UNIVERSITY OF CALIFORNIA
SANTA CRUZ

NEURAL SCHRÖDINGER BRIDGE WITH SINKHORN LOSSES

A thesis submitted in partial satisfaction of the
requirements for the degree of

MASTER OF SCIENCE

in

ELECTRICAL AND COMPUTER ENGINEERING

by

Charlie Yan

May 2023

The thesis of Charlie Yan
is approved:

Abhishek Halder

Professor Abhishek Halder

Qi Gong

Professor Qi Gong

DocuSigned by:

5F739D6B0881448...

Professor Dejan Multinovic

Peter F. Biehl
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

Abstract

Neural Schrödinger Bridge

with Sinkhorn Losses

by

Charlie Yan

We consider the problem of directly controlling (i.e., reshaping) stochastic uncertainties subject to continuous-time controlled nonlinear dynamics and fixed deadline constraints. This problem is known as the generalized Schrödinger bridge problem. Historically, a version of the problem originated in the works of Erwin Schrödinger in 1931-32, and can be seen as the stochastic version of the optimal mass transport problem with prior nonlinear dynamics. From a control-theoretic viewpoint, this is an atypical stochastic optimal control problem–"atypical" because instead of boundary conditions on a finite dimensional state space, the problem involves endpoint boundary conditions on the space of joint state probability density functions, or probability measures in general. In recent years, significant strides have been made in the control community to solve such problems for some special classes of prior nonlinear dynamics including control-affine gradient and mixed conservative-dissipative systems. However, the existing algorithms in the literature exploit special structures of such prior nonlinear dynamics, but in the absence of such structures, no general computational framework is available.

In this work, we propose to leverage recent advances in machine learning, specifically physics-informed neural networks (PINNs), to numerically solve generalized Schrödinger bridge problems. We introduce a variant of the standard PINN to

account for the endpoint joint distributional constraints via the Sinkhorn divergence that exploits the geometry on the space of probability measures. We explain how this architecture can be implemented as differentiable layers. Our proposed framework allows numerically solving variants of Schrödinger bridge problems for which no algorithms are available otherwise. This includes systems with control non-affine as well as nonlinear non-autonomous (i.e., explicit time dependent) drifts and diffusions, as well as situations where the controlled dynamics may only be available in a data-driven manner (e.g., in the form of neural networks).

We demonstrate the efficacy of our computational framework using two engineering case studies. The first case study involves optimally steering the stochastic angular velocity of a rotating rigid body from a given statistics to another over a prescribed time horizon. This is of interest, for example, in controlling the spin of a spacecraft in the presence of stochastic uncertainties. The second case study involves controlled colloidal self-assembly for the purpose of advanced manufacturing of materials with desirable properties. In this case, first principle physics-based control-oriented models are difficult to obtain due to complex molecular dynamics, quantum effects and thermal fluctuations. We show how such colloidal self-assembly problems are amenable to generalized Schrödinger bridge formulation, and solve the data-driven distribution steering problems for such systems using our proposed framework. We provide detailed numerical results and discuss the implementation details for the proposed computational architecture and algorithms.

Acknowledgments

I'm very grateful to have met, been taught by, and done research under Professor Halder. I want to thank him for being the best lecturer I've ever met, a genuinely funny and quotable guy, and great advisor. I would also like to thank him for introducing me to convex optimization and optimal transport, dangerous topics to try to teach myself. I would like to thank Iman Nodozi, for his friendship and guidance jumping into research.

I'm very grateful to my family, for giving me a home so I can focus on my studies, and for our dog Winston for keeping me human and being there for me. Finally, I'd like to thank mRNA vaccines, for giving my health back to finish this degree.

Chapter 1

Introduction

In this introductory Chapter, we provide the necessary backgrounds for two problems–the optimal mass transport (OMT) problem, and the Schrödinger bridge problem (SBP)–that lie at the heart of our work. The two problems are related but had very different historical origins. Their connections with each other, and with the disciplines of control theory, optimization, probability, machine learning and the partial differential equations, have only recently come to the fore.

For each of these two problems, we first present the "classical" version: the incarnation in which the problem originated historically. This is then followed up by the "generalized" version: the incarnation in which the problem becomes relevant in our context. The high level difference between the classical and the generalized versions, for both OMT and SBP, is that the generalized version has a prior dynamics encapsulating the underlying "physics" of the problem. In contrast, the corresponding classical version has no prior dynamics, i.e., the problem is formulated in the Wiener path space without any other information.

1.1 Optimal Mass Transport

1.1.1 Classical OMT

We start by summarizing the rudiments on classical OMT. Well-known references for this topic are [74, 75]; for a brief summary see e.g., [36].

Static Formulation. The *static formulation* of classical OMT goes back to Gaspard Monge in 1781, which concerns with finding a mass preserving transport map $\boldsymbol{\theta} : \mathbb{R}^n \mapsto \mathbb{R}^n$ pushing a given measure μ_0 to another μ_T while minimizing a transportation cost $\int_{\mathbb{R}^n} c(\boldsymbol{x}, \boldsymbol{\theta}(\boldsymbol{x}))\mathrm{d}\mu_0$ where c is some ground cost functional, i.e.,

$$c : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}_{\geq 0}.$$

In other words, Monge's formulation is a static optimization problem:

$$\inf_{\text{measurable } \boldsymbol{\theta}:\mathbb{R}^n \mapsto \mathbb{R}^n} \int_{\mathbb{R}^n} c(\boldsymbol{x}, \boldsymbol{\theta}(\boldsymbol{x}))\mathrm{d}\mu_0 \tag{1.1}$$

$$\text{subject to} \qquad \boldsymbol{x} \sim \mu_0, \quad \boldsymbol{\theta}(\boldsymbol{x}) \sim \mu_T. \tag{1.2}$$

Notice that this is an infinite dimensional nonlinear nonconvex problem over measurable maps $\boldsymbol{\theta}(\cdot)$.

A common choice for c is half of the squared Euclidean distance, but in general, the choice of the functional c plays an important role for guaranteeing the existence-uniqueness of the minimizer $\boldsymbol{\theta}^{\mathrm{opt}}(\cdot)$. Even when the existence-uniqueness of the *optimal transport map* $\boldsymbol{\theta}^{\mathrm{opt}}(\cdot)$ can be guaranteed, Monge's formulation is computationally less malleable as it requires solving a nonlinear nonconvex problem over all measurable pushforward mappings $\boldsymbol{\theta} : \mathbb{R}^n \mapsto \mathbb{R}^n$ taking μ_0 to μ_T. For $c(\boldsymbol{x}, \boldsymbol{y}) \equiv \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2$, $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, $\mathrm{d}\mu_0(\boldsymbol{x}) = \rho_0(\boldsymbol{x})\mathrm{d}\boldsymbol{x}, \mathrm{d}\mu_T(\boldsymbol{x}) = \rho_T(\boldsymbol{y})\mathrm{d}\boldsymbol{y}$, it is known [6] that $\boldsymbol{\theta}^{\mathrm{opt}}$ exists, is unique, and admits a representation $\boldsymbol{\theta}^{\mathrm{opt}} = \nabla\psi$ for some convex function ψ. Even then, the direct computation of ψ is numerically

challenging because it reduces to solving a second order nonlinear elliptic Monge-Ampère PDE [74, p. 126]:

$$\det\left(\nabla^2\psi(\boldsymbol{x})\right)\rho_T(\nabla\psi(\boldsymbol{x})) = \rho_0(\boldsymbol{x})$$

where det and ∇^2 denote the determinant and the Hessian, respectively.

A more tractable reformulation of the static OMT is due to Leonid Kantorovich in 1942 [46], which instead of finding the optimal transport map $\boldsymbol{\theta}^{\text{opt}}$, seeks to compute an *optimal coupling* π^{opt} between the given measures μ_0, μ_T that solves

$$\underset{\pi\in\Pi_2(\mu_0,\mu_T)}{\arg\inf}\int_{\mathbb{R}^n\times\mathbb{R}^n} c(\boldsymbol{x},\boldsymbol{y})\mathrm{d}\pi(\boldsymbol{x},\boldsymbol{y}) \tag{1.3}$$

where $\Pi_2(\mu_0,\mu_T)$ denotes the set of all joint probability measures π supported over the product space $\mathbb{R}^n\times\mathbb{R}^n$ with $\boldsymbol{x}$ marginal μ_0, and $\boldsymbol{y}$ marginal μ_T. Notice that (1.3) is an infinite dimensional linear program. The map $\boldsymbol{\theta}^{\text{opt}}$ is precisely the support of the optimal coupling π^{opt}. In the other direction, we can recover π^{opt} from $\boldsymbol{\theta}^{\text{opt}}$ as $\pi^{\text{opt}} = \left(\text{Id}\times\boldsymbol{\theta}^{\text{opt}}\right)\sharp\,\mu_0$ where Id denotes the identity map, and $\sharp$ denotes the pushforward of a probability measure.

Dynamic Formulation. Let $\mathcal{P}_2(\mathbb{R}^n)$ denote the collection of probability density functions (PDFs) supported over $\mathbb{R}^n$ that have finite second moments, i.e.,

$$\mathcal{P}_2(\mathbb{R}^n) := \left\{\rho : \mathbb{R}^n \mapsto \mathbb{R}_{\geq 0} \mid \int_{\mathbb{R}^n}\rho\mathrm{d}\boldsymbol{x} = 1, \int_{\mathbb{R}^n}\|\boldsymbol{x}\|_2^2\rho\mathrm{d}\boldsymbol{x} < \infty\right\}.$$

The *dynamic formulation* of OMT due to Benamou and Brenier [4] appeared at the turn of the 21st century. When $c(\boldsymbol{x},\boldsymbol{y}) \equiv \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$ and μ_0,μ_T admit respective PDFs ρ_0,ρ_T, the dynamic formulation is the following stochastic optimal control

problem:

$$\underset{(\rho^{\boldsymbol{u}},\boldsymbol{u})\in\mathcal{P}_2(\mathbb{R}^n)\times\mathcal{U}}{\arg\inf}\int_0^T\int_{\mathbb{R}^n}\frac{1}{2}\|\boldsymbol{u}\|_2^2\,\rho^{\boldsymbol{u}}(\boldsymbol{x}^{\boldsymbol{u}},t)\mathrm{d}\boldsymbol{x}^{\boldsymbol{u}}\mathrm{d}t \tag{1.4a}$$

$$\frac{\partial\rho^{\boldsymbol{u}}}{\partial t}+\nabla_{\boldsymbol{x}^{\boldsymbol{u}}}\cdot(\rho^{\boldsymbol{u}}\boldsymbol{u})=0, \tag{1.4b}$$

$$\rho^{\boldsymbol{u}}(\boldsymbol{x}^{\boldsymbol{u}},t=0)=\rho_0,\quad \rho^{\boldsymbol{u}}(\boldsymbol{x}^{\boldsymbol{u}},t=T)=\rho_T. \tag{1.4c}$$

The constraint (1.4b) is the *Liouville PDE* (see e.g., [33]) that governs the evolution of the state PDF $\rho^{\boldsymbol{u}}(\boldsymbol{x}^{\boldsymbol{u}},t)$ under a feasible control policy $\boldsymbol{u}\in\mathcal{U}$. So (1.4) is a problem of optimally steering a given joint PDF ρ_0 to another ρ_T over time horizon $[0,T]$ using a vector of single integrators, i.e., with full control authority in $\mathcal{U}$. The solution $(\rho^{\mathrm{opt}},\boldsymbol{u}^{\mathrm{opt}})$ for (1.4) satisfies

$$\rho^{\mathrm{opt}}(\boldsymbol{x}^{\boldsymbol{u}},t)=\boldsymbol{\theta}_t\sharp\,\rho_0,\quad \boldsymbol{\theta}_t:=\left(1-\frac{t}{T}\right)\mathrm{Id}+\frac{t}{T}\boldsymbol{\theta}^{\mathrm{opt}}, \tag{1.5a}$$

$$\boldsymbol{u}^{\mathrm{opt}}(\boldsymbol{x}^{\boldsymbol{u}},t)=\nabla_{\boldsymbol{x}^{\boldsymbol{u}}}\phi(\boldsymbol{x}^{\boldsymbol{u}},t),\quad \frac{\partial\phi}{\partial t}+\frac{1}{2}\|\nabla_{\boldsymbol{x}^{\boldsymbol{u}}}\phi\|_2^2=0. \tag{1.5b}$$

Thus, (1.5a) tells that the optimally controlled PDF is obtained as pushforward of the initial PDF via a map that is a linear interpolation between identity and the optimal transport map. Consequently, the PDF ρ^{opt} itself is a (nonlinear) McCann's displacement interpolant [56] between ρ_0 and ρ_T. The optimal control in (1.5b) is obtained as the gradient of the solution of a *Hamilton-Jacobi-Bellman (HJB) PDE*.

The $\psi(\boldsymbol{x})$ in static OMT and the $\phi(\boldsymbol{x},t)$ in dynamic OMT are related [74, Thm. 5.51] through the *Hopf-Lax representation formula*

$$\phi(\boldsymbol{x},t)=\inf_{\boldsymbol{y}\in\mathbb{R}^n}\left(\phi(\boldsymbol{y},0)+\frac{1}{2t}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2\right),\ t\in(0,T], \tag{1.6a}$$

$$\phi(\boldsymbol{y},0)=\psi(\boldsymbol{y})-\frac{1}{2}\|\boldsymbol{y}\|_2^2, \tag{1.6b}$$

i.e., $\phi(\boldsymbol{x},t)$ is the Moreau-Yosida proximal envelope [65, Ch. 3.1] of $\phi(\boldsymbol{y},0)=\psi(\boldsymbol{y})-\frac{1}{2}\|\boldsymbol{y}\|_2^2$, and hence $\phi(\boldsymbol{x},t)$ is continuously differentiable w.r.t. $\boldsymbol{x}\in\mathbb{R}^n$.

Classical OMT allows defining a distance metric, called the *Wasserstein metric* W, on the manifold of probability measures or PDFs. In particular, when $c(\boldsymbol{x}, \boldsymbol{y}) \equiv \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2$, the infimum value achieved in (1.3) is the one half of the squared Wasserstein metric between μ_0 and μ_T, i.e.,

$$W^2(\mu_0, \mu_T) := \inf_{\pi \in \Pi_2(\mu_0, \mu_T)} \int_{\mathbb{R}^n \times \mathbb{R}^n} \|\boldsymbol{x} - \boldsymbol{y}\|_2^2 \, \mathrm{d}\pi(\boldsymbol{x}, \boldsymbol{y}), \tag{1.7}$$

which is also equal to the infimum value achieved in (1.4), provided μ_0, μ_T are absolutely continuous. The tuple $(\mathcal{P}_2(\mathbb{R}^n), W)$ defines a complete separable metric space, i.e., a polish space. This offers a natural way to metrize the topology of weak convergence of probability measures w.r.t. the metric W.

1.1.2 Generalized OMT

The dynamic version of the generalized OMT is the following stochastic optimal control problem:

$$\inf_{(\rho^{\boldsymbol{u}}, \boldsymbol{u}) \in \mathcal{P}_2(\mathbb{R}^n) \times \mathcal{U}} \int_0^T \int_{\mathbb{R}^n} \left(q(\boldsymbol{x}^{\boldsymbol{u}}) + r(\boldsymbol{u})\right) \, \rho^{\boldsymbol{u}}(\boldsymbol{x}^{\boldsymbol{u}}, t) \mathrm{d}\boldsymbol{x}^{\boldsymbol{u}} \mathrm{d}t \tag{1.8a}$$

$$\frac{\partial \rho^{\boldsymbol{u}}}{\partial t} + \nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \cdot \left(\rho^{\boldsymbol{u}} \boldsymbol{f}\left(t, \boldsymbol{x}^{\boldsymbol{u}}, \boldsymbol{u}\right)\right) = 0, \tag{1.8b}$$

$$\rho^{\boldsymbol{u}}(\boldsymbol{x}^{\boldsymbol{u}}, t = 0) = \rho_0, \quad \rho^{\boldsymbol{u}}(\boldsymbol{x}^{\boldsymbol{u}}, t = T) = \rho_T. \tag{1.8c}$$

We suppose that the cost functions $q(\cdot), r(\cdot)$ and the controlled vector field $\boldsymbol{f}$ are sufficiently smooth to make the problem (1.8) well-posed. For instance, we suppose that $q + r$ is at least lower bounded.

Notice that the problem (1.8) reduces to the problem (1.4) when

$$q(\cdot) \equiv 0, \quad r(\cdot) \equiv \frac{1}{2}\|\cdot\|_2^2, \quad \boldsymbol{f} \equiv \boldsymbol{u}.$$

In other words, (1.8) generalizes (1.4) in two ways. One generalization comes from considering more general (separable) cost-to-go in the objective. Another

generalization comes from considering more general (both time, state and control-dependent) vector field in the Liouville PDE constraint.

Notice that the PDE (1.4b) is induced by the underlying controlled ODE: $\dot{\boldsymbol{x}}^{\boldsymbol{u}} = \boldsymbol{u}$. In contrast, the PDE (1.8b) is induced by the underlying controlled ODE: $\dot{\boldsymbol{x}}^{\boldsymbol{u}} = \boldsymbol{f}\left(t, \boldsymbol{x}^{\boldsymbol{u}}, \boldsymbol{u}\right)$.

