MULTI-AGENT CONTROL UNDER COMMUNICATION CONSTRAINTS

Commande d’une flottille de robots sous-marines avec des contraintes de communication

Lara Briñón Arranz
CNRS, Control Systems Department, GIPSA-lab,
NeCS INRIA-CNRS team-project,
Responsable: Carlos Canudas de Wit
lara.brinon-arranz@inrialpes.fr

February-Jun 2008

ENSIEG - INPG

INRIA Rhône-Alpes
1 Abstract

2 Introduction
  2.1 Project Context .......................................................... 5
  2.2 Review of State of the Art .............................................. 6
    2.2.1 Models ................................................................. 6
    2.2.2 Formation Control .................................................. 7
    2.2.3 Formation control with collision avoidance ...................... 10
  2.3 Source Seeking control .................................................. 10

3 General Control Scheme

4 Formation Control
  4.1 Circle Formation .......................................................... 13
    4.1.1 Circle with centre in (0,0) ........................................ 13
    4.1.2 Circle with random coordinates of the centre .................... 16
    4.1.3 Circle with centre in \((c_1, c_2)\) .................................. 16
  4.2 Uniform Distribution ..................................................... 17

5 Variable Formation Control
  5.1 Main Transformations .................................................... 19
  5.2 Limitation of the previous schemes ................................... 20
    5.2.1 Transitory response ................................................. 20
    5.2.2 Slow variations ..................................................... 21
  5.3 Control Redesign based on previous schemes ......................... 23
    5.3.1 Contraction Algorithm .............................................. 24

6 Gradient Search Method

7 Conclusions
  7.1 Review of results ........................................................ 29
  7.2 Perspectives ............................................................... 30

8 Review in French
  8.1 Introduction .............................................................. 31
    8.1.1 Contexte de stage .................................................. 31
    8.1.2 Etude de l'état de l'art ........................................... 31
  8.2 Méthodologie ............................................................... 33
  8.3 Contrôle de la formation ............................................... 33
    8.3.1 Formation du cercle ................................................. 33
8.3.2 Distribution uniforme .................................................. 34
8.4 Contrôle des variations de la formation .................................. 34
8.5 Recherche du gradient ....................................................... 35
8.6 Conclusions ................................................................. 36

9 Attached Document A ......................................................... 37
  9.1 Operators ................................................................. 37
  9.2 Calculations developed .................................................. 37
    9.2.1 Formation Control .................................................. 37
    9.2.2 Uniform Distribution Control ...................................... 39
  9.3 Contraction circle control .............................................. 40
  9.4 Gradient Search Control law .......................................... 42
  9.5 Program code ........................................................... 44
  9.6 Simulink Diagrams ...................................................... 55

10 Attached Document B: Review of the state of art ..................... 56
  10.1 Introduction ........................................................... 57
  10.2 Models ................................................................. 57
  10.3 Formation Control .................................................... 59
    10.3.1 Lyapunov-based control design ................................... 59
    10.3.2 Laplacian-based consensus algorithms ......................... 62
    10.3.3 Formation control with collision avoidance ..................... 66
  10.4 Source Seeking control .............................................. 68
    10.4.1 Single agent with a single sensor ................................ 68
    10.4.2 Source seek using multiple sensors ............................. 70
1 Abstract

In this report we present the project *Multi-agent control under communication constraints*. The main objective is that a multi-agent system, composed of underwater vehicles which have to work in a cooperative way, find the position of a source (fresh water, chemical source, methane’s source). We show the simulation results and the mathematical development of the Lyapunov theorem for each problem.

Ce rapport présente le stage *Commande d’une flottille de robots sous-marines avec des contraintes de communication*. L’objectif traité est qu’une flottille d’agents sous-marins travaillent en collaboration, pour réaliser la recherche (par gradient) et la localisation d’une source (eaux douce, source chimique, source de méthane, etc.) On montre les résultats obtenus dans les simulations et les calculs théoriques réalisés en appliquant le théorème de Lyapunov.
2 Introduction

Control community had formerly focused mainly on control of vehicle formations. However, distributed motion and cooperation systems have emerged as topics of significant interest and matters of concern to the control theory and robotics specialists over the past few years.

This project *Commande d’une flottille de robots sous-marines avec des constraintes de communication* tackles the problem of multi-agents control with communications constraints. The multi-agents system is composed of AUVs (Autonomous Underwater Vehicles). The aim of AUVs is to find the position of a source (fresh water, chemical source, methane’s source). In order to achieve this objective, the agents have to work in a cooperative way carrying out a gradient search through the concentration measurements of each agent.

2.1 Project Context

This work is included in the project CONNECT. It has been developed in five months but due to the complexity of this subject the thesis *Control Design for Multi-Agent systems under communication constraints* will continue with this research project.

**Project CONNECT** This project is funded by the ANR (National Research Agency). The project deals with the problem of controlling multi-agent systems, i.e. systems composed of several sub-systems interconnected between them by an heterogeneous communication network. The main challenge is to learn how to design controllers accounting for constraints on the network topology, but also on the possibility to share computational resources during the system operation, while preserving closed-loop system stability. The control of an agents cluster composed of autonomous underwater vehicles, marine surface vessels, and possibly aerial drones will be used as a support example all along the proposal.

This project belongs to the NeCS (Networked Controlled System Team) which is a INRIA-GIPSA-lab joint team-projet supported by the CNRS, INRIA, INPG and UJF. The team goal is to develop a new control framework for assessing problems raised by the consideration of new technological low-cost and wireless components, the increase of systems complexity and the distributed and dynamic location of sensors (sensor networks) and actuators.

**GIPSA laboratory** The GIPSA laboratory devotes to the fundamental research about the speech, the perception, the knowledge, the brain, the diagnosis and the control of systems. This laboratory develops the applications in several areas, as multimode interaction, telecommunications, the energy, the environment, on-board systems, robotic mechatronics, the healthy, the transports, etc.

GIPSA-lab comprises several laboratories and a research team: Laboratoire d’automatique de Grenoble (LAG, UMR CNRS-INPG-Université Joseph Fourier) Institut de la communication...
2.2 Review of State of the Art

This section reviews various methods and control approaches dealing with multi-agents systems and formation control. It is completed also with a short review on some of the most commonly used models for control design and analysis in this context. At the end we also review gradient search methods that will be of use when using the fleet formation for source searching.

2.2.1 Models

Simple linear state space models are in some cases used to represent the agent’s dynamics [1]. Nevertheless, nonlinear massless models based on the velocity kinematics are better adapted to describe the motions of AUV’s. They can be formulated in three-dimensional space, but for seek of simplicity some works are carried out in the plane (two-dimensional models) [1, 2].

\textbf{Kinematic models} Many of the studies carried out in the field of control of multi-agent control are based on simplified models. Some studies deal with simple point masses in the plane, \( \dot{x}_i = u_i \) where each \( x_i \) describes the agent coordinate vector, and \( u_i \), the associated control input. In many cases, the studies limited to motion in the plane with \((x_i, u_i) \in \mathbb{R}^2\). A more general models formulate linear models, of the standard form \( \dot{x}_i = A_i x_i + B_i u_i \).

In the applications concerning mobile robots and also underwater robots, it seems to be customary to model each vehicle from its kinematic equations, that in certain cases, see [3], can be simplified to point mass motion subject to planar steering control, i.e.

\begin{align*}
\dot{r}_k &= v e^{i \theta_k} \\
\dot{\theta}_k &= u_k
\end{align*}

where \( r = x_k + iy_k, \in \equiv \mathbb{R}^2 \) and \( \theta_k \in S^1 \) are the position and heading of each vehicle, \( v \) is the vehicle velocity that often is normalized to one, \( v = 1 \). The particular reason to make this assumption, in the context of underwater vehicles is due to the fact that underwater vehicles with a single propel, like the one considered in the CONNECT project, are energy-efficiently operated under constant velocity motion.

Equation (33) is not fundamental, it only asses the fact that the control input is related to the velocity steering rather than to the steering angle directly. It is however possible to simply further the model by assuming that heading time-scale, is at least of an order of magnitude faster than the position time scale. Therefore, it makes sense to consider the steering angle \( \theta_k \) directly the control input, with \( v = 1 \)

\[ \dot{r}_k = e^{i u_k} \]
where $u_M = \theta_{\text{max}}$ describes the maximum possible turning angle for the considered vehicle.

**Dynamics models** An example of dynamic model in the plane is [1],

\[
\begin{align*}
\dot{x}_{1k} &= x_{2k} \\
\dot{x}_{2k} &= f_k \cos(u_k) - 2x_{2k} \\
\dot{y}_{1k} &= y_{2k} \\
\dot{y}_{2k} &= f_k \sin(u_k) - 2y_{2k}
\end{align*}
\]

where $f_k$ represents the input force given by the vehicle’s thruster, $u_k$ is the angle of the vehicle or heading, and the states $x_k$ and $y_k$ represent the position of the vehicle in the plane. The model can be completed with the linear damping $-2x_k$ and $-2y_k$ associated to each direction of motion.

**Group models** As the agents are controlled in a coordinated fashion, rather than independent each other, it become interesting to introduce other variables of interest that better describe the group motion. For instance, in [3] they introduce the center of mass, $R$,

\[
R = \frac{1}{N} \sum_{j=1}^{N} r_k
\]

and the velocity of the center of mass of the group:

\[
p = \frac{1}{N} \sum_{k=1}^{N} e^{iu_k} = \frac{1}{N} \sum_{k=1}^{N} \dot{r}_k = \dot{R}
\]

However, these appellations may be not exact as motion of the body in water are not subject to pure gravity forces, but also influenced by buoyancy, and by fluid nonlinear damping. It may be of interest to consider also other possible points of interest better adapted to motion in fluid environments.

Other quantities of interest are the potential function $U$, defined as

\[
U(u) = \frac{N}{2} |p|^2
\]

which reflect a kind of "kinetic energy". The gradient of this function $\frac{\partial U}{\partial u_k} = \langle ie^{iu_k}, p \rangle = \Re \{ -ie^{-iu_k} p \}$ is often used for control design. (See attached document for the inner product definition)

### 2.2.2 Formation Control

In this section we review some control strategies assuming all-to-all communications assumptions.

**Lyapunov-based control design** The paper [3] resumes several control options based on potential and Lyapunov functions. We summarize some of these results here below.
Circular motions & relative headings  To stabilize circular motions of the group about its center of mass, and also to stabilize a particular arrangement (phase-locked patterns) of the vehicles in their circular formation, the following control law has been proposed in [3],

$$\dot{u}_k = \omega_0(1 + \kappa \langle \tilde{r}_k, \dot{r}_k \rangle) - \frac{\partial U}{\partial u_k}$$  \hspace{1cm} (5)

where, $\tilde{r}_k = r_k - \mathbf{R}$ describes the distance between each vehicle $k$ to the center of mass, i.e.

$$\tilde{r}_k = r_k - \mathbf{R} = \frac{1}{N} \sum_{j=1}^{N} r_{kj}$$

$r_{kj}$ being the relative position $r_{kj} = r_k - r_j$.

The control law allows the stabilization to a circle with a radius $\rho_0 = |\omega_0|^{-1}$ with the rotation direction of $\omega_0$. By the second term, this control law allows the stabilization of symmetric $(M - N)$-patterns characterized by $2 \leq M \leq N$ heading clusters separated by a multiple of $2\pi$. To this to happen, an addition requirement is that the potential $U(u)$ be invariant to rigid rotations, i.e. $\langle \nabla U, \vec{1} \rangle = \langle \mathbf{p}, \mathbf{p} \rangle = 0$.

Stability and convergence to the circular motions, are show by means of the Lyapunov function $V$, $V(r, u) = \kappa \frac{1}{2} \sum_{k=1}^{N} |e^{i u_k} - i \omega_0 \tilde{r}_k|^2 + U(u)$

composed of a function which has a minimum zero for circular motions around the center of mass with radius $\rho_0 = 1/\omega_0$, and the potential function $U$.

The control law (40) does not necessarily allows of a evenly spaced location of the vehicles. For instance, it is possible to reach equilibrium so that the vehicles superimposes. To avoid this, authors [3], and [4], proposed a new form of potential including "high-order harmonics", i.e.

$$U_{m}(u) = \frac{N}{2} |\mathbf{p}_m|^2, \quad \mathbf{p}_m = \frac{1}{mN} \sum_{k=1}^{N} e^{imu_k}$$

Coordinated subgroups  The extension to the coordination of sub-groups allows to design control laws to coordinate vehicles in sub-groups using block all-to-all interconnections. The idea can be of interests, if we decide that we should split the vehicle fleet motion into a subsets of smaller subgroups. Or if it is interesting to split the gradient source search in 3D, into cuts of subgroups at different deeps.

Shape control: Elliptical Beacon control laws  Other possible extension is to modify the circular motions and to stabilize a single vehicle on an elliptical trajectory about a fixed beacon. In addition, it was also shown by the same authors that it is possible to couple several vehicles via their relative heading in order to synchronize the vehicle phase of each ellipse.

Lara Briñon Arranz 8 Jun 2008
Laplacian-based consensus algorithms

**Gossip protocols** A gossip protocol is a protocol designed to mimic the way that information spreads when people gossip about some fact, and it satisfies some particular conditions. The core of the protocol involves periodic, pairwise, inter-process interaction and the information exchanged during these interactions is of bounded size. When agents interact in a gossip protocol, the state of one or both changes in a way that reflects the state of the other and reliable communication is not assumed.

Inspired by these ideas Laplacian-based consensus control algorithms has been derived to deal with the control of multi-agent systems with limited information and they are based in the review recently provided in the paper [5], which is based on the five-key papers [6, 7, 8, 9, 10].

