Cooperative Source Seeking and Level Curve Tracking for Multi-Agent Systems

Vom Promotionsausschuss der
Technischen Universität Hamburg-Harburg
zur Erlangung des akademischen Grades
Doktor-Ingenieur
genehmigte Dissertation

von

Esteban Emilio Rosero García

aus

Potosí, Kolumbien

2016

1. Gutachter:
Prof. Dr. Herbert Werner

2. Gutachter:
Prof. Dr. Karl-Heinz Zimmermann

Vorsitzende des Promotionsverfahrens:
Prof. Dr. Volker Turau

Tag der mündlichen Prüfung: 14.10.2016

To my family, my daughter Sara Sofía and Ximena

Abstract

The field of autonomous systems control is young, but operational experience is rapidly growing, making research on collaborative systems of great importance. Improving aerial robots in particular could be key in facing future environmental challenges or disaster situations by, for example, enabling information sharing and coordinated action by a groups of agents.

In this work, two main problems are addressed: the cooperative source seeking problem and the cooperative level curve tracking problem by a group of agents under undirected constrained communications. In the first case, agents should move towards a scalar field's source such as the highest temperature spot of a large scale fire or the highest concentration point of an oil spill. Agents ought to locate a quantifiable signal source anywhere in a scalar field, relying only on information gathered locally by each one of them, and having therefore to effectively cooperate. For this purpose, distributed control algorithms are proposed, enabling agents to drive towards an unknown scalar field's source. In the second case, when a source becomes inaccessible or approaching it results highly dangerous, a level curve tracking approach is proposed. Level curve tracking enables agents to move along a curve, alongside which concentration values stay constant. This procedure allows agents to still obtain the needed information. To solve the cooperative level curve tracking problem, distributed algorithms steering a group of agents along a desired scalar field's level curve are proposed.

The control algorithms presented are based on both a formation-control component (keeping agents in a desired geometric driving pattern) and a trajectory control component (steering agents in the source's direction or along the desired level curve). This control architecture takes into account that communications between agents are limited, that there are diverse agent dynamics and that each agent has to cooperatively compute a gradient in order to obtain direction. Agents are assumed to only have access to other agents' relative position information and to scalar field's concentration values at their own locations. Firstly, simple agent dynamics are considered and distributed navigation controllers for both single and double integrator models are designed. Secondly, complex dynamics are analysed extending the presented approach to general linear time-invariant (LTI) models and non-holonomic systems. Stability conditions are provided for single and double integrators, as well as LTI systems. The applied method is verified using formation flight simulation. Finally, non- holonomic systems are considered and a simple distributed controller avoiding obstacles or collisions between agents and steering agents towards the scalar field's maximum value is presented.

Acknowledgements

I would like to express my sincere gratitude to my advisor Prof. Dr. Herbert Werner for giving me the opportunity of being his student and for his invaluable support and guidance during my study and research time at the Institute of Control Systems of the Hamburg University of Technology. I am very glad of having taken part in his seminars which widened my academic view and enriched my understanding of the importance of an interdisciplinary view in our field.

I am also very grateful with the School of Electrical & Electronic Engineering of the Universidad del Valle, the German Academic Exchange Service (DAAD), and the Colombian Administrative Department of Science, Technology and Innovation (COLCIENCIAS) for giving me the financial support for my doctoral studies and with the first of them for giving me the time needed to complete these studies.

I wish to express my gratitude to Ximena for her constant support and help during my time in Germany. A special dedication to my daughter Sara Sofía who is the light in my life and a special thanks to my parents, Rafael and Emperatriz, and brothers, Javier and Patricia, for their faith and good energy during this time.

Nomenclature

Abbreviations

AAVs	Autonomous Aerial Vehicles
AUVs	Autonomous Underwater Vehicles
LTI	Linear Time-Invariant
MAS	Multi-Agent Systems
MIMO	Multiple Inputs Multiple Outputs
UAVs	Unmanned Air Vehicles
UUVs	Unmanned Underwater Vehicles

Symbols

$\bar{\hat{g}}(t)$	Average of the estimated gradient
$\bar{\mathcal{A}}$	Normalized adjacency matrix
$\bar{\mathcal{L}}$	Normalized Laplacian matrix
$\bar{a}_{ij} = \frac{1}{\lVert\mathcal{N}_i\rVert}$	Normalized elements of matrix $\bar{\mathcal{A}}$
$\bar{e}_g$	Average of the estimated gradient error vector
$\bar{e}_p$	Average of the position error vector
β	Tuning parameter
$\mathbf{1} = [1\ 1\ 1\ ...\ 1]^T$	Vector of ones
δ	Disagreement vector
$\Delta = \text{diag}(\mathcal{A} \cdot \mathbf{1})$	Degree matrix
$\Delta_{ii} = d_i = \sum_j a_{ij}$	Degree of agent i
$\dot{v}$	Linear acceleration vector
$\dot{v}_i = \ddot{r}_i$	Linear acceleration of agent i

$\eta(t) = x(t) - \mathbf{1}r(t)$	Error vector
$\gamma,\ \alpha$	Tuning parameters
$\hat{g}_i$	Estimated gradient
$\hat{r}(t)$	Average of the positions
$\hat{v}$	Average of the velocity
λ_1	First eigenvalue of matrix $\mathcal{L}$
λ_2	Second eigenvalue of matrix $\mathcal{L}$
$\lambda_i(A)$	i^{th} eigenvalue of matrix A
$\lambda_{max}(A)$	Maximum eigenvalue of matrix A
$\lambda_{min}(A)$	Minimum eigenvalue of matrix A
$\Vert\mathcal{N}_i\Vert$	cardinality of the set of neighbours $\mathcal{N}_i$
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^p$	Set of real numbers in dimension p
$\mathbb{R}^{N\times N}$	Set of real $N \times N$ matrix
$\mathbb{R}^N$	Set of real matrix with dimension N
$\mathcal{A}$	Adjacency matrix
$\mathcal{E}$	Set of edges
$\mathcal{G}$	Graph
$\mathcal{L}$	Laplacian matrix
$\mathcal{M}$	Projection matrix
$\mathcal{N}_i$	Set of neighbours of agent i
$\mathcal{V}$	Set of nodes
μ	Bound of the estimated gradient
μ	Bound of the estimated gradient
$\nabla\psi(r_i)$	True gradient at position r_i
$\nabla^2\psi(r_i)$	Hessian matrix at position r_i
ν	Bounded rate
ω	Angular velocity vector
ω_i	Angular velocity of agent i
ω_{ci}	Bandwidth
$\otimes$	Kronecker product
ϕ_i	Orientation of agent i respects to r_x
$\phi_{di} = \text{Atan2}\,(E_{yi}, E_{xi})$	Desired orientation angle for agent i

$\psi(r)$	Smooth scalar function
$\psi(r_i)$	Scalar function value at position r_i
$\psi(r_s) = L_\psi$	Maximum value of the scalar function
$\\|\cdot\\|$	Euclidian norm for vectors
$\Sigma = \text{diag}(\Sigma_1, \ldots, \Sigma_N)$	Covariance matrix of $n(t)$
θ	Tuning parameter
$A \succ 0$	Positive definite matrix
$A \succeq 0$	Positive semidefinite matrix
$A,\ B,\ C,\ D$	State, input, output, feed-through matrices
A^{-1}	Inverse of matrix A
A^T	Transpose of matrix A
a_{ij}	Elements of matrix $\mathcal{A}$
d_a	Distance function
e_p	Position error vector
$e_{\nabla i}$	Estimated gradient error
$e_{\phi i} = \phi_i - \phi_{di}$	Orientation error of agent i
e_ϕ	Orientation error vector
I	Identity matrix
i	Index of agents, $i = 1, \ldots, N$
j	Index for the neighbours of agent i
K	Control matrix
k_a, k_o	Tuning parameters
k_T, k_F	Scalar tuning parameters
l_a	Maximum distance between agents in the formation
L_H	Bound of the Hessian matrix
N	Number of agents
$n(t)$	Zero-mean Gaussian noise vector
$n_i(t)$	Zero-mean Gaussian noise
p	State's dimension
r	Position vector
r^*	Position of the equilibrium point
r_F	Desired formation vector
$r_i = [r_{xi}\ r_{yi}\ r_{zi}]^T$	Position of agent i

$r_s = [r_{xs}\ r_{ys}\ r_{zs}]^T$	Position of the maximum value
r_{xi}	x-position of agent i
r_{yi}	y-position of agent i
r_{zi}	z-position of agent i
S	Detection region's radius
s	Avoidance region's radius
u	Control law vector
u_i	Control law for agent i
v	Linear velocity vector
$v_i = \dot{r}_i$	Linear velocity of agent i
x_i	State vector of agent i

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Multi-Agent Systems

Today, advances in exploration and rescue technologies become more relevant than ever before. Besides being a very important field of research, automatic control of robots, unmanned vehicles, and any kind of autonomous self-operating devices is a substantial issue for improving exploration and survival technologies. Equipped with sensors, automated agents can be used to operate in hazardous environments with little resources available. Localizing and monitoring disaster zones such as fire spots, chemical spill belts, etc., searching for survivors, or conducting environmental studies are examples of future missions. Agents capable of autonomous operation and prepared for information measurement and/or intervention should be able to completely replace humans in environments too dangerous or precarious. This kind of agents are still being developed and enhanced. Agents' size is becoming highly adaptable nowadays, even to a very small scale. These agents can be provided with wireless communications and the fastest processors. Costs of such devices are steadily decreasing and their integrated equipment is more sophisticated than ever.

For the purposes of this work, an agent is a dynamic system, capable of autonomous action in its environment, which can be directed towards attaining certain goals. It is equipped with sensors, processing units and actuators. Autonomous means that it is able to operate without human intervention. Furthermore, each agent is able to fulfill different kinds of functions: sensing, processing, communicating and executing.

A group of such agents can perform tasks more efficiently than a single agent and accomplish missions not executable by a single one. Individual agents have restricted capabilities such as temporal or spatial separation, fuel and time constraints, and additionally, limited sensor range and accuracy. Furthermore, due to restricted communication bandwidth, communication latency and limited computation and memory, their manoeuvres are more costly and less effective compared to them of collective agents.

Agents operating in groups are structured as systems of agents. A multi-agent system (MAS) is a system of autonomous agents interacting with each other and their environment in order to perform a common task; it can be modelled by differential equations. To achieve a common objective, agents need to gather and exchange information through information flow channels. However, sensors can collect huge amounts of data turning the information flow into a challenge for processing units in the individual system. Initial approaches solved this problem by means of a central entity in the group responsible for processing all the gathered information and steering the whole system.

Centralized control means that there is one processor steering all agents from a central point. It requires high computational power and its implementation is expensive due to computational costs and an eventual communication overhead. As a result, system's performance decays due to communication delays. With the development of more complex systems including a larger number of agents communication costs turned too high; agents required considerable power to operate, and energy efficiency decreased. Furthermore, system's reliability was not guaranteed due to the single point of failure problem. A failure in the central entity could cause the whole system to collapse, or in the best case, data would be lost. A central processing unit is highly vulnerable to internal failure or external attacks. In this case the complete network would become inoperative.

This risk can be avoided with an alternative approach: information can be processed inside each agent creating communication networks with a more evenly distributed flow and computation of information. Such networks will be called distributed networks for the purposes of this work. Distributed control means that each agent senses independently, gathers information from its neighbours, processes all available local information, and is then able to compute all this data in order to coordinate its movements with other agents and follow the group. Although each agent has only topical access to communications, limited computational resources and restricted sensing capabilities, all together the capacity of the system improves significantly being not only an addition of the individual agent's processing force but a multiplication of its functions.

In order to develop challenging applications as distributed dynamical systems, science and engineering have put their attention on biological systems, aiming to gain understanding of collective behaviour in animal groups. A fish school, a flock of birds, a swarm of bees are all examples providing inspiration to engineers. Animals in groups have developed cooperative strategies that allow them to function as high performance networked dynamical systems. They employ distributed feedback strategies with astounding success despite their sensing, information processing and execution limitations. As a group, they have to constantly adapt their behaviour to uncertain and changing environments.

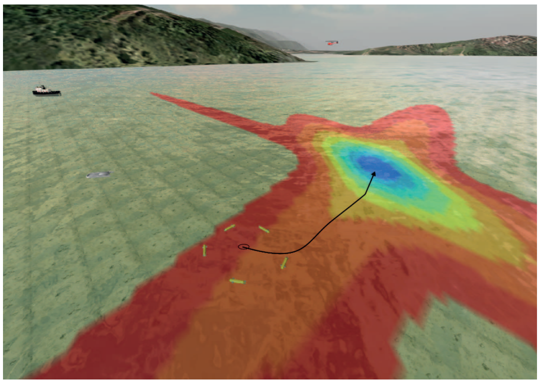
Biological systems are models of cooperative behaviour aiming to perform tasks in a coordinated fashion. Coordination involves teamwork; agents collaborate to acquire, fuse and share information, in order to overcome environmental constraints and achieve a global team goal. The ability of animals to communicate and coordinate their motion with neighbours leads to complex self-organized collective behaviour. A group of ants doing collaborative work is able to locate food sources, transport them to feed the colony and defend its territory with astounding efficiency. A single ant is neither capable of remembering the precise location of a food source, nor of carrying huge amounts of food (Okubo, 1986).

In an effort to develop distributed mobile agent systems resembling their natural counterparts, engineers have been experimenting with mobile sensor networks trying, for example, to implement flocking applications. The goal has been to create self-organized networks capable of coordinated group behaviour (Olfati-Saber, 2006). For this purpose, heuristic rules were introduced by (Reynolds, 1987) in order to explain any form of collective behaviour of a large number of individuals with a common goal. These rules are known as cohesion, separation, and alignment. Cohesion means the attempt to stay close to the neighbours, separation means avoiding collisions with neighbours, and alignment means the attempt to match velocity with neighbour agents.

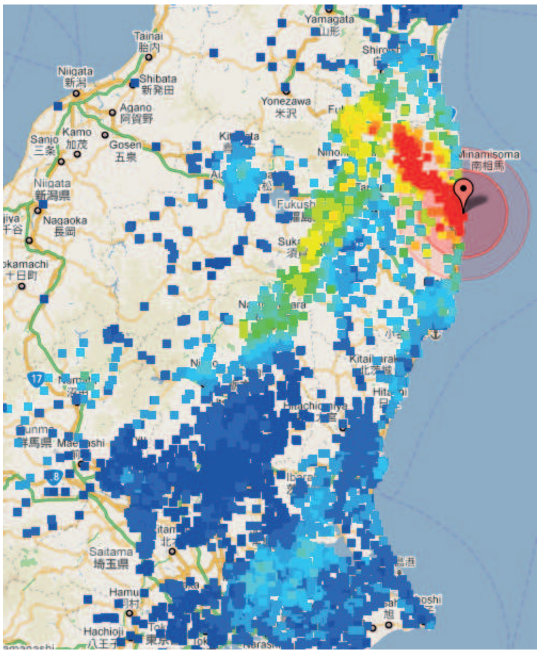
These principles have been used to design and implement first prototypes of autonomous multi-agent systems such as ground vehicles, unmanned air vehicles (UAVs), unmanned underwater vehicles (UUVs), robot mobile systems, or sensor networks (Fax and Murray, 2004; Oh and Ahn, 2010; Olfati-Saber et al., 2007; Olfati-Saber and Murray, 2004; Ren et al., 2007) to name just a few.

Such cooperative networks of agents act as “an intelligent array of sensors” (Ogren et al., 2004). In some sense they can operate better than its natural models because of an improved capacity for rapidly adapting to the environment (e.g. regulating inter vehicle spacing more efficiently than fish, which need additional space for mating behaviour). Sensors can be set up to improve performance and optimize detection and value measurement. Such a system offers many advantages and overcomes the dangers of individual agent failure or changes in the number of agents.

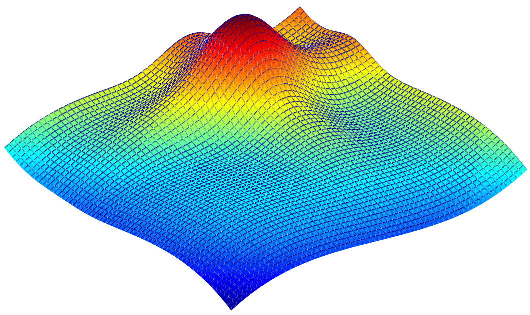
Cooperative systems can be characterized as a number of interconnected decision-making components with limited processing capabilities, locally sensing information, and limited communication range, which seek and achieve a collective objective (Shamma, 2008). The main features of cooperative systems are: distributed information flow, restriction in information exchange between agents keeping this exchange in a local range, and complexity of the resulting control dynamics. Distribution of information means that every agent needs to share information with its neighbour agents in order to coordinate with the group and agree on a common goal. The information exchange is local due to limitations in the communication bandwidth and the sensor range. Complex dynamics refer to a large number of interacting dynamic agents and the difficulties of dealing with analysis and synthesis of controllers, each one with partial and/or overlapping information. Thus, the main challenge of cooperative systems is to achieve desirable collective behaviour. In the following chapters this collective behaviour will be called consensus. For the purpose of achieving consensus, distributed control algorithms for individual agents must be designed, algorithms capable of dealing with limited information exchange and dynamic interaction topology.



(a) Oil spill [1]



(b) Fukushima's environment radiation levels after nuclear disaster [2]



(c) Toxic cloud

Figure 1.1: Scalar fields

In this work, two important issues in the field of cooperative control of multi-agent systems are considered: the cooperative source seeking problem and the cooperative level curve tracking problem. The problem of source seeking is to locate a quantifiable signal source somewhere in a scalar field using several and effectively cooperating agents. A scalar field represents the changing strength of a spatially distributed signal (scalar signal) and each point in a given area is associated with a local measurement of this signal. For example, oil spilled in the sea generates a scalar field of oil concentration values (Fig. 1.1a). Similarly, radiation levels after nuclear disaster like in Fukushima (Fig. 1.1b), or a toxic

[1]Picture taken from FeedNetBack project
[2]Picture taken from seaandskyjp.wordpress.com

substance cloud moving in space are phenomena that can be represented as scalar fields (Fig. 1.1c). When a scalar field represents temperature values distributed over an area, the source is the point of maximum heat and the signal is given by the actual temperature measurement in each point of this area.

The scalar field's source can be fixed or it can be moving in space, e.g., when the toxic substance is moving by the effect of wind. A scalar field can be expanding or contracting, for example, when oil spill is spreading by the effect of water streams. Furthermore, a scalar field can have either a symmetric or an irregular pattern, without affecting our approach. For our purposes, the signal strength is assumed to decay as the sensor moves away from the source's position. Finally, the source's position is equal to the maximum value in a scalar field's distribution.

The problem of level curve tracking arises in the cases when the source is inaccessible or approaching it is highly dangerous or unreasonable. Level curve tracking enables agents to keep moving in a constant concentration value or range near to the source, allowing them to obtain the information needed. Any scalar field can be also represented by a contour of level curves as shown in Fig. 1.2. The level curve can vary in space and time depending on if the scalar field is moving, expanding or contracting.

To solve the cooperative source seeking problem, distributed algorithms to drive a group of agents towards the unknown moving scalar field's source or unknown fixed scalar field's source are proposed in this work. Agents have access to the signal's position and value at its own location, only. The distributed control algorithms here presented are based on both formation control and trajectory control. The formation controller part maintains the agents in a desired geometric formation and the trajectory controller part drives the agents to the scalar field's source. Based on a distributed gradient estimation in each agent, the direction to be followed by the agents is calculated. The so estimated gradient can also be used to track the direction followed by rapid and maximum changes in the measured values, e.g., if the scalar field represents spatially distributed air temperature, the gradient can reveal the direction in which temperature is increasing most rapidly.

To solve the cooperative level curve tracking problem same assumptions as in the source seeking problem are made. A distributed algorithm to steer a group of agents, at constant velocity, tracking a desired level curve of the scalar field is proposed. Information of agents' relative positions and scalar field's concentration values is used in the algorithm. In the proposed methods, information of an estimated Hessian matrix is not required.

1.2 Current State of Research

The source seeking problem has already been addressed in the field of automated control. In the last decade, several source seeking algorithms have been developed

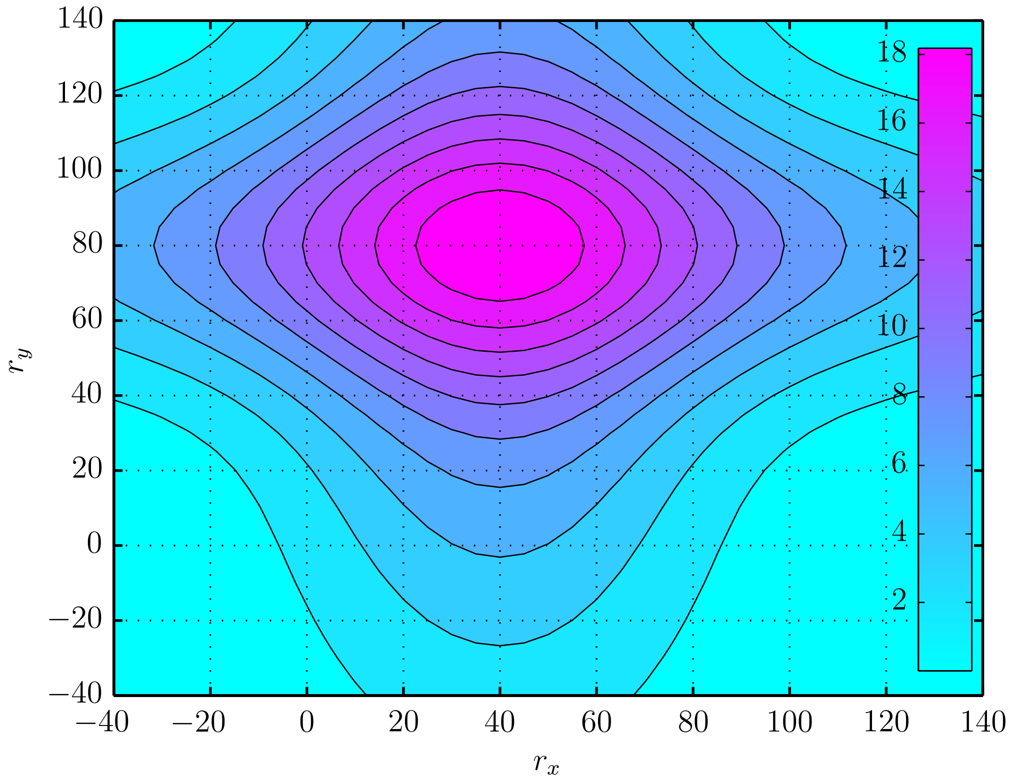


Figure 1.2: Contour of the scalar field

to guide one or more mobile agents towards the signal source. There are different approaches to deal with the source seeking problem in literature. They can be classified depending on the number of agents considered: approaches for only one agent or approaches for formations of multiple agents.

Firstly, in the case of source seeking concepts for only one agent, some researchers have focused on developing exploration missions in which measurements are done while the agent changes its position in a regular sequence over time (Azuma et al., 2012; Cochran and Krstic, 2009; Cochran et al., 2009; Liu and Krstic, 2010; Matveev et al., 2011; Mayhew et al., 2008; Stankovic and an M. Stipanovic, 2010; Zhang et al., 2007). Further, for an isolated non-holonomic agent, an angular velocity controller has been proposed in order to locate the maximum of the scalar field (Cochran and Krstic, 2009). In (Zhang et al., 2007) a periodic forward-backward movement of the unicycle is employed and the forward velocity is tunable to move the agent towards the source. An extension of (Cochran and Krstic, 2009) to 3-D source seeking is proposed in (Cochran et al., 2009). In (Matveev et al., 2011), a sliding mode navigation strategy has been proposed to drive an agent to the maximum of the scalar field. A stochastic source seeking approach for a nonholonomic mobile agent is presented in (Azuma et al., 2012). The main drawback of these approaches is that the agent must travel long distances to compute the gradient, sometimes using special manoeuvres such as sinusoidal inputs. In (Fabbiano et al., 2014) the problem of one agent with multiple sensors localizing the source of a diffusion process is considered. The solution is based on gradient computation and higher-order derivatives such as the Hessian matrix from Poisson integrals.

Secondly, in the case of multiple agents, a number of algorithms to improve mission performance have been developed (Biyik and Arcak, 2007; Brinon-Arranz et al., 2011; Cortes, 2005; Ghods and Krstic, 2010; Li and Guo, 2012; Moore and Canudas-de Wit, 2010; Ogren et al., 2004; Zhang and Leonard, 2010). Most of them depend on all-to-all communication between agents and on the formation's center of mass in order to estimate a unified gradient and design a convergence-achieving control law. In (Ogren et al., 2004) the source seeking problem is solved by means of decoupling the control strategy into formation maintenance and leader following. In this case, the coordination framework uses virtual bodies and artificial potentials. The measured field's gradient is approximated at the virtual body's position using centralized computation. A cooperative Kalman filter to estimate the gradient at the center of the formation is designed in (Zhang and Leonard, 2010). In (Biyik and Arcak, 2007), a gradient climbing method for an agents' formation to be steered towards the maximum of the scalar field is addressed. Here a leader is required and the gradient is estimated by the leader. The work presented in (Brinon-Arranz et al., 2011) solves the source seeking problem for a group of agents through distributing them uniformly in a fixed circular formation. To be able to estimate the gradient in the center of mass, agents rotate around this center, which is assumed as fixed and known to all agents. In (Fabbiano et al., 2014) a distributed source localization with no position information is presented. The authors suppose that the network's communications are described by a time-invariant ring-topology and the gradient is estimated based on a distributed implementation of Poisson integral formula. In (Li and Guo, 2012; Li et al., 2014) a control law combining both all-to-all and limited communications is presented. This algorithm is too complex and requires the center of mass's information.

In (Zhu et al., 2013), cooperation of multiple UAVs is carried out by adopting a leader-follower formation strategy. Scalar field's gradient is estimated at the leader UAVs location based on the measurements of all UAVs. In (Zhu et al., 2014), an extension of (Zhu et al., 2013) to locate an unknown moving scalar field using a leader-follower formation strategy is presented.

Formation control has been studied by (Borrelli and Keviczky, 2008; Fax and Murray, 2004; Massioni and Verhaegen, 2009; Popov and Werner, 2009), among many others. In (Fax and Murray, 2004), the control problem of an N identical agents' formation under unknown but fixed topologies is considered. Robust stability condition for the formation is derived, being equivalent to the same condition on a single agent. Using the framework presented in (Fax and Murray, 2004), a distributed Linear Quadratic Regulator (LQR) design for a formation is considered in (Borrelli and Keviczky, 2008). A feedback strategy based on a decomposition method and linear matrix inequalities is proposed in (Massioni and Verhaegen, 2009). In (Popov and Werner, 2009), using the framework of (Fax and Murray, 2004) and employing graph theory results, authors convert the formation stability problem into a robust control problem for a single agent. This way, performance requirements' management turns easier, and stability for fixed and varying topologies with communication delays is guaranteed. This result is applied in (Pilz et al., 2009).

Furthermore, (Ahmadi et al., 2015; Mendez et al., 2015; Mendez and Werner, 2014), have considered the topic of formation control for non-holonomic agents. In (Mendez and Werner, 2014), non-holonomic agents are represented as linear parameter-varying (LPV) models. A leader-follower configuration is analysed allowing the group of agents to track the leader's path, maintain the formation and follow a reference trajectory. In (Mendez et al., 2015), a distributed controller for a group of agents with non-holonomic dynamics using a directed switching communication topology and local information is considered. The controller's design is developed within the framework of linear parameter varying and linear fractional transformation (LPV-LFT) control. In (Ahmadi et al., 2015), a distributed controller for non-holonomic agents maintaining the desired formation and avoiding collisions between agents and obstacles is considered. In the last study only local information and undirected communication topology are used.

The problem of level curve tracking has been addressed by (Ogren et al., 2004; Zhang et al., 2007; Zhang and Leonard, 2010). These solutions exhibit several weaknesses such as the network requiring special shapes to estimate the gradient and the Hessian matrix of the formation's center of mass. They also assume that all agents know the center of mass' information. In (Williams and Sukhatme, 2012), a probabilistic method for spatial process mapping by a distributed muti-agent system is considered. They introduce a coordinated level curve tracking algorithm for adaptive sampling. Potential functions to avoid collision between agents are included. The drawback of such an approximation is that both the Hessian matrix and the gradient at the formation's center of mass must be estimated to be able to apply their control law.

1.3 Communication Graphs

Agents can share information with their surrounding peers through communication channels or by sensing. The information exchange inside a network is represented by a communication topology. Graphs are natural abstractions that provide an appropriate representation of how information is shared between agents in a network. The network's topology can be fixed (time-invariant) or switching (time-varying). The graph-based abstraction contains descriptions of the network's topology in terms of objects referred to as nodes and edges. The following section summarizes the graph theory concepts (Mesbahi and Egerstedt, 2010) which will be used in the following sections.

In the following, $\|\cdot\|$ denotes the Euclidian norm for vectors and the induced 2-norm for matrices. The superscript T means the transpose for real matrices and $\mathbb{R}^p$ denotes the p-dimensional Euclidian case. For any symmetric square matrix, $A \in \mathbb{R}^{m\times m}$ is the set of $m \times m$ real matrices. The minimum and maximum eigenvalues of a symmetric matrix A are denoted by $\lambda_{min}(A)$ and $\lambda_{max}(A)$ respectively. $A \succ 0$ denotes a positive definite matrix and $A \succeq 0$ denotes a positive semidefinite matrix. The matrix A^{-1} denotes the inverse of a square matrix A.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph that models the interaction among agents where $\mathcal{V} = \{1, ..., N\}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. Each node represents an agent and each edge corresponds to an information exchange channel. An edge $(i, j) \in \mathcal{E}$ indicates that the agent i and j exchange information.

Let $\mathcal{N}_i = \{j \in \mathcal{V} : a_{ij} \neq 0\}$ denote the set of neighbours of node i. The adjacency matrix $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N\times N}$ of a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with N nodes specifies the interconnection topology of the network. Here $a_{ij} = 1$ if $(i, j) \in \mathcal{E}$, else $a_{ij} = 0$. Note that here $\mathcal{A}$ is symmetric.

The Laplacian matrix $\mathcal{L}$ of the graph $\mathcal{G}$ is defined as $\mathcal{L} = \Delta - \mathcal{A}$, where Δ =diag($\mathcal{A} \cdot \mathbf{1}$) is a diagonal matrix with the agents' degrees on its diagonal, i.e., $\Delta_{ii} = d_i = \sum_j a_{ij}$. Here $\mathbf{1} = [1\ 1\ 1\ ...\ 1]^T \in \mathbb{R}^N$ denotes the vector of ones which is an eigenvector of $\mathcal{L}$ corresponding to $\lambda_1 = 0$, i.e., $\mathcal{L} \cdot \mathbf{1} = 0$. A projection matrix is defined as $\mathcal{M} = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T \in \mathbb{R}^{N\times N}$ and $\mathcal{M}$ satisfies $\mathcal{M}\mathbf{1} = 0$.

For an undirected graph, λ_2 is the algebraic connectivity, which is positive ($\lambda_2 > 0$) if and only if the undirected graph is connected. The second smallest eigenvalue λ_2 of $\mathcal{L}$ determines the algorithm's speed of convergence. $|\mathcal{N}_i|$ indicates the cardinality of the set of neighbours $\mathcal{N}_i$ and $\otimes$ denotes the Kronecker product. I denotes the identity matrix of appropriate dimension.

Here the normalized adjacency matrix used by (Fax and Murray, 2004) and (Popov and Werner, 2012) is used, where $\bar{a}_{ij} = \frac{1}{|\mathcal{N}_i|}$ if $(i, j) \in \mathcal{E}$, else $\bar{a}_{ij} = 0$. The normalized Laplacian matrix $\bar{\mathcal{L}}$ of the graph $\mathcal{G}$ is defined as $\bar{\mathcal{L}} = I - \bar{\mathcal{A}}$.

The covariance matrix of the noise signal $n(t)$ is assumed to be diagonal $\Sigma = \text{diag}(\Sigma_1, \dots, \Sigma_N)$, i.e., $n_i(t)$ and $n_j(t)$ are uncorrelated.

1.4 Multi-Agent Dynamics

Different models have been used to represent multi-agent system dynamics. In this thesis only continuous-time models will be treated. Multi agent systems consist of N dynamic agents, labelled as $i = 1, \dots, N$, interconnected via relative information exchange links.

In the next sections, the agent dynamics used in this thesis will be described. For simple agent dynamics, single and double integrators are used. For more complex linear dynamics, a Linear Time-Invariant (LTI) model is used; this representation also includes single and double integrator agent dynamics. For nonlinear dynamics, the simplest non-holonomic agent dynamics are used.

1.4.1 Single Integrator Models

The single integrator is a particular case of a kinematic linear model that will be used and discussed in Chapter 4. The dynamics of each agent i can be represented by

$$\dot{r}_i(t) = u_i(t), \tag{1.1}$$

where the position state of agent i is denoted by $r_i(t) \in \mathbb{R}^p$, p is the state's dimension and the control input of the system is denoted by $u_i(t) \in \mathbb{R}^p$.

A consensus protocol is proposed in (Olfati-Saber and Murray, 2004; Ren and Atkins, 2007) as

$$u_i(t) = -\sum_{j \in \mathcal{N}_i} a_{ij} \left(r_i(t) - r_j(t) \right), \tag{1.2}$$

where a_{ij} denotes the adjacency matrix $\mathcal{A}$'s elements of the information exchange topology and $\mathcal{N}_i$ denotes agent i's set of neighbours.

1.4.2 Double Integrator Models

Several vehicle dynamics can be represented by double integrator models which can be controlled using acceleration as an input. These models will be discussed in Chapters 4 and 7. For each agent i, the dynamics are defined as

$$\begin{aligned} \dot{r}_i(t) &= v_i(t), \\ \dot{v}_i(t) &= u_i(t), \end{aligned} \tag{1.3}$$

where $v_i(t)$ is the velocity state. When agents are modelled by single integrator dynamics, these agents achieve agreement with regard to their position signals. When agents have to reach agreement regarding both their position and their velocity signals, a double integrator model representation is more suitable.

For some purposes such as formation stabilization applications, it might be desirable that $v \to 0$ when $t \to \infty$. A second order-consensus protocol (Ren and Atkins, 2007) is given by

$$u_i(t) = -\sum_{j \in \mathcal{N}_i} a_{ij} \left[(r_i(t) - r_j(t)) + \gamma (v_i(t) - v_j(t)) \right] - \alpha v_i(t), \tag{1.4}$$

where $\alpha > 0$, $\gamma > 0$. Using this protocol agents achieve consensus on their relative position at zero velocity. When $\alpha = 0$, agents achieve consensus, but their velocity is not zero, $v \neq 0$. Note that consensus protocols 1.2 and 1.4 are distributed, meaning that each agent needs only information from its local neighbours.

1.4.3 Linear Time-Invariant Models

Some agent dynamics can be represented using a more general form such as a continuous linear time-invariant (LTI) system. LTI systems include single integrator, double integrator and higher-order integrator dynamics as special cases. Several types of electromechanical systems like robots, mobile agents or mechatronic systems can be modelled as LTI systems. A general Linear Time-Invariant model for each agent i can be represented as

$$\begin{aligned} \dot{x}_i(t) &= Ax_i(t) + Bu_i(t), \\ r_i(t) &= Cx_i(t) + Du_i(t), \end{aligned} \tag{1.5}$$

where $x_i \in \mathbb{R}^n$ denotes the state vector, $r_i \in \mathbb{R}^q$ denotes the output vector, $u_i \in \mathbb{R}^p$ denotes the control vector, $A \in \mathbb{R}^{n\times n}$ denotes the state matrix, $B \in \mathbb{R}^{n\times p}$ denotes the input matrix, $C \in \mathbb{R}^{q\times n}$ denotes the output matrix and $D \in \mathbb{R}^{q\times p}$ denotes the feed-through matrix, for each agent i.

A consensus algorithm is given by

$$u_i(t) = -K \sum_{j\in\mathcal{N}_i} a_{ij} \left(r_i(t) - r_j(t)\right), \tag{1.6}$$

where K is a control matrix.

1.4.4 Unicycle Kinematic Models

Many types of mobile agents like Autonomous Underwater vehicles (AUVs) or Unmanned Aerial Vehicles (UAVs) can be represented using a kinematic model equivalent to a unicycle. These models cover, in a realistic way, the nonlinear dynamics of the simplest non-holonomic constraints, i.e., the agent wheels roll without slipping. These models' constraints reduce the instantaneous movements that one agent can perform. The unicycle dynamics of each agent i in the group is given by

$$\begin{bmatrix} \dot{r}_{xi}(t) \\ \dot{r}_{yi}(t) \\ \dot{\phi}_i(t) \end{bmatrix} = \begin{bmatrix} \cos(\phi_i) & 0 \\ \sin(\phi_i) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i(t) \\ \omega_i(t) \end{bmatrix}, \tag{1.7}$$

where r_{xi} and r_{yi} are the Cartesian coordinates, $\phi_i \in [0, 2\pi)$ the orientation of the agent with respect to the r_x axis, v_i is the linear velocity input, and ω_i is the angular velocity input.

1.5 Problem Description and Approach

As initially explained, the problem of cooperative source seeking and level curve tracking by a formation of N mobile agents under undirected constrained communication is addressed in this thesis. It is assumed that agents are equipped with position and signal strength measurement sensors. Task of all agents is to find equilibrium around the scalar field's source or to move along a defined level curve inside the scalar field. Agents exchange information with neighbouring agents through a communication network. For every agent, and based on relative output information between each agent and its neighbours, an unweighted gradient, a weighted gradient and a gradient direction are estimated.

1.5.1 Assumptions about Formation and Estimated Gradient

Let $\bar{e}_\nabla$ be an average vector of the estimated gradient's error, e_0 be a bounded value of the estimated gradient's error and $e_{\phi i}$ be the orientation error. In order to estimate the gradient in each agent, assumptions 1 and 2 have been made:

Assumption 1. *Graph $\mathcal{G}$ is undirected and connected. It is assumed that the formation satisfies $|\mathcal{N}_i| \geq p$, $N \geq p+1$ and the agent i and their neighbours are not collinear, i.e., each agent has at least p neighbours and the minimum number of agents in the space is $p+1$.*

Assumption 2. *The scalar field has an isolated global maximum. The estimated gradient is bounded by $\|\hat{g}_i(r_i)\| \leq \mu$, i.e., the estimated gradient will not be infinite for any time t. The average of the estimated gradient's error is bounded by $\|\bar{e}_\nabla\| \leq e_0$.*

Assumptions 1 and 2 are discussed in Chapter 2.

In order to solve the source seeking problem and level curve tracking problem for agents modelled as LTI systems, assumption 3 has been made:

Assumption 3. *It is assumed that the state matrix A_p of each agent is marginally stable.*

In order to solve the source seeking problem for non-holonomic agents, assumption 4 has been made:

Assumption 4. *Agents' trajectory is smooth and satisfies*

$$e_{\phi 1} = e_{\phi 2} = \cdots = e_{\phi N} \neq \frac{\pi}{2}.$$

1.5.2 Assumptions about a Scalar Field Scenario

Consider a scenario that can be modelled as a scalar field such as concentration levels of radiation, temperature, pressure or a toxic substance. Let p be the dimension of the space in which the agents move ($p = 1, 2,$ or 3); let $r \in \mathbb{R}^p$ be the position of a single agent in the dimension p; let $\psi(r) : \mathbb{R}^p \to \mathbb{R}$ be an unknown smooth scalar function that is twice continuously differentiable and has a maximum $\psi(r_s) = L_\psi$ at $r_s = [r_{xs}\ r_{ys}\ r_{zs}]^T$, with $\nabla\psi(r_s) = 0$. Also consider that the Hessian $\nabla^2\psi(r_i)$ is bounded by $\|\nabla^2\psi(r_i)\| \leq L_H$. In this thesis following assumptions have been made:

- The scalar field's shape is not known by the agents.
- Agents have access only to the values of the field at its own location.
- The scalar field's source is fixed in space. When level curve tracking is analysed, for simulation purposes, the scalar field is assumed to be expanding or contracting as a function of time.
- Communication constraints such as packet loss and time delays are not considered.