Just like the classical dynamic OMT (1.4) corresponds to the classical static OMT (1.3), similar correspondence can be associated with (1.8). To do this, we slightly generalize the setting: we replace $\mathbb{R}^n$ in (1.3) with an n dimensional Riemannian manifold $\mathcal{M}$. Consider an absolutely continuous curve $\boldsymbol{\gamma}(t) \in \mathcal{M}$, $t \in [0, T]$, and $(\boldsymbol{\gamma}, \dot{\boldsymbol{\gamma}}) \in \mathcal{TM}$ (tangent bundle). Then for $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{M}$, we think of $c(\boldsymbol{x}, \boldsymbol{y})$ in (1.3) to be derived from a Lagrangian $L : [0, T] \times \mathcal{TM} \mapsto \mathbb{R}$, i.e., express c as an action integral

$$c(\boldsymbol{x}, \boldsymbol{y}) = \inf_{\boldsymbol{\gamma}(\cdot) \in \Gamma_{\boldsymbol{xy}}} \int_0^T L(t, \boldsymbol{\gamma}(t), \dot{\boldsymbol{\gamma}}(t))\, \mathrm{d}t, \tag{1.9}$$

where

$$\Gamma_{\boldsymbol{xy}} := \{\boldsymbol{\gamma} : [0, T] \mapsto \mathbb{R}^n \mid \boldsymbol{\gamma}(\cdot) \text{ is absolutely continuous}, \boldsymbol{\gamma}(0) = \boldsymbol{x}, \boldsymbol{\gamma}(T) = \boldsymbol{y}\}.$$

In particular, the choice $\mathcal{M} \equiv \mathbb{R}^n$ and $L(t, \boldsymbol{\gamma}, \dot{\boldsymbol{\gamma}}) \equiv \frac{1}{2}\|\dot{\boldsymbol{\gamma}}\|_2^2$ results in $c(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_2^2$, i.e., the standard Euclidean OMT (1.3). When $\mathcal{M} \equiv \mathbb{R}^n$ and $\boldsymbol{f}\left(t, \boldsymbol{x}^{\boldsymbol{u}}, \boldsymbol{u}\right) \equiv \widetilde{\boldsymbol{f}}(t, \boldsymbol{x}^{\boldsymbol{u}}) + \boldsymbol{B}(t)\boldsymbol{u}$, i.e., a control-affine vector field with $\boldsymbol{B}(t) \in \mathbb{R}^{n \times n}$ nonsingular for all $t \in [0, T]$, then the Lagrangain L in (1.9) is

$$L(t, \boldsymbol{\gamma}, \dot{\boldsymbol{\gamma}}) = q(\boldsymbol{\gamma}) + r\left(\left(\boldsymbol{B}(t)\right)^{-1}\left(\dot{\boldsymbol{\gamma}} - \widetilde{\boldsymbol{f}}(t, \boldsymbol{\gamma})\right)\right).$$

We say that a Lagrangian L is *superlinear* (1-coercive) if

$$\lim_{\|\dot{\boldsymbol{\gamma}}\|_2 \to \infty} \frac{L}{\|\dot{\boldsymbol{\gamma}}\|_2} = +\infty. \tag{1.10}$$

For the identified Lagrangian L in (1.9), if the mapping $\dot{\boldsymbol{\gamma}} \mapsto L(\cdot, \cdot, \dot{\boldsymbol{\gamma}})$ is strictly convex as well as superlinear, then we say L is a *weak Tonelli Lagrangian* [75, p. 118], [29, Ch. 6.2]. It is known that if L in (1.9) is a weak Tonelli Lagrangian, then [29, Thm. 1.4.2] the existence and uniqueness of the minimizing pair $(\rho^{\text{opt}}, \boldsymbol{u}^{\text{opt}})$ for (1.8) is guaranteed.

1.2 Schrödinger Bridge Problem

From a control-theoretic viewpoint, both the SBP and the OMT are stochastic optimal control problems. Even then, it is helpful to think about the SBP as a further stochastic generalization of the OMT. We explain this next.

1.2.1 Classical SBP

The classical SBP concerns with the minimum effort additive control needed to move a given distribution to another over a presribed finite time horizon subject to the constraint that the uncontrolled sample paths evolve according to Brownian motion (i.e., standard Wiener process). This is a stochastic optimal control of the form

$$\underset{(\rho^{\boldsymbol{u}}, \boldsymbol{u}) \in \mathcal{P}_2(\mathbb{R}^n) \times \mathcal{U}}{\arg\inf} \int_0^T \int_{\mathbb{R}^n} \frac{1}{2} \|\boldsymbol{u}\|_2^2 \, \rho^{\boldsymbol{u}}(\boldsymbol{x}^{\boldsymbol{u}}, t) \mathrm{d}\boldsymbol{x}^{\boldsymbol{u}} \mathrm{d}t \tag{1.11a}$$

$$\frac{\partial \rho^{\boldsymbol{u}}}{\partial t} + \nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \left(\rho^{\boldsymbol{u}} \boldsymbol{u}\right) = \beta^{-1} \Delta_{\boldsymbol{x}^{\boldsymbol{u}}} \rho^{\boldsymbol{u}}, \tag{1.11b}$$

$$\rho^{\boldsymbol{u}}(\boldsymbol{x}^{\boldsymbol{u}}, t = 0) = \rho_0, \quad \rho^{\boldsymbol{u}}(\boldsymbol{x}^{\boldsymbol{u}}, t = T) = \rho_T. \tag{1.11c}$$

The constant β is referred to as the inverse temperature. Clearly, if $\beta^{-1} \downarrow 0$, then the classical SBP (1.11) reduces to the classical OMT (1.4).

Notice that unlike the first order Liouville PDE (1.4b), the PDE constraint (1.11b) involves a second order Laplacian term. Thus, (1.11b) is a Fokker-Planck-

Kolmogorov (FPK) PDE. This can be motivated as follows. The macroscopic dynamics (1.11b) is induced by the controlled sample path dynamics:

$$\mathrm{d}\boldsymbol{x}^{\boldsymbol{u}} = \boldsymbol{u}\,\mathrm{d}t + \sqrt{2\beta^{-1}}\mathrm{d}\boldsymbol{w} \tag{1.12}$$

where $\boldsymbol{u} \in \mathbb{R}^n$ is the control, and $\boldsymbol{w}$ is the standard Wiener process in $\mathbb{R}^n$. In other words, the trajectory-level controlled dynamics for the classical SBP is "single integrator + Gaussian white noise". The controlled Itô stochastic differential equation (SDE) generalizes the controlled ODE

$$\frac{\mathrm{d}\boldsymbol{x}^{\boldsymbol{u}}}{\mathrm{d}t} = \boldsymbol{u},$$

which was indeed the sample path dynamics associated with (1.4b).

The SBP originated in the works of Erwin Scrödinger [71, 72, 76] and as such predates both the mathematical theory of stochastic processes and feedback control. Scrödinger's original motivation behind this study was to seek a probabilistic interpretation of quantum mechanics.

1.2.2 Generalized SBP

The generalized SBP is a stochastic optimal control problem of the form:

$$\inf_{(\rho^{\boldsymbol{u}},\boldsymbol{u})\in\mathcal{P}_2(\mathbb{R}^n)\times\mathcal{U}} \int_0^T \int_{\mathbb{R}^n} \left(q(\boldsymbol{x}^u) + r(\boldsymbol{u})\right)\rho^{\boldsymbol{u}}(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x}\mathrm{d}t \tag{1.13a}$$

$$\frac{\partial\rho^{\boldsymbol{u}}}{\partial t} + \nabla_{\boldsymbol{x}^{\boldsymbol{u}}}\cdot\left(\rho^{\boldsymbol{u}}\boldsymbol{f}\left(t,\boldsymbol{x}^{\boldsymbol{u}},\boldsymbol{u}\right)\right) = \beta^{-1}\langle\boldsymbol{G}\left(t,\boldsymbol{x}^{\boldsymbol{u}},\boldsymbol{u}\right),\nabla^2_{\boldsymbol{x}^{\boldsymbol{u}}}\rho^{\boldsymbol{u}}\rangle, \tag{1.13b}$$

$$\rho^{\boldsymbol{u}}(\boldsymbol{x}^{\boldsymbol{u}},t=0) = \rho_0, \quad \rho^{\boldsymbol{u}}(\boldsymbol{x}^{\boldsymbol{u}},t=T) = \rho_T, \tag{1.13c}$$

where $\langle\cdot,\cdot\rangle$ in (1.13b) is the Frobenius inner product, $\boldsymbol{G}\left(t,\boldsymbol{x}^{\boldsymbol{u}},\boldsymbol{u}\right)\in\mathbb{R}^{n\times n}$ is the diffuison tensor, and $\nabla^2_{\boldsymbol{x}^{\boldsymbol{u}}}$ is the Hessian operator. The diffusion tensor $\boldsymbol{G}\left(t,\boldsymbol{x}^{\boldsymbol{u}},\boldsymbol{u}\right) := \left(\boldsymbol{g}\left(t,\boldsymbol{x}^{\boldsymbol{u}},\boldsymbol{u}\right)\right)^\top\boldsymbol{g}\left(t,\boldsymbol{x}^{\boldsymbol{u}},\boldsymbol{u}\right)$ for some diffusion coefficient $\boldsymbol{g}\left(t,\boldsymbol{x}^{\boldsymbol{u}},\boldsymbol{u}\right)\in\mathbb{R}^{n\times p}$. Thus, the diffusion tensor, by definition, is symmetric positive semi-definite.

As in the case for the OMT, here too, the qualifier "generalized" refers to the presence of prior dynamics given by the controlled drift-diffusion coefficient pair $(\boldsymbol{f}, \boldsymbol{g})$ which was not considered in Schrödinger's original investigations [71,72]. Specifically, the FPK PDE (1.13b) is induced by a controlled sample path dynamics

$$\mathrm{d}\boldsymbol{x}^{\boldsymbol{u}} = \boldsymbol{f}\left(t, \boldsymbol{x}^{\boldsymbol{u}}, \boldsymbol{u}\right)\, \mathrm{d}t + \sqrt{2\beta^{-1}}\, \boldsymbol{g}\left(t, \boldsymbol{x}^{\boldsymbol{u}}, \boldsymbol{u}\right) \mathrm{d}\boldsymbol{w}, \tag{1.14}$$

where $\boldsymbol{u} \in \mathbb{R}^m$ is the control, and $\boldsymbol{w}$ is a standard Wiener process in $\mathbb{R}$.

Let us now comment on how the classical SBP (1.11) can be seen as a special case of the gernealized SBP (1.13). By setting $\boldsymbol{f} \equiv \boldsymbol{u}$ and $\boldsymbol{g} \equiv \boldsymbol{I}_n$, we see that (1.14) reduces to (1.12), and consequently the macroscopic dynamics (1.13b) specializes to (1.11b). The reduction of (1.13) to (1.11) then follows by setting $q(\cdot) \equiv 0$, $r(\cdot) = \frac{1}{2}\|\boldsymbol{u}\|_2^2$.

In recent years, Schrödinger bridge problems and their connections to OMT have come to prominence in both control [9, 12, 16, 60] and machine learning [13, 24, 78] communities.

1.3 Organization of this Thesis

The remaining of this Thesis is organized as follows. In Chapter 2. we discuss some results about the solution structure of the SBPs: both classical and generalized. In Chapter 3, we explain the proposed idea of using the physics informed neural networks (PINNs) and its variants to numerically solve the generalized SBP. Chapter 4 details a numerical case study where we solve an instance of generalized SBP. In this particular case study, the prior dynamics in the generalized SBP comes from Euler equation describing the spin dynamics for a rigid body. Chapter 5 details another numerical case study solving a model-free instance of generalized SBP appearing in

controlled colloidal self-assembly applications. We provide the concluding remarks and a summary of this work in Chapter 6.

1.4 Publications

The following two manuscripts came out from the research reported in this Thesis.

- C. Yan, I. Nodozi, A. Halder, "Optimal Mass Transport over the Euler Equation", *2023 IEEE Conference on Decision and Control (CDC)*, [under review].
- I. Nodozi, C. Yan, M. Khare, A. Halder, A. Mesbah, "Neural Schrödinger Bridge for Minimum Effort Controlled Colloidal Self Assembly", [in preparation for submission to IEEE Transactions on Control Systems Technology].

In particular, the first manuscript formed the basis for the material in Chapter 4. The second manuscript's content is reported in Chapter 5.

Chapter 2

Solution Structure for the SBP

In this Chapter, we discuss the structural results for the solutions of the SBPs. These will be useful in the ensuing Chapters.

2.1 Solution structure for the classical SBP

Let us consider the classical SBP (1.11). From the first order conditions of optimality, it is easy to verify that the optimal pair $(\rho^{\boldsymbol{u}}_{\text{opt}}, \boldsymbol{u}_{\text{opt}})$ for (1.11), satisfies the following system of PDEs:

$$\frac{\partial \psi}{\partial t} + \frac{1}{2}\|\nabla_{\boldsymbol{x}^{\boldsymbol{u}}}\psi\|_2^2 = -\beta^{-1}\Delta\psi, \tag{2.1a}$$

$$\frac{\partial}{\partial t}\rho^{\boldsymbol{u}}_{\text{opt}} + \nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \cdot (\rho^{\boldsymbol{u}}_{\text{opt}} \nabla_{\boldsymbol{x}^{\boldsymbol{u}}}\psi) = \beta^{-1}\Delta_{\boldsymbol{x}^{\boldsymbol{u}}}\rho^{\boldsymbol{u}}_{\text{opt}}, \tag{2.1b}$$

with boundary conditions:

$$\rho^{\boldsymbol{u}}_{\text{opt}}(\boldsymbol{x}^{\boldsymbol{u}}, , 0) = \rho_0, \quad \rho^{\boldsymbol{u}}_{\text{opt}}(\boldsymbol{x}^{\boldsymbol{u}}, T) = \rho_T, \tag{2.2}$$

and the optimal control is given by $\boldsymbol{u}_{\text{opt}}(\boldsymbol{x}^{\boldsymbol{u}}, t) = \nabla_{\boldsymbol{x}^{\boldsymbol{u}}}\psi(\boldsymbol{x}^{\boldsymbol{u}}, t)$.

The system (2.1) can further be transformed into a system of *linear* PDEs

via the mapping $\big(\rho^{\boldsymbol{u}}_{\text{opt}}, \psi\big) \mapsto (\varphi, \hat{\varphi})$ given by

$$\varphi(\cdot, t) = \exp\left(\frac{\psi(\cdot, t)}{2\beta^{-1}}\right), \tag{2.3a}$$

$$\hat{\varphi}(\cdot, t) = \rho^{\boldsymbol{u}}_{\text{opt}}(\cdot, t) \exp\left(-\frac{\psi(\cdot, t)}{2\beta^{-1}}\right). \tag{2.3b}$$

The transformed variables $(\varphi, \hat{\varphi})$ satisfy the following pair of forward-backward heat equations:

$$\frac{\partial \varphi}{\partial t} = -\beta^{-1}\Delta\varphi, \tag{2.4a}$$

$$\frac{\partial \hat{\varphi}}{\partial t} = \beta^{-1}\Delta\hat{\varphi}, \tag{2.4b}$$

with coupled boundary conditions

$$\varphi(\cdot, t = 0)\hat{\varphi}(\cdot, t = 0) = \rho_0(\cdot), \tag{2.5a}$$

$$\varphi(\cdot, t = T)\hat{\varphi}(\cdot, t = T) = \rho_T(\cdot). \tag{2.5b}$$

To simplify the notation, let

$$\varphi_T(\cdot) := \varphi(\cdot, t = T), \quad \hat{\varphi}_0(\cdot) := \hat{\varphi}(\cdot, t = 0). \tag{2.6}$$

Then, the solution of (2.4) can be formally written as

$$\varphi(\boldsymbol{x}, t) = \int_{\mathbb{R}^n} K(t, \boldsymbol{x}, T, \boldsymbol{y})\varphi_T(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y}, \quad t \le T, \tag{2.7a}$$

$$\hat{\varphi}(\boldsymbol{x}, t) = \int_{\mathbb{R}^n} K(0, \boldsymbol{y}, t, \boldsymbol{x})\hat{\varphi}_0(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y}, \quad t \ge 0, \tag{2.7b}$$

where

$$K(t, \boldsymbol{x}, s, \boldsymbol{y}) := \big(4\pi\beta^{-1}(t - s)\big)^{-n/2} \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|_2^2}{4\beta^{-1}(t - s)}\right) \tag{2.8}$$

is the heat kernel or the Markov kernel associated with the *uncontrolled* diffusion SDE $\mathrm{d}\boldsymbol{x} = \sqrt{2\beta^{-1}}\, \mathrm{d}\boldsymbol{w}(t)$.

Combining (2.5) and (2.7), it follows that finding the minimizer for the classical SBP reduces to solving for the pair $(\varphi_T, \hat{\varphi}_0)$ which satisfies the following

system of nonlinear integral equations, referred to as the *Schrödinger system*, given by

$$\rho_0(\boldsymbol{x}) = \hat{\varphi}_0(\boldsymbol{x}) \int_{\mathbb{R}^n} K(0, \boldsymbol{x}, T, \boldsymbol{y}) \varphi_T(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y}, \tag{2.9a}$$

$$\rho_T(\boldsymbol{x}) = \varphi_T(\boldsymbol{x}) \int_{\mathbb{R}^n} K(0, \boldsymbol{y}, T, \boldsymbol{x}) \hat{\varphi}_0(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y}. \tag{2.9b}$$

The existence and uniqueness of solutions to the Schrödinger system (2.9) were established in [5, 31, 45]. To compute the pair $(\varphi_T, \hat{\varphi}_0)$ from (2.9), a fixed point recursion was proposed in [15]. Such a recursion was also proved [15, Sec. III] to be contractive in Hilbert's projective metric [8, 41]. Once the pair $(\varphi_T, \hat{\varphi}_0)$ is obtained from (2.9), then using (2.7) one can compute the pair $(\varphi, \hat{\varphi})$. Finally, from (2.3), the original decision variables $(\rho^{\boldsymbol{u}}_{\text{opt}}, \boldsymbol{u}_{\text{opt}})$ can be recovered via the mapping

$$\rho^{\boldsymbol{u}}_{\text{opt}}(\cdot, t) = \varphi(\cdot, t) \, \hat{\varphi}(\cdot, t), \tag{2.10a}$$

$$\boldsymbol{u}^{\text{opt}}(\cdot, t) = 2\beta^{-1} \nabla_{(\cdot)} \log \varphi(\cdot, t). \tag{2.10b}$$

From (2.10a), the optimal controlled joint state PDF at any time is a product of the factors φ and $\hat{\varphi}$ at that time, and hence we refer to $(\varphi, \hat{\varphi})$ as the *Schrödinger factors*. Notice that the factors solve the boundary-coupled PDE system (2.4)-(2.5).

2.2 Solution structure for the minimum energy generalized SBP

We now consider the minimum energy generalized SBP, which is (1.13) with the control cost $r(\cdot) := \frac{1}{2}\|\cdot\|_2^2$. For notational ease, let $\boldsymbol{G}_\beta := \beta^{-1}\boldsymbol{G}$. In this case, the analogue of the optimality conditions (2.1) is given by the following result.

