IMPERIAL

Imperial College London
Faculty of Natural Sciences
Department of Mathematics

Control of Agent-Based Models across Scales

Sara Bicego

Supervisor: Dr. Dante Kalise
Co-supervisor: Dr. Nikolas Kantas

Examiners: Dr. Panos Parpas
Professor Lorenzo Pareschi

Thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy and the Diploma of the Imperial College of Science, Technology and Medicine

Academic Year 2024-2025

Declaration of Originality

I, Sara Bicego, declare that the work contained in this thesis is my own, or else is appropriately referenced. Any material for which I am not responsible is explicitly indicated by the appropriate use of references and citations.

Copyright

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC). Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose. When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes. Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Please seek permission from the copyright holder for uses of this work that are not included in the license mentioned above.

Abstract

This thesis presents a comprehensive investigation into modeling and controlling high dimensional agent-based models within a multi-scale framework. At the core of this research lies the fundamental challenge of overcoming computational bottlenecks in solving optimal control problems with many-agent dynamics. Solutions to this class of optimization problems are doubly cursed by the dimensionality of the system, first in the number of interacting agents, and second in the dimensionality of each agent's underlying dynamics. This dual nature suggests a broad spectrum of techniques to enhance the efficiency and scalability of optimal control computations across various modeling resolutions. At the microscopic scale, we address this double trouble at the agent-to-agent interaction level through data-driven approximation techniques. We combine neural network approximation models with control theory and existing solvers for synthetic data generation, resulting in efficient and effective control laws. When the computational challenge is mostly linked to a large number of agents, we switch to a macroscopic viewpoint, formulating an optimal control problem for the distribution of agents evolving in the underlying physical space. The resulting mean field dynamics offer an ideal framework to analyze emergent behaviors of many-agent systems and addressing PDE-constrained optimization problems to stabilize the system around desired steady states. When both the number of agents and the dimensionality of their underlying dynamics make traditional control synthesis impractical, we introduce a mesoscopic kinetic approximation framework that models agent densities as plasma-like matter. The mean-field behavior is approximated by averaging over controlled subsystems of interacting agent pairs in a Monte Carlo fashion. The accuracy of this approximation relies on high-frequency sampling of controlled binary interactions, for which we leverage neural network speed-ups similar to those used at the microscopic scale.

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor, Dante Kalise, for his unwavering support, guidance, and encouragement throughout my PhD studies. His insight and advice were instrumental not only in shaping this work but also in my personal and academic development as a researcher.

I am also grateful to Giacomo Albi, Greg Pavliotis, the LC2 research group, and all my collaborators, whose contributions enriched this thesis and informed several of the key ideas explored within. Their expertise and perspective helped broaden the scope of this research and created a stimulating and supportive environment in which to grow.

My sincere thanks go to my examiners, Panos Parpas and Lorenzo Pareschi, for their time, thoughtful feedback, and valuable suggestions during the viva and revision process.

I would also like to thank the following incredible people — all main characters in my PhD journey. To Ollie, for teaching me how to not take myself too seriously and for being my anchor. To my friends, near and far, for providing both distraction and encouragement during the most demanding phases of these past few years. A special mention goes to the office crew - particularly Karina, Thivan, Henry, Connor and Jamie, for your camaraderie, humour and friendship.

To my family, who believed in me when I could not — this thesis is dedicated to you. To my parents, Cristina and Paolo, for your steadfast support and for making each visit feel like a special occasion. To my brother, Davide, for being my best friend. And to my grandparents, Meri and Graziano — I finally finished my homework!

To my family

Contents

1 Introduction **9**

1.1 Constructing Agent-Based Models 10
1.2 Controlling Agent-Based Systems 13
1.3 Contributions and Publications 17
1.4 Outline of the Thesis 19

2 Background **21**

2.1 Optimal Control 21
2.1.1 Dynamic Programming and HJB equation 23
2.1.2 Pontryagin's Maximum Principle 26
2.1.3 State Dependent Riccati Equation 33
2.1.4 PMP vs. SDRE update for model predictive control, a comparison . 37
2.2 Data-driven control strategies 39
2.2.1 Feedforward Neural Networks 40
2.2.2 Recurrent Neural Networks 42
2.2.3 Training 43
2.2.4 Example: Gradient-augmented supervised learning of SDRE feedback laws 44
2.3 Deflation method for polynomial systems and PDEs 49
2.3.1 Computing the derivative of the deflated operator 50
2.3.2 Numerical considerations and limitations of deflation methods . . . 52
2.3.3 Example: Time optimal feedback control of nilpotent linear systems using neural networks 53
2.4 Concluding remarks 67

3 Microscopic Scale
Learning gradient-augmented feedback control synthesis **69**

3.1 Consensus Control of Cucker Smale dynamics 71
3.2 Pontryagin Optimality System 73
3.3 Semi-linear formulation 74

3.4 Gradient-augmented regression and data generation . . . 75
3.5 Numerical Test . . . 76
3.6 Concluding remarks . . . 77

4 Macroscopic Scale

Computation and Control of Unstable Steady States for Mean Field Multiagent Systems **79**

4.1 Introduction . . . 80
4.2 The McKean-Vlasov PDE and Characterization of Steady States . . . 84
4.3 Computing multiple steady states in the McKean-Vlasov PDE . . . 87
4.3.1 Spectral Galerkin method for the Stationary Fokker–Planck equation 88
4.3.2 Deflation algorithm for steady state identification . . . 90
4.3.3 The Haken-Kelso-Bunz Model . . . 91
4.3.4 Phase transitions for the $O(2)$ model with magnetic field . . . 96
4.4 Optimal feedback stabilization of unstable steady states . . . 98
4.4.1 First-order necessary optimality conditions . . . 99
4.4.2 Nonlinear Model Predictive Control . . . 103
4.5 Numerical Tests . . . 104
4.5.1 Hegselmann-Krause opinion formation model . . . 105
4.5.2 Two-dimensional Von Mises interaction potential . . . 107
4.6 Concluding remarks . . . 109

5 Mesoscopic Scale

Control of kinetic collective dynamics by deep neural feedback laws **111**

5.1 A Boltzmann formulation for the control of mean field dynamics . . . 117
5.1.1 Binary controlled dynamics . . . 118
5.1.2 Consistency with the mean field formulation . . . 120
5.1.3 Full state controlled binary dynamics . . . 122
5.2 Asymptotic Monte Carlo methods for constrained mean field dynamics . . . 123
5.3 Infinite horizon optimal control of binary dynamics . . . 126
5.3.1 Discrete-time Linear Quadratic Regulator . . . 128
5.3.2 Discrete-time Non-Linear Quadratic Regulator . . . 130
5.3.3 Discrete-time SDRE approach . . . 131
5.4 Synthetic data generation for Neural Network feedback laws . . . 132
5.5 Numerical Tests . . . 134
5.5.1 Test 0: Sznajd model . . . 135
5.5.2 Test 1: Cucker-Smale model . . . 138
5.5.3 Test 2: quasi-Morse potential . . . 142
5.6 Concluding remarks . . . 146

6 Conclusions **150**
6.1 Future directions . . . 151
6.1.1 Robust kinetic control for parametric uncertainty . . . 152
6.1.2 Towards a quantitative understanding of collective animal behaviour under climatic events . . . 155

Appendix A: *Supplementary material to Macroscopic scale chapter* **162**
A.1 Fréchet differentiability of the cost functional $\mathcal{J}$ . . . 162
A.2 Derivation of the continuous optimality system . . . 165
A.3 Consistency of the discretized optimality condition . . . 166

Appendix B: *Supplementary material to Mesoscopic scale chapter* **168**
B.4 Proof of Theorem 5.1.2 . . . 168
B.5 Asymptotic constrained symmetric Nanbu algorithm . . . 171

Acronyms

ABM Agent-based model

OCP Optimal control problem

MF OCP Mean field optimal control problem

MFG Mean field game

HJB Hamilton-Jacobi-Bellman (equation)

HJI Hamilton-Jacobi-Isaacs (equation)

LQR Linear Quadratic Regulator

NLQR Non-linear Quadratic Regulator

ARE Algebraic Riccati equation

SDRE State Depended Riccati equation

DARE Discrete time Algebraic Riccati equation

DSDRE Discrete time State Depended Riccati equation

NN Neural Network

MSE Mean Squared Error

PRMSE Percent Root MEan Squared Error

HKB Haken-Kelso-Bunz

LLS Linear Least Squares

MC Monte Carlo

RL Reinforcement Learning

Symbols

N Number of agents in the system

d Dimension of physical space where the agents dynamics take place

$n = dN$ Full dimensionality of the system

$\mathbf{s}(t) := (s_1, s_2, \dots, s_N)^\top$ Ensemble state at time t, $s_i(t) \in \mathbb{R}^{2d}$ state of i-th agent

$\mathbf{x}(t) := (x_1, x_2, \dots, x_N)^\top$ Ensemble position at time t, $x_i(t) \in \mathbb{R}^d$ position of i-th agent

$\mathbf{v}(t) := (v_1, v_2, \dots, v_N)^\top$ Ensemble velocity at time t, $v_i(t) \in \mathbb{R}^d$ velocity of i-th agent

$\mathbf{u}(t) := (u_1, u_2, \dots, u_N)^\top$ Ensemble control at time t, $u_i(t) \in \mathbb{R}^m$ control of i-th agent

$\mathbf{y}(t) := (y_1, y_2, \dots, y_N)^\top$ Ensemble initial condition, with $y_i \in \mathbb{R}^n$ initial state of i-th agent

$\boldsymbol{\lambda}(t) := (\lambda_1, \lambda_2, \dots, \lambda_N)^\top$ Ensemble adjoint variable or co-state, $\lambda_i(t)$ adjoint to $s_i(t)$

$P(x_i, x_j)$ Interaction kernel between i-th and $j-$th agent

$\mathcal{J}(\mathbf{s}, \mathbf{u})$ Cost functional

$\mathcal{L}(\mathbf{s}, \mathbf{u}; t)$ Running cost

$l(\mathbf{s})$ Tracking error

$\mathcal{G}(u)$ Control energy penalization

γ Control energy penalization coefficient

U_{adm} Set of admissible controls

$\mathcal{H}(\mathbf{s}, \mathbf{u}, \boldsymbol{\lambda}; t)$ Hamiltonian

$f(\mathbf{s})$ Free dynamics, with semilinearization $f(\mathbf{s}) = \mathcal{A}(\mathbf{s})\mathbf{s}$

$g(\mathbf{s})$ Control-to-state operator, with semilinearization $g(\mathbf{s}) = \mathcal{B}(\mathbf{s})$

Ω State space

$\mathbb{O}$ Zero-valued matrix

$\mathbb{I}$ Identity operator

$\mathcal{Q}(\rho, \rho)$ Boltzmann-type collision operator

(s, s_*) Pair of interacting agents

$('v,' v_*)$ Pre-interation velocities generating (v, v_*) after binary interaction

(v', v'_*) Post-interation velocities for a pair of interacting agents with velocities (v, v_*)

$V(\mathbf{y})$ Value function, denoting the optimal cost-to-go from the initial condition $\mathbf{y}$

$\Psi(\mathbf{s}; T)$ Terminal cost associated with the state $\mathbf{s}$ at final time T

$\mathbf{u}^*(\mathbf{s})$ Optimal feedback control

$(\mathbf{s}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t))$ Extremal trajectories solving PMP optimality system

Π ARE solution operator, introduced for the quadratic ansatz $V(\mathbf{s}) = \mathbf{s}^\top \Pi \mathbf{s}$

$\Pi(\mathbf{s})$ SDRE solution operator, introduced for the quadratic ansatz $V(\mathbf{s}) = \mathbf{s}^\top \Pi(\mathbf{s})\mathbf{s}$

$\mathcal{K}$ Feedback operator, computed by solving the ARE

$\mathcal{K}(\mathbf{s})$ State-dependent feedback operator, computed by solving the SDRE

t_h Discrete time in the uniformly spaced discretization of $[0, T]$

dt Step-size in the discretization of the time interval $[0, T]$

ψ_θ Neural network approximation of the function ψ with trainable parameters θ

l_k k-th layer of the neural network

σ_k Activation function of the k-th layer

$\mathcal{T}$ Training set, with N_s samples

$\mathcal{T}_v$ Validation set, with N_v samples

$\mathcal{T}_t$ Test set, with N_t samples

$\mathcal{L}^{NN}$ Loss function for neural network training

r^2 Coefficiant of determination

$\mathcal{F}$ Residual of a root-finding problem

$\mathcal{D}$ Deflated residual of the root-finding problem $\mathcal{F}$

$\mathcal{M}$ Deflation operator, with deflation power p and shift parameter ξ

μ Weight parameter in gradient-augmented regression loss

$W(x, y)$ Interaction potential between agents having states x and y

$U(x)$ Confining potential at state x

$\mathcal{F}[\rho]$ Free energy functional

L Number of Fourier modes for truncated Fourier series approximation

β Inverse temperature of particle bath (measure of noise amplitude)

κ Interaction strenght between particles

$\mathcal{J}_N(\mathbf{s}, \mathbf{u})$ Cost functional for the $N-$agents system

λ_ε Interaction frequency

Chapter 1

Introduction

In nature, complex group behaviors often emerge from the interactions of individuals following simple rules. Honeybees use basic communication and task allocation strategies, yet their collective behavior enables the hive to function with remarkable sophistication, appearing almost intelligent at the group level. Similarly, bird flocks are able to change direction in perfect unison without a central coordinator, and ant colonies can efficiently optimize food paths through simple local interactions. This approach reflects a core paradigm of mathematical research: breaking down formidable challenges into manageable, solvable components.

Inspired by such natural phenomena, Agent-Based Models (ABMs) were designed to model complex collective behaviors across diverse domains. ABMs consist of networks of intelligent agents, each with independent decision-making capabilities and specific goals, while also engaging in dynamic interactions that may not always be rational. Applications of ABMs are both diverse and impactful, enabling extensive research into emergent behaviors driven by decentralized, collaborative, or competitive interactions. Beyond ecological simulations, ABMs find applications in robotic coordination, urban planning, traffic management, and distributed artificial intelligence, among others. In robotics, interactions between drones are used to coordinate swarm dynamics towards military missions, disaster management, rescue operations, infrastructure inspection, and surveillance [230, 155].

In urban settings, analyzing pedestrian behavior in crowded environments plays a critical role in designing public venues with effective evacuation plans and congestion mitigation strategies [128, 11]. Similarly, modeling vehicles as agents that share information like speed, direction, and location is a cornerstone of automated driving [30, 42]. ABMs have also proven instrumental in ecological preservation, where they are used to model, monitor, and manage complex ecosystems, enabling researchers to simulate species interactions, predict environmental changes, and design strategies for biodiversity conservation.

1.1 Constructing Agent-Based Models

Drawing inspiration from physics and classical mechanics, we encode fundamental interaction forces through first- or second-order dynamical systems. Using principles from Newtonian physics, these models effectively capture both external influences and the microscopic interactions between agents, including attraction, repulsion, self-propulsion, and alignment forces. These equations govern the evolution of the state variables of each agent in the ensemble, often describing spatial movement. As a clarifying example, we consider a Cucker-Smale system [91] of N agents within the physical space $\mathbb{R}^d$, where the $i-$th agent is described by the state $s_i := (x_i, v_i) \in \mathbb{R}^{2d}$ collecting position and velocity information for $i = 1, \dots, N$. The evolution of the system can be described as

$$\begin{cases} \dot{x}_i(t) = v_i(t) \\ \dot{v}_i(t) = \sum_{j=1}^{N} P(x_i(t), x_j(t))(v_j(t) - v_i(t)) + u_i(t) \end{cases}, \tag{1.1}$$

for $t \geq 0$, where $P(\cdot, \cdot)$ denotes the communication function between two interacting agents, for example the Cucker-Smale interaction kernel

$$P(x_i, x_j) = \frac{\alpha_1}{(1 + |x_i - x_j|^2)^{\alpha_2}}. \tag{1.2}$$

In (1.1), u_i encodes all the external forces acting on the $i-$th agent. The interplay between internal interactions and external control forces is a fundamental feature of all the models

discussed in this thesis.

A fascinating property of these agent-based dynamical systems is their capacity for **self-organization**. Based on the nature of the interaction rules described in $P(\cdot,\cdot)$ coherent global patterns emerge spontaneously from purely local interactions. Without centralized control or external coordination, the ensemble dynamics can evolve toward ordered collective behaviors through the complex interplay of individual agent rules. Emergent patterns such as synchronization, flocking, or lane formation, represent macroscopic structures that cannot be directly inferred from the properties of individual agents alone [205]. This self-organizing principle provides a powerful framework for understanding how complex, coordinated behaviors can arise across diverse systems, from bird murmurations and fish schools to pedestrian flows and cellular aggregations. Remarkably, similar pattern formations can emerge from systems with vastly different underlying mechanisms, suggesting fundamental organizational principles that transcend specific domains. For instance, flocking dynamics in bird flocks and robotic swarms can be captured by Cucker-Smale-type interaction kernels like (1.2) [91, 182]. Even in abstract spaces, social influence models exhibit consensus formation akin to physical clustering, where opinion dynamics evolve based on similarity-weighted interactions [127, 237], much like how oscillators synchronize in the Kuramoto model through phase coupling [157]

$$P(x_i, x_j) = \kappa \sin(x_i - x_j). \tag{1.3}$$

Similarly, lane formation in pedestrian flows arises from short-range repulsion and long-range attraction forces between individuals [215, 184, 53], a mechanism that also governs swarming behaviors in fish schools and drone formations [176, 177]. These interactions are often modeled using Morse-type potentials, which balance attraction and repulsion,

$$P(x_i, x_j) = C_1 e^{-|x_i - x_j|/\sigma_1} - C_2 e^{-|x_i - x_j|/\sigma_2}. \tag{1.4}$$

As the number of agents grows large ($N \to \infty$), interactions at the agent level can be

approximated by a macroscopic description in terms of the distribution $\rho(t, x, v)$ which characterizes the probability of finding an agent with state $s = (x, v)$ at time t [243]. Under this statistical interpretation, the dynamics for the density ρ follow a mean-field model over the physical space $\mathbb{R}^{2d}$:

$$\partial_t \rho + v \cdot \nabla_x \rho = \nabla_v \cdot \Big((\mathcal{P}[\rho] + u)\rho \Big), \tag{1.5}$$

where the mean-field interaction operator is given by

$$\mathcal{P}[\rho](x, v; t) = \int_{\mathbb{R}^{2d}} P(x, y)(w - v)\rho(y, w; t)\, dy\, dw. \tag{1.6}$$

From a dynamical perspective, this formulation provides a powerful framework for studying emergent behavior. The system typically evolves toward stable configurations that correspond to steady states of the Vlasov-type dynamics, often minimizing an underlying energy functional or satisfying specific equilibrium conditions [90, 115]. Notably, small variations in the interaction kernel $P(\cdot, \cdot)$ can induce bifurcations, leading to qualitatively different collective behaviors [52, 246]. By analyzing the stability of these equilibria, we can study the conditions under which specific patterns emerge and persist.

An alternative description of the macroscopic dynamics can be derived from kinetic theory by modeling the evolution of the probability distribution of agents while incorporating explicit microscopic interaction effects through a Boltzmann-type collision operator [69, 174, 16]. In this mesoscopic model, the evolutionary equation takes the form

$$\partial_t \rho + v \cdot \nabla_x \rho = \mathcal{Q}(\rho, \rho), \tag{1.7}$$

where the operator $\mathcal{Q}(\rho, \rho)$ encodes binary collision dynamics between agents, accounting for both the gain and loss of particles in a given state at time t. A standard form of the Boltzmann collision operator can be expressed based on the interaction dynamics in (1.1),

simplified to the case $N = 2$ particles with velocities (v, v_*).

$$\mathcal{Q}_\eta(\rho, \rho)(t, x, v) = \int_{\mathbb{R}^{2d}} \frac{1}{\mathcal{J}_\eta} \rho(t, x, {}' v)\rho(t, x_*, {}' v_*) - \rho(t, x, v)\rho(t, x_*, v_*)\, dx_*\, dv_*\,, \tag{1.8}$$

where (v', v'_*) represents the velocities update after a forward Euler step of length η, and $\mathcal{J}_\eta$ is the Jacobian of the binary interactions map $(v, v_*) \longmapsto (v', v'_*)$, while $({}'v, {}' v_*) \longmapsto (v, v_*)$ are the pre-interaction velocities that generate the couple (v, v_*).

As discussed in [5], the choice of resolution scale in modeling ABMs remains a subject of debate, as each level of description offers distinct advantages and limitations. Microscopic models, such as agent-based dynamics, provide the highest fidelity by explicitly tracking individual agents and their interactions. However, their computational complexity grows rapidly with system size, making them impractical for large-scale analysis. At the macroscopic level, hydrodynamic-type models emerge from asymptotic approximations of these agent-to-agent interactions, describing the system through continuum fields that capture global patterns while sacrificing individual variability. Although these models enable efficient large-scale predictions, they inherently smooth out local heterogeneities that may be crucial in certain contexts. Kinetic models bridge the gap between microscopic and macroscopic descriptions by approximating the system's state through a probability distribution over individual agent states. While this approach allows for a more tractable representation of collective dynamics, it relies on the assumption that the probability density is well-defined and sufficiently smooth. Rather than endorsing a single *correct* resolution, in this thesis we adopt a **multi-scale perspective** that leverages the strengths of each modeling approach. By selecting the most suitable resolution for each case study, we ensure both analytical tractability and practical applicability while maintaining consistency across different scales.

1.2 Controlling Agent-Based Systems

Moving past the mere modeling purposes, a fascinating question is how to influence systems and induce a specific collective behaviour. This problem has a variety of applications,

e.g. designing collision-avoidance protocols for swarm robotics [79, 194, 208, 119], formulating pedestrian evacuation strategies for crowd dynamics [11, 89, 107, 62], assessing interventions for traffic flow management [239, 136, 226], and learning mechanisms for opinion dynamics [12, 18, 123].

ABMs can be influeced by guiding the evolution of agents using an external force, represented in our model by $\mathbf{u} := (u_1, \dots, u_N)^\top$. Within the **optimal control** framework, this external force becomes a variable that can be adjusted and tuned to reach the desired outcome. Referred to as the control action, $\mathbf{u}$ is determined by solving an optimization problem. Specifically, the control is computed as the minimizer of a cost functional, subject to the constraints imposed by the controlled dynamics. The cost encapsulates the trade-off between competing goals, such as minimizing effort while maximizing performance. Typically, we consider optimal control problems (OCPs) of the form:

$$\mathbf{u}^* = \underset{\mathbf{u}\in U_{adm}}{\operatorname{argmin}} \left\{ \mathcal{J}(\mathbf{u}, \mathbf{s}) := \frac{1}{N} \int_0^{+\infty} \sum_{i=1}^{N} \mathcal{L}(s_i,\, u_i;\, t)\, dt \right\} \text{ s.t. } (1.1)\,, \tag{1.9}$$

where the running cost function

$$\mathcal{L}(s,\, u;\, t) := l(s(t)) + \gamma\, \mathcal{G}(u(t)) \tag{1.10}$$

penalizes both the state tracking error $l(s)$ and the control effort $\mathcal{G}(u)$, weighted by a coefficient $\gamma > 0$.

The choice of cost functional depends on the problem setting and desired control objectives. A common choice for state penalization is a quadratic function, $l(s) = \|s - s_{\text{ref}}\|_{l_2}^2$, which encourages the system to converge toward a desired reference state s_{ref} w.r.t. the Euclidean norm. Similarly, the control cost is often taken as a quadratic term, $\mathcal{G}(u) = \|u\|_{l_2}^2$, promoting energy-efficient control strategies by discouraging excessive control effort. The coefficient γ plays a crucial role in balancing the trade-off between minimizing state deviations and reducing control expenditure, influencing the overall control aggressiveness.

The motivation for quadratic penalties is rooted in their convexity, which ensures that the

optimization problem remains well-posed and computationally tractable [1]. Convex cost functions guarantee the existence of a unique global minimizer, and enable efficient numerical solutions using gradient-based optimization techniques and dynamic programming principles. These advantages make quadratic penalties particularly appealing for a wide range of applications. More sophisticated cost functionals can be used to encode additional problem-specific constraints. For example, ℓ_1-type penalties on the control, such as $\mathcal{G}(u) = \|u\|_{\ell_1}$, promote sparse control strategies. This is particularly useful in networked or distributed systems where actuation is costly or must be applied selectively [225, 61]. Barrier functions can enforce safety constraints, ensuring that agents avoid restricted areas, which is crucial in applications like collision avoidance for multi-robot coordination [166, 255]. In time-sensitive scenarios, such as emergency response or automated driving, exponentially weighted costs can prioritize short-term performance, ensuring rapid reaction to immediate changes while balancing long-term objectives [193]. In this thesis, we focus on convex cost functionals to preserve the well-posedness and tractability of the optimization problem.

Centralized vs. decentralized control strategies

When designing control strategies for ABMs, a fundamental distinction arises between centralized and decentralized approaches. In **centralized control**, a single decision-maker has global access to all agents' states and determines the control actions for the entire system, as seen in (1.9). While this approach can achieve globally optimal solutions, it suffers from scalability challenges. As the number of agents grows, the computational complexity and communication overhead increase exponentially, making real-time implementation impractical.

In contrast, **decentralized control** distributes the decision-making process among individual agents, each relying on local information and limited communication with neighbors to determine its actions. The optimization problem for each agent is posed as:

$$u_i^* = \underset{u_i \in U_{adm}}{\operatorname{argmin}} \left\{ \mathcal{J}_i(s_i, u_i) := \int_0^{+\infty} \mathcal{L}(s_i, u_i; t)\, dt \right\} \quad \text{s.t.} \quad (1.1) \tag{1.11}$$

The running cost $\mathcal{L}$ might encode limited communication between agents, but each agent optimizes without full access to the global state $\mathbf{s}$. This approach significantly enhances scalability, but the reliance on partial information typically leads to suboptimal performance, as individual agents lack a global perspective on the system's evolution.

This distinction between centralized and decentralized control has a parallel at the macroscopic level. In **Mean Field Optimal Control Problems** (MF OCPs), the global decision-maker optimizes the control actions for a large population of agents using information about the overall distribution ρ of agent states [45, 116]. This is formulated as the following PDE-constrained optimization problem:

$$\mathbf{u}^* = \underset{\mathbf{u}\in U_{adm}}{\operatorname{argmin}} \left\{ \mathcal{J}(\rho, \mathbf{u}) := \int_0^{+\infty} \int_{\mathbb{R}^{2d}} \mathcal{L}(s, u; t)\, d\rho(s)\, dt \right\} \quad \text{s.t.} \quad (1.5)\,. \tag{1.12}$$

In **Mean Field Games** (MFGs), each agent makes its decision based on its own state s_i and the distribution of states across all agents, essentially playing a game against the rest of the population [162]. The population's collective behavior is captured through the mean field, which reflects the aggregate state of all agents and influences each agent's optimal strategy. In this framework, the optimal solution is described by a *Nash equilibrium*, where no agent can improve its outcome in the optimization horizon $(0, T)$ by unilaterally changing its strategy, given the strategies of all other agents. This is typically formulated as a forward-backward system of equations:

$$\begin{cases} \partial_t \rho + v \cdot \nabla_x \rho = \nabla_v \cdot ((\mathcal{P}[\rho(t)] + u)\, \rho) & \text{in } \mathbb{R}^{2d} \times (0, T), \\ -\partial_t V(s,t) = \min_u \{\mathcal{L}(s,u) + \nabla_s V(s,t) \cdot (\mathcal{P}[\rho(t)] + u)\} & \text{in } \mathbb{R}^{2d} \times (0, T), \\ \rho(0) = \rho_0, \quad V(s,T) = 0 & \text{in } \mathbb{R}^{2d}, \end{cases} \tag{1.13}$$

where the first equation describes the evolution of the state distribution of the agents, while the second equation represents the Hamilton-Jacobi-Bellman (HJB) equation that characterizes the optimal control strategy given the time-dependent mean field $\rho(t)$.

Challenges of centralized control for high-dimensional agent-based systems

In this thesis, we focus on centralized control frameworks across different scales and work to mitigate the challenges associated with this class of dynamic optimization problems. Solving such OCPs generally requires advanced numerical techniques, as analytical solutions are rarely available except in special cases, such as linear-quadratic regulators [158]. Common approaches include indirect methods based on Pontryagin's maximum principle [212], which transform the problem into a two-point boundary value problem, and direct methods that discretize the control and state trajectories, converting the OCP into a finite-dimensional nonlinear programming problem. For large-scale systems with many agents, these approaches become computationally prohibitive, necessitating approximation strategies such as mean-field control or macroscopic formulations that capture the system's emergent behavior while reducing computational complexity.

In the ABMs context, the numerical solution of these problems is shadowed by the challenges deriving from the dimensionality of the system being controlled. The controlled dynamics describe the evolution of the system ensemble state, whose dimension scales according to both the number N of agents being considered, and the dimensionality d of the space where the agent-to-agent dynamics take place. Addressing this dual challenge is the central focus of this dissertation, with the aim of developing methods to effectively mitigate the computational complexities associated with solving OCPs across scales for high-dimensional, many-agent systems.

1.3 Contributions and Publications

The dual nature of the curse of dimensionality we aim at alleviating suggests a broad spectrum of potential techniques. This has motivated a variety of projects, all unified by the overarching goal of addressing the computational challenges associated with the (approximate) solution of OCPs for high dimensional many-agent systems. Below, we provide a taxonomy of the techniques and contributed papers presented in this thesis, organized according to their modeling resolution:

- **Microscopic scale** At the agent-to-agent iteraction level, we rely on Neural Network approximation techniques to mimick and accelerate existing numerical OCP solvers. Performance is further enhanced by incorporating information about the gradient of the target variable, which comes as a by-product the optimization routines under consideration.

  [8] G. Albi, S. Bicego and D. Kalise, *Gradient-augmented Supervised Learning of Optimal Feedback Laws Using State-dependent Riccati Equations*. IEEE Control Systems Letters 6(2022): 836 -841, doi: 10.1109/LCSYS.2021.3086697

  [6] G. Albi, S. Bicego, M. Herty, Y. Huang, D. Kalise, and C. Segala, *Data/moment-driven approaches for fast predictive control of collective dynamics*. ArXiv preprint, doi: arXiv.2402.15611 to appear as a book chapter on Model Predictive Control, Series on Dynamic Modeling and Econometrics in Economics and Finance, Springer.

  An alternative research direction explores the use of deflated Newton's method to solve time-optimal control problems governed by $n-$th integrator dynamics. The algorithm solves a polynomial system to determine the switching times of the optimal bang-bang control and is used to generate a synthetic dataset for training a Neural Network classifier that approximates the state-to-control map.

  [49] S. Bicego, S. Gue, D. Kalise, and N. Villamizar, *Time Optimal Feedback Control of Nilpotent Linear Systems using Neural Networks*. Arxiv preprint, doi: arXiv.2503.17581, submitted to Communications on Applied Mathematics and Computation

- **Macroscopic scale** Since the number of agents contributes to the dimensionality of the system's ensemble state dimension, a viable approach for many-agent OCPs is to adopt a mean-field perspective. This shifts the focus from an agent-based description to modeling the density of agents as a whole. Within this framework, we consider a stabilization mean-field OCP governed by McKean–Vlasov PDEs.

[50] S. Bicego, D. Kalise and G.A. Pavliotis, *Computation and control of unstable steady states for mean field multiagent systems.* Arxiv preprint, to appear in Proceedings of the Royal Society A, doi: arXiv.2406.11725.

We highlight that the goal of the project is twofold: first, stable and unstable stationary configurations of the governing PDE are identified using a deflated root-finding approach in Banach spaces. Second, a control problem is formulated and solved to steer the system towards a desired (unstable) steady state.

- **Mesoscopic scale** In many-agent systems with high-dimensional underlying dynamics, a mean-field approximation may fall short in achieving sufficient computational complexity reduction. One way to address this limitation is to treat the agents' density as plasma matter, approaching the mean-field solution from suboptimality via averaging over controlled subsystems of interacting pairs of agents. This routine still requires efficient solution of many 2-agents OCPs, for which we rely on Neural Network approximation.

  [9] G. Albi, S. Bicego and D. Kalise, *Supervised learning for kinetic consensus control.* Proceedings of the 25th International Symposium on Mathematical Theory of Networks and Systems (I), pp. 308-313, doi: 10.1016/j.ifacol.2022.11.036

  [10] G. Albi, S. Bicego and D. Kalise, *Control of high-dimensional collective dynamics by deep neural feedback laws and kinetic modelling.* Arxiv preprint, doi: arXiv.2404.02825, submitted to Journal of Computational Physics.

The specific contributions of these publications are explicitly stated at the beginning of the respective chapters.

1.4 Outline of the Thesis

The rest of this thesis is structured as follows. Chapter 2 establishes the theoretical foundations of this research. It introduces optimal control theory, explores neural network approximation techniques for accelerating numerical solvers for optimal control problems,

and details the deflation algorithm from [50]. This chapter also presents two motivating numerical examples, adapted from [8] and [49], which investigate the use of neural networks for accelerating feedback control computations and the application of deflated Newton's method to time-optimal OCPs, respectively. Chapter 3 examines the microscopic resolution level, focusing on gradient-augmented neural network approximations of controllers for large-scale agent-based models, as developed in [8, 6]. Chapter 4 transitions from an agent-based descriptions to the mean-field frameworks, exploring the use of deflation techniques for identifying and stabilizing stationary states in the McKean–Vlasov equation, following [50]. Chapter 5 presents the kinetic approximation approach for controlling high-dimensional many-agent systems, synthesizing results from [9, 10]. Finally, Chapter 6 provides concluding remarks and perspectives for future research.

Chapter 2

Background

In this chapter, we present the theoretical and computational foundations for the multi-scale control strategies developed in the rest of the thesis. We begin with a concise overview of optimal control theory, introducing its core concepts and emphasizing the computational challenges posed by high-dimensional problems. We then discuss neural network approximation techniques as data-driven tools to accelerate control synthesis, with a focus on learning efficient state-to-control mappings. As a concrete example, we illustrate how neural networks can be used to speed up feedback computation, following the approach developed by the author in [8]. The chapter concludes with an introduction to the deflation technique, a numerical strategy for systematically identifying multiple solutions to nonlinear systems. Its relevance is motivated through the analysis of polynomial root-finding problems arising in time-optimal control, as explored in [49].

2.1 Optimal Control

Optimal control is a mathematical framework used for determining the best possible strategy for influencing dynamical systems towards a desired objective. The control variable can be interpreted as an external force that acts on the system and that can be optimized while balancing performance and efficiency, as seen in (1.9).

Optimal control emerged as a distinct mathematical field in the aftermath of World War

II, driven largely by applications in military defense. Although the problem's roots can be traced to early developments in the Calculus of Variations, the nonlinear nature of the dynamical systems under consideration often resulted in differential equations that were unsolvable at the time. In a technological race characteristic of the Cold War, scientists from both the East and the West independently developed the two foundational methodologies in optimal control: Pontryagin's Maximum Principle and Dynamic Programming. We refer the reader to [207] for an historical overview of these contributions.

In a general framework, denoting as $\mathbf{s}(t)$ the state of the system at time t, and $\mathbf{u}(\cdot)$ as the control action, we consider quadratic cost functionals

$$\min_{\mathbf{u}(\cdot)\in U_{adm}} \mathcal{J}(\mathbf{y}, \mathbf{u}(\cdot)) := \int_0^{+\infty} \mathbf{s}^\top(t) Q \mathbf{s}(t) + \mathbf{u}^\top(t) R \mathbf{u}(t) dt, \tag{2.1}$$

subject to nonlinear, control-affine constraint

$$\dot{\mathbf{s}}(t) = f(\mathbf{s}(t)) + g(\mathbf{s}(t))\mathbf{u}(t)\,, \qquad \mathbf{s}(0) = \mathbf{y}\,, \tag{2.2}$$

where we denote the state of the system as $\mathbf{s} \in \mathbb{R}^n$, and the control variable $u(\cdot) \in U_{adm} := \{u(t) : \mathbb{R}^+ \to \mathbb{R}^m \text{ is measurable}\}$. We further assume $0 \prec Q \in \mathbb{R}^{n\times n}$ and $0 \preceq R \in \mathbb{R}^{n\times m}$ to be symmetric.

Without imposing a priori any regularity on the control $\mathbf{u}$, the measurability assumption ensures existence and uniqueness of the controlled trajectory (2.2) via Carathéodory theorem ([110] Chapter 8.1).

Theorem 2.1.1: Carathéodory

Assume that

(*i*) For every $\mathbf{u}(t)$, the function $\mathbf{s} \mapsto f(\mathbf{s}) + g(\mathbf{s})\mathbf{u}$ is continuous.

(*ii*) The function $f(\mathbf{s}(t)) + g(\mathbf{s}(t))\mathbf{u}(t)$ is measurable in $t \in \mathbb{R}_+$.

(*iii*) There exists a function $L(\mathbf{s}) \in L^1_{\text{loc}}(\mathbb{R}_+)$ s. t. $\|f(\mathbf{s}) + g(\mathbf{s})\mathbf{u}\| \leq L(\mathbf{s})$, $\forall \mathbf{u} \in U_{adm}$.

Then,

$$\mathbf{s}(t) = \mathbf{y} + \int_0^t f(\mathbf{s}(\tau)) + g(\mathbf{s}(\tau))\mathbf{u}(\tau)\, d\tau \tag{2.3}$$

is the unique solution to (2.2).

In what follows, we consider dynamics $f(\mathbf{s}) : \mathbb{R}^n \to \mathbb{R}^n$ and $g(\mathbf{s}) : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ in $\mathcal{C}^1(\mathbb{R}^n)$, satisfying $f(0) = 0$ and $g(0) = 0$. These assumptions ensure that the conditions of Theorem 2.1 are met.

Moreover, the assumption of a quadratic running cost as in (2.1), with symmetric positive semi-definite operators Q and R, ensures convexity of the optimization objective. This, in turn, implies the existence of a unique solution and facilitates efficient computation using well-established optimization algorithms.

The design of the cost functional (2.1) is key to ensuring the convergence of the controlled system to the desired behavior. Depending on the application, the optimization can be performed over a finite horizon T, where the control strategy is tailored to a specific time interval, or an infinite horizon $T \to +\infty$, which enforces asymptotic stability in the sense of Lyapunov [131]. In this thesis we will mostly be concerned with the latter framework.

2.1.1 Dynamic Programming and HJB equation

We start by addressing the continuous-time deterministic infinite horizon OCP in (2.1-2.2) by means of dynamic programming. The approach stands on the principle formulated by R. Bellman in 1950s, stating that the optimal policy for a sequential decision problem has the property that – in each intermediate time – the remaining decisions must form an

optimal policy with respect to the current system configuration.

Mathematically, this principle can be expressed in terms of the value function associated with the problem, defined as

$$V(\mathbf{y}) := \inf_{\mathbf{u}(\cdot)\in U_{adm}} \mathcal{J}(\mathbf{y}, \mathbf{u}(\cdot))\,. \tag{2.4}$$

The set of admissible controls U_{adm} is typically chosen based on the problem's requirements. A common choice is $L^2([0,+\infty);\mathbb{R}^m)$, which ensures finite control energy, or $L^\infty([0,+\infty);U)$, where U is a compact subset of $\mathbb{R}^m$, enforcing bounded control inputs. Alternatively, a more general choice is to define U_{adm} as the set of all measurable functions $u(\cdot) : [0,+\infty) \to \mathbb{R}^m$, allowing for unconstrained controls. The specific choice of U_{adm} depends on the problem setting and the desired regularity and constraints on the control function. We refer to [38] for details.

The value function encodes the cost associated with the optimally controlled dynamics starting from $\mathbf{y}$. Bellman's Dynamic Programming Principle the provides a first characterization of the value function.

Theorem 2.1.2: Dynamic Programming Principle

Let the system dynamics $f(\mathbf{s})$ and $g(\mathbf{s})$ in (2.2) satisfy the assumptions of Carathéodory's theorem 2.1. Additionally, assume that the cost functional is of the form (2.1), with $0 \prec Q \in \mathbb{R}^{n\times n}$ and $0 \preceq R \in \mathbb{R}^{n\times m}$ symmetric. Then, for all $\mathbf{y} \in \mathbb{R}^n$ and $t > 0$

$$V(\mathbf{y}) = \inf_{\mathbf{u}(\cdot)\in U_{adm}} \left\{ \int_0^t \mathbf{s}^\top(\tau) Q \mathbf{s}(\tau) + \mathbf{u}^\top(\tau) R \mathbf{u}(\tau) d\tau + V(\mathbf{s}(t)) \right\}. \tag{2.5}$$

The value function can also be characterized as the unique viscosity solution of a convex Hamilton-Jacobi-type partial differential equation, commonly known as the Hamilton-Jacobi-Bellman (HJB) equation. We now formally derive the HJB equation from Bellman's principle, while a rigorous justification follows from viscosity solution theory; see [38].

To do this, we consider equation (2.5) over a short interval of length Δt:

$$V(\mathbf{s}(t)) = \inf_{\mathbf{u}(\cdot)\in U_{adm}} \left\{ \int_{t}^{t+\Delta t} \mathbf{s}^\top(\tau) Q \mathbf{s}(\tau) + \mathbf{u}^\top(\tau) R \mathbf{u}(\tau) d\tau + V(\mathbf{s}(t+\Delta t)) \right\}. \tag{2.6}$$

We approximate the change in the value function over small time intervals Δt using a first-order Taylor expansion

$$V(\mathbf{s}(t+\Delta t)) = V(\mathbf{s}(t)) + \Delta t \left[\nabla V(\mathbf{s}(t))^\top \left(f(\mathbf{s}(t)) + g(\mathbf{s}(t))\mathbf{u}(t) \right) \right] + \mathcal{O}(\Delta t^2), \tag{2.7}$$

while the integral cost, for short intervals, can be approximated as

$$\int_{t}^{t+\Delta t} \mathbf{s}^\top(\tau) Q \mathbf{s}(\tau) + \mathbf{u}^\top(\tau) R \mathbf{u}(\tau) d\tau \approx \Delta t \left[\mathbf{s}^\top(t) Q \mathbf{s}(t) + \mathbf{u}^\top(t) R \mathbf{u}(t) \right]. \tag{2.8}$$

Substituting in equation (2.6), while letting $\Delta t \to 0$, we obtain the HJB equation

$$\inf_{\mathbf{u}(\cdot)\in U_{adm}} \left\{ \mathbf{s}^\top(t) Q \mathbf{s}(t) + \mathbf{u}^\top(t) R \mathbf{u}(t) + \nabla V(\mathbf{s}(t))^\top \left(f(\mathbf{s}(t)) + g(\mathbf{s}(t))\mathbf{u}(t) \right) \right\} = 0 \tag{2.9}$$

The optimal control $\mathbf{u}^*$ is found in the unconstrained case by imposing the first-order optimality condition, i.e. by requiring the gradient w.r.t. $\mathbf{u}$ of the objective functional (2.9) to vanish, leading to

$$\nabla_{\mathbf{u}} (\mathbf{u}^\top(t) R \mathbf{u}(t) + \nabla V(\mathbf{s}(t))^\top g(\mathbf{s}(t)) \mathbf{u}(t)) = 0 \implies \mathbf{u}^*(\mathbf{s}) = -R^{-1} g(\mathbf{s})^\top \nabla V(\mathbf{s}). \tag{2.10}$$

Substituting $\mathbf{u}^*$ back into the HJB equation gives us the closed form:

$$\nabla V(\mathbf{s})^\top f(\mathbf{s}) - \frac{1}{2} \nabla V(\mathbf{s})^\top g(\mathbf{s}) R^{-1} g(\mathbf{s})^\top \nabla V(\mathbf{s}) + \mathbf{s}^\top Q \mathbf{s} = 0. \tag{2.11}$$

In this approach, the control design (2.10) is expressed in **feedback** form, also known as **closed-loop control**, meaning that the optimal action is determined as a function of the state variable. This approach is particularly desirable as it ensures that the control adapts dynamically to changes (and external perturbations) in the state.

Curse of dimensionality

The Hamilton-Jacobi-Bellman equation (2.11) is a fully nonlinear, first-order PDE for which explicit solutions are generally unavailable, except in specific cases (e.g., linear-quadratic problems). Moreover, the HJB equation is posed over the state space of the optimal control problem, which can be of arbitrarily high dimension. This issue is particularly pronounced in agent-based models, where the state dimension scales with both the number N of agents and the dimension d of the underlying state space.

Following the notation introduced before, the ensemble state $\mathbf{s} = (s_1, \dots, s_N)^\top$ counts $n = 2dN$ entries. As a consequence, numerical methods that rely on state-space discretization, such as dynamic programming or finite difference schemes, suffer from an exponential growth in computational cost and memory requirements as n increases.

To circumvent these difficulties we approach the solution from suboptimality by means of numerical approximation. We present two alternative approaches: long horizon first-order necessary conditions by means of Pontryagin's Maximum Principle (PMP) and locally asymptotically stabilizing feedback design via State Dependent Riccati Equation (SDRE). These methods enable practical implementation of control strategies while mitigating the computational challenges posed by the curse of dimensionality.

2.1.2 Pontryagin's Maximum Principle

We now consider the finite-horizon optimal control problem, where the cost functional takes the form

$$\mathcal{J}(\mathbf{y}, \mathbf{u}(\cdot)) := \int_0^T \mathcal{L}(\mathbf{s}, \mathbf{u}; t)dt + \Psi(\mathbf{s}; T)\,, \tag{2.12}$$

where $\mathcal{L}$, Ψ encode the running and terminal cost respectively, and the system dynamics evolve according to (2.2) over the fixed time interval $[0, T]$, $T < +\infty$. We also introduce the associated Hamiltonian function, defined as

$$\mathcal{H}(\mathbf{s}, \mathbf{u}, \boldsymbol{\lambda}; t) = \boldsymbol{\lambda}^\top \big(f(\mathbf{s}(t)) + g(\mathbf{s}(t))\mathbf{u}(t)\big) - \mathcal{L}(\mathbf{s}, \mathbf{u}; t)\,, \tag{2.13}$$

which is defined as a function of the current state $\mathbf{s}$, control $\mathbf{u}$, and adjoint variable $\boldsymbol{\lambda}$, appearing in this formulation as a time-dependent Lagrange multiplier.

Theorem 2.1.3: Pontryagin's Maximum Principle [213]

Let the system dynamics $f(\mathbf{s})$ and $g(\mathbf{s})$ in (2.2) satisfy the assumptions of Carathéodory's theorem 2.1. Additionally, assume that the running cost $\mathcal{L}$ and terminal penalty Ψ in (2.12) to be continuously differentiable w.r.t. $\mathbf{s}$ so that the Hamiltonian (2.13) is differentiable in its arguments $(\mathbf{s}, \mathbf{u})$. If $(\mathbf{s}^*, \mathbf{u}^*)$ is an optimal trajectory, then there exists an adjoint variable $\boldsymbol{\lambda}^*(t)$ such that the following two-point boundary value problem holds:

$$\dot{\mathbf{s}}^*(t) = \partial_{\boldsymbol{\lambda}} \mathcal{H}(\mathbf{s}^*, \mathbf{u}^*, \boldsymbol{\lambda}; t), \qquad \text{(state)} \qquad (2.14)$$

$$\dot{\boldsymbol{\lambda}}^*(t) = -\partial_{\mathbf{s}} \mathcal{H}(\mathbf{s}^*, \mathbf{u}^*, \boldsymbol{\lambda}; t), \qquad \text{(adjoint)} \qquad (2.15)$$

$$\mathbf{s}^*(0) = \mathbf{y}\,, \quad \boldsymbol{\lambda}^*(T) = \nabla_{\mathbf{s}} \Psi(\mathbf{s}^*; T)\,, \qquad \text{(b.c.'s)} \qquad (2.16)$$

closed with first-order optimality condition for the optimal control:

$$\mathbf{u}^*(t) = \arg \min_{\mathbf{u}(\cdot) \in U} \mathcal{H}(\mathbf{s}^*, \mathbf{u}, \boldsymbol{\lambda}^*; t). \qquad (2.17)$$

For the quadratic case, where the running is of the form

$$\mathcal{L}(\mathbf{s}, \mathbf{u}; t) = \mathbf{s}^\top(t) Q \mathbf{s}(t) + \mathbf{u}^\top(t) R \mathbf{u}(t), \qquad (2.18)$$

with $0 \prec Q \in \mathbb{R}^{n \times n}$ and $0 \preceq R \in \mathbb{R}^{n \times m}$ symmetric, the optimality condition (2.17) simplifies to

$$\mathbf{u}^*(t) = R^{-1} g^\top(\mathbf{s}^*(t)) \boldsymbol{\lambda}^*(t). \qquad (2.19)$$

This procedure only provides the necessary conditions for optimality, meaning that PMP solutions are good candidates, with only some of them being optimal. Triples $(\mathbf{s}^*, \mathbf{u}^*, \boldsymbol{\lambda}^*)$ solving (2.14-2.17) are called *extremal trajectories*, and are computed departing from a fixed initial configuration $\mathbf{y}$. To determine whether an extremal trajectory is indeed globally optimal, additional verification techniques, such as second-order conditions or convexity arguments, are required.

The optimal control derived via PMP is time-dependent, meaning it is computed as a function of time rather than the current system state. This is often referred to as **open-loop control**, as opposed to feedback (closed-loop) control strategies. A significant limitation of this approach is its sensitivity to disturbances and uncertainties: any deviation from the nominal trajectory during execution cannot be corrected without re-solving the optimal control problem with an updated initial condition.

Reduced gradient iteration

The Pontryagin optimality system is a fully coupled system of forward and backward equations, with boundary conditions specified at $t = 0$ and the terminal time $t = T$ of the optimization horizon. To address this problem, we iteratively integrate the state equation (2.14) and the adjoint equation (2.15), while employing a gradient descent algorithm to update the control signal [140, 34, 32].

Following the derivation provided in [32], we reformulate the optimal control problem (2.12)-(2.2) with quadratic running cost (2.18) by expressing it as an unconstrained dynamic optimization problem depending on the control variable $\mathbf{u}$ alone:

$$\min_{\mathbf{u}(\cdot)} \Big\{ \bar{\mathcal{J}}(\mathbf{u}) := \mathcal{J}(\mathbf{s}, \mathbf{u}) \mid e(\mathbf{s}, \mathbf{u}) = 0 \Big\}, \tag{2.20}$$

where

$$e(\mathbf{s}, \mathbf{u}) := \begin{pmatrix} \dot{\mathbf{s}}(t) - (f(\mathbf{s}(t)) + g(\mathbf{s}(t))\mathbf{u}(t)) \\ \mathbf{s}(0) - \mathbf{y} \end{pmatrix}. \tag{2.21}$$

Assuming the solution operator $\mathbf{s}(\mathbf{u}) = \mathbf{s}(\mathbf{y}, \mathbf{u})$ to (2.2) to be well-defined and continuously differentiable, we obtain the Gateaux derivative of the reduced cost functional $\bar{\mathcal{J}}$ at $\boldsymbol{\nu} \in L_2(0, T; \mathbb{R}^m)$ in a direction $\delta\mathbf{u}$ by computing

$$\bar{\mathcal{J}}'(\boldsymbol{\nu})\delta\mathbf{u} = \langle \nabla\bar{\mathcal{J}}(\boldsymbol{\nu}), \delta\mathbf{u} \rangle_{L_2(0,T;\mathbb{R}^m)} = \langle (\mathbf{s}'(\boldsymbol{\nu}))^\dagger \partial_{\mathbf{s}} \mathcal{J}(\mathbf{s}, \boldsymbol{\nu}) + \partial_{\mathbf{u}} \mathcal{J}(\mathbf{s}, \boldsymbol{\nu}), \delta\mathbf{u} \rangle_{L_2(0,T;\mathbb{R}^m)} \tag{2.22}$$

where $\mathbf{s}'(\boldsymbol{\nu})$ is given by

$$\mathbf{s}'(\boldsymbol{\nu})\delta\mathbf{u} = -(\partial_{\mathbf{s}} e(\mathbf{s}, \boldsymbol{\nu}))^{-1} \partial_{\mathbf{u}} e(\mathbf{s}, \boldsymbol{\nu})\delta\mathbf{u}\,, \tag{2.23}$$

and the superscript † stands for the adjoint operator.

