) = 0.$$

Using this calculations, and since $\delta\boldsymbol{z}_j^\kappa$ was arbitrary, we can conclude that

$$\boldsymbol{\lambda}_j^{N_t^{j,k+1}} = \boldsymbol{r}_j^{N_t^{j,k}}$$

and

$$\boldsymbol{r}_j^{N_t^{j,k}-1} = \nabla_{\boldsymbol{z}}\boldsymbol{c}(z_j^{N_t^{j,k}-1})^\top\boldsymbol{\lambda}_j^{N_t^{j,k+1}}.$$

Hence for all $j\in[N]$ and $k\in[N_j]$, the terminal condition for jumps in the adjoint model are given by

$$\boldsymbol{r}_j^{N_t^{j,k}-1} = \nabla_{\boldsymbol{z}}\boldsymbol{c}(z_j^{N_t^{j,k}-1})^\top\boldsymbol{r}_j^{N_t^{j,k}}. \tag{4.4}$$

Furthermore, we get from (4.3) the dynamics

$$\boldsymbol{r}_j^\kappa = \boldsymbol{r}_j^{\kappa+1} + \nabla_{\boldsymbol{z}}\boldsymbol{a}(\boldsymbol{z}_j^\kappa,\boldsymbol{\mu}^\kappa,t^\kappa)^\top\boldsymbol{r}_j^{\kappa+1} + \nabla_{\boldsymbol{z}}\boldsymbol{b}(\boldsymbol{z}_j^\kappa,t^\kappa)^\top\boldsymbol{r}_j^{\kappa+1} - \nabla_{\boldsymbol{z}}j^h(\boldsymbol{z}_j^\kappa,\boldsymbol{\mu}^\kappa). \tag{4.5}$$

Summarizing, we obtain the following adjoint model

$$\begin{aligned}\boldsymbol{r}_j^\kappa = \boldsymbol{r}_j^{\kappa+1} &+ \nabla_{\boldsymbol{z}}\boldsymbol{a}(\boldsymbol{z}_j^\kappa,\boldsymbol{\mu}^\kappa,t^\kappa)^\top\boldsymbol{r}_j^{\kappa+1} + \nabla_{\boldsymbol{z}}\boldsymbol{b}(\boldsymbol{z}_j^\kappa,t^\kappa)^\top\boldsymbol{r}_j^{\kappa+1} \\ &- \nabla_{\boldsymbol{z}}j^h(\boldsymbol{z}_j^\kappa) \qquad \text{for } \kappa\in[N_t^{j,k},N_t^{j,k+1}-1],\end{aligned} \tag{4.6a}$$

$$\boldsymbol{r}_j^{N_t^{j,k}-1} = \nabla_{\boldsymbol{z}} \boldsymbol{c}(z_j^{N_t^{j,k}-1})^\top \boldsymbol{r}_j^{N_t^{j,k}}. \tag{4.6b}$$

Notice that (4.6) evolves backwards in time. There are some important differences between the discretize-before-optimize (DBO) approach that we exploit in this work and the optimize-before-discretize approach (OBD). First of all, we do not have to solve Forward-Backward-SDE (FBSDE) since we do not have to take care of the filtration that evolves forward in time; see, e.g., [33, Chapter 7, Definition 3.1] for the definition of solutions to Backward SDEs. Since we discretized before we optimized, we cosider for the adjoint equation the same Brownian motion increments and jump-times as for the original system; see also [9]. Furthermore, the adjoint jump kernel coincides with one of the model and hence the adjoint jump frequency also coincides with the model jump frequency. This is different in the OBD case [5].

Analogously to Theorem 3.1, we can formulate a result for the adjoint equation. For this let $\boldsymbol{r}^h$ define the continuous representation of a numerical solution for the j-th particle in the m-th realization as the piecewise constant function as $\boldsymbol{r}^h = \{\boldsymbol{r}^h_{j,m}\}_{j,m=1}^{M,N}$ with

$$\boldsymbol{r}^h_{j,m}(t) = \sum_{\kappa=0}^{N_t^{j,m}-1} \boldsymbol{r}^\kappa_{j,m} \chi_{[t^\kappa, t^{\kappa+1}]}(t) \qquad t \in [0,T]. \tag{4.7}$$

Theorem 4.1. *Let Assumptions 2.1 and 2.2 hold. Let $\boldsymbol{r}^h$ be the piecewise constant approximation given in (4.7). Then there exists for any $\boldsymbol{\mu} \in L^2(0,T)^N$ a constant $C > 0$ and $N_t^* \in \mathbb{N}$ such that for all $N_t \geq N_t^*$*

$$\mathbb{E} \sup_{t\in[0,T]} [|\boldsymbol{r}^h(t) - \boldsymbol{r}(t)|^2] \leq C\Delta t(1 + \mathbb{E}[|\boldsymbol{r}(0)|^2]). \tag{4.8}$$

The next step is to derive the reduced gradient $\nabla_{\boldsymbol{\mu}} \widehat{j}^h$ of $\widehat{j}^h$ defined in (3.9). For this, we take the directional derivative of $\mathcal{L}$ with respect to $\boldsymbol{\mu}^\kappa$ for $\kappa \in [N_t^u]$ and calculate for the arbitrary direction $\delta\boldsymbol{\mu}^\kappa$

$$\nabla_{\boldsymbol{\mu}^\kappa} \mathcal{L}\, \delta\boldsymbol{\mu}^\kappa = \frac{\Delta t}{N} \sum_{j=1}^N \delta\boldsymbol{\mu}^\kappa \left\{ \alpha\, \boldsymbol{\phi}(\boldsymbol{z}_j^h(t^\kappa)) \boldsymbol{\phi}(\boldsymbol{z}_j^h(t^\kappa))^\top \boldsymbol{\mu}^\kappa - \Psi(\boldsymbol{z}_j^h(t^\kappa))^\top \boldsymbol{r}_j^h(t^\kappa) \right\}. \tag{4.9}$$

Hence, the reduced gradient is given by

$$\nabla_{\boldsymbol{\mu}^\kappa} \widehat{j}^h(\boldsymbol{\mu}^\kappa) = \frac{\Delta t}{N} \sum_{j=1}^N \alpha\, \boldsymbol{\phi}(\boldsymbol{z}_j^h(t^\kappa)) \boldsymbol{\phi}(\boldsymbol{z}_j^h(t^\kappa))^\top \boldsymbol{\mu}^\kappa - \Psi(\boldsymbol{z}_j^h(t^\kappa))^\top \boldsymbol{r}_j^h(t^\kappa), \tag{4.10}$$

where $\boldsymbol{z}$ and $\boldsymbol{r}$ are solutions to (3.5b) and (4.6), respectively, corresponding to $\boldsymbol{\mu}$. Hence, we have that at optimality it must hold for all $\kappa \in [N_t^u]$ that

$$\nabla_{\boldsymbol{\mu}^\kappa} \widehat{j}^h(\boldsymbol{\mu}^\kappa) = 0 \tag{4.11}$$

which is equivalent to

$$\sum_{j=1}^N \alpha\, \boldsymbol{\phi}(\boldsymbol{z}_j^h(t^\kappa)) \boldsymbol{\phi}(\boldsymbol{z}_j^h(t^\kappa))^\top \boldsymbol{\mu}^\kappa = \sum_{j=1}^N \Psi(\boldsymbol{z}_j^h(t^\kappa))^\top \boldsymbol{r}_j^h(t^\kappa) \qquad \forall \kappa \in [N_t^u]. \tag{4.12}$$

Notice that (4.12) is a high-dimensional nonlinear system with respect to $\boldsymbol{\mu}^\kappa$.

5. Numerical implementation and optimization procedure

In this section, we explain our strategy to solve the model and adjoint problem and the procedure to solve the full optimization problem (3.5). We use Monte Carlo strategies to solve the arising equations. In principle, these methods are meshless. However, to define our shape functions $\boldsymbol{\phi}$, we shall introduce a computational domain and a discrete phase space. We want to point out, that we do not specify boundary conditions, as this computational domain is needed only to define the number and the centers of the shape functions.