Theorem 1. ***(Conditions for optimality)***

The pair $(\rho^{\boldsymbol{u}}_{\text{opt}}, \boldsymbol{u}_{\text{opt}})$ *that solves* (1.13) *with* $r(\cdot) := \frac{1}{2}\|\cdot\|_2^2$, *must satisfy the system*

of coupled PDEs

$$\frac{\partial \psi}{\partial t} = q + \frac{1}{2}\|\boldsymbol{u}_{\text{opt}}\|_2^2 - \langle \nabla_{\boldsymbol{x}^{\boldsymbol{u}}}\psi, \boldsymbol{f}\rangle - \langle \boldsymbol{G}_\beta, \mathbf{Hess}(\psi)\rangle \tag{2.11a}$$

$$\frac{\partial \rho^{\boldsymbol{u}}_{\text{opt}}}{\partial t} = -\nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \cdot (\rho^{\boldsymbol{u}}_{\text{opt}} \boldsymbol{f}) + \langle \boldsymbol{G}_\beta, \mathbf{Hess}(\rho^{\boldsymbol{u}}_{\text{opt}})\rangle, \tag{2.11b}$$

$$\boldsymbol{u}_{\text{opt}} = \nabla_{\boldsymbol{u}_{\text{opt}}} \left(\langle \nabla_{\boldsymbol{x}^{\boldsymbol{u}}}\psi, \boldsymbol{f}\rangle + \langle \boldsymbol{G}_\beta, \mathbf{Hess}(\psi)\rangle \right). \tag{2.11c}$$

with boundary conditions (5.2)*, where* $\psi(\boldsymbol{x}^{\boldsymbol{u}}, t)$ *is a* $C^1(\mathbb{R}^n; [0, T])$ *Lagrange multiplier.*

Proof. Consider the Lagrangian associated with (1.13) and $r(\cdot) := \frac{1}{2}\|\cdot\|_2^2$, given by

$$\begin{aligned}\mathcal{L}(\rho^{\boldsymbol{u}}, \boldsymbol{u}, \psi) := \int_0^T \int_{\mathbb{R}^n} \Big\{ q(\boldsymbol{x}^{\boldsymbol{u}}) + \frac{1}{2}\|\boldsymbol{u}(\boldsymbol{x}^{\boldsymbol{u}}, t)\|_2^2 \, \rho^{\boldsymbol{u}}(\boldsymbol{x}^{\boldsymbol{u}}, t) + \psi(\boldsymbol{x}^{\boldsymbol{u}}, t) \times \\ \left(\frac{\partial \rho^{\boldsymbol{u}}}{\partial t} + \nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \cdot (\rho^{\boldsymbol{u}} \boldsymbol{f}) - \langle \boldsymbol{G}_\beta, \mathbf{Hess}\,(\rho^{\boldsymbol{u}})\rangle \right) \Big\} \mathrm{d}\boldsymbol{x}^{\boldsymbol{u}}\, \mathrm{d}t\end{aligned} \tag{2.12}$$

where $\psi(\boldsymbol{x}^{\boldsymbol{u}}, t)$ is a $C^1(\mathbb{R}^n; [0, T])$ Lagrange multiplier.

Define the set of feasible controls $\mathcal{U}$ comprising of finite energy inputs, i.e.,

$$\mathcal{U} := \{\boldsymbol{u} : \mathbb{R}^n \times [0, T] \mapsto \mathbb{R}^m \mid \langle \boldsymbol{u}, \boldsymbol{u}\rangle < \infty\}. \tag{2.13}$$

Let

$$\mathcal{P}_{0T}(\mathbb{R}^n) := \left\{ \rho(\cdot, t) \geq 0 \mid \int_{\mathbb{R}^n} \rho \mathrm{d}(\cdot) = 1, \rho(\cdot, t = 0) = \rho_0, \quad \rho(\cdot, t = T) = \rho_T \right\}. \tag{2.14}$$

The idea now is to perform the unconstrained minimization of the Lagrangian $\mathcal{L}$ over $\mathcal{P}_{0T}(\mathbb{R}^n) \times \mathcal{U}$. Carrying out integration by parts, the Lagrangian (5.3) can be written as

$$\begin{aligned}\mathcal{L}(\rho^{\boldsymbol{u}}, \boldsymbol{u}, \psi) = \int_0^T \int_{\mathbb{R}^n} \Big(q(\boldsymbol{x}^{\boldsymbol{u}}) + \frac{1}{2}\|\boldsymbol{u}(\boldsymbol{x}^{\boldsymbol{u}}, t)\|_2^2 - \frac{\partial \psi}{\partial t} - \langle \nabla \psi, \boldsymbol{f}\rangle \\ - \langle \boldsymbol{G}_\beta, \mathbf{Hess}(\psi)\rangle \Big) \rho^{\boldsymbol{u}} \mathrm{d}\boldsymbol{x}^{\boldsymbol{u}}\, \mathrm{d}t.\end{aligned} \tag{2.15}$$

Minimizing (5.5) with respect to $\boldsymbol{u}$ for a fixed PDF gives (5.1c). By substituting (5.1c) in (5.5) and equating the resulting expression to zero, we get the dynamic programming equation

$$\int_0^T \int_{\mathbb{R}} \Big(q\left(\boldsymbol{x}^{\boldsymbol{u}}\right) + \frac{1}{2} \| \nabla_{\boldsymbol{u}_{\text{opt}}} \left(\langle \nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \psi, \boldsymbol{f} \rangle + \langle \boldsymbol{G}_\beta, \mathbf{Hess}(\psi) \rangle \right) \|_2^2 \\ - \frac{\partial \psi}{\partial t} - \langle \nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \psi, \boldsymbol{f} \rangle - \langle \boldsymbol{G}_\beta, \mathbf{Hess}(\psi) \rangle \Big) \rho^{\boldsymbol{u}}(\boldsymbol{x}^{\boldsymbol{u}}, t) \mathrm{d}\boldsymbol{x}^{\boldsymbol{u}} \, \mathrm{d}t = 0. \tag{2.16}$$

For above equation to be satisfied for an arbitrary $\rho^{\boldsymbol{u}}$, we must have

$$\frac{\partial \psi}{\partial t} = q\left(\boldsymbol{x}^{\boldsymbol{u}}\right) + \frac{1}{2} \| \nabla_{\boldsymbol{u}_{\text{opt}}} \left(\langle \nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \psi, \boldsymbol{f} \rangle + \langle \boldsymbol{G}_\beta, \mathbf{Hess}(\psi) \rangle \right) \|_2^2 - \langle \nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \psi, \boldsymbol{f} \rangle \\ - \langle \boldsymbol{G}_\beta, \mathbf{Hess}(\psi) \rangle, \tag{2.17}$$

which is the Hamilton-Jacobi-Bellman (HJB) PDE (5.1a). The associated FPK PDE (2.11b) and the boundary conditions (5.2) follow from primal feasibility. ■

We notice that the conditions for optimality (5.1) for the minimum energy generalized SBP is qualitatively different from (2.1) in the sense we now get a system of three (as opposed to two) coupled PDEs. The implicit equation (5.1c) is a result of the *control non-affine* drift and diffusion $\boldsymbol{f}, \boldsymbol{g}$.

2.3 Existing literature on solving the minimum energy Generalized SBP

While solution methods for (5.1) in general are not available in the current literature, specialized algorithms for particular cases have appeared. For instance, [9] considered the case when the drift coefficient is control affine and the diffusion coefficient is $C\left([0,T]\right)$ matrix that is independent of state and input:

$$\boldsymbol{f}\left(t, \boldsymbol{x}^{\boldsymbol{u}}, \boldsymbol{u}\right) \equiv \widetilde{\boldsymbol{f}}\left(t, \boldsymbol{x}^{\boldsymbol{u}}\right) + \boldsymbol{B}(t)\boldsymbol{u}, \quad \boldsymbol{g}\left(t, \boldsymbol{x}^{\boldsymbol{u}}, \boldsymbol{u}\right) \equiv \boldsymbol{B}(t) \in \mathbb{R}^{n \times m}.$$

In this case $m = p$, and the stochastic process noise enters through the input channels (modeling e.g., disturbance in forcing and/or actuation uncertainties). The results in [9] showed that if $\widetilde{\boldsymbol{f}}$ is gradient of a potential, or when $\left(\widetilde{\boldsymbol{f}}, \boldsymbol{B}(t)\right)$ is of mixed conservative-dissipative form, then certain proximal recursions can be designed to numerically solve for the optimal pair $(\rho^{\boldsymbol{u}}_{\text{opt}}, \boldsymbol{u}_{\text{opt}})$. In [12], this result was extended for the case when additional (deterministic) state inequality constraints are present. SBP with nonlinear drifts with full state feedback linearizable structures were considered in [11, 32]. The SBPs for both first and second order noisy nonuniform Kuramoto oscillator models were solved in [60]. Closest to our generalized SBP is the work in [61], which considered control non-affine drift and diffusion coefficients and solved the resulting system of three PDEs via PINN. We will take a similar approach for the case study in Chapter 5.

All the works mentioned above considered minimum energy SBP with the state cost $q \equiv 0$.

We mention here that there exist works on controlling the density of robotic or cellular swarms [58, 59, 64]. The generalized SBPs considered here differ from these works in that SBPs directly impose two-point boundary value constraints on the space of measures in a non-parametric sense. Notice that the space of joint PDFs is an infinite dimensional manifold, and our development utilize the metric geometry underlying this manifold (without assuming square integrability of the joint state density functions which need not hold).

Chapter 3

Neural Networks to Learn the Solution of the Minimum Energy Generalized SBP

3.1 Physics Informed Neural Networks

PINNs are deep neural networks which can be trained to minimize a (possibly weighted) sum of equation errors and initial-boundary condition losses, to approximate the solution of a system of equations such as (5.1). In the next two Chapters we will detail such PINN implementations to solve instances of the minimum energy generalized SBP (1.13). The instances we consider are those for which alternative methods (e.g., Hopf-Cole transform techniques) are not available due to the non-affine control structures.

Since we explain the specific PINN implementations in the ensuing Chapter 4 and Chapter 5, here we only outline the main idea. For solving a spatio-temporal

PDE system such as (5.1), PINN considers the joint space-time domain as the "feature space". In our context, initial-boundary condition losses correspond to the satisfaction of (5.2). Likewise, equation losses correspond to the satisfaction of (5.1). The output of the PINN, in our case, will be the tuple $\big(\rho^{\boldsymbol{u}}_{\text{opt}}, \boldsymbol{u}_{\text{opt}}, \psi\big)$.

As standard, the training of the PINN is done via first order search algorithms such as stochastic gradient descent and its variants such as ADAM. The training is performed over a number of epochs so that all the (equation and initial-boundary condition) residuals are small. The learnt solutions can be verified either graphically (e.g., by inspecting the optimally controlled transient joint PDFs or their marginals) or by performing closed-loop simulation with the learnt control policy and then estimating the controlled PDFs from the ensemble of simulated closed-loop sample paths. We refer to this proposed computational framework of learning the generalized SBP as the *neural Schrödinger bridge*.

3.2 PINNs with Sinkhorn Losses

Typically, PINN implementations in the literature use standard 2-norm losses. While such losses are easy to implement, they do not necessarily take into account the pertinent geometry of the decision variables. For example, the SBP involves the endpoint boundary conditions (5.2) on only one variable (PDF) level. However, the standard 2-norm is less meaningful as a metric to measure the closeness of PDFs.

A more meaningful loss for (5.2) could be in terms of the Wasserstein metric (1.7). However, a direct discretization of the Wasserstein loss (1.7) requires computing a large scale linear program (LP), and differentiating through this large scale LP is not scalable for the training of PINN. This issue can be alleviated by

instead computing the Wasserstein losses with small entropic regularization a.k.a. the Sinkhorn divergences.

For a regularization parameter $\varepsilon > 0$, we refer to the entropy-regularized version of (1.7) as *Sinkhorn divergence*

$$W_\varepsilon^2(\mu_0, \mu_T) := \inf_{\pi \in \Pi_2(\mu_0, \mu_T)} \int_{\mathbb{R}^n \times \mathbb{R}^n} \left\{ \|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \varepsilon \log \pi(\boldsymbol{x}, \boldsymbol{y}) \right\} \mathrm{d}\pi(\boldsymbol{x}, \boldsymbol{y}). \tag{3.1}$$

As $\varepsilon \downarrow 0$, the Sinkhorn divergence (3.1) approaches the Wasserstein metric (1.7). It turns out that the computation of (3.1) reduces to certain matrix scaling problem which can be solved via the so-called Sinkhorn iterations involving modest computation. In particular, we peopose to use losses of the form (3.1) for the endpoint conditions (5.2). We find that it is easy to differentiate through the Sinkhorn computation facilitating training of the PINN. This will be discussed in detail in Chapter 5. We refer to this variant of the proposed framework as the *neural Schrödinger bridge with Sinkhorn losses*.

Chapter 4

Case study: Steering Angular Velocity Distribution

In this Chapter, we present a numerical case study for finite horizon optimal steering of the joint state probability distribution subject to the angular velocity dynamics governed by the Euler equation. The problem and its solution amounts to controlling the spin of a rigid body via feedback, and is of practical importance, for example, in angular stabilization of a spacecraft with stochastic initial and terminal states.

4.1 Problem Formulation

The controlled angular velocity dynamics for a rotating rigid body such as a spacecraft, is given by the well-known Euler equation

$$\boldsymbol{J}\dot{\boldsymbol{\omega}} = -[\boldsymbol{\omega}]^{\times}\boldsymbol{J}\boldsymbol{\omega} + \boldsymbol{\tau}, \tag{4.1}$$

where the positive diagonal matrix $\boldsymbol{J} := \mathrm{diag}(J_1, J_2, J_3)$ comprises of the principal moments of inertia, the vector $\boldsymbol{\omega} := (\omega_1, \omega_2, \omega_3)^{\top} \in \mathbb{R}^3$ denotes the body's angular

velocity (in rad/s) along its principal axes, the vector $\boldsymbol{\tau} := (\tau_1, \tau_2, \tau_3)^\top \in \mathbb{R}^3$ denotes the torque input applied about the principal axes, and

$$[\boldsymbol{\omega}]^\times := \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \in \mathfrak{so}(3).$$

As usual, $\mathfrak{so}(3)$ denotes the Lie algebra of the three dimensional rotation group SO(3). Motivated by the problem of steering the probabilistic uncertainties in angular velocities over a prescribed time horizon, we consider the deterministic and stochastic variants of the optimal mass transport (OMT) [4, 74, 75] over the Euler equation, which we refer to as the OMT-EE.

Specifically, let $\mathcal{P}_2(\mathbb{R}^3)$ denote the manifold of probability measures supported on $\mathbb{R}^3$ with finite second moments. Given two probability measures $\mu_0, \mu_T \in \mathcal{P}_2(\mathbb{R}^3)$, the OMT-EE associated with (4.1) is a stochastic optimal control problem:

$$\inf_{\boldsymbol{u}\in\mathcal{U}} \int_0^T \mathbb{E}_{\mu^{\boldsymbol{u}}} \left[q(\boldsymbol{x}^{\boldsymbol{u}}) + r(\boldsymbol{u})\right] \mathrm{d}t \tag{4.2a}$$

$$\text{subject to} \quad \dot{\boldsymbol{x}}^{\boldsymbol{u}} = \boldsymbol{\alpha} \odot \boldsymbol{f}(\boldsymbol{x}^{\boldsymbol{u}}) + \boldsymbol{\beta} \odot \boldsymbol{u}, \; i \in [\![3]\!] := \{1, 2, 3\}, \tag{4.2b}$$

$$\mu^{\boldsymbol{u}}(\boldsymbol{x}^{\boldsymbol{u}}, t = 0) = \mu_0 \text{ (given)}, \; \mu^{\boldsymbol{u}}(\boldsymbol{x}^{\boldsymbol{u}}, t = T) = \mu_T \text{ (given)}, \tag{4.2c}$$

where the fixed time horizon is $[0, T]$ for some prescribed $T > 0$, and $\mathbb{E}_{\mu^{\boldsymbol{u}}}[\cdot]$ denotes the expectation w.r.t. the controlled state probability measure $\mu^{\boldsymbol{u}}(\boldsymbol{x}^{\boldsymbol{u}}, t)$ for $t \in [0, T]$, i.e., $\mathbb{E}_{\mu^{\boldsymbol{u}}}[\cdot] := \int(\cdot)\, \mathrm{d}\mu^{\boldsymbol{u}}$. The superscript $\boldsymbol{u}$ for a variable indicates that variable's dependence on the choice of control $\boldsymbol{u}$. The symbol $\odot$ denotes the elementwise (Hadamard) vector product.

The correspondence between (4.1) and (4.2b) follows by noting that the

controlled state $\boldsymbol{x}^{\boldsymbol{u}} \equiv$ controlled $\boldsymbol{\omega}$, the control $\boldsymbol{u} \equiv \boldsymbol{\tau}$, the vector field

$$\boldsymbol{f}(\boldsymbol{z}) := (z_2 z_3, z_3 z_1, z_1 z_2)^\top \text{ for } \boldsymbol{z} \in \mathbb{R}^3, \tag{4.3}$$

and the parameter vectors $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^3$ have entries

$$\alpha_i := (J_{i+1 \mod 3} - J_{i+2 \mod 3})/J_i,\ \beta_i := 1/J_i,\ i \in [\![3]\!]. \tag{4.4}$$

The cost-to-go in (4.2a) comprises of an additive state cost $q(\cdot)$, and a strictly convex and superlinear (i.e., 1-coercive) control cost $r(\cdot)$. Of particular interest is the case $q(\cdot) \equiv 0$ and $r(\cdot) = \frac{1}{2}\|\cdot\|_2^2$ which corresponds to minimum effort control. We suppose that $q + r$ is lower bounded.

Let Ω be the space of continuous functions $\eta : [0, T] \mapsto \mathbb{R}^3$, which is a complete separable metric space endowed with the topology of uniform convergence on compact time intervals. With Ω, we associate the σ-algebra $\mathscr{F} = \sigma\{\eta(s) \mid 0 \leq s \leq T\}$, and consider the complete filtered probability space $(\Omega, \mathscr{F}, \mathbb{P})$ with filtration $\mathscr{F}_t = \sigma\{\eta(s) \mid 0 \leq s \leq t \leq T\}$. So, $\mathscr{F}_0$ contains all $\mathbb{P}$-null sets and $\mathscr{F}_t$ is right continuous. The stochastic initial condition $\boldsymbol{x}^u(t = 0)$ in (4.2) is $\mathscr{F}_0$ measurable. For a given control policy $\boldsymbol{u}$, the controlled state $\boldsymbol{x}^u(t)$ is $\mathscr{F}_t$-adapted (i.e., non-anticipating) for all $t \in [0, T]$.