**Consensus in Networks** The interaction topology of a network of agents is represented using a directed graph \( G = (V, E) \) with the set of nodes \( V = \{1, 2, \ldots, n\} \) and edges \( E \subseteq V \times V \). The neighbors of agent \( i \) are denoted by \( N_i = \{j \in V : (i, j) \in E\} \).

This set describes the information that is accessible to the agent \( i \), either all the time (fixe network topology), or at particular time instant (variable network topology). Assuming now that the control law for each agent is designed on the basis of limited information, \( u_i = \sum_{j \in N_i} (x_j(t) - x_i(t)) \), a simple consensus algorithm to reach an agreement regarding the state of \( n \) integrator agents with dynamics \( \dot{x}_i = u_i \) can be then expressed as an nth-order linear system on a graph. Hence, the collective dynamics of the group of agents can be compactly written as \( \dot{x} = Lx \) where \( L \) is graph Laplacian of the network and its elements are defined as follows:

\[
l_{ij} = \begin{cases} 
-1 & j \in N_i \\
\frac{1}{|N_i|} & j = i 
\end{cases}
\]

where \(|N_i|\) denotes the number of neighbors of node \( i \) (or out-degree of node \( i \)). With this control law, the authors in [5] had shown that the states converge to the same equilibrium, \( x \rightarrow x^* = \frac{1}{n} \sum x_i(0)1_n \) which shows that all states (nodes) agrees (consent). It is also shown that \( x^* \) is an unique equilibrium as long as the graph is connected.

**Distributed Formation Control** The above controller is not able to impose a particular formation. This can be reached by using vectors of relative positions of neighboring vehicles, and then using consensus based-controllers with input bias [7]. For this, the authors considered the problem of minimizing locally the cost function

\[ U(x) = \sum_{j \in N_i} ||x_j - x_i - r_{ij}||^2 \]

via a distributed algorithm, where \( r_{ij} \) is the desired inter-vehicle relative position vector.

If the agents use the gradient decent algorithm to minimize the cost \( U(x) \), this leads to

\[ \dot{x}_i = \sum_{j \in N_i} (x_j(t) - x_i(t) - r_{ij}) = \sum_{j \in N_i} (x_j(t) - x_i(t)) + b_i \]
with input bias $b_i = -\sum_{j \in N_i} r_{ij}$. This is equivalent to the consensus problem mentioned before, with the bias term added. Although this bias does not play any role in the stability of the system, it helps to modify the equilibrium of the state. In that way a particular formation can be obtained by modifying each of the components of $r_{ij}$.

2.2.3 Formation control with collision avoidance

Flocking is a form of collective behavior of large number of interacting agents with a common group objective, inspired in the clouding animal behavior. The three rules of Reynolds generate the main principles for the flocking method control. This rules, known as cohesion, separation and alignment rules, are gathered in [11]. Following theses rules, the gradient-based algorithm equipped with velocity consensus protocol and flocking with obstacle avoidance [12] achieve methods for formation control.

A graph model is used to tackle the flocking problem. Thus each node represents one agent, called $\alpha$-agent, and the set of neighbors for each $\alpha$-agent is defined as an open ball with radius $r$ [12]:

$$N_i = \{ j \in V : \| q_j - q_i \| < r \}$$

where $\| \cdot \|$ is the Euclidean norm in $\mathbb{R}^m$, and $q_i$ describes the Euclidean generalized coordinated of agent $i$.

Distributed algorithms for flocking in free-space, or free-flocking, use $\alpha$-agents to represent each agent in the network system. We introduce virtual agents called $\beta$-agents and $\gamma$-agents which model the effect of obstacles and collective objective of a group, respectively [12]. We use agent-based representation of all nearby obstacles; therefore, we defined the set of $\beta$-neighbors of an $\alpha$-agent as follows:

$$N^\beta_i = \{ k \in V^\beta : \| q_{i,k} - q_i \| < r' \}$$

where $r' > 0$ is interaction range of an $\alpha$-agent with neighboring $\beta$-agents.

To achieve flocking in presence of obstacles, in [12] the authors use a multi-species collective potential function for the particle system, composed for three terms $V_\alpha(q), V_\beta(q)$ and $V_\gamma(q)$, each of them describes a problem (neighbors, obstacles, collective objective). They attempt to relate the control functions with attractive/repulsive pairwise potential [13].

2.3 Source Seeking control

There are several works dealing with the problem of source seeking in different scenarios. They rate from the source seeking for signal fields which are static (light sources, magnetic fields) with respect to the target emitting them [14], to the cases where signals are diffusive [15]. In this latter case the source signal is governed by a diffusion equation. Application concerning plume tracking and source seeking using multiple sensors on a single agent, or multiple agents communicating between each other are reporting in works [16, 17, 18, 19].
Works in [14], and [15] concern a single agent modeled as a pointless mass similar to the model (34) – (35). In particular [15] treat the case of diffusive sources and show how this source tracking problem can be tackled with a simple control law which has two terms. The first term in the control law is used to produce an excitation such as to always collect sufficient rich concentration information, while the second term is used to direct the vehicle heading towards the minimum (or maximum) source concentration. Stability and convergence toward the source focus are shown in [14].

**Contour-shape formations** In [19] general curve evolutionary theory has been used to the decentralized control of contour-shape formations of underwater vehicles commissioned to adapt to a certain level set of the source concentration. Several interesting ideas are here proposed. In particular the concept of flexible (or deformable) shape formations, where the shape of the formation is dictated by the environment. The forces used for that purpose need to be in adequation to those needed to preserve the formation.

**Adaptive gradient climbing** The authors in [17] present a stable control strategy for groups of vehicles to move and reconfigure cooperatively in response to a sensed, distributed environment. The underlying coordination framework uses virtual bodies and artificial potentials. This work address the problem of a gradient climbing missions in which the mobile sensor network seeks out local maxima or minima in the environmental field.

### 3 General Control Scheme

After developing the review of the state of the art on the control design of multi-agents systems that has been condensed previously, we tackle the project purpose of generating a general scheme. This scheme will steer the search of the problem solution.

Henceforth, in order to simplify the approach, the followings assumptions are considered:

**A1** First, assuming that all the states are measured, and that a ”centralized” controller is to be considered

**A2** It is also assumed that only kinematic models are to be considered

**A3** Communication bandwidth is unlimited

Control problems ordered as a function of its complexity.

1. **Track simple forms.** Learn how to design feedbacks \((u_k, v_k)\) such as to produce a circular formation, with equally spaced angles along the circle. Other items are:
• Which is the best kinematic representation between the Euler angle representation given before, and eventually the modified Rodriguez Parameters (MRP) $\sigma_k = G_k(\sigma_k)\omega_k$ with $\sigma_k \in \mathbb{R}^3$, being the orientation of the rigid body

\[
G_k(\sigma_k) = \frac{1}{2} \left(1 - \frac{\sigma_k^T \sigma_k}{2} I_3 - S(\sigma_k) + \sigma_k^T \sigma_k \right)
\]

• Make some simulations using different initial conditions

2. **Track complex forms.** Learn how to design controllers for any particular form, i.e. ellipses with arbitrarily center, shape and orientation. Try to use also a generalized operations for translation, rotation, and eventually contraction/expansion (scaling).

3. **Avoid collisions.** Introduce a mechanisms that allows to avoid collisions between agents.

4. **Track gradients of concentration.** Learn how to modify the previous law in order to track gradient variations of the concentration contours, using the previously 3 mentioned generalized operations.

The following diagram of the general system points out the mainly objective, the gradient search. Even then we must build the formation control in order to achieve the control motion of the multi-agent system depending on the concentration measurements.

![General system diagram](image)

*Figure 1: General system diagram*

It is going to be attempted generalize the proposed problem. It would be useful to be able to construct control law with additive properties as the ones in the potential function approaches, where the various control capabilities are obtained by adding, or changing the potential functions.

However, we have taken into account the time limitations for the development of all previous items and that this work *Commande d’une flottille de robots sous-marines avec des constraints de communication*, included in the CONNECT project, constitutes the background work for the thesis *Control Design for Multi-Agent systems under communication constraints*, we have followed a methodology more specific that highlights the main aspects of the project.

This methodology is composed of the followings steps:
1. Formation Model

2. Control Formation

3. Transformations of the circle Formation
   (a) Translation
   (b) Contraction
   (c) Rotation

4. Gradient Search

4 Formation Control

After the first month of the project, carrying out the review of the state of the art, we concluded that a kinematic model, as the model explained in [3], is appropriate for this system:

$$\dot{r}_k = ve^{i\theta_k} \quad (7)$$

$$\dot{\theta}_k = u_k \quad (8)$$

Henceforth, this model is used to represent the multi-agents system and all mathematical calculations take into account these equations. Although these equations can be simplified if we consider $v = 1$.

4.1 Circle Formation

According to the paper [3], the circle formation is possible through the application of only control law that depends on each agent. Thus, when the system becomes stable, all agents follows a circular trajectory the radius $\rho = 1/\omega_0$.

This control law is obtained applying the Lyapunov second theorem on stability to the following Lyapunov function candidate:

$$S(r, \theta) = \frac{1}{2} \sum_{k=1}^{N} |ve^{i\theta_k} - i\omega_0\tilde{r}_k|^2$$

where $\tilde{r}_k = r_k - R$ defines the relative distance between the position of each agent $r_k$ and the centre of mass of the formation $R$. If the aim is building a fixed circle, the equation $\dot{R} = 0$ is verified.

4.1.1 Circle with centre in $(0,0)$

In order to simplify the first implementation of the model, it is consider $R = (0,0)$, hence the centre of circle formation will be in the origin of coordinate system.
Developing the Lyapunov theorem:

\[
\dot{S}(r, \theta) = \sum_{k=1}^{N} \left[ <ve^{i\theta_k} - i\omega_0 \tilde{r}_k, -i\omega_0 \dot{r}_k > + <ve^{i\theta_k} - i\omega_0 \tilde{r}_k, i\omega_0 \dot{\theta}_k > \right]
\]

Choosing the control law as follows

\[
u_k = \omega_0 (1 + \kappa \langle \tilde{r}_k, \dot{r}_k \rangle)
\]

the system is stable because

\[
\dot{S}(r, \theta) = -\kappa \sum_{k=1}^{N} \langle \omega_0 \tilde{r}_k, \dot{r}_k \rangle^2 \leq 0
\]

therefore Lyapunov stability theorem is proved.

The program Matlab and its application Simulink give us a suitable simulation environment. The simulations deal with understand the theoretical equations and permit an appropriate analysis of the stability.

- **Simulation Parameters**

There are many parameters in the system equations and in the control law that have influence on simulation results. The time of convergence, system stability and the shape of the formation depend on the parameters values.

\( r_0 \): The initial conditions determines the duration of transitory response until achieve a stable formation. Besides, the final relative heading value of each agent is generated according to the initial conditions. We have carried out several simulations with random initial conditions. All of them were stables.

\( n \): The value of \( n \) represents the number of agents in the simulation. This is the main parameter to analyze the convergence time of the system. The convergence is faster with small values of \( n \). An usual number of agents used in the simulations is four.

\( v \): This is the speed of each agent. The circle radius is defined as \( \frac{v}{\omega_0} \) thus \( v \) determines the formation size.

\( \omega_0 \): This is a parameter of the control law. Due to the previous equation, the radius of the circle depends on \( \omega_0 \). On the other hand, the system stability is conditioned too but this influence depends mainly on the relation with \( \kappa \). We have chosen \( \omega_0 = 1 \) in almost all simulations.
\(\kappa\): This parameter appears on the control law hence it has influence on the system stability. After several simulations it is concluded that the better values for this parameter is \(\kappa = \omega_0 = 1\) thus the convergence time is reduced. For higher values like \(\kappa = 10\) each agent carries out a circular motion with bigger radius therefore convergence time is increased. Whereas the values as \(\kappa = 0.1\) produces independent circular motion with smaller radius for each agent. The convergence time is increased this time because the independent motion predominates over the part of the control law that search the final stable circle.

- Stability

Before, the stability of the system has been proved using the Lyapunov second theorem. Nevertheless it is worth mentioning that only control law is used and the speed of each agent is considered constant. Thus the agents must penetrate in the final stable trajectory through the several exterior circumferences.

In order to observe the system stability in the simulation, we have programmed a MATLAB file that generate an animation in real-time. The measurement of relative headings and relative angular velocity enhances the system stability.

The following figures show the procedure of control law convergence, through the relative measurements of \(\theta_k\) and \(\dot{\theta}_k\):

Each term \(\theta_k - \theta_j\) converges to a constant value because all agents have a fixed distribution in the final stable circle. Moreover, for each agent the relative angular velocity \(\dot{\theta}_k - \dot{\theta}_j\) converges to zero due to all agents track the same circular trajectory and turn with the same angular velocity. Thus all previous subtractions will be zero when the system becomes stable.
4.1.2 Circle with random coordinates of the centre

Following with the work developed in [3], the centre of mass is defined as

\[ \mathbf{R} = \frac{1}{N} \sum_{j=1}^{N} \mathbf{r}_j \]

hence the relative position of each agent can be expressed as follows:

\[ \tilde{\mathbf{r}}_k = \mathbf{r}_k - \mathbf{R} = \frac{1}{N} \sum_{j=1}^{N} \mathbf{r}_{kj} \]

where \( \mathbf{r}_{kj} = \mathbf{r}_k - \mathbf{r}_j \).

The control law obtained is the same that in the case of the centre in \((0,0)\). The main difference in the simulation results is the convergence time and obviously, the coordinates of the formation centre. The centre of the final stable circle depends on the initial conditions. The agents attempt to build a circle based in the centre of mass in each instant. Therefore the centre of stable circle will be positioned near the first system’s centre of mass that initial conditions establish. This is the reason for the faster stability in this second case.