In Section 5.1, we introduce the computational domain and define the shape functions in Section 5.2. In Section 5.3, we explain our procedure to handle the jump processes before we

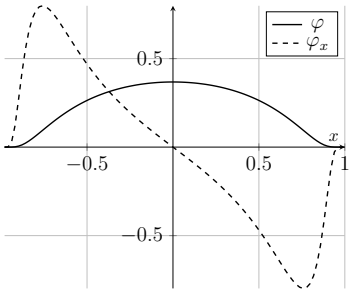


FIGURE 5.1. A radial basis function φ and its derivative φ_x.

discuss our strategy to solve the model and adjoint equation in Section 5.4 and Section 5.5, respectively. In Section 5.6, we discuss our optimization strategy.

5.1. **Discretization.** We consider our two-dimensional computational domain as $\Omega_x \times \Omega_v \coloneqq (-x_{\max}, x_{\max}) \times (-v_{\max}, v_{\max})$, with given $x_{\max}, v_{\max} > 0$. We use a standard finite-volume discretization in phase space and define the discrete phase space $\Omega_{\Delta x,\Delta v} \coloneqq \Omega_{\Delta x} \times \Omega_{\Delta v}$ as follows; see also [4, 5]. We choose a partition of Ω_x and Ω_v of equally-spaced, non-overlapping square cells with side length $\Delta v = 2v_{\max}/N_v$ where $N_v \geq 2$ and $\Delta x = 2x_{\max}/N_x$ with $N_x \geq 2$, respectively. On this partition, we consider a cell-centered representation as follows:

$$\Omega_{\Delta x} \coloneqq \left\{ x^i \in \Omega_x \;\middle|\; i \in [N_x] \right\}, \qquad x^i \coloneqq (i - 1/2)\Delta x - x_{\max}$$

and

$$\Omega_{\Delta v} \coloneqq \left\{ v^l \in \Omega_v \;\middle|\; l \in [N_v] \right\}, \qquad v^l \coloneqq (l - 1/2)\,\Delta v - v_{\max}.$$

Notice that the computational phase space is centered at $(0, 0)$ by this choice.

In order to discretize the control $\boldsymbol{\mu}_\ell$, $\ell \in [L]$, with $L = N_x\, N_v$, we use the time discretization given in (3.1). We identify the control and the solution $\boldsymbol{z}$ as the constant approximation between the grid points corresponding to this discretization of $[0, T]$; cf. (3.6) and (3.4).

5.2. **Shape functions.** For our implementation of the shape functions φ_ℓ, $\ell \in [L]$, we consider *radial-basis functions* (RBF) [17]. More specifically, we choose a certain form of a bump function with center x_c and shape parameter $\varepsilon_\varphi > 0$ given by

$$\varphi(x, x_c, \varepsilon_\varphi) \coloneqq \begin{cases} \exp\left(-\frac{1}{1-(\varepsilon_\varphi|x-x_c|)^2}\right) & \text{for } |x - x_c| < \frac{1}{\varepsilon_\varphi}, \\ 0 & \text{else.} \end{cases} \tag{5.1}$$

The functions ϕ^x_ℓ, ϕ^v_ℓ, $\ell \in [L]$, are defined using φ by

$$\phi^x_\ell : \mathbb{R} \to \mathbb{R}, \qquad \phi^x_\ell(x) = \varphi(x, x^\ell, \varepsilon_\varphi), \tag{5.2a}$$

$$\phi^v_\ell : \mathbb{R} \to \mathbb{R}, \qquad \phi^v_\ell(v) = \varphi(v, v^\ell, \varepsilon_\varphi). \tag{5.2b}$$

There are results for the convergence of the approximation of L^2 functions using RBF with compact support present in the literature; see, e.g., [17, Theorem 6.7].

The derivative of the shape functions is given by:

$$\varphi_x(x, x_c, \varepsilon_\varphi) = \begin{cases} \exp\left(-\frac{1}{1-(\varepsilon_\varphi|x-x_c|)^2}\right) \frac{-2\varepsilon_\varphi|x-x_c|}{(1-(\varepsilon_\varphi|x-x_c|)^2)^2} & \text{for } |x - x_c| < \frac{1}{\varepsilon_\varphi}, \\ 0 & \text{else.} \end{cases}$$

In Figure 5.1, we visualize the RBF function together with its derivative. We point out, that both of them are defined on whole $\mathbb{R}$ and are compactly supported. Notice that we choose the same shape function for position and velocity for the sake of simplicity. However, it is in principle possible to choose different functions for each component.

5.3. **Jump process.** For our implementation, we consider the jump part $\boldsymbol{c}$ to be related to the Keilson-Storer kernel introduced in [29]. For a pre-jump velocity $v \in \mathbb{R}$ and post-jump velocity $w \in \mathbb{R}$, it is given by

$$A(v,w) = \Gamma\sqrt{\frac{\beta}{\pi}}\exp(-\beta|w-\gamma v|^2), \tag{5.3}$$

with parameters $\gamma \in [-1,1]$ and $\beta > 0$.

With this kernel, many physical phenomena can be modeled including Brownian motion, telegraphic noise, and the famous Bhatnagar–Gross–Krook (BGK) collision operator [12]. It was also used in previous work [5, 6] by the authors. For this work, we define for a position $x \in \mathbb{R}$ and velocity $v \in \mathbb{R}$

$$\boldsymbol{c}(x,v) = (x, \gamma v + \varsigma_\beta)^\top, \tag{5.4}$$

where ς_β are precalculated random numbers obeying $\mathcal{N}(0,(2\beta)^{-1})$. We denote by $\mathcal{N}(\xi,\varpi)$ the Gaussian distribution with mean ξ and variance ϖ. The parameters γ and β are given in the definition of the Keilson-Storer kernel (5.3).

Furthermore, we need to sample the jump times $T_{j,k}$. For this, we use the fact that the mean free time is given as the reciprocate of jump frequency of the Keilson-Storer kernel defined as

$$\sigma = \int_{\mathbb{R}} A(v,w)\,\mathrm{d}v = \sqrt{\frac{\beta}{\pi}}. \tag{5.5}$$

In order to determine the time instances at which a particle undergoes a velocity transition due to jump using σ, one can follow the procedure described in, e.g., [11, 27].

If σ is the jump frequency, then $\sigma\,\mathrm{d}t$ is the probability that a particle has a jump during the time $\mathrm{d}t$. Now, assuming that a particle has a jump at time t, the probability that it will be subject to another jump at time $t+\delta t$ dt is computed according to a Poisson distribution given by

$$\exp\left(-\int_t^{t+\delta t}\sigma\,\mathrm{d}t'\right) = \exp\left(-\delta t\,\sigma\right). \tag{5.6}$$

The formula in (5.6) is the probability distribution of the distance between two events of a Poisson process. Following a standard approach [27] and using a uniformly distributed random number $\nu \in (0,1)$, one obtains the following rule

$$\delta t = -\sigma^{-1}\log(\nu). \tag{5.7}$$

Notice that we do not have an adjoint jump frequency, like we have it in [11].