A sensitivity analysis on the OCP of interest leads to a first-order perturbation $\tilde{\mathbf{s}}$ of the state trajectory in response to variations in the control $\delta\mathbf{u}$ and an external perturbation $\boldsymbol{\phi}$. The perturbation $\tilde{\mathbf{s}}$ satisfies the linearized state sensitivity equation

$$\begin{cases} \dot{\tilde{\mathbf{s}}}(t) - \partial_{\mathbf{s}}(f(\mathbf{s}(t)) + g(\mathbf{s}(t))\mathbf{u}(t))\tilde{\mathbf{s}}(t) = \boldsymbol{\phi}, \\ \tilde{\mathbf{s}}(0) = \tilde{\mathbf{s}}_0, \end{cases} \tag{2.24}$$

where $\tilde{\mathbf{s}}_0$ represents the perturbation in the initial state. Similarly, the first-order perturbation $\tilde{\boldsymbol{\lambda}}$ of the adjoint trajectory in response to variations in the adjoint terminal condition $\tilde{\boldsymbol{\lambda}}_T$ and an external perturbation $\boldsymbol{\psi}$ satisfies the linearized adjoint equation:

$$\begin{cases} -\dot{\tilde{\boldsymbol{\lambda}}}(t) - (\partial_{\mathbf{s}}(f(\mathbf{s}(t)) + g(\mathbf{s}(t))\mathbf{u}(t)))^{\top}\tilde{\boldsymbol{\lambda}}(t) = \boldsymbol{\psi}, \\ \tilde{\boldsymbol{\lambda}}(T) = \tilde{\boldsymbol{\lambda}}_T. \end{cases} \tag{2.25}$$

It can be shown that $(\partial_{\mathbf{s}} e(\mathbf{s}, \mathbf{u}))^{-1}$ in (2.23), defined by $(\boldsymbol{\phi}, \tilde{\mathbf{s}}_0) \mapsto \tilde{\mathbf{s}}$, is the solution operator of equation (2.24), whilst the adjoint operator $(\partial_{\mathbf{s}} e(\mathbf{s}, \mathbf{u}))^{-\dagger}$, defined by $(\boldsymbol{\psi}, \tilde{\boldsymbol{\lambda}}_T) \mapsto \tilde{\boldsymbol{\lambda}}$, is the solution operator of equation (2.25). Using these elements, we can now compute the gradient as

$$\nabla\bar{\mathcal{J}}(\boldsymbol{\nu}) = \partial_{\mathbf{u}} J(\mathbf{s}, \mathbf{u}) - (\partial_{\mathbf{s}} e(\mathbf{s}, \mathbf{u}))^{-\dagger}(\partial_{\mathbf{u}} e(\mathbf{s}, \mathbf{u}))^{*}\partial_{\mathbf{s}} J(\mathbf{s}, \mathbf{u}). \tag{2.26}$$

Thus, we conclude that the reduced gradient reads

$$\nabla\bar{\mathcal{J}}(\mathbf{u})(t) = 2R\mathbf{u}(t) + g(\mathbf{s}(t))^{\top}\boldsymbol{\lambda}(t), \tag{2.27}$$

where $\boldsymbol{\lambda}$ is the solution to

$$\begin{cases} -\dot{\boldsymbol{\lambda}}(t) - (\partial_{\mathbf{s}}(f(\mathbf{s}(t)) + g(\mathbf{s}(t))\mathbf{u}(t)))^{\top}\boldsymbol{\lambda}(t) = \partial_{\mathbf{s}}\mathcal{J}(\mathbf{s}(t)), \\ \boldsymbol{\lambda}(T) = \nabla_{\mathbf{s}}\Psi(\mathbf{s}^*; T). \end{cases} \tag{2.28}$$

The explicit form (2.27) can be used to implement a gradient descent algorithm for solving the optimal control problem. Starting from an initial guess for the control $\mathbf{u}^0$, the control is updated iteratively as

$$\mathbf{u}^{k+1}(t) = \mathbf{u}^k(t) - \delta^k\Big(2R\mathbf{u}^k(t) - g(\mathbf{s}^k(t))^{\top}\boldsymbol{\lambda}^k\Big), \tag{2.29}$$

where $\mathbf{s}^k, \boldsymbol{\lambda}^k$ denote the state and adjoint trajectory associated with the control signal $\mathbf{u}^k$, and the step size δ^k is either fixed or adaptively determined at each iteration, for example, using the Barzilai-Borwein method [40]. The steps of this numerical scheme are summarized in Algorithm 1.

Algorithm 1: Reduced gradient iteration

Data: $\mathbf{y}$ initial condition; T time horizon; $\mathbf{u}^0$ control signal guess; $\varepsilon > 0$ tolerance
$k = 0$;
converged $\leftarrow$ *False*;
while *not converged* **do**
  $\mathbf{s}^k \leftarrow$ integration of (2.14) from $\mathbf{y}$ forward in time with control $\mathbf{u}^k$;
  $\boldsymbol{\lambda}^k(T) \leftarrow$ (2.16);
  $\boldsymbol{\lambda}^k \leftarrow$ integration of (2.15) from $\boldsymbol{\lambda}^k(T)$ backward in time with control $\mathbf{u}^k$;
  $\mathbf{u}^{k+1} \leftarrow \mathbf{u}^k(t) - \delta^k(2R\mathbf{u}^k(t) - g(\mathbf{s}^k(t))^{\top}\boldsymbol{\lambda}^k)$; `gradient step`
  converged $\leftarrow$ $\|\mathbf{u}^{k+1} - \mathbf{u}^k\|_2^2 < \varepsilon$;
  $k = k + 1$;
Result: Approximated extremal trajectory $(\mathbf{s}^*, \mathbf{u}^*, \boldsymbol{\lambda}^*)$

A key drawback of PMP is its sensitivity to the initial guess for the control signal $\mathbf{u}^0$. For Algorithm 1 to converge, the initial guess must correspond to an asymptotically stabilizing control. If the initial guess is inaccurate, the iterative process may converge to a suboptimal solution or fail to converge altogether. This dependence on a good initial guess can make PMP challenging to implement, especially in complex systems where determining a reliable initial guess is difficult.

On the suboptimality of PMP control signals We reiterate that the PMP approach is inherently suboptimal. The two-point boundary value problem defined by equations (2.14-2.16) provides necessary conditions for optimality, but does not guarantee sufficiency unless additional convexity conditions on the Hamiltonian and the value function are satisfied. In practice, if the Hamiltonian has multiple stationary points, the computed control may fulfill the first-order optimality conditions but fail to represent the true minimizing control.

Furthermore, solving the optimality system requires numerically integrating both the forward state dynamics and the backward costate equations. However, these equations are often stiff and highly sensitive to initial conditions, causing discretization errors to accumulate. This can degrade the quality of the resulting feedback law.

Another source of suboptimality arises from horizon truncation error. Unlike HJB-based approaches, which inherently encode the full cost-to-go, PMP only considers the backward-propagated costates from a finite horizon ($T < \infty$). This limitation can lead to myopic control behavior if the chosen horizon is insufficient to accurately approximate the infinite-horizon solution.

Model Predictive Control

To retrieve closed loop controls from this method, one can rely on subsequent open-loop solves departing from the current system configuration $\mathbf{s}$, and retrieve a control action of feedback nature by associating the first time instance of the control signal to $\mathbf{y} = \mathbf{s}$, i.e. $\mathbf{u}(\mathbf{s}) := \mathbf{u}(t = 0)$ along the trajectory departing at $\mathbf{s}$ [32]. Defining a uniformly spaced time-discretization on the time interval $[0, T]$ with N_{steps} steps, we denote the step size as $dt = \frac{T}{N_{\text{steps}}+1}$ and define the discrete time points as $t_h = h\,dt$ for $h = 0, 1, \ldots, N_{\text{steps}}$. The corresponding state at each time point is denoted by $\mathbf{s}_h := \mathbf{s}(t_h)$. Starting from the initial configuration of the system, $\mathbf{s}_0 = \mathbf{y}$, we employ a receding horizon strategy to guide the controlled evolution, as summarized in Algorithm 2.

Algorithm 2: MPC

Data: $\mathbf{y}$ initial condition; T time horizon; N_steps number of time steps in $[0, T]$; $\varepsilon > 0$ tolerance; $\mathbf{u}^0$ control signal guess

$dt = T/(N\_steps + 1)$

$\mathbf{s}_0 \leftarrow \mathbf{y}$

for $h = 1, \dots, N\_steps$ **do**

  $(\mathbf{s}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t)) \leftarrow$ Algorithm 1 $(\mathbf{s}_{h-1}, T, \mathbf{u}^0, \varepsilon)$

  $\mathbf{u}(\mathbf{s}_h) \leftarrow \mathbf{u}(t = 0)$; `feedback control`

  $\mathbf{s}_h \leftarrow$ integration of (2.14) from $\mathbf{s}_{h-1}$ for $t \in [0, dt]$ with control $\mathbf{u}(\mathbf{s}_h)$

Result: Feedback control $\mathbf{u}(\mathbf{s}_h)$ and controlled trajectory $\mathbf{s}_h$ for $h = 1, \dots, N\_steps$

On the link between PMP and the HJB equation

The connection between PMP and DP approach in finite horizon settings dates back to Bellman's early work and has been a central topic in optimal control theory since the foundational contributions of Pontryagin [213].

Under suitable regularity assumptions on the HJB solution (here $V \in \mathcal{C}^1$), the forward-backward dynamics derived from PMP can be interpreted as characteristic curves of the HJB equation [258, 227]. Consequently, the value function at a given space-time point is determined by solving the two-point boundary value problem (2.14-2.17), with its gradient corresponding to the adjoint variable [39, 85]. We now outline this connection and refer to [38] for further details.

In finite horizon setting, the value function is defined as

$$V(\mathbf{y}, t) = \inf_{\mathbf{u}(\cdot) \in U_{adm}} \left\{ \int_t^T \mathcal{L}(\mathbf{s}, \mathbf{u}; \tau) d\tau + \Psi(\mathbf{s}; T) \right\} \tag{2.30}$$

where $\mathbf{s}$ evolves according to controlled dynamics as in (2.2). The associated HJB equation reads

$$\partial_t V(\mathbf{s}, t) + \inf_{\mathbf{u}(\cdot) \in U_{adm}} \left\{ (f(\mathbf{s}) + g(\mathbf{s})\mathbf{u}) \cdot \nabla V(\mathbf{s}, t) + \mathcal{L}(\mathbf{s}, \mathbf{u}; t) \right\} = 0, \tag{2.31}$$

recalling the definition of Hamiltonian in (2.13) and selecting $\boldsymbol{\lambda}(t) = \nabla V(\mathbf{s}, t)$ as costate,

this can be rewritten as

$$\partial_t V(\mathbf{s}, t) + \inf_{\mathbf{u}(\cdot) \in U_{adm}} \mathcal{H}(\mathbf{s}, \mathbf{u}, \nabla V; t) = 0\,. \tag{2.32}$$

Assuming the existence of an optimal control $\mathbf{u}^*$, the infimum is attained and leads to

$$\partial_t V(\mathbf{s}, t) = -\mathcal{H}(\mathbf{s}^*, \mathbf{u}^*, \nabla V; t). \tag{2.33}$$

We conclude by considering the state gradient on both sides, obtaining the adjoint equation (2.15) for $\boldsymbol{\lambda}^*(t) = \nabla V(\mathbf{s}^*, t)$:

$$\partial_t \nabla V(\mathbf{s}^*, t) = -\nabla_{\mathbf{s}} \mathcal{H}(\mathbf{s}^*, \mathbf{u}^*, \nabla V(\mathbf{s}^*); t)\,, \tag{2.34}$$

with terminal condition $\nabla V(\mathbf{s}^*, T) = \nabla_{\mathbf{s}} \Psi(\mathbf{s}^*; T)$. This describes how the value function gradient evolves along an optimal trajectory, effectively establishing the connection between the PMP optimality system and the HJB equation.

This interpretation does not readily extend to infinite horizon optimal control problems of the type (2.1-2.2), as the final condition for the adjoint variable would pathologically be set at infinity. Several methods have been proposed in the literature to approach $T \to \infty$ with PMP [63]; in this present work, we simply consider a finite horizon T *long enough* for the system to reach consensus [24].

2.1.3 State Dependent Riccati Equation

As an alternative approach for circumventing the direct solution of the HJB PDE for V, we resort to the State Dependent Riccati Equation (SDRE) approach, providing suboptimal feedback laws that ensure locally stabilizing effect at the limits. This is applicable to nonlinear quadratic optimal control problems in infinite horizon setting (2.1-2.2), where the system dynamics can be represented in semi-linear form

$$\dot{\mathbf{s}} = \mathcal{A}(\mathbf{s})\mathbf{s} + \mathcal{B}(\mathbf{s})\mathbf{u}\,, \tag{2.35}$$

with state-to-state matrix valued map $f(\mathbf{s}) = \mathcal{A}(\mathbf{s})\mathbf{s}$, $\mathcal{A} : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ and control operator $g(\mathbf{s}) = \mathcal{B}(\mathbf{s})$, with $\mathcal{B} : \mathbb{R}^n \to \mathbb{R}^{n\times m}$. Carrying over the assumtions on the system dynamics in (2.2), we further have $\mathcal{A}, \mathcal{B} \in C^1$, satisfying $\mathcal{A}(\mathbb{O}_n) = \mathbb{O}_{n\times n}$ and $\mathcal{B}(\mathbb{O}_n) = \mathbb{O}_{n\times m}$. In what follows, we further assume pointwise controllability in the state space, i.e. $\forall \mathbf{s} \in \mathbb{R}^n$ the pair $(A(\mathbf{s}), B(\mathbf{s}))$ is controllable [23].

This method is based on the idea that the value function (2.4) is non-negative by construction, and with no loss of generality it can be represented as a quadratic form $V(\mathbf{s}) = \mathbf{s}^\top \Pi(\mathbf{s})\mathbf{s}$, with $\Pi(\mathbf{s}) \in \mathbb{R}^{n\times n}$ a symmetric matrix-valued function. In order to better describe the difficulties related to the solution of this optimal control problem, we first focus on the linear quadratic case, where the optimal solution is computed via linear quadratic regulator (LQR). The optimality of the feedback law is ensured by a direct link between the LQR solution and the dynamic programming one, obtained from equation (2.11). In non-linear settings, the parallelism between the non-linear quadratic regulator (NLQR) and the dynamic programming approach is broken, together with the optimality of the solution.

Linear Quadratic Regulator

Under linearity assumption, i.e. $f(\mathbf{s}) \equiv A \in \mathbb{R}^{n\times n}$, $g(\mathbf{s}) \equiv B \in \mathbb{R}^{n\times m}$, by using the quadratic ansatz $V(\mathbf{s}) = \mathbf{s}^\top \Pi \mathbf{s}$, $\Pi \in \mathbb{R}^{n\times n}$, we can rewrite the HJB equation (2.11) as

$$\inf_{\mathbf{u}} \left\{ 2\mathbf{s}^\top \Pi A \mathbf{s} + 2\mathbf{s}^\top \Pi B \mathbf{u} + \frac{1}{2}\mathbf{u}^\top R \mathbf{u} + \mathbf{s}^\top Q \mathbf{s} \right\} = 0\,. \tag{2.36}$$

which can be solved as before by means of first order optimality conditions, leading to an optimal feedback of the form

$$\mathbf{u}^*(\mathbf{s}) = -\mathcal{K}\mathbf{s} = -R^{-1}B^\top \Pi \mathbf{s}\,. \tag{2.37}$$

and substituting it in (2.36), we obtain

$$2\mathbf{s}^\top \Pi A \mathbf{s} - \mathbf{s}^\top \Pi B R^{-1} B^\top \Pi \mathbf{s} + \mathbf{s}^\top Q \mathbf{s} = 0\,. \tag{2.38}$$

For (2.38) to hold for every state $\mathbf{s} \in \mathbb{R}^n$, Π satisfies the following algebraic Riccati equation (ARE):

$$\Pi A + A^\top \Pi - \Pi B R^{-1} B^\top \Pi + Q = 0\,. \tag{2.39}$$

We highlight that the feedback (2.37) is derived directly from the application of the dynamic programming principle in the linear quadratic case, ensuring its optimality.

Nonlinear Quadratic Regulator

In order to extend LQR to the nonlinear case, we re-arrange the dynamics in semilinear state-dependent form as in (2.35) and we impose the quadratic ansatz $V(\mathbf{s}) = \mathbf{s}^\top \Pi(\mathbf{s})\mathbf{s}$. Similarly to the LQR design, the feedback control policy can be calculated via first order optimality conditions. By approximating $\nabla V(\mathbf{s}) \approx 2\Pi(\mathbf{s})\mathbf{s}$ and inserting it into the HJB equation (2.11), we obtain the State-dependent Riccati Equation

$$\mathcal{A}^\top(\mathbf{s})\Pi(\mathbf{s}) + \Pi(\mathbf{s})\mathcal{A}(\mathbf{s}) - \Pi(\mathbf{s})\mathcal{B}(\mathbf{s})R^{-1}\mathcal{B}^\top(\mathbf{s})\Pi(\mathbf{s}) + Q = 0 \tag{2.40}$$

to be solved for $\Pi(\mathbf{s})$. The resulting feedback law then reads

$$\mathbf{u}(\mathbf{s}) = -\mathcal{K}(\mathbf{s})\mathbf{s} = -R^{-1}\mathcal{B}^\top(\mathbf{s})\Pi(\mathbf{s})\mathbf{s}\,. \tag{2.41}$$

On the suboptimality of SDRE feedback laws In the SDRE approach, suboptimality arises primarily from the way it approximates the optimal control solution by enforcing a state-dependent linearization of the system dynamics. Instead of solving the fully nonlinear optimal control problem, SDRE reformulates it as a sequence of locally linear-quadratic subproblems. A key issue with this approach is that the semi-linearization of the free dynamics, given by $f(\mathbf{s}) = \mathcal{A}(\mathbf{s})\mathbf{s}$, is not unique for state dimensions $n > 1$. This non-uniqueness means that different choices of $\mathcal{A}(\cdot)$ lead to different Riccati equations and, consequently, different feedback control laws. While this produces a stabilizing feedback controls, it does not necessarily yield the globally optimal law. A more fundamental source of suboptimality lies in the approximation of the value function gradient. Equa-

tion (2.40) holds for $\nabla V(\mathbf{s}) \approx 2\Pi(\mathbf{s})\mathbf{s}$, whereas a chain rule calculation accounting for the state-dependency of $\Pi(\cdot)$ shows

$$\nabla V(\mathbf{s}) = 2\Pi(\mathbf{s})\mathbf{s} + \mathbf{s}^\top \nabla \Pi(\mathbf{s})\mathbf{s}\,. \tag{2.42}$$

which is not captured in the SDRE (2.40), leading to a discrepancy w.r.t. the true optimal control derived from the HJB solution. These two sources of suboptimality are entertwined, as the choice of $\mathcal{A}(\cdot)$ directly influences the quality of the SDRE approximation. A way to leverage this fact is to optimize the choice of $\mathcal{A}(\cdot)$ to minimize this misalignment [143, 101], potentially reducing suboptimality while maintaining computational tractability.

Anoter significant limitation of the SDRE method is its dependence on the stability of the system's eigenvalues. If the eigenvalues of the linearized system become unstable, the Riccati equation may fail to converge to a stabilizing solution, leading to instability in the closed-loop system. As a result, the SDRE-based controller can become ineffective in controlling systems with unstable eigenvalues, limiting its applicability in such scenarios.

MPC with SDRE update

The state dependency of the feedback operator $\mathcal{K}(\mathbf{s})$ in (2.41) suggests the need for solutions $\Pi(\mathbf{s})$ of (2.40) at every $\mathbf{s} \in \Omega \subset \mathbb{R}^n$, where n can be arbitrarily large. Thus, solving the SDRE (2.40) does not offer any computational advantage compared to directly solving the original HJB equation (2.11). Between the several numerical approaches that have been proposed to address this, we cite Taylor series method [26] and interpolation for $\Pi(\mathbf{s})$ [35], while we refer the interested reader to [189] for a more exhaustive review.

Following a similar approach as in [36], we realize the feedback design in a model predictive control fashion: given the current configuration $\bar{\mathbf{s}}$ of the system, we freeze the operator $\Pi(\bar{\mathbf{s}}) \equiv \Pi \in \mathbb{R}^{n\times n}$, meaning that (2.40) reduces to its algebraic form (2.39), where all the state dependencies are neglected by accordingly evaluating all the operators at $\bar{\mathbf{s}}$. The system is then evolved for a short time, after which the current state $\bar{\mathbf{s}}$ is updated. This

procedure requires subsequently solving the linear quadratic problem associated with the frozen system, which is addressed via the linear quadratic regulator routine.

This numerical procedure is sketched in Algorithm 3, where a controlled trajectory is computed along the time interval $[0, T]$. Defining a uniformly spaced time-discretization on the time interval $[0, T]$ with N_{steps} steps and discrete times $t_h = h\,dt$ for $h = 0, 1, \ldots, N_{\text{steps}}$. Departing from the initial configuration of the system $\mathbf{s}(t_0) = \mathbf{y}$, we follow a receding horizon strategy for the controlled evolution.

Algorithm 3: MPC with SDRE update

Data: $\mathbf{y}$ initial condition; T time horizon; N_steps number of time steps in $[0, T]$

$dt = T/(N\_steps + 1)$

$\mathbf{s}(t_0) \leftarrow \mathbf{y}$; `initial condition`

for $h = 0, ..., N\_steps$ **do**

- $\bar{\mathbf{s}} \leftarrow \mathbf{s}(t_h)$; `read current state`
- $A \leftarrow \mathcal{A}(\bar{\mathbf{s}}),\ \ B \leftarrow \mathcal{B}(\bar{\mathbf{s}})$; `freeze the system accordingly`
- solve (2.40) for Π;
- $\mathbf{u} \leftarrow -(R + B^\top \Pi B)^{-1} B^\top \Pi A \bar{\mathbf{s}}$; `approximated feedback control`
- $\mathbf{s}(t_{h+1}) \leftarrow \bar{\mathbf{s}} + dt(A\bar{\mathbf{s}} + B\mathbf{u})$; `the control system evolves for` dt

Result: Feedback controlled trajectory $\mathbf{s}(t_h)$ for $h = 1, \ldots, N\_steps$

2.1.4 PMP vs. SDRE update for model predictive control, a comparison

We conclude this Optimal Control background section by providing some comparative remarks to guide the choice between the two methodologies in different problem settings. In Fig. 2.1, we sketch the MPC update computation using SDRE versus PMP to approximate the feedback control.

From an **implementation perspective**, the SDRE approach is particularly effective when system dynamics allow for a semi-linear factorization. It benefits from well-established Riccati solvers, which typically have polynomial complexity $O(n^3)$ in the system's state dimension. While efficient for moderately large-scale systems, it becomes computationally prohibitive for very high-dimensional problems due to intensive matrix operations.

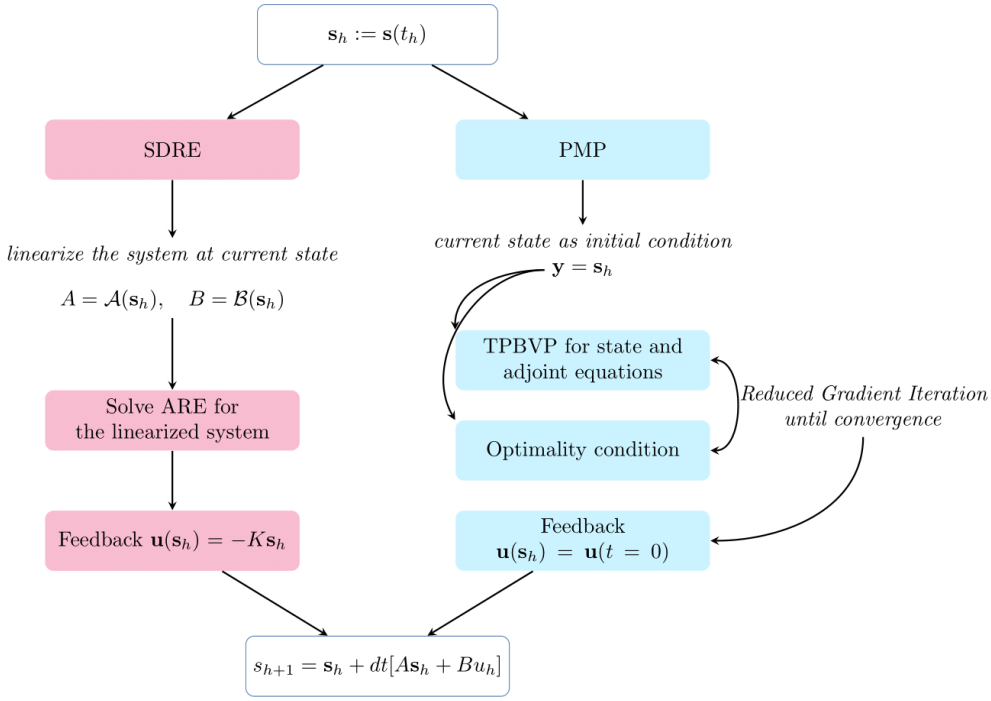


Figure 2.1: MPC update computed via SDRE versus PMP.

On the other hand, PMP-based control synthesis requires solving adjoint equations, making implementation more intricate. However, it can exploit problem-specific sparsity and structural properties to improve efficiency. In scenarios where agent interactions are sparse, each agent's costate equation (2.15) depends only on a subset of other agents, enabling problem decomposition into smaller, independent subproblems. This facilitates scalable, distributed computation, reducing overall complexity. Inherent system symmetries can further simplify the problem by enforcing common optimality conditions across similar agents [195].

Both methods provide approximation of the optimal feedback control associated with the solution of the HJB equation (2.11). The resulting state-to-control maps are suboptimal by construction. A fist common reason for **suboptimality** is the receding horizon approach, which inherently introduces a numerical integration of the controlled trajectory, but deeper reasons can be found for each numerical routine, as discussed earlier. A summary of the comparison between PMP and SDRE updates for MPC can be found in Table 2.1.

For high-dimensional problems, both PMP and SDRE-based control synthesis can still

	PMP update	**SDRE update**
Formulation	TPBVP for state and adjoint equations + optimality condition	ARE for the linearized system at current state
Suboptimality	• Truncation error for the optimization horizon $T < +\infty$ • Necessary but non-sufficient optimality condition	• Non-uniqueness of semilinearization $f(\mathbf{s}) = \mathcal{A}(\mathbf{s})\mathbf{s}$ for state dimensions $n > 1$ • Inexact gradient of the value function $\nabla V(\mathbf{s}) \approx 2\Pi(\mathbf{s})\mathbf{s}$ for nonlinear systems
Numerical solvers	Reduced Gradient Iteration	Linear Quadratic Regulator
Complexity	Iterative, can exploit sparsity and structural properties of the system	Riccati solvers are $\mathcal{O}(n^3)$ due to intensive matrix operations
Shortcomings	• Asymptotically stabilizing initial guess for the control signal is required • The state and adjoint equations are often stiff	• Stability of the system's eigenvalues is required • Can only be applied to nonlinear quadratic OCP that admit semilinearization.

Table 2.1: Comparison of PMP and SDRE updates for Model Predictive Control

demand significant computational resources in a receding horizon framework. These challenges motivate the exploration of alternative methods for efficient and scalable feedback control approximation.

2.2 Data-driven control strategies

The numerical approximation of high-dimensional Hamilton-Jacobi-Bellman equations in deterministic optimal control has been widely studied. While an exhaustive review is beyond our scope, we highlight several key approaches reported in the literature.

Classical grid-based methods include sparse grids [121, 150, 149], tree-structured algorithms [22], and max-plus methods [3], each designed to mitigate the curse of dimensionality in structured settings. Another line of research leverages viscosity solutions [233, 129],

providing theoretical guarantees for convergence. Additionally, representation formulas for HJB equations have been combined with fast convex optimization solvers [83, 80, 93], enabling efficient feedback synthesis in certain problem classes.

Beyond direct numerical solvers, alternative approaches aim to approximate the feedback control law without explicitly solving the HJB equation. One early example is [43], where feedback policies were constructed by interpolating finite-horizon open-loop solutions, reducing the computational burden for high-dimensional systems. Polynomial-based methods, including sparse polynomial approximations [145, 146, 32], and tensor decomposition techniques, such as tensor trains [130, 196, 197, 100, 102], have also been explored for scalable approximations.

In recent years, artificial neural networks (NNs) have gained prominence as a powerful tool for approximating solutions to high-dimensional HJB equations. These methods can be broadly classified into unsupervised learning techniques [92, 82] and supervised learning approaches [148, 81, 187, 186, 147], with [198] offering a comparative study of different approximation models. More generally, deep learning algorithms for solving PDEs have been extensively investigated in works such as [252, 109, 135, 169], highlighting their potential beyond HJB-specific applications.

Building on this extensive body of work, the majority of the projects presented in this thesis utilize a supervised learning-based computational method to accelerate the solution of large-scale optimal control problems. By circumventing the explicit solution of the HJB PDE, this approach relies on offline approximations of the feedback law, significantly reducing the computational burden during online implementation. In particular, we focus on neural network approximation models, which are briefly introduced in the remainder of this section.

2.2.1 Feedforward Neural Networks

An artificial neural network of feedforward type is associated with a directed acyclic graph $(\mathcal{V}, \mathcal{E})$ defining the *architecture* of the model, with $\mathcal{E}$ finite set of directed edges and $\mathcal{V}$ a

finite set of vertices (or *nodes*), which is partitioned in three proper distinguished subsets:

- Set I consisting of the input vertices, having only outgoing edges, while receiving input scalars from outside the network.
- Set O containing the output vertices, with only incoming edges. For every network input, these vertices will store the corresponding output value of the model.
- Set $H = \mathcal{V} \setminus \{I, O\}$ of hidden vertices storing intermediate values.

Vertices $v \in \mathcal{V} \setminus I$ are associated with a non-linear *activation function* $\sigma_v : \mathbb{R} \to \mathbb{R}$ and a *bias* $b_v \in \mathbb{R}$, whilst for every edge $e \in \mathcal{E}$ there is a *weight* $w_e \in \mathbb{R}$. The weights and biases are the *trainable parameters* of the network, and they define a family of many output functions. At each vertex $v \in \mathcal{V} \setminus \mathcal{I}$ we place a computational unit (*neuron*) defined as

$$x_v := \sigma_v\left(b_v + \sum_{\substack{v' \text{ s.t. } \exists e \in \mathcal{E} \\ e=(v',v)}} w_e x_{v'}\right), \tag{2.43}$$

where $x_{v'}$ is the output of the neuron at vertex v'.

The first class of model we will consider is the one of fully-connected FNNs, where the vertices are organized into layers, and each vertex of every layer is connected via outgoing edges to all vertices in the next one. This family of models is characterized by the flowing of information from the input nodes to the output ones without forming any cycles or loops, as shown in Fig. 2.2.

Under the above assumptions, these architectures approximate general functions ψ through a sequence of layered transformations $l_1, \ldots, l_K$, resulting in $\tilde{\psi} = \psi_\theta \approx \psi$ where

$$\psi_\theta(\mathbf{x}) = l_K \circ \ldots \circ l_2 \circ l_1(\mathbf{x}), \qquad l_k(\mathbf{x}) = \sigma_k(\mathbf{W}_k \mathbf{x}_k + \mathbf{b}_k). \tag{2.44}$$

Each layer l_k applies a nonlinear activation function σ_k element-wise to a linear transformation of its input $\mathbf{x}_k$. Assuming the k-th layer to have n_k neurons, the parameters $\mathbf{W}_k \in \mathbb{R}^{n_{k-1} \times n_k}$ and $\mathbf{b}_k \in \mathbb{R}^{n_k}$ represent the weight matrix and bias vector for the k-th

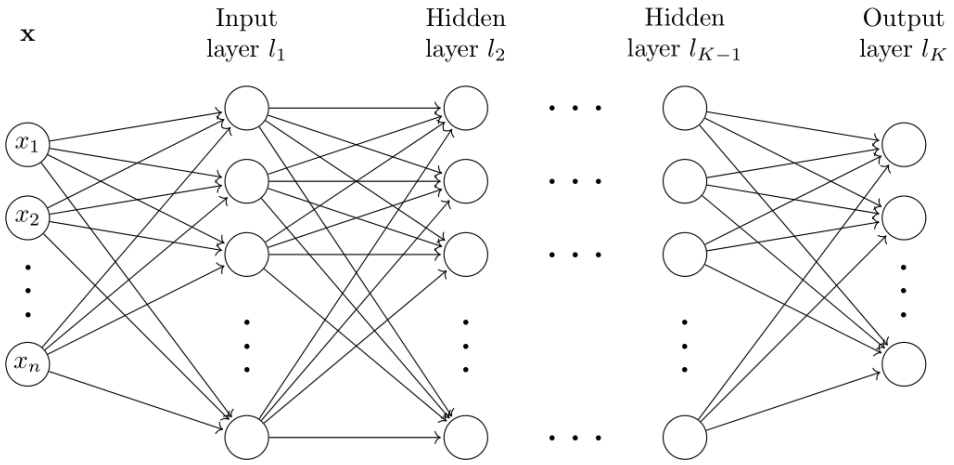


Figure 2.2: Directed acyclic graph of a fully connected FNN.

layer respectively, defined as

$$\mathbf{b}_k = \begin{bmatrix} b_{v_1} \\ b_{v_2} \\ \vdots \\ b_{v_{n_k}} \end{bmatrix} \qquad \Big[\mathbf{W}_k\Big]_{i,j} = w_{(v'_i, v_j)} \tag{2.45}$$

where v'_i and v_j denote vertices in the $(k-1)$-th and k-th layers, respectively, with $i = 1, \ldots, n_{k-1}$ and $j = 1, \ldots, n_k$. We use the same activation function for all neurons within a given layer, with $\sigma_1 = \sigma_K$ being the identity function.

2.2.2 Recurrent Neural Networks

Recurrent neural networks are a class of models which can be obtained from FNNs by allowing loop connections between layers. This architecture is particularly well-suited for processing sequential data. When a sequence is fed as an input, while the output is a single vector, the RNN is said *many-to-one*; vice versa, a RNN receiving a single input vector and producing a sequence of outputs is called *one-to-many*. A *many-to-many* RNN is characterized by input and output of sequential type, while *one-to-one* RNNs deal with non-sequential data.

In this thesis, we extend FNNs into RNNs by allowing hidden layers to be one-to-one

Long Short Term Memory (LSTM) cells which – for every input $\mathbf{x}$ – generate output h according to the following system of equations

$$\begin{aligned} i &= \rho(W_i\,\mathbf{x} + b_i) && \textit{input gate} \\ \tilde{c} &= \sigma(W_c\,\mathbf{x} + b_c) && \textit{candidate value} \\ c &= i \odot \tilde{c} && \textit{cell value} \\ o &= \rho(W_o\,\mathbf{x} + b_o) && \textit{output gate} \\ h &= o \odot \sigma(c) && \textit{final output} \end{aligned} \tag{2.46}$$

where (assuming $\mathbf{x}_k \in \mathbb{R}^{n_k}$), the weight matrices W_i, W_c, $W_o \in \mathbb{R}^{n_k \times n_{k+1}}$ and the bias vectors b_i, b_c, $b_o \in \mathbb{R}^{n_{k+1}}$ are the trainable parameters, $\sigma(\cdot)$ is the activation function, and $\rho(\cdot)$ is the recurrent activation function. The number n_k denotes the number of neurons in the $k-$th cell. For convenience, we denote $\mathbf{W}_k = (W_i, W_c, W_o)^\top$ and $\mathbf{b}_k = (b_i, b_c, b_o)^\top$.A visual representation of the flux of information through a LSTM cell can be found in Figure 2.3.

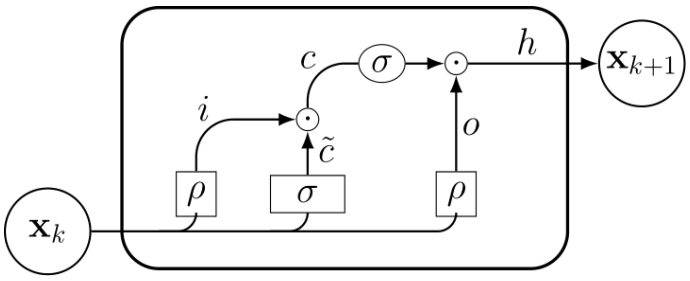


Figure 2.3: One-to-one LSTM cell, visualization of equation (2.46)

2.2.3 Training

In a supervised learning environment, the trainable parameters $\theta = \{\mathbf{W}_k, \mathbf{b}_k\}_{k=1}^{K}$ are to be computed as minimizers of a suitable *loss function* measuring the approximation error within a set of sampled data (*training set*) $\mathcal{T} = \{\mathbf{x}^{(i)}, \psi^{(i)}\}_{i=1}^{N_s}$ where $\psi^{(i)} := \psi(\mathbf{x}^{(i)})$:

$$\theta^* = \underset{\theta}{argmin}\ \mathcal{L}^{NN}(\psi, \psi_\theta)\,, \qquad \mathcal{L}^{NN}(\psi, \psi_\theta) = \sum_{\mathcal{T}} l^{NN}(\psi^{(i)}, \psi_\theta(\mathbf{x}^{(i)}))\,. \tag{2.47}$$

In particular, we consider

$$\mathcal{L}^{NN}(\psi, \psi_\theta) = MSE(\psi, \psi_\theta)\,, \qquad l^{NN}(\psi^{(i)}, \psi_\theta(\mathbf{x}^{(i)})) = \frac{\|\psi^{(i)} - \psi_\theta(\mathbf{x}^{(i)})\|^2}{N_s}\,. \tag{2.48}$$

Training convergence is assessed via the loss $\mathcal{L}^{NN}$, computed on a held-out validation set. Models are trained for a fixed number of epochs, and the best-performing configuration is selected based on minimum validation loss. The number of layers K, the number of neurons per layer, the set of activation functions $\{\sigma_k\}_{k=2}^{K-1}$ in the hidden layers, and eventual other parameters represent further degrees of freedom defining the *architecture* Θ of the NN. They are to be computed as result of the *hyper-parameter tuning phase*, which amounts to choose, between trained models $\psi_\theta[\Theta]$ having different architectures, the one performing better w.r.t. some error measure (e.g. the *root mean squared error*)

$$\Theta^* = \underset{\tilde{\Theta}}{argmin}\ \sqrt{MSE(\psi, \psi_\theta[\Theta])}\,, \qquad MSE(\psi, \psi_\theta) = \frac{1}{N_t}\sum_{\mathcal{T}_v} \frac{\|\psi^{(j)} - \psi_\theta(\mathbf{x}^{(j)})\|^2}{\|\psi^{(j)}\|^2} \tag{2.49}$$

where $\mathcal{T}_v = \{\mathbf{x}^{(j)}, \psi^{(j)}\}_{j=1}^{N_v}$ is the *validation set* such that $\mathcal{T} \cap \mathcal{T}_v = 0$.

2.2.4 Example: Gradient-augmented supervised learning of SDRE feedback laws

We proceed by providing an example of the NN approximation routine for SDRE feedback controls in the context of PDE-constrained optimization, based on the work published in [8]. This test builds upon a similar example from [100], where we consider the control of the nonlinear Allen-Cahn PDE

$$\partial_t x(\xi, t) = 0.1\partial^2_{\xi\xi} x + x(1 + x^2) - \chi_\omega(\xi)u(t) \tag{2.50}$$

in $[0, 1] \times \mathbb{R}_+$ with Neumann boundary conditions, where the scalar control signal $u : [0, +\infty] \to \mathbb{R}$ acts through the indicator function of the interval $\omega = [0.6, 0.9]$. Without control action, these dynamics are bistable with $x \equiv \pm 1$ being the stable equilibria. We

are interested in minimizing

$$\mathcal{J}(\mathbf{y}, \mathbf{u}(\cdot))(u, x) = \int_0^{+\infty} ||x(\xi, t)||^2 + 0.1u^2(t)dt\,, \tag{2.51}$$

thus stabilizing the dynamics towards the equilibrium $x = 0$. The PDE (2.50) is discretized in space via finite differences with $N = 51$ degrees of freedom, leading to a system of nonlinear ODEs

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{x} \odot (1 + \mathbf{x} \odot \mathbf{x}) + \mathbf{B}u(t), \tag{2.52}$$

where $\mathbf{x}(t) = (x(\xi_1, t), ..., x(\xi_N, t))$ is the discrete state, $\odot$ denotes the Hadamard product and $\mathbf{A}, \mathbf{B}$ correspond to a discretization of the Laplace operator and the indicator function $\chi_\omega(\xi)$ over a uniform grid $\{\xi_i\}_{i=1}^N$.

Synthetic data generation

The control laws to be approximated rely on the pointwise solution of the SDRE (2.40), for which we resort to the `lqr` routine in MATLAB. The samples for training were generated from initial condition vectors $\{\mathbf{x}^{(i)}\}_{i=1}^{N_s} \in \Omega \subset \mathbb{R}^n$, computed using Halton quasi-random sequences in $[0, 1]^n$.

Similarly to what is presented in Section 3.4, we consider two alternative NN approximation targets:

- **Learning $\mathbf{u}(\mathbf{x})$**: we train a model for the vector-valued feedback law $\mathbf{u}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^m$ upon a set of N_s training states $\{\mathbf{x}^{(i)}\}_{i=1}^{Ns}$, the solution of the corresponding $\Pi(\mathbf{x})$, and the controls $\mathbf{u}(\mathbf{x})$ via (2.37).

- **Learning $V(\mathbf{x})$** we train a model for the scalar function $V(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ from $V(\mathbf{x}) = \mathbf{x}^\top \Pi(\mathbf{x})\mathbf{x}$ and its gradient $\nabla V(\mathbf{x}) = 2\Pi(\mathbf{x})$, where $\Pi(\mathbf{x})$ is a positive definite solution of (2.40) for each $\mathbf{x}^{(i)}$. The feedback law is then expressed as $\mathbf{u}(\mathbf{x}) = -\frac{1}{2}\mathbf{R}^{-1}\mathbf{B}(\mathbf{x})^\top \nabla V(\mathbf{x})$.

Both alternatives are a direct supervised learning formulation of the SDRE approach, with the common objective of synthesizing a feedback requiring a reduced number of operations

for online implementation. The advantage of the second formulation resides in the training of a scalar function, for which both function and gradient values are available as a by-product of solving the SDRE. Once the solution of the SDRE is computed for each sample $\mathbf{x}^{(i)}$, $i = 1, \dots, N_s$, the training set $\{\mathbf{x}^{(i)}, \mathbf{u}(\mathbf{x}^{(i)})\}_{i=1}^{N_s}$ for u_θ can be computed as in (2.37), while the ANN V_θ is trained upon an enriched dataset, containing both the value function $V(\mathbf{x}^{(i)})$ and its gradient $\nabla V(\mathbf{x}^{(i)})$.

Training and goodness of fit

The sampling datasets are split into *training set* $\mathcal{T}$ and *validation set* $\mathcal{T}_v$ with a ratio of 80/20. For training the different models, we rely on the mean squared error as measure of the approximation error $\mathcal{L}_0^{NN}$. In the gradient-augmented regression for V, we additionally incorporate a penalty term to account for discrepancies in the derivative information:

$$\begin{aligned} \mathcal{L}_0^{NN}(\psi, \psi_\theta) &= \sum_{i=1}^{N_s} \frac{\|\psi^{(i)} - \psi_\theta(\mathbf{x}^{(i)})\|^2}{N_s}, \\ \mathcal{L}_1^{NN}(\psi, \psi_\theta) &= \mathcal{L}_0^{NN}(\psi, \psi_\theta) + \mu \sum_{j=1}^{n} \mathcal{L}_0^{NN}\left(\frac{\partial \psi}{\partial x_j}, \frac{\partial \psi_\theta}{\partial x_j}\right), \end{aligned} \tag{2.53}$$

where x_j denotes the $j-$th dimension of the input variable $\mathbf{x}$, and ψ is a general function of the state $\mathbf{x}$ approximated by the NN ψ_θ. In this example, we focus on FNN models and select the architecture based on the goodness of fit in the validation set, measured using the coefficient of determination

$$r^2 = 1 - \frac{\sum_{\mathcal{T}_v} \|\psi(\mathbf{x}^{(j)}) - \psi_\theta(\mathbf{x}^{(j)})\|^2}{\sum_{\mathcal{T}_v} \|\psi(\mathbf{x}^{(j)}) - \bar{\psi}\|^2}, \tag{2.54}$$

for $\bar{\psi}$ sample mean of the target variable in the data.

The minimization of the loss function (2.53) is performed using the quasi-Newton method `lbfgs`. Other than that, the parameters to be optimized are the number of hidden layers, the number of neurons per layer, the activation function, and the number of epochs taken into account during the training (we fixed the batches' size to 100). For $V_\theta(\mathbf{x})$, we also optimize the hyper-parameter μ weighting the gradient term in $\mathcal{L}_1^{NN}$. The goodness of fit

of the trained models is finally evaluated in the *test set*, a uniform grid of $N = 10^4$ points within the state space, where the approximated control is compared with the pointwise computation through the SDRE solution.

Numerical Test

We consider a dataset $\{\mathbf{x}^{(i)}, V(\mathbf{x}^{(i)}), \nabla V(\mathbf{x}^{(i)})\}_{i=1}^{N_s}$ with $N_s = 1000$, where the states have been sampled from $[-2, 2]^{51}$. We train a model for V_θ with 3 hidden layers, 500 neurons per layer, and activation function $\sigma(x) = \max(0, x)$. The best configuration of hyper-parameters is found to be $\mu = 7.5$, with the NN being trained for 71 epochs. Finally, we test the trained model V_θ in a test grid of points in $[-2, 2]^{51}$. For the model $\mathbf{u}_\theta$, the architecture is built with 4 hidden layers, with 500 neurons per layer, and activation function $\sigma(x) = \max(0, x)$. In this example, also the output layer for $\mathbf{u}_\theta$ is made only of a single neuron, since the feedback law is a scalar in $\mathbb{R}$. The model was trained for 50 epochs.

predicted variable	r^2	*MSE*
V_θ	0.81681	0.00024628
∇V_θ	0.87114	0.00025852
$\mathbf{u}_V$	0.91976	0.012181
$\mathbf{u}_\theta$	0.85443	0.022098

Table 2.2: Goodness of fit.

In Figs 2.4 and 2.5 we compare the trajectories resulting from the integration of the discretized dynamics (2.52) with $t \in [0, 10]$, for an initial condition $x(\xi, 0) = 1 + (1 - \xi)\xi$, and different feedback laws: the constant zero function, the feedback obtained considering the linear control operator $\mathbf{K}_0$, the control resulting from the gradient-augmented approximation V_θ, and the one given by $\mathbf{u}_\theta$. In this high-dimensional local problem, the approximation done through the gradient-augmented model V_θ happens to outperform $\mathbf{u}_\theta$ in terms of goodness of fit. On the other hand, observing the different closed-loop evolutions and control signals, we can see how both approximate feedback laws succeed in stabilizing the trajectories near $x = 0$, while the uncontrolled system is stable in $x = 1$ and the $\mathbf{u}_0$ results in a system which, for $t = 10$ has not yet approached the equilibrium.

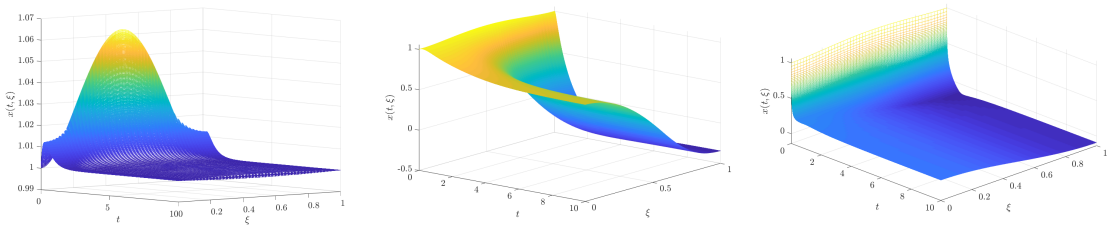


Figure 2.4: Right: uncontrolled state, converges to $x = 1$. Middle: controlled state with linearized feedback $\mathbf{u}_0(\mathbf{x}) = -\mathbf{K}_0\mathbf{x}$ around the origin, fails to stabilize. Left: the nonlinear feedback $\mathbf{u}_V$ stabilizes the system towards the origin.

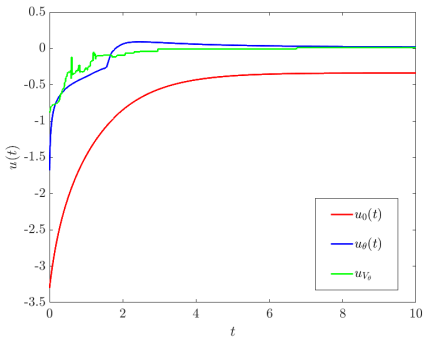


Figure 2.5: Different control signals: $\mathbf{u}_0$, $\mathbf{u}_\theta$ and $\mathbf{u}_V$. In this test the feedback law trained with gradient-augmented data $\mathbf{u}_V$ outperforms $\mathbf{u}_\theta$.

Concluding remarks This numerical example presented the computational method from [8] for approximating stabilizing feedback laws in nonlinear dynamics using a supervised learning approach. The training data is derived from the pointwise solution of the SDRE. We examined the approximation of feedback control via FNNs, exploring various architectural choices and loss functions for training. Our computational results demonstrate that for high-dimensional nonlinear problems, the SDRE feedback law can be effectively approximated using FNNs, eliminating the need for a fast ARE solver in real-time closed-loop control.

Problems where a linear structure is more prominent, such as in the control of semilinear parabolic PDEs, learning a model for a local approximation of the value function $V(\mathbf{x})$ and computing the control from its gradient is an accurate and more efficient alternative w.r.t. directly learnign the feedback control $\mathbf{u}(\mathbf{x})$.

2.3 Deflation method for polynomial systems and PDEs

We conclude this background chapter by introducing a numerical technique that plays a pivotal role in identifying distinct solutions to non-linear systems of equations. The technique combines Newton's iteration with a deflation routine, which systematically eliminates previously identified solutions to uncover new ones.

Consider the root-finding problem for scalar polynomials as a motivating example. A standard Newton method typically converges to only a single root within the basin of attraction where the initial guess resides. To compute multiple roots, one would traditionally need to initialize different starting points, hoping they fall into distinct basins of attraction. Instead, the deflation technique transforms the landscape of the root-finding problem by effectively "hiding" previously identified solutions, reshaping the basins of attraction. As illustrated in Figure 2.6, this process allows a single initial condition to be used iteratively to recover all solutions.

If an iterative algorithm is used to determine the zeros $a_1, \dots, a_n$ of a polynomial $\mathrm{q}(a)$, distinct roots can be identified by applying the same iterative algorithm to the modified function:

$$\mathrm{d}(a) = \frac{\mathrm{q}(a)}{\prod_{i=1}^{n} (a - a_i)} \,. \tag{2.55}$$

Deflation was introduced in the study of polynomials in [250] and later extended in [60] to differentiable finite-dimensional maps $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}^n$. The generalization of this deflation technique to infinite-dimensional Banach spaces was proposed in [111]. This technique has gained popularity due to its effectiveness and is widely used in various fields, often in combination with continuation and barrier methods, particularly in topology optimization and other engineering applications. and has been successfully applied in various contexts [73, 164, 199]. Below, we briefly outline the key elements of the deflation algorithm.

The procedure aims at systematically changing the PDE (or polynomial system) residual $\mathcal{F}$ so that known solutions are progressively ruled out. Namely, we are going to construct a new function $\mathcal{D}(\cdot)$ having all the zeros coinciding with the roots of $\mathcal{F}(\cdot)$, except for those

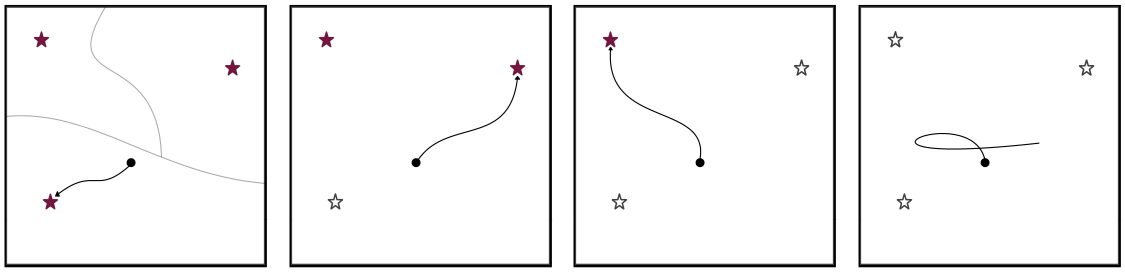


Figure 2.6: Illustration of the deflation technique applied to a root-finding problem. The first panel shows a standard Newton method converging to a root (pink star) from an initial guess (black circle) within the root's basin of attraction. Subsequent panels demonstrate how deflation progressively alters the basins of attraction, allowing the same initial condition to reach different solutions iteratively, until the root-finding method ultimately diverges for the given initial guess.

we have already found. As a clarifying example, let us assume we have already identified a solution $\mathbf{r}$ of $\mathcal{F}(\mathbf{a}) = 0$, then

$$\mathcal{D}(\mathbf{a}) = \mathcal{M}(\mathbf{a}, \mathbf{r})\mathcal{F}(\mathbf{a}) \tag{2.56}$$

where $\mathcal{M}(\mathbf{a}, \mathbf{r})$ is a deflation operator, i.e. $\mathcal{M}$ is invertible, and $\mathcal{M}(\mathbf{a}, \mathbf{r})\mathcal{F}(\mathbf{a}) = 0$ if and only if $\mathcal{F}(\mathbf{a}) = 0$.

Different classes of alternative choices of deflation operator $\mathcal{M}$ are proposed in [111], among which we consider the *shifted exponential-norm deflation operator* that specifies

$$\mathcal{D}(\mathbf{a}) = \left(\frac{\mathbb{I}}{\eta(\mathbf{a})} + \mathbb{I}\xi\right)\mathcal{F}(\mathbf{a})\,, \qquad \eta(\mathbf{a}) = \|\mathbf{a} - \mathbf{r}\|^p\,, \tag{2.57}$$

where $\mathbb{I}$ denotes the identity operator, p is the deflation power and $\xi \geq 0$ is the shift parameter. The deflating term $\eta(\mathbf{a})^{-1}$ ensures that $\mathcal{D}(\mathbf{r}) \neq 0$, while the shift operator $\mathbb{I}\xi$ forces the norm of the deflated residual to not vanish artificially as $\|\mathbf{a} - \mathbf{r}\| \to \infty$.

2.3.1 Computing the derivative of the deflated operator

In the following we assume that roots are found using Newton's method. Provided that, starting from an initial condition $\mathbf{a}_0$, the first root $\mathbf{r}_0$ of $\mathcal{F}$ has already been found, the

problem reduces to $\mathcal{D}(\cdot)$=0, for which the Newton's iteration reads

$$\mathbf{a}^{n+1} = \mathbf{a}^n - \frac{\mathcal{D}(\mathbf{a}^n)}{\mathcal{D}'(\mathbf{a}^n)}\,. \tag{2.58}$$

This motivates a brief discussion of how to conveniently compute the derivative $\mathcal{D}'(\cdot)$. We focus on the case

$$\mathcal{D}(\mathbf{a}) = \left(\frac{\mathbb{I}}{\eta(\mathbf{a})}\right)\mathcal{F}(\mathbf{a})\,, \qquad \eta(\mathbf{a}) = \prod_{i=0}^{n} \|\mathbf{a} - \mathbf{r}_i\|^p\,, \tag{2.59}$$

as the generalization to (2.57) is straightforward. We write the deflation routine in a recursive way w.r.t. the number of roots.