1.5.3 Problem Statement

Consider multi-agent systems consisting of N agents moving in the space $\mathbb{R}^p$ with a communication graph $\mathcal{G}$. The following two problems are considered in this thesis:

- **Source seeking problem:** For a given N agents formation with undirected communication topology, find a distributed control law $u_i(t)$ that allows the formation's center to reach equilibrium in the neighbourhood of the scalar field's maximum.

  This first problem will be addressed in Chapters 4, 5 and 6.

- **Level curve tracking problem:** For a given N agents formation with undirected communication topology, find a distributed control law $u_i(t)$ that allows the formation to move along a given level curve inside an unknown scalar field.

  This second problem will be addressed in Chapter 7.

A scalar field is a model characterizing the spatial distribution of a scalar magnitude (i.e. concentration measurements) that allows analysis of the spatial presence of this given magnitude by associating a value with each point in space. Such a scalar field can contain several maxima and minima, spread through a given area. For this study, a scalar field containing only one maximum is assumed.

The signal based on which measurements are conducted and computed for each point in space can be impaired by noise or by varying concentration values. In this work both pure signals and signals affected by noise are considered. Signals affected by noise are given specific attention in Chapter 3.

Depending on the real environmental conditions to which this model should be applied, there will be situations when the scalar field's source is stable (e.g. lava emission due to volcanic eruption) and other situations, when the scalar field's source will be moving at constant or variable speed (e.g. fire), or expanding or contracting (e.g. oil spill). Taking this circumstance in account, at the end of this work, in the comments to simulation results, some conclusions will be presented about a possible extension of these results to the case of a scalar field's source moving at constant velocity.

In order to address the two problems mentioned above, three challenges were considered. The first challenge is to develop cooperative schemes to estimate a gradient in each agent using only its relative position values and the strength signal values of the neighbours. This problem will be addressed in Chapter 2.

The second challenge is to develop a cooperative consensus filter to estimate the gradient direction in each agent when signals are corrupted by noise. This problem will be addressed in Chapter 3. This Chapter applies only for cases when signals are contaminated by noise. When gathering pure signals is possible, such a filter is not necessary.

The third challenge is to develop cooperative distributed controllers which are able to stabilize both agents individually and the whole formation in space, and additionally, steer agents either to an equilibrium in the immediate neighbourhood of the scalar field's source or allow them to navigate along a desired level curve. In order to solve the source seeking problem, cooperative distributed controllers for single and double integrator models, LTI systems and non-holonomic systems are developed. This problem will be addressed in Chapters 4, 5 and 6. In order to solve the level curve tracking problem, cooperative distributed controllers for double integrator models and LTI systems are developed. This problem will be addressed in Chapter 7.

When simple agent dynamics are considered, distributed navigation controllers for both single and double integrator models are designed and agents are programmed for computing a distributed gradient estimation individually and locally. In the case of complex dynamics, our approach is extended to general linear time-invariant (LTI) models and non-holonomic systems. Stability conditions are provided and the presented method is verified using formation flight simulation for quad-rotor helicopters. Performance requirements are incorporated into our design using mixed-sensitivity loop shaping. The H_∞ synthesis technique is used to design the controller. It will be demonstrated that the presented distributed controllers enable agents to converge towards the scalar field's source while their formation is maintained. In the case of non-holonomic systems, a simple distributed controller to avoid obstacles and collisions between agents and to steer agents towards the maximum of the scalar field is presented.

1.5.4 Control Architecture

As part of the solution strategy, a cooperative control architecture as shown in Fig. 1.3 is proposed. Distributed controllers composed of two parts are proposed: a formation control law and a trajectory control law. The formation control component aims is to keep agents in a desired geometric formation and the trajectory control component aims is to steer the formation towards the source or track a desired level curve. In this diagram the control input vector is represented by $u = [u_1, \ldots, u_N]^T$ where N indicates the number of agents, the agents' position vector is represented by $r = [r_1, \ldots, r_N]^T$ with $r_1 = [r_{1x}, r_{1y}, r_{1z}]^T$, the concentration measurement is represented by $\psi(r) = [\psi_1(r_1), \ldots, \psi_N(r_N)]^T$, and the estimated gradient is represented by $\hat{g}(\psi, r) = [\hat{g}_1, \ldots, \hat{g}_N]^T$. This control architecture takes into account that there is limited communication between agents, that there are different agent dynamics and that cooperative estimation of the gradient direction takes place in each agent.

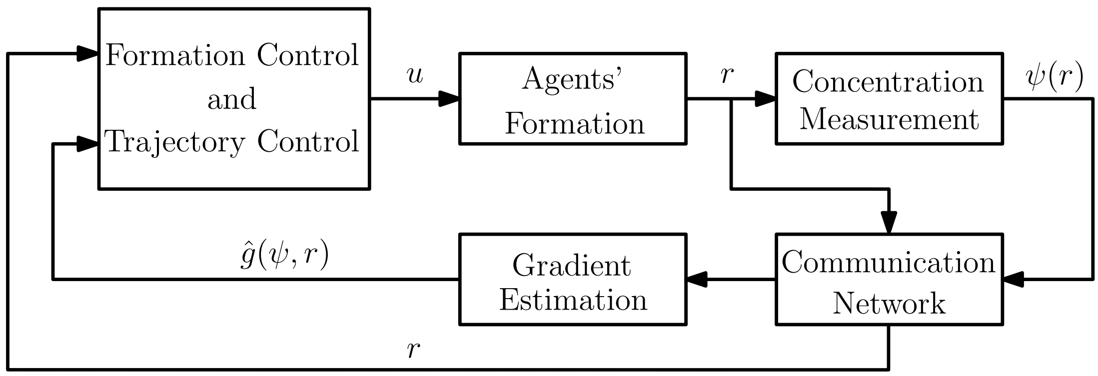


Figure 1.3: Architecture of the control strategy

1.6 Simulation Framework

The approach proposed in this work is illustrated using simulation studies. In this simulations $N = 7$ mobile agents are considered. The group of agents is ordered in a geometric formation and its communications are described by a communication graph $\mathcal{G}$ as shown by Fig. 1.4. The communication topology is undirected and chosen to be an hexagon. Each corner of the hexagon represents an agent and each edge represents a communication channel between agents.

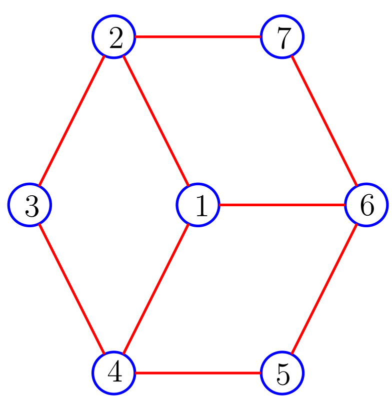


Figure 1.4: Geometric formation and communication topology of agents

A scalar field is defined as

$$\psi(r) = A_0 e^{-\left((r-r_s)^T H_1 (r-r_s)\right)} + A_0 e^{-\left((r-r_s)^T H_2 (r-r_s)\right)}, \tag{1.8}$$

where $A_0 = 3$, $H_1 = \begin{bmatrix} \frac{1}{2\sigma_{x1}^2} & 0 \\ 0 & \frac{1}{2\sigma_{y1}^2} \end{bmatrix}$, $H_2 = \begin{bmatrix} \frac{1}{2\sigma_{x2}^2} & 0 \\ 0 & \frac{1}{2\sigma_{y2}^2} \end{bmatrix}$, $\sigma_{x1} = 30$, $\sigma_{y1} = 75$, $\sigma_{x2} = 80$ and $\sigma_{y2} = 25$.

The maximum is located at $r_s = [40\ 80]^T$.

1.7 Contributions of this Work

In order to solve the problems mentioned above, either localizing the source or tracking a level curve, three important steps are made: firstly, three schemes to estimate the gradient direction in each agent are proposed. Secondly, a modified consensus filter under limited communication conditions is proposed. Thirdly, cooperative source seeking algorithms for single and double integrators, linear time-varying (LTI) systems and non-holonomic systems are proposed. In the case of cooperative level curve tracking, algorithms for double integrators and LTI systems are presented.

1. **Distributed gradient estimation:** Three simple distributed schemes to estimate the gradient direction in each agent using only neighbours' relative signal strength values and relative positions are proposed. The schemes are: unweighted, weighted and average gradient. Both unweighted and weighted gradient are estimated using least squares estimation. The average gradient is estimated using an average of the slopes between agent i and its neighbours j.

2. **Modified consensus filter:** If the position and concentration values are corrupted by noise, a modified consensus filter to estimate the gradient based on (Olfati-Saber and Shamma, 2005) is proposed (Rosero and Werner, 2014c). It is very useful under limited communication conditions. This proposed filter algorithm significantly improves the accuracy of time-varying signal tracking and attenuates high frequency noise. This new solution allows to define the desired bandwidth of the distributed consensus filter by tuning a control parameter. With this information, estimating the gradient direction becomes feasible.

3. **Distributed source seeking algorithms:** To solve the source seeking problem using multi-agent systems, a cooperative distributed source seeking algorithm for each one of the following agent models is proposed: single and double integrator (Rosero and Werner, 2014a), linear time-invariant dynamics (Rosero and Werner, 2014b) and non-holonomic systems. Convergence analysis is shown. The algorithms are designed under a fixed undirected communication topology. Position and concentration values of

neighbours are needed by each agent. In our approach, it is neither necessary to know the center of mass' position and the estimated gradient at this position, nor to keep the agents rotating in a circular formation, nor to have a leader in the formation, nor to have all-to-all communication between agents. These differences provide additional advantages over one-agent schemes. Applying the presented approach, agents use energy more efficiently, they can cover larger spatial areas, and this solution is suitable for a larger number of agents and for any type of formation.

4. **Distributed level curve tracking algorithms:** To solve the level curve tracking problem and based on distributed source seeking algorithms, a cooperative distributed level curve tracking algorithm for double integrator models and linear time-invariant dynamics is proposed. The algorithms are designed under a fixed undirected communication topology.

1.8 Structure of this Thesis

The thesis is organized as follows.

In Chapter 2 three methods to estimate the gradient in each agent in a cooperative way are derived. The methods are called unweighted, weighted and directional gradient. These methods are used later to design a distributed control law to move the agents towards the source of the scalar field or to move a group of agents along a level curve.

In Chapter 3 a modified consensus filter for signals corrupted by noise is presented. This filter is useful to reject noise at higher frequencies and track the time-varying signals. Based on this consensus filter and the methods to estimate the gradient in each agent presented in Chapter 2, two distributed gradient estimation schemes with position signals and signal strength corrupted by noise are provided.

In Chapter 4 two simple distributed control laws for single and double integrator agents to solve the source seeking problem are proposed. These control laws are based on both a formation controller and a trajectory controller. Stability analysis is provided to show that the agents move towards the source of the scalar field. Simulation results for both control laws support the effectiveness of our approach.

Chapter 5 extends the results of single and double integrator models, given in Chapter 4, to linear time invariant systems. The distributed control law for LTI systems solves the source seeking problem and allows us to include performance in the formation controller design process, too.

In Chapter 6 a simple distributed controller for non-holonomic systems (unicycle model) in order to solve the source seeking problem is proposed. The control law includes both collision avoidance between agents and between agents and obstacles.

In Chapter 7 the level curve tracking problem is solved for double- integrator and linear time-invariant systems. The proposed distributed control laws do not require the Hessian matrix estimation. Furthermore, the control laws are applied to an unknown fixed scalar field source and an unknown moving scalar field's source, as well as to an unknown moving scalar field's source inside an expanding or contracting scalar field.

In *Conclusions and Outlook* the results presented in this thesis and future directions of research are summarized.

Chapter 2

Distributed Gradient Estimation

Distributed gradient estimation is a central issue to be considered on the way to solve the scalar field's source or level curve tracking problem. As mentioned in the last chapter, both relative position and concentration signals are needed, and the gradient estimation takes place using neighbours' information only. This chapter presents three simple distributed schemes to estimate the gradient of the scalar field for each mobile agent in a network. The unweighted and weighted gradient estimation schemes are based on least-squares methods while the directional gradient estimation scheme is based on an average of the slopes between neighbouring agents. A group of N spatially distributed agents inside an unknown scalar field is considered. Each agent is equipped with both position and signal strength sensors. Each sensor measures unknown signals, computes its local gradient estimate and sends the information to its neighbours. Using the available information, each agent then follows the computed gradient. Consequently, the estimated gradient is then used to develop a cooperative control law which moves a group of agents in an autonomous manner towards the source of the scalar field or steers the agents to track a level curve.

2.1 Introduction

A direction based on gradient estimation is required for designing a cooperative controller that moves all agents towards the source of a scalar field. In (Li and Guo, 2012; Ogren et al., 2004; Zhang and Leonard, 2010) the gradient in the formation's center of mass is estimated using information of all agents. The authors assume that the center of mass is known to all agents. In (Brinon-Arranz et al., 2011) the gradient in the formation's center of mass is estimated using information of all network agents and agents rotate around this center. It is also assumed that the center of mass is known to all agents.

All these previous approaches compute the gradient in the geometric formation's center of mass and they require all-to-all communication between agents. Furthermore, all agents must know the center of mass' position. Thanks to collaborative field sensing and data exchange, the gradient estimation can be computed more efficiently. Since agents are equipped with sensors that are able to measure position and signal intensity inside the local field, it is possible for each agent to estimate the gradient direction taking advantage of the undirected communication scheme of the network.

In this chapter, a way to estimate the gradient at each agent's position using relative information is addressed. An arbitrary but pre-specified formation of N identical mobile agents under undirected constrained communication is used. Each agent estimates the gradient based on its own and neighbours' data. The estimated gradient will be later used to design distributed controllers moving the formation towards the source of the scalar field.

This chapter is organized as follows. Section 2.2 provides three methods for gradient approximation based on least squares and directional derivative. Section 2.3 provides simulation results. Finally, concluding remarks are made in Section 2.4.

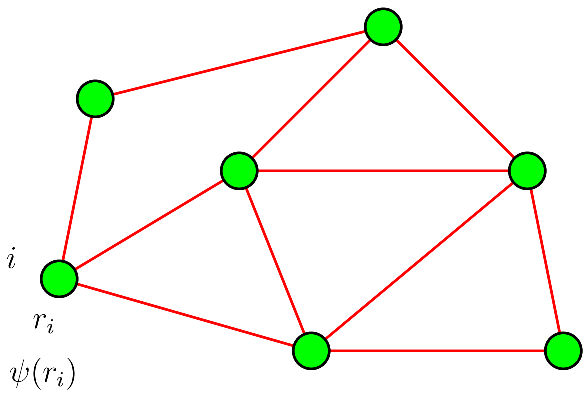


Figure 2.1: Mobile agents into a scalar field

2.2 Distributed Gradient Estimation

Consider a number of N mobile agents, spatially distributed, with communication graph $\mathcal{G}$ (Fig. 2.1). Each agent i measures the signal strength $\psi(r_i)$ at its position r_i where $i = 1, 2, ..., N$. Since $\psi(r_i)$ is twice continuously differentiable, it can be approximated by a Taylor series. The value at r_i is given by

$$\psi(r_i) = \psi(r_j) + (r_i - r_j)^T \nabla\psi(r_i) + \frac{1}{2}(r_i - r_j)^T \nabla^2\psi(r_i)(r_i - r_j) + H.O.T., \quad (2.1)$$

where i denotes the reference agent's index, j denotes a neighbour agent sending information to agent i, $\nabla\psi(r_i) \in \mathbb{R}^{p\times 1}$ is the true gradient, $\nabla^2\psi(r_i) \in \mathbb{R}^{p\times p}$ denotes the Hessian matrix and $H.O.T.$ the higher order terms.

2.2.1 Unweighted Gradient

For each agent i, it is possible to compute the slope between agent i and their neighbours $\mathcal{N}_i$. Neglecting high order terms, Equation (2.1) can be combined into

$$\underbrace{\begin{bmatrix} \psi(r_i) - \psi(r_1) \\ \vdots \\ \psi(r_i) - \psi(r_{|\mathcal{N}_i|}) \end{bmatrix}}_{\mathsf{b}_i} = \underbrace{\begin{bmatrix} (r_i - r_1)^T \\ \vdots \\ (r_i - r_{|\mathcal{N}_i|})^T \end{bmatrix}}_{\mathsf{R}_i} \nabla\psi(r_i), \tag{2.2}$$

where $\mathsf{b}_i \in \mathbb{R}^{|\mathcal{N}_i|\times 1}$ is a column vector of the scalar field's relative signal strength $\psi(r)$, $\mathsf{R}_i \in \mathbb{R}^{|\mathcal{N}_i|\times p}$ is a matrix whose coefficients depend on the relative position of the formation's geometric shape in the space $\mathbb{R}^p$. The problem can be solved minimizing $\|\mathsf{R}_i\hat{g}_i - \mathsf{b}_i\|^2$, then the local gradient $\hat{g}_i$ in each agent i can be estimated as

$$\begin{aligned} \hat{g}_i &= \left(\mathsf{R}_i^T\mathsf{R}_i\right)^{-1}\mathsf{R}_i^T\mathsf{b}_i, \\ &= \left(\sum_{j\in\mathcal{N}_i} a_{ij}(r_i - r_j)(r_i - r_j)^T\right)^{-1}\left(\sum_{j\in\mathcal{N}_i} a_{ij}(r_i - r_j)(\psi(r_i) - \psi(r_j))\right), \end{aligned} \tag{2.3}$$

where the distributed estimated gradient is $\hat{g}_i = f(r_i, r_j, \psi_i, \psi_j, \forall\, j \in \mathcal{N}_i) \in \mathbb{R}^{p\times 1}$ and it is assumed that the inverse of $\mathsf{R}_i^T\mathsf{R}_i$ exists. This means the matrix R_i must be full column rank (rank(R_i)$= p$). The rank $(\mathsf{R}_i) = p$ if and only if agent i and their neighbours $\mathcal{N}_i$ are non-collinear, i.e., they do not collapse in a plane ($p = 3$), in a line ($p = 2$) or in a point ($p = 1$). Since $\mathsf{R}_i \in \mathsf{R}^{|\mathcal{N}_i|\times p}$, this requires that $|\mathcal{N}_i| \geq p$. Note that the minimum number of agents to compute the gradient is $N_{min} = p + 1$. Fig. 2.2 shows the distributed estimated gradient for each mobile agent into a scalar field.

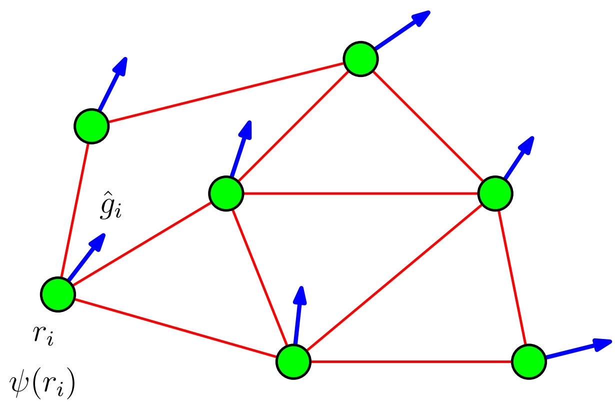


Figure 2.2: Distributed estimated gradient for each mobile agent

Note that the unweighted gradient computation can be arranged as

$$\hat{g}_i = \left(\sum_{j\in\mathcal{N}_i}(r_i - r_j)(r_i - r_j)^T\right)^{-1}\left(\sum_{j\in\mathcal{N}_i}(r_i - r_j)(\psi(r_i) - \psi(r_j))\right),$$

$$\hat{g}_i = \left(\sum_{j\in\mathcal{N}_i} z_{ij}z_{ij}^T\right)^{-1}\left(\sum_{j\in\mathcal{N}_i} z_{ij}\psi_{ij}\right),$$

where $z_{ij} = r_i - r_j$, $\psi_{ij} = \psi(r_i) - \psi(r_j)$,

$$\mathsf{R}_i^T\mathsf{R}_i = \sum_{j\in\mathcal{N}_i}(r_i - r_j)(r_u - r_j)^T = \sum_{j\in\mathcal{N}_i}\begin{bmatrix}(r_{xi} - r_{xj})^2 & (r_{xi} - r_{xj})(r_{yi} - r_{yj}) \\ (r_{xi} - r_{xj})(r_{yi} - r_{yj}) & (r_{yi} - r_{yj})^2\end{bmatrix},$$

$$\mathsf{R}_i^T\mathsf{b}_i = \sum_{j\in\mathcal{N}_i}(r_i - r_j)(\psi(r_i) - \psi(r_j)) = \sum_{j\in\mathcal{N}_i}\begin{bmatrix}(r_{xi} - r_{xj})\,(\psi(r_i) - \psi(r_j)) \\ (r_{yi} - r_{yj})\,(\psi(r_i) - \psi(r_j))\end{bmatrix},$$

for all $i \in \mathcal{V}$. Let us consider the properties of $\mathsf{R}_i^T\mathsf{R}_i$. $\mathsf{R}_i^T\mathsf{R}_i$ is symmetric and the first leading principal minor of $\mathsf{R}_i^T\mathsf{R}_i$ is nonnegative

$$\sum_{j\in\mathcal{N}_i}(r_{xi} - r_{xj})^2 \geq 0,$$

for all $i \in \mathcal{V}$. The equality holds if and only if agent i and all its neighbours are in the same r_x coordinate. Moreover, the second leading principal minor of $\mathsf{R}_i^T\mathsf{R}_i$ is nonnegative by the Cauchy-Schwarz inequality

$$\sum_{j\in\mathcal{N}_i}(r_{xi} - r_{xj})^2\sum_{j\in\mathcal{N}_i}(r_{yi} - r_{yj})^2 - \left(\sum_{j\in\mathcal{N}_i}(r_{xi} - r_{xj})(r_{yi} - r_{yj})\right)^2 \geq 0,$$

for all $i \in \mathcal{V}$. The equality holds if and only if the coordinates of the agents are linearly dependent (agents are collinear). For this reason, to compute the gradient $\hat{g}_i$ at the location of agent i, it is necessary to guarantee that both agent i and the neighbours $j \in \mathcal{N}_i$ are not collinear in the space $\mathbb{R}^p$. With this requirement fulfilled, all leading principal minors of $\mathsf{R}_i^T\mathsf{R}_i$ are positives. It follows that $\mathsf{R}_i^T\mathsf{R}_i$ is also positive definite. Furthermore, the matrix $(\mathsf{R}_i^T\mathsf{R}_i)^{-1}$ is also positive definite by the positive definiteness of $\mathsf{R}_i^T\mathsf{R}_i$.

Neglecting the higher order terms, the estimation error $e_{\nabla i}$ for each agent i can be computed as

$$e_{\nabla i} = (\mathsf{R}_i^T\mathsf{R}_i)^{-1}\mathsf{R}_i^T e_{di}, \tag{2.4}$$

where

$$e_{di} = \begin{bmatrix}\frac{1}{2}(r_i - r_1)^T\nabla^2\psi(r_i)(r_i - r_1) \\ \vdots \\ \frac{1}{2}(r_i - r_{|\mathcal{N}_i|})^T\nabla^2\psi(r_i)(r_i - r_{|\mathcal{N}_i|})\end{bmatrix},$$

with $e_{di} \in \mathbb{R}^{|\mathcal{N}_i|\times 1}$. Defining the maximum distance between agents in the formation as $l_a = \max_{j\in\mathcal{N}_i}\|r_i - r_j\|$ and considering that the Hessian $\nabla^2\psi(r_i)$ is bounded by $\|\nabla^2\psi(r_i)\| \le L_H$, then $\frac{1}{2}(r_i - r_j)^T\nabla^2\psi(r_i)(r_i - r_j) \le \frac{1}{2}L_{Hi}\|r_i - r_j\|^2 \le \frac{1}{2}L_{Hi}l_a^2$. Thus, for each agent i, the error e_{di} is bounded by $\|e_{di}\| \le \frac{1}{2}L_{Hi}l_a^2\sqrt{|\mathcal{N}_i|}$, and the estimation error by

$$\begin{aligned}\|e_{\nabla i}\| \le& \|(\mathsf{R}_i^T\mathsf{R}_i)^{-1}\mathsf{R}_i^T\|\|e_{di}\|,\\ \le&\frac{1}{2}L_{Hi}l_a^2\sqrt{|\mathcal{N}_i|}\|(\mathsf{R}_i^T\mathsf{R}_i)^{-1}\mathsf{R}_i^T\| = e_{0i}.\end{aligned} \tag{2.5}$$

Note that the estimated error depends on the square of the distance between agents. To make the lower order terms in the Taylor expansion dominate, the distances between agents $\|r_i - r_j\|$ must be sufficiently small.

2.2.2 Weighted Gradient

If Equation (2.1) is normalized by the relative distance's norm between the agent i an their neighbours j and $w_{ji} = \frac{1}{\|r_j - r_i\|}$, then the slope in the direction between agents i and j is found. For each agent i, and using a first order approximation, it is possible to write

$$\mathsf{W}_i\underbrace{\begin{bmatrix}\psi(r_i)-\psi(r_1)\\ \vdots\\ \psi(r_i)-\psi(r_{|\mathcal{N}_i|})\end{bmatrix}}_{\mathsf{b}_i} = \mathsf{W}_i\underbrace{\begin{bmatrix}(r_i-r_1)^T\\ \vdots\\ (r_i-r_{|\mathcal{N}_i|})^T\end{bmatrix}}_{\mathsf{R}_i}\nabla\psi(r_i), \tag{2.6}$$

where $\mathsf{b}_i \in \mathbb{R}^{|\mathcal{N}_i|\times 1}$ is a column vector of the relative signal strength of the scalar field $\psi(r)$, $\mathsf{R}_i \in \mathbb{R}^{|\mathcal{N}_i|\times p}$ is a matrix whose coefficients depend on the relative position of the formation's geometric shape in the space $\mathbb{R}^p$ and $\mathsf{W}_i = \mathrm{diag}\{w_{ij}\} \in \mathbb{R}^{|\mathcal{N}_i|\times|\mathcal{N}_i|}$ is a diagonal matrix containing the weights of all neighbouring agents $|\mathcal{N}_i|$ that are sending information to agent i. Then, for each agent i, it is possible to write

$$\mathsf{W}_i\mathsf{b}_i = \mathsf{W}_i\mathsf{R}_i\nabla\psi(r_i).$$

Using least-squares, the estimate can be computed by

$$\hat{g}_i = \left(\mathsf{R}_i^T\mathsf{W}_i^2\mathsf{R}_i\right)^{-1}\mathsf{R}_i^T\mathsf{W}_i^2\mathsf{b}_i, \tag{2.7}$$

where the distributed estimated gradient $\hat{g}_i = f(r_i, r_j, \psi_i, \psi_j, \forall\ j \in \mathcal{N}_i) \in \mathbb{R}^{p\times 1}$ and provided that the inverse of $\mathsf{R}_i^T\mathsf{R}_i$ exists. This means the matrix R_i must be full column rank (rank(R_i)$= p$).

Neglecting the higher order terms, the estimation error for each agent i can be computed as

$$e_{\nabla i} = (\mathsf{R}_i^T\mathsf{W}_i^2\mathsf{R}_i)^{-1}\mathsf{R}_i^T e_{di}, \tag{2.8}$$

where $e_{di} \in \mathbb{R}^{|\mathcal{N}_i| \times 1}$ and

$$e_{di} = \mathsf{W}_i \begin{bmatrix} \frac{1}{2}(r_i - r_1)^T \nabla^2 \psi(r_i)(r_i - r_1) \\ \vdots \\ \frac{1}{2}(r_i - r_{|\mathcal{N}_i|})^T \nabla^2 \psi(r_i)(r_i - r_{|\mathcal{N}_i|}) \end{bmatrix}.$$

Defining the maximum distance between two agents in the formation as $l_a = \max_{j \in \mathcal{N}_i} \|r_i - r_j\|$ and considering that the Hessian $\nabla^2 \psi(r_i)$ is bounded by $\|\nabla^2 \psi(r_i)\| \leq L_H$, then $\frac{1}{2} w_{ji}(r_i - r_j)^T \nabla^2 \psi(r_i)(r_i - r_j) \leq \frac{1}{2} L_{Hi} \|r_i - r_j\| \leq \frac{1}{2} L_{Hi} l_a$. Thus, for each agent i, the error e_{di} is bounded by $\|e_{di}\| \leq \frac{1}{2} L_{Hi} l_a \sqrt{|\mathcal{N}_i|}$ and the estimation error by

$$\begin{aligned} \|e_{\nabla i}\| \leq & \|(\mathsf{R}_i^T \mathsf{W}_i^2 \mathsf{R}_i)^{-1} \mathsf{R}_i^T\| \|e_{di}\|, \\ \leq & \frac{1}{2} L_{Hi} l_a \sqrt{|\mathcal{N}_i|} \|(\mathsf{R}_i^T \mathsf{W}_i^2 \mathsf{R}_i)^{-1} \mathsf{R}_i^T\| = e_{0i}. \end{aligned} \tag{2.9}$$

Note that e_{di} only depends on the distance between agents. In this case, when the Taylor approximation is normalized by the relative distance between agents, the estimated gradient's accuracy is improved in relation to the unweighted gradient approach proposed in (Rosero and Werner, 2014a).

As mentioned in Section 1.5, following assumptions have been made:

Assumption 1. *Graph $\mathcal{G}$ is undirected and connected. It is assumed that the formation satisfies $|\mathcal{N}_i| \geq p$, $N \geq p+1$ and the agent i and their neighbours are not collinear, i.e., each agent has at least p neighbours and the minimum number of agents in the space is $p+1$.*

Assumption 2. *The scalar field has an isolated global maximum. The estimated gradient is bounded by $\|\hat{g}_i(r_i)\| \leq \mu$, i.e., the estimated gradient will not be infinite for any time t. The average of the estimated gradient's error is bounded by $\|\bar{e}_\nabla\| \leq e_0$.*

2.2.3 Average Gradient

If Equation (2.1) is normalized by the relative distance between agents and multiplied by the unit direction vector between agent i and its neighbours j, the gradient can be computed as an average of the slopes in the direction of the normalized vector as

$$\hat{g}_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \frac{\psi(r_i) - \psi(r_j)}{\|r_i - r_j\|^2} (r_i - r_j). \tag{2.10}$$

This approximation is useful when some agents inside the communication topology do not entirely satisfy the condition $|\mathcal{N}_i| \geq p$. At the other hand, the same approximation is not optimal based on least squares estimator.

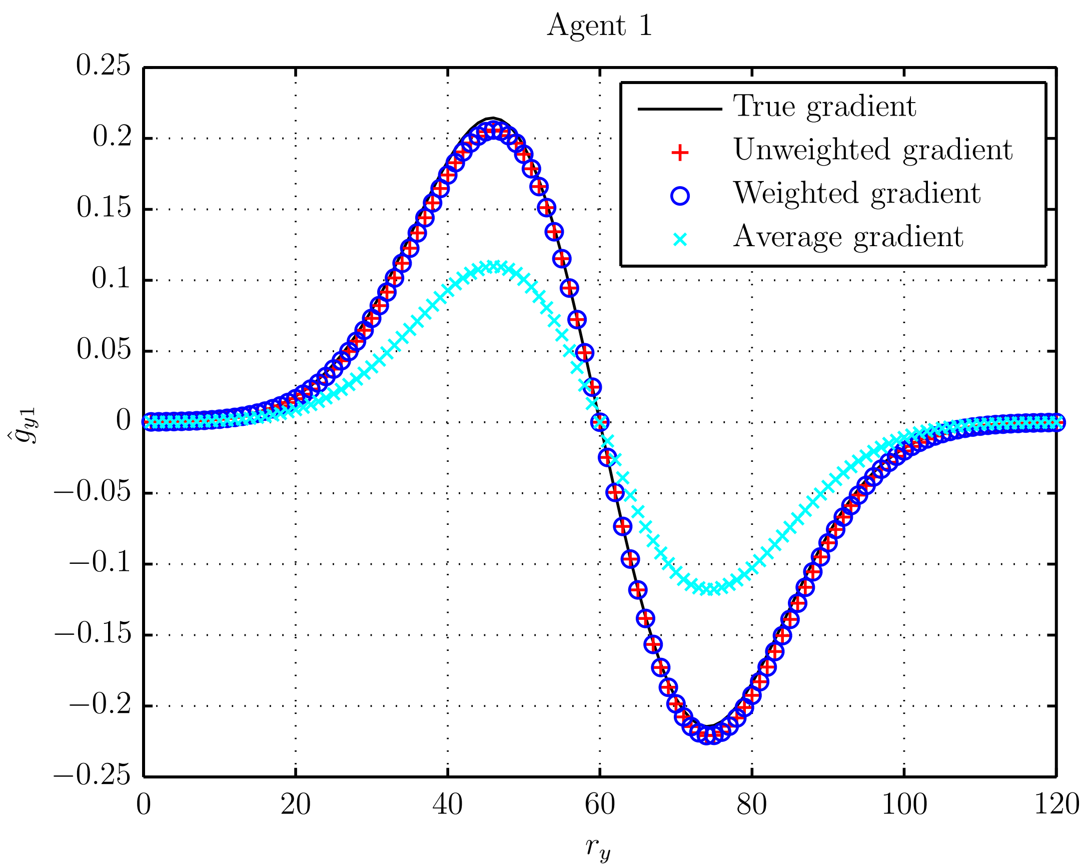


Figure 2.3: $\hat{g}_{y1}$-component's estimated gradient of agent 1 with $r_s = [r_{x0}\ r_{y0}]^T = [60\ 60]^T$

2.3 Simulation Results

This section shows simulation results for the estimated gradient method previously computed. The scalar field is defined as $\psi(r) = Ae^{-(\frac{(r_x - r_{x0})^2}{2\sigma_x^2} + \frac{(r_y - r_{y0})^2}{2\sigma_y^2})}$, the source is centred at the maximum value $r_s = [r_{x0}\ r_{y0}]^T = [60\ 60]^T$, with $A = 10$ and $\sigma_x = \sigma_y = 20$. Note that level curves are circular. A formation of $N = 7$ agents with undirected communication topology and a geometric formation are both considered, as illustrated in Fig. 1.4.

Figs. 2.3 and 2.4 present a comparison between the true gradient and the 3 proposed estimated gradient methods (unweighted, weighted and average estimated gradient) for agent 1 only. Agents travel across the scalar field with a fixed geometric shape and at constant velocity. The results reveal that the unweighted and weighted gradient approximations result in very similar values to the true gradient while the average gradient approximation's error is larger. When the distance between agents increases, the estimated gradient error increases, too. Furthermore, this effect can be observed in Equations (2.8) and (2.9).

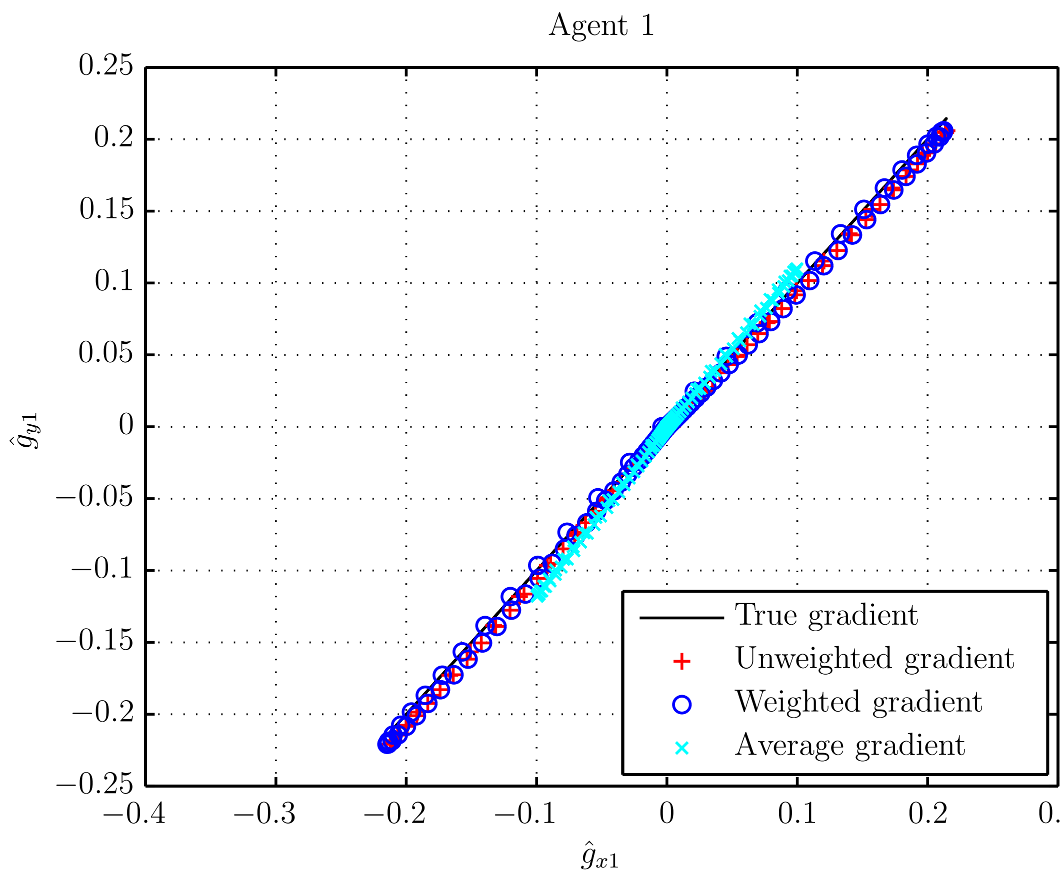


Figure 2.4: $\hat{g}_{x1} - \hat{g}_{y1}$ components' estimated gradient of agent 1

2.4 Conclusions

In this chapter, three methods to estimate gradient in a cooperative way (estimation takes place in each agent separately) have been presented. The unweighted and weighted gradient are computed by using least squares estimator while the average gradient is computed using only the average of the slopes between neighbour agents. These methods require gathering relative position and relative signal strength information from each agent's local neighbours only. The gradient estimated this way is defined by a vector with a magnitude and a direction. The magnitude is the value of the rate of change measured in the direction of the highest rate of change inside the scalar field.

The average gradient and the unweighted gradient methods are useful because they are easy to implement. Nevertheless, the average gradient method is less precise than the others because it is not computed taking the least squares estimator into account. The estimation error of the weighted gradient is smaller than the estimation error of the unweighted gradient and the estimation error of the average gradient, because the weighted gradient method uses the distance between agent i and its neighbours as the weighting factor. However, the weighted gradient method requires more processing capability than the unweighted and the average gradient methods.

Chapter 3

Modified Distributed Consensus Filter for Sensor Networks

This chapter considers distributed consensus filters for sensor networks. In such networks each group of sensors measures the same signal, in this case a signal corrupted by noise. An algorithm which significantly improves the accuracy of time-varying signal tracking and attenuates high frequency noise is proposed. It is a modification of an existing algorithm (Olfati-Saber and Shamma, 2005). This new algorithm allows us to define the desired bandwidth of the distributed consensus filter by tuning a control parameter. The consensus filter solves data fusion problems in a distributed way, programming each sensor to use information only from its local neighbours to perform filtering in a fixed communication topology. The filtering is performed in each node of the sensor network. Convergence analysis is provided. Simulation results illustrate the performance of the proposed filter.

3.1 Introduction

Sensor networks have a growing range of applications in fields requiring collaborative information processing. Examples are measurement of physical parameters, monitoring environmental conditions, sensor management, etc. A sensor network consists of a number of sensor nodes that are spatially distributed. Each sensor receives an input and estimates the average of inputs to all the sensors in the network. It then communicates this estimation to its neighbours. Since each sensor estimates the average of inputs, this is called an average consensus problem.