5.4. **Model equation.** For our numerical examples in Section 6, we consider in the following structure the coefficient functions

$$\boldsymbol{a}(\boldsymbol{z},u,t) = \mathrm{e}\otimes\begin{pmatrix}0\\ u(\boldsymbol{z},t)\end{pmatrix} + H\boldsymbol{z} + h(\boldsymbol{z},t), \qquad H \coloneqq \begin{pmatrix}0 & 1\\ -\eta & 0\end{pmatrix}\otimes I_N, \tag{5.8a}$$

where h is a given smooth nonlinear function such that $\boldsymbol{a}$ fulfills Assumption 2.2, $I_N \in \mathbb{R}^{N\times N}$ is the N-dimensional identity matrix and

$$\boldsymbol{b}(\boldsymbol{z},t) = \begin{pmatrix}\mathrm{b}_1 & 0\\ 0 & \mathrm{b}_2\end{pmatrix}\otimes I_N, \qquad \mathrm{b}_1,\mathrm{b}_2 > 0. \tag{5.8b}$$

During the free flight, we consider Euler's method for integration using stepsize Δt that is smaller than the difference of two jump times. The Euler-Maruyama method is given by

$$\boldsymbol{x}_j^{\kappa+1} = \boldsymbol{x}_j^\kappa + \boldsymbol{v}_j^\kappa\,\Delta t + \mathrm{b}_1\,\Delta\boldsymbol{B}_j^{x,\kappa} \qquad \boldsymbol{x}_j^{N_t^{j,k}} = \boldsymbol{x}_j^{N_t^{j,k}-1}, \tag{5.9a}$$

$$\boldsymbol{v}_j^{\kappa+1} = \boldsymbol{v}_j^\kappa + \left(u(\boldsymbol{z}_j^\kappa,t^\kappa) - \eta\,\boldsymbol{x}_j^\kappa\right)\Delta t + \mathrm{b}_2\,\Delta\boldsymbol{B}_j^{v,\kappa} \qquad \boldsymbol{v}_j^{N_t^{j,k}} = \gamma\boldsymbol{v}_j^{N_t^{j,k}-1} + \varsigma_\beta, \tag{5.9b}$$

where $\Delta \boldsymbol{B}_j^{x,\kappa}$ and $\Delta \boldsymbol{B}_j^{v,\kappa}$ for $j \in [N]$ and $\kappa \in [N_t]$ are precomputed Gaussian samples for position and velocity with zero mean and standard deviation $\sqrt{\Delta t^\kappa}$; see [34]. The whole method for solving the forward model is summarized in Algorithms 5.1 and 5.2.

5.5. **Adjoint equation.** Recall that the adjoint equation (4.6) evolves backward. We introduce the adjoint position $\boldsymbol{q}$ and adjoint velocity $\boldsymbol{p}$ for the adjoint phase space coordinate $\boldsymbol{r} = (\boldsymbol{q}, \boldsymbol{p})$. The adjoint equations for the coefficients fixed in (5.8) are given by

$$\boldsymbol{q}_j^\kappa = \boldsymbol{q}_j^{\kappa+1} + \Delta t\Big(\big(u_x(\boldsymbol{z}_j^\kappa) - \eta\big)\, \boldsymbol{p}_j^{\kappa+1} - \nabla_{\boldsymbol{x}} j^h(\boldsymbol{z}_j^\kappa, u(\boldsymbol{z}_j^\kappa))\Big), \tag{5.10a}$$

$$\boldsymbol{p}_j^\kappa = \boldsymbol{p}_j^{\kappa+1} + \Delta t\Big(\boldsymbol{q}_j^{\kappa+1} + u_v(\boldsymbol{z}_j^\kappa)\boldsymbol{p}_j^{\kappa+1} - \nabla_{\boldsymbol{v}} j^h(\boldsymbol{z}_j^\kappa, u(\boldsymbol{z}_j^\kappa))\Big), \tag{5.10b}$$

with terminal conditions

$$\boldsymbol{q}_{j,T_{N_{j,k}}} = -\nabla_{\boldsymbol{x}} j^h\left(\boldsymbol{z}_{j,T_{N_{j,k}}}, u(\boldsymbol{z}_{j,T_{N_{j,k}}})\right), \qquad \boldsymbol{p}_{j,T_{N_{j,k}}} = -\nabla_{\boldsymbol{v}} j^h\left(\boldsymbol{z}_{j,T_{N_{j,k}}}, u(\boldsymbol{z}_{j,T_{N_{j,k}}})\right). \tag{5.11}$$

The derivatives of u are given by

$$u_x(x,v,t) = \sum \boldsymbol{\mu}_\ell \partial_x \boldsymbol{\phi}_\ell^x(x)\boldsymbol{\phi}_\ell^v(v), \qquad u_v(x,v,t) = \sum \mu_\ell \boldsymbol{\phi}_\ell^x(x)\partial_v \boldsymbol{\phi}_\ell^v(v).$$

The derivatives of the functional are given by

$$\nabla_{\boldsymbol{x}} j^h(\boldsymbol{z}_j, u) = \nabla_{\boldsymbol{x}} \mathcal{J}(\boldsymbol{z}_j) + \alpha\, u(\boldsymbol{z}_j) u_x(\boldsymbol{z}_j), \qquad \nabla_{\boldsymbol{v}} j^h(\boldsymbol{z}_j, u) = \nabla_v \mathcal{J}(\boldsymbol{z}_j) + \alpha\, u(\boldsymbol{z}_j) u_v(\boldsymbol{z}_j). \tag{5.12}$$

5.6. **Optimization procedure.** In this section, we describe our numerical strategy to solve the discrete optimization problem (3.5). We exploit a stochastic gradient method (SGD) with linesearch [39]. To assemble the gradient in every iteration, we need to integrate the equations of motion for the model (5.9) and adjoint equation (5.10). In Algorithm 5.1, we summarize this procedure taking into account the jumps.

Algorithm 5.1 Integration of the equations of motion

Require: Initial position $\boldsymbol{x}_j^0$ and velocity $\boldsymbol{v}_j^0$, number of jump times N_j of the j-th particle
Set $t^\kappa \leftarrow 0$, $t \leftarrow 0$, $\bar{t} \leftarrow 0$, $\kappa \leftarrow 0$
while $t < T$ **do**
  if $t^{\kappa+1} > T^{j,k}$ **then** ▷ *jump*
    $\bar{t} \leftarrow T^{j,k} - t$
    Update position and velocity using Euler's method according to (5.9) (respectively (5.10)) using $\bar{t}$ as stepsize in time
    Calculate the initial condition of the next time interval using $\boldsymbol{c}(\boldsymbol{v}_{j,k})$ in (5.4) (respectively $\nabla \boldsymbol{c}$)
    $t \leftarrow T^{j,k}$
  else ▷ *no jump*
    $\bar{t} \leftarrow t^{k+1} - t$
    Update position and velocity using Euler's method according to (5.9) (respectively (5.10)) using Δt as stepsize in time $\bar{t}$.
    Set $t \leftarrow t^{\kappa+1}$
  end if
  $\kappa \leftarrow \kappa + 1$
end while

In Algorithm 5.2, we summarize the procedure to solve the state equation. Notice that since the particles evolve independently from each other, one can heavily parallelize this method. Furthermore, notice that we generate jump times for each particle a priori. This jump times are then also used in the solver for the adjoint equation.

Algorithm 5.2 Model solver

Require: Parameters $\gamma, \beta > 0$, initial particles $\boldsymbol{z}^0$
for each particle $j = 1$ **to** N **do** ▷ *particles evolve independently from each other*
  $T_{j,0} \leftarrow 0$, $\tilde{k} \leftarrow 0$
  while $T_{j,\tilde{k}} < T$ **do** ▷ *Sample jump times*

Set $\tilde{k} \leftarrow \tilde{k} + 1$
Sample new jump time $\delta t_{j,\tilde{k}}$ using (5.6) and set $T_{j,\tilde{k}} \leftarrow T_{j,\tilde{k}-1} + \delta t_{j,\tilde{k}}$
end while ▷ *Create* $\cup_{k=1}^{N_j}(T_{j.k-1}, T_{j,k}] = (0, T]$
Set $N_j \leftarrow \tilde{k} - 1$
Sample ς_β^k, $k \in [N_j]$ obeying $\mathcal{N}(0, (2\beta)^{-1})$ for each jump time $T_{j,k}$
Integrate the equations of motion of the particles using Algorithm 5.1
end for
return

The method to calculate the reduced gradient (4.10) is presented in Algorithm 5.3. This procedure is executed iteratively in the complete optimization algorithm summarized in Algorithm 5.4. In each iteration, we sample a new set of initial conditions for the particles obeying the initial distribution $\boldsymbol{z}$. Then we solve the model equation where we generate the jump times and Brownian motion increments. These are then also used in the solver for the adjoint equation. The resulting (discrete) trajectories for all particles are taken in to account to assemble the reduced gradient for this iteration.

