ABSTRACT

Title of Dissertation: Collaborative Control of Autonomous Swarms
with Resource Constraints

Wei Xi, Doctor of Philosophy, 2006

Dissertation directed by: Professor John S. Baras
Department of Electrical and Computer Engineering

This dissertation focuses on the collaborative control of homogeneous UAV swarms. A two-level scheme is proposed by combining the high-level path planning and the low-level vehicle motion control. A decentralized artificial potential function (APF) based approach, which mimics the bacteria foraging process, is studied for the high-level path planning. The deterministic potential based approach, however, suffers from the local minima entrapment dilemma, which motivate us to fix the "flaw" that is naturally embedded. An innovative decentralized stochastic approach based on the Markov Random Filed (MRF) theory is proposed; this approach traditionally used in statistical mechanics and in image processing. By modeling the local interactions as Gibbs potentials, the move-

ments of vehicles are then decided by using Gibbs sampler based simulated annealing (SA) algorithm.

A two-step sampling scheme is proposed to coordinate vehicle networks: in the first sampling step, a vehicle is picked through a properly designed, configuration-dependent proposal distribution, and in the second sampling step, the vehicle makes a move by using the local characteristics of the Gibbs distribution. Convergence properties are established theoretically and confirmed with simulations. In order to reduce the communication cost and the delay, a fully parallel sampling algorithm is studied and analyzed accordingly. In practice, the stochastic nature of the proposed algorithm might lead to a high traveling cost. To mitigate this problem, a hybrid algorithm is developed by combining the Gibbs sampler based method with the deterministic gradient-flow method to gain the advantages of both approaches.

The robustness of the Gibbs sampler based algorithm is also studied. The convergence properties are investigated for different types sensor errors including *range-error* and *random-error*. Error bounds are derived to guarantee the convergence of the stochastic algorithm.

In the low-level motion control module, a model predictive control (MPC) approach is investigated for car-like UAV model. Multiple control objectives, for example, minimizing tracking error, avoiding actuator/state saturation, and minimizing control effort, are easily encoded in the objective function. Two numerical optimization approaches, gradient descendent approach and dynamic programming approach, are studied to strike the balance between computation time and complexity.

Collaborative Control of Autonomous Swarms with Resource Constraints

by

Wei Xi

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2006

Advisory Committee:

Professor John S. Baras, Chairman / Advisor
Professor Eyad Abed
Professor Armand Makowski
Professor Nuno Martins
Professor Balakumar Balachandran, Dean's of Representative

DEDICATION

To Mom, Dad, and my wife Fang

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my adviser, Professor John S. Baras, for his excellent guidance and continuous support during my graduate studies. I am grateful to him for giving me an invaluable opportunity to work on challenging and extremely interesting projects over the past four years. Thanks for broadening my scope and helping me achieve my goals.

I would like to thank Professor Eyad Abed, Professor Armand Makowski, Professor Nuno Martins and Professor Balakumar Balachandran for serving in my advisory committee and providing many insightful suggestions and comments.

I would also like to thank Professor P. S. Krishnaprasad for his enthusiasm in teaching and guidance on my course project. I benefited a lot from his valuable courses on adaptive control and geometric control.

I would like to express my gratitude to my colleagues and friends at Maryland, who offer help on my research and job searching. I owe them a lot.

I want to thank for the financial support of my graduate studies and research by the Army Research Office under the ODDR&E MURI01 Program Grant No. DAAD19-01-1-0465 to the Center for Networked Communicating Control Systems (through Boston University), and under ARO Grant No. DAAD190210319. I also want to thank for administrative staff of ISR for their assistance of my payroll, health insurance, and etc.

Finally, but most importantly, I am deeply indebted to my wife and my parents for

their unconditional love, support, and care. No matter what happens, they always stand on my back and share my happiness and suffering.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

With rapid advances in sensing, communication, computation, and actuation capabilities, it is extremely appealing to cooperatively/coordinatively control a large number of unmanned autonomous vehicles (UAVs) to perform dangerous or explorative tasks in various hazardous, unknown or remote environments. Originally driven by the need for saving labor costs and protecting personnel loss from dangerous environments and like, the applications of UAVs have been potentially extended to a broad range both in military and industry, for example, automated highway systems, mobile sensor networks, ocean resources exploration, spacecraft interferometry, satellite formations, robotic border patrol and search-and-rescue operations [1, 2].

In such applications, the number of autonomous vehicles involved can be very large, ranging from hundreds to thousands, which forbids the centralized control approaches to be applied. The primary reason is that the communication cost and computation cost usually increase exponentially with the size of UAV swarms. Therefore, decentralized/distributed control approaches are especially appealing considering the large scale of the vehicle networks and the bandwidth constraint on the communication [3, 4, 5, 6]. The main advantages of such approaches are flexibility, scalability, robustness and cost-effectiveness. The flexibility lies in the fact that the distributed/decentralized algorithm is easy to migrate to different size of UAV networks. The scalability is due to fact that

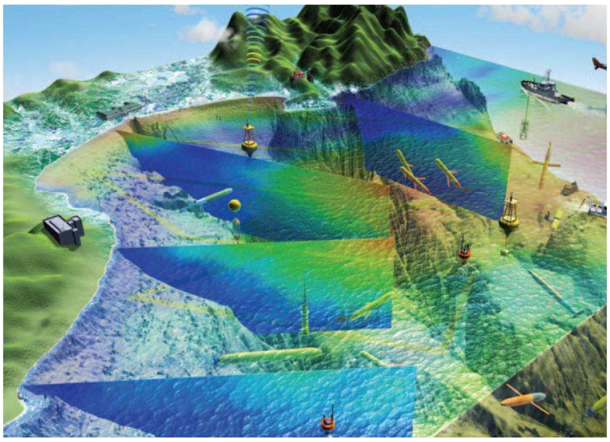

Fig. 1.1: Ocean resource exploration by autonomous underwater vehicles

the communication cost and computation cost only increase linearly with swarm size. The decentralized/distributed approach depends primarily on local information, which reduces the requirement of sophisticated sensing and data processing capabilities. Therefore, cost-effective sensors and actuators are usually preferred to reduce the total system cost. For example, resource exploration and information gathering in oceans, Mars, or polluted areas (by chemical or radioactive materials) can often be accomplished more efficiently by groups of low-cost mobile vehicles (Fig. 1.1) than a single sophisticated and expensive one. Moreover, the robustness implies that a UAV network's performance will not degenerate too much with the loss of a few UAVs.

In this dissertation, distributed distributed coordination control methodologies for UAV swarms are studied. The aforementioned communication, sensing, and actuation constraints present theoretical challenges in designing a fully distributed algorithm for achieving group objectives. Recently, bio-inspired approaches have been receiving growing attentions to addressed these problems. By mimicking the coordination/collaboration mechanisms in social animals, for example, ant colony optimization, social potential, which lead to many amazing and interesting collective behaviors, people have been trying to use them in controlling artificial systems, like UAV networks/swarms. A brief review can be found in the next chapter.

In this research work, learning inspiration from bacteria foraging process, an artificial potential function (APF) based approach is investigated for coordinating UAV swarms in a battle field scenario. The deterministic potential descent approach, however, suffers from the local minima entrapment dilemma, which motivate us to fix the “flaw” that is naturally embedded. The details can be found in chapter 3. The solution is inspired from Markov Random Field (MRF), a well known model used to describe phase transition in ferromagnetic material. This approach is systematically studied in this dissertation.

In the design of applications involving UAV swarm networks, a commonly used two level hierarchical scheme is adopted. The high level module generates way-points for vehicle swarms. Vehicles are usually treated as point masses moving on a 2D or 3D mission space. The aforementioned potential based approach belongs to this category.

The lower level module deals with vehicle dynamics. Given desired way-points generated from high level module, the low level module is responsible for the trajectory tracking, actuator saturation, and many other practical concerns. In this dissertation, a Model Predictive Control (MPC) based approach is proposed to address the multiple objectives oriented trajectory tracking control. We now outline the contributions of this dissertation.

1.1 Contributions of the Dissertation

In this dissertation, decentralized/distributed collective approaches are studied for a military based example. These approaches can be easily extended to many other UAV applications, for example, formation control, flocking control, and etc. The major contribution of this dissertation is that a fundamental dilemma of APF approach, local minima entrapment problem, is fully solved by using a novel simulated annealing algorithm based on gibbs sampler and Markov Random Field theory. The other contribution is to apply MPC based approach for multiple objectives oriented trajectory tracking control.

1.1.1 Innovative decentralized collaborative control of vehicle swarms based on MRF and SA

As aforementioned, inspired by the emergent collective behaviors demonstrated in nature, for example, bacteria foraging, APF approaches are usually used in collaborative system design [7, 8, 9, 10, 11]. By encoding local interactions between UAV swarms and environments in potential functions, vehicles make moves by following the negative gradient flow of potentials. Despite the simple and elegant nature of the APF approach, it suffers from the dilemma that system dynamics could be entrapped in the local minima due to the nonconvexity of the potential function [12]. Researchers attempted to address this problem by designing potential functions that have no other local minima [13, 14], or by escaping from local minima by using ad hoc techniques, for example, random walk [15], virtual obstacles [16], and virtual local targets [17].

Inspired by the Ising model in statistical physics, a novel, systematic approach is proposed to coordinate an autonomous swarm based on the theory of Markov random fields (MRFs) and Gibbs sampling, a tool that has been traditionally used in statistical mechanics, image processing, computer vision, and data analysis [18, 19, 20, 21, 22]. In this approach, a swarm is modeled as an MRF on a graph, where the (mobile) vehicles and their communication/sensing links constitute the vertices and the edges [23] of the graph, respectively. As in the APF approach, global objectives and constraints (e.g., obstacles) are reflected through the design of potential functions - in this case, Gibbs potentials. The movement of vehicles is then decided by using simulated annealing (SA) based on the Gibbs sampler. The dynamic graph associated with the evolution of vehicle networks, however, presents significant challenges in convergence analysis since classical MRF theory does not apply directly.

In this dissertation, convergence of the novel stochastic approach is systematically studied. The local minima entrapment problem can be generally solved with an appropriate cooling schedule. The robustness of the stochastic algorithm in the presence of sensor uncertainty is also studied. Moreover, a fully distributed parallel algorithm and its convergence for saving sampling time are also investigated.

In practice, the stochastic nature of the proposed algorithm might lead to a high traveling cost and long maneuvering time. To mitigate this problem, a hybrid algorithm is developed by combining the Gibbs sampler-based method with the deterministic gradient-flow method to gain the advantages of both approaches. A vehicle switches between the two schemes to achieve a sound tradeoff between efficiency and optimality. Analysis is performed on the choice of switching parameters. The impact of vehicle memory on performance enhancement is also investigated.

1.1.2 Multiple objectives oriented motion control based on MPC approach

As mentioned earlier, in this dissertation, a two-level based scheme is adopted to control UAV swarms. The potential based stochastic algorithm belongs to a high-level path planning algorithm. A point mass model is considered to represent a UAV, instead of a real dynamic and kinematic model. Way-points are generated to lead vehicle networks to the global objective. The goal of low-level motion control module is to compute control input for real UAV/robot to follow desired way-points.

In the low-level module, a model predictive based (MPC) based motion control is proposed for (but not limited to) car-like nonholonomic UAVs. This approach is appealing because it can easily accommodate multiple control objectives, for example, minimizing tracking error, avoiding actuator/state saturation, and minimizing control effort, in an elegant and compact way. By applying the well-known Pontryagin Maximum Principle (PMP), a gradient based approach is first proposed to solve the sub-optimal control problem. The computation cost and delay, however, presents difficulties in real-time control. A dynamic programming based approach is then constructed by assuming that the actuator has limited control options. Extensive simulations confirm our analysis.

1.2 Organization of the Dissertation

In Chapter 2, a literature review of the current understanding of the collective behaviors exist in nature and how they affect engineering design in multiple agent system is provided. The deterministic APF based approach and its limitation are introduced in chapter 3. In chapter 4, the focus is the novel stochastic algorithm based on MRF and SA.

Chapter 5 contains the convergence analysis of the fully distributed Gibbs sampler based algorithm. In chapter 6, a hybrid scheme is proposed to improve the system performance. The impact of memory is also discussed. The robustness analysis considering sensor uncertainty is discussed in chapter 7. Low-level motion control based on MPC approach is studied in chapter 8. Conclusions and future work are provided in chapter 9.

Chapter 2

Literature Review

2.1 Understanding the swarming behavior in nature

For centuries, swarming or aggregation behaviors in wild nature have been noticed by people. Swarm behaviors exist in many species, ranging from small organisms as simple as bacteria to large mammals like whales [24, 25]. For example, at the end of every autumn, the Canada geese fly in a V-formation in loose groups when they migrate from north to south (see figure 2.1). Many kinds of social animals, like birds, fishes, and ants, easily organize themselves in a large orderly group and move as a single organism without a central commander. Some social insects, such as ants and bees, cooperate to forage food in places far away from their home and to build sophisticated giant hives comparing to their tiny size without advanced communication technology (e.g., language) and computation tools (e.g., brain). Biologists have spent a long time to understand the underlying mechanism of these swarming behaviors. Although good progress has been made in the past decade, most of them are still in mysteries for us.

To understand the nature of the aggregation behavior, the first thing needed to be studied is the motives that drive these animals to aggregate in groups. In [26], Parrish *et al.* pointed out that aggregation is actually an evolutionarily advantageous state. First of all, it is believed that forming a group may increase the chances of survival of newborns and juveniles from being eaten by predators, such that the reproduction of the species can

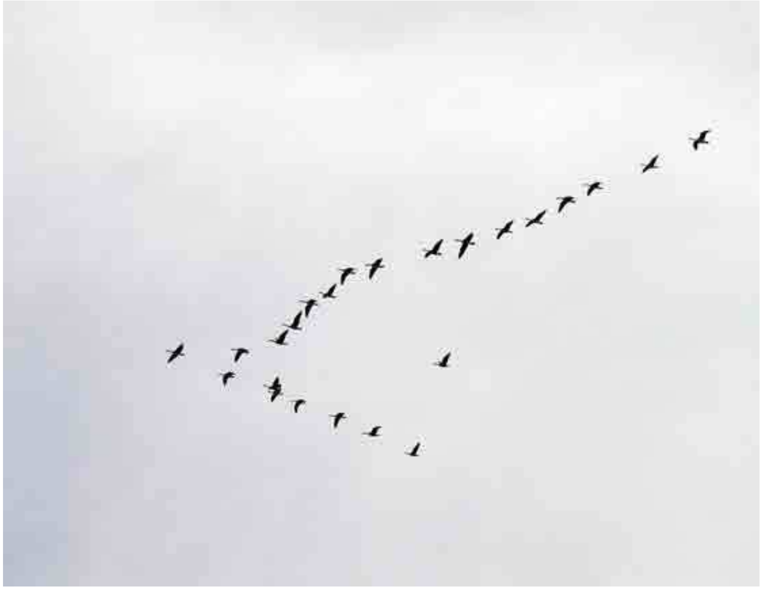

Fig. 2.1: Geese migrating from north to south in a V-formation

be continued. Second, the aggregation also helps to find food because a large group of animals has more capability to “sense” and “search” than a single one. Other benefits include energetic saving for flying or swimming by certain relative position among neighbors (e.g. group of wild geese in V-shape), and reproduction success by mating aggregation. These facts may explain some aspects of aggregation. In many other cases, however, it is hard to find functional purposes associated with aggregation patterns. Additional background can be found in [27, 28, 29] and references there in.

Another important issue is how individual members coordinate with each other such that the whole group exhibits emergent collective behaviors. For decades, many mathematical models were proposed by biologists to get insights into the nature of swarming behaviors [27]. Most of these models are focusing on spatial approaches, where the space is directly or indirectly considered in the model; others are non-spatial approaches [30].

In spatial approaches, there are mainly three different approaches that have been used for the analysis of swarm dynamics [27, 31].

The first one is based on a statistical model, which uses the “Eulerian” framework to describe the mean-field density of swarm. In this model, each member of the swarm is not treated as a single identity. The swarm is described as density in spatial space by a partial differential equation that is based on a diffusion approximation of the random motion [32]. Recent work in [33] extended the approach by integrating non-local interactions, such as visual or auditory sensing. Although many analytical results can be produced, this approach is limited to large, dense swarms with no big discontinuities [27].

The second spatial approach is based on individual-based path generation, where “Lagrangian” equations are used to describe motion of individual members in the swarm [32, 30]. In this model, all interactions among individuals are modeled as attraction and repulsion forces. An attractive feature of this approach is that all interplays between members in the swarm can be modeled as potential functions, and the motion of each individual follow the negative gradient of the potential surface. By constructing a Lyapunov function associated with the potential, it is straightforward to show that the minimizer correspond to the stable state of the swarm system. The common understanding from this approach is that aggregation is caused by the long-range attraction and the short-range repulsion, although the form of attraction/repulsion functions can be varied [34, 35]. For example, in [35], Mogilner suggested a model where attraction and repulsion terms were exponentials with different magnitudes. By using this model, the author also derived the individual distance of a large group, which revealed a condition on the attraction and repulsion to avoid collapse of the swarm.

The third spatial approach uses behavior based model, where no explicit mathematical equations are prescribed. All interactions among individual members are described by some behavior rules. In 1986, it is known that Crag Reynolds wrote a small computer program that simulate the flocking of organisms by letting a number of identical ones follow simple rules. In this model [36], named “boids”, three simple steering rules are used: separation (avoiding local crowding), alignment (matching the heading of neighbors), and cohesion (moving towards the average of local flockmates). The simulation based approaches can create real-like animations of flock animals that have been successfully applied in movie industry, for example, “The lion king” and “Batman Return” [37].

In parallel to Reynolds’ work, physicist Vicsek proposed a similar rule based model to study collective behaviors of self-propelling particles (SPP)[38]. In his simulation, orientation consensus behaviors were observed by a nearest neighbor rule. In Vicsek’s model, each particle has constant speed v_0, and tries to align the orientation to the local average velocity. Phase transition can be observed by numerical results even in the noise environments. Other works that study the motion and related transport phenomena in non-equilibrium process of multi-particles system can be found in [39, 40, 41, 42, 43, 44, 45].

Although this approach provides eyeball attractive appealing, it is hard to analyze and thus to get insight of the nature of the emergent collective behavior due the lack of mathematical description. Nonetheless, there are still some successes in analyze the behavior based swarming effects, for example, consensus and flocking. In [39], Toner and Tu analyzed the Vicsek model by using a continuous “hydrodynamic” model. In [6], they analyzed the “boids” model. By investigating the algebraic graph theoretical properties of the underlying interconnection graph, they showed the relationship between stability of the flocking motion and the graph connectivity.

These convergence analysis results are further relaxed by Jadbabaie and Ren in [5, 46], they showed that consensus can be achieved even if the network topology is not connected all the time. The main results are follows: if there exist a connected spanning tree in the union of network topology over finite time interval, the nearest neighbor rule guarantees the consensus convergence. In [47], Olfati further study the flocking behavior of “boids” model.

In the non-spatial approaches, space is not a factor to describe the swarm dynamics [27, 48]. They actually treat the group size as a primary factor to described the swarming behavior. In [27], it was argued that group size is determined by the balance of payoffs to individual members, i.e. the size of group affects the individual benefits. It was believed that there is a theoretically optimum group size n^*, at which point the average benefits of group members are maximal compare to loner, and the equilibrium or the critical group size $\bar{n}$ happens when average benefits are equal to the loner. When the group size is over critical group size $\bar{n}$, the whole swarm is then divided into small ones. In [48], the authors used a continuous model to describe the distribution of animals group size. They focused on a fixed size group divided into groups of various dynamic sizes.

All these approaches gain success in explaining or demonstrating collective behaviors of the swarm in some aspects. None of them, however, can explain them all. For example, the “Eulerian” framework unveils the density evolving of a whole swarm system, but hard to bridge the individual to collective level. On the contrary, “lagrangian” framework concentrates on individual behavior. This approach can easily describe the motion of each member in the swarm, but is hard to analyze the transformation of the collective behavior in group level. Excellent background material of mathematical models for swarming behaviors can be found in [32] and references there in.

2.2 Bio-inspired Design for collaborative control system

In the previous section, different models were proposed by biologists and physicists to analyze the underlying mechanism of the swarming behaviors. Inspired by this, people try to integrate those results into engineering design for, in particular, applications of robot/UAV swarms. In the literature, there are mainly three approaches have been reported: artificial potential function approach, behavior-based approach, and leader/follower approach.

2.2.1 Artificial potential function approach

The artificial potential function based approach has been explored for path planning and control of robotic manipulators and mobile robots over the past two decades [49, 50, 51]. Models involving in the approach often use potential functions to describe the relationship between individual agents and environments. And the motion of each individual is determined by the negative gradient flow of the potential functions. This approach mimic the collective behaviors demonstrated in large social insects or bacterial, where social potential was used to modeling swarm aggregation and cohesion [52].

In [3], Olfati-Saber used structure potential function to achieve collision-free, distributed formation stabilization of autonomous swarms. By imposing constraints on the formation graph to eliminate ambiguities, the potential function was designed in a way which leads autonomous swarms to a unique formation. The state feedback control law were used to control the motions of swarms, and the local formation stability and bounded feedback stability was then established by constructing corresponding lyapunov function. Similar results can be found in [7, 53], where structure potential are constructed by introduce virtual leader. Without eliminating the ambiguities, the formation of swarms in their model may converge to one of many possible formations.

Besides the formation control, the artificial potential approach is also used in path planning of autonomous swarms [54]. In [9], the authors demonstrated a interesting application in battle field scenario, where static obstacles and moving threats coexist in the same environment. By following the gradient flow of potential functions, which was described by a weighted sum of sub-potentials reflecting the objectives or constraints, the swarm vehicles demonstrated fascinating collective behavior to reach the target without being collided with obstacles and catched by moving threats. Despite the simple, local, and elegant nature of the potential-based approach, this approach suffers from the problem that the system dynamics could be trapped at the local minima of potential functions [12]. To avoid this situation, Ge [54] suggested to adjust the coefficients of repulsive potential according to the minimum distance between robot and target position, thus the local minimum can be eliminate. Kim *et. al.* [11] proposed a similar method to get around local minimum by adjusting the ratio between weighting coefficients of sub-potentials. Although these improvements can solve the problem in some special situations, for general case, it is still unsolved.

2.2.2 Behavior-based approach

Behavior-based method is another important approach originating form swarm behavior of creatures, for example, Reynolds “boids” model[36]. By designing a set of actions or behaviors for each element in the group, it hopes that desired group behavior emerges as a result. In [55], Miller proposed behavior-controlled diagram to control micro-rovers to execute various tasks, like grabbing rocks. The advantage of behavior-based approaches are robust, flexible, computational efficient and communication-cost effective. However, it is hard to decompose desired group behavior to element individual behavior. To address the problem, Parker proposed a mixed strategy that combine local behavior control and global knowledge in terms of cumulative position error and time to control robots maintain a line formation when they are navigating past waypoints to a final destination [56]. And the results showed that system performance improved by considering global knowledge. In [57], Balch and Arkin extended the Park’s work, and showed that the unit-center reference techniques provides better performance than the leader-referenced approach in Park’s paper. Though simulations in these papers showed success of the behavior based approach, it is hard to systemically study because the approaches lack of analysis basis.

2.2.3 Leader/Follower approach

In leader/follower approach, the leader’s motion is prescribed. The other agents in the swarm robotics are followers which track the motion of leader or subset of neighbor agents with some offset. This approach can be also traced back to the V-formation demonstrated in geese migration. Variants of the approach are proposed in many papers to improve the performance and study the convergence property, which include multiple leaders, chain topologies, tree topologies and network topologies, and thus induced various information flow among the autonomous swarms.

In [58], a vision-based formation control of nonholonomic robots were proposed. By specifying the desired formation in the image plane, the control problem is translated into a distributed leader/follower formation control. By using nonlinear feedback tracking controller for each follower agent, the formation input-to-state stability were established. In [59],Wang studied the approach for formation keeping and relative attitude alignment based on nearest neighbor tracking. Different topologies of the approaches, like nearest neighbor tracking, barycenter tracking, etc., are compared in this paper. Some advanced extension, for example, actuator saturation and adaptive control, were reported in [60].

In [61], the leader/follower model is described by a double-graph model, where one graph is used to describe the leader states flow, and the other one is used to describe neighbors states flow. Experiment and simulation shows that the system performance is improved by increasing the communication cost (the amount information need to transmitted through the network). The mesh stability of the interconnected multi-vehicles system are established in this paper as long as the formation topology is acyclic.

Unlike traditional leader/follower model, in [62], Fax and Murray assume all agents are identical and no leaders are explicit designated. The communication interconnection between vehicles forms a directed graph. By using tools from algebraic graph theory, a Nyquist criterion is proved, which states how the eigenvalues of the graph Laplacian matrix determine formation stability give the communication topology.

Besides to aforementioned approaches, several other approaches have been studied in formation control and path planning. In [63, 64], they used the virtual structure approach for formation control, where the formation is describe by a single rigid body. The translation from the virtual structure motion to individual motion needs centralized mechanism which limit the applications of the method. Justh used the planar Frenet-Serret framework for multi-UAVs formation with constant speed [65]. In [66], a dynamic programming approach was used to find near-optimal search path in the presence of uncertainty and moving constraints.

Chapter 3

Problem formulation and an APF based approach

As many applications have been introduced in chapter 1, UAV swarms are potentially able to perform various dangerous or complicated missions in remote or poisonously environments. For example, in defense applications, missions like hostage rescue, surveillance and ground mines clearance can be performed by a group of UAV swarms to help reduce human losses. In these types of applications, the full information of environments are usually unknown. Moreover, limited sensing capabilities, computation power, battery power are often required for UAVs in order to reduce the manufacturing cost. On the other hand, UAV swarms can exchange information through wireless communication , e.g., Radio Frequence (RF) communication, shortwave communication. However, environment conditions (e.g., severe weather), and large number agents in UAV networks introduce communication error and delay. Hence, decentralized approaches which require only local information are especially appealing.

Inspired by the emergent collective behaviors demonstrated in insects, bacteria, and even human being, the decentralized coordination mechanisms existing in nature have been received great interests. As has been discussed in the last chapter, the collective behaviors are often involved with so called “social potential” [52]. Borrowed from here, the Artificial Potential Function (APF) approach has been adopted and explored in collaborative control design over the past two decades, for example, mobile robot networks path planning and control of robotic manipulators, see e.g., [49, 50, 51] and the references therein. In this approach, the interaction between individual agents and environments, as well as group tasks are often encoded in the potential function. The potential functions should be designed such that desired configurations of multiple robotics correspond to the global minimizer of the ensemble potential functions.

In this chapter, a APF based decentralized approach is investigated to collaboratively control UAV swarms to achieve group tasks in a battle field scenario. The mission goal is to let the UAV swarms autonomously explore their ways to a predetermined target area while avoiding unknown obstacles and moving threats. To address multiple objectives instantaneously, a linear combination of individual potential components is used, where each individual potential term reflects one single objective or constraint, e.g., coherence among inter vehicles, obstacle avoidance, collision avoidance, and etc.

The potential functions are constructed such that only local information is involved, which guarantees that the potential function can be evaluated locally. There are usually two sources for getting local information. One is the information captured by the sensors (e.g., camera, sonar) on the UAV. For example, the distance been UAV and obstacles. The other one is through the communication link among neighboring UAVs. They could share the information they have in their memories. With potential function encoded in the UAV, at every time instant each vehicle evaluates its potential function profile with local information and decides its velocity by following the negative gradient flow.

Despite the simple and elegant nature of the APF approach, it suffers from the dilemma that system dynamics could be entrapped in the local minima due to the nonconvexity of the potential surface. A novel approach based on the Markov Random Field

(MRF) and Simulated Annealing (SA) is then proposed to solve the problem and studied systemically in the remainder of the dissertation.

3.1 Problem setup: A battle field scenario

The problem setup follows [9]. Consider a kinematic path planning problem in an unknown 2D space (the extension to 3D space is straightforward), which is called mission space hereafter. There are N vehicles moving on the ground. For simplicity, we assume that all vehicle are identical, and has point mass first order kinematic equation, i.e.,

$$\dot{p}_k(t) = u_k(t), \quad k = 1, ..., N, \tag{3.1}$$

where $p_k(t) = (x_k(t), y_k(t))$ is the location of vehicle k at time instant t, and $u_k(t)$ is the control input.

Each vehicle carries multiple sensors (e.g, camera, sonar) for exploring local surrounding area within the sensing range R_s. Two vehicle are neighbors if they are within interaction distance R_i, where $R_i \leq R_s$. There is a desired inter-vehicle distance r_0 for several reasons: staying too close leads to small area of coverage, good chance of collision, and easy targeting by the enemy fire, while staying too far apart leads to loss of communication and coordination.

The mission scenario is shown in figure 3.1. There are several unknown obstacles distributed in the mission space. For simplicity, we first assume that obstacles are of circular or eclipse shape. More complicated shape of obstacles may lead vehicle swarms fail to accomplish group tasks with simply APF approach. Denote p_l^o as the centroid of the obstacle l, and R_l^o as the radius. It is also assumed that each vehicle has the knowledge of

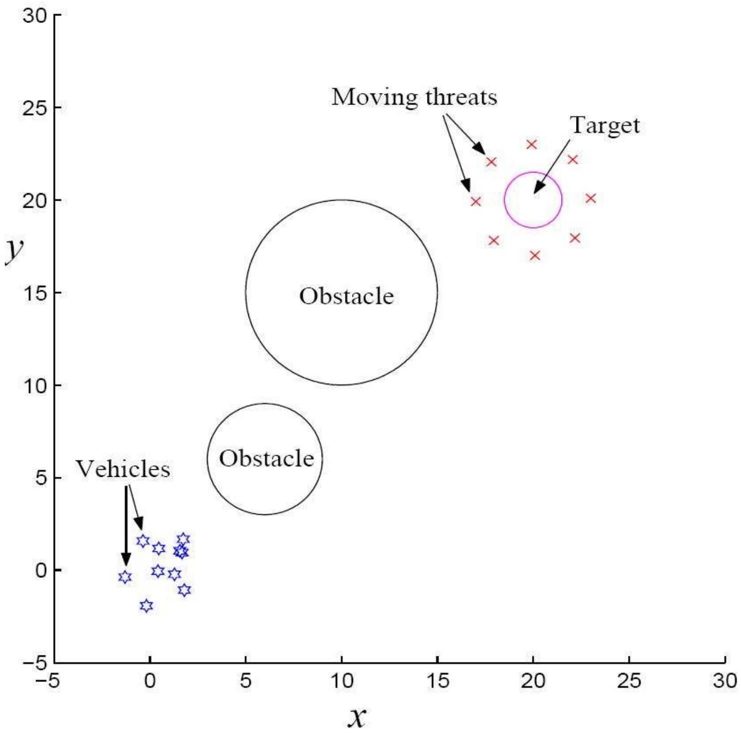

Fig. 3.1: A battle field scenario

target area location. The target area is also of the circular shape which located are p^g and is guarded by moving threats surrounding outside. The moving threats with predetermined trajectory $p^m(t)$ will destroy vehicle if the distance between them is within is range R_e.

Let $\mathscr{V}(t)$ be the set of vehicles that are alive at t, and $\mathscr{N}_k(t)$ be the neighboring set of the vehicle k defined by

$$\mathscr{N}_k(t) = \{j \in \mathscr{V}(t) : j \neq k, \|p_k(t) - p_j(t)\| \leq R_i\}.$$

The mission task has two folds: first, generate collision-free paths such that multiple vehicle could reach to the target area without being trapped by obstacles. second, enter into the target area without being killed by moving threats and cover the target area.