- At the very first step, with only one known solution $\mathbf{r}_0$ to $\mathcal{F}(\mathbf{a}) = 0$, we have

$$\mathcal{D}(\mathbf{a}) = \left(\frac{\mathbb{I}}{\eta_0(\mathbf{a})}\right)\mathcal{F}(\mathbf{a})\,, \qquad \eta_0(\mathbf{a}) := \|\mathbf{a} - \mathbf{r}_0\|^p\,, \tag{2.60}$$

  with derivative

$$\mathcal{D}'(\mathbf{a}) = \frac{\mathcal{F}'(\mathbf{a})}{\eta_0(\mathbf{a})} - \frac{\mathcal{F}(\mathbf{a})}{(\eta_0(\mathbf{a}))^2} \otimes \eta_0'(\mathbf{a})\,. \tag{2.61}$$

  where $\otimes$ denotes the outer product. Assume that, by applying Newton's method for $\mathcal{D}(\mathbf{a}) = 0$ from the same initial guess $\mathbf{a}_0$, we find the second root $\mathbf{r}_1$.

- At the $n-$th iteration of the deflation loop, assuming to know roots $\mathbf{r}_0, \dots, \mathbf{r}_{n-1}$, the adjusted residual operator $\mathcal{D}$ reads

$$\mathcal{D}(\mathbf{a}) = \left(\frac{\mathbb{I}}{\eta_n(\mathbf{a})}\right)\mathcal{F}(\mathbf{a})\,, \qquad \eta_n(\mathbf{a}) = \eta_{n-1}(\mathbf{a})\,\tau(\mathbf{a})\,, \tag{2.62}$$

  for the temporal variable $\tau(\mathbf{a}) = \|\mathbf{a} - \mathbf{r}_{n-1}\|^p$ and, dropping all the dependencies w.r.t. $\mathbf{a}$, we have the derivative

$$\mathcal{D}' = \frac{\mathcal{F}'}{\eta_n} - \frac{\mathcal{F}}{\eta_n^2} \otimes \left(\eta_{n-1}'\,\tau + \eta_{n-1}\,\tau'\right). \tag{2.63}$$

  Newton's method is then used to find the next root $\mathbf{r}_n$ of $\mathcal{D}(\mathbf{a}) = 0$.

The deflation loop stops when the root finding technique diverges for a given deflated operator.

A pseudocode for this routine is provided in Algorithm 4. A crucial consideration for the deflation algorithm is the selection of suitable stopping criteria, as once all roots have been found, the algorithm will naturally fail to converge. However, numerical instability can also prevent convergence as the number of roots in the deflation operator increases.

Algorithm 4: Deflated Newton

Data: Function $\mathcal{F}$ and its derivative $\mathcal{F}'$, initial `guess`, deflation parameters p, ξ, convergence tolerance $\varepsilon \ll 1$

`max_iteration` $= 10^3$, `solutions` $\leftarrow \{\}$, `exit_flag` $=$ `false`; `initializing`

$\eta_0(*) \leftarrow 1$, $\eta_0'(*) \leftarrow 0$, $\tau(*) \leftarrow 1$, $\tau'(*) \leftarrow 0$; `deflation operators`

$m \leftarrow 0$; `count of identified roots`

while not *exit_flag* **do**

  $\boldsymbol{t} \leftarrow$ `guess` ; `initial guess`

  `update` $\gg 1$, `iteration` $\leftarrow 0$; `reset convergence flags`

  while *update* $> \varepsilon$ *and* *iteration* $<$ *max_iteration* **do**

    `iteration` $+= 1$;

    $f \leftarrow \mathcal{F}(\boldsymbol{t})$, $df \leftarrow \mathcal{F}'(\boldsymbol{t})$;

    $\mathcal{D} \leftarrow$ (2.57), $\mathcal{D}' \leftarrow$ (2.63); `compute deflated system`

    $\boldsymbol{t}_{\text{new}} \leftarrow \boldsymbol{t} -$ `pinv`$(\mathcal{D}') \cdot \mathcal{D}$; `update root candidate`

    `update` $\leftarrow \|\boldsymbol{t}_{\text{new}} - \boldsymbol{t}\|$;

    $\boldsymbol{t} \leftarrow \boldsymbol{t}_{\text{new}}$;

  if *iteration* $=$ *max_iteration* **then**

    `exit_flag` $=$ `true`;

  else

    `solutions` $\leftarrow$ `solutions` $\cup \{\boldsymbol{t}\}$; `store solution`

    $m += 1$;

    $\tau(*) \leftarrow \| * - \boldsymbol{t}\|^p$; `Update deflation operators`

    $\tau'(*) \leftarrow p \cdot \| * - \boldsymbol{t}\|^{p-2} \cdot (* - \boldsymbol{t})$;

    $\eta_m(*) = \eta_{m-1}(*)\tau(*)$;

Result: list `solutions`, containing the roots of $\mathcal{F}$

2.3.2 Numerical considerations and limitations of deflation methods

While deflation techniques have proven effective for computing multiple solutions to nonlinear problems, they are not without numerical limitations.

The core idea of deflation is to systematically modify the nonlinear residual to introduce singularities at previously discovered solutions, hence preventing the solver from converg-

ing to them again. However, this modification leads to a cumulative degradation in the conditioning of the resulting nonlinear system. As additional solutions are deflated, this can adversely affect the convergence behavior of Newton-type solvers.

In practical applications, these issues are most apparent after a large number of solutions have been deflated. For instance, in problems with translation-invariant steady states or symmetry-induced multiplicities, even a well-tuned Newton solver may diverge after several deflations. This phenomenon has been documented in the literature (e.g., [111]) and was also observed in the experiments conducted in this thesis.

Deflation has been applied successfully to high-dimensional problems, including PDEs discretized on fine spatial grids. However, its performance in such settings is sensitive to the size of the problem and the accumulation of deflation terms. In high dimensions, numerical instabilities, increased memory usage, and deteriorating convergence rates all present challenges that limit the scalability of the method.

Furthermore, deflation methods assume that the Newton solver remains effective near the solution set. This assumption breaks down in the presence of degenerate critical points—that is, locations where the gradient vanishes but the Hessian is singular. At such points, standard Newton updates become undefined or unreliable, and deflation can exacerbate the problem by introducing additional singular behavior in the residual.

Overall, while deflation offers a powerful framework for exploring solution multiplicity, its numerical behavior must be carefully monitored, particularly in high-dimensional or degenerate regimes.

2.3.3 Example: Time optimal feedback control of nilpotent linear systems using neural networks

In this section, we present an example illustrating the use of the deflation algorithm to find multiple roots of polynomial systems linked to the solution of a specific type of OCPs. More specifically, summarizing the work in [49], we study the synthesis of time-optimal control laws, i.e. external inputs that steer a dynamical system to a target in the least

amount of time.

Let A be an $n \times n$ real matrix, $B = (b_1, \ldots, b_n)^\top \in \mathbb{R}^n$ an n-dimensional real vector, and $\mathbf{y} \in \mathbb{R}^n$ a given point. We aim at finding solutions of the form $(T, \Omega, u, \mathbf{s})$ to the linear system of differential equations

$$\begin{cases} \dot{\mathbf{s}}(t) = A\mathbf{s}(t) + Bu(t) \\ \mathbf{s}(0) = \mathbf{y} \\ \mathbf{s}(T) = \mathbf{0}, \end{cases} \tag{2.64}$$

where T is called the *total time* of the system 2.64, and is it given by $T = t_1 + \cdots + t_n$ for non-negative $t_i \in \mathbb{R}$, which we call *switching times*. The domain Ω is the union of subintervals $\Omega_i = [T_{i-1}, T_i)$, with $T_0 = t_0 = 0$, and $T_i = \sum_{j=1}^{i} t_j$, for every $1 \leq i \leq n-1$, and $\Omega_n = [T_{n-1}, T_n]$.

According to Pontryagin's maximum principle [212, Theorem 2], time-optimal control signals for this problem can be parametrized as sequences that switch among a finite set of values. The function $u(t)$ is hence a piecewise-constant function on Ω called the *control function* of the system (2.64) taking values between 1 and -1 on each Ω_i. In particular, the theorem shows that the optimal control is constant on each subinterval Ω_i, taking values of only -1 or 1. Such a control is known as *bang-bang* for its discontinuous switching of signs.

The *trajectory* function $\mathbf{s}\colon \Omega \to \mathbb{R}^n$ is defined as the unique solution to the differential equation (2.64). We say that such a solution $(T, \Omega, u, \mathbf{s})$ is *optimal* if for any other solution $(\tilde{T}, \tilde{\Omega}, \tilde{u}, \tilde{\mathbf{s}})$, we have that the total time $T \leq \tilde{T}$.

For this class of linear dynamical systems, this parametrization allows us to reformulate the feasibility of an optimal switching sequence as the existence of non-negative solutions to a polynomial system of equations. To determine feasible switching sequences, we compute all roots of the associated polynomial system and identify the real, non-negative solutions. Since multiple roots naturally arise in this setting, the deflation technique introduced earlier offers a systematic approach for recovering all distinct solutions. By applying

deflation iteratively, we obtain the complete set of roots and extract the real, non-negative ones, which correspond to valid optimal switching sequences.

While this approach provides open-loop control solutions, real-world applications often require robustness to uncertainties and disturbances. A more resilient alternative is to construct state-to-control mappings, allowing for real-time adaptation through feedback control.

Nilpotent Linear Time-invariant Systems

In this section, A is assumed to be a nilpotent matrix with nilpotency index n. Since in this case $A = (A_{ij})_{i,j}$ is similar to an $n \times n$ nilpotent Jordan block, without loss of generality, we assume that $A_{ij} = \delta_{i,j-1}$, for $i, j = 1, \dots, n$, where δ is the Kronecker delta.

Proposition: Let $\boldsymbol{b}, \boldsymbol{x} \in \mathbb{R}^n$ be fixed. If $(T, \Omega, u, \mathbf{s})$ is a solution to (2.64), where

$$\mathbf{s}(t) = e^{tA}\mathbf{s}(0) + e^{tA} \int_0^t e^{-\tau A} B\, u(\tau)\, d\tau, \tag{2.65}$$

for $t \in \Omega$ and an $n \times n$ nilpotent Jordan block A, then

$$0 = -u(0) \sum_{k=i}^{n} b_k \sum_{\substack{\alpha_1, \dots, \alpha_n \geq 0 \\ \alpha_1 + \cdots + \alpha_n = k}} (-1)^{\max\{i:\, \alpha_i \neq 0\}} \frac{t_1^{\alpha_1} \cdots t_n^{\alpha_n}}{\alpha_1! \cdots \alpha_n!} + \sum_{j=0}^{i-1} x_{n-j} \frac{T^{i-j-1}}{(i-j-1)!}, \tag{2.66}$$

for every $i = 1, \dots, n$, where $(x_1, \dots, x_n)^\top = \mathbf{y}$.

Example: In $\mathbb{R}^3$, a solution (2.64) leads to a the system of equations (2.66), which in the case $n = 3$ is of the form

$$\begin{aligned}
0 &= u(0)(t_1 - t_2 + t_3) + x_3, \\
0 &= u(0)\left(\frac{t_1^2}{2} + t_1 t_2 + t_1 t_3 - \frac{t_2^2}{2} - t_2 t_3 + \frac{t_3^2}{2}\right) + x_2 + (t_1 + t_2 + t_3) x_3, \\
0 &= u(0)\left(\frac{t_3^3}{6} + \frac{t_1^2 t_2}{2} + \frac{t_1^2 t_3}{2} + \frac{t_1 t_2^2}{2} + t_1 t_2 t_3 + \frac{t_1 t_3^2}{2} - \frac{t_2^3}{6} - \frac{t_2^2 t_3}{2} - \frac{t_2 t_3^2}{2} + \frac{t_3^3}{6}\right) \\
&\quad + x_1 + (t_1 + t_2 + t_3) x_2 + \frac{(t_1 + t_2 + t_3)^2}{2} x_3,
\end{aligned}$$

for $u(0) = \pm 1$. Note that from a solution of (2.66), with $u(0) = \pm 1$ and non-negative switching times t_i, we can recover a control function and thus a solution of the form (2.65) to the system of differential equations (2.64). In the following, we focus on finding solutions to (2.66) subject to these two conditions. By results found in [212, Theorem 11, 12], if there exists a solution of (2.66) with $u(0) = \pm 1$ and non-negative switching times t_i, then the recovered control function $u(t)$ is the unique optimal control for the system (2.64).

Deflated Newton's method

To determine all possible solutions of the polynomial system (2.66), we rely on the deflation routine discussed in Section 2.3. In the context of this project, $\mathcal{F}(\boldsymbol{t}) = (f_1, \ldots, f_n)^\top$ will be the polynomial system from (2.66), with $\mathbf{t} := (t_1, \ldots, t_n)$ vector of switching times. Moreover, we consider the deflated system of equation as in (2.57), with Jacobian

$$\mathcal{F}' = \begin{pmatrix} \partial_{t_1} f_1 & \partial_{t_2} f_1 & \cdots & \partial_{t_n} f_1 \\ \partial_{t_1} f_2 & \partial_{t_2} f_2 & \cdots & \partial_{t_n} f_2 \\ \vdots & \vdots & \ddots & \vdots \\ \partial_{t_1} f_n & \partial_{t_2} f_n & \cdots & \partial_{t_n} f_n \end{pmatrix}, \tag{2.67}$$

where the partial derivative of some f_k with respect to the switching time t_q is the polynomial

$$\partial_{t_q} f_k = -u_0 \sum_{\substack{\alpha_1, \ldots, \alpha_n \geq 0 \\ \alpha_1 + \cdots + \alpha_n = k-1}} (-1)^{\min\{\min(1 \leq i \leq n | \alpha_i \neq 0), q\}} \frac{t_1^{\alpha_1} \cdots t_n^{\alpha_n}}{\alpha_1! \cdots \alpha_n!} + \sum_{l=0}^{k-2} x_{n-l+1} \frac{T^l}{l!}. \tag{2.68}$$

This explicit form of the Jacobian of $\mathcal{F}$ is instrumental to computing the derivative of the deflated system, as discussed in 2.3.1.

We terminate the deflation loop either when the root-finding method fails to converge or when the theoretical maximum number of solutions is reached. In the reference paper [49], we discuss the Hermite quadratic form and show how its signature can be used to determine the existence of non-negative real roots of a system of polynomial equations.

This can be used as a methods to bound the number of feasible solutions of 2.3.1, which are exact in most cases, and are better than then naive factorial bound $n!$.

Feedforward Neural Network Classifier

The numerical procedure developed so far generates open-loop controls, which are effective under ideal conditions where the system dynamics and initial conditions are precisely known. However, in real-world applications, uncertainties, disturbances, and modeling inaccuracies can lead to deviations from the desired trajectory. This issues are well understood in the control literature, particularly in the context of stabilizing control, and are typically mitigated by adopting a closed-loop control approach, which is inherently more robust. Closed-loop controls are state-to-control mappings that determine the optimal action based on the system's current state. Unlike open-loop control signals, which prescribe an optimal trajectory based solely on the initial condition, feedback controls are computed dynamically as the system evolves. This adaptability allows them to correct deviations from the optimal trajectory, enhancing robustness to uncertainties and disturbances.

Open-loop solvers, like the one here proposed, can be utilized to synthesize feedback controls in a MPC fashion. However, as discussed in Section 2.1.2, these approaches typically come with a high computational cost due to the need for frequent recomputation of the optimal trajectories. Similarly to what we discussed in Section 2.2, we alleviate this issue by considering a Neural Network approximation of the state-to-control mapping.

Given the bang-bang (binary) nature of the control in the problem under consideration, we frame the feedback approximation as a **classification** task rather than relying on regression. This means that the output of our model will be of discrete nature in $\{-1, 1\}$. In particular, we focus on FNN-type of architectures with a sigmoidal final layer

$$l_M(\boldsymbol{z}_M) = \frac{1}{1 + e^{-\boldsymbol{z}_M}}, \tag{2.69}$$

where, as before in (2.44), the function is applied component-wise to the entries of the layer input $\boldsymbol{z}_M$. This function maps any real-valued input into the range $(0, 1)$, making it

particularly suitable for binary classification tasks, as the output can be interpreted as a probability:

$$\mathbb{P}\big[u(\boldsymbol{z}) = 1\big] =: u_\theta(\boldsymbol{z}), \qquad \mathbb{P}\big[u(\boldsymbol{z}) = -1\big] =: 1 - u_\theta(\boldsymbol{z}). \tag{2.70}$$

By denoting the approximation of the feedback control as $\tilde{u}(\boldsymbol{z}) \approx u(\boldsymbol{z})$, we define it as the most likely event under the probability u_θ:

$$\tilde{u}(\boldsymbol{z}) = \begin{cases} 1, & \text{if } u_\theta(\boldsymbol{z}) \geq 0.5, \\ -1, & \text{otherwise.} \end{cases} \tag{2.71}$$

This formulation not only provides a classification decision but also quantifies the confidence of the model in the forecast.

The network is trained to minimize a cross-entropy loss function on data, ensuring the output probabilities align with the provided labels. The number of layers and neurons is selected to balance model complexity and computational efficiency, and hyperparameters such as learning rate and batch size are tuned using a grid search and aiming at maximizing the approximation performance. These choices are problem-specific and will be clarified for each numerical test.

In this case, the binary cross-entropy loss function measures how well the neural network's predicted probabilities align with the true binary labels ($u(\boldsymbol{z}) \in \{-1, 1\}$). The binary cross-entropy loss for a single data point $\boldsymbol{z}$ is given by:

$$l^{NN}(\boldsymbol{z}, u(\boldsymbol{z})) = -\big[\zeta \log(u_\theta(\boldsymbol{z})) + (1 - \zeta) \log(1 - u_\theta(\boldsymbol{z}))\big],$$

where:

- $\zeta \in \{0, 1\}$ is the true label, with $\zeta = 0$ corresponding to $u(\boldsymbol{z}) = -1$ and $\zeta = 1$ corresponding to $u(\boldsymbol{z}) = 1$,
- $u_\theta(\boldsymbol{z})$ is the network's predicted likelihood that $u(\boldsymbol{z}) = 1$, as in (2.70).

For a dataset of N samples, the total binary cross-entropy loss is:

$$\mathcal{L}^{NN} = \frac{1}{N}\sum_{j=1}^{N}\left[-\zeta_j \log(u_\theta(\boldsymbol{z}^{(j)})) - (1-\zeta_j)\log(1-u_\theta(\boldsymbol{z}^{(j)}))\right]. \tag{2.72}$$

This loss function encourages the network to predict probabilities $u_\theta(\boldsymbol{z}^{(j)})$ close to 1 when $\zeta_j = 1$ and close to 0 when $\zeta_j = 0$, effectively learning the mapping from states $\boldsymbol{z}^{(j)}$ to control labels $u^{(j)}$.

The model's performance is assessed through the accuracy, measuring the proportion of correctly predicted labels out of the total number of samples

$$\text{Accuracy} = \frac{1}{N}\sum_{j=1}^{N}\mathbf{1}_{(\tilde{u}(\boldsymbol{z}^{(j)})=u^{(j)})},$$

where $\mathbf{1}$ denotes the indicator function.

Numerical Tests

In this section, we evaluate the proposed methodology on an nth order integrator system, where n ranges from 2 to 5. The discussion includes the process of data generation, the architecture of the model, and the chosen hyperparameters.

For the generation of training and testing datasets, we utilize the deflation routine and solve the polynomial system in (2.66). We begin by generating N_s sample points in the state space, denoted as $\{\mathbf{y}^{(i)}\}_{i=1}^{N_s}$. For each initial condition, we evaluate the solvability of the polynomial system under the two possible initial control values, $u_0 = 1$ and $u_0 = -1$.

To ensure the accurate detection of switching surfaces, we further store the state and control information for a sequence of 100 points along the resulting optimal trajectory generating from each initial condition. This process results in a dataset consisting of $N = 100N_s$ entries:

$$\mathcal{T} = \{\mathbf{y}^{(i)}, u_0^{(i)}\}_i \cup \{\mathbf{s}^{(i)}(t_k), u^{(i)}(t_k)\}_{i,k} \qquad \text{for } i = 1,\ldots,N_s,\ k = 1,\ldots,100, \tag{2.73}$$

where $\mathbf{s}^{(i)}(t)$ is the optimal trajectory departing from $\mathbf{y}^{(i)}$ and t_k's are uniformly spaced discrete times in $(0, T]$. For convenience, we will denote the input $u_\theta(\cdot)$ as $\boldsymbol{z}$, and rearrange the training set as

$$\mathcal{T} = \{\boldsymbol{z}^{(j)}, u^{(j)}\}_{j=1}^{N}, \tag{2.74}$$

where $N = 100N_s$.

The training is performed using the Adam optimizer, which efficiently handles the stochastic gradient descent steps with adaptive learning rates. The dataset is split into two subsets: 90% for training, 10% for testing.

Across the numerical examples, we demonstrate how leveraging the confidence interpretation of u_θ can improve approximation performance. We evaluate the robustness of the feedback approximation under noisy trajectories and highlight the computational advantages of using Hermite quadratic forms to bound the number of roots identified through deflation.

2nd order We consider the time-optimal control problem

$$\min_{u(t)} T \qquad \text{s.t.} \qquad \begin{cases} \dot{s}_1 = y_2\,, & s_1(0) = y_1\,, & s_1(T) = 0\,, \\ \dot{s}_2 = u\,, & s_2(0) = y_2\,, & s_2(T) = 0\,, \end{cases} \tag{2.75}$$

whose solution is a piecewise constant function, starting from an initial control $u_0 \in \{-1, 1\}$ and switching at times $\boldsymbol{t} := (t_1, t_2)^\top$, computed by solving the associated polynomial system (2.66), which for $n = 2$ reads:

$$\begin{cases} t_1 - t_2 + y_2 = 0 \\ \dfrac{t_1^2}{2} + t_1 t_2 - \dfrac{t_2^2}{2} + y_1 + y_2(t_1 + t_2) = 0 \end{cases} \quad \text{for } (u_0 = 1), \qquad \begin{cases} t_1 - t_2 - y_2 = 0 \\ \dfrac{t_1^2}{2} + t_1 t_2 - \dfrac{t_2^2}{2} - y_1 - y_2(t_1 + t_2) = 0 \end{cases} \quad \text{for } (u_0 = -1). \tag{2.76}$$

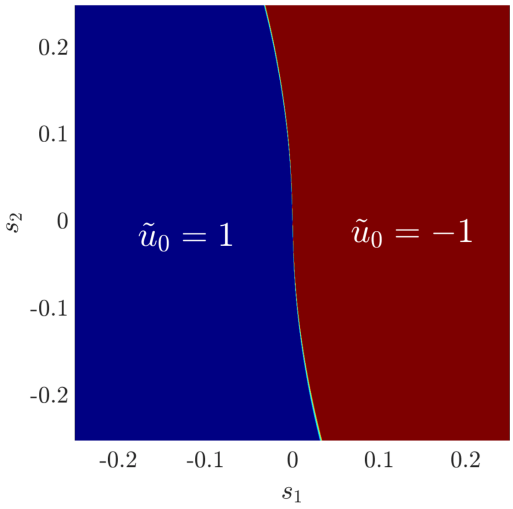


Figure 2.7: The trained classifier identifies the switching surface as a low-confidence region.

Admissible solutions of (2.76) specify the optimal trajectory, starting from $\mathbf{y} := (x_1, x_2)$ with control u_0 and switching times $\boldsymbol{t}$. We generate a dataset by sampling $N_s = 50$ normally distributed initial conditions $\{\mathbf{y}^{(i)}\}_{i=1}^{N_s}$ in the numerical domain $[-1, 1]^2$ and coupling, along the optimal trajectories, states $\mathbf{s}^{(i)}(t_k)$ with the associated control $u^{(i)}(t_k) \in \{-1, 1\}$, for $0 = t_0 < \ldots t_k < \ldots t_{100} = T$.

We rely on the $N = 100 \cdot N_s$ generated data pairs in $\mathcal{T}$ to train a NN classifier mapping states $\boldsymbol{z}$ into the associated feedback $u_\theta(\boldsymbol{z})$. After a grid search over a number of possible feedforward architectures, we select $u_\theta(\cdot)$ to have a single hidden layer of width $n_2 = 100$ neurons and $\sigma_2 = tanh(\cdot)$ as activation function. We train the model to minimize the loss function (2.72) over the training set via Adam, a first-order gradient-based optimization method [151].

After training, the model achieves a test set accuracy of 99.38% with a corresponding loss $\mathcal{L} = 0.0156$. In Figure 2.7, we show how the trained classifier labels points in $[-0.25, 0.25]^2$. The model assigns controls with high probability to points on either side of the switching surface, while points lying directly on the surface are associated with lower confidence, as indicated by the lighter colors in the figure.

All optimal trajectories eventually intersect the switching hyperplane and travel along

it until reaching the origin. Therefore, the model's accuracy on the switching surface is crucial for achieving a reliable feedback approximation. At this point, we have two options:

- we rely on the classification provided by the neural network u_θ, disregarding its low confidence,
- or we define a confidence threshold, and if the confidence falls below this threshold, we invoke the open-loop solver to determine the feedback control at the low-confidence point.

We compare these two procedures in Figure 2.8. The trajectory controlled through the NN feedback approximation, is integrated via explicit Euler, with time-step 5^{-4}. The enhanced approximation, relying on the polynomial system solution when the classifier confidence is below $\varepsilon = 0.01$, is triggered on 1% of the trajectory points and doesn't improve significantly the approximated optimal horizon T_{NN}. We conclude that, in this first example, even low-confidence points are classified correctly via u_θ.

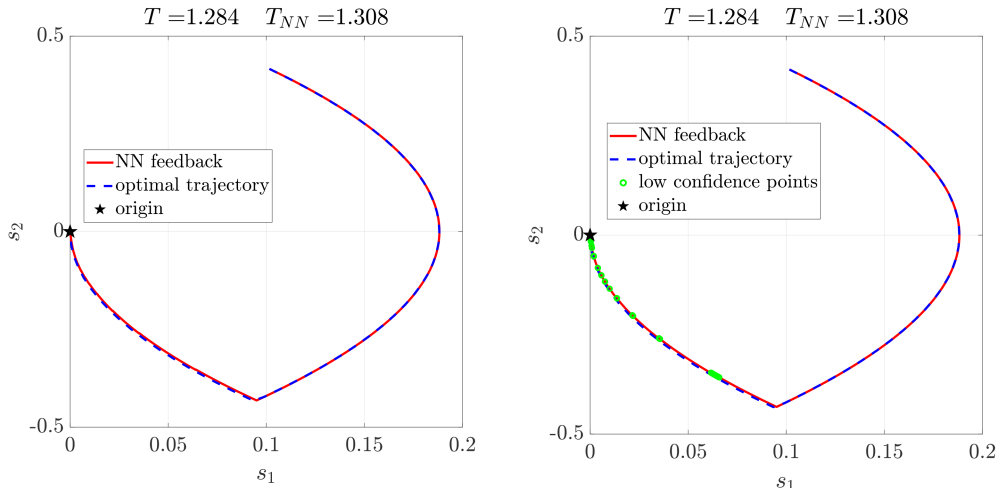


Figure 2.8: Comparison of the optimal trajectory computed via deflation with the one controlled with the classifier u_θ. On the left, the NN feedback is solely determined via u_θ, whilst on the right we rely on the identification of the solution of the polynomial system (2.76) to identify the feedback at low-confidence points.

3rd order Similar to what we have discussed in the previous section, we consider the

time-optimal control problem subject to

$$\begin{cases} \dot{s}_1 = s_2 \\ \dot{s}_2 = s_3 \\ \dot{s}_3 = u \end{cases} \tag{2.77}$$

with state variable $\mathbf{s} := (s_1, s_2, s_3)$ and boundary conditions $\mathbf{s}(0) = \mathbf{y} := (y_1, y_2, y_3)$, $\mathbf{s}(T) = \mathbf{0}$. The polynomial system (2.66) features 3 equations for the unknown switching times $(t_1, t_2, t_3) =: \boldsymbol{t}$. Relying on the deflation routine, we sample state and controls along the optimal trajectories departing from $N_s = 5000$ normally distributed initial conditions in $[-1, 1]^3$. The resulting dataset is used to train a model u_θ characterized by a single hidden layer with width $n_2 = 80$ and $\sigma_2 = tanh(\cdot)$ activation function. After training, the model reaches 99.12% accuracy in the test set, associated with a loss $\mathcal{L} = 0.0334$. In this example, the enhanced feedback approximation, calling the polynomial solver for identifying the control at points with confidence below $\varepsilon = 0.005$, occurs on 4.08% of the trajectory points, and improves the approximated horizon when compared to the trajectory controlled solely via u_θ. We show the results in Figure 2.9, where the NN-controlled trajectory is computed via a forward Euler scheme with stepsize 10^{-3}, whilst the optimal trajectory is computed through exact integration.

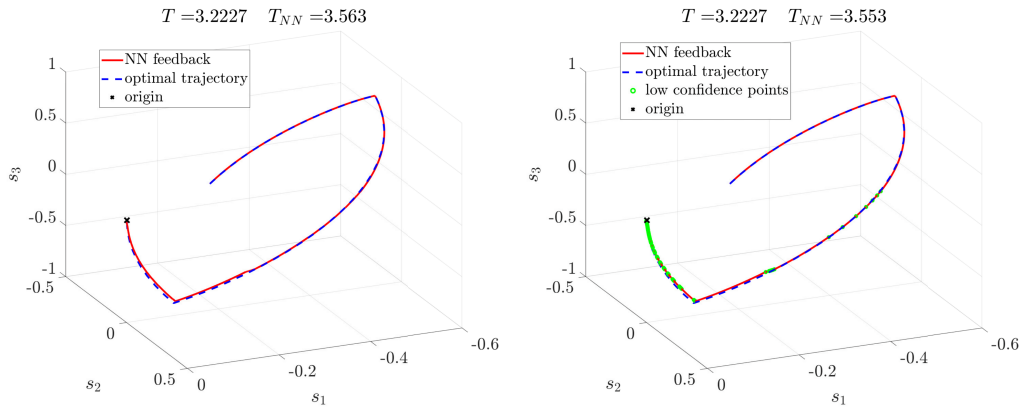


Figure 2.9: Comparison of the optimal trajectory with the u_θ-controlled (left) and with the confidence-enhanced approximation (right). The reliance on the polynomial solver for low-confidence points improves the approximation, as the time T_{NN} needed to reach the origin decreases.

We motivated the introduction of the feedback NN approximation for achieving robustness against noise. To evaluate this, we introduce white noise into the last component of the system. Specifically, at every discrete integration time t we add $(0, 0, \eta(t))$, where $\eta(t) \sim \mathcal{N}(0, \sigma^2)$, with $\sigma^2 = 0.02$. Starting from the same initial condition, we perform a Monte Carlo simulation for both the open-loop and the approximated feedback control approaches. Figure 2.10 illustrates the mean and variance obtained from 1000 noisy controlled trajectories. As expected, while the open-loop solution diverges on average from the target final state, the approximated feedback law successfully drives the system towards the origin.

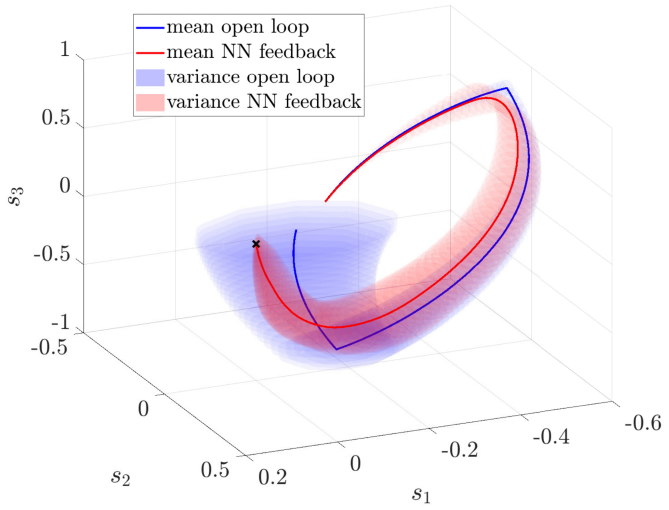


Figure 2.10: Mean and variance of a Monte Carlo simulation with 1000 trajectories perturbed by Gaussian noise. The robustness of the approximated feedback law ensures that the controlled system reaches the target destination, while the open-loop solution, obtain as solution of the polynomial system (2.66), diverges from it.

4th order We consider the time-optimal control problem subject to the fourth order integrator for state variable $\mathbf{s} = (s_1, \ldots, s_4)$ and boundary conditions $\mathbf{s}(0) = \mathbf{y} := (y_1, \ldots, y_4)$, $\mathbf{s}(T) = \mathbf{0}$. In this higher-dimensional example, we examine the upper bound on the number of solutions of the associated system (2.66) for the switching times $t_1, \ldots, t_4$. The theoretical bound on the total number of real and complex solutions is $4! = 24$. However, as discussed in [49], the use of Hermite quadratic forms allows us to refine this bound by identifying the number of real solutions more precisely.

We sample initial conditions $\mathbf{y}^{(i)} \in ([-1,1] \cap \mathbb{Q})^4$, for $i = 1, \dots, N_s = 10{,}000$, and for each $\mathbf{y}^{(i)}$ we compute the number of real roots of the polynomial system associated with $u_0 = 1$ and $u_0 = -1$. Note the transition from $\mathbb{R}$ to $\mathbb{Q}$, which we do as computation of the Hermite quadratic form over the reals is computationally unreliable, if not impossible. This preliminary step reduces the computational cost of data generation, as the bound for the deflation algorithm is tightened relative to the theoretical one. Additionally, some $\mathbf{y}^{(i)}$'s will exhibit 0 real solutions for one system or the other, directly determining the sign of the initial control u_0. These new bounds reduce the CPU time for data generation from 108 to 49 minutes when compared to the theoretical bound.

We train a model u_θ characterized by 2 hidden layers, with width $n_2 = n_3 = 100$ and $\sigma_2 = \sigma_3 = tanh(\cdot)$ activation functions. After training, we evaluate the model in the test set, obtaining accuracy of 96.76% and loss $\mathcal{L} = 0.0882$. In Figure 2.11 we analyze the controlled trajectories comparing the optimal solution with the approximated one along the dimensions y_2, y_3, y_4. We consider a low-confidence threshold $\varepsilon = 0.005$, under which we rely on the deflation routine for determining the control. This interests 1.3% of the trajectory. Even more than in the lower order examples, this enhanced approximated feedback improves the performance of the control law. As done for the previous example, we also test the robustness of the feedback approximation by injecting additive noise of the form $(0, 0, 0, \eta(t))$, with $\eta(t) \sim \mathcal{N}(0, \sigma^2)$, where $\sigma = 2^{-2}$. A Monte Carlo simulation of 1000 controlled noisy trajectories shows how the approximated feedback control successfully steers the system towards the origin, while the open loop solution diverges.

5th order As an extreme case for the proposed numerical procedure, we consider the 5th order integrator for state $\mathbf{s} = (s_1, \dots, s_5)$ with boundary conditions $\mathbf{s}(0) = \mathbf{y} := (y_1, \dots, y_5)$ and $\mathbf{s}(T) = \mathbf{0}$. The theoretical upper bound for the number of solutions $(t_1, \dots, t_5)$ of the associated polynomial system is $5! = 120$. However, excessively deflating a polynomial system can lead to instability. Heuristically, as the number of identified and deflated roots increases, the root-finding procedure may diverge, failing to locate additional existing solutions. This pathological behaviour worsen as the upper bound increases.

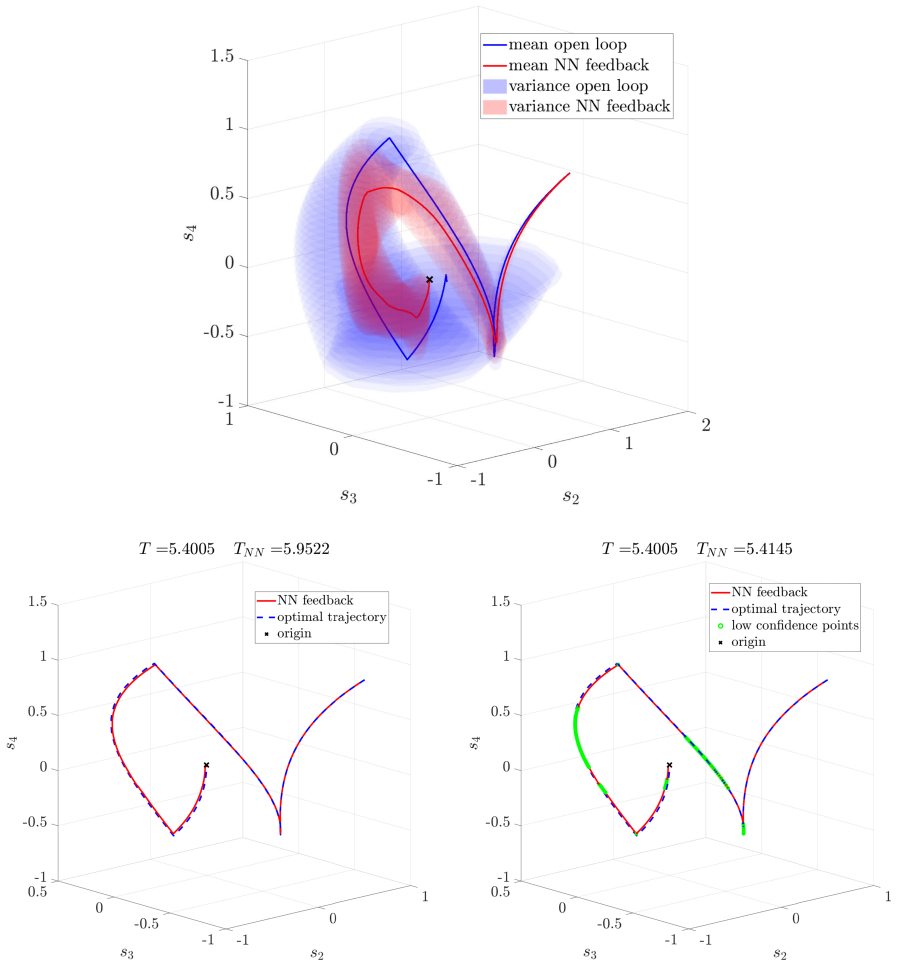


Figure 2.11: Monte Carlo simulation of 1000 controlled noisy trajectories via open loop control signal vs. approximated feedback control (top). Comparison of the optimal trajectory with the one controlled via the approximated feedback law (left). Calling the polynomial solver for the low-confidence points improves the approximation performance in terms of time horizon T_{NN} (right).

We sample $N_s = 50,000$ points $\mathbf{y}^{(i)} \in [-1,1]^5$, $i = 1,\dots,N_s$, and include in the dataset $\mathcal{T}$ only the optimal trajectories corresponding to initial conditions for which the deflation algorithm successfully identifies an admissible solution. This selection process excludes approximately 20% of the initial conditions. We consider a model u_θ with 2 hidden layers having width $n_2 = n_3 = 80$ and $\sigma_2 = \sigma_3 = tanh(\cdot)$ activation functions. The trained classifier achieves accuracy 99.61% in the test set, associated with loss $\mathcal{L} = 0.0141$. In Figure 2.12 we test the performance of the NN feedback controller against the optimal trajectory. The performance of the approximation is significantly improved by selectively

intervening with the polynomial system solver whenever the classifier confidence drops below $\varepsilon = 0.1$. This occurs only at 5 low-confidence points, all located at the intersection of switching surfaces.

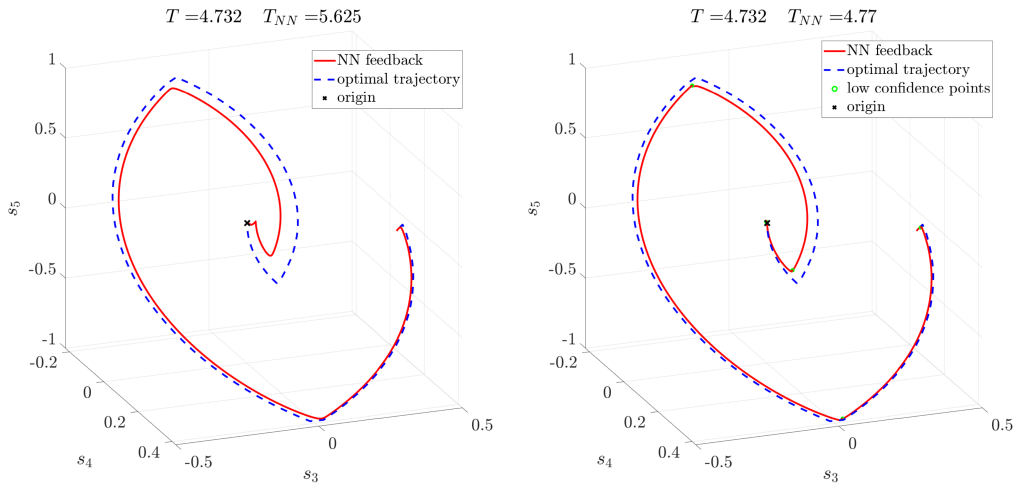


Figure 2.12: Comparison of the optimal trajectory with the trajectory controlled using the approximated feedback law (left). Leveraging the polynomial solver for low-confidence points enhances the approximation's performance in terms of the time horizon T_{NN} (right).

2.4 Concluding remarks

In this chapter, we have provided the methodological background necessary for the developments in the remainder of this thesis. We began by introducing optimal control theory, outlining its fundamental principles and highlighting the challenges associated with solving optimal control problems. In particular, we compared MPC with PMP and SDRE update, highlighting their respective advantages and computational challenges. We then explored neural network approximation techniques as a means to accelerate these solvers by learning efficient state-to-control mappings. As an illustrative example, we reviewed the use of neural networks to accelerate feedback control computations, following the approach in [8].

Next, we presented the deflation technique, a powerful numerical method for systematically identifying multiple solutions to nonlinear systems. This was motivated by its application in time-optimal control problems, where solutions often correspond to multiple switching

sequences that must be computed efficiently. In particular, we examined the deflated Newton's method in the context of solving polynomial systems linked to OCPs, as detailed in [49].

In the following chapters, we will apply these techniques to increasingly complex systems, transitioning from microscopic agent-based models to macroscopic mean-field formulations and mesoscopic kinetic approximations. This multi-scale perspective will allow us to develop and analyze computational methods for high-dimensional control problems, leveraging the theoretical tools introduced here.

Chapter 3

Microscopic Scale

Learning gradient-augmented feedback control synthesis

In this chapter, we address the challenge of improving the computational efficiency and scalability of solving optimal control problems at the agent-to-agent interaction level. By leveraging data-driven control synthesis, we aim at bridging the gap between traditional numerical solvers and the growing demands of real-time, high-dimensional control applications.

In the following, we provide an overview of the methodologies and results presented in [8, 7] in the context of data-driven feedback control strategies for large-scale agent-based models.

Related literature and contributions

The high dimensionality of the physical state space d, combined with the large number of interacting agents N, poses significant challenges for synthesizing optimal feedback controls using conventional direct methods (e.g., [64, 34]). As the number of agents grows, the computational cost of solving these problems escalates exponentially. Addressing this issue requires the development of novel, efficient algorithms capable of overcoming the computational bottlenecks associated with high-dimensional optimization.

A promising framework for addressing these challenges is Model Predictive Control, as detailed in Section 2.1.2. However, a major drawback of MPC is its computational complexity, as it requires repeatedly solving an optimal control problem at each time step. Accelerating MPC computations while maintaining accuracy is crucial for practical applications, particularly those requiring real-time execution.

The research community has proposed various effective solutions to mitigate these issues, involving the simplification of either the predictive models or the optimization problems, to reduce the computational complexity while maintaining acceptable control performance. Among the various proposed methodologies, we cite Reduced-Order MPC [256, 142, 134], linearization techniques [46], lookup tables (e.g., explicit MPC [20, 44]), fast-update schemes [251], and instantaneous control strategies [14], which can be interpreted as a one-step MPC approach. Approximation techniques for the optimization tasks, mostly based on machine learning algorithms, have also been explored. For instance, Learning-based MPC addresses the data-driven approximation of possibly different elements of the MPC formulation, such that the overall performance is improved. For these methods, a fundamental categorisation can be made between offline and online paradigms. In offline learning, the model targets the controller design and is trained using synthetic samples, or data obtained from previous simulations [211, 172]. In online learning instead, the model is trained to approximate the system dynamics, and data are collected while the system evolves, hence the model is actively adjusted during the closed-loop operation. Among other methods in this class, we find Gaussian Processes [153, 152] – providing a non-parametric and probabilistic model of the system's behavior – and Reinforcement Learning [161, 167, 181], particularly useful when the dynamics are complex, unknown, or hard to model accurately using traditional approaches. We refer the reader to [141, 191] for a more comprehensive discussion of the matter.

The main contributions of our work, inspired by the aforementioned results, can be summarized as follows:

- This project is based on supervised learning MPC, where the control policy is learned

offline using training data generated by sampling optimal agent trajectories [102]. Notably, both numerical procedures discussed in Sections 2.1.2 (MPC with PMP update) and 2.1.3 (MPC with SDRE update) adhere to the receding horizon paradigm. Rather than solving high-dimensional optimal control problems from scratch at each time step, we employ neural network approximations to accelerate these numerical procedures.

- A key innovation of this approach is the incorporation of gradient information into the training process (as introduced in Section 2.2.4), which significantly enhances the learned policy's accuracy and generalization. Since the gradient of the value function is naturally obtained as a by-product of standard optimization routines, using this information is an efficient way to improve learning efficiency. This method offers a scalable alternative to traditional solvers, combining data-driven function approximation with classical control-theoretic insights.

Outline of this chapter

The rest of this chapter is organized as follows. In Section 3.1, we introduce the OCP of interest, along with the associated first-order optimality conditions, presented in Section 3.2, and the semi-linear form of the SDRE approach in Section 3.3. Section 3.4 explores these optimal control techniques within the context of supervised learning, presenting a method for enabling real-time feedback control. Finally, Section 3.5 concludes with a numerical test to evaluate the effectiveness of the proposed approaches.

3.1 Consensus Control of Cucker Smale dynamics

We consider second-order systems of N interacting agents with state variables described by $s_i = (x_i, v_i) \in \mathbb{R}^d \times \mathbb{R}^d$ for $i = 1 \dots N$, where x_i and v_i represent the position and the velocity of the $i-$th agent, respectively. Given an initial configuration $\mathbf{s}(0) = \mathbf{y}$ of the

ensemble, the evolution of the agents is governed by

$$\begin{cases} \dot{x}_i(t) = v_i(t), \\ \dot{v}_i(t) = \dfrac{1}{N}\displaystyle\sum_{j=1}^{N} P\left(\|x_j(t) - x_i(t)\|\right)\left(v_j(t) - v_i(t)\right), \qquad i = 1, \dots, N, \end{cases} \tag{3.1}$$

where $P \in C^1(\mathbb{R}_+)$ is a non-increasing positive function encoding the interaction kernel. In particular, we focus on kernels of Cucker-Smale type

$$P(r) = \frac{1}{(1+r^2)}\,. \tag{3.2}$$

A remarkable property of the Cuker-Smale model is the emergence of consensus by isotropic averaging, i.e., convergence towards

$$\bar{v} = \frac{1}{N}\sum_{j=1}^{N} v_j.$$

Consensus convergence depends on the interaction strength, encoded in $P(\cdot)$, and the cohesiveness of the initial configuration [133]. Whenever consensus is not guaranteed to emerge, an external control action can be introduced to enforce convergence towards the desired pattern. To this end, the controlled collective evolution can be modeled as

$$\begin{cases} \dot{x}_i(t) = v_i(t), \\ \dot{v}_i(t) = \dfrac{1}{N}\displaystyle\sum_{j=1}^{N} P\left(\|x_j(t) - x_i(t)\|\right)\left(v_j(t) - v_i(t)\right) + u_i(t), \end{cases} \tag{3.3}$$

where each agent is influenced by an ad-hoc control component, with $\mathbf{u}(t) := (u_1, \dots, u_N)^\top$ representing a dynamical external force applied in a centralized manner. This external influence can be tuned by solving an optimal control problem

$$\mathbf{u}^*(t) = \underset{\mathbf{u}(\cdot)\in U_{adm}}{\operatorname{argmin}} \left\{ \mathcal{J}_N(\mathbf{u}(\cdot)) := \int_0^{+\infty} \frac{1}{N}\sum_{i=1}^{N} \left(\|\bar{v}(t) - v_i(t)\|^2 + \gamma \|u_i(t)\|^2 \right) dt \right\}, \tag{3.4}$$

subject to the dynamics (3.3), and where the control signal lies in the admissible set $U_{adm} := \Big\{\mathbf{u} : \mathbb{R}_0^+ \to (\mathbb{R}^d)^N \text{ is measurable } ; \mathcal{J}_N(\mathbf{u}(\cdot)) < \infty\Big\}$. In such a way, the control action is designed to penalize both the distance of the agent's velocities from the target $\bar{v}$ and the control energy required to steer the system towards the consensus manifold, with parameter $\gamma > 0$.

In the following, we compare PMP and SDRE feedback synthesis to generate data for training a neural network that approximates the state-to-control map.

3.2 Pontryagin Optimality System

As discussed in Section 2.1.2, after troncating the optimization horizon at $T < +\infty$, necessary optimality conditions for the optimal control problem (3.3)-(3.4) lead to a two-point boundary value problem for the state variables $s_i = (x_i, v_i)$, the associated adjoint variables $\lambda_i = (p_i, q_i)$, and the control signals u_i.

$$\begin{cases} \dot{x}_i &= v_i \\ \dot{v}_i &= \frac{1}{N}\sum\limits_{j=1}^{N} \dfrac{v_i - v_j}{1 + \|x_i - x_j\|^2} + u_i \\ \dot{p}_i &= -\frac{1}{N}\sum\limits_{j=1}^{N} \dfrac{2(x_i - x_j)}{(1 + \|x_j - x_i\|^2)^2}\langle q_j - q_i, v_j - v_i\rangle \\ \dot{q}_i &= -\frac{1}{N}\sum\limits_{j=1}^{N} \dfrac{q_j - q_i}{1 + \|x_i - x_j\|^2} - \frac{2}{N}(\bar{v} - v_i) + p_i \end{cases} \tag{3.5}$$

together with boundary conditions given by $(x_i(0), v_i(0)) = y_i$ and $(p_i(T), q_i(T)) = (0, 0)$ for all $i = 1, \dots, N$. The PMP system is closed with the optimality condition

$$\mathbf{u}^*(t) = \underset{u}{\operatorname{argmin}}\left(\sum_{i=1}^{N}\langle q_i, \dot{v}_i\rangle + \frac{\gamma}{N}\|u_i\|_2^2\right) = -\frac{N}{2\gamma}\mathbf{q}^\top , \tag{3.6}$$

where $\mathbf{q} = (q_1, \dots q_N)^\top$ [34]. It is possible to approximate the optimality conditions above by following a reduced gradient approach (see Section 2.1.2). Starting with an initial guess for the control signal over the whole horizon, this method iterates forward-backward passes over (3.5), while updating the control signal $u(\cdot)$ using (3.6) as a gradient step.

3.3 Semi-linear formulation

The semi-linear formulation of the dynamics of interest is derived for the ensemble configuration $\mathbf{s}$, where a two-dimensional agent's state is denoted as $(x, v) = (x^1, x^2, v^1, v^2)$:

$$\mathbf{s} := \Big(x_1^1,\, x_1^2, \dots, x_N^1,\, x_N^2,\, v_1^1,\, v_1^2, \dots, v_N^1,\, v_N^2\Big)^\top, \tag{3.7}$$

and it holds as in (2.35) for operators: $\mathcal{B} = (\mathbb{O}_{2n}, \mathbb{I}_{2n})^\top$,

$$\mathcal{A}(\mathbf{s}) = \begin{bmatrix} \mathbb{O}_{2n} & \mathbb{I}_{2n} \\ \mathbb{O}_{2n} & \tilde{\mathcal{A}}(\mathbf{s}) \end{bmatrix}, \qquad \tilde{A}_{i,j}(\mathbf{s}) = \begin{cases} P(\|x_j - x_i\|) & \text{if } i \text{ odd},\ j = i+1 \\ P(\|x_j - x_i\|) & \text{if } i \text{ even},\ j = i-1 \\ -P(\|x_j - x_i\|) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}, \tag{3.8}$$

where $\mathbb{O}_{2n}, \mathbb{I}_{2n}$ indicate the zero and the identity matrices in $\mathbb{R}^{2n\times 2n}$ respectively, $n = dN$, and $P(\cdot,\cdot)$ Cucker-Smale type kernel defined in (3.2). Furthermore, aiming at writing the consensus cost functional in quadratic form as in (2.1), we notice that

$$\sum_{i=1}^{N} \|v_i - \bar{v}\|^2 = \|\mathbf{v} - C\mathbf{v}\|^2 \tag{3.9}$$

for $\mathbf{v} = (v_1, \dots v_N)^\top$ and a matrix $C \in \mathbb{R}^{n\times n}$ full of zeros, except for the $N \times N$ diagonal blocks, which are valued $N^{-1} \cdot \mathbb{I}_N$, for $\mathbb{I}_N$ denoting a matrix in $\mathbb{R}^{N\times N}$ full of ones. It follows that

$$\|\mathbf{v} - C\mathbf{v}\|^2 = \mathbf{v}^\top \Big(\mathbb{I}_n + C^\top C - 2C\Big)\mathbf{v}\,. \tag{3.10}$$

Thus, we can write the cost (3.4) in the form (2.1) w.r.t. linear operators

$$Q = \mathbb{I}_n + C^\top C - 2C\,, \qquad R = \frac{\gamma}{N} \cdot \mathbb{I}_n\,. \tag{3.11}$$

3.4 Gradient-augmented regression and data generation

We aim to provide a model $\tilde{\mathbf{u}} \approx \mathbf{u}$ that acts as a surrogate for the feedback law, accelerating the receding horizon control of the system. The primary candidate for approximation is the state-to-control map $\mathbf{s} \mapsto \mathbf{u}(\mathbf{s})$, as it directly synthesizes the control action and serves as the objective of the learning operation.