4.1.3 Circle with centre in \((c_1, c_2)\)

In order to make progress in the methodology proposed, we contribute in the project development applying a variation in the model definition. The centre of mass is defined as a complex number \( \mathbf{R} = c_1 + ic_2 \). This change permits controlling the position of the circle because \( c_1 \) and \( c_2 \) are the coordinates of the centre. Henceforth, the centre of mass is called centre of the circle and its nomenclature is \( \mathbf{C} \), thus the definition of relative position changes: \( \tilde{\mathbf{r}}_k = \mathbf{r}_k - \mathbf{C} \)

The simulation results are the same than in the case of circle with centre in \((0,0)\). All simulation parameters have identical influence for the two cases.
The main difference in this model is that we can choose the centre of the final circle. This is an advantage for the later approach than we want to move the formation.

4.2 Uniform Distribution

The control formation has been developed in several ways. The simulations show the applicability and stability of each control law. After that, the following step is to achieve the uniform distribution of all agents in the formation. According to the paper [3], a potential function can be added to the formation control law for control the relative-headings.

The stability analysis begins modifying the Lyapunov function candidate. An other objective function is defined as follows:

\[ V(\mathbf{r}, \theta) = \kappa S(\mathbf{r}, \theta) + U(\theta) \]

Differentiating

\[ \dot{V}(\mathbf{r}, \theta) = \kappa \dot{S}(\mathbf{r}, \theta) + \frac{\partial U}{\partial \theta} \dot{\theta} \]

\[ \dot{V}(\mathbf{r}, \theta) = \sum_{k=1}^{N} \kappa < \omega_0 \mathbf{r}_k, \mathbf{i} \dot{\mathbf{r}}_k > (\omega_0 - u_k) + \frac{\partial U}{\partial \theta} \dot{\theta} \]

\[ \dot{V}(\mathbf{r}, \theta) = \sum_{k=1}^{N} |\kappa < \omega_0 \mathbf{r}_k, \mathbf{i} \dot{\mathbf{r}}_k > (\omega_0 - u_k) + \frac{\partial U}{\partial \theta} \dot{\theta}_k| \]

The function \( U(\theta) \) has an important quality:

\[ < \nabla U, 1 > = 0 \]

that is equivalent to

\[ \sum_{k=1}^{N} \frac{\partial U}{\partial \theta_k} = 0 \]

If this property is applied the previous equation can be expressed as:

\[ \dot{V}(\mathbf{r}, \theta) = \sum_{k=1}^{N} |\kappa < \omega_0 \mathbf{r}_k, \mathbf{i} \dot{\mathbf{r}}_k > - \frac{\partial U}{\partial \theta_k} \dot{\theta}_k| (\omega_0 - u_k) \]

Thus the control law changes as follows:

\[ u_k = \omega_0(1 + \kappa \langle \mathbf{r}_k, \mathbf{i} \dot{\mathbf{r}}_k \rangle) - \frac{\partial U}{\partial \theta_k} \]

where \( \frac{\partial U}{\partial \theta_k} \) is defined as:

\[ \frac{\partial U}{\partial \theta_k} = \frac{K}{N} \sum_{j=1}^{N} \sum_{m=1}^{[N/2]} \frac{\sin m \theta_{kj}}{m} \]

and \( \theta_{kj} = \theta_k - \theta_j \).

In this control law is included a new constant \( K \) hence it is necessary to know its influence.

Lara Briñón Arranz 17 Jun 2008
• Simulation Parameters

**previous parameters:** In this case, each previous parameter has the same influence in the convergence time, stability or radius value that in the formation control law.

**$K$:** This constant determines the weight of the relative heading control opposite the formation control. The $\kappa/K$ relation will decide what part has preference. Usually it is chosen a slow value for the relative heading control as $\kappa = 10K$ because it is the equilibrated relation to improve the convergence. If $K$ is ten times bigger, the uniform distribution is faster than circle formation. The system takes a longtime to form the final stable circle. In other hand, if $K$ is too small, the final circle will be formed quick but achieving the correct relative headings is more difficult due to the constant speed of all agents. The agents are forced to track several circular trajectories external to the final circle until the uniform distribution is completed.

• Stability

We are going to use a Matlab animation and the measurement of relative headings and relative angular velocity in order to analyze the system stability.

Again, for each agent the relative angular velocity $\dot{\theta}_k - \dot{\theta}_j$ converges to zero due to all agents track the same circular trajectory and turn with the same angular velocity. Each term $\theta_k - \theta_j$ converges to a constant value because all agents have a fixed distribution in the final stable circle. Nevertheless, the relative-heading control determines the value of each term. The value of relative heading must represent a uniform distribution in the circle, thus for each agent $\theta_k - \theta_j = \frac{2\pi}{N}(k - j)$ where $N$ is the number of agents in the system.

The following figures show the procedure of control law convergence, through the relative measurements of $\theta_k$ and $\dot{\theta}_k$:

We summarize the simulation parameters here below:
5 Variable Formation Control

The control of variations in the circle formation is a previous step before the gradient search. The development of this new approach begins gathering all the information of the review of state of the art.

5.1 Main Transformations

The final objective is to steer the formation until it finds the source searched depending on the concentration measures. This task only can be achieved if the formation carries out the basic motions in the plane (due to it is considered two-dimensional problem). The formation motion is composed to three main transformations:

1. Translation
2. Contraction and Expansion
3. Rotation

**Translation** A translation is moving every point a constant distance in a specific direction. The translation motion can also be interpreted as the movement that change the position of an object.

A circle undergoes a translation when the coordinates of its centre changes the value and the radius remain constant. Thus, the translation can be carried out controlling the value of \((c_1, c_2)\)
Contraction and Expansion A contraction or expansion is changing the size of the formation without changing its shape. In a circle this transformation depends on its radius. When the radius is reduced, it is called contraction and the expansion is an increase of its radius. Then, this transformation can be controlled changing $\omega_0 = 1/\rho$.

Rotation A rotation is a movement which keeps a point fixed. This definition applies to an ellipse can be interpreted as the change of the orientation of every point keeping a focus fixed. Nevertheless, in a circle, any change keeping the centre fixed results the same circle. Then, we are not going to consider this transformation.

5.2 Limitation of the previous schemes

The first idea for check the ability of the system achieving transformations and variations, is to deal with change the simulation parameters directly. The constant parameters which have influence in the circle transformations are $\omega_0$, it determines the radius of the circle, and $C$ than represents the coordinates of its centre. We must to simulate each transformation separately thus the influence of each variation in the system stability can be analyzed.

With the simulation results and the analysis of the control law and model equation, we can conclude that there is a transitory response when we impose an abrupt variation in the circle parameters $\omega_0$ and $C$. We show these conclusions here below.

5.2.1 Transitory response

The control law described previously for the circle formation can’t keep the system stability while a sudden variation is applied to the variables $\omega_0$ and $C$. This is, if the parameters change quickly, the formation is destroyed during the contraction or translation respectively.

Nevertheless the control law is able to achieve the objective. if the aim is the formation of a circle with bigger or smaller radius or if it is the formation of the circle with others coordinates of the centre, finally the system is stable.

With these results, we think that this system can reach the stability despite the perturbations. Our objective is to verify if the system is stable subject to soft variations.

The measurement of the relative headings determines the stability of the system. We have made several simulations to show the transitory response. After attaining the stable formation, we introduce a variation step in the parameters for $t = 100\text{seconds}$. In the circle contraction, the variation is produced from $\omega_0 = 0.3$ to $\omega_0 = 1.5$. The simulation for change the coordinates of the centre the variation is from $(-1, -1)$ to $(2, 2)$. We can observe that the relative headings change suddenly. Although after the transitory response the system becomes stable.

Surprisingly, the expansion of the circle is more stable than the contraction.

The duration of the transitory response depends mainly on the number of agents. All these results are shown in the followings figures:
5.2.2 Slow variations

After the previous results, we are going to apply softer variations. Each variation is simulated individually and then, both transformations will be analyzed together.

**Translation**  The coordinates of centre are new input variables for the system. The transformation is carried out introducing a soft ramp instead of a value constant for \( c_1 \) and \( c_2 \).

This ramp is attenuated by means of a second order filter which avoids the sudden changes that can affect the formation stability. The simulation results are satisfactory. The translation of the circle is produced without lost of the stability. That is, the formation is conserved during the motion.
Figure 6: (a) Relative headings during translation motion. (b) Relative angular velocity during translation motion.

Depending on the slope of the ramp the system keeps the stable value for the relative headings. If the ramp is too hard the formation is conserved but the agents stop having an uniform distribution.

**Contraction and Expansion** In this other variation analysis, the aim is change the radius of the circle $\rho = 1/\omega_0$. Then, $\omega_0$ is a new input variable for the system. Again the transformation is carried out introducing a soft ramp as $\omega_0$ value. This ramp is attenuated by means of a second order filter which avoids the sudden changes that can affect the formation stability.

Figure 7: Contraction and Expansion of the circle. The figure shows the formation during the contraction motion between the two stable position initial and final.

The simulation shows that the circle is stable during the contraction and expansion motion. The formation and relative headings are conserved.

As the previous transformation (translation), depending on the slope of the ramp the system keeps the stable value for the relative headings. If the ramp is too hard the formation is conserved.
but the agents stop having an uniform distribution.

![Graph](image1.png)

![Graph](image2.png)

**Figure 8:** (a) Relative headings during contraction motion. (b) Relative angular velocity during contraction motion.

This figures shows that the circle keeps the relative headings during the contraction transformation. The system is really stable although the perturbations in the value of $\omega_0$ are introduced.

**Combined Motions** To conclude this section, we present the simulation results for the combined motions. Both transformations separately have resulted feasible and the system is stable for the two cases. If the translation and contraction are produced in the same simulation the system is still stable. The formation conserves the uniform distribution of the agents in the circle during all combined motion. Whith this conclusions we have come to the end of the slow variations section. From here on we can develop the gradient search using $\omega_0$ and $(c_1, c_2)$ as system perturbations which permits the control of circle motion.

**5.3 Control Redesign based on previous schemes**

The stability and performances of the system have been analyzed introducing externals perturbations. The conclusion is that the control law is efficient for carry out the transformations of the circle but we want to control the transformation parameters directly. The following step is redesign a control law that permits including the variation of the parameters $\omega_0$ and $(c_1, c_2)$ in the control closed-loop.

We search to achieve a contribution in the multi-agents and formation control fields. This is a mainly objective, as the formation control, analyzed previously, in this project.

At the beginning we are going to develop the control law for the contraction and expansion due to the previous results which shows that the system is more stable with this kind of transformation.
### 5.3.1 Contraction Algorithm

According to the section *Formation Control*, we search a Lyapunov function candidate which represents the stability objective. In this case, the parameter \( \omega_0 \) is variable then, henceforth, it is called \( \omega_d \). We choose the previous Lyapunov function but now it depends on \( \omega_d \):

\[
\dot{V}(r, \theta, \omega_d) = \kappa \dot{S}(r, \theta, \omega_d) + \frac{\partial U}{\partial \theta_k} \dot{\theta}_k
\]

where

\[
S(r, \theta, \omega_d) = \frac{1}{2} \sum_{k=1}^{N} |e^{iu_k} - i\omega_d \tilde{r}_k|^2
\]

Using the following notation: \( f(r_k, \theta_k, \omega_d) = e^{iu_k} - i\omega_0 \tilde{r}_k \) the derivative of previous function is defined as:

\[
\dot{S}(r, \theta, \omega_d) = \sum_{k=1}^{N} < f(r_k, \theta_k, \omega_d), \frac{\partial f(r_k, \theta_k, \omega_d)}{\partial r_k} \dot{r}_k > + \sum_{k=1}^{N} < f(r_k, \theta_k, \omega_d), \frac{\partial f(r_k, \theta_k, \omega_d)}{\partial \theta_k} \dot{\theta}_k > + \sum_{k=1}^{N} < f(r_k, \theta_k, \omega_d), \frac{\partial f(r_k, \theta_k, \omega_d)}{\partial \omega_d} \dot{\omega}_d >
\]

Calculating all partial derivatives, the development of the Lyapunov theorem continues:

\[
\dot{S}(r, \theta, \omega_d) = \sum_{k=1}^{N} < \dot{r}_k - i\omega_0 \tilde{r}_k, i\omega_d \dot{r}_k > + \sum_{k=1}^{N} < \dot{r}_k - i\omega_0 \tilde{r}_k, i\dot{r}_k \dot{\theta}_k > + \sum_{k=1}^{N} < \dot{r}_k - i\omega_0 \tilde{r}_k, -i\tilde{r}_k \dot{\omega}_d >
\]

The two first terms can be merged as the formation control obtaining:

\[
\dot{S}(r, \theta, \omega_d) = \sum_{k=1}^{N} < \omega_d \tilde{r}_k, \dot{r}_k > (\omega_d - u_k) + \dot{\omega}_d \sum_{k=1}^{N} (< \dot{r}_k, -i\tilde{r}_k > + < -i\omega_d \tilde{r}_k, -i\tilde{r}_k >)
\]

Therefore the derivative of Lyapunov function can be expressed as:

\[
\dot{V}(r, \theta, \omega_d) = \kappa \sum_{k=1}^{N} < \omega_d \tilde{r}_k, \dot{r}_k > (\omega_d - u_k) + \frac{\partial U}{\partial \theta_k} \dot{\theta}_k + \kappa \dot{\omega}_d \sum_{k=1}^{N} (< \dot{r}_k, -i\tilde{r}_k > + < -i\omega_d \tilde{r}_k, -i\tilde{r}_k >)
\]
\[ \dot{V}(r, \theta, \omega_d) = \sum_{k=1}^{N} (\kappa \langle \omega_d \tilde{r}_k, \dot{r}_k \rangle + \frac{\partial U}{\partial \theta_k})(\omega_d - u_k) + \kappa \dot{\omega}_d \sum_{k=1}^{N} (\langle \dot{\tilde{r}}_k, -i\tilde{r}_k \rangle + \omega_d |\tilde{r}_k|^2) \]

with the control law as follows we pretend minimize the new Lyapunov function:

\[ u_k = \omega_d (1 + \kappa \langle \dot{\tilde{r}}_k, \dot{r}_k \rangle - \frac{\partial U}{\partial \theta_k} + a_k) \]

where \( a_k \) search erase the not positives termes in \( \dot{V}(r, \theta, \omega_d) \):

\[ a_k = \frac{\kappa \dot{\omega}_d (\langle \dot{\tilde{r}}_k, -i\tilde{r}_k \rangle + \omega_d |\tilde{r}_k|^2)}{\kappa \langle \omega_d \tilde{r}_k, \dot{r}_k \rangle + \frac{\partial U}{\partial \theta_k}} \]

The theorem of Lyapunov is verified:

\[ \dot{V}(r, \theta, \omega_d) = -\sum_{k=1}^{N} (\kappa \langle \omega_d \tilde{r}_k, \dot{r}_k \rangle - \frac{\partial U}{\partial \theta_k})^2 \leq 0 \]

We are going to make several simulations in the order to verify the control law obtained for the contraction of the circle. The simulink model (see attached document), consists of the formation control model and we add the new control term \( a_k \).