3.2 APF Approach based Decentralized control of UAV swarms

To accomplish the mission tasks, a linear combination of multiple potential function terms are designed to accommodate multiple objectives and constraints. In particularly, the potential function for vehicle k is of the following form

$$J_k = \lambda_g J^g(p_k, p^g) + \lambda_n J_k^n(p_k, p^{\mathscr{N}_k}) + \lambda_o J_k^o(p_k, p^o) + \lambda_m J_k^m(p_k, p^m(t)), \tag{3.2}$$

where $J_k^g, J_k^n, J_k^o, J_k^m$ are potential terms accounting for target attraction, inter-vehicles cohesion, obstacle avoidance and moving threats avoidance. $\lambda_g, \lambda_n, \lambda_o, \lambda_m$ are corresponding coefficients. The design of the coefficients is also very challenging.

As aforementioned, the desired configuration p^* of the vehicle swarms attains the global minimum potential values, i.e., $J(p^*) = \min_p(J)$, where $J = \sum_k J_k$ is the *total potential value*. The control objective can then be precisely described as follows: find decentralized control input $u_k(t)$, such that $p(t) \to p^*$ as time $t \to \infty$. The gradient descent approach is commonly used for the controller design, i.e.,

$$u_k = -\frac{\partial}{\partial p_k} J_k \tag{3.3}$$

Each component in (3.2) is described in the follows.

- Target attraction potential J_k^g

  The target attraction potential is used to lead vehicles move towards the target. A frequently used form is $J_k^g = f(\|p_k(t) - p^g\|)$, where $f(\cdot)$ is a strictly increasing function with respect to $\|p_k(t) - p^g\|$. When vehicle is far away from the target, the target potential has high value, and vice versus. In a free space, if there is no other

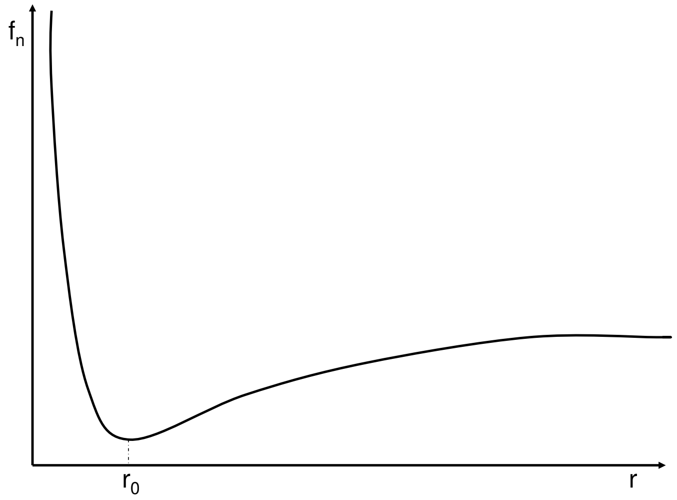

Fig. 3.2: The inter-vehicle potential function

terms in the potential function (3.2), by taking the velocity as the negative gradient of J_k^g only, the vehicle k moves directly to the target.

- inter-vehicle potential J_k^n

  The inter-vehicle potential maintains the desired distance r_0 between neighboring vehicle with the following form

  $$J_k^n = \sum_{j \in \mathcal{N}_k(t)} f_n(r_{jk}),$$

  where $r_{jk} = \|p_k(t) - p_j(t)\|$. When two vehicle are too close to each other $(r_{jk} \leq r_0)$, the potential function $f_n(r_{jk})$ tends to infinity as $r_{jk} \to \infty$, and is strictly deceasing on $(0, r_0]$. When two vehicle are far away to each other $(r_{jk} \geq r_0)$, the potential function $f_n(r_{jk})$ is strictly increasing on $(r_0, R_s]$ and holding constant on $[R_s, \infty)$. A typical inter-vehicle potential $f_n(r_{jk})$ is shown in figure 3.2

- The obstacle potential J_k^o

The obstacle potential prevents vehicle from being entrapped by obstacles. When vehicle is moving towards the obstacle, the potential function approaches to infinity, and is strictly decreasing on $(0, R_s]$. An example satisfying the property is

$$J_k^o = \frac{1}{r_k^o}$$

where r_k^o is the shortest distance between vehicle and obstacle surface.

- The moving threats avoidance potential J_k^m

  The moving threats are modeled as moving points. Let $\mathscr{M}_k$ be the set of moving threats within the detection range of vehicle k. J_k^m is expressed as

  $$J_k^m = \sum_{j \in \mathscr{M}_k} f_m(r_{k,m_j}),$$

  where $r_{k,m_j} = \|p_k - p_{m_j}^m\|$ is the distance between vehicle k and moving threats m_j. $f_m(\cdot)$ is a monotone decreasing function on $[R_e, R_s]$, and constant on (R_s, ∞). Moreover, $f_m(\cdot)$ goes to infinity as r_{k,m_j} approaches to R_e. The design of the moving threats potential function is similar as the obstacle potential function.

3.3 Local minimum entrapment dilemma and existing solutions

As mentioned in the last chapter, the potential based approach provide people an extremely simple and elegant way to understand the collective behaviors demonstrated in nature, e.g., insects swarming behavior, bacteria foraging. Inspired from nature, people utilize this approach in designing artificial multiple agent system, e.g., multiple UAV system, mobile sensor network. In [9], extensive simulation results demonstrate the success of the APF based approach in the application of a battle field scenario. However, the stability analysis via the celebrated Lyapunov stability theory shows that the gradient decent approach may lead to local minima such that the group mission fails, i.e., the final configuration of the vehicle swarms may not be the desired one. Unless one design the potential function very carefully such that only one minimum exists on the potential surface, there is no guarantee that the gradient decent approach can lead the vehicle swarms to achieve desired configuration.

Proposition 3.3.1 *The autonomous system (3.3) is locally stable. Let $\mathscr{E} = \{p^* \triangleq (p_1^*, ..., p_N^*) : \frac{\partial}{\partial p_k} J_k \mid_{p^*} = 0, \forall\ k\}$ be the equilibria set. Starting from arbitrary initial condition p_0. The final configuration $p_\infty = \Phi(p_0) \in \mathscr{E}$. If the potential surface is strictly convex, i.e., equilibrium set has only one stable equilibrium, the system asymptotically reach to the global minimizer.*

Proof. Let $J = \sum_{k=1}^N J_k$. It is straightforward to verify that J is a valid candidate lyapunov function. With control designed using gradient decent in (3.3), we have

$$\begin{aligned} \frac{d}{dt}J &= \sum_{k=1}^{N} \frac{\partial}{\partial p_k} J \cdot \dot{P}_k \\ &= \sum_{k=1}^{N} -(\frac{\partial}{\partial p_k} J_k)^2 \leq 0 \end{aligned} \tag{3.4}$$

By applying lyapunov stability theorem [68], one concludes that the system (3.3) is locally stable.

$\mathscr{E}$ defines a limiting set. By applying the lasalle's theorem [68], the system approaches to the limiting set $\mathscr{E}$ as time tends to infinity. If the potential function J is strictly convex, it is straightforward to show that the global minimizer of J is the unique component in $\mathscr{E}$. Thus, the gradient decent approach is guaranteed to lead the system to the global minimizer□

In the proposition 3.3.1, one can see that the final configuration p_∞ depends on the initial configuration p_0 if the the potential surface J is non-convex. To substantiate the fact, simulations in the next section show that in the battle field scenario UAV swarms could fail to reach the designated target given the existence of a non-convex shape obstacle. To address the problem, several approaches in the literature were proposed.

As stated in the proposition 3.3.1, one way to avoid local minimum entrapment is through design potential function such that it is strictly convex. For example, in [14], a harmonic function was proposed to construct potential function which is free from local minima. In [13], Volpe and Khosla proposed to use superquadric function to eliminate local minima.

Another popular way to solve the problem comes from heuristic point of view. In [16], virtual obstacles were proposed the push the vehicle away from the local minima due to the nonconvex obstacles. In [17], an approach involving virtual local targets were used to lead the vehicle move out from obstacles. Unlike introducing additional virtual obstacles or targets, a probabilistic method approach was introduced in [15]. In this approach, when vehicles get trapped, they would explore all possible directions with equal probabilities. After certain time interval, the vehicles switch back to gradient decent approach. Despite the success demonstrated in simulation, these heuristic approaches usually are lack of rigorous analysis.

Recently, Chang proposed a new solution that combines the gyroscopic force and the gradient decent approach [69]. In this approach, the control input u for each vehicle consists of three parts as follows:

$$u = F_p + F_d + F_g,$$

Where F_p and F_d are forces coming from the gradient flow and the damping force respectively, the other term F_g is the gyroscopic force, which is orthogonal to the gradient decent direction. When the vehicle was trapped by the obstacles, the gyroscopic force drive the vehicle move around the obstacle to get out from local minima.

All these approach can only solve the local minima entrapment problem in some special cases, e.g., non-convex shape obstacles. There is no general way in the past research that could guarantee the final configuration approaches to the desired the configuration. This motivate us to solve the general problem. In the next chapter, a general solution is proposed and systemically studied to solve this problem.

3.4 Simulation Results

In [9], extensive simulations demonstrated the success of using the APF approach. By designing weighting coefficients of potential function components, in a battle field scenario, simulations showed that vehicles networks successfully enter into target area without being captured by the moving threats.

In this section, two simulations are conducted to show that UAV swarms fail to accomplish group task if the designed APF is nonconvex and has multiple local minima.

(1) a battle field scenario

In this example, a similar battle filed scenario as in [9] is used for demonstrating the fact. For simplicity, we do not assume that moving threats are circulating around the

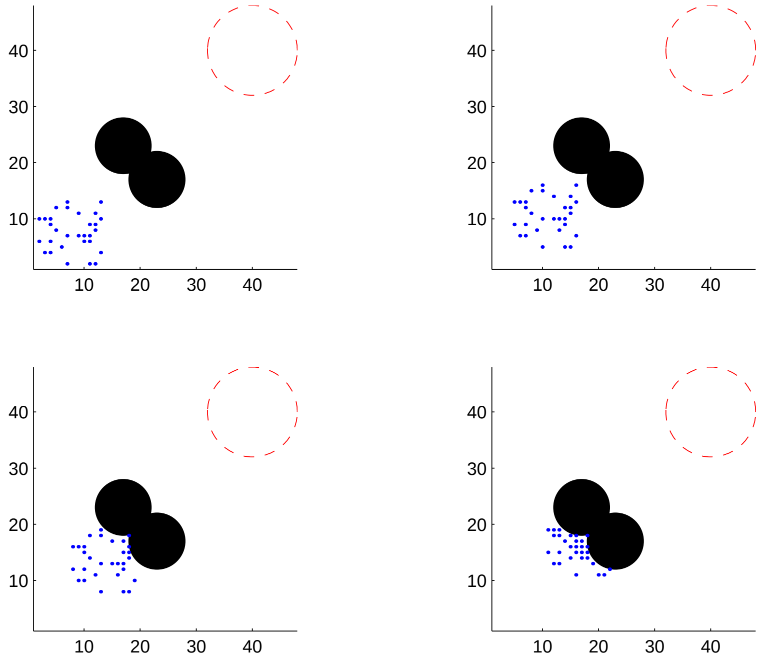

Fig. 3.3: Snap shots of vehicle swarms that are trapped by nonconvex shape obstacles

target area. Instead of circular obstacles, we introduce a non-convex shape of obstacle. In the simulation, there are 20 vehicles on a 48×48 area (see Fig. 3.1). The target is located at the corner (43,43) with radius $R_g = 5$, and two overlapped circular obstacles with radius 5 are centered at (17, 23) and (23, 17), respectively. Initially the vehicles are randomly distributed close to the other corner which is opposite to the target. The parameters used are: $\lambda_g = 10$, $\lambda_o = 1$, $\lambda_n = 0.2$, $R_m = \sqrt{2}$, $R_s = 6\sqrt{2}$. The mission goal is completed if all vehicles reach and gather in the target area. In addition, it is desired that vehicles have more neighbors.

The snapshots of the UAV swarms using the APF based approach is shown in figure 3.3. Clearly, due to the nonconvex shape of the obstacle, a part of vehicle networks are trapped, which makes the group task failed.

(2) Grid formation

The goal is to have the vehicles to form certain structure by simply define the inter-vehicle distance. In the simulation, 6 vehicle are moving on the 5 by 5 ground. Each vehicle can detect the distance with other vehicles. The desired inter-vehicle distance is set to be r_0. For simplicity, the inter-vehicle potential function $f_n(\cdot)$ for any two neighboring vehicle i and j has the following form,

$$f_n(p_i, p_j) = (\|p_i - p_j\| - r_0)^2. \tag{3.5}$$

Since only one term shows up in the APF, the coefficient λ_n is set to be 1. In the simulations, two equilibrium configurations are observed. See figure 3.4 and 3.5,where a hexagon-like configuration, and a star shape configuration were observed due to the different initial condition. The simulations confirm that the APF approach can only lead to local minima.

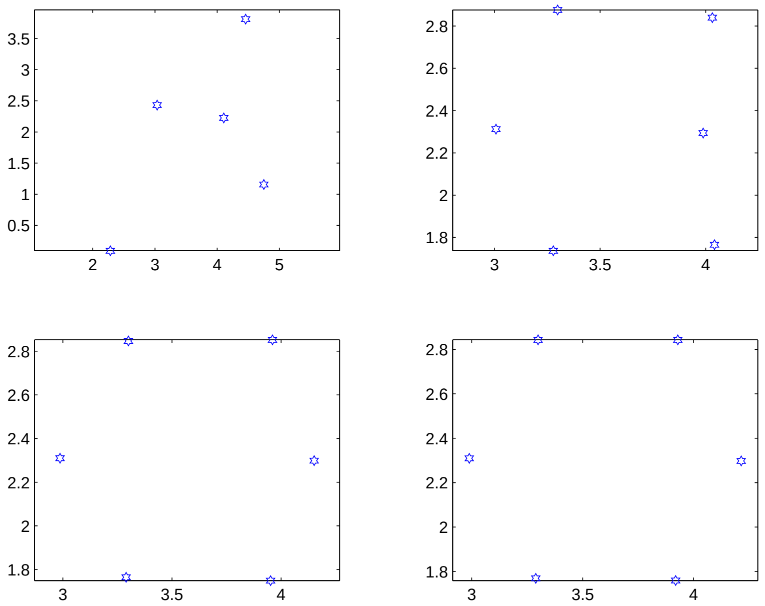

Fig. 3.4: Snap shots for a hexagon-like equilibrium configuration

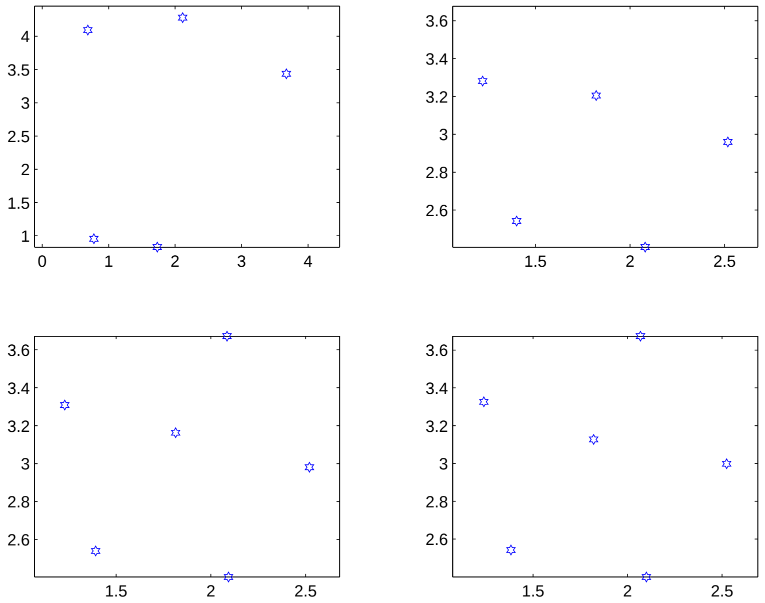

Fig. 3.5: Snap shots for a star-shape equilibrium configuration

Chapter 4

A Novel Gibbs Sampling Based Approach

To address the local minima entrapment dilemma existed in the gradient descent approach, In this chapter, a novel, systematic approach is proposed based on the theory of Markov random fields (MRFs) and Gibbs sampling, a tool that has been traditionally used in statistical mechanics, image processing and computer vision [18, 19, 20], and data analysis and clustering [21, 22]. A discrete-time path planning setting is considered, where vehicles are allowed to move on a discretized grid in a two-dimensional space (three-dimensional case can be easily extended). We focus on the high-level coordination and planning problem in this chapter, and thus the vehicle dynamics is not included in the analysis. In chapter 8, a model predictive control (MPC) based motion control approach is proposed for car-like wheeled vehicle's dynamics during the low-level vehicle control in executing the planned trajectory.

In the new approach, a swarm is modeled as a MRF on a graph, where the (mobile) vehicles and their communication/sensing links constitute the vertices and the edges [23] of the graph, respectively. As in the APF approach, global objectives and constraints (e.g., obstacles) are reflected through the design of potential functions - in this case, Gibbs potentials. The movement of vehicles is then decided using simulated annealing based on the Gibbs sampler. The dynamic graph associated with the evolution of vehicle networks, however, presents significant challenges in convergence analysis since classical MRF theory does not apply directly.

In this chapter, we first present analysis on the single-vehicle case before embarking on the more general multi-vehicle case. The aforementioned dynamic graph now degenerates to a single vertex and the analysis is much simplified. The purpose of studying the single-vehicle case is two-fold. Firstly, this is of interest in its own right by showing the convergence to the optimal location despite the existence of local minima, a clear advantage over the traditional gradient-flow method. Secondly, some analytical techniques and results for the single-vehicle case, e.g., the convergence bound and the impact of potential function design on convergence speed, also apply to the multi-vehicle case. Therefore, they provide insight into the latter case.

To deal with the self-organization of multiple vehicles, a two-step sampling scheme is proposed: in the first sampling step a vehicle is picked through a properly designed, *configuration-dependent* proposal distribution, and in the second sampling step the vehicle makes a move using the local characteristics of the Gibbs distribution. It is shown that simulated annealing with such a two-step scheme leads to convergence to the configuration(s) of minimal potential. This scheme requires mostly local interactions among vehicles except a global communication mechanism for notifying the newly selected vehicle. Such a mechanism could be provided, e.g., by a dedicated base station, or by individual vehicles with short-time, long-range communication capabilities.

The remainder of the chapter is organized as follows. Section 4.1 reviews basic mathematical background of MRF, Gibbs sampler and simulated annealing approach. The problem formulation and system modeling is addressed in section 4.2. The single-vehicle case is treated in section 4.3, while the multi-vehicle case studied in Section 4.4. Section

4.5 substantiates the convergence analysis with extensive simulation results.

4.1 Mathematical Background

4.1.1 Markov Random Field (MRF)

One can refer to, e.g., [20, 70], for a more comprehensive review of MRFs. Let S be a finite set of cardinality σ, with elements denoted by s and called *sites*, and let Λ be a finite set called the *phase space*. A *random field* on S with phases in Λ is a collection $X = \{X_s\}_{s\in S}$ of random variables X_s with values in Λ. A *configuration* of the system is $x = \{x_s, s \in S\}$ where $x_s \in \Lambda$, $\forall s$. The product space Λ^σ is called the *configuration space*.

A *neighborhood system* on S is a family $\mathscr{N} = \{\mathscr{N}_s\}_{s\in S}$, where $\mathscr{N}_s \subset S$, and $\forall s \in S$,

1. $s \notin \mathscr{N}_s$
2. $r \in \mathscr{N}_s$ if and only if $s \in \mathscr{N}_r$.

$\mathscr{N}_s$ is called the *neighborhood* of site s. The couple $(S, \mathscr{N})$ is called a *graph* or a *topology* , where S is the set of *vertices* and $\mathscr{N}$ defines *edges*. Site s and t are linked by a edge if and only if they are neighbors, i.e., $t \in \mathscr{N}_s$.

The random field X is called a *Markov random field* (MRF) with respect to the neighborhood system $\mathscr{N}$ if, $\forall s \in S$,

$$\Pi(X = x) > 0, \qquad \forall x \in \Lambda^\sigma$$

$$\Pi(X_s = x_s | X_r = x_r, r \neq s) = \Pi(X_s = x_s | X_r = x_r, r \in \mathscr{N}_s),$$

i.e., the conditional probabilities associated with the joint probability distribution of X are local in character and they obey the spatial Markovian relationship. The *local characteristic* of the MRF at site s is defined by the condition distribution function $\pi^s(x)$:

$$\pi^s(x) = \Pi(X_s = x_s | X_r = x_r, r \in \mathscr{N}_s)$$

The family $\{\pi^s(x)\}_{s \in S}$ is called the *local specification* of the MRF

The probability distribution π on a finite configuration space Λ^S is said to satisfy the *positive condition* if for any site s, $\pi(x_s) = 0$ implies $\pi(x_s, x_{S \setminus s}) = 0$, where $S \setminus s$ denotes the compliment of s in S. The positive condition guarantees the uniqueness of the MRF given the local specification specified.

4.1.2 Gibbs random field and Cliques

A random field X is a *Gibbs random field* if and only if its joint probability distribution is of the form:

$$\Pi(X = x) = \frac{e^{-\frac{U(x)}{T}}}{Z}, \forall x \in \Lambda^\sigma,$$

where T is the temperature variable (widely used in simulated annealing algorithms), $U(x)$ is the Gibbs potential (or energy) of the configuration x, and Z is the normalizing constant, called the *partition function*:

$$Z = \sum_{x \in \Lambda^\sigma} e^{-\frac{U(x)}{T}}.$$

The Gibbs field originally comes from physics, where it was introduced by Gibbs (1902). It was interesting for physics because the probability distribution of the special MRF is interpreted through energy and potential function which is useful in describing the local interaction of particles. In their study, the notion of *clique* plays a important role, which used to refer a small group of people who know each other.

A subset $c \subset S$ is called a *clique* with respect to a topology $(S, \mathscr{N})$ if every pair of distinct sites (s,t) in c are neighbors, except for single-site cliques $\{s\}$. With cliques defined, one can then construct potential functions $\Psi_c(x)$ associated with each clique to describe the local interaction among neighboring nodes in c. A *Gibbs potential* on configuration space Λ^S w.r.t. $(S, \mathscr{N})$ is a collection $\{\Psi_c(x)\}_{c \subset S}$ of functions $\Psi_c(x)$: $\Lambda^s \rightarrow R$ such that

1. $\Psi_c(x) = 0$ if c is not a clique

2. for all $x, x' \in \Lambda^S$, and all $c \subset S$, $x(c) = x'(c) \Rightarrow \Psi_c(x) = \Psi_c(x')$

The value of $\Psi_c(x)$ depends only on the local configuration on clique c, which guarantees the local nature of the Gibbs Field.

One then considers the following very useful class of potential functions

$$U(x) = \sum_{c \in \mathscr{C}} \Psi_c(x),$$

which is a sum of clique potentials $\Psi_c(x)$ over all possible cliques $\mathscr{C}$. A classical example, usually referred as *Ising Model*, was introduced by Ising (1925) for understanding qualitatively the phenomenon of phase transition in ferromagnetic materials. One can refer to A for details.

By the Hammersley-Clifford theorem [71], a Gibbs random field defined by a Gibbs distribution leads to an MRF with a consistent set of conditional probabilities; and conversely, for any MRF with a consistent set of conditional probabilities, there exists an equivalent Gibbs distribution expressed in terms of local potentials.

4.1.3 Gibbs Sampler

Consider a random field that changes randomly with time, which defines a stochastic process $\{X_n(s), s \in S\}_{n \geq 0}$. At each time instant n, the state X_n is a random field take values in Λ^S. The stochastic process $\{X_n\}$ in called a *dynamic random field*.

One then interested to generate a finite state Markov chain whose stationary distribution equals to a marginal distribution of a given random field. Suppose a homogenous Markov chain is irreducible and aperiodic, the fundamental Perron-Frobinius theorem shows that there exist an unique stationary distribution. This implies that from any initial distribution, the state distribution at time n will converges to the stationary distribution. The transition mechanism of the Markov chain is called a *sampling algorithm*, or a *simulation algorithm*. There are two major sampling algorithms in literature: Metropolis sampler, and Gibbs sampler. The latter is adopted in our approach.

The *Gibbs sampler* belongs to the class of *Markov Chain Monte Carlo* (MCMC) methods, which sample Markov chains leading to stationary distributions. The algorithm updates the configuration by visiting each site sequentially or with a strictly positive probability distribution ($q_s, s \in S$) on S, and sampling from the local conditional distributions of a Gibbs field. Picking an deterministic enumeration scheme for $S = s_1, \ldots, s_\sigma$, which is called a *visiting scheme* , the Gibbs sampler (basic version) works as following:

- Step 1. Pick an initial configuration $x \in \Lambda^\sigma$

- Step 2. Update the configuration by visiting each site $s_1 \ldots s_\sigma$ in turn specified by the visiting scheme. When visiting s_i, hold fixed the values at $s_j, j \neq i$, and change x_{s_i} to $z \in \Lambda$ with probability distribution $P(z)$ defined through the local characteristics:

$$P(z) = \frac{\Pi(z|r \in \mathscr{N}_{s_i})}{\sum\limits_{z' \in \Lambda} \Pi(z'|r \in \mathscr{N}_{s_i})},$$

- Step 3. Repeat step 2.

Note that in the above procedure the evaluation of $\Pi(z)$ is in general easy thanks to the local nature of potential functions. By incorporating the Gibbs distribution, it can be easily proved that the local sampling distribution in step 2 only depend on the clique potentials associated with current visiting site s_i:

$$P(z) = \frac{\exp(-\sum\limits_{c \in \mathscr{C}_{s_i}} \Psi_c(x_s = z))}{\sum\limits_{z' \in \Lambda} \exp(-\sum\limits_{c \in \mathscr{C}_{s_i}} \Psi_c(x_s = z'))},$$

where $\mathscr{C}_{s_i}$ denotes the set of cliques associated with site s_i. Let $\Phi_s(x) = \sum\limits_{c \in \mathscr{C}_s} \Psi_c(x)$ be the potential function associated with site s, the above formula can be simplified as following:

$$P(z) = \frac{\exp(-\Phi s_i(x_{s_i} = z))}{\sum\limits_{z' \in \Lambda} \exp(-\Phi_{s_i}(x_{s_i} = z'))}. \tag{4.1}$$

From the definition of $\Phi_s(x)$, it is straightforward to show that $\Phi_s(x)$ depends only the value on site s and its neighbors site $r \in \mathscr{N}_s$, i.e., $\Phi_s(x) = \Phi_s(y)$ whenever $x_s = y_s$, and $x_r = y_r, \forall r \in \mathscr{N}_s$.

Step 2 essentially defines the transition probabilities for a random field-valued Markov chain $X(n)$. The convergence of the Markov chain defined by Gibbs sampler was studied by D. Geman and S. Geman in the context of image processing [72]. By denoting one sequential visit to all sites as a *sweep*, it can be shown that as the number of sweeps goes to infinity, the distribution of $X(n)$ converges to the Gibbs distribution Π. Furthermore, with an appropriate cooling schedule, simulated annealing using the Gibbs sampler yields a uniform distribution on the space of configurations corresponding to the minimum energy $U(x)$. Thus the global objectives could be achieved through appropriate design of the Gibbs potential function.

4.1.4 Simulated Annealing approach

Both the gradient descent and the simulated annealing (SA) approach are popular optimization methods that numerically solve the linear and nonlinear optimization problem. As discussed in the last chapter, the gradient descent approach belongs to local optimization techniques. By taking steps proportional to the negative of the gradient flow, this approach always end up with local minimizer if the corresponding potential function is not convex.

The SA approach, on the other hand, numerically searches global optimizer given a function in a large search space, e.g., combinatorial optimization problem. It was S. Kirkpatrick, *et. al.* who first proposed SA approach in 1983 [73]. Then V. Cerny independently invented a similar approach in 1985[74]. The approach is inspired from the annealing procedure in metallurgy, a technique involving heating and slowly cooling of a material to get better mechanical performance. Inside the material, the annealing procedure results in larger size of crystals which have lower internal force.

By mimicking this physical process, the SA approach iteratively replace the current solution by randomly pick a local solution. The probability of making transition from current state x to a candidate state x' depends on a potential function difference $(U(x) - U(x'))$ and a temperature T. For example, in the Boltzmann annealing (BA) algorithm, the transition probability is expressed as follows.

$$p_{x \to x'} = \begin{cases} 1, & if\ U(x') < U(x); \\ e^{-\frac{U(x')-U(x)}{T}}, & if\ U(x') \geq U(x) \end{cases}$$

By gradually deceasing the temperature, the method asymptotically approaches to the global minimizer.

The rationale behind the approach lies in that, at each temperature, the transition probability corresponds to the solution of a entropy maximization problem constrained by the averaged energy under certain value. Using the lagrange multiplier, the solution can be solved by minimizing the lagrange $L = \beta E_p(U(x)) + S(p)$, where $S(p)$ is the entropy, and $\beta = 1/T$. Instead of greedily search the global minimizer along the negative gradient flow in the gradient descent approach, the random sampling in the SA approach minimizes a linear combination of the averaged energy and the entropy. As mentioned in the previous section, [18] proved that with logarithm cooling schedule, the SA approach converges to the global minimizer as temperature goes to zero.

Although the SA approach have been theoretical proved and practically applied in solving many optimization problems, it frequently suffers from the long computation time. To address the problem, studies on accelerating cooling schedule and parallel sampling have been visited by many researchers. One may refer to [75] for more detailed reviews of the SA approach.

4.2 Modeling vehicle swarms as MRF

Consider a 2D mission space (the extension to 3D space is straightforward), which is discretized into a lattice of cells. For ease of presentation, each cell is assumed to be square with unit dimensions. One could of course define cells of other geometries (e.g., hexagons) and of other dimensions (related to the coarseness of the grid) depending on the problems at hand. Label each cell with its coordinates (i, j), where $1 \leq i \leq N_1, 1 \leq j \leq N_2$, for $N_1, N_2 > 0$. There is a set of vehicles (or *mobile nodes*) S indexed by $s = 1, \cdots, \sigma$ on the mission space. To be precise, each vehicle (node) s is assumed to be a point mass located at the center of some cell (i_s, j_s), and the position of vehicle s is taken to be $p_s = (i_s, j_s)$. At most one vehicle is allowed to stay in each cell at any time instant.

The distance between two cells, (i_a, j_a) and (i_b, j_b), is defined to be the L_2 norm, in specific, $R \overset{\triangle}{=} \sqrt{(i_a - i_b)^2 + (j_a - j_b)^2}$. There might be multiple *obstacles* in the space, where an obstacle is defined to be a set of adjacent cells that are inaccessible to vehicles. For instance, a "circular" obstacle centered at (i^o, j^o) with radius R^o can be defined as $O \overset{\triangle}{=} \{(i, j) : \sqrt{(i - i^o)^2 + (j - j^o)^2} \leq R^o\}$. The *accessible area* is the set of cells in the mission space that are not occupied by obstacles. An *accessible-area graph* can then be induced by letting each cell in the accessible area be a vertex and connecting neighboring cells with edges. The mission space is *connected* if the associated accessible-area graph is connected, which will be assumed hereafter. There can be at most one *target area* in the space. A target area is a set of adjacent cells that represent desirable destinations of mobile nodes. A "circular" target area can be defined similarly as a "circular" obstacle. An example scenario is shown in Fig. 4.2.

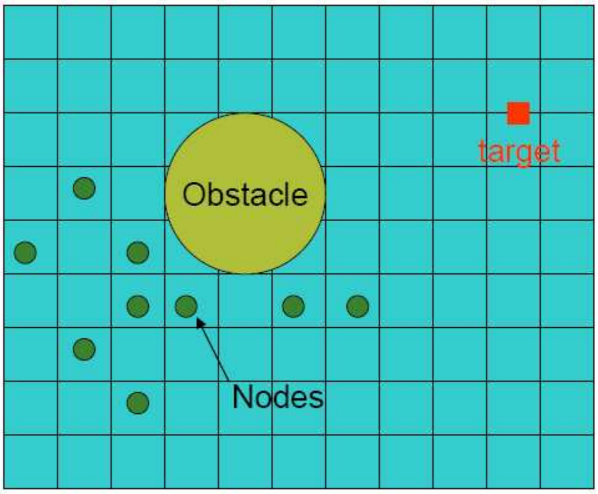

Fig. 4.1: An example mission scenario.