Average consensus estimation for sensor networks has been discussed in (Bai et al., 2010; Brinon-Arranz et al., 2011; Freeman et al., 2006; Li and Guo, 2013; Olfati-Saber and Shamma, 2005; Spanos et al., 2005; Xiao et al., 2007, 2005). In (Xiao et al., 2005), a static average consensus protocol is applied and each node computes a locally weighted least-squares estimate in order to reach convergence for the overall network. In (Xiao et al., 2007), a distributed average computation of time-varying signals is studied for inputs affected by zero-mean noise. In (Olfati-Saber and Shamma, 2005) and (Spanos et al., 2005), a dynamic consensus filter for distributed low-pass sensor fusion is proposed. It tracks the input's average of all sensors in a network. It is assumed that each sensor measures the same signal corrupted by noise.

In the case that inputs were not equal for each sensor, a proportional-integral (PI) consensus filter is proposed in (Freeman et al., 2006), (Bai et al., 2010) and (Li and Guo, 2013). This filter let the inputs accurately converge when they are time-invariant. In (Bai et al., 2010), the authors design a generalized PI consensus filter to track the average of ideal time-varying inputs by exploiting the internal model principle.

In (Li and Guo, 2013) authors modify the PI consensus filter proposed in (Freeman et al., 2006) and remove the requirement of bi-directional exchange of neighboring gains. They also extend the results to balanced directed graphs with switching topologies via hybrid filter in order to compensate the effect introduced by switching.

An application of the algorithm based on (Olfati-Saber and Shamma, 2005) is presented in (Brinon-Arranz et al., 2011). The authors estimate the gradient direction of signal propagation using a fixed circular sensor formation. A gain is introduced to modify the eigenvalues of the Laplacian matrix, and authors consider that the input is a different vector for each sensor.

In this chapter a modification of the algorithm in (Olfati-Saber and Shamma, 2005) is proposed, which significantly improves the accuracy of time-varying signal tracking. By defining the sensor network's bandwidth it attenuates high frequency noise. A tuning parameter which increases the convergence speed and reduces the convergence region is included. This result will be used to estimate the gradient in a distributed way in Sections 3.4 and 3.5 when the concentration and position signals are corrupted by noise at high frequency (unweighted and weighted gradient estimation).

This chapter is organized as follows: Section 3.2 provides the main results on design and convergence analysis of the proposed distributed consensus filter. In Section 3.3 detailed simulation results are presented. In Sections 3.4 and 3.5, the unweighted and weighted distributed gradient estimation with noise is presented. Finally, concluding remarks follow in Section 3.6.

3.2 Consensus Filter

Consider a sensor network of size N, spatially distributed with undirected communication graph $\mathcal{G}$. Assume that each sensor measures a signal $r(t) \in \mathbb{R}^p$ corrupted by noise $n_i(t) \in \mathbb{R}^p$. The sensing model is given by

$$u_i(t) = r(t) + n_i(t), \ i \in \mathcal{V}, \tag{3.1}$$

where $n_i(t)$ is a zero-mean Gaussian noise. Define the vectors $u(t) = [u_1^T(t), \dots, u_N^T(t)]$, $r(t) = [r_1^T(t), \dots, r_N^T(t)]^T$ and $n(t) = [n_1^T(t), \dots, n_N^T(t)]^T$, then the sensing model for the whole network is

$$u(t) = \mathbf{1}r(t) + n(t). \tag{3.2}$$

The covariance matrix of $n(t)$ is assumed to be diagonal $\Sigma = \mathrm{diag}(\Sigma_1, \dots, \Sigma_N)$, i.e., $n_i(t)$ and $n_j(t)$ are uncorrelated. A distributed consensus algorithm proposed in (Olfati-Saber and Shamma, 2005) is given by

$$\begin{aligned}\dot{x}_i(t) = &- \sum_{j\in\mathcal{N}_i} a_{ij}(u_i(t) - u_j(t)) - \sum_{j\in\mathcal{N}_i} a_{ij}(x_i(t) - x_j(t)) \\ &+ (1 + d_i)(u_i(t) - x_i(t)),\end{aligned} \tag{3.3}$$

where $x_i(t) \in \mathbb{R}^p$ is the estimation of the target r, obtained in sensor node i by taking a linear combination of neighbouring estimates and measurements. In order to rebuild the signal r, the following modified algorithm is proposed

$$\begin{aligned}\dot{x}_i(t) =& \beta \sum_{j\in\mathcal{N}_i} a_{ij}(u_i(t) - u_j(t)) - \beta \sum_{j\in\mathcal{N}_i} a_{ij}(x_i(t) - x_j(t)) \\ &+ \beta(1 + d_i)(u_i(t) - x_i(t)),\end{aligned} \tag{3.4}$$

where $\beta > 0$ is a tuning parameter. The difference between the consensus algorithm proposed by (Olfati-Saber and Shamma, 2005) and our approach is that a tuning parameter β and change of sign of the term $\sum_{j\in\mathcal{N}_i} a_{ij}(u_i(t) - u_j(t))$ is added. The resulting Equation (3.4) allows significant improvements in the consensus filter, as shown below.

In vector notation, and using the definition of the Laplacian graph, Equation (3.4) can be rewritten as

$$\dot{x} = -\beta(I_N + \Delta + \mathcal{L})(x - u) = -\beta A(x - u), \tag{3.5}$$

where $A = I_N + \Delta + \mathcal{L}$ is a symmetric positive definite matrix ($A \succ 0$). Applying the Gersgorin theorem to matrix A, it is shown that $1 + d_{min} \leq \lambda_{min}(A) \leq \lambda_{max}(A) \leq 1 + 3d_{max}$. Each state of a node i is initialized with $x_i(0) = x_{i0}$.

3.2.1 Filter Analysis

To show the effect of parameter β on performance and stability of the algorithm, the proposed consensus filter is analysed as follows. Let $U \in \mathbb{R}^{N\times N}$ be a unitary matrix such that $U^{-1}(\Delta + \mathcal{L})U = \Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_N)$, where $0 < \lambda_1, \dots, \lambda_N$. Let $x = (U \otimes I_p)\tilde{x}$, where $\tilde{x} = [\tilde{x}_1^T, \dots, \tilde{x}_N^T]^T$. Further, rewrite the input signals as $\mathbf{1}r = (U \otimes I_p)(\mathbf{1}\tilde{r})$, and $n = (U \otimes I_p)\tilde{n}$, where $\tilde{n} = [\tilde{n}_1^T, \dots, \tilde{n}_N^T]^T$, then Equation (3.5) can be rewritten as

$$\dot{\tilde{x}} = -\beta((I_N + \Lambda) \otimes I_p)\tilde{x} + \beta((I_N + \Lambda) \otimes I_p)(\mathbf{1}\tilde{r} + \tilde{n}). \tag{3.6}$$

Note that the weight factor β is the same for all directions of the state. Therefore, for each sensor i, it is possible to write

$$\dot{\tilde{x}}_i = -\beta(1+\lambda_i)\otimes I_p)\tilde{x}_i + \beta(1+\lambda_i)\otimes I_p)(\tilde{r}+\tilde{n}_i).$$

The transfer function of the consensus filter for each sensor i is given by

$$\tilde{x}_i(s) = [s+\beta(1+\lambda_i)\otimes I_p]^{-1}\,\beta\,[(1+\lambda_i)\otimes I_p]\,(\tilde{r}(s)+\tilde{n}_i(s)).$$

If $p=1$, the above equation can be written as

$$\tilde{x}_i(s) = \frac{\beta(1+\lambda_i)}{s+\beta(1+\lambda_i)}(\tilde{r}(s)+\tilde{n}_i(s)). \tag{3.7}$$

The pole of the transfer function is strictly negative and thus the filter is stable. Furthermore, the transfer function in (3.7) is strictly proper, it means it is a low-pass filter. When β is increased and $\beta > 1$, the poles move to the left and the speed of the sensor network becomes faster. If β is reduced such that $0 < \beta < 1$, the poles move towards the imaginary axis and the speed of the sensor network becomes slower. Note that β does not modify the static gain of the transfer function ($t \to \infty$, $s \to 0$). This static gain is always equal to 1 for all the eigenvalues resulting in improved tracking accuracy.

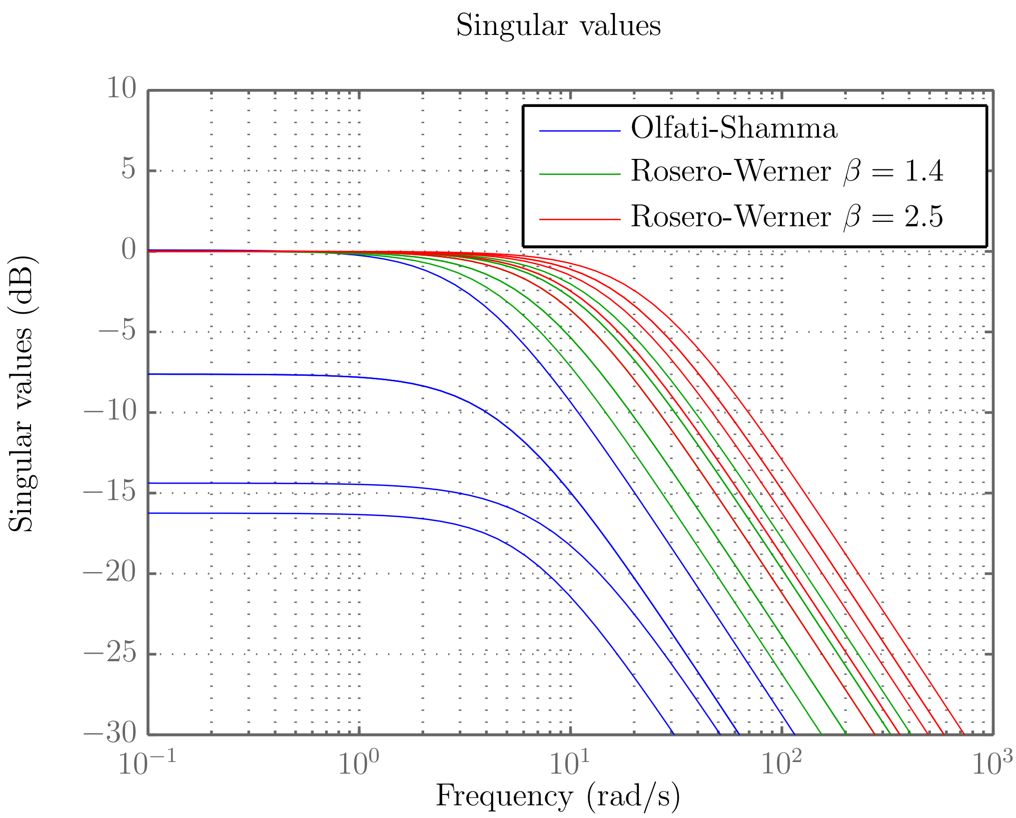


Figure 3.1: Singular value plots of the consensus filter

Since tracking and noise rejection are conflicting design objectives, it is necessary to define the frequency range. Therefore β is selected ensuring that the target will be tracked and noise will be rejected. The algorithm suppresses the effect of noise at high frequency. If the bandwidth is defined as $\omega_{ci} = \beta(1+\lambda_i)$, it is possible to write for each eigenvalue $\beta(1+\lambda_1) \leq \omega_{ci} \leq \beta(1+\lambda_N) = \omega_{cmax}$. Tuning the parameter β, it is possible to design the sensor network's bandwidth in order to attenuate the high frequency noise as $\beta = \frac{\omega_{cmax}}{1+\lambda_N}$. If $\omega \ll \omega_{cmax}$, then $\left|\frac{\tilde{x}_i(s)}{\tilde{u}_i(s)}\right| = 1$. If $\omega \gg \omega_{cmax}$, then $\left|\frac{\tilde{x}_i(s)}{\tilde{u}_i(s)}\right| = \frac{\omega_{cmax}}{\omega}$.

Fig. 3.1 shows a comparison of singular values of the system (3.5) between our approach (with $\beta = 1.4$ and $\beta = 2.5$) and the approach given by (Olfati-Saber and Shamma, 2005), with the communication topology given by Fig. 1.4. Fig. 3.1 shows that the proposed approach has a gain 1 at low-frequency for all singular values. Tuning β (with $\beta \geq 1$), it is possible to track the targets faster. The tracking is slower if $0 < \beta < 1$. In both cases the filter rejects the high frequency noise.

3.2.2 Convergence Analysis

To analyse the convergence of the consensus filter algorithm the following theorem is proposed.

Theorem 1. *Let $r(t)$ be a signal with an uniformly bounded rate $\|\dot{r}(t)\| \leq \nu$. Then $x^*(t) = r(t)\mathbf{1}$ is a globally asymptotically ϵ-stable equilibrium of the consensus filter dynamics given by (3.3) with input $u(t) = r(t)\mathbf{1}$ and*

$$\epsilon = \frac{\nu\sqrt{N}\,(1+d_{max})}{\beta\lambda_{min}^2(A)}\sqrt{\frac{\lambda_{max}(A)}{\lambda_{min}(A)}}. \tag{3.8}$$

Proof. The equilibrium is defined at $x(t) = \mathbf{1}r(t)$. Define the error vector $\eta(t) = x(t) - \mathbf{1}r(t)$, then, the dynamics of the error can be written as

$$\dot{\eta} = -\beta A\eta - \mathbf{1}\dot{r}. \tag{3.9}$$

Now define the Lyapunov function $V(\eta) = \frac{1}{2}\eta^T A\eta$ for the dynamic errors in (3.9). Then the derivative of $V(\eta)$ along the trajectories of the system is given by

$$\dot{V}(\eta) = -\beta\eta^T A^T A\eta - \dot{r}^T\mathbf{1}^T A\eta. \tag{3.10}$$

The derivative of the Lyapunov function is bounded by

$$\dot{V}(\eta) \leq -\beta\lambda_{min}^2(A)\|\eta\|^2 + \|\dot{r}^T\|\|\mathbf{1}^T A\eta\|. \tag{3.11}$$

Let $\|\dot{r}(t)\| \leq \nu$, then

$$\dot{V}(\eta) \leq -\beta\lambda_{min}^2(A)\|\eta\|^2 + \nu\sqrt{N}\,(1+d_{max})\,\|\eta\|, \tag{3.12}$$

where $\|\mathbf{1}^T A\eta\| \leq \sqrt{N}(1+d_{max}\|\eta\|)$ and $1+d_{min} \leq \lambda_{max}(A) \leq \lambda_{max}(A) \leq 1+3d_{max}$. Completing the square, it is possible to obtain

$$\dot{V}(\eta) \leq -\beta\lambda_{min}^2(A)\left(\|\eta\| - \frac{\nu\sqrt{N}\,(1+d_{max})}{2\beta\lambda_{min}^2(A)}\right)^2 + \frac{\left(\nu\sqrt{N}\,(1+d_{max})\right)^2}{4\beta\lambda_{min}^2(A)}. \tag{3.13}$$

Following the reasoning of proposition 2 in (Olfati-Saber and Shamma, 2005), let B_ρ be a closed ball centred at $\eta = 0$ wit radius

$$\rho \leq \frac{\nu\sqrt{N}\,(1+d_{max})}{\beta\lambda^2_{min}(A)}, \tag{3.14}$$

and let $\Omega_c = \{\eta : V(\eta) \leq c\}$ be a level-set of the Lyapunov function $V(\eta)$ with $c = \frac{1}{2}\lambda_{max}(A)\rho^2$. Then, B_ρ is contained in Ω_c because

$$\|\eta\| \leq \rho \Rightarrow V(\eta) = \frac{1}{2}\eta^T A\eta \leq \frac{1}{2}\eta^T\lambda_{max}(A)\rho^2 = c,$$

and thus $\eta \in \Omega_c$. As a result, any solution of (3.9) starting in $\mathbb{R}^p \setminus \Omega_c$ satisfies $\dot{V}(\eta) < 0$. Thus, it enters Ω_c in some finite time and remains in Ω_c thereafter (i.e. Ω_c is an invariant level-set). This guarantees global asymptotic ϵ-stability of $\eta = 0$ with radius $\epsilon = \rho\lambda_{max}(A)/\lambda_{min}(A)$. To show this, note that

$$\frac{1}{2}\lambda_{min}(A)\|\eta\|^2 \leq V(\eta) \leq \frac{1}{2}\lambda_{max}(A)\rho^2. \tag{3.15}$$

Thus, the solutions enter the region

$$\|\eta\| \leq \rho\sqrt{\frac{\lambda_{max}(A)}{\lambda_{min}(A)}}, \tag{3.16}$$

which implies the radius of ϵ- stability is

$$\epsilon = \rho\frac{\lambda_{max}(A)}{\lambda_{min}(A)} = \frac{\nu\sqrt{N}\,(1+d_{max})}{\beta\lambda^2_{min}(A)}\sqrt{\frac{\lambda_{max}(A)}{\lambda_{min}(A)}}. \tag{3.17}$$

Of course, ϵ-stability of $\eta = 0$ implies ϵ-tracking of $r(t)$ by every node of the network (i.e., ϵ-consensus is asymptotically reached). □

Remark 1. *Using the consensus algorithm (3.3) proposed by (Olfati-Saber and Shamma, 2005), the radius of $\tilde{\epsilon}$-stability is given by*

$$\tilde{\epsilon} = \beta\epsilon. \tag{3.18}$$

3.2.3 Convergence Analysis for Regular Networks

If a regular network $\mathcal{G}$ of degree $\bar{d}$ is considered, where $d_{max} = d_{min} = \bar{d} = \delta N^\gamma$, $I_N + \Delta = (\bar{d}+1)I_N$ and $\lambda_{max} = \lambda_{min}(A) = 1 + \bar{d}$, the expression for ϵ simplifies to

$$\epsilon = \frac{\nu\sqrt{N}(1+\bar{d})}{\beta(1+\bar{d})^2} = \frac{\nu\sqrt{N}}{\beta(1+\delta N^\gamma)}, \tag{3.19}$$

while the authors in (Olfati-Saber and Shamma, 2005) obtained

$$\tilde{\epsilon} = \beta\epsilon. \tag{3.20}$$

Note that if β is increased ($\beta > 1$), the radius of ϵ is smaller.

The advantages of the algorithm here presented over the algorithm presented in (Olfati-Saber and Shamma, 2005) are:

- Its gain for all the eigenvalues is 1, resulting in improved tracking accuracy.
- Tuning of the parameter β improves the convergence speed and the convergence region.
- $\beta = \frac{\omega_{cmax}}{1+\lambda_N}$ is introduced as a suitable choice for designing the sensor network's bandwidth.

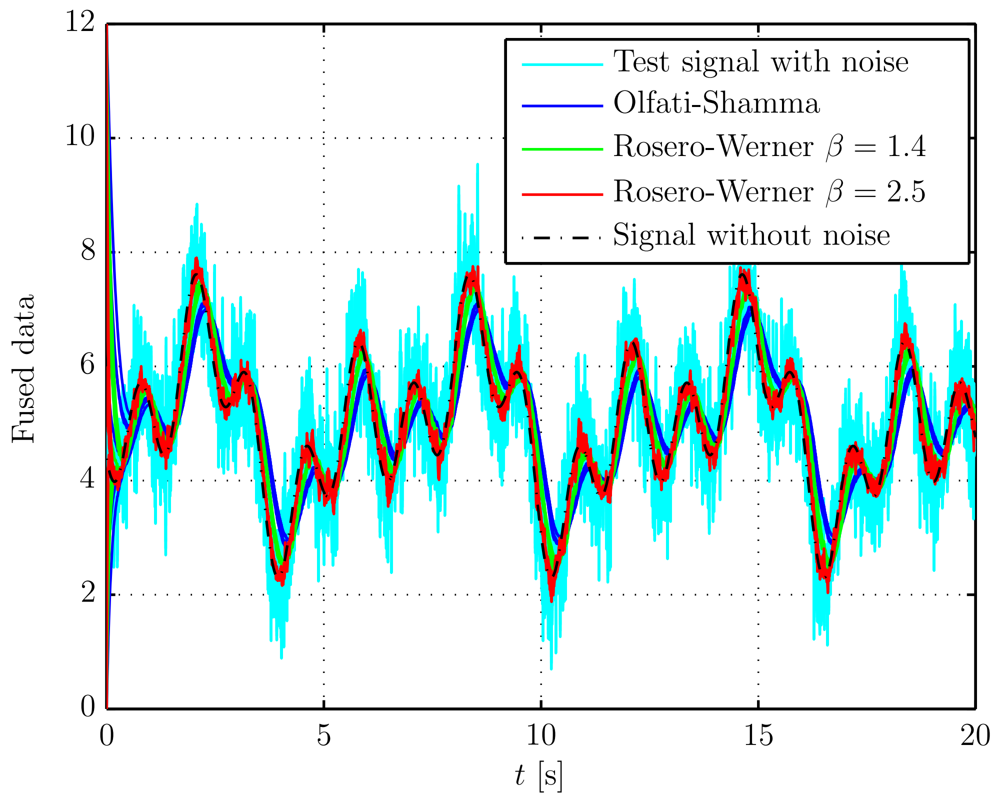


Figure 3.2: Fused sensor data for sensor measurements $r_a(t) + n_i(t)$

3.3 Simulation Results

Simulation results for sensor networks with $N = 7$ in a dimension $p = 1$ are presented. The communication topology of the graph is shown in Fig. 1.4. The results are tested via sinusoidal target tracking, using the following test signals as

$$r_a(t) = 5 + \sin(t) + \sin(2t + 3) + \sin(5t + 4),$$
$$r_b(t) = 5 + \sin(2t),$$
$$r_c(t) = 5 + \sin(5t).$$

The covariance matrix is taken as $\Sigma = \mathrm{diag}(0.5\ 0.29\ 0.15\ 0.3\ 0.45\ 0.4\ 0.6)$ for each sensor, with initial conditions $x(0) = [0\ 2\ 4\ 6\ 8\ 10\ 12]^T$.

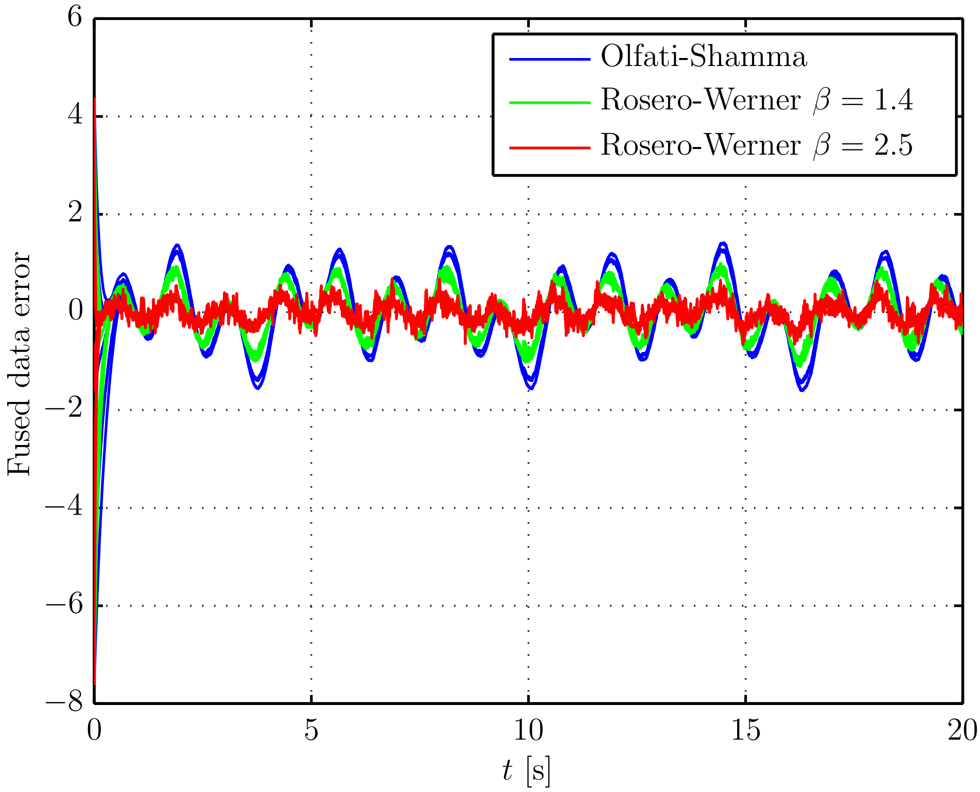


Figure 3.3: Fused sensor data error for sensor measurements $r_a(t) + n_i(t)$

Figs. 3.2, 3.4 and 3.6 show sensor fusion using a low-pass consensus filter based on the approach presented by (Olfati-Saber and Shamma, 2005) and our approach with $\beta = 1.4$ and $\beta = 2.5$. Figs. 3.3, 3.5, and 3.7 show the error between the reference signals and the estimated signals for each sensor. For $\beta \geq 1$ the performance of our approach is better than the approach presented by (Olfati-Saber and Shamma, 2005), and the convergence speed is faster with a smaller error. This result is supported by equation (3.17). Figs. 3.3 and 3.5 show that for signals with frequency lower than the maximum bandwidth of the network and $\beta \geq 1$ the signal tracking error of our approach is smaller than the tracking error in the approach presented by (Olfati-Saber and Shamma, 2005). If $\beta \gg 1$, the signal tracking error is smaller and the effect of noise appears.

Figs. 3.6 and 3.7 show the comparison between the approach presented by (Olfati-Saber and Shamma, 2005) and our approach with $\beta = 1.4$ and $\beta = 2.5$ for a target signal with frequency 5 rad/s. The maximum bandwidth for the approach based on (Olfati-Saber and Shamma, 2005) is 3.6 rad/s while for our approach the maximum bandwidth is 13.3 rad/s for $\beta = 1.4$ and 24 rad/s for $\beta = 2.5$, respectively. The tracking signal error of our approach is smaller than the tracking error of the approach presented by (Olfati-Saber and Shamma, 2005) because the network's bandwidth is increased by increasing the parameter β.

All in all, the proposed consensus algorithm has a better performance than

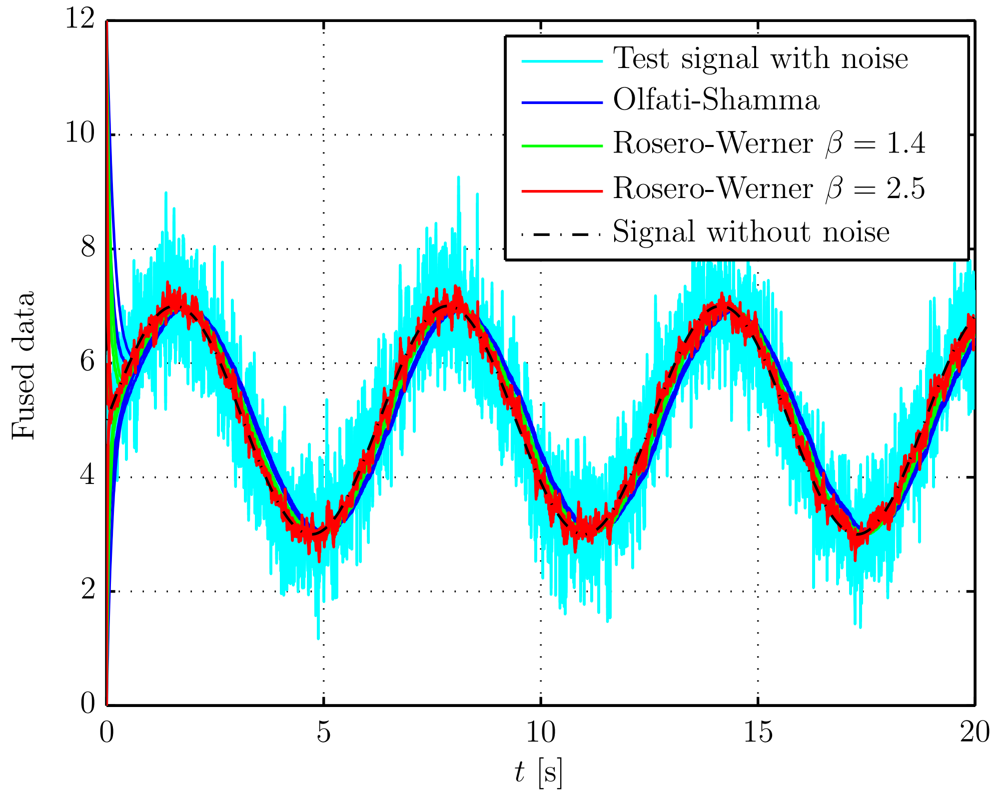


Figure 3.4: Fused sensor data for sensor measurements $r_b(t) + n_i(t)$

the algorithm proposed by (Olfati-Saber and Shamma, 2005). As shown, the network's bandwidth has been modified tuning the parameter β. For $\beta \geq 1$ the convergence velocity increases and the signal tracking error and the convergence region are reduced. Note that for $\beta \gg 1$ the signal tracks the real signal and the noise. For this reason, it is necessary to find a trade-off between noise rejection at high frequency and algorithm's performance. The consensus algorithm is clearly capable of target tracking within the network's bandwidth.

3.4 Unweighted Distributed Gradient Estimation with Noise

Assume that each agent measures both concentration signal $\psi_i(r_i(t)) \in \mathbb{R}$ and position signal $r_i(t) \in \mathbb{R}^p$ corrupted by noise $n_{\psi i}(t) \in \mathbb{R}$ or $n_{ri}(t) \in \mathbb{R}^p$. The sensing models are given by

$$\begin{aligned} u_{\psi i}(t) &= \psi_i(t) + n_{\psi i}(t), \\ u_{ri}(t) &= r_i(t) + n_{ri}(t), \end{aligned} \tag{3.21}$$

where $n_{\psi i} \in \mathbb{R}$ and $n_{ri} \in \mathbb{R}^p$ are zero-mean Gaussian noise, and the pairs $n_{ri}(t)$ - $n_{rj}(t)$ and $n_{\psi i}(t)$ - $n_{\psi j}(t)$ are uncorrelated. In order to estimate the signals, a

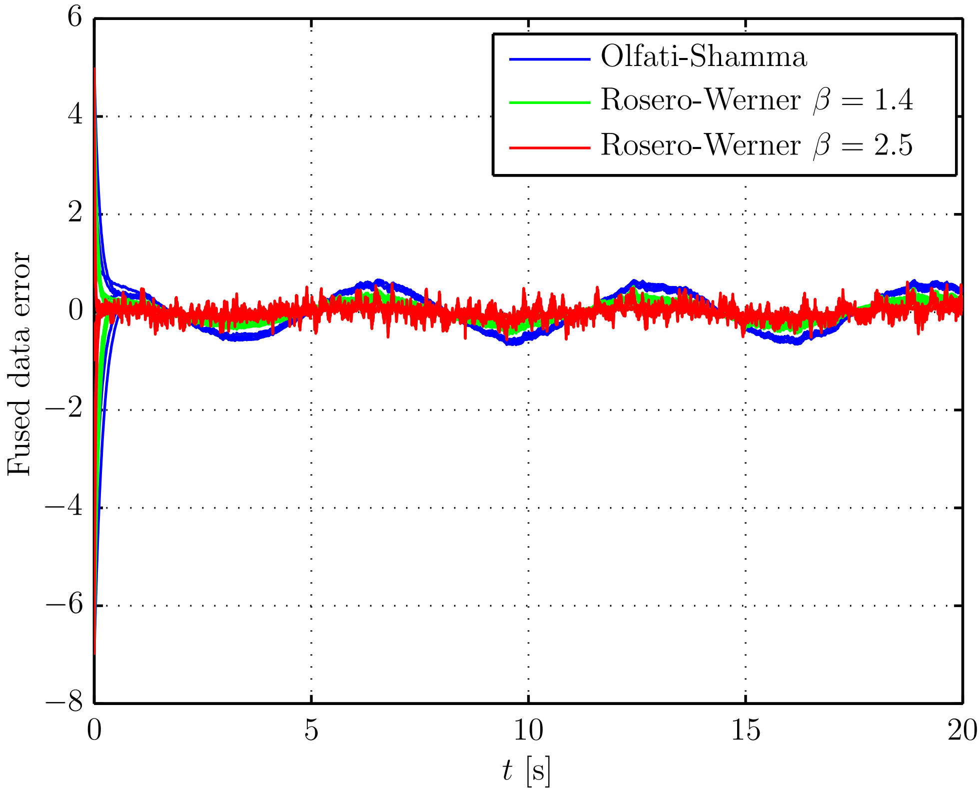


Figure 3.5: Fused sensor data error for sensor measurements $r_b(t) + n_i(t)$

distributed consensus algorithm is proposed as

$$\begin{aligned}\dot{\varphi}_i(t) =&\beta_\varphi \sum_{j\in\mathcal{N}_i} a_{ij}\left(u_{\psi i} - u_{\psi j}\right) - \beta_\varphi \sum_{j\in\mathcal{N}_i} a_{ij}(\varphi_i - \varphi_j) \\ &+ \beta_\varphi(1 + d_i)\left(u_{\psi i} - \varphi_i\right),\end{aligned} \tag{3.22}$$

where $\varphi_i \in \mathbb{R}$ is the target's estimation ψ_i and β_φ is a tuning parameter. To estimate the position signal, the following algorithm is proposed

$$\begin{aligned}\dot{\nu}_i(t) =&\beta_\nu \sum_{j\in\mathcal{N}_i} a_{ij}\left(u_{ri} - u_{rj}\right) - \beta_\nu \sum_{j\in\mathcal{N}_i} a_{ij}(\nu_i - \nu_j) \\ &+ \beta_\nu(1 + d_i)\left(u_{ri} - \nu_i\right),\end{aligned} \tag{3.23}$$

where $\nu_i \in \mathbb{R}^p$ is the estimation of the target r_i and β_ν is a tuning parameter. Each state of sensor i is initialized with $\varphi_i(0) = \varphi_{i0}$ and $\nu_i(0) = \nu_{i0}$. The stability analysis of these algorithms is introduced in (Rosero and Werner, 2014c). Using the outputs of consensus filter, it is possible to compute the slope between agent i and their neighbours $\mathcal{N}_i$ as

$$\underbrace{\begin{bmatrix} \varphi(\nu_i) - \varphi(\nu_1) \\ \vdots \\ \varphi(\nu_i) - \varphi(\nu_{|\mathcal{N}_i|}) \end{bmatrix}}_{\mathsf{b}_i} = \underbrace{\begin{bmatrix} (\nu_i - \nu_1)^T \\ \vdots \\ (\nu_i - \nu_{|\mathcal{N}_i|})^T \end{bmatrix}}_{\mathsf{R}_i} \nabla\varphi(\nu_i), \tag{3.24}$$

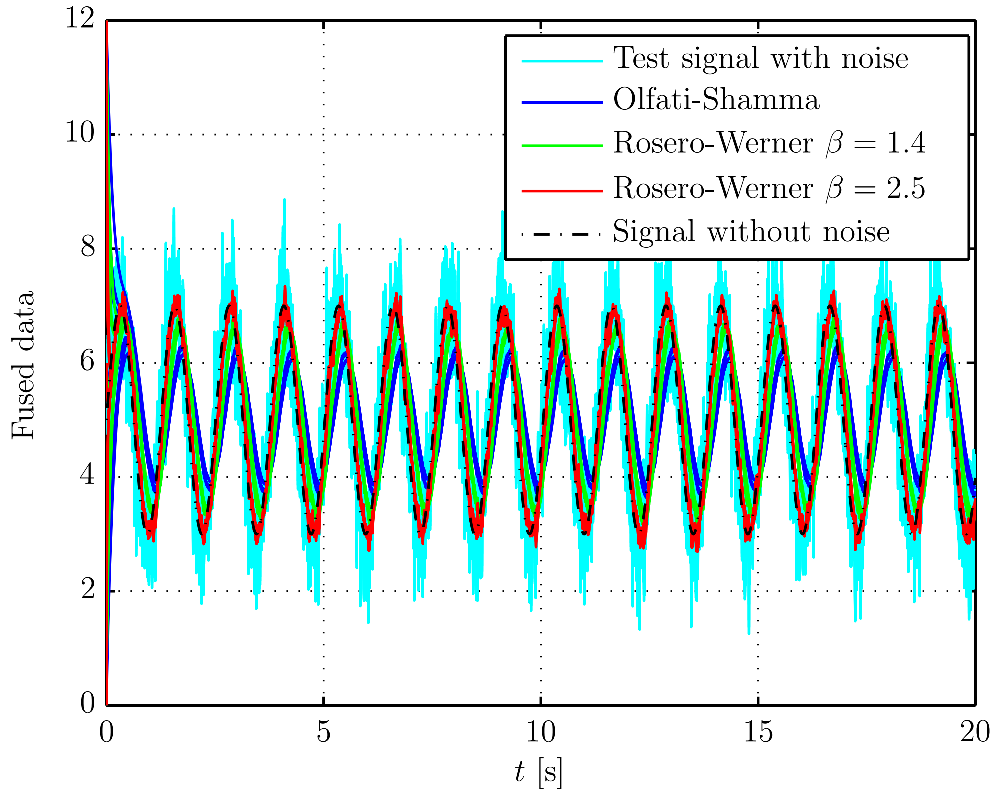


Figure 3.6: Fused sensor data for sensor measurements $r_c(t) + n_i(t)$

where $\mathsf{b}_i \in \mathbb{R}^{|\mathcal{N}_i|\times 1}$, $\mathsf{R}_i \in \mathbb{R}^{|\mathcal{N}_i|\times p}$ and $\nabla\varphi(\nu_i) \in \mathbb{R}^{p\times 1}$. The problem can be solved using least-squares as

$$\hat{g}_{ei} = \left(\mathsf{R}_i^T\mathsf{R}_i\right)^{-1}\mathsf{R}_i^T\mathsf{b}_i. \tag{3.25}$$

3.5 Weighted Distributed Gradient Estimation with Noise

Using both sensing models (Equation 3.21) and the consensus filters for concentration and position signals (Equations 3.22 and 3.23) it is possible to write

$$\mathsf{W}_i \underbrace{\begin{bmatrix} \varphi(\nu_i) - \varphi(\nu_1) \\ \vdots \\ \varphi(\nu_i) - \varphi(\nu_{|\mathcal{N}_i|}) \end{bmatrix}}_{\mathsf{b}_i} = \mathsf{W}_i \underbrace{\begin{bmatrix} (\nu_i - \nu_1)^T \\ \vdots \\ (\nu_i - \nu_{|\mathcal{N}_i|})^T \end{bmatrix}}_{\mathsf{R}_i} \nabla\varphi(\nu_i), \tag{3.26}$$

where $\mathsf{b}_i \in \mathbb{R}^{|\mathcal{N}_i|\times 1}$, $\mathsf{R}_i \in \mathbb{R}^{|\mathcal{N}_i|\times p}$, $\nabla\varphi(\nu_i) \in \mathbb{R}^{p\times 1}$ and $\mathsf{W}_i = \mathrm{diag}\{w_{ij}\} \in \mathbb{R}^{|\mathcal{N}_i|\times|\mathcal{N}_i|}$. Using least-squares estimator, the problem can be solved as

$$\hat{g}_{ei} = \left(\mathsf{R}_i^T\mathsf{W}_i^2\mathsf{R}_i\right)^{-1}\mathsf{R}_i^T\mathsf{W}_i^2\mathsf{b}_i. \tag{3.27}$$

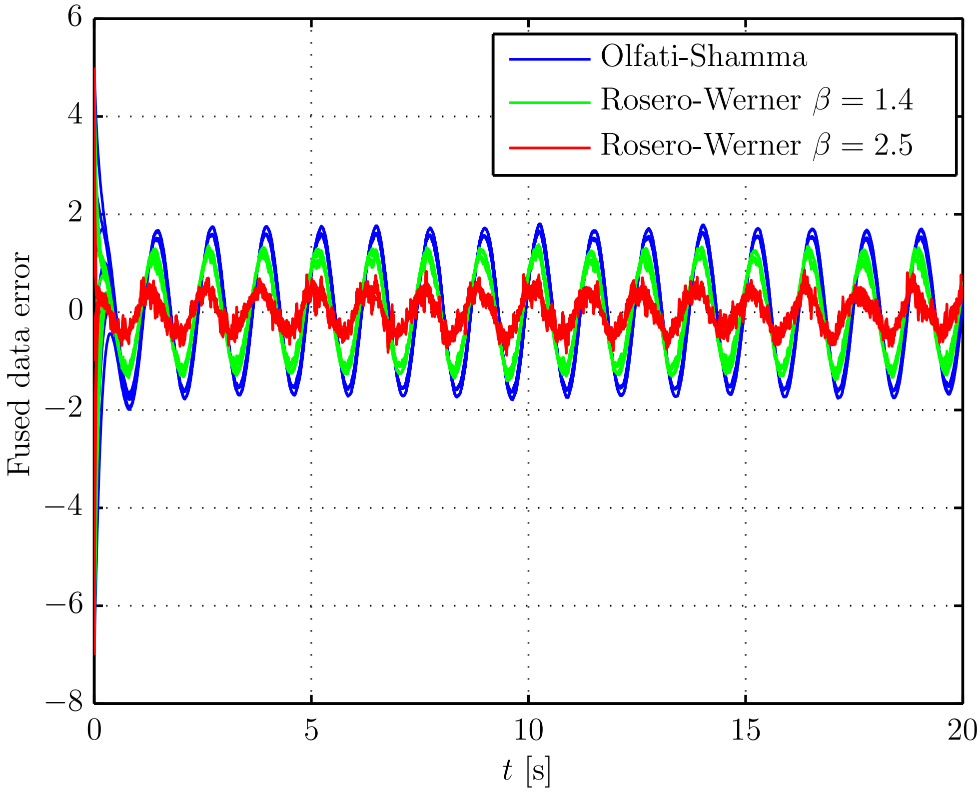


Figure 3.7: Fused sensor data error for sensor measurements $r_c(t) + n_i(t)$

Equations (3.25) and (3.27) can be used to reject noise at higher frequency when concentration and position signals are corrupted by noise.