Again the transformation is carried out introducing a soft ramp as \( \omega_0 \) value. This ramp is attenuated by means of a second order filter which avoids the sudden changes that can affect the formation stability. It is necessary to include an other filter to obtain \( \dot{\omega}_d \) since it is an input variable in the new term of the control law.

![Figure 9: Contraction control. This figure shows the formation during the contraction motion between the two stable position initial and final. The circle don't keep the uniform distribution.](image)

The simulation results show that the system is finally stable because it is able to achieve the aim (change the radius of the formation) but during the motion, the circle loses the uniform distribution. The relative headings are kept only if the ramp is too soft.

Analyzing several times the equations which composes the Lyapunov method, we have not found calculation mistakes. Thus the simulations results are unsatisfactory due to two possible causes:
1. Implementation mistakes in the program code. The Matlab file and the simulink model have been debugged for one week. Even so, it has not been found obvious mistakes to conclude that it is the reason.

2. Concept error. It is possible that some hypothesis used in the calculation of the formation control is not true any more considering $\omega_d$ variable.

### 6 Gradient Search Method

The gradient research is the main challenge of this project. We have develop the control laws previous for the circle formation and the analysis of the variation formation with the idea of moving the circle according to the measurement of each agent.

In order to develop the control for the gradient research we make a change of coordinates as follows:

![Change of coordinates](image)

**Equation for each agent:**

$$
\begin{align*}
\dot{r}_k &= ve^{i\theta_k} \\
\dot{\theta}_k &= \omega_d(1 + \kappa < \tilde{r}_k, \dot{r}_k>) - \frac{\partial U}{\partial \theta_k}
\end{align*}
$$

Stable circle formation system:

$$
\begin{align*}
\dot{r}_k &= ve^{i\psi}e^{i\theta_k^*} \\
\psi &= \omega_d
\end{align*}
$$

where $\theta_k^* = 2\pi \frac{k-1}{N}$

This new system can be considered because two reasons: When the circle formation is stable, all relative headings converges to zero, therefore $\frac{\partial U}{\partial \theta_k}$ converges to zero too. The second reason is that the formation control law can be separated in two parts: $\omega_d$ is the slow term and $\kappa < \tilde{r}_k, \dot{r}_k)$ is the fast term. Thus the control law converges to the slow term.
We choose a change of coordinate system turning with the same angular speed $\omega_d$ of each agent and its origin is in the coordinates of the centre of the circle formation $(c_1, c_2)$.

\[
\hat{\mathbf{r}}_k = \mathbf{r}_k e^{-i\psi}
\]  
(11)

\[
\dot{\hat{\mathbf{r}}}_k = \dot{\mathbf{r}}_k e^{-i\psi} - i\dot{\psi} \mathbf{r}_k e^{-i\psi}
\]  
(12)

And finally if we define $\dot{c}_d e^{-i\psi} = \hat{c}_d$ the new equation model is:

\[
\dot{\hat{\mathbf{r}}}_k = ve^{i\theta^*_k} - \hat{c}_d - i\omega_d \hat{\mathbf{r}}_k
\]

The Lyapunov function, that we want to maximize this time, depends on the measurements of each agent, called $\eta_k$. The objective is to approach the values of measurements to the highest source value. Hence, it is attempted to achieve the purpose using this function:

\[
V(\eta) = \frac{1}{2} \sum_{k=1}^{N} \eta_k^2
\]

Differentiating

\[
\dot{V}(\eta) = \sum_{k=1}^{N} \eta_k \dot{\eta}_k
\]

Due to that, each measurement is a real number but the gradient of the concentration function is a vector, we have to introduce the definition as follows for the derivative of measurements:

\[
\dot{\eta}_k(\hat{\mathbf{r}}_k) = \langle \frac{\partial \eta_k}{\partial \hat{\mathbf{r}}_k}, \dot{\hat{\mathbf{r}}}_k \rangle = \langle \frac{\partial \eta_k}{\partial \hat{\mathbf{r}}_k}, ve^{i\theta^*_k} - \hat{c}_d - i\omega_d \hat{\mathbf{r}}_k \rangle
\]

in such a way that it is possible to tackle the Lyapunov derivative:

\[
\dot{V}(\eta) = \sum_{k=1}^{N} \eta_k \langle \frac{\partial \eta_k}{\partial \hat{\mathbf{r}}_k}, ve^{i\theta^*_k} - \hat{c}_d - i\omega_d \hat{\mathbf{r}}_k \rangle
\]

If we define

\[
\omega_d = \sum_{k=1}^{N} \eta_k \langle \frac{\partial \eta_k}{\partial \hat{\mathbf{r}}_k}, -\hat{\mathbf{r}}_k \rangle
\]

a quadratic term is obtained in the Lyapunov function $\dot{V}(\eta_k)$.

Moreover we can separate the others terms in their real components for search a control equation for $\hat{c}_d$ as follows:

\[
\sum_{k=1}^{N} \eta_k \langle \frac{\partial \eta_k}{\partial \hat{\mathbf{r}}_k}, -\hat{c}_d \rangle = -\sum_{k=1}^{N} \eta_k \langle \frac{\partial \eta_k}{\partial \hat{\mathbf{r}}_k}, \hat{c}_d x \rangle + \langle \frac{\partial \eta_k}{\partial \hat{\mathbf{r}}_k}, \hat{c}_d y \rangle
\]

Thus it seems logical to define the two real components that they forms an other quadratic term:

\[
\hat{c}_d x = -\sum_{k=1}^{N} \eta_k \frac{\partial \eta_k}{\partial \hat{\mathbf{r}}_k} |_x
\]  
(13)

\[
\hat{c}_d y = -\sum_{k=1}^{N} \eta_k \frac{\partial \eta_k}{\partial \hat{\mathbf{r}}_k} |_y
\]  
(14)
Although, the real components equations are going to be used for simulate, it is possible to find a complex expression:

\[
\hat{c}_d = - \sum_{k=1}^{N} \eta_k \frac{\partial \eta_k}{\partial \hat{r}_k}
\]

The complex equation permits achieving the change of coordinate system readily:

\[
\hat{c}_d = \hat{c}_k e^{-i\psi} \Rightarrow \hat{c}_d = \hat{c}_d e^{i\psi}
\]

However it is necessary to define the concentration gradient for obtaining it in the coordinates of original system:

\[
\frac{\partial \eta_k}{\partial \hat{r}_k} = \frac{\partial \eta_k}{\partial r_k} \frac{\partial r_k}{\partial \hat{r}_k}
\]

with this definition the expression for the control of the centre:

\[
\dot{c}_d = -e^{i\psi} \sum_{k=1}^{N} \eta_k \frac{\partial \eta_k}{\partial r_k} e^{i\psi} = \sum_{k=1}^{N} \eta_k \frac{\partial \eta_k}{\partial r_k} e^{i2\psi}
\]

\[
\dot{c}_{dx} = -\sum_{k=1}^{N} \eta_k [\frac{\partial \eta_k}{\partial r_k} |_x \cos 2\psi - \frac{\partial \eta_k}{\partial r_k} |_y \sin 2\psi] \quad (15)
\]

\[
\dot{c}_{dy} = -\sum_{k=1}^{N} \eta_k [\frac{\partial \eta_k}{\partial r_k} |_y \cos 2\psi + \frac{\partial \eta_k}{\partial r_k} |_x \sin 2\psi] \quad (16)
\]

Nevertheless there is a not quadratic term in the Lyapunov function. The influence of this third term is unknown.

\[
\dot{V}(\eta) = \left[ - \sum_{k=1}^{N} \eta_k \frac{\partial \eta_k}{\partial r_k} |_x \right]^2 + \left[ - \sum_{k=1}^{N} \eta_k \frac{\partial \eta_k}{\partial r_k} |_y \right]^2 + \left[ \sum_{k=1}^{N} \eta_k \right] < \frac{\partial \eta_k}{\partial r_k}, -\hat{r}_k > + \sum_{k=1}^{N} \eta_k \left< \frac{\partial \eta_k}{\partial r_k}, ve^{i\theta_k} \right>
\]
7 Conclusions

This project consists of two main objectives, the gradient search and the formation control that permits to carry out this search. We have explained previously the different steps to achieve each aim. Despite that the review of the state of the art developed supplies the necessary information to tackle each approach, we have reduced the items according to the durations of the project.

As it has been explained in the introduction, this project represents a thesis blackout. The thesis will tackle the approaches and challenges that this project has not been able to achieve.

7.1 Review of results

We summarize the simulation results and the mathematical conclusions presented in the previous sections, here below.

Formation control Choosing the system model and its representation mathematical, the development of formation control begins. We have followed the paper [3] obtaining a control law for a circle formation. The main challenge has been to understand every aspects of the Lyapunov stability theorem and its development. The understanding of these questions has permitted to attempt new control laws.

The control law used determines a stable formation for the four cases analyzed and simulated with satisfactory results:

1. Circle with centre in (0, 0)
2. Circle with random coordinates of centre
3. Circle with centre in \((c_1, c_2)\)
4. Uniform distribution, that is applicable to the three previous cases and the system continues stable.

Variation control Introducing slow variations of the parameters \(\omega_0\) and \((c_1, c_2)\) the simulation results show that the stability of the system is conserved. Thus it is possible to use these parameters as input variables to control the gradient search.

We have developed a new contraction law. This control law is obtained applying the Lyapunov method, but the simulation results show lost of stability during the transformation.

Gradient search The main difficulty is to achieve a suitable approach to develop the control law which carries out the gradient search. Working on the mathematical equations we have found a possible control law. This law is obtained with verify the Lyapunov stability theorem. The simulation results don’t show a stable system and the gradient search is only carried out with fixed parameters. This control law is not very robust.
7.2 Perspectives

Applying the satisfactory results obtained the following step is to complete the project by means of the thesis *Control Design for Multi-Agent systems under communication constraints*. This project supplies the mathematical base and the appropriated simulation environment for the development of several items:

- Improvement of the variation control laws (contraction and translation).
- To achieve a robust control law for the gradient search.
- Development of new models assuming communication constraints. According to the paper [5] the better solution is a graph model.
- To include a term in the formation control for the collision avoidance. The flocking algorithm seems more appropriate for the AUVs.
8 Review in French

8.1 Introduction

8.1.1 Contexte de stage

La communauté scientifique dans le domaine du contrôle s’est focalisée principalement dans la commande de la formation des véhicules. Cependant, les systèmes de coopération sont apparus comme sujets d’intérêt importants les dernières années.

Le stage est inclus dans le projet CONNECT, lequel a été fondé par the ARN (National Research Agency. Le projet traite le problème de la commande de systèmes composés des plusieurs sous-systèmes connectés entre eux grâce à une ressort de communication hétérogène. Le principal objectif est d’apprendre comment on peut dessiner des lois de contrôle en train de préserver la stabilité du système en boucle fermée. Le project CONNECT appartient à l’équipe NeCS (Networked Controlled System Team) lequel est un ensemble des laboratoires INRIA et GIPSA-lab soutenus par CNRS, INRIA, INPG et UJF.

Dans le cas précis de ce stage, on cherche à aborder le problème du contrôle des multi-agents systèmes avec des contraintes de communication. On veut réussir à ce qu’une flottille d’agents sous-marins travaillent en collaboration, pour réaliser la recherche (par gradient) et la localisation d’une source (eaux douce, source chimique, source de méthane, etc.). Chaque AUV est équipé d’un capteur qui mesure la concentration de la source, et d’un système de communication (sonar) pour échanger des données relatives à leur position absolue. L’objectif de la commande est de commander la flottille à fin de réaliser la tâche d’inspection décrite précédemment sous des contraintes de communication.

8.1.2 Étude de l’état de l’art

On commence avec la réalisation d’une étude de l’état de l’art. Nous allons résumer les travaux les plus caractéristiques sur le domaine du contrôle de systèmes distribués, contrôle d’une formation et d’algorithmes pour rechercher le gradient, ci-dessous.

Modèles On a trouvé plusieurs modèles pour représenter le mouvement des agents. Ils peuvent être formulés en trois dimensions, mais pour simplifier, on va utiliser les modèles en deux dimensions traités dans [1, 2, 3].