In this thesis, all nodes are assumed to be identical. Each node has a *sensing range* R_s: it can sense whether a cell within distance R_s is occupied by some node or obstacle. Communication between two nodes that are within a distance of R_s is regarded as local. The moving decision of each node s depends on other nodes located within distance R_i ($R_i \leq R_s$), called the *interaction range*. These nodes form the set $\mathscr{N}_s$ of *neighbors* of node s. A node can travel at most R_m ($R_m \leq R_s$), called *moving range*, within one move.

The neighborhood system defined earlier naturally leads to a dynamic graph, where each vehicle represents a vertex of the graph and the neighborhood relation prescribes the edges between vehicles. An MRF can then be defined on the graph, where each vehicle s is a site and the associated phase space Λ_s is the set of all cells located within the moving range R_m from location p_s and not occupied by obstacles or other vehicles. The configuration space of the MRF is denoted as $\mathscr{X}$.

The Gibbs potential $U(x) = \sum\limits_{c \in \mathscr{C}} \Psi_c(x)$, where $\Psi_c(x)$ is the clique potential and may

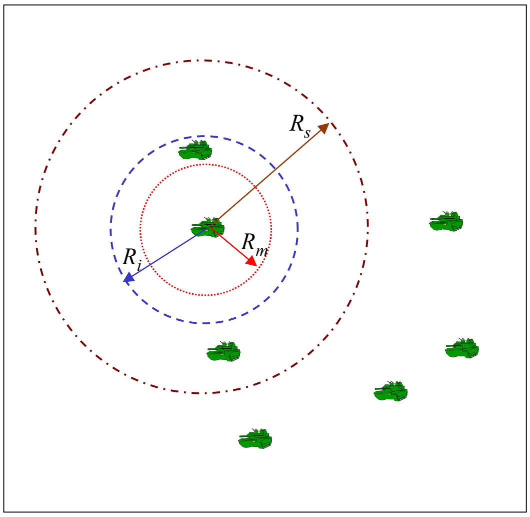

Fig. 4.2: Illustration of the sensing range R_s, the interaction range R_i, and the moving range R_m. Note since the mission space is a discretized grid, a cell is taken to be within a disk if its center is.

include different types of potential functions. To characterize the constraints and objectives as we mentioned earlier, the Gibbs potential can be formulated as a sum of three kinds of clique potentials. To be specific,

$$U(x) = \lambda_g \sum_{s\in S} \Psi_s^g + \lambda_o \sum_{s\in S} \Psi_s^o + \lambda_n \sum_{\langle s,t\rangle \in \mathcal{N}} \Psi_{(s,t)}^n, \tag{4.2}$$

$$\Phi_s(x) = \lambda_g \Psi_s^g + \lambda_o \Psi_s^o + \lambda_n \sum_{t\in \mathcal{N}_s} \Psi_{(s,t)}^n, \tag{4.3}$$

where Ψ_s^g, Ψ_s^o, and $\Psi_{(s,t)}^n$ account for the attraction from the target area, the repelling from obstacles, and the pulling force from neighbors, respectively. And $\langle s,t\rangle$ indicates a pair of sites s and t are neighbors of each other. $\lambda_g, \lambda_o, \lambda_n$ are the weighting coefficients for adjusting the potential surface. Note that the design of these constants is also a challenging and important issue as it may directly impact the nodes behavior and the convergence rate of the algorithm [9, 76]. The movement of each vehicle is then determined by the Gibbs sampler-based simulated annealing. Without loss of generality, it is assumed that there is one circular target area centered at p^g, and there are K (possibly overlapping) circular obstacles centered at p^{ok}, $1 \leq k \leq K$. The following potential functions are used:

$$
\begin{aligned}
J_s^g &= \|p_s - p^g\| \\
J_s^o &= \sum_{k=1}^{K} \frac{1}{\|p_s - p^{ok}\|} \\
J_s^n &= \begin{cases} \frac{1}{\sum\limits_{z \in \mathcal{N}_s} \frac{1}{\|p_s - p_z\|}}, & if \mathcal{N}_s \neq \emptyset \\ \Delta, & if \mathcal{N}_s = \emptyset \end{cases}
\end{aligned}
\tag{4.4}
$$

where J_s^n tends to be smaller when site s has more neighbors. $\Delta > 0$ is a relative large constant and it represents the penalty for having no neighbors.

Two major differences exist between a classical MRF introduced before and the MRF defined for the vehicle networks:

1. In a classical MRF, the neighborhood $\mathcal{N}_s$ depends only on s and not on x_s. However, for autonomous vehicles, $\mathcal{N}_s$ also depends on the phase x_s;

2. The phase space Λ is the same for each s and is time-invariant for a classical MRF. However, the phase space in vehicle networks is site-dependent and time-varying.

Due to these differences, classical MRF theory cannot be adopted directly to analyze the convergence behavior of the path planning algorithm, although simulations results in [77] showed a great success of the stochastic algorithm that lead autonomous swarms to the global objective without getting trapped by local minima.

4.3 Gibbs sampling based algorithm for Single Vehicle

4.3.1 Gibbs sampling based algorithm

For single vehicle case, the gibbs potentials only consist of clique potentials associated with singleton, namely obstacle potential, and target attraction potential, i.e.,

$$\Phi_s(x) = \lambda_g \Psi_s^g + \lambda_o \Psi_s^o.$$

So the mission goal is to reach the target area while avoiding the obstacles.

Starting from arbitrary initial position p_0, the vehicle make moves according to the following algorithm

- Step 1. Pick an cooling schedule $T(n) = \frac{c}{\log(n)}$, where c is a coefficient for adjusting the cooling rate. Let n=1.

- Step 2. At each time instant n, the vehicle moves from current location $x_s(n)$ to candidate location $z \in \Lambda$ (empty cells within the moving range) with probability distribution $P(z)$ defined through the local characteristics:

$$P(z) = \frac{e^{-\Phi_s(z)}}{\sum\limits_{z' \in \Lambda} e^{-\Phi_s(z')}},$$

- Step 3. Let n=n+1. Repeat step 2.

In the next subsection, analysis is provided to show that the vehicle will reach to the target for sure given arbitrary shape of obstacles provided the accessible area is connected.

4.3.2 Convergence Analysis

For the single-vehicle case, the MRF has only one site and the individual potential Φ_s consists of only first two terms of (4.2), which makes the analysis simpler. It can be shown that the Markov chain of the MRF defined by the Gibbs sampler converges to a stationary distribution Π_T as time tends to infinity for a fixed temperature T. By choosing an appropriate cooling schedule, the vehicle approaches the configuration(s) of minimum potential. This is made precise by the following theorems. The proofs of the theorems are adapted from some results on classical MRFs [20].

Theorem 4.3.1 *Assume that the accessible area is connected. For a fixed temperature T, let P_T be the kernel of the homogeneous Markov chain $X(n)$ defined by the Gibbs sampler. Then $X(n)$ has a unique stationary distribution Π_T:*

$$\pi_T(x) = \frac{e^{-\frac{U(x)}{T}} \sum\limits_{\|p_z - p_x\| \leq R_m} e^{-\frac{U(z)}{T}}}{Z_T}, \tag{4.5}$$

where Z_T is the partition function, defined as:

$$Z_T = \sum_{x \in X} (e^{-\frac{U(x)}{T}} \sum_{\|p_z - p_x\| \leq R_m} e^{-\frac{U(z)}{T}})$$

Furthermore, starting from any distribution ν

$$\lim_{n \to \infty} \nu P^n = \Pi_T. \tag{4.6}$$

Proof. Since the MRF has only one site, the Markov kernel P_T defined by Gibbs sampler is:

$$P_T(x,y) = \begin{cases} \frac{e^{-\frac{U(y)}{T}}}{\sum\limits_{\|p_z - p_x\| \leq R_m} e^{-\frac{U(z)}{T}}} & if \|p_y - p_x\| \leq R_m \\ 0 & if \|p_y - p_x\| > R_m \end{cases}. \tag{4.7}$$

Due to the connectivity of the accessible area, there exists at least one path between any two configurations x and y (i.e., a sequence of moves $\{x, x_1, \cdots, y\}$), and the shortest path is bounded by τ moves for some finite τ. This implies that P_T has a *strictly positive power* P_T^{τ}, i.e., the τ-step Markov chain reaches each state with positive probability from any state. Hence the Markov chain is ergodic and it has a unique invariant distribution Π_T, which implies (4.6). One can verify that (4.5) is a stationary distribution for P_T. □

Due to the limited moving range, τ sweeps will be performed for each temperature $T(n)$ in the simulated annealing to guarantee the convergence.

For a Markov kernel P, define its *contraction* coefficient $c(P)$ by

$$c(P) = (1/2)\max_{x,y} \|P(x,\cdot) - P(y,\cdot)\|_1,$$

where $P(x,\cdot)$ denotes the vector of conditional distributions $p(\cdot|x)$. The following lemma [20] will be useful in the proof of Theorem 4.3.2.

Lemma 4.3.1 *Let μ and ν be probability distributions, and P and Q be Markov kernels. Then*

$$\|\mu P - \nu P\|_1 \leq c(P)\|\mu - \nu\|_1,$$

$$c(PQ) \leq c(P)c(Q),$$

$$c(P) \leq 1 - |X| \min\{(P(x,y) : x, y \in X\},$$

and for a primitive P,

$$c(P^n) \to 0 \textit{ as } n \to \infty,$$

where X is the state space of the Markov chain and $|X|$ denotes its cardinality.

Theorem 4.3.2 *Assume that the accessible area is connected. Let $\{T(n)\}_{n\geq 1}$ be a cooling schedule decreasing to 0 such that eventually*

$$T(n) \geq \frac{\tau\Delta}{\ln n} \tag{4.8}$$

where $\Delta = \max_{x,y}\{|U(x)-U(y)| : \|p_x - p_y\| \leq R_m\}$[1]*, and τ is as defined in the proof of Theorem 4.3.1. Let $Q_n = P^{\tau}_{T(n)}$. Then from any initial distribution ν,*

$$\lim_{n\to\infty} \nu Q_1 \cdots Q_n = \Pi_\infty, \tag{4.9}$$

where Π_∞ is the distribution (4.5) evaluated at $T = 0$. Let M be the set of configurations achieving the minimum of $U(x)$. Assume that $\|p_x - p_y\| > R_m$, $\forall x,y \in M, x \neq y$. Then

$$\lim_{n\to\infty} \nu Q_1 ... Q_n = \begin{cases} \frac{1}{|M|} & \textit{if } x \in M \\ 0 & \textit{otherwise} \end{cases}. \tag{4.10}$$

Proof. Define $m_x = min\{U(z) : \|p_z - p_x\| \leq R_m\}$. Then, if $\|p_y - p_x\| \leq R_m$,

$$P_T(x,y) = \frac{e^{-\frac{U(y)-m_x}{T}}}{\sum\limits_{\|p_z-p_x\|\leq R_m} e^{-\frac{U(z)-m_x}{T}}} \geq |\overline{\Lambda}|^{-1} e^{\frac{-\Delta}{T}} \tag{4.11}$$

where $|\overline{\Lambda}|$ is the upper-bound on the cardinality of the phase space. For $Q = P^{\tau}_{T}$, from Lemma 4.3.1,

$$\begin{aligned} c(Q) &\leq 1 - |X| \min_{x,y} Q(x,y) \leq 1 - |X| (\min_{x,y} P_T(x,y))^{\tau} \\ &\leq 1 - \lambda e^{\frac{-\tau\Delta}{T}}, \end{aligned} \tag{4.12}$$

where $\lambda = |X| \cdot |\overline{\Lambda}|^{-\tau} \leq 1$. Then

$$\begin{aligned} \prod_{k=i}^{n} c(Q_k) &\leq \prod_{k=i}^{n} (1 - \lambda e^{\frac{-\tau\Delta}{T(k)}}) \leq \prod_{k=i}^{n} (1 - \lambda k^{-1}) \\ &\leq e^{-\sum\limits_{k=i}^{n} \lambda k^{-1}} \leq (\frac{i}{n})^{\lambda} \end{aligned} \tag{4.13}$$

[1] In this paper, $\|\cdot\|$ denotes the Euclidean norm, and $\|\cdot\|_1$ denotes the 1-norm.

where the third inequality holds because

$$(1-z) \leq e^{-z}, \forall z \in (0,1),$$

and the last one follows by

$$\begin{aligned} \ln(ni^{-1}) &< \ln(n+1) - \ln(i) = \sum_{k=i}^{n} (\ln(1+k) - \ln k) \\ &= \sum_{k=i}^{n} \ln(1+k^{-1}) \leq \sum_{k=i}^{n} k^{-1}. \end{aligned}$$

With some abuse of notation, $\Pi_{T(n)}$ will be written as Π_n to simplify the expressions. For $i \leq n$, one has

$$\begin{aligned} & \|\nu Q_1 ... Q_n - \Pi_\infty\|_1 \\ = \; & \|(\nu Q_1 ... Q_{i-1} - \Pi_\infty) Q_i ... Q_n + \Pi_\infty Q_i ... Q_n - \Pi_\infty\|_1 \\ \leq \; & \|(\nu Q_1 ... Q_{i-1} - \Pi_\infty)\|_1 c(Q_i ... Q_n) + \|\Pi_\infty Q_i ... Q_n - \Pi_\infty\|_1 \\ \leq \; & 2c(Q_i ... Q_n) + \|\Pi_i - \Pi_\infty\|_1 + \|\Pi_n - \Pi_\infty\|_1 + \sum_{k=i}^{k=n-1} \|\Pi_k - \Pi_{k+1}\|_1, \end{aligned} \tag{4.14}$$

where the last inequality is established by Dobrushin's theorem [20]. From (4.13), the first term of the last inequality approaches 0 as $n \longrightarrow \infty$. By Theorem (4.3.1), the second and third terms also approach 0 as $i, n \longrightarrow \infty$. To show that the fourth term vanishes as $i \longrightarrow \infty$, lemma 7.2.1 in [20] can be applied to prove the result. The only condition need to be investigated is the monotonicity and boundness of the denominator $(e^{-\frac{U(x)}{T}} \sum_{\|p_z - p_x\| \leq R_m} e^{-\frac{U(z)}{T}})$ of (4.5) with temperature T, which is straightforward to verify. This completes the proof of (4.9).

Let m denote the (global) minimal value of Gibbs potential $U(x)$. Then

$$\pi_T(x) = \frac{e^{-\frac{U(x)-m}{T}} \sum\limits_{\|p_z-p_x\| \le R_m} e^{-\frac{U(z)-m}{T}}}{\sum\limits_{y \in X} (e^{-\frac{U(y)-m}{T}} \sum\limits_{\|p_z-p_y\| \le R_m} e^{-\frac{U(z)-m}{T}})}$$

If $U(x)$ or $U(y)$ is a minimum then the respective exponential equals to 1 no matter what temperature T may be. The other exponentials decrease to 0 as T tends to 0. Eq. (4.10) then follows from (4.9) and $\|p_x - p_y\| > R_m$, $\forall x, y \in M$. $\square$

Note one can easily characterize Π_∞ when more than one minimizers of $U(x)$ are located within a distance of R_m. Theorem 4.3.2 establishes that under the algorithm, a vehicle can reach the target ultimately for *arbitrarily shaped* obstacles.

4.3.3 Convergence rate and Gibbs potential design

In Theorem 4.3.2 a condition on the cooling schedule $T(n)$ is specified to guarantee convergence. In the condition the maximal energy difference Δ among neighboring configurations plays an important role. It is of interested to study how to improve the convergence speed by appropriately designing the potential functions. The study here will be focused on the single-vehicle case.

Proposition 4.3.1 *For the single-vehicle case, the convergence speed is characterized by*

$$\|\nu Q_1 ... Q_n - \Pi_\infty\| = O(n^{-\frac{2\lambda\tilde{m}}{2\tilde{m}+\lambda\Delta\tau}}) \tag{4.15}$$

where $\tilde{m} = \min\limits_{y \neq x_m}(U(y) - m)$, *and* m, λ, Δ, *and* τ *are defined as in Theorem 4.3.2.*

Proof. From (4.14), $\|\nu Q_1 ... Q_n - \Pi_\infty\|$ is bounded by four terms. From (4.13), the first term is bounded by $(\frac{i}{n})^\lambda$. The other three terms basically have same convergence speeds because of the monotonicity of the stationary distribution $\Pi_n(x)$. For large i, the third term is dominated by

$$\begin{aligned} \|\Pi_{n+1}-\Pi_i\|_1 &\leq \|\Pi_{n+1}-\Pi_\infty\|_1+\|\Pi_i-\Pi_\infty\|_1 \\ &\leq 2\|\Pi_i-\Pi_\infty\|_1. \end{aligned}$$

Take $T(i)=\frac{\tau\Delta}{\ln i}$. If x is not a minimizer of U, $e^{-\frac{U(x)-m}{T(i)}} \leq e^{-\frac{\tilde{m}\ln i}{\tau\Delta}} = i^{-\frac{\tilde{m}}{\tau\Delta}}$. Then

$$\begin{aligned} &\|\Pi_k(x)-\Pi_\infty(x)\|_1 \\ =\; & \frac{e^{-\frac{U(x)-m}{T}} \sum\limits_{\|p_z-p_x\|\leq R_m} e^{-\frac{U(z)-m}{T}}}{\sum\limits_{y\in X}\big(e^{-\frac{U(y)-m}{T}} \sum\limits_{\|p_z-p_y\|\leq R_m} e^{-\frac{U(z)-m}{T}}\big)} \\ \leq\; & \frac{|\overline{\Lambda}| i^{-\frac{2\tilde{m}}{\Delta\tau}}}{|M|} = O(i^{-\frac{2\tilde{m}}{\Delta\tau}}) \end{aligned}$$

Similarly, it can be shown that for a minimizer x, the distance converges at the order $O(i^{-\frac{2\tilde{m}}{\Delta\tau}})$ too. And so is $\|\Pi_i-\Pi_\infty\|_1$. Then a bound for $\|\nu Q_1...Q_n-\Pi_\infty\|$ is

$$(\frac{i}{n})^\lambda + const\cdot i^{-\frac{2\tilde{m}}{\Delta\tau}} \tag{4.16}$$

This becomes optimal for

$$\begin{aligned} i^* &= (const\cdot\frac{2\tilde{m}}{\Delta\tau})^{\frac{1}{\lambda+\frac{2\tilde{m}}{\lambda\Delta\tau}}}\cdot n^{\frac{\lambda}{\lambda+\frac{2\tilde{m}}{\Delta\tau}}} \\ &= const\cdot n^{\frac{\lambda}{\lambda+\frac{2\tilde{m}}{\Delta\tau}}}. \end{aligned}$$

Eq. (4.15) then follows by plugging i^* into (4.16). □

Proposition 4.3.1 shows that the potential surface (in particular: $g \overset{\triangle}{=} \frac{2\lambda\tilde{m}}{2\tilde{m}+\lambda\Delta\tau}$) determines the convergence speed of the algorithm. It is thus natural to use g as a design indicator. Simulation has been conducted to verify the above analysis. In the simulation a

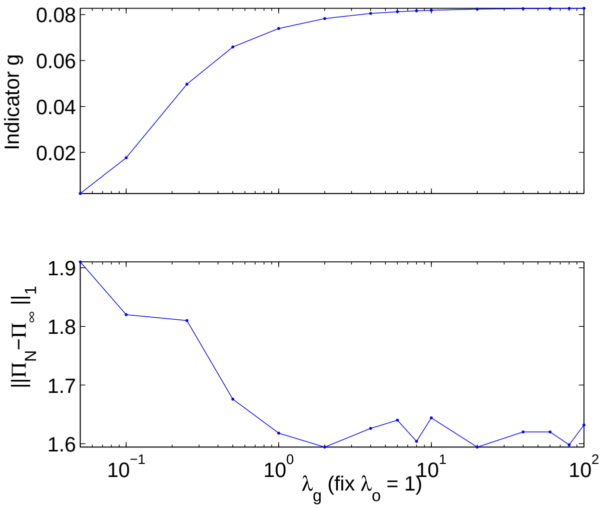

Fig. 4.3: Convergence *vs.* design parameter - comparison of simulation results with analysis.

similar scenario as in Fig. 4.2 is used but with a 10 by 10 grid. Two overlapping circular obstacles are located in the middle of the field, which forms a non-convex shape. A single vehicle starts from one corner, and wants to reach the target area at the other corner. The potential functions used are: $J_s^g = \|p_s - p^g\|$, $J_s^o = \sum_{k=1}^{K} \frac{1}{\|p_s - p^{ok}\|}$, where p^g and p^{ok} denote the centers of the target area and of the obstacles. λ_g is varied from 0.05 to 100 while λ_o is fixed to 1. For each pair of coefficients, the algorithm is run $N = 10,000$ steps, and the number of times w that the vehicle visited target during the last 100 steps is counted. The empirical distance is then given by $\|\Pi_N - \Pi_\infty\|_1 = 2(1 - w/100)$. Comparison with the numerically calculated design indicator g reveals good agreement with the bound (4.15) (Fig. 4.3).

4.4 A Novel Gibbs sampling algorithm for multi-vehicle system

In the previous section, preliminary results show the global convergence of single vehicle case. However, it is still not clear about the convergence of the general case of multi-vehicles with limited moving/sensing range. In this section, a novel distributed stochastic algorithm is presented. The algorithm requires only limited sensing, communication, and moving ranges for vehicles, and a mechanism for minimal full-range information transfer. Such a mechanism could be provided, e.g., by a dedicated base station, or by individual vehicles with short-time, long-range communication capabilities. It is shown that the algorithm, with primarily local interactions, leads to globally optimal vehicle configurations represented by the global minima of Gibbs potentials.

4.4.1 A novel decentralized algorithm

The algorithm uses a randomized sequence for updating the nodes, and a key idea involved is the *configuration-and-temperature-dependent* proposal distribution $G_T^x(s)$. In particular, given a configuration x and a temperature T,

$$G_T^x(s) = \frac{\sum_{z \in \mathcal{N}_m^x(s)} e^{-\frac{U(z)}{T}}}{\sum_{s' \in S} \sum_{z \in \mathcal{N}_m^x(s')} e^{-\frac{U(z)}{T}}}. \tag{4.17}$$

In (4.17) $\mathcal{N}_m^x(s)$ denotes the set of *s-neighbors of configuration x within one move*:

$$\mathcal{N}_m^x(s) \overset{\triangle}{=} \{z : z_{S\backslash s} = x_{S\backslash s}, \|z_s - x_s\| \leq R_m\},$$

where $S\backslash s$ denotes the set of all nodes except s.

Since for $z \in \mathcal{N}_m^x(s)$, $U(z) - U(x) = \Phi_s(z) - \Phi_s(x)$, (4.17) can be rewritten as

$$\begin{aligned} G_T^x(s) &= \frac{\sum_{z \in \mathcal{N}_m^x(s)} e^{-\frac{U(z)-U(x)}{T}}}{\sum_{s'} \sum_{z \in \mathcal{N}_m^x(s')} e^{-\frac{U(z)-U(x)}{T}}} \\ &= \frac{\sum_{z \in \mathcal{N}_m^x(s)} e^{-\frac{\Phi_S(z)-\Phi_S(x)}{T}}}{\sum_{s'} \sum_{z \in \mathcal{N}_m^x(s')} e^{-\frac{\Phi_S(z)-\Phi_S(x)}{T}}}. \end{aligned}$$

Note that, from (4.2), each node s would be able to evaluate $D_T^x(s) = \sum_{z \in \mathcal{N}_m^x(s)} e^{-\frac{\Phi_S(z)-\Phi_S(x)}{T}}$ if $R_s \geq R_i + R_m$.

In sampling node s is first randomly selected with probability $G_T^x(s)$, and then x_s is updated according to its local characteristics while $x_{S \backslash s}$ is kept fixed:

$$P(x_s = l) = \frac{e^{-\frac{\Phi_S(x_S = l, x_{S \backslash s})}{T}}}{\sum_{l' \in C_m^s} e^{-\frac{\Phi_S(x_S = l', x_{S \backslash s})}{T}}}, \tag{4.18}$$

where C_m^s is the set of candidate locations node s can take, i.e., $l \in C_m^s$ is not occupied by any obstacle or other nodes, and $\|x_s - l\| \leq R_m$. One can verify that there exists a smallest integer τ, such that after τ steps of sampling, any configuration x has a positive probability of becoming any other configuration y.

The self-organization algorithm works as follows. Pick an appropriate cooling schedule $T(n)$ with $T(n) \to 0$ as $n \to \infty$. Pick a sufficiently large N_{max}. For each temperature $T(n)$, run τ steps of sampling as described above (this will be called *one annealing step*). To be specific:

- **Step 1. Initialization.** Start with an arbitrary configuration $x(0)$ and let $n = 1$, $k = 1$. Pick an arbitrary node $s(0)$. Have all nodes to evaluate and send $D_{T(1)}^{x(0)}(s)$ to $s(0)$. Node $s(0)$ calculates the proposal distribution $G_{T(1)^{x(0)}}(s)$ according to

(4.17),namely,

$$G^{x(0)}_{T(1)}(s) = \frac{D^{x(0)}_{T(1)}(s)}{\sum_{s'} D^{x(0)}_{T(1)}(s')}.$$

Node $s(0)$ then selects a node $s^1(1)$ [2] for updating by sampling the distribution $G^{x(0)}_{T(1)}(s)$, and it sends the vector $\{D^{x(0)}_{T(1)}(s), s \in S\}$ to $s^1(1)$;

- **Step 2. Updating the selected node.** Node $s^k(n)$ updates its location by sampling its local characteristics (see (4.18)). Denote the new configuration as $x^k(n)$;

- **Step 3. Selecting the next node.** Note that the neighborhood $\mathcal{N}_s$ of a node s changes *only if* node $s^k(n)$ was in $\mathcal{N}_s$ before its updating or is currently in $\mathcal{N}_s$. For either case, the distance between such s (denoting the set of such nodes as $\bar{\mathcal{N}}^k(n)$) and $s^k(n)$ is now no greater than $R_i + R_m \leq R_s$ and they can communicate *locally*. The node $s^k(n)$ thus collects and updates $D^{x^k(n)}_{T(n)}(s)$ for nodes in $\bar{\mathcal{N}}^k(n)$. Let $k = k+1$. If $k = \tau$, let $k = 0$ and $n = n+1$. The current node evaluates and samples new proposal distribution, selects the next node to be updated, and communicates the updated $\{D_{T(n)(s)}\}$ to the next node (the superscript of D is omitted when it is clear from the context);

- **Step 4.** If $n < N_{max}$, go to Step 2; otherwise quit.

Remark 4.4.1 *Long-range (over a distance greater than R_s) communication is only required for initialization and for transferring $\{D^x_T(s)\}$ to the newly selected node at each step. Since $\{D^x_T(s)\}$ is just a σ-dimensional vector, information exchange in the algorithm is primarily at the local level. The (minimal) global communication can be achieved*

[2] In the notation $x^k(n)$ or $s^k(n)$, n indexes the annealing temperature, while k (from 1 to τ) indexes the sampling step within a fixed temperature.

through, e.g., fixed base stations, or individual vehicles with short-time, long-range transmission capabilities.

4.4.2 Convergence Analysis

Let P_T denote the Markov kernel defined by the random update scheme (4.17) and (4.18), i.e.,

$$\begin{aligned}
P_T(x,y) &\overset{\triangle}{=} Pr(X(n+1)=y|X(n)=x) \\
&= \sum_{s\in S} G_T^x(s)\cdot \mathbf{1}(y\in \mathcal{N}_m^x(s)) \frac{e^{-\frac{U(y)}{T}}}{\sum_{z\in\mathcal{N}_m^x(s)} e^{-\frac{U(z)}{T}}} \\
&= \frac{\sum_{s\in S}\sum_{z\in\mathcal{N}_m^x(s)} e^{-\frac{U(z)}{T}}}{\sum_{s'\in S}\sum_{z\in\mathcal{N}_m^x(s')} e^{-\frac{U(z)}{T}}} \cdot \frac{e^{-\frac{U(y)}{T}}\cdot \mathbf{1}(y\in\mathcal{N}_m^x(s))}{\sum_{z\in\mathcal{N}_m^x(s)} e^{-\frac{U(z)}{T}}} \\
&= \sum_{s\in S} \frac{e^{-\frac{U(y)}{T}}\cdot \mathbf{1}(y\in\mathcal{N}_m^x(s))}{\sum_{s'\in S}\sum_{z\in\mathcal{N}_m^x(s')} e^{-\frac{U(z)}{T}}}.
\end{aligned} \tag{4.19}$$

Let τ be the integer as selected in Section 4.4.1, and let $Q_T = P_T^\tau$.

Theorem 4.4.1 *The Markov kernel Q_T has a unique stationary distribution Π_T with*

$$\Pi_T(x) = \frac{e^{-\frac{U(x)}{T}}\sum_{s\in S}\sum_{z\in\mathcal{N}_m^x(s)} e^{-\frac{U(z)}{T}}}{Z_T}, \tag{4.20}$$

where $Z_T = \sum_y e^{-\frac{U(y)}{T}}\sum_{s\in S}\sum_{z\in\mathcal{N}_m^y(s)} e^{-\frac{U(z)}{T}}$ is the partition function.

Proof. First one can show that Π_T is a stationary distribution of P_T. From (4.19) and (5.2),

$$\begin{aligned} & \sum_y \Pi(y) P_T(y,x) \\ = \quad & \sum_y \frac{e^{-\frac{U(y)}{T}} \sum_{s'' \in S} \sum_{z \in \mathscr{N}_m^y(s'')} e^{-\frac{U(z)}{T}}}{Z_T} \cdot \\ & \sum_{s \in S} \frac{e^{-\frac{U(x)}{T}} \cdot \mathbf{1}(x \in \mathscr{N}_m^y(s))}{\sum_{s' \in S} \sum_{z \in \mathscr{N}_m^y(s')} e^{-\frac{U(z)}{T}}} \\ = \quad & \frac{e^{-\frac{U(x)}{T}} \sum_y e^{-\frac{U(y)}{T}} \sum_{s \in S} \mathbf{1}(x \in \mathscr{N}_m^y(s))}{Z_T} \\ = \quad & \frac{e^{-\frac{U(x)}{T}} \sum_{s \in S} \sum_{z \in \mathscr{N}_m^x(s)} e^{-\frac{U(z)}{T}}}{Z_T} = \Pi_T(x). \end{aligned}$$

Since $Q_T = P_T^\tau$, Π_T is also a stationary distribution for Q_T. Due to the choice of τ, $Q_T(x,y) > 0$, $\forall x,y$. Thus from the Perron-Frobenius theorem, Q_T has a unique stationary distribution, which is Π_T. □

Let Δ be the *maximal local oscillation* of the potential U:

$$\Delta \overset{\triangle}{=} \max_x \max_{y \in \mathscr{N}_m^x} |U(x) - U(y)|,$$

where $\mathscr{N}_m^x = \cup_{s \in S} \mathscr{N}_m^x(s)$.