As an alternative, the value function V can also serve as a learning target, from which a control policy can be derived using $\mathbf{u} \approx \nabla V$, according to the optimality conditions in equations (2.37) and (2.17). Besides its theoretical significance, a key advantage of selecting V as a learning objective is its property of being a scalar function, hence relatively easy to approximate. Furthermore, information about ∇V is generated for free as a byproduct of both the SDRE and PMP routines. In the SDRE case, this is expressed as $\nabla V(\mathbf{s}) \approx 2\Pi(\mathbf{s})\mathbf{s}$, where $\Pi(\cdot)$ is the solution operator. In PMP, the gradient of the value function corresponds to the adjoint variable λ.

We compare control designs learned through the state-to-control map and through the value function, with the training datasets sampled appropriately. Specifically, we sample state configurations $\{\mathbf{s}^{(i)}\}_{i=1}^{N_s}$ and associate them with the associated labels

$$\mathbf{u}^{(i)} := \mathbf{u}(\mathbf{s}^{(i)}),\ V^{(i)} := V(\mathbf{s}^{(i)}),\ \nabla V^{(i)} := \nabla V(\mathbf{s}^{(i)})$$

- In the PMP framework, we solve the optimality system (3.5-3.6) fixing each sample as initial configuration $\mathbf{y} = \mathbf{s}^{(i)}$. As done in [32], once the extremal trajectory $(\mathbf{s}^*, \mathbf{u}^*, \lambda^*)$ is computed for time $t \in [0, T]$, the variables of interest can be obtained as follows: the feedback control associated with the initial condition $\mathbf{s}^{(i)}$ is $\mathbf{u}^{(i)} = \mathbf{u}^*(t = 0)$, the optimal cost-to-go from $\mathbf{s}^{(i)}$ can be retrieved as the cost integrated along the extremal trajectory, and coincides with the value function $V^{(i)} = V(\mathbf{s}^{(i)}, t = 0)$, its gradient being $\nabla V^{(i)} = \lambda^*(t = 0)$.

- For the SDRE approach, we solve the Algebraic Riccati Equation (2.36) associated with freezing (2.40) at each current state $\mathbf{s}^{(i)}$. The target variables can then be re-

trieved by using the quadratic ansatz for $V^{(i)} = \mathbf{s}^{(i)^\top}\Pi(\mathbf{s}^{(i)})\mathbf{s}^{(i)}$, $\nabla V^{(i)} = 2\mathbf{s}^{(i)^\top}\Pi(\mathbf{s}^{(i)})$ and the first order optimality condition for $\mathbf{u}$ in (2.41) [8, 9].

Once the training data are collected, we consider models for both the control designs and target variables, denoted as:

- $\mathbf{u}_\theta^{SDRE}$, $\mathbf{u}_\theta^{PMP}$ – for the direct approximation of the mapping $\mathbf{s} \mapsto \mathbf{u}(\mathbf{s})$, with loss function $\mathcal{L}_0^{NN}$ given by the mean squared error

$$\mathcal{L}_0^{NN}(\mathbf{u}, \mathbf{u}_\theta) = \sum_{i=1}^{N_s} \frac{\|\mathbf{u}^{(i)} - \mathbf{u}_\theta(\mathbf{s}^{(i)})\|^2}{N_s} \tag{3.12}$$

- $\mathbf{u}_V^{SDRE}$, $\mathbf{u}_V^{PMP}$ – for the approximation of the feedback map through a model for the value function V. A deterministic layer is stacked on top of the FNN, encoding (2.17,2.41). In this case, we additionally incorporate a penalty term in the loss function $\mathcal{L}_1^{NN}$ to account for discrepancies in the derivative information:

$$\mathcal{L}_1^{NN}(\mathbf{u}, \mathbf{u}_V) = \mathcal{L}_0^{NN}(\mathbf{u}, \mathbf{u}_V) + \mu \sum_{j=1}^{n} \mathcal{L}_0^{NN}\left(\frac{\partial \mathbf{u}}{\partial \mathbf{s}_j}, \frac{\partial \mathbf{u}_V}{\partial \mathbf{s}_j}\right), \tag{3.13}$$

where $\mathbf{s}_j$ denotes the $j-$th dimension of the input variable $\mathbf{s}$.

3.5 Numerical Test

We apply the proposed methodologies to the high-dimensional second-order consensus problem (3.4) for a system of $N = 50$ agents in $\mathbb{R}^2$, following dynamics of Cucker-Smale type (3.3) and a control penalization parameter $\gamma = 0.1$. We consider the integration of the system in a time-frame $t \in [0, T]$ with $T = 10$ and time-step $dt = 0.01$. The solution of this optimal control problem is associated with an HJB equation cast in a $2dN = 200$ dimensional space Ω, hence the need to rely on numerical representations for the optimal feedback design.

We prescribe the initial condition for the ensemble state $\mathbf{s} \in [0, 1]^{2dN}$, and a grid search in the parametric space is performed, to select an appropriate model architecture.

	M	layers width	σ	μ	PRMSE
$\mathbf{u}_\theta^{SDRE}$	1	1000	tanh	–	0.7921%
$\mathbf{u}_\theta^{PMP}$	3	400	sigmoid	–	1.1477%
$\mathbf{u}_V^{SDRE}$	2	200, 100	sigmoid	0.07	4.1843%
$\mathbf{u}_V^{PMP}$	2	800	sigmoid	0.1	4.8775%

Table 3.1: Different model specifics, including architectural details and Percent RMSE in a test-set.

In Table 3.1, we compare the architectures of the different models, characterized by the number M of layers, the width of each one of them, the activation functions σ on the hidden layers and the gradient augmentation parameter μ. All models have been trained with the Adam solver [151] over batches of 200 samples. In the table, we also display the Coefficient of Variation or Percent Root Mean Squared Error (PRMSE) for the different models when applied to a testing dataset of 10^6 unseen sample initial conditions in $[0,1]^{2dN}$.

In Figure 3.1 a plot of the distance from the target configuration is displayed for the different models, and compared with the one resulting from both the PMP SDRE approach. Another interesting comparison is the one in terms of the required CPU time for integrating the system trajectories while controlling them via the different methods. This is presented in Table 3.2, versus the CPU time used by the forward simulation with the original solvers.

$\mathbf{u}_\theta^{SDRE}$	$\mathbf{u}_\theta^{PMP}$	$\mathbf{u}_V^{SDRE}$	$\mathbf{u}_V^{PMP}$	SDRE	PMP
13.9152	9.2503	11.8068	4.0699	1821.21	425.1578

Table 3.2: CPU time (in seconds) required for the system integration along time $t \in [0, 10]$ under the different alternative feedback designs.

3.6 Concluding remarks

In this section, we reviewed the numerical procedure proposed in [8, 7] for accelerating MPC algorithms when controlling large-scale agent-based models at the microscopic scale. Our findings show that supervised learning, combined with deep neural networks, provides an effective approach for constructing a suboptimal feedback law using a finite set of offline

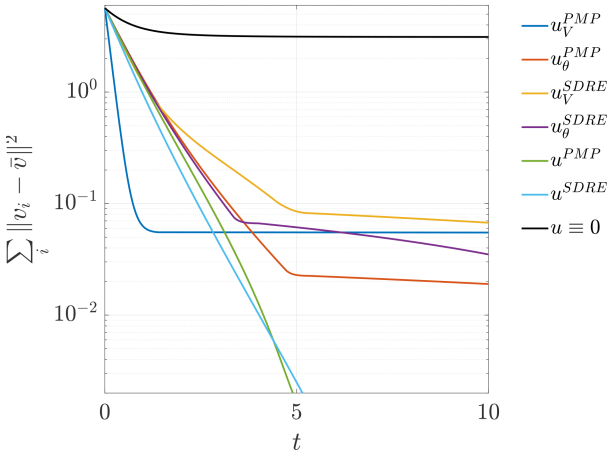


Figure 3.1: Comparison of learned and target trajectories for a random initial condition in $[0,1]^{2dN}$. All the models improve consensus with respect to the uncontrolled dynamics. In terms of learned models, the $\mathbf{u}_V^{PMP}$ model can be considered the optimal one, with an overall cost (at final time horizon T) $\mathcal{J} \approx 1.44$.

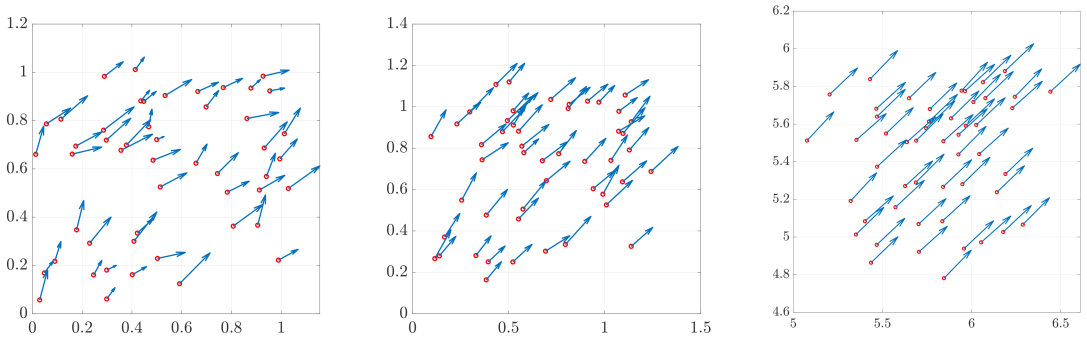


Figure 3.2: System configuration at time $t = 0, 1, 10$ seconds respectively, under the control action provided by the feedback map $\mathbf{u}_V^{PMP}$.

samples from optimal trajectories. Synthetic trajectory generation can be performed using either first-order optimality conditions in Pontryagin form or the SDRE approach. In both cases, the high dimensionality of the control system renders online computation prohibitively expensive, but this challenge can be mitigated through an offline training phase, enabling efficient real-time feedback control.

Chapter 4

Macroscopic Scale

Computation and Control of Unstable Steady States for Mean Field Multiagent Systems

As the number of agents in an ABM increases, the dimensionality of the system's state space grows rapidly, making modeling and control computationally intractable at the microscopic agent-to-agent interaction level. In these scenarios, a more efficient approach is to adopt a mean-field perspective, replacing the explicit agent-based description with a density-based representation of the population. Collective behaviors and coherent structures at the mean-field level are intrinsically linked to the underlying microscopic interactions [236, 112, 19, 201, 182]. These emergent phenomena encompass a wide range of examples, including swarming, flocking, bacterial synchronization, and socio-economic dynamics [188, 96, 4], as well as the formation of clusters in opinion dynamics [137].

Outline of this chapter.

This chapter presents the findings published in [50] and is organized as follows. Section 4.1 introduces the main objectives of this project: the computation and control of steady states for weakly interacting diffusion processes at the mean-field limit. This introduction is followed by a review of the related literature and a summary of our contributions. Sec-

tion 4.2 introduces the class of models under consideration, ranging from noisy ABMs to the Fokker-Planck equation arising in the mean-field limit. Section 4.3 details the identification of stationary states at the macroscopic level. We employ a spectral Fourier-Galerkin method for spatial discretization of the steady-state PDE. The resulting system of nonlinear equations is solved via Newton's method combined with a deflation algorithm to handle the multiplicity of solutions, a characteristic of this class of problems. We also present two numerical examples demonstrating the proposed steady-state-finding procedure and validating it against established linear stability analysis and bifurcation results. In Section 4.4, we augment the evolutionary Fokker-Planck equation with an external control force. This control signal is designed to stabilize the system at a desired unstable stationary state by optimizing a suitable cost functional. The resulting finite-horizon optimal control problem is addressed at the discrete level using first-order optimality conditions, and the control action is synthesized in feedback form via nonlinear model predictive control. Section 4.5 presents numerical results, testing this stabilization routine for two different models across one- and two-dimensional domains. Finally, Section 6.1.2 provides an appendix addressing some theoretical aspects of our numerical methodology.

4.1 Introduction

The mean-field description of ABMs driven by noise involves replacing the complex N-particle Fokker-Planck equation with a single, nonlinear, and nonlocal PDE for the evolution of the one-particle distribution function—the McKean-Vlasov equation [118, 229]. This mean-field approach has proven valuable across diverse fields, including physics and astrophysics [87, 74], biology, traffic flow modeling [138], opinion dynamics [105], and the convergence analysis of Artificial Neural Networks [245, 217, 179, 224, 223]. Emergent patterns in these systems are captured by the stationary states of the McKean-Vlasov PDE. A common characteristic is the coexistence of multiple stable and unstable configurations, with transitions between them representing disorder-order phase transitions. The number and nature of these steady states depend on the interplay between interaction and diffusion, quantified by the strengths of interaction and noise [94, 220, 67].

The **first objective** of this project is to develop accurate and efficient computational methodologies for identifying both stable and unstable stationary states of noise-driven interacting particle systems in their mean-field limit, particularly for systems exhibiting multiple emergent behaviours [97, 118, 37].

These ABMs are characterized by rich self-organization landscapes. While stable equilibria often possess large basins of attraction, convergence can be slow depending on confinement and interaction potentials [180]. For linear Fokker-Planck-type PDEs with a unique Gibbsian stationary state, introducing external forcing terms can accelerate convergence towards globally stable steady states [58, 163]. This is naturally addressed within the framework of nonlinear control theory and feedback stabilization.

Following this control-theoretical approach, our **second objective** is to study the optimal stabilization of McKean-Vlasov PDEs towards unstable steady states. As a prototypical example, we focus on opinion dynamics, particularly the noisy Hegselmann-Krause model, where the goal is to either promote or prevent the emergence of polarization or consensus, especially in the presence of a phase transition [248, 123, 67, 108]. Optimal control methodologies have been applied to synchronization problems for Kuramoto-type models [221]. The methodology developed here applies directly to this problem; indeed, we study a variant of the noisy Kuramoto model with an external potential in Section 4.3.4.

Unstable steady states, while not typically observed in forward-time simulations, are often of significant physical or functional interest. In applications ranging from climate science and neuroscience to chemical reaction networks and engineering systems, unstable steady states frequently embody functionally relevant but dynamically inaccessible targets.

Models, methods, and related literature.

For ABMs with weakly interacting diffusions, the passage to the mean field limit [229, 192] and the structure and nature of phase transitions for the mean field dynamics is well understood [94, 76, 67]. We consider McKean-Vlasov PDEs characterized by confinement and interaction potentials. Similarly to [126], see also [2, 75], we discretize the stationary

McKean-Vlasov PDE using a spectral Galerkin method, approximating the steady state solution as a truncated Fourier series. This translates into a system of nonlinear equations for the Fourier coefficients which can be written in vector form. The integrals involved in the discretized operators are computed using Gaussian quadrature. The constraint that mass is conserved, i.e. that the solution of the McKean-Vlasov PDE integrates to 1, is imposed by adding a constraint to the system. Positivity of the density is enforced by filtering out solutions with negative values.

Finding multiple solutions to nonlinear PDEs such as the Allen-Cahn or the Yamabe equation has been previously studied in [111], where an abstract deflated Newton method in Banach spaces was proposed. The same method has been applied for bifurcation analysis in [73]. This algorithm, throughly presented in Section 2.3, yields a complete taxonomy of the equilibrium points of the system, for which we formulate an optimal stabilization problem.

We focus on the control of densities at the mean field scale, that we formulate as a PDE-constrained optimization problem for the McKean-Vlasov PDE. First-order optimality conditions for this problem have been previously studied in [31, 114] for the Fokker-Planck equation, and in [13, 56, 2] for the McKean-Vlasov PDE. Although related, our work is conceptually different as it focuses on the mean-field/McKean-Vlasov nonlinear and nonlocal PDEs where multiple steady states can exist, and which cannot be stabilized using open-loop controllers. To circumvent this difficulty, we resort to the synthesis of feedback laws [58] via nonlinear mean-field MPC [28, 95, 33], and in particular to PMP policy iteration algorithm, presented in Section 2.1.2. We also mention the (formal) link between optimal control methodologies for mean field dynamics and linear response theory, e.g. the study of the response of the system to a weak external forcing (control), in particular close to the transition point [171, 254]. It was shown in these references that linear response theory breaks down close to the transition point, and that a weak external forcing can have a substantial impact on the behaviour of the mean field dynamics.

Contributions.

In summary, the main contributions of this project are:

1. The development of a numerical scheme for identifying all steady states (up to translation invariance), both stable and unstable, of noise-driven interacting particle systems in their mean field limit, posed on the torus. The proposed method combines a spectral Galerkin approximation of the McKean-Vlasov PDE with a deflated Newton's method to handle the multiplicity of solutions. This allows for a comprehensive characterization of the self-organization landscape of these systems, including the coexistence of various stationary configurations and the transitions between them.

2. The formulation and solution of an optimal control problem for stabilizing the McKean-Vlasov PDE towards an unstable steady state. The control enters as an external forcing term in the PDE and is computed by solving a finite-horizon optimization problem using first-order optimality conditions. The control is synthesized using MPC, enabling the stabilization of unstable stationary solutions in a receding horizon fashion.

3. The application of the proposed methodology to several examples of interacting particle systems exhibiting phase transitions and multiple steady states, including the noisy Hegselmann-Krause model for opinion dynamics and the Haken-Kelso-Bunz model from biophysics. The numerical results demonstrate the effectiveness of the approach in capturing the rich self-organization behaviour of these systems and stabilizing unstable configurations of interest. Detailed numerical studies illustrate the role of noise and interaction strength in shaping the landscape of stationary solutions in the considered models. We analyse the bifurcations and phase transitions in the number and stability of steady states as the system parameters are varied, providing insights into the collective behaviour of these systems.

4.2 The McKean-Vlasov PDE and Characterization of Steady States

We consider a system of N interacting particles driven by additive noise:

$$dX_t^i = -\nabla U(X_t^i)\,dt - \frac{1}{N}\sum_{j=1}^{N}\nabla W(X_t^i, X_t^j)\,dt + \sqrt{2\beta^{-1}}\,dB_t^i\,, \quad X_0^i \sim \nu\,,\ i = 1,\dots,N, \tag{4.1}$$

where X_t^i denotes the state of the $i-$th agent at time t. Here, U and W denote the confining and interaction potential, respectively, β denotes the inverse temperature, and $b_t := (B_t^1, \dots, B_t^N)$ is a standard $N-$dimensional Brownian motion. The confining potential is a symmetric function, $W(x,y) = W(y,x)$. We consider the dynamics on the torus $\mathbb{T}^d$, in one and two dimensions. The noisy Hegselmann-Krause model [248, 122, 125] that we will study later on is a prototypical example of a system of the form (4.1). The white noise in (4.1) models uncertainty and it can be easily replaced by noise processes with a non-trivial spatio-temporal structure [126]. As is well known [206, Ch. 4] that the dynamics (4.1) has a gradient structure with respect to the Hamiltonian

$$\mathcal{H}^N(X^1, \dots, X^N) := \sum_{i=1}^{N} U(x^i) + \frac{1}{2N}\sum_{i,j=1}^{N} W(x^i, x^j) \tag{4.2}$$

and reads

$$dX_t^i = -\nabla \mathcal{H}^N(X_t^1, \dots, X_t^N)\,dt + \sqrt{2\beta^{-1}}\,dB_t^i\,. \tag{4.3}$$

The $N-$particle distribution function $\rho^N(t, x_1, x_2, \dots, x_N)$ satisfies the Fokker-Planck equation associated with the $N-$particle system as

$$\partial_t \rho_t^N = \nabla_N \cdot \left(\nabla_N \mathcal{H}^N \rho_t^N\right) + \beta^{-1}\Delta_N \rho_t^N\,. \tag{4.4}$$

Under appropriate assumptions on the Hamiltonian and certainly when considering the dynamics on the torus with smooth confining and interaction potentials, the $N-$particle

Fokker-Planck equation has a unique steady state given by the Gibbs measure

$$\rho_\infty^N := \frac{1}{Z_\beta^N} e^{-\beta \mathcal{H}^N(x_1,\dots,x_N)}, \qquad Z_\beta^N := \int e^{-\beta \mathcal{H}^N(x_1,\dots,x_N)}\, dx_1 \dots x_N\,. \tag{4.5}$$

We refer to [206, Ch. 4] for details.

For exchangeable interacting particle systems and with chaotic initial data, and when the interaction strength is inversely proportional to the number of particles as in (4.2), we can pass to the mean field limit $N \to +\infty$. In particular, the $N-$particle distribution function becomes approximately the product of $1-$particle distribution functions, $\rho^N(x_1,\dots,x_N,t) \approx \Pi_{\ell=1}^N \rho(x_\ell,t)$; the $1-$particle distribution function satisfies the McKean-Vlasov PDE

$$\partial_t \rho = \nabla \cdot \Big(\rho[(\nabla W * \rho) + \nabla U] \Big) + \beta^{-1} \Delta \rho\,, \tag{4.6}$$

in the domain $\mathbb{T}^d$ with periodic boundary conditions and initial distribution $\rho(0,x) = \rho_0$, with

$$(W * \rho)(x) := \int_{\mathbb{T}^d} W(x,y)\rho(y)dy\,. \tag{4.7}$$

We refer to the classic references [229, 192] for proofs and refinements of the above and to [98, 160] for modern approaches to entropic propagation of chaos.

Unlike the $N-$particle system (4.3) that is always reversible with respect to the unique Gibbs invariant measure (4.5), it is well known that the mean field dynamics can exhibit phase transitions, i.e. non-uniqueness of stationary states [94, 220, 118, 67]. This is the case, for example, for the mean field dynamics in unbounded domains and for a non-convex confining potential and a quadratic interaction potential at sufficiently low temperatures [94, 242]. For the dynamics on the torus, which is the case that we will consider in this chapter, the concept of convexity is replaced by that of $H-$*Stability* [218, 76, 67]: The mean field dynamics exhibits a unique stationary state at all temperatures if and only if the interaction potential has no non-negative Fourier coefficients. When the interaction potential has negative Fourier coefficients, then a phase transition occurs at sufficiently

low temperatures. We refer to [76, 67] for a detailed analysis of phase transitions in the absence of a confining potential and to [231, 97, 48] for partial results when a confining potential is present on the torus. It is important to note that, in the absence of a confining potential, the uniform distribution on the torus is always a stationary state. Furthermore, past the phase transition, steady states are translation invariant [47]. We can identify a particular steady state by choosing appropriately the initial condition [48]. Translation invariance is removed by the addition of a confining potential, e.g. a "magnetic field" [97].

Several equivalent characterizations of stable and unstable steady states of (4.6) exist: they can be defined as solutions to the steady state PDE

$$\beta^{-1}\Delta\rho + \nabla\cdot\Big(\rho\big[(\nabla W * \rho) + \nabla U\big]\Big) = 0\,, \tag{4.8}$$

as critical points of the free energy functional

$$\mathcal{F}[\rho] = \beta^{-1}\int_{\mathbb{T}^d}\rho\log(\rho)\,dx + \int_{\mathbb{T}^d}U\rho\,dx + \frac{1}{2}\int_{\mathbb{T}^d}\int_{\mathbb{T}^d}W(x,y)\rho(x)\rho(y)\,dxdy\,, \tag{4.9}$$

or fixed points of the Euler-Lagrange equations for the free energy functional [41]

$$\rho = \frac{1}{Z}e^{-\beta(U+W*\rho)}, \quad Z := \int_{\mathbb{T}^d}e^{-\beta(U+W*\rho)}\,dx. \tag{4.10}$$

Alternatively, it can be shown that this integral equation, the *Kirkwood-Monroe map* is equivalent to the stationary McKean-Vlasov PDE (4.6); see [231, Thm 4.1] or [104, Thm. 1]. A rigorous proof of the equivalence between steady states obtained through the previously mentioned methods is presented in [67, Prop 2.4]. Furthermore, [67, Thm 2.3] establishes the regularity of steady state solutions as $\rho \in H^1 \cap \mathcal{P}_{ac}(\Omega)$, where $\mathcal{P}_{ac}(\Omega)$ denotes the space of absolutely continuous probability measures over Ω. The solutions are in fact strictly positive [190].

We reiterate that finding critical points of the free energy functional (4.9), roots of the nonlinear map $F_\kappa[\rho] := \rho - \frac{1}{Z}e^{-\beta(U+W*\rho)}$ in (4.10) and stationary solutions of the McKean-

Vlasov PDE are *equivalent* [Prop 2.4][67].[1] One of the main goals of this project will be the development of efficient numerical methodologies that enable us to calculate all solutions of the Kirkwood-Monroe equation, i.e. all critical points of the free energy functional.

4.3 Computing multiple steady states in the McKean-Vlasov PDE

The variety of applications of this problem has motivated a flourishing literature on numerical solvers for (4.6). For a review of finite difference and finite element methods, we refer the reader to [210], beyond which we cite the structure preserving scheme proposed in [204], where the positivity of ρ and its mass preservation are guaranteed by construction. Alternative solvers include Spectral Methods [257, 70], Deep Learning [253, 77, 168, 232], and Tensor Train approximation [103].

We are concerned with the identification of invariant probability measures of (4.1) obtained as solutions of the stationary McKean-Vlasov PDE (4.8). The proposed numerical procedure relies on a spectral approximation of ρ in terms of L Fourier modes, for which (4.6) can be written as a bilinear system of L equations. In these settings, the identification of steady states reduces to a root finding problem, addressed via Newton's method. In order to identify distinct solutions, typically one applies the routine multiple times starting from many different initial guesses, and relying on the assumption that those starting points lie in different basins of attraction. Here instead, we rely on the infinite-dimensional deflation algorithm proposed in [111] and presented in Section 2.3, where the PDE residual is iteratively modified in such a way to eliminate known solutions from consideration. Alternative approaches include numerical continuation [72, 78], and numerical integration of the Davidenko differential equation associated with the original nonlinear problem [57].

[1] In fact, these statements are also equivalent to finding global minimizers of the entropy dissipation functional, but this observation will not be used in this chapter.

4.3.1 Spectral Galerkin method for the Stationary Fokker–Planck equation

We consider a Galerkin approximation of the stationary Fokker–Planck equation (4.8). Its weak form over a periodic domain Ω reads

$$\beta^{-1} \int_\Omega \Delta\rho\, \varphi \, dx + \int_\Omega \nabla \cdot \Big(\rho \big[(\nabla W * \rho) + \nabla U \big] \Big) \, \varphi \, dx \; = 0, \tag{4.11}$$

with test functions $\varphi \in \mathcal{V} := \{\varphi : \Omega \to \mathbb{R} \mid \varphi \text{ smooth, periodic}\}$. After integrating by parts and under the periodicity assumption for both ρ and φ, this leads to

$$0 = -\beta^{-1} \Big(\int_\Omega \nabla\rho \cdot \nabla\varphi \, dx \Big) - \Big(\int_\Omega \big(\rho \big[(\nabla W * \rho) + \nabla U \big] \cdot \nabla\varphi dx \Big) \,, \qquad \forall \varphi \in \mathcal{V} \,. \tag{4.12}$$

We consider a global spectral basis $\{\psi_i\}_{i=1}^L$ spanning an $L-$dimensional subspace $\mathcal{V}_L \subset \mathcal{V}$. The approximate solution ρ^L is then to be found in $\mathcal{V}_L$, parametrized as $\rho^L(x) = \sum_{n=1}^L a_n \psi_n(x)$, and to be tested against an arbitrary $\varphi \in \mathcal{V}_L$ of the form $\varphi(x) = \sum_{m=1}^L k_m \psi_m(x)$. In particular, we consider $k_m = e_m$ to be the $m-$th standard unit vector, leading to $\varphi = \psi_m$ for all $m = 1, \ldots, L$. This leads to the equation

$$-\beta^{-1} \sum_{n=1}^L a_n \int_\Omega \left(\nabla\psi_n + \psi_n \, \nabla U \right) \cdot \nabla\psi_m \, dx - \sum_{n,k=1}^L a_n a_k \int_\Omega \left(\psi_n \big[(\nabla W * \psi_k) \big] \right) \cdot \nabla\psi_m \, dx = 0 \tag{4.13}$$

holding for all $m = 1, \ldots, L$. This system of nonlinear equation can be written in vector form for $\mathbf{a} := (a_1, \ldots, a_L)^\top$ as

$$0 = -(\beta^{-1} K + C)\mathbf{a} - bil(\mathbf{a}) \tag{4.14}$$

with stiffness matrix $K \in \mathbb{R}^{L \times L}$, forcing term $C \in \mathbb{R}^{L \times L}$, and bilinear form $bil(\cdot) \in \mathbb{R}^L$ defined as

$$\big[K\big]_{n,m} = \int_\Omega \nabla\psi_n(x) \cdot \nabla\psi_m(x) \, dx \,, \qquad \big[C\big]_{n,m} = \int_\Omega \psi_n(x) (\nabla U \cdot \nabla\psi_m(x)) \, dx \tag{4.15}$$

$$bil(\mathbf{a}) = \begin{bmatrix} \mathbf{a}^\top b_1 \, \mathbf{a} \\ \vdots \\ \mathbf{a}^\top b_L \, \mathbf{a} \end{bmatrix} \qquad \mathbb{R}^{L\times L} \ni \left[b_k\right]_{n,m} = \int_\Omega \psi_n (\nabla W * \psi_k) \cdot \nabla \psi_m \, dx \tag{4.16}$$

In the numerical tests, all the above integrals, including the convolution in (4.16), are computed by means of Gaussian quadrature rule. The number of quadrature points considered will be specified in each example. Without loss of generality, we consider $\psi_1, \dots, \psi_L$ to be the first L Fourier modes in Ω, but any orthonormal basis of functions globally defined in space can be used instead. The main burden it terms of computational time (and memory) relies in the calculation and storage of the three-way tensor $b \in \mathbb{R}^{L\times L\times L}$, and can be partially mitigated by parallelization.

Imposing the positivity and density conditions

Since $\rho^L \approx \rho$ is the approximation of a probability density function we need to impose that it is strictly positive (for a strictly positive initial condition) and that it conserves mass at all times, and in particular at the level of steady states

$$\rho^L(x) > 0 \quad \forall x \in \Omega \,, \qquad \int_\Omega \rho^L(x) dx = 1 \,. \tag{4.17}$$

The mass conservation constraint can be imposed directly on the system (4.14) by means of an additional matrix row. We sketch the idea in the two-dimensional case $x = (x^1, x^2)$, but it can be easily generalized to arbitrary dimensions. Assuming we have discretized the domain Ω in a mesh having $N+1$ nodes in both the dimensions, with the same discretization step h, we have

$$\sum_{i=1}^{L} a_i \sum_{j,k=1}^{N+1} \psi_i(x_j^1, x_k^2) h^2 \approx \int_\Omega \rho(x) dx \,. \tag{4.18}$$

Moreover, using the notation $\zeta_i = \sum_{j,k=1}^{N+1} \psi_i(x_j^1, x_k^2)h^2$, the imposition of the density condition is equivalent to require

$$\bar{\zeta}\mathbf{a} = 1\,, \qquad \text{where} \quad \bar{\zeta} = \begin{bmatrix} \zeta_1 & \dots & \zeta_L \end{bmatrix}. \tag{4.19}$$

Then, our modified system of equations will be

$$0 = -\tilde{A}\mathbf{a} - \tilde{b}(\mathbf{a})\,. \tag{4.20}$$

with operators

$$\tilde{A} = \begin{bmatrix} \beta^{-1}K + C \\ \bar{\zeta} \end{bmatrix}, \qquad \tilde{b}(\mathbf{a}) = \begin{bmatrix} bil(\mathbf{a}) \\ -1 \end{bmatrix}. \tag{4.21}$$

On the other hand, the positivity constraint is handled by filtering out – once a set of candidate steady states is generated – all the ρ_∞'s that display negative values for at least one point in the mesh.

4.3.2 Deflation algorithm for steady state identification

After the spectral discretization and the imposition of the L^1 conservation constraint, we are left with the task of finding the (possibly) multiple solutions of the discretized Kirkwood-Monroe map

$$\mathcal{F}(\mathbf{a}) := -\tilde{A}\mathbf{a} - \tilde{b}(\mathbf{a}) = 0\,. \tag{4.22}$$

As proposed in [111] and presented in Section 2.3, we combine the dard Newton's method with a deflation technique to systematically eliminate known solutions from consideration. The PDE residual $\mathcal{F}$ is iteratively modified so that the roots of the deflated operator $\mathcal{D}$ coincide with the remaining unknown solutions:

$$\mathcal{D}(\mathbf{a}) = \mathcal{M}(\mathbf{a}, \mathbf{r})\mathcal{F}(\mathbf{a}) \tag{4.23}$$

where for each $\mathbf{r} \in \Omega$, $\mathbf{a} \in \Omega \setminus \{\mathbf{r}\}$ $\mathcal{M}(\mathbf{a}, \mathbf{r})\mathcal{F}(\mathbf{a}) = 0$ if and only if $\mathcal{F}(\mathbf{a}) = 0$. The operator $\mathcal{M}$ is constructed as a shifted exponential-norm deflation operator:

$$\mathcal{M}(\mathbf{a}, \mathbf{r}) = \left(\frac{\mathbb{I}}{\eta(\mathbf{a})} + \mathbb{I}\xi \right), \tag{4.24}$$

while the derivative of $\mathcal{D}$ is efficiently computed using a recursive formula (see 2.3.1).

The deflation loop stops when the root finding technique diverges for a given deflated operator and initial condition $\hat{\mathbf{a}}$. In the numerical tests we adopt the following stopping criteria for the Newton's iteration: at the $n-$th step, either we have $\|\mathbf{a}^n - \mathbf{a}^{n-1}\| \leq \varepsilon$, with $\varepsilon = 10^{-10}$, or $n \geq 10^3$.

Remark 1. *In this project, we apply the deflated Newton's method to find steady states of a PDE arising as the gradient flow of a free energy functional. The deflated operator $\mathcal{D}$ used in this work is not to be considered as the gradient of a modified energy. The purpose of deflation is purely numerical: it alters the residual of the steady-state equation to avoid convergence to previously known solutions when applying Newton-type solvers. Although $\mathcal{D}$ shares the same nontrivial roots as the original operator (excluding deflated ones), it is not meaningful to simulate $\mathcal{D}$ as a dynamical system and convergence to a steady state of the original system is not guaranteed.*

4.3.3 The Haken-Kelso-Bunz Model

In this section, we apply the root-finding procedure described above to a one-dimensional mean-field model on $[0, 2\pi]$ with periodic boundary conditions. In particular, we consider the Haken-Kelso-Bunz (HKB) model [118, Sec. 5.3.3] which was introduced as a model for human hand movement. This model exhibits phase transitions and has a rich set of (stable and unstable) stationary states; see also [27]. We then compare the results of our numerical simulations with known analytical results.

We consider the PDE (4.6) for $\rho : [0, 2\pi] \to \mathbb{R}$ for confining and interaction potentials,

respectively,

$$W(x,y) = -\kappa\, cos(x-y)$$
$$U_{HKB}(x) = -c_1\, cos(x) - c_2\, cos(2x) \tag{4.25}$$

with parameters c_1, $c_2 \in \mathbb{R}$ and $\kappa \geq 0$. We consider the case for $c_1 = 0, c_2 = -\alpha$, $\alpha = -1$. For the initial guess in Newton's method, we consider a uniform random vector a_0 with entries normalized such that $\rho(0)$ is a distribution. We consider bifurcations with respect to the interaction parameter κ, and we fix the inverse temperature $\beta^{-1} = 1$. The choice of the model parameter values is motivated by the analysis provided in [118, Sec. 5.3.3]. As expected, we see that, when the interaction strength κ is below the critical value $\kappa < \kappa_c$, the PDE has a unique stationary solution. This stationary state, for the interaction and confining potentials (4.25), is calculated in [27, 48]. However, when κ is above the critical value $\kappa > \kappa_c$, we expect to have multiple steady states, as displayed in Figure 4.1.

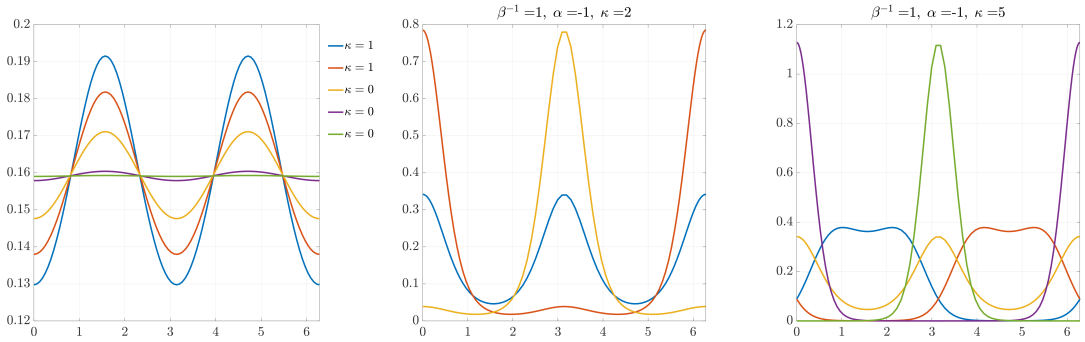


Figure 4.1: Steady states for the McKean–Vlasov PDE with potentials in Eq. (4.25) via deflated Newton's method for different interaction strength κ. Weak interaction regimes lead to a unique stable steady state, that collapses into the uniform distribution as $\kappa \to 0$ (left). However, for stronger interactions with $\kappa = 2$ and $\kappa = 5$ (centre and right, respectively), multiple stable and unstable stationary configurations coexist.

Verification of Solutions

We solve the time-dependent McKean–Vlasov PDE (4.6) for the HKB potential (4.25). The stability and bifurcation analysis presented in [118, Sec. 5.3.3], on which we rely to validate the deflated Newton's method as a steady-state finding procedure.

The McKean–Vlasov PDE with potential as in (4.25) reads

$$\partial_t \rho = \beta^{-1}\partial_x^2 \rho + \partial_x \Big[\rho \Big(2\alpha \, sin(2x) + \kappa \int_\Omega sin(x-y)\rho(y)dy \Big) \Big] , \tag{4.26}$$

with steady states that are solutions of the Kirkwood-Monroe integral equation

$$\rho_\infty(x) = Z^{-1} \exp\Big\{ -\beta \Big(\alpha \, cos(2x) - \kappa \int_\Omega cos(x-y)\rho_\infty(y)dy \Big) \Big\}, \tag{4.27}$$

for Z normalization constant. Introducing the notation

$$\langle g(X) \rangle := \int_\Omega g(x)\rho(t,x)dx \, , \tag{4.28}$$

we can write equation (4.26) in the form

$$\partial_t \rho = \beta^{-1}\partial_x^2 \rho + \kappa \, \partial_x \Big[\rho \Big(2\alpha \, sin(2x) + \big(sin(x)\langle cos(X) \rangle + cos(x)\langle sin(X) \rangle \big) \Big) \Big] , \tag{4.29}$$

for which the steady states come as solutions of the integral equation

$$\rho_\infty(x, \mathbf{m}) = Z(\mathbf{m})^{-1} \exp\Big\{ -\beta \Big(\alpha \, cos(2x) - \kappa \big[cos(x) \, m_1 + sin(x) \, m_2 \big] \Big\} . \tag{4.30}$$

Here, the parameters $\mathbf{m} = (m_1, m_2)$ are placeholders for the expectations $\langle cos(X) \rangle$, $\langle sin(X) \rangle$ w.r.t. ρ_∞. This means we can solve equation (4.30) by coupling with the self-consistency equation

$$\mathbf{m} = \mathbf{r}(\mathbf{m}) \qquad \text{for } \; \mathbf{r} = (R_1, \, R_2) \tag{4.31}$$

$$\text{for} \quad \mathbf{r}(\mathbf{m}) := \Big(\int_\Omega cos(x)\rho_\infty(x, \mathbf{m})dx, \int_\Omega sin(x)\rho_\infty(x, \mathbf{m})dx \Big) . \tag{4.32}$$

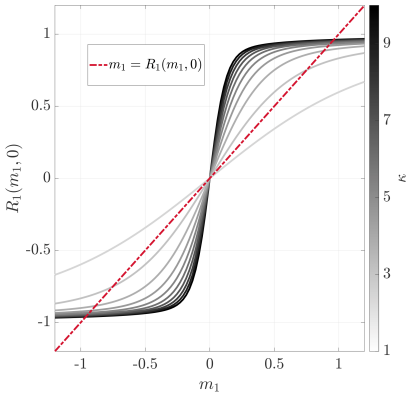


Figure 4.2: Solution to the self-consistency equation (4.31) for fixed $\beta = 1$ and $\alpha = -1$, with varying interaction strength parameter κ. The dashed red line represents self-consistency, and its intersections with the curves identify symmetric stationary solutions of Eq. (4.30). In weak interaction regimes (similar to high noise $\beta^{-1} \gg 1$), all solutions collapse into a single stable solution, indicating a saddle-node bifurcation.

Remark 2. *Each one of the (possibly) many solutions* $\mathbf{m}$ *of the self-consistency equation* (4.31) *is uniquely linked to a steady state. Therefore,* $\mathbf{m}$ *serves as a vector-valued order parameter for detecting phase transitions. Given a steady state* $\tilde{\rho}$*, it is possible to estimate the associated* $\mathbf{m}$ *via Linear Least Squares regression (LSS) for coefficients* a, b, c *defined as*

$$\underbrace{log(\tilde{\rho})}_{Y} = -\beta\, U(x) \underbrace{-log(Z)}_{a} + \underbrace{\beta\, \kappa\, cos(x)\, m_1}_{U_1(X)\, b} + \underbrace{\beta\, \kappa\, sin(x)\, m_2}_{U_2(X)\, c} . \tag{4.33}$$

We will use this to estimate the order parameters $\mathbf{m}$ *associated with the stationary solutions that we find via deflated Newton's method, and then check if they are consistent with* (4.31).

From the stability analysis developed in [118], we expect a phase transition between a monostable and a multistable regime, which is studied here separately with and without additional symmetry assumptions that are discussed below.

Symmetric solutions

Due to the even symmetry of the confining potential $U(x)$, the second component $R_2(m_1, 0)$ vanishes. We focus then on solutions of the self-consistency equation (Eq. (4.31)) of the form:

$$m_1 = R_1(m_1, 0), \qquad m_2 = 0, \tag{4.34}$$

which correspond to symmetric steady state configurations. Figure 4.2 illustrates how the number of solutions to equation (4.34) varies with the inverse temperature parameter. Following [118], we expect the non-negative solutions $m_1 \geq 0$ to parameterize stable steady states, while negative solutions lead to unstable configurations. The self-consistency argument ensures that at least one symmetric stable steady state always exists (associated with $R_1(0,0) = R_2(0,0) = 0$). Figure 4.4 verifies that the deflation algorithm finds all the expected symmetric solutions. This is obtained by considering $l = 100$ Fourier modes, 200 Gauss-points, $p = 2$, $\xi = 1$, and with initial condition $\mathbf{a}^0$ given by the zero vector. All the steady states obtained via Newton's method give a residual (4.22) $\|\mathcal{F}(a)\|_{\ell^2} \leq 10^{-15}$.

Dropping the symmetry assumption

For validating both symmetric and non-symmetric solutions (as in Figure 4.1), the full self-consistency equation (4.31) with $\mathbf{m} = (m_1, m_2)$ is needed. We verify these solutions by comparing the LSS-estimated $\mathbf{m}$ with the intersection of the diagonal planar sections in Figure 4.3. In Figure 4.5 we superimpose the self-consistency solutions over the free

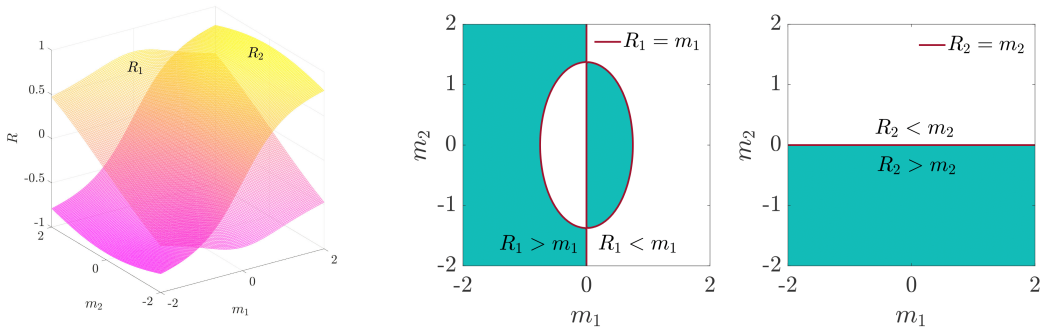


Figure 4.3: $\kappa = \mathbf{2}$: The complete self-consistency equation is linked to two surfaces, one for each component of $\mathbf{r}(\mathbf{m})$ (left). The solutions $\mathbf{m}$ lie then on the diagonal planar section of those surfaces w.r.t. m_1 and m_2 respectively (right).

energy levels; this allows us to identify stable and unstable configurations. The figures have been obtained by considering $L = 50$ Fourier modes, 100 Gauss-points, $p = 4$, $\xi = 1$, and with initial condition $\mathbf{a}^0$ given by a random uniform vector normalized such that the associated ρ is a distribution. All the steady states obtained via Newton's method give a residual (4.22) $\|\mathcal{F}(a)\|_{\ell^2} \leq 10^{-15}$.

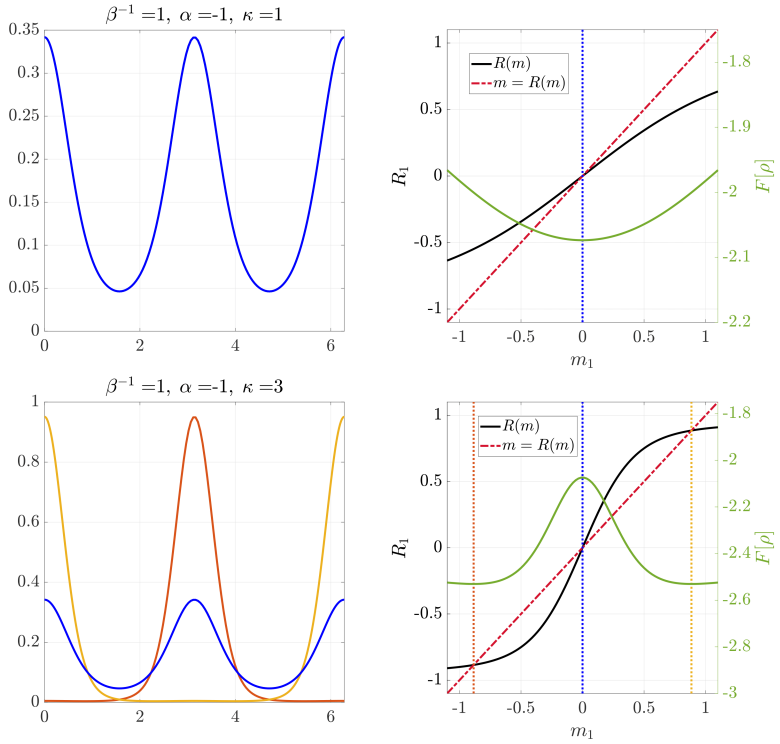


Figure 4.4: **Left**: Solutions obtained via the deflated Newton's method for two different parameter configurations, leading to a single solution (top) and triple solutions (bottom), respectively. **Right**: The self-consistency equation (4.34) and the free energy landscape. The dashed vertical lines represent the m_1 values estimated by the LSS method. The colouring of the curves corresponds to the solutions displayed on the left. All figures are generated with $\alpha = -1$, $\beta = 1$, and varying interaction strength $\kappa = 1$ (top) and $\kappa = 3$ (bottom). The green curve shows the free energy $F[\rho]$ of the system, where stable and unstable configurations correspond to minimizing and non-minimizing critical points, respectively.

4.3.4 Phase transitions for the $O(2)$ model with magnetic field

To validate the proposed root-finding procedure numerically, we apply it to the problem of the $O(2)$ model with an external magnetic field; for this problem, the stationary states can be computed analytically [97, Lemma 1.26]. We consider the state space $\Omega = [0, 2\pi]$ with interaction and confining potentials

$$W(x, y) = -cos(2\pi(x - y)), \qquad U(x) = -\eta cos(2\pi x), \tag{4.35}$$

respectively, with $\eta \in (0, 1)$. The presence of a small confinement (e.g. $\eta = 0.05$) is enough to break the translation invariance of the problem. In [97, Lemma 1.26], it is

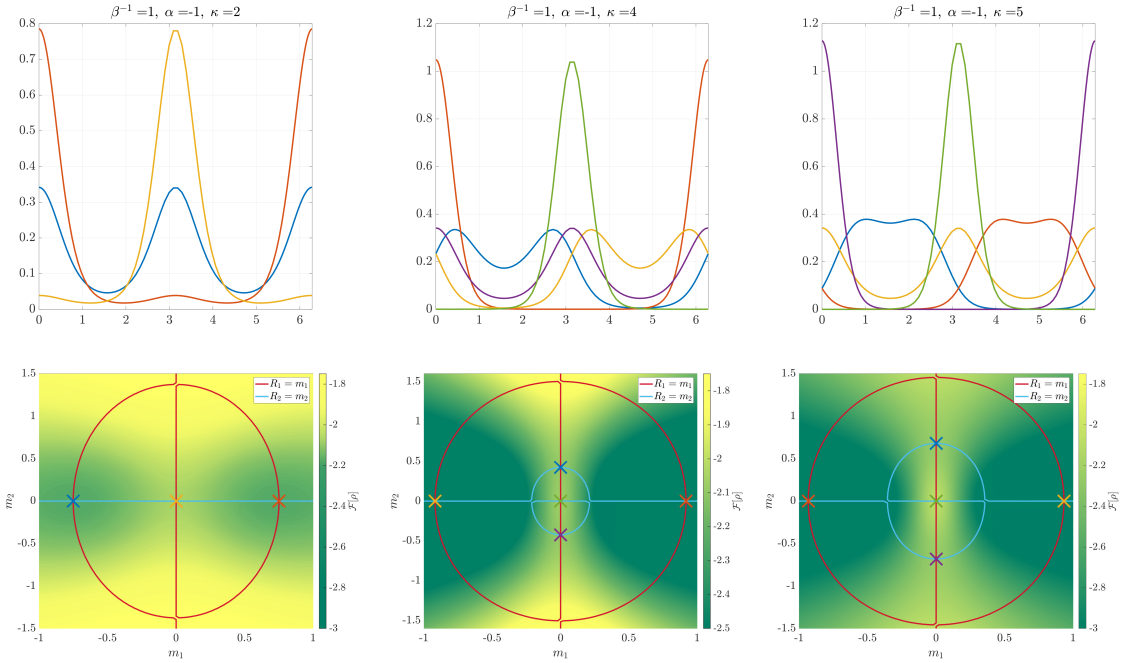


Figure 4.5: We display all the solutions (symmetric and asymmetric) found via deflation and we compared the estimated values of m_1 and m_2 with the contour plot and free energy landscape for different parameter configurations: $\kappa = \mathbf{2}$ (left), $\kappa = \mathbf{4}$ (center), $\kappa = \mathbf{5}$ (right). The estimated parameters lie in the desired intersections, validating the numerical solutions.

proved that there exists a critical temperature β_c^{-1} such that:

- for $\beta^{-1} > \beta_c^{-1}$ there exists a unique stable steady state parameterized by

$$\rho_{min} = Z_{min}^{-1} e^{\alpha_{min} cos(2\pi x)}, \qquad Z_{min} = \int_\Omega e^{\alpha_{min} cos(2\pi x)} dx \qquad \text{with } \alpha_{min} > 0; \quad (4.36)$$

- for $\beta^{-1} < \beta_c^{-1}$ we expect *at least* two steady state solutions of the form

$$\begin{aligned} \rho_{min} &= Z_{min}^{-1} e^{\alpha_{min} cos(2\pi x)}, \qquad Z_{min} = \int_\Omega e^{\alpha_{min} cos(2\pi x)} dx \\ \rho_* &= Z_*^{-1} e^{\alpha_* cos(2\pi x)}, \qquad Z_* = \int_\Omega e^{\alpha_* cos(2\pi x)} dx \end{aligned} \quad (4.37)$$

for $\alpha_* < 0 < \alpha_{min}$ with both constants depending on β^{-1}. At low temperatures, ρ_{min} is the unique minimizer (ρ_* being a non-minimizing critical point) of the free energy. This classifies ρ_{min} and ρ_* as stable and unstable steady states, respectively.

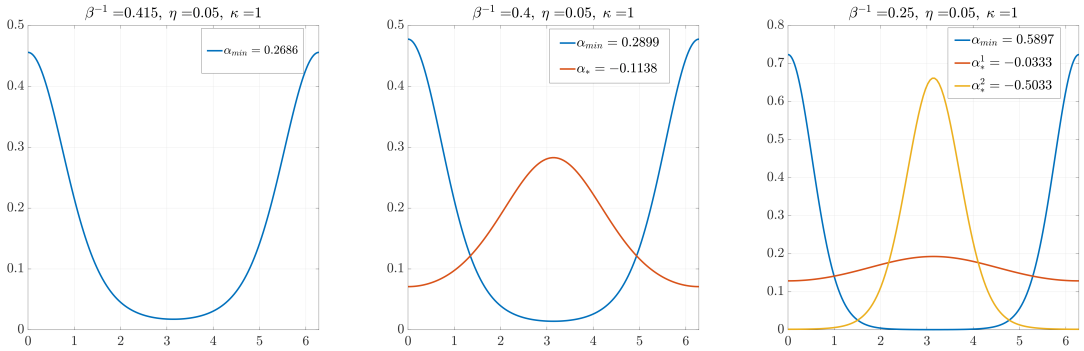


Figure 4.6: Steady states for the modified noisy Kuramoto model (confining strength parameterized by $\eta = 0.05$) obtained via deflated Newton's method. We label the solutions with the LSS-estimated coefficients α_{min} and α_*'s.

The results of our numerical experiments are displayed in Figure 4.6 , and are obtained with the following parameters choice: $L = 10$ Fourier modes, 100 Gauss-points, $p = 2$, $\xi = 0.05$, and $\mathbf{a}^0$ is the uniform distribution. The coefficients α_{min} and α_*'s, estimated using the LSS method as done in the previous example, match the expected behaviour. In particular, regardless the choice of β^{-1}, we always retrieve a positive $\alpha_{min} > 0$, associated with a stable configuration. Below the critical temperature, which we place numerically between $0.4 < \beta_c^{-1} < 0.415$, other unstable steady states, parametrized by $\alpha_* < 0$, start to appear.