In (4.2), the set of feasible Markovian control policies

$$\mathcal{U} := \{\boldsymbol{u} : \mathbb{R}^3 \times [0, T] \mapsto \mathbb{R}^3 \mid \int_0^T \mathbb{E}_{\mu^{\boldsymbol{u}}}\left[r(\boldsymbol{u})\right] \mathrm{d}t < \infty\}. \tag{4.5}$$

Thus, solving (4.2) amounts to designing an admissible Markovian control policy $\boldsymbol{u} \in \mathcal{U}$ that transfers the stochastic angular velocity state from a prescribed initial to a prescribed terminal probability measure under the controlled sample path dynamics constraint (4.2b), and hard deadline constraint. The initial and terminal measures can be interpreted as the estimated and allowable statistical uncertainty

specifications, respectively, and therefore, problem (4.2) asks to directly control or reshape uncertainties in a nonparametric sense [18].

4.2 Related Works

Continuous time deterministic optimal control subject to the angular velocity dynamics given by the Euler equation, has been studied in several prior works. In the finite horizon setting, Athans et. al. [1] derived the minimum time, minimum fuel (assuming free terminal time) and minimum energy (assuming fixed terminal time) controllers–all steering an arbitrary initial angular velocity vector to zero. Considering free terminal state and no terminal cost, Kumar [48] showed that a tangent hyperbolic feedback is optimal for finite horizon problem w.r.t. quadratic state and quadratic control cost-to-go. Again considering free terminal state, Dwyer [27] derived the optimal finite horizon controller w.r.t. quadratic state and quadratic control cost-to-go, as well as quadratic terminal cost. Infinite horizon optimal control problem w.r.t. quadratic state and control objective was studied in [73].

Besides control design, systems-theoretic properties for (4.1) are known too. Thanks to the periodicity of unforced motion, (4.1) enjoys global controllability guarantees and it is known [7, Thm. 4 and disussions thereafter], [2, Reamrk in p. 895] that the controlled dynamics is reachable on entire $\mathbb{R}^3$. See also [21].

Formulating and solving the OMT with prior dynamics is a relatively recent endeavor, see e.g., [11, 17, 28, 44]. To the best of the author's knowledge, OMT over the Euler equation has not been investigated before.

4.3 OMT-EE as Generalized OMT

We suppose that the endpoint measures $\mu_0, \mu_T \in \mathcal{P}_2(\mathbb{R}^3)$ in (4.2) are absolutely continuous with respective PDFs ρ_0, ρ_T, and rewrite (4.2) as

$$\inf_{(\rho^{\boldsymbol{u}},\boldsymbol{u})\in\mathcal{P}_2(\mathbb{R}^3)\times\mathcal{U}} \int_0^T \int_{\mathbb{R}^3} (q(\boldsymbol{x}^{\boldsymbol{u}}) + r(\boldsymbol{u}))\ \rho^{\boldsymbol{u}}(\boldsymbol{x}^{\boldsymbol{u}}, t)\mathrm{d}\boldsymbol{x}^{\boldsymbol{u}}\mathrm{d}t \tag{4.6a}$$

$$\frac{\partial \rho^{\boldsymbol{u}}}{\partial t} + \nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \cdot (\rho^{\boldsymbol{u}} (\boldsymbol{\alpha} \odot \boldsymbol{f}(\boldsymbol{x}^{\boldsymbol{u}}) + \boldsymbol{\beta} \odot \boldsymbol{u})) = 0, \tag{4.6b}$$

$$\rho^{\boldsymbol{u}}(\boldsymbol{x}^{\boldsymbol{u}}, t = 0) = \rho_0, \quad \rho^{\boldsymbol{u}}(\boldsymbol{x}^{\boldsymbol{u}}, t = T) = \rho_T. \tag{4.6c}$$

Clearly, problem (4.6) is an instance of the generalized OMT (1.8).

We clarify here that the solution of the Liouville PDE (4.6b) is understood in the weak sense, i.e., for all compactly supported smooth test functions $\zeta(\boldsymbol{x}^{\boldsymbol{u}}, t) \in C_c^\infty\left([0, T] \times \mathbb{R}^3\right)$, the function $\rho^{\boldsymbol{u}}(\boldsymbol{x}^{\boldsymbol{u}}, t)$ satisfies:

$$\int_0^T \int_{\mathbb{R}^3} \left(\rho^{\boldsymbol{u}} \frac{\partial \zeta}{\partial t} + \rho^{\boldsymbol{u}} \langle \boldsymbol{\alpha} \odot \boldsymbol{f}(\boldsymbol{x}^{\boldsymbol{u}}) + \boldsymbol{\beta} \odot \boldsymbol{u}, \nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \zeta \rangle\right) \mathrm{d}\boldsymbol{x}^{\boldsymbol{u}}\mathrm{d}t + \int_{\mathbb{R}^3} \rho_0(\boldsymbol{x}^{\boldsymbol{u}}) \zeta(\boldsymbol{x}^{\boldsymbol{u}}, t = 0)\mathrm{d}\boldsymbol{x}^{\boldsymbol{u}} = 0. \tag{4.7}$$

4.3.1 Static OMT-EE

At this point, a natural question arises: if (4.6) is the Euler equation generalization of the dynamic OMT (1.4), then what is the corresponding generalization of the static OMT (1.3)?

Answering this requires us to identify the Lagrangian in (1.9). For OMT-EE, $\mathcal{M} \equiv \mathbb{R}^3$ and we have the Lagrangian

$$L(t, \boldsymbol{\gamma}, \dot{\boldsymbol{\gamma}}) \equiv q(\boldsymbol{\gamma}) + r((\dot{\boldsymbol{\gamma}} - \boldsymbol{\alpha} \odot \boldsymbol{f}) \oslash \boldsymbol{\beta}) \tag{4.8}$$

where $\oslash$ denotes vector element-wise (Hadamard) division. In particular, L in (4.8) has no explicit dependence on t, i.e., $L : \mathcal{TM} \mapsto \mathbb{R}$.

This identification allows us to define the *static OMT-EE* as the linear program (1.9) wherein the functional c is given by (4.8). The feasible set (1.9) $\Pi_2(\mu_0, \mu_T)$ here denotes the set of all joint probability measures π supported over the product space $\mathbb{R}^3 \times \mathbb{R}^3$ with $\boldsymbol{x}$ marginal μ_0, and $\boldsymbol{y}$ marginal μ_1. We next show that identifying (4.8) also helps establish the existence-uniqueness of solution for the specific generalized OMT (4.6).

4.3.2 Back to Dynamic OMT-EE (4.6): Existence and Uniqueness

Theorem 2. ***(Existence-uniqueness)*** *Let $r : \mathbb{R}^3 \mapsto \mathbb{R}_{\geq 0}$ be strictly convex and superlinear function. Then the minimizing tuple $(\rho^{\text{opt}}, \boldsymbol{u}^{\text{opt}})$ for problem (4.6) exists and is unique.*

Proof. Since r is strictly convex, so L in (4.8) viewed as function of $\dot{\boldsymbol{\gamma}} \in \mathbb{R}^3$, is strictly convex composed with an affine map. Therefore, L is strictly convex in $\dot{\boldsymbol{\gamma}}$.

We next show that L in (4.8) is also superlinear in $\dot{\boldsymbol{\gamma}} \in \mathbb{R}^3$. To see this, notice that

$$\begin{aligned}\lim_{\|\dot{\boldsymbol{\gamma}}\|_2 \to \infty} \frac{L}{\|\dot{\boldsymbol{\gamma}}\|_2} &= \lim_{\|\dot{\boldsymbol{\gamma}}\|_2 \to \infty} \frac{r((\dot{\boldsymbol{\gamma}} - \boldsymbol{\alpha} \odot \boldsymbol{f}) \oslash \boldsymbol{\beta})}{\|\dot{\boldsymbol{\gamma}}\|_2} \\ &= \lim_{\|\boldsymbol{z}\|_2 \to \infty} \frac{r(\boldsymbol{z})}{\|\boldsymbol{\alpha} \odot \boldsymbol{f} + \boldsymbol{\beta} \odot \boldsymbol{z}\|_2}. \end{aligned} \tag{4.9}$$

Using triangle inequality: $\|\boldsymbol{\alpha} \odot \boldsymbol{f} + \boldsymbol{\beta} \odot \boldsymbol{z}\|_2 \leq \|\boldsymbol{\alpha} \odot \boldsymbol{f}\|_2 + \|\boldsymbol{\beta} \odot \boldsymbol{z}\|_2 \leq \|\boldsymbol{\alpha} \odot \boldsymbol{f}\|_2 + \|\boldsymbol{\beta}\|_\infty \|\boldsymbol{z}\|_2$, and hence

$$\frac{r(\boldsymbol{z})}{\|\boldsymbol{\alpha} \odot \boldsymbol{f} + \boldsymbol{\beta} \odot \boldsymbol{z}\|_2} \geq \frac{r(\boldsymbol{z})}{\|\boldsymbol{\alpha} \odot \boldsymbol{f}\|_2 + \|\boldsymbol{\beta}\|_\infty \|\boldsymbol{z}\|_2}.$$

Taking the limit $\|\boldsymbol{z}\|_2 \to \infty$ to both sides of above, we obtain

$$\begin{aligned}(4.9) &\geq \lim_{\|\boldsymbol{z}\|_2 \to \infty} \frac{r(\boldsymbol{z})}{\|\boldsymbol{\alpha} \odot \boldsymbol{f}\|_2 + \|\boldsymbol{\beta}\|_\infty \|\boldsymbol{z}\|_2} \\ &= \lim_{\|\boldsymbol{z}\|_2 \to \infty} \frac{r(\boldsymbol{z})/\|\boldsymbol{z}\|_2}{\|\boldsymbol{\beta}\|_\infty} = +\infty, \end{aligned}$$

since r is superlinear, and $\|\boldsymbol{\alpha} \odot \boldsymbol{f}\|_2, \|\boldsymbol{\beta}\|_\infty > 0$. Thus, (4.9) itself equals to $+\infty$, thereby proving that L is indeed superlinear.

The Lagrangian (4.8) being both strictly convex and superlinear in $\dot{\boldsymbol{\gamma}}$, is a weak *Tonelli Lagrangian* [75, p. 118], [29, Ch. 6.2], and therefore guarantees [29, Thm. 1.4.2] the existence and uniqueness of the minimizing pair $(\rho^{\text{opt}}, \boldsymbol{u}^{\text{opt}})$ for problem (4.6). ■

4.4 Conditions of Optimality, $q(\cdot) \equiv 0$, $r = \frac{1}{2}\|\cdot\|_2^2$

A specific instance of (4.6) that is of practical interest is *minimum energy angular velocity steering*, i.e., the case

$$q(\cdot) \equiv 0, \quad r = \frac{1}{2}\|\cdot\|_2^2.$$

Then, (4.6) resembles the Benamou-Brenier dynamic OMT (1.4) except that the controlled Liouville PDE (4.6b) has a prior bilinear drift which (1.4b) does not have.

Theorem 3. *(**Necessary conditions of optimality for minimum energy steering of angular velocity PDF without process noise**) The optimal tuple* $(\rho^{\text{opt}}, \boldsymbol{u}^{\text{opt}})$ *solving problem* (4.6) *with* $q(\cdot) \equiv 0$, $r = \frac{1}{2}\|\cdot\|_2^2$, *satisfies the following first order necessary conditions of optimality:*

$$\frac{\partial \phi}{\partial t} + \frac{1}{2}\|\boldsymbol{\beta} \odot \nabla_{\boldsymbol{x}^{\boldsymbol{u}}}\phi\|_2^2 + \langle \nabla_{\boldsymbol{x}^{\boldsymbol{u}}}\phi, \boldsymbol{\alpha} \odot \boldsymbol{f}(\boldsymbol{x}^{\boldsymbol{u}})\rangle = 0, \tag{4.10a}$$

$$\frac{\partial \rho^{\text{opt}}}{\partial t} + \nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \cdot \big(\rho^{\text{opt}}\big(\boldsymbol{\alpha} \odot \boldsymbol{f}(\boldsymbol{x}^{\boldsymbol{u}}) + \boldsymbol{\beta}^2 \odot \nabla_{\boldsymbol{x}^{\boldsymbol{u}}}\phi\big)\big) = 0, \tag{4.10b}$$

$$\rho^{\text{opt}}(\boldsymbol{x}^{\boldsymbol{u}}, t = 0) = \rho_0, \quad \rho^{\text{opt}}(\boldsymbol{x}^{\boldsymbol{u}}, t = T) = \rho_T, \tag{4.10c}$$

$$\boldsymbol{u}^{\text{opt}} = \boldsymbol{\beta} \odot \nabla_{\boldsymbol{x}^{\boldsymbol{u}}}\phi, \tag{4.10d}$$

where $\boldsymbol{\beta}^2$ *denotes the vector element-wise square.*

Proof. Consider problem (4.6) with $q \equiv 0$, $r(\cdot) = \frac{1}{2}\|\cdot\|_2^2$, and its associated Lagrangian

$$\mathcal{L}\left(\rho^{\boldsymbol{u}}, \boldsymbol{u}, \phi\right) := \int_0^T \int_{\mathbb{R}^3} \Bigg\{ \frac{1}{2}\|\boldsymbol{u}(\boldsymbol{x}^{\boldsymbol{u}}, t)\|_2^2 \, \rho^{\boldsymbol{u}}(\boldsymbol{x}^{\boldsymbol{u}}, t) + \phi(\boldsymbol{x}^{\boldsymbol{u}}, t) \left(\frac{\partial \rho^{\boldsymbol{u}}}{\partial t} + \nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \cdot \left(\rho^{\boldsymbol{u}} \left(\boldsymbol{\alpha} \odot \boldsymbol{f}(\boldsymbol{x}^{\boldsymbol{u}}) + \boldsymbol{\beta} \odot \boldsymbol{u} \right) \right) \right) \Bigg\} \mathrm{d}\boldsymbol{x}^{\boldsymbol{u}} \mathrm{d}t \tag{4.11}$$

where the Lagrange multiplier $\phi \in C^1\left(\mathbb{R}^3; [0, T]\right)$. Let $\mathcal{P}_{0T}$ denote the family of PDF-valued curves over $[0, T]$ satisfying (4.10c). We perform unconstrained minimization of (4.11) over $\mathcal{P}_{0T} \times \mathcal{U}$.

Performing integration-by-parts of the right-hand-side of (4.11) and assuming the limits for $\|\boldsymbol{x}^{\boldsymbol{u}}\|_2 \to \infty$ are zero, we arrive at the unconstrained minimization of

$$\int_0^T \int_{\mathbb{R}^3} \Bigg(\frac{1}{2}\|\boldsymbol{u}(\boldsymbol{x}^{\boldsymbol{u}}, t)\|_2^2 - \frac{\partial \phi}{\partial t} - \langle \nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \phi, \boldsymbol{\alpha} \odot \boldsymbol{f}(\boldsymbol{x}^{\boldsymbol{u}}) + \boldsymbol{\beta} \odot \boldsymbol{u} \rangle \Bigg) \rho^{\boldsymbol{u}}(\boldsymbol{x}^u, t) \, \mathrm{d}\boldsymbol{x}^{\boldsymbol{u}} \, \mathrm{d}t. \tag{4.12}$$

Pointwise minimization of the integrand in (4.12) w.r.t. $\boldsymbol{u}$ for each fixed PDF-valued curve in $\mathcal{P}_{0T}$, gives

$$\boldsymbol{u}^{\mathrm{opt}} = \mathrm{diag}\left(\boldsymbol{\beta}\right) \nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \phi,$$

which is the same as (4.10d). Substituting the above expression for optimal control back in (4.12), and equating the resulting expression to zero, we obtain the dynamic programming equation

$$\int_0^T \int_{\mathbb{R}^3} \left(-\frac{\partial \phi}{\partial t} - \frac{1}{2} \left\| \boldsymbol{\beta} \odot \nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \phi \right\|_2^2 - \langle \nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \phi, \boldsymbol{\alpha} \odot \boldsymbol{f}(\boldsymbol{x}^{\boldsymbol{u}}) \rangle \right) \rho^{\boldsymbol{u}}(\boldsymbol{x}^u, t) \, \mathrm{d}\boldsymbol{x}^{\boldsymbol{u}} \, \mathrm{d}t = 0. \tag{4.13}$$

For (4.13) to hold for any feasible $\rho^{\boldsymbol{u}}(\boldsymbol{x}^u, t)$, the expression within the parentheses must vanish, which gives us the HJB PDE (4.10a).

Since ρ^{opt} must satisfy the feasibility conditions (4.6b)-(4.6c), hence substituting (4.10d) therein yields (4.10b)-(4.10c). ■

Remark 1. *Equations* (4.10d) *and* (4.10a) *generalize the condition* (1.5b) *in classical dynamic OMT. The solution of the coupled system of HJB PDE* (4.10a) *and Liouville PDE* (4.10b) *with boundary conditions* (4.10c) *yields the optimal PDF* ρ^{opt}.

4.5 Numerical Results

4.5.1 Uncontrolled PDF Evolution

Before delving into the approximate numerical solution for the optimally controlled PDF evolution, we briefly remark on the *uncontrolled* PDF evolution. Specifically, we show next that the bilinear structure of the drift vector field in Eulerian dynamics (4.2b) allows analytic handle on the uncontrolled PDFs, which will come in handy later for comparing the optimally controlled versus uncontrolled evolution of the stochastic states.