3.6 Conclusions

In this chapter a distributed consensus filter for sensor networks with time-varying signal tracking has been presented. The filter is designed to run locally on each sensor. It requires information only from agent's local neighbours. Analysis shows that the proposed algorithm is stable and exhibits a better performance than the algorithm proposed by (Olfati-Saber and Shamma, 2005). Simulations confirm the effectiveness of the proposed consensus algorithm.

Chapter 4

Cooperative Source Seeking with Single and Double Integrator Agents

In this chapter the problem of cooperative source seeking by a formation of single and double integrator mobile agents is considered. As mentioned above, each agent is equipped with position and field sensors. Agents are part of a network and exchange information with neighbouring agents through a communication network. Distributed navigation controllers for single and double integrator agents are presented. When the field measurements are corrupted by noise, distributed consensus filters are used in order to reject the noise at high frequencies. Stability conditions are presented. Numerical simulations illustrate the effectiveness of the proposed control law.

4.1 Introduction

Here the source seeking problem as defined in Chapter 1 will be addressed using two simple distributed navigation strategies for single and double integrator models.

The simplest mathematical representation of agents are single and double integrator models. A single integrator model is also called a kinematic agent model because it ignores the lower-level agent dynamics of individual agents; they achieve consensus using only relative position information.

Double integrator models are extended single integrator models that use an additional state to represent the velocity. Double integrator dynamics represent the dynamics of a point mass without friction under the effect of a time-varying force input. Double integrator modelling is required to show how agents reach an agreement not only in their positions but also in their velocities. The double integrator models allow the design of a control law based on acceleration as input and therefore, both position and velocity achieve agreement.

Aim of this Chapter is to demonstrate that, using the presented controllers, agents are able to converge towards the scalar field's source while maintaining the formation.

The chapter is structured as follows. In Sections 4.2 and 4.3, control laws and stability analysis for single and double integrator models are presented. Simulation results illustrate the proposed approach in Section 4.4. Finally, concluding remarks and a reference to the second part of this work are made in Section 4.5.

4.2 Single Integrator Agents

In order to simplify the notation, in the following sections only the case $p = 1$ will be considered. The analysis can be extended to a higher dimension p rewriting the equations in terms of Kronecker product.

Agents modelled by single integrator dynamics are described by

$$\dot{r}_i(t) = u_i(t), \tag{4.1}$$

where $i = 1, ..., N$, $r_i(t) \in \mathbb{R}^p$ is the position vector and $u_i(t) \in \mathbb{R}^p$ its control input. The following distributed control law is considered

$$u_i(t) = k_F \sum_{j \in \mathcal{N}_i} a_{ij} \left[(r_{Fi}(t) - r_{Fj}(t)) - (r_i(t) - r_j(t)) \right] + k_T \hat{g}_i(r_i(t)), \tag{4.2}$$

where

$$\hat{g}_i(r_i) = \left(\sum_{j \in \mathcal{N}_i} a_{ij} (r_i - r_j)(r_i - r_j)^T \right)^{-1} \left(\sum_{j \in \mathcal{N}_i} a_{ij} (r_i - r_j)(\psi(r_i) - \psi(r_j)) \right)$$

is the distributed estimated gradient computed in each agent i in Chapter 2. $k_T > 0$ and $k_F > 0$ are scalar tuning parameters, $u_{Ti}(t) = k_T \hat{g}_i(r_i)$ denotes the control input used to track the direction of the gradient for agent i, $\hat{g}_i$ denotes the estimated gradient computed by Equations (2.3) and (3.25), $u_{Fi}(t) = k_F \sum_{j \in \mathcal{N}_i} [r_{Fi}(t) - r_{Fj}(t)) - (r_i(t) - r_j(t)]$ denotes the formation control input for agent i and r_{Fi} denotes a reference signal for agent i.

Let $r(t) = [r_1^T(t), \ldots, r_N^T(t)]^T$, $r_F(t) = [r_{F1}^T(t), \ldots, r_{FN}^T(t)]^T$, $\hat{g}(t) = [\hat{g}_1^T(t), \ldots, \hat{g}_N^T(t)]^T$, and use the definition of the Laplacian $\mathcal{L}$, then it is possible to write

$$\dot{r} = k_F \mathcal{L}(r_F - r) + k_T \hat{g}(r). \tag{4.3}$$

Note that if $\hat{g}(r) = 0$ and $k_F = 1$, Equation (4.3) is reduced to the well known results about consensus for agents' formation presented by (Mesbahi and Egerstedt, 2010; Olfati-Saber and Murray, 2004). Consensus is reached asymptotically with a rate of convergence $\lambda_2(\mathcal{G})$, if and only if the graph $\mathcal{G}$ is connected.

Let $\bar{r}(t) = \frac{1}{N} \sum_{i \in \mathcal{V}} r_i(t) = \frac{1}{N} \mathbf{1}^T r(t)$ be the average of the position states, and $\bar{g}(r(t)) = \frac{1}{N} \sum_{i \in \mathcal{V}} \hat{g}_i(r(t)) = \frac{1}{N} \mathbf{1}^T \hat{g}(r(t))$ be the average of the estimated gradient.

Since $\mathbf{1}^T\mathcal{L} = \mathbf{0}^T$, the time derivative of the position states' average $\bar{r}(t)$ is given by

$$\dot{\bar{r}}(t) = \frac{1}{N}\mathbf{1}^T\dot{r}(t) = \frac{k_T}{N}\mathbf{1}^T\hat{g} = k_T\bar{\hat{g}}, \tag{4.4}$$

with initial average $\bar{r}_0 = \bar{r}(0) = \frac{1}{N}\mathbf{1}^T r(0)$ and $\bar{\hat{g}}_0 = \bar{\hat{g}}(r(0)) = \frac{1}{N}\mathbf{1}^T\hat{g}(r(0))$. Therefore it holds that $\bar{r}(t) = \bar{r}(0) + k_T\int_0^t \bar{\hat{g}}(\tau, r(\tau))d\tau$ for all $t \geq 0$. Note that the agents move inside a scalar field with the varying velocity $k_T\bar{\hat{g}}$ in the gradient direction. If the estimated gradient $\hat{g}_i$ for each agent i is normalized as $\hat{g}_i^N = \frac{\hat{g}_i}{\|\hat{g}_i\|}$, the agents' velocity will be constant and defined by the parameter k_T.

The equilibrium point of Equation (4.3) is given by

$$k_F\mathcal{L}r^* = k_F\mathcal{L}r_F + k_T\hat{g}(r^*), \tag{4.5}$$

where $r^* = [r_1^{*T}, \ldots, r_N^{*T}]^T$. The position's average of the equilibrium points are given by

$$\dot{\bar{r}}(t) = 0 = \bar{\hat{g}}(r^*). \tag{4.6}$$

Note that Equation (2.3) can be rewritten as $\hat{g}_i = \left(r_p^T\bar{\mathcal{L}}_i r_p\right)^{-1}\left(r_p^T\bar{\mathcal{L}}_i\psi\right)$, where $r_p = [r_{p1}, \ldots, r_{pN}]^T$, $\psi = [\psi_1, \ldots, \psi_N]^T$, $r_{pi} = [r_{xi}\ r_{yi}]^T$ and $\bar{\mathcal{L}}_i$ is the Laplacian matrix for the tree generated by agent i only with neighbours $\mathcal{N}_i$. $\bar{\mathcal{L}}_i$ has only one zero eigenvalue, the same eigenvalue of $\mathcal{L}$. In equilibrium, the average of Equation (4.5) can be written as

$$\sum_{i=1}^{N}\hat{g}_i^* = \sum_{i=1}^{N}\left(r_p^{*T}\bar{\mathcal{L}}_i r_p^*\right)^{-1}\left(r_p^{*T}\bar{\mathcal{L}}_i\psi^*\right) = 0. \tag{4.7}$$

When the gradient's average is zero, agents are in an equilibrium point, i.e., $\dot{\bar{r}}(t) = \bar{\hat{g}} = 0$, and they do not move anymore.

For single integrator agents the following lemma is used.

Lemma 1. *((Seyboth et al., 2011)). Suppose $\mathcal{L}$ is the Laplacian of an undirected connected graph $\mathcal{G}$. Then, $\forall\ t \geq 0$ and $\mathsf{v} \in \mathbb{R}^N$ with $\mathbf{1}^T\mathsf{v} = 0$, it is valid that $\|e^{\mathcal{L}t}\mathsf{v}\| \leq e^{-\lambda_2 t}\|\mathsf{v}\|$.*

In order to discuss the convergence, the state $r(t)$ can be decomposed as

$$r(t) = \frac{1}{N}\mathbf{1}\mathbf{1}^T r(0) + \delta(t),$$

where $\mathbf{1}$ is a vector of ones and δ is referred to as disagreement vector. By definition, the disagreement vector has zero average, i.e., $\mathbf{1}^T\delta(t) \equiv 0$ (Olfati-Saber and Murray, 2004).

In order to solve the source seeking problem for single integrator agents, theorem 2 is presented. This theorem recalls assumptions 1 and 2 which have been presented in Chapter 1.

Theorem 2. *Consider the multi-agent system (4.1) with control law (4.2). Suppose that assumptions 1 and 2 are fulfilled. Then, for all $r_i(0) \in \mathbb{R}^p$ and $t \geq 0$, agents locate the unknown source's position r_s of the scalar field $\psi(r)$ and the disagreement vector δ of the closed-loop system converges to a ball centred at the origin with radius*

$$\epsilon = \frac{2k_T\sqrt{N}\mu}{k_F\lambda_2}, \tag{4.8}$$

with the distance between the formation's center of mass and the source of the scalar field

$$\|\bar{r}^* - r_s\| \leq \frac{2}{L_H}\left(e_0 + L_H\frac{h}{2}\right). \tag{4.9}$$

Proof. Considering the position error $e_p(t) = r(t) - r^*$ and using Equation (4.5), the dynamic error can be written as

$$\dot{e}_p(t) = -k_F\mathcal{L}e_p + k_Te_g, \tag{4.10}$$

where the gradient error is given by

$$e_g = \hat{g}(r) - \hat{g}(r^*). \tag{4.11}$$

The state $e_p(t)$ can be decomposed according to

$$\delta(t) = e_p(t) - \mathbf{1}\bar{e}_p(t), \tag{4.12}$$

where $\delta(t)$ is the disagreement vector of the multi-agent system and $\bar{e}_p = \frac{1}{N}\mathbf{1}^Te_p(t) = \bar{r}(t) - \bar{r}^*$. By definition the disagreement vector has zero average, i.e., $\mathbf{1}^T\delta(t) \equiv 0$, following the notation of (Olfati-Saber and Murray, 2004). Since the reference of the formation is fixed, the disagreement dynamics are given by

$$\begin{aligned}\dot{\delta}(t) &= \dot{e}_p(t) - \mathbf{1}\dot{\bar{e}}_p(t),\\ &= -k_F\mathcal{L}\delta + k_T\mathcal{M}e_g,\end{aligned} \tag{4.13}$$

where $\mathcal{M} = I_N - \frac{1}{N}\mathbf{1}\mathbf{1}^T$, initial conditions $\delta(0) = e_p(0) - \mathbf{1}\bar{e}_p(0)$ and $\dot{r}_F = 0$. Therefore, the solution is given by

$$\delta(t) = e^{-k_F\mathcal{L}t}\delta(0) + \int_0^t e^{-k_F\mathcal{L}(t-\tau)}k_T\mathcal{M}e_gd\tau,$$

where e_g is a function of δ and the disagreement vector is bounded by

$$\|\delta(t)\| \leq \|e^{-k_F\mathcal{L}t}\delta(0)\| + \int_0^t \|e^{-k_F\mathcal{L}(t-\tau)}k_T\mathcal{M}e_g\|d\tau.$$

Since $\mathcal{M}\hat{g}$ has zero average, Lemma 1 can be applied. This yields

$$\|\delta(t)\| \leq e^{-k_F\lambda_2 t}\|\delta(0)\| + k_T\|\mathcal{M}\| \int_0^t e^{-k_F\lambda_2(t-\tau)}\|e_g\|d\tau,$$

where $\lambda_2 = \lambda(\mathcal{G})$ and $\|\mathcal{M}\| = 1$. Since $\|\hat{g}_i\| \leq \mu$, we have $\|\hat{g}\| \leq \sqrt{N}\mu$ and $\|e_g\| \leq \|\hat{g}(e_p + r^*) - \hat{g}(r^*)\| \leq 2\sqrt{N}\mu$; and it follows that

$$\|\delta(t)\| \leq \frac{2k_T\sqrt{N}\mu}{k_F\lambda_2} + \left(\|\delta(0)\| - \frac{2k_T\sqrt{N}\mu}{k_F\lambda_2}\right) e^{-k_F\lambda_2 t}. \tag{4.14}$$

This estimate shows that the zero-input response of the disagreement vector decays to zero exponentially fast, while the zero-state response is bounded for every bounded input.

The derivative of the position error's average is given by

$$\dot{\bar{e}}_p = \dot{\bar{r}}(t) - \dot{\bar{r}}^* = k_T\bar{e}_g(e_p), \tag{4.15}$$

where $\bar{e}_g = \frac{1}{N}\mathbf{1}^T\left(\hat{g}(e_p + r^*) - \hat{g}(r^*)\right)$, $\bar{e}_g$ satisfies $\bar{e}_g(0) = 0$, and $\bar{e}_p^T\bar{e}_g(e_p) < 0$ for all $e_p \neq 0$. Then the equilibrium point r^* is stable because the agents starting on either side of the source will have to move towards the scalar field's source due to the sign of the derivative $\dot{\bar{e}}_p$. In order to arrive at the same conclusion, consider the system $\dot{\bar{e}}_p = -f(e_p)$, where $f(e_p) = k_T\bar{e}_g(e_p)$ and satisfies $f(0) = 0$ and $\bar{e}_p^T(e_p)f(e_p) > 0$ for all $e_p \neq 0$. Consider the Lyapunov function candidate $V = \int_0^{\bar{e}_p} f^T(z)dz$, V is continuously differentiable, $V(0) = 0$, and $V > 0$ for all $e_p \neq 0$. Then $\dot{V} = -f^T(e_p)f(e_p) = -k_T\bar{e}_g^T(e_p)\bar{e}_g(e_p) < 0$. Therefore, it can be concluded that the equilibrium is stable.

At this point, it is necessary to compute the distance between the scalar field's source and the formation's center of mass. Fig. 4.1 presents some ideas about a possible equilibrium point for agents near to the scalar field's source r_s, the formation's center of mass $\bar{r}^*$, and the position r_{av}^* when the estimated gradient's average is zero, $\bar{\hat{g}}(r_{av}^*) = 0$. When $\bar{\hat{g}}(r_{av}^*) = 0$, agents stop and maintain the formation.

To find the position error between the scalar field's source and the formation's center of mass $\|\bar{r}^* - r_s\|$ the Taylor expansion is used. It is assumed that $m \leq \|\nabla^2\|\psi(x) \leq L_H$ near to the source. The Taylor expansion of $\psi(r)$ at $\bar{r}^*$ yields

$$\psi(r) = \psi(\bar{r}^*) + \nabla\psi(\bar{r}^*)^T(r - \bar{r}^*) + \frac{1}{2}(r - \bar{r}^*)^T\nabla^2\psi(s_1)(r - \bar{r}^*), \tag{4.16}$$

for some s_1 on the line segment between $\bar{r}$ and r. For the equilibrium of the formation's center of mass $r = r_s$, this equation yields

$$\psi(r^*) = \psi(\bar{r}^*) + \nabla\psi(\bar{r}^*)^T(r^* - \bar{r}^*) + \frac{1}{2}(r^* - \bar{r}^*)^T\nabla^2\psi(x_1)(r^* - \bar{r}^*). \tag{4.17}$$

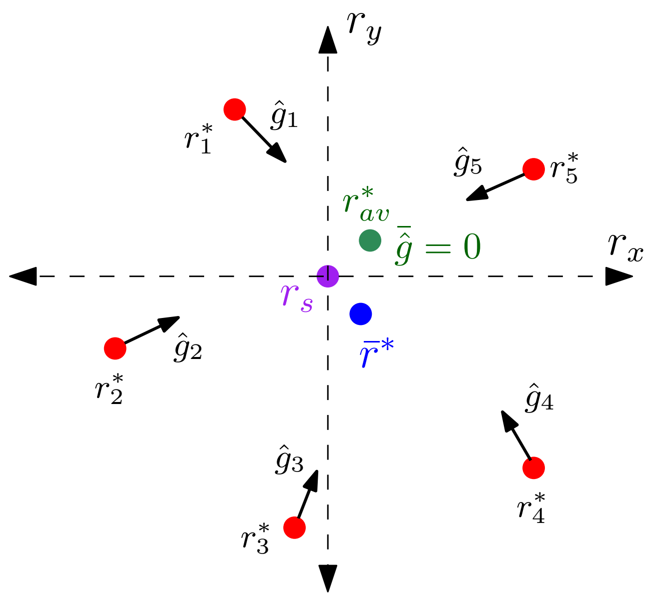


Figure 4.1: Agents in equilibrium

The upper bound on the Hessian $\|\nabla^2\psi(x)\| \leq L_H$, implies

$$\psi(r_s) \geq \psi(\bar{r}^*) + \nabla\psi(\bar{r}^*)^T(r_s - \bar{r}^*) + \frac{1}{2}L_H\|r_s - \bar{r}^*\|^2, \tag{4.18}$$

and

$$\psi(r_s) \geq \psi(\bar{r}^*) - \|\nabla\psi(\bar{r}^*)\|\|r_s - \bar{r}^*\| + \frac{1}{2}L_H\|r_s - \bar{r}^*\|^2. \tag{4.19}$$

Since $\psi(r_s) \geq \psi(\bar{r}^*)$, it holds that

$$-\|\nabla\psi(\bar{r}^*)\|\|r_s - \bar{r}^*\| + \frac{1}{2}L_H\|r_s - \bar{r}^*\|^2 \leq 0, \tag{4.20}$$

then

$$\|r_s - \bar{r}^*\| \leq \frac{2}{L_H}\|\nabla\psi(\bar{r}^*)\|. \tag{4.21}$$

A second order approximation of the gradient evaluated at the formation's center of mass $\nabla\psi(\bar{r}^*)$ can be written as

$$\begin{aligned}\nabla\psi(\bar{r}^*) &= \nabla\psi(r_{av}^*) + \nabla^2\psi(s_1)(\bar{r}^* - r_{av}^*), \\ &= \bar{\hat{g}}(r_{av}^*) - \bar{e}_\nabla(r_{av}^*) + \nabla^2\psi(s_1)(\bar{r}^* - r_{av}^*),\end{aligned} \tag{4.22}$$

where $\nabla\psi(r_{av}^*) = \bar{\nabla}\psi(r_{av}^*) = \bar{\hat{g}}(r_{av}^*) - \bar{e}_\nabla(r_{av}^*)$ and $\bar{\hat{g}}(r_{av}^*) = 0$. Taking the norm, it is possible to write

$$\|\nabla\psi(\bar{r}^*)\| \leq e_0 + L_H\frac{h}{2}, \tag{4.23}$$

since the estimated error is bounded by $\|\bar{e}_\nabla\| \leq e_0$, $\|\nabla^2\psi(s_1)\| \leq L_H$ and $\|\bar{r}^* - r^*_{av}\| \leq \frac{h}{2}$. Then the distance between the scalar field's source and the formation's center of mass is bounded by

$$\|r_s - \bar{r}^*\| \leq \frac{2}{L_H}\left(e_0 + L_H\frac{h}{2}\right). \tag{4.24}$$

Note that e_0 depends on the square of the maximum distance between agents and the bound of Hessian matrix. In order to reduce the estimated error, the distance between agents must be sufficiently small.

□

Remark 2. *If the estimated gradient is normalized,* $\hat{g}^N(t) = \left[\frac{\hat{g}_1^T(t)}{\|\hat{g}_1(t)\|}, \ldots, \frac{\hat{g}_N^T(t)}{\|\hat{g}_N(t)\|}\right]^T$, *then* $\|\frac{\hat{g}_i}{\|\hat{g}_i\|}\| = 1$, $\mu = 1$ *and* $\|\hat{g}^N\| = \sqrt{N}$. *Thus, the convergence region is given by*

$$\|\delta(t)\| \leq \frac{2k_T\sqrt{N}}{k_F\lambda_2}, \tag{4.25}$$

and the agents' average velocity is given by $\|\dot{\bar{r}}(t)\| \leq k_T$.

Remark 3. *Note that the previous result is given for* $r_s \neq \bar{r}^* \neq r_{av}$. *If* $r^*_{av} = \bar{r}^*$, *then* $h = \max_{j\in\mathcal{N}_i}\|\bar{r}^* - r^*_{av}\| = 0$, *and* $\|\nabla\psi(\bar{r}^*)\| \leq e_0$. *Consequently, error is bounded by*

$$\|r_s - \bar{r}^*\| \leq \frac{2}{L_H}e_0. \tag{4.26}$$

If $r^*_{av} = \bar{r}^* = r_s$, *then* $h = \max_{j\in\mathcal{N}_i}\|\bar{r}^* - r^*_{av}\| = 0$, $\nabla\psi(\bar{r}^*) = 0$, *and then* $\|r_s - \bar{r}^*\| = 0$, *i.e., the formation's center of mass is exactly at the scalar field's source and the distance between this source and the formation's center of mass is zero.*

Remark 4. *Note that since* $\|\hat{g}_i\| \leq \mu$, *the convergence region is conservative.*

4.3 Double Integrator Agents

Agents modelled by double integrator dynamics are described by

$$\begin{aligned}\dot{r}_i(t) &= v_i(t),\\ \dot{v}_i(t) &= u_i(t),\end{aligned} \tag{4.27}$$

where $i = 1, ..., N$, $r_i(t) \in \mathbb{R}^p$ is the position vector, $v_i \in \mathbb{R}^p$ is the velocity vector, and $u_i(t) \in \mathbb{R}^p$ is the control input. Again, agents are controlled in a distributed fashion, i.e., $u_i(t)$ depends only on information from its neighbours $j \in \mathcal{N}_i$.

The following distributed control law is considered

$$\begin{aligned}u_i(t) =& k_F\sum_{j\in\mathcal{N}_i} a_{ij}[(r_{Fi}(t) - r_{Fj}(t)) - (r_i(t) - r_j(t)) - \theta(v_i(t) - v_j(t))]\\ &+ k_T\left(\hat{g}_i(r_i(t)) - \gamma v_i(t)\right),\end{aligned} \tag{4.28}$$

where

$$\hat{g}_i(r_i) = \left(\sum_{j\in\mathcal{N}_i} a_{ij}(r_i - r_j)(r_i - r_j)^T\right)^{-1} \left(\sum_{j\in\mathcal{N}_i} a_{ij}(r_i - r_j)(\psi(r_i) - \psi(r_j))\right)$$

is the distributed estimated gradient computed in each agent i in Chapter 2, $k_F > 0$, $k_T > 0$, $\gamma > 0$ and $\theta > 0$ are scalar control parameters in the formation, $u_{Ti}(t) = k_T(\hat{g}_i(t) - \gamma v_i(t))$ denotes the control input tracking the gradient for agent i, $u_{Fi}(t) = k_F \sum_{j\in\mathcal{N}_i} a_{ij}[(r_{Fi}(t) - r_{Fj}(t)) - (r_i(t) - r_j(t)) - \theta(v_i(t) - v_j(t))]$ denotes the formation control input for agent i, $r_{Fi}(t)$ denotes a formation's desired reference signal for agent i, and $\hat{g}_i(t)$ is the estimated gradient for agent i computed by Equations (2.3) and (3.25).

Let $r(t) = [r_1^T(t), \ldots, r_N^T(t)]^T$, $r_F(t) = [r_{F1}^T(t), \ldots, r_{FN}^T(t)]^T$, $v(t) = [v_1^T(t), \ldots, v_N^T(t)]^T$, and $\hat{g}(t) = \left[\hat{g}_1^T(t), \ldots, \hat{g}_N^T(t)\right]^T$, then the closed-loop dynamics can be written as

$$\begin{bmatrix} \dot{r} \\ \dot{v} \end{bmatrix} = \Sigma \begin{bmatrix} r \\ v \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ k_F\mathcal{L} & k_T I_N \end{bmatrix} \begin{bmatrix} r_F \\ \hat{g}(r) \end{bmatrix}, \tag{4.29}$$

where

$$\Sigma = \begin{bmatrix} 0 & I_N \\ -k_F\mathcal{L} & -k_F\theta\mathcal{L} - k_T\gamma I_N \end{bmatrix}.$$

Note that if $\hat{g}(r) = 0$, $k_T = 1$ and $k_F = 1$, Equation (4.29) is reduced to a well known result on agents' formation presented by (Ren and Atkins, 2007).

Let $\bar{r}(t) = \frac{1}{N}\sum_{i\in\mathcal{V}} r_i(t) = \frac{1}{N}\mathbf{1}^T r(t)$ be the average of the position states, $\bar{v}(t) = \frac{1}{N}\sum_{i\in\mathcal{V}} v_i(t) = \frac{1}{N}\mathbf{1}^T v(t)$ be the average of the velocity states, and $\bar{\hat{g}}(r(t)) = \frac{1}{N}\sum_{i\in\mathcal{V}} \hat{g}_i(r(t)) = \frac{1}{N}\mathbf{1}^T\hat{g}(r(t))$ be the average of the estimated gradient. Since $\mathbf{1}^T\mathcal{L} = \mathbf{0}^T$, the time derivative of both the position states average $\bar{r}$ and the velocity states average $\bar{v}$ is given by

$$\begin{aligned} \dot{\bar{r}}(t) &= \bar{v}(t), \\ \dot{\bar{v}}(t) &= -k_T\gamma\bar{v}(t) + k_T\bar{\hat{g}}(r), \end{aligned} \tag{4.30}$$

with initial average $\bar{r}_0 = \bar{r}(0) = \frac{1}{N}\mathbf{1}^T r(0)$, $\bar{v}_0 = \bar{v}(0) = \frac{1}{N}\mathbf{1}^T v(0) = 0$ and $\bar{\hat{g}}_0 = \bar{\hat{g}}(r(0)) = \frac{1}{N}\mathbf{1}^T\hat{g}(r(0))$. Note that agents move with varying velocity into the scalar field following the gradient direction. If the estimated gradient $\hat{g}_i$ for each agent i is normalized as $\hat{g}_i^N = \frac{\hat{g}_i}{\|\hat{g}_i\|}$, agents' velocity will be constant.

Let σ be the eingenvalues of matrix Σ. With the purpose of analysing the consensus of Equation (4.29), first the Equation $det(\sigma I_{2N} - \Sigma) = 0$ has to be solved to find the eigenvalues of Σ. It holds that

$$det(\sigma I_{2N} - \Sigma) = det\left(\sigma^2 I_N + (k_T\gamma I_N + k_F\theta\mathcal{L})\sigma + k_F\mathcal{L}\right).$$

Then, the eigenvalues of Σ are given by

$$\sigma_{i1,\ i2} = \frac{-(k_T\gamma - k_F\theta\lambda_i) \pm \sqrt{(k_T\gamma - k_F\theta\lambda_i)^2 + 4k_F\lambda_i}}{2},$$

where $i = 1, \ldots, N$, σ_{i1} and σ_{i2} are the eigenvalues of Σ that are associated with eigenvalue λ_i of the Laplacian matrix $\mathcal{L}$.

Since $\lambda_1 = 0$, two eigenvalues $\sigma_{11} = 0$ and $\sigma_{12} = -k_T\gamma$ are obtained. Then, Σ has one zero eigenvalue because $\mathcal{L}$ has one zero eigenvalue, and all the other eigenvalues have negative real parts. Note that Σ can be written in Jordan canonical form as

$$\begin{aligned}\Sigma =& PJP^{-1},\\ =& \begin{bmatrix}\mathsf{w}_1, \ldots, \mathsf{w}_{2N}\end{bmatrix} \begin{bmatrix} 0 & 0_{1\times(2N-1)} \\ 0_{(2N-1)\times 1} & J' \end{bmatrix} \begin{bmatrix} \mathsf{v}_1^T \\ \vdots \\ \mathsf{v}_{2N}^T \end{bmatrix},\end{aligned}$$

where J' is the Jordan upper diagonal block matrix corresponding to $2N-1$ non-zero eigenvalues and w_i, $\mathsf{v}_i \in \mathbb{R}^{2N} for\ i = 1, \ldots, 2N$. w_i and v_i can be chosen to be the right and left generalized eigenvectors of Σ, respectively corresponding to the eigenvalue zero. It is well-known that an undirected graph is connected if and only if the second smallest eigenvalue is larger than zero. Since Σ has only one zero eigenvalue, $\mathcal{L}$ has a simple zero eigenvalue. It implies that there exists a non-negative vector p such that $\mathsf{p}^T\mathcal{L} = 0$ and $\mathsf{p}^T\mathbf{1} = 1$. Without loss of generality, $\mathsf{w}_1 = [\mathbf{1}^T\ \mathbf{0}^T]^T$ is chosen as the right eigenvector and $\mathsf{v}_1 = [\mathsf{p}^T\ \frac{1}{k_T\gamma}\mathsf{p}^T]^T$ is chosen as the left eigenvector corresponding to the eigenvalue zero, where $\mathsf{v}_1^T\mathsf{w}_1 = 1$ and $\mathsf{p} = \frac{1}{N}\mathbf{1}$. Since all other eigenvalues of Σ have negative real parts, it follows that

$$\begin{aligned}\lim_{t\to\infty} e^{\Sigma t} =& \lim_{t\to\infty} Pe^{Jt}P^{-1},\\ =& \frac{1}{N}\begin{bmatrix} \mathsf{w}_1\mathsf{v}_1^T & 0_{N\times N} \\ 0_{N\times N} & 0_{N\times N}\end{bmatrix}.\end{aligned}$$

Given any initial positions and velocities $r(0) = r_0$ and $v(0) = v_0$ respectively, $r_F = 0$ and $\hat{g} = 0$, it follows that

$$\lim_{t\to\infty}\begin{bmatrix} r(t) \\ v(t)\end{bmatrix} = \frac{1}{N}\begin{bmatrix}\mathbf{1}\mathbf{1}^T & \frac{1}{k_T\gamma}\mathbf{1}\mathbf{1}^T \\ 0 & 0\end{bmatrix}\begin{bmatrix} r(0) \\ v(0)\end{bmatrix},$$

then $\lim_{t\to\infty} r(t) = \frac{1}{N}\mathbf{1}\mathbf{1}^T r(0) + \frac{1}{k_T\gamma N}\mathbf{1}\mathbf{1}^T v(0)$ and $\lim_{t\to\infty} v(t) = 0$. It is straightforward to see that consensus is achieved if and only if the graph is connected, $k_F > 0$, $k_T > 0$, $\gamma > 0$, $\theta > \max_{2\leq i\leq N}\frac{k_T\gamma}{k_F\lambda_i}$ and the graph is connected. Similar results are obtained applying a procedure based on (Ren and Atkins, 2007; Zhu, 2011). Thus, matrix Σ has exactly one zero eigenvalue and only one independent eigenvector corresponding to the zero eigenvalue.

To order the eigenvalues of matrix Σ, let $\kappa_1 = \sigma_{11}$, $\kappa_2 = \sigma_{12}$, $\kappa_3 = \sigma_{21}$, and so forth. In consequence the eigenvalues of Σ can be organised as $-\kappa_{2N}(\Sigma) \leq \cdots \leq -\kappa_2(\Sigma) < \kappa_1(\Sigma) = 0$. Then for all $t \geq 0$ and all vectors $\mathsf{v} \in \mathbb{R}^{2N}$ with $\mathsf{w}_1^T\mathsf{v} = 0$ and $\mathsf{v}_1^T\mathsf{v} = 0$, it holds that

$$\|e^{\Sigma t}\mathsf{v}\| \leq e^{-\kappa_2 t}c_d\|\mathsf{v}\|, \tag{4.31}$$

where $c_d = \|P^{-1}\|\|P\|$.

The equilibrium point of Equation (4.29) is given by

$$\begin{aligned} v^* =& 0 \\ k_F \mathcal{L} r^* =& k_F \mathcal{L} r_F + k_T \hat{g}(r^*), \end{aligned} \tag{4.32}$$

where $r^* = [r_1^{*T}, \ldots, r_N^{*T}]^T$. The position's and velocity's average of the equilibrium points are given by

$$\dot{\bar{r}}(t) = \bar{v}(t) = \dot{\bar{v}}(t) = 0 = \bar{\hat{g}}. \tag{4.33}$$

In equilibrium, the average of Equation (4.32) is defined as

$$\sum_{i=1}^{N} \hat{g}_i^* = 0 = \sum_{i=1}^{N} \left(r_p^{*T} \bar{\mathcal{L}}_i r_p^*\right)^{-1} \left(r_p^{*T} \bar{\mathcal{L}}_i \psi^*\right). \tag{4.34}$$

When the average of the gradient is zero, agents are in an equilibrium point, i.e., $\bar{\hat{g}} = 0$ and they do not move anymore.

In order to solve the source seeking problem for double integrator agents, theorem 3 is presented. This theorem recalls assumptions 1 and 2 which have been presented in Chapter 1.

Theorem 3. *Consider the multi-agent system (4.27) with control law (4.28). Suppose that assumptions 1 and 2 are fulfilled. Then, for all $r_i(0) \in \mathbb{R}^p$ and $t \geq 0$, agents locate the unknown source's position r_s of the scalar field $\psi(r)$ and the disagreement vector δ of the closed-loop system converges to a ball centred at the origin with radius*

$$\epsilon = \frac{2 c_d k_T \sqrt{N} \mu}{\kappa_2}, \tag{4.35}$$

with a distance between the formation's center of mass and the scalar field's source

$$\|r_s - \bar{r}^*\| \leq \frac{2}{L_H}\left(e_0 + L_H \frac{h}{2}\right). \tag{4.36}$$

Proof. Consider the position error $e_p(t) = r(t) - r^*$ and velocity error $e_v(t) = v(t) - v^*$. Using Equation (4.32), the error is governed by

$$\begin{bmatrix} \dot{e}_p \\ \dot{e}_v \end{bmatrix} = \Sigma \begin{bmatrix} e_p \\ e_v \end{bmatrix} + k_T \begin{bmatrix} 0 \\ e_g \end{bmatrix}, \tag{4.37}$$

where $e_g = \hat{g}(r) - \hat{g}(r^*)$. Define the average of the position error states $\bar{e}_p(t) = \frac{1}{N}\sum_{i\in\mathcal{V}} e_{pi}(t) = \frac{1}{N}\mathbf{1}^T e_p(t) = \bar{r} - \bar{r}^*$, and the average of the velocity error states $\bar{e}_v(t) = \frac{1}{N}\sum_{i\in\mathcal{V}} e_{vi}(t) = \frac{1}{N}\mathbf{1}^T e_v(t) = \bar{v} - \dot{\bar{r}}^*$. Then, the state vector can be decomposed according to

$$\begin{aligned} e_p(t) &= \mathbf{1}\bar{e}_p(t) + \delta_p(t), \\ e_v(t) &= \mathbf{1}\bar{e}_v(t) + \delta_v(t), \end{aligned} \tag{4.38}$$

such that the disagreement vectors $\delta_p(t)$ and $\delta_v(t)$ have zero average, i.e., $\mathbf{1}^T\delta_p(t) \equiv \mathbf{1}^T\delta_v(t) \equiv 0$. Derivation of Equation (4.38) with respect to time t and considering that $v(0) = 0$, yields

$$\begin{bmatrix} \dot{\delta}_p \\ \dot{\delta}_v \end{bmatrix} = \Sigma \begin{bmatrix} \delta_p \\ \delta_v \end{bmatrix} + k_T \begin{bmatrix} 0 \\ \mathcal{M} \end{bmatrix} e_g, \tag{4.39}$$

where $\mathcal{M} = I_N - \frac{1}{N}\mathbf{1}\mathbf{1}^T$. With $\delta(t) = [\delta_p(t)^T \;\; \delta_v(t)^T]^T$ and $\Psi = \begin{bmatrix} 0 \\ \mathcal{M} \end{bmatrix}$, the disagreement dynamics are given by

$$\dot{\delta}(t) = \Sigma\delta(t) + \Psi e_g \tag{4.40}$$

and the solution is given by

$$\delta(t) = e^{\Sigma t}\delta(0) + k_T \int_0^t e^{\Sigma(t-\tau)}\Psi e_g d\tau.$$

Then the disagreement vector is bounded by

$$\|\delta(t)\| \leq \|e^{\Sigma t}\delta(0)\| + k_T \int_0^t \|e^{\Sigma(t-\tau)}\Psi e_g\| d\tau.$$

Applying Equation (4.31) and using $\kappa_2 = \kappa_2(\Sigma)$, this yields

$$\|\delta(t)\| \leq \frac{2c_d k_T \sqrt{N}\mu}{\kappa_2} + \left(c_d\|\delta(0)\| - \frac{2c_d k_T \sqrt{N}\mu}{\kappa_2} \right) e^{-\kappa_2 t}, \tag{4.41}$$

because $\|\Psi\| = \|\mathcal{M}\| = 1$, $\|\hat{g}_i\| \leq \mu$ and $\|\hat{g}\| \leq \sqrt{N}\mu$. This estimate shows that the zero-input response decays to zero exponentially fast while the zero-state response is bounded for every bounded input.

Since $\mathbf{1}^T\mathcal{L} = \mathbf{0}^T$, the time derivative of $\bar{e}_p$ and $\bar{e}_v$ are given by

$$\begin{aligned} \dot{\bar{e}}_p(t) &= \bar{e}_v(t), \\ \dot{\bar{e}}_v(t) &= -k_T\gamma\bar{e}_v(t) + k_T\bar{e}_g(e_p), \end{aligned} \tag{4.42}$$

with initial average $\bar{e}_p(0) = \frac{1}{N}\mathbf{1}^T e_p(0) = \bar{e}_{p0}$, $\bar{e}_v(0) = \frac{1}{N}\mathbf{1}^T e_v(0) = \bar{e}_{v0}$, $\bar{e}_g = \frac{1}{N}\mathbf{1}^T(\hat{g}(e_p + r^*) - \hat{g}(r^*))$, $\bar{e}_g$ satisfies $\bar{e}_g(0) = 0$ and $e_p^T\bar{e}_g(e_p) < 0$ for all $e_p \neq 0$.