4.4 Optimal feedback stabilization of unstable steady states

Having determined all possible steady state solutions to our problem, we now turn our attention to the design of control laws which can effectively steer the McKean-Vlasov PDE towards a given stationary distribution. For this, we consider the mean-field equation with a control term in the drift

$$\partial_t \rho = \beta^{-1} \Delta \rho + \nabla \cdot \Big(\rho((\nabla W * \rho) + \nabla U + u) \Big), \qquad (t, x) \in \Omega \times [0, T], \tag{4.38}$$

where $u = u(t, x) \in L_2(0, T; L_\infty(\Omega))$ is a time-dependent vector field representing an external control signal. In the following we refer to $\rho \in \mathcal{W}(0, T)$ as the weak solution to

(4.38), where the Bochner space reads

$$\mathcal{W}(0,T) := \{\rho \in L_2(0,t;H^1(\Omega)) \,|\, \partial_t \rho \in L_2(0,T;H^{-1})\}\,,$$

with H^{-1} dual space of H^1.

A natural control problem arising in this context is the acceleration of convergence towards a stable steady state, which can be achieved by studying the linearized control problem around an equilibrium [58]. In this project, our goal is to choose a control law to steer the solution to the McKean-Vlasov PDE towards a prescribed unstable steady state. The synthesis of such a control signal is achieved by casting the following dynamic optimization problem:

$$\min_u \mathcal{J}(u) := \frac{1}{2}\int_0^T \|\rho(t,\cdot) - \rho_\infty\|^2_{L_2(\Omega)} + \gamma \|u(t,\cdot)\|^2_{L_2(\Omega,\mathbb{R}^d)}\, dt \, + \Psi(\rho(T,x))\,, \tag{4.39}$$

subject to (4.38). In this formulation $T > 0$ denotes a prescribed prediction horizon for stabilization, ρ_∞ corresponds to the desired unstable steady state to be tracked, $\gamma > 0$ is a control energy penalty, and $\Psi(\rho)$ is a terminal cost. In finite horizon optimal control problems, this terminal penalty plays a crucial role in enforcing stabilization towards the terminal state, representing a relaxation of the terminal constraint $\rho(T,x) = \rho_\infty$. In our case, we consider a terminal penalty given by

$$\Psi(\rho) = \eta \, \|\rho(T,\cdot) - \rho_\infty\|^2_{L_2(\Omega)}\,, \qquad \eta \geq 0\,. \tag{4.40}$$

Remark 3. *The non-negativity and mass conservation conditions have not yet been addressed in the optimal control problem. However, these properties are inherently satisfied when the initial condition $\rho_0 \in L_2(\Omega)$ is non-negative and has an integral equal to 1. For a more detailed discussion on this topic, we direct the reader to Corollary 2.4 in [159].*

4.4.1 First-order necessary optimality conditions

To solve the optimal control problem (4.39), we derive first-order conditions for optimality, consistently to what was presented in Section 2.1.2. These conditions are obtained by analysing the behaviour of $\mathcal{J}$ under infinitesimal perturbations of the control and, consequently, the state variable. Assuming $\mathcal{J}(u)$ is Fréchet differentiable (proof in Appendix A.1) with derivative denoted by $D\mathcal{J}(u)[\cdot]$, every optimal control signal $u^* \in L_2(0,T;L_\infty(\Omega))$ satisfies

$$D\mathcal{J}(u^*)[h] = 0 \qquad \forall h \in L_2(0,T;L_\infty(\Omega))\,. \tag{4.41}$$

The condition (4.41) can be rewritten in a more convenient form by means of a system of equations for the triple $(u,\, \rho,\, \lambda)$, where $\lambda \in \mathcal{W}(0,T)$ denotes the adjoint state variable associated with the pair (ρ, u)

$$\begin{cases} \partial_t \rho = \beta^{-1}\Delta\rho + \nabla\cdot\Big(\rho\big((\nabla W * \rho) + \nabla U + u\big)\Big) & \text{(state)} \\ -\partial_t \lambda = \beta^{-1}\Delta\lambda - \nabla\lambda\cdot\Big(\nabla W * \rho + \nabla U + u\Big) + \Big(\nabla W \star (\nabla\lambda\,\rho)\Big) + (\rho - \rho_\infty) & \text{(adjoint)} \\ \rho(x,0) = \rho_0(x), \qquad \lambda(t,x) = \nabla_\rho \Psi(\rho(T,x)) & \text{(b.c.'s)} \\ -\nabla\lambda\rho + \gamma u = 0 & \text{(optimality)} \end{cases} \tag{4.42}$$

and where the operation $\star$ is defined as

$$(f \star g)(x) := \int_\Omega f(y,x)g(y)dy\,. \tag{4.43}$$

This optimality system can be immediately derived from the first order condition (4.41) combined with the derivation of a computational expression for $D\mathcal{J}(u^*)[\cdot]$ [241, 114, 159, 31, 29]. The full derivation can be found in Appendix A.2

If an optimal control signal u^* exists that solves (4.39), then the optimal triple (ρ^*, λ^*, u^*), comprising the u^*-controlled state ρ^*, the corresponding adjoint λ^*, and u^* itself, satisfy eq. (4.41). For a comprehensive derivation and well-posedness analysis of optimality

conditions for a similar control problem, we refer the reader to [114].

In the following, we discuss the construction of a numerical scheme for the approximation of such an optimal triple by following a discretize-then-optimize strategy. That is, instead of approximating the continuous optimality system (4.41), we derive optimality conditions for a discretized version of (4.39), which has a similar forward-backward structure as in the continuous case. For this, following the method of lines, we consider the spatial approximation in $\mathcal{V}_L$ of the target steady state ρ_∞, the distribution $\rho(t,\cdot)$ and the control signal $u(t,\cdot)$ at each fixed time t as $L-$truncated Fourier series. A semi-discrete problem is then obtained by embedding the time dependency at the level of the Fourier coefficients

$$\rho_\infty^L(x) = \sum_{i=1}^{L} \tilde{a}_i \psi_i(x)\,, \qquad \rho^L(t,x) = \sum_{i=1}^{L} a_i(t)\psi_i(x)\,, \qquad u^L(t,x) = \sum_{i=1}^{L}\sum_{j=1}^{d} u_{i,j}(t)\psi_i(x)\,. \tag{4.44}$$

The discretization maintains the assumption that the control signal $u^L(t,\cdot) \in L_\infty(\Omega)$ for all $t \in [0,T]$. This is ensured by the boundedness of our basis functions $|\psi_i(x)|L\infty \leq 1$ $i = 1,\dots,L$, and by u^L being a finite sum of such bounded functions, hence remaining in L_∞. The associated L-dimensional approximation of (4.38) reads

$$M\,\dot{\mathbf{a}}(t) = -(\beta^{-1}K + C)\,\mathbf{a}(t) - bil(\mathbf{a}(t)) - d(\mathbf{a}(t),\mathbf{u}(t))\,, \tag{4.45}$$

where the operators K, C, $bil(\cdot)$ are as in (4.15-4.16), $\mathbf{u} \in \mathbb{R}^{L\times d}$, and the mass matrix M has entries

$$M_{n,m} = \int_\Omega \psi_n(x)\psi_m(x)dx\,, \tag{4.46}$$

whilst the bilinear control operator reads

$$d(\mathbf{a},\mathbf{u}) = \sum_{i,k=1}^{L}\sum_{j=1}^{d} a_i(t)\,u_{k,j}(t)\int_\Omega \psi_i \frac{\partial \psi_m}{\partial x_j}\psi_k\,dx, \qquad m = 1,\dots,L\,. \tag{4.47}$$

Under this discretization, denoting the $j-$th column of the control coefficients as $\mathbf{u}_j \in \mathbb{R}^L$,

the finite-dimensional approximation of the optimal control problem reads

$$\min_{\mathbf{u}(\cdot)\in\mathcal{V}_L^d} \mathcal{J}_L(\mathbf{u}) := \frac{1}{2}\int_0^T (\mathbf{a}(t)-\tilde{\mathbf{a}})^\top M\,(\mathbf{a}(t)-\tilde{\mathbf{a}}) + \gamma \sum_{j=1}^d (\mathbf{u}_j(t)^\top M\,\mathbf{u}_j(t))\,dt + P(\mathbf{a}(T))\,, \quad (4.48)$$

subject to (4.45), where final cost is given by

$$P(\mathbf{a}(T)) = \eta\,(\mathbf{a}(T)-\tilde{\mathbf{a}})^\top M\,(\mathbf{a}(T)-\tilde{\mathbf{a}})\,. \quad (4.49)$$

For the sake of simplicity, we will drop the time dependence and denote $\mathbf{a} = \mathbf{a}(t),\ \mathbf{u} = \mathbf{u}(t)$. Writing the system dynamics in (4.45) in control-affine form as

$$\dot{\mathbf{a}} = f(\mathbf{a}) + \sum_{j=1}^d g_j(\mathbf{a})\,\mathbf{u}_j \quad (4.50)$$

$$f(\mathbf{a}) := M^{-1}\Big(-(\beta^{-1}K+C)\,\mathbf{a} - bil(\mathbf{a})\Big)\,, \quad g_j(\mathbf{a}) := M^{-1}\Big(-D_j(\mathbf{a})\Big)\,, \quad g(\mathbf{a}) := (g_1(\mathbf{a}),\dots,g_d(\mathbf{a})), \quad (4.51)$$

$$[D_j]_{k,m}(\mathbf{a}) := \sum_{n=1}^L a_n \int_\Omega \psi_n \frac{\partial \psi_m}{\partial x_j}\psi_k\,dx\,, \qquad k,m = 1,\dots,L, \quad (4.52)$$

we define the Hamiltonian associated to the optimal control problem

$$\mathcal{H}(\mathbf{a},\mathbf{u},\mathbf{p}) := \mathbf{p}^\top\Big(f(\mathbf{a}) + g(\mathbf{a})\,\mathbf{u}\Big) + \mathcal{L}(\mathbf{a},\mathbf{u})\,, \quad (4.53)$$

with adjoint variable $\mathbf{p}\in\mathbb{R}^L$ and running cost

$$\mathcal{L}(\mathbf{a},\mathbf{u}) := (\mathbf{a}-\tilde{\mathbf{a}})^\top M\,(\mathbf{a}-\tilde{\mathbf{a}}) + \gamma\sum_{j=1}^d (\mathbf{u}_j^\top M\,\mathbf{u}_j)\,. \quad (4.54)$$

Analogously as for the continuous problem, first-order necessary optimality conditions for the semi-discrete problem (4.48) are then expressed as

$$\dot{\mathbf{a}} = \frac{\partial\mathcal{H}}{\partial\mathbf{p}}(\mathbf{a},\mathbf{u},\mathbf{p})\,, \quad -\dot{\mathbf{p}} = \frac{\partial\mathcal{H}}{\partial\mathbf{a}}(\mathbf{a},\mathbf{u},\mathbf{p})\,, \quad \frac{\partial\mathcal{H}}{\partial\mathbf{u}}(\mathbf{a},\mathbf{u},\mathbf{p}) = 0\,, \quad (4.55)$$

closed with initial and terminal boundary conditions. This two-point boundary value problem reads

$$\dot{\mathbf{a}} = M^{-1}\Big(-(\beta^{-1}K + C)\,\mathbf{a} - bil(\mathbf{a}) - d(\mathbf{a},\mathbf{u})\Big) \qquad \text{(state)} \tag{4.56}$$

$$-\dot{\mathbf{p}} = \mathbf{p}^\top M^{-1}\nabla_{\mathbf{a}}\Big(-(\beta^{-1}K + C)\,\mathbf{a} - bil(\mathbf{a}) - d(\mathbf{a},\mathbf{u})\Big) + M\,(\mathbf{a} - \tilde{\mathbf{a}}) \qquad \text{(adjoint)} \tag{4.57}$$

$$\mathbf{a}(0) = \hat{\mathbf{a}}\,, \quad \mathbf{p}(T) = \nabla_{\mathbf{a}} P(\mathbf{a}(T)) = 2\eta\,(\mathbf{a}(T) - \tilde{\mathbf{a}})^\top M \qquad \text{(b.c.'s)} \tag{4.58}$$

$$0 = \gamma\,\mathbf{u}_j^\top M - \mathbf{p}^\top g_j(\mathbf{a}), \qquad j = 1,\dots,d \qquad \text{(optimality)} \tag{4.59}$$

The system is addressed via policy iteration, combining the forward-backward integration of the state (4.56) and adjoint (4.57) equations, with a gradient descent algorithm for the update of the control signal. This iterative procedure is presented and discussed in Section 2.1.2.

Remark 4. *(On the consistency of optimality conditions). The continuous optimal control problem* (4.38) *and its discretized counterpart* (4.48) *yield different optimality systems. Establishing consistency between these two approaches is essential for ensuring the reliability and accuracy of our numerical solutions. We provide a detailed proof of this consistency in Appendix A.3.*

4.4.2 Nonlinear Model Predictive Control

Our original goal is the design of a control law for stabilizing the McKean-Vlasov PDE towards an unstable steady state. In general, optimal asymptotic stabilization is addressed by means of infinite horizon optimal control, that is, taking $T = +\infty$. This formulation is not approachable with the proposed methodology as it would translate into an ill-posed two-point boundary value problem with an asymptotic terminal condition as $t \to \infty$.

On the other hand, finite horizon $T < +\infty$ optimal control problems are generally not suitable for stabilizing systems around unstable stationary configurations. Small perturbations or deviations from these points tend to derail the system from the desired convergence and in our case, trajectories will be attracted towards stable steady states. Even if the open-loop optimal control successfully steers the system towards an unstable steady state ρ_∞ within the finite horizon, the system is likely to diverge from that beyond T due to disturbances. In this context, a fundamental aspect is the use of a feedback control law which can continuously steer the dynamics towards the desired configuration based on measurements of the current state of the system, hence robust to perturbations.

Nonlinear Model Predictive Control (MPC) is a technique for synthesizing stabilizing feedback control maps from subsequent solves of the optimality system (4.56)-(4.59) over a moving time frame of reduced horizon $T < +\infty$. The idea is that state dependency of the open loop solution only resides in the first entry in the control signal, which can be registered as feedback w.r.t. the initial condition. This procedure is presented in Algorithm 2 and discussed in Section 2.1.2. The receding horizon nature of MPC allows for subsequent feedback and correction of deviations from the unstable target configuration.

4.5 Numerical Tests

In this section we assess the proposed deflation and control methodology on a test cases related to opinion dynamics. In this context, the uncontrolled dynamics converge to a single cluster (or consensus), and we wish to prevent such behaviour and steering the system towards an unstable multi-cluster or uniform distribution.

The PDE model parameters, such as noise and interaction strengths, are selected to ensure the system operates below the transition point where multiple solutions (both stable and unstable) to the integral equation emerge. For a comprehensive theoretical analysis of these models, including the examination of critical temperatures and interaction strengths, readers are referred to [123, 67] for the Hegselmann-Krause model and in [132] for the two dimensional Von Mises model.

In both the following numerical tests, at a fixed time, the control signal and density are discretized in space as in (4.44) as linear combination of spectral basis functions, here given by the first L Fourier modes. For the time stepping we will use a 4th order Runge Kutta scheme.

4.5.1 Hegselmann-Krause opinion formation model

We consider the controlled McKean–Vlasov equation with $V \equiv 0$ and Hegselmann-Krause interaction potential

$$\nabla W(x,y) := \begin{cases} x-y & \text{if } |x-y| \le r \\ 0 & \text{otherwise} \end{cases} \tag{4.60}$$

where r encodes the radius of interactions. The jump discontinuities in this potential are treated by shifting either the base or the end of the jump by a small factor of $5e-3$.

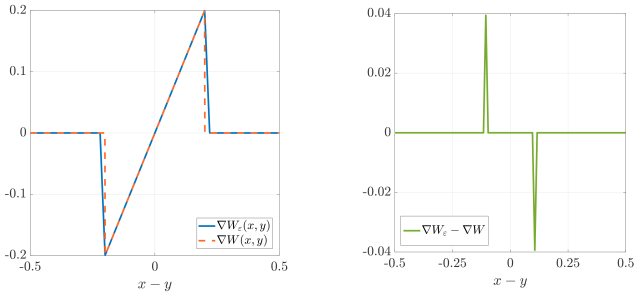


Figure 4.7: The Hegselmann-Krause interaction potential and its regularization (left). Difference between original and regularized potential (right).

In the one-dimensional case with $\Omega = [0,1]$ and periodic boundary conditions, the Mckean-Vlasov equation reads

$$\partial_t \rho = \beta^{-1} \partial_x^2 \rho + \partial_x \left[\rho \left(\int_\Omega (x-y)\rho(y) \mathbb{I}_{|x-y|\le r}(y)\, dy \right) \right], \tag{4.61}$$

where $\mathbb{I}_{|z|\le r}$ denotes the indicator function of the ball of radius R centred at the origin.

The emergent behaviour in this model depends on both the noise amplitude β^{-1} and

the range of interaction r. In Figure 4.8 we fix $r = 0.1$ and we analyse the asymptotic behaviour while changing β^{-1}. The figures have been obtained with $l = 50$ Fourier modes, 200 Gauss points, and deflation parameters $p = 3$, $\xi = 1$. In high noise regimes (similarly to what happens for long interaction ranges), we identify a unique stationary configuration, converging to the uniform distribution as $\beta^{-1} \to 1$. For lower inverse temperature $\beta^{-1} \leq \beta_c^{-1}$, agents' opinions tend to cluster into distinct groups [123]. Once the opinion clusters are formed, they remain relatively stable for a long period of time. This is a manifestation of the dynamical metastability phenomenon that characterizes the dynamics of systems exhibiting discontinuous phase transitions, and it will be studied in detail in forthcoming work. Eventually, the opinion clusters merge, leading to the stable stationary consensus configuration.

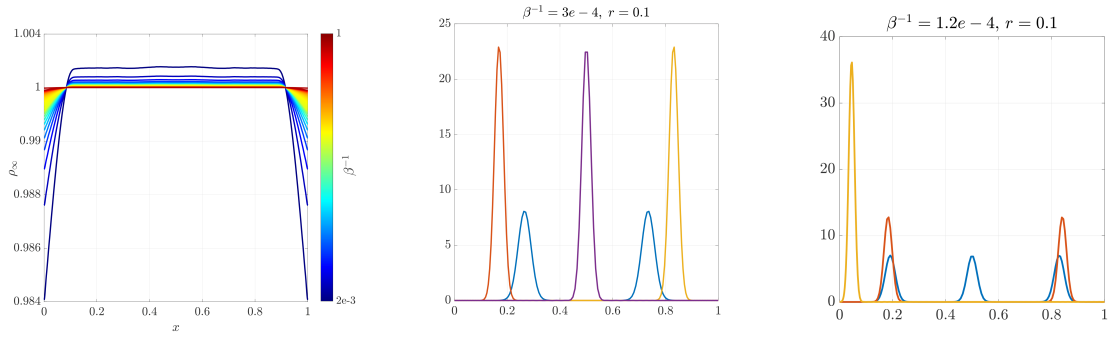


Figure 4.8: In high noise regimes ($\beta^{-1} > \beta_c^{-1} \approx 1e-3$), a single stable steady state converges towards the disordered uniform state (left). For low noise regimes (centre-right), opinions start condensing into clusters. In both parameter regimes, the stable configuration is consensus. However, deflated Newton's method identifies some unstable steady states displaying concentration around multiple positions. All solutions can be found translated along the domain, but only a selection of those is displayed for clarity of presentation.

Having identified the multi-cluster distributions as metastable, we solve the optimal control problem (4.38) for stabilizing the system at this configuration while preventing consensus formation. We consider $\beta^{-1} = 3e-4$, with the stabilization target distribution ρ_∞ being the blue line in Figure 4.8 (centre). We address the stabilization problem with control energy penalization parameter $\gamma = 0.01$, and terminal cost weight $\eta = 100$, over a total time interval $[0, T]$, $T = 1e3$ seconds. The parameters for the gradient descend are a step size $\delta = 10^{-6}$ and convergence tolerance $\varepsilon = 0.01$. In a receding horizon approach,

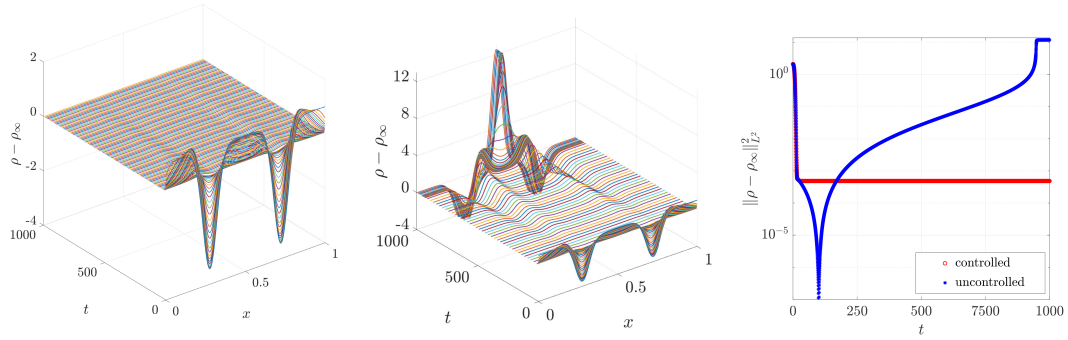


Figure 4.9: We address the optimal control problem for stabilizing the system in (Fig. 4.8 - centre) at the unstable configuration in blue via MPC. (Left) evolution of the discrepancy w.r.t. ρ_∞ in the controlled case, departing from a uniform initial condition. (Centre) the free dynamics diverge from the target state, towards the stable consensus configuration. (Right) discrepancy in L_2 norm between density $\rho(t,x)$ and target steady state $\rho_\infty(x)$.

the control is computed by subsequent solves of (4.59) over a rolling window of reduced time horizon (100s). In Figure 4.9, we compare the controlled and uncontrolled evolution of $\rho(t,x)$, where the MPC action successfully stabilizes the distribution at the desired state, preventing the formation of the concentration profile appearing in the uncontrolled dynamics.

4.5.2 Two-dimensional Von Mises interaction potential

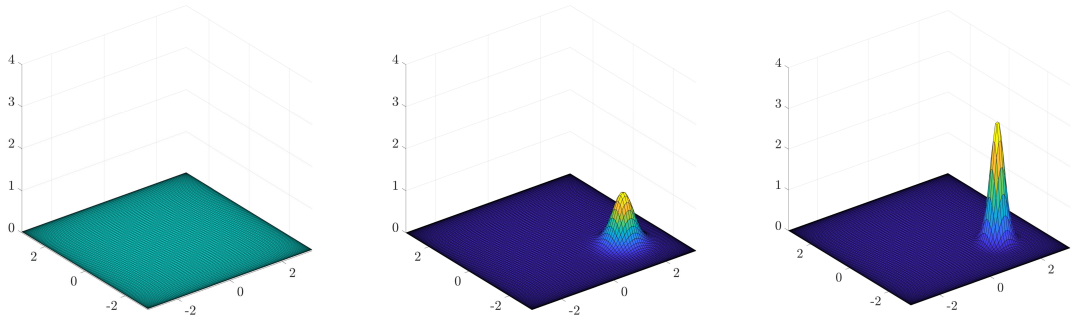
In this numerical test we apply the proposed methodology to a two-dimensional problem, with $\Omega = [-\pi, \pi] \times [-\pi, \pi]$ and Von Mises interaction potential

$$W(x,y) = -I_0(k)^{-2} exp\Big\{\theta(cos(x) + cos(y))\Big\}, \tag{4.62}$$

without any confining force $U(x) \equiv 0$. We use this a "periodized" version of the two-dimensional attractive Gaussian interaction potential that was used in, e.g. [175, 120].[2] For sufficiently small interaction parameter, relative to the size of the domain, this system displays a discontinuous phase transition between a disordered phase, where at high tem-

[2] Alternatively, we could have also considered the wrapped Gaussian. For small variances, the case of interest for us, the wrapped Gaussian, the Gaussian potential and the Von Mises potential give identical numerical results. On the other hand, the rigorous theory put forward in [67] upon which we rely in order to prove the existence of a discontinuous phase transition for the bounded confidence model, applies only to periodic interaction potentials.

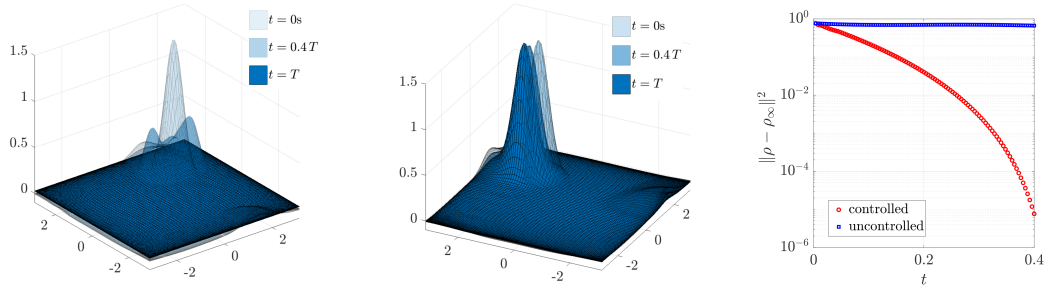
peratures $\beta^{-1} \geq \beta_c$ the stable steady state is the uniform distribution, and a cooperative one, where the uniform configuration is still a stationary state (now unstable), together with a stable, "droplet" concentration profile [67]. In this numerical test we fix $\theta = 1$. In the deflation routine, as initial guess $\mathbf{a}_0$, we consider the one resulting from the iterative scheme presented in [120]. As expected, for large values of β^{-1}, we have a single (uniform) solution, whilst the solutions become two when the inverse temperature is low enough. Figure 4.10 shows the results obtained for $\kappa = 1$, with $l = 10$ Fourier modes and 200 Gauss points.



(a) $\beta^{-1} > 0.3702$ (b) $\beta^{-1} = 0.3701$ (c) $\beta^{-1} = 0.2$

Figure 4.10: Steady states identified with the deflated Newton's method. For $\beta \geq 0.3702$, the iterative scheme converges into a uniform distribution, and the root finding method does not identify other steady states. For smaller values of β^{-1}, we have convergence towards a droplet configuration, whilst the uniform distribution remains an unstable steady state.

We consider an optimal control problem for stabilizing the system towards the uniform distribution in a parameter regime where this is an unstable steady state ($\beta^{-1} = 0.3701$). Following a receding horizon strategy, we solve the problem in $t \in [0, T]$, $T = 1s$ with a rolling window of reduced 0.1s length, for control penalization parameter $\gamma = 0.001$ and terminal cost weight $\eta = 1000$. The parameters for the gradient descend are $\delta = 10^{-3}$ and $\varepsilon = 0.1$. In Figure 4.11 we compare the evolution of $\rho(t, x)$ with and without the control action. In the controlled case, the distribution of particles is successfully steered to the target uniform distribution, whilst the free dynamics move towards the stable concentration profile.



(a) (b) (c)

Figure 4.11: (a) evolution of the controlled dynamics towards ρ_∞ at three sample times. (b) the free dynamics diverge from the target state, forming the stable droplet configuration. (c) discrepancy in L_2 norm between density $\rho(t, x)$ and target steady state $\rho_\infty(x)$.

4.6 Concluding remarks

We have presented a comprehensive methodology for computation and control of unstable steady states for McKean-Vlasov PDEs. The proposed pipeline consists of a spectral discretization of the dynamics, the use of a deflation algorithm for identification of all possible steady states, and a nonlinear model predictive control law for asymptotic stabilization towards the desired unstable state. While unstable steady states do not naturally arise from studying the evolutionary system, there are applications where it might be useful to artificially force the system around an unstable configuration. For example, in opinion dynamics, one may wish to break down clustering in a highly polarized society. In this context, we have demonstrated the effectiveness of our method in multiagent models for opinion dynamics both in one and two dimensions.

Even though in this project we applied our methodology to a relatively simple class of examples, namely weakly interacting diffusions on the torus, for which a well developed theory of phase transitions exists, our approach opens up new possibilities for understanding and controlling the collective behaviour of complex interacting particle systems, with potential impact in various scientific and engineering domains. We are particularly interested in multiagent systems in the social sciences, such as models for consensus formation, pedestrian dynamics and sociodynamics, as well as multispecies models in ecology [124].

Our identification-control pipeline can also be applied to mean field models arising from rational strategies rather than interactions forces, e.g., mean field games with multiple solutions [96], or mean field game models for systemic risk [65]. Moreover, the impact of the proposed methodology expands well beyond the realm of physical multiagent systems to problems in computational statistics, where there is interest in accelerating convergence to the target distribution for Fokker-Planck dynamics in applications related to sampling from probability measures in high dimensions [58, 59, 163].

Chapter 5

Mesoscopic Scale

Control of kinetic collective dynamics by deep neural feedback laws

In the study of large-scale ABMs, when both the number N of agents and the intrinsic dimensionality d of the microscopic dynamics exacerbate the curse of dimensionality, rendering the methodologies discussed in previous chapters insufficient for practical implementation.

To address these limitations, in this chapter we summarise the work published in [9, 10], where we introduce a different perspective that occupies the middle ground between microscopic agent-to-agent interactions and macroscopic mean-field descriptions. This mesoscopic scale serves as a crucial bridge between these approaches by examining the dynamics of smaller agent subsystems rather than focusing exclusively on either individual or collective behaviour.

In this chapter, we consider a modeling paradigm that conceptualizes the agents' density as a plasma-like matter distribution. This approach solves the large-scale OCP by focusing on controlled subsystems, specifically pairs of interacting agents. While we continue to leverage mean-field principles for large-scale approximations, our method introduces an enhanced level of granularity through averaging over controlled binary interactions.

This technique effectively alleviates the computational challenges linked to both the large number of agents N and the high dimensionality d of the state space.

To make matters more concrete, let us consider a second-order system with N agents in $\mathbb{R}^d$, where the state of the i-th agent is encoded by the pair $s_i = (x_i, v_i) \in \mathbb{R}^{2d}$, representing position and velocity, respectively, evolving according to transport-interaction dynamics of the form

$$\begin{aligned} \dot{x}_i(t) &= v_i(t) \\ \dot{v}_i(t) &= \frac{1}{N}\sum_{j=1}^{h} P(x_i(t), x_j(t))(v_j(t) - v_i(t)) + u_i(t)\,, \qquad i = 1, ..., N. \end{aligned} \tag{5.1}$$

Here, $P(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ denotes an interaction kernel, while $u_i(t)$ is a control signal influencing agent i. The ensemble of control signals is denoted by $\mathbf{u}(t) = (u_1(t), \dots, u_N(t))^\top$. The core of the self-organization behaviour of the free dynamics resides in P, which can induce clustering, polarization or alignment, among many others. This self-organization behaviour can be modified by the influence of an external control law. In the framework of dynamic optimization, this control is synthesized by minimizing a cost functional which rewards the convergence of the system towards a cooperative goal, e.g. consensus

$$\tilde{\mathbf{u}}(t) = \underset{\mathbf{u}(\cdot)}{\operatorname{argmin}} \left\{ \mathcal{J}_N(\mathbf{x}, \mathbf{v}, \mathbf{u}) := \frac{1}{N} \int_0^{+\infty} \sum_{i=1}^{h} \|v_i - \bar{v}\|^2 + \gamma\, \mathcal{G}(u_i)\, dt \right\}, \qquad \gamma > 0\,, \tag{OCP}$$

where the first term in the cost is promoting consensus towards the target velocity $\bar{v}$, while $\mathcal{G} : \mathbb{R}^d \to \mathbb{R}_+ \cup \{0\}$ is a convex function penalizing the energy spent by the control u_i.

The double trouble of the curse of dimensionality

The solution of the optimal control problem defined by minimizing (OCP) subject to (5.1) is *twice cursed* by the dimensionality of the problem: solving the OCP becomes prohibitively expensive for large values of N (as in swarm robotics or collective animal behaviour where hundreds or thousands of agents are present), as well as for high-dimensional state spaces (i.e. $d \gg 1$, as in portfolio optimization). As discussed in Chapter 4, a partial

remedy to this problem comes from statistical mechanics by assuming the number of agents $N \to \infty$, so that the dynamics of the individual-based problem can be approximated by a mean field equation

$$\partial_t \rho + v \cdot \nabla_x \rho = -\nabla_v \cdot \Big((\mathcal{P}[\rho] + u)\rho \Big) , \tag{5.2}$$

where $\rho(t, x, v)$ is the probability density of having an agent with state (x, v) at time t, and the mean field interaction force is given by the non-local operator

$$\mathcal{P}[\rho](t, x, v) := \int_{\mathbb{R}^{2d}} P(x, x_*)(v_* - v)\rho(t, x_*, v_*) dx_* \, dv_* \, . \tag{5.3}$$

A direct transcription of the objective (OCP) leads to the mean field optimal control problem [117]

$$\min_u \mathcal{J}(\rho, u) := \int_0^{+\infty} \int_{\mathbb{R}^{2d}} \Big(\|v - \bar{v}\|^2 + \gamma \, \mathcal{G}(u) \Big) \, \rho(t, x, v) \, dx \, dv \, dt \, . \tag{MFOC}$$

Even though the formulation using (5.2) alleviates the curse of dimensionality with respect to N, it leads to a PDE-constrained optimization problem over $2d + 1$ dimensions, which becomes prohibitively expensive already for moderate values of d. The solution of high-dimensional mean field optimal control problems has been addressed using deep learning techniques in [219] in the case of linear dynamics, however, the case with nonlinear agents remains open.

A kinetic remedy

In this chapter, we discuss the synthesis of a feedback control for (5.2) which is a sub-optimal solution to (MFOC). In order to overcome the obstacle posed by the treatment of high-dimensional, nonlinear agents, we resort to modeling the evolution of the agents density ρ from a kinetic viewpoint, reformulating the mean field controlled dynamics as a Povzner-Boltzmann type equation

$$\partial_t \rho(t, x, v) + v \cdot \nabla_x \rho = \lambda_\varepsilon \mathcal{Q}_{\kappa,u}(\rho, \rho)(t, x, v) \, , \tag{5.4}$$

where the operator $\mathcal{Q}$ takes into account the gain and the loss of particles in (x, v) at time t, due to the motion of individuals via free transport and the velocity changes resulting from the controlled interaction dynamics, see [214, 201]. If we consider the interaction dynamics in (5.1), reduced to the case of $N = 2$ particles with velocities (v, v_*), by denoting as (v', v'_*) the update of those velocities after a forward Euler step of length κ, we have

$$\mathcal{Q}_{\kappa,u}(\rho, \rho)(t; x, v) = \int_{\mathbb{R}^{2d}} \frac{1}{\mathcal{J}_\kappa} \rho(t, x, {}' v)\rho(t, x_*, {}' v_*) - \rho(t, x, v)\rho(t, x_*, v_*)\, dx_*\, dv_*\,, \tag{5.5}$$

where $({}'v, {}' v_*) \longmapsto (v, v_*)$ are the pre-interaction velocities that generate the couple (v, v_*), while $\mathcal{J}_\kappa$ is the Jacobian of the binary interactions map $(v, v_*) \longmapsto (v', v'_*)$.

We will show that the alternative mesoscopic description of ABMs via eq. (5.4) is consistent with a sub-optimal, controlled solution of the constrained mean field PDE (5.2), when assuming high frequency λ_ε and weak interactions κ between particles, similarly to the *grazing-collision* limit in kinetic theory [178]. The convenience of such kinetic formulation relies on the reduced computational cost required for its solution via *direct simulation Monte Carlo* (DSMC) methods, [16]. Such a sampling-based technique approximates the solution of the Boltzmann eq.(5.4) by computing a collection of binary sub-problems for couples of agents sampled from the population density function $\rho(t, \cdot)$ [201].

Assuming a high interaction frequency between agents requires, at each evolution step, the solution of a large number of reduced 2-agent (OCP). For efficiency purposes, we rely on supervised learning to build an NN approximating the solution of the binary OCP, thus circumventing its online solution at every sampling instance. For training, we generate synthetic data via the solution of the binary OCP using a discrete time state-dependent Riccati equation (dSDRE) approach [35, 259]. The overall procedure is outlined in Figure 5.1.

Related literature and contributions The modeling of high-dimensional ABMs systems has been extensively studied within the kinetic research community, aiming at reducing the computational complexity of simulations [51, 54, 99]. The formulation of a

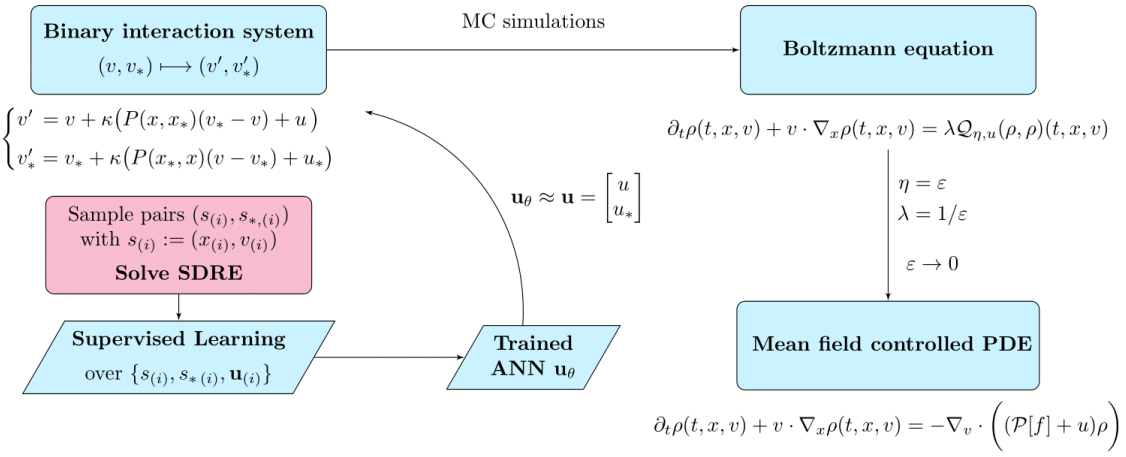


Figure 5.1: A diagram summarizing the main building blocks of the proposed numerical method. The mean field optimal control problem is approximated by a kinetic equation, which relies on sampling interactions between pairs of agents. To reconstruct the complete microscopic dynamical system, these interaction equations are coupled with a free transport process (omitted here for simplicity). Since every sample requires the solution of a binary optimal control problem, we resort to supervised learning techniques to build a feedback control law which can be called in the sampling step without solving online optimization problems. We denote by $s_{(i)} := (x_{(i)}, v_{(i)})$ the state of the $i-$th agent.

kinetic description for evolutionary models has inspired a flourishing literature in ABMs [66, 202, 238, 240]. In this direction, the authors of [17] proposed a Monte Carlo approach based on binary collision dynamics [201], inspired from plasma physics. The proposed methodology addresses the integration of the mean field formulation of ABMs by means of a Boltzmann scheme, allowing to retrieve the mean field evolution of the system as a limit of 2-agents sub-problems. Such complexity reduction has motivated the extension of this numerical scheme to fit the optimal control framework [16, 203, 209].

In [13, 16], consistency was shown between the feedback controlled, non-local, mean field equation (5.2) and the homogeneous non-linear Boltzmann equation (5.4) with controlled binary interactions in the quasi-invariant scaling, and where the control action has been optimized by means of (MPC) 2.1.2, or via dynamic programming 2.1.1. Similar approaches can be found also in [18] for leader-follower multi-agent systems.

The main contributions of our work, inspired by the aforementioned results, can be sum-

marized as follows:

- Aiming at a further reduction in computational complexity with respect to [16, 13], we rely on data-driven approximation models fitted in a supervised learning fashion over synthetic data for the reduced binary OCP. For this, we consider candidate approximation models of both feed-forward (see 2.2.1) and recurrent (see 2.2.2) neural network type. The architecture of such fitted approximation models conveniently allows for batch evaluations of data, meaning that at each time step the controls can be computed for all the sampled 2-agents sub-systems at once. The precision of the fitted models and their efficiency as the dimensionality of the problem increases have been assessed through numerical tests.

- The dataset guiding the ANNs training phase is collected from synthetic data, obtained from the solution of the infinite horizon (OCP) reduced to the binary case. Unlike the approaches taken in [13], we circumvent the solution of the HJB equation associated to the binary OCP, which would be unfeasible even in moderate agent dimensions d. Instead, we rely on the State-Dependent Riccati Equation approach for synthesizing nonlinear feedback controls, as discussed in Section 2.1.3. In particular, we will focus on discrete time settings [106, 71].

- A first, intuitive, choice is to consider the feedback control as target variable of the approximation task, as done in [216, 8, 148, 156, 102]. Furthermore, we also train networks for approximating the whole of the controlled right hand side of the discrete-time binary controlled dynamics. For uncontrolled systems, approximation techniques has been studied for detection and approximation of interaction kernels $P(\cdot,\cdot)$ [55, 170]. However, to our knowledge, this work has not been yet extended to the controlled framework.

Outline of the Chapter

The rest of the chapter has been organized as follows. In Section 5.1 we construct an approximation for the sub-optimal control of the mean field PDE (5.2) from a kinetic

viewpoint. This leads to a Boltzmann description of the dynamics (5.4), which is then proven to converge to its mean field counterpart when performing a proper scaling of the frequency and strength of interactions. In Section 5.2 we discuss how to approximate the evolution of system according to the Boltzmann dynamics for the distribution of agents via Monte Carlo simulation: from the current system configuration, we sample a pool of agents, which are then randomly coupled according to statistics of interaction. The post-interaction agents are then considered, so that their sampling distribution models the updated population density. Following this scheme, the evolution of the system is tracked at discrete times. This requires the computation of the feedback control in discrete-time settings, which is addressed in Section 5.3 by means of a discrete time state-dependent Riccati Equation approach. In Section 5.4, we discuss a supervised learning approximation to the solution of the binary OCP. Numerical tests are presented in Section 5.5.

5.1 A Boltzmann formulation for the control of mean field dynamics

The accurate modeling of self-organization phenomena and optimal control in ABMs requires a large number of interacting individuals, implying the need for the solution of a very high dimensional optimal control problem, which often comes at a prohibitively expensive computational cost. An alternative way to address this problem is to model instead the distribution function $\rho(t, x, v)$ describing the density of individuals having state variable $(x, v) \in \mathbb{R}^d \times \mathbb{R}^d$ at time $t \geq 0$. The evolution of $\rho(t, x, v)$ can be characterized by a kinetic equation accounting for the motion of individuals undergoing pairwise interactions, as modeled in (5.8). Thus, the mean field dynamics can be retrieved by suitable scaling of the interactions, also referred as *quasi-invariant scaling*, or grazing collision limit [201, 247]. In particular, the quasi-invariant limit consists of considering an interaction regime where low intensity interactions occur with high frequency. In this regime, the density $\rho(t, \cdot, \cdot)$ is expected to converge pointwise in time to the solution of the mean field controlled PDE (5.2), which corresponds to a sub-optimal solution of the mean field optimal control problem (MFOC).

5.1.1 Binary controlled dynamics

We denote by $(x, v), (x_*, v_*) \in \mathbb{R}^d \times \mathbb{R}^d$ the position and velocity states of two agents in the population, and we assume that they modify their velocity states according to binary interaction maps $(v, v_*) \mapsto (v', v'_*)$ as follows

$$\begin{aligned} v' &= v + \kappa P(x, x_*)(v_* - v) + \kappa u, \\ v'_* &= v_* + \kappa\, P(x_*, x)(v - v_*) + \kappa u_*\,, \end{aligned} \tag{5.6}$$

where κ is the strength of interaction, $\mathbf{u} = (u, u_*)^\top$ is the forcing term associated to interaction between agents. The goal of such external influence is to steer a couple of agents toward consensus. In particular, we will address the optimal control $\bar{u}$ as the solution of an infinite horizon binary optimal control problem, as follows

$$\tilde{\mathbf{u}} = \underset{\mathbf{u}\in U_{\text{adm}}}{argmin}\Big\{ \int_0^{+\infty} \|v - \bar{v}\|^2 + \|v_* - \bar{v}\|^2 + \gamma \left(\mathcal{G}(u) + \mathcal{G}(u_*)\right) dt \Big\}, \tag{5.7}$$

where $U_{adm} = L^\infty([0, +\infty); \mathbb{R}^{2d})$ is the space of admissible controls. The numerical procedure for the feedback control synthesis will be addressed in section 5.3.

Furthermore, we model the evolution in time of $\rho(t, x, v)$ with a kinetic integro-differential equation of Boltzmann type [13, 200, 214]

$$\partial_t \rho(t, x, v) + v \cdot \nabla_x \rho(t, x, v) = \lambda_\varepsilon \mathcal{Q}_{\kappa,u}(\rho, \rho)(t, x, v)\,, \tag{5.8}$$

where the parameter λ_ε encodes the interaction frequency, and the operator $Q_{\kappa,u}(\rho, \rho)$ accounts for the gain and loss of particles with state (x, v) at time t

$$\mathcal{Q}_{\kappa,u}(\rho, \rho) = \mathcal{Q}^+_{\kappa,u}(\rho, \rho) - \mathcal{Q}^-_{\kappa,u}(\rho, \rho). \tag{5.9}$$

In particular, we can express respectively the gain and loss operators as follows

$$\mathcal{Q}^{+}_{\kappa,u}(\rho,\rho)(t,x,v) = \int\limits_{\mathbb{R}^d\times\mathbb{R}^d} \frac{1}{\mathcal{J}_{\kappa,u}}\rho(t,x,'v)\rho(t,x_*,'v_*)\,dx_*\,dv_*\,, \tag{5.10a}$$

$$\mathcal{Q}^{-}_{\kappa,u}(\rho,\rho)(t,x,v) = \int\limits_{\mathbb{R}^d\times\mathbb{R}^d} \rho(t,x,v)\rho(t,x_*,v_*)\,dx_*\,dv_*\,, \tag{5.10b}$$

where $('v,'v_*) \longmapsto (v,v_*)$ are the pre-interaction velocities that generate the couple (v,v_*), while $\mathcal{J}_{\kappa,u}$ is the Jacobian of the binary interactions map (5.6). To avoid the presence of the jacobian we can introduce a test function $\varphi(x,v) \in C^2_0(\mathbb{R}^d\times\mathbb{R}^d)$ and express the Povzner-Boltzmann operator in the weak form as

$$\langle \mathcal{Q}_{\kappa,u}(\rho,\rho),\varphi\rangle = \lambda_\varepsilon \iint_{\mathbb{R}^{2d}\times\mathbb{R}^{2d}} \left(\varphi(x,v') - \varphi(x,v)\right)\rho\rho_*\,dx_*\,dv_*\,dx\,dv, \tag{5.11}$$

where v' denotes the post-interaction velocity as in (5.6), and and we adopted the shorten notation $\rho = \rho(t,x,v)$, and $\rho_* = \rho(t,x_*,v_*)$.

Remark 5. *Here we considered the unbounded domain in space and velocity, where the operator (5.9) accounts interactions with constant collision frequency λ_ε, analogously to the Boltzmann equation for Maxwell molecules. In general, the interaction frequency is ruled by a non-linear kernel, where in the Povzner-Boltzmann model it reads as follows*

$$\lambda_\varepsilon\langle \mathcal{Q}_{\kappa,u},\varphi\rangle = \iint_{\mathbb{R}^{2d}\times\mathbb{R}^{2d}} \mathcal{P}_B(x,x_*,v,v_*)\left(\varphi(x,v') - \varphi(x,v)\right)\rho\,\rho_*\,dx_*\,dv_*\,dx\,dv, \tag{5.12}$$

where the kernel $\mathcal{P}_B(\cdot)$ in the Povzner approach [214] considers non-local interactions among particles.

If we consider a bounded domain, further conditions should be included in (5.8). Of particular interest is the case, when the velocity space is bounded, e.g. $v,v_*\in\Omega_v\subset\mathbb{R}^d$. In this situation, we can directly impose in (5.12) that the boundary is satisfied introducing a non-linear interaction kernel $\mathcal{P}_B$ such as $\mathcal{P}_B \equiv \chi(v'\in\Omega_v)\chi(v'_*\in\Omega_v)$. This latter choice automatically satisfies the boundaries, however such nonlinear kernel causes major difficulties when we are interested in studying asymptotic properties of the model (5.8), such

as mean field approximations. Alternatively, following the approach proposed in [201], one can ask that the discrete interaction (5.6) preserves the boundary. In this setting, such requirement is possible by properly designing the admissible space of controls U_{adm} in such a way that the boundary cannot be violated. We also observe that the control acts by forcing the state within an admissible position in the bounded domain. Thus, to preserve numerically the bounds a possible strategy is to require a sufficiently small penalty parameter $\gamma > 0$ in the cost functional (5.7), as for example shown in [240], where an explicit form of the control is obtained.

5.1.2 Consistency with the mean field formulation

Here we focus on the consistency of the Boltzmann operator (5.9) with a mean field controlled dynamics of the type (5.2), in particular introducing a *quasi-invariant optimality limit* we can regularize such operator, considering a regime where interactions strength is low and frequency is high. This technique, analogous to the grazing collision limit in plasma physics, has been thoroughly studied in [247] and specifically for first order models in [88, 21], and allows to pass from Boltzmann equation (5.8) to a mean field equation [16, 18].

In what follows we consider the change of notation for the control in the binary dynamics (5.6)

$$u \to u_\kappa(x, v, x_*, v_*), \quad u_* \to u_\kappa(x_*, v_*, x, v)$$

to give explicit dependence on the parameters and the state variables, since we focus on feedback type controls, and we introduce the following assumptions

(*i*) the system (5.6) constitutes an invertible changes of variables from (v, w) to (v', w');

(*ii*) there exists an integrable function $u(x, v, x_*, v_*)$ such that the following limit is well defined

$$\lim_{\kappa \to 0} u_\kappa(x, v, x_*, v_*) = u(x, v, x_*, v_*). \tag{5.13}$$

Hence, in accordance with [66] and with [13], where an analogous of this argument is furnished for the controlled dynamics in stochastic settings, we state the following theorem

Theorem 5.1.1:

Consider the Boltzmann-type equation (5.8), with $\kappa, \lambda_\varepsilon > 0$ and the control $u_\kappa \in U_{\text{adm}}$, where U_{adm} is the class the admissible controls. Furthermore, assume the kernel function $F_\kappa(\cdot) \in L^2_{loc}(\mathbb{R}^{2d} \times \mathbb{R}^{2d})$ for all $\kappa > 0$, where

$$F_\kappa(x, v, x_*, v_*) = P(x, x_*)(v_* - v) + u_\kappa(x, v, x_*, v_*),$$

and define the parameter $\varepsilon > 0$ to introduce the quasi-invariant scaling as follows

$$\kappa = \varepsilon, \qquad \lambda_\varepsilon = 1/\varepsilon, \tag{5.14}$$

which links the strength and the frequency of the interactions in the Boltzmann-type equation (5.8).

Thus, if $\rho^\varepsilon(x, v, t)$ is a solution for the scaled equation (5.8), for $\varepsilon \to 0$ $\rho^\varepsilon(t, x, v)$ converges pointwise, up to a subsequence, to $\rho(t, x, v)$ where ρ satisfies the following controlled mean field equation,

$$\partial_t \rho + v \cdot \nabla_x \rho = -\nabla_v \cdot ((\mathcal{P}[\rho] + \mathcal{U}[\rho])\rho) \tag{5.15}$$

with initial data $\rho^0(x, v) = \rho(0, x, v)$ and where $\mathcal{P}[\cdot]$ represents the interaction kernel (5.3) and the control is such that

$$\mathcal{U}[\rho](t, x, v) = \int_{\mathbb{R}^d \times \mathbb{R}^d} u(x, v, x_*, v_*)\rho(t, x_*, v_*)\, dx_*\, dv_* \tag{5.16}$$

where $u(x, v, x_*, v_*)$ is defined as the limiting value in (5.13).

We report the proof of this result in Appendix A.3, as a reformulation of the result proposed in [13].

5.1.3 Full state controlled binary dynamics

A further generalization consists in introducing a binary interaction dynamics where, differently from the previous section, we consider a binary exchange of information for the full states of the agents $s = (x, v)^\top$, $s_* = (x_*, v_*)^\top \in \mathbb{R}^{2d}$. The interacting process defining $(s, s_*) \to (s', s'_*)$ is described by a binary rules of the following type

$$\begin{aligned} s' &= s + \kappa G(s, s_*) + \kappa H u \\ s'_* &= s_* + \kappa G(s_*, s) + \kappa H u_* \end{aligned} \tag{5.17}$$

where the controls are respectively $u = u(s, s_*), u_* = u_*(s, s_*)$, and the operators G and H are defined accordingly to (5.6) as follows

$$G(s, s_*) = \begin{pmatrix} \mathbb{O}_d & \mathbb{I}_d \\ \mathbb{O}_d & -P(x, x_*)\mathbb{I}_d \end{pmatrix} \begin{pmatrix} x \\ v \end{pmatrix} + \begin{pmatrix} \mathbb{O}_d & \mathbb{O}_d \\ \mathbb{O}_d & P(x, x_*)\mathbb{I}_d \end{pmatrix} \begin{pmatrix} x_* \\ v_* \end{pmatrix}, \qquad H = \begin{pmatrix} \mathbb{O}_d \\ \mathbb{I}_d \end{pmatrix}. \tag{5.18}$$

Hence, the kinetic density $\rho(t, s)$ evolves according to a Boltzmann-type model of the following type

$$\partial_t \rho(t, s) = \lambda_\varepsilon \hat{\mathcal{Q}}_{\kappa,u}(\rho, \rho)(t, s), \tag{5.19}$$

with frequency $\lambda_\varepsilon > 0$, and where, differently from the formulation in (5.8), the transport term is now encoded in the interaction operator $\hat{Q}(\rho, \rho)(t, s)$. The interaction operator $\hat{Q}(\rho, \rho)(t, s)$ is now defined as follows

$$\hat{\mathcal{Q}}_{\kappa,u}(\rho, \rho)(t, s) = \int_{\mathbb{R}^{2d}} \left(\frac{1}{\hat{\mathcal{J}}_{\kappa,u}} \rho(t,' s)\rho(t,' s_*) - \rho(t, s)\rho(t, s_*) \right) ds, \tag{5.20}$$

where $\hat{\mathcal{J}}_{\kappa,u}$ is the jacobian of the binary rule (5.17), with $('s,' s_*)$ the precollisional states.