In the absence of control, we denote the uncontrolled state vector as $\boldsymbol{x}$, and the uncontrolled joint state PDF as ρ (i.e., without the $\boldsymbol{u}$ superscripts). In that case, (4.6b) specializes to the uncontrolled Liouville PDE

$$\frac{\partial \rho}{\partial t} + \nabla_{\boldsymbol{x}} \cdot (\rho \boldsymbol{\alpha} \odot \boldsymbol{f}(\boldsymbol{x})) = 0. \tag{4.14}$$

Since the drift in (4.2b) is divergence free, we can explicitly solve (4.14) with known initial condition $\rho(\boldsymbol{x}, t = 0) = \rho_0$, as

$$\rho(\boldsymbol{x}, t) = \rho_0 \left(\boldsymbol{x}_0 \left(\boldsymbol{x}, t\right)\right) \tag{4.15}$$

where $\boldsymbol{x}_0(\boldsymbol{x}, t)$ is the *inverse flow map* associated with the unforced initial value problem:

$$\dot{\boldsymbol{x}} = \boldsymbol{\alpha} \odot \boldsymbol{f}(\boldsymbol{x}), \quad \boldsymbol{x}(t=0) = \boldsymbol{x}_0. \tag{4.16}$$

For an asymmetric rigid body, we have $J_1 \neq J_2 \neq J_3$, and the corresponding flow map $\boldsymbol{x}(\boldsymbol{x}_0, t)$ for (4.16) is given component-wise by (see e.g., [49, equation (37.10)])

$$x_1 = \overline{x}_{10}\,\mathrm{cn}\left(\omega_p t + \lambda_1, \lambda_2\right), \tag{4.17a}$$

$$x_2 = \overline{x}_{20}\,\mathrm{sn}\left(\omega_p t + \lambda_1, \lambda_2\right), \tag{4.17b}$$

$$x_3 = \overline{x}_{30}\,\mathrm{dn}\left(\omega_p t + \lambda_1, \lambda_2\right), \tag{4.17c}$$

where cn (elliptic cosine), sn (elliptic sine), dn (delta amplitude) are the Jacobi elliptic functions, and the variables $\overline{x}_{i0} \forall i \in [\![3]\!]$, ω_p, λ_1, λ_2 depend only on $\boldsymbol{x}_0$. In Sec. 4.5, we numerically compute the inverse flow map $\boldsymbol{x}_0(\boldsymbol{x}, t)$ associated with (4.17).

For an axisymmetric rigid body, we have $J_1 = J_2 \neq J_3$, and the inverse flow map $\boldsymbol{x}_0(\boldsymbol{x}, t)$ for (4.16) can be computed component-wise analytically as

$$\gamma := \frac{x_2 - x_1 \tan\left(\alpha_2 x_3 t\right)}{x_1 + x_2 \tan\left(\alpha_2 x_3 t\right)}, \tag{4.18a}$$

$$x_{10} = \left(\frac{x_1^2 + x_2^2}{1 + \gamma^2}\right)^{\frac{1}{2}}, \tag{4.18b}$$

$$x_{20} = \gamma\, x_{10} = \gamma \left(\frac{x_1^2 + x_2^2}{1 + \gamma^2}\right)^{\frac{1}{2}}, \tag{4.18c}$$

$$x_{30} = x_3, \tag{4.18d}$$

To see this, notice that for $J_1 = J_2 \neq J_3$, (4.16) specializes to

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} \alpha_1 x_2 x_3 \\ \alpha_2 x_3 x_1 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} x_1(t=0) \\ x_2(t=0) \\ x_3(t=0) \end{pmatrix} = \begin{pmatrix} x_{10} \\ x_{20} \\ x_{30} \end{pmatrix}, \tag{4.19}$$

giving $x_3(t) = x_{30}$, and

$$\dot{x}_1 = (\alpha_1 x_{30}) x_2 \Rightarrow \ddot{x}_1 = (\alpha_1 x_{30}) \dot{x}_2 = \left(\alpha_1 \alpha_2 x_{30}^2\right) x_1 = -\left(\alpha_2^2 x_{30}^2\right) x_1, \ (\text{as } \alpha_1 = -\alpha_2),$$

which is a simple harmonic oscillator with angular frequency $\alpha_2 x_{30}$, yielding general solution:

$$x_1(t) = A\cos\left(\alpha_2 x_{30} t + \varphi\right), \quad x_2(t) = A\sin\left(\alpha_2 x_{30} t + \varphi\right).$$

The constants A and φ can be determined from the initial conditions: $x_1(t=0) = x_{10}$, $\dot{x}_1(t=0) = \alpha_1 x_{20} x_{30}$, as $A = \sqrt{x_{10}^2 + x_{20}^2}$ and $\varphi = \arctan(x_{20}/x_{10})$. Then we express A, φ as function of the current states as

$$A = \sqrt{x_1^2 + x_2^2}, \quad \varphi = \arctan(x_2/x_1) - \alpha_2 x_3 t.$$

Finally back out A, φ to x_{10}, x_{20} using $\varphi = \arctan(x_{20}/x_{10})$ as

$$\begin{aligned} & \alpha_2 x_3 t = \arctan\left(x_2/x_1\right) - \arctan(x_{20}/x_{10}) \\ \Leftrightarrow \quad & x_{20}/x_{10} = \arctan\left(x_2/x_1\right) - \arctan\tan\left(\alpha_2 x_3 t\right) \\ & \qquad = \arctan\left(\frac{\frac{x_2}{x_1} - \tan\left(\alpha_2 x_3 t\right)}{1 + \frac{x_2}{x_1}\tan\left(\alpha_2 x_3 t\right)}\right) \\ \Leftrightarrow \quad & x_{20}/x_{10} = \gamma. \end{aligned} \tag{4.20}$$

On the other hand,

$$A = \sqrt{x_{10}^2 + x_{20}^2} = \sqrt{x_1^2 + x_2^2},$$

whereupon using $x_{20} = \gamma x_{10}$, and then solving for x_{10} yields (4.18b). Now invoking $x_{20} = \gamma x_{10}$ again, we obtain (4.18c).

Using (4.18), we see that (4.15) takes the form

$$\rho(x_1, x_2, x_3, t) = \rho_0 \left(\left(\frac{x_1^2 + x_2^2}{1+\gamma^2} \right)^{\frac{1}{2}}, \gamma \left(\frac{x_1^2 + x_2^2}{1+\gamma^2} \right)^{\frac{1}{2}}, x_3 \right).$$

4.5.2 Minumum Energy Stochastic OMT-EE

To facilitate the numerical solution of the dynamic OMT-EE discussed in Sec. 4.4, i.e., the solution of (4.6) with $q \equiv 0$, $r(\cdot) \equiv \frac{1}{2}\|\cdot\|_2^2$, we perturb the sample path dynamics (4.2b) with an additive process noise resulting in the Itô stochastic differential equation (SDE):

$$\mathrm{d}\boldsymbol{x}^{\boldsymbol{u}} = (\boldsymbol{\alpha} \odot \boldsymbol{f}(\boldsymbol{x}^{\boldsymbol{u}}) + \boldsymbol{\beta} \odot \boldsymbol{u})\,\mathrm{d}t + \sqrt{2\delta}\,\mathrm{d}\boldsymbol{w},\ \delta > 0. \tag{4.21}$$

The $\boldsymbol{w}$ in (4.21) denotes standard Wiener process in $\mathbb{R}^3$. Due to process noise, the first order Liouville PDE (4.6b) is replaced by the second order Fokker-Planck-Kolmogorov (FPK) PDE

$$\frac{\partial \rho^{\boldsymbol{u}}}{\partial t} + \nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \cdot \left(\rho^{\boldsymbol{u}} \left(\boldsymbol{\alpha} \odot \boldsymbol{f}(\boldsymbol{x}^{\boldsymbol{u}}) + \boldsymbol{\beta} \odot \boldsymbol{u}\right)\right) = \delta \Delta_{\boldsymbol{x}^{\boldsymbol{u}}} \rho^{\boldsymbol{u}}, \tag{4.22}$$

which has both advection and diffusion. The corresponding necessary conditions of optimality are then transformed as follows.

Theorem 4. ***(Necessary conditions of optimality for minimum energy steering of angular velocity PDF with process noise)*** *Let $\delta > 0$. The optimal tuple $(\rho^{\mathrm{opt}}, \boldsymbol{u}^{\mathrm{opt}})$ solving problem (4.6) with $q(\cdot) \equiv 0$, $r = \frac{1}{2}\|\cdot\|_2^2$, and (4.6b) replaced by (5.1b), satisfies the following first order necessary conditions of optimality:*

$$\frac{\partial \phi}{\partial t} + \frac{1}{2}\|\boldsymbol{\beta} \odot \nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \phi\|_2^2 + \langle \nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \phi, \boldsymbol{\alpha} \odot \boldsymbol{f}(\boldsymbol{x}^{\boldsymbol{u}}) \rangle = -\delta \Delta_{\boldsymbol{x}^{\boldsymbol{u}}} \phi, \tag{4.23a}$$

$$\frac{\partial \rho^{\mathrm{opt}}}{\partial t} + \nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \cdot \left(\rho^{\mathrm{opt}} \big(\boldsymbol{\alpha} \odot \boldsymbol{f}(\boldsymbol{x}^{\boldsymbol{u}}) + \boldsymbol{\beta}^2 \odot \nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \phi\big)\right) = \delta \Delta_{\boldsymbol{x}^{\boldsymbol{u}}} \rho^{\mathrm{opt}}, \tag{4.23b}$$

$$\rho^{\mathrm{opt}}(\boldsymbol{x}^{\boldsymbol{u}}, t = 0) = \rho_0, \quad \rho^{\mathrm{opt}}(\boldsymbol{x}^{\boldsymbol{u}}, t = T) = \rho_T, \tag{4.23c}$$

$$\boldsymbol{u}^{\mathrm{opt}} = \boldsymbol{\beta} \odot \nabla_{\boldsymbol{x}^{\boldsymbol{u}}} \phi, \tag{4.23d}$$

where $\boldsymbol{\beta}^2$ denotes the vector element-wise square.

Proof. The proof follows the same line of arguments, mutatis mutandis, as in the proof of Theorem 3. See also [9, proof of Prop. 1], [19, p. 275]. ■

Remark 2. *In the limit $\delta \downarrow 0$, the conditions* (4.23) *reduce to the conditions* (4.10)*.*

While (4.23) is valid for arbitrary (not necessarily small) $\delta > 0$, we are particularly interested in numerically solving (4.23) for small δ since then, its solution is guaranteed [53,57] to approximate the solution of (4.10). Indeed, the second order terms in (4.23) contribute toward smoother numerical solutions, i.e., behave as stochastic dynamic regularization in a computational sense. This idea of leveraging the stochastic version of the OMT for approximate numerical solution of the corresponding deterministic dynamic OMT has appeared, e.g., in [32].

4.5.3 Solving the Conditions of Optimality using A Modified Physics Informed Neural Network

We propose leveraging recent advances in neural network-based computational frameworks to numerically solve (4.23) for small $\delta > 0$. Specifically, we propose training a modified physics informed neural network (PINN) [55,67] to numerically solve (4.23a)-(4.23c), which is a system of two second order coupled PDEs together with the endpoint PDF boundary conditions.

We point out here that one can alternatively use the Hopf-Cole [20, 42] a.k.a. Fleming's logarithmic transform [30] to rewrite the system (4.23a)-(4.23c) into a system of forward-backward Kolmogorov PDEs with the unknowns being the so-called "Schrödinger factors". Unlike (4.23), these PDEs are coupled via nonlinear boundary conditions; see e.g., [9, Sec. II, III.B], [19, Sec. 5]. However, the

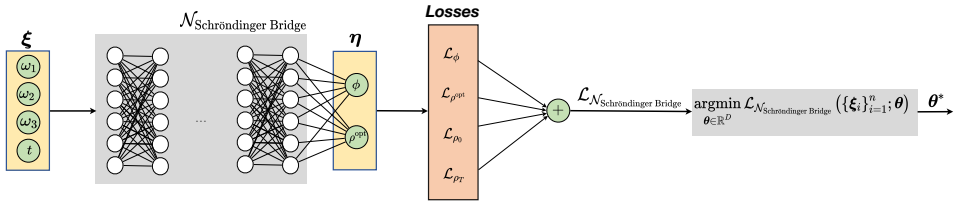

Figure 4.1: The architecture of the PINN with $\boldsymbol{\xi} := (\omega_1, \omega_2, \omega_3, t)$ as the input features. The PINN output $\boldsymbol{\eta}$ comprises of the value function and the optimally controlled PDF, i.e., $\boldsymbol{\eta} := (\phi, \rho^{\text{opt}})$.

numerical solution of the resulting system is then contingent on the availability of two initial value problem solvers: one for the forward Kolmogorov PDE and another for the backward Komogorov PDE. While specialized solvers may be designed for certain classes of prior nonlinear drifts [9, 10], in general one resorts to particle-based methods such as Monte Carlo and Feynman-Kac solvers. Directly solving the conditions of optimality by adapting PINNs, as pursued here, offers an alternative computational method.

As in [61], our training of PINN in this work involves minimizing a sum of four losses: two losses encoding the equation errors in (4.23a)-(4.23b), and the other two encoding the boundary condition errors in (4.23c). However, different from [61], we penalize the boundary condition losses using the discrete version of the Sinkhorn divergence (3.1) computed using contractive Sinkhorn iterations [22].

Because the Sinkhorn iterations involve a sequence of differentiable linear operations, it is Pytorch auto-differentiable to support backpropagation. Compared to the computationally demanding task of differentiating through a large linear program involving the Wasserstein losses, the Sinkhorn losses for the endpoint boundary conditions offer approximate solutions with far less computational cost allowing us to train the PINN on nontrivial problems.

The proposed architecture of the PINN is shown in Fig. 5.1. In our

problem, $\boldsymbol{\xi} := (\omega_1, \omega_2, \omega_3, t)$ comprises the features given to the PINN, and the PINN output $\boldsymbol{\eta} := (\phi, \rho^{\text{opt}})$. We parameterize the output of the fully connected feed-forward network via $\boldsymbol{\theta} \in \mathbb{R}^D$, i.e.,

$$\boldsymbol{\eta}(\boldsymbol{\xi}) \approx \mathcal{N}_{\text{Schrödinger Bridge}}(\boldsymbol{\xi}; \boldsymbol{\theta}), \tag{4.24}$$

where $\mathcal{N}_{\text{Schrödinger Bridge}}(\cdot; \boldsymbol{\theta})$ denotes the neural network approximant parameterized by $\boldsymbol{\theta}$, and D is the dimension of the parameter space (i.e., the total number of to-be-trained weight, bias and scaling parameters for the network).

The overall loss function for the network denoted as $\mathcal{L}_{\mathcal{N}_{\text{Schrödinger Bridge}}}$, consists of the sum of the equation error losses and the losses associated with the boundary conditions. Specifically, let $\mathcal{L}_\phi$ be the mean squared error (MSE) loss term for the HJB PDE (4.23a), and let $\mathcal{L}_{\rho^{\text{opt}}}$ be the MSE loss term for the FPK PDE (4.23b). For (4.23c), we consider Sinkhorn regularized losses $\mathcal{L}_{\rho_0}$ and $\mathcal{L}_{\rho_T}$. Then,

$$\mathcal{L}_{\mathcal{N}_{\text{Schrödinger Bridge}}} := \mathcal{L}_\phi + \mathcal{L}_{\rho^{\text{opt}}} + \mathcal{L}_{\rho_0} + \mathcal{L}_{\rho_T}, \tag{4.25}$$

where each summand loss term in (5.17) is evaluated on a set of n collocation points $\{\boldsymbol{\xi}_i\}_{i=1}^n$ in the domain of the feature space $\Omega := \mathcal{X} \times [0, T]$ for some $\mathcal{X} \subset \mathbb{R}^3$, i.e., $\{\boldsymbol{\xi}_i\}_{i=1}^n \subset \Omega$.

We train the PINN with a Pytorch backend to compute the optimal training parameter

$$\boldsymbol{\theta}^* := \underset{\boldsymbol{\theta} \in \mathbb{R}^D}{\operatorname{argmin}}\, \mathcal{L}_{\mathcal{N}_{\text{Schrödinger Bridge}}}(\{\boldsymbol{\xi}_i\}_{i=1}^n; \boldsymbol{\theta}). \tag{4.26}$$

In the next Section, we detail the simulation setup and report the numerical results.

We consider the stochastic dynamics (4.21) with $\delta = 0.1$. The vector field $\boldsymbol{f} : \mathbb{R}^3 \mapsto \mathbb{R}^3$ is given in (4.3). For the parameter vectors in (4.4), we consider

$J_1 = 0.45$, $J_2 = 0.50$, and $J_3 = 0.55$.

The control objective is to steer the prescribed joint PDF of the initial condition $\boldsymbol{x}(t=0) \sim \rho_0 = \mathcal{N}(\boldsymbol{m}_0, \boldsymbol{\Sigma}_0)$ to the prescribed joint PDF of the terminal condition $\boldsymbol{x}(t=T) \sim \rho_T = \mathcal{N}(\boldsymbol{m}_T, \boldsymbol{\Sigma}_T)$ over $t \in [0, T]$, subject to (4.21), while minimizing (4.6a) with $q(\cdot) \equiv 0$, $r = \frac{1}{2}\|\cdot\|_2^2$. Here, we fix the final time $T = 4$ s, and

$$\boldsymbol{m}_0 = (2,2,2)^\top,\ \boldsymbol{m}_T = (0,0,0)^\top,\ \boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_T = 0.5\boldsymbol{I}_3.$$

Due to the prior nonlinear drift, the optimally controlled transient joint state PDFs are expected to be non-Gaussian even when the endpoint joint state PDFs are Gaussian.

For training the $\mathcal{N}_{\text{Schrödinger Bridge}}$, we use a network with 3 hidden layers with 70 neurons in each layer. The activation functions are chosen to be $\tanh(\cdot)$. The input-output structure of the network is as explained in Sec. 5.3.1.

We fix the state-time collocation domain $\Omega = \mathcal{X} \times [0, T] = [-5, 5]^3 \times [0, 4]$. We trained the PINN for 80,000 epochs with the Adam optimizer [47] and with a learning rate 10^{-3}. We used $n = 100,000$ pseudorandom samples (using Hammersley distribution) between the endpoint boundary conditions at $t = 0$ and $t = T$ for the training. Additionally, to satisfy compute constraints, we uniformly randomly sampled 35,000 samples every 40,000 epochs. For computing the Sinkhorn losses at the endpoint boundary conditions, we use the entropic regularization parameter (see (3.1)) $\varepsilon = 0.1$.