If the graph is an undirected graph and consensus is achieved, the state of every agent will converge to the consensus dynamics' solution of the Equation (4.42). Then the equilibrium point r^* is asymptotically stable because the agents starting on either side of the source will have to move toward the source of the scalar field due to the sign of $\bar{e}_g$. To arrive to the same conclusion, consider the system $\dot{\bar{e}}_p(t) = \bar{e}_v(t)$, $\dot{\bar{e}}_v(t) = -k_T\gamma\bar{e}_v(t) - f(e_p)$, where $f(e_p) = -k_T\bar{e}_g(e_p)$ and satisfies $f(0) = 0$; $\bar{e}_p^T(e_p)f(e_p) > 0$ for all $e_p \neq 0$. Consider the Lyapunov function candidate $V = \frac{1}{2}\bar{e}_v^T\bar{e}_v + \int_0^{\bar{e}_p} f^T(z)dz$, V is continuously differentiable, $V(0) = 0$, and $V > 0$ for all $e_p \neq 0$ and $e_v \neq 0$. Then $\dot{V} = -k_T\gamma\bar{e}_v^T\bar{e}_v \leq 0$. It can be concluded that the equilibrium point is stable.

Applying the Lasalle's principle, it is inferred that the equilibrium point is also stable because $\bar{e}_v = 0 \Rightarrow \bar{e}_p = 0 \Rightarrow \dot{\bar{e}}_v = 0 = -f(0) = 0$. Note that the feedback gains k_T and γ determine the consensus dynamics. k_F and θ determine both whether the consensus is achieved and they also determine consensus velocity.

The distance between the scalar field's source and the formation's center of mass is bounded by

$$\|r_s - \bar{r}^*\| \leq \frac{2}{L_H}\left(e_0 + L_H \frac{h}{2}\right). \tag{4.43}$$

This result is obtained applying a similar analysis as done for demonstrating Theorem 2. $\square$

Remark 5. *If the estimated gradient is normalized,* $\hat{g}^N(t) = \left[\frac{\hat{g}_1^T(t)}{\|\hat{g}_1(t)\|}, \dots, \frac{\hat{g}_N^T(t)}{\|\hat{g}_N(t)\|}\right]^T$, $\|\frac{\hat{g}_i}{\|\hat{g}_i\|}\| = 1$ *and* $\mu = 1$, *then* $\|\hat{g}^N\| = \sqrt{N}$. *Thus, the convergence region is given by* $\|\delta(t)\| \leq \frac{2c_d k_T \sqrt{N}}{\kappa_2}$ *and the agents travel at constant velocity.*

4.4 Simulation Results

In order to demonstrate convergence of the proposed source seeking algorithms, mobile agents ($N = 7$) with the formation and communication graph $\mathcal{G}$ (Fig. 1.4) are considered. The communication topology is undirected and connected. The scalar field is defined as

$$\psi(r) = A_0 e^{-\left((r-r_s)^T H_1 (r-r_s)\right)} + A_0 e^{-\left((r-r_s)^T H_2 (r-r_s)\right)},$$

where $A_0 = 3$, $H_1 = \begin{bmatrix} \frac{1}{2\sigma_{x1}^2} & 0 \\ 0 & \frac{1}{2\sigma_{y1}^2} \end{bmatrix}$, $H_2 = \begin{bmatrix} \frac{1}{2\sigma_{x2}^2} & 0 \\ 0 & \frac{1}{2\sigma_{y2}^2} \end{bmatrix}$, $\sigma_{x1} = 30$, $\sigma_{y1} = 75$, $\sigma_{x2} = 80$ and $\sigma_{y2} = 25$. Their maximum is located at $r_s = [40\ 80]^T$.

	x_{Fi}	y_{Fi}
Agent 1	0	0
Agent 2	-3	6
Agent 3	-6	0
Agent 4	-3	-6
Agent 5	3	-6
Agent 6	6	0
Agent 7	3	6

Table 4.1: Desired formation for cooperative source seeking with single and double integrator agents

	x_{0i}	y_{0i}
Agent 1	0	0
Agent 2	0	1
Agent 3	1	1
Agent 4	1	0
Agent 5	2	0
Agent 6	2	1
Agent 7	3	0

Table 4.2: Initial positions for cooperative source seeking with single and double integrator agents

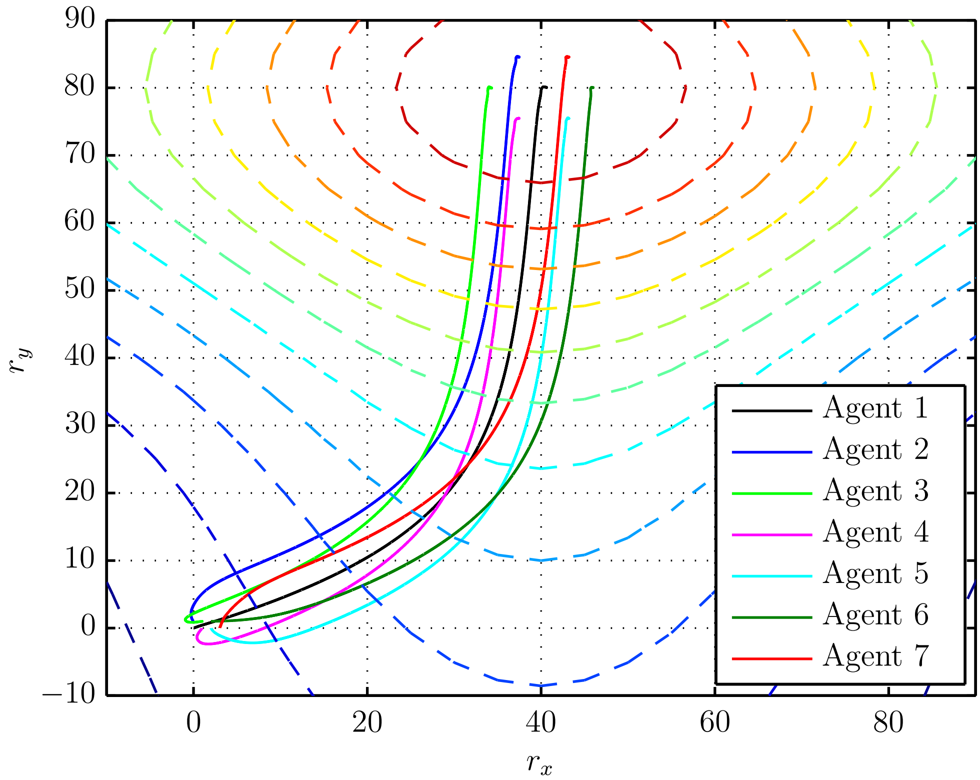


Figure 4.2: Formation with double integrator agents

The desired formation $r_{Fi} = [x_{Fi}\ y_{Fi}]^T$ is defined as shown in Table 4.1, and the initial positions $r_{0i} = [x_{0i}\ y_{0i}]^T$ are set as shown in Table 4.2.

To estimate the gradient, Equation (3.25) is implemented with signals corrupted by noise. Their initial values are $\varphi(0) = \nu(0) = 0$. The tuning parameters are set to $\beta_\varphi = 1$ and $\beta_\nu = 1.2$. For r_i and ψ_i, the noise covariance matrix is set to diag(0.23 0.24 0.21 0.22 0.23 0.24 0.25) and diag(0.23 0.26 0.24 0.26 0.28 0.29 0.3), respectively.

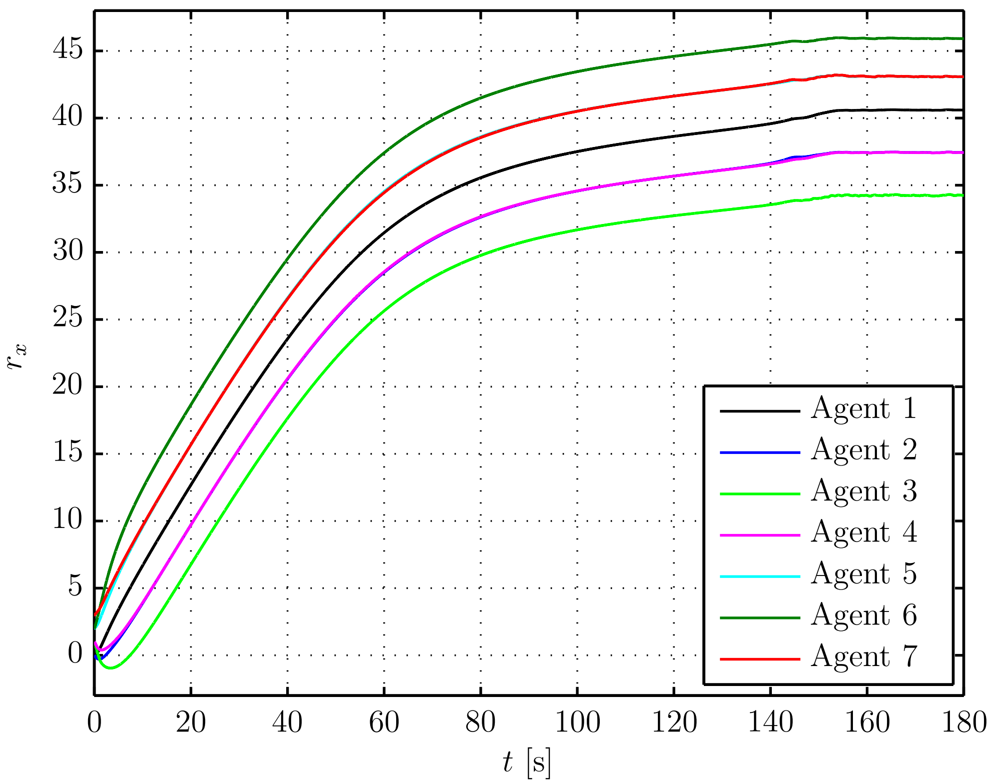


Figure 4.3: Formation's x-positions for double integrator agents

Simulation results for double integrator models are presented. The source seeking algorithm (4.28) from Theorem 2 is implemented. The distributed controller's tuning parameters are set to $k_F = 10$, $\theta = 5$, $\gamma = 1.5$, and $k_T = 2$. The second eigenvalue λ_2 of $\mathcal{L}$ is 1.382 and the convergence radius is defined as $\epsilon = 5.743$.

Fig. 4.2 shows the transient response of the formation. Thanks to the distributed control law (4.28), agents maintain the formation and locate the scalar field's maximum at $r_s = [40\ 80]^T$.

Fig. 4.3 shows each agent's x-position. Agents start from its initial positions, and after the transient, they achieve the desired formation maintaining a relative distance between them. Note that when agents achieve the scalar field's source, they stay at their final positions.

Fig. 4.4 shows the agents' velocity in y direction $\dot{r}_{yi}$, and the magnitude of the velocity $\|\dot{r}_i\|$ of the formation. Note that the colors of the lines correspond to agents as explained in Fig. 4.3. The velocity $\dot{r}_{yi}$ converges to a common value and the magnitude of the agents' velocity converges to a constant velocity due to the normalization of the estimated gradient. The velocity is determined by the parameters k_T and γ. When agents reach the maximum, the velocity of all of them goes to zero. Note that the agents' velocity oscillates around its equilibrium points when the signal measurements are corrupted by noise. The proposed algorithm is able to drive the formation to the scalar field's source. Agents travel at constant velocity and locate the maximum value while the formation is maintained.

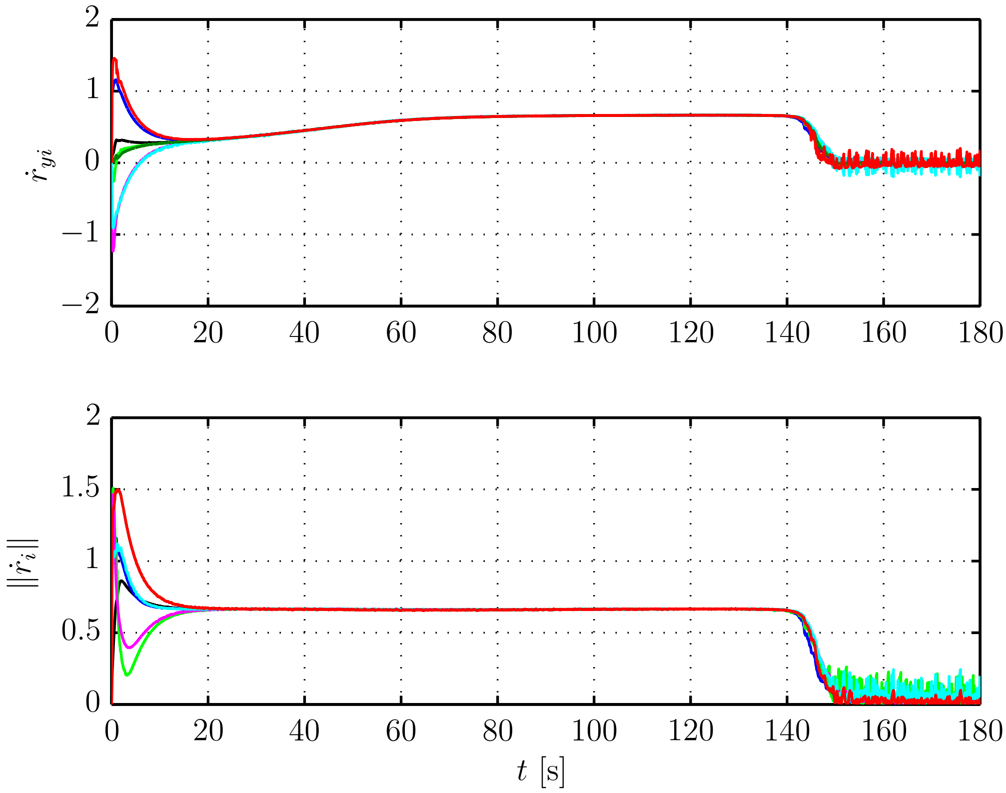


Figure 4.4: Formation's $\dot{r}_{yi}$- and $\|\dot{r}_i\|$-velocity for double integrator agents

Similar results have been obtained for single integrator agents. The source seeking algorithm (4.2) presented in Theorem 1 has been implemented. The distributed controller's tuning parameters for single integrator agent's formation are set to $k_F = 10$, $\theta = 5$, $\gamma = 1.5$, and $k_T = 2$. The second eigenvalue λ_2 of $\mathcal{L}$ is 1.382.

Fig. 4.5 shows the formation's transient response. Due to the distributed control law (4.2), agents maintain the desired formation and locate the scalar field's maximum at $r_s = [40\ 80]^T$. Fig. 4.6 shows each agent's x-positions. Agents start from its initial positions, and after the transient, they achieve the wanted formation maintaining a relative distance between them. Note that when agents achieve the scalar field's source, they maintain their final positions.

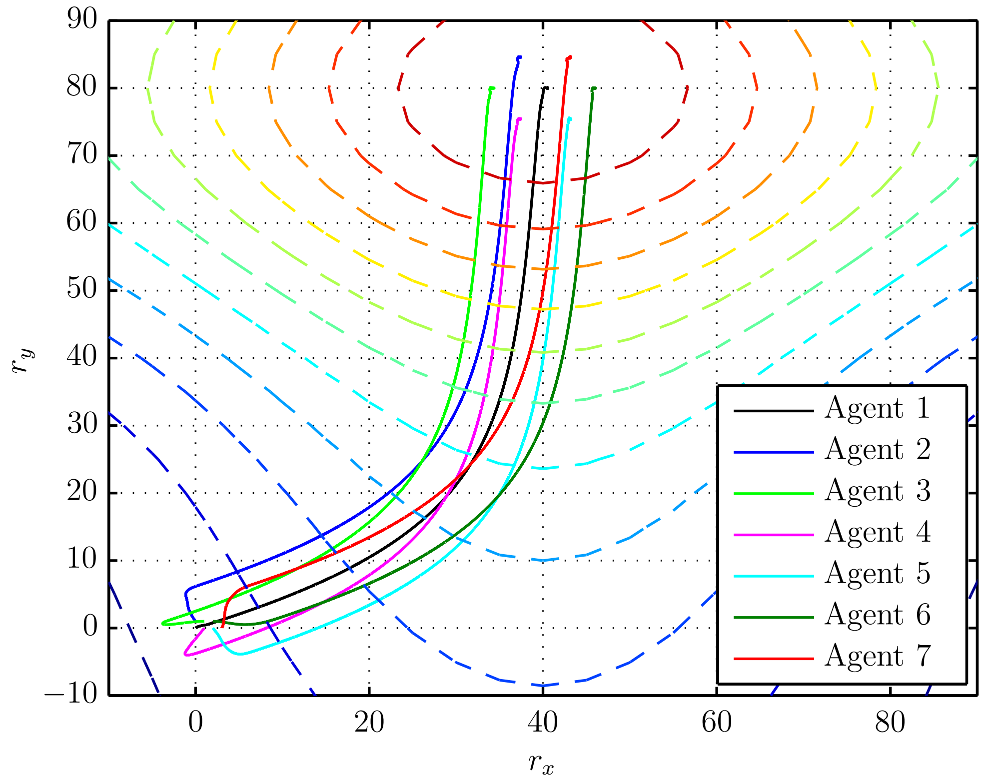


Figure 4.5: Formation with agents modelled as single integrators

Fig. 4.7 shows agents' velocity in y direction $\dot{r}_{yi}$, and the magnitude of the formation's velocity $\|\dot{r}_i\|$. Note that the colors of the lines are explained in Fig. 4.6. Velocity $\dot{r}_{yi}$ converges to a common value and the magnitude of the agents' velocity converges to a constant velocity due to the estimated gradient's normalization. Agents' velocity is determined by parameter k_T. When agents reach the maximum, their velocity becomes zero. Note that agents' velocity oscillates around their equilibrium points when signal measurements are corrupted by noise.

The proposed distributed algorithms for single and double integrator agents are able to drive the formation to the scalar field's source. Agents travel at constant velocity and locate the maximum value while the formation is maintained.

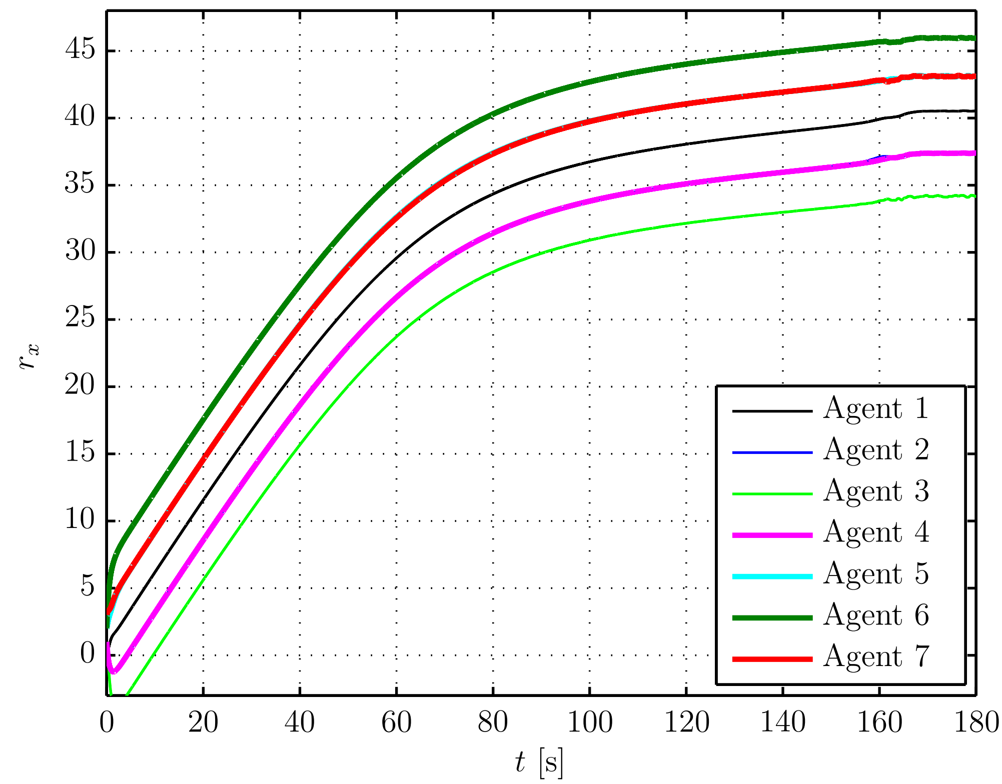


Figure 4.6: x-positions of the formation for agents modelled as single integrator

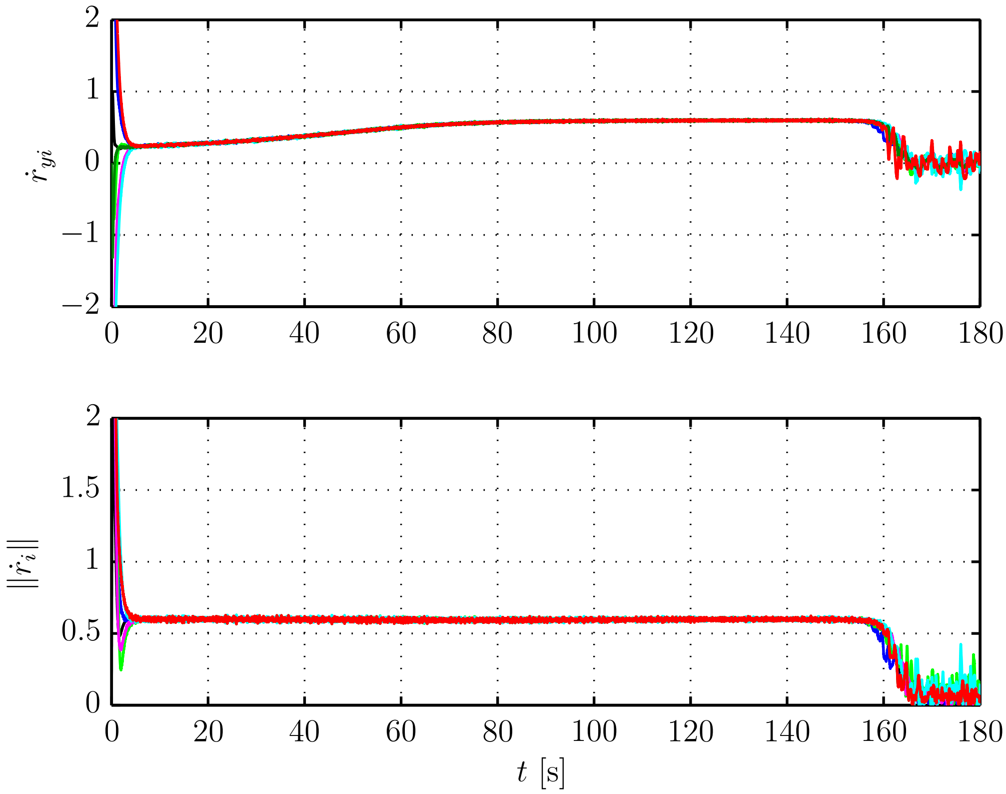


Figure 4.7: Formation's $\dot{r}_{yi}$- and $\|\dot{r}_i\|$-velocity for single integrator agents

4.5 Conclusions

In this chapter, cooperative multi-agent controllers for single and double integrator agents have been presented. They enable the agents' formation to locate the unknown source of the scalar field under constrained communications. A distributed navigation strategy based on both a trajectory and a formation controller has been successfully implemented as shown by the simulation results. Theoretical analysis demonstrates that agents are able to converge towards the source of the scalar field while the formation is maintained. The results here presented are applicable to any agents' formation independently of the formation's size.

Chapter 5

Cooperative Source Seeking with LTI Agent Models

In many real life applications agents possess higher order dynamics than single and double integrator dynamics. The former can be often represented by Linear Time-Invariant (LTI) models. LTI systems are described by sets of linear, ordinary differential equations that have constant coefficients. In this chapter, the focus will be shifted to modelling a dynamic agent's behaviour as a system with multiple inputs and multiple outputs (MIMO) using the state space representation, i.e., LTI systems with coupled first-order linear differential equations with constant coefficients.

After having considered the cooperative source seeking problem for simple dynamics (single and double integrator agents), the method developed in this work is extended in this chapter to general LTI systems. Stability conditions are provided and the proposed approach is illustrated with formation flight simulation for quad-rotor helicopters.

5.1 Introduction

To guarantee that agents modelled as LTI systems localize the source of the scalar field and the formation's stability is maintained, more advanced controllers are required than the controllers presented in the previous chapters. For this purpose, a distributed dynamic controller using only the position outputs and the estimated gradient is designed. In the design process, it is also possible to include performance requirements.

A simple distributed control strategy was considered in Chapter 4, which was composed by formation control and trajectory control. The dynamic controller presented in this chapter has the same structure, but some additional considerations about formation control will be presented.

In this chapter the problem of locating an unknown scalar field's maximum using a formation of N identical LTI agents under undirected constrained communication is studied. For every agent, based on relative information from its neighbour agents, a gradient is estimated. (see Chapter 2). A distributed dynamic output feedback protocol based on both a gradient estimation algorithm and a formation controller is proposed. The formation control component's goal is to keep agents in a desired geometric formation, while the trajectory control component's goal is to steer the formation towards the source. Performance requirements are incorporated to our design using mixed-sensitivity loop shaping. The mixed-sensitivity synthesis technique is used to design the controller. It will be demonstrated that the presented distributed controllers enable agents to converge towards the scalar field's source while the formation is maintained. When position and concentration measurements are corrupted by noise, distributed consensus filters can be used in order to estimate the gradient direction.

The contribution of this chapter is a distributed dynamic output feedback controller for LTI systems. Stability conditions are provided. In our approach it is neither necessary to know the formation's center of mass and its estimated gradient nor to keep the agents rotating in a circular formation.

In Sections 5.2 and 5.3, control law and stability analysis for LTI models are presented, respectively. Section 5.4 provides formation flight simulation results for quad-rotor helicopters. Finally, concluding remarks are made in Section 5.5.

5.2 LTI Models

Consider a group of N identical agents with linear time-varying dynamics described by

$$\begin{aligned}
\dot{x}_i(t) &= A_p x_i(t) + B_u u_i(t) + B_w \bar{w}_i(t),\\
z_i(t) &= C_z x_i(t) + D_{zu} u_i + D_{zw} \bar{w}_i(t),\\
r_i(t) &= C_y x_i(t) + D_{yw} \bar{w}_i(t),
\end{aligned} \tag{5.1}$$

where $x_i \in \mathbb{R}^n$ is the state of agent i, $u_i \in \mathbb{R}^{m_1}$ is the control input, $\bar{w}_i \in \mathbb{R}^{m_2}$ is the external disturbance and noise, $r_i \in \mathbb{R}^p$ is the measured position output, and $z_i \in \mathbb{R}^q$ is the controlled output of agent i. It is assumed that (A_p, B_u) is stabilizable, (A_p, C_y) is detectable, and without loss of generality, B_u is full column rank.

Based on the neighbours' relative output measurements between agents, a distributed dynamic output feedback control for agent i is considered as

$$\begin{aligned}
\dot{v}_i(t) =& A_K v_i(t) + B_K(e_{Fi}(t) + e_{Ti}(t)),\\
u_i(t) =& C_K v_i(t) + D_K(e_{Fi}(t) + e_{Ti}(t)),
\end{aligned} \tag{5.2}$$

where

$$\begin{aligned}
e_{Fi}(t) =& \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} a_{ij} \left[(r_{Fi}(t) - r_{Fj}(t)) - (r_i(t) - r_j(t)) \right],\\
e_{Ti}(t) =& v_{Ti}(t) = k_T \hat{g}_i(r_i(t)),
\end{aligned} \tag{5.3}$$

where

$$\hat{g}_i(r_i) = \left(\sum_{j\in\mathcal{N}_i} a_{ij}(r_i - r_j)(r_i - r_j)^T\right)^{-1} \left(\sum_{j\in\mathcal{N}_i} a_{ij}(r_i - r_j)(\psi(r_i) - \psi(r_j))\right),$$

is the distributed estimated gradient computed in each agent i in Chapter 2, and $v_i(t) \in \mathbb{R}^{m_K}$ is the state of the dynamic output feedback controller. If $m_K = 0$ Equation (5.2) is reduced to a static output feedback law. Let $x = [x_1^T, \ldots, x_N^T]^T$, $v = [v_1^T, \ldots, v_N^T]^T$, $\bar{w} = [\bar{w}_1^T, \ldots, \bar{w}_N^T]^T$, $r_F = [r_{F1}^T, \ldots, x_{FN}^T]^T$, $v_T = [v_{T1}^T, \ldots, v_{TN}^T]^T$, and $z = [z_1^T, \ldots, z_N^T]^T$. Substituting the controller (5.2) into the system (5.1), the closed-loop network dynamics can be written as

$$\begin{aligned}
\begin{bmatrix}\dot{x}\\ \dot{v}\end{bmatrix} &= \begin{bmatrix} I_N \otimes A_p - \mathcal{L}\otimes B_u D_K C_y & I_N \otimes B_u C_K \\ -\mathcal{L}\otimes B_K C_y & I_N \otimes A_K \end{bmatrix} \begin{bmatrix}x\\ v\end{bmatrix} \\
&\quad + \begin{bmatrix} \mathcal{L}\otimes B_u D_K & I_N \otimes B_u D_K & I_N \otimes B_w - \mathcal{L}\otimes B_u D_K D_{yw} \\ \mathcal{L}\otimes B_K & I_N \otimes B_K & -\mathcal{L}\otimes B_K D_{yw} \end{bmatrix} \begin{bmatrix} r_F \\ v_T \\ \bar{w}\end{bmatrix}, \\
z &= \begin{bmatrix} I_N \otimes C_z - \mathcal{L}\otimes D_{zw} D_K C_y & I_N \otimes D_{zw} C_K \end{bmatrix} \begin{bmatrix}x\\ v\end{bmatrix} \\
&\quad + \begin{bmatrix} \mathcal{L}\otimes D_{zw} D_K & I_N \otimes D_K & I_N \otimes D_{zw} - \mathcal{L}\otimes D_{zw} D_K D_{yw} \end{bmatrix} \begin{bmatrix} r_F \\ v_T \\ \bar{w}\end{bmatrix}, \\
r &= \begin{bmatrix} I_N \otimes C_y & 0 \end{bmatrix} \begin{bmatrix}x\\ v\end{bmatrix} + \begin{bmatrix} 0 & 0 & I_N \otimes D_{yw} \end{bmatrix} \begin{bmatrix} r_F \\ v_T \\ \bar{w}\end{bmatrix},
\end{aligned} \tag{5.4}$$

where $w(t) = [r_F^T \ v_T^T \ \bar{w}^T]^T$ and $\otimes$ denotes the Kronecker product.

5.2.1 Controller Synthesis

A closed-loop representation of our approach is shown in Fig. 5.1. Each agent is assumed to be locally stabilized. Based on the approach presented in (Pilz et al., 2009) and (Popov and Werner, 2009), local controllers are proposed. First, controllers to stabilize the formation are designed, then the formation's convergence to the scalar field's source is verified.

To design a local controller K_L, it is assumed that the system's sensors are capable of measuring all states. A full state feedback LQR controller is chosen as $\zeta_i = K_L x_i + u_i$, where K_L is designed to stabilize the dynamics of each agent i and u_i is the control law. The latter steers the agent's position in the formation and tracks the gradient direction. The new dynamics with a local controller is given by

$$\begin{aligned}
\dot{x}_i(t) &= (A_p + B_u K_L)x_i(t) + B_u u_i(t) + B_w \bar{w}_i(t), \\
\dot{x}_i(t) &= A x_i(t) + B_u u_i(t) + B_w \bar{w}_i(t), \\
z_i(t) &= C_z x_i(t) + D_{zu} u_i + D_{zw} \bar{w}_i(t), \\
r_i(t) &= C_y x_i(t) + D_{yw} \bar{w}_i(t),
\end{aligned} \tag{5.5}$$

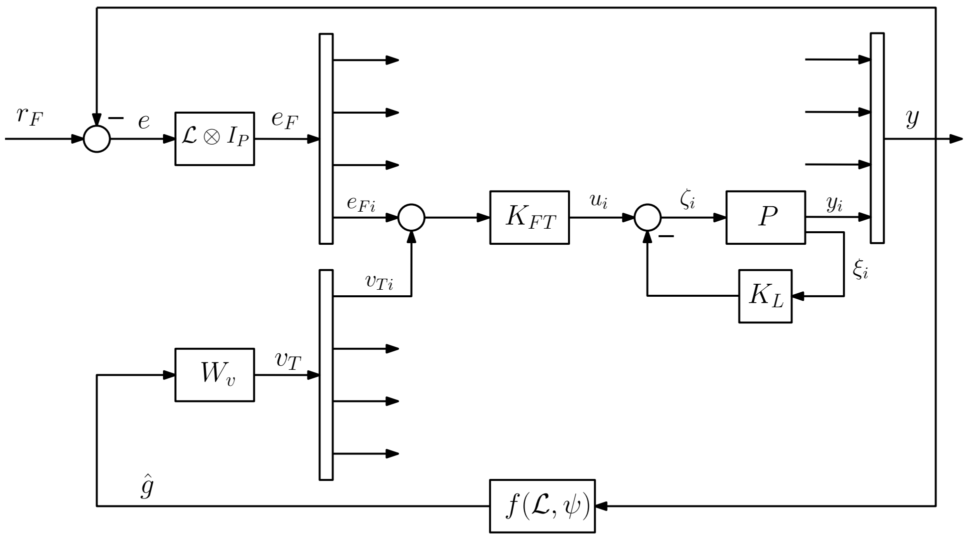


Figure 5.1: Closed-loop representation of a formation

where $A = A_p + B_u K_L$ and the matrix is Hurwitz. The formation controller K_{FT} is designed based on Theorems 3 and 4 of (Pilz et al., 2009). Performance requirements are incorporated into our design by means of mixed-sensitivity loop shaping. H_∞ synthesis technique is used to design the controller. A generalized plant construction with sensitivity $W_S(s)$ and control sensitivity $W_K(s)$ is built. These theorems reduce the formation stability problem to an H_∞ design problem for a single agent with uncertainty. They also guarantee stability for any formation if the controller K_{FT} satisfies the design requirements.

As mentioned in Section 1.5, the following assumption has been made.

Assumption 3. *It is assumed that the state matrix A_p of each agent is marginally stable.*

5.2.2 H_∞ Performance Analysis

Let $0 = \lambda_1 < \lambda_2 \leq, \ldots, \leq \lambda_N$ be the eigenvalues of the Laplacian matrix $\mathcal{L}$. Since $\mathcal{L}$ is symmetric, there exists a orthogonal matrix $U \in \mathbb{R}^{N \times N}$ such that $U^{-1}\mathcal{L}U = \Lambda = diag[\lambda_1, \ldots, \lambda_N]$. Let $x = (U \otimes I_n)\tilde{x}$, $v = (U \otimes I_{mK})\tilde{v}$, $z = (U \otimes I_q)\tilde{z}$, $\bar{w} = (U \otimes I_{m2})\tilde{\bar{w}}$, $v_T = (U \otimes I_p)\tilde{v}_T$, $r_F = (U \otimes I_p)\tilde{r}_F$. Then, Equation (5.4) can be rewritten as

$$\begin{aligned}
\begin{bmatrix} \dot{\tilde{x}} \\ \dot{\tilde{v}} \end{bmatrix} &= \begin{bmatrix} I_N \otimes A_p - \Lambda \otimes B_u D_K C_y & I_N \otimes B_u C_K \\ -\Lambda \otimes B_K C_y & I_N \otimes A_K \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{v} \end{bmatrix} \\
&\quad + \begin{bmatrix} \Lambda \otimes B_u D_K & I_N \otimes B_u D_K & I_N \otimes B_w - \Lambda \otimes B_u D_K D_{yw} \\ \Lambda \otimes B_K & I_N \otimes B_K & -\Lambda \otimes B_K D_{yw} \end{bmatrix} \begin{bmatrix} \tilde{r}_F \\ \tilde{v}_T \\ \tilde{\bar{w}} \end{bmatrix}, \\
\tilde{z} &= \begin{bmatrix} I_N \otimes C_z - \Lambda \otimes D_{zw} D_K C_y & I_N \otimes D_{zw} C_K \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{v} \end{bmatrix} \\
&\quad + \begin{bmatrix} \Lambda \otimes D_{zw} D_K & I_N \otimes D_K & I_N \otimes D_{zw} - \Lambda \otimes D_{zw} D_K D_{yw} \end{bmatrix} \begin{bmatrix} \tilde{r}_F \\ \tilde{v}_T \\ \tilde{\bar{w}} \end{bmatrix}, \\
\tilde{r} &= \begin{bmatrix} I_N \otimes C_y & 0 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{v} \end{bmatrix} + \begin{bmatrix} 0 & 0 & I_N \otimes D_{yw} \end{bmatrix} \begin{bmatrix} \tilde{r}_F \\ \tilde{v}_T \\ \tilde{\bar{w}} \end{bmatrix}.
\end{aligned} \tag{5.6}$$

Note that Equation (5.6) is composed of N individual systems as

$$\begin{aligned}
\begin{bmatrix} \dot{\tilde{x}} \\ \dot{\tilde{v}} \end{bmatrix} &= \begin{bmatrix} A_p - \lambda_i B_u D_K C_y & B_u C_K \\ -\lambda_i B_K C_y & A_K \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{v} \end{bmatrix} \\
&\quad + \begin{bmatrix} \lambda_i B_u D_K & B_u D_K & B_w - \lambda_i B_u D_K D_{yw} \\ \lambda_i B_K & B_K & -\lambda_i B_K D_{yw} \end{bmatrix} \begin{bmatrix} \tilde{r}_F \\ \tilde{v}_T \\ \tilde{\bar{w}} \end{bmatrix}, \\
\tilde{z} &= \begin{bmatrix} C_z - \lambda_i D_{zw} D_K C_y & D_{zw} C_K \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{v} \end{bmatrix} \\
&\quad + \begin{bmatrix} \lambda_i D_{zw} D_K & D_K & D_{zw} - \lambda_i D_{zw} D_K D_{yw} \end{bmatrix} \begin{bmatrix} \tilde{r}_F \\ \tilde{v}_T \\ \tilde{\bar{w}} \end{bmatrix}, \\
\tilde{r} &= \begin{bmatrix} C_y & 0 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{v} \end{bmatrix} + \begin{bmatrix} 0 & 0 & D_{yw} \end{bmatrix} \begin{bmatrix} \tilde{r}_F \\ \tilde{v}_T \\ \tilde{\bar{w}} \end{bmatrix},
\end{aligned} \tag{5.7}$$

$\forall\ i = 1, \ldots, N$. For $\lambda_1 = 0$, the state matrix of the system (5.4) is unstable if the given matrix A_p is unstable.

Denote the transfer function matrices of systems (5.4) and (5.6) by $T_{\tilde{w}\tilde{z}}$ and $T_{\tilde{w}_i\tilde{z}_i}$, respectively. Then it follows $T_{\tilde{w}\tilde{z}} = \mathrm{diag}(T_{\tilde{w}_1\tilde{z}_1},\ T_{\tilde{w}_2\tilde{z}_2},\ \ldots,\ T_{\tilde{w}_N\tilde{z}_N}), = (U^{-1} \otimes I_{m2}) T_{\tilde{w}\tilde{z}} (U \otimes I_{m1})$. Thus, the relationships between the H_∞ norm of T_{wz}, $T_{\tilde{w}\tilde{z}}$ and $T_{\tilde{w}_i\tilde{z}_i}$ is $\|T_{wz}\|_\infty = \|T_{\tilde{w}\tilde{z}}\|_\infty = \max_{i=1,2,\ldots,N} \|T_{\tilde{w}_i\tilde{z}_i}\|_\infty$. The previous equations convert the distributed H_∞ control problem of the multi-agent network into a H_∞ control problem of a single agent. These transformations are based on the works of (Eichler and Werner, 2013; Li et al., 2011; Massioni and Verhaegen, 2009).