In this framework, we can retrieve the consistency with mean field model (5.15) via The-

orem 5.1.2, considering now as a kernel function the following

$$F_\kappa(s, s_*) = \kappa G(s, s_*) + \kappa H u(s, s_*), \tag{5.21}$$

and assuming $F_\kappa(\cdot) \in L^2_{loc}(\mathbb{R}^{2d} \times \mathbb{R}^{2d}; \mathbb{R}^{2d})$. Then, in the quasi-invariant scaling (5.14), and taking the limit $\varepsilon \to 0$ we have pointwise convergence of the kinetic model (5.19) with the following mean field equation

$$\partial_t \rho(t, s) = -\nabla_s \cdot \left(\rho(t, s) \int_{\mathbb{R}^{2d}} \left(G(s, s_*) + H u(s, s_*) \right) \rho(t, s_*) \, ds_* \right). \tag{5.22}$$

Recalling that $\nabla_s = (\nabla_x, \nabla_v)^\top$ and that the operators G, H are defined as in (5.18), we have that (5.22) is equivalent to the mean field equation (5.15).

This result follows the same steps as the proof of Theorem (5.1.2), in this regard further details can be found in A.3.

We stress that the main differences, with respect to the results of the previous section, consist in treating the transport term as part of the interaction dynamics, and requiring that (5.21) is L^2_{loc}. This last requirement, useful for the validity of the consistency Theorem 5.1.2, is in general more restrictive. Nonetheless the full state binary interaction (5.17), in a reformulated version, allow to provide an efficient control synthesis for the second-order dynamics. This aspect will be discussed in more details in Section 5.3.

5.2 Asymptotic Monte Carlo methods for constrained mean field dynamics

We provide a fast simulation numerical scheme for the Povzner-Boltzmann-type controlled equation, in the asymptotic regime, reminiscent of Direct Simulaton Monte Carlo methods (DSMCs), used in plasma physics, [54, 51, 17] and later adapted to collective dynamics [17, 237].

First of all, considering a splitting of the transport and collisional term of the Boltzmann-

type equation (5.8) in the asymptotic regime in two different steps [17]:

$$\begin{cases} \partial_t \rho = -v \cdot \nabla_x \rho & \textit{transport} \\ \partial_t \rho = \varepsilon^{-1}\, \mathcal{Q}_{\kappa,u}(\rho,\rho). & \textit{interaction} \end{cases} \tag{5.23}$$

The purpose of the splitting scheme (5.23) is to focus the discussion on the convergence of the collisional term, since the free transport process coincides with the mean field formulation and the Boltzmann one.

In the previous section, consistency between the time evolution of the agents' population has been proven between the mean field model for the population density function, and a scaled Boltzmann description for the dynamics. The latter modeling of $\rho(t,x,v)$ is guided by the reduced microscopic binary interactions (5.6) between agents. This motivates the resorting to Monte Carlo simulation techniques for the approximation of the population density function under the Boltzmann formulation [201, 200, 17, 13].

The evolution (5.23) of ρ in $[0,T]$ can be modeled by means of a Forward Euler method, for which we define a time step Δt, and discrete times $t_h = h\Delta t$ for $h = 0,\dots,N_T$

$$\rho(t_{h+1},x,v) = \rho(t_h,x,v) + \Delta t\, \partial_t \rho(t_h,x,v)\,. \tag{5.24}$$

In the spirit of DSMC methods, we can design a stochastic simulation scheme where we sample N_s agents from ρ^0 and we approximate the time variation of $\partial_t f$ as the evolution of the sampled distribution according to the post transport/interaction states.

Concerning the free transport process, the time evolution amounts to the exact free flow at time t_h of sample particles $\left\{(x_i^h, v_i^h)\right\}_{i=1}^{N_s}$ and evolves as follows

$$x_i^{h+1} = x_i^h + \Delta t v_i^h \qquad i = 1,\dots,N_s. \tag{5.25}$$

The collisional term in the Boltzmann-type equation (5.23), can be rewritten in dependence

of the gain and loss components of the operator $\mathcal{Q}_{\varepsilon,u}$:

$$\partial_t \rho(t,x,v) = \frac{1}{\varepsilon}\mathcal{Q}^+_{\varepsilon,u}(\rho,\rho)(t,x,v) - Z\rho(t,x,v)\,, \tag{5.26}$$

where $\rho > 0$ represent the total mass

$$Z = \int_{\mathbb{R}^{2d}} \rho(t,x_*,v_*)\,dx_*\,dv_*\,. \tag{5.27}$$

Assuming ρ to be a probability density function we will consider $Z = 1$. The Forward Euler scheme for (5.26), with the notation $\rho^h = \rho(t_h,x,v)$ reads

$$\rho^{h+1} = \left(1 - \frac{\Delta t}{\varepsilon}\right)\rho^h + \frac{\Delta t}{\varepsilon}\,\mathcal{Q}^+_{\varepsilon,u}(\rho^h,\rho^h)\,, \tag{5.28}$$

since ρ^h is a probability density, thanks to mass conservation $\mathcal{Q}^+_{\varepsilon,u}$ is again probability density function. Moreover, under the restriction $\Delta t \le \varepsilon$, also ρ^{h+1} is also a probability density. Equation (5.28) can be interpret as follows: an agent with state (x,v) has probability $(1-\Delta t/\varepsilon) \in [0,1]$ to avoid collision with other agents in each time interval $[t, t+\Delta t]$. When, instead, the collision does happen (event with probability $\Delta t/\varepsilon \in [0,1]$), the evolution follows the interaction law $\mathcal{Q}^+_{\varepsilon,u}$ described by the scaled binary interaction, then a sampled pair of agents at time t_h

$$(x,v) := (x_i^h, v_i^h), \qquad (x_*,v_*) := (x_j^h, v_j^h)$$

evolves according to (5.6) as follows

$$\begin{aligned} v_i^{h+1} &= v_i^h + \varepsilon\, P\left(x_i^h, x_j^h\right)\left(v_j^h - v_i^h\right) + \varepsilon\, u\left(x_i^h, v_i^h, x_j^h, v_j^h\right), \\ v_j^{h+1} &= v_j^h + \varepsilon\, P\left(x_j^h, x_i^h\right)\left(v_i^h - v_j^h\right) + \varepsilon\, u\left(x_j^h, v_j^h, x_i^h, v_i^h\right). \end{aligned} \tag{5.29}$$

In what follows we consider the asymptotic regime with $\Delta t = \varepsilon$, where the binary interactions (5.29) are equivalent to a Forward Euler scheme for the two agent dynamics, and the particles that interacts at each iteration are maximized according to the scheme (5.28).

The choice of ε is of paramount importance to approximate consistently the mean-field model (5.15) according to Theorem 5.1.2. Nonetheless, there is a trade-off between the number of samples N_s and the size of ε, indeed it is possible to see that below a certain threshold there is no further improvement in approaching the mean field model. We refer to [17] for further discussion in this direction.

The asymptotic Monte-Carlo algorithm can be formalized as the procedure outlined in Algorithm 8, which is comparable to Algorithm 1 in [13]. This approach differs from the conventional Nanbu scheme, which was originally developed for free particle systems lacking external control inputs.

Up to this moment, the discussion overlooked the derivation of the forcing terms u in the interaction (5.29). These control variable is meant to be of feedback type, as they only depend on the current agents' states. Nevertheless, the discrete-time nature of the interaction map (5.29), embeds within the control variable a dependency on the time-step Δt, which in turn is related to the parameter ε, see for example [13, 16]. In the following section, we address the solution of the discrete time formulation (5.25)(5.29) of the binary interaction control problem.

5.3 Infinite horizon optimal control of binary dynamics

In this section we study the solution of the infinite horizon optimal control problem for the reduced 2-agent system.

We begin by recalling some fundamental notions on optimal control for discrete-time systems [185], to then present its numerical approximation using a discrete-time State-Dependent Riccati Equation approach. We highlight that this formulation has fundamental differences w.r.t. its continuous version (presented in section 2.1.3).

We reformulate (5.29) in a general semi-linear control-affine discrete time systems of the form:

$$\mathbf{z}(t_{h+1}) = f(\mathbf{z}(t_h)) + g(\mathbf{z}(t_h))\mathbf{u}(t_h)\,, \tag{5.30}$$

where $\mathbf{z}(t_h) \in \mathbb{R}^n$ and $\mathbf{u}(t_h) \in \mathbb{R}^m$ denote the state of the system and the control signal at time $t_h = h\Delta t$, such that $\mathbf{z}(t_h) := (s^h, s^h_*)^\top \in \mathbb{R}^n$, with $n = d$ or $2d$ depending on whether the agent dynamics are of first or second order, respectively. The control vector is defined as $\mathbf{u}(t_h) := (u^h, u^h_*)^\top \in \mathbb{R}^m$, with $m = 2d$.

The free dynamics map $f : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ and the control operator $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are assumed to be $C^1(\mathbb{R}^n)$, satisfying $f(\mathbb{O}_n) = \mathbb{O}_{n\times n}$ and $g(\mathbb{O}_n) = \mathbb{O}_{n\times m}$. Note that binary systems of the form (5.29) fit this setting. Given $Q \in \mathbb{R}^{n\times n}$, $Q \succeq 0$ and $R \in \mathbb{R}^{m\times m}$, $R \succ 0$, we are interested in the infinite horizon optimal control problem

$$\min_{\mathbf{u}(\cdot)} \mathcal{J}(\mathbf{u};\mathbf{s}) := \sum_{n=0}^{+\infty} \mathbf{z}(t_h)^\top Q\mathbf{z}(t_h) \,+\, \mathbf{u}(t_h)^\top R\mathbf{u}(t_h)\,, \tag{5.31}$$

subject to the dynamics (5.30) with initial state $\mathbf{z}(t_0) = \mathbf{s}$. We look for a solution $\tilde{\mathbf{u}}$ to (5.31) in feedback form, that is, an optimal control map $\tilde{\mathbf{u}} : \mathbb{R}^n \to \mathbb{R}^m$ which is expressed a function of the state, $\tilde{\mathbf{u}}(t_h) = \tilde{\mathbf{u}}(\mathbf{z}(t_h))$. The computation of an optimal feedback law follows a dynamic programming argument, for which we define $V(\mathbf{s}) : \mathbb{R}^n \to \mathbb{R}$ as the optimal cost-to-go departing from $\mathbf{s}$:

$$V(\mathbf{s}) = \min_{\mathbf{u}(\cdot)} \mathcal{J}(\mathbf{u};\mathbf{s})\,, \tag{5.32}$$

where V satisfies the Bellman equation

$$V(\mathbf{s}) = \min_{\mathbf{u}\in\mathbb{R}^m} \Big(\mathbf{s}^\top Q\mathbf{s} + \mathbf{u}^\top R\mathbf{u} + V\big(f(\mathbf{s}) + g(\mathbf{s})\mathbf{u}\big) \Big)\,, \qquad \text{for all } \ \mathbf{s} \in \mathbb{R}^n\,. \tag{5.33}$$

From this point onwards, as we work globally in the state space, $\mathbf{z}$ and $\mathbf{s}$ are treated indistinctly.

In order to better describe the difficulties related to the solution of this optimal control problem, we first focus on the linear quadratic case, where the optimal solution is computed via the discrete-time linear quadratic regulator (LQR). The optimality of the feedback law is ensured by a direct link between the LQR solution and the dynamic programming one, obtained from equation (5.33). In non-linear settings, the parallelism between the

nonlinear QR and DP is broken, together with the optimality of the solution.

5.3.1 Discrete-time Linear Quadratic Regulator

The discrete time linear quadratic problem is a particular instance of the optimal control problem (5.31) when $f(\mathbf{s}) \equiv A \in \mathbb{R}^{n\times n}$, and $g(\mathbf{s}) \equiv B \in \mathbb{R}^{n\times m}$. Under these assumptions, we make the ansatz $V(\mathbf{s}) = \mathbf{s}^\top \Pi \mathbf{s}$, so that the Bellman equation (5.33) becomes

$$\mathbf{s}^\top \Pi\, \mathbf{s} = \min_{\mathbf{u}\in\mathbb{R}^m} \Big(\mathbf{s}^\top Q\, \mathbf{s} + \mathbf{u}^\top R\, \mathbf{u} + (A\mathbf{s} + B\mathbf{u})^\top \Pi\, (A\mathbf{s} + B\mathbf{u}) \Big) \qquad \text{for all } \ \mathbf{s} \in \mathbb{R}^n\,. \tag{5.34}$$

Solving the equation above leads to an optimal feedback of the form

$$\tilde{\mathbf{u}} = \tilde{\mathbf{u}}(\mathbf{s}) = -(R + B^\top \Pi\, B)^{-1} B^\top \Pi\, A\mathbf{s}\,, \tag{5.35}$$

where $\Pi \in \mathbb{R}^{n\times n}$ is the unique positive definite solution of the discrete-time algebraic Riccati equation (DARE):

$$\Pi = Q + A^\top \Pi\, A - A^\top \Pi\, B(R + B^\top \Pi\, B)^{-1} B^\top \Pi\, A\,. \tag{5.36}$$

Remark 6. *Consider a discrete first-order binary system as in (5.29) with constant interaction kernel $P(x, x_*) = p$, with individual states $v, v_* \in \mathbb{R}^d$, target $\tilde{v} = 0$ and $\varepsilon = \Delta t$. In this case, the optimal control problem (5.31) reads*

$$\min_{\mathbf{u}(\cdot)} \ \sum_{n=0}^{+\infty} \|v^h\|^2 + \|v_*^h\|^2 + \gamma\, (\|u^h\|^2 + \|u_*^h\|^2) \tag{5.37}$$

$$s.t. \quad \begin{cases} v^{h+1} &= v^h \ + \Delta t(p(v_*^h - v^h) + u^h) \\ v_*^{h+1} &= v_*^h + \Delta t(p(v^h - v_*^h) + u_*^h)\,. \end{cases} \tag{5.38}$$

This formulation corresponds to a linear-quadratic control problem by setting $\mathbf{s} = (v, v_*)^\top$,

$\mathbf{u} = (u, u_*)^\top$ *and*

$$A = \mathbb{I}_{2d} + \Delta t \tilde{A}\,, \quad \tilde{A} = \begin{pmatrix} -p & p \\ p & -p \end{pmatrix} \otimes \mathbb{I}_d, \qquad B = \Delta t \mathbb{I}_{2d} \qquad Q = \Delta t \mathbb{I}_{2d}, \qquad R = \Delta t \gamma \mathbb{I}_{2d}.$$

Similarly as in Theorem 5.1.2, we are interested in the asymptotic limit for $\Delta t \to 0$*, in which case* (5.36) *becomes*

$$2\tilde{A}\Pi - \gamma \Pi^2 + \mathbb{I}_{2d} = \mathbb{O}_{2d}. \tag{5.39}$$

Exploiting to the symmetric structure of the two agent interaction, Π *can be reduced to diagonal and off-diagonal components, see e.g. [139, 15], obtaining the following*

$$\Pi = \begin{pmatrix} \pi_D & \pi_O \\ \pi_O & \pi_D \end{pmatrix} \otimes \mathbb{I}_d\,, \qquad \pi_D = \frac{1}{2\gamma}\left(\sqrt{\gamma} - 2p + \sqrt{4p^2 + \gamma}\right)\,, \quad \pi_O = \frac{1}{\sqrt{\gamma}} - \pi_D\,. \tag{5.40}$$

The limiting optimal feedback law $\tilde{\mathbf{u}}(\mathbf{s}) = (\tilde{u}, \tilde{u}_*)$ *is given by*

$$\tilde{\mathbf{u}}(\mathbf{s}) = -\frac{1}{\gamma}\Pi \mathbf{s} = -\frac{1}{\gamma}\begin{pmatrix} \pi_D v + \pi_O v_* \\ \pi_O v + \pi_D v_* \end{pmatrix}\,, \tag{5.41}$$

from where it follows that the individual feedback laws are given by

$$\tilde{u}(v, v_*) = -\frac{1}{\gamma}(\pi_D v + \pi_O v_*) = \tilde{u}_*(v_*, v).$$

In the limiting kinetic equation this corresponds to the mean field control model

$$\partial_t \rho + v \cdot \nabla_x \rho = -\nabla_v \cdot \left(\int_{\mathbb{R}^{2d}} (p(v_* - v) - \frac{1}{\gamma}(\pi_D v + \pi_O v_*))\rho(t, x_*, v_*)\, dx_*\, dv_* \right)\,. \tag{5.42}$$

Notice that the structure of $\Pi \in \mathbb{R}^{2d \times 2d}$ *in* (5.40) *implies isotropic action of the control in the d-dimensional state space.*

5.3.2 Discrete-time Non-Linear Quadratic Regulator

Similarly to the continuous case (2.1.3), it is feasible to apply LQR to a linearization of the system (5.38). However, if we want to compute a law accounting for nonlinearities, then we compromise the convenient connection between the DARE and the Bellman equation. Moreover, the computational cost required for the solution consistently increases.

In order to extend LQR to the nonlinear case, we begin by re-arranging the difference equation (5.30) in semilinear state-dependent form

$$\mathbf{z}(t_{h+1}) = \mathcal{A}(\mathbf{z}(t_h))\mathbf{z}(t_h) + \mathcal{B}(\mathbf{z}(t_h))\mathbf{u}(t_h)\,, \tag{5.43}$$

where $f(\mathbf{z}) = \mathcal{A}(\mathbf{z})\mathbf{z}$ and $g(\mathbf{z}) = \mathcal{B}(\mathbf{z})$. Note that this semi-linearization is not unique for systems of order greater than 1. In what follows, we further assume pointwise controllability, i.e. $\forall \mathbf{z} \in \Omega$ the pair $(\mathcal{A}(\mathbf{z}), \mathcal{B}(\mathbf{z}))$ is controllable.

Similarly to the LQR design, the feedback control policy can be calculated as

$$\begin{aligned} \tilde{\mathbf{u}}(\mathbf{z}(t_h)) &= -\mathbf{z})\mathbf{z} \\ \mathcal{K}(\mathbf{z}) &= (\mathcal{B}(\mathbf{z})^\top \Pi(\mathbf{z})\mathcal{B}(\mathbf{z}) + R)^{-1}\mathcal{B}(\mathbf{z})^\top \Pi(\mathbf{z})\mathcal{A}(\mathbf{z}) \end{aligned} \tag{5.44}$$

where the argument $\mathbf{z}$ for the operators denotes dependency of the current state $\mathbf{z}(t_h)$ with discrete time scale Δt, and $\Pi(\mathbf{z})$ is the solution of a DARE with state-dependent coefficients (DSDRE):

$$\Pi(\mathbf{z}) = Q + \mathcal{A}(\mathbf{z})^\top \Pi(\mathbf{z})\,\mathcal{A}(\mathbf{z}) - \mathcal{A}(\mathbf{z})^\top \Pi(\mathbf{z})\,\mathcal{B}(\mathbf{z})(R + \mathcal{B}(\mathbf{z})^\top \Pi(\mathbf{z})\,\mathcal{B}(\mathbf{z}))^{-1}\mathcal{B}(\mathbf{z})^\top \Pi(\mathbf{z})\,\mathcal{A}(\mathbf{z})\,. \tag{5.45}$$

The state dependency of the feedback operator $\mathcal{K}(\mathbf{z}) = \mathcal{K}(\mathbf{z}(t_h))$ suggests the need for consecutive sequential solutions $\Pi(\mathbf{z}) = \Pi(\mathbf{z}(t_h))$ of (5.45) at every discrete time t_h along a trajectory. Thus, for high dimensions of the state space $\Omega \subset \mathbb{R}^h$, the exact solution of the DSDRE (5.45) comes at a cumbersome computational cost. Between the several numerical approaches that have been proposed to address this, we cite Taylor series method [26] and interpolation for $\Pi(\mathbf{z})$ [35], while we refer the interested reader to [189] for a more

exhaustive review. In what follows, we rely on a discrete version of the SDRE approach proposed in [35].

5.3.3 Discrete-time SDRE approach

We aim at circumventing the computational challenge of solving (5.45) at each time step, by realizing the DSDRE feedback law in a model predictive control fashion: given the current state $\mathbf{z}(t_h)$ of the system, we assume the operator $\Pi(\mathbf{z})$ to be a positive definite matrix in $\Pi \in \mathbb{R}^{n\times n}$, meaning that (5.45) reduces to its algebraic form (5.36), where the state dependencies in (5.45) are neglected by accordingly freezing all the operators at the current configuration. The resulting feedback variable leads to a suboptimal approximation of the controlled trajectory between t_h and t_{h+1}, after which the procedure is repeated by freezing the system at the updated state $\mathbf{z}(t_{h+1})$.

Algorithm 5: MPC-DSDRE approach

$\mathbf{z}(0) \leftarrow \mathbf{s}$; `initial condition`

for $h = 0, ..., N_T$ **do**

- $\mathbf{s} \leftarrow \mathbf{z}(t_h)$; `read current state`
- $A \leftarrow \mathcal{A}(\mathbf{s}),\ \ B \leftarrow \mathcal{B}(\mathbf{s})$; `freeze the system accordingly`
- solve (5.36) for Π;
- $\mathbf{u} \leftarrow -(R + B^\top \Pi B)^{-1} B^\top \Pi A\mathbf{s}$; `approximated feedback control`
- $\mathbf{z}(t_{h+1}) \leftarrow \mathbf{s} + \Delta t(A\mathbf{s} + B\mathbf{u})$; `the control system evolves for` Δt

The main computational bottleneck still persisting with this approach is the availability of a sufficiently fast solver for (5.36). Efficiency in retrieving the DARE solution is key in the settings under consideration. The consistency between the mean field and the kinetic dynamics only holds for $\varepsilon \ll 1$ which in turn is associated with high frequency sampling, i.e. large numbers of couple of interacting particles in the Monte Carlo simulation. At each time step, we aim at conveniently approximate the controlled post-interaction states for a large number of paired agents, living in an arbitrarily high dimensional state space

$\Omega \subset \mathbb{R}^h$. We address this task by means of supervised learning approximation, relying on models within the family of Artificial Neural Networks.

5.4 Synthetic data generation for Neural Network feedback laws

The computational method presented in this chapter models the time evolution of the population density $\rho(t, x, v)$ according to the Forward Euler scheme (5.28), where the updated density is to be approximated by the sampling distribution of *controlled* post-interaction states in a Monte Carlo fashion.

In this section we will discuss supervised learning of feedback laws for allowing efficient high frequency MC sampling. In particular, we will focus on neural network models of feedforward or recurrent type, as previously introduced in Section 2.2.

Both the training and hyper-parameter tuning phases for the neural network approximation rely on the availability of datasets containing input data points coupled with the corresponding target function evaluations. Since our goal is to approximate the feedback law for the reduced binary controlled dynamics, a straightforward approach is to treat the feedback control itself as the output variable in our approximation. As an alternative, we also train networks for approximating the whole of the controlled right hand side of the discrete-time binary controlled dynamics. In Section 5.5, we compare the two approaches in two numerical tests, in terms of precision and computational time. In both cases, we consider the MSE as loss function.

To create training data and establish a benchmark for precision, we solve the optimal control problem using the discrete-time SDRE approach. The difference equations for the

binary system at the discrete time t_h hold as follows:

$$\begin{cases} x^{h+1} = x^h + \Delta t\, v^h \\ x_*^{h+1} = x_*^h + \Delta t\, v_*^h \end{cases} \qquad \begin{cases} v^{h+1} = v^h + \Delta t[P(x^h, x_*^h)(v_*^h - v^h) + u(x^h, x_*^h, v^h, v_*^h)] \\ v_*^{h+1} = v_*^h + \Delta t[P(x_*^h, x^h)(v^h - v_*^h) + u_*(x^h, x_*^h, v^h, v_*^h)] \end{cases} \tag{5.46}$$

being $(x, v), (x_*, v_*) \in \Omega \subset \mathbb{R}^{2d}$ the states for a couple of interacting agents. We introduce the notation $\mathbf{s} = (x, x_*, v, v_*)^\top \in \mathbb{R}^{4d}$, $\mathbf{u} = (u, u_*)^\top \in \mathbb{R}^{2d}$, for which the difference equation reads as in (5.30)

$$\mathbf{s}^{h+1} = f(\mathbf{s}^h) + g(\mathbf{s}^h)\mathbf{u}^h\,, \tag{5.47}$$

for $f(\mathbf{s}) = (\mathbb{I}_{4d} + \Delta t\, G(\mathbf{s}))\mathbf{s}$ and $g(\mathbf{s}) = \Delta t\, H(\mathbf{s})$, with G and H defined as in (5.18). From this, comes the Δt-dependency of the feedback law $\mathbf{u}$: the DSDRE parameters $\mathcal{A}$, $\mathcal{B}$ are dependent of the time-step, as will be the DARE solution associated to the system frozen at the current configuration.

Algorithm 6: Synthetic data generation

$\{\mathbf{s}_{(i)}, \Delta t_{(i)}\}_{i=1}^{N_s}$ *i.i.d.*; — N_s `uniform samples in` $\Omega^2 \times [0, T]$
for $i = 1, ..., N_s$ **do**
  $A \leftarrow \mathbb{I}_{4d} + \Delta t_{(i)}\, G(\mathbf{s}_{(i)})$;
  $B \leftarrow \Delta t_{(i)}\, H(\mathbf{s}_{(i)})$; — G, H `as in` (5.18)
  solve (5.36) for Π;
  $\mathbf{u}_{(i)} \leftarrow -(R + B^\top \Pi B)^{-1} B^\top \Pi A \mathbf{s}_{(i)}$; — `feedback control`
  $\mathbf{s}'_{(i)} \leftarrow \mathbf{s}_{(i)} + \Delta t_{(i)}(A\mathbf{s}_{(i)} + B\mathbf{u}_{(i)})$; — `post-interaction state`
$\mathcal{T}_\mathbf{u} = \{\mathbf{s}_{(i)}, \Delta t_{(i)}, \mathbf{u}_{(i)}\}_{i=1}^{N_s}$; — `dataset for` $\mathbf{u}_\theta$
$\mathcal{T}_{\mathbf{s}'} = \{\mathbf{s}_{(i)}, \Delta t_{(i)}, \mathbf{s}'_{(i)}\}_{i=1}^{N_s}$; — `dataset for` $\mathbf{s}'_\theta$

The generation of datasets $\mathcal{T}_\mathbf{u}$, $\mathcal{T}_{\mathbf{s}'}$ for the supervised learning approximation of the feedback law $\mathbf{u}_\theta$, and post-interaction states $\mathbf{s}'_\theta$ is summarized in Algorithm 6. The number N_s of samples in the datasets is discussed in Section 5.5 specifically for each numerical test, whilst the reliability and generalizability of the sampled data are ensured through cross validation of the trained models over an unseen *test set* $\mathcal{T}_t \nsubseteq \mathcal{T} \cup \mathcal{T}_v$ also generated

through Algorithm 6. The addition of the time step Δt to the models' input allows to train the networks to provide accurate approximations even when applied to adaptive time-step integration techniques.

Remark 7. *As the controlled density evolution is carried by a Monte Carlo type method, the computational efficiency of evaluating a batch of controlled particle interactions is crucial. Our choice of control/state architectures is motivated by its favorable computational complexity. The evaluation cost of a single point through FNNs is $O(n)$, n being the number of neurons in the network. Furthermore, in our non-sequential framework, the evaluation cost of a one-to-one LSTM cell with n neurons is comparable to the one of a FNN with $4n$ neurons. This linear scaling in both architectures ensures efficient batch evaluation, as the computational cost grows proportionally with network complexity. This efficiency is particularly advantageous for large-batch evaluation, associated with $\varepsilon \ll 1$.*

5.5 Numerical Tests

In this section we will assess the proposed methodology over consensus control problems for two high dimensional (both in d and in N) ABMs. We aim at modeling the time evolution of the agents' distribution $\rho(t, x, v)$ via Monte Carlo simulation of the approximated binary post interaction states. In this section we will compare two different approaches: the approximation of the feedback control map $\mathbf{u}_\theta$, acting in the binary system (5.46), and the direct approximation of the controlled dynamics $\mathbf{s}'_\theta$. Since the approximation of the post-interactions positions $\mathbf{s}'_x$ is direct, the neural network model is restricted to post-interaction velocities $\mathbf{s}'_v$, that is $\mathbf{s}'_\theta = (\mathbf{s}'_x, \mathbf{s}'_{v,\theta})^\top$. Algorithm 7 [17] summarizes the proposed numerical procedure for the density evolution once an approximated optimal feedback law $\mathbf{u}_\theta \approx \tilde{\mathbf{u}}$ has been constructed. When a neural network $\mathbf{s}'_\theta$ is built, we replace line 6 in the algorithm with

$$\mathbf{s}_k^{h+1} \leftarrow \mathbf{s}'_\theta(\mathbf{s}_k^h)\,. \tag{5.48}$$

The assumption of constant time-step Δt can be easily relaxed, as both the approximants $\mathbf{u}_\theta$ and $\mathbf{s}'_\theta$ take Δt as input. In the following numerical tests, the NN training has been

Algorithm 7: Monte Carlo simulation for Boltzmann dynamics

$\{(x,v)_i\}_{i=1}^h \sim \rho(t_0,x,v)$ *i.i.d.*; `N samples from the initial distribution`

for $h = 0, ..., T$, $t_h = h \cdot \Delta t$ **do**

  select $\{(i_k, j_k)\}_{k=1}^{N//2}$; `random pairs of agents without repetitions`

  $\{\mathbf{s}_k^h\}_{k=1}^{N//2}$, $\mathbf{s}_k^h = (x_{i_k}^h, x_{j_k}^h, v_{i_k}^h, v_{j_k}^h)$; `N//2 couples of agents`

  for $k = 1, ..., N//2$ **do**

    $s_k^{h+1} \leftarrow \mathcal{A}(\mathbf{s}_k^h)\mathbf{s}_k^h + \mathcal{B}(\mathbf{s}_k^h)\tilde{\mathbf{u}}(\mathbf{s}_k^h, \Delta t)$; `controlled dynamics (5.46)`

    $(x_{i_k}^{h+1}, v_{i_k}^{h+1}, x_{j_k}^{h+1}, v_{j_k}^{h+1}) \leftarrow \mathbf{s}_k^{h+1}$;

done via *Adam* optimizer [151] with a learning rate $\alpha = 0.01$ over batches of 100 samples.

5.5.1 Test 0: Sznajd model

To illustrate the relationship between the control strategy derived at the supervised learning kinetic approximation level and its mean-field counterpart, we consider a preliminary numerical example in a simplified setting. We focus on consensus control for a one-dimensional first-order Sznajd model for opinion dynamics [228]. At the mean-field level, the control problem reads as a first-order version of (MFOC):

$$\begin{aligned} \min_u \mathcal{J}(\rho, u) := \int_0^T \int_\Omega \left(\|x - \bar{x}\|^2 + \gamma \|u\|^2\right) \rho(t, x_*)\, dx\, dt\,, \\ \text{subject to } \ \partial_t \rho + \partial_x \Big[(\mathcal{P}[\rho] + u)\rho \Big] = 0\,, \end{aligned} \tag{5.49}$$

where $\bar{x}$ denotes the target configuration, and the interaction operator is defined as

$$\mathcal{P}[\rho](t,x) = \int_\Omega P(x, x_*)(x_* - x)\rho(t,x) dx_*\,, \qquad P(x, x_*) = \beta(1 - x^2)\,, \tag{5.50}$$

for $\beta \in \mathbb{R}$. In particular, we consider $\Omega = [-1, 1]$, representing the opinion space of a population of voters ranging between two extreme positions represented by $\{-1, 1\}$, and a negative interaction coefficient $\beta = -1$, which induces polarization behavior. The control energy penalization coefficient is fixed to $\gamma = 0.05$.

This simplified framework enables the numerical solution of the mean-field optimal control problem, which would be computationally prohibitive in the higher-dimensional, second-

Model	K	# Neurons	Activations
$\mathbf{u}_\theta^{FNN}$	1	100	$\sigma(\mathbf{z}) = \log(\exp(\mathbf{z}) + 1)$
$\mathbf{u}_\theta^{RNN}$	1	100	$\sigma(\mathbf{z}) = tanh(\mathbf{z})$, $\rho(\mathbf{z}) = (1 + \exp(-\mathbf{z}))^{-1}$
$\mathbf{s}'^{FNN}_{v,\theta}$	2	60	$\sigma(\mathbf{z}) = tanh(\mathbf{z})$
$\mathbf{s}'^{RNN}_{v,\theta}$	1	100	$\sigma(\mathbf{z}) = tanh(\mathbf{z})$, $\rho(\mathbf{z}) = log(exp(\mathbf{z}) + 1)$

Table 5.1: Summary of models and their configurations for Test 0.

order models considered in the subsequent numerical tests.

At the binary interaction level, the problem reads

$$\min_{\mathbf{u}(\cdot)} \frac{\Delta t}{2} \sum_{n=0}^{+\infty} (\|\mathbf{x}^n\|^2 + \gamma \|\mathbf{u}^n\|^2), \text{ s.t. } \begin{cases} x^{n+1} = x^n + \Delta t(\beta(1 - (x^n)^2)(x_* - x) + u^n) \\ x_*^{n+1} = x^n + \Delta t(\beta(1 - (x_*^n)^2)(x - x_*) + u_*^n) \end{cases} \tag{5.51}$$

for $\mathbf{u}^n := (u^n, u_*^n)^\top$ and $\mathbf{x} = (x^n, x_*^n)^\top$. The synthetic data generation procedure associates $N_s = 10^5$ uniform samples of interacting couples of agents $\mathbf{x}_{(i)}^n \in \Omega$ to their associated dSDRE feedback control $\mathbf{u}_{(i)}^n$ and the controlled state update $\mathbf{x}_{(i)}^{n+1}$ obtained with interaction strenght $\varepsilon = 0.01$, for $i = 1, \dots, N_s$. With a ratio of 80/20, we designate those samples to form the training set $\mathcal{T}$ and the validation set $\mathcal{T}_v$ respectively. We test the architectures reported in Table 5.1.

The error comparison for the binary-controlled evolution of a randomly selected pair of interacting agents is presented in Fig. 5.2, where all approximation models demonstrate similar performance to the dSDRE control.

We conclude this preliminary example by quantifying the performance gap between the solution of (5.49) and its neural network-accelerated kinetic approximation, both implemented with a time step $\Delta t = \varepsilon = 0.01$. This comparison offers a more concrete understanding of the suboptimality of the control strategy at the kinetic level with respect to the MFOC. The approximation of the mean field control law under the numerical procedure

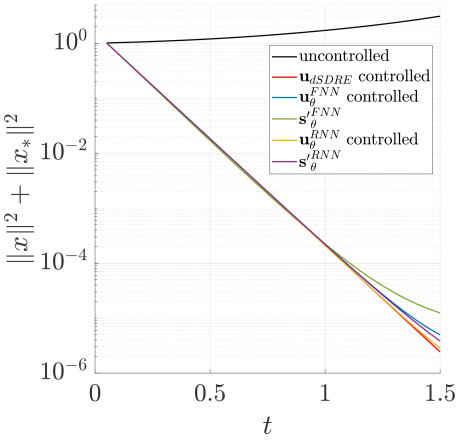


Figure 5.2: Convergence to consensus for two randomly sampled interacting agents with states $x, x_* \in \Omega$, when controlled via dSDRE feedback law vs. the approximation models.

proposed in this paper is retrieved as

$$\int_\Omega \mathbf{u}_\theta(x, x_*)\rho(t, x_*)dx_* \approx u(t, x)\,. \tag{5.52}$$

In Table 5.2, we compare the MPC solution of problem (5.49) with the forward integration of the mean-field dynamics controlled via (5.52), where the control input $\mathbf{u}_\theta$ is given by the previously trained neural networks $\mathbf{u}_\theta^{FNN}$ and $\mathbf{u}_\theta^{RNN}$. The optimality conditions arising from problem (5.49) are approximated with a first-order semi-Lagrangian scheme, as done in [13], but neglecting diffusion. Moreover, to mirror the feedback nature of the control resulting from the numerical routine discussed so far, the solver is applied in an MPC fashion, where the optimality conditions are solved iteratively over a reduced time horizon $T = 0.5s$ [216]. Using the same semi-Lagrangian scheme, we simulate the forward dynamics controlled via the kinetic approximation models through (5.52).

The table reports comparable outcomes in terms of cost functionals, density evolutions, and control profiles across the three approaches. The mean-field control, denoted here by u_{MPC}, retains the best performance. It is worth noting that even u_{MPC} should be regarded as close-to-optimal for an infinite horizon cost. This is because it is computed as the numerical solution of the first-order necessary optimality conditions, implemented in a receding horizon fashion over a sequence of finite-horizon MFOCs. In Figure 5.3,

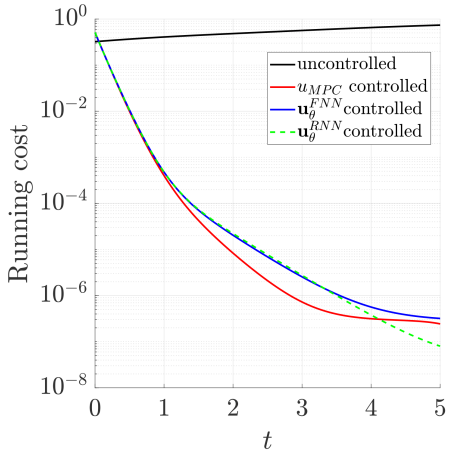


Figure 5.3: Comparison of the time evolution of the running cost for the MFOC (5.49).

we compare the running cost evolution, i.e. the integral w.r.t. time in (5.49), which is consistent with the results reported in Table 5.2.

5.5.2 Test 1: Cucker-Smale model

We proceed by testing the proposed methodology in a consensus control problem for non-linear and non-local dynamics governing the evolution of a N-agent system. In its reduced binary semilinear formulation (5.46), the model is written for the state $\mathbf{s} = (x, x_*, v, v_*)^\top \in \mathbb{R}^{2d} \times \mathbb{R}^{2d}$, with an interaction kernel given by

$$P(x, x_*) = \frac{1}{1 + \|x - x_*\|^2}\,. \tag{5.53}$$

The control variable $\mathbf{u} = (u, u_*) \in \mathbb{R}^{2d}$ is here computed as

$$\tilde{\mathbf{u}} = \underset{u(\cdot)}{argmin}\ \frac{1}{2} \int_0^{+\infty} \|v - \bar{v}\|^2 + \|v_* - \bar{v}\|^2 + \gamma \|\mathbf{u}\|^2 dt\,, \tag{5.54}$$

with $d = 15$, $\gamma = 0.01$, and target velocity $\bar{v} = \dfrac{v + v_*}{2}$. Furthermore, we notice that the cost can be parametrized in quadratic form as done in equation (3.11).

As discussed before, the key ingredient of the proposed methodology is a NN approxima-

$\mathbf{u}_\theta^{RNN}$	$\mathbf{u}_\theta^{FNN}$	u_{MPC}
$\mathcal{J} = 0.063444$	$\mathcal{J} = 0.063504$	$\mathcal{J} = 0.063230$

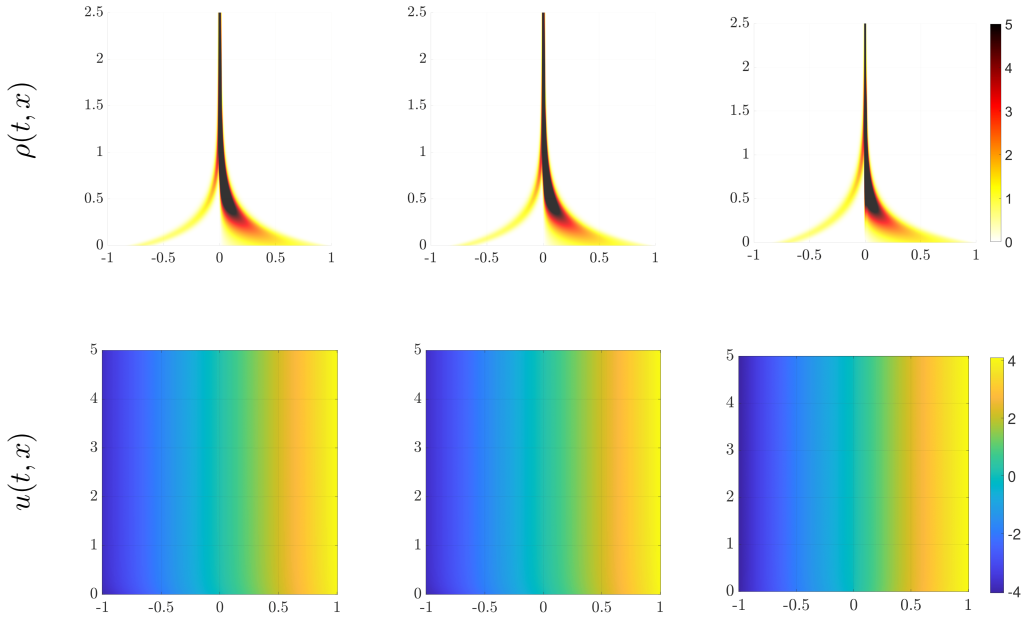


Table 5.2: Comparison of the mean field optimal control problem solved via the proposed numerical procedure against the MPC routine relying on the first-order semi-Lagrangian scheme discussed in [13]. For the three cases, we display the resulting total cost (top), controlled evolution (middle), and mean field control (bottom), with u_{MPC} retaining the best performance.

tion of either the control $\mathbf{u}_\theta$ or the (controlled) update $\mathbf{s}'_\theta$ of the reduced binary problem. In both cases, the synthetic data have been generated from $N_s = 10^5$ uniform samples of interacting couples of agents $\{\mathbf{s}_{(i)}\}_{i=1}^{N_s}$ within $\Omega \times \Omega = [-5, 5]^{4d}$, together with their associated (sub)optimal DSDRE feedback control for the discrete-time infinite horizon OCP (5.44) starting from each $\mathbf{s}_{(i)}$ for the sampled time-step $\Delta t_{(i)} \in [0, 1]$. We highlight that while the dynamics and controls are defined in the entire space, we consider a finite computational domain Ω for sampling.

Once the problem has been written in semi-linear form, we rely on the DSDRE approach (described in algorithm 5) for the generation of a dataset collecting N_s samples of coupled states, time-steps, associated feedback laws, and controlled state updates. With a ratio of 80/20, we designate those samples to form the training set $\mathcal{T}$ and the validation set $\mathcal{T}_v$

Model	K	# Neurons	Activations
$\mathbf{u}_\theta^{FNN}$	1	100	$\sigma(\mathbf{z}) = \log(\exp(\mathbf{z}) + 1)$
$\mathbf{u}_\theta^{RNN}$	1	100	$\sigma(\mathbf{z}) = \max(0, \mathbf{z})$, $\rho(\mathbf{z}) = (1 + \exp(-\mathbf{z}))^{-1}$
$\mathbf{s}'^{FNN}_{v,\theta}$	3	100	$\sigma(\mathbf{z}) = \max(0, \mathbf{z})$
$\mathbf{s}'^{RNN}_{v,\theta}$	1	100	$\sigma(\mathbf{z}) = \max(0, \mathbf{z})$, $\rho(\mathbf{z}) = (1 + \exp(-\mathbf{z}))^{-1}$

Table 5.3: Summary of models and their configurations for Test 1.

respectively. In table 5.3 we summarize the selected architectures.

The behaviour of the approximated controlled binary dynamics for a single couple of agents is shown in Figure 5.4, where we compare the true DSDRE solution with our NN approximations. As the trajectories evolve, the approximated states deviate from the DSDRE closed-loop. However, this does not significantly affect the MC simulation, as a sampled pair of agents will only interact for the duration of a single time step. The goodness of fit of the trained models measured in a *test set* $\mathcal{T}_t \nsubseteq \mathcal{T} \cup \mathcal{T}_v$ is presented in Table 5.6.

In Figure 5.5, we show the evolution of the density of agents' velocities according to a Monte Carlo simulation of binary controlled dynamics, as described in Algorithm 7. All the approximation models equally succeed in steering the density distribution to concentration profiles. Snapshots of the mean field distributions at specific time intervals show how similar the consensus configurations are across the trained models.

$d = 15$	$N_s = 10^2$	$N_s = 10^3$	$N_s = 10^4$	$N_s = 10^5$
$\mathbf{s}'^{FNN}_\theta$	0.045515	0.227243	2.097497	24.060892
$\mathbf{s}'^{RNN}_\theta$	0.067627	0.283442	3.210856	34.087541
$\mathbf{u}_\theta^{FNN}$	0.293578	2.447205	21.754862	2.2594×10^2
$\mathbf{u}_\theta^{RNN}$	0.340493	2.225319	22.486736	2.2559×10^2
$\mathbf{u}$	2.3866×10^2	2.3738×10^3	–	–

Table 5.4: CPU times (seconds) for coupled agents in $\mathbb{R}^{4d}$, $d = 15$, when considering different number of MC samples. The omitted records exceeded a time threshold $t_{max} = 24h$.

So far, the reliance on approximants for the binary controlled state update has been

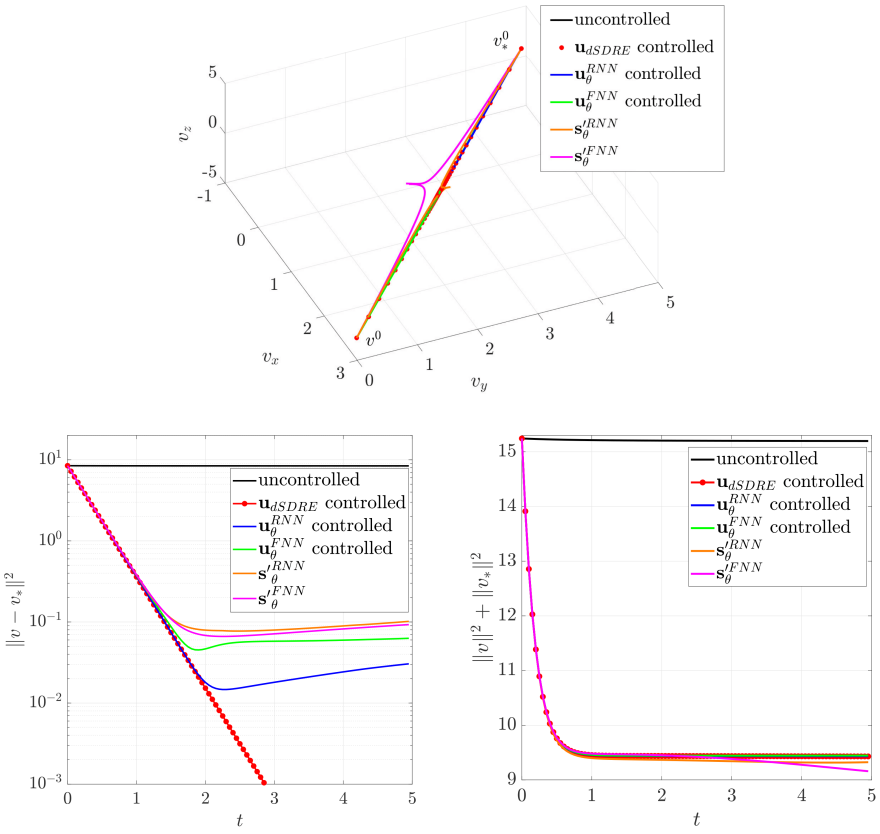


Figure 5.4: Test 1: Evolution of the two interacting agents' in the velocity space restricted to the first 3 dimensions (top); convergence to consensus in logarithmic scale (bottom left); consensus configurations for the different approximators (bottom right).

motivated in terms of efficiency. In Tables 5.4 and 5.5, we present CPU times[1] (in seconds) for the MC simulation with the different models, compared with the DSDRE solution $\mathbf{u}$. The improved efficiency achieved through approximation is crucial, as the alignment between the mean field dynamics and their kinetic approximation relies on high-frequency sampling ($\varepsilon \ll 1$). The evolution of the agents' distribution is approximated along a sequence of discrete times $t_h = n\Delta t$, $\Delta t = 0.05$, $n = 1, \dots, 100$. The tables address the computational cost associated to sampling a number of controlled binary interactions in the MC simulation, and the dimensionality d of the agents physical space, for a total of $4d$ dimensions. The reliance on NN approximation models allows for a speedup of 2 to 3 orders of magnitude. For $d \geq 7$ and $N_s \geq 10^4$, the computational cost resulting from the use of $\mathbf{u}_\theta$ displays a linear growth, whilst $\mathbf{s}'_\theta$ performs even better. CPU times for the true DSDRE controlled dynamics exceed 24 hours.

$N_s = 10^4$	$d = 3$	$d = 7$	$d = 10$	$d = 15$	$d = 30$
$\mathbf{s}'^{FNN}_\theta$	1.048226	1.212633	1.390813	2.142041	2.617840
$\mathbf{s}'^{RNN}_\theta$	2.033726	2.243084	2.493256	3.210856	3.893368
$\mathbf{u}^{FNN}_\theta$	7.712628	11.006977	15.041731	21.754862	70.172311
$\mathbf{u}^{RNN}_\theta$	7.734224	11.224325	15.991421	22.486736	70.564372
$\mathbf{u}$	1.1979×10^3	5.2136×10^3	–	–	–

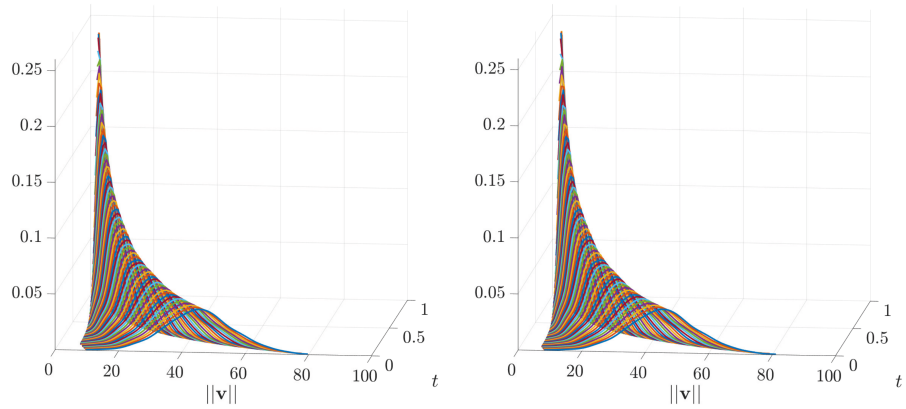
Table 5.5: CPU times (seconds) when considering 10^4 MC samples of coupled agents in $\mathbb{R}^{4d}$, with varying d. The omitted records exceeded a time threshold $t_{max} = 24h$.

5.5.3 Test 2: quasi-Morse potential

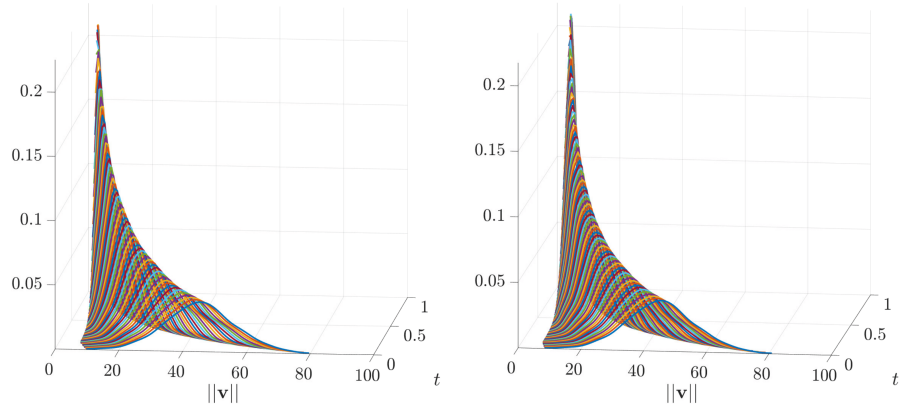
As a second numerical test, we consider the consensus control problem analysed in [68] for the interacting particles of a second order system of agents in the physical space $\mathbb{R}^3$. The velocities are here governed by both a self-propulsion force, expressed in the $i-th$ agent by the term $(\alpha - \beta\|v_i(t)\|^2)v_i(t)$, for fixed $\alpha \geq 0$, $\beta > 0$, and an attraction-repulsion force acting though the pairwise interaction potential W. We consider a radial potential of the form

$$W(x) = V(\|x\|) - C\,V\left(\frac{\|x\|}{l}\right), \qquad V(r) := -exp\left\{\frac{-r^p}{p}\right\} \tag{5.55}$$

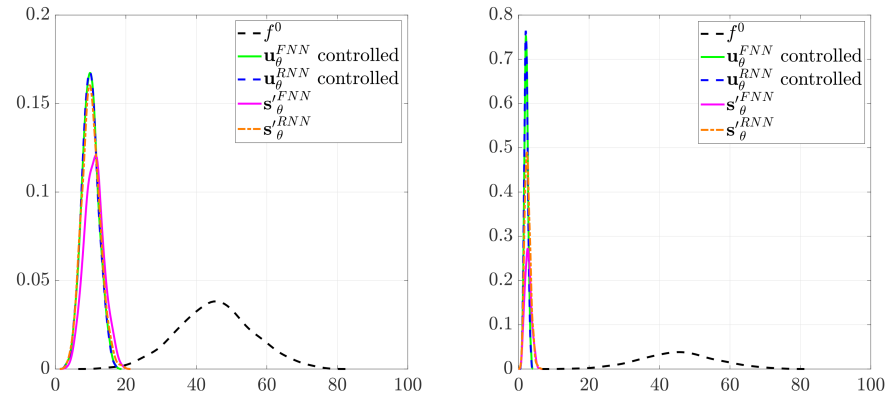
[1]The experiments have been executed in MATLAB R2022a installed on a machine with Intel Core i7-10700 processor running at 2.90GHz.