Algorithm 5.3 Calculate the gradient

Require: current control iterate $\boldsymbol{\mu}^n = (\boldsymbol{\mu}^n, \dots, \boldsymbol{\mu}_{N_t}^n)$, initial distribution $\mathring{\boldsymbol{z}} = (\mathring{\boldsymbol{x}}, \mathring{\boldsymbol{v}})$, desired trajectory $\bar{\boldsymbol{z}}_{\mathrm{d}}(t) = (\bar{x}_{\mathrm{d}}(t), \bar{v}_{\mathrm{d}}(t))$
Sample initial condition of N particles obeying $\mathring{\boldsymbol{z}}$
Solve the model system (3.5b) using Algorithm 5.2
Calculate terminal condition for adjoint equation according to (5.11)
Solve adjoint system (4.6) using the timesteps generated in Algorithm 5.2 and the integration analog to the one in Algorithm 5.1
Calculate the reduced gradient $\nabla_{\boldsymbol{\mu}}\widehat{j}^h$ according to (4.10).
return $\nabla_{\boldsymbol{\mu}}\widehat{j}^h$

The resulting gradient is then used as a stepdirection as usual in the stochastic gradient descent; see, e.g, [24]. Additionally, we consider the stochastic version of the classical Armijo-linesearch to determine a stepsize $\zeta_n > 0$

$$\widehat{j}^h\left(\boldsymbol{\mu}^n - \zeta_n \nabla\widehat{j}^h(\boldsymbol{\mu}_n)\right) \leq \widehat{j}^h(\boldsymbol{\mu}_n) - c\,\zeta_n \|\nabla\widehat{j}^h(\boldsymbol{\mu}_n)\|^2, \tag{5.13}$$

with a hyperparameter $c > 0$; see [19, 39]. The algorithm terminates if the difference between two subsequent control iterates is closer than a given tolerance or if the maximum iteration depth $n_{\max}$ is reached. The whole strategy is summarized in Algorithm 5.4.

Algorithm 5.4 Gradient descent scheme

Require: desired state $\bar{\boldsymbol{z}}_{\mathrm{d}}(t) = (\bar{x}_{\mathrm{d}}(t), \bar{v}_{\mathrm{d}}(t))$, initial guess of the control $\boldsymbol{\mu}^0 = (\boldsymbol{\mu}^0, \dots, \boldsymbol{\mu}_{N_t}^0)$, initial distribution $\mathring{\boldsymbol{z}} = (\mathring{\boldsymbol{x}}, \mathring{\boldsymbol{v}})$, tolerance tol > 0, maximum iteration depth $n_{\max}$
Set $n \leftarrow 0$ and initialize $\mathrm{E} \gg tol$
while $\mathrm{E} >$ tol **and** $n < n_{\max}$ **do**
  Compute reduced gradient $\boldsymbol{h}^n$ using Algorithm 5.3 ▷ *Sampling of random data included here*
  Determine the step-size ζ_n along $\boldsymbol{h}^n$ satisfying (5.13)
  Update control: $\boldsymbol{\mu}^{n+1} \leftarrow \boldsymbol{\mu}^n + \zeta_n\,\boldsymbol{h}^n$
  $\mathrm{E} \leftarrow \|\boldsymbol{\mu}^{n+1} - \boldsymbol{\mu}^n\|_2$
  Set $n \leftarrow n + 1$
end while
return $\boldsymbol{U}^\ell$

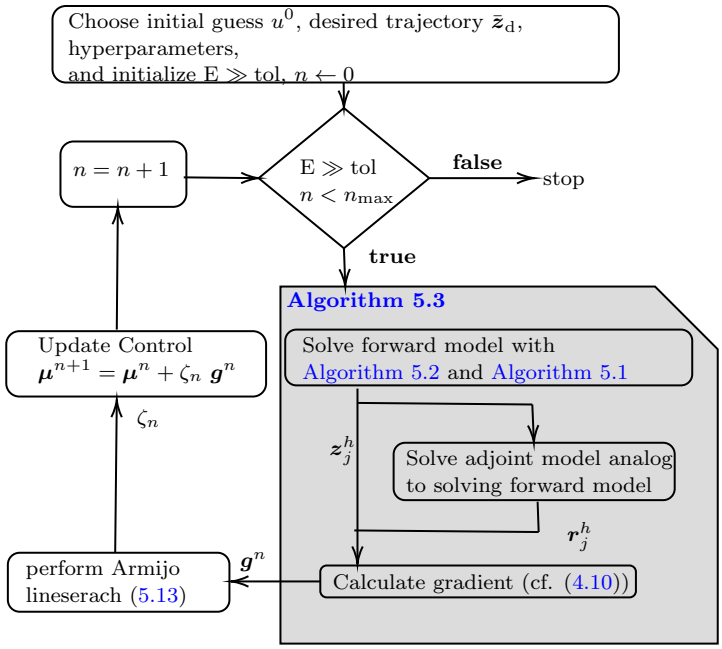


FIGURE 5.2. UML flowchart of the optimization procedure Algorithm 5.4.

6. Numerical experiments

In this section, we show the results several numerical experiments in order to validate our optimization procedure. For all of them, we fix the numerical computational phase space having the bounds $x_\mathrm{max} = 2$ and $v_\mathrm{max} = 2$. For the basis functions φ in (5.1), we take the grid-points of $\Omega_{\Delta x} \times \Omega_{\Delta v}$ as centers of the radial basis functions. Hence, we have $L = N_x\, N_v$ basis functions. We iterate through all center points (x^i, v^l) while assembling u in (2.4). For the final time, we choose $T = 1$.

As tracking cost, we consider for $\boldsymbol{z} = (x, v) \in \mathbb{R}^2$

$$\mathcal{J}(\boldsymbol{z}) = -\exp\left(-\frac{1}{2\sigma_\mathcal{J}^2}(|x - \bar{x}_\mathrm{d}(t)|^2 + |v - \bar{v}_\mathrm{d}(t)|^2)\right) \tag{6.1}$$

with a desired (mean) phase space trajectory $\bar{\boldsymbol{z}}_\mathrm{d}(t) = (\bar{x}_\mathrm{d}(t), \bar{v}_\mathrm{d}(t))$. Notice that $\mathcal{J}$ is bounded from above and below and smooth and hence satisfies Assumption 2.1. Its partial derivatives are then given by

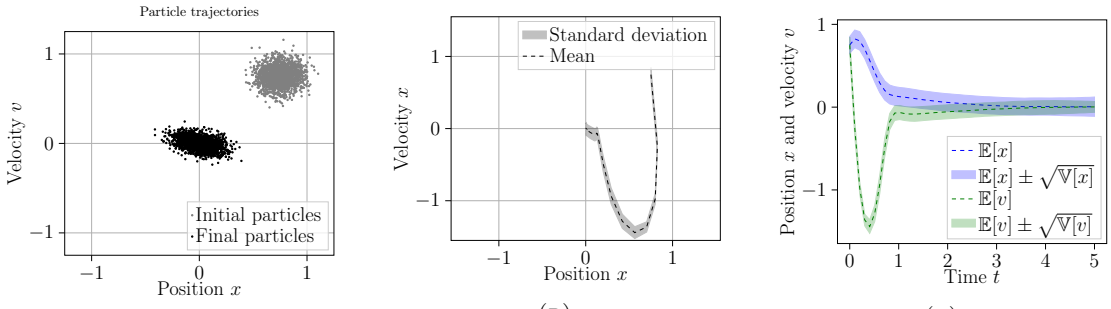
$$\nabla_x \mathcal{J}(\boldsymbol{z}) = \sigma_\mathcal{J}^{-2}(x - \bar{x}_\mathrm{d}) \exp\left(-\frac{1}{2\sigma_\mathcal{J}^2}(|x - \bar{x}_\mathrm{d}|^2 + |v - \bar{v}_\mathrm{d}|^2)\right),$$
$$\nabla_v \mathcal{J}(\boldsymbol{z}) = \sigma_\mathcal{J}^{-2}(v - \bar{v}_\mathrm{d}) \exp\left(-\frac{1}{2\sigma_\mathcal{J}^2}(|x - \bar{x}_\mathrm{d}|^2 + |v - \bar{v}_\mathrm{d}|^2)\right).$$

In all experiments, we use $\beta = 10$ and $\gamma = 0.9$. Furthermore, for the temporal discretization, we choose $N_t = 50$, $\Delta t = 0.1$ which leads to $T = 5$. Furthermore, we consider $N_x = N_v = 10$, $\Delta x = \Delta v = 0.4$ and use $\varepsilon_\varphi = 0.5$ and consider $N = 2 \cdot 10^3$ particles.