Theorem 4.4.2 *Let $\{T(n)\}$ be a cooling schedule decreasing to 0 such that eventually,*

$$T(n) \geq \frac{\tau \Delta}{\ln n}.$$

Let $Q_n = P_{T(n)}^\tau$, and let $\mathscr{M}$ be the set of global minima of $U(\cdot)$. Then for any initial distribution ν,

$$\lim_{n \to \infty} \nu Q_1 \cdots Q_n \to \nu_\infty, \tag{4.21}$$

where ν_∞ is the distribution (5.2) evaluated at $T = 0$. In particular,

$$\sum_{x \in \mathscr{M}} \nu_\infty(x) = 1. \tag{4.22}$$

Proof. Let $\alpha_x = \min_{y \in \mathscr{N}_m^x} U(z)$. For $y \in \mathscr{N}_m^x$, from (4.19),

$$P_T(x,y) = \frac{e^{-\frac{U(y)-\alpha_x}{T}}}{\sum_{s' \in S} \sum_{z \in \mathscr{N}_m^x(s')} e^{-\frac{U(z)-\alpha_x}{T}}} \geq \frac{e^{-\frac{\Delta}{T}}}{\sigma|\mathscr{X}|},$$

where $|\mathscr{X}|$ denotes the cardinality of the configuration space $\mathscr{X}$. For $Q_T = P_T^\tau > 0$,

$$\min_{x,y} Q_T(x,y) \geq (\min_{x',y' \in \mathscr{N}_m^{x'}} P_T(x',y'))^\tau \geq \frac{e^{-\frac{\tau\Delta}{T}}}{(\sigma|\mathscr{X}|)^\tau}.$$

Let $C(Q_T)$ denotes the *contraction coefficient* [20] of Q_T, i.e.,

$$C(Q_T) \stackrel{\triangle}{=} \frac{1}{2} \max_{x,y} \|Q_T(x,\cdot) - Q_T(y,\cdot)\|_1.$$

Using Lemma 4.2.3 of [20], one has

$$C(Q_T) \leq 1 - |\mathscr{X}| \min_{x,y} Q_T(x,y) \leq 1 - \lambda e^{-\frac{\tau\Delta}{T}},$$

where $\lambda = \frac{|\mathscr{X}|}{(\sigma|\mathscr{X}|)^\tau} < 1$. This implies $C(Q_n) \leq 1 - \lambda e^{-\frac{-\tau\Delta}{T(n)}}$. The claim (5.4 can then be proved following the proof of Theorem 3.2 in [76]. As $T(n) \to 0$, $\Pi_{T(n)}(x) \to 0$, for all $x \notin \mathscr{M}$, as one can verify from (5.2). Eq. (5.5) thus follows. □

From Theorem 5.1.1, the distributed algorithm can achieve global objectives provided that the global minimizers of $U(\cdot)$ correspond to the desired configurations.

4.4.3 Convergence rate of the novel algorithm

In section 4.4, we mentioned that the algorithm is a special case of single site MRF. The convergence rate bound can then be followed by the proposition 4.3.1.

Let $\tilde{m} = \min_{x \notin \mathscr{M}} U(x) - m$, i.e., the minimal potential difference between other configurations and the global minimizers. The following result characterizes the convergence speed of the distributed algorithm:

Proposition 4.4.1 *Consider the distributed self-organization algorithm with a cooling schedule $T(n) = \frac{\tau\Delta}{\ln n}$. Then the following estimate holds for any initial distribution ν:*

$$\|\nu Q_1 \cdots Q_n - \Pi_\infty\| = O(n^{-\frac{2\lambda\tilde{m}}{2\tilde{m}+\lambda\Delta\tau}}) = O(n^{-g}), \qquad (4.23)$$

where $\lambda, \tau,$ and Δ are as defined in Theorem 5.1.1, and $g = \frac{2\lambda\tilde{m}}{2\tilde{m}+\lambda\Delta\tau}$ is called the indicator of convergence speed in this proposal

4.5 Simulation results

Simulation was conducted to verify the analysis. The emphasis was on scenarios involving inter-vehicle interactions (e.g., formation control). Two examples are presented, one on clustering and the other on formation control. Other objectives or constraints, such as target-approaching and obstacle avoidance, can be easily incorporated, as was done in the single-vehicle case.

Clustering

The goal is to cluster all the nodes without specifying a specific target area. This is more challenging than the case of having an explicit target, as the latter provides persistent attraction from a fixed location. The potential function used was:

$$U(x) = \sum_{r \neq s,\, \|x_r - x_s\| \leq R_i} -\frac{c}{\|x_r - x_s\|},$$

where $c > 0$ is some constant. Clearly, the more neighbors each node has and the closer they are, the lower the potential U. Simulation was performed for 50 nodes on a 30 by 30 grid, and the following parameters were used: $R_I = 4\sqrt{2} + \varepsilon$ ($\varepsilon > 0$ and very small), $R_m = 2\sqrt{2} + \varepsilon$, $R_s = R_I + R_m$ (this was also the case for all other simulation results presented in this section), $c = 2$, annealing schedule $T(n) = \frac{1}{0.08 \ln n}$, and $\tau = 50$.

Fig. 4.4 shows the snapshots of the network evolution. The algorithm's ability to overcome local minima is evident from the figure: the nodes initially evolved into two separated (farther than R_s) sub-clusters, and yet they merged into one cluster after 500 annealing steps.

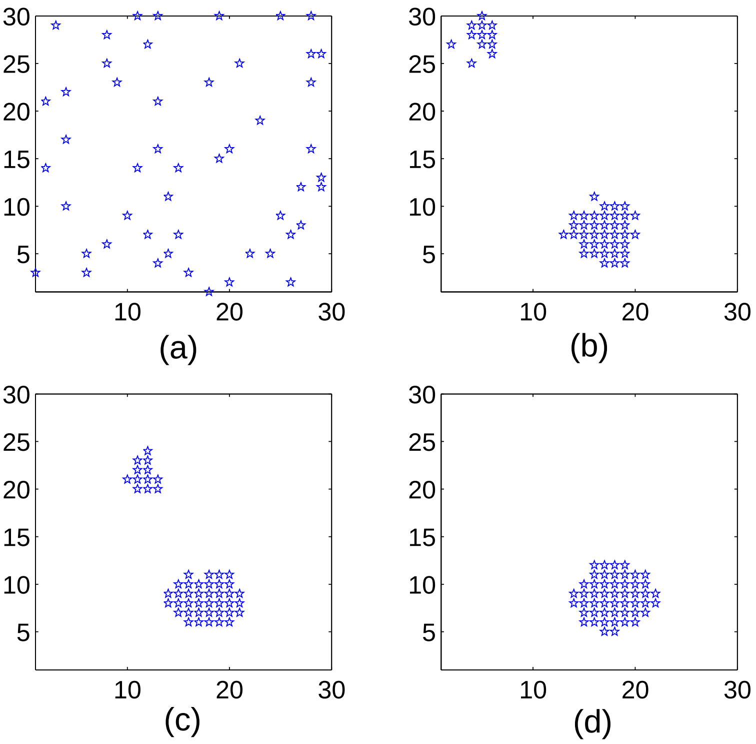

Fig. 4.4: Snapshots of clustering operation. (a) initial configuration; (b) after 100 annealing steps; (c) after 400 annealing steps; and (d) after 500 annealing steps.

Formation

The goal is to have the nodes to form (square) lattice structures with a desired inter-vehicle distance R_{des}. The potential function used was:

$$U(x) = \sum_{r \neq s,\, \|x_r - x_s\| \leq R_i} c_1(|\|x_r - x_s\| - R_{des}|^{\alpha} - c_2),$$

where $c_1 > 0$, $c_2 > 0$, and $\alpha > 0$. A proper choice of c_2 encourages nodes to have more neighbors. The power α shapes the potential function. In particular, for $|\|x_r - x_s\| - R_{des}| < 1$, smaller α leads to larger potential difference from the global minimum.

Simulation was first conducted for 9 nodes on an 8 by 8 grid. Parameter used were: $R_i = 2\sqrt{2} - \varepsilon$, $R_m = \sqrt{2} + \varepsilon$, $R_{des} = 2$, $c_1 = 10$, $c_2 = 1.05$, $\alpha = 0.02$, $T(n) = \frac{1}{0.01 \ln n}$, and $\tau = 20$. The desired configuration (global minimizer of U) is shown in Fig. 4.5 (modulo vehicle permutation and formation translation on the grid). Simulated annealing was performed for 10^4 steps. Empirical distributions with respect to configuration potentials were calculated based on the average of every 2500 steps (Fig. 4.6). The trend of convergence to the configuration of the lowest potential is clear from Fig. 4.6. One can further calculate the error $\|\nu_n - \Pi_\infty\|_1$, where ν_n is the empirical distribution of configurations (again modulo vehicle permutation and network translation). In particular, ν_n is only calculated for $n = 1000, 2000, 3000, \cdots$, and for a configuration x, $\nu_n(x)$ equals the relative frequency of x between the $(n-1000)$-th and the n-th sampling steps. From Theorem 5.1.1, Π_∞ satisfies:

$$\Pi_\infty(x) = \begin{cases} 1 \text{ if } x \text{ is desired} \\ 0 \text{ otherwise} \end{cases} .$$

Therefore,

$$\|\nu_n - \Pi_\infty\|_1 = 1 - \nu_n(x^*) + |0 - (1 - \nu_n(x^*)| = 2(1 - \nu_n(x^*)), \tag{4.24}$$

where x^* denotes the desired formation. The evolution of $\|\nu_n - \Pi_\infty\|_1$ is shown in Fig. 4.7.

It is of interest to investigate the robustness of the proposed scheme with respect to the discretization level (i.e., coarseness) of the grid. For this purpose, the same mission space was discretized into a 16 by 16 grid with cell size of 0.5, and simulation was performed based on the new lattice. All mission-related parameters were kept the same as in the 8 by 8 case. In particular, the exclusion condition (one cell cannot contain more than one vehicle) on the 8 by 8 grid was properly translated into a condition for the new grid. Five simulation runs were performed for both discretization schemes, and each was run for 10^4 annealing steps. For each simulation, the evolution of $\|\nu_n - \Pi_\infty\|_1$ was calculated according to (4.24) and the total annealing time was recorded. Fig. 4.8 compares the performance under the two discretization schemes, where $\|\nu_n - \Pi_\infty\|_1$ and the annealing time were averaged over the five runs for each case. The results indicate that the convergence to the desired formation can be achieved under both discretization schemes, while it takes longer computational time for the finer discretization.

Simulation was also performed for a group of 20 vehicles on a 15 by 15 grid, and Fig. 4.9 shows the snapshots after different annealing steps. One can see that the group achieves an almost optimal configuration after 2000 steps.

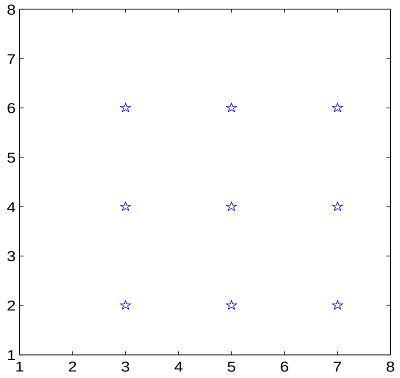

Fig. 4.5: The desired formation for 9 vehicles on an 8 by 8 grid.

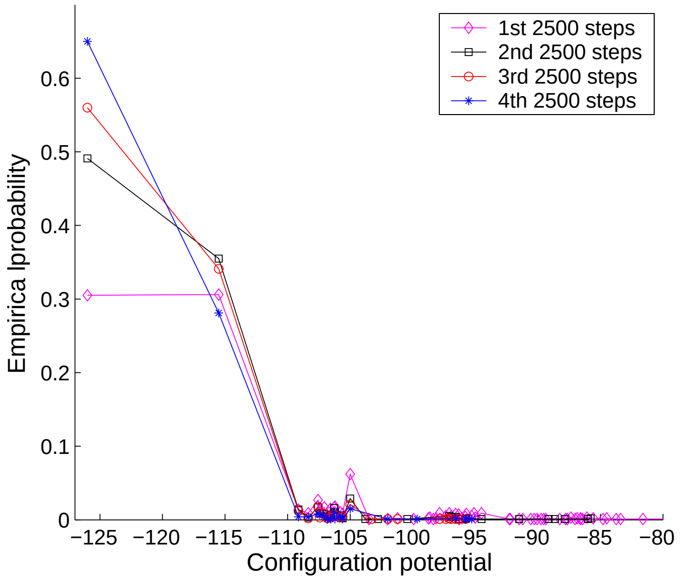

Fig. 4.6: Evolution of the empirical distribution of configuration potentials.

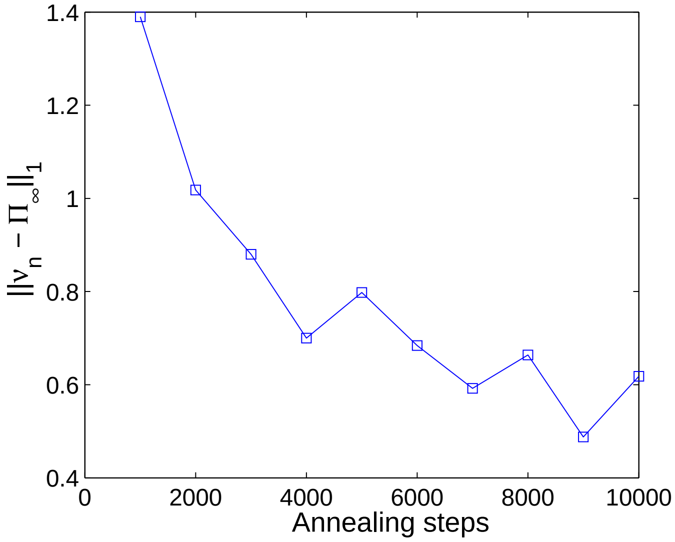

Fig. 4.7: Evolution of $\|v_n - \Pi_\infty\|_1$.

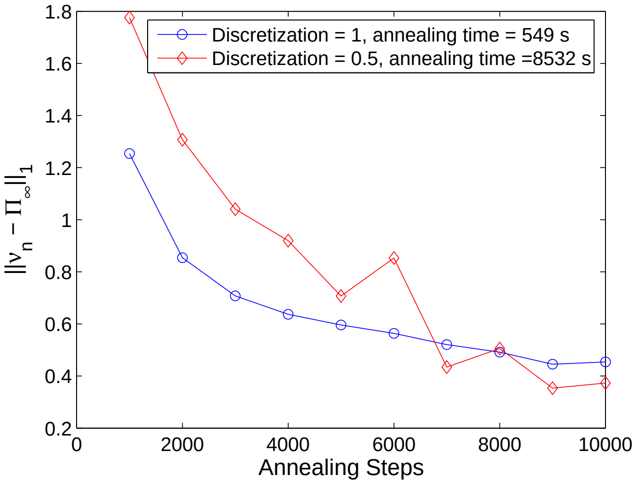

Fig. 4.8: Comparison of annealing schemes with different discretization levels.

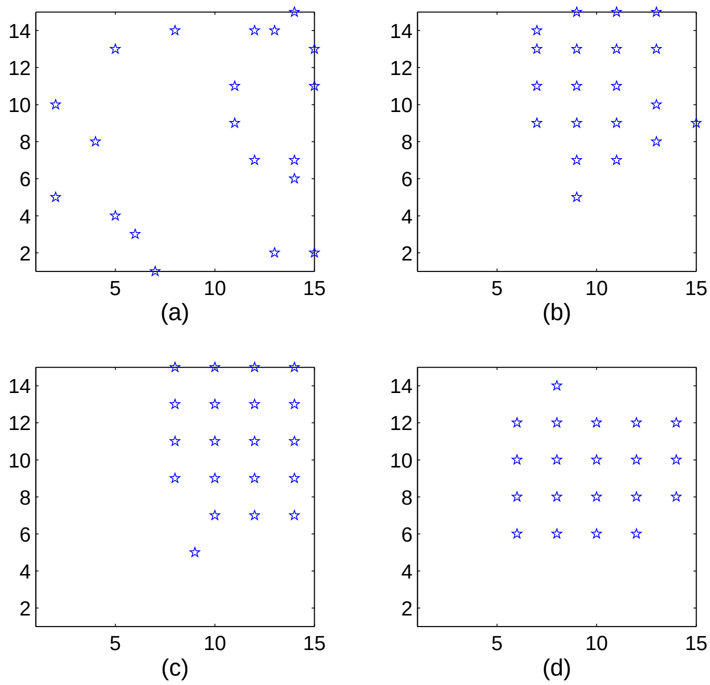

Fig. 4.9: Snapshots of formation operation. (a) Initial configuration; (b) after 1000 annealing steps; (c) after 2000 annealing steps; (d) after 3000 annealing steps.

4.6 Summary

In this chapter, a stochastic algorithm was presented for coordination of autonomous swarms. The algorithm was based on the Gibbs sampler with a random visiting scheme. The specific choice of the proposal distribution results in Gibbs-type distributions for vehicle configurations, leading to the convergence of the algorithm.

The gibbs sampling based stochastic algorithm belongs to a reactive coordination approach. At each time instance, vehicles only need current state information, including its own and neighbors' location, and the distance to the obstacles, to do gibbs sampling. The algorithm do not require to have the full information of the environments or history data. Therefore, the data processing time for making decision can be greatly reduced, which is very beneficial in real-time control.

Moreover, since the algorithm do not require UAV use memories storing the environments information, e.g., obstacle position and shapes, trajectory history, and maps, the limited memory spaces can and save as well as the manufacturing cost.

The random visiting scheme entails long-range communication for notifying newly selected nodes although such information exchange is minimal. A deterministic sequential visiting scheme would eliminate this requirement; however, the convergence behavior would be unclear since the stationary distribution for each T is no longer of the Gibbs-type.

However, to guarantee that vehicle swarms achieve the group mission goal, long maneuvering time have to be paid. This is primarily due to the stochastic nature of the algorithm. This problem is addressed in the next two chapters. Two different approaches are investigated to improve the efficiency and the performance comparing with the original approach. Substantial improvements are observed via extensive simulations.

In this chapter, the coordination schemes presented are meant for high-level path planning. They should be combined with low-level planning and control modules in implementation. Since vehicles move on a discrete lattice, the choice of discretization level becomes an important problem. From the simulation results, a practical and efficient strategy is to use the coarsest discretization that is compatible with the problem setup (e.g., the minimum inter-vehicle separation). The cell geometry is another factor in discretization. In this paper the mission space was discretized into a lattice with square cells. One could use cells of other geometries without changing the algorithm (except the numbering scheme for cells) to implement, e.g., triangular or hexagonal formations.

Chapter 5

Parallel Gibbs Sampling Approach

In the last chapter, a novel Gibbs sampling based stochastic approach was proposed to coordinate vehicle networks to perform group tasks. Theoretical studies have shown that, with a special sequential sampling scheme, it was possible to achieve global goals despite the presence of local minima of potentials, which is further confirmed with simulations [76, 78]. However, to maintain global indices in a large vehicle networks, which is crucial for sequential sampling, is difficult when there exist node failures. Moreover, the nature of sequential sampling lead to long executing time which presents difficulties in practice.

The above problems can be solved using parallel sampling [77], i.e., each node in the vehicle swarms does the local Gibbs sampling in parallel. Parallel sampling techniques have been studied for many years in order to accelerate the slow convergence rate of sequential simulated annealing algorithm [79]. It is usually required that nodes update their locations at the same time clock (synchronously). However, the synchronization incurs the communication cost and the delay which degrade the performance. This can be solved by using the asynchronous parallel sampling, i.e., each vehicle takes its own clock to do the local sampling. Extensive simulations suggest that global goals (by this, we mean self-organization of autonomous swarms) can be achieved by the parallel sampling algorithm, whereas it has to be noted that the parallel simulation annealing algorithm might not lead to the global minimizer.

In this chapter, we firstly investigated the convergence properties of a synchronous parallel sampling algorithm. In the study of the asynchronous parallel algorithm, the fact that "time-varying" number of active nodes presents challenges. Fortunately, by applying partial parallel model in [80], the asynchronous algorithm could be described by a homogeneous Markov chain. The convergence of asynchronous parallel algorithm then follows. Finally, a special example based on a battle field scenario was investigated. Sufficient conditions that guarantee the optimality of the parallel sampling algorithm were analyzed.

5.1 Synchronous parallel sampling algorithm

The problem formulation can be found in the last chapter. To trade space with time, one may refer to section 4.2 for detail. Basically, in the synchronous parallel sampling algorithm, vehicle swarms make their moves by sampling their local potential function profile in parallel at the same time clock ticks. In particular, using the synchronous parallel sampling algorithm, vehicles update their locations as follows:

- Step 1. Pick a cooling schedule $T(\cdot)$ and the total number N of annealing steps. Let annealing step n=1;

- Step 2. Conduct location updates for node 1 through S in *parallel* at the same time clock, where node $s, 1 \leq s \leq S$, performs following:

- Determine the set L_s of candidate locations for the next move:

$$L_s = \Lambda_s \cap \{(i,j) : \sqrt{(i-i_s)^2 + (j-j_s)^2} \leq R_m\},$$

where Λ_s represents the set of cells not occupied by other vehicles or obstacles;

- When two neighboring vehicles $(s < s')$ have their candidate locations conflicted, i.e., $L_s \cap L_{s'} \neq \emptyset$. The vehicle with lower index updates its candidate locations to $L_s \cap L^c_{s'}$. Repeat this procedure until $L_s \cap L_{s'} = \emptyset$, for all $s \neq s'$.

- Each vehicle s evaluate potential function for every $l \in L_s$,

$$\Phi_s(x_s = l, X(S \backslash s) = x(S \backslash s))$$

where $S \backslash s$ denotes the complement of s in S. And update the location of each vehicle s in parallel by sampling the local distribution

$$p(z) = \frac{\exp(-\frac{\Phi_s(x_s=l,X(S\backslash s)=x(S\backslash s))}{T(n)})}{\sum\limits_{l' \in L_s} \exp(-\frac{\Phi_s(x_s=l',X(S\backslash s)=x(S\backslash s))}{T(n)})}.$$

- Step 3. Let $n = n+1$. If $n = N$, stop; otherwise go to Step 2.

For a fixed temperature T, the underline mathematical model of the synchronous parallel sampling algorithm is a homogenous Markov chain defined by

$$\begin{aligned} & P_T(x(n+1)|x(n)) \\ = & \prod_{s \in S} (p(x_s = x_s(n+1)|x_{\mathscr{N}_s} = x_{\mathscr{N}_s}(n))) \\ = & \prod_{s \in S} \frac{e^{-\frac{\Phi_S(x_S=x_S(n+1),x_{\mathscr{N}_S}=x_{\mathscr{N}_S}(n))}{T(n)}}}{\sum\limits_{l \in L_S(x(n))} e^{-\frac{\Phi_S(x_S=l,x_{\mathscr{N}_S}=x_{\mathscr{N}_S}(n))}{T(n)}}} \end{aligned} \tag{5.1}$$

where $\|x_s(n+1) - x_s(n)\| \leq R_m$ for all $s \in S$. $\Phi_s(x_s = x_s(n+1), x_{\mathscr{N}(s)} = x_{\mathscr{N}(s)}(n))$ is the local energy which could be evaluated by vehicle s with only local information.

Proposition 5.1.1 *For a fixed temperature T, the homogeneous Markov chain (5.1) has a unique invariant distribution Π_T. From any initial distribution ν_0*

$$\lim_{n\to\infty} \nu_0 P_T^n = \Pi_T \tag{5.2}$$

Sketch of Proof. Due to the connectivity of the accessible area, there exists at least one path between any two configurations x and y (i.e., a sequence of multiple moves $\{x, x_1, \cdots, y\}$), and the shortest path is bounded by τ moves for some finite τ. This implies that P_T has a *strictly positive power* P_T^τ, i.e., the τ-step Markov chain reaches each state with positive probability from any state. The irreducibility and aperiodicity of the kernel then follows. Hence the Markov chain is ergodic and has a unique invariant distribution Π_T for a fixed temperature T [81]. □

Picking an appropriate cooling schedule $T(n)$ and τ, the simulated annealing algorithm yields unique distribution Π_∞. This is made precise by the following theorem.

Theorem 5.1.1 *Let $\tilde{U}(x,y) : X \times X \to \mathbb{R}$ be an induced energy function defined on the clique potentials*

$$\tilde{U}(x,y) = \begin{cases} \sum_{s\in S}\sum_{c\ni s} \Psi_c(y_s, x_{S\backslash s}), & when\, y \in \mathcal{N}^m(x); \\ 0, & o.w.. \end{cases} \tag{5.3}$$

where $\mathcal{N}^m(x) = \{z \in X : \forall s, \|z_s - x_s\| \le R_m\}$. Let $\tilde{\Delta}$ be:

$$\tilde{\Delta} \overset{\triangle}{=} \max_{y,z\in\mathcal{N}^m(x)} |\tilde{U}(x,y) - \tilde{U}(x,z)|.$$

Let $T(n)$ be a cooling schedule decreasing to 0 such that eventually,

$$T(n) \ge \frac{\tau\tilde{\Delta}}{\ln n}.$$

Let $Q_n = P^{\tau}_{T(n)}$. Then for any initial distribution ν,

$$\lim_{n\to\infty} \nu Q_1 \cdots Q_n \to \Pi_\infty, \tag{5.4}$$

where Π_∞ is the limit distribution of (5.2) as T tends to zero. In particular,

$$\lim_{T\to 0} \Pi_T(x) = \Pi_\infty(x). \tag{5.5}$$

Proof. Let $\alpha_x = \min_{y\in\mathscr{N}^m(x)} \tilde{U}(x,y)$. From (5.1), we have

$$P_T(x,y) = \frac{\exp(-\frac{\tilde{U}(x,y)-\alpha_x}{T})}{\sum\limits_{z\in\mathscr{N}^m(x)} \exp(-\frac{\tilde{U}(x,z)-\alpha_x}{T})} \leq \frac{e^{-\frac{\tilde{\Delta}}{T}}}{|\mathscr{N}^m(x)|},$$

where $|\mathscr{N}^m(x)|$ denotes the cardinality of the configuration space $\mathscr{N}^m(x)$. Following analogous arguments to those in the proof of Theorem 4.2 in [76] , one can show

$$c(Q_n) \leq 1 - \lambda e^{-\frac{-\tau\tilde{\Delta}}{T(n)}},$$

where $c(Q_n)$ denotes the contraction coefficient of Q_n, and $\lambda = \frac{|\mathscr{X}|}{|\mathscr{N}^m(x)|^\tau}$. Similarly, one can prove the claim (5.4). □

Remark 5.1.1 *For the parallel sampling algorithm, the explicit expression of invariant distribution (5.2) is generally lacking. It is hard to analytically study the equilibrium properties. Here, we just have some brief discussions.*

Let Ω^0 be the set of limiting configuration(s) which is defined by

$$\Omega^0 \triangleq \{x : \Pi_\infty(x) > 0\}. \tag{5.6}$$

Let Ω^L be the set of all the local minima of U. Then we have $\Omega^0 \subset \Omega^L$. If the potential function U is "well behaved", i.e., $\{x^* : U(x^*) = \min_x U(x)\} \subset \Omega^0$, there is a positive *chance that the parallel annealing algorithm leads the final configuration to x^* as temperature tends to zero, which is confirmed by extensive simulations in[77]. In section 5.3, we analytically studied the limiting configurations with a special example.*

5.2 Asynchronous parallel sampling algorithm

The asynchronous parallel sampling algorithm works similar as the synchronous version, except each vehicle s makes moves independently by following its own time clock $t^s = \{t_1^s, t_2^s, ...\}$. Thus, at one time instance n, there are only a part of vehicles make a move. The transition probability from configuration $x(n)$ to $x(n+1)$ can be written down as follows

$$\begin{aligned} & \tilde{P}_T(x(n+1)|x(n)) \\ = & \prod_{s:n\in t^s} (p_T(x_s = x_s(n+1)|x_{\mathcal{N}(s)} = x_{\mathcal{N}(s)}(n))). \end{aligned} \tag{5.7}$$

Clearly this formulation leads to a inhomogeneous Markov chain. In general, a inhomogeneous Markov chain may not have unique stationary distribution. This presents challenges in convergence analysis. To deal with the difficulty, we adopt the partial parallel model in [80] and model the asynchronous parallel algorithm as a hieratical Markov chain.

Let $t = \bigcup_{s\in S} t^s$ denote the set of updating time for all vehicles. Clearly, t is a countable set. For each time instance $t_i \in t$, each vehicle s has a probability p_s to make a move, which is defined by

$$p_s \triangleq \lim_{|t|\to\infty} \frac{|t^s|}{|t|},$$

where $|t^s|$ and $|t|$ denote the cardinality of t^s and t respectively. For synchronous case, $p_s \equiv 1$; whereas, for asynchronous one $0 < p_s < 1$. Then, the associated Markov chain kernel P_T can be expressed as

$$\begin{aligned}
&\tilde{P}_T(x(n+1)|x(n)) \\
&= \prod_{s \in S} ((1-p_s)\mathbf{1}_{x_s(n+1)=x_s(n)} + p_s P_T(x(n+1)|x(n)))
\end{aligned} \tag{5.8}$$

Since the kernel (5.8) defines a homogeneous Markov chain, followed by proposition 5.1.1, the Markov chain has unique stationary distribution $\tilde{\Pi}_T$ for a fixed temperature. Then, by using similar argument in theorem 5.1.1, with an appropriate cooling schedule, the asynchronous parallel annealing algorithm converges to a unique distribution $\tilde{\Pi}_\infty$, where $\tilde{\Pi}_\infty = \lim_{T \to \infty} \tilde{\Pi}_T$.

5.3 Equilibrium Analysis of the synchronous parallel algorithm in an example

In this section, an explicit Π_T is derived for a particular example based on a battle field scenario in section 4.2. Sufficient conditions that guarantee the optimality of the parallel sampling algorithm are studied.

Proposition 5.3.1 *For the synchronous Markov chain kernel of (5.1), suppose that $\tilde{U}(x,y)$ defined in (5.3) has a symmetric form, i.e., $\tilde{U}(x,y) = \tilde{U}(y,x)$ for all $x, y \in X$. For a fixed temperature T, the synchronous Markov chain has a unique stationary distribution Π_T*

given by

$$\Pi_T(x) = \frac{\sum\limits_{z\in\mathcal{N}^m(x)} \exp(-\tilde{U}(x,z)/T)}{\sum\limits_{y\in X}\sum\limits_{z\in\mathcal{N}^m(x)} \exp(-\tilde{U}(y,z)/T)} \tag{5.9}$$

proof. The existence and uniqueness of stationary distribution are followed by proposition 5.1.1. The markov chain kernel (5.1) can be rewritten as

$$P_T(x,y) = \frac{\exp(-\frac{\tilde{U}(x,y)}{T}}{\sum\limits_{z\in\mathcal{N}^m(x)} \exp(-\frac{\tilde{U}(x,z)}{T})} \tag{5.10}$$

(5.9) can then be verified due to the fact the balance equation fulfilled, i.e.,$\Pi_T(x) * P_T(x,y) = \Pi_T(y) * P_T(y,x)\Box$

In general, the symmetry of energy function $\tilde{U}(x,y)$ does not hold. However, in some special cases, one can construct a symmetric energy function for the same parallel Markov chain kernel.

Theorem 5.3.1 *Suppose that the Markov Random Field defined in section 4.2 only consists of singleton and pairwise cliques, and the neighborhood system is time-invariant, then there exist a symmetrization potential function $\hat{U}$ which defines the same parallel Markov chain kernel defined by $\tilde{U}$ in (5.10). In specific,*

$$\hat{U}(x,y) = \tilde{U}(x,y) + \sum_{s\in S} \Psi_{\{s\}}(x) \tag{5.11}$$

Proof. We first show the symmetry of $\hat{U}$.

$$\begin{aligned}
\hat{U}(x,y) &= \tilde{U}(x,y) + \sum_{s\in S}\Psi_{\{s\}}(x_s) \\
&= \sum_{s\in S}\sum_{c\ni s}\Psi_c(y_s, x_{S\backslash s}) + \sum_{s\in S}\Psi_{\{s\}}(x_s) \\
&= \sum_{s\neq t}\Psi_{\{s,t\}}(y_s, x_t) + \sum_{s\in S}\Psi_{\{s\}}(y_s) + \sum_{s\in S}\Psi_{\{s\}}(x_s) \\
&= \sum_{s\neq t}\Psi_{\{s,t\}}(x_s, y_t) + \sum_{s\in S}\Psi_{\{s\}}(x_s) + \sum_{s\in S}\Psi_{\{s\}}(y_s) \\
&= \tilde{U}(y,x) + \sum_{s\in S}\Psi_{\{s\}}(y_s) = \hat{U}(y,x)
\end{aligned}$$

Because the difference between $\hat{U}(x,y)$ and $\tilde{U}(x,y)$ depends only on configuration x, the two potential functions actually defines the same Markov chain kernel. In precise, for any two configurations x and y, we have

$$\begin{aligned}
P_T(x,y) &= \frac{\exp(-\tilde{U}(x,y)/T)}{\sum\limits_{z\in\mathcal{N}^m(x)}\exp(-\tilde{U}(x,z)/T)} \\
&= \frac{\exp(-\tilde{U}(x,y)/T - \sum\limits_{s\in S}\Psi_{\{s\}}(x)/T)}{\sum\limits_{z\in\mathcal{N}^m(x)}\exp(-\tilde{U}(x,z)/T - \sum\limits_{s\in S}\Psi_{\{s\}}(x)/T)} \\
&= \frac{\exp(-\hat{U}(x,y)/T)}{\sum\limits_{z\in\mathcal{N}^m(x)}\exp(-\hat{U}(x,z)/T)} \quad \square
\end{aligned}$$

Let $\tilde{H}(x)$ be the induced energy from the invariant distribution $\Pi_{T(n)}$ of the Markov chain kernel $Q_{T(n)}$. In specific,

$$\tilde{H}(x) = -\ln\left(\sum_{z\in\mathcal{N}^m(x)}\exp(-\hat{U}(x,z))\right) \tag{5.12}$$

Pick an appropriate cooling schedule $T(n)$ as in theorem 5.1.1, one could conclude that the asymptotic configuration(s) Ω^0 of the parallel sampling algorithm are the minimizer of $\tilde{H}(x)$. In the next, one would like to study whether Ω^0 minimize the original configuration energy $U(x)$.