(a) $\mathbf{u}_\theta^{RNN}$ (b) $\mathbf{u}_\theta^{FNN}$



(c) $\mathbf{s}'^{RNN}_\theta$ (d) $\mathbf{s}'^{FNN}_\theta$



(e) $t_{50} = 0.5s$ (f) $t_{100} = 1s$

Figure 5.5: Density of agents' velocities in norm for Test 1. Comparison of the time evolution of the MC pdf obtained considering different approximation approaches for the controlled binary interactions. In the bottom row, we display a comparison of the different approximated densities at two different discrete times.

	mlticolumn3cTest 1			mlticolumn3cTest 2		
Model	r^2	*MSE*	*MRE %*	r^2	*MSE*	*MRE %*
$\mathbf{s}'^{FNN}_{\theta}$	0.99998	0.075252	0.38964	0.9994	0.038274	0.60308
$\mathbf{s}'^{RNN}_{\theta}$	0.99999	0.0069192	0.3739	0.9998	0.012784	0.20956
$\mathbf{u}^{FNN}_{\theta}$	0.99996	0.045596	0.63555	0.9997	7.6882	3.4018
$\mathbf{u}^{RNN}_{\theta}$	0.99998	0.018018	0.40136	0.99979	5.5492	2.7483

Table 5.6: Goodness of fit for Tests 1 and 2 in terms of: *coefficient of determination* r^2, *mean squared error* and *mean percentage error*. $\mathcal{T}_t$ is a collection of sampled states $\mathbf{s}^{(i)} \in (\Omega \times \Omega)$ and related target DSDRE values for $i = 1, ..., 10^5$.

with $C = 0.6$, $p = 1.5$, $l = 0.5$, $\alpha = 2$, $\beta = 1.5$, in a similar configuration as in [68]. Accordingly, for the couple of interacting agents $\mathbf{s} = (x, x_*, v, v_*)$ we define

$$P_x(x, x_*) = \|x - x_*\|^{(p-2)} \left(\frac{C}{l^p} e^{-\frac{\|x - x_*\|^p}{p\, l^p}} - e^{-\frac{\|x - x_*\|^p}{p}} \right) \tag{5.56}$$

$$P_v(v) = (\alpha - \beta \|v(t)\|^2)\,, \tag{5.57}$$

with which we write the reduced binary dynamics in semilinear form as

$$A = \begin{bmatrix} \mathbb{O}_{2d} & \mathbb{I}_{2d} \\ A_x & A_v \end{bmatrix} \qquad \Big[A_x\Big]_{i,j} = \begin{cases} -P_x(x, x_*) & \text{if } i = j \\ P_x(x, x_*) & \text{if } m \text{ odd}, \ j = i + 1 \\ P_x(x, x_*) & \text{if } m \text{ even}, \ j = i - 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.58}$$

where A_v is the diagonal matrices with diagonal vector the component-wise application of $P_v(\cdot)$ to the vector with d repetitions of v, v_*. Moreover, the cost operators for the consensus goal Q, R hold as in (3.11).

As discussed for the previous numerical example, we aim at approximating the binary control $\mathbf{u}$, resulting from the DSDRE approach, and the related controlled state update

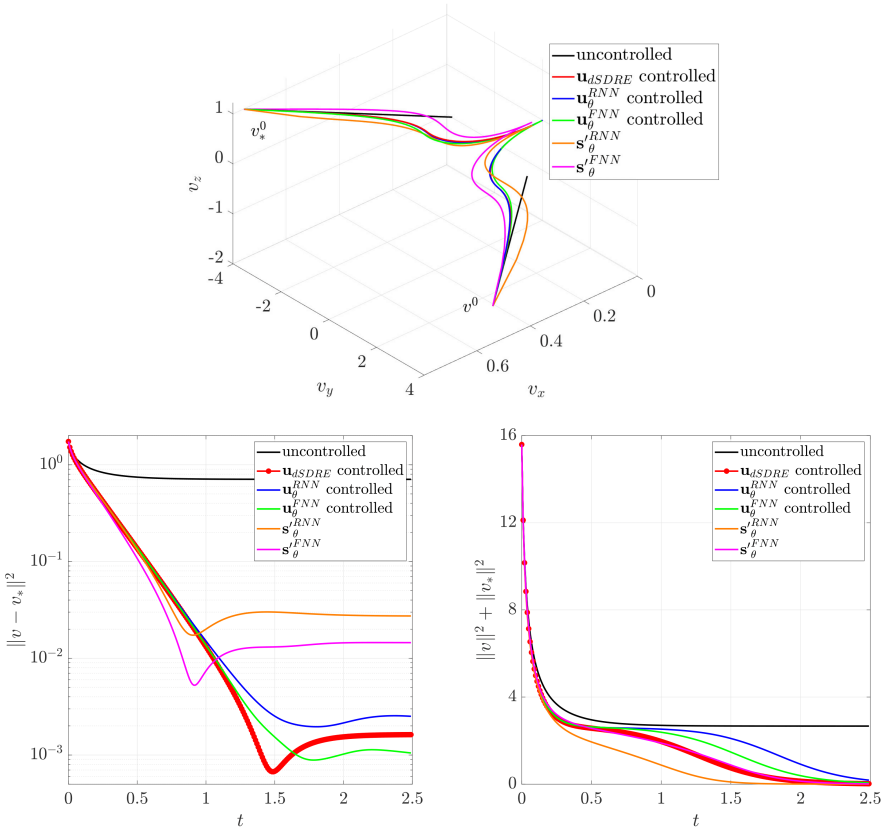


Figure 5.6: Evolution (in velocities) of a couple of approximately controlled interacting agents under the different models, versus the uncontrolled dynamics and the dSDRE-controlled ones (top); convergence to consensus in logarithmic scale (bottom left); consensus configurations for the different approximators (bottom right).

Model	K	# Neurons	Activations
u_θ^{FNN}	3	100	$\sigma(\mathbf{z}) = \log(1+\exp(\mathbf{z}))$
u_θ^{RNN}	5	100	$\sigma(\mathbf{z}) = \rho(\mathbf{z}) = \log(1+\exp(\mathbf{z}))$
$\mathbf{s}'^{FNN}_{v,\theta}$	4	100	$\sigma(\mathbf{z}) = \log(1+\exp(\mathbf{z}))$
$\mathbf{s}'^{RNN}_{v,\theta}$	3	100	$\sigma_1(\mathbf{z}) = \rho(\mathbf{z}) = \log(1+\exp(\mathbf{z}))$, $\sigma_2(\mathbf{z}) = \sigma_3(\mathbf{z}) = elu(\mathbf{z})$

Table 5.7: Summary of models and their configurations for Test 2.

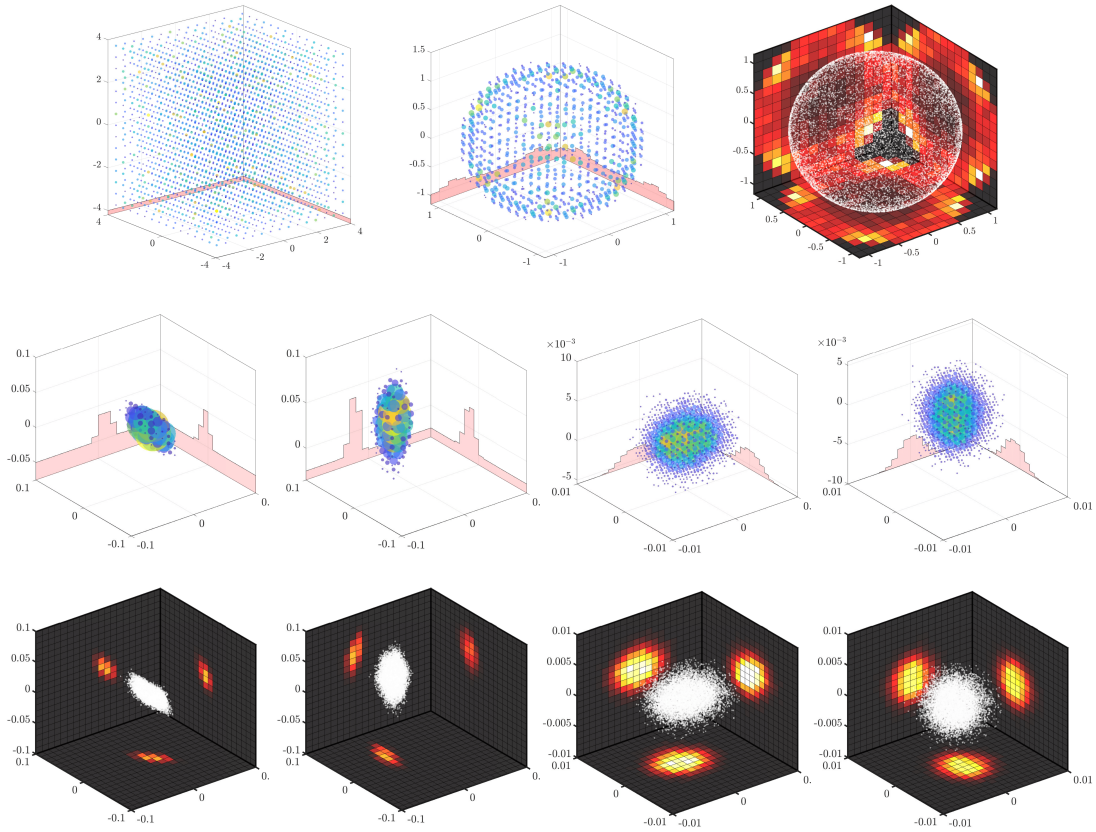
$\mathbf{s}'$. We train the model described in Table 5.7, where

$$elu(\mathbf{z}) := \begin{cases} e^{\mathbf{z}} - 1 & \text{if } \mathbf{z} \leq 0 \\ \mathbf{z} & \text{if } \mathbf{z} > 0 \end{cases} \tag{5.59}$$

The goodness of fit of these models outside $\mathcal{T} \cup \mathcal{T}_v$ is displayed in table 5.6, whilst a comparison of the approximately controlled binary system is displayed in Figure 5.6, where the dynamics of a couple of agents with states randomly sampled in $[-4, 4]^{12}$ evolve throughout 40 discrete time intervals of length $\Delta t = 0.02s$. Whilst all the NN models lead to a similar final state, those approximating the feedback design perform closer to the reference DSDRE solution. As shown in Figure 5.7, the controlled evolution converges to concentrated density profiles, exhibiting slight differences in mean and variance among the various models. This can also be noticed from the 5.8, which depicts the density of agents along the first two dimensions, together with the associated vector field of the velocities.

5.6 Concluding remarks

We have developed a novel computational method for mitigating the curse of dimensionality arising in the optimal control of large-scale, high-dimensional, agent-based models. The key ingredients of the proposed approach are: the use of a kinetic model to reduce the complexity associated to the particle ensemble to the sampling of two-agent subproblems, and the synthesis of control laws for the two-agent system by means of deep neural networks, supervised learning, and a discrete-time state-dependent Riccati equation approach.

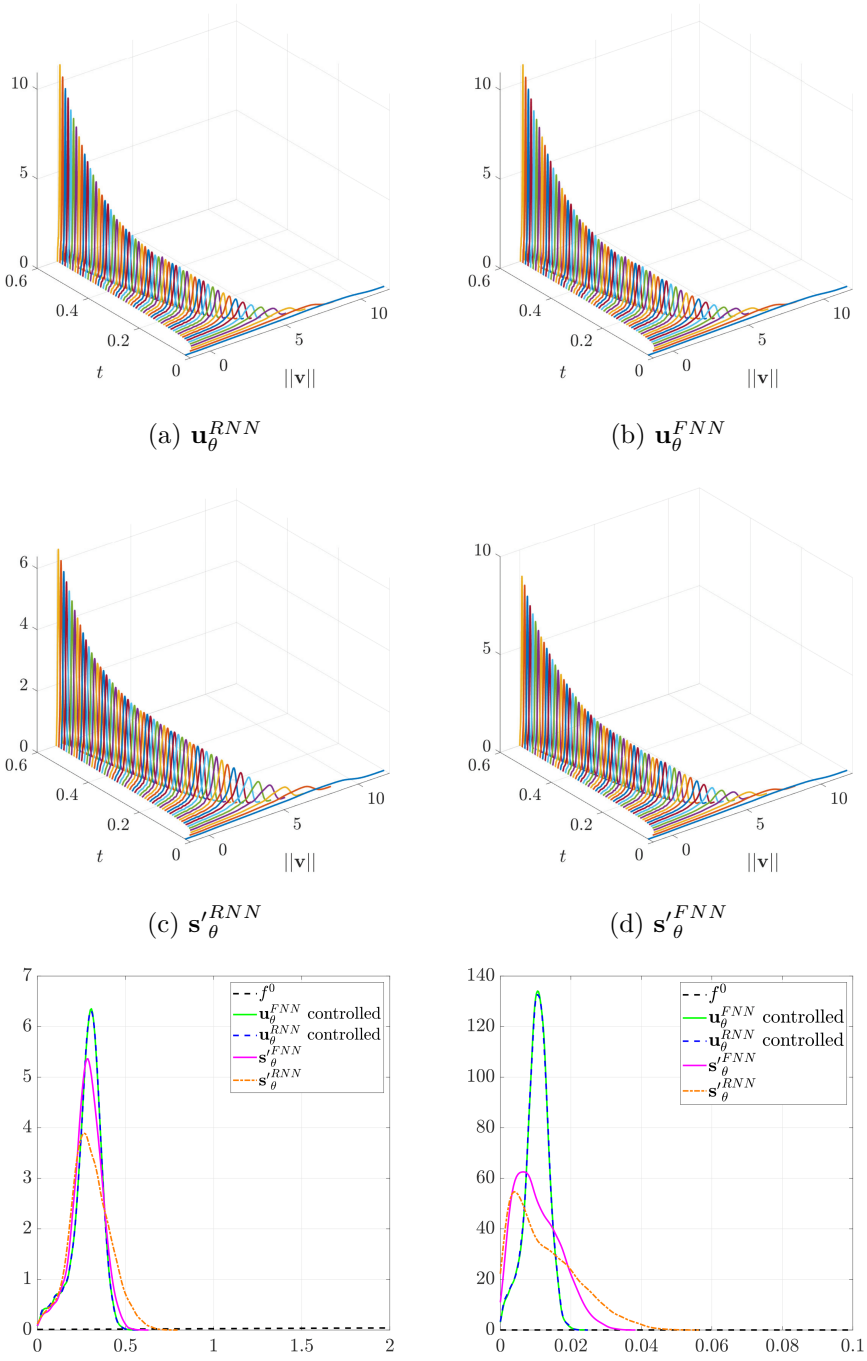


(a) $\mathbf{s}'^{RNN}_{\theta}$ (b) $\mathbf{s}'^{FNN}_{\theta}$ (c) $\mathbf{u}^{RNN}_{\theta}$ (d) $\mathbf{u}^{FNN}_{\theta}$

Figure 5.7: Density of the system in the velocity space. First row: density of the uncontrolled system at initial and final times $t \in \{0s, 2s\}$. Second row: controlled system configuration at time $T = 2s$, with projection of partial densities along the first and second dimension. Third row: configuration of the controlled system at the final time $T = 2s$, with projection of partial densities along all the three dimensions.

Our numerical experiments validate the effectiveness of the approach in controlling consensus and attraction-repulsion dynamics in high-dimensional interacting particle systems. The use of neural network approximation models for fast feedback synthesis allows for a speedup of 2 to 3 orders of magnitude compared to solving a binary optimal control problem every time two interacting agents are sampled. The proposed framework can be extended to construct a deep neural network that directly predicts the post-interaction states, or to different control problems provided that a solver for synthetic data generator is available.

Potential future research directions include the generalization of the proposed approach



(a) $\mathbf{u}_\theta^{RNN}$ (b) $\mathbf{u}_\theta^{FNN}$

(c) $\mathbf{s}'^{RNN}_\theta$ (d) $\mathbf{s}'^{FNN}_\theta$

(e) $t_5 = 0.5s$ (f) $t_{100} = 1s$

Figure 5.8: Density of agents' velocities in norm for test 2. Comparison of the time evolution of the MC pdf obtained considering different approximation approaches for the controlled binary interactions. In the bottom row, we display a comparison of the different approximated densities at two different discrete times.

to the $\mathcal{H}_\infty$ robust control framework, which could further enhance the applicability of the method to real-world scenarios where robustness to uncertainties in the agents' interaction forces is crucial. Additionally, exploring the integration of adaptive time-stepping techniques with the neural network approximation models could lead to faster and more accurate simulations of the controlled agent-based system. Finally, investigating the scalability of the proposed approach to even higher-dimensional problems and larger agent populations could enable applications in consensus-based optimization and mean field limits of neural networks.

Chapter 6

Conclusions

This thesis has addressed fundamental computational challenges in solving optimal control problems for high-dimensional many-agent systems, with a particular focus on overcoming the curse of dimensionality. Inspired by collective behaviors in nature and their mathematical description through agent-based modeling, we have developed a multi-scale framework that significantly improves the efficiency and scalability of optimal control computations across different modeling resolutions.

At the **microscopic level**, we leveraged Neural Network approximation techniques to accelerate existing numerical OCP solvers. By incorporating gradient information from optimization routines into our learning architecture, we achieved remarkable computational efficiency gains in high-dimensional settings. Specifically, in a 200-dimensional state space, our method demonstrated speedups of up to two orders of magnitude compared to traditional numerical solvers, while preserving reasonable solution accuracy. We validated this approach on two challenging problems: achieving consensus in high-dimensional second-order ABMs and stabilizing nonlinear parabolic PDEs. The substantial performance improvement make real-time control applications feasible, which would otherwise be computationally prohibitive using conventional techniques.

At the **macroscopic level**, we adopted a mean-field perspective to tackle the dimensionality challenges inherent in large-scale agent systems driven by stochastic dynamics, in

particular weakly interacting diffusions on the torus. This led to the formulation of stabilization problems for McKean–Vlasov PDEs, where we developed a systematic pipeline for computing and controlling unstable steady states. Our approach combined spectral discretization, a deflation algorithm for identifying multiple steady states, and a nonlinear model predictive control strategy to achieve asymptotic stabilization towards a desired unstable equilibrium. While unstable steady states do not naturally emerge from the system's evolution, there are scenarios where deliberately steering a system towards an unstable configuration is beneficial. A key example is opinion dynamics, where breaking down clustering in highly polarized societies can be desirable. To demonstrate the effectiveness of our method, we applied it to agent-based opinion dynamics models in both one and two dimensions, showing its potential to shape collective behaviors in social systems.

At the **mesoscopic level**, we tackled the control of ABMs where both the number of agents N and the dimensionality of the underlying agent dynamics d render traditional control synthesis infeasible. To approximate the mean-field OCP, we introduced a kinetic approximation framework that models agent densities as a form of plasma matter. This hybrid approach balances between computational efficiency and control accuracy by averaging over controlled subsystems of interacting agent pairs. Further enhancing performance, we implemented neural network-based controllers for the reduced $2-$agent subsystems, significantly accelerating control synthesis. Compared to conventional methods, our approach demonstrated a speed-up of at least 3 orders of magnitude in terms of CPU times, reinforcing its potential for scalable control of high-dimensional agent-based models.

6.1 Future directions

While this thesis has developed computational techniques to tackle optimal control challenges across multiple modeling scales, many research directions remain open. Building on the frameworks established in this work, we identify two main avenues for future exploration: robust control of ABMs under modeling uncertainties at the mesoscopic scale and inverse optimal control of agent-based models in conservation science. These directions

extend our multi-scale approach to address practical challenges in handling model uncertainty and inferring control objectives from observed collective behaviors. The remainder of this section introduces these problems and presents some preliminary work on these topics.

6.1.1 Robust kinetic control for parametric uncertainty

In many practical scenarios, the parameters governing agent interactions are not precisely known, posing challenges for control design. To address this, we propose leveraging the H_∞ control framework at the kinetic level, extending our current methodologies (Chapter 5) by incorporating robustness against model uncertainties. A key aspect of this approach is the use of neural networks to approximate optimal feedback laws while accounting for worst-case disturbances in the system dynamics.

Case study

As a preliminary example, we consider a system of N interacting agents governed by a Sznajd-type opinion dynamics model, where uncertainties in interaction kernels can significantly impact collective behavior:

$$\dot{x}_i = \frac{1}{N}\sum_{j}^{N} P(x_i, x_j)(x_j - x_i) + u_i, \tag{6.1}$$

where the interaction kernel

$$P(x_i, x_j) = \beta(1 - x_i)^2 \tag{6.2}$$

depends on the unknown parameter β. We model this uncertainty as a confidence interval $\beta = \bar{\beta} \pm \omega$ around a known reference value $\bar{\beta}$. The perturbation term $|\omega| \ll 1$ acts as an *adversarial signal* opposing our control action $\mathbf{u} = (u_1, \dots, u_N)^\top$.

Our objective is to design a robust control law that ensures stability despite these uncertainties. Using the kinetic argument from Chapter 5, we approximate the original optimal control problem via averaging of many two-agent subproblems. The simplified system for

$N = 2$ agents is given by:

$$\begin{aligned} \dot{x}_1 &= \bar{\beta}(1 - x_1)^2(x_2 - x_1) \pm \omega(1 - x_1)^2(x_2 - x_1), \\ \dot{x}_2 &= \bar{\beta}(1 - x_2)^2(x_1 - x_2) \pm \omega(1 - x_2)^2(x_1 - x_2). \end{aligned} \tag{6.3}$$

Rewriting in affine control form:

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u} + h(\mathbf{x})\omega, \tag{6.4}$$

where $\mathbf{x} = (x_1, x_2)^\top$ and $\mathbf{u} = (u_1, u_2)^\top$, with initial condition $\mathbf{x}(0) = \mathbf{y}$. We assume the adversarial signal satisfies $\omega \in [-\delta, \delta]$ with $\delta \ll 1$ and remains constant over time.

Robust binary control

The control goal is to ensure both internal stability and disturbance attenuation [244, 145], formulated as:

$$\int_0^T \ell(\mathbf{x}; t) + \mathbf{u}^\top R\mathbf{u}\, dt \leq \gamma^2 \int_0^T \omega^\top P\omega\, dt, \tag{6.5}$$

where γ is a robustness parameter quantifying the control's ability to mitigate adversarial effects. A minimal value γ^* (the H_∞ norm of the system) exists, ensuring the feasibility of an asymptotically stabilizing feedback law.

For a chosen $\gamma \geq \gamma^*$, applying the dynamic programming principle leads to the Hamilton-Jacobi-Isaacs equation:

$$\min_{\mathbf{u}\in\mathcal{U}} \max_{\omega\in[-\delta,\delta]} \left\{ \nabla V(\mathbf{x})^\top (f(\mathbf{x}) + g(\mathbf{x})\mathbf{u} + h(\mathbf{x})\omega) + \ell(\mathbf{x}) + \mathbf{u}^\top R\mathbf{u} - \gamma^2 \omega^\top P\omega \right\} = 0, \tag{6.6}$$

providing a recipe for computing robust optimal feedback policies that account for worst-case parameter uncertainties. As done in standard optimal control theory for the HJB equation 2.11, we synthesize a stabilizing feedback law $\mathbf{u}_\gamma$ along with its associated worst-case disturbance ω_γ:

$$\mathbf{u}_\gamma(\mathbf{x}) = -\frac{1}{2} R^{-1} g(\mathbf{x})^\top \nabla V(\mathbf{x}), \tag{6.7}$$

$$\omega_\gamma(\mathbf{x}) = \frac{1}{2\gamma^2} P^{-1} h(\mathbf{x})^\top \nabla V(\mathbf{x}), \tag{6.8}$$

where $V(\mathbf{x})$ solves the HJI equation:

$$\nabla V(\mathbf{x})^\top f(\mathbf{x}) + \frac{1}{4} \nabla V(\mathbf{x})^\top S(\mathbf{x}) \nabla V(\mathbf{x}) + \ell(\mathbf{x}) = 0, \tag{6.9}$$

with

$$S(\mathbf{x}) = \frac{1}{2\gamma^2} h(\mathbf{x}) P^{-1} h(\mathbf{x})^\top - g(\mathbf{x}) R^{-1} g(\mathbf{x})^\top. \tag{6.10}$$

We highlight that, in the limit $\gamma \to \infty$, the disturbance attenuation term vanishes, and the Hamilton-Jacobi-Isaacs equation reduces to the standard HJB.

Physics Informed Neural Networks for the binary HJI equation

Since standard numerical techniques such as LQR and SDRE are not directly applicable to solving the HJI equation arising from this H_∞ OCP, we propose addressing it using a semi-supervised learning approach based on Physics-Informed Neural Networks (PINNs). Inspired by [222], we approximate the solution of the HJI equation by minimizing a residual loss function over the domain. Given a general PDE of the form

$$\mathcal{N}[V](x) = 0, \qquad x \in \mathbb{R}^{2d}, \tag{6.11}$$

where $\mathcal{N}$ is a differential operator (in our case the LHS of equation (6.9)), we approximate its solution by minimizing the loss function

$$\mathcal{J}(V_\theta) = \|\mathcal{N}[V_\theta]\|^2_{2,\Omega} + \mu \sum_{i=1}^{N_d} \|\nabla V_\theta(\mathbf{x}^{(i)}) - \nabla V^{(i)}\|^2, \tag{6.12}$$

where V_θ is the neural network parameterized by θ, and the additional term penalizes deviations from the reference gradients $\nabla V^{(i)} := \nabla V(\mathbf{x}^{(i)})$ computed using data synthesized via the PMP. The dataset $\{\mathbf{x}^{(i)}\}_{i=1}^{N_s} \subset \mathbb{R}^{2d}$ consists of sampled states from optimal trajectories obtained through PMP [183, 84], ensuring that the learned value function aligns with optimal control solutions. The parameter μ balances the contributions of the

physics-informed residual and the data-driven regularization term.

Next steps

This project is still in its early stages, and several challenges remain in effectively solving these problems. One of the primary obstacles encountered is the optimization process, particularly in achieving a proper balance between the unsupervised and supervised components within the multi-objective optimization framework. The current optimizer selection does not fully capture the complexities inherent in the problem, necessitating the exploration of more robust optimization methods.

Future research should focus on identifying and developing alternative optimization strategies that can better address the intricacies of multi-objective optimization in kinetic control problems. Notably, recent work in [25] on the challenges of the nonlinear regime for PINNs provides valuable insights into improving the training process in the presence of complex nonlinearities. Their approach offers a promising direction to tackle some of the difficulties encountered when optimizing these systems, potentially leading to more effective solutions of the binary HJI equation.

6.1.2 Towards a quantitative understanding of collective animal behaviour under climatic events

This open research project, supported by the Turner-Kirk Trust and carried out in collaboration with Vincent Savolainen from the Life Science department at Imperial College London, aims to address the pressing challenge of understanding collective animal behavior in the context of extreme climatic events. Climate change poses significant threats to biodiversity, with severe weather events such as hurricanes and wildfires causing widespread devastation to animal populations. In these circumstances, collective behavior emerges as a key survival strategy, where individual animals interact in response to environmental stressors.

Understanding the dynamics of these interactions is crucial for uncovering the self-organizing principles and decision-making mechanisms that guide animal populations during such

emergencies. This project, with the ultimate objective of inferring the social structure of animal groups and the decision rules that govern their behavior in order to maximize survival, employs a combination of mathematical modeling, computational simulations, and empirical data collection to analyze collective animal behavior.

Case study

The primary case study for this work is the macaque population on Cayo Santiago, a small island off the coast of Puerto Rico. Since 1938, the island has been home to a free-ranging population of rhesus macaques (*Macaca mulatta*). Today, the colony consists of approximately 1,800 individuals, all direct descendants of the original 409 monkeys. The Cayo Santiago Field Station serves as a key research site for behavioral and noninvasive studies on the colony [86], offering a unique opportunity to study the dynamics of collective behavior in a controlled yet natural environment.

On the 20th of September 2017, Hurricane Maria made landfall in Puerto Rico as a Category 4 storm, with winds reaching 250 km/h and torrential rain. The storm devastated the island, triggering a major humanitarian crisis, and it became the strongest hurricane to hit Puerto Rico in nearly 90 years, resulting in a death toll exceeding 3000 people. Cayo Santiago was directly in the storm's path, causing significant damage to the research facilities on the island. Despite the widespread destruction, remarkably, only about 50 macaques (approximately 2% of the colony) from the colony perished. This event attracted a lot of research interest, with extensive studies focused on both the social [234, 235] and biological [249] reaction to the trauma of the storm. The high survival rate among the macaques has been attributed to their collective behavior, especially the role of leadership and the social structure, which enabled the group to navigate the storm's challenges and ensure survival. This episode became the focus of our study, as it provided a compelling case to investigate the role of social dynamics and leadership in survival strategies under extreme climatic events.

Mathematical Modeling

A dynamical system was developed to represent macaque populations as interacting agents driven by a combination of interaction forces and self-propulsion. A two-population model was specifically constructed to distinguish followers from leaders, incorporating alignment, repulsion from the hurricane, and leader anticipation of the hurricane's trajectory.

The colony follows a *hierarchical social structure*, where groups of macaques follow a designated leader. It is assumed that the leaders are experienced individuals, typically elders, who possess knowledge of extreme events and safe routes or locations. Additionally, macaques with stronger social ties are hypothesized to have a higher likelihood of survival during the hurricane due to their knowledge of safe areas and their ability to stay close to leaders.

We model the macaque colony on the island as a system of interacting agents, represented by a set of position-velocity pairs $(\mathbf{x}_i, \mathbf{v}_i)_{i=1}^N$, where $\mathbf{x}_i, \mathbf{v}_i \in \mathbb{R}^2$ denote the position and velocity of the i-th agent, respectively, for $i = 1, \ldots, N$. The agents move in space according to Newton's Second Law:

$$\ddot{\mathbf{x}}_i = \sum \mathbf{F}(\mathbf{x}_i)\,, \tag{6.13}$$

where the forces $\mathbf{F}$ acting on the ith agent can be distinguished between endogenous and exogenous. An agent is influenced from within the system via topological interactions with other agents. Agents communicate within a neighborhood $\mathcal{N}$ comprising M closest agents (and a leader):

$$F_i^{\text{endo}} = \sum_{j \in \mathcal{N}} C_1 \left[(\mathbf{x}_j - \mathbf{x}_i) \mathbb{I}_{(\|\mathbf{x}_j - \mathbf{x}_i\| > r_1)} \right] - C_2 \left[e^{-\|\mathbf{x}_j - \mathbf{x}_i\|^\gamma} \frac{\mathbf{x}_j - \mathbf{x}_i}{\|\mathbf{x}_j - \mathbf{x}_i\|} \mathbb{I}_{(\|\mathbf{x}_j - \mathbf{x}_i\| \leq r_2)} \right], \tag{6.14}$$

where C_1 and C_2 denote long-range (r_1) attraction and short-range (r_2) repulsion strength respectively. We further enhance follower-leader attraction with higher value of C_1 when j is the index of a leader.

Agents are also influenced by the environment, in this case the moving storm. This is translated in our model as a wind-induced perpendicular movement and a repulsion from

the storm center $\mathbf{x}_c$

$$\mathbf{F}_i^{\text{exo}} = \Phi(\mathbf{x}_i) \cdot \hat{\mathbf{n}}_i + \kappa \frac{\mathbf{x}_i - \mathbf{x}_c(t)}{\|\mathbf{x}_i - \mathbf{x}_c(t)\|^3} \tag{6.15}$$

where $\Phi(\mathbf{x})$ denotes the wind power at the position $\mathbf{x}$ and κ is the repulsion strength parameter. Leaders have knowledge of the storm trajectory in a window Δt in the future, and

$$\mathbf{F}_{leader}^{exo} = \Phi(\mathbf{x}_i) \cdot \hat{\mathbf{n}}_i + \kappa \int_t^{t+\Delta t} \frac{\mathbf{x}_i - \mathbf{x}_c(s)}{\|\mathbf{x}_i - \mathbf{x}_c(s)\|^3} ds \,. \tag{6.16}$$

This detailed mathematical model provides the foundation for our future work in inverse optimal control, where we aim to infer the underlying optimization principles that guided the macaques' survival strategies during Hurricane Maria. By observing the emergent collective behaviors and comparing them with our model predictions, we can reverse-engineer the implicit cost functions that leaders and followers may have been optimizing, potentially revealing key insights into adaptive decision-making under extreme environmental stress.

Simulations

Computational experiments simulated the movement of a macaque group of in response to a modeled tornado on Cayo Santiago. The results show that leaders play a critical role in ensuring group survival, with significantly higher mortality rates when leader anticipation is absent.

Field Data Collection

Fieldwork at Longleat Safari Park captured drone footage of a group of macaques of the same type as the ones in Cayo Santiago. The animals were filmed while interacting, and reacting to food introduction. This serves as a proxy to better undestand the response of the collective to external disturbances. The dataset will be used for model calibration, ensuring that the simulated behaviors align with real-world observations.

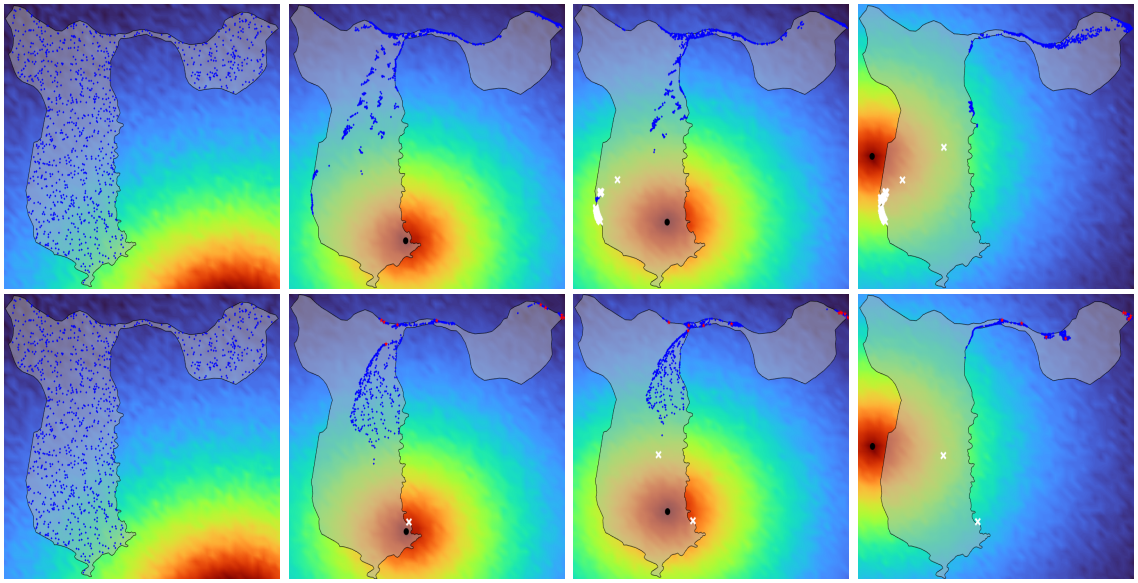


Figure 6.1: Simulation of $N = 1000$ agents (blue dots), neighborhood size $M = 5$. **Top:** no leaders. **Bottom:** 10 leaders (red stars). White crosses denote dead agents.

Next steps

Data post-processing: Extract and analyze movement trajectories from drone footage to quantify both individual and group behavior. This will involve applying computer vision techniques, such as convolutional neural networks for object detection and tracking, alongside optical flow methods to capture motion dynamics. Additionally, deep learning-based multi-object tracking and behavioral clustering techniques can be explored to differentiate social interactions from responses to environmental factors. These approaches will help infer leadership roles, quantify coordination strategies, and analyze collective movement patterns under extreme conditions.

Inverse problems: Using the tracked trajectories, we aim to infer the underlying decision-making mechanisms that drive collective behavior in the colony. This can be approached at multiple levels:

1. *Inferring social interaction rules*: By analyzing movement patterns, we can estimate the strength and nature of social forces that influence decision-making, such as attraction, repulsion, and alignment. These forces determine group cohesion, escape strategies, and responses to environmental threats. Recent work on learn-



Figure 6.2: Drone footage from Longleat Safari Park. Studying the behaviour of the macaque group to food serve as a proxy to better understand the system an exogenous disturbance.

ing interaction kernels from trajectory data [165] and inferring social dynamics in multi-agent systems [113] offers promising approaches to reconstructing underlying decision mechanisms. Variational methods [55] can further aid in recovering interaction rules by formulating the problem as an optimization task constrained by observed behaviors.

2. *Network-based models*: the macaque society follows a hierarchical structure, where social bonds influence movement decisions. By modeling these interactions through graph-based representations (as done in [173]), we can infer the underlying social network and its role in optimizing collective responses.

3. *Leaders as optimal controllers*: leaders can be modeled as agents optimizing group survival by anticipating environmental risks. This can be formulated as an inverse optimal control problem, where the objective is to recover the cost function that best explains observed behaviors. Inverse optimal control with constraints [144] provides a framework for recovering both the objective function and the constraints shaping agent behavior, which is particularly relevant in understanding how animal groups balance survival strategies with social cohesion. As an alternative avenue, insights from reinforcement learning (RL) suggest that inverse optimal control can be used to infer cost structures [154]. Since RL can be viewed as an optimal control problem with unknown dynamics, where an agent learns an optimal policy through interaction

with the environment, inverse RL techniques allow us to recover the implicit cost function that drives decision-making.

By integrating these techniques, we aim to reconstruct the implicit optimization strategies that macaques employ during extreme climatic events, advancing our understanding of adaptive collective behavior.

Supplementary material to Macroscopic Scale Chapter

A.1 Fréchet differentiability of the cost functional $\mathcal{J}$

This section examines the differentiability of the cost functional (4.39), which serves as the objective in our optimal control problem. The differentiability assumption is fundamental to the derivation of first-order optimality conditions, upon which our model predictive control algorithm is built.

By definition, $\mathcal{J}$ is Fréchet differentiable at $u \in \mathcal{U} := L_2(0,T;L_\infty(\Omega))$ if there exists a bounded linear operator $D\mathcal{J}(u)[\cdot]$ such that

$$\lim_{\|h\|_{\mathcal{U}}\to 0} \frac{\left|\mathcal{J}(u+h) - \mathcal{J}(u) - D\mathcal{J}(u)[h]\right|}{\|h\|_{\mathcal{U}}} = 0, \tag{17}$$

then $D\mathcal{J}(u)[h]$ is the Fréchet derivative of $\mathcal{J}$.

The proof follows 3 steps assessing the existence of a candidate operator matching the definition (17), its linearity, and boundedness.

1. **Candidate derivative -** We start by showing that the candidate $D\mathcal{J}(u)[h]$ we used to derive the optimality system (4.41) satisfies (17). The expression for $D\mathcal{J}(u^*)[\cdot]$ was derived following [241] and reads:

$$D\mathcal{J}(u)[h] := \int_0^T \int_\Omega (\rho(t,x) - \rho_\infty(x))\delta\rho(t,x) + \gamma\, u(t,x)\, h(t,x)\, dx\, dt\,,$$

where $\delta\rho$ denotes the change in the state ρ induced by the perturbation $u \mapsto u + h$.

By expanding the perturbed cost functional $\mathcal{J}(u+h)$ we obtain:

$$\begin{aligned}\mathcal{J}(u+h) &= \frac{1}{2}\int_0^T\int_\Omega (\rho-\rho_\infty+\delta\rho)^2+\gamma(u+h)^2 dx\,dt \\ &= \mathcal{J}(u)+\frac{1}{2}\int_0^T\int_\Omega 2\delta\rho(\rho-\rho_\infty)+\delta\rho^2+\gamma h^2+2\gamma uh\,dx\,dt\,.\end{aligned}$$

Hence

$$\mathcal{J}(u+h)-\mathcal{J}(u)-D\mathcal{J}(u)[h]=\frac{1}{2}\int_0^T \|\delta\rho\|^2+\gamma\|h\|^2\,dt\,,$$

where a standard energy estimate gives

$$\|\delta\rho\|_{L_2(0,T;L_2(\Omega))}\leq C\|h\|_{\mathcal{U}}\,. \tag{18}$$

We conclude upon noticing that

$$\frac{|\mathcal{J}(u+h)-\mathcal{J}(u)-D\mathcal{J}(u)[h]|}{\|h\|_{\mathcal{U}}} \;\leq\; \frac{\frac{(C+\gamma)}{2}\|h\|_{\mathcal{U}}^2}{\|h\|_{\mathcal{U}}} \xrightarrow[\|h\|_{\mathcal{U}}\mapsto 0]{} 0\,, \tag{19}$$

hence the definition of Fréchet differential is satisfied.

2. **Linearity -** The differential operator $D\mathcal{J}(u)[h]$ features both h and the associated state perturbation $\delta\rho$. To show linearity in the argument h, we need to investigate the relationship between h and $\delta\rho$. To this end, we derive the Fokker Planck equation for the state perturbation:

$$\partial_t\delta\rho=\beta^{-1}\Delta\delta\rho+\nabla\cdot(\delta\rho((\nabla W*\rho)+\nabla U+u))+\nabla\cdot(\rho h)\,, \tag{20}$$

which is linear in both $\delta\rho$ and h, so we can rewrite it as $\partial_t\delta\rho=\mathcal{L}(\delta\rho)+\nabla\cdot(\rho h)$, where $\mathcal{L}$ is a linear operator. We introduce the solution operator S, so that $\delta\rho=S(h)$, which we will now show to be linear. Let $\delta\rho_1=S(h_1)$ and $\delta\rho_2=S(h_2)$, and consider $\delta\rho_{1,2}=a\delta\rho_1+b\delta\rho_2$, for $a,b\in\mathbb{R}$. Substituting into the PDE (20), by linearity of $\mathcal{L}$

and $\nabla\cdot$ we have

$$\partial_t \delta\rho_{1,2} = \mathcal{L}(\rho_{1,2}) + \nabla \cdot (\rho(ah_1 + bh_2)),$$

hence $\delta\rho_{1,2}$ satisfies the PDE for the control perturbation $ah_1 + bh_2$ and $S(ah_1 + bh_2) = aS(h_1) + bS(h_2)$. This implies that $\delta\rho$ depends linearly on h. We can conclude that $D\mathcal{J}(u)[h]$ is indeed linear in h, as both h and $\delta\rho$ appear linearly in the expression for $D\mathcal{J}(u)[h]$.

3. **Boundedness -** We need to show that there exists a $C > 0$ such that

$$|D\mathcal{J}(u)[h]| \leq C\|h\|_{\mathcal{U}}.$$

This reads

$$|D\mathcal{J}(u)[h]| \leq \left| \int_0^T \int_\Omega (\rho - \rho_\infty)\, \delta\rho \, dx \right| + \gamma \left| \int_0^T \int_\Omega u\, h \, dx \right|.$$

By Cauchy-Schwarz inequality and the estimate (18), we can bound the first term as

$$\begin{aligned} \left| \int_0^T \int_\Omega (\rho - \rho_\infty)\, \delta\rho \, dx \right| &\leq \|\rho - \rho_\infty\|_{L_2(0,T;L_2(\Omega))}\, \|\delta\rho\|_{L_2(0,T;L_2(\Omega))} \\ &\leq \|\rho - \rho_\infty\|_{L_2(0,T;L_2(\Omega))}\, C\|h\|_{\mathcal{U}} \\ &\leq C_1\|h\|_{\mathcal{U}} \end{aligned}$$

due to the regularity of ρ and ρ_∞. Similarly, the second term reads

$$\gamma \left| \int_0^T \int_\Omega u\, h \, dx \right| \leq \gamma \|u\|_{\mathcal{U}} \|h\|_{\mathcal{U}} \leq C_2 \|h\|_{\mathcal{U}}\,. \tag{21}$$

Combining the estimates for both terms, we obtain

$$|D\mathcal{J}(u)[h]| \leq (C_1 + C_2)\|h\|_{\mathcal{U}}$$

which concludes the proof.

A.2 Derivation of the continuous optimality system

In this section explicitely derive the PMP optimality system associated with the continuous OCP (4.38)-(4.39).

We start by considering the Frechèt derivative of the cost functional $\mathcal{J}$ in a direction $h \in L_2(\mathbb{R}^2, \mathbb{R}^2)$

$$D\mathcal{J}(u)[h] = \int_0^T \int_\Omega (\rho - \rho_\infty)\dot{\rho} + \gamma u h \, dt \,, \tag{22}$$

where we define

$$\dot{\rho} := D_u \rho[u] \tag{23}$$

which solves the following FP equation

$$\partial_t \dot{\rho} = \sigma^2 \Delta \dot{\rho} + div\Big(\dot{\rho}[\nabla W * \rho + \nabla V + u]\Big) + div\Big(\rho[\nabla W * \dot{\rho}] + h\Big) \,. \tag{24}$$

By Riesz representation theorem, given $\mathcal{L} : L_2 \to \mathbb{R}$ is a linear functional, there exists $v \in L_2$ such that $\mathcal{L}(u) = \langle v, u \rangle_{L_2}$. Since $D_u\rho[u]$ is a linear functional we can compute $\dot{\rho}$ by finding a suitable function v.

With this objective in mind, we can rewrite the derivative in (22) as

$$\int_0^T \int_\Omega (\rho - \rho_\infty)\dot{\rho} \, dt = \int_0^T \int_\Omega \partial_t \dot{\rho} - \sigma^2 \Delta \dot{\rho} - div\Big(\dot{\rho}[\nabla W * \rho + \nabla V + u]\Big) - div\Big(\rho[\nabla W * \dot{\rho}]\Big) dt \,, \tag{25}$$

so that

$$\int_0^T \int_\Omega (\rho - \rho_\infty)\dot{\rho} \, dt = \int_0^T \int_\Omega \lambda \, div(\rho h) dt \tag{26}$$

for λ adjoint operator.

An integration by parts of the RHS of (25) yields

$$\int_0^T \int_\Omega (\rho - \rho_\infty)\dot{\rho} \, dt = \int_0^T \int_\Omega \Big(- \partial_t \lambda - \sigma^2 \Delta \lambda + \nabla \lambda \cdot \big(\nabla W * \rho + \nabla V + u\big) - \nabla W \star (\nabla \lambda \rho)\Big) \dot{\rho} \, dt \,, \tag{27}$$

with

$$f \star g(y) := \int_{\Omega} f(x-y)g(x)dx\,. \tag{28}$$

From this, the adjoint operator λ follows backward dynamics defined as

$$-\partial_t\lambda - \sigma^2\Delta\lambda + \nabla\lambda\cdot(\nabla W * \rho + \nabla V + u) - \nabla W \star (\nabla\lambda\rho) = (\rho - \rho_\infty)\,. \tag{29}$$

We conclude by going back to the definition of Frechèt derivative of the cost functional $\mathcal{J}$, that by (26) reads

$$\begin{aligned} D\mathcal{J}(u)[h] &= \int_0^T\int_\Omega \lambda\, div(\rho h)\, dt + \int_0^T\int_\Omega \gamma u h\, dt\,, \\ &= \int_0^T\int_\Omega \langle -\nabla\lambda\rho + \gamma u, h\rangle_{L_2}\, dt\,. \end{aligned} \tag{30}$$

Every optimal control signal u satisfies

$$D\mathcal{J}(u)[h] = 0 \quad \forall\, h \in L_2(\mathbb{R}^2, \mathbb{R}^2)\,, \tag{31}$$

which translates into the sufficient optimality conditions

$$-\nabla\lambda\rho + \gamma u = 0\,. \tag{32}$$

This concludes the informal derivation of the optimality system (4.42) for the stabilizing optimal control problem at the continuous level.

A.3 Consistency of the discretized optimality condition

In this section we prove consistency between the two approaches to solving the PDE-constrained optimal control problem (4.38). The crucial distinction lies at the stage at which discretization is applied to the solution process:

- **optimize-then-discretize** - The optimality system is derived at the continuous,

infinite-dimensional level, as in (4.41). The functions ρ, u, and λ are then discretized and the two point boundary value problem is rewritten in finite-dimensional form.

- **discretize-then-optimize** - Discretization is performed at the PDE level, resulting in a finite-dimensional optimal control problem. The optimality conditions for this approach are given by (4.56 - 4.59).

Consistency between these approaches can be shown by comparing the first order optimality conditions in (4.59) and (4.41). The explicit expression of (4.59) reads

$$\gamma\, \mathbf{u}_j^\top M - \mathbf{p}^\top g_j(\mathbf{a}) = \gamma \int u_j^L\, \psi_m\, d\Omega - \frac{1}{\int \psi_m \psi_m d\Omega} \int \rho^L\, \frac{\partial \lambda^L}{\partial x_j}\, \psi_m\, d\Omega$$

for $j = 1, \ldots, d$ and $m = 1, \ldots, L$ and $\lambda^L(t,x) := \sum_{n=1}^{L} \lambda_n(t)\psi_n(x)$. Noting that

- since $u(t,\cdot) \in L_2(\Omega) \subset L_\infty(\Omega)$, and $\rho, \lambda \in \mathcal{W}(0,T)$ implies $\rho(t,\cdot) \in L_2(\Omega)$, $\nabla\lambda \in L_2(\Omega)$, as a direct result of Paserval's identity we have convergence of the truncated Fourier series ρ^L, u^L, $\frac{\partial \lambda^L}{\partial x_j}$;
- $\frac{1}{\int \psi_m \psi_m d\Omega}$ is constant and independent of the number of Fourier modes L, hence it represents a scaling that does not affect consistency;

we have

$$\gamma\, \mathbf{u}_j^\top M - \mathbf{p}^\top g_j(\mathbf{a}) \underset{L_2}{\longrightarrow} \int (-\nabla\lambda\rho + \gamma u)\psi_m\, d\Omega$$

hence the discrete optimality condition converges in L_2 in variational sense to it's continuous counterpart, ensuring consistency of the discretization scheme.

Supplementary material to Mesoscopic scale chapter

B.4 Proof of Theorem 5.1.2

Step 1

Let us introduce a test function $\varphi \in C_0^2(\mathbb{R}^d \times \mathbb{R}^d)$, is the state space hosting the N_a agents of the system. We consider the following weak formulation of the Boltzmann equation (5.8)

$$\begin{aligned} \frac{d}{dt}\langle f, \varphi\rangle + \langle f, v \cdot \nabla_x \varphi\rangle &= \lambda \langle \mathcal{Q}_{\eta,u}(\rho,\rho), \varphi\rangle \\ &= \lambda \iint\limits_{\mathbb{R}^{2d}\times\mathbb{R}^{2xd}} (\varphi(x,v') - \varphi(x,v))\rho(t,x_*,v_*)\, dx_*\, dv_*\, dx\, dv \end{aligned} \tag{33}$$

Step 2

By definition of the binary post-interaction dynamics (5.6), we can explicitly write $v' - v$ as

$$v' - v = \eta\, P(x,x_*)(v_* - v) + \eta\, u_\eta(x,v,x_*,v_*) =: \eta F_\eta(x,v,x_*,v_*) \tag{34}$$

where we introduce the function $F_\eta(x,v,x_*,v_*)$ for the controlled dynamics binary dynamics. Moreover, we expand $\varphi(v')$ inside the operator (33) in Taylor series of $v' - v$ up to the second order, obtaining

$$\varphi(x,v') - \varphi(x,v) = \eta\, F_\eta(x,v,x_*,v_*) \cdot \nabla_v\, \varphi(x,v) + R^\varphi(\eta^2) \tag{35}$$

where the term $R^{\varphi}(\eta^2)$ represents the reminder of the Taylor expansion as follows

$$\begin{aligned} R^{\varphi}(\eta^2) &= \sum_{i,j=1}^{d} \left(\left(\partial_v^{(i,j)} \varphi(x,v) - \partial_v^{(i,j)} \varphi(x,\tilde{v}) \right) (v'-v)_i (v'-v)_j \right) \\ &= \eta^2 \sum_{i,j=1}^{d} \left(\left(\partial_v^{(i,j)} \varphi(x,v) - \partial_v^{(i,j)} \varphi(x,\tilde{v}) \right) (F_\eta)_i (F_\eta)_j \right) =: \eta^2 \tilde{R}_\eta^{\varphi}(x,v,x_*,v_*) \end{aligned} \tag{36}$$

where $\tilde{v} = (1-\theta)v' + \theta v$ for some $\theta \in [0,1]$ and we use the multi-index notation for the second order partial derivatives of φ.

Step 3

Embedding (35) into the interacting operator for the weak form (33), and introducing the quasi invariant scaling (5.14), i.e. $\lambda = 1/\varepsilon$, $\eta = \varepsilon$ we have

$$\frac{1}{\varepsilon} \langle \mathcal{Q}_{\varepsilon,u}(\rho,\rho), \varphi \rangle = \iint\limits_{\mathbb{R}^{2d}\times\mathbb{R}^{2d}} F_\varepsilon(x,v,x_*,v_*) \cdot \nabla_v \varphi(x,v)\, f\, f_*\, dx_*\, dv_*\, dx\, dv + \varepsilon \mathcal{R}_\varepsilon^{\varphi}, \tag{37}$$

with reminder term

$$\mathcal{R}_\varepsilon^{\varphi} = \iint\limits_{\mathbb{R}^{2d}\times\mathbb{R}^{2d}} \tilde{R}_\varepsilon^{\varphi}(x,v,x_*,v_*)\, f f_*\, dx_*\, dv_*\, dx\, dv. \tag{38}$$

For $\varepsilon \to 0$, assuming that the reminder vanishes to zero, and integrating by parts the scaled weak form of the Povzner-Boltzmann model (33) we have

$$\left\langle \partial_t \rho + v \cdot \nabla_x \rho + \nabla_v \cdot \left[f \int\limits_{\mathbb{R}^{2d}} \left(P(x,x_*)(v_*-v) + u(x,v,x_*,v_*) \right) \rho(t,x_*,v_*)\, dx_*\, dv_* \right], \varphi \right\rangle = 0$$

for every $\varphi \in C^2(\mathbb{R}^d \times \mathbb{R}^d)$. Hence, in strong form, we retrieve consistency of the scaled Povzner-Boltzmann model (5.8) with controlled mean field model (5.15). Hence, in the next step we show that the reminder is bounded and the previous limit hold true for $\varepsilon \to 0$.