We start with a test case where we want to center all particles in the middle of the computational domain given a Gaussian distribution as initial configuration and continue with the case where we have a uniform distribution in a subdomain of the computational domain and also want to have the particles centered in the middle (cf. Section 6.1). Then, in Section 6.2, we average the control found in this case in time and apply it to a random initial configuration that also has particles outside the previous subdomain. The next test case is to follow a (non-smooth) trajectory in time (cf. Section 6.3). After this, we test our framework with a system

of particles including coupling in Section 6.4. Finally, we present in Section 6.5 a strategy that avoids storing the forward trajectories and test it for the setting of the previous case.

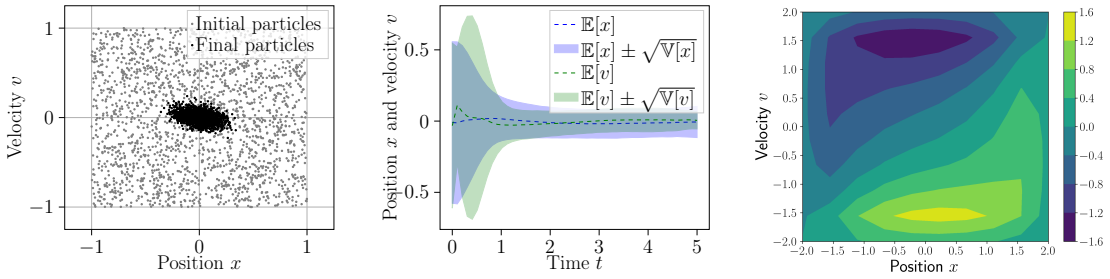
6.1. **Centering of the particles.** For our first example, we choose $\bar{\boldsymbol{z}}_{\mathrm{d}}(t) = (0,0)$. Moreover, we take $\eta = 1$ in (5.8). We start first with a normal distributed initial condition $\mathring{z} \sim \mathcal{N}(0.75, 0.75)^\top, 0.01\, I_2)$. In Figure 6.1, we plot the result of this numerical experiment. In Figure 6.1(a), we plot the initial configuration (gray) and the one obtained in the final timestep using our optimized control. The corresponding evolution of the mean and variance in phase space is plotted in Figure 6.1(b). In the uncontrolled case, all particles will (up to the effects of jumps and diffusion) remain on their initial orbit around the center. In the control case, the orbit is immediately decreased in position, i.e. the particles move directly close to the center. On the other side in velocity, there is a small overshoot before converging to the center. From Figure 6.1(c), we can obtain that the final configuration is kept stable. In the uncontrolled case, one expects sinusoidal behaviour for the mean in position and velocity due to Hook's law given by H (cf. (5.8)).



(A) (B) (C)

FIGURE 6.1. Results of numerical experiments trying to center the particles starting from a normal distribution. (a) Initial (gray) and final (black) configuration; (b) Mean and standard deviation in phase space; (c) Mean and standard deviation in position and velocity over time.

Now, we start with a uniform distribution in a subset of our computation domain, i.e. uniform distributed in $[-1,1]^2$. Our goal is again to center the particles in the middle, i.e. $\bar{\boldsymbol{z}}_{\mathrm{d}}(t) = (0,0)$, while again considering $\eta = 1$ in (5.8). In Figure 6.2, we show the results of this test case. In Figure 6.2(a) the initial and final configuration is visualized. From Figure 6.2(b) it is evident that also here the final configuration is stable. In Figure 6.2(c), we plot the values of $\mu(t)$ for $t = 0$ to give an impression of how the control might look like. In Section 6.2, we will use the results of this test case the generate a stabilization control for general random initial data.



(A) (B) (C)

FIGURE 6.2. Results of numerical experiments trying to center the particles starting from a uniform distribution. (a) Initial and final configuration; (b) Evolution of mean and variance; (c) Control $\boldsymbol{\mu}(t)$ at final time $t = T$.

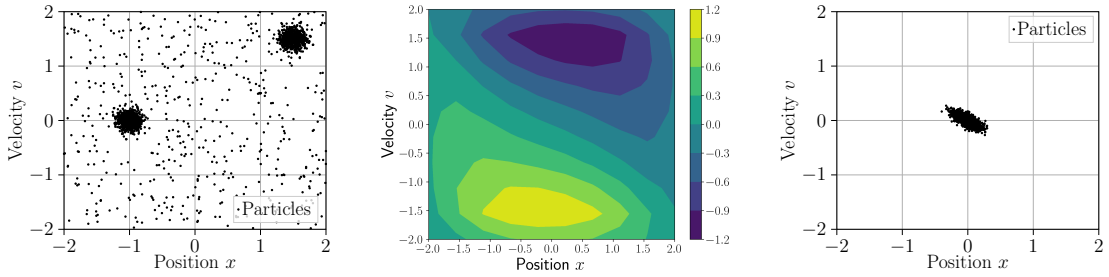
6.2. **Stabilization.** In this test case, we aim at stabilizing the system at $\bar{\boldsymbol{z}}_{\mathrm{d}}(t) = (0,0)$ with a random initial condition, in particular with an initial condition different from the one that was used to calculate the control. More specifically, we show that by taking the average in time of the control obtained using our procedure above, we might be able to apply it to any input data and observe a stabilizing behavior. Since our control depends by construction on position and velocity and hence on the state of the system, we can interpret it as a feedback-like control. The idea of taking the time average of an open-loop control to obtain a closed-loop one can already be found in [6]. However, in the current work, there are now distribution functions involved. We consider still $\eta = 1$ in (5.8). This means, in particular, that the desired configuration $\bar{\boldsymbol{z}}_{\mathrm{d}}$ is not reached in the uncontrolled case since the particles stay on their initial orbit (up to perturbations by jump and diffusion).

We define the time-averaged control given the optimized control from Section 6.1 where we started with a uniform initial distribution

$$\bar{u}(x,v) = \sum_{\ell=1}^{L} \bar{\boldsymbol{\mu}}_\ell \boldsymbol{\phi}_\ell^x(x) \boldsymbol{\phi}_\ell^v(v) \qquad \text{with } \bar{\boldsymbol{\mu}}_\ell \coloneqq \frac{1}{T}\int_0^T \boldsymbol{\mu}_\ell(t)\,\mathrm{d}t. \tag{6.2}$$

Notice that this control $\bar{u}$ is now independent of time. Hence it can also be used in an infinite time interval. We want to point out that there is no additional optimization procedure executed.