With the Gibbs potential function defined in (4.4), the induced energy function $\hat{U}(x,y)$ satisfies the following inequality

$$\begin{aligned}
\hat{U}(x,y) &= \sum_{s\neq t}\lambda_n\|y_s - x_t\| \\
&\quad + \sum_{s\in S}(\lambda_g(J_s^g(x)+J_s^g(y))+\lambda_o(J_s^o(x)+J_s^o(y))) \\
&\leq \sum_{s\neq t}\lambda_n(\|y_s - x_s\| + \|x_s - x_t\|) \\
&\quad + \sum_{s\in S}(\lambda_g(J_s^g(x)+J_s^g(y))+\lambda_o(J_s^o(x)+J_s^o(y))) \\
&\leq \sum_{s\neq t}\lambda_n R_m + U(x) + U(y) \\
&\leq c_1 R_m + 2U(x) + \Delta
\end{aligned} \tag{5.13}$$

where $\Delta = max_{y\in\mathscr{N}^m(x)}|U(x)-U(y)|$ is the *maximal local oscillation* of the potential U. And $c_1 = \sum_{s\neq t}\lambda_n$

From (5.12) and (5.13), we have

$$\tilde{H}(x) \leq \overline{\mathscr{M}}\,(2U(x)+c_1R_m+\Delta) \tag{5.14}$$

where $\overline{\mathscr{M}} = \max_x \ln|\mathscr{N}^m(x)|$. Let x^* be the minimizer of $U(x)$, i.e., $x^* = \arg\min_{x\in\mathscr{X}} U(x)$. Minimize both side of (5.14), we have

$$\min_{x\in\mathscr{X}}\tilde{H}(x) \leq \overline{\mathscr{M}}\,(2U(x^*)+c_1R_m+\Delta) \tag{5.15}$$

Similarly, using the fact $\|x_s - x_t\| \leq R_m + \|y_s - x_t\|$, it can be shown that

$$\tilde{H}(x) \geq \underline{\mathscr{M}}\,(2U(x)-c_1R_m-\Delta)$$

where $\underline{\mathscr{M}} = \min_x \ln|\mathscr{N}^m(x)|$.

Lemma 5.3.1 *Let the set $A = \{x : \tilde{H}(x) \leq \overline{\mathscr{M}}\,(2U(x^*)+c_1R_m+\Delta)\}$, and the set $B = \{x : U(x)-U(x^*) < \frac{\overline{\mathscr{M}}}{\underline{\mathscr{M}}}(c_1R_m+\Delta)\}$. Then $B \supset A$*

Proof. $\forall x \in \bar{B}$, we have

$$\begin{aligned}\tilde{H}(x) &\geq \underline{\mathscr{M}}(2U(x) - c_1 R_m - \Delta) \\ &> \overline{\mathscr{M}}(2U(x^*) + c_1 R_m + \Delta)\end{aligned}$$

which implies $x \in \bar{A}$. So, $\bar{B} \subset \bar{A}$, which is equivalent to $A \subset B$. □

From the lemma, one could conclude that the minimizer of $\tilde{H}(x)$ lies in a ball Ω^B with radius $c_1R_m + \Delta$ from minimizer of $U(x)$. In section 5.1, we have Ω^0 is a subset of local minima Ω^L. With lemma 5.3.1, we have

$$\Omega^0 \subset (\Omega^L \cap \Omega^B) \tag{5.16}$$

If $(\Omega^L \cap \Omega^B) = \{x^*\}$, the parallel algorithm minimizes the original potential function U and desired configuration(s) can be achieved. For many applications, goal configurations might not restrict to ones with minimum energy. If all configurations contained in $(\Omega^L \cap \Omega^B)$ are desired, parallel algorithm then achieves the global goals for sure.

5.4 Simulation Results

In [77], several examples on vehicle formation control using parallel sampling algorithm, e.g., line formation, clustering. In this section, simulations with the battle field scenario were conducted to verify the analysis result in the previous sections. The simulations suggest that the parallel sampling algorithm successfully lead the vehicle networks to the target area without being trapped by the obstacles.

In the simulation there are 50 vehicles on a 48 × 48 area (see Fig. 3.1). The target is located at the corner (43,43) with radius $R_g = 5$, and two overlapped circular obstacles with radius 5 are centered at (17, 23) and (23, 17), respectively. Initially the vehicles are randomly distributed close to the other corner which is opposite to the target. The parameters used are: $\lambda_g = 5$, $\lambda_o = 1$, $\lambda_n = 0.2$, $R_m = \sqrt{2}$, $R_s = 6\sqrt{2}$. The cooling schedule is set to be $\frac{100}{\log(n)}$, where n is the number of annealing iterations. Simulated annealing was performed for 1000 steps. Figure 5.1 shows the snapshots after different annealing steps. One can see that the whole group reaches target after 750 steps.

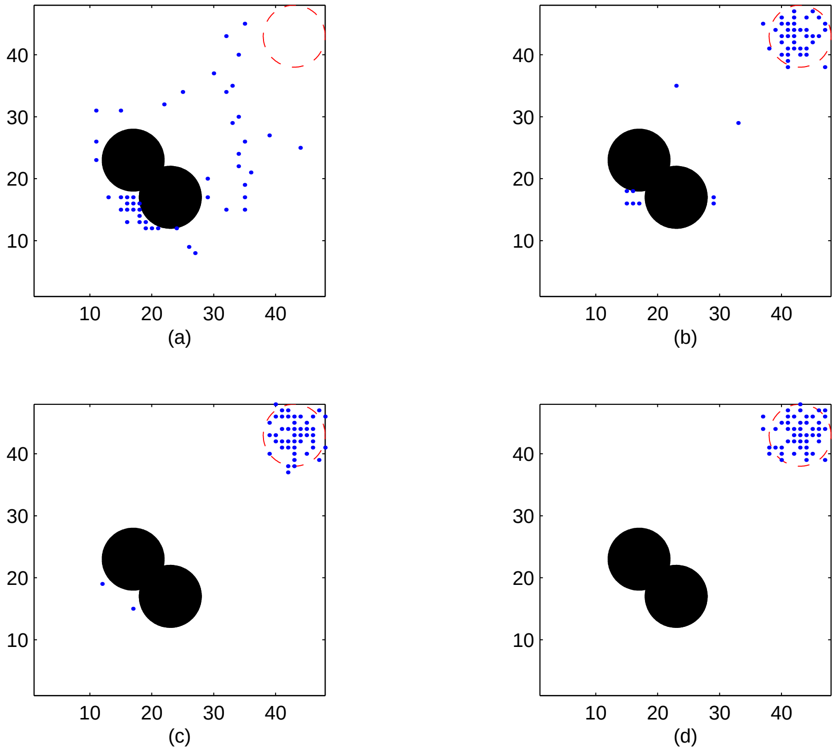

Fig. 5.1: Snapshots of formation operation. (a) initial configuration; (b) after 250 annealing steps; (c) after 500 annealing steps; and (d) after 750 annealing steps.

Chapter 6

Performance improvement through a hybrid scheme

Analysis and simulations in chapter 4 and 5 have shown that simulated annealing with the Gibbs sampler can lead to the achievement of global objectives with limited moving capabilities and communication ranges [77, 76]. However, high traveling cost arising from the stochastic nature of the algorithm presents a barrier to its application in practice.

In this chapter a hybrid approach is proposed, which combines the advantages of the deterministic gradient-flow method and the stochastic simulated annealing method. The algorithm works as follows. Each vehicle in a swarm makes its own decision to switch between the two methods: with initial choice of the gradient-flow scheme, a vehicle switches to simulated annealing when it determines that it gets trapped by some obstacles. After a certain number *N* of annealing steps, it switches back to the gradient-flow scheme to save the traveling time and cost. A notion of memory is introduced to further improve the performance. Each vehicle records the “dangerous” locations where it has been trapped before, and adaptively takes this information into account when making moving decisions.

6.1 A Hybrid control algorithm

In the application of the battle field scenario, the gradient-flow method alone provides fast march toward the target in the absence of obstacles but it may get vehicles trapped in non-target areas. On the other hand, the Gibbs sampler-based simulated annealing complements the gradient method in that it could move vehicles out of otherwise trapping areas, but one has to pay the cost associated with probabilistic exploration - longer execution times and traveling distance. The hybrid control scheme aims to combine advantages of both schemes while avoiding their disadvantages.

To demonstrate the idea, a battle field scenario is used for simulations. The problem setup can be referred to section 4.2. And the mission goal is to maneuver multi-vehicles to cover target area while avoiding obstacles and collide with other vehicles. For simplicity, it is assumed that there is one circular target area centered at p^g, and there are K (possibly overlapping) circular obstacles centered at p^{ok}, $1 \leq k \leq K$. To achieve such objective, the following Clique potential functions in (4.2) are used.

$$\begin{aligned} \Psi_s^g &= \|p_s - p^g\| \\ \Psi_s^o &= \sum_{k=1}^{K} \frac{1}{\|p_s - p^{ok}\|} \\ \Psi_s^n &= \begin{cases} \frac{1}{\sum\limits_{z \in \mathcal{N}_s} \frac{1}{\|p_s - p_z\|}}, & if \mathcal{N}_s \neq \emptyset \\ \Delta, & if \mathcal{N}_s = \emptyset \end{cases} \end{aligned} \tag{6.1}$$

where Ψ_s^n tends to be smaller when site s has more neighbors. $\Delta > 0$ is a relative large constant and it represents the penalty for having no neighbors.

In the proposed scheme vehicles make moving decisions simultaneously and hence the scheme is fully parallel and scalable. The algorithm works as follows:

- Step 1. Each vehicle starts with the gradient-flow method (see below for more detail on the implementation of the gradient-flow method) and goes to Step 2;

- Step 2. If for d consecutive time instants a vehicle cannot move under the gradient method and its location is not within the target area, then it is considered to be trapped. The vehicle then switches to the simulated annealing method with a predetermined cooling schedule (see below for more detail) and goes to Step 3;

- Step 3. After performing simulated annealing for N time instants, the vehicle switches to the gradient method and goes to Step 2.

In the case of a conflict (multiple nodes contend for one spot), a uniform sampling is performed, and the winner will take the spot while the other contenders will stay put for the current time instant. Note that the resolution of conflict can be achieved *locally* since $R_s \geq 2R_m$ and potentially contending nodes are within the local communication range. In the simulation the algorithm will stop if

$$u_g = \sum_{s \in S} \|p_s - p^g\|^2 \leq \varepsilon, \tag{6.2}$$

where u_g is an indicator measuring how far the vehicles, as a whole, are away from the target area.

Implementation of the gradient-flow scheme and the simulated annealing scheme is provided next for completeness.

1). The gradient-flow scheme

In the gradient-flow method the velocities of vehicles follow the (negative) gradient flows of their potential surfaces. To be specific, at each time instant,

- Step 1. A vehicle s determine the set Λ_s of candidate locations for the next move, i.e., the set of cells with the distance R_m and not occupied by other vehicles or obstacles;
- Step 2. For each $l \in \Lambda_s$, evaluate potential function $\Phi_s(X(S\backslash s) = x(S\backslash s), x_s = l)$, where $S\backslash s$ denotes the complement of s in S;
- Step 3. Update the location of vehicle s by taking

$$x_s = \arg\min_{l \in \Lambda_s} \Phi_s(X(S\backslash s) = x(S\backslash s), x_s = l).$$

2). Gibbs sampler-based simulated annealing

Unlike the gradient-flow scheme, in simulated annealing each vehicle updates its next location by sampling a probability distribution. First a cooling schedule $T(n)$ is determined (how to choose a cooling schedule for best convergence performance is itself a vast subject and is beyond the scope of this paper).

- Step 1. Let $n = 1$;
- Step 2. The vehicle s determines the set Λ_s of candidate locations for the next move as in the gradient-flow method;
- Step 3. For each $l \in \Lambda_s$, the vehicle evaluates the Gibbs potential function $\Phi_s(l) = \Phi_s(X(S\backslash s) = x(S\backslash s), x_s = l)$, and calculates the probability distribution

$$P(x_s = l) = \frac{e^{-\frac{\Phi_s(l)}{T(n)}}}{\sum\limits_{z \in \Lambda_s} e^{-\frac{\Phi_s(z)}{T(n)}}};$$

- Step 4. Update x_s to $l \in \Lambda_s$ with probability $P(x_s = l)$;

- Step 5. Let $n = n + 1$, and repeat Step 2 to step 5.

6.2 Qualitative Analysis of switching parameters

In the hybrid algorithm there are two key parameters that determine the performance of the system: d and N. d is the waiting time that triggers a vehicle to switch from the deterministic approach to the stochastic one, and N is used to determine the duration for stochastic exploration.

6.2.1 Waiting time d

When d is smaller, it's more likely for a vehicle to make a wrong decision and switch to simulated annealing. In particular, a vehicle may be "trapped" temporarily due to the presence of its neighbors. Premature switching to stochastic exploration adds to the traveling cost. On the other hand, if d is too large, it will also be a waste of time if indeed the current cell is a trapping spot.

This tradeoff is verified through simulation. In the simulation there were 20 vehicles on a 48 × 48 grid (see Fig. 4.2). The target is located at the corner (43,43) with radius $R_g = 5$, and two overlapped circular obstacles with radius 5 are centered at (17, 23) and (23, 17), respectively. Initially the vehicles are randomly distributed close to the other corner which is opposite to the target. The parameters used are: $\lambda_g = 5$, $\lambda_o = 1$, $\lambda_n = 0.2$, $R_m = \sqrt{2}$, $R_s = 6\sqrt{2}$. The cooling schedule is set to be $\frac{100}{\log(n)}$, where n is the number of annealing iterations. The stopping criterion $\varepsilon = 200$ (for the distance indicator u_g) is

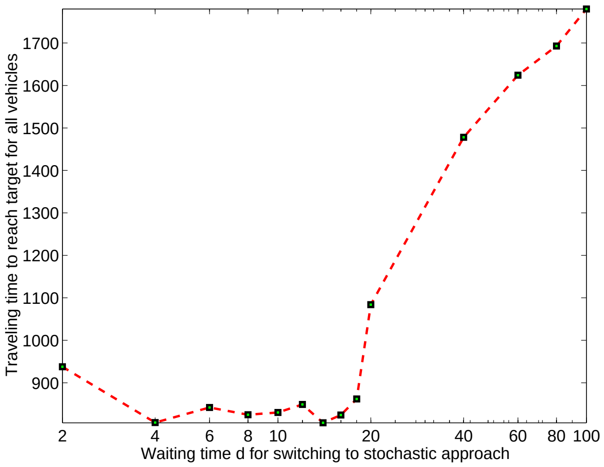

Fig. 6.1: Average traveling time versus the switching parameter d (waiting time).

chosen. So at the end of each simulation, the average distance between the target and vehicles is about $\sqrt{10}$ which is less than the target radius R_g. While fixing the duration N to be 100, the switch waiting time d is increased from 2 to 100. For each d, 10 simulation runs were performed and the traveling times were averaged. Fig. 6.1 shows the average traveling time versus the switch waiting time d.

In the figure, as d is very small ($d = 2$), vehicles take about 950 steps to arrive the target. Then the traveling time drops to 800 when d is between 4 and 18. After that, as d becomes larger and is comparable to stochastic exploration duration N, the performance is dramatically degraded. Clearly, a moderate d should be chosen for the best efficiency.

6.2.2 Duration N

Duration time N for stochastic perturbation is another key parameter. Intuitively a very small N may not provide a trapped vehicle enough opportunities to get out; and a very big N will kill the time-saving advantage offered by the gradient-flow algorithm.

Therefore, it is of interest to study how the duration time N affects performance analytically.

A simplifying assumption is adopted to make the analysis tractable. Considering that each vehicle makes its own moving decision, one might approximate the multi-vehicle system as a collection of independent single vehicles. Furthermore, it is assumed that the time spent on the gradient-flow method is much shorter than the time spent on the stochastic approach, and can be neglected. To justify the latter assumption, it has been found in simulation that a vehicle takes 40-50 times more steps to get the target using the stochastic approach than using the gradient-flow approach in the absence of obstacles. Define the *reachable* area to be the set of cells from which a vehicle can reach the target area under the gradient-flow method, and the *unreachable* area to be its complement. Starting from the unreachable area with the gradient method, a vehicle will be trapped and switch to the simulated annealing. For the duration N of stochastic exploration, let $P(N)$ be the probability that a vehicle will move to the reachable area after N steps of simulated annealing with the Gibbs sampler. Then the expected time for the vehicle to reach the target is approximately

$$\begin{aligned} T_{total} &= \sum_{k=1}^{\infty} k \cdot N \cdot P(N)(1-P(N))^{k-1} \\ &= N \cdot P(N)^{-1}, \end{aligned} \tag{6.3}$$

where the vehicle is assumed to start from the unreachable area, otherwise $T_{total} = 0$.

The key question then becomes how to evaluate $P(N)$. In section **??** a bound (4.15) is derived to reflect the convergence rate of the probability distribution $\Pi_n = \nu Q_1 \cdot Q_n$. From section **??** we know, by designing the potential function such that the target location

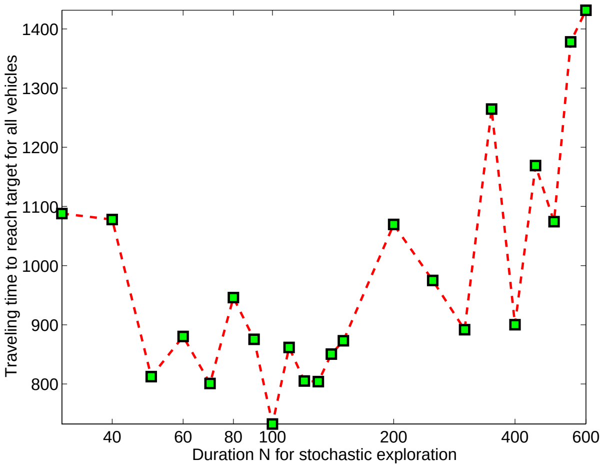

Fig. 6.2: Average traveling time versus the duration N for stochastic exploration.

has the lowest potential, Π_∞ has mass 1 in the target area. Since the target belongs to the reachable area, the probability $P(N)$ satisfies the following inequality:

$$\begin{aligned} P(N) &\geq P(\text{vehicle reaches target}) \\ &\geq 1-\frac{1}{2}const\cdot N^{-\frac{2\lambda\tilde{m}}{2\tilde{m}+\lambda\Delta\tau}}. \end{aligned} \tag{6.4}$$

Combining (6.3) and (6.4), one obtains

$$T_{total} \leq \frac{N}{1-\frac{1}{2}const\cdot N^{-\frac{2\lambda\tilde{m}}{2\tilde{m}+\lambda\Delta\tau}}}. \tag{6.5}$$

Eq. (6.5) clearly indicates that an optimal N exists to minimize the bound on $T(N)$. This analysis has confirmed by simulation. Same simulation scenario and parameters were used earlier except that the waiting time d for switching was 8 and the duration N for stochastic exploration was varies from 20 to 600. Fig. 6.2 shows the average traveling time versus N, and one can see that a choice of N between 60 and 200 would achieve good time-efficiency.

Remark 6.2.1 *The hybrid control scheme proposed is essentially a kind of stochastic* *relaxation algorithm. However, simulated annealing based on the Gibbs sampler provides advantages over many other stochastic perturbation methods.*

Take the single vehicle case as one example. One can show that the probability of a vehicle getting trapped again after N annealing steps goes to zero as N goes to infinity. This is not the case with just any stochastic scheme. Take a random walk-type perturbation scheme as an example. As the number N of random walks goes to infinity, the distribution of the vehicle approaches the stationary distribution of the Markov chain defined by the random walk. In general there will be positive masses falling in the unreachable area. This implies that there is no guarantee for the vehicle to reach the target no matter how many steps are used for the random perturbation.

6.3 The impact of memory

In this section the notion of memory is introduced to further enhance the hybrid control scheme. The idea is to record the trapping spots and reduce the probability of repeatedly being trapped at the same spots. Each vehicle keeps track of the *risk level* of dangerous cells, and accordingly modify its potential surface to lower the probability of accessing high-risk regions. To be specific, the enhanced algorithm works as follows:

- Step 1. Initialize the algorithm, set parameter d, N, and the cooling schedule $T(n)$, and let all vehicles initially choose the gradient-flow method;

- Step 2. When a vehicle s determines that it has been trapped at cell y by obstacles, it increases the risk level R^s_y by 1 (the default risk level for every location is 1). Then the vehicle switches to simulated annealing with $n = 1$;

- Step 3. At each annealing step, vehicle s determines the set L_s of candidate location for its next move. For $l \in L_s$, it evaluates the Gibbs potential function $\Phi_s(X(S\backslash s) = x(S\backslash s), x_s = l)$, which is simply denoted as $\Phi_s(l)$. Then vehicle s will take l with the probability
$$P(x_s = l) = \frac{e^{-\frac{\Phi^s(l)}{T(n)}}/R_l^s}{\sum_{z\in L_s} e^{-\frac{\Phi^s(z)}{T(n)}}/R_z^s};$$
- Step 4. Increase n by 1 and repeat Step 3 until n reaches N. The vehicle s then switches back to the gradient-flow algorithm and goes to Step 2;
- Step 5. The algorithm stops if the aggregate distance indicate $u_g \leq \varepsilon$.

6.4 Simulation Results

To compare the performance with the original hybrid control scheme, simulation has been performed with the same setup as in the previous section. The waiting time was set to be $d = 6$ and the duration N varied from 30 to 600. As seen in Fig. 6.3, the hybrid control scheme with memory always achieves better performance than the original one for all N.

Summary

In this chapter, a hybrid algorithm was developed in an attempt to improve the efficiency of the Gibbs sampling based stochastic algorihtm. Some analysis was provided to gain insight into the algorithm. The algorithm is heuristic in nature and treats each vehicle independently, and thus one cannot expect to get analytical convergence results for the network as a whole. In some sense the hybrid control scheme offers stochastic pertur-

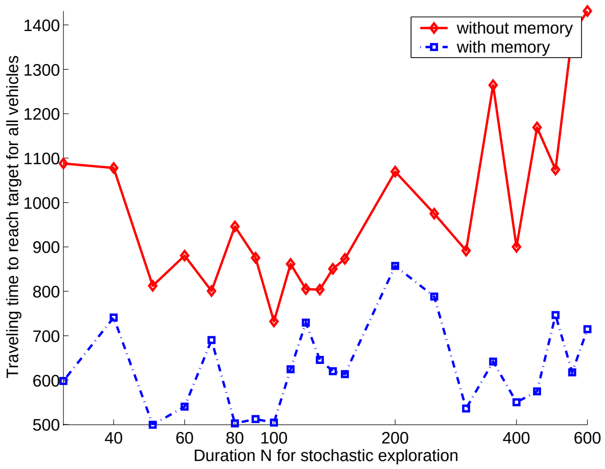

Fig. 6.3: Averaged traveling time under the hybrid scheme with memory, in comparison with that under the memoryless scheme.

bation to the deterministic gradient-flow method; however, it has advantages over random walk-type perturbation schemes [15]. Take the single-vehicle case as an example. One can show that the probability of a vehicle getting trapped again after N annealing steps goes to zero as N goes to infinity. For a random-walk perturbation scheme, however, as the number N of random walks goes to infinity, the configuration distribution approaches the stationary distribution of the Markov chain defined by the random walk, and there is no guarantee for the vehicle to reach the target no matter how many steps are used.

Chapter 7

Robustness of the Gibbs sampling approach in the presence of sensor errors

The Gibbs sampler based stochastic algorithm proposed in the previous chapters have been demonstrated great success in controlling UAVs networks. However, an underline assumption in our previous studies was that the potential function can be precisely evaluated. In practice, the potential values have to be calculated via sensor measurements. In many applications, e.g., the battle field scenario, cost-effective sensors are preferred to reduce the total expense. As a result, sensor uncertainties introduce noises to Gibbs potential evaluations. It is then of interests to study the robustness of the annealing algorithm. In the past, this issue have been studied for the annealing algorithm based on the classical MRF. In [82], Grover presented an analysis of the impact of fixed *range-error* on equilibrium properties. Later on, Gelfand and Mitter studied the effects of state-independent Gaussian noise. They showed that in certain conditions, slowly decreasing *random-error* will not affect the limiting configurations [83, 84]. Greening studied the impact of errors for the metropolis annealing algorithm in VLSI applications[85]. By assuming the errors as the AWGN, a explicit expression of equilibrium distribution was derived for the underline Markov chain. Convergence rate was also investigated in his dissertation. In this chapter, we studied the impact of both fixed *range-error* and bounded *random-error* on the annealing algorithm proposed in [78]. In our analysis, unlike previous studies, we

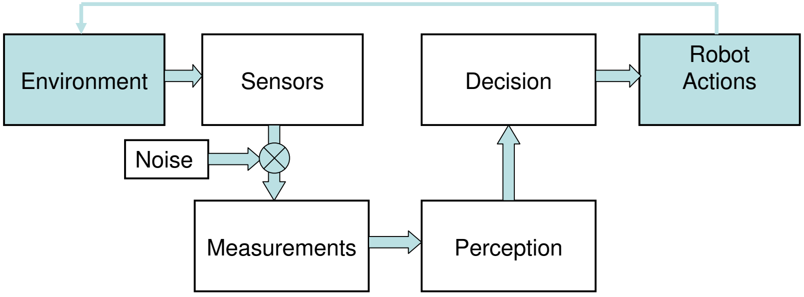

Fig. 7.1: The diagram of the general robotics system

do not require the random-error follows Gaussian distribution. Sufficient conditions that guarantee the convergence to the global minimizer were analyzed. Simulations are further confirmed the analysis results.

7.1 Sensors and Sensor Errors

In robotics applications, sensors play important roles in understanding/exploring environment and handling the environment uncertainty. The diagram of the general robotics system is depictedx in figure 7.1,

In the diagram, sensors are firstly used to capture the measurements from surrounding environment, e.g., the UAV's location, the distance between the UAV and obstacles, the environment photos taken by cameras, and etc. The measurements are then fed into certain data processing module to acquire necessary information for decision making. This module may consist of but not limited to noise filtration, image processing and understanding, data fusion, and etc. The decision making module uses the processed data to compute the control command for achieving certain tasks and send back to robotics actuators. For example, both the gradient descent approach, and the gibbs sampler based approach belongs to this category.

There are different types of sensors have been developed in the robotics field in the past few decades. From their functions, sensors can be divided into two categories: internal sensors and external sensors. Internal sensors are mainly used to measure and monitor the internal states of a robot, e.g. velocity, acceleration, temperature, pressure, balance, attitude, and etc., so that low level controller can be designed for UAV maneuvering. It also help to detect and avoid failure situations, e.g., actuator saturation, unstable mode. Most internal sensors are non-contact. They include but not limited to synchros, resolves, proximity, accelerometers, compasses, optic encoders. Both synchros and resolves are used to measure angular position. compasses generate global coordinates. Optic encoders are used for measuring and controlling motor speed and acceleration.

External sensors, on the other hand, are responsible for acquire information from environments as aforementioned. For example, GPS is usually used to get vehicle's position; camera can be used to get the shape of obstacles; sonar can be used to measure the distance between vehicle and obstacles. There are even odor sensors have been recently developed to detect the chemical ingredients and densities.

As discussed before, in multiple UAVs applications, failure or even loss of certain amount of UAVs is tolerable because of the dangerous working conditions. Therefore, cost-effective sensors are preferred to reduce the total cost of a single UAV. As a result, sensor noises and uncertainties may introduce fairly large errors into the system. It is then of interest to study the robustness of the Gibbs sample based algorithm.

The sensor error considered in the thesis fall into two categories: *range-error* and *random-error*. A potential function is said to has *range-errors* if the difference between the nominal potential value $U(x)$ and the observed one $\hat{U}(x)$ is confined to a fixed range, and does not change with time. The *range-error* is usually caused by the system error of defected sensors. On the contrary, we consider that the potential function $\tilde{U}(x)$ has *random-errors* if the difference between the $U(x)$ and $\tilde{U}(x)$ is an independent random variable, which is denoted as $Z(x)$. The *random-error* introduces time-varying potential evaluation.

7.2 Convergence analysis in the presence of sensor errors

In this section, we study the impact of sensor errors on the convergence properties of the annealing algorithm in the subsection 4.4.

7.2.1 Gibbs potential with range-error

When sensors carried by vehicles have *range-error*, the observed potential $\hat{U}(x)$ of a configuration x can be expressed as

$$\hat{U}(x) = U(x) + e(x), \tag{7.1}$$

where $e(x)$ is a finite constant. We assume $\underline{e} \leq e(x) \leq \overline{e}$, where $\underline{e}$ and $\overline{e}$ are the upper bound and lower bound of range error respectively. The observed potential $\hat{U}(x)$ satisfies

$$U(x) + \underline{e} \leq \hat{U}(x) \leq U(x) + \overline{e}, \tag{7.2}$$

Since the *range-error* is time-invariant, the Gibbs sampler defines a homogeneous Markov chain at a fixed temperature T. By directly applying theorem 4.4.1, one could conclude that there exist a unique equilibrium distribution $\hat{\pi}_T$ at temperature T,

$$\hat{\Pi}_T(x) = \frac{e^{-\frac{\hat{U}(x)}{T}} \sum_{s \in S} \sum_{z \in \mathscr{N}_m^x(s)} e^{-\frac{\hat{U}(z)}{T}}}{Z_T}. \tag{7.3}$$

Proposition 7.2.1 *Let $\Pi_T(x)$ denote the equilibrium distribution in (5.2). Let the maximum oscillation of range error $\Delta_e = \overline{e} - \underline{e}$. Then,*

$$e^{-\frac{2\Delta_e}{T}} \Pi_T(x) \leq \hat{\Pi}_T(x) \leq e^{\frac{2\Delta_e}{T}} \Pi_T(x) \tag{7.4}$$

Moreover,

$$\|\hat{\Pi}_T - \Pi_T\| \leq e^{\frac{2\Delta_e}{T}} - 1, \tag{7.5}$$

where $\|\cdot\|$ stands for L_1 norm in this paper.