Modèles cinématiques: Beaucoup d’études réalisées dans le domaine du contrôle des systèmes multi-agent proposent des modèles simples . On trouve principalement des modèles avec une masse ponctuelle du type: \( \dot{x}_i = u_i \) ou modèles linéaires. Cependant, pour les applications concernant les robots sous-marins, il semble qu’un modèle non linéaire est plus approprié [3]:

\[
\begin{align*}
\dot{r}_k &= v e^{i \theta_k} \\
\dot{\theta}_k &= u_k
\end{align*}
\]
Modèles dynamiques: Ces modèles sont moins utilisés dû à leur complicité. Ils tiennent compte de la force appliquée à l’agent, [1].

Modèles pour un groupe: On introduit nouvelles variables pour décrire le mouvement du groupe. Le centre de masse est défini en [3]: \( R = \frac{1}{N} \sum_{j=1}^{N} r_k \)

Contrôle de la formation On montre quelques stratégies pour la commande de la formation. On considère que tous les agents peuvent communiquer avec tous les autres.

1. Contrôle basé sur Lyapunov

**Mouvement circulaire et distribution uniforme:** L’article [3] propose la loi du contrôle suivant pour accomplir les deux objectives:

\[
\dot{u}_k = \omega_0 \left( 1 + \kappa \langle \tilde{r}_k, \dot{r}_k \rangle \right) - \frac{\partial U}{\partial u_k},
\]

Cette loi est obtenue en appliquant le théorème de stabilité de Lyapunov.

**Groupes coordonnées.**

**Commande pour une forme déterminée.**

2. Algorithmes basés sur Laplacien

**Gossip:** C’est un protocole qui est dessiné pour imiter la perte d’information quand les agents font des commérages pour les mettre en relation avec les contraintes de communication.

**Consensus:** On définit les concepts graphe et voisins pour un ensemble d’agents. Le graphe permet de connaître les relations de communication de chaque agent avec ses voisins.

**Contrôle d’une formation distribuée.**

3. Contrôle qui évite les obstacles Flocking, mot défini en [12], c’est une forme qui représente un groupe d’agents connectés entre eux avec un objectif commun. Cet algorithme utilise le contrôle de consensus pour définir les obstacles de façon à ce qu’ils soient considérés comme de nouveaux voisins dans le graphe.

**Contrôle pour chercher une source** Il y a plusieurs travaux qui traitent de la recherche d’une source. On trouve principalement les modèles de la source qui suivent une équation diffusive.

**Formations avec une forme déformable:** On essaie d’ajuster cette forme aux courbes de niveau de la source.

**Adaptation à la croissance du gradient:** On détermine une fonction objective pour la recherche du gradient qui sera maximisée.

Lara Briñón Arranz 32 Jun 2008
8.2 Méthodologie

Après avoir réuni les différents approches sur le contrôle des systèmes multi-agents on confectionne la méthodologie nécessaire pour développer ce stage. Les objectifs proposés sont détaillés ensuite:

1. Modèle de la formation
2. Contrôle de la formation
3. Transformations du cercle
   (a) Translation
   (b) Contraction
   (c) Rotation
4. Recherche du gradient

8.3 Contrôle de la formation


Pour développer chaque section, on va effectuer une analyse théorique et une application pratique. On fera les simulations pour vérifier les résultats obtenus, avec le program Matlab et Simulink.

8.3.1 Formation du cercle

Le modèle non linéaire suivant est considéré pour représenter notre système:

\[
\dot{r}_k = ve^{i\theta_k} \quad (20)
\]
\[
\dot{\theta}_k = u_k \quad (21)
\]

Après ce choix on simule le système en boucle fermée avec la loi de commande exprimée en [3] pour la formation du cercle:

\[
u_k = \omega_0(1 + \kappa(\bar{r}_k, \tilde{r}_k))
\]

ou \(\bar{r}_k = r_k - R\).

Cette formule a été développée grâce au théorème de stabilité de Lyapunov, appliqué à la fonction

\[
S(r, \theta) = \frac{1}{2} \sum_{k=1}^{N} |ve^{i\theta_k} - i\omega_0\bar{r}_k|^2
\]

Les résultats des simulations montrent que la loi de commande utilisée est capable de former le cercle de radius \(\rho = 1/\omega_0\) et stabiliser le système pour les trois cas suivants:

Lara Briñón Arranz 33 Jun 2008
Cercle avec centre en $(0, 0)$: Pour obtenir le cercle en l’origine de coordonnées, il faut annuler le centre de masse du système $R = 0$. Les valeurs des angles relatifs sont constantes par conséquent la formation est stable.

Cercle avec coordonnées aléatoires du centre: On définit le centre de masse $R = \frac{1}{N} \sum_{j=1}^{N} r_k$. Le centre du cercle dépend des conditions initiales. Dû à cette définition, le système réussit à être stable plus rapidement.

Cercle avec centre en $(c_1, c_2)$: Les résultats des simulations sont égaux que dans le premier cas. Pourtant, affiner la définition du centre de masse comme cela $R = C$ permettra d’accomplir la translation du cercle.

8.3.2 Distribution uniforme

Une fois qu’on a réussi à contrôler la formation, le but suivant est distribuer les agents de façon uniforme dans le cercle. On peut réaliser cette tâche rien qu’en ajouter un terme à la loi de commande:

$$ u_k = \omega_0 (1 + \kappa \langle \hat{r}_k, \hat{r}_k \rangle) - \frac{\partial U}{\partial \theta_k} $$

où ce terme est défini comme suit:

$$ \frac{\partial U}{\partial \theta_k} = -\frac{K}{N} \sum_{j=1}^{N} \sum_{m=1}^{[N/2]} \frac{\sin m \theta_{kj}}{m} $$

et $\theta_{kj} = \theta_k - \theta_j$.

On déduit cette expression parce qu’on a introduit une nouvelle fonction potentielle dans la fonction de Lyapunov:

$$ V(r, \theta) = \kappa S(r, \theta) + U(\theta) $$

Les simulations réalisées pour ce cas montrent que le système est stable et encore les angles relatifs vérifient $\theta_k - \theta_j = \frac{2\pi}{N} (k - j)$ où $N$ est le nombre d’agents. Par conséquent, les agents ont une distribution uniforme dans le cercle.

8.4 Contrôle des variations de la formation

Les principaux transformations qu’on peut réaliser dans une formation sont:

1. Translation
2. Contraction
3. Rotation

Pourtant, pour un cercle la rotation n’a aucun sens, donc on va considérer seulement les deux premières transformations.
Réponse transitoire: On commence en analysant les réponses du système par rapport les variations des paramètres $\omega_0$ et les coordonnées du centre. On conclut que si les variations de ces paramètres sont brusques, le système arrête d’être stable pendant le transformation.

Variations lentes: Par conséquent, l’objectif suivant est d’évaluer la stabilité en introduisant des variations doucement. Après avoir réalisé plusieurs simulations on peut affirmer que chaque transformation est accomplie sans perdre la stabilité avec une étendue marge de variation des paramètres. De plus la distribution uniforme est conservée autant pour la translation que pour la contraction.

Contrôle de la contraction: Bien qu’on a réussi à déplacer la formation et à changer sa forme, on veut inclure le contrôle des transformations dans la boucle fermée du système global. On a contribué en développant une loi de commande qui essaie de faire stable le système pendant la contraction si on considère $\omega_0 = \omega_d$ variable. La fonction de Lyapunov pour analyser la stabilité est égale à l’antérieure mais quand on derive, il faut deriver sur $\omega_d$. Finalement on obtiens:

$$u_k = \omega_d(1 + \kappa < \tilde{r}_k, \dot{r}_k >) - \frac{\partial U}{\partial \theta_k} + a_k$$

où $a_k$ cherche à éliminer les terms négatifs dans la dérivée de la fonction de Lyapunov $V(r, \theta, \omega_d)$:

$$a_k = \frac{\kappa \omega_d(< \tilde{r}_k, -i\tilde{r}_k > + \omega_d|\tilde{r}_k|^2)}{\kappa < \omega_d \tilde{r}_k, \dot{r}_k > + \frac{\partial U}{\partial \theta_k}}$$

Cependant les simulations montrent que le système devient instable pendant la transformation.

8.5 Recherche du gradient

Une fois qu’on a réalisé la commande de la formation d’une manière satisfaisante, l’objectif essentiel est de chercher une méthode pour accomplir la recherche du gradient.

Les paramètres $\omega_d$ et $(c_1, c_2)$ peuvent être considérés variables d’entrée au système. On va développer une loi de commande pour chaque paramètre en dépendant des mesures de la concentration pris pour les agents. Pour ce faire, on propose une changement des coordonnées du système qui tourne avec une vitesse angulaire égale à la rotation de chaque agent dans le cercle:

$$\tilde{r}_k = \tilde{r}_k e^{-i\psi} \quad (22)$$
$$\dot{\tilde{r}}_k = \dot{\tilde{r}}_k e^{-i\psi} - i\tilde{r}_k \psi \tilde{r}_k e^{-i\psi} \quad (23)$$

On définit une nouvelle modèle pour le cercle stable:

$$\dot{r}_k = vr e^{i\theta_k} \quad (24)$$
$$\dot{\psi} = \omega_d \quad (25)$$

Lara Briñón Arranz
Jun 2008
où $\theta^r_k = 2\pi \frac{k-1}{N}$.

Avec ces équations on cherche à maximiser la fonction de Lyapunov:

$$V(\eta) = \frac{1}{2} \sum_{k=1}^{N} \eta^2_k$$

puisque vers la source, les valeurs des mesures $\eta_k$ seront plus grandes. Donc, si la fonction de Lyapunov grandit, ça veut dire qu'on s’approche à la source.

Après plusieurs calculs on trouve les suivants lois de commande pour chaque paramètre:

$$\omega_d = \sum_{k=1}^{N} \eta_k < \frac{\partial \eta_k}{\partial \hat{r}_k}, -\hat{r}_k >$$

(26)

$$\dot{c}_d = - \sum_{k=1}^{N} \eta_k \frac{\partial \eta_k}{\partial r_k} e^{i\psi}$$

(27)

il faut tenir compte de changement des coordonnées:

$$\frac{\partial \eta_k}{\partial r_k} = \frac{\partial \eta_k}{\partial r_k} e^{i\psi}$$

Mais ces formules ne réussissent pas rendre positifs tous les termes de la fonction de Lyapunov, c’est-à-dire, on ne peut pas assurer la stabilité du système.

Effectivement, les simulations ne sont pas satisfaisantes. On a trouvé une combinaison de paramètres qui accomplit la recherche du gradient. C’est la raison pour laquelle on considère que la loi de commande obtenue est près d’être la solution mais il faut analyser attentivement les calculs et améliorer l’implémentation des simulations.

8.6 Conclusions

Ce projet permet d’analyser plusieurs facettes différentes dans le domaine du systèmes multi-agents. On a appris à développer la méthode de Lyapunov pour chercher la loi de commande qui fait stable le system. Les formules théoriques ont été appliquées d’une manière satisfaisante pour le contrôle de la formation du cercle. Les lois de commande utilisées gardent la formation stable et la distribution uniforme même si les paramètres, $\omega_0$ et les coordonnées du centre, varient doucement.

On a trouvé difficultés pour accomplir la recherche du gradient dû à la complexité des équations qui représentent le système comme un cercle stable et aussi dû au problème de définir la valeur du gradient grâce à les mesures pris pour chaque agent.

Ce stage constitue l’ensemble des connaissances pour développer la thèse dans le même domaine qui nous occupera les trois prochaines années Control Design for Multi-Agent systems under communication constraints. Par conséquent, toutes les simulations non satisfaisantes seront analysées au même temps qu’on aborde les diverses approches pour le problème des contraintes de communication.
9 Attached Document A

9.1 Operators

**Operator** $<,>$ **inner product:**

$a, b \in \mathbb{C}^N$ therefore the operator is defined as:

$$<a, b> = \text{Re}\{\bar{a}^T b\}$$

This definition can be used for developer the derivate of the same kind functions witch we use as Lyapunov functions.

If

$$V = \frac{|f(\varphi)|^2}{2} = \frac{1}{2}(f(\varphi)\overline{f(\varphi)})$$

where $f : \mathbb{R}^N \to \mathbb{C}^N$ then

$$\dot{V} = \frac{\partial}{\partial t} \left[ \frac{1}{2}(f(\varphi)\overline{f(\varphi)}) \right] = \frac{1}{2}\left[ f(\varphi)\overline{\dot{f}(\varphi)} + f(\varphi)\dot{\overline{f(\varphi)}} \right]$$

9.2 Calculations developed

9.2.1 Formation Control

In order to obtain the formation control law, we apply the Lyapunov second theorem on stability to the following Lyapunov function candidate:

$$S(r, \theta) = \frac{1}{2} \sum_{k=1}^{N} |ve^{i\theta_k} - i\omega_0 \tilde{r}_k|^2$$

where $\tilde{r}_k = r_k - R$ defines the relative distance between the position of each agent $r_k$ and the centre of mass of the formation $R$.