Fig. 4.2 depicts fifty optimally controlled state sample paths for this simulation. These sample paths are obtained via closed-loop simulation with the optimal control policy $\boldsymbol{u}^{\text{opt}}$ resulting from the training of the PINN.

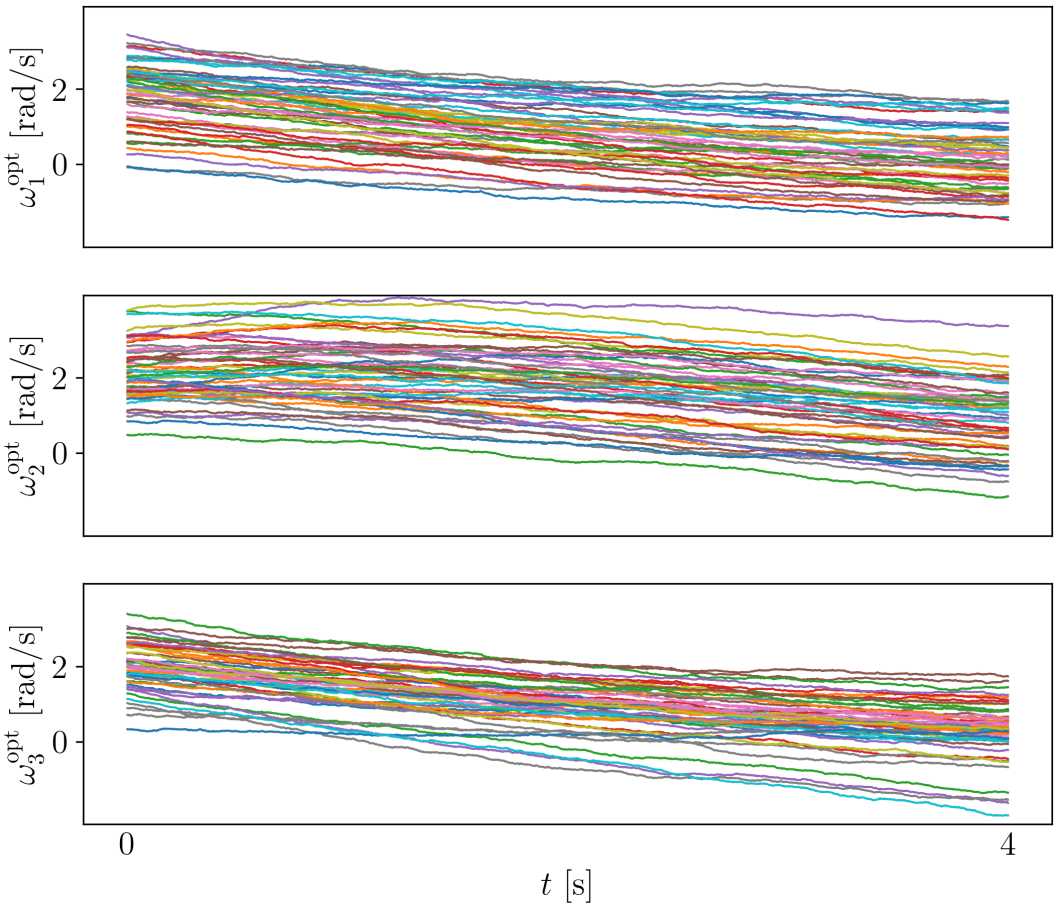

Figure 4.2: Fifty optimally controlled closed-loop state sample paths $\omega_i^{\mathrm{opt}}(t)$, $i \in [\![3]\!]$, for the simulation reported in Sec. 4.5.

Fig. 4.3 shows the snaphsots of the univariate marginal PDFs under optimal control and the same without control, for the aforesaid numerical simulation. Following Sec. 4.5.1, computing the uncontrolled PDFs for the deterministic dynamics (i.e., $\delta = 0$) requires inverting (4.17). We used the method-of-characteristics [33] to solve the corresponding unforced Liouville PDE, thereby obtaining the uncontrolled joint PDF snapshots. The marginals ρ_i^{unc}, $i \in [\![3]\!]$, in Fig. 4.3 were obtained by numerically integrating these uncontrolled joints.

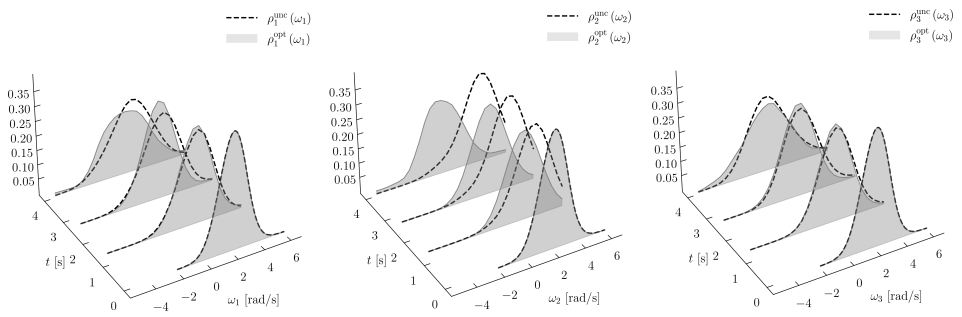

Figure 4.3: Four snapshots for the *optimally controlled* univariate marginals ρ_i^{opt} and the corresponding *uncontrolled* univariate marginals ρ_i^{unc}, $i \in [\![3]\!]$, for the numerical simulation in Sec. 4.5.

Chapter 5

Case study: Data-driven Controlled Colloidal Self-assembly

5.1 Problem Formulation

We now consider a controlled colloidal self-assembly application where the interest is to solve the *minimum energy* variant of the generalized SBP, i.e., to solve (1.13) with

$$q(\cdot) \equiv 0, \quad r(\cdot) \equiv \frac{1}{2}\|\cdot\|_2^2,$$

where the associated drift and diffusion coefficient pair $(\boldsymbol{f}, \boldsymbol{g})$ in (1.14), are not available from first principle physics, but are instead approximated in a data-driven manner (e.g., using neural networks). This is particularly challenging since both $\boldsymbol{f}$ and $\boldsymbol{g}$ can in general be non-autonomous (i.e., may have explicit t-dependence) as well as nonlinear in state, and non-affine in control.

5.2 Conditions of Optimality

In the following, we derive the first-order optimality conditions for the *minimum energy generalized SBP* as a coupled system of nonlinear PDEs.

Theorem 5. *The pair* $(\rho^{\boldsymbol{u}}_{\text{opt}}(\boldsymbol{x},t), \boldsymbol{u}_{\text{opt}}(\boldsymbol{x},t))$ *that solves* (1.13) *with* $q(\cdot) \equiv 0$ *and* $r(\cdot) \equiv \frac{1}{2}\|\cdot\|_2^2$*, must satisfy the following system of coupled PDEs*

$$\frac{\partial \psi}{\partial t} = \frac{1}{2}\|\boldsymbol{u}_{\text{opt}}\|_2^2 - \langle \nabla\psi, \boldsymbol{f}\rangle - \langle \boldsymbol{G}, \mathbf{Hess}(\psi)\rangle \tag{5.1a}$$

$$\frac{\partial \rho^{\boldsymbol{u}}_{\text{opt}}}{\partial t} = -\nabla.(\rho^{\boldsymbol{u}}_{\text{opt}}\boldsymbol{f}) + \langle \boldsymbol{G}, \mathbf{Hess}(\rho^{\boldsymbol{u}}_{\text{opt}})\rangle, \tag{5.1b}$$

$$\boldsymbol{u}_{\text{opt}} = \nabla_{\boldsymbol{u}_{\text{opt}}}\left(\langle \nabla_{\boldsymbol{x}}\psi, \boldsymbol{f}\rangle + \langle \boldsymbol{G}, \mathbf{Hess}(\psi)\rangle\right). \tag{5.1c}$$

with boundary conditions

$$\rho^{\boldsymbol{u}}_{\text{opt}}(\boldsymbol{x},0) = \rho_0, \quad \rho^{\boldsymbol{u}}_{\text{opt}}(\boldsymbol{x},T) = \rho_T, \tag{5.2}$$

where $\psi(\boldsymbol{x},t)$ *is a* $C^1(\mathbb{R}^n;\mathbb{R}_{\geq 0})$ *value function.*

Proof. Consider the Lagrangian

$$\begin{aligned}\mathcal{L}(\rho^{\boldsymbol{u}}, \boldsymbol{u}, \psi) := \int_0^T \int_{\mathbb{R}^n} \Big\{ &\frac{1}{2}\|\boldsymbol{u}(\boldsymbol{x},t)\|_2^2 \rho^{\boldsymbol{u}}(\boldsymbol{x},t) + \psi(\boldsymbol{x},t)\times \\ &\left(\frac{\partial \rho^{\boldsymbol{u}}}{\partial t} + \nabla.(\rho^{\boldsymbol{u}}\boldsymbol{f}) - \langle \boldsymbol{G}, \mathbf{Hess}\,(\rho^{\boldsymbol{u}})\rangle\right)\Big\} \mathrm{d}\boldsymbol{x}\mathrm{d}t\end{aligned} \tag{5.3}$$

where $\psi(\boldsymbol{x},t)$ is a $C^1(\mathbb{R}^n;\mathbb{R}_{\geq 0})$ Lagrange multiplier. Let

$$\mathcal{P}_{0T}(\mathbb{R}^n) := \big\{\rho(\boldsymbol{x},t) \geq 0 \mid \int_{\mathbb{R}^n} \rho \mathrm{d}\boldsymbol{x} = 1, \rho(\boldsymbol{x},t=0) = \rho_0, \quad \rho(\boldsymbol{x},t=T) = \rho_T\big\}. \tag{5.4}$$

We seek to minimize the Lagrangian over over $\mathcal{P}_{0T}(\mathbb{R}^n) \times \mathcal{U}$.

Performing integration by parts, Lagrangian (5.3) can be written as

$$\mathcal{L}(\rho^{\boldsymbol{u}}, \boldsymbol{u}, \psi) = \int_0^T \int_{\mathbb{R}^n} \left(\frac{1}{2}\|\boldsymbol{u}(\boldsymbol{x},t)\|_2^2 - \frac{\partial \psi}{\partial t} - \langle \nabla\psi, \boldsymbol{f}\rangle - \langle \boldsymbol{G}, \mathbf{Hess}(\psi)\rangle\right) \rho^{\boldsymbol{u}} \mathrm{d}\boldsymbol{x}\, \mathrm{d}t. \tag{5.5}$$

Minimizing (5.5) with respect to $\boldsymbol{u}$ for a fixed PDF ρ results (5.1c). By substituting (5.1c) in (5.5) and equating the resulting expression to zero, we get the dynamic programming equation

$$\begin{aligned}\int_0^T \int_{\mathbb{R}} \Big(\frac{1}{2} \| \nabla_{\boldsymbol{u}_{\text{opt}}} \left(\langle \nabla_{\boldsymbol{x}} \psi, \boldsymbol{f} \rangle + \langle \boldsymbol{G}, \mathbf{Hess}(\psi) \rangle \right) \|_2^2 \\ - \frac{\partial \psi}{\partial t} - \langle \nabla \psi, \boldsymbol{f} \rangle - \langle \boldsymbol{G}, \mathbf{Hess}(\psi) \rangle \Big) \rho^{\boldsymbol{u}}(x,t) \mathrm{d}x \, \mathrm{d}t = 0.\end{aligned} \tag{5.6}$$

For the above equation to be satisfied for arbitrary $\rho^{\boldsymbol{u}}$, we must have

$$\frac{\partial \psi}{\partial t} = \frac{1}{2} \| \nabla_{\boldsymbol{u}} \left(\langle \nabla_{\boldsymbol{x}} \psi, \boldsymbol{f} \rangle + \langle \boldsymbol{G}, \mathbf{Hess}(\psi) \rangle \right) \|_2^2 - \langle \nabla \psi, \boldsymbol{f} \rangle - \langle \boldsymbol{G}, \mathbf{Hess}(\psi) \rangle \tag{5.7}$$

which is the HJB PDE (5.1a). Finally, (5.1b) and (5.2) follow from (1.13b) and (1.13c), respectively. ■

Remark 3. *Unlike the conditions for optimality for control-affine SBP in Ch. 4 (see also [19, eq. (5.7)-(5.8)], [9, eq. (20)-(21)], [12, eq. (4)]) where we get two coupled PDEs, one being the HJB PDE and another being the controlled FPK PDE, the system* (5.1) *comprises of* $m+2$ *coupled PDEs where* m *is the number of control inputs. This is because the policy equation* (5.1c) *is implicit in* $\boldsymbol{u}_{\text{opt}}$*. Due to the non-affine control, we can no longer express* $\boldsymbol{u}_{\text{opt}}$ *as the scaled gradient of the value function* ψ*. Instead, we now need to solve the coupled system* (5.1)-(5.2)*.*

5.3 Numerical Results

5.3.1 Solving the Conditions for Optimality using a Modified PINN

We propose leveraging recent advances in neural network-based computational frameworks to jointly learn the solutions of (5.1)-(5.2). In the following,

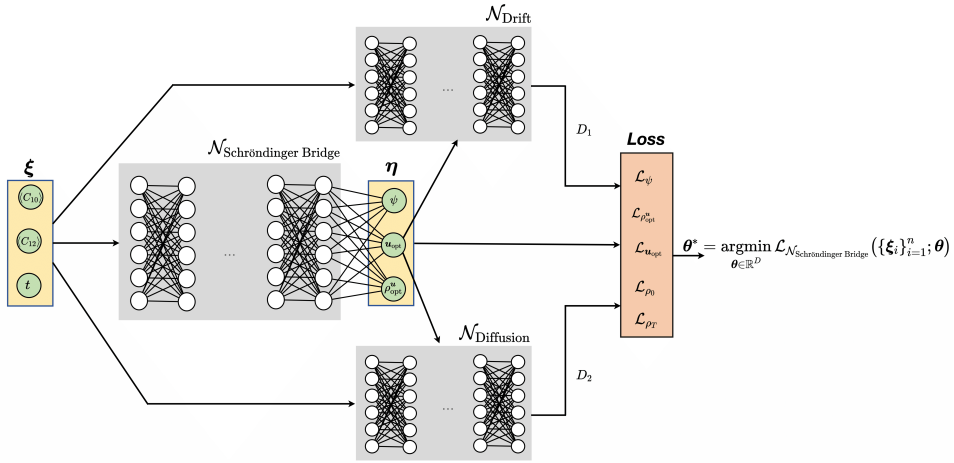

Figure 5.1: The architecture of the physics-informed neural network with the system order parameters $\langle C_{10}\rangle$, $\langle C_{12}\rangle$, and time as the input features $\boldsymbol{\xi} := (\langle C_{10}\rangle, \langle C_{12}\rangle, t)$. The output $\boldsymbol{\eta}$ comprises of the value function, optimally controlled PDF, and optimal control policy, i.e., $\boldsymbol{\eta} := (\psi, \rho^{\boldsymbol{u}}_{\text{opt}}, \boldsymbol{u}_{\text{opt}})$. The networks $\mathcal{N}_{\text{Drift}}$ and $\mathcal{N}_{\text{Diffusion}}$ are fully trained from MD simulation.

we discuss the training of a modified PINN [55,67] to numerically solve (5.1)-(5.2), which is a system of three coupled PDEs together with the endpoint PDF boundary conditions.

In this regard, we first introduce the 2-Wasserstein distance, which will play an important role in the development of the modified PINN structure.

To solve (1.7), we implement Sinkhorn iteration approach as following. We first we write the discrete version of (1.7) as

$$W^2(\rho_1, \rho_2) = \min_{\boldsymbol{M}\in\Pi(\rho_1,\rho_2)} \langle \boldsymbol{C}, \boldsymbol{M}\rangle \tag{5.8}$$

where the matrix $\boldsymbol{C} \in \mathbb{R}^{d\times d}$ is given by

$$\boldsymbol{C}(i,j) = \|\boldsymbol{x}^i - \boldsymbol{y}^j\|_2^2, \quad i,j = 1,2,\dots d \tag{5.9}$$

and $\Pi(\rho_1,\rho_2)$ stands for the set of all coupling matrices $\boldsymbol{M} \in \mathbb{R}^{d\times d}$ such that

$$\boldsymbol{M} \geq 0, \quad \boldsymbol{M}\mathbf{1} = \rho_1 \quad \boldsymbol{M}^\top \mathbf{1} = \rho_2. \tag{5.10}$$

We define the entropy of a matrix as

$$H(\boldsymbol{M}) := \langle \boldsymbol{M}, \log \boldsymbol{M} \rangle \tag{5.11}$$

and by including this entropic regularization in (5.8), we get Sinkhorn regularized squared Wasserstein distance, which encourages smoother coupling matrices

$$\begin{aligned} W_\varepsilon^2(\rho_1, \rho_2) = &\min_{\boldsymbol{M} \in \Pi(\rho_1, \rho_2)} \langle \boldsymbol{C}, \boldsymbol{M} \rangle + \varepsilon \langle \boldsymbol{M}, \log \boldsymbol{M} \rangle \\ &\text{Subject to} \quad \boldsymbol{M}\mathbf{1} = \rho_1 \quad \boldsymbol{M}^\top \mathbf{1} = \rho_2 \end{aligned} \tag{5.12}$$

where $\varepsilon > 0$ is a regularization parameter. The convex optimization problem given in (5.12) can be solved iteratively using the Sinkhorn iterations algorithm [22]. The solution can be written in the form

$$\boldsymbol{M} = \text{diag}(\boldsymbol{u}) \boldsymbol{\Gamma} \text{diag}(\boldsymbol{v}) \tag{5.13}$$

where $\boldsymbol{\Gamma} := \exp\left(\frac{-\boldsymbol{C}}{2\varepsilon}\right)$ and the iterations alternate between updating $\boldsymbol{u}$ and $\boldsymbol{v}$

$$\boldsymbol{u}^{k+1} = \rho_1 \oslash \left(\boldsymbol{\Gamma} \boldsymbol{v}^k\right) \tag{5.14}$$

$$\boldsymbol{v}^{k+1} = \rho_2 \oslash \left(\boldsymbol{\Gamma}^\top \boldsymbol{u}^{k+1}\right). \tag{5.15}$$

Computing the Wasserstein distance as a linear program is very computationally expensive, and even more so to get a differentiable distance that is trainable. Because the Sinkhorn iterations algorithm is a series of differentiable linear operations, it is Pytorch auto-differentiable to support backpropagation. The Sinkhorn algorithm offers an approximately equivalent distance for far less computational cost allowing us to train on this metric for nontrivial problems.