Proof. Pick any configuration $x \in \mathscr{X}$. For each configuration $y \in \{x \cup \mathscr{N}^m(x)\}$, let $U(y) = U(y) + \underline{e}$. For any other configurations ($x' \in \{x \cup \mathscr{N}^m(x)\}^c$), let $\hat{U}(x') = U(x') + \overline{e}$. Then, we have

$$\hat{\Pi}_T(x) \leq \frac{e^{-\frac{U(x)+\underline{e}}{T}} \sum_{s \in S} \sum_{y \in \mathscr{N}^m(x)} e^{-\frac{U(y)+\underline{e}}{T}}}{Z_T(\hat{U})},$$

where $Z_T(\hat{U})$ denotes the partition function by taking Gibbs potential $\hat{U}(x)$. Clearly,

$$Z_T(\hat{U}) > Z_T(U + \overline{e}).$$

Then,

$$\begin{aligned}
\hat{\Pi}_T(x) &\leq \frac{e^{-\frac{2\underline{e}}{T}} e^{-\frac{U(x)}{T}} \sum_{s \in S} \sum_{y \in \mathscr{N}^m(x)} e^{-\frac{U(y)}{T}}}{Z_T(U + \overline{e})} \\
&= \frac{e^{-\frac{2\underline{e}}{T}} e^{-\frac{U(x)}{T}} \sum_{s \in S} \sum_{y \in \mathscr{N}^m(x)} e^{-\frac{U(y)}{T}}}{e^{-\frac{2\overline{e}}{T}} Z_T(U)} \\
&= e^{\frac{2(\overline{e} - \underline{e})}{T}} \pi_T(x) = e^{\frac{2\Delta_e}{T}} \pi_T(x).
\end{aligned}$$

The converse arguments supplies the lower bound. Then, By inequality (7.4)

$$\begin{aligned}\|\hat{\Pi}_T - \Pi_T\| &\leq \max\{\|e^{\frac{2\Delta_e}{T}}\Pi_T - \Pi_T\|\,,\|e^{-\frac{2\Delta_e}{T}}\Pi_T - \Pi_T\|\} \\ &= e^{\frac{2\Delta_e}{T}} - 1.\end{aligned}$$

The last equality holds because $e^{\frac{2\Delta_e}{T}} - 1 > 1 - e^{-\frac{2\Delta_e}{T}}$. □

Proposition 7.2.1 unveils the basic impact of range-error on the equilibrium distribution for a fixed temperature T. Moreover, pick an appropriate cooling schedule as in theorem 5.1.1, i.e., logarithm cooling rate, it can be shown that the SA algorithm leads to limiting configurations with minimum energy of $\hat{U}(x)$. If the global minimizer of $\hat{U}(x)$ minimizes the nominal Gibbs potential $U(x)$, the range-error does not affect limiting configurations. A sufficient condition is formally stated in the following proposition.

Proposition 7.2.2 *For the Gibbs potential with range-error, the simulated annealing algorithm leads to the global minimizer x^* of the nominal Gibbs potential $U(x)$, if the following condition satisfies:*

$$\Delta_e \leq \frac{1}{2}\Delta_U, \tag{7.6}$$

where Δ_U is the minimum potential difference with global minimizer, i.e.,

$$\Delta_U = \min_{x\in\mathscr{X},x\neq x^*} |U(x) - U(x^*)|.$$

Proof. Let configuration x be any configuration other than the global minimizer x^*, i.e.,$x \neq x^* \in \mathscr{X}$. By equation (7.2), we have

$$\begin{aligned}\hat{U}(x^*) &\leq U(x^*) + \Delta_e \leq U(x) - \Delta_U + \Delta_e \\ &\leq U(x) - \Delta_e \leq \hat{U}(x).\end{aligned}$$

One then concludes that x^* minimizes potential function $\hat{U}(x)$ □

If the maximum oscillation of range-error is too large, the simulated annealing algorithm may not be able to lead the limiting configurations to global minimizer.

7.2.2 Gibbs potential with random-error

In the previous section, the potential error $e(x)$ is assumed to be a fixed value for each configuration x. In practice, the potential error due to sensor noise usually varies with time, i.e., the Gibbs potential has *random-errors*. For the ease of analysis, let the *random-error* Z_x be an independent random variable associated with each configuration x. The observed Gibbs potential $\tilde{U}$ with random-error can then be expressed as

$$\tilde{U}(x, z_x) = U(x) + Z_x, \tag{7.7}$$

where Z_x follows a probability distribution f_{z_x}

Proposition 7.2.3 *Let $Z = \{Z_x : x \in \mathscr{X}\}$ be the vector of random-error. The Gibbs sampler with random-error define a homogenous Markov chain at a fixed temperature with kernel matrix satisfies*

$$\tilde{P}_T = E_Z(P_T(z)). \tag{7.8}$$

where $P_T(z)$ is the kernel matrix with fixed range-error z. Moreover, there exist a unique equilibrium distribution $\tilde{\Pi}_T$ at a fixed temperature T. Starting from any initial distribution ν_T^0,

$$\lim_{n\to\infty} \|\nu_T^0(\tilde{P}_T)^n - \tilde{\Pi}_T\| = 0 \tag{7.9}$$

Proof. For any two configurations $x,y \in \mathscr{X}$, the transition probability $\tilde{p}(x,y)$ satisfies

$$\tilde{p}_T(x,y) \quad = \quad \int p_T(x,y|Z=z)f(z)dz. \tag{7.10}$$

One could then conclude (7.8) holds. Given any fixed range-error z, we know the kernel matrix $P_T(z)$ is primitive, i.e., the markov chain is irreducible and aperiodic. Since $\tilde{P}_T$ is a superposition of $P_T(z)$, the primitivity of $\tilde{P}_T$ is obvious. The uniqueness and existence of the equilibrium distribution then follow accordingly. The final statement is followed by the ergodicity of the primitive Markov chain. □

Unfortunately, the lacking of the explicit form of the stationary distribution for the Markov chain $\tilde{P}_T$ presents challenges to analyze the robustness of the SA algorithm under the *random-error*. To simplify the analysis, we assume that the *random-error* has only limited support. Similar idea for analyzing *range-error* in the previous subsection can then be applied.

Proposition 7.2.4 *Assume that the random-error z is bounded, i.e.,* $\underline{z} \leq z_x \leq \bar{z}$, $\forall x$. *Let* $\Delta_z = \bar{z} - \underline{z}$. *Let* $C(P_T)$ *be the contraction coefficient of a Markov kernel* P_T *(see [20]). The equilibrium distribution* $\tilde{\Pi}_T$ *satisfies the following inequality:*

$$||\tilde{\Pi}_T - \Pi_T|| \leq \frac{e^{\frac{2\Delta_z}{T}} - 1}{1 - C(\tilde{P}_T)} \tag{7.11}$$

Proof. By assumption, since Z is bounded, given any $z \in Z$, it is easy to show that, for all $x,y \in \mathscr{X}$, the matrix $P_T(Z=z)$ satisfies

$$e^{-\frac{2\Delta_z}{T}} P_T(x,y) \leq P_T(x,y|Z=z) \leq e^{\frac{2\Delta_z}{T}} P_T(x,y), \tag{7.12}$$

where $P_T = P_T(Z=0)$ is the Markov chain kernel matrix with nominal Gibbs potential. Then

$$\begin{aligned} \|\tilde{\Pi}_T - \Pi_T\| &= \|\tilde{\Pi}_T \tilde{P}_T - \Pi_T \tilde{P}_T + \Pi_T \tilde{P}_T - \Pi_T P_T| \\ &\leq \|\tilde{\Pi}_T - \Pi_T\| C(\tilde{P}_T) + \|\Pi_T \tilde{P}_T - \Pi_T P_T| \\ &\leq \|\tilde{\Pi}_T - \Pi_T\| C(\tilde{P}_T) + (e^{\frac{2\Delta_z}{T}} - 1). \end{aligned}$$

This is equivalent to

$$\|\tilde{\Pi}_T - \Pi_T\|(1 - C(\tilde{P}_T)) \leq e^{\frac{2\Delta_z}{T}} - 1.$$

The inequality (7.11) then follows. □

Clearly, as the maximum oscillation of random-error Δ_z tends to zero, the distribution $\nu_T^n = \nu_T^0 (\tilde{P}_T)^n$ tends to approach the nominal equilibrium distribution Π_T.

Proposition 7.2.5 *Pick an appropriate cooling schedule $T(n)$ such that* $\lim_{n\to\infty} T(n) = 0$ *and the Markov chain $\tilde{P}_T$ converges as temperature tends to zero. Assume $\Delta_z \leq \frac{1}{2}\Delta_U$. Then, From any initial distribution ν*

$$\lim_{n\to\infty} \nu \prod_{i=1}^{n} \tilde{P}_{T(i)} = \Pi_\infty, \tag{7.13}$$

i.e., the limiting configurations tends to global minimizers of nominal potential $U(x)$

Proof. Since the Markov chain kernel matrix $\{\tilde{P}_{T(i)}\}$ is primitive, by picking an appropriate logarithm cooling schedule ($e.g., T(n) = c/log(n)$), the simulated annealing algorithm converges to a limiting distribution $\tilde{\Pi}_\infty$, i.e., $\lim_{n\to\infty} \nu \prod_{i=1}^{n} \tilde{P}_{T(i)} = \tilde{\Pi}_\infty$, where $\tilde{\Pi}_\infty = \lim_{T\to 0} \tilde{\Pi}_T$. In the next, one will show the limiting distribution $\tilde{\Pi}_\infty$ actually equals Π_∞.

Let $\Pi_T(z)$ denote the equilibrium distribution of the Markov chain kernel $P_T(z)$. For any $w \in Z$, one has

$$\begin{aligned}
&\Pi_T(w)\tilde{P}_T = \Pi_T(w)\int_z P_T(z)f(z)dz \\
&= \int_z (\Pi_T(w) - \Pi_T(z))P_T(z)f(z)dz + \int_z \Pi_T(z)f(z)dz.
\end{aligned}$$

Let $\bar{\Pi}$ be the mean of $\Pi(z)$ with respect to the probability distribution $f(z)$. In specific, $\bar{\Pi} = \int_z \Pi_T(z)f(z)dz$. Let $\Delta_{\Pi_T} = \int_w \int_z (\Pi_T(w) - \Pi_T(z))(P_T(z) - P_T(w))f(z)dz$. Integrating the both sides of (7.14) with respect to w, one then has

$$\begin{aligned}
& \bar{\Pi}_T\tilde{P} = \int_w \Pi_T(w)\tilde{P}_T dw \\
= \;& \int_w f(w)\int_z P_T(z)(\Pi_T(w) - \Pi_T(z))f(z)dzdw \\
+ \;& \int_w f(w)\int_z \Pi_T(z)f(z)dzdw \\
= \;& \frac{1}{2}\{\int_w\int_z (\Pi_T(w) - \Pi_T(z))P_T(z)f(z)f(w)dzdw \\
- \;& \int_w\int_z (\Pi_T(w) - \Pi_T(z))P_T(w)f(z)f(w)dzdw\} \\
+ \;& \int_w f(w)\bar{\Pi}_T dw \\
= \;& \frac{1}{2}\Delta_{\Pi_T} + \bar{\Pi}_T \qquad (7.14)
\end{aligned}$$

The primitivity of $\tilde{P}$ implies that $\lim_{n\to\infty}\bar{\Pi}_T\tilde{P}^n = \tilde{\Pi}_T$. Assuming $(I - \tilde{P}_T)^{-1}$ exist, with (7.14), the left hand side of the above equation can be rewritten as

$$\begin{aligned}
\bar{\Pi}_T\tilde{P}^n &= \sum_{i=1}^{n}\frac{1}{2}\Delta_{\Pi_T}\tilde{P}_T^{i-1} + \bar{\Pi}_T \\
&= \frac{1}{2}\Delta_{\Pi_T}(I - \tilde{P}_T^n)(I - \tilde{P}_T)^{-1} + \bar{\Pi}_T
\end{aligned}$$

As n tends to ∞, the equilibrium distribution can then be explicitly expressed as

$$\tilde{\Pi}_T = \frac{1}{2}\Delta_{\Pi_T}(I - \tilde{P}_T^{\infty})(I - \tilde{P}_T)^{-1} + \bar{\Pi}_T \qquad (7.15)$$

By proposition 7.2.2, one has $\lim_{T\to 0}\Pi_T(z) = \Pi_\infty\ \forall z$, since $\Delta_z \leq \frac{1}{2}\Delta_U$. Then, we have

$$\lim_{T\to 0}\Delta_{\Pi_T} = 0,\ \textit{and}\ \lim_{T\to 0}\bar{\Pi}_T = \Pi_\infty.$$

Take the limit of T for equation (7.15), and plug in the above equations. The final conclusion then follows:

$$\begin{aligned}\tilde{\Pi}_\infty &= \lim_{T\to 0}\left(\frac{1}{2}\Delta_{\Pi_T}(I-\tilde{P}_T^\infty)(I-\tilde{P}_T)^{-1}+\bar{\Pi}_T\right)\\ &= \Pi_\infty.\square\end{aligned}$$

The result shows that if the bound of the *random-error* is constrained by $\frac{\Delta_U}{2}$, an appropriated cooling schedule leads to global minimizers.

7.3 Simulation Results

Simulation was conducted to verify the robustness analysis in the previous section. A formation control example involving inter-vehicle interactions are used to demonstrate the impact of the sensor error on the convergence of the Gibbs sampler based approach. Other objectives or constraints, such as target-approaching and obstacle avoidance, can be similarly analyzed.

The goal of the simulation is to have the nodes to form (square) lattice structures with a desired inter-vehicle distance R_{des}. The potential function used was:

$$U(x) = \sum_{r\neq s,\,\|x_r-x_s\|\leq R_i} c_1(|\|x_r-x_s\|-R_{des}|^\alpha - c_2),$$

where $c_1 > 0$, $c_2 > 0$, and $\alpha > 0$. A proper choice of c_2 encourages nodes to have more neighbors. The power α shapes the potential function. In particular, for $|\|x_r-x_s\|-$

$R_{des}| < 1$, smaller α leads to larger potential difference from the global minimum.

In the simulation, 9 nodes were initially randomly placed on an 8 by 8 grid (see Fig. 7.2 (a)). Parameters used were: $R_i = 4\sqrt{2} - \varepsilon$, $R_m = 2\sqrt{2} + \varepsilon$, $R_{des} = 2$, $c_1 = 10$, $c_2 = 1.05$, $\alpha = 0.02$, $T(n) = \frac{1}{0.01 \ln n}$, and $\tau = 20$. The desired configuration (global minimizer of U) is shown in Fig. 7.2 (b) (modulo vehicle permutation and formation translation on the grid). Simulated annealing was performed for 10^4 steps.

The sensor error was modeled as additive noise Z_x as in (7.7). Uniform distribution was selected for Z_x. Other distributions can be studied accordingly. The potential difference of the example was calculated to be $\Delta_U = 11$. So the potential error bound Δ_z should less than 5.5 in order to guarantee the convergence. In the simulation, we compared with 3 different cases: noise-free, $\Delta_z = 5$, and $\Delta_z = 30$. Moreover, for comparison, we studied cases where the sensor error is modeled as additive white gaussian noise (AWGN). Due to the lack of analytical results, numerical studies were provided instead. Two different variances, $\sigma = 1$ and 5, are used in the simulation respectively.

To demonstrate the trend of convergence to the lowest potential, one can calculate the error $||\nu_n - \Pi_\infty||_1$ as metric, where ν_n is the empirical distribution of configurations (again modulo vehicle permutation and network translation), and

$$\Pi_\infty(x) = \begin{cases} 1 \text{ if } x \text{ is desired} \\ 0 \text{ otherwise} \end{cases} .$$

Therefore,

$$||\nu_n - \Pi_\infty||_1 = 1 - \nu_n(x^*) + |0 - (1 - \nu_n(x^*)| = 2(1 - \nu_n(x^*)),$$

where x^* denotes the desired formation. The evolution of $||\nu_n - \Pi_\infty||_1$ for different potential error bound are shown in Fig. 7.3 , where $v_n(x^*)$ is calculated as the relative frequency of sampling x^* in 1000 annealing steps. The plot suggests that when the potential error bound $\Delta_z \leq \frac{1}{2}\Delta_U$, the convergence trend is roughly same as the noise-free case. On the other hand, when Δ_z is relative large, the convergence trend is barely observed.

With the sensor random-error being modeled as AWGN, similar convergence properties were observed in simulations. As one knows, for normal distribution, 99.7% samples lie in $[-3\sigma, +3\sigma]$, which is roughly comparable to the former cases with $\Delta_z = 6\sigma$. Hence, the case $\sigma = 1$ should be comparable with the case $\Delta_z = 5$, and the case $\sigma = 5$ corresponds to the case $\Delta_z = 30$. In the simulation, it was observed that the convergence rate of $\sigma = 1$ is slightly faster than the case $\Delta_z = 5$ with uniform distribution. Similar results can be observed by comparing cases $\sigma = 5$ and $\Delta_z = 30$. The reason is due to the bell shape of the normal distribution, where probability densities concentrate at the center.

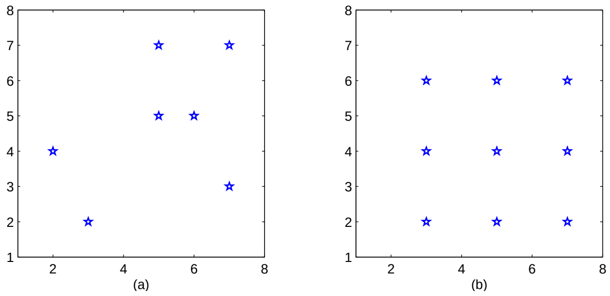

Fig. 7.2: The initial and desired configuration for 9 vehicles. (a) initial configuration; and (b) desired configuration

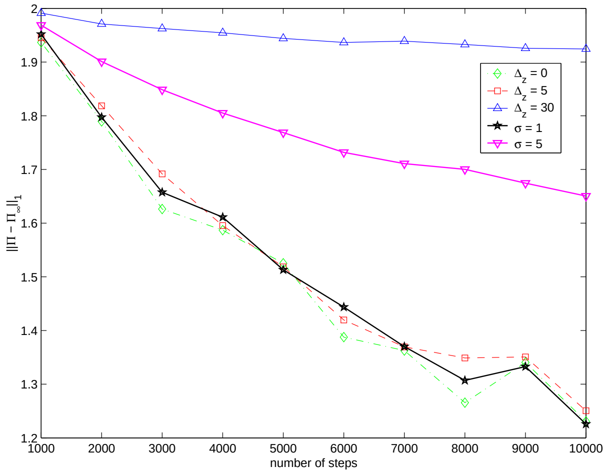

Fig. 7.3: Comparation of the evolution of $\|\nu_n - \Pi_\infty\|_1$ for different sensor noise.

Chapter 8

MPC based motion control of Car-like autonomous vehicles

8.1 Introduction

In this thesis, a two layer hierarchical scheme is adopted in the collaborative UAV swarms control system design (see figure 8.1). In the system diagram, it can be seen that the Gibbs sampler based stochastic algorithm proposed in previous chapters belongs to the high level path planning module. It generates a sequence of way-points $\{p_i(n), n = 1, ..., N\}$ for each UAV/robot i. The sequence of way-points are then used to generate smooth continuous curves $\{x_i(t)\}$ which pass the way-point $p_i(n)$ at time $t(n)$. The continuous curve $\{x_i(t)\}$ for UAV/robot i is usually called a *reference trajectory*. One can imagine $\{x_i(t)\}$ as the *image* of the discrete *path* $\{p_i(n)\}$ on the continuous mission space. The goal of lower-level system in figure 8.1 is to design controllers such that UAVs/robots can follow the *reference trajectories*.

In practice, however, dynamic and kinematic constraints, e.g., nonholonomic constraints, prohibit autonomous vehicles from following arbitrary *reference trajectories*. So the *reference trajectory* have to be generated by solving the vehicles' dynamic and kinematic differential equations, which is considered in the low level motion control design. Moreover, in the high-level path planning module, local collision avoidance, small obstacle/moving threat avoidance, as well as input/state saturation are not fully addressed. In this chapter, a Nonlinear Model Predictive Control (NMPC) approach based low-level

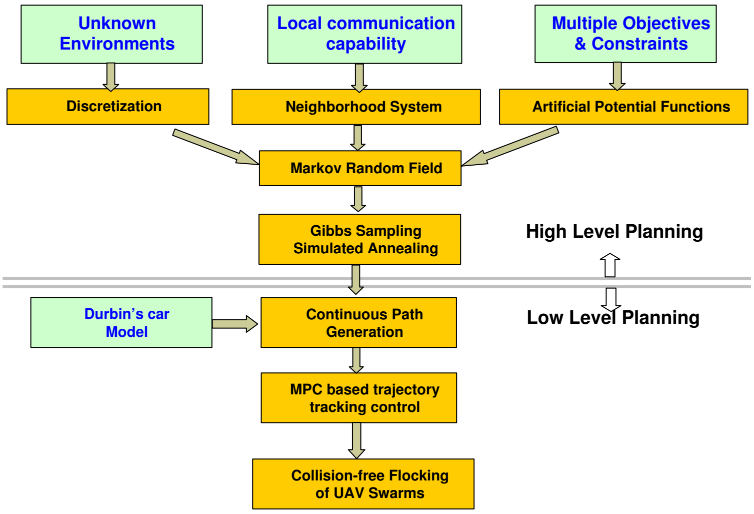
Unknown Environments
Local communication capability
Multiple Objectives & Constraints
Discretization
Neighborhood System
Artificial Potential Functions
Markov Random Field
Gibbs Sampling
Simulated Annealing
High Level Planning
Low Level Planning
Durbin's car Model
Continuous Path Generation
MPC based trajectory tracking control
Collision-free Flocking of UAV Swarms

Fig. 8.1: Gibbs sampler based collaborative control diagram

motion controller is proposed to address all aforementioned requirements. The low-cost, robust and multi-objectives oriented control scheme was then verified through simulations.

The remaining of this chapter is organized as follows. In section 8.2, the kinematic model of car-like vehicle and controllability is first studied. Section 8.3 discusses the generation of admissible *reference trajectory* with nonholonomic constraints for a car-like vehicle. Finally, the design and implementation of a NMPC based approach is addressed in section 8.4.

8.2 Kinematic model of car-like vehicle and controllability analysis

The kinematic and dynamic models of wheeled mobile robots have been considerably studied in the robotics control communities in past decades. The main feature of the kinematic model is the presence of nonholonomic constraints due to the *rolling* *without slipping* constraint between the wheels and the ground. The nonholonomic constraints impose many difficulties in control design. In particular, Brockett have shown that a linearized nonholonomic model has deficiency in controllability and there is no time-invariant linear control to guarantee the tracking error convergence zero [86]. To illustrate the nonholonomic constraints, we first look at the simplest wheeled mobile robot model, unicycle, as shown in figure 8.2. The configuration of the unicycle $q = (x, y, \theta)$ consists of generalized coordinates: position coordinates (x, y) at which the wheel of unicycle contact with ground, and the angle θ measuring the wheel orientation with respect to the x axis. All possible configurations form configuration space are denoted as $Q = \{q\}$. Due to the *rolling without slipping* constraints, the generalized velocity variables $\dot{q} = [\dot{x}, \dot{y}, \dot{\theta}]$ satisfy the following equation:

$$\begin{bmatrix} \sin\theta & \cos\theta & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = 0. \tag{8.1}$$

This equation is usually called the nonholonomic constraint equation. In general case, the constraint can be written as

$$C(q)\dot{q} = 0, \tag{8.2}$$

where $C(q)$ is the constraint matrix. The nonholonomic constraints restrict general velocities $\dot{q}$ living in the null space of the constraint matrix $C(q)$.

For unicycle model, by considering the nonholonomic constraint equation (8.1), the

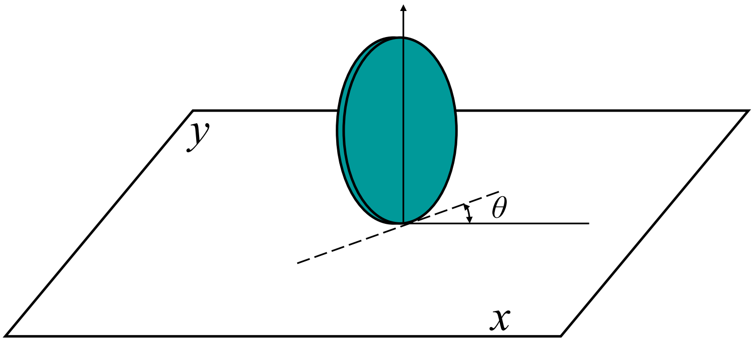

Fig. 8.2: The unicycle model

kinematic model can be expressed as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{8.3}$$

where v and ω are two control inputs, linear velocity and angular velocity respectively.

The unicycle is the simplest wheel-base robot model, which is only of conceptual interest in studies. In practice, people frequently adopt car-like mobile robot in ground UAV applications due to the relative lower cost and the convenience for applications. There are two types of car-like mobile robot: front-wheel driving vehicle and rear-wheel driving vehicle. They basically share similar structure of kinematic model.

The configuration of the rear-wheel driving vehicle is shown in fig 8.3. In this model, the generalized coordinates $q = (x, y, \theta, \phi)$, where (x, y) are the cartesian coordinates of the center point of the rear axle, θ is the heading angle of the car body with respect to the x axis, and ϕ is the steering angle. In figure 8.3, l is the distance between the front axle and the rear axle.

In the kinematic model of car-like vehicle, four nonholonomic constraints must be

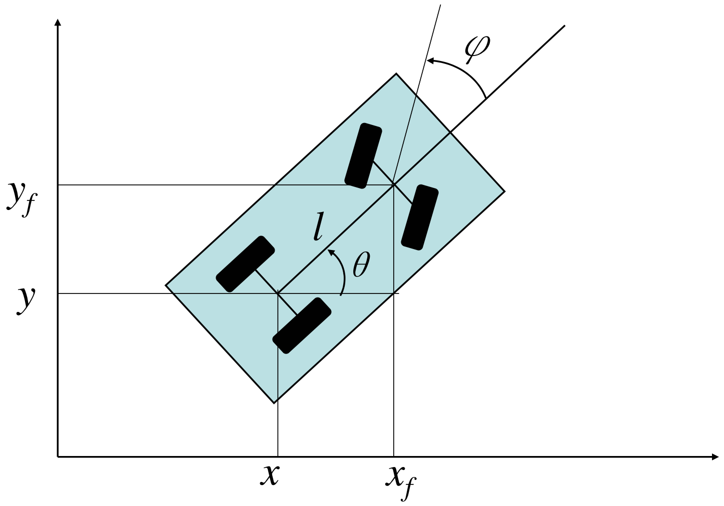

Fig. 8.3: The rear-wheel driving car model

satisfied with each one associated with one wheel. Let $(x_{fr}y_{fr})$ and (x_{fl},y_{fl}) be coordinates of the center point of the right front wheel and the left front wheel respectively. Assuming the two wheels are parallel, the nonholonomic constraints for the front wheels can be expressed as

$$\begin{aligned}\dot{x}_{fr}\sin(\theta+\phi)-\dot{y}_{fr}\cos(\theta+\phi)&=0\\ \dot{x}_{fl}\sin(\theta+\phi)-\dot{y}_{fl}\cos(\theta+\phi)&=0.\end{aligned}\tag{8.4}$$

Denote (x_f,y_f) as the coordinate of the front axle center point. Since $x_f=x_{fr}+x_{fl}$ and $y_f=y_{fr}+y_{fl}$, the two nonholonomic constraints are linearly dependent. So, they can be simplified as one constraint. Same procedure can be used to reduce the nonholonomic constraints of rear wheels. Therefore, four nonholonomic constraints are boiled down two nonholonomic constraints. In specific,

$$\begin{aligned} \dot{x}_f \sin(\theta+\phi) - \dot{y}_f \cos(\theta+\phi) &= 0 \\ \dot{x}\sin\theta - \dot{y}\cos\theta &= 0. \end{aligned} \tag{8.5}$$

Considering the rigid body constraints, the center of front axle (x_f, y_f) satisfies

$$\begin{bmatrix} x_f \\ y_f \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} l + \begin{bmatrix} x \\ y \end{bmatrix} \tag{8.6}$$

The first nonholonomic constraint of (8.5) can be rewritten with only general configuration q involved.

$$\dot{x}\sin(\theta+\phi) - \dot{y}\cos(\theta+\phi) - \dot{\theta} l \cos\theta = 0 \tag{8.7}$$

Assume that control input v, ω are linear velocity and steering velocity respectively, the kinematic model of rear-driving vehicle can be expressed as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ \tan\phi/l & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{8.8}$$

One can easily verify that (8.8) satisfies the nonholonomic constraints (8.5) and (8.7). Note that when $\phi = \pm\frac{\pi}{2}$, the model becomes singular. This corresponds to the situation where the front wheel heading is orthogonal to the longitudinal axis of the car. In practice, the range of steering angle ϕ should be restricted to prevent the singularity case.

The kinematic front-wheel driving vehicle can be obtained similarly with linear veloctiy $|v| = \sqrt{\dot{x}_f^2 + \dot{y}_f^2}$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\phi & 0 \\ \sin\theta\cos\phi & 0 \\ \sin\phi/l & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{8.9}$$

Remark 8.2.1 *In the previous study, the kinematic model of car-like vehicle is simplified as a bicycle kinematic model. In practice, the kinematic model with wheel coordinates should be used to address the presence of actuators and sensors on the wheel axle, as well as tire deformation. Nevertheless, the kinematic model discussed in this section contains main feature of vehicle kinematics and is feasible for illustrate the control design in the rest sections.*

Without loss of generality, in the rest of this section, the rear-driving vehicle model is considered in the motion control design. The controllability of the wheeled car is firstly studied. We start the analysis by rewriting the system equation (8.8) as

$$\dot{q} = g_1(q)v + g_2(q)\omega, \tag{8.10}$$

where $g_1(q) = [\cos\theta \quad \sin\theta \quad \tan\phi/l \quad 0]^T$, $g_2(q) = [0 \quad 0 \quad 0 \quad 1]^T$. Clearly, the kinematic model (8.10) is a driftless, nonlinear system. The driftless means that any configuration $q \in Q$ is an equilibrium point provided zero input. Given the initial condition $q(t_0) = q_0 \in Q$ and admissible control input $u(t) \in \mathscr{U}$ on $[t_0, t]$, the unique solution of (8.10) is denoted as $q(t, t_0, q_0, u)$, or simply $q(t)$.

Definition 8.2.1 *A system is controllable if for every two configuration $q_1, q_2 \in Q$, there* *exist a finite time* $T > 0$ *and an admissible control* $u : [0,T] \in \mathscr{U}$ *such that* $q(T,0,q_1,u) = q_2$

Following the nonlinear control theory in Isdori's book [87], the *Lie algebra rank condition* is a very powerful tool used to test the controllability of a driftless nonlinear system. In this case, one just need to check if the general controllable form is of full rank. In specific,

$$rank[g_1, g_2, [g_1, g_2], [g_1, [g_1, g_2]], [g_2, [g_1, g_2]], ...] = 4. \tag{8.11}$$

Observing that

$$g_3 \triangleq [g_1, g_2] = \frac{\partial g_2}{\partial q} g_1 - \frac{\partial g_1}{\partial q} g_2 \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\cos^2 \phi l} \\ 0 \end{bmatrix} \textit{ and } [g_1, g_3] = \begin{bmatrix} -\frac{\sin\theta}{\cos^2\theta l} \\ \frac{\cos\theta}{\cos^2\theta l} \\ 0 \\ 0 \end{bmatrix}, \tag{8.12}$$

one can conclude that the rank condition of the controllable form is indeed full rank, and thus the controllability holds.

8.3 Trajectory Generation for car-like autonomous vehicle

In this section, the generation of the continuous *reference trajectory* is investigated for car-like vehicle given predetermined way-point sequences in the free space. One of the most interesting problem in the literature is the shortest path problem, which usually associated with the time-optimal trajectory. However, nonholonomic constrains of the car-like vehicle present difficulties in solving such kind of problems.

In 1957 [88], Dubin studied the shortest trajectory problem for a very simple model, the unicycle model with constant linear velocity 1, i.e.,

$$\begin{cases} \dot{x} = \cos\theta \\ \dot{y} = \sin\theta \\ \dot{\theta} = \omega \end{cases}$$

with configuration $q = [x, y, \theta] \in \mathbb{R}^2 \times S^1$, and control input ω. The goal is to find the time-optimal trajectory to connect the initial configuration q_0 and the terminal configuration q_t with input constraints $|\omega| \leq 1$.