Using the following notation $f(r_k, \theta_k) = ve^{i\theta_k} - i\omega_0 \tilde{r}_k$ then

$$S(r, \theta) = \frac{1}{2} \sum_{k=1}^{N} |f(r_k, \theta_k)|^2$$

and the derivative of the previous function is defined as:

$$\dot{S}(r, \theta) = \sum_{k=1}^{N} <f(r_k, \theta_k), \frac{\partial f(r_k, \theta_k)}{\partial r_k} \dot{r}_k> + \sum_{k=1}^{N} <f(r_k, \theta_k), \frac{\partial f(r_k, \theta_k)}{\partial \theta_k} \dot{\theta}_k>$$

Partial derivatives:

$$\frac{\partial f(r_k, \theta_k)}{\partial r_k} = -i\omega_0$$

$$\frac{\partial f(r_k, \theta_k)}{\partial \theta_k} = ive^{i\theta_k} = i\dot{r}_k$$
and we have defined previously $\dot{\theta}_k = u_k$, therefore

$$
\dot{S}(r, \theta) = \sum_{k=1}^{N} < f(r_k, \theta_k), -i\omega_0 \dot{r}_k > + \sum_{k=1}^{N} < f(r_k, \theta_k), i\dot{r}_k u_k >
$$

$$
< f(r_k, \theta_k), -i\omega_0 \dot{r}_k > = -\omega_0 < f(r_k, \theta_k), -i\omega_0 \dot{r}_k >
$$

$$
< f(r_k, \theta_k), i\dot{r}_k u_k > = < f(r_k, \theta_k), i\dot{r}_k > u_k
$$

then

$$
\dot{S}(r, \theta) = \sum_{k=1}^{N} < f(r_k, \theta_k), i\dot{r}_k > (u_k - \omega_0) = \sum_{k=1}^{N} < ve^{i\theta_k} - i\omega_0 \dot{r}_k, i\dot{r}_k > (u_k - \omega_0) = \sum_{k=1}^{N} < \dot{r}_k - i\omega_0 \dot{r}_k, i\dot{r}_k > (u_k - \omega_0) = \sum_{k=1}^{N} [< \dot{r}_k, i\dot{r}_k > - \omega_0 < i\dot{r}_k, i\dot{r}_k >](u_k - \omega_0) =
$$

Operating:

$$
< \dot{r}_k, i\dot{r}_k >= Re\{\overline{r_k}^T i\dot{r}_k\} = Re\{(\dot{x}_k - i\dot{y}_k)(\dot{x}_k + i\dot{y}_k)\} = Re\{i\dot{x}_k^2 - \dot{x}_k \dot{y}_k + \dot{x}_k \dot{y}_k + i\dot{y}_k^2\} = 0
$$

$$
< i\dot{r}_k, i\dot{r}_k >= Re\{\overline{i\dot{r}_k}^T i\dot{r}_k\} = Re\{(-\ddot{x}_k - \ddot{y}_k)i(\dot{x}_k + i\dot{y}_k)\} = Re\{\ddot{x}_k \dot{x}_k + i\ddot{x}_k \dot{y}_k - i\ddot{y}_k \dot{x}_k + \ddot{y}_k \dot{y}_k\} = < \ddot{r}_k, \dot{r}_k >
$$

Finally

$$
\dot{S}(r, \theta) = \sum_{k=1}^{N} [< \dot{r}_k, i\dot{r}_k > - \omega_0 < i\dot{r}_k, i\dot{r}_k >](u_k - \omega_0) = \sum_{k=1}^{N} [0 - \omega_0 < \ddot{r}_k, \dot{r}_k >](u_k - \omega_0) = \sum_{k=1}^{N} [\omega_0 < \ddot{r}_k, \dot{r}_k >](\omega_0 - u_k)
$$
9.2.2 Uniform Distribution Control

In order to obtain the uniform distribution formation control law, we combine the previous circular control law with a potential control term:

\[ u_k = \omega_0 (1 + \kappa \langle \tilde{r}_k, \dot{r}_k \rangle) - \frac{\partial U}{\partial \theta_k} \]

The new Lyapunov function candidate is composed by two terms:

\[ V(r, \theta) = \kappa S(r, \theta) + U(\theta) \]

Differentiating \( \dot{V}(r, \theta) = \kappa \dot{S}(r, \theta) + \frac{\partial U}{\partial \theta} \dot{\theta} \):

\[ \sum_{k=1}^{N} \kappa \langle \omega_0 \tilde{r}_k, i \dot{r}_k \rangle (\omega_0 - u_k) + \frac{\partial U}{\partial \theta_k} \dot{\theta} = \]

\[ \sum_{k=1}^{N} (\kappa \langle \omega_0 \tilde{r}_k, i \dot{r}_k \rangle (\omega_0 - u_k) + \frac{\partial U}{\partial \theta_k} u_k) \]

Due to the property

\[ \langle \nabla U, 1 \rangle = 0 \]

\[ \sum_{k=1}^{N} \frac{\partial U}{\partial \theta_k} = 0 \]

\[ \omega_0 \sum_{k=1}^{N} \frac{\partial U}{\partial \theta_k} = 0 \]

we obtained

\[ \dot{V}(r, \theta) = \sum_{k=1}^{N} [\kappa \langle \omega_0 \tilde{r}_k, i \dot{r}_k \rangle (\omega_0 - u_k) + \frac{\partial U}{\partial \theta_k} u_k] - \omega_0 \sum_{k=1}^{N} \frac{\partial U}{\partial \theta_k} = \]

\[ \sum_{k=1}^{N} [\kappa \langle \omega_0 \tilde{r}_k, i \dot{r}_k \rangle (\omega_0 - u_k) + \frac{\partial U}{\partial \theta_k} (u_k - \omega_0)] = \]

\[ \sum_{k=1}^{N} [\kappa \langle \omega_0 \tilde{r}_k, i \dot{r}_k \rangle - \frac{\partial U}{\partial \theta_k}] (\omega_0 - u_k) \]

If we defined the uniform distribution control law as

\[ u_k = \omega_0 (1 + \kappa \langle \tilde{r}_k, \dot{r}_k \rangle) - \frac{\partial U}{\partial \theta_k} \]

Operating

\[ \dot{V}(r, \theta) = \sum_{k=1}^{N} [\kappa \langle \omega_0 \tilde{r}_k, i \dot{r}_k \rangle - \frac{\partial U}{\partial \theta_k}] (\omega_0 - (\omega_0 (1 + \kappa \langle \tilde{r}_k, \dot{r}_k \rangle) - \frac{\partial U}{\partial \theta_k})) = \]

Lara Briñón Arranz 39 Jun 2008
\[ \sum_{k=1}^{N} [\kappa < \omega_0 \hat{r}_k, i \hat{r}_k] - \frac{\partial U}{\partial \theta_k} (\omega_0 - \omega_0 - \kappa < \omega_0 \hat{r}_k, \hat{r}_k) + \frac{\partial U}{\partial \theta_k} = \]

\[ \sum_{k=1}^{N} [\kappa < \omega_0 \hat{r}_k, i \hat{r}_k] - \frac{\partial U}{\partial \theta_k} (-\kappa < \omega_0 \hat{r}_k, \hat{r}_k) + \frac{\partial U}{\partial \theta_k} = \]

\[ - \sum_{k=1}^{N} [\kappa < \omega_0 \hat{r}_k, i \hat{r}_k] - \frac{\partial U}{\partial \theta_k} ]^2 \]

Finally

\[ \dot{V}(r, \theta) = - \sum_{k=1}^{N} [\kappa < \omega_0 \hat{r}_k, i \hat{r}_k] - \frac{\partial U}{\partial \theta_k} ]^2 \leq 0 \]

### 9.3 Contraction circle control

We need to control \( \omega_0 \), therefore we define \( \omega_0 = \omega_d \). Then \( \check{\xi} \) depends on \( \omega_d \) now. The new Lyapunov function is:

\[ V(r, \theta, \omega_d) = \kappa S(r, \theta, \omega_d) + U(\theta) \]

and its derivative is expressed as follows:

\[ \dot{V}(r, \theta, \omega_d) = \kappa \dot{S}(r, \theta, \omega_d) + \frac{\partial U}{\partial \theta_k} \dot{\theta}_k \]

Differentiating the first term:

\[ \dot{S}(r, \theta, \omega_d) = \sum_{k=1}^{N} \langle f(r_k, \theta_k, \omega_d), \frac{\partial f(r_k, \theta_k, \omega_d)}{\partial r_k} \dot{r}_k \rangle + \]

\[ \sum_{k=1}^{N} \langle f(r_k, \theta_k, \omega_d), \frac{\partial f(r_k, \theta_k, \omega_d)}{\partial \theta_k} \dot{\theta}_k \rangle + \]

\[ \sum_{k=1}^{N} \langle f(r_k, \theta_k, \omega_d), \frac{\partial f(r_k, \theta_k, \omega_d)}{\partial \omega_d} \dot{\omega}_d \rangle \]

Partial derivatives:

\[ \frac{\partial f(r_k, \theta_k, \omega_d)}{\partial r_k} = -i \omega_d \]

\[ \frac{\partial f(r_k, \theta_k, \omega_d)}{\partial \theta_k} = i \nu e^{i \theta} = i \hat{r}_k \]

\[ \frac{\partial f(r_k, \theta_k, \omega_d)}{\partial \omega_d} = -i \hat{r}_k \]

Operating

\[ \dot{S}(r, \theta, \omega_d) = \sum_{k=1}^{N} \langle \dot{r}_k - i \omega_0 \hat{r}_k, i \omega_d \hat{r}_k \rangle + \]
Therefore the derivative of Lyapunov function can be expressed as:

\[
\dot{V}(\mathbf{r}, \theta, \omega_d) = \sum_{k=1}^{N} \left[ \kappa < \omega_d \mathbf{r}_k, \mathbf{r}_k > (\omega_d - u_k) + \frac{\partial U}{\partial \theta_k} u_k \right] + \kappa \dot{\omega}_d \sum_{k=1}^{N} (\dot{\mathbf{r}}_k, -i \mathbf{r}_k) + \omega_d |\mathbf{r}_k|^2
\]

with a control law as follows

\[
u_k = \omega_d (1 + \kappa < \mathbf{r}_k, \dot{\mathbf{r}}_k >) - \frac{\partial U}{\partial \theta_k} + a_k\]

we simplifies

\[
\dot{V}(\mathbf{r}, \theta, \omega_d) = \sum_{k=1}^{N} \left[ \kappa < \omega_d \mathbf{r}_k, i \dot{\mathbf{r}}_k > - \frac{\partial U}{\partial \theta_k} (\omega_d - (\omega_d(1 + \kappa < \mathbf{r}_k, \dot{\mathbf{r}}_k >) - \frac{\partial U}{\partial \theta_k} + a_k)\right] +
\]

\[
\kappa \dot{\omega}_d \sum_{k=1}^{N} (\dot{\mathbf{r}}_k, -i \mathbf{r}_k) + \omega_d |\mathbf{r}_k|^2
\]
\[
\kappa \omega_d \sum_{k=1}^{N} (\langle \dot{\mathbf{r}}_k, -i\mathbf{r}_k \rangle + \omega_d |\mathbf{r}_k|^2)
\]

We search to erase the two last terms:
\[
\sum_{k=1}^{N} (\kappa < \omega_d \mathbf{r}_k, i\mathbf{r}_k > - \frac{\partial U}{\partial \theta_k}) (-a_k) + \kappa \omega_d \sum_{k=1}^{N} (\langle \dot{\mathbf{r}}_k, -i\mathbf{r}_k \rangle + \omega_d |\mathbf{r}_k|^2) = 0
\]

Hence
\[
a_k = \frac{\kappa \omega_d \langle \mathbf{r}_k, -i\mathbf{r}_k \rangle + \omega_d |\mathbf{r}_k|^2}{\kappa < \omega_d \mathbf{r}_k, \mathbf{r}_k > + \frac{\partial U}{\partial \theta_k}}
\]

and the Lyapunov second theorem on stability is verified:
\[
\dot{V}(\mathbf{r}, \theta) = -\sum_{k=1}^{N} [\kappa < \omega_0 \mathbf{r}_k, i\mathbf{r}_k > - \frac{\partial U}{\partial \theta_k}]^2 \leq 0
\]

### 9.4 Gradient Search Control law

Equation for each agent:
\[
\begin{align*}
\dot{\mathbf{r}}_k &= ve^{i\theta_k} \\
\dot{\theta}_k &= \omega_d (1 + \kappa < \mathbf{r}_k, \dot{\mathbf{r}}_k >) - \frac{\partial U}{\partial \theta_k}
\end{align*}
\]
when the circle formation is stable \( u_k \to u_k^* \) then formation system can be defined by:
\[
\begin{align*}
\dot{\mathbf{r}}_k^* &= ve^{i\psi}e^{i\theta_k^*} \\
\dot{\psi} &= \omega_d
\end{align*}
\]

where \( \theta_k^* = 2\pi \frac{k-1}{N} \)

We define the new relative position variable \( \tilde{r}_k \) for each agent
\[
\tilde{r}_k = (\mathbf{r}_k + c_d)e^{-i\psi} = \mathbf{r}_k e^{-i\psi} + c_d e^{-i\psi}
\]
and for simplicity we consider \( \tilde{c}_d = c_d e^{-i\psi} \), then
\[
\tilde{r}_k = \mathbf{r}_k e^{-i\psi} + \tilde{c}_d
\]

its derivative is
\[
\dot{\tilde{r}}_k = \dot{\mathbf{r}}_k e^{-i\psi} - i\psi \mathbf{r}_k e^{-i\psi} + \dot{\tilde{c}}_d = ve^{i\theta_k} e^{-i\psi} - i\omega_d \mathbf{r}_k e^{-i\psi} + \dot{\tilde{c}}_d = ve^{-i\theta_k} - i\omega_d (\mathbf{r}_k - c_d) + \tilde{c}_d
\]
and the measurements of the agent \( k \) is expressed in function of the position of each vehicle
\[
\eta_k = G(\mathbf{r}_k)
\]
where the measurement is a real number $\eta_k \in \mathbb{R} \forall k$ and its derivative can be defined as follows:

$$\dot{\eta}_k(\hat{r}_k) = \frac{\partial \eta_k}{\partial \hat{r}_k} \cdot \hat{r}_k = <\frac{\partial \eta_k}{\partial \hat{r}_k}, v e^{i\theta_k} - \hat{c}_d - i \omega_d \hat{r}_k>$$