The proposed architecture of the PINN is shown in Fig. 5.1. In our problem, $\boldsymbol{\xi} := (\langle C_{10} \rangle, \langle C_{12} \rangle, t)$ comprises the features given to the DNN, and the DNN

output $\boldsymbol{\eta} := (\psi, \rho^{\boldsymbol{u}}_{\text{opt}}, \boldsymbol{u}_{\text{opt}})$ comprises of the value function, optimally controlled PDF, and optimal policy. We parameterize the output of the fully connected feed-forward DNN via $\boldsymbol{\theta} \in \mathbb{R}^D$, i.e.,

$$\boldsymbol{\eta}(\boldsymbol{\xi}) \approx \mathcal{N}_{\text{Schrödinger Bridge}}(\boldsymbol{\xi}; \boldsymbol{\theta}), \tag{5.16}$$

where $\mathcal{N}_{\text{Schrödinger Bridge}}(\cdot; \boldsymbol{\theta})$ denotes the neural network approximant parameterized by $\boldsymbol{\theta}$, and D is the dimension of the parameter space (i.e., the total number of to-be-trained weight, bias and scaling parameters for the DNN). As discussed in the previous sections, the explicit expression for drift and diffusion functions are not given, and we learn them from an MD simulation. After finishing the MD simulations, the output of the fully trained networks $\mathcal{N}_{\text{Drift}}$ and $\mathcal{N}_{\text{Diffusion}}$, which are the learned drift and learned diffusion functions, will be fed to the loss of $\mathcal{L}_{\mathcal{N}_{\text{Schrödinger Bridge}}}$.

The overall loss function for the network denoted as $\mathcal{L}_{\mathcal{N}_{\text{Schrödinger Bridge}}}$, consists of the sum of the losses associated with the three equations in (5.1) and the losses associated with the boundary conditions (5.2). Specifically, let $\mathcal{L}_{\psi}$ be the Mean Squared Error (MSE) loss term for the HJB PDE (5.1a), let $\mathcal{L}_{\rho^{\boldsymbol{u}}_{\text{opt}}}$ be the MSE loss term for the FPK PDE (5.1b), and because the control policy has m components $(u_1, \ldots, u_m)$, let $\mathcal{L}_{u_{j_{\text{opt}}}}|_{j=1,\ldots,m}$ be the corresponding MSE loss term for each control policy component in (5.1c). The MSE loss function is insufficient to train and solve the joint PDF boundary conditions in (5.2). So, we consider Sinkhorn regularized squared Wasserstein distance $\mathcal{L}_{\rho_0}$ and $\mathcal{L}_{\rho_T}$ for (5.2) and implement the Sinkhorn iteration algorithm to obtain the Sinkhorn regularized squared

Wasserstein distance for the boundary conditions (5.2). Then,

$$\mathcal{L}_{\mathcal{N}_{\text{Schrödinger Bridge}}} := \mathcal{L}_{\psi} + \mathcal{L}_{\rho^{\boldsymbol{u}}_{\text{opt}}} + \mathcal{L}_{u_{j_{\text{opt}}}}\,|_{j=1,\ldots,m} + \mathcal{L}_{\rho_0} + \mathcal{L}_{\rho_T}, \tag{5.17}$$

where each summand loss term in (5.17) is evaluated on a set of n collocation points $\{\boldsymbol{\xi}_i\}_{i=1}^n$ in the domain of the feature space $\Omega := [0,1]^2 \times [0,T]$, i.e., $\{\boldsymbol{\xi}_i\}_{i=1}^n \subset \Omega$, and

$$\mathcal{L}_{\psi} := \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\partial\psi}{\partial t}\bigg|_{\boldsymbol{\xi}_i} - \frac{1}{2}\|\boldsymbol{u}_{\text{opt}}\|_2^2\bigg|_{\boldsymbol{\xi}_i} + \langle\nabla\psi, D_1\rangle|_{\boldsymbol{\xi}_i} + \langle D_2, \mathbf{Hess}(\psi)\rangle|_{\boldsymbol{\xi}_i}\right)^2,$$

$$\mathcal{L}_{\rho^{\boldsymbol{u}}_{\text{opt}}} := \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\partial\rho^{\boldsymbol{u}}_{\text{opt}}}{\partial t}\bigg|_{\boldsymbol{\xi}_i} + \nabla.(\rho^{\boldsymbol{u}}_{\text{opt}} D_1)\big|_{\boldsymbol{\xi}_i} - \langle D_2, \mathbf{Hess}(\rho^{\boldsymbol{u}}_{\text{opt}})\rangle\big|_{\boldsymbol{\xi}_i}\right)^2,$$

$$\mathcal{L}_{u_{j_{\text{opt}}}}\,|_{j=1,\ldots,m} := \frac{1}{n}\sum_{i=1}^{n}\left(u_{j_{\text{opt}}}\big|_{\boldsymbol{\xi}_i} - \frac{\partial}{\partial u_{j_{\text{opt}}}}\left(\langle\nabla_{\boldsymbol{x}}\psi, D_1\rangle + \langle D_2, \mathbf{Hess}(\psi)\rangle\right)|_{\boldsymbol{\xi}_i}\right)^2,$$

$$\mathcal{L}_{\rho_0} := W_{\varepsilon}^2\left(\rho^{\boldsymbol{u}}_{\text{opt}}(\boldsymbol{x}, t=0), \rho_0\right),$$

$$\mathcal{L}_{\rho_T} := W_{\varepsilon}^2\left(\rho^{\boldsymbol{u}}_{\text{opt}}(\boldsymbol{x}, t=T), \rho_T\right)$$

for each collocation point $\boldsymbol{\xi}_i$, $i = 1, \ldots, n$. Since PINN's activations function is tanh, its output tensors could be negative, positive, trivial, take on any distribution shape, and may not be a valid PDF during training. Therefore, computing a distance measure between such an output can result in numerical singularities or non-differentiable distances, or local minima distances that jeopardize overall PINN

training. To avoid this problem in Sinkhorn iteration implementation, we use the log-sum-exp (LSE) technique to maintain numerical stability during iteration irrespective of the input numerical properties at the expense of some memory overhead costs. We expose ρ_0 and ρ_T to the network during training, and for every epoch, we compute the Sinkhorn regularized squared Wasserstein distance respective network output to these two distributions. To further reduce memory footprint, we practice mini-batching, where we sample our output and use the same sample indices to sample our desired boundary distributions. We compute the matrix C (5.9) from the output batch points. While the LSE technique is a good way to guarantee numerical stability for arbitrary neural network output distributions, it does not guarantee that the computed distance will not be a local minimum if the input distributions are not valid PDFs. It is still necessary to introduce another term to the loss function to penalize invalid distribution outputs, specifically negative outputs. There are many possible ways to ask this as a loss function, and one 'convex' way that performs well is to express it as the negative sum of all the outputs that violate it. We also weight this loss component by a scalar multiplier (of 10) so that the network penalizes violating this constraint more and thus learns it 'before' learning the Sinkhorn regularized squared Wasserstein distance. Another successful 'convex' ways of asking for network output include raising a sum to the power of 2 and using the count of violations as a loss is not 'convex' and confuses training. Finally, we found in our numerical experiments that constraining the boundary condition distributions to have a sum or trapze of 1 to satisfy the PDF definition yields a network that scales down the control magnitudes to trivial ranges and failed closed-loop simulations. Such an outcome suggests some implicit relationship between the

control magnitude and the ρ_{opt} equation and that Sinkhorn regularized squared Wasserstein distance similarity is more necessary for reasonable optimal transport control than strictly satisfying PDF definitions.

We used the PINN software library with a Pytorch backend to perform numerical experiments using the above loss functions. The PINN library was not written for optimal transport-type problems, so we needed to modify it to suit our needs. One modification was to program PINN to compute loss between outputs and distributions directly and integrate the Sinkhorn iteration algorithm into the library. We also had to modify it to perform the mini-batching we needed. In summary, for training the PINN, we minimize the overall loss (5.17) over $\boldsymbol{\theta} \in \mathbb{R}^D$ by solving

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}\in\mathbb{R}^D}{\operatorname{argmin}} \mathcal{L}_{\mathcal{N}_{\text{Schrödinger Bridge}}}(\{\boldsymbol{\xi}_i\}_{i=1}^n; \boldsymbol{\theta}). \tag{5.18}$$

Next we detail the simulation setup and report the numerical results.

5.3.2 Numerical Case Study of Isotropic Colloids in an NPT ensemble

System Description

We consider a *in-silico* representation of a system of isotropic colloids with Lennard-Jones interaction potentials

$$U(r) = 4\epsilon((\frac{\sigma}{r})^{12} - (\frac{\sigma}{r})^6) \tag{5.19}$$

in an NPT ensemble. In this system, 2048 particles with identical ϵ and σ are initialized at a temperature and stress value. The system can be controlled with a linear temperature or pressure ramp.

$\langle C_{10}\rangle, \langle C_{12}\rangle$ denote the Steinhart bond order parameters defined below

$$C(i) = \left(\frac{4\pi}{2l+1} \sum_{m=-l}^{l} |C_{lm}(i)|^2 \right)^{\frac{1}{2}} \tag{5.20}$$

where $C_{lm}(i)$

$$C_{l\,m}(i) = \frac{1}{N(i)} \sum_{j=1}^{N(i)} Y_{lm}\left(r_{i\,j}\right) \tag{5.21}$$

Y_{lm} represents the spherical harmonic and $N(i)$ is the number of neighbors. The subscript in the order parameters correspond to the quantum number used in defining the spherical harmonics ($\langle C_{10}\rangle$ would correspond to 10 fold ordering). $\langle C_{10}\rangle, \langle C_{12}\rangle$ were chosen due to their ability to differentiate between BCC and FCC structures. The simulations used to train $\mathcal{N}_{\text{Drift}}$ and $\mathcal{N}_{\text{Diffusion}}$ were trajectories with a batch size of 10^5, sampled 500 times per trajectory. Alternative sampling intervals were tested with a sensitivity analysis prior to selecting 500. Temperature and stress ramp rate combinations were sampled with a hypercube distribution. MD simulations were conducted in the python package HOOMD-blue.

$$\boldsymbol{x} := (\langle C_{10}\rangle, \langle C_{12}\rangle) \in [0,1]^2$$

$$\boldsymbol{u} = (u_1, u_2) := (\text{temperature ramp}, \text{stress ramp})$$

Learning the Dynamics

$\mathcal{N}_{\text{Drift}}$ and $\mathcal{N}_{\text{Diffusion}}$ were each modeled with a neural network trained using the python package torchsde, a stochastic differential equation solver. The input to the model was the current system state (value of $\boldsymbol{x}$ at the start of the MD simulation trajectory), and the temperature and stress ramps ($\boldsymbol{u}$). The model was then trained to predict the evolution of $\boldsymbol{x}$ over the trajectory. 5-fold cross validation was used to train the model.

Controller Synthesis

We consider the self-assembly mechanism of colloidal particles is given by (4.21),

D_1, D_2 are the outputs of neural network representations $\mathcal{N}_{\text{Drift}}$ and $\mathcal{N}_{\text{Diffusion}}$, respectively, and learnt from the high fidelity MD simulation data.

The control objective is to steer the prescribed joint PDF of the initial condition $\boldsymbol{x}(t=0) \sim \rho_0 = \mathcal{N}(\boldsymbol{m}_0, \boldsymbol{\Sigma}_0)$ to the prescribed joint PDF of the terminal condition $\boldsymbol{x}(t=T) \sim \rho_T = \mathcal{N}(\boldsymbol{m}_T, \boldsymbol{\Sigma}_T)$ (fcc point) over $t \in [0, T]$, subject to (4.21), while minimizing (**??**). Here, we fix the final time $T = 200$ s, and

$$\boldsymbol{m}_0 = (0.2, 0.2)^\top,\ \boldsymbol{m}_T = (0.4, 0.375)^\top,\ \boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_T = 0.1\boldsymbol{I}_2.$$

The domain for state-time collocation is $\Omega = [0,1]^2 \times [0, 200]$, and we trained the Physics-Informed Neural Network (PINN) for $100,000$ epochs with the Adam optimizer [47], using a learning rate of 10^{-3}. To train the model, we used $n = 3000$ pseudorandom samples, drawn using Sobol distribution, between the endpoint boundary conditions at $t = 0$ and $t = 200$. We also uniformly randomly sampled $3,000$ samples every $20,000$ epochs to satisfy compute constraints. For computing the Sinkhorn losses at the endpoint boundary conditions, we used an entropic regularization parameter of $\varepsilon = 0.1$ as given in (5.12).Fig. 5.2 shows the residuals for each loss functions given in (5.17).

Fig.5.3 shows the evolution of $\rho^{\text{opt}}(\boldsymbol{x}, t)$ over time, starting from the initial distribution and reaching the terminal distribution after 200 seconds. By comparing Fig. 5.4 and Fig. 5.5 with Fig. 5.6, it becomes clear that the controls are high (low) in regions where the value function ψ changes rapidly (when the gradient of ψ is large/small).

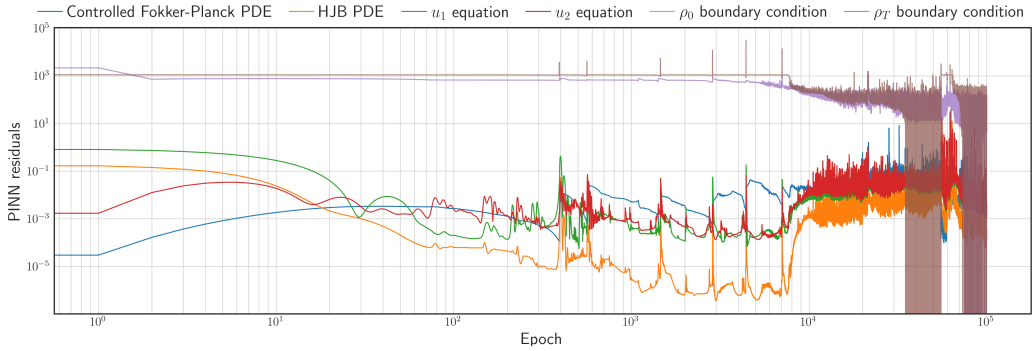

Figure 5.2: The PINN residuals for solving the conditions for optimality (5.1)-(5.2).

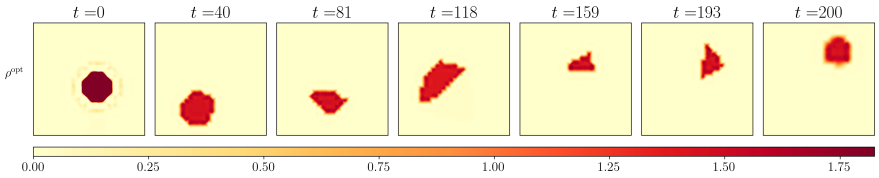

Figure 5.3: Contour plots of the optimally controlled state PDFs $\rho^{\text{opt}}(\boldsymbol{x}, t)$ over $[0, 1]^2$.

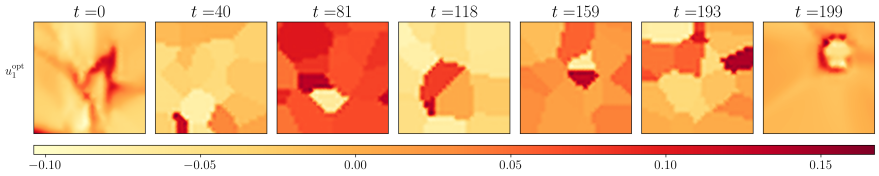

Figure 5.4: Contour plots of the magnitude of the optimal control $u_1^{\text{opt}}(\boldsymbol{x}, t)$ over $[0, 1]^2$.

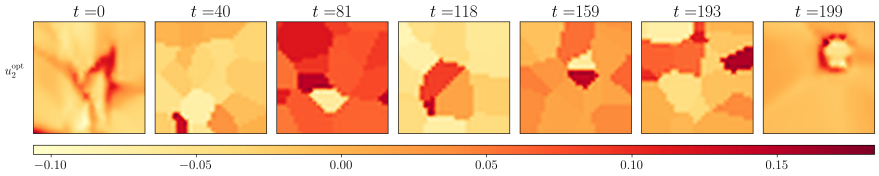

Figure 5.5: Contour plots of the magnitude of the optimal control $u_2^{\text{opt}}(\boldsymbol{x}, t)$ over $[0, 1]^2$.

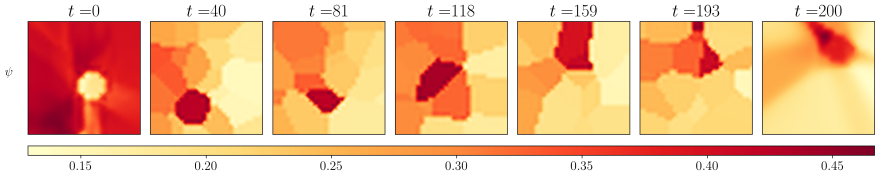

Figure 5.6: Contour plots of the value function $\psi(\boldsymbol{x}, t)$ over $[0, 1]^2$.

Figure 5.7: Simulation results for the optimal PDF steering for the self-assembly mechanism of colloidal particles over $t \in [0, 200]$. The color denotes the value of the plotted variable; see colorbar (dark red = high, light yellow = low).

Chapter 6

Conclusions

In this thesis, we present a solution to a specific instance of the generalized Schrodinger Bridge Problem (SBP) with control cost $r(\cdot) := \frac{1}{2}\|\cdot\|_2^2, q \equiv 0$, the minimum energy generalized SBP. We cover the existence-uniqueness of such problems. We prove the conditions for optimality for such systems using integration by parts on the Lagrangians, minimizing with respect to u and performing substitution and equating the expressions to zero to yield dyanmic programing expressions. These operations yield the system of coupled partial differential equations (PDEs) that must be satisfied for $(\rho^{\boldsymbol{u}}_{\text{opt}}, \boldsymbol{u}_{\text{opt}})$

We propose using Physics-Informed Neural Networks (PINNs) to solve these PDE systems, and provide the architectures, loss functions, and implementation details for 2 case studies. Whereas prior work has used mean squared error (MSE) loss functions, those loss functions do not respect the geometry necessary for outputs, such as in this case that the boundary conditions at time T_0, T_t must be probability density functions (PDFs). After some experimentation we confirm that computing the direct squared Wasserstein-2 distance metric (from classical opti-

mal mass transport (OMT)) between the network output and the desired boundary condition PDFs with backpropable gradients is computationally intractable for non-trivial problems.