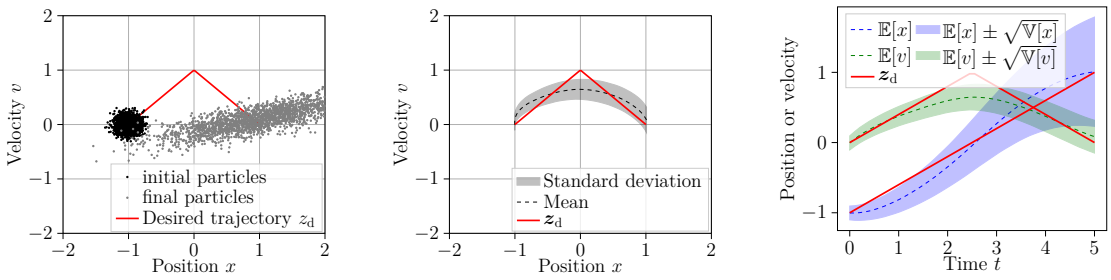
In Figure 6.3 we show the results of this test case. We start with a superposition of a bimodal and uniform configuration as initial condition that is shown in Figure 6.3(a). After applying our control $\bar{u}$ that is visualized in Figure 6.3(b), we end up with a final configuration shown in Figure 6.3(c). We observe that our average control is capable of collecting the particles in the center and keeping them there. Notice that we initialized the particles outside of the domain where we generated uniform particles (cf. Figure 6.2). Nevertheless, since our control u is defined in whole $\mathbb{R}^2$, it is possible the stabilize all particles in the center $(0,0)$.



(A) (B) (C)

FIGURE 6.3. Results of numerical experiments of stabilization with time-averaged control. (a) Initial configuration; (b) Contourplot of averaged $\bar{\boldsymbol{\mu}}$; (c) Final configuration obtained with averaged control $\bar{\boldsymbol{\mu}}$ without additional optimization process.

6.3. **Following a time-dependent trajectory.** In the last test case, we search for an optimal control such that the particles follow a desired non-smooth time-dependent trajectory $\bar{\boldsymbol{z}}_{\mathrm{d}}(t)$ for $t \in [0,T]$. We choose $\bar{\boldsymbol{z}}_{\mathrm{d}}(t) = (-\frac{x_{\max}}{2} + \frac{x_{\max}}{T}t, -|\frac{v_{\max}}{2T}t| + \frac{v_{\max}}{2})$. Furthermore, we choose $\eta = 0$ for this experiment and $\varepsilon_\varphi = 0.1$. In particular, the particles will not feel a harmonic oscillator force, as this is not sensible for following a time-dependent trajectory that is not a circular motion. We present the results of this test case in Figure 6.4, where in all plots the desired trajectory is plotted in red. In Figure 6.4(a), the initial configuration and final configurations are shown. In Figure 6.4(b), the desired trajectory and the resulting mean and variance of particles applying the optimal control are depicted in phase space, whereas in Figure 6.4(c) the mean and variance in position and velocity are plotted over time. We see that the mean of the particles follows closely the prescribed desired trajectory. Since we cannot control the variance with out control mechanism, it grows as expected.



(A) (B) (C)

FIGURE 6.4. Results of numerical experiments with desired time-dependent trajectory (red). (a) Initial and final configuration; (b) Desired trajectory (red) and evolution of mean and standard deviation in phase space; (c) Desired trajectory (red) and evolution of mean and standard deviation over time.

6.4. **Interacting particles.** As a next example, we now consider a system of particles that interact with each other. More in detail, we consider a system of N particles, in which all nearest neighbours are coupled. For three particles with positions x_i, velocities v_i, v_2, $i = 1, 2, 3$, and with positive parameters η and ω, we have the system:

$$\dot{x}_i = v_i \qquad \dot{v}_i = u(x_i, v_i, t) - \eta x_i - \omega \left(2x_i - \sum_{i=1, i \neq j}^{3} x_i \right) \tag{6.3}$$

With this system of coupled oscillators, we can model in particular interacting phonons [30].

As a test case, we initialize N particles equidistantly distributed on an ellipse in phase space described by

$$\left(\frac{x}{A_x} \right)^2 + \left(\frac{v}{A_v} \right)^2 = 1, \tag{6.4}$$

with given $A_x, A_v > 0$. The phonons should stay on this ellipse. However, since we have diffusion and jumps in the process, the phonons leave this orbit. With our control, we aim to keep the orbit as stable as possible. For this reason, we implement also another functional $\mathcal{J}_c$ for this test case with coupled particles:

$$\mathcal{J}_c(z, t) = -\exp\left(-\frac{1}{2\sigma_{\mathcal{J}}^2} \left(\left| \frac{x^2}{A_x^2} + \frac{v^2}{A_v^2} - 1 \right|^2 \right) \right). \tag{6.5}$$

This implements our desire to keep the particles on the ellipse described in (6.4).

In the following figures, we present the results of this test case. We set $A_x = 1.5$, $A_v = 1.7071067811865475$, and $\eta = 1$, $\omega = 0.5$.

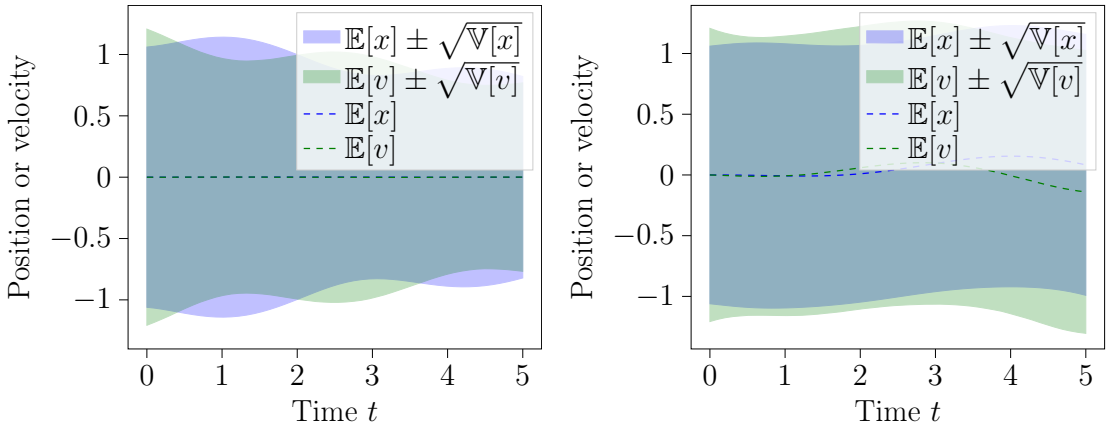


FIGURE 6.5. Results of experiment with coupled particles. (a): Uncontrolled case; (b): Applying optimized control mechanism.

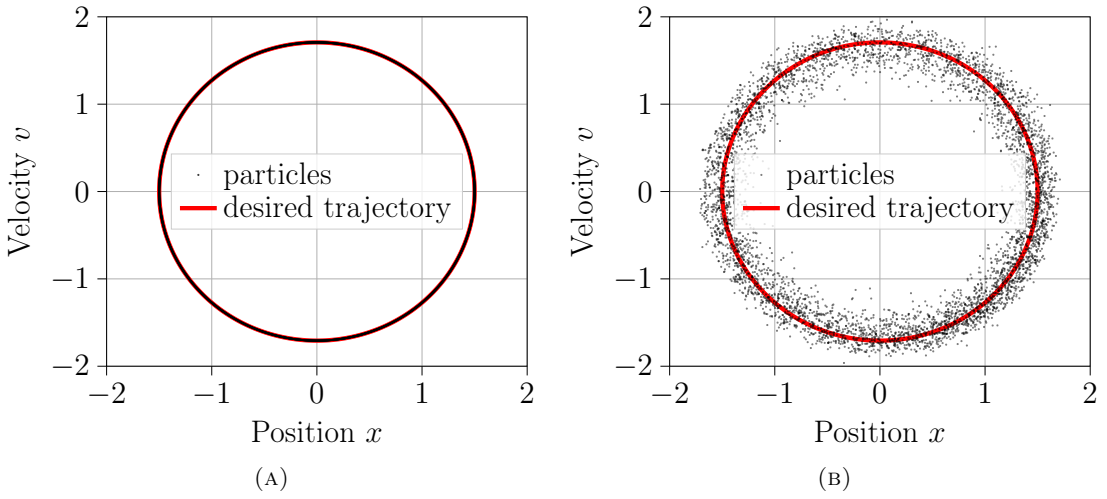
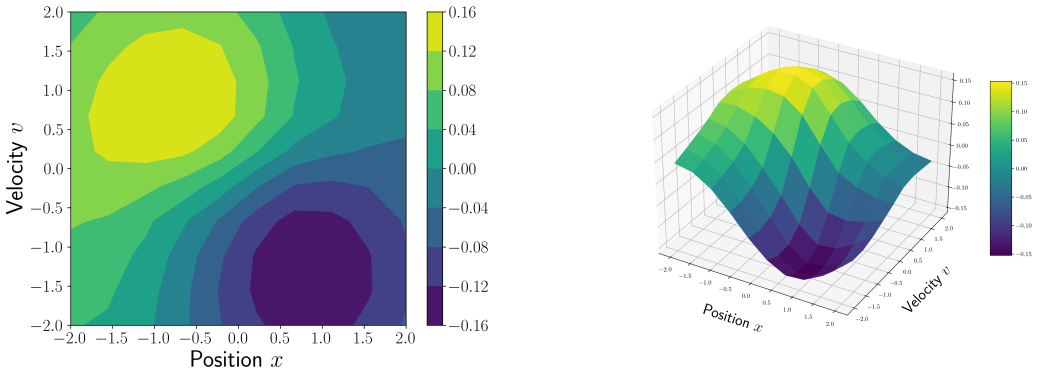