The Lyapunov function or objective function depends on the measurements because we search reach the maximum value for the concentration measured:

$$V(\eta) = \frac{1}{2} \sum_{k=1}^{N} \eta_k^2$$

Differentiating

$$\dot{V}(\eta) = \sum_{k=1}^{N} \eta_k \dot{\eta}_k$$

with the previous definition of $\dot{\eta}_k$, in such a way that it is possible to tackle the Lyapunov derivative:

$$\dot{V}(\eta) = \sum_{k=1}^{N} \eta_k <\frac{\partial \eta_k}{\partial \hat{r}_k}, v e^{i\theta_k} - \hat{c}_d - i \omega_d \hat{r}_k>$$

$$\dot{V}(\eta) = \sum_{k=1}^{N} \eta_k <\frac{\partial \eta_k}{\partial \hat{r}_k}, v e^{i\theta_k} > + \sum_{k=1}^{N} \eta_k <\frac{\partial \eta_k}{\partial \hat{r}_k}, -\hat{c}_d > + \omega_d \sum_{k=1}^{N} \eta_k <\frac{\partial \eta_k}{\partial \hat{r}_k}, -\hat{r}_k >$$

If we define

$$\omega_d = \sum_{k=1}^{N} \eta_k <\frac{\partial \eta_k}{\partial \hat{r}_k}, -\hat{r}_k >$$

a quadratic term is obtained in the Lyapunov function:

$$\dot{V}(\eta_k) = \sum_{k=1}^{N} \eta_k <\frac{\partial \eta_k}{\partial \hat{r}_k}, v e^{i\theta_k} > + \sum_{k=1}^{N} \eta_k <\frac{\partial \eta_k}{\partial \hat{r}_k}, -\hat{c}_d > + \sum_{k=1}^{N} \eta_k <\frac{\partial \eta_k}{\partial \hat{r}_k}, -\hat{r}_k >^2$$

Moreover we can separate the others terms in their real components for search a control equation for $\hat{c}_d$ as follows:

$$\sum_{k=1}^{N} \eta_k <\frac{\partial \eta_k}{\partial \hat{r}_k}, -\hat{c}_d > = - \sum_{k=1}^{N} \eta_k [\frac{\partial \eta_k}{\partial \hat{r}_k}]_x \hat{c}_{dx} + \frac{\partial \eta_k}{\partial \hat{r}_k} [y \hat{c}_{dy}]$$

Because $\hat{c}_d = \hat{c}_{dx} + i \hat{c}_{dy}$, it seems logical to define the two real components that they forms an other quadratic term:

$$\hat{c}_{dx} = - \sum_{k=1}^{N} \eta_k \frac{\partial \eta_k}{\partial \hat{r}_k} |_x$$

$$\hat{c}_{dy} = - \sum_{k=1}^{N} \eta_k \frac{\partial \eta_k}{\partial \hat{r}_k} |_y$$

Lara Briñón Arranz 43 Jun 2008
Although, the real components equations are going to be used for simulate, it is possible to find a complex expression:

\[ \dot{c}_d = - \sum_{k=1}^{N} \eta_k \frac{\partial \eta_k}{\partial r_k} \]

and the Lyapunov function obtains an other positive term

\[ \dot{V}(\eta_k) = \sum_{k=1}^{N} \eta_k < \frac{\partial \eta_k}{\partial r_k}, e^{i\theta_k} > + \sum_{k=1}^{N} \eta_k |\frac{\partial \eta_k}{\partial r_k}|^2 + \sum_{k=1}^{N} \eta_k < \frac{\partial \eta_k}{\partial r_k}, -\dot{r}_k >^2 \]

The complex equation permits achieving the change of coordinate system readily:

\[ \dot{c}_d = \dot{c}_k e^{-i\psi} \Rightarrow \dot{c}_d = \dot{c}_d e^{i\psi} \]

However it is necessary to define the concentration gradient for obtaining it in the coordinates of original system:

\[ \frac{\partial \eta_k}{\partial r_k} = \frac{\partial \eta_k}{\partial r_k} e^{i\psi} \]

with this definition the expression for the control of the centre:

\[ \dot{c}_d = -e^{i\psi} \sum_{k=1}^{N} \eta_k \frac{\partial \eta_k}{\partial r_k} e^{i\psi} = \sum_{k=1}^{N} \eta_k \frac{\partial \eta_k}{\partial r_k} e^{i2\psi} \]

\[ \dot{c}_{dx} = - \sum_{k=1}^{N} \eta_k [\frac{\partial \eta_k}{\partial r_k} |_x \cos 2\psi - \frac{\partial \eta_k}{\partial r_k} |_y \sin 2\psi] \] (30)

\[ \dot{c}_{dy} = - \sum_{k=1}^{N} \eta_k [\frac{\partial \eta_k}{\partial r_k} |_y \cos 2\psi + \frac{\partial \eta_k}{\partial r_k} |_x \sin 2\psi] \] (31)

Nevertheless there is a not quadratic term in the Lyapunov function. The influence of this third term is unknown.

\[ \dot{V}(\eta) = \left[ - \sum_{k=1}^{N} \eta_k \frac{\partial \eta_k}{\partial r_k} |_x \right]^2 + \left[ - \sum_{k=1}^{N} \eta_k \frac{\partial \eta_k}{\partial r_k} |_y \right]^2 + \left[ \sum_{k=1}^{N} \eta_k < \frac{\partial \eta_k}{\partial r_k}, -\dot{r}_k > \right]^2 + \sum_{k=1}^{N} \eta_k < \frac{\partial \eta_k}{\partial r_k}, e^{i\theta_k} > \]

9.5 Program code

In order to achieve the problem solution, we have carried out several simulations in Matlab and Simulink. The program code for the M-files is shown here below. Each simulation is composed for different steps: initial conditions, run simulink model, animation and relative headings graphs.

**initial conditions:** *init_conditions.m* file:
%condiciones iniciales
%inicializacion de variables
x=0; y=0; xpto=0; ypto=0; theta=0; x0=0; y0=0;

%numero de agentes
global n n=5

%velocidad de los agentes
global v v=1

%1/radio del circulo formado
global w0 w0=1

%constantes de los controladores
global kappa
kappa=1
global beta
beta=.1

%coordenadas del centro
global c1 c2 c1=1 c2=1

%condiciones iniciales para cada agente
for j=1:n
  x0(j)=random('norm',1,1);
  y0(j)=random('norm',1,1);
end

x0=[1.3 -2.1 -.8 2.5 1.4]
y0=[-2.1 .5 2.4 -1.4 2.5]

**simulink model:**  *bucle_agentes.m* file:

```
function bucle_agentes2(block)
  setup(block);
  function setup(block) global n
  % Registre number of input and output ports
  block.NumInputPorts = 1;
  block.NumOutputPorts = 2;
```
% Setup functional port properties to dynamically inherited.
block.SetPreCompInpPortInfoToDynamic;
block.SetPreCompOutPortInfoToDynamic;

    block.InputPort(1).Complexity = 'Real';
    block.InputPort(1).DataTypeId = 0;
    block.InputPort(1).SamplingMode = 'Sample';
    block.InputPort(1).Dimensions = n;

    block.OutputPort(1).Complexity = 'Real';
    block.OutputPort(1).DataTypeId = 0;
    block.OutputPort(1).SamplingMode = 'Sample';
    block.OutputPort(1).Dimensions = n;

    block.OutputPort(2).Complexity = 'Real';
    block.OutputPort(2).DataTypeId = 0;
    block.OutputPort(2).SamplingMode = 'Sample';
    block.OutputPort(2).Dimensions = n;

% Register methods
block.RegBlockMethod('ProcessParameters', @ProcessPrms);
block.RegBlockMethod('PostPropagationSetup', @DoPostPropSetup);
block.RegBlockMethod('Start', @Start);
block.RegBlockMethod('WriteRTW', @WriteRTW);
block.RegBlockMethod('Outputs', @Outputs);

% Block runs on TLC in accelerator mode.
block.SetAccelRunOnTLC(true);

function DoPostPropSetup(block)
function ProcessPrms(block)
block.AutoUpdateRuntimePrms;
function Start(block)
function Outputs(block)
global n v
salida1=zeros(1,n);
salida2=zeros(1,n);
\begin{verbatim}
theta = block.InputPort(1).Data; 
for k=1:n 
salida1(k) = v*cos(theta(k)); 
salida2(k) = v*sin(theta(k)); 
end 
block.OutputPort(1).Data = salida1; 
block.OutputPort(2).Data = salida2;
\end{verbatim}

control\_agentes.m file:

\begin{verbatim}
function control\_agentes2(block) 
setup(block);

function setup(block) 
global n 
% Registe number of input and output ports 
block.NumInputPorts = 5; 
block.NumOutputPorts = 1; 

% Setup functional port properties to dynamically inherited. 
block.SetPreCompInpPortInfoToDynamic; 
block.SetPreCompOutPortInfoToDynamic; 

block.InputPort(1).Complexity = 'Real'; 
block.InputPort(1).DataTypeId = 0; 
block.InputPort(1).SamplingMode = 'Sample'; 
block.InputPort(1).Dimensions = n; 

block.InputPort(2).Complexity = 'Real'; 
block.InputPort(2).DataTypeId = 0; 
block.InputPort(2).SamplingMode = 'Sample'; 
block.InputPort(2).Dimensions = n; 

block.InputPort(3).Complexity = 'Real'; 
block.InputPort(3).DataTypeId = 0; 
block.InputPort(3).SamplingMode = 'Sample'; 
block.InputPort(3).Dimensions = n;
\end{verbatim}
block.InputPort(4).Complexity = 'Real';
block.InputPort(4).DataTypeId = 0;
block.InputPort(4).SamplingMode = 'Sample';
block.InputPort(4).Dimensions = n;

block.InputPort(5).Complexity = 'Real';
block.InputPort(5).DataTypeId = 0;
block.InputPort(5).SamplingMode = 'Sample';
block.InputPort(5).Dimensions = n;

block.OutputPort(1).Complexity = 'Real';
block.OutputPort(1).DataTypeId = 0;
block.OutputPort(1).SamplingMode = 'Sample';
block.OutputPort(1).Dimensions = n;

% Register methods
block.RegBlockMethod('ProcessParameters', @ProcessPrms);
block.RegBlockMethod('PostPropagationSetup', @DoPostPropSetup);
block.RegBlockMethod('Start', @Start);
block.RegBlockMethod('WriteRTW', @WriteRTW);
block.RegBlockMethod('Outputs', @Outputs);

% Block runs on TLC in accelerator mode.
block.SetAccelRunOnTLC(true);

function DoPostPropSetup(block)
function ProcessPrms(block)
block.AutoUpdateRuntimePrms;

function Start(block)
function Outputs(block)
global n kappa beta w0 c1 c2
% constantes
s1=zeros(n);
s1x=zeros(n);
s1y=zeros(n);
s2=zeros(n);
u1=0;
u2=0;
uk=0;

%entradas
x=block.InputPort(1).Data;
y=block.InputPort(2).Data;
xpto=block.InputPort(3).Data;
ypto=block.InputPort(4).Data;
theta=block.InputPort(5).Data;

%bucle de control para cada agente k
for k=1:n
for j=1:n
for m=1:round(n/2)

%control relative headings
s2(k) = s2(k) + sin(m*(theta(k)-theta(j)))/m;
end
end

s1(k) = (x(k)-c1)*xpto(k) + (y(k)-c2)*ypto(k);
u1(k) = w0*(1 + kappa*s1(k));
u2(k) = beta*s2(k)/n;
uk(k) = u1(k) + u2(k);
end

%salida
block.OutputPort(1).Data = uk;

contra ction_control.m file:

function contraction_control(block)
setup(block);

function setup(block)
global n
% Registe number of input and output ports
block.NumInputPorts = 7;
block.NumOutputPorts = 1;

% Setup functional port properties to dynamically inherited.
block.SetPreCompInpPortInfoToDynamic;
block.SetPreCompOutPortInfoToDynamic;

block.InputPort(1).Complexity = 'Real';
block.InputPort(1).DataTypeId = 0;
block.InputPort(1).SamplingMode = 'Sample';
block.InputPort(1).Dimensions = n;

block.InputPort(2).Complexity = 'Real';
block.InputPort(2).DataTypeId = 0;
block.InputPort(2).SamplingMode = 'Sample';
block.InputPort(2).Dimensions = n;

block.InputPort(3).Complexity = 'Real';
block.InputPort(3).DataTypeId = 0;
block.InputPort(3).SamplingMode = 'Sample';
block.InputPort(3).Dimensions = n;

block.InputPort(4).Complexity = 'Real';
block.InputPort(4).DataTypeId = 0;
block.InputPort(4).SamplingMode = 'Sample';
block.InputPort(4).Dimensions = n;

block.InputPort(5).Complexity = 'Real';
block.InputPort(5).DataTypeId = 0;
block.InputPort(5).SamplingMode = 'Sample';
block.InputPort(5).Dimensions = n;

block.InputPort(6).Complexity = 'Real';
block.InputPort(6).DataTypeId = 0;
block.InputPort(6).SamplingMode = 'Sample';
block.InputPort(6).Dimensions = n;

block.InputPort(7).Complexity = 'Real';
block.InputPort(7).DataTypeId = 0;
block.InputPort(7).SamplingMode = 'Sample';
block.InputPort(7).Dimensions = n;

block.OutputPort(1).Complexity = 'Real';
block.OutputPort(1).DataTypeId = 0;
block.OutputPort(1).SamplingMode = 'Sample';
block.OutputPort(1).Dimensions = n;

% Register methods
block.RegBlockMethod('ProcessParameters', @ProcessPrms);
block.RegBlockMethod('PostPropagationSetup', @DoPostPropSetup);
block.RegBlockMethod('Start', @Start);
block.RegBlockMethod('WriteRTW', @WriteRTW);
block.RegBlockMethod('Outputs', @Outputs);