Instead we use an entropy-regularized version of the Wasserstein metric we refer to as *Sinkhorn divergence* which as $\epsilon \downarrow 0$ converges to the classical Wasserstein metric. Such a divergence metric can be computed using *Sinkhorn iterations* in a backpropable, and numerically stable, implementation, making the entire system computationally tractable for PINN to solve.

We demonstrate these techniques on 2 case distribution steering problems, the first where the state is angular velocities for a rigid body, the second states for colloidal self-assembly. In the first case study, we derive the conditions for optimality and present a system of 2 coupled PDE equations (FPK and HJB PDEs). We show a sufficient neural network architecture, loss functions, and numerical experiment. The numerical experiment is non-trivial: it is 3 dimensional, the system dynamics are nonlinear, the inertias are non-axis-symmetric. We also performed post-processing, and applied the learned control policy to a closed-loop simulation and confirmed it could steer a Gaussian distribution of 50 samples at a given initial PDF to a desired PDF. In the second case study, we had a different system of 3 coupled PDEs and different architecture. Here, we use the same Sinkhorn iteration technique to train for the boundary conditions. Like in the first case study, we provide conditions for optimality and define loss functions and implementation details. Unlike in the first case study where the system dynamics were well defined equations, here we have data-driven system dynamics encoded as frozen neural nets themselves. Still we show that PINN can be trained to learn to steer a distribution.

During our experiments it was clear that PINN could achieve low residual values without providing a good control policy (for example when there are too many hidden layers, or when the training distribution was sufficiently dense), or that PINN could not converge on a good policy when the proposed problem was not solvable. Future work could investigate this further, to get a clearer understanding of what numerical and implementation details are necessary for BOTH good loss function results and good closed loop control policies. Another interesting finding is that clamping the HJB PDEs in both cases resulted in a *trivial* control policy, and removing any numerical upper bound limitation on HJB PDE net output resulted in good control policies, so it is important to state that the network outputs are not *strict PDFs / PMFs*. Future work might revisit why this is another necessary condition for good training to yield good control policies. In addition, future work might extend to more colorful boundary condition distributions such as Gaussian mixtures, or explore how feasible these methods are for different nonlinear system dynamics and problem constraints.

Bibliography

[1] M Athans, P Falb, and R Lacoss. Time-, fuel-, and energy-optimal control of nonlinear norm-invariant systems. *IEEE transactions on automatic control*, 8(3):196–202, 1963.

[2] John Baillieul. The geometry of homogeneous polynomial dynamical systems. *Nonlinear Analysis: Theory, Methods & Applications*, 4(5):879–900, 1980.

[3] Kaivalya Bakshi, David D Fan, and Evangelos A Theodorou. Schrödinger approach to optimal control of large-size populations. *IEEE Transactions on Automatic Control*, 66(5):2372–2378, 2020.

[4] Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

[5] Arne Beurling. An automorphism of product measures. *Annals of Mathematics*, pages 189–200, 1960.

[6] Yann Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on pure and applied mathematics*, 44(4):375–417, 1991.

[7] Roger W Brockett. Nonlinear systems and differential geometry. *Proceedings of the IEEE*, 64(1):61–72, 1976.

[8] Peter J Bushell. Hilbert's metric and positive contraction mappings in a Banach space. *Archive for Rational Mechanics and Analysis*, 52(4):330–338, 1973.

[9] Kenneth Caluya and Abhishek Halder. Wasserstein proximal algorithms for the Schrödinger bridge problem: Density control with nonlinear drift. *IEEE Transactions on Automatic Control*, 2021.

[10] Kenneth F Caluya and Abhishek Halder. Gradient flow algorithms for density propagation in stochastic systems. *IEEE Transactions on Automatic Control*, 65(10):3991–4004, 2019.

[11] Kenneth F Caluya and Abhishek Halder. Finite horizon density steering for multi-input state feedback linearizable systems. In *2020 American Control Conference (ACC)*, pages 3577–3582. IEEE, 2020.

[12] Kenneth F Caluya and Abhishek Halder. Reflected Schrödinger bridge: Density control with path constraints. In *2021 American Control Conference (ACC)*, pages 1137–1142. IEEE, 2021.

[13] Tianrong Chen, Guan-Horng Liu, and Evangelos A Theodorou. Likelihood training of Schrödinger bridge using forward-backward SDEs theory. *arXiv preprint arXiv:2110.11291*, 2021.

[14] Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. Optimal steering of a linear stochastic system to a final probability distribution, part i. *IEEE Transactions on Automatic Control*, 61(5):1158–1169, 2015.

[15] Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. Entropic and displacement interpolation: a computational approach using the Hilbert metric. *SIAM Journal on Applied Mathematics*, 76(6):2375–2396, 2016.

[16] Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. On the relation between optimal transport and Schrödinger bridges: A stochastic control viewpoint. *Journal of Optimization Theory and Applications*, 169:671–691, 2016.

[17] Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. Optimal transport over a linear dynamical system. *IEEE Transactions on Automatic Control*, 62(5):2137–2152, 2016.

[18] Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. Controlling uncertainty. *IEEE Control Systems Magazine*, 41(4):82–94, 2021.

[19] Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. Stochastic control liaisons: Richard Sinkhorn meets Gaspard Monge on a Schrodinger bridge. *Siam Review*, 63(2):249–313, 2021.

[20] Julian D Cole. On a quasi-linear parabolic equation occurring in aerodynamics. *Quarterly of Applied Mathematics*, 9(3):225–236, 1951.

[21] Peter Crouch. Spacecraft attitude control and stabilization: Applications of geometric control theory to rigid body models. *IEEE Transactions on Automatic Control*, 29(4):321–331, 1984.

[22] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[23] Inc. CVX Research. CVX: Matlab software for disciplined convex programming, version 2.0. `http://cvxr.com/cvx`, August 2012.

[24] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion Schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.

[25] Cynthia Dwork. Differential privacy. In *International Colloquium on Automata, Languages, and Programming*, pages 1–12. Springer, 2006.

[26] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

[27] Thomas Dwyer. The control of angular momentum for asymmetric rigid bodies. *IEEE Transactions on Automatic Control*, 27(3):686–688, 1982.

[28] Karthik Elamvazhuthi, Piyush Grover, and Spring Berman. Optimal transport over deterministic discrete-time nonlinear systems using stochastic feedback laws. *IEEE control systems letters*, 3(1):168–173, 2018.

[29] Alessio Figalli. *Optimal transportation and action-minimizing measures*. PhD thesis, Lyon, École normale supérieure (sciences), 2007.

[30] Wendell H Fleming. Logarithmic transformations and stochastic control. In *Advances in Filtering and Optimal Stochastic Control*, pages 131–141. Springer, 1982.

[31] Robert Fortet. Résolution dâun systeme dâéquations de m. Schrödinger. *J. Math. Pure Appl. IX*, 1:83–105, 1940.

[32] Shadi Haddad, Kenneth F Caluya, Abhishek Halder, and Baljeet Singh. Prediction and optimal feedback steering of probability density functions for safe automated driving. *IEEE Control Systems Letters*, 5(6):2168–2173, 2020.

[33] Abhishek Halder and Raktim Bhattacharya. Dispersion analysis in hypersonic flight during planetary entry using stochastic Liouville equation. *Journal of Guidance, Control, and Dynamics*, 34(2):459–474, 2011.

[34] Abhishek Halder and Raktim Bhattacharya. Model validation: A probabilistic formulation. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 1692–1697. IEEE, 2011.

[35] Abhishek Halder and Raktim Bhattacharya. Further results on probabilistic model validation in Wasserstein metric. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 5542–5547. IEEE, 2012.

[36] Abhishek Halder and Raktim Bhattacharya. Geodesic density tracking with applications to data driven modeling. In *2014 American Control Conference*, pages 616–621. IEEE, 2014.

[37] Abhishek Halder and Raktim Bhattacharya. Probabilistic model validation for uncertain nonlinear systems. *Automatica*, 50(8):2038–2050, 2014.

[38] Abhishek Halder and Tryphon T Georgiou. Gradient flows in uncertainty propagation and filtering of linear Gaussian systems. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 3081–3088. IEEE, 2017.

[39] Abhishek Halder, Kooktae Lee, and Raktim Bhattacharya. Optimal transport approach for probabilistic robustness analysis of F-16 controllers. *Journal of Guidance, Control, and Dynamics*, 38(10):1935–1946, 2015.

[40] Abhishek Halder and Eric DB Wendel. Finite horizon linear quadratic gaussian density regulator with Wasserstein terminal cost. In *2016 American Control Conference (ACC)*, pages 7249–7254. IEEE, 2016.

[41] David Hilbert. Über die gerade linie als kürzeste verbindung zweier punkte. *Mathematische Annalen*, 46(1):91–96, 1895.

[42] Eberhard Hopf. The partial differential equation $u_t + uu_x = \mu_{xx}$. *Communications on Pure and Applied mathematics*, 3(3):201–230, 1950.

[43] SafeGraph Inc. `https://www.safegraph.com/`.

[44] Kaito Ito and Kenji Kashima. Sinkhorn MPC: Model predictive optimal transport over dynamical systems. In *2022 American Control Conference (ACC)*, pages 2057–2062. IEEE, 2022.

[45] Benton Jamison. The Markov processes of Schrödinger. *Probability Theory and Related Fields*, 32(4):323–331, 1975.

[46] Leonid V Kantorovich. On the translocation of masses. In *Dokl. Akad. Nauk. USSR (NS)*, volume 37, pages 199–201, 1942.

[47] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[48] KSP Kumar. On the optimum stabilization of a satellite. *IEEE Transactions on Aerospace and Electronic Systems*, (2):82–83, 1965.

[49] L.D. Landau and E.M. Lifshitz. *Course of Theoretical Physics. Volume 1: Mechanics*. Butterworth-Heinemann, 3 edition, 1976.

[50] John M Lee. *Introduction to Riemannian manifolds*. Springer, 2018.

[51] Kooktae Lee, Abhishek Halder, and Raktim Bhattacharya. Performance and robustness analysis of stochastic jump linear systems using Wasserstein metric. *Automatica*, 51:341–347, 2015.

[52] Yin Tat Lee and Aaron Sidford. Path finding methods for linear programming: Solving linear programs in $\tilde{O}(\sqrt{\text{rank}})$ iterations and faster algorithms for maximum flow. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 424–433. IEEE, 2014.

[53] Christian Léonard. From the Schrödinger problem to the Monge–Kantorovich problem. *Journal of Functional Analysis*, 262(4):1879–1920, 2012.

[54] Yong Lin, Linyuan Lu, and Shing-Tung Yau. Ricci curvature of graphs. *Tohoku Mathematical Journal, Second Series*, 63(4):605–627, 2011.

[55] Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. Deepxde: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228, 2021.

[56] Robert J McCann. A convexity principle for interacting gases. *Advances in mathematics*, 128(1):153–179, 1997.

[57] Toshio Mikami. Monge's problem with a quadratic cost by the zero-noise limit of h-path processes. *Probability theory and related fields*, 129(2):245–260, 2004.

[58] Dejan Milutinović and Pedro Lima. Modeling and optimal centralized control of a large-size robotic population. *IEEE Transactions on Robotics*, 22(6):1280–1285, 2006.

[59] Dejan Lj Milutinović and Pedro U Lima. *Cells and robots: modeling and control of large-size agent populations*. Springer, 2007.

[60] Iman Nodozi and Abhishek Halder. Schrödinger meets kuramoto via feynman-kac: Minimum effort distribution steering for noisy nonuniform kuramoto oscillators. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 2953–2960. IEEE, 2022.

[61] Iman Nodozi, Jared O'Leary, Ali Mesbah, and Abhishek Halder. A physics-informed deep learning approach for minimum effort stochastic control of colloidal self-assembly. *arXiv preprint arXiv:2208.09182*, 2022.

[62] Yann Ollivier. Ricci curvature of Markov chains on metric spaces. *Journal of Functional Analysis*, 256(3):810–864, 2009.

[63] Yann Ollivier. A survey of ricci curvature for metric spaces and markov chains. In *Probabilistic approach to geometry*, pages 343–381. Mathematical Society of Japan, 2010.

[64] Aaron Palmer and Dejan Milutinović. A hamiltonian approach using partial differential equations for open-loop stochastic optimal control. In *Proceedings of the 2011 American Control Conference*, pages 2056–2061. IEEE, 2011.

[65] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and trends® in Optimization*, 1(3):127–239, 2014.

[66] Gabriel Peyré and Marco Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[67] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[68] Areejit Samal, R.P. Sreejith, Jiao Gu, Shiping Liu, Emi Saucan, and Jürgen Jost. Comparative analysis of two discretizations of Ricci curvature for complex networks. *Scientific Reports*, 8(8560):2–3, 2018.

[69] Romeil Sandhu, Tryphon Georgiou, Ed Reznik, Liangjia Zhu, Ivan Kolesov, Yasin Senbabaoglu, and Allen Tannenbaum. Graph curvature for differentiating cancer networks. *Scientific reports*, 5(1):1–13, 2015.

[70] Romeil S Sandhu, Tryphon T Georgiou, and Allen R Tannenbaum. Ricci curvature: An economic indicator for market fragility and systemic risk. *Science advances*, 2(5):e1501495, 2016.

[71] Erwin Schrödinger. Über die umkehrung der naturgesetze. *Sitzungsberichte der Preuss. Phys. Math. Klasse*, 10:144–153, 1931.

[72] Erwin Schrödinger. Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. In *Annales de l'institut Henri Poincaré*, volume 2, pages 269–310, 1932.

[73] Panagiotis Tsiotras, Martin Corless, and Mario Rotea. Optimal control of rigid body angular velocity with quadratic cost. *Journal of optimization theory and applications*, 96:507–532, 1998.

[74] Cédric Villani. *Topics in optimal transportation*. Number 58. American Mathematical Soc., 2003.

[75] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.

[76] A Wakolbinger. Schrödinger bridges from 1931 to 1991. In *Proc. of the 4th Latin American Congress in Probability and Mathematical Statistics, Mexico City*, pages 61–79, 1990.

[77] Chi Wang, Edmond Jonckheere, and Reza Banirazi. Wireless network capacity versus Ollivier-Ricci curvature under heat-diffusion (HD) protocol. In *2014 American Control Conference*, pages 3536–3541. IEEE, 2014.

[78] Gefei Wang, Yuling Jiao, Qian Xu, Yang Wang, and Can Yang. Deep generative learning via Schrödinger bridge. In *International Conference on Machine Learning*, pages 10794–10804. PMLR, 2021.

Dropbox Sign

Audit trail

Title	Charlie Yan MS Thesis
File name	Charlie_Yan_UCSC_MS_Thesis.pdf
Document ID	9787ca325a9b9b6b7ba442fb02cfbb86a8bf6d59
Audit trail date format	MM / DD / YYYY
Status	● Pending signature

Document History

SENT	**05 / 23 / 2023** 18:18:40 UTC	Sent for signature to Qi Gong (qigong@soe.ucsc.edu), Dejan Milutinovic (dmilutin@ucsc.edu) and Abhishek Halder (ahalder@ucsc.edu) from cyan140@ucsc.edu IP: 156.68.49.7
VIEWED	**05 / 23 / 2023** 18:22:26 UTC	Viewed by Qi Gong (qigong@soe.ucsc.edu) IP: 4.53.201.89
VIEWED	**05 / 23 / 2023** 18:27:36 UTC	Viewed by Abhishek Halder (ahalder@ucsc.edu) IP: 69.181.126.157
SIGNED	**05 / 23 / 2023** 18:29:20 UTC	Signed by Abhishek Halder (ahalder@ucsc.edu) IP: 69.181.126.157
SIGNED	**05 / 23 / 2023** 19:34:04 UTC	Signed by Qi Gong (qigong@soe.ucsc.edu) IP: 4.53.201.89
INCOMPLETE	**05 / 23 / 2023** 19:34:04 UTC	This document has not been fully executed by all signers.

DocuSign

Certificate Of Completion

Envelope Id: 0B162CDEF9EC481C810FE59C87132B2F
Status: Completed
Subject: Please sign Charlie Yan MS Thesis
Source Envelope:
Document Pages: 71
Signatures: 1
Certificate Pages: 1
Initials: 0
AutoNav: Enabled
EnvelopeId Stamping: Enabled
Time Zone: (UTC-08:00) Pacific Time (US & Canada)

Envelope Originator:
Charlie Yan
1156 High Street
Santa Cruz, CA 95064
cyan140@ucsc.edu
IP Address: 156.68.49.7

Record Tracking

Status: Original
5/23/2023 5:27:45 PM
Holder: Charlie Yan
cyan140@ucsc.edu
Location: DocuSign

Signer Events	Signature	Timestamp
Dejan Milutinovic dmilutin@ucsc.edu University of California, Santa Cruz Security Level: Email, Account Authentication (Optional)	DocuSigned by:  5F739D6B0881448... Signature Adoption: Uploaded Signature Image Using IP Address: 142.254.101.252	Sent: 5/23/2023 5:29:45 PM Viewed: 5/24/2023 12:29:09 AM Signed: 5/24/2023 12:33:46 AM

Electronic Record and Signature Disclosure:
Not Offered via DocuSign

In Person Signer Events	Signature	Timestamp

Editor Delivery Events	Status	Timestamp

Agent Delivery Events	Status	Timestamp

Intermediary Delivery Events	Status	Timestamp

Certified Delivery Events	Status	Timestamp

Carbon Copy Events	Status	Timestamp

Witness Events	Signature	Timestamp

Notary Events	Signature	Timestamp

Envelope Summary Events	Status	Timestamps
Envelope Sent	Hashed/Encrypted	5/23/2023 5:29:45 PM
Certified Delivered	Security Checked	5/24/2023 12:29:09 AM
Signing Complete	Security Checked	5/24/2023 12:33:46 AM
Completed	Security Checked	5/24/2023 12:33:46 AM

Payment Events	Status	Timestamps