FIGURE 6.6. Behaviour of particles; (a): Initial particles on the desired ellipse; (b): final particles distributed on average on the ellipse.

6.5. **Adjoint equation independent of realization of forward trajectories.** In this section, we consider the same setting as in Section 6.3 but we do not take the solution of the forward model into account for the calculation of the solution to the adjoint model. The main reason for this is that it is then possible to avoid the storing of the forward trajectories and hence save memory. More specifically, it is possible to save the memory for $N \cdot N_t$ datapoints for the N particles with N_t timesteps. Instead of using $\boldsymbol{z}_j^h$, we use the (known) desired trajectory $\bar{\boldsymbol{z}}_\mathrm{d}(t)$ and sample for each particle and realization a position and velocity according to $\mathcal{N}(\bar{\boldsymbol{z}}_\mathrm{d}(t), \varsigma(t))$ with $\varsigma(t) : [0,T] \to \mathbb{R}^+$ being the two-dimensional variance in phase space that mirrors the behavior of the variance of the forward model, i.e it growths linearly over time.

Notice, that in the generation of the gradient (4.10), we consider the forward trajectories for everything else expect the calculation of the adjoint variable $\boldsymbol{r}$ since this can be done during the calculation of the forward model.

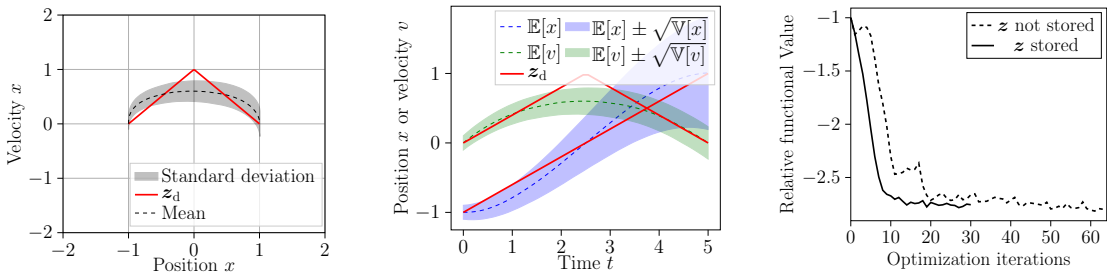
In Figure 6.8, we present the results of this test case. We observer a similar behavior as in the test case of Section 6.3 which stresses the effectiveness of our idea in this test case in order to save memory capacity. When we compare the behavior of the convergence of the (relative) functional,



(A) (B)

FIGURE 6.7. Averaged (in time) optimal control; (a) Contour plot; (b) Surface plot

we see that more steps are needed in order to reach a certain functional value compared the the case in which we take the exact forward trajectories (cf. Figure 6.8(c)).



(A) (B) (C)

FIGURE 6.8. Results of numerical experiments with desired time-dependent trajectory (red). (a) Evolution of mean and standard deviation in phase space; (b) Evolution of mean and standard deviation for position and velocity in time; (c) Comparison of convergence of relative functional $\hat{j}^h(u^\ell)/\hat{j}^h(u^0)$ over optimization iterations.

7. Conclusion

In this work, we developed and analyzed an adjoint-based optimization method for the control of systems governed by jump-diffusion processes, staying within the microscopic framework of stochastic differential equations. Our approach avoids the curse of dimensionality associated with macroscopic PDE-based methods and allows for highly parallelizable computations.

We derived the optimality system using a discretize-then-optimize approach and validated the theoretical results with Monte Carlo methods tailored to the microscopic context. Numerical experiments demonstrated the effectiveness of the proposed method in diverse scenarios, including centering particles, stabilization, and following time-dependent trajectories and using interacting and non-interacting particles. Moreover, we also presented a memory saving strategy that avoids storing the forward trajectories. These results underscore the robustness and versatility of our approach in solving complex control problems in stochastic settings.

Future work could explore extensions to more complex jump processes including the control also in the diffusive and jump part. The integration of machine learning for enhanced control strategies also presents an interesting direction for further research.

Acknowledgments

This work was partially funded by the Deutsche Forschungsgemeinschaft within SFB 1432, Project-ID 425217212. The authors acknowledge the Young Starting Fund of the University of Konstanz for partially funding J.R.

Declaration of generative AI and AI-assisted technologies in the writing process. During the preparation of this work the authors used ChatGPT in order to improve the style of the writing. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Author contribution. J.B. procured funding and led the project. A.B. initiated the project and gave crucial ideas to improve the work. G.C. initiated the project and worked together with J.B. to derive the theoretical results. J.R. implemented the code and conducted numerical experiments under the supervision of J.B. All authors edited and revised the manuscript.

Conflict of interest. The authors have no relevant financial or non-financial interests to disclose.

References

[1] Ahmed, N. U., and Charalambous, C. D. Stochastic minimum principle for partially observed systems subject to continuous and jump diffusion processes and driven by relaxed controls. *SIAM J. Control Optim. 51*, 4 (2013), 3235–3257.

[2] Atkinson, K. *An introduction to numerical analysis*. John wiley & sons, 1991.

[3] Bao, F. Lévy backward SDE filter for jump diffusion processes and its applications in material sciences. *Commun. Comput. Phys. 27*, 2 (2019).

[4] Barth, T., Herbin, R., and Ohlberger, M. Finite volume methods: foundation and analysis. *Encyclopedia of Computational Mechanics Second Edition* (2018), 1–60.

[5] Bartsch, J., and Borzì, A. MOCOKI: A Monte Carlo approach for optimal control in the force of a linear kinetic model. *Comput. Phys. Commun. 266* (2021), 108030.

[6] Bartsch, J., and Borzì, A. On the stabilization of a kinetic model by feedback-like control fields in a Monte Carlo framework. *Kinet. Relat. Models* (2024).

[7] Bartsch, J., Borzì, A., Fanelli, F., and Roy, S. A theoretical investigation of Brockett's ensemble optimal control problems. *Calc. Var. Partial Differential Equations 58*, 5 (2019), Paper No. 162, 34.

[8] Bartsch, J., Borzì, A., Fanelli, F., and Roy, S. A numerical investigation of Brockett's ensemble optimal control problems. *Numer. Math. 149*, 1 (2021), 1–42.

[9] Bartsch, J., Denk, R., and Volkwein, S. Adjoint-based calibration of nonlinear stochastic differential equations. *Appl. Math. Optim.* (2024).