5.3 Stability Analysis

In this section, agents' convergence inside a scalar field will be analysed. For this purpose, Theorem 4 is presented. It is based on assumptions 1, 2 and 3, which were presented in Chapter 1. Let

$$\tilde{B} = \begin{bmatrix} 0 & B_u D_K \\ 0 & B_K \end{bmatrix} \text{ and } \tilde{G} = \begin{bmatrix} -B_u D_K & 0 \\ B_K & 0 \end{bmatrix}.$$

Theorem 4. *Consider the multi-agent system (5.1) with control law (5.2). Suppose that assumptions 1, 2 and 3 are fulfilled. Then, for all $r_i(0) \in \mathbb{R}^p$ and $t \geq 0$ agents locate the unknown source's position r_s of the scalar field $\psi(r)$ and the closed-loop system converges to an equilibrium point centred ball with radius*

$$\epsilon \leq \frac{2k_T\sqrt{N}\mu c_{\hat{A}}\|C_y\|\left(\|\tilde{B}\| + \|\mathcal{L}\|\|\tilde{G}\|\right)}{\lambda_{\hat{A}}}, \tag{5.8}$$

with the distance between the formation's center of mass and the scalar field's source

$$\|\bar{r}^* - r_s\| \leq \frac{2}{L_H}\left(e_0 + L_H\frac{h}{2}\right). \tag{5.9}$$

Proof. Consider the closed-loop system of (5.4), with $\bar{w} = 0$, as

$$\begin{aligned}
\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} &= \begin{bmatrix} I_N \otimes A - \mathcal{L} \otimes B_u D_K C_y & I_N \otimes B_u C_K \\ -\mathcal{L} \otimes B_K C_y & I_N \otimes A_K \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} \\
&+ \begin{bmatrix} \mathcal{L} \otimes B_u D_K & I_N \otimes B_u D_K \\ \mathcal{L} \otimes B_K & I_N \otimes B_K \end{bmatrix} \begin{bmatrix} r_F \\ k_T \hat{g}(r) \end{bmatrix}, \\
r &= \begin{bmatrix} I_N \otimes C_y & 0 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix}.
\end{aligned} \tag{5.10}$$

Let $\zeta = [x^T \ v^T]^T$ and $\xi = [r_F^T \ k_T\hat{g}(r)^T]^T$. Equation (5.10) can be written as

$$\begin{aligned}
\dot{\zeta} &= \hat{A}\zeta + \hat{B}\xi, \\
r &= \hat{C}\zeta,
\end{aligned} \tag{5.11}$$

where $\hat{A} = I_N \otimes \tilde{A} + \mathcal{L} \otimes \tilde{F}$ and this term is Hurwitz matrix, $\hat{B} = I_N \otimes \tilde{B} + \mathcal{L} \otimes \tilde{G}$, $\hat{C} = I_N \otimes \tilde{C}$, with

$$\begin{aligned}
\tilde{A} &= \begin{bmatrix} A & B_u C_K \\ 0 & A_K \end{bmatrix}, \ \tilde{F} = \begin{bmatrix} -B_u D_K C_y & 0 \\ B_K & 0 \end{bmatrix}, \\
\tilde{B} &= \begin{bmatrix} 0 & B_u D_K \\ 0 & B_K \end{bmatrix}, \ \tilde{G} = \begin{bmatrix} -B_u D_K & 0 \\ B_K & 0 \end{bmatrix}, \\
\tilde{C} &= \begin{bmatrix} C_y & 0 \end{bmatrix}.
\end{aligned}$$

The equilibrium point is given by

$$\begin{aligned}\zeta^* =& \hat{A}^{-1}\hat{B}\xi^*,\\ r^* =& \hat{C}\zeta^*.\end{aligned} \tag{5.12}$$

Let the state error be $e_\zeta = \zeta - \zeta^*$ and the position error $e_r = r - r^*$. The dynamic state error can be written as

$$\dot{e}_\zeta = \dot{\zeta} = \hat{A}e_\zeta + \hat{B}(\xi - \xi^*), \tag{5.13}$$

where

$$\xi - \xi^* = \begin{bmatrix} 0 \\ k_T\left(\hat{g}(r) - \hat{g}(r^*)\right)\end{bmatrix} = \begin{bmatrix} 0 \\ k_T\left(\hat{g}(e_r + r^*) - \hat{g}(r^*)\right)\end{bmatrix}.$$

The solution is given by

$$e_\zeta = e^{\hat{A}t}e_\zeta(0) + \int_0^t e^{\hat{A}(t-\tau)}\hat{B}(\xi - \xi^*)d\tau, \tag{5.14}$$

and the position error can be written as

$$e_r = \hat{C}e_\zeta, = \hat{C}e^{\hat{A}t}e_\zeta(0) + \hat{C}\int_0^t e^{\hat{A}(t-\tau)}\hat{B}(\xi - \xi^*)d\tau. \tag{5.15}$$

The solution is bounded by

$$\|e_r\| \le c_{\hat{A}}\|\hat{C}\|\|e_\zeta(0)\|e^{-\lambda_{\hat{A}}t} + 2k_T\sqrt{N}\mu c_{\hat{A}}\|\hat{C}\|\|\hat{B}\|\int_0^t e^{-\lambda_{\hat{A}}(t-\tau)}d\tau, \tag{5.16}$$

since $\|e^{\hat{A}t}\| \le c_{\hat{A}}e^{-\lambda_{\hat{A}}t}$, $\|\xi - \xi^*\| \le 2k_T\sqrt{N}\mu$. It follows

$$\begin{aligned}\|e_r\| \le& \frac{2k_T\sqrt{N}\mu c_{\hat{A}}\|C_y\|\left(\|\tilde{B}\| + \|\mathcal{L}\|\|\tilde{G}\|\right)}{\lambda_{\hat{A}}} + \\ & c_{\hat{A}}\|C_y\|\left(\|e_\zeta(0)\| - \frac{2k_T\sqrt{N}\mu\left(\|\tilde{B}\| + \|\mathcal{L}\|\|\tilde{G}\|\right)}{\lambda_{\hat{A}}}\right)e^{-\lambda_{\hat{A}}t},\end{aligned} \tag{5.17}$$

since $\|\hat{C}\| \le \|C_y\|$, $\|\hat{B}\| \le \|\tilde{B}\| + \|\mathcal{L}\|\|\tilde{G}\|$.

The distance between the scalar field's source and the formation's center of mass is bounded by

$$\|r_s - \bar{r}^*\| \le \frac{2}{L_H}\left(e_0 + L_H\frac{h}{2}\right). \tag{5.18}$$

This result is obtained with a similar analysis done by the proof of Theorem 1. □

Remark 6. *If the estimated gradient is normalized, $\hat{g}^N(t) = \left[\frac{\hat{g}_1^T(t)}{\|\hat{g}_1(t)\|}, \ldots, \frac{\hat{g}_N^T(t)}{\|\hat{g}_N(t)\|}\right]^T$, then $\|\frac{\hat{g}_i}{\|\hat{g}_i\|}\| = 1$, $\mu = 1$ and $\|\hat{g}^N\| = \sqrt{N}$. Thus, the convergence region is defined by $\|e_r\| \le \frac{2k_T\sqrt{N}c_{\hat{A}}\|C_y\|\left(\|\tilde{B}\|+\|\mathcal{L}\|\|\tilde{G}\|\right)}{\lambda_{\hat{A}}}$, and the agents travel at constant velocity.*

5.4 Simulation Results

Simulation results showing convergence of the proposed source seeking algorithm for LTI models are presented in this section. The scalar field is given by

$$\psi(r) = A_0 e^{-\left((r-r_s)^T H_1 (r-r_s)\right)} + A_0 e^{-\left((r-r_s)^T H_2 (r-r_s)\right)},$$

where $A_0 = 3$, $H_1 = \begin{bmatrix} \frac{1}{2\sigma_{x1}^2} & 0 \\ 0 & \frac{1}{2\sigma_{y1}^2} \end{bmatrix}$, $H_2 = \begin{bmatrix} \frac{1}{2\sigma_{x2}^2} & 0 \\ 0 & \frac{1}{2\sigma_{y2}^2} \end{bmatrix}$, $\sigma_{x1} = 30$, $\sigma_{y1} = 75$, $\sigma_{x2} = 80$, $\sigma_{y2} = 25$, and the maximum is located at $r_s = [40\ 80]^T$.

A formation network of $N = 7$ identical mobile agents under an undirected and connected communication topology $\mathcal{G}$ is considered, as shown in Fig. 1.4. Each agent is an underactuated and unstable multi-input multi-output 12^{th} order dynamic model as proposed in (Lara et al., 2006) and (Pilz et al., 2009), with 4 inputs and 3 position outputs. The linearised model of a quad-rotor helicopter is described in Appendix A.

The desired formation $r_{Fi} = [x_{Fi}\ y_{Fi}\ z_{Fi}]^T$ is defined as shown in Table 5.1, and the initial positions $r_{0i} = [x_{0i}\ y_{0i}\ z_{0i}]^T$ are set as shown in Table 5.2.

	x_{Fi}	y_{Fi}	z_{Fi}
Agent 1	0	0	0
Agent 2	-3	6	0
Agent 3	-6	0	0
Agent 4	-3	-6	0
Agent 5	3	-6	0
Agent 6	6	0	0
Agent 7	3	6	0

Table 5.1: Desired formation for cooperative source seeking with LTI agent models

	x_{0i}	y_{0i}	z_{0i}
Agent 1	0	0	0
Agent 2	0	1	0
Agent 3	1	1	0
Agent 4	1	0	0
Agent 5	2	0	0
Agent 6	2	1	0
Agent 7	3	0	0

Table 5.2: Initial positions for cooperative source seeking with LTI agent models

To estimate a gradient in each agent, equation (3.27) is implemented when the signal measurements are corrupted by zero-mean Gaussian noise. The noise covariance matrix is set to diag(0.21 0.22 0.23 0.24 0.25 0.26 0.27) and diag(0.26 0.27 0.26 0.27 0.26 0.26 0.27) for r_i and ψ_i, respectively. Their initial conditions are $\varphi(0) = \nu(0) = 0$. The consensus filters' tuning parameters are set to $\beta_\varphi = 1.2$ and $\beta_\nu = 1$.

A full state feedback LQR controller is chosen in order to stabilize a single agent. The weighting matrices are

$R = \text{diag}(100,\ 0.1,\ 25,\ 25)$ and

$Q = \text{diag}(0.04,\ 1,\ 0.04,\ 1,\ 0.5,\ 20,\ 0.25,\ 1,\ 10^3,\ 50,\ 10^3,\ 50)$.

In order to keep the agents in a desired geometric formation and to steer the formation towards the maximum, the source seeking algorithm (5.2) is implemented. The sensitivity and control sensitivity weighting filters are $W_s = I_3 \otimes \left(\frac{1}{s+0.0001}\right)$ and $W_K = I_4 \otimes \left(50\frac{s+10^3}{s+10^6}\right)$. A H_∞ synthesis technique is used to design the controller; the designed H_∞ controller is of 19th order. The achieved H_∞ norm is $0.992 < 1$.

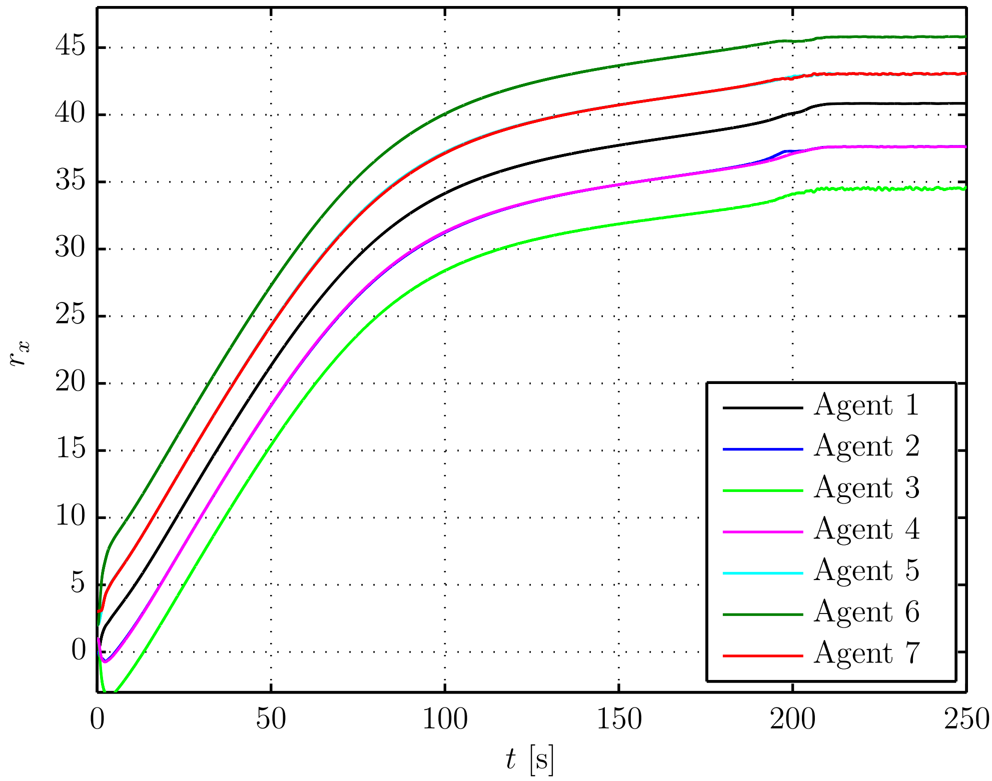


Figure 5.2: Formation's r_x-positions for quad-rotor helicopters

Fig. 5.2 shows each agent's x-position. Agents start from its initial positions, and after the transient, they achieve the desired formation maintaining a relative distance between them. Note that when agents achieve the scalar field's source, they maintain their final positions.

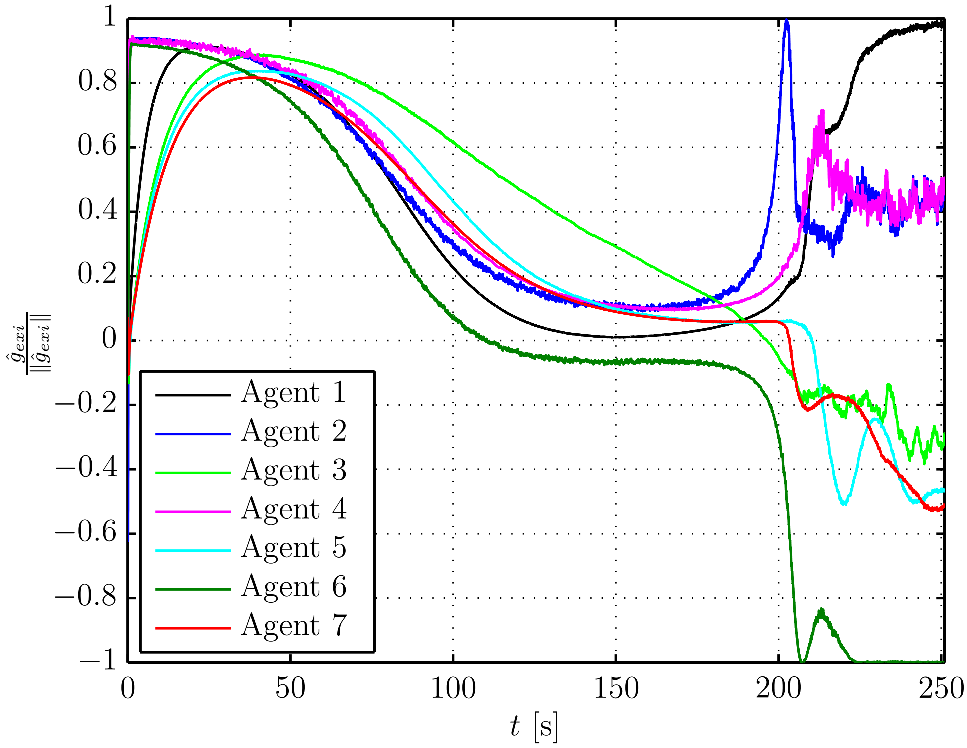


Figure 5.3: Normalized estimated gradient $\frac{\hat{g}_{exi}}{\|\hat{g}_{exi}\|}$ for quad-rotor helicopters

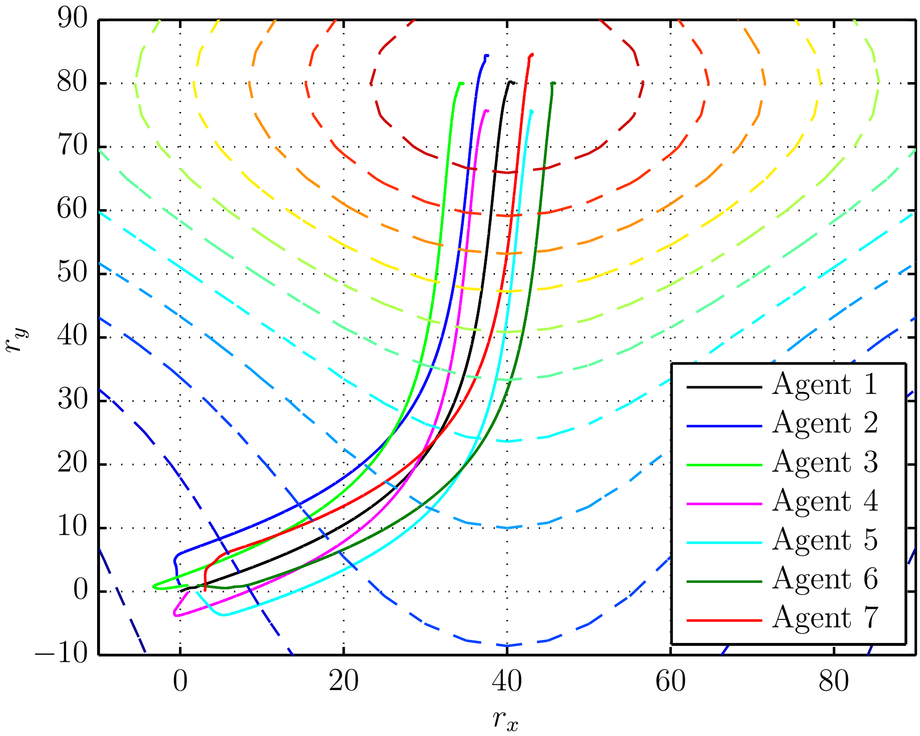


Figure 5.4: Formation's response for quad-rotor helicopters

Fig. 5.3 shows the x-component of the normalized weighted estimated gradient $\frac{\hat{g}_{exi}}{\|\hat{g}_{exi}\|}$ for each agent. Since concentration and position signals are corrupted by noise, a distributed consensus filter designed in Chapter 3 is used. This consensus filter rejects noise at higher frequencies. Fig. 5.4 shows that the agents locate the maximum of the scalar field, $r_s = [40\ 80]^T$, while they maintain the desired formation.

The formation's $\dot{r}_{yi}$- and $\|\dot{r}_i\|$-velocities for quad-rotor helicopters are shown in Fig. 5.5. Note that the color of the lines describe agents as explained in Figs. 5.2, 5.3 and 5.4. The agents' velocity in $\dot{r}_{yi}$ converges to common values and these values turn zero when the formation reaches the maximum. The magnitude of the agents' velocity $\|\dot{r}_i\|$ converges to a constant velocity determined by parameter $k_T = 0.5$, due to the normalization of the estimated gradient. Note that the agents' velocities oscillate around their equilibrium values when signal measurements are corrupted by noise.

The proposed algorithm is able to drive the formation to the scalar field's source at constant velocity. When agents reach the maximum, the velocity of all agents goes to zero.

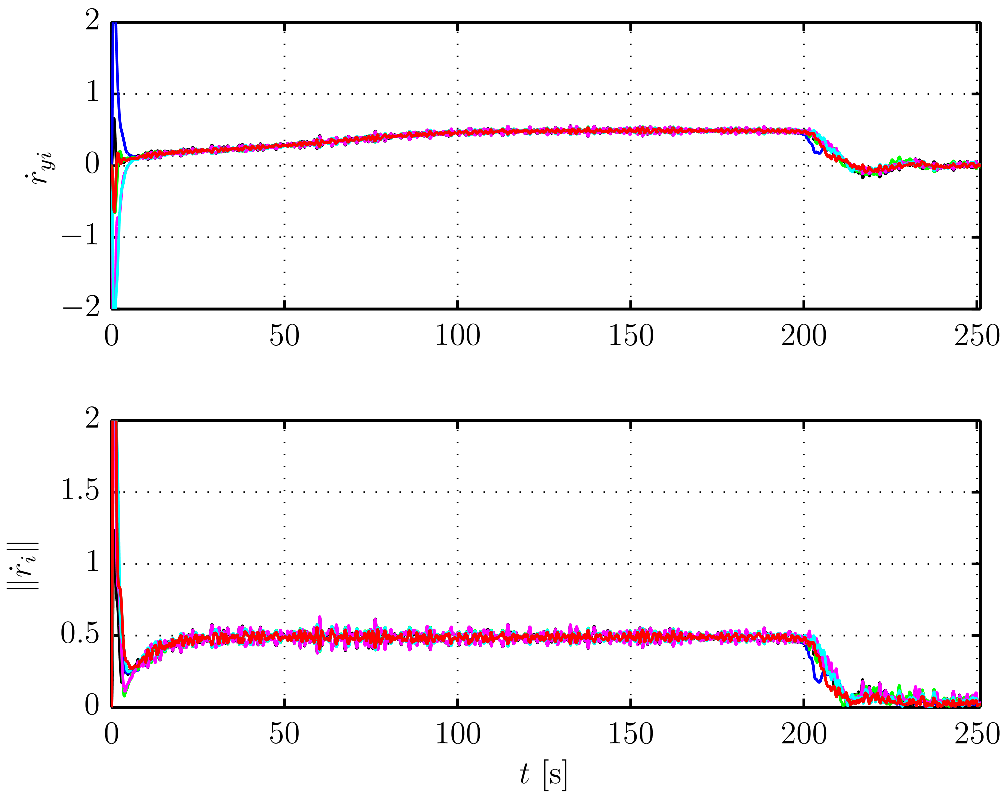


Figure 5.5: Formation's $\dot{r}_{yi}$- and $\|\dot{r}_i\|$-velocity for quad-rotor helicopters

5.5 Conclusions

In this chapter a cooperative multi-agent source seeking algorithm for LTI models has been presented. Agents are aimed to locate an unknown scalar field's source under fixed constrained communication topologies. A distributed controller based on both gradient estimation and a formation controller is proposed. Theoretical analysis proves that the proposed algorithms enable agents to converge towards the scalar field's source while the formation is maintained. Here the results presented in Chapters 2 and 3 are also used to deal with the gradient estimation by noise corrupted signals.

Chapter 6

Cooperative Source Seeking and Collision Avoidance with Non-Holonomic Multi-Agent Systems

This chapter presents a distributed control strategy for steering a swarm with non-holonomic agent dynamics towards a scalar field's source. A simple distributed control strategy is proposed, that preserves the desired formation while locating the scalar field's source and avoiding collision between agents and obstacles. Applying this strategy, agents exchange local information according to an undirected communication topology.

Stability analysis of the overall closed-loop system is provided. The suggested control strategy enables an agents' network to maintain a desired formation and find a scalar field's source. Results are illustrated by simulation studies on a constant scalar field.

6.1 Introduction

Many mechanical systems, like robots, can not be modelled as LTI systems. These kind of systems have fewer controllable degrees of freedom than total degrees of freedom. They are called non-holonomic. In our context, a ground vehicle that moves in two dimensional space has three degrees of freedom (r_x position, r_y position and the orientation ϕ), but at any position, the vehicle can be only moved by a forward motion and a steering angle. The vehicle moves (non-slipping) only in the direction of ϕ. Considering these kind of systems in this work brings a more realistic approach to some situations of the scenarios considered, when agents have this kind of restrictions, i.e. they have non-holonomic constraints.

As previously discussed in this work, hazardous environments and disaster scenarios can often be depicted as scalar fields, which are functions of space. In the literature, different approaches to explore distributed scalar fields have been investigated, such as climbing gradients (Biyik and Arcak, 2007; Ogren et al., 2004), extremum-seeking (Matveev et al., 2011; Zhang et al., 2007), monitoring environmental areas (Zhang and Leonard, 2010), to name a few.

The problem of cooperative source seeking inside a scalar field using non-holonomic agents has been recently addressed. Source seeking algorithms have been presented to steer an agent or a group of agents towards the maximum of a scalar field, or at least, to a point as near as possible to this source. In (Brinon-Arranz et al., 2011), a control strategy for a group of agents uniformly distributed in a circular formation is proposed. This approach requires all-to-all communication and knowledge about the center of mass for all agents. In (Li and Guo, 2012; Li et al., 2014) a controller that combines all-to-all and limited communications is presented. These algorithms are complex and require the center of mass' information to estimate the gradient. Distributed controllers for single and double integrator agent models, as well as for linear time invariant agent models, are proposed in (Rosero and Werner, 2014a,b).

In this chapter, a distributed controller that is implemented locally on each agent is designed. Each controller is composed of a formation part, trajectory part and avoidance part. The formation controller part uses only relative position information to maintain a desired formation in space. The trajectory controller part uses the relative position and relative field signals to estimate the gradient and to drive the group of agents towards the source of the scalar field. The avoidance controller part uses locally defined potential functions based on obstacle's position information, as well as other mobile agents' position information, in order to detect the presence of any object within a given range and therefore, avoid crashing. The analysis of avoidance controller part was made on (Ahmadi et al., 2015).

The contribution of this chapter is a distributed control law that combines formation control, avoidance function and an estimated gradient for non-holonomic systems. Aim of this control law is to steer agents towards the maximum of a scalar field under limited communications and to avoid collisions between agents and obstacles.

Chapter 6 is organized as follows. The distributed control law for a group of non-holonomic systems is presented in Section 6.2. In Section 6.3 the avoidance function used in the distributed control law is explained. In Section 6.4, stability analysis is presented for the whole closed-loop system. Section 6.5 presents simulation results. Finally, concluding remarks are made in Section 6.6.

6.2 Distributed Control Law

The dynamics of an individual agent are assumed to be given by the well-known kinematic model of the non-holonomic unicycle

$$\begin{bmatrix} \dot{r}_{xi}(t) \\ \dot{r}_{yi}(t) \\ \dot{\phi}_i(t) \end{bmatrix} = \begin{bmatrix} \cos(\phi_i) & 0 \\ \sin(\phi_i) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i(t) \\ \omega_i(t) \end{bmatrix}, \tag{6.1}$$

where $i = 1, \ldots, N$, r_{xi} and r_{yi} are the Cartesian coordinates, $\phi_i \in [0, 2\pi)$ the agent's orientation with respect to the x axis, v_i is the linear velocity input and ω_i is the angular velocity input. The vector $r_i = [r_{xi}\ r_{yi}]^T$ represents the position of agent i.

In this section a distributed controller for a group of N non-holonomic mobile agents is introduced. This controller allows agents to locate the source and track the estimated gradient direction without colliding with static objects or other mobile agents. Let V_a be the avoidance function between agents and V_o be the avoidance function between agents and obstacles. The following distributed control law is proposed

$$\begin{aligned} v_i &= \cos(e_{\phi i}) D_i, \\ \omega_i &= -k_\phi e_{\phi i} + \dot{\phi}_{di}, \end{aligned} \tag{6.2}$$

with

$$\begin{aligned} e_{\phi i} &= \phi_i - \phi_{di}, \\ \phi_{di} &= \text{Atan2}\left(E_{yi}, E_{xi}\right), \\ D_i &= \sqrt{E_{xi}^2 + E_{yi}^2}, \\ E_i &= [E_{xi}\ E_{yi}]^T = k_F e_i + k_T \hat{g}_i + k_a \frac{\partial V_{ai}}{\partial r_i} + k_o \frac{\partial V_{oi}}{\partial r_i}, \\ e_i &= [e_{xi}\ e_{yi}]^T = \sum_{j \in \mathcal{N}_i} a_{ij} \left[(r_{Fi} - r_{Fj}) - (r_i - r_j) \right], \\ \hat{g}_i(r_i) &= [\hat{g}_{xi}\ \hat{g}_{yi}]^T, \\ &= \left(\sum_{j \in \mathcal{N}_i} a_{ij} (r_i - r_j)(r_i - r_j)^T \right)^{-1} \left(\sum_{j \in \mathcal{N}_i} a_{ij} (r_i - r_j)(\psi(r_i) - \psi(r_j)) \right), \end{aligned}$$

where $k_F,\ k_T,\ k_\phi,\ k_a$ are tuning parameters, $k_F > 0$ affects the formation error e_i, $k_T > 0$ affects the estimated gradient direction $\hat{g}_i$, $k_\phi > 0$ affects the orientation error $e_{\phi i}$, $k_a > 0$ affects the changes of the avoidance function between agents $\frac{\partial V_{ai}}{\partial r_i}$, $k_o > 0$ affects the changes of the avoidance function for obstacles $\frac{\partial V_{oi}}{\partial r_i}$, r_{Fi} denotes the reference of agent i, $e_{\phi i}$ is the orientation error, and ϕ_{di} defines a desired direction of motion that depends on the formation error e_i. The estimated gradient is computed by Equation (2.3) which depends on the relative concentration values $\psi(r_i) - \psi(r_j)$ and distance $r_i - r_j$ between agents.

Based on this approach, progress in formation control of non-holonomic agents with collision avoidance between agents and obstacles has been made by us in a shared project with Siavash Ahmadi on (Ahmadi et al., 2015). For this reason, in this chapter the stability analysis for multi agents with collision avoidance is not included.

6.3 Avoidance Function

In order to understand how the avoidance function works, consider only one mobile agent and one obstacle as shown in Fig. (6.1) (Mastellone et al., 2008). The coordinates of the object to be avoided are defined as $(x_a,\ y_a)$ and a distance function $d_a = \sqrt{(r_x - x_a)^2 + (r_y - y_a)^2}$. The avoidance function is defined as

$$V_o = \left(\min\left\{0, \frac{d_a^2 - S^2}{d_a - s^2}\right\}\right)^2, \tag{6.3}$$

where $s > 0$ is the avoidance region's radius and $S > 0$ is the detection region's radius, with $S > s$ as represented in Fig. (6.1).

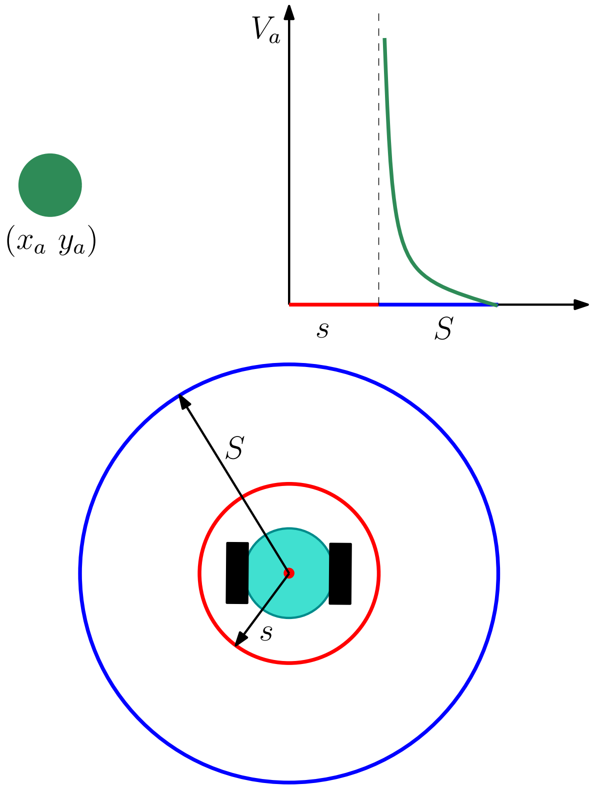


Figure 6.1: Avoidance and detection region around an agent

Taking the partial derivatives of V_a with respect to r_x and r_y coordinates, it is possible to write $\frac{\partial V_o}{\partial r_x} = \begin{cases} 0 & \text{if } d_a \geq S, \\ 4\frac{(S^2-s^2)(d_a^2-S^2)}{(d_a^2-s^2)^3}(r_x - x_a) & \text{if } S > d_a > s, \\ 0 & \text{if } d_a < s, \end{cases}$

and

$$\frac{\partial V_o}{\partial r_y} = \begin{cases} 0 & \text{if } d_a \geq S, \\ 4\frac{(S^2-s^2)(d_a^2-S^2)}{(d_a^2-s^2)^3}(r_y - y_a) & \text{if } S > d_a > s, \\ 0 & \text{if } d_a < s. \end{cases}$$

Now consider a scenario in which a group of agents moves in the direction given by the computed gradient while avoiding collision with each other in a cooperative way. The obstacle avoidance problem using the avoidance function is based on the results presented by (Mastellone et al., 2008). The avoidance function between agents is defined as

$$V_{ai} = \sum_{j=1, j\neq i}^{N-1} V_{aij},$$

where

$$V_{aij} = \left(\min\left\{0, \frac{d_{aij}^2 - S^2}{d_{aij}^2 - s^2}\right\}\right)^2$$

and

$$d_{aij} = \sqrt{(r_{xi} - r_{xj})^2 + (r_{yi} - r_{yj})^2},\ j \neq i,\ j = 1, \ldots, N-1.$$

Each agent i might potentially collide with the remaining $N-1$ agents, therefore for each agent i, $N-1$ avoidance functions V_{aij} are defined for $j = 1, \ldots, N-1$. The avoidance function V_{oi} is defined in the same way as V_{ai} in order to avoid collision between agents.

As mentioned in Section 1.5, the following assumption has been made.

Assumption 4. *Agents' trajectory is smooth and satisfies*

$$e_{\phi 1} = e_{\phi 2} = \cdots = e_{\phi N} \neq \frac{\pi}{2}.$$

Assumption 4 is the only restrictive condition. All agents turn around in their places if the condition is not satisfied.

6.4 Stability Analysis

In order to analyse the stability of the multi agent system without the collision avoidance functions, consider $k_a = k_o = 0$. Taking Equations 6.1 and 6.2 into account, the closed loop system dynamics are given by

$$\begin{aligned} \dot{r}_i(t) =& R(e_{\phi_i})\left(k_F \sum_{j\in\mathcal{N}_i} a_{ij}\left[(r_{Fi} - r_{Fj}) - (r_i - r_j)\right] + k_T \hat{g}_i\right), \\ \dot{\phi}_i(t) =& -k_\phi e_{\phi_i} + \dot{\phi}_{di}, \end{aligned} \tag{6.4}$$

where

$$R(e_{\phi_i}) = \begin{bmatrix} \cos^2(e_{\phi_i}) & -\cos(e_{\phi_i})\sin(e_{\phi_i}) \\ \cos(e_{\phi_i})\sin(e_{\phi_i}) & \cos^2(e_{\phi_i}) \end{bmatrix}.$$

Taking $\dot{r}_i = 0$ and $\dot{\phi}_i = \dot{\phi}_{di} = 0$, and considering $e_{\phi_i} \neq \frac{\pi}{2}$, then $R(e_{\phi_i}) \neq 0$ for all $e_{\phi_i} \neq \frac{\pi}{2}$. Equation (6.4) possesses an equilibrium point given by

$$\begin{aligned} k_F \sum_{j \in \mathcal{N}_i} a_{ij} \left[(r_{Fi} - r_{Fj}) - (r_i^* - r_j^*) \right] + k_T \hat{g}_i^* =& 0, \\ \phi_i^* =& \phi_{di}^*, \end{aligned} \tag{6.5}$$

or in vector form

$$\begin{aligned} k_F \hat{\mathcal{L}}(r_F - r^*) + k_T \hat{g}^* =& 0, \\ \phi^* =& \phi_d^*, \end{aligned} \tag{6.6}$$

where $\hat{\mathcal{L}} = \mathcal{L} \otimes I_2$. Note that the formation's equilibrium point is similar to the equilibrium point of the group of agents modelled as single integrators.

In order to analyse the stability, define the position error as

$$e_{pi} = r_i - r_i^*, \tag{6.7}$$

then the dynamic errors can be written as

$$\dot{e}_{pi} = R(e_{\phi_i}) \left(-k_F \sum_{j \in \mathcal{N}_i} a_{ij}(e_{pi} - e_{pj}) + k_T(\hat{g}_i - \hat{g}_i^*) \right). \tag{6.8}$$

Deriving the orientation error e_{pi} and taking into account Equation (6.4), the orientation error's dynamics is given by

$$\dot{e}_{\phi i} = - k_\phi e_{\phi i}, \tag{6.9}$$

then the rotation matrix $R(e_{\phi_i}) = I_2$ when $t \to \infty$ and this dynamic is stable if $k_\phi > 0$. Therefore Equations (6.8) and (6.9) can be written as

$$\begin{bmatrix} \dot{e}_p \\ \dot{e}_\phi \end{bmatrix} = - \begin{bmatrix} k_F \hat{R}_{e\phi} \hat{L} & 0 \\ 0 & k_\phi I \end{bmatrix} \begin{bmatrix} e_p \\ e_\phi \end{bmatrix} + k_T \hat{R}_{e\phi} \begin{bmatrix} e_g \\ 0 \end{bmatrix}, \tag{6.10}$$

where $e_p \in \mathbb{R}^{2N}$, $e_g \in \mathbb{R}^{2N}$, $e_\phi \in \mathbb{R}^N$, $\hat{R}_{e\phi} \in \mathbb{R}^{2N\text{x}2N}$
with $\hat{R}_{e\phi} = \text{diag}(R(e_{\phi_1}), \ldots, (R(e_{\phi_N}))$, $\hat{L} = \mathcal{L} \otimes I_2$, $e_p = [e_{p1}^T, \ldots, e_{pN}^T]$, $e_g = [e_{g1}^T, \ldots, e_{gN}^T]$ and $e_{gi} = \hat{g}_i - \hat{g}_i^*$. The orientation error e_ϕ converges to zero since $k_\phi > 0$. The parameter k_ϕ defines the orientation error's transient speed, this means, if $k_\phi \gg 0$, agents achieve the desired angles rapidly. As well as e_ϕ goes to zero, the matrix $\hat{R}_{e\phi}$ goes to I, since $\hat{R}_{e\phi}$ depends on e_ϕ. Then, agents achieve the equilibrium points around the scalar field's source.

Note that so far, it has been shown that a formation of nonholonomic agents reach the scalar field's source without the effect of collision avoidance between agents and collision avoidance between agents and obstacles. However, implemented simulations based on the distributed control law proposed in Equation (6.2) show that a formation of nonholonomic agents can avoid collision between agents, can evade collision between agents and obstacles and can maintain the desired formation, while they are seeking the scalar field's source. Some progress in formation control of non-holonomic agents with only collision avoidance has been made in (Ahmadi et al., 2015).

6.5 Results

In this section, simulation results illustrating convergence of the proposed algorithm are provided. Following simulations are displayed:

- Formation control only.
- Formation control and collision avoidance between agents.
- Formation control, source seeking and collision avoidance between agents.
- Formation control, source seeking and collision avoidance between agents and obstacles.

The scalar field is defined as

$$\psi(r) = A_0 \left(e^{-(r^T H_1 r)} + e^{-(r^T H_2 r)} \right),$$

where $A_0 = 0.5$, $H_1 = \begin{bmatrix} \frac{1}{2\sigma_{x1}^2} & 0 \\ 0 & \frac{1}{2\sigma_{y1}^2} \end{bmatrix}$, $H_2 = \begin{bmatrix} \frac{1}{2\sigma_{x2}^2} & 0 \\ 0 & \frac{1}{2\sigma_{y2}^2} \end{bmatrix}$, $\sigma_{x1} = 30$, $\sigma_{y1} = 75$, $\sigma_{x2} = 80$ and $\sigma_{y2} = 25$.

The scalar field's source is located at $r_s = [70\ 70]^T$. Scalar field's contour is shown in Fig. 1.2. A formation of 7 agents with the undirected communication topology shown in Fig. 1.4 is considered.

The desired formation $r_{Fi} = [x_{Fi}\ y_{Fi}\ z_{Fi}]^T$ is defined as shown in the table 5.1, and the initial positions $q_{0i} = [x_{0i}\ y_{0i}\ \phi_{0i}]^T$ are set as shown in Table 6.1, respectively.