Although this problem was initial proposed by A. A. Markov in 1989, Dubins was the first one who studied the problem in detail [89]. After then, similar problems have been studied in [90], [91].

Dubins showed that the optimal trajectories are concatenations of at most 3 pieces of basic elements. The basic elements include a line segment (L) and an arc (C) with radius 1. Moreover, he showed that there are only seven combinations are allowed to ensure optimality of the generated trajectory. Precisely, they must be of the forms in $L, C, L-C, C-C, C-L, C-C-C, C-L-C$. For example, the form $L-C-L$ is excluded from optimal trajectory. Moreover, for the case of $C-C-C$, he showed that the middle arc should be lie in $(\pi, 2\pi)$, and total length of three arcs should less than 2π. A typical example of the optimal trajectory for dubins' car is shown in figure 8.4, where a C-L-C type trajectory is used to connect the initial and terminal configurations. In [92], the shortest path synthesis for dubins' car are studied completely.

However, in the dubins' optimal trajectory, there may exist discontinuous curvature at the connection point of two successive pieces, line-arc or arc-arc (with opposite

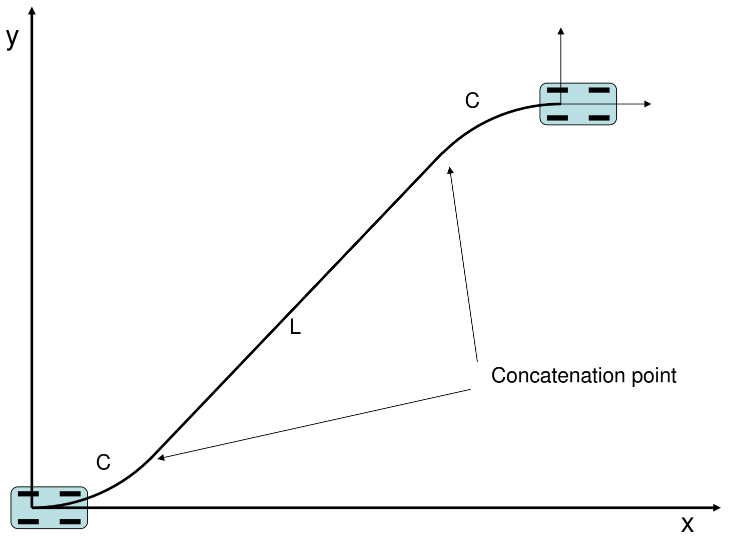

Fig. 8.4: An example of the optimal trajectory for dubins' car

direction of rotation). To follow (exactly) such trajectory a car-like vehicle would be constrained to stop at the end of each connection point. To deal with the problem, Sussmann [89] studied a generalization problem by controlling the angular acceleration in stead of the angular velocity, which is the equivalent model of the car-like vehicle, i.e.,

$$\begin{cases} \dot{x} = \cos\theta \\ \dot{y} = \sin\theta \\ \dot{\theta} = \omega \\ \dot{\omega} = u \end{cases}$$

where the control input is bounded ($u < u_{max}$), and the curvature of the state trajectory $\kappa \in [-\kappa_{max}, \kappa_{max}]$ is bounded. The goal is to find the time-optimal trajectory in the feasible configuration space $C \in \mathbb{R}^2 \times S^1 \times [-\kappa_{max}, \kappa_{max}]$. He showed that the optimal trajectory consists of line segments, arcs and clothoids.

A clothoid, also known as a Cornu spiral, is a curve parameterized by two Fresnel integrals $(S(t),C(t))$. In the canonical clothoid (Cl), the $x(t)$ and $y(t)$ coordinates can be expressed as:

$$\begin{aligned} x(t) &= \int_0^t \cos(\frac{B\tau^2}{2})d\tau \qquad (8.13) \\ y(t) &= \int_0^t \sin(\frac{B\tau^2}{2})d\tau \qquad (8.14) \end{aligned}$$

Along a clothoid curve, the curvature depends linearly on the arc length and varies continuously from $-\infty$ to $+\infty$. For the clothoid defined in (8.13), it corresponds to the state trajectory $(x(t),y(t))$ of system (8.13) with control input equals to constant B. Because the curvature along the reference trajectory should be bounded, only a part of the clothoid can be utilized in the optimal path generation.

Moreover, Sussmann showed that along the time-optimal trajectory for system (8.13), the number of basic elements (L, C, and Cl) cannot be bound above. In some extreme case, there may exist a time-optimal trajectory that involves with infinite chattering [89], which is not allowed in practice.

A practical way addressing the problem is to study a sub-optimal problem by restricting the maximum number of basic elements. The analytical study of the Sub-optimal Continuous-Curvature (SCC) trajectory planning can be found in [93]. In the SCC trajectory, there exist at most 9 pieces of basic elements. For each dubins' optimal trajectory there is a corresponding sub-optimal trajectory. For example, the C-L-C in dubins' model may become Cl-C-Cl-L-Cl-C-Cl in the sub-optimal trajectory. The curvature profile in this example is shown in figure 8.5. By replacing the arc A and C in the left plot with curve 1-2-3 and 5-6-7 in the right plot, the curvature profile is continuous.

The key part of this approach is to replace any arc segment in dubins' optimal

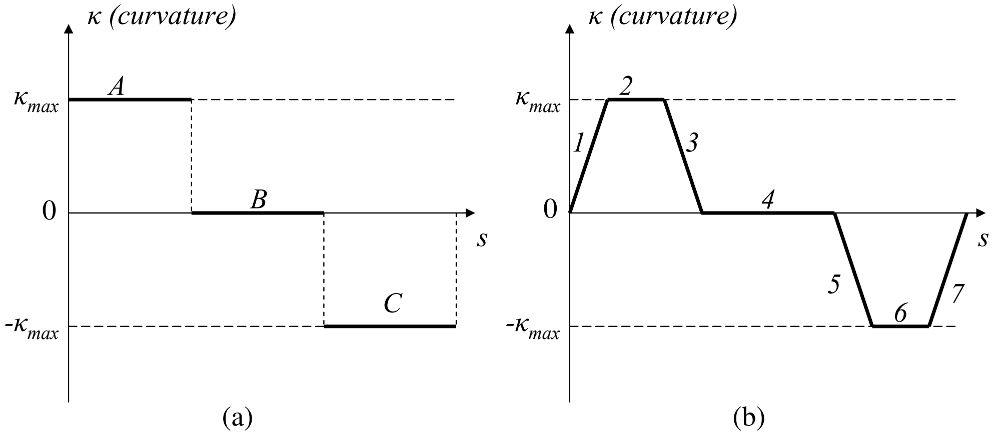

Fig. 8.5: Curvature profile: (a) Dubins' optimal trajectory and (b) SCC trajectory

trajectory with a *continuous-curve-turn*. In precise, the arc is replaced by a Cl-C-Cl combination, where the start and end point has curvature equals to 0 [93].

In general, the local sub-optimal trajectory planner works as follows. First generate the dubins' optimal trajectory using the synthesis approach in [92]. For each arc segment, replace it by a curve consisting of Cl-C-Cl. A typical sub-optimal trajectory is shown in figure 8.6.

In [94], they further compared the dubins's optimal trajectory and the SCC trajectory. In their extensive simulation results, they showed that the total length of the SCC trajectory is only about 1.1 times longer than the dubins' optimal trajectory. In the rest of this chapter, it is assumed that the reference trajectory $q_r(t)$ is generated by the sub-optimal method. The corresponding reference control input are linear velocity $v_r(t)$ and steering velocity ω_r.

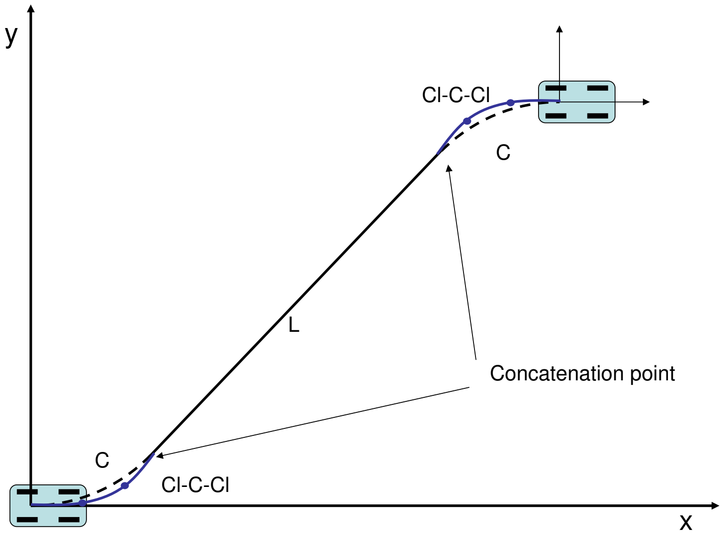

Fig. 8.6: An example of the sub-optimal trajectory

8.4 Model Predictive Control (MPC) based trajectory tracking control

8.4.1 Review of Model Predictive Control

The Model Predictive Control (MPC) approach is an advanced method of the process control that integrates optimal control, stochastic control and multivariable control approaches. The MPC approach has been considerably developed and widely used from chemical industries, oil refineries to food process industries since late seventies [95]. One of the most important advantages of MPC lies in that it can easily used to handle the non-linear system control with input/output constraints, e.g., actuator saturation, which are frequently encountered in practice [96]. The main idea of MPC approach is to choose the control action by repeatedly solving online an optimal control problem. The general framework of MPC approach is depicted in figure 8.7.

First, a model is used to predict the process output at future time instants (hori-

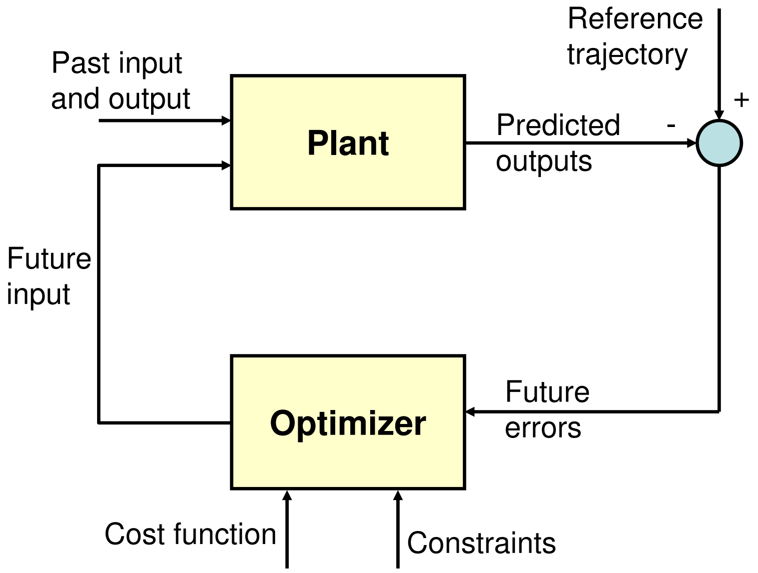

Fig. 8.7: Diagram of Model Predictive Control (MPC)

zon) given the current states and the candidate optimal future control input sequences. The proposed optimal future control sequences are then generated by optimizer minimizing the cost function (the cost is incurred by future errors and control input) subject to (input/output/state) constraints. Finally, a receding strategy, which involves recursively solving a finite horizon optimal control at each time instant, is used for control purpose.

More precisely, in discrete time the model predictive control approach can be formulated as follows:

$$\begin{aligned} x(t+1) &= f(x(t),u(t),w(t)) \\ y(t) &= g(x(t),u(t)), \end{aligned} \tag{8.15}$$

where $u(t)$ is the control input, and $w(t)$ is the noise or disturbance. At time t, Given a control input sequences $\{u_k\}_t^{t+N}$ and initial state $x(t)$ at time instant t, for a finite time horizon $H = t,...,t+N$, the output $\{y_k\}_t^{t+N}$ can be calculate by the predict model (8.15). Let us define a cost/objective function as $J(\{x_k\}_t^{t+N}, \{y_k\}_t^{t+N}, \{u_k\}_t^{t+N})$ involving the future state trajectory, output trajectory and control effort. The optimal control input u^* is generated by minimizing the cost function, that is,

$$\{u_k^*\}_t^{t+N} = \arg \min_{\{u_k\}_t^{t+N}} J(\{x_k\}_t^{t+N}, \{y_k\}_t^{t+N}, \{u_k\}_t^{t+N}). \tag{8.16}$$

From time instant t to $t+\tau$, the optimal control inputs $\{u_k^*\}_t^{t+\tau}$ are then applied to the process or the plant. At time instant $t+\tau$, a new time horizon $T = \{t+\tau,...,t+N+\tau\}$ is generated. $\{u_k^*\}_{t+\tau}^{t+N+\tau}$ is calculated by solving similar finite horizon optimal control problem as (8.16) given $x(t+\tau)$ as initial state and the predictive model (8.15). $\{u_k^*\}_{t+\tau}^{t+2\tau}$ are then used for control during time interval $\{t+\tau,...,t+2\tau\}$. One can imagine the finite horizon as a window with size N moving along the time axis. The recursively solving finite horizon optimal control problem with size N and applied the optimal control during first tau step is essentially the core of MPC strategy. For this particular reason, MPC is also called Receding Horizon Predictive Control (RHPC).

8.4.2 Gradient Descent based MPC approach

In this section, a MPC approach is proposed for solving low-level trajectory tracking problem of a car-like autonomous vehicle. The stabilization and tracking control of nonholonomic vehicle has been addressed by many people in the past decades. The challenge is explicitly demonstrate by Brockett in [86], that is nonholonomic systems cannot be stabilized via smooth time-invariant state feedback. To address the problem, time-varying feedback control approaches were proposed by many researchers, for examples, approaches based on Control Lyapunov Function (CLF) [97][98], and adaptive control approach [99][100].

As mentioned in section 8.1, the low level motion control module has to simultaneously handle multiple objectives, such as bounded tracking error, collision avoidance, local obstacle avoidance, input/state saturation, and etc. However, aforementioned approaches fail to solve our problem since they usually only address stability and zero tracking error in the controller design. Inspired by [101], a Nonlinear Model Predictive Control (NMPC) based approach is proposed for the multi-objective oriented motion control of car-like vehicle.

Most commercially available MPC approaches were developed based on linear model and frequently demonstrated poor performance due to the high nonlinearity of the plant. This has motivated people to develop NMPC approach, where a nonlinear model was used to describe the plant more precisely. While the NMPC approach offers potential for improved the performance, it also imposes challenges in theoretical study and practical implementation, which are usually associated with the on-line solving nonlinear optimization problem at each sampling period to generate optimal control sequence. To solve the nonlinear optimization problem, the Maximum Principle can be used to derive a set of necessary conditions for optimal control sequence u^*. Unfortunately, it is hard to get explicit solutions from the necessary conditions in general. Gradient descent approach can then be used to numerically compute the optimal control.

In the next, the approach for a continuous time nonlinear system is briefly described, and then applied to solve our problem. we start to illustrate the approach by investigating a general finite horizon optimal control problem. The goal of the problem is to find the optimal control input $u^*(t) \in \mathscr{U}$, $t \in [t_0, t_f]$, such that

$$\begin{aligned} \text{minimize } J &= \psi(q(t_f)) + \int_{t_0}^{t_f} L(q(t), u(t), t)dt \\ \text{subject to } \dot{q} &= f(q(t), u(t), t) \\ q(t_0) &= q_0 \end{aligned} \tag{8.17}$$

where the first term $\psi(q(t_f))$ is the terminal cost, and the second term is the running cost. By introducing the costate vector $\lambda(t)$, the Maximum Principle indicates that the optimal control should satisfies the following conditions (the derivation can be found in appendix):

$$L_u + \lambda^T f_u = 0 \tag{8.18}$$

$$L_q + \lambda^T f_q + \dot{\lambda}^T = 0 \tag{8.19}$$

$$\psi_q(q_{t_f}) - \lambda^T(t_f) = 0 \tag{8.20}$$

where L_u, L_q are denoted as the partial derivative of L respect to u and q respectively. Similar for f_u and f_q. One can find that the costate propagates backward time with initial condition $\lambda^T(t_f) = \psi_q(q_{t_f})$, whereas the state propagates forwards time. This fact presents challenges to solve the optimal control $u^*(t)$ analytically. A numerical method was then proposed using the gradient descent method. It is outlined as follows:

1. For a given q_0, pick a control history $u^0(t)$. Let $i = 0$.

2. Propagate $\dot{q} = f(q, u, t)$ forward in time to create a state trajectory.

3. Evaluate $\lambda^T(t_f) = \psi_q(q_{t_f})$, and solve backward λ^T using $\dot{\lambda}^T = L_q + \lambda^T f_q$

4. Update control input $u^{i+1} = u^i + \delta_u$ with $\delta_u = -K(L_u + \lambda^T f_u)$, where K is a positive scalar.

5. Calculate $\delta J = J(u^{i+1}) - J(u^i)$. If $\delta J > 0$, reduce K and go back to step 4.

6. Let $i = i + 1$, and go back to step 2 until the solution converges.

The numerical method is widely used for solving optimal control problem in complex system. The major difficulties of the approach is the computation cost. In order to speed the solution converges, one should carefully select appropriate control trajectory to start the iteration. Once the optimal solution is available, one can then plug the nonlinear optimizer in the general MPC framework in the last section to get the NMPC approach.

The NMPC can easily handle multi-objectives oriented tracking control by jointly encoding individual objectives in in the objective function J. In our problem, the following form could be used:

$$J = \psi_{t_f} + \int_{t_0}^{t_f} (\mu^{tk} J^{tk} + \mu^{sc} J^{sc} + \mu^u J^u + \mu^o J^o + \mu^c J^c) dt, \qquad (8.21)$$

where J^{tk}, J^{sc}, J^u, J^o, and J^c are the objective/potential functions account for tracking performance, state saturation, control effort and saturation, obstacle avoidance, and collision avoidance respectively. μ^{tk},μ^{sc},μ^u,μ^o, and μ^c are weighting coefficients for each objectives. The design of the potential functions as well as weighting coefficient are challenging in order to get robust performance. In our problem, the potential function are designed in the following forms:

- Tracking performance J^{tk}

  Assume that the desired trajectory is $q_d(t) = \{[x_d(t), y_d(t), \theta_d(t), \phi_d(t)]\}$. The quadratic form of tracking error can be a good candidate, i.e.,

$$J^{tk} = (q - q_d)^T Q(q - q_d),$$

where Q is a positive diagonal matrix, which determines the relative importance of tracking error for different states.

- Terminal cost ψ_{t_f}

  Similar as tracking performance, ψ_{t_f} is selected as:

  $$J^{tk} = (q_{t_f} - q_d(t_f))^T Q_0(q_{t_f} - q_d(t_f)),$$

- Input/state saturation J^{tk}

  As aforementioned, when the steering angle $\phi = \pm\phi/2$, the kinematic model of rear-wheel vehicle will degenerate. This enforce the the steering angle must live in the safe range $[-phi^{sat}, phi^{sat}]$, where phi^{sat} is a positive scalar. The saturation cost could be

  $$J^{sc} = max(0, |\phi| - \phi^{sat})^2,$$

- Control effort J^u

  We use quadratic form to represent control effort:

  $$J^u = u^T Ru,$$

- Obstacle avoidance J^o A repulsive potential function is used for avoid obstacles. Assume the closest point on the obstacle surface to the vehicle are (x^o, y^o). J^o is proposed of following form

  $$J^o = \frac{1}{(x - x^o)^2 + (y - y^o)^2},$$

where (x,y) is location of vehicle.

- Collision avoidance J^c

$$J^c = \sum_j \frac{1}{\max(\sqrt{(x-x^j)^2+(y-y^j)^2}-R_{safe},\varepsilon)},$$

where R_{safe} is the safety range and ε is a small scalar used to prevent J^o from infinite or negative.

It is straightforward to extend the approach to discrete time nonlinear system in order to implement this approach to low-cost digital controller, one may refer to [101] to get the details.

8.4.3 Dynamic Programming based NMPC approach

In the last section, a gradient descent approach was proposed to solve the nonlinear finite horizon optimal control problem in the NMPC approach. It has to be noted that the convergence of the gradient descent approach is not guaranteed. Without carefully selecting the weighting coefficients in objective function, as well as initial control sequence and step size of control updates, this approach may lead to unstable performance, which brings out a lot of troubles in the controller design. The other difficulty of the gradient descent approach is that the computation time in different sampling period may varies a lot which may cause instability due to the maximum delay by computation.

To address these problems, a dynamic programming (DP) based approach is proposed. It is well known that the Dynamic Programming approach suffers from "curse of dimensionality" in general. However, since it is usually assume that autonomous vehicles have only limited actuation capability, by reducing the size of admissible control input, the DP approach can be used to solve the finite horizon optimal problem (8.22) in a reasonable time.

For discrete time case, the finite horizon optimal problem (8.22) can be rewritten as follows

$$\begin{aligned} \text{minimize } J &= \psi(q(N)) + \sum_{k=0}^{N-1} L(q(k), u(k)) \\ \text{subject to } q_{k+1} &= g(q(k), u(k)) \\ q(0) &= q_0. \end{aligned} \tag{8.22}$$

We assume that the control input $u(k) \in \mathscr{U}$ takes only discrete values. Denote $|\mathscr{U}|$ be cardinality of admissible control. The optimal control sequence can be recursively solved by the following DP algorithm:

1. Initially, let $J_0(q_0) = 0$.

2. For k=0,...,N-1, we have

   $$J_{k+1}(q(k+1)) = \min_{q(k)} (L(q(k), u(k)) + J_k^*(q(k))),$$

   where $q(k+1) = g(q(k), u(k))$.

3. Find the optimal control sequence associated with the optimal cost:

   $$J^* = \min_{q(N)} (\psi(q(N)) + J_N(q(N))).$$

The advantages to use the DP algorithm lie in two folds. First, the DP algorithm do not require that the objective function $L(q(k), u(k))$ and system dynamics $g(q(k), u(k))$ differentiable. Second, it guarantees the finite convergence time which is very important to in real-time control applications. However, we have to admit the “curse of dimensionality” of the DP algorithm prohibits the approach from being used in more complex system.

8.5 Simulation results

In this section, the performance of the proposed NMPC based low-level vehicle control is demonstrated in three simulation examples. In all simulations, the distance (l) between front axle and rear axle is set to be 0.8. The steering angle (ϕ) lies in $[-pi/4, pi/4]$. The saturation range of control input v, ω are: $v \in [0,5]$ and $\omega \in [-1,1]$.

8.5.1 Free-space Way-point Navigation

In this scenario, a single vehicle moving in a free space is considered. In the simulation, the reference trajectory is a SCC trajectory consisting of 7 pieces basic elements (Cl, C, L). Starting at origin $(0,0)$, the reference linear velocity v_r is constant 1. As shown

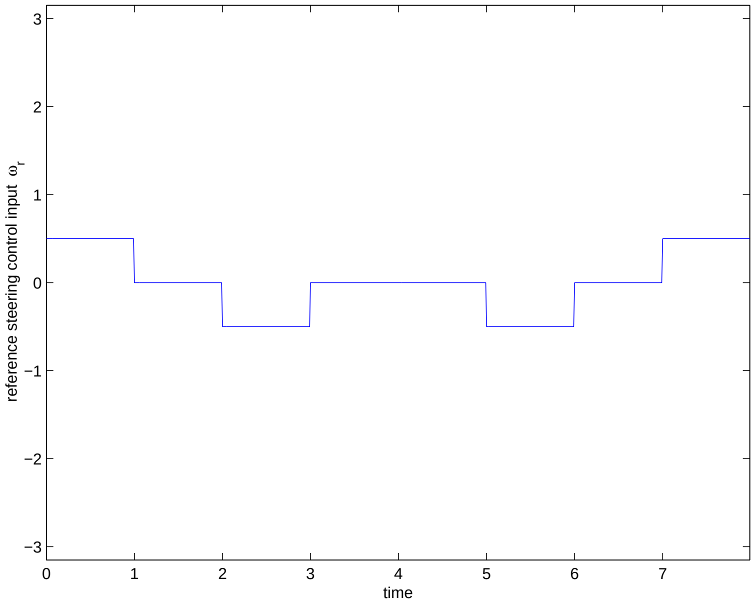

Fig. 8.8: The reference steering control input for free space trajectory tracking

in figure 8.8, the reference steering velocity $\omega_r(t)$ is set to be follows

$$\omega_r(t) = \begin{cases} 0.5, & 0 \leq t \leq 1 \\ 0, & 1 < t \leq 2; \\ -0.5, & 2 < t \leq 3; \\ 0, & 3 < t \leq 5; \\ -0.5, & 5 < t \leq 6; \\ 0, & 6 < t \leq 7; \\ 0.5, & 7 < t \leq 8; \\ 0, & t > 8. \end{cases}$$

In the simulations using gradient descent MPC approach, the selected weighting coefficient are: $\mu^{tk} = 20$,$\mu^{sc} = 300$,$\mu^{u} = 5$,$\mu^{o} = 0$. The horizon time for each sampling period is set to be 0.3s, and the step size K are initially set to be 1-e3. In the DP based

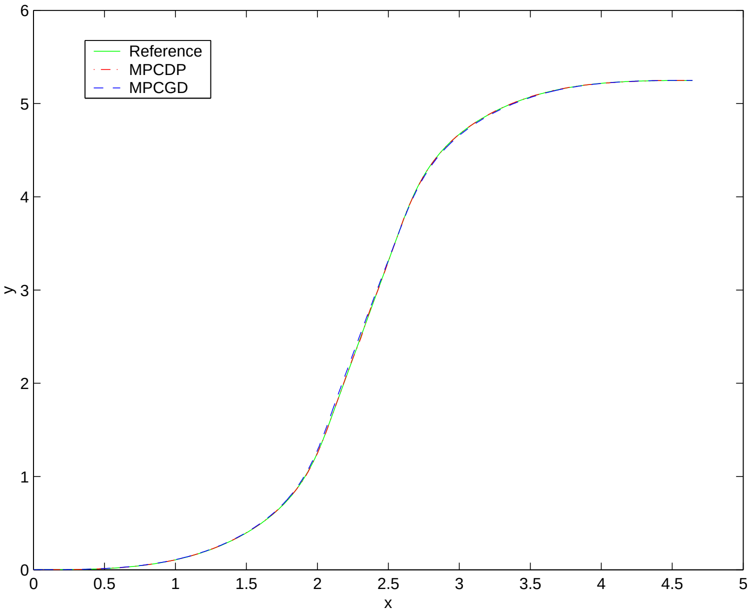

Fig. 8.9: Free space trajectory tracking with MPC based approach

NMPC approach, the sampling period is set to be the same as the one in the gradient approach. The linear velocity input is set to be constant 1. The steering input can only take three values: -2.5, 0, and 2.5. Simulations results shows that both approaches demonstrate an excellent tracking performance 8.9.

Moreover, we compared the trajectory tracking error $\sqrt{(x(t)-x_r(t))^2+(y(t)-y_r(t))^2}$, and heading angle tracking error $\theta(t)-\theta_r(t)$ in figure 8.10. The trajectory tracking errors for both approaches are bounded below 0.005. The DP based approach shows larger heading angle deviation than gradient descent approach. The main reason is due to the limited steering control capability.

We further compared the range of computation time at each sampling period. The range for gradient based MPC approach is about (0, 1.5) second, which is much larger than the sampling period 0.3 second. The range for DP based approach is roughly around

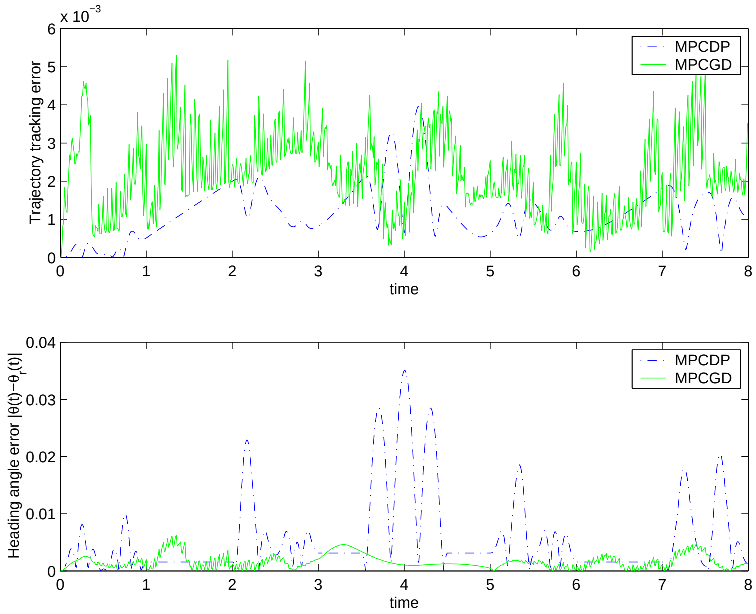

Fig. 8.10: Comparison of free space tracking error for two MPC based approaches

0.02 second. Although the coding efficiency of Matlab may greatly affect the computation time, the large variation of computation cost potentially present barriers to apply gradient based MPC approach in real-time control.

8.5.2 Trajectory tracking with obstacle avoidance

In the high-level path planning module, obstacles are assumed much larger than the size of autonomous vehicle. In order to avoid small obstacles within vehicle's local sensing range, low-level controllers are designed to deal with the problem. For simplicity, it is assumed that obstacles have circular or eclipse shaped. General shapes of obstacles can be approximated by an eclipse envelope. In the simulation, the reference trajectory start at origin $(0,0)$. The linear velocity v_r is constant 1. The reference steering velocity

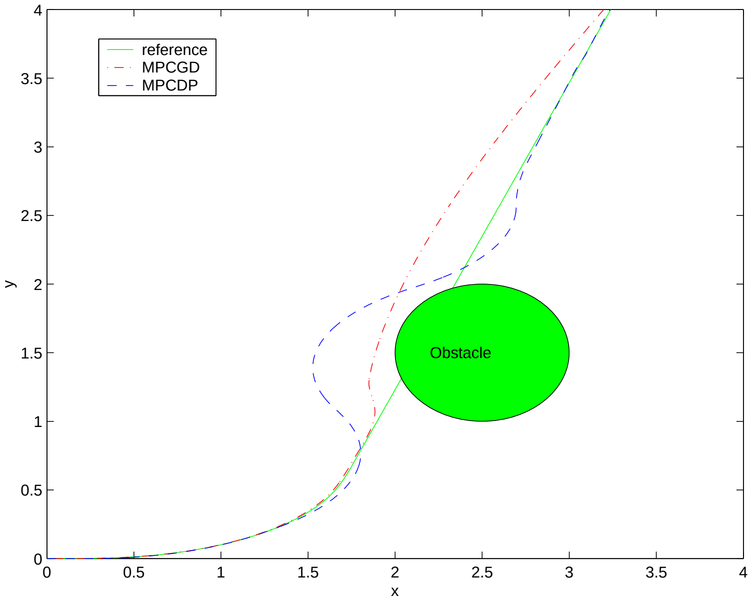

Fig. 8.11: Comparison of local obstacle avoidance for two MPC based approaches

$\omega_r(t)$ is set to be follows

$$\omega_r(t) = \begin{cases} 1, & 0 \le t \le 0.5; \\ 0, & 0.5 < t \le 2; \\ -1, & 2 < t \le 2.5; \\ 0, & t > 2.5. \end{cases}$$

A circular obstacle is located along the reference trajectory with center at $(2.5, 1.5)$ and radius 0.5. Figure 8.11 shows that both approach successfully avoid from hitting towards the obstacle. The vehicle trajectory with DP based MPC approach has larger deviation from reference trajectory than that with gradient descent approach. The primary reason is due to the limited steering control capability.