[10] Bartsch, J., Knopf, P., Scheurer, S., and Weber, J. Controlling a Vlasov–Poisson Plasma by a Particle-in-Cell Method Based on a Monte Carlo Framework. *SIAM J. Control Optim. 62*, 4 (2024), 1977–2011.

[11] Bartsch, J., Nastasi, G., and Borzì, A. Optimal control of the Keilson-Storer master equation in a Monte Carlo framework. *J. Comput. Theor. Transp. 50*, 5 (2021), 454–482.

[12] Bhatnagar, P. L., Gross, E. P., and Krook, M. A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems. *Phys. Rev. 94* (May 1954), 511–525.

[13] Billingsley, P. *Convergence of probability measures*, second ed. Wiley Series in Probability and Statistics: Probability and Statistics. John Wiley & Sons, Inc., New York, 1999. A Wiley-Interscience Publication.

[14] Bismut, J.-M. An introductory approach to duality in optimal stochastic control. *SIAM Rev. 20*, 1 (1978), 62–78.

[15] Bruti-Liberati, N., and Platen, E. Approximation of jump diffusions in finance and economics. *Comput. Econ. 29* (2007), 283–312.

[16] Buckdahn, R., Labed, B., Rainer, C., and Tamer, L. Existence of an optimal control for stochastic control systems with nonlinear cost functional. *Stochastics 82*, 3 (2010), 241–256.

[17] Buhmann, M. D. *Radial basis functions: theory and implementations*, vol. 12 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 2003.

[18] del Castillo-Negrete, D., Carreras, B., and Lynch, V. Fractional diffusion in plasma turbulence. *Phys. Plasmas 11*, 8 (2004), 3854–3864.

[19] Dvinskikh, D., Ogaltsov, A., Gasnikov, A., Dvurechensky, P., and Spokoiny, V. On the line-search gradient methods for stochastic optimization. *IFAC-PapersOnLine 53*, 2 (2020), 1715–1720.
[20] Framstad, N. C., Øksendal, B., and Sulem, A. Sufficient stochastic maximum principle for the optimal control of jump diffusions and applications to finance. *J. Optim. Theory Appl. 121*, 1 (2004), 77–98.
[21] Gaviraghi, B., Schindele, A., Annunziato, M., and Borzì, A. On optimal sparse-control problems governed by jump-diffusion processes. *Appl. Math. 7*, 16 (2016), 1978–2004.
[22] Grenander, U., and Miller, M. I. Representations of knowledge in complex systems. *J. R. Stat. Soc. Ser. B Methodol. 56*, 4 (1994), 549–581.
[23] Grinstead, C. M., and Snell, J. L. *Introduction to probability*. American Mathematical Soc., 1997.
[24] Higham, C. F., and Higham, D. J. Deep learning: an introduction for applied mathematicians. *SIAM Rev. 61*, 4 (2019), 860–891.
[25] Higham, D. J., and Kloeden, P. E. Convergence and stability of implicit methods for jump-diffusion systems. *Int. J. Numer. Anal. Model. 3*, 2 (2006), 125–140.
[26] Higham, D. J., and Kloeden, P. E. *An introduction to the numerical simulation of stochastic differential equations*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2021.
[27] Jacoboni, C., and Reggiani, L. The Monte Carlo method for the solution of charge transport in semiconductors with applications to covalent materials. *Rev. Mod. Phys. 55*, 3 (1983), 645–705.
[28] Jain, R., Ginot, F., and Krüger, M. Micro-rheology of a particle in a nonlinear bath: Stochastic Prandtl-Tomlinson model. *Phys. Fluids 33*, 10 (Oct. 2021), 103101.
[29] Keilson, J., and Storer, J. E. On Brownian motion, Boltzmann's equation, and the Fokker-Planck equation. *Quart. Appl. Math. 10* (1952), 243–253.
[30] Koniakhin, S. V., Utesov, O. I., and Yashenkin, A. G. Coupled-oscillator model for hybridized optical phonon modes in contacting nanosize particles and quantum dot molecules. *Phys. Rev. Res. 5*, 1 (Feb. 2023), 013153.
[31] Kushner, H. J., and Di Masi, G. Approximations for functionals and optimal control problems on jump diffusion processes. *J. Math. Anal. Appl. 63*, 3 (1978), 772–800.
[32] Kushner, H. J., and Dupuis, P. *Numerical methods for stochastic control problems in continuous time*, second ed., vol. 24 of *Applications of Mathematics (New York)*. Springer-Verlag, New York, 2001. Stochastic Modelling and Applied Probability.
[33] Mao, X. *Stochastic differential equations and applications*, second ed. Horwood Publishing Limited, Chichester, 2008.
[34] Maruyama, G. Continuous Markov processes and stochastic equations. *Rend. Circ. Mat. Palermo (2) 4* (1955), 48–90.
[35] Platen, E., and Bruti-Liberati, N. *Numerical solution of stochastic differential equations with jumps in finance*, vol. 64 of *Stochastic Modelling and Applied Probability*. Springer-Verlag, Berlin, 2010.
[36] Sachs, E. W., and Schu, M. Gradient computation for model calibration with pointwise observations. In *Control and optimization with PDE constraints*, vol. 164 of *Internat. Ser. Numer. Math.* Birkhäuser/Springer Basel AG, Basel, 2013, pp. 117–136.
[37] Sobczyk, K. *Stochastic differential equations: with applications to physics and engineering*, vol. 40. Springer Science & Business Media, 2013.
[38] Tröltzsch, F. *Optimal control of partial differential equations: Theory, methods, and applications*, vol. 112 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, Providence, RI, 2010.
[39] Vaswani, S., Mishkin, A., Laradji, I., Schmidt, M., Gidel, G., and Lacoste-Julien, S. Painless stochastic gradient: Interpolation, line-search, and convergence rates. *Adv. Neural. Inf. Process. Syst. 32* (2019).
[40] Yong, J., and Zhou, X. Y. *Stochastic controls*, vol. 43 of *Applications of Mathematics (New York)*. Springer-Verlag, New York, 1999. Hamiltonian systems and HJB equations.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

23/2025 Antonietti, P. F.; Caldana, M.; Mazzieri, I.; Re Fraschini, A.
MAGNET: an open-source library for mesh agglomeration by Graph Neural Networks

22/2025 Leimer Saglio, C. B.; Pagani, S.; Antonietti P. F.
A p-adaptive polytopal discontinuous Galerkin method for simulating brain electrophysiology

21/2025 Caldera, L., Masci, C., Cappozzo, A., Forlani, M., Antonelli, B., Leoni, O., Ieva, F.
Uncovering mortality patterns and hospital effects in COVID-19 heart failure patients: a novel Multilevel logistic cluster-weighted modeling approach

20/2025 Botti, M.; Prada, D.; Scotti, A.; Visinoni, M.
Fully-Mixed Virtual Element Method for the Biot Problem

19/2025 Bortolotti, T.; Wang, Y. X. R.; Tong, X.; Menafoglio, A.; Vantini, S.; Sesia, M.
Noise-Adaptive Conformal Classification with Marginal Coverage

Bortolotti, T.; Wang, Y. X. R.; Tong X.; Menafoglio, A.; Vantini, S.; Sesia, M.
Noise-Adaptive Conformal Classification with Marginal Coverage

18/2025 Antonietti, P.F.; Corti, M.; Gómez, S.; Perugia, I.
A structure-preserving LDG discretization of the Fisher-Kolmogorov equation for modeling neurodegenerative diseases

17/2025 Botti, M.; Mascotto, L.
Sobolev--Poincaré inequalities for piecewise $W^{1,p}$ functions over general polytopic meshes

16/2025 Radisic, I.; Regazzoni, F.; Bucelli, M.; Pagani, S.; Dede', L.; Quarteroni, A.
Influence of cellular mechano-calcium feedback in numerical models of cardiac electromechanics

15/2025 Fois, M.; de Falco, C.; Formaggia L.
Efficient particle generation for depth-averaged and fully 3D MPM using TIFF image data