Step 4

Finally we provide estimate on the reminder of the Taylor expansion, to show that such term vanishes for $\varepsilon \to 0$. We observe that

$$\|\mathcal{R}_\varepsilon^\varphi\| \leq \frac{C}{2} \iint_{\mathbb{R}^{2d}\times\mathbb{R}^{2d}} \|F_\varepsilon(x,v,x_*,v_*)\|^2 \, \rho(t,x,v)\rho(t,x_*,v_*)\, dx\, dv\, dx_*\, dv_*$$

where C is the bounding constant of $C_0^2(\mathbb{R}^d \times \mathbb{R}^d)$, and since $F_\varepsilon(x,v,x_*,v_*) \in L^2_{loc}$ we can conclude that the quasi-invariant limit holds true.

□

Remark 8. *The proof provided here can be readapted seamlessly in the case of the full state binary dynamics (5.17) by assuming that $F_\eta(s,s_*) := G(s,s_*) + Hu(s,s_*) \in L^2_{loc}(\mathbb{R}^{2d} \times \mathbb{R}^{2d})$. More precisely, in Step 1, we consider the test function $\Phi \in C_0^2(\mathbb{R}^{2d})$ and the weak form of (5.19) as follows*

$$\frac{d}{dt}\langle f, \Phi\rangle = \lambda \iint\limits_{\mathbb{R}^{2d}\times\mathbb{R}^{2d}} (\Phi(s') - \Phi(s))\rho(t,s_*)\rho(t,s)\, ds_* ds, \tag{39}$$

where $s' = (x', v')$ is the post-collisional state in (5.17). In Step 2, by Taylor's expansion around $s = (x,v)$, we retrieve

$$\Phi(s') - \Phi(s) = \eta\,(G(s,s_*) + Hu(s,s_*)) \cdot \nabla_s \Phi(s) + \eta^2 \tilde{R}_\eta^\Phi, \tag{40}$$

with reminder term defined now as

$$\tilde{R}_\eta^\Phi(s,s_*) := \sum_{i,j=1}^{2d} \left(\left(\partial_v^{(i,j)}\Phi(s) - \partial_v^{(i,j)}\Phi(\tilde{s}) \right) (F_\eta(s,s_*))_i (F_\eta(s,s_*))_j \right), \tag{41}$$

where $\tilde{s} = (1-\theta)s + \theta s'$, $\theta \in [0,1]$, similarly to (36). Then in Step 3, introducing the scaling (5.14) and using (40) in (39) we have

$$\frac{d}{dt}\langle f, \Phi\rangle = \iint\limits_{\mathbb{R}^{2d}\times\mathbb{R}^{2d}} \left((G(s,s_*) + Hu(s,s_*)) \cdot \nabla_s \Phi(s) \right) \rho(t,s_*)\rho(t,s)\, ds_* ds + \varepsilon \mathcal{R}_\varepsilon^\Phi, \tag{42}$$

where the reminder $\mathcal{R}_\varepsilon^\Phi$ is defined as in (38), as integrating $\tilde{R}_\varepsilon^\Phi(s, s_*)$. Thus, integrating by parts, we have that for any test function Φ is

$$\left\langle \partial_t \rho + \nabla_s \cdot \left(f \int_{\mathbb{R}^{2d}} (G(s, s_*) + Hu(s, s_*))\, \rho(s_*)\, ds_* \right), \Phi \right\rangle = 0. \tag{43}$$

Finally, the reminder vanishes in the limit $\varepsilon \to 0$ following the same argument of Step 4.

B.5 Asymptotic constrained symmetric Nanbu algorithm

Algorithm 8:

$\{(x_i^0, v_i^0)\}_{i=1}^{N_s} \sim \rho^0$ *i.i.d.*; N_s `samples from the initial distribution`

for $h = 0, ..., N_T - 1$, $t_n = n \cdot \Delta t$ **do**

  select $\{(i_k, j_k)\}_{k=1}^{N_s/2}$ random pairs of agents without repetitions

  for $k = 1, ..., N_s/2$ **do**

    Compute $(v_{i_k}^{n+1}, v_{j_k}^{n+1})$ according to (5.29) ; `interaction`

  for $i = 1, ..., N_s$ **do**

    Compute x_i^{n+1} according to (5.25) ; `transport`

For the generalized binary dynamics (5.17) an analogous stochastic simulation technique can be designed, where in this case `interaction` and `transport` are updated simultaneously considering the particle states at time t_n as follows $s_i^n = (x_i^n, v_i^n)^\top$, $s_j^n := (x_j^n, v_j^n)^\top$.

Bibliography

[1] Abhijeet, M. N. G. Mohamed, A. Sharma, and S. Chakravorty. Convexity in optimal control problems. In *Proceedings of the 63rd IEEE Conference on Decision and Control (CDC)*, pages 261–266. IEEE, 2024.

[2] M. Aduamoah, B. D. Goddard, J. W. Pearson, and J. C. Roden. Pseudospectral methods and iterative solvers for optimization problems from multiscale particle dynamics. *BIT Numerical Mathematics*, 62(4):1703–1743, 2022.

[3] M. Akian, S. Gaubert, and A. Lakhoua. The max-plus finite element method for solving deterministic optimal control problems: basic properties and convergence analysis. *SIAM J. Control Optim.*, 47(2):817–848, 2008.

[4] G. Albi, D. Balagué, J. A. Carrillo, and J. von Brecht. Stability analysis of flock and mill rings for second order models in swarming. *SIAM Journal on Applied Mathematics*, 74(3):794–818, 2014.

[5] G. Albi, N. Bellomo, L. Fermo, S.-Y. Ha, J. Kim, L. Pareschi, D. Poyato, and J. Soler. Vehicular traffic, crowds, and swarms: From kinetic theory and multiscale methods to applications and research perspectives. *Mathematical Models and Methods in Applied Sciences*, 29(10):1901–2005, 2019.

[6] G. Albi, S. Bicego, M. Herty, Y. Huang, D. Kalise, and C. Segala. Data/moment-driven approaches for fast predictive control of collective dynamics. *arXiv preprint arXiv.2402.15611*, 2024.

[7] G. Albi, S. Bicego, M. Herty, Y. Huang, D. Kalise, and C. Segala. Data/moment-driven approaches for fast predictive control of collective dynamics. *arXiv preprint arXiv:2402.15611*, 2024.

[8] G. Albi, S. Bicego, and D. Kalise. Gradient-augmented supervised learning of optimal feedback laws using state-dependent riccati equations. *IEEE Control Systems Letters*, 6:836–841, 2022.

[9] G. Albi, S. Bicego, and D. Kalise. Supervised learning for kinetic consensus control. *IFAC-PapersOnLine*, 55(30):103–108, 2022. 25th International Symposium on Mathematical Theory of Networks and Systems MTNS 2022.

[10] G. Albi, S. Bicego, and D. Kalise. Control of high-dimensional collective dynamics by deep neural feedback laws and kinetic modelling. *arXiv preprint arXiv:2404.02825*, 2024.

[11] G. Albi, M. Bongini, E. Cristiani, and D. Kalise. Invisible control of self-organizing agents leaving unknown environments. *SIAM Journal on Applied Mathematics*, 76(4):1683–1710, 2016.

[12] G. Albi, E. Calzola, and G. Dimarco. A data-driven kinetic model for opinion dynamics with social network contacts. *European Journal of Applied Mathematics*, 36(2):264–290, 2025.

[13] G. Albi, Y.-P. Choi, M. Fornasier, and D. Kalise. Mean field control hierarchy. *Applied Mathematics and Optimization*, 76(1):93–135, 2017.

[14] G. Albi, G. Dimarco, F. Ferrarese, and L. Pareschi. Instantaneous control strategies for magnetically confined fusion plasma. *Journal of Computational Physics*, 527:113804, 2025.

[15] G. Albi, M. Herty, D. Kalise, and C. Segala. Moment-driven predictive control of mean-field collective dynamics. *SIAM Journal on Control and Optimization*, 60(2):814–841, 2022.

[16] G. Albi, M. Herty, and L. Pareschi. Kinetic description of optimal control problems and applications to opinion consensus. *Commun. Math. Sci.*, 13(6):1407–1429, 2015.

[17] G. Albi and L. Pareschi. Binary interaction algorithms for the simulation of flocking and swarming dynamics. *Multiscale Modeling & Simulation*, 11(1):1–29, 2013.

[18] G. Albi, L. Pareschi, and M. Zanella. Boltzmann-type control of opinion consensus through leaders. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 372(2028):20140138, 2014.

[19] D. Aldous. Interacting particle systems as stochastic social dynamics. *Bernoulli*, 19(4):1122 – 1149, 2013.

[20] A. Alessio and A. Bemporad. A survey on explicit model predictive control. *Nonlinear Model Predictive Control: Towards New Challenging Applications*, pages 345–369, 2009.

[21] G. Aletti, G. Naldi, and G. Toscani. First order continuous models of opinion formation. *SIAM Journal on Applied Mathematics*, 67:837–853, 2007.

[22] A. Alla, M. Falcone, and L. Saluzzi. An efficient DP algorithm on a tree-structure for finite horizon optimal control problems. *SIAM J. Sci. Comput.*, 41(4):A2384–A2406, 2019.

[23] A. Alla, D. Kalise, and V. Simoncini. State-dependent riccati equation feedback stabilization for nonlinear pdes. *Advances in Computational Mathematics*, 49:1–32, 2021.

[24] N. Altmüller and L. Grüne. Distributed and boundary model predictive control for the heat equation. *GAMM-Mitteilungen*, 35(2):131–145, 2012.

[25] C. C. Andrea Bonfanti, Giuseppe Bruno. The challenges of the nonlinear regime for physics-informed neural networks. In *Proceedings of NeurIPS 2024*, 2024.

[26] G. C. Andreas Wernli. Suboptimal control for the nonlinear quadratic regulator problem. *Automatica*, 11(1):75–84, 1975.

[27] L. Angeli, J. Barré, M. Kolodziejczyk, and M. Ottobre. Well-posedness and stationary solutions of mckean-vlasov (s) pdes. *Journal of Mathematical Analysis and Applications*, 526(2):127301, 2023.

[28] M. Annunziato and A. Borzì. A fokker–planck control framework for stochastic systems. *EMS Surveys in Mathematical Sciences*, 5(1/2):65—98, 2018.

[29] M. Annunziato and A. Borzì. A fokker-planck control framework for multimensional stochastic processes. *Journal of Computational and Applied Mathematics*, pages 487–507, 2013.

[30] C. Appert-Rolland, A. Nicolas, A. Seyfried, A. Tordeux, and D. Ullmo. Modelling vehicle and pedestrian collective dynamics: Challenges and advancements. *arXiv preprint arXiv:2410.22896*, 2024.

[31] M. S. Aronna and F. Tröltzsch. First and second order optimality conditions for the control of fokker-planck equations. *ESAIM: COCV*, 27, 2021.

[32] B. Azmi, D. Kalise, and K. Kunisch. Optimal feedback law recovery by gradient-augmented sparse polynomial regression. *J. Mach. Learn. Res.*, 22:Paper No. 48, 32, 2021.

[33] B. Azmi and K. Kunisch. On the stabilizability of the burgers equation by receding horizon control. *SIAM Journal on Control and Optimization*, 54(3):1378–1405, 2016.

[34] R. Bailo, M. Bongini, J. A. Carrillo, and D. Kalise. Optimal consensus control of the Cucker-Smale model. *IFAC-PapersOnLine*, 51(13):1–6, 2018.

[35] H. Banks, B. Lewis, and H. Tran. Nonlinear feedback controllers and compensators: a state-dependent Riccati equation approach. *Computational Optimization and Applications*, 37:177–218, 6 2007.

[36] H. T. Banks, B. M. Lewis, and H. T. Tran. Nonlinear feedback controllers and compensators: a state-dependent riccati equation approach. *Computational Optimization and Applications*, 37:177–218, 2007.

[37] A. B. T. Barbaro, J. A. Cañizo, J. A. Carrillo, and P. Degond. Phase transitions in a kinetic flocking model of cucker–smale type. *Multiscale Modeling & Simulation*, 14(3):1063–1088, 2016.

[38] M. Bardi, I. C. Dolcetta, et al. *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*, volume 12. Springer, 1997.

[39] E. N. Barron and R. Jensen. The pontryagin maximum principle from dynamic programming and viscosity solutions to first-order partial differential equations. *Trans. Amer. Math. Soc.*, 298:635–641, 1986.

[40] J. Barzilai and J. M. Borwein. Two-Point Step Size Gradient Methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 01 1988.

[41] F. Bavaud. Equilibrium properties of the Vlasov functional: the generalized Poisson-Boltzmann-Emden equation. *Rev. Modern Phys.*, 63(1):129–148, 1991.

[42] L. E. Beaver and A. A. M. records. Constraint-driven optimal control of multiagent systems: A highway platooning case study. *IEEE Control Systems Letters*, 6:1754–1759, 2022.

[43] S. C. Beeler, H. T. Tran, and H. T. Banks. Feedback control methodologies for nonlinear systems. *J. Optim. Theory Appl.*, 107(1):1–33, 2000.

[44] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[45] A. Bensoussan, J. Frehse, and P. Yam. *Mean field games and mean field type control theory*. SpringerBriefs in Mathematics. Springer, New York, 2013.

[46] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer. Linear tracking MPC for nonlinear systems—part i: The model-based case. *IEEE Transactions on Automatic Control*, 67(9):4390–4405, 2022.

[47] L. Bertini, G. Giacomin, and K. Pakdaman. Dynamical aspects of mean field plane rotators and the Kuramoto model. *J. Stat. Phys.*, 138(1-3):270–290, 2010.

[48] B. Bertoli, B. D. Goddard, and G. A. Pavliotis. Stability of stationary states for mean field models with multichromatic interaction potentials. *arXiv preprint arXiv:2406.04884*, 2024.

[49] S. Bicego, S. Gue, D. Kalise, and N. Villamizar. Time optimal feedback control of nilpotent linear systems using neural networks. *arXiv preprint arXiv:2503.17581*, 2025.

[50] S. Bicego, D. Kalise, and G. A. Pavliotis. Computation and control of unstable steady states for mean field multiagent systems. *arXiv preprint arXiv:2406.11725*, 2024.

[51] G. Bird. *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. Number v. 42 in International Series of Monographs on Physics. Clarendon Press, 1994.

[52] D. Biro and T. Vicsek. Collective behavior and bifurcations in multi-agent systems. *Physical Review E*, 77(1):011905, 2008.

[53] K. Bisht et al. Lanes and lattice structures in a repulsive model for self-propelled agents. *arXiv preprint arXiv:2412.10577*, 2023.

[54] A. V. Bobylev and K. Nanbu. Theory of collision algorithms for gases and plasmas based on the Boltzmann equation and the Landau-Fokker-Planck equation. *Physical Review E*, 61:4576, 4 2000.

[55] F. Bongini, M. Fornasier, M. Maggioni, and L. Portinale. Inferring interaction rules from observations of evolutive systems i: The variational approach. *Mathematical Models and Methods in Applied Sciences*, 27(5):909–951, 2017.

[56] M. Bongini, M. Fornasier, F. Rossi, and F. Solombrino. Mean-field pontryagin maximum principle. *Journal of Optimization Theory and Applications*, 175(1):1–38, 2017.

[57] F. H. Branin. Widely convergent method for finding multiple solutions of simultaneous nonlinear equations. *IBM Journal of Research and Development*, 16(5):504–522, 1972.

[58] T. Breiten, K. Kunisch, and L. Pfeiffer. Control strategies for the Fokker-Planck equation. *ESAIM Control Optim. Calc. Var.*, 24(2):741–763, 2018.

[59] T. Breiten and K. K. Kunisch. Improving the convergence rates for the kinetic Fokker-Planck equation by optimal control. *SIAM J. Control Optim.*, 61(3):1557–1581, 2023.

[60] K. M. Brow and W. B. Gearhart. Deflation techniques for the calculation of further solutions of a nonlinear system. *Numerische Mathematik*, 16(4):334–342, Jan 1971.

[61] A. E. Bryson. Space trajectory optimization and l1-optimal control problems. *Acta Astronautica*, 61(1-6):50–54, 2007.

[62] M. Burger, R. Pinnau, C. Totzeck, O. Tse, and A. Roth. Instantaneous control of interacting particle systems in the mean-field limit. *J. Comput. Phys.*, 405:109181, 20, 2020.

[63] P. Cannarsa and H. Frankowska. Value function, relaxation, and transversality conditions in infinite horizon optimal control. *Journal of Mathematical Analysis and Applications*, 457(2):1188–1217, 2018. Special Issue on Convex Analysis and Optimization: New Trends in Theory and Applications.

[64] M. Caponigro, M. Fornasier, B. Piccoli, and E. Trélat. Sparse stabilization and control of alignment models. *Math. Models Methods Appl. Sci.*, 25(3):521–564, 2015.

[65] R. Carmona, J.-P. Fouque, and L.-H. Sun. Mean field games and systemic risk. *Commun. Math. Sci.*, 13(4):911–933, 2015.

[66] J. A. Carrillo, M. Fornasier, J. Rosado, and G. Toscani. Asymptotic flocking dynamics for the kinetic Cucker-Smale model. *SIAM J. Math. Anal.*, 42(1):218–236, 2010.

[67] J. A. Carrillo, R. S. Gvalani, G. A. Pavliotis, and A. Schlichting. Long-time behaviour and phase transitions for the Mckean-Vlasov equation on the torus. *Arch. Ration. Mech. Anal.*, 235(1):635–690, 2020.

[68] J. A. Carrillo, D. Kalise, F. Rossi, and E. Trélat. Controlling Swarms toward Flocks and Mills. *SIAM J. Control Optim.*, 60(3):1863–1891, 2022.

[69] C. Cercignani. *Kinetic Theory and the Boltzmann Equation.* Springer Science & Business Media, New York, 2000.

[70] G. Chai and T. jun Wang. Mixed generalized hermite–fourier spectral method for fokker–planck equation of periodic field. *Applied Numerical Mathematics*, 133:25–40, 2018. "The 20th IMACS World Congress" held in Xiamen University, Xiamen,China, December 10-14, 2016.

[71] I. Chang and J. Bentsman. Constrained discrete-time state-dependent Riccati equation technique : A model predictive control approach. *Proceedings of the IEEE Conference on Decision and Control*, pages 5125–5130, 2013.

[72] K.-S. Chao, D.-k. Liu, and C.-t. Pan. A systematic search method for obtaining multiple solutions of simultaneous nonlinear equations. *IEEE Transactions on Circuits and Systems*, 22(9):748–753, 1975.

[73] E. Charalampidis, N. Boullé, P. Farrell, and P. Kevrekidis. Bifurcation analysis of stationary solutions of two-dimensional coupled gross–pitaevskii equations using deflated continuation. *Communications in Nonlinear Science and Numerical Simulation*, 87:105255, 2020.

[74] P. Chavanis. Nonlinear mean-field fokker–planck equations and their applications in physics, astrophysics and biology. *Comptes Rendus Physique*, 7(3):318–330, 2006. Statistical mechanics of non-extensive systems.

[75] P.-H. Chavanis. The Brownian mean field model. *Eur. Phys. J. B*, 87(5):Art. 120, 33, 2014.

[76] L. Chayes and V. Panferov. The McKean-Vlasov equation in finite volume. *J. Stat. Phys.*, 138(1-3):351–380, 2010.

[77] X. Chen, L. Yang, J. Duan, and G. E. Karniadakis. Solving inverse stochastic problems from discrete particle observations using the fokker–planck equation and physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(3):B811–B830, 2021.

[78] M.-J. Chien. Searching for multiple solutions of nonlinear systems. *IEEE Transactions on Circuits and Systems*, 26(10):817–827, 1979.

[79] Y.-P. Choi, D. Kalise, J. Peszek, and A. A. Peters. A collisionless singular Cucker-Smale model with decentralized formation control. *SIAM J. Appl. Dyn. Syst.*, 18(4):1954–1981, 2019.

[80] Y. T. Chow, J. Darbon, S. Osher, and W. Yin. Algorithm for overcoming the curse of dimensionality for time-dependent non-convex Hamilton-Jacobi equations arising from optimal control and differential games problems. *J. Sci. Comput.*, 73(2-3):617–643, 2017.

[81] Y. T. Chow, J. Darbon, S. Osher, and W. Yin. Algorithm for overcoming the curse of dimensionality for time-dependent non-convex Hamilton-Jacobi equations arising from optimal control and differential games problems. *J. Sci. Comput.*, 73(2-3):617–643, 2017.

[82] Y. T. Chow, J. Darbon, S. Osher, and W. Yin. Algorithm for overcoming the curse of dimensionality for certain non-convex Hamilton-Jacobi equations, projections and differential games. *Ann. Math. Sci. Appl.*, 3(2):369–403, 2018.

[83] Y. T. Chow, J. Darbon, S. Osher, and W. Yin. Algorithm for overcoming the curse of dimensionality for state-dependent Hamilton-Jacobi equations. *J. Comput. Phys.*, 387:376–409, 2019.

[84] C. Cipriani, A. Scagliotti, and T. Wohrer. A minimax optimal control approach for robust neural odes. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):5923–5936, 2023.

[85] F. H. Clarke and R. B. Vinter. The relationship between the maximum principle and dynamic programming. *SIAM Journal on Control and Optimization*, 25(5):1291–1311, 1987.

[86] J. Clive, E. Flintham, and V. Savolainen. Same-sex sociosexual behaviour is widespread and heritable in male rhesus macaques. *Nature Ecology & Evolution*, 7:1287–1301, 2023.

[87] H. Cohn and R. Kulsrud. The stellar distribution around a black hole: numerical integration of the Fokker-Planck equation. *Astrophysical Journal*, 226:1087–1108, Dec. 1978.

[88] S. Cordier, L. Pareschi, and G. Toscani. On a kinetic model for a simple market economy. *Journal of Statistical Physics*, 120(1):253–277, 2005.

[89] E. Cristiani, B. Piccoli, and A. Tosin. *Multiscale modeling of pedestrian dynamics*, volume 12 of *MS&A. Model. Simul. Appl.* Springer, Cham, 2014.

[90] F. Cucker and S. Smale. On the emergence of equilibria in collective dynamics of 'small' particles. *IEEE Transactions on Automatic Control*, 49(10):1676–1681, 2004.

[91] F. Cucker and S. Smale. Emergent behavior in flocks. *IEEE Trans. Automat. Control*, 52:852–862, 2007.

[92] J. Darbon, G. P. Langlois, and T. Meng. Overcoming the curse of dimensionality for some Hamilton-Jacobi partial differential equations via neural network architectures. *Res. Math. Sci.*, 7(3):Paper No. 20, 50, 2020.

[93] J. Darbon and S. Osher. Algorithms for overcoming the curse of dimensionality for certain Hamilton-Jacobi equations arising in control theory and elsewhere. *Res. Math. Sci.*, 3:Paper No. 19, 26, 2016.

[94] D. A. Dawson. Critical dynamics and fluctuations for a mean-field model of cooperative behavior. *J. Statist. Phys.*, 31(1):29–85, 1983.

[95] P. Degond, M. Herty, and J. Liu. Mean-field games and model predictive control. *Communications in Mathematical Sciences*, 15(5):1403 —- 1422, 2014.

[96] P. Degond, J.-G. Liu, and C. Ringhofer. Large-scale dynamics of mean-field games driven by local Nash equilibria. *J. Nonlinear Sci.*, 24(1):93–115, 2014.

[97] M. Delgadino, R. Gvalani, and G. Pavliotis. On the diffusive-mean field limit for weakly interacting diffusions exhibiting phase transitions. *Archive for Rational Mechanics and Analysis*, 2021.

[98] M. G. Delgadino, R. S. Gvalani, G. A. Pavliotis, and S. A. Smith. Phase Transitions, Logarithmic Sobolev Inequalities, and Uniform-in-Time Propagation of Chaos for Weakly Interacting Diffusions. *Comm. Math. Phys.*, 401(1):275–323, 2023.

[99] G. Dimarco, Q. Li, L. Pareschi, and B. Yan. Numerical methods for plasma physics in collisional regimes. *Journal of Plasma Physics*, 81(1):305810106, 2015.

[100] S. Dolgov, D. Kalise, and K. K. Kunisch. Tensor decomposition methods for high-dimensional Hamilton-Jacobi-Bellman equations. *SIAM J. Sci. Comput.*, 43(3):A1625–A1650, 2021.

[101] S. Dolgov, D. Kalise, and L. Saluzzi. Optimizing semilinear representations for state-dependent riccati equation-based feedback control. *IFAC-PapersOnLine*, 55(30):510–515, 2022. 25th International Symposium on Mathematical Theory of Networks and Systems MTNS 2022.

[102] S. Dolgov, D. Kalise, and L. Saluzzi. Data-driven tensor train gradient cross approximation for hamilton-jacobi-bellman equations. *SIAM Journal on Scientific Computing*, 45(5):A2153–A2184, 2023.

[103] S. V. Dolgov, B. N. Khoromskij, and I. V. Oseledets. Fast solution of parabolic problems in the tensor train/quantized tensor train format with initial application to the fokker–planck equation. *SIAM Journal on Scientific Computing*, 34(6):A3016–A3038, 2012.

[104] K. Dressler. Stationary solutions of the Vlasov-Fokker-Planck equation. *Math. Methods Appl. Sci.*, 9(2):169–176, 1987.

[105] B. During, P. Markowich, J.-F. Pietschmann, and M.-T. Wolfram. Boltzmann and fokker–planck equations modelling opinion formation in the presence of strong leaders. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 465(2112):3687–3708, 2009.

[106] A. S. Dutka, A. W. Ordys, and M. J. Grimble. Optimized discrete-time state dependent Riccati equation regulator. *Proceedings of the 2005, American Control Conference*, 4:2293–2298, 2005.

[107] J. R. Dyer, A. Johansson, D. Helbing, I. D. Couzin, and J. Krause. Leadership, consensus decision making and collective behaviour in humans. *Philos. Trans. Roy. Soc. B*, 364(1518):781–789, 2009.

[108] B. Düring, J. Franceschi, M.-T. Wolfram, and M. Zanella. Breaking consensus in kinetic opinion formation models on graphons. *SIAM Rev.*, page 577–621, jan 2024.

[109] W. E and B. Yu. The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. *Commun. Math. Stat.*, 6(1):1–12, 2018.

[110] M. Falcone and R. Ferretti. *Semi-Lagrangian Approximation Schemes for Linear and Hamilton–Jacobi Equations*, volume 13 of *MOS-SIAM Series on Optimization*. Society for Industrial and Applied Mathematics (SIAM), 2014.

[111] P. E. Farrell, A. Birkisson, and S. W. Funke. Deflation techniques for finding distinct solutions of nonlinear partial differential equations. *SIAM Journal on Scientific Computing*, 37(4):A2026–A2045, 2015.

[112] E. Fausti and A. Sojmark. An interacting particle system for the front of an epidemic advancing through a susceptible population. *arXiv preprint arXiv:2210.09286*.

[113] M. Fazel, T. Lian, M. Maggioni, and M. Mesbahi. Learning representations and dynamics for multiagent systems from data. *Proceedings of the IEEE*, 109(5):693–719, 2021.

[114] A. Fleig and R. Guglielmi. Optimal control of the fokker–planck equation with space-dependent controls. *Journal of Optimization Theory and Applications*, 174(2):408–427, 2017.

[115] W. H. Fleming and H. M. Soner. Mean field limits for systems of agents with interactions. *SIAM Journal on Control and Optimization*, 53(4):1803–1836, 2015.

[116] M. Fornasier and F. Solombrino. Mean-field Optimal Control. *ESAIM: Control, Optimisation and Calculus of Variations*, 20(4):1123–1152, 2014.

[117] M. Fornasier and F. Solombrino. Mean-field optimal control. *ESAIM Control Optim. Calc. Var.*, 20(4):1123–1152, 2014.

[118] T. D. Frank. *Nonlinear Fokker-Planck Equations (Section 5.3.3)*. Springer Berlin, Heidelberg, 2005.

[119] G. Freudenthaler and T. Meurer. PDE-based multi-agent formation control using flatness and backstepping: analysis, design and robot experiments. *Automatica*, 115:108897, 13, 2020.

[120] M. Gabrié, G. M. Rotskoff, and E. Vanden-Eijnden. Adaptive Monte Carlo augmented with normalizing flows. *Proc. Natl. Acad. Sci. USA*, 119(10):Paper No. e2109420119, 9, 2022.

[121] J. Garcke and A. Kröner. Suboptimal feedback control of PDEs by solving HJB equations on adaptive sparse grids. *J. Sci. Comput.*, 70(1):1–28, 2017.

[122] J. Garnier, G. Papanicolaou, and T.-W. Yang. Large deviations for a mean field model of systemic risk. *SIAM J. Financial Math.*, 4(1):151–184, 2013.

[123] J. Garnier, G. Papanicolaou, and T.-W. Yang. Consensus convergence with stochastic effects. *Vietnam J. Math.*, 45(1-2):51–75, 2017.

[124] V. Giunta, T. Hillen, M. A. Lewis, and J. R. Potts. Detecting minimum energy states and multi-stability in nonlocal advection–diffusion models for interacting species. *Journal of Mathematical Biology*, 85(5):56, 2022.

[125] B. D. Goddard, B. Gooding, H. Short, and G. A. Pavliotis. Noisy bounded confidence models for opinion dynamics: the effect of boundary conditions on phase transitions. *IMA Journal of Applied Mathematics*, 11 2021. hxab044.

[126] S. N. Gomes, G. A. Pavliotis, and U. Vaes. Mean Field Limits for Interacting Diffusions with Colored Noise: Phase Transitions and Spectral Numerical Methods. *Multiscale Model. Simul.*, 18(3):1343–1370, 2020.

[127] J. Gómez-Serrano, C. Graham, and J.-Y. Le Boudec. The bounded confidence model of opinion dynamics. *Math. Models Methods Appl. Sci.*, 22(2):1150007, 46, 2012.

[128] X. Gong, M. Herty, B. Piccoli, and G. Visconti. Crowd dynamics: Modeling and control of multiagent systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 6(Volume 6, 2023):261–282, 2023.

[129] A. Gooran Orimi, S. Effati, and M. H. Farahi. Approximate solution of the Hamilton-Jacobi-Bellman equation. *J. Math. Model.*, 10(1):71–91, 2022.

[130] A. Gorodetsky, S. Karaman, and Y. Marzouk. High-dimensional stochastic optimal control using continuous tensor decompositions. *Int. J. Robot. Res.*, 37:340–377, 2018.

[131] L. Grüne and J. Pannek. Infinite horizon optimal control. In *Nonlinear Model Predictive Control*, Communications and Control Engineering, pages 71–90. Springer, Cham, 2017.

[132] R. Gvalani, G. Pavliotis, and A. Schlichting. Coarsening dynamics for mean-field interacting particle systems. preprint, 2024.

[133] S.-Y. Ha, T. Ha, and J.-H. Kim. Emergent behavior of a cucker-smale type particle model with nonlinear velocity couplings. *IEEE Transactions on Automatic Control*, 55(7):1679–1683, 2010.

[134] J. Hahn, U. Kruger, and T. F. Edgar. Application of model reduction for model predictive control. *IFAC Proceedings Volumes*, 35(1):393–398, 2002. 15th IFAC World Congress.

[135] J. Han, A. Jentzen, and W. E. Solving high-dimensional partial differential equations using deep learning. *Proc. Natl. Acad. Sci. USA*, 115(34):8505–8510, 2018.

[136] Y. Han, A. Hegyi, Y. Yuan, S. Hoogendoorn, M. Papageorgiou, and C. Roncoli. Resolving freeway jam waves by discrete first-order model-based predictive control of variable speed limits. *Transportation Research Part C: Emerging Technologies*, 77:405–420, 2017.

[137] L. Helfmann, N. Djurdjevac Conrad, P. Lorenz-Spreen, and C. Schütte. Modelling opinion dynamics under the impact of influencer and media strategies. *Scientific Reports*, 13(1):19375, 2023.

[138] M. Herty and L. Pareschi. Fokker-planck asymptotics for traffic flow models. *Kinetic and Related Models*, 3(1):165–179, 2010.

[139] M. Herty, L. Pareschi, and S. Steffensen. Mean–field control and Riccati equations. *Networks and Heterogeneous Media*, 10(3):699–715, 2015.

[140] R. Herzog and K. Kunisch. Algorithms for pde-constrained optimization. *GAMM-Mitteilungen*, 33(2):163–176, 2010.

[141] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1):269–296, 2020.

[142] S. Hovland, K. Willcox, and J. T. Gravdahl. MPC for large-scale systems via model reduction and multiparametric quadratic programming. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 3418–3423, 2006.

[143] A. Jones and A. Astolfi. On the solution of optimal control problems using parameterized state-dependent Riccati equations. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 1098–1103, 2020.

[144] U. Kalabic, E. A. Theodorou, and P. Tsiotras. Inverse optimal control with constraints. *Automatica*, 50(1):137–147, 2014.

[145] D. Kalise, S. Kundu, and K. Kunisch. Robust feedback control of nonlinear pdes by numerical approximation of high-dimensional hamilton-jacobi-isaacs equations. *SIAM J. Appl. Dyn. Syst.*, 19:1496–1524, 2020.

[146] D. Kalise and K. Kunisch. Polynomial approximation of high-dimensional Hamilton-Jacobi-Bellman equations and applications to feedback control of semilinear parabolic PDEs. *SIAM J. Sci. Comput.*, 40(2):A629–A652, 2018.

[147] W. Kang, Q. Gong, and T. Nakamura-Zimmerer. Algorithms of Data Development For Deep Learning and Feedback Design, 2019. arXiv preprint:1912.00492.

[148] W. Kang, Q. Gong, T. Nakamura-Zimmerer, and F. Fahroo. Algorithms of data generation for deep learning and feedback design: a survey. *Phys. D*, 425:Paper No. 132955, 10, 2021.

[149] W. Kang and L. Wilcox. *A Causality Free Computational Method for HJB Equations with Application to Rigid Body Satellites*. 2015. AIAA Guidance, Navigation, and Control Conference.

[150] W. Kang and L. C. Wilcox. Mitigating the curse of dimensionality: sparse grid characteristics method for optimal feedback control and HJB equations. *Comput. Optim. Appl.*, 68(2):289–315, 2017.

[151] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[152] J. Kocijan and R. Murray-Smith. *Nonlinear Predictive Control with a Gaussian Process Model*, pages 185–200. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[153] J. Kocijan, R. Murray-Smith, C. Rasmussen, and B. Likar. Predictive control with gaussian process models. In *The IEEE Region 8 EUROCON 2003. Computer as a Tool.*, volume 1, pages 352–356 vol.1, 2003.

[154] S. Krishnan, A. Garg, R. Liaw, L. Miller, and D. Fox. Shaping reward functions for reinforcement learning with inverse optimal control. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[155] S. N. e. a. Kumar P.A., Manoj N. Uav swarm objectives: A critical analysis and comprehensive review. *SN Computer Science*, 2024.

[156] K. Kunisch and D. Walter. Semiglobal optimal feedback stabilization of autonomous systems via deep neural network approximation. *ESAIM Control Optim. Calc. Var.*, 27:Paper No. 16, 59, 2021.

[157] Y. Kuramoto. Self-entrainment of a population of coupled non-linear oscillators. In *International Symposium on Mathematical Problems in Theoretical Physics*, pages 420–422. Springer, 1975.

[158] H. Kwakernaak and R. Sivan. *Linear Optimal Control Systems*. Wiley-Interscience, New York, 1972.

[159] Körner, Jacob and Borzì, Alfio. Second-order analysis of fokker–planck ensemble optimal control problems. *ESAIM: COCV*, 28:77, 2022.

[160] D. Lacker and L. Le Flem. Sharp uniform-in-time propagation of chaos. *Probab. Theory Related Fields*, 187(1-2):443–480, 2023.

[161] S. Lale, K. Azizzadenesheli, B. Hassibi, and A. Anandkumar. Model learning predictive control in nonlinear dynamical systems. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 757–762, 2021.

[162] J.-M. Lasry and P.-L. Lions. Mean field games. *Japanese Journal of Mathematics*, 2(1):229–260, 2007.

[163] T. Lelièvre, F. Nier, and G. A. Pavliotis. Optimal non-reversible linear drift for the convergence to equilibrium of a diffusion. *J. Stat. Phys.*, 152(2):237–274, 2013.

[164] Li, Wang, and Li. An efficient spectral trust-region deflation method for multiple solutions. *Journal of Scientific Computing*, 95:105–255, 2020.

[165] Q. Li and M. Maggioni. Learning interaction kernels in heterogeneous systems of agents from trajectory data. *Journal of Machine Learning Research*, 22(181):1–63, 2021.

[166] X. Li, Z. Sun, H. Wang, S. Li, and J. Wang. A predictive cooperative collision avoidance for multi-robot systems using control barrier function. *arXiv preprint arXiv:2501.10447*, 2025.

[167] D. Limon, J. Calliess, and J. Maciejowski. Learning-based nonlinear model predictive control. *IFAC-PapersOnLine*, 50(1):7769–7776, 2017. 20th IFAC World Congress.

[168] S. Liu, W. Li, H. Zha, and H. Zhou. Neural parametric fokker–planck equation. *SIAM Journal on Numerical Analysis*, 60(3):1385–1449, 2022.

[169] Z. Liu, Y. Yang, and Q. Cai. Neural network as a function approximator and its application in solving differential equations. *Appl. Math. Mech. (English Ed.)*, 40(2):237–248, 2019.

[170] F. Lu, M. Maggioni, and S. Tang. Learning interaction kernels in heterogeneous systems of agents from multiple trajectories. *Found Comput Math*, 22, 2022.

[171] V. Lucarini, G. A. Pavliotis, and N. Zagli. Response theory and phase transitions for the thermodynamic limit of interacting identical systems. *Proc. A.*, 476(2244):20200688, 27, 2020.

[172] E. Maddalena, C. da S. Moraes, G. Waltrich, and C. Jones. A neural network architecture to learn explicit MPC controllers from data. *IFAC-PapersOnLine*, 53(2):11362–11367, 2020. 21st IFAC World Congress.

[173] M. Maggioni, J. Mercer, and A. S. Rinaldo. Graph-based methods for learning from high-dimensional data. *Annals of Statistics*, 48(5):2820–2851, 2020.

[174] D. Marin and M. Rousset. From kinetic theory to continuum models: Boltzmann and fluid dynamics. *Mathematical Models and Methods in Applied Sciences*, 17(1):49–74, 2007.

[175] N. Martzel and C. Aslangul. Mean-field treatment of the many-body Fokker-Planck equation. *J. Phys. A*, 34(50):11225–11240, 2001.

[176] T. Matsumoto and T. Kubo. An ordinary differential equation model for fish schooling. *arXiv preprint arXiv:1508.05597*, 2015.

[177] T. Matsumoto and T. Kubo. A stochastic differential equation model for predator-avoidance fish schooling. *Mathematical Biosciences*, 368:108357, 2023.

[178] S. McNamara and W. R. Young. Kinetics of a one-dimensional granular medium in the quasielastic limit. *Phys. Fluids A*, 5(1):34–45, 1993.

[179] S. Mei, A. Montanari, and P. Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

[180] P. Monmarché and J. Reygner. Local convergence rates for wasserstein gradient flows and mckean-vlasov equations with multiple stationary solutions. *arXiv preprint arXiv:2404.15725*, 2024.

[181] J. E. Morinelly and B. E. Ydstie. Dual MPC with reinforcement learning. *IFAC-PapersOnLine*, 49(7):266–271, 2016. 11th IFAC Symposium on Dynamics and Control of Process SystemsIncluding Biosystems DYCOPS-CAB 2016.

[182] S. Motsch and E. Tadmor. Heterophilious dynamics enhances consensus. *SIAM Review*, 56(4), 2014.

[183] M. Motta and F. Rampazzo. An abstract maximum principle for constrained minimum problems. *arXiv preprint arXiv:2310.09845*, 2023.

[184] H. Murakami, D. Yanagisawa, and K. Nishinari. Advancing lane formation and high-density simulations in bidirectional pedestrian flow. In *Transportation Research Procedia*. Elsevier, 2021.

[185] D. S. Naidu and R. C. Dorf. *Optimal Control Systems*. CRC Press, Inc., USA, 2003.

[186] T. Nakamura-Zimmerer, Q. Gong, and W. Kang. Adaptive Deep Learning for High-Dimensional Hamilton-Jacobi-Bellman Equations, 2019. arXiv preprint:1907.05317.

[187] T. Nakamura-Zimmerer, Q. Gong, and W. Kang. Qrnet: optimal regulator design with lqr-augmented neural networks. *IEEE Control Syst. Lett.*, 5:1303–1308, 10 2021.

[188] G. Naldi, L. Pareschi, and G. Toscani, editors. *Mathematical modeling of collective behavior in socio-economic and life sciences*. Modeling and Simulation in Science, Engineering and Technology. Birkhäuser Boston Inc., Boston, MA, 2010.

[189] S. R. Nekoo. Tutorial and review on the state-dependent Riccati equation. *J. Appl. Nonlinear Dyn.*, 8(2):109–166, 2019.

[190] R. Nickl, G. A. Pavliotis, and K. Ray. Bayesian nonparametric inference in mckean-vlasov models. *arXiv preprint arXiv:2404.16742*, 2024.

[191] A. Norouzi, H. Heidarifar, H. Borhan, M. Shahbakhti, and C. R. Koch. Integrating machine learning and model predictive control for automotive applications: A review and future directions. *Engineering Applications of Artificial Intelligence*, 120:105878, 2023.

[192] K. Oelschläger. A martingale approach to the law of large numbers for weakly interacting stochastic processes. *Ann. Probab.*, 12(2):458–479, 1984.

[193] K. Oh and J. Seo. Development of an adaptive and weighted model predictive control algorithm for autonomous driving with disturbance estimation and grey prediction. *IEEE Access*, 10:35251–35264, 2022.

[194] K.-K. Oh, M.-C. Park, and H.-S. Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015.

[195] T. Ohsawa. Symmetry reduction of optimal control systems and principal connections. *SIAM Journal on Control and Optimization*, 51(1):96–120, 2013.

[196] L. Oster, MathiasSallandt and R. Schneider. Approximating the stationary bellman equation by hierarchical tensor products. *Journal of Computational Mathematics*, 42(3):638–661, 2024.

[197] M. Oster, L. Sallandt, and R. Schneider. Approximating Optimal feedback Controllers of Finite Horizon Control Problems Using Hierarchical Tensor Formats. *SIAM J. Sci. Comput.*, 44(3):B746–B770, 2022.

[198] M. Oster, L. Saluzzi, and T. Wenzel. A comparison study of supervised learning techniques for the approximation of high dimensional functions and feedback control. *Dynamic Games and Applications*, Dec 2024.

[199] I. P. A. Papadopoulos, P. E. Farrell, and T. M. Surowiec. Computing multiple solutions of topology optimization problems. *SIAM Journal on Scientific Computing*, 43(3):A1555–A1582, 2021.

[200] L. Pareschi and G. Russo. An introduction to Monte Carlo method for the Boltzmann equation. *ESAIM: Proceedings*, 10:35–75, 2001.

[201] L. Pareschi and G. Toscani. *Interacting multiagent systems. Kinetic equations and Monte Carlo methods*. Oxford: Oxford University Press, 2014.

[202] L. Pareschi and G. Toscani. Wealth distribution and collective knowledge: a Boltzmann approach. *Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.*, 372(2028):20130396, 15, 2014.

[203] L. Pareschi, T. Trimborn, and M. Zanella. Mean-field control variate methods for kinetic equations with uncertainties and applications to socioeconomic sciences. *Int. J. Uncertain. Quantif.*, 12(1):61–84, 2022.

[204] L. Pareschi and M. Zanella. Structure Preserving Schemes for Nonlinear Fokker–Planck Equations and Applications. *Journal of Scientific Computing*, 74:1575–1600, 2018.

[205] H. V. D. Parunak, R. Savit, and R. L. Riolo. Emergence in complex systems: A survey of agent-based models. *Computational and Mathematical Organization Theory*, 8(2):91–104, 2002.

[206] G. A. Pavliotis. *Stochastic processes and applications*, volume 60 of *Texts in Applied Mathematics*. Springer, New York, 2014. Diffusion processes, the Fokker-Planck and Langevin equations.

[207] H. J. Pesch and M. Plail. The maximum principle of optimal control: a history of ingenious ideas and missed opportunities. *Control and Cybernetics*, 38(4A):973–995, 2009.

[208] A. A. Peters, R. H. Middleton, and O. Mason. Leader tracking in homogeneous vehicle platoons with broadcast delays. *Automatica*, 50(1):64–74, 2014.

[209] B. Piccoli, F. Rossi, and E. Trélat. Control to flocking of the kinetic Cucker-Smale model. *SIAM J. Math. Anal.*, 47(6):4685–4719, 2015.

[210] L. Pichler, A. Masud, and L. A. Bergman. *Numerical Solution of the Fokker–Planck Equation by Finite Difference and Finite Element Methods—A Comparative Study*, pages 69–85. Springer Netherlands, Dordrecht, 2013.

[211] S. S. Pon Kumar, A. Tulsyan, B. Gopaluni, and P. Loewen. A deep learning architecture for predictive control. *IFAC-PapersOnLine*, 51(18):512–517, 2018. 10th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2018.

[212] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. *The mathematical theory of optimal processes*. Translated from the Russian by K. N. Trirogoff; edited by L. W. Neustadt. Interscience Publishers John Wiley & Sons, Inc. New York-London, 1962.

[213] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. *The Mathematical Theory of Optimal Processes*. Wiley, 1962.

[214] A. J. Povzner. On the Boltzmann equation in the kinetic theory of gases. *Mat. Sb. (N.S.)*, 58(100):65–86, 1962.

[215] J. Procházka and K. Olševičová. Monitoring lane formation of pedestrians: Emergence and entropy. In *Intelligent Information and Database Systems (ACIIDS 2015)*, volume 9011 of *Lecture Notes in Computer Science*, pages 221–228. Springer, Cham, 2015.

[216] S. V. Raković and W. S. Levine, editors. *Handbook of model predictive control*. Control Engineering. Birkhäuser/Springer, Cham, 2019.

[217] G. Rotskoff and E. Vanden-Eijnden. Trainability and accuracy of artificial neural networks: An interacting particle system approach. *Communications on Pure and Applied Mathematics*, 75(9):1889–1935, 2022.

[218] D. Ruelle. *Statistical mechanics: Rigorous results*. World Scientific, 1999.

[219] L. Ruthotto, S. J. Osher, W. Li, L. Nurbekyan, and S. W. Fung. A machine learning framework for solving high-dimensional mean field game and mean field control problems. *Proceedings of the National Academy of Sciences*, 117(17):9183–9193, 2020.

[220] M. Shiino. Dynamic behavior of stochastic-systems of infinitely many coupled nonlinear oscillators exhibiting phase-transitions of mean-field type - h-theorem on asymptotic approach to equilibrium and critical slowing down of order-parameter fluctuations. *Phys. Rev. A*, 36(5):2393–2412, SEP 1 1987.

[221] C. Sinigaglia, F. Braghin, and S. Berman. Optimal control of velocity and nonlocal interactions in the mean-field kuramoto model. *2022 American Control Conference (ACC)*, pages 290–295, 2022.

[222] J. Sirignano and K. Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, 2018.

[223] J. Sirignano and K. Spiliopoulos. Mean field analysis of neural networks: A central limit theorem. *Stochastic Processes and their Applications*, 130(3):1820–1852, 2020.

[224] J. Sirignano and K. Spiliopoulos. Mean field analysis of neural networks: A law of large numbers. *SIAM Journal on Applied Mathematics*, 80(2):725–752, 2020.

[225] A. N. Specified. Sparse sensor selection for distributed systems: An l1-relaxation approach. *Automatica*, 150:110939, 2024.

[226] R. E. Stern, S. Cui, M. L. Delle Monache, R. Bhadani, M. Bunting, M. Churchill, N. Hamilton, H. Pohlmann, F. Wu, B. Piccoli, et al. Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments. *Transp. Research Part C: Emerging Techn.*, 89:205–221, 2018.

[227] N. Subbotina. The method of characteristics for Hamilton—Jacobi equations and applications to dynamical optimization. *Journal of mathematical sciences*, 135:2955–3091, 2006.

[228] K. Sznajd-Weron and J. Sznajd. Opinion evolution in closed community. *International Journal of Modern Physics C*, 11:1157–1165, 9 2000.

[229] A.-S. Sznitman. Topics in propagation of chaos. In *École d'Été de Probabilités de Saint-Flour XIX—1989*, volume 1464 of *Lecture Notes in Math.*, pages 165–251. Springer, Berlin, 1991.

[230] A. Tahir, J. Böling, M.-H. Haghbayan, H. T. Toivonen, and J. Plosila. Swarms of unmanned aerial vehicles — a survey. *Journal of Industrial Information Integration*, 16:100106, 2019.

[231] Y. Tamura. On asymptotic behaviors of the solution of a nonlinear diffusion equation. *J. Fac. Sci. Univ. Tokyo Sect. IA Math.*, 31(1):195–221, 1984.

[232] K. Tang, X. Wan, and Q. Liao. Adaptive deep density approximation for fokker-planck equations. *Journal of Computational Physics*, 457:111080, 2022.

[233] Z.-Z. Tao and B. Sun. A feedback design for numerical solution to optimal control problems based on Hamilton-Jacobi-Bellman equation. *Electron. Res. Arch.*, 29(5):3429–3447, 2021.

[234] M. Testard, S. Larson, E. Watowich, et al. Rhesus macaques build new social connections after a natural disaster. *Current Biology*, 31(11):2299–2309, 2021.

[235] M. Testard, S. Shergold, and L. Brent. Ecological disturbance alters the adaptive benefits of social ties. *Science*, 380(6648):1320–1324, 2024.

[236] T. Thieu and R. Melnik. Modelling the behavior of human crowds with interacting particle systems in active-passive population dynamics. *arXiv preprint arXiv:2201.07163*, 2021.

[237] G. Toscani. Kinetic models of opinion formation. *Comm. Math. Sci.*, 4(3):481–496, 2006.

[238] A. Tosin and M. Zanella. Boltzmann-type models with uncertain binary interactions. *Commun. Math. Sci.*, 16(4):963–985, 2018.

[239] A. Tosin and M. Zanella. Kinetic-controlled hydrodynamics for traffic models with driver-assist vehicles. *Multiscale Model. Simul.*, 17(2):716–749, 2019.

[240] A. Tosin and M. Zanella. Boltzmann-type description with cutoff of follow-the-leader traffic models. In *Trails in kinetic theory—foundational aspects and numerical methods*, volume 25 of *SEMA SIMAI Springer Ser.*, pages 227–251. Springer, Cham, [2021] ©2021.

[241] F. Tröltzsch. *Optimal Control of Partial Differential Equations: Theory, Methods and Applications (chapter 2.10)*, volume 112 of *Graduate Studies in Mathematics*. American Mathematical Society, 2010.

[242] J. Tugaut. Phase transitions of McKean-Vlasov processes in double-wells landscape. *Stochastics*, 86(2):257–284, 2014.

[243] T. L. V. *Mean Field Theory for Large Interacting Systems*. Springer, 2013.

[244] A. J. van der Schaft. l_2-gain analysis of nonlinear systems and nonlinear state feedback h_∞ control. *IEEE Transactions on Automatic Control*, 37(6):770–784, 1992.

[245] S. Vellmer and B. Lindner. Fokker–Planck approach to neural networks and to decision problems. *The European Physical Journal Special Topics*, 230:2929–2949, 2021.

[246] T. Vicsek and A. Zafeiris. Bifurcation theory in systems of interacting agents: From cucker-smale to synchronization. *Physics Reports*, 517(3):71–140, 2012.

[247] C. Villani. On a new class of weak solutions to the spatially homogeneous Boltzmann and Landau equations. *Archive for rational mechanics and analysis*, 143(3):273–307, 1998.

[248] C. Wang, Q. Li, W. E, and B. Chazelle. Noisy Hegselmann-Krause systems: phase transition and the $2R$-conjecture. *J. Stat. Phys.*, 166(5):1209–1225, 2017.

[249] E. Watowich, H. Chiou, L. Montague, et al. Natural disaster and immunological aging in a nonhuman primate. *Proceedings of the National Academy of Sciences*, 119(8):e2115306119, 2022.

[250] J. H. Wilkinson. *Rounding errors in algebraic processes*. Classics in applied mathematics ; 89. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1964.

[251] I. J. Wolf and W. Marquardt. Fast NMPC schemes for regulatory and economic NMPC – a review. *Journal of Process Control*, 44:162–183, 2016.

[252] T. Xiao and M. Frank. Using neural networks to accelerate the solution of the Boltzmann equation. *J. Comput. Phys.*, 443:Paper No. 110521, 22, 2021.

[253] Y. Xu, H. Zhang, Y. Li, K. Zhou, Q. Liu, and J. Kurths. Solving Fokker-Planck equation using deep learning. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(1), 2020.

[254] N. Zagli, V. Lucarini, and G. A. Pavliotis. Spectroscopy of phase transitions for multiagent systems. *Chaos*, 31(6):Paper No. 061103, 8, 2021.

[255] S. Zhang, K. Garg, and C. Fan. Neural graph control barrier functions guided distributed collision-avoidance multi-agent control. In J. Tan, M. Toussaint, and K. Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 2373–2392. PMLR, 06–09 Nov 2023.

[256] T. Zhao, Y. Zheng, J. Gong, and Z. Wu. Machine learning-based reduced-order modeling and predictive control of nonlinear processes. *Chemical Engineering Research and Design*, 179:435–451, 2022.

[257] M. Zheng, F. Liu, I. Turner, and V. Anh. A novel high order space-time spectral method for the time fractional fokker–planck equation. *SIAM Journal on Scientific Computing*, 37(2):A701–A724, 2015.

[258] X. Zhou. Maximum principle, dynamic programming, and their connection in deterministic control. *Journal of Optimization Theory and Applications*, 65(2):363–373, 1990.

[259] T. Çimen. State-dependent Riccati equation (SDRE) control: A survey. *IFAC Proceedings Volumes*, 41(2):3761–3775, 2008. 17th IFAC World Congress.