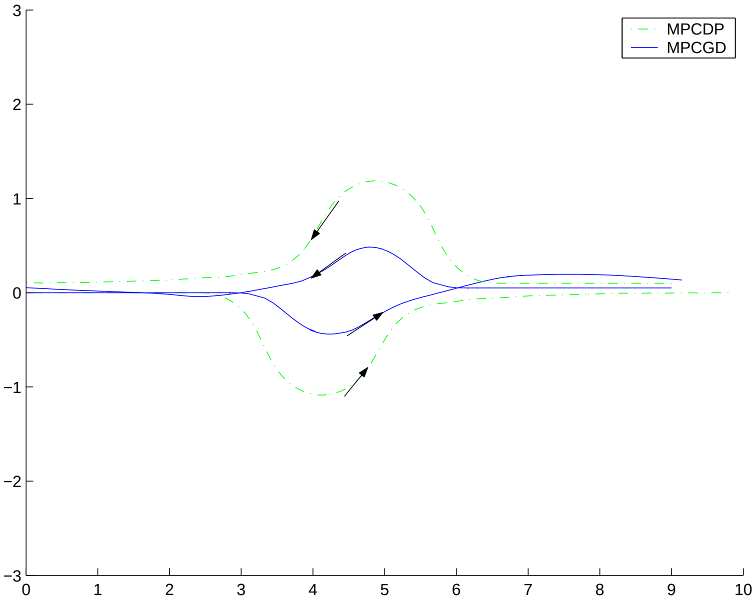

Fig. 8.12: Comparison of local collision avoidance for two MPC based approaches

8.5.3 Multiple vehicle tracking with collision avoidance

In this simulation, two vehicles' heading directions are initially opposite to each other. We assume that the two vehicle are planning to move the other's location. So they are supposed to collide in the mid of their ways. By adding the collision avoidance potential function component in the objective function, simulations show that both approaches yield good performance in avoiding collision (see figure 8.12). The dashed curves are vehicles' trajectories with DP based MPC approach. Similar as previous simulations, the limited actuator's capability results in large deviation from reference trajectory comparing with the gradient based approach.

As discussed previously, A typical case that two vehicles colliding into each other happens when vehicle *A* moves from cell at southwest to cell at northeast, and at the same time vehicle *B* moves from cell at northwest to cell at southeast (or similar permutation).

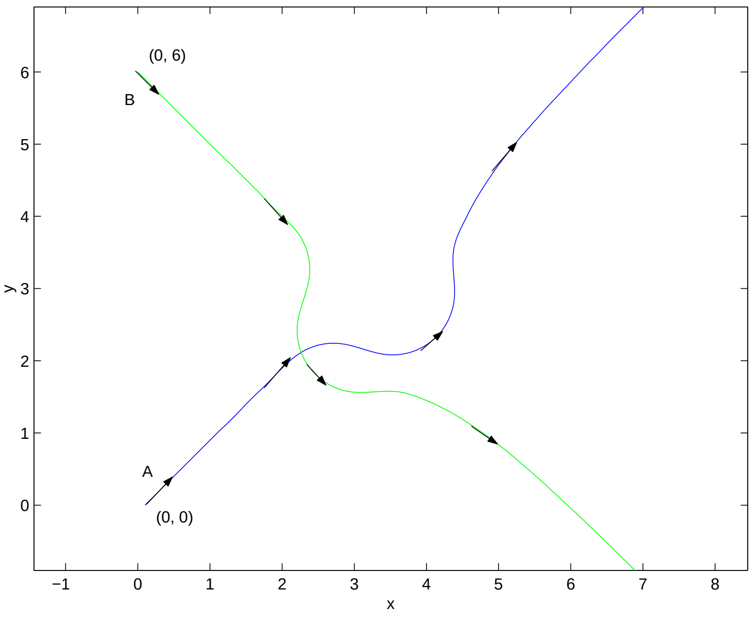

Fig. 8.13: An example of local collision avoidance with DP based MPC approach

We simulate the scenario using the DP based MPC approach. Two vehicle are initially sitting at $(0,0)$ and $(0,6)$ with orientation $pi/4$ and $-pi/4$ respectively. Without collision avoidance control, they will collide at $(3,3)$. Figure 8.13 shows our DP based approach achieves collision avoidance successfully.

Remark 8.5.1 *The MPC approach demonstrates great advantages in low-level autonomous vehicle control design. The main advantage of using MPC control lies in that it can easily accommodate several control objectives in a fairly simple framework. Comparing with the gradient descent, the DP based approach offers a lower computational cost and more robust performance. However, the DP based approach suffers from "curse of dimensionality", which limit its applications.*

Chapter 9

Conclusions

This dissertation concentrates on the collaborative control of homogeneous UAV swarms with communication, sensing, and actuation constraints. In particular, a fundamental dilemma, local minima entrapment problem, that existed in the artificial potential function (APF) approach is addressed.

In this dissertation, a novel distributed stochastic approach is proposed to solve the local minima entrapment problem. By modeling vehicle networks as a dynamic Markov random field (MRF), the Gibbs sampler based simulated annealing approach is proposed to coordinate vehicles' motion. Similar to the APF approach, this approach belongs to potential based approach that mimics the bacteria foraging in nature. All interactions among vehicles and environments are encoded as Gibbs potential functions. However, compared to the APF approach, the present approach provides a more general framework by adding the annealing process. The APF approach in fact is only a special case of the Gibbs sampler based approach with temperature fixed at zero all the time. By tuning the cooling scheduling, the system can achieve different convergence rate. Theoretical studies and extensive simulation results have been conducted to validate the novel approach. It have been shown that by choosing appropriate cooling schedule, this stochastic approach is guaranteed to lead the vehicle networks to achieve group objects/tasks. Two examples, a battle field scenario and formation control, were used in the simulation. But it is easy to extend our approach to many applications that have been mentioned in the first chapter.

Although the present approach provides a general solution to complement the APF approach, the stochastic nature and sequential sampling introduce long travel times and maneuvering costs to UAV networks that present barriers to apply our approach in practice. Two approaches are proposed in this dissertation to reduce the execution time and improve the performance.

First, a fully distributed parallel sampling algorithm was investigated to reduce the delay caused by the sequential sampling. Generally, it is hard to show that fully parallel sampling algorithm lead to the same global minimizer as the sequential counterpart. Fortunately, in some special cases, we proved that the parallel algorithm will lead to quasi-desired configurations.

Second, we proposed a hybrid scheme that combines the advantages of both the stochastic exploration algorithm and the deterministic gradient decent method. The hybrid scheme switches between two approaches to achieve both fast maneuvering and free of entrapment. There are two key parameters, switch waiting time d and the duration for stochastic perturbation N, that significantly affect the system performance. Some preliminary analysis on how to optimally select the parameters are provided. Convex trends of performance change are observed through simulations. By implementing memories to vehicles, an improved hybrid scheme is also proposed. In this scheme, vehicles use memory to online learn "dangerous" locations and reduce the probability to reach those locations again. In nature, the improvement is to reconstruct the potential surface by estimating the local minimum gradually. Simulation results have shown that the improved scheme achieves better performance than the original one.

The robustness of our approach is also addressed in the dissertation. We study the convergence properties of the stochastic approach under different types sensor errors including *range-error* and *random-error*. It is shown that under moderate sensor noise, i.e., the sensor noise is bounded by half of the minimum potential difference, the convergence properties of our proposed algorithm are conserved.

We further investigated the low-level motion control to deal with nonholonomic constraints in car-like UAV model. The proposed MPC based approach have shown its advantages to easily accommodate multiple objectives and constraints, e.g., minimizing tracking error, avoiding actuator/state saturation, local obstacle and collision avoidance. We compared two approaches, gradient descent based approach and dynamic programming based approach. Both of them have advantages and disadvantages. For car-like vehicle with limit control input, the dynamic programming based approach is more preferred.

There are several possible directions to extend the work reported in this dissertation.

In the modeling of vehicle networks as MRF, only singleton and pairwised cliques are considered in this dissertation to describe the interactions between vehicles and environments. One interesting direction is to study more complicated forms of cliques, which might lead to more interesting collective behaviors. Another way to generate appealing emergent aggregative behaviors is via potential function design, in this dissertation and [77], we have cooked several examples with potential applications. In practice, it is of interest to study a set of element collective behaviors and associated artificial potential functions (APFs), which could be used to construct an APF component library. Based on this, one could then study the synthesis of basic collective behaviors. For example, in this dissertation, a linear combination of basic APF components is proposed for group task synthesis. As we have previously discussed, the design of weighting coefficients is a challenging and important issue as it may directly impact the nodes behavior and the convergence rate of the algorithm. So this is could be a future direction to improve the algorithm performance.

Although in the theoretical study of convergence properties of the stochastic algorithm, the logarithm rate cooling schedule is used which only leads to a polynomial convergence rate. In practice, however, many fast cooling schedule have been used in general simulated annealing algorithm with good performance [20]. One could then “borrow” the existing cooling schedules to accelerate the convergence rate in the future study. Moreover, it is of interest to estimate the convergence rate. With more accurate convergence rate estimation, one could theoretically determine the optimal switch parameters in the hybrid control scheme.

In the hybrid scheme, there are several possible directions for the future study. First of all, the switching algorithm can be treated as a extreme case of cooling schedule, i.e., the temperature follows a non-continuous curve. One could easily extend the switching pattern by combining several different rate of cooling schedule. One could also try adaptively decreasing or increasing cooling schedule to achieve optimal performance. The second direction is to study the selection of switching parameters. In the dissertation, we only provide a preliminary study for off-line parameters selection. One could design a online adaptive scheme to determine best switch parameters based on previous experience. The last but not the least is to extend the use of memories. There are several possible ways along this direction. First, one could share the memory information with neighbors to accelerate the learning speed. Second, collaboratively map building can be used to enable vehicle networks have more knowledge of environments than “risk” level map, and thus may potentially lead to better performance.

Appendix A

Appendix: Ising Model

In this appendix, we provide a classical example, Ising model ,to illustrate several basic concepts in Markov Random Field and Gibbs Field.

The *Ising Model* was introduced by Ising (1925) for understanding qualitatively the phenomenon of phase transition in ferromagnetic materials. In the Ising's finite model, the ferromagnetic materials are modeled as a large 2D magnetic particles array. Each particle has two phases: pointing upward and downward. The attraction/replusion force between two (spatial) adjacent particles induces magnetic potential energy. The spatial structure and the local interactions can then be modeled as a Markov Random Field (MRF), where the sites is described as a 2D lattices, i.e., $S = Z_m^2$, and the phase space $\Lambda = \{+1, -1\}$. The neighborhood system is depicted as in figure A.1, where each node has at most 8 neighboring nodes. There are two types of cliques in this model: singleton and pairwise clique. The Gibbs potential is

$$\Psi_{\{s\}}(x) = -\frac{H}{k}x(s),$$
$$\Psi_{\{s,t\}}(x) = -\frac{J}{k}x(s)t(s),$$

where $\{s,t\}$ is a pairwise clique, with $s \in \mathscr{N}_S$; k is the Boltzmann constant; H is the external magnetic field, and J is the internal energy of an element dipole. The potential/energy

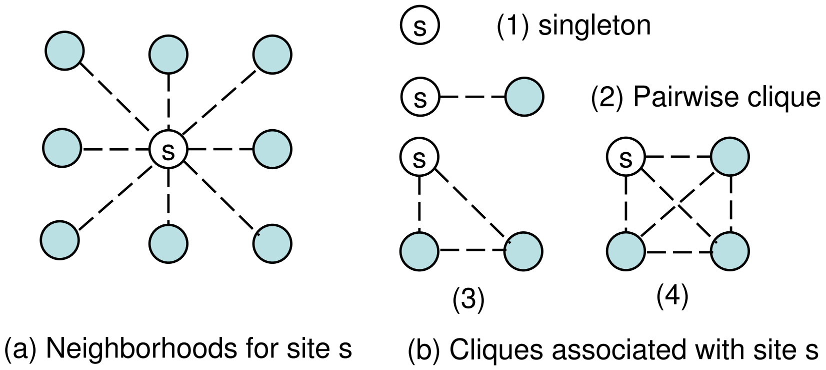
s
(1) singleton
s
(2) Pairwise clique
s
s
s
(3)
(4)
(a) Neighborhoods for site s
(b) Cliques associated with site s

Fig. A.1: Neighborhoods and cliques in Ising model: (a) neighborhoods for site *s*, (b) cliques associated with site *s*

function for the whole lattice is therefore

$$U(x) = -\frac{H}{k}\sum_{s\in S} - \frac{J}{k}\sum_{\{s,t\}} x(s)t(s)$$

Let the marginal distribution (π) of the magnetic lattices in the Ising model be

$$\pi(x) = \frac{1}{Z}e^{-U(x)},$$

which is usually called *Gibbs distribution*. By the celebrated Hammersley-Clifford theorem [71], if the distribution π for the ising model with respect to the topology $(S, \mathscr{N})$ satisfy positive condition, there is an unique corresponding Markov random field with local characteristics as follows:

$$\pi^s(x) = \frac{e^{-\frac{1}{k}(\sum_{t\in\mathscr{N}_S} x(t)+H)x(s)}}{e^{+\frac{1}{k}(\sum_{t\in\mathscr{N}_S} x(t)+H)} + e^{-\frac{1}{k}(\sum_{t\in\mathscr{N}_S} x(t)+H)}}.$$

The converse part of the theorem shows the for any MRF with a consistent set of conditional probabilities, there exists an equivalent Gibbs distribution expressed in terms of local potentials.

Appendix B

Appendix: Continuous-time finite horizon optimal control

In this section, we present a quick review of using the Pontrjagin Maximum Principle (PMP) to solve the finite horizon optimal control problem. The goal of the problem is to find the optimal control input $u^*(t) \in \mathscr{U}$, where $t \in [t_0, t_f]$, such that

$$\begin{aligned} \text{maximize } J &= \psi(q(t_f)) + \int_{t_0}^{t_f} L(q(t), u(t), t)dt \\ \text{subject to } \dot{q} &= f(q(t), u(t), t) \\ q(t_0) &= q_0 \end{aligned} \tag{B.1}$$

where the first term in objective function J is the terminal cost, and the second term is the running cost. To derive a general solution, one can use the method of calculus of variations. The first step is to augment the cost by adding the costate vector $\lambda(t)$

$$J = \psi(q(t_f)) + \int_{t_0}^{t_f} (L + \lambda^T (f - \dot{q}))dt \tag{B.2}$$

Along the optimal trajectory the variation of J should be vanished, which follows the fact that J is continuous in q, u, and t. The variation δJ can be expressed as

$$J = \psi_q \delta q(t_f) + \int_{t_0}^{t_f} (L_q \delta q + L_u \delta_u + \lambda^T f_q \delta q + \lambda^T f_u \delta u - \lambda^T \dot{q}))dt \tag{B.3}$$

The last term can be integrated by parts, namely

$$\int_{t_0}^{t_f} -\lambda^T \dot{q} dt = -\lambda^T (t_f) \delta q(t_f) + \lambda^T (t_0) \delta q(t_0) + \int_{t_0}^{t_f} \dot{\lambda}^T \delta q dt.$$

Then,

$$\delta J = \psi_q(q_{t_f})\delta q(t_f) - \lambda^T(t_f)\delta q(t_f) + \lambda^T(t_0)\delta q(t_0) + \int_{t_0}^{t_f}((L_u + \lambda^T f_u)\delta_u + (L_q + \lambda^T f_q + \dot{\lambda}^T)\delta q))dt \tag{B.4}$$

To make sure the variation of J vanishes, the following condition should be satisfied.

$$L_u + \lambda^T f_u = 0 \tag{B.5}$$

$$L_q + \lambda^T f_q + \dot{\lambda}^T = 0 \tag{B.6}$$

$$\psi_q(q_{t_f}) - \lambda^T(t_f) = 0 \tag{B.7}$$

In general, to solve the optimal control analytically from above conditions is difficult. One can used gradient methods to numerically compute the optimal control input. When the system is linear, and the objective function has quadratic form, the three conditions are boiled down to riccati equation. Efficient softwares can be found to solve the equation.

BIBLIOGRAPHY

[1] D. A. Schoenwald, "AUVs: In space, air, water, and on the ground," *IEEE Control Systems Magazine*, vol. 20, no. 6, pp. 15–18, 2000.

[2] M. H. Douglas and A. Patricia, "Reducing swarming theory to practice for uav control," in *Proceedings of IEEE Aerospace Conferenc*, 2004.

[3] R. Olfati-Saber and R. M. Murray, "Distributed cooperative control of multiple vehicle formations using structural potential functions," in *Proceedings of the 15th IFAC World Congress*, Barcelona, Spain, 2002.

[4] J. R. T. Lawton, R. W. Beard, and B. J. Young, "A decentralized approach to formation maneuvers," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 6, pp. 933–941, 2003.

[5] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[6] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Stable flocking of mobile agents, Part I: Fixed topology," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, 2003, pp. 2010–2015.

[7] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proceedings of the 40th IEEE Conference on Decision and Control*, Orlando, FL, 2001, pp. 2968–2973.

[8] P. Song and V. Kumar, "A potential field based approach to multi-robot manipulation," in *Proceedings of the IEEE International Conference on Robots and Automation*, Washington, DC, 2002, pp. 1217–1222.

[9] J. S. Baras, X. Tan, and P. Hovareshti, "Decentralized control of autonomous vehicles," in *Proceedings of the 42nd IEEE Conference on Decision and Control, Maui*, vol. 2, Maui, Hawaii, 2003, pp. 1532–1537.

[10] P. Ogren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic Control*, vol. 49, no. 8, pp. 1292–1302, 2004.

[11] D. H. Kim, H. O. Wang, G. Ye, and S. Shin, "Decentralized control of autonomous swarm systems using artificial potential functions: Analytical design guidelines," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, vol. 1, Atlantis, Paradise Island, Bahamas, 2004, pp. 159–164.

[12] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Sacramento, CA, 1991, pp. 1398–1404.

[13] R. Volpe and P. Khosla, "Manipulator control with superquadric artificial potential functions: Theory and experiments," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 6, pp. 1423–1436, 1990.

[14] J. Kim and P. Khosla, “Real-time obstacle avoidance using harmonic potential functions,” *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 338–349, 1992.

[15] J. Barraquand, B. Langlois, and J.-C. Latombe, “Numerical potential field techniques for robot path planning,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 2, pp. 224–241, 1992.

[16] C. Liu, M. H. A. Jr, H. Krishna, and L. S. Yong, “Virtual obstacle concept for local-minimum-recovery in potential-field based navigation,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, CA, 2000, pp. 983–988.

[17] X. Zou and J. Zhu, “Virtual local target method for avoiding local minimum in potential field based robot navigation,” *Journal of Zhejiang University Science*, vol. 4, no. 3, pp. 264–269, 2003.

[18] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions and automation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.

[19] R. Chellappa and A. Jain, *Markov Random Fields: Theory and Applications*. Boston: Academic Press, 1993.

[20] G. Winkler, *Image Analysis, Random Fields, and Dynamic Monte Carlo Methods : A Mathematical Introduction*. New York: Springer-Verlag, 1995.

[21] B. P. Carlin and T. A. Louis, *Bayes and Empirical Bayes Methods for Data Analysis*. Boca Raton, FL: Chapman and Hall/CRC, 2000.

[22] M. Medvedovic, K. Y. Young, and R. E. Bumgarner, “Bayesian mixture model based clustering of replicated microarray data,” *Bioinformatics*, vol. 20, no. 8, pp. 1222–1232, 2004.

[23] R. Diestel, *Graph Theory*, ser. Graduate Texts in Mathematics. New York: Springer-Verlag, 1997, vol. 173.

[24] W.C.Allee, *Animal Aggregations*. Chicago,IL: Univ. of Chicago Press, 1931.

[25] S. Camazine, J.-L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau, *Self-Organization in Biological Systems*. Princeton, NJ: Princeton University Press, 2001.

[26] J. K. Parrish and L. Edelstein-Keshet, “Complexity, pattern, and evolutionary trade-offs in animal aggregation,” *Science*, vol. 284, pp. 99–101, April 1999.

[27] J. K. Parrish, S. V. Viscido, and D. Grunbaum, “Self-organized fish schools: An examination of emergent properties,” *Biol. Bull.*, vol. 202, June 2002.

[28] W. D. Hamilton, “Geometry for the selfish herd,” *J. theor. Biol*, vol. 31, pp. 295–311, 1971.

[29] K. M. Passino, “Biomimicry of bacterial foraging for distributed optimization and control,” *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52–67, 2002.

[30] V. Gazi and K. M. Passino, “Stability analysis of social foraging swarms,” *IEEE Trans. on Systems, Man, and Cybernetics-Part B Cybernetics*, vol. 34, no. 1, pp. 539–557, 2004.

[31] D. Grunbaum and A. Okubo, “Modeling social animal aggregations,” *Frontiers in Theoretical Biology*, vol. 100, pp. 296–325, 1999.

[32] L. Edelstein-Keshet, “Mathematical models of swarming and social aggregation,” in *Proceedings of the 2001 International Symposium on Nonlinear Theory and its Applications, (NOLTA 2001) Miyagi, Japan*, 2001, pp. 159–164, invited lecture.

[33] A. Mogilner and L. Edelstein-Keshet, “A nonlocal model for a swarm,” *J. Math. Biol.*, vol. 38, pp. 534–570, 1999.

[34] C. M. Breder, “Equations descriptive of fish schools and other animal aggregations,” *Ecology*, vol. 35, pp. 361–370, 1954.

[35] A. Mogilner, L. Edelstein-Keshet, L. Bent, and A. Spiros, “Mutual interactions, potentials, and individual distance in a social aggregation,” *J. Math. Biol.*, vol. 47, pp. 353–389, 2003.

[36] C. W. Reynolds, “Flocks, herds, and schools: A distributed behavior model,” in *Proceedings of SIGGRAPH 87*, 1987.

[37] B. Schechter, “Birds of a feather,” *New Scientist‘*, pp. 30–33, Jan 1999.

[38] T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, and O.Shochet, “Novel type of phase transition in a system of self-driven particles,” *Phys. Rev. Lett.*, vol. 75, pp. 1226–1229, 1995.

[39] J. Toner and Y. Tu, “Flocks, herds, and schools: A quantitative theory of flocking,” *Phys. Rev. E*, vol. 58, 1998.

[40] A. Czirok and T. Vicsek, “Collective motion,” *Statistical Mechanics of Biocomplexity, Lecture Notes in Physics 527*, 1999.

[41] N. Shimoyama, K. Sugawa, T. Mizuguchi, Y. Hayakawa, and M. Sano, “Collective motion in a system of motile elements,” *Physical Review Letters*, vol. 76, 1996.

[42] A. Czirok, H. E. Stanley, and T. Vicsek, “Spontaneously ordered motion of selfpropelled partciles,” *Journal of Physics A: Mathematical, Nuclear and General*, vol. 30, 1997.

[43] G. Grégoire, H. Chaté, and Y. Tu, “Moving and staying together without a leader,” *Physica D*, vol. 181, 2003.

[44] E. M. Rauch, M. M. Millonas, and D. R. Chialvo, “Pattern formation and functionality in swarm models,” *Phys. Rev. A*, vol. 207, 1995.

[45] H. Levine, W.-J. Rappel, and I. Cohen, “Self-organization in systems of self-propelled particles,” *Phys. Rev. Lett.*, vol. 63, no. 1, 2001.

[46] R. W. B. W. Ren, “Consensus seeking in multiagent systems under dynamically changing interaction topologies,” *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 655–661, 2005.

[47] R. Olfati-Saber, “Flocking for multi-agent dynamic systems: Algorithms and theory,” June 2004, submitted to the IEEE Transactions on Automatic Control.

[48] S. Gueron and S. A. Levin, “The dynamics of group formation,” *Math. Biosci.*, vol. 128, pp. 243–264, 1995.

[49] O. Khatib, “Real time obstacle avoidance for manipulators and mobile robots,” *International Journal of Robotic Research*, vol. 5, no. 1, pp. 90–98, 1986.

[50] R. Shahidi, M. Shayman, and P. S. Krishnaprasad, “Mobile robot navigation using potential functions,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, Sacramento, CA, 1991, pp. 2047–2053.

[51] E. Rimon and D. E. Kodistschek, “Exact robot navigation using artificial potential functions,” *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.

[52] V. Gazi and K. M. Passino, “Stability analysis of swarms,” *IEEE Trans. Trans. on Automatic Control*, vol. 48, no. 4, pp. 692–697, 2003.

[53] R. Bachmayer and N. E. Leonard, “Vehicle networks for gradient descent in a sampled environment,” in *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, NV, 2002, pp. 112–117.

[54] S. S. Ge and Y. J. Cui, “New potential functions for mobile robot path planning,” *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 615–620, Jan 2000.

[55] D. P. Miller, “Multiple bahavior-controlled micro-robots for planetary surface missions,” in *Proceedings of IEEE Systems, Man and Cybernetics Conference, Studio City, CA*, 1990.

[56] L. Parker, “Design control laws for cooperative agent teams,” in *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, 1993.

[57] T. Balch and R. C. Arkin, “Behavior-based formation control for multi-robot teams,” *IEEE Transactions on Robotics and Automation*, vol. 14, 1998.

[58] R. Vidal, O. Shakernia, and S. Sastry, “Distributed formation control with omnidirectional vision based motion segmentation and visual servoing,” *Robotics and Automation Magazine, Special issue on Panoramic Robotics*, 2003.

[59] P. K. C. Wang and F. Y. Hadaegh, “Coordination and control of multiple microspacecraft moving in formation,” *J. Astronautical Sci*, vol. 44, no. 3, 1996.

[60] F. Y. Hadaegh, W.-M. Lu, and P. K. C. Wang, “Adaptive control of formation flying spacecraft for interferometry,” in *Proceedings of the International Federation Automatic Control*, 1998.

[61] Z. Jin and R. M. Murray, “Double-graph control strategy of multi-vehicle formations,” in *Proceedings of the 43rd IEEE Conference on Decision and Control, Paradise Island, Bahamas*, 2004.

[62] J. A. Fax and R. M. Murray, “Information flow and cooperative control of vehicle formations,” *IEEE Transaction on Automatic Control*, vol. 49, no. 9, 2004.

[63] R. W. Beard and W. Ren, “Virtual structure based spacecraft formation control with formation feedback,” in *Proceedings of the AIAA Guidance, Navigation, and Control Conference, Monterey, CA*, 2002.

[64] M. Egerstedt and X. Hu, “Formation constrained multi-agent control,” *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, 2001.

[65] E. W. Justh and P. S. Krishnaprasad, "Steering laws and continuum models for planar formations," in *Proceedings of the 42rd IEEE Conference on Decision and Control, Mauri, Hawaii*, 2003.

[66] M. Flint, M. Polycarpou, and E. F. Gaucherand, "Cooperative path-planning for autonomous vehicles using dynamic programming," in *IFAC 15th Triennial World Congress, Barcelona*, Spain, 2002, pp. 1694–1699.

[67] W. Li. and C. G. Cassandras, "Centralized and distributed cooperative receding horizon control of autonomous vehicle missions," *Journal of Mathematical and Computer Modelling*, 2004, accepted.

[68] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002.

[69] D. Chang, S. C. Shadden, J. E. Marsden, and R. Olfati-Saber, "Collision avoidance for multiple agent systems," in *Proceedings of on Decision and Control, Hawaii*, 2003.

[70] P. Bremaud, *Markov Chains, Gibbs Fields, Monte Carlo Simulation and Queues.* New York: Springer Verlag, 1999.

[71] ——, *Markov Chains, Gibbs Fields, Monte Carlo Simulation and Queues.* New York: Springer Verlag, 1999.

[72] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions and automation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.

[73] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, 4598, pp. 671–680, 1983. [Online]. Available: citeseer.ist.psu.edu/kirkpatrick83optimization.html

[74] V. Cerny, "A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, pp. 41–51, 1985.

[75] L. Ingber, "Simulated annealing: Practice versus theory," *Mathl. Comput. Modelling*, vol. 18, no. 11, pp. 29–57, 1993.

[76] W. Xi, X. Tan, and J. S. Baras, "Gibbs sampler-based path planning for autonomous vehicles: Convergence analysis," in *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, 2005.

[77] J. S. Baras and X. Tan, "Control of autonomous swarms using Gibbs sampling," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, 2004, pp. 4752–4757.

[78] W. Xi, X. Tan, and J. S. Baras, "A stochastic algorithm for self-organization of autonomous swarms," in *Proceedings of the 44th IEEE Conference on Decision and Control, Seville*, Seville, Spain, 2005.

[79] R. Azencott, *Simulated Annealing: Parlalleization Techniques*, R. Azencott, Ed. Willey Inter. Science, 1992.

[80] A. Trouve, "Massive parallelization of simulated annealing: A mathematical study," *Simulated Annealing: Parlalleization Techniques*, pp. 145–162, 1992.

[81] R. Horn and C. R. Johnson, *Matrix Analysis*. New York: Cambridge University Press, 1985.

[82] L. K. Grover, "Simulated annealing using approximate calculation," in *Progress in Computer Aided VLSI Design*, 1989.

[83] S. B. Gelfand and S. K. Mitter, "Simulated annealing with noisy or imprecise energy measurements," *Journal of Optimization Theory and Applications*, vol. 61, no. 1, pp. 49–62, 1989.

[84] ——, "Simulated annealing with noisy or imprecise energy measurements," *Algorithmica*, vol. 6, no. 3, pp. 419–436, 1991.

[85] D. R. Greening, "Simulated annealing with errors," Ph.D. dissertation, UNIVERSITY OF CALIFORNIA Los Angeles, 1995.

[86] J. C. Alexander and J. H. Maddocks, "Asymptotic stability and feedback stabilization," *Differential Geometric Control Theory,* 1989.

[87] A. Isidori, *Nonlinear Control Systems*, 3rd ed. New York: Springer-Verlag, 1995.

[88] L. E. Dubins, "On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, pp. 497–516, 1957.

[89] H. Sussmann, "The markov-dubins problem with angular acceleration control," in *Proceedings of the 36th Conference on Decision and Control, San Diego, CA*, Dec. 1997.

[90] J. A. Reeds and R. A. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Journal of Mathematics*, vol. 145, no. 2, 1990.

[91] P. Soueres and J. Laumond, "Shortest paths synthesis for a car-like robot," *IEEE Transaction on Automatic Control*, vol. 41, no. 5, pp. 672–668, 1996.

[92] X. N. Bui, P. Soueres, J. D. Boissonnat, and J. P. Laumond, "Shortest path synthesis for dubins nonholonomic robot," in *Proceedings of IEEE Int. Conf. on Robotics and Automation, San Diego, CA*, May 1994.

[93] A. Scheuer and T. Fraichard, "Continuous curvature path planning for car-like vehicles," in *Proceedings of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems, Grenoble, FR*, September 1997.

[94] A. Scheuer, "Suboptimal continuous-curvature path planning for non-holonomic robots," in *Proceedings of 11st Journes Jeunes Chercheurs en Robotique*, 1999, pp. 107–112.

[95] S. J. Qin and T. A. Badgewell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, 2003.

[96] G. J. Sutton and R. R. Bitmead, "Computational implementation of nmpc to nonlinear submarine," *Chemical Process Control*, vol. 26, 2000.

[97] W. Ren and R. Beard, “Trajectory tracking for unmanned air vehicles with velocity and heading rate constraints,” *IEEE Transactions on Control Systems Technology*, vol. 12, no. 5, 2004.

[98] F. M. Y. J. Kanayama, Y. Kimura and T. Noguchi, “A stable tracking control method for an autonomous mobile robot,” in *Proceedings of the IEEE Int. Conf. Robotics and Automation, Cincinnatti, OH*, 1990.

[99] P. Morin and C. Samson, “Time-varying exponential stabilization of chained systems based on a backstepping technique,” in *Proceedings of the IEEE Conf. on Decision and Control, Kobe, J*, 1996.

[100] T. Fukao, H. Nakagawa, and N. Adachi, “Adaptive tracking control of a nonholonomic mobile robot,” *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, 2000.

[101] D. H. Shim, H. J. Kim, H. Chung, and S. Sastry, “A flight control system for aerial robots: Algorithms and experiments,” in *Proceedings of the 15th IFAC World Congress on Automatic Control*, July 2002.