Simulation results are presented ahead.

1. **Formation control only.**

   Fig. 6.2 shows the agents' trajectories described by control law (6.2). In this case, only the control law's formation control part has been applied. Agents start from their initial positions and with their initial orientations as shown in table 6.1. Tuning parameters are chosen as $k_F = 1$, $k_\phi = 0.2$ and $k_T = k_a = k_o = 0$. As shown in Fig. 6.2, all agents reach the desired formation defined in Fig. 1.4.

	x_{0i}	y_{0i}	ϕ_{0i}
Agent 1	1	-4	0
Agent 2	13	9	0
Agent 3	5	15	90
Agent 4	-15	1	0
Agent 5	-3	4	90
Agent 6	7	2	30
Agent 7	7	2	30

Table 6.1: Initial positions and initial orientations for non-holonomic multi-agent systems

2. **Formation control and collision avoidance between agents.**

   Fig. 6.3 shows agents' trajectories as described by control law (6.2). Furthermore, only the control law's formation control part and collision avoidance function are applied. Agents start from different initial positions and with different initial orientations. Tuning parameters are chosen as $k_F = 1$, $k_\phi = 0.2$, $k_a = 1$ and $k_T = k_o = 0$. Note that all agents reach the desired formation defined in Fig. 1.4. and agents avoid entering in the detection region of other agents. Each agent considers neighbour agents as obstacles and all agents try to maintain the formation all the time.

3. **Formation control, source seeking and collision avoidance between agents.**

   Fig. 6.4 shows the agents' trajectories mentioned above. Three parts of the control law are applied here: formation control part, collision avoidance function and estimated gradient. Agents start moving ahead from different initial positions and different initial orientations. The tuning parameters are chosen as $k_F = 1$, $k_\phi = 0.2$, $k_a = 1$, $k_T = 1$ and $k_o = 0$. Note that the group of agents move in the gradient direction while agents reach the desired formation and avoid entering in the detection region of other agents. After some time, agents locate the scalar field's source and maintain their positions.

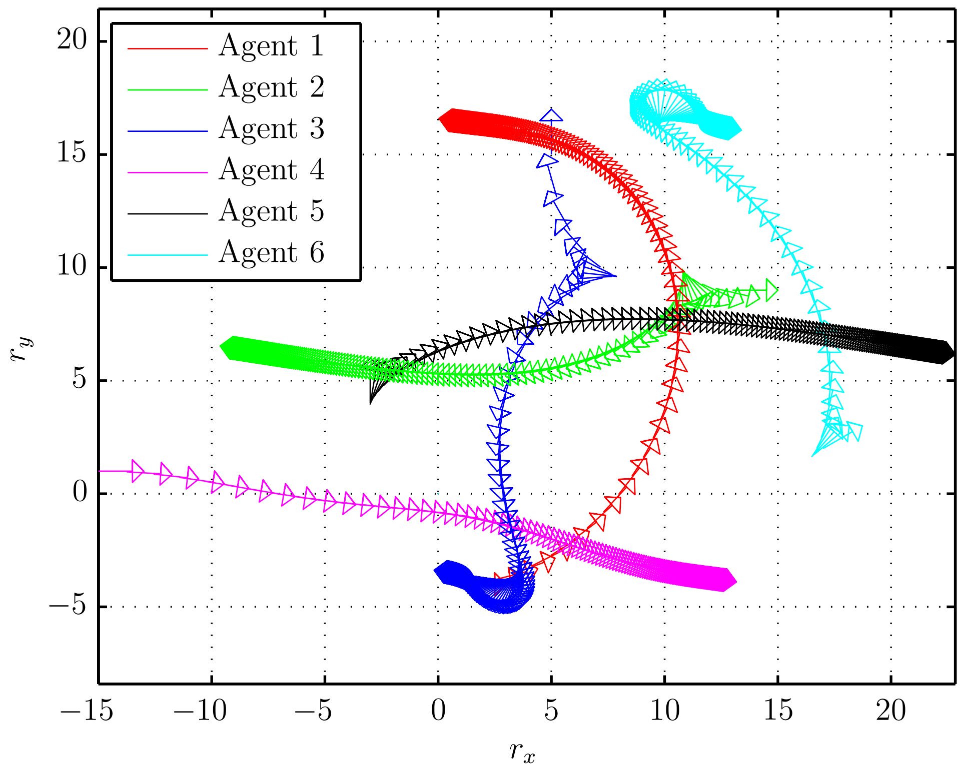


Figure 6.2: Formation control

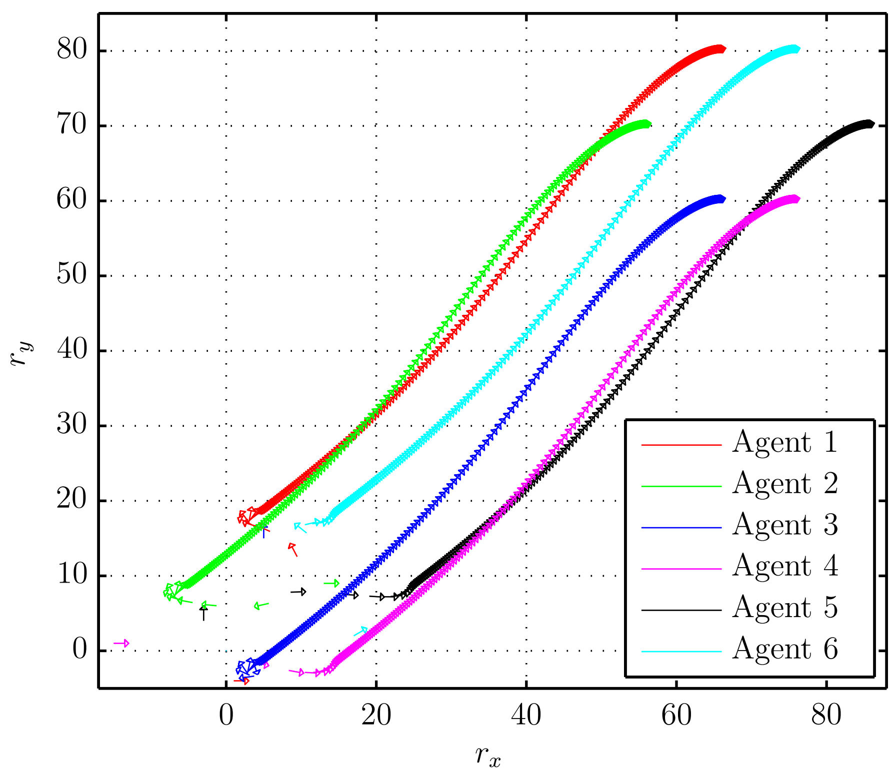


Figure 6.4: Formation control, collision avoidance between agents and tracking of the gradient direction

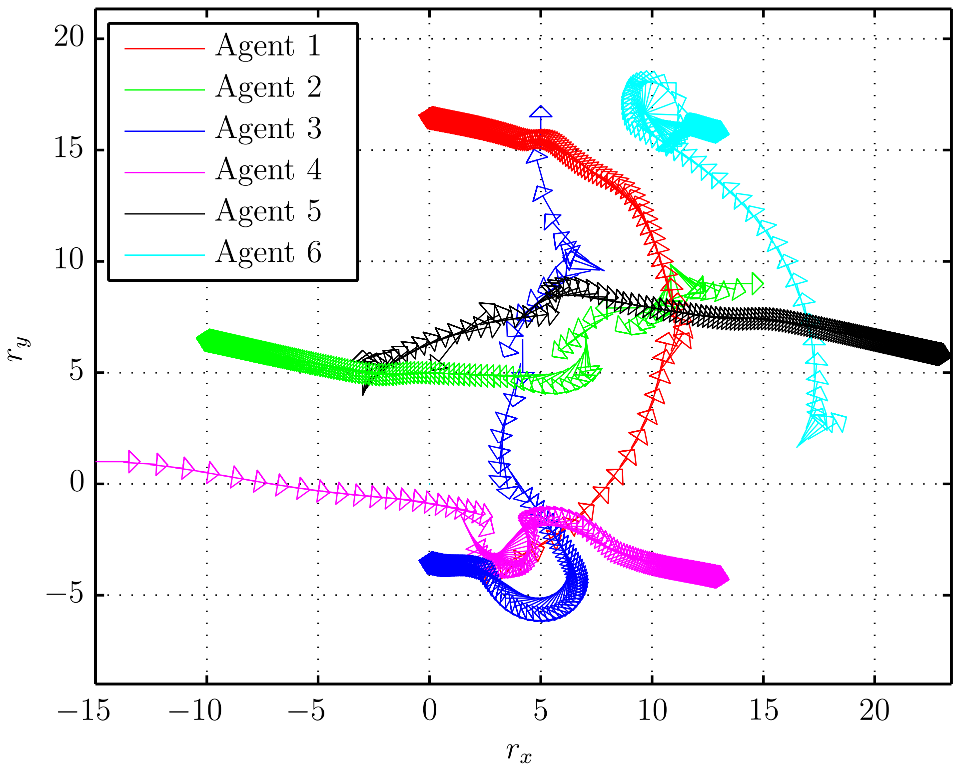


Figure 6.3: Formation control and collision avoidance between agents

4. **Formation control, collision avoidance between agents and obstacles and source seeking.**

Fig. 6.5 shows the same agents' trajectories depicted above. All control law's components are applied: formation control, collision avoidance function between agents, collison avoidance function between agents and obstacles, and gradient estimation. Agents start from different initial positions and have different initial orientations. Tuning parameters are chosen as $k_F = 1$, $k_\phi = 0.2$, $k_a = 1$, $k_T = 1$ and $k_o = 1$. As can be seen, the group of agents move in the gradient direction while agents reach the desired formation and avoid entering other agents' detection range. If agents find obstacles, they avoid them and move in the estimated gradient's direction. After some time they locate the source of the scalar field and maintain their positions.

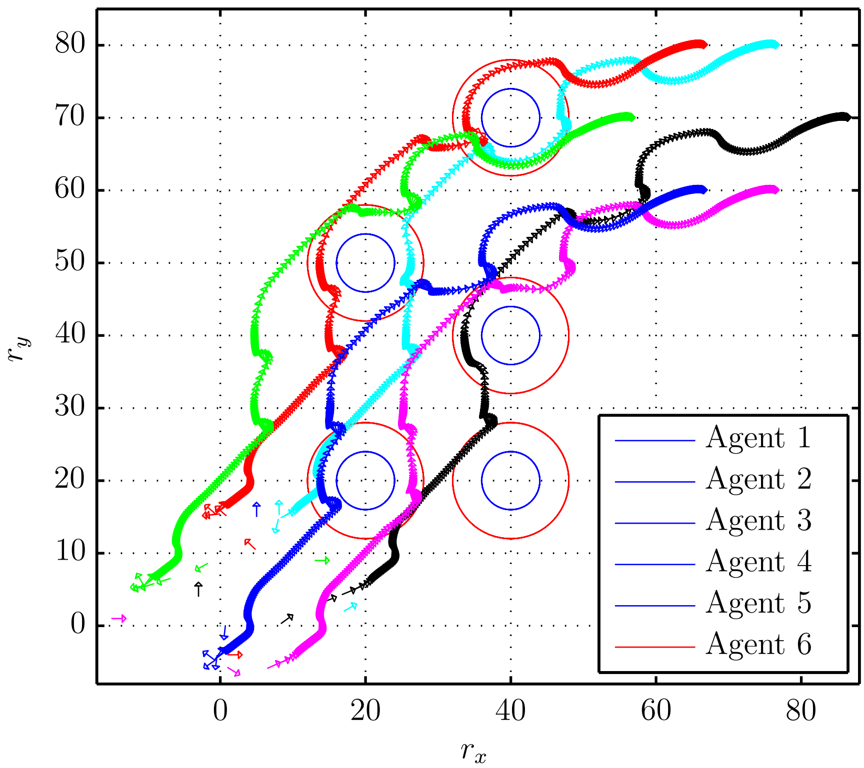


Figure 6.5: Formation control, collision avoidance between agents and obstacles, and source seeking

6.6 Conclusions

In this chapter, both the source seeking problem and collision avoidance problem for non-holonomic systems have been successfully solved by a simple distributed controller. This controller is composed of three parts: the formation control part maintains the group of agents in a desired formation, the trajectory control part provides agents the direction in which agents' networks shall move to locate the scalar field's source, and an avoidance function allows agents to evade collisions between agents and obstacles. A stability analysis has been provided to study overall network's stability.

Chapter 7

Cooperative Level Curve Tracking

In this chapter, two distributed controllers enabling multi-agent systems to move in a defined formation, along a defined level curve, into a a scalar field are proposed. Here, agents are modelled as double integrators and LTI systems. As in previous Chapters, agents exchange information with neighbours using only a constrained communication topology. This control strategy maintains agents in a desired formation and steers the whole group towards an aimed level curve. Convergence of the presented control laws is proved. Results are supported by simulations for double integrator models and quad-rotor helicopter formations. These results confirm that the presented control laws can be successfully applied.

7.1 Introduction

Information collected via level curve tracking inside any substance's concentration field allows the synthesis of temperature maps, pressure maps, altitude maps, salinity charts or concentration charts of toxic substances. These are few examples of how improving this technology will contribute to the study of environmental scenarios in the future.

The problem of spatial mapping and level curve tracking has been addressed by (Ogren et al., 2004; Williams and Sukhatme, 2012; Zhang et al., 2007; Zhang and Leonard, 2010). However, these approaches require the group of agents to maintain special formation shapes, as well as to gather information about estimated gradient, estimated Hessian matrix and position of the formation's center of mass.

In this chapter, based on the cooperative source seeking strategy for single, double integrator models and LTI systems presented in Chapters 4 and 5, the problem of tracking a level curve inside an unknown scalar field is addressed using an N identical agents formation, whereas agents operate under undirected constrained communication. To solve the **level curve tracking problem** as defined in Chapter 1, new distributed control laws will be presented. If signals are corrupted by noise, distributed consensus filters are used, as explained in Chapter 3. Note that for implementing the solution here presented, in contrast to previously cited authors, it is not necessary to know or estimate gradient and Hessian matrix at the center of mass. Taking the present approach the center of mass' position needs not to be known.

This chapter is organized as follows. In Section 7.2, problem statement and definition of level curves are introduced. In Sections 7.3 and 7.4, distributed control laws and stability analysis for double integrator models and LTI systems are presented. Simulation results illustrate the feasibility of the proposed approach in Section 7.5. Finally, Section 7.6 presents conclusions to this chapter.

7.2 Background and Problem Statement

7.2.1 Level Curves

A level curve is a two dimensional ($p = 2$) curve that builds a path within the domain of a function. Along this path the function values remain constant. For example, if the function represents temperature on a plate's surface, then temperature values are constant along the level curve. Level curves and contour plots are one way of visualizing functions of two variables.

Consider a scenario which can be described by a scalar field such as an area with changing concentration levels of radiation, temperature, pressure or a toxic substance. Consider a differentiable scalar field $\psi = \psi(r)$, described by a mapping $\psi : \mathbb{R}^2 \to \mathbb{R}$, where $r = [r_x \ r_y]^T \in \mathbb{R}^2$ define agents' position in space. The following definition has been made.

Let c be a constant. A level curve of a function $\psi(r_x, r_y)$ is a curve in the $r_x r_y$-plane such that, for all points (r_x, r_y) on the curve, $\psi(r_x, r_y) = c$. Then, the level curves of $\psi(r)$ are the horizontal traces of the graph of ψ in the plane $z = c$, projected onto the $r_x r_y$-plane. A graph depicting this level curves is called a contour plot.

7.2.2 Problem Statement

The **level curve tracking problem** is defined as follows: For a given N agents formation with undirected communication topology, find a distributed control law $u_i(t)$ that allows the formation to move along a given level curve inside an unknown scalar field.

In the next sections, control laws for double integrator agents and quadcopter helicopter agents are presented.

7.3 Cooperative Level Curve Tracking for Double Integrator Agents

The dynamics of each agent i in a network modelled as double integrator are described by

$$\begin{aligned}\dot{r}_i(t) &= v_i(t),\\ \dot{v}_i(t) &= u_i(t),\end{aligned} \tag{7.1}$$

where $i = 1, ..., N$, $r_i(t) \in \mathbb{R}^p$ is the position vector, $v_i \in \mathbb{R}^p$ is the velocity vector, and $u_i(t) \in \mathbb{R}^p$ is the acceleration's control input. To solve the level curve tracking problem, a distributed control law consisting of two parts is presented: the two parts are, first, a formation control law u_{Fi} and second, a trajectory control law u_{Ti} as

$$\begin{aligned}u_i(t) =& u_{Fi}(t) + u_{Ti}(t),\\ u_{Fi}(t) =& k_F \sum_{j\in\mathcal{N}_i} a_{ij}[(r_{Fi}(t) - r_{Fj}(t)) - (r_i(t) - r_j(t)) - \theta(v_i(t) - v_j(t))],\\ u_{Ti}(t) =& k_\psi(\psi_{ref} - \psi_i(r_i(t)))\frac{\hat{g}_i(r_i(t))}{\|\hat{g}_i(r_i(t))\|} + k_T\left(T\frac{\hat{g}_i(r_i(t))}{\|\hat{g}_i(r_i(t))\|} - \frac{\gamma}{k_T}v_i(t)\right),\end{aligned} \tag{7.2}$$

where

$$\hat{g}_i(r_i) = \left(\sum_{j\in\mathcal{N}_i} a_{ij}(r_i - r_j)(r_i - r_j)^T\right)^{-1}\left(\sum_{j\in\mathcal{N}_i} a_{ij}(r_i - r_j)(\psi(r_i) - \psi(r_j))\right)$$

is the distributed estimated gradient computed in each agent i in Chapter 2, $u_{Fi}(t)$ keeps agents in their desired relative positions, $u_{Ti}(t)$ drives the whole group of agents to a desired level curve of the scalar field, $r_{Fi}(t)$ denotes a formation's desired reference signal for agent i and $k_\psi > 0$, $k_F > 0$, $k_T > 0$, $\gamma > 0$, $\theta > 0$ are scalar control parameters in the formation. Since level curve tracking implies movement towards the orthogonal direction of the gradient, a rotation matrix T is chosen for counterclockwise rotation as $T = \begin{bmatrix} 0 & -1\\ 1 & 0\end{bmatrix}$, or clockwise rotation $-T$ in the case of $\mathbb{R}^2$. The term $T\frac{\hat{g}_i(r_i(t))}{\|\hat{g}_i(r_i(t))\|}$ drives agents in orthogonal direction of the gradient at constant velocity. The term $k_\psi(\psi_{ref} - \psi_i(t))\frac{\hat{g}_i}{\|\hat{g}_i\|}$ moves agents in the gradient direction, when there is an error between current and desired level curve. The error $\psi_{ref} - \psi_i(t)$ depends on the difference between current and desired level curve. For this reason, when agents are far from a desired level curve the gain of this term is large and cause fast movements; while, if agents are close to a desired level curve the gain will be smaller or zero. This term multiplied by the estimated gradient moves agents in the gradient direction, reducing the error between current and desired level curve.

Let $r(t) = [r_1^T(t), \ldots, r_N^T(t)]^T$, $r_F(t) = [r_{F1}^T(t), \ldots, r_{FN}^T(t)]^T$, $v(t) = [v_1^T(t), \ldots, v_N^T(t)]^T$, and $\hat{g}(t) = \left[\frac{\hat{g}_1^T}{\|\hat{g}_1\|}, \ldots, \frac{\hat{g}_N^T}{\|\hat{g}_N\|}\right]^T$; the closed-loop dynamics can be written as

$$\begin{bmatrix} \dot{r} \\ \dot{v} \end{bmatrix} = \Sigma \begin{bmatrix} r \\ v \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ k_F\mathcal{L} & I_N \end{bmatrix} \begin{bmatrix} r_F \\ \left(k_\psi P(\psi) + k_T\bar{T}\right)\hat{g} \end{bmatrix}, \tag{7.3}$$

where

$$\Sigma = \begin{bmatrix} 0 & I_N \\ -k_F\mathcal{L} & -k_F\theta\mathcal{L} - \gamma I_N \end{bmatrix},$$

and $P(\psi) = diag\left[\psi_{ref} - \psi_1, \ldots, \psi_{ref} - \psi_N\right] = P(\psi) \otimes I_p$, $\bar{T} = diag\left[T, \ldots, T\right] = I_N \otimes T$.

Let $\bar{r}(t) = \frac{1}{N}\sum_{i\in\mathcal{V}} r_i(t) = \frac{1}{N}\mathbf{1}^T r(t)$ be the average of the position states, $\bar{v}(t) = \frac{1}{N}\sum_{i\in\mathcal{V}} v_i(t) = \frac{1}{N}\mathbf{1}^T v(t)$ be the average of the velocity states, and $\bar{\hat{g}}(r(t)) = \frac{1}{N}\sum_{i\in\mathcal{V}} \hat{g}_i(r(t)) = \frac{1}{N}\mathbf{1}^T\hat{g}(r(t))$ be the average of the estimated gradient, $\bar{f} = \frac{1}{N}\mathbf{1}^T P(\psi)\hat{g}$ and $\bar{h} = \frac{1}{N}\mathbf{1}^T\bar{T}\hat{g}$. Since $\mathbf{1}^T\mathcal{L} = \mathbf{0}^T$, the time derivative of both the position states average $\bar{r}$ and the velocity states average $\bar{v}$ is given by

$$\begin{aligned} \dot{\bar{r}}(t) &= \bar{v}(t), \\ \dot{\bar{v}}(t) &= -\gamma\bar{v}(t) + k_\psi\bar{f} + k_T\bar{h}, \end{aligned} \tag{7.4}$$

with initial average $\bar{r}(0) = \frac{1}{N}\mathbf{1}^T r(0) = \bar{r}_0$, $\bar{v}(0) = \frac{1}{N}\mathbf{1}^T v(0) = \bar{v}_0 = 0$ and $\bar{\hat{g}}(r(0)) = \frac{1}{N}\mathbf{1}^T\hat{g}(r(0)) = \bar{\hat{g}}_0$. Again, note that the agents' formation moves in the orthogonal direction of the gradient according to the term $k_T\bar{h}$, and the agents' formation moves in the gradient direction according to the term $k_\psi\bar{f}$, to eliminate the error between the current and desired level curve. Note that the terms $k_T\bar{h}$ and $k_\psi\bar{f}$ depend on the estimated gradient and these two terms generate a perpendicular vector.

Again, the proposed distributed controller has two parts: formation controller and tracking controller. The formation controller maintains agents in a desired relative position and the tracking controller steers agents in the perpendicular direction of the gradient in order to move the agent along a desired level curve inside an unknown scalar field.

To find the equilibrium points set the Equation (7.3) as

$$\begin{aligned} \dot{r}^* =& v^*, \\ 0 =& k_F\mathcal{L}(r_F - r^*) - (k_F\theta\mathcal{L} + \alpha I)v^* + k_\psi f^* + k_T h^*, \end{aligned} \tag{7.5}$$

where $r^* = [r_1^{*T}, \ldots, r_N^{*T}]^T$, $v^* = [v_1^{*T}, \ldots, v_N^{*T}]^T$, $f^* = P(\psi^*)\hat{g}(r^*)$ and $h^* = \bar{T}\hat{g}(r^*)$. Note that since agents need to move along a defined level curve at constant velocity, then the velocity is set to $\dot{r}^* = v^*$. Then the equilibrium points are given by

$$\begin{aligned} r^* =& \int_0^t v^* d\tau, \\ v^* =& \left(k_F\theta\mathcal{L} + \gamma I\right)^{-1}\left(k_F\mathcal{L}(r_F - r^*) + k_\psi f^* + k_T h^*\right). \end{aligned} \tag{7.6}$$

In order to discuss convergence, the state $r(t)$ can be decomposed as

$$r(t) = \frac{1}{N}\mathbf{1}\mathbf{1}^T r(0) + \delta(t),$$

where $\mathbf{1}$ is a vector of ones and δ is referred to as disagreement vector. By definition, the disagreement vector has zero average, i.e., $\mathbf{1}^T\delta(t) \equiv 0$ (Olfati-Saber and Murray, 2004).

In order to solve the level curve tracking problem for double integrator agents, the following theorem is presented.

Theorem 5. *Consider the multi-agent system (7.1) with control law (7.2). Suppose that assumptions 1 and 2 are fulfilled. Then, for all $r_i(0) \in \mathbb{R}^2$ and $t \geq 0$, agents track an unknown scalar field's level curve $\psi(r)$ and the disagreement vector δ of the closed-loop system converges to a ball centred at the origin with radius*

$$\epsilon = \frac{2c_d\sqrt{N}}{\kappa_2}(k_T + ak_\psi). \tag{7.7}$$

Proof. Consider the position error $e_p(t) = r(t) - r^*$ and velocity error $e_v(t) = v(t) - v^*$. The dynamic error can be computed as

$$\begin{bmatrix} \dot{e}_p \\ \dot{e}_v \end{bmatrix} = \Sigma \begin{bmatrix} e_p \\ e_v \end{bmatrix} + \begin{bmatrix} 0 \\ k_\psi e_f + k_T e_h \end{bmatrix}, \tag{7.8}$$

where $e_f = f - f^* = P(\psi)\hat{g}(e_p + r^*) - P(\psi^*)\hat{g}(r^*)$ and $e_h = h - h^* = \bar{T}\left(\hat{g}(r) - \hat{g}(r^*)\right)$. Define the average of position error states $\bar{e}_p(t) = \frac{1}{N}\sum_{i\in\mathcal{V}} e_{pi}(t) = \frac{1}{N}\mathbf{1}^T e_p(t) = \bar{r} - \bar{r}^*$, and the average of the velocity error states $\bar{e}_v(t) = \frac{1}{N}\sum_{i\in\mathcal{V}} e_{vi}(t) = \frac{1}{N}\mathbf{1}^T e_v(t) = \bar{v} - \dot{\bar{r}}^*$. Then, the state vector can be decomposed according to

$$\begin{aligned} e_p(t) &= \mathbf{1}\bar{e}_p(t) + \delta_p(t), \\ e_v(t) &= \mathbf{1}\bar{e}_v(t) + \delta_v(t), \end{aligned} \tag{7.9}$$

such that the disagreement vectors $\delta_p(t)$ and $\delta_v(t)$ have zero average, i.e., $\mathbf{1}^T\delta_p(t) \equiv \mathbf{1}^T\delta_v(t) \equiv 0$. Derivation of Equation (7.9) with respect to time t and considering that $v(0) = 0$, yields

$$\begin{bmatrix} \dot{\delta}_p \\ \dot{\delta}_v \end{bmatrix} = \Sigma \begin{bmatrix} \delta_p \\ \delta_v \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ k_\psi\mathcal{M} & k_T\mathcal{M} \end{bmatrix} \begin{bmatrix} e_f \\ e_h \end{bmatrix}, \tag{7.10}$$

where $\mathcal{M} = I_N - \frac{1}{N}\mathbf{1}\mathbf{1}^T$. With $\delta(t) = [\delta_p(t)^T \;\; \delta_v(t)^T]^T$, $e_T = [e_f^T \;\; e_h^T]^T$, and $\Psi = \begin{bmatrix} 0 & 0 \\ k_\psi\mathcal{M} & k_T\mathcal{M} \end{bmatrix}$. The disagreement dynamics are given by

$$\dot{\delta}(t) = \Sigma\delta(t) + \Psi e_T, \tag{7.11}$$

and the solution is given by

$$\delta(t) = e^{\Sigma t}\delta(0) + \int_0^t e^{\Sigma(t-\tau)}\Psi e_T d\tau.$$

Then the disagreement vector is bounded by

$$\|\delta(t)\| \leq \|e^{\Sigma t}\delta(0)\| + \int_0^t \|e^{\Sigma(t-\tau)}\Psi e_T\| d\tau.$$

Applying Equation (4.31) and using $\kappa_2 = \kappa_2(\Sigma)$ yields

$$\begin{aligned}\|\delta(t)\| \leq & c_d e^{-\kappa_2 t}\|\delta(0)\| \\ & + 2c_d\sqrt{N}(k_T + ak_\psi)\int_0^t e^{-\kappa_2(t-\tau)}d\tau,\end{aligned}$$

because $\|\Psi e_T\| \leq k_\psi\|\mathcal{M}\|\|e_f\| + k_T\|\mathcal{M}\|\|e_h\|$, $\|e_f\| \leq \|P(\psi)\hat{g} - P(\psi^*)\hat{g}^*\| \leq 2a\sqrt{N}$, $\|e_h\| \leq \|\bar{T}\hat{g} - \bar{T}\hat{g}^*\| \leq 2\sqrt{N}$, $\|\mathcal{M}\| = 1$, $\|\bar{T}\| = 1$, $\|P(\psi)\| = a$, $\|\tilde{g}_i\| \leq \mu$, and $\|\hat{g}\| \leq \sqrt{N}$. Then $\|\Psi e_T\| \leq 2\sqrt{N}(k_T + ak_\psi)$. Finally

$$\begin{aligned}\|\delta(t)\| \leq & \frac{2c_d\sqrt{N}}{\kappa_2}(k_T + ak_\psi) \\ & + \left(c_d\|\delta(0)\| - \frac{2c_d\sqrt{N}}{\kappa_2}(k_T + ak_\psi)\right)e^{-\kappa_2 t}. \end{aligned} \tag{7.12}$$

Since $\mathbf{1}^T\mathcal{L} = \mathbf{0}^T$, the time derivative of $\bar{e}_p$ and $\bar{e}_v$ is given by

$$\begin{aligned}\dot{\bar{e}}_p(t) &= \bar{e}_v(t), \\ \dot{\bar{e}}_v(t) &= -\gamma\bar{e}_v(t) + k_\psi\bar{e}_f(e_p) + k_T\bar{e}_h(e_p),\end{aligned} \tag{7.13}$$

with initial average $\bar{e}_p(0) = \frac{1}{N}\mathbf{1}^T e_p(0) = \bar{e}_{p0}$, $\bar{e}_v(0) = \frac{1}{N}\mathbf{1}^T e_v(0) = \bar{e}_{v0}$, $\bar{e}_f = \frac{1}{N}\mathbf{1}^T\left(P(\psi)\hat{g}(e_p + r^*) - P(\psi^*)\hat{g}(r^*)\right)$, $\bar{e}_f$ satisfies $\bar{e}_f(0) = 0$, and $\bar{e}_p^T\bar{e}_f(e_p) < 0$ for all $e_p \neq 0$; $\bar{e}_h = \frac{1}{N}\mathbf{1}^T\left(\bar{T}\hat{g}(e_p + r^*) - \bar{T}\hat{g}(r^*)\right)$, $\bar{e}_h$ satisfies $\bar{e}_h(0) = 0$, and $\bar{e}_p^T\bar{e}_h(e_p) < 0$ for all $e_p \neq 0$.

Each agent's state converges to the consensus dynamics' solution of Equation (7.13) if the graph is undirected and consensus is achieved. Then the equilibrium is stable, because agents starting on any place of the scalar field move towards the desired level curve at constant velocity.

To arrive at the same conclusion, consider the system

$$\begin{aligned}\dot{\bar{e}}_p(t) =& \bar{e}_v(t), \\ \dot{\bar{e}}_v(t) =& -\gamma\bar{e}_v(t) - s_f(e_p) - s_h(e_p),\end{aligned}$$

where $s_f(e_p) = -k_\psi\bar{e}_f(e_p)$ and satisfies $s_f(0) = 0$; $\bar{e}_p^T(e_p)s_f(e_p) > 0$ for all $e_p \neq 0$. $s_h(e_p) = -k_\psi\bar{e}_h(e_p)$ and satisfies $s_h(0) = 0$; $\bar{e}_p^T(e_p)s_h(e_p) > 0$ for all $e_p \neq 0$.

Consider the Lyapunov function candidate with line integrals as

$$V = \frac{1}{2}\bar{e}_v^T\bar{e}_v + \int_0^{\bar{e}_p} s_f^T(z)dz + \int_0^{\bar{e}_p} s_h^T(z)dz, \tag{7.14}$$

where V is continuously differentiable, $V(0) = 0$, and $V > 0$ for all $e_p \neq 0$ and $e_v \neq 0$. Then

$$\dot{V} = -\gamma\bar{e}_v^T\bar{e}_v \leq 0. \tag{7.15}$$

Applying the Lasalle's principle, it is possible to conclude that the equilibrium point is stable because $\bar{e}_v = 0 \Rightarrow \bar{e}_p = 0 \Rightarrow \dot{\bar{e}}_v = 0 = -s_f(0) = -s_h(0) = 0$. $\square$

7.4 Cooperative Level Curve Tracking for LTI Models

Consider a group of N identical agents with linear time-varying dynamics described by

$$\begin{aligned}
\dot{x}_i(t) &= A_p x_i(t) + B_u u_i(t) + B_w \bar{w}_i(t), \\
z_i(t) &= C_z x_i(t) + D_{zu} u_i + D_{zw} \bar{w}_i(t), \\
r_i(t) &= C_y x_i(t) + D_{yw} \bar{w}_i(t),
\end{aligned} \tag{7.16}$$

where $x_i \in \mathbb{R}^n$ is the state of agent i, $u_i \in \mathbb{R}^{m_1}$ is the control input, $\bar{w}_i \in \mathbb{R}^{m_2}$ is the external disturbance and noise, $r_i \in \mathbb{R}^p$ is the measured position output, and $z_i \in \mathbb{R}^q$ is the controlled output of agent i. It is assumed that $(A_p,\ B_u)$ is stabilizable, $(A_p,\ C_y)$ is detectable, and without loss of generality, B_u is full column rank.

Based on the neighbours' relative output measurements between agents, a distributed dynamic output feedback controller for agent i is considered as

$$\begin{aligned}
\dot{v}_i(t) =& A_K v_i(t) + B_K(e_{Fi}(t) + e_{Ti}(t)), \\
u_i(t) =& C_K v_i(t) + D_K(e_{Fi}(t) + e_{Ti}(t)), \\
e_{Fi}(t) =& \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} a_{ij} \left[(r_{Fi}(t) - r_{Fj}(t)) - (r_i(t) - r_j(t)) \right], \\
e_{Ti}(t) =& v_{Ti}(t) = k_\psi(\psi_{ref} - \psi_i) \frac{\hat{g}_i(r_i(t))}{\|\hat{g}_i(r_i(t))\|} + k_T T \frac{\hat{g}_i(r_i(t))}{\|\hat{g}_i(r_i(t))\|},
\end{aligned} \tag{7.17}$$

where

$$\hat{g}_i(r_i) = \left(\sum_{j \in \mathcal{N}_i} a_{ij}(r_i - r_j)(r_i - r_j)^T \right)^{-1} \left(\sum_{j \in \mathcal{N}_i} a_{ij}(r_i - r_j)(\psi(r_i) - \psi(r_j)) \right)$$

is the distributed estimated gradient computed in each agent i in Chapter 2, and $v_i(t) \in \mathbb{R}^{m_K}$ is the state of the dynamic output feedback controller.

Let $x = [x_1^T, \ldots, x_N^T]^T$, $v = [v_1^T, \ldots, v_N^T]^T$, $\bar{w} = [\bar{w}_1^T, \ldots, \bar{w}_N^T]^T$,

$r_F = [r_{F1}^T, \ldots, x_{FN}^T]^T$, $v_T = [v_{T1}^T, \ldots, v_{TN}^T]^T$, and $z = [z_1^T, \ldots, z_N^T]^T$. Substituting the controller (7.17) into the system (7.16), the closed-loop network dynamics can be written as

$$\begin{aligned}
\begin{bmatrix}\dot{x}\\ \dot{v}\end{bmatrix} &= \begin{bmatrix} I_N \otimes A_p - \mathcal{L}\otimes B_u D_K C_y & I_N \otimes B_u C_K \\ -\mathcal{L}\otimes B_K C_y & I_N \otimes A_K \end{bmatrix}\begin{bmatrix}x\\ v\end{bmatrix} \\
&\quad + \begin{bmatrix} \mathcal{L}\otimes B_u D_K & I_N\otimes B_u D_K & I_N \otimes B_w - \mathcal{L}\otimes B_u D_K D_{yw} \\ \mathcal{L}\otimes B_K & I_N \otimes B_K & -\mathcal{L}\otimes B_K D_{yw}\end{bmatrix}\begin{bmatrix} r_F \\ v_T \\ \bar{w}\end{bmatrix}, \\
z &= \begin{bmatrix} I_N\otimes C_z - \mathcal{L}\otimes D_{zw} D_K C_y & I_N \otimes D_{zw} C_K\end{bmatrix}\begin{bmatrix}x\\ v\end{bmatrix} \\
&\quad + \begin{bmatrix} \mathcal{L}\otimes D_{zw} D_K & I_N\otimes D_K & I_N\otimes D_{zw} - \mathcal{L}\otimes D_{zw} D_K D_{yw}\end{bmatrix}\begin{bmatrix} r_F \\ v_T \\ \bar{w}\end{bmatrix}, \\
r &= \begin{bmatrix} I_N \otimes C_y & 0\end{bmatrix}\begin{bmatrix}x\\ v\end{bmatrix} + \begin{bmatrix} 0 & 0 & I_N\otimes D_{yw}\end{bmatrix}\begin{bmatrix} r_F \\ v_T \\ \bar{w}\end{bmatrix},
\end{aligned} \tag{7.18}$$

where $v_T(t) = \left(k_\psi P(\psi) + k_T \bar{T}\right)\hat{g}$, $P(\psi) = diag\,[\psi_{ref} - \psi_1, \ldots, \psi_{ref} - \psi_N] = P(\psi)\otimes I_p$ and $\bar{T} = diag\,[T, \ldots, T] = I_N \otimes T$.

7.4.1 Controller Synthesis

The proposed distributed controller has two parts: formation controller and tracking controller. The formation controller maintains agents in a desired relative position and the tracking controller steers agents in the gradient's perpendicular direction in order to track a level curve.

Based on the previous works, (Pilz et al., 2009) and (Popov and Werner, 2009) a robust control approach to formation control is implemented. First, a local controller to stabilize a single quad-rotor helicopter is designed. Second, a robust formation controller to maintain agents in a desired relative position is designed.

To design a local controller K_L, a full state feedback LQR controller is used as $\zeta_i = K_L x_i + u_i$, where K_L stabilizes local dynamics of each agent i and u_i is the control law for the formation and tracking part. u_i steers agent's position in the formation and tracks the gradient direction. The new dynamics after applying a local controller are given by

$$\begin{aligned}
\dot{x}_i(t) &= (A_p + B_u K_L)x_i(t) + B_u u_i(t) + B_w \bar{w}_i(t),\\
\dot{x}_i(t) &= A x_i(t) + B_u u_i(t) + B_w \bar{w}_i(t),\\
z_i(t) &= C_z x_i(t) + D_{zu} u_i + D_{zw}\bar{w}_i(t),\\
r_i(t) &= C_y x_i(t) + D_{yw}\bar{w}_i(t),
\end{aligned} \tag{7.19}$$

where $A = A_p + B_u K_L$ and the matrix is Hurwitz. The formation controller K_{FT} is designed based on Theorems 3 and 4 of (Pilz et al., 2009). Performance requirements are incorporated into our design by means of mixed-sensitivity loop shaping. H_∞ synthesis technique is used to design the controller. A generalized plant construction with sensitivity $W_S(s)$ and control sensitivity $W_K(s)$ is shown in Fig. 7.1, where $w_p = r_F$. These theorems reduce the formation stability problem to an H_∞ design problem for a single agent with uncertainty. They also guarantee stability for any formation if controller K_{FT} satisfies the design requirements.

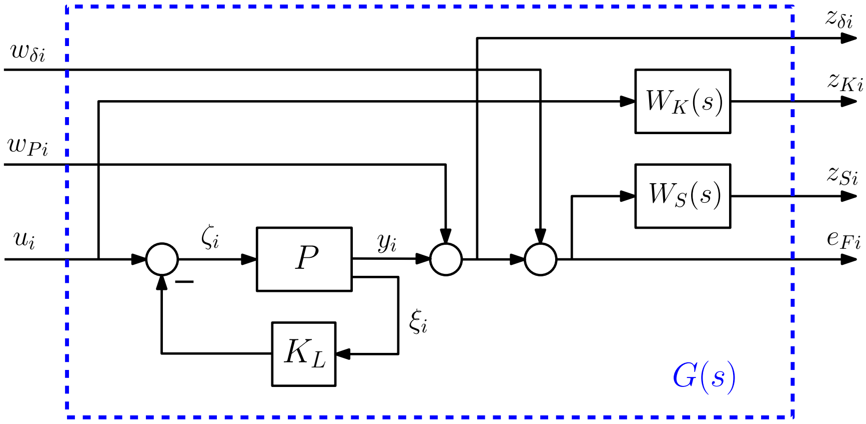


Figure 7.1: Generalized plant

7.4.2 Stability Analysis

Convergence of agents inside a scalar field is analysed as follows.

Theorem 6. *Consider the multi-agent system (7.16) with control law (7.17). Suppose that the assumptions 1, 2 and 3 are fulfilled. Then, for all $r_i(0) \in \mathbb{R}^2$ and $t \geq 0$, agents move along a given level curve of an unknown scalar field $\psi(r)$ and the closed-loop system converge to an equilibrium point (7.23) centred ball with radius*

$$\epsilon \leq \frac{2\sqrt{N} c_{\hat{A}} (k_\psi a + k_T) \|C_y\| \left(\|\tilde{B}\| + \|\mathcal{L}\| \|\tilde{G}\| \right)}{\lambda_{\hat{A}}}. \tag{7.20}$$

Proof. Consider the closed-loop system of (7.18), with $\bar{w} = 0$, as

$$\begin{aligned}
\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} &= \begin{bmatrix} I_N \otimes A - \mathcal{L} \otimes B_u D_K C_y & I_N \otimes B_u C_K \\ -\mathcal{L} \otimes B_K C_y & I_N \otimes A_K \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} \\
&\quad + \begin{bmatrix} \mathcal{L} \otimes B_u D_K & I_N \otimes B_u D_K \\ \mathcal{L} \otimes B_K & I_N \otimes B_K \end{bmatrix} \begin{bmatrix} r_F \\ v_T \end{bmatrix}, \\
r &= \begin{bmatrix} I_N \otimes C_y & 0 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix}.
\end{aligned} \tag{7.21}$$

Let $\zeta = [x^T \ v^T]^T$ and $\xi = [r_F^T \ v_T^T]^T$. Equation (7.21) can be written as

$$\begin{aligned}
\dot{\zeta} &= \hat{A}\zeta + \hat{B}\xi, \\
r &= \hat{C}\zeta,
\end{aligned} \tag{7.22}$$

where $\hat{A} = I_N \otimes \tilde{A} + \mathcal{L} \otimes \tilde{F}$ and this term is Hurwitz matrix, $\hat{B} = I_N \otimes \tilde{B} + \mathcal{L} \otimes \tilde{G}$, $\hat{C} = I_N \otimes \tilde{C}$, and

$$\begin{aligned}
\tilde{A} &= \begin{bmatrix} A & B_u C_K \\ 0 & A_K \end{bmatrix}, \ \tilde{F} = \begin{bmatrix} -B_u D_K C_y & 0 \\ B_K & 0 \end{bmatrix}, \\
\tilde{B} &= \begin{bmatrix} 0 & B_u D_K \\ 0 & B_K \end{bmatrix}, \ \tilde{G} = \begin{bmatrix} -B_u D_K & 0 \\ B_K & 0 \end{bmatrix}, \\
\tilde{C} &= \begin{bmatrix} C_y & 0 \end{bmatrix}.
\end{aligned}$$

The equilibrium point is given by

$$\begin{aligned}
\zeta^* =& \hat{A}^{-1} \hat{B} \xi^*, \\
r^* =& \hat{C} \zeta^*.
\end{aligned} \tag{7.23}$$

Let the state error $e_\zeta = \zeta - \zeta^*$ and the position error $e_r = r - r^*$. The dynamic state error can be written as

$$\dot{e}_\zeta = \hat{A} e_\zeta + \hat{B}(\xi - \xi^*), \tag{7.24}$$

where

$$\xi - \xi^* = \begin{bmatrix} 0 \\ k_\psi \left(P(\psi)\hat{g} - P(\psi^*)\hat{g}^* \right) + k_T \bar{T} \left(\hat{g} - \hat{g}^* \right) \end{bmatrix},$$

$\hat{g} = \hat{g}(e_r + r^*)$ and $\hat{g}^* = \hat{g}^*(r^*)$. The solution is given by

$$e_\zeta = e^{\hat{A}t} e_\zeta(0) + \int_0^t e^{\hat{A}(t-\tau)} \hat{B}(\xi - \xi^*) d\tau, \tag{7.25}$$

and the position error can be written as

$$\begin{aligned}
e_r &= \hat{C} e_\zeta, \\
&= \hat{C} e^{\hat{A}t} e_\zeta(0) + \hat{C} \int_0^t e^{\hat{A}(t-\tau)} \hat{B}(\xi - \xi^*) d\tau.
\end{aligned} \tag{7.26}$$

The solution is bounded by

$$\begin{aligned}
\|e_r\| \leq & c_{\hat{A}} \|\hat{C}\| \|e_\zeta(0)\| e^{-\lambda_{\hat{A}} t} + \\
& 2\sqrt{N} \mu (k_\psi a + k_T) c_{\hat{A}} \|\hat{C}\| \|\hat{B}\| \int_0^t e^{-\lambda_{\hat{A}}(t-\tau)} d\tau,
\end{aligned} \tag{7.27}$$

because $\|e^{\hat{A}t}\| \leq c_{\hat{A}} e^{-\lambda_{\hat{A}} t}$ and $\|\xi - \xi^*\| \leq 2\sqrt{N}(k_\psi a + k_T)$, since $\|P(\psi)\| \leq a$, $\|\bar{T}\| = 1$, $\|\hat{g}\| \leq \sqrt{N}$. It follows

$$\begin{aligned}
\|e_r\| \leq & \frac{2\sqrt{N} c_{\hat{A}} (k_\psi a + k_T) \|\hat{C}\| \|\hat{B}\|}{\lambda_{\hat{A}}} + \\
& c_{\hat{A}} \|\hat{C}\| \left(\|e_\zeta(0)\| - \frac{2\sqrt{N}(k_\psi a + k_T) \|\hat{B}\|}{\lambda_{\hat{A}}} \right) e^{-\lambda_{\hat{A}} t}.
\end{aligned} \tag{7.28}$$

Finally

$$\begin{aligned}\|e_r\| \leq & \frac{2\sqrt{N} c_{\hat{A}}(k_\psi a + k_T)\|C_y\| \left(\|\tilde{B}\| + \|\mathcal{L}\|\|\tilde{G}\|\right)}{\lambda_{\hat{A}}} + c_{\hat{A}}\|C_y\| \left(\|e_\zeta(0)\|\right) e^{-\lambda_{\hat{A}} t} - \\ & c_{\hat{A}}\|C_y\| \left(\frac{2\sqrt{N}(k_\psi a + k_T)\left(\|\tilde{B}\| + \|\mathcal{L}\|\|\tilde{G}\|\right)}{\lambda_{\hat{A}}}\right) e^{-\lambda_{\hat{A}} t}, \end{aligned} \tag{7.29}$$

since $\|\hat{C}\| \leq \|C_y\|$, $\|\hat{B}\| \leq \|\tilde{B}\| + \|\mathcal{L}\|\|\tilde{G}\|$. □

7.5 Simulation Results

To evaluate the proposed level curve tracking algorithms, a formation of 7 mobile agents with a communication graph $\mathcal{G}$ as shown in Fig. 1.4 is considered. The communication topology is fixed and undirected. The scalar field is defined as

$$\psi(r) = A_0 e^{-\left((r-r_s)^T H_1 (r-r_s)\right)} + A_0 e^{-\left((r-r_s)^T H_2 (r-r_s)\right)},$$

where $A_0 = 3$, $H_1 = \begin{bmatrix} \frac{1}{2\sigma_{x1}^2} & 0 \\ 0 & \frac{1}{2\sigma_{y1}^2} \end{bmatrix}$, $H_2 = \begin{bmatrix} \frac{1}{2\sigma_{x2}^2} & 0 \\ 0 & \frac{1}{2\sigma_{y2}^2} \end{bmatrix}$, $\sigma_{x1} = 30$, $\sigma_{y1} = 75$, $\sigma_{x2} = 80$ and $\sigma_{y2} = 25$. Fig. 1.2 shows the level curve described by the previous equations.

To estimate the gradient, Equation (2.3) should be implemented in absence of noise; when signals are corrupted by noise Equation (3.25) should be implemented. Initial conditions of the consensus filter are set as $\xi(0) = \nu(0) = 0$. Tuning parameters are set to $\beta_\xi = 1$ and $\beta_\nu = 1.2$. For r_i and ψ_i, the noise covariance matrix is set to diag(0.23 0.24 0.21 0.22 0.23 0.24 0.25) and diag(0.23 0.26 0.24 0.26 0.28 0.29 0.3), respectively.

In this scenario, the goal is to move the agents' formation along a desired level curve maintaining a desired geometric agent formation. At the same time, aim of the controller is to reject disturbance at high frequency noise.

7.5.1 Double Integrator Models

The desired formation r_{Fi} for double integrator models is defined as $r_{F1} = [0\ 0]^T$, $r_{F2} = [-3\ 6]^T$, $r_{F3} = [-6\ 0]^T$, $r_{F4} = [-3\ \ -6]^T$, $r_{F5} = [3\ \ -6]^T$, $r_{F6} = [6\ 0]^T$, and $r_{F7} = [3\ 6]^T$. Their initial positions r_{0i} are set to $r_{01} = [0\ 0]^T$, $r_{02} = [0\ 1]^T$, $r_{03} = [1\ 1]^T$, $r_{04} = [1\ 0]^T$, $r_{05} = [2\ 0]^T$, $r_{06} = [2\ 1]^T$, and $r_{07} = [3\ 0]^T$ with respect to any position inside the scalar field. The distributed controller's tuning parameters are set to $k_F = 1$, $\theta = 1$ and $\gamma = 1.5$.

Fig. 7.2 shows agents' velocity and the formation's response when agents are inside a scalar field. Due to the distributed level curve tracking controller (7.2),

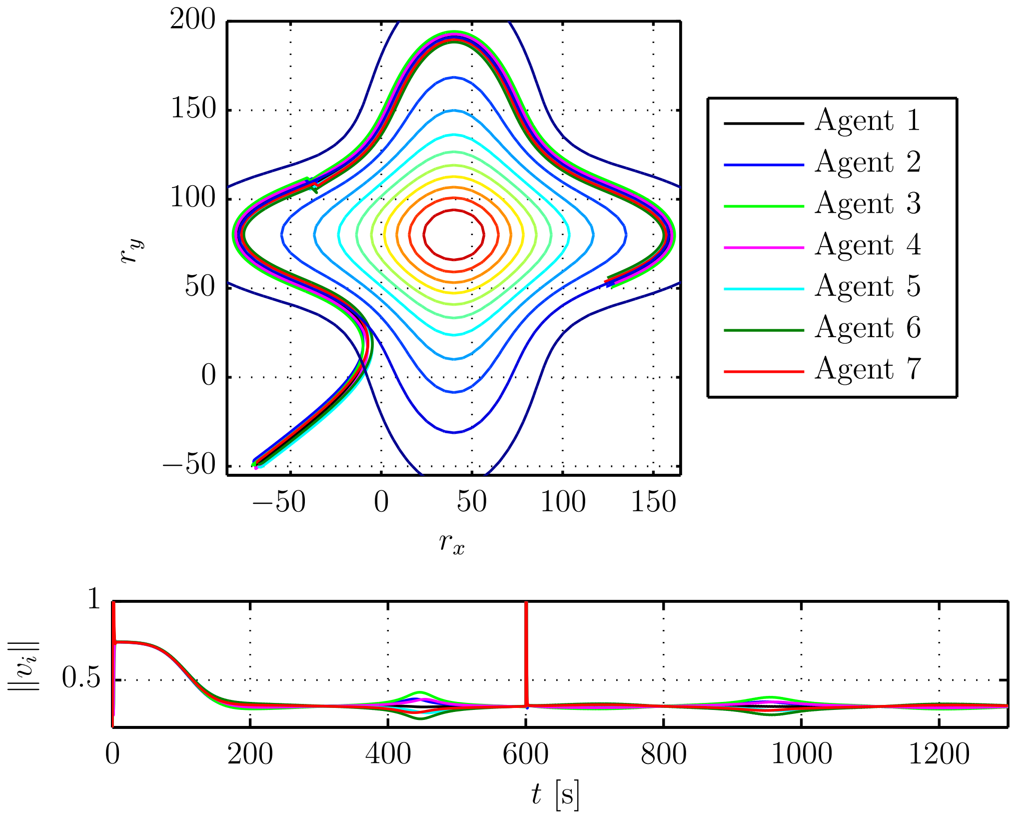


Figure 7.2: Level curve tracking using Equation (7.2)

agents start moving from their initial positions towards the desired level curve $\psi_{ref} = 1$ maintaining the desired formation. Note that agents' velocity is higher when agents are far from the desired level curve because error $\psi_{ref} - \psi_i$ is large. When agents are close to the desired level curve, they move at a constant average velocity defined by $\frac{k_T}{\gamma} = 0.4\ m/s$. At time $t = 600\ s$ disturbance steps in agents 1, 3 and 4 have been included. The controller rejects such disturbance steps and agents go back to the desired level curve and formation. When agents move along a strong arc, inner agents move slower and outer agents move faster than the average velocity of the formation. Their velocity converges to a common value and agents converge to the desired relative positions.

To avoid higher velocities when agents are far from the desired level curve and to drive agents at constant velocity to any place of the scalar field, two distributed controllers can be used. When agents are far, the source seeking algorithm (4.28) can be implemented to enable agents to track the gradient direction. When agents are close to the desired level curve, the level curve tracking algorithm (7.2) can be used to drive agents in a direction perpendicular to the gradient. Fig. 7.3 illustrates the effect of combining these two controllers. When agents start moving, the source seeking algorithm is used and agents travel at an average velocity predefined as $\frac{k_T}{\gamma} = 0.66\ m/s$. At time $t = 150\ s$ the level curve tracking algorithm (7.2) is used and agents travel at an average velocity predefined as $\frac{k_T}{\gamma} = 0.5\ m/s$. Note that the whole formation is able to track different level

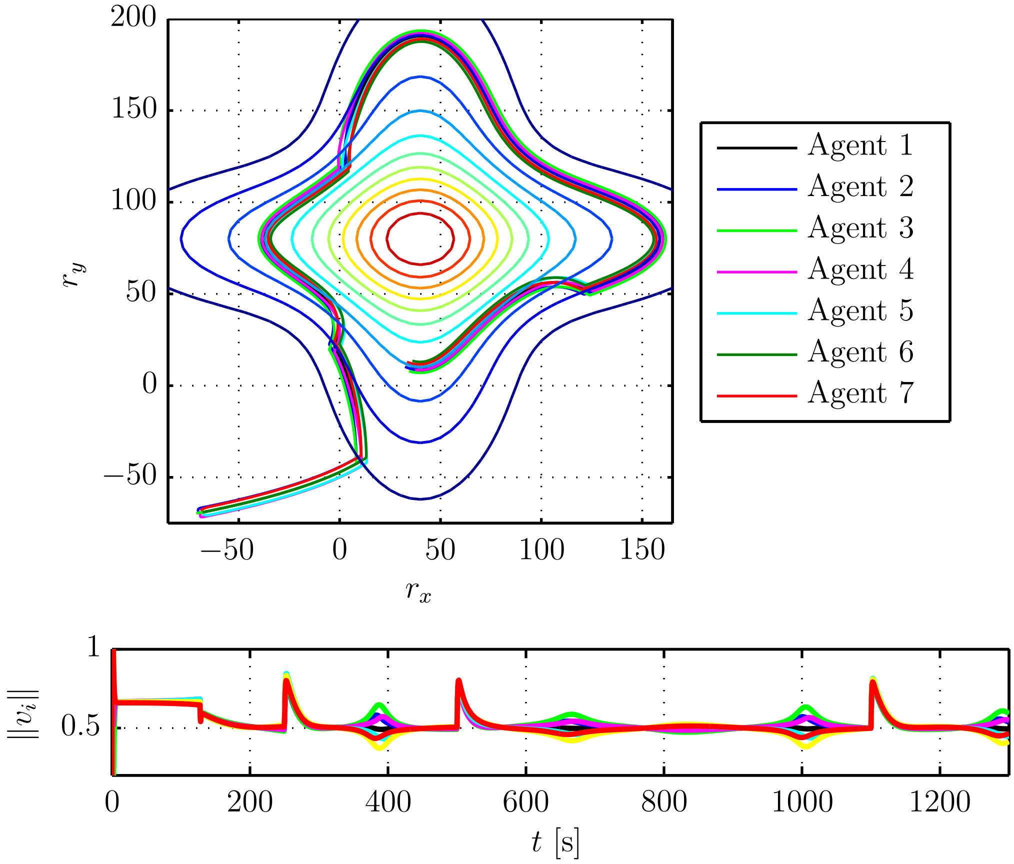


Figure 7.3: Level curve tracking combining Equations (4.28) and (7.2)

curves. In this simulation the level curve reference is changing between $\psi_{ref} = 1$ and $\psi_{ref} = 2$.

7.5.2 Quad-rotor Helicopter

Simulation results for LTI models are presented in this section. Each agent is an underactuated and unstable multi-input/multi-output (MIMO) 12^{th} order dynamic model as proposed in (Lara et al., 2006) and (Pilz et al., 2009), with 4 inputs and 3 position outputs (see the model of quad-rotor helicopter in Appendix A). The desired formation r_{Fi} and the initial positions of quad-rotor helicopters r_{0i} are defined in the same way as for double integrator models with component $r_{zi} = 0$ for all agents.

To stabilize a single agent, a full state feedback LQR controller is chosen. The weighting matrices are

$R = diag(100,\ 0.1,\ 25,\ 25)$ and

$Q = diag(0.04,\ 1,\ 0.04,\ 1,\ 0.5,\ 20,\ 0.25,\ 1,\ 10^3,\ 50,\ 10^3,\ 50)$.

To track the desired level curve and maintain agents in a desired relative position, the level curve tracking algorithm (7.17) is implemented. The sensitivity and control sensitivity weighting filters are $W_s = I_3 \otimes \left(\frac{1}{s+0.0001}\right)$ and $W_K =$

$I_4 \otimes \left(50\frac{s+10^3}{s+10^6}\right)$. The H_∞ synthesis technique is used to design the controller. The designed H_∞ controller is of 19th order. The robust stability H_∞ norm is $0.992 < 1$.

In the case of a quad-rotor helicopter formation, simulation results are similar to them obtained under a double integrator model scenario. This results apply only when the scalar field's source do not move, they apply only for constant scalar fields. For this reason simulations are not repeated.

Now a simulation considering a scalar field's source moving at slow constant velocity and a scalar field expanding and contracting at slow velocity are considered.

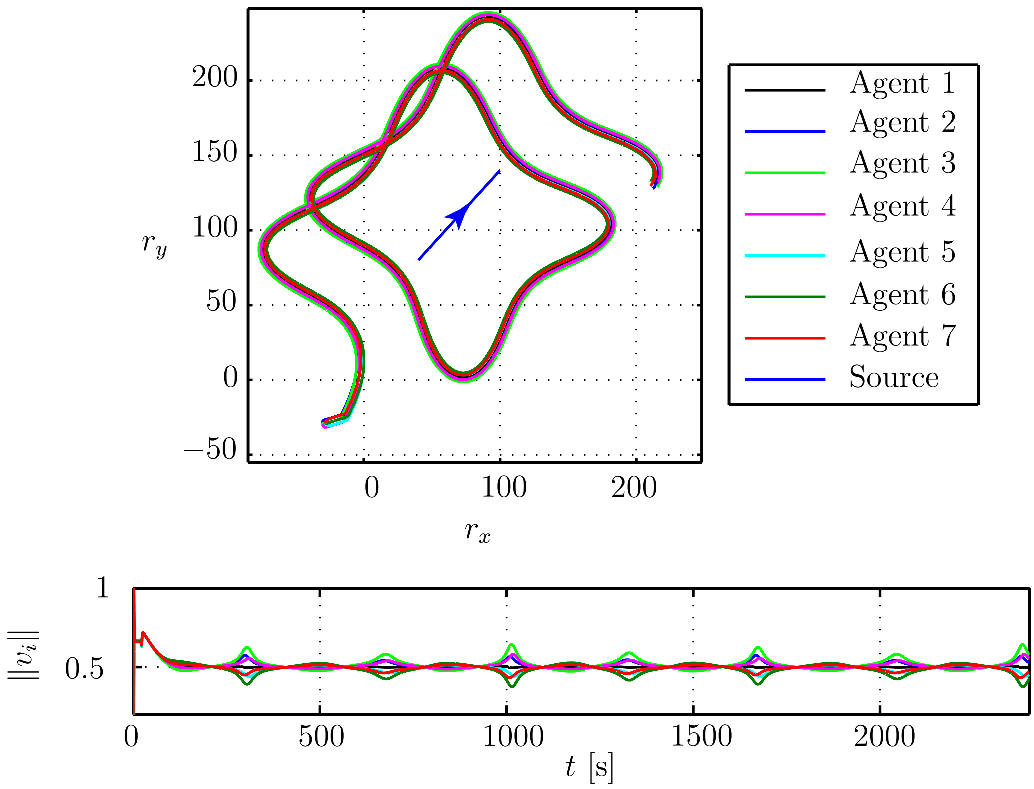


Figure 7.4: Level curve tracking for quad-rotor helicopters operating into a moving scalar field using Equation 7.17

Implementing the approach here presented, agents travel at constant velocity to any place into the scalar field. In real scenarios scalar fields may travel in different directions and also expand or contract; for example under influence of changing environmental conditions. If agents' velocity is less than the scalar field source's velocity and/or extension-contraction velocity of the scalar field, agents localize a desired level curve and move along this curve. Fig. 7.4 shows agents tracking the desired level curve $\psi_{ref} = 1$ at constant velocity while the scalar field's source is moving along the blue line (source). The source moves at velocity $\dot{r}_s = 0.2\ m/s$ while agents move at $\dot{\bar{r}} = 0.5\ m/s$.

7.6 Conclusions

In this chapter, the problem of cooperative level curve tracking has been discussed. Distributed controllers based on both an estimated gradient and a formation controller for double integrator models and LTI systems have been proposed. Agents are able to move along a desired level curve inside the scalar field. Theoretical analysis demonstrates that the algorithms here proposed enable agents to converge to a discretionary level curve into the a scalar field while geometric formation is preserved. When concentration and position signals are corrupted by noise, distributed consensus filters are used to estimate the gradient and agents are still able to move towards a desired level curve.

Conclusions and Outlook

In this thesis distributed algorithms to solve both the source seeking problem and the level curve tracking problem for a group of N identical agents have been presented. These algorithms enable agents to operate under an undirected constrained communication topology. As a contribution to the field of cooperative control, a combined distributed control strategy has been proposed: the distributed algorithms are composed of two parts, a formation controller and a trajectory controller. The distributed formation controller uses information of relative positions and velocities and its goal is to maintain agents on their desired relative positions. The distributed trajectory controller is based on a locally estimated gradient, which is computed for each agent in a distributed way, and its goal is to steer the group of agents towards the scalar field's source or to move agents along the desired level curve. Calculation of this estimated gradient is based also on information of relative agent's positions and information of relative signal strength between agent i and their neighbours.

An important new feature of the proposed approach is the estimation of the gradient in each agent, which opens the door for combining the best attributes of formation controller and trajectory controller, therefore functioning in a distributed way and avoiding constraints of previous approaches. For the proposed source seeking algorithms, it is neither necessary to know the formation's center of mass and its estimated gradient, nor to keep agents rotating in a circular formation, nor to have a leader in the formation. Further, for the proposed level curve tracking algorithms, it is neither necessary to estimate the Hessian matrix, nor to know the formation's center of mass. The solution is found using only an estimated gradient in each one of the agents. The proposed approach is suitable for a large number of agents and for any type of formation. Consequently, applying our approach agents can cover larger spatial areas and use energy more efficiently.

A modified consensus filter operating under constrained communications has been proposed; it is very useful when signals are corrupted by noise. This consensus filter algorithm rejects high frequency noise and improves the accuracy of time-varying signal tracking, too. Therefore, it significantly improves the gradient direction's estimation.

In order to illustrate the proposed source seeking approach through conceivable models of dynamic agents, distributed algorithms for linear and non-linear dynamic agents have been proposed. In the linear case, simple distributed controllers for single and double integrator agents have been designed. To present an improved network of heightened performance and to generalize the results of single and double integrator agents to more complex dynamics, a distributed dynamic controller for LTI systems has been proposed. In the nonlinear case, a simple distributed control law for a group of non-holonomic agents has been designed. An algorithm for collision avoidance between agents and with obstacles has been included in the control law. The distributed source seeking algorithms designed here are apt to locate and track the scalar field's source even if the source is moving at constant velocity. If the average velocity of the formation is higher than the scalar field's source velocity, then our approach allows the group of agents to locate and follow the source or maximum of the scalar field.

To solve the level curve tracking problem, distributed controllers for double integrator agents and LTI systems have been designed. As in the algorithms previously presented, information of the estimated Hessian matrix is not required here. In contrast to other approaches, our approach uses only the estimated gradient computed in each agent and only relative information between agents. The distributed level curve tracking controllers designed here enable agents to successfully move along level curves of a scalar field independently of the state of the scalar field. It can be fixed or can be moving at constant velocity, and at the same time, the scalar field can be expanding and contracting at low velocities.

All mentioned algorithms are provided with stability analysis, showing that the group of agents effectively move towards the scalar field's source or move along a desired level curve while agents maintain the desired formation. The results obtained in the simulations additionally support the effectiveness of this approach.

Future Directions

- In future applications, where keeping a fixed geometric formation poses limits to the applications' efficiency, swarm behaviour for agents can be achieved using this distributed estimated gradient method in a new formulation of the presented control law which changes the trajectory controller. In this case agents could react with more flexibility like a fish school looking for food.

- Natural environment signals are subject to many sorts of disturbances. Therefore, the scalar field could corrupted by spatial noise, which would generate multiple maxima within this field. In order to locate the real maximum, the formation controller should be modified, allowing the formation to expand or shrink the sensing range until the real maximum or source is located.

- In the course of a single mission many different tasks can arise or be indispensable for a successful accomplishment. Examples of thinkable tasks are tracking the gradient direction, taking pictures, changing formation's shape or removing failing agents. In this situation, simple decision-making controllers should be designed to allow optimal task switching. This scenario would require or a renewed stability analysis.

- When there is limited or no position information available (environments where Global Positioning Systems -GPS- does not work, for example in situations of vapour emitting sources such as explosive detection, searching for illegal drugs, chemical leaking or hazardous chemical sensing), distributed controllers must be redesigned allowing agents to find the source or move along a level curve without position information.

Appendix A

Quad-rotor Helicopter Model

The linearised model of the quad-rotor helicopter, written as a state space model, is given by

$$\begin{aligned}\dot{\varrho}(t) =& A\varrho(t) + Bu(t),\\ r(t) =& C\varrho(t),\end{aligned}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \frac{1}{m} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T,$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T,$$

with $m = 0.64$ kg as the total mass of the quad-rotor helicopter and $g = 9.81$ m/s as the gravitational constant.

The state vector is defined as

$$\varrho(t) = \begin{bmatrix} r_x & \dot{r}_x & r_y & \dot{r}_y & r_z & \dot{r}_z & o_r & \dot{o}_r & o_p & \dot{o}_p & o_{yz} & \dot{o}_{yz} \end{bmatrix}^T,$$

where r_x and r_y describe the position of the quad-rotor helicopter in the horizontal plane, r_z describe the vertical coordinate, o_r is the roll angle around the r_x-axis, o_p is the pitch angle around the r_y-axis and o_{yz} is the yaw angle around the r_z-axis.

The control inputs are defined as

$$u(t) = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}^T,$$

and the output vector is defined as

$$r(t) = \begin{bmatrix} r_x & r_y & r_z \end{bmatrix}^T.$$

Bibliography

Ahmadi, S., E. Rosero, and H. Werner (2015). Formation control of nonholonomic agents with collision avoidance. In *2015 American Control Conference (ACC), 2015*, pp. 757–762.

Azuma, S., M. S. Sakar, and G. Pappas (2012). Stochastic source seeking by mobile robots. *Automatic Control, IEEE Transactions on 57*(9), 2308–2321.

Bai, H., R. Freeman, and K. Lynch (2010). Robust dynamic average consensus of time-varying inputs. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pp. 3104–3109.

Biyik, E. and M. Arcak (2007). Gradient climbing in formation via extremum seeking and passivity-based coordination rules. In *Decision and Control, 2007 46th IEEE Conference on*, pp. 3133 –3138.

Borrelli, F. and T. Keviczky (2008). Distributed LQR design for identical dynamically decoupled systems. *Automatic Control, IEEE Transactions on 53*(8), 1901–1912.

Brinon-Arranz, L., A. Seuret, and C. Canudas-de Wit (2011). Collaborative estimation of gradient direction by a formation of auvs under communication constraints. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pp. 5583 –5588.

Cochran, J. and M. Krstic (2009). Nonholonomic source seeking with tuning of angular velocity. *Automatic Control, IEEE Transactions on 54*(4), 717 –731.

Cochran, J., A. Siranosian, N. Ghods, and M. Krstic (2009). 3-D source seeking for underactuated vehicles without position measurement. *Robotics, IEEE Transactions on 25*(1), 117 –129.

Cortes, J. (2005). Achieving coordination tasks in finite time via nonsmooth gradient flows. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pp. 6376 – 6381.

Eichler, A. and H. Werner (2013). Performance bounds on decomposable systems. In *2013 American Control Conference (ACC), 2013.*

Fabbiano, R., C. Canudas De Wit, and F. Garin (2014). Distributed source localisation with no position information. In *13th European Control Conference (ECC 2014)*, Strasbourg, France.

Fabbiano, R., C. C. de Wit, and F. Garin (2014). Source localization by gradient estimation based on poisson integral. *Automatica 50*(6), 1715 – 1724.

Fax, J. and R. Murray (2004). Information flow and cooperative control of vehicle formations. *Automatic Control, IEEE Transactions on 49*(9), 1465–1476.

Freeman, R., P. Yang, and K. Lynch (2006). Stability and convergence properties of dynamic average consensus estimators. In *Decision and Control, 2006 45th IEEE Conference on*, pp. 338–343.

Ghods, N. and M. Krstic (2010). Multi-agent deployment around a source in one dimension by extremum seeking. In *American Control Conference (ACC), 2010*, pp. 4794 –4799.

Lara, D., A. Sanchez, R. Lozano, and P. Castillo (2006). Real-time embedded control system for vtol aircrafts: Application to stabilize a quad-rotor helicopter. In *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*, pp. 2553–2558.

Li, S. and Y. Guo (2012). Distributed source seeking by cooperative robots: All-to-all and limited communications. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 1107–1112.

Li, S. and Y. Guo (2013). Distributed consensus filter on directed graphs with switching topologies. In *American Control Conference (ACC), 2013*, pp. 6151–6156.

Li, S., R. Kong, and Y. Guo (2014). Cooperative distributed source seeking by multiple robots: Algorithms and experiments. *IEEE/ASME Transactions on Mechatronics 19*(6), 1810–1820.

Li, Z., Z. Duan, and G. Chen (2011). On and performance regions of multi-agent systems. *Automatica 47*(4), 797 – 803.

Liu, S.-J. and M. Krstic (2010). Stochastic source seeking for nonholonomic unicycle. *Automatica 46*(9), 1443 – 1453.

Massioni, P. and M. Verhaegen (2009). Distributed control for identical dynamically coupled systems: A decomposition approach. *Automatic Control, IEEE Transactions on 54*(1), 124–135.

Mastellone, S., D. M. Stipanovic, C. R. Graunke, K. A. Intlekofer, and M. W. Spong (2008). Formation control and collision avoidance for multi-agent nonholonomic systems: Theory and experiments. *I. J. Robotic Res. 27*(1), 107–126.

Matveev, A. S., H. Teimoori, and A. V. Savkin (2011). Navigation of a unicycle-like mobile robot for environmental extremum seeking. *Automatica 47*(1), 85 – 91.

Mayhew, C., R. Sanfelice, and A. Teel (2008). Robust source-seeking hybrid controllers for nonholonomic vehicles. In *American Control Conference, 2008*, pp. 2722 –2727.

Mendez, A., C. Hoffmann, and H. Werner (2015). LPV formation control for a class of non-holonomic agents with directed and switching communication topologies. In *Proc. 54th IEEE Conf. Decision Control*, pp. 757–762.

Mendez, A. and H. Werner (2014). LPV formation control of non-holonomic multi-agent systems. In *Proc. 19th IFAC World Congr.*, pp. 1997–2002.

Mesbahi, M. and M. Egerstedt (2010). *Graph theoretic methods in multiagent networks*. Princeton series in applied mathematics. Princeton: Princeton University Press.

Moore, B. and C. Canudas-de Wit (2010). Source seeking via collaborative measurements by a circular formation of agents. In *American Control Conference (ACC), 2010*, pp. 6417 –6422.

Ogren, P., E. Fiorelli, and N. Leonard (2004). Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *Automatic Control, IEEE Transactions on 49*(8), 1292 – 1302.

Oh, K.-K. and H.-S. Ahn (2010). A survey of formation of mobile agents. In *Intelligent Control (ISIC), 2010 IEEE International Symposium on*, pp. 1470–1475.

Okubo, A. (1986). Dynamical aspects of animal grouping: swarms, schools, flocks, and herds. *Advances in Biophysics 22*, 1–94.

Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: algorithms and theory. *Automatic Control, IEEE Transactions on 51*(3), 401–420.

Olfati-Saber, R., J. Fax, and R. Murray (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE 95*(1), 215–233.

Olfati-Saber, R. and R. Murray (2004). Consensus problems in networks of agents with switching topology and time-delays. *Automatic Control, IEEE Transactions on 49*(9), 1520 – 1533.

Olfati-Saber, R. and J. S. Shamma (2005). Consensus filters for sensor networks and distributed sensor fusion. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC 05. 44th IEEE Conference on*, pp. 6698 – 6703.

Pilz, U., A. Popov, and H. Werner (2009). Robust controller design for formation flight of quad-rotor helicopters. In *Proceedings of the Joint 48th IEEE Conference on Decision and Control and Chinese Control Conference*, pp. 8322–8327.

Popov, A. and H. Werner (2009). A robust control approach to formation control. In *Proceedings of the European Control Conference*, pp. 4428–4433.

Popov, A. and H. Werner (2012). Robust stability of a multi-agent system under arbitrary and time-varying communication topologies and communication delays. *Automatic Control, IEEE Transactions on 57*(9), 2343–2347.

Ren, W. and E. Atkins (2007). Distributed multi-vehicle coordinated control via local information exchange. *International Journal of Robust and Nonlinear Control 17*(10-11), 1002–1033.

Ren, W., R. Beard, and E. Atkins (2007). Information consensus in multivehicle cooperative control. *Control Systems, IEEE 27*(2), 71–82.

Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavioral model. In *Computer Graphics*, pp. 25–34.

Rosero, E. and H. Werner (2014a). Cooperative source seeking via gradient estimation and formation control (part 1). In *United Kingdom Automatic Control Council (UKACC), 2014*.

Rosero, E. and H. Werner (2014b). Cooperative source seeking via gradient estimation and formation control (part 2). In *United Kingdom Automatic Control Council (UKACC), 2014*.

Rosero, E. and H. Werner (2014c). Modified distributed consensus filter for sensor networks. In *Control Conference (ECC), 2014 European*, pp. 892–895.

Seyboth, G., D. V. Dimarogonas, and K. H. Johansson (2011). Control of multi-agent systems via event-based communication. In *Proc. 18th IFAC World Congress*, Milan, Italy, pp. 10086–10091.

Shamma, J. (2008). *Cooperative Control of Distributed Multi-Agent Systems*. Massachusetts: Wiley-Interscience.

Spanos, D., R. Olfati-Saber, and R. Murray (2005). Dynamic consensus on mobile networks. In *The 16th IFAC World Congress, Prague, Czech*.

Stankovic, M. S. and D. an M. Stipanovic (2010). Extremum seeking under stochastic noise and applications to mobile sensors. *Automatica 46*(8), 1243 – 1251.

Williams, R. K. and G. S. Sukhatme (2012). Probabilistic spatial mapping and curve tracking in distributed multi-agent systems. In *ICRA'12*, pp. 1125–1130.

Xiao, L., S. Boyd, and S.-J. Kim (2007). Distributed average consensus with least-mean-square deviation. *J. Parallel Distrib. Comput. 67*(1), 33–46.

Xiao, L., S. Boyd, and S. Lall (2005). A scheme for robust distributed sensor fusion based on average consensus. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, IPSN '05, Piscataway, NJ, USA. IEEE Press.

Zhang, C., D. Arnold, N. Ghods, A. Siranosian, and M. Krstic (2007). Source seeking with non-holonomic unicycle without position measurement and with tuning of forward velocity. *Systems and Control Letters 56*(3), 245 – 252.

Zhang, F., E. Fiorelli, and N. Leonard (2007). Exploring scalar fields using multiple sensor platforms: Tracking level curves. In *Decision and Control, 2007 46th IEEE Conference on*, pp. 3579–3584.

Zhang, F. and N. Leonard (2010). Cooperative filters and control for cooperative exploration. *Automatic Control, IEEE Transactions on 55*(3), 650 –663.

Zhu, J. (2011). On consensus speed of multi-agent systems with double-integrator dynamics. *Linear Algebra and its Applications 434*(1), 294 – 306.

Zhu, S., D. Wang, and C. Low (2013). Cooperative control of multiple uavs for source seeking. *Journal of Intelligent & Robotic Systems 70*(1-4), 293–301.

Zhu, S., D. Wang, and C. Low (2014). Cooperative control of multiple uavs for moving source seeking. *Journal of Intelligent & Robotic Systems 74*(1-2), 333–346.

Publications

Rosero Esteban, and Werner Herbert. Modified distributed consensus filter for sensor networks. *2014 European Control Conference (ECC)*. June, 2014.

Rosero Esteban, and Werner Herbert. Cooperative Source Seeking via Gradient Estimation and Formation Control (Part 1). *2014 United Kingdom Automatic Control Council (UKACC)*. July, 2014.

Rosero Esteban, and Werner Herbert. Cooperative Source Seeking via Gradient Estimation and Formation Control (Part 2). *2014 United Kingdom Automatic Control Council (UKACC)*. July, 2014.

Siavash Ahmadi, Rosero Esteban, and Werner Herbert. Formation control of non-holonomic agents with collision avoidance. *2015 American Control Conference (ACC)*. July, 2015: 757-762.

Curriculum Vitae

Last name	Rosero García
First name	Esteban Emilio
Date of birth	21.06.1976
Place of birth	Potosí, Colombia

08.1983 - 07.1988	Elementary School, Potosí, Colombia
08.1988 - 07.1994	Secondary School, Potosí, Colombia
08.1994 - 07.1995	Military Service, Tunja, Colombia
08.1995 - 12.2001	Bachelor in Mechanical Engineering Universidad del Valle, Cali, Colombia
01.2002 - 07.2002	English course
08.2002 - 02.2003	Process Engineer CENICAÑA, Cali, Colombia
03.2003 - 07.2006	Master of Science in Engineering, emphasis: Automática Universidad del Valle, Cali, Colombia
08.2006 - 07.2007	Lecturer Universidad del Valle, Cali, Colombia
08.2007 - 09.2010	Auxiliary Professor Universidad del Valle, Cali, Colombia
10.2010 - 09.2014	Ph.D. student at the Institute of Control Systems Hamburg University of Technology (TUHH), Germany
10.2014 - today	Assistant Professor at the Electrical & Electronic School Universidad del Valle, Cali, Colombia