% Block runs on TLC in accelerator mode.
block.SetAccelRunOnTLC(true);

function DoPostPropSetup(block)
function ProcessPrms(block)
block.AutoUpdateRuntimePrms;
function Start(block)
function Outputs(block)
global n kappa beta c1 c2
% constantes
s1=zeros(n,1);
s2=zeros(n,1);
u1=0;
u2=0;
den=0;
num=0;
kapa2=.1;
cuadrado=0;
termino=0;

% entradas
wd=block.InputPort(1).Data;
wdpto=block.InputPort(2).Data;
x=block.InputPort(3).Data;
y=block.InputPort(4).Data;
xpto=block.InputPort(5).Data;
ypto=block.InputPort(6).Data;
theta=block.InputPort(7).Data;
% bucle de control de la contraccion para cada agente k
for k=1:n
  for j=1:n
    for m=1:round(n/2)
      % control relative headings
      s2(k) = s2(k) + sin(m*(theta(k)-theta(j)))/m;
    end
  end

  s1(k) = (x(k)-c1)*xpto(k) + (y(k)-c2)*ypto(k);
  u1(k) = wd*kappa*s1(k);
  u2(k) = beta*s2(k)/n;

  % denominador de la funcion de control
  den(k) = u1(k) + u2(k);

  cuadrado(k) = wd*((x(k)-c1)^2 + (y(k)-c2)^2);
  termino(k) = (y(k)-c2)*xpto(k) - (x(k)-c1)*ypto(k) + cuadrado(k);

  % numerador de la funcion de control
  num(k) = kappa*wdpto*termino(k);

  % ley de control para la contraccion
  uk2(k)= num(k) / den(k);
end

% salida
block.OutputPort(1).Data = uk2;

animation animation_formations.m file:
% simulation collective motion: control cercle

\[ m, n \] = \text{size}(x);
modulo=0;

\[ p1=1; \] if ypto(1,1)==0
\[ p1=0; \]
\[ p2=1; \] else
\[ p2=p1*xpto(1,1)/ypto(1,1); \] end
\[ \text{norma} = \sqrt{p1^2+p2^2}; \]
\[ p1=p1/\text{norma}; \]
\[ p2=p2/\text{norma}; \]
\[ a0=\begin{bmatrix} 0;-.1 \end{bmatrix}; \]
\[ b0=\begin{bmatrix} .2;0 \end{bmatrix}; \]
\[ c0=\begin{bmatrix} 0;.1 \end{bmatrix}; \]

figure

for j=1:m-1

plot(1,1,'*')
hold on
%
rotacion + traslacion
for k=1:n
\[ \text{modulo}(k)=\sqrt{xpto(j,k)^2+ypto(j,k)^2}; \]
\[ a = \begin{bmatrix} xpto(j,k)/\text{modulo}(k) - ypto(j,k)/\text{modulo}(k) \\ ypto(j,k)/\text{modulo}(k) xpto(j,k)/\text{modulo}(k) \end{bmatrix} * \]
\[ a0 + [x(j,k); y(j,k)]; \]
\[ b = \begin{bmatrix} xpto(j,k)/\text{modulo}(k) - ypto(j,k)/\text{modulo}(k) \\ ypto(j,k)/\text{modulo}(k) xpto(j,k)/\text{modulo}(k) \end{bmatrix} * \]
\[ b0 + [x(j,k); y(j,k)]; \]
\[ c = \begin{bmatrix} xpto(j,k)/\text{modulo}(k) - ypto(j,k)/\text{modulo}(k) \\ ypto(j,k)/\text{modulo}(k) xpto(j,k)/\text{modulo}(k) \end{bmatrix} * \]
\[ c0 + [x(j,k); y(j,k)]; \]

coordx = \begin{bmatrix} b(1)c(1)a(1)b(1) \end{bmatrix};
coordy = \begin{bmatrix} b(2)c(2)a(2)b(2) \end{bmatrix};

plot(x(j,k), y(j,k), 'k.'); axis([-1.53.5, -23.5])
hold on
patch(coordx, coordy, 'k'), axis([-1.53.5, -23.5]);
title('Formation du cercle: centre en (c1,c2)')
end
hold off
F(j)=getframe;
end

relative heading graph  relative_heading.m file:

[m,n]=size(x);
k=1;
t=linspace(0,100,m);

figure
subplot(4,1,1)
plot(t,theta(:,k)-theta(:,2))
ylabel('theta1-theta2')
title('Relative headings')
hold on
subplot(4,1,2)
plot(t,theta(:,k)-theta(:,3))
ylabel('theta1-theta3')
subplot(4,1,3)
plot(t,theta(:,k)-theta(:,4))
ylabel('theta1-theta4')
subplot(4,1,4)
plot(t,theta(:,k)-theta(:,5))
xlabel('time (s)')
ylabel('theta1-theta5')

figure
subplot(4,1,1)
plot(t,thetapto(:,k)-thetapto(:,2))
ylabel('Vrel1-Vrel2')
title('Relative angular velocity')
hold on
subplot(4,1,2)
plot(t,thetapto(:,k)-thetapto(:,3))
ylabel(‘Vrel1-Vrel3’)
subplot(4,1,3)
plot(t,thetapto(:,k)-thetapto(:,4))
ylabel(‘Vrel1-Vrel4’)
subplot(4,1,4)
plot(t,thetapto(:,k)-thetapto(:,5))
xlabel(‘time (s)’)
ylabel(‘Vrel1-Vrel5’)

9.6 Simulink Diagrams

Figure 11: Formation control model in Simulink
Figure 12: Translation control model in Simulink

Figure 13: Contraction control model in Simulink

10 Attached Document B: Review of the state of art

NOTES ON MULTI-AGENT CONTROL UNDER COMMUNICATIONS CONSTRAINTS

This note is devoted to the review of the state of the art on the control design of multi-agent systems subject to communication constraints. This work is carried out within the framework of the ANR-CNRS CONNECT projet. More information on the project objectives can be found...
10.1 Introduction

Control community had formerly focused mainly on control of vehicle formations. However, distributed motion and cooperation systems have emerged as topics of significant interest and matters of concern to the control theory and robotics specialists over the past few years. This report review various methods and control approaches dealing with multi-agents systems and formation control. The report is completed also with a short review on some of the most commonly used models for control design and analysis in this context. At the end we also review gradient search methods that will be of use when using the fleet formation for source searching.

10.2 Models

Simple linear state space models are in some cases used to represent the agent’s dynamics [1]. Nevertheless, nonlinear massless models based on the velocity kinematics (and/or the accelerations) are better adapted to describe the motions of AUV’s. They can be formulated in three-dimensional space, but for seek of simplicity some works are carried out in the plane (two-dimensional models) [1, 2].

Models can also be discretized in tome, but in this review we limited to models that are described in continuous time. As the time scales of the AUV’s motion is rather slow, the time discretization is not generally an issue here.

**Kinematic models** Many of the studies carried out in the field of control of multi-agent control are based on simplified models. Some studies deal with simple point masses in the plane,

\[ \dot{x}_i = u_i \]

where each \( x_i \) describes the agent coordinate vector, and \( u_i \), the associated control input. In many cases, the studies limited to motion in the plane with \((x_i, u_i) \in \mathbb{R}^2\). A more general models formulate linear models, of the standard form:

\[ \dot{x}_i = A_i x_i + B_i u_i \]
In the applications concerning mobile robots and also underwater robots, it seems to be customary to model each vehicle from its kinematic equations, that in certain cases, see [3], can be simplified to point mass motion subject to planar steering control, i.e.

\[
\dot{r}_k = ve^{i\theta_k} \tag{32}
\]

\[
\dot{\theta}_k = u_k \tag{33}
\]

where \( r = x_k + iy_k \in \mathbb{R}^2 \) and \( \theta_k \in S^1 \) are the position and heading of each vehicle, \( v \) is the vehicle velocity that often is normalized to one, \( v = 1 \). The particular reason to make this assumption, in the context of underwater vehicles is due to the fact that underwater vehicles with a single propel, like the one considered in the CONNECT project, are energy-efficiently operated under constant velocity motion.

Equation (33) is not fundamental, it only asses the fact that the control input is related to the velocity steering rather than to the steering angle directly. It is however possible to simply further the model by assuming that heading time-scale, is at least of an order of magnitude faster than the position time scale. This assumption is justified in practice, as vehicle masses, added masses forces, fluid friction contact areas, are larger for the vehicle body motion, than for the lateral flaps controlling the vehicle heading. This makes the vehicle body dynamics (32) much more slower. Therefore, it makes sense to consider the steering angle \( \theta_k \) directly the control input, with \( v = 1 \)

\[
\dot{r}_k = e^{iu_k} \tag{34}
\]

\[
|u_k| \leq u^M = \theta_{\text{max}} \tag{35}
\]

where \( u^M = \theta_{\text{max}} \) describes the maximum possible turning angle for the considered vehicle. The constraint (35) is important as it can be used to model different types of vehicles.

An alternative, and equivalent representation to (34)-(35) is

\[
\dot{x}_k = \cos(u_k) \tag{36}
\]

\[
\dot{y}_k = \sin(u_k) \tag{37}
\]

\[
|u_k| \leq u^M = \theta_{\text{max}} \tag{38}
\]

but the compact description (34) – (35) is often preferred.

**Dynamics models**  An example of dynamic model in the plane is [1],

\[
\dot{x}_{1k} = x_{2k} \tag{36}
\]

\[
\dot{x}_{2k} = f_k \cos(u_k) - 2x_{2k} \tag{37}
\]

\[
\dot{y}_{1k} = y_{2k} \tag{38}
\]

\[
\dot{y}_{2k} = f_k \sin(u_k) - 2y_{2k} \tag{39}
\]
where $f_k$ represents the input force given by the vehicle’s thruster, $u_k$ is the angle of the vehicle or heading, and the states $x_k$ and $y_k$ represent the position of the vehicle in the plane. The model can be completed with the linear damping $-2x_k$ and $-2y_k$ associated to each direction of motion.

**Group models.** As the agents are controlled in a coordinated fashion, rather than independent each other, it become interesting to introduce other variables of interest that better describe the group motion. For instance, in [3] they introduce the center of mass, $\mathbf{R}$,

$$\mathbf{R} = \frac{1}{N} \sum_{j=1}^{N} \mathbf{r}_k$$

and the velocity of the center of mass of the group $\mathbf{p} = \dot{\mathbf{R}}$.

$$\mathbf{p} = \frac{1}{N} \sum_{k=1}^{N} e^{iu_k} = \frac{1}{N} \sum_{k=1}^{N} \dot{\mathbf{r}}_k = \dot{\mathbf{R}}$$

However, these appellations may be not exact as motion of the body in water are not subject to pure gravity forces, but also influenced by buoyancy, and by fluid nonlinear damping. It may be of interest to consider also other possible points of interest better adapted to motion in fluid environments.

As a curiosity, we can think in the Barycenter; a point in the space around which two bodies are orbiting around their common center of mass as if they were one object.

Other quantities of interest are the potential function $U$, defined as

$$U(u) = \frac{N}{2} |\mathbf{p}|^2$$

which reflect a kind of “kinetic energy”. The gradient of this function $\frac{\partial U}{\partial u_k} = \langle ie^{iu_k}, \mathbf{p} \rangle = \Re\{ie^{iu_k} \mathbf{p}\}$ is often used for control design\(^1\).

### 10.3 Formation Control

In this section we review some control strategies assuming all-to-all communications assumptions. The case where the communication information is restricted, will be treated in subsequent sections. Note also that these controller does not capture any particular feature for source gradient search. This will also be reviewed in latter sections.

#### 10.3.1 Lyapunov-based control design

The paper [3] resume several control options based on potential and Lyapunov functions. We summarize some of these results here below.

\(^1\)The inner product for complex numbers is denoted here as $\langle x_1, x_2 \rangle = \Re\{(x_1^\top)^T x_2\}$, being $x_1$, and $x_2$ elements of the real vector space $\mathbb{C}^N$, which is isomorphic to $\mathbb{R}^{2N}$.
Circular motions & relative headings  To stabilize circular motions of the group about its center of mass, and also to stabilize a particular arrangement (phase-locked patterns) of the vehicles in their circular formation, the following control law has been proposed in [3],

\[
\dot{u}_k = \omega_0 (1 + \kappa \langle \tilde{r}_k, \dot{r}_k \rangle) - \frac{\partial U}{\partial u_k}
\]

where, \( \tilde{r}_k = r_k - R \) describes the distance between each vehicle \( k \) to the center of mass, i.e.

\[
\tilde{r}_k = r_k - R = \frac{1}{N} \sum_{j=1}^{N} r_{kj}
\]

\( r_{kj} \) being the relative position \( r_{kj} = r_k - r_j \).

Figure 14: The six possible different symmetric patterns for \( N = 12 \) corresponding to \( M = 1; 2; 3; 4; 6; \) and 12. The top left is the synchronized state and the bottom right is the splay state. The number of collocated headings is illustrated by the width of the black annulus denoting each phase cluster. Figure extracted from [3].

The control law allows the stabilization to a circle with a radius \( \rho_0 = |\omega_0|^{-1} \) with the rotation direction of \( \omega_0 \). By the second term, this control law allows the stabilization of symmetric \((M - N)\)-patterns characterized by \( 2 \leq M \leq N \) heading clusters separated by a multiple of \( \frac{2\pi}{M} \). To this to happen, an additional requirement is that the potential \( U(u) \) be invariant to rigid rotations, i.e. \( \langle \nabla U, \tilde{1} \rangle = \langle iP, p \rangle = 0 \).

Stability and convergence to the circular motions, are show by means of the Lyapunov function \( V \), that combines the function \( S \)

\[
S(r, u) = \frac{1}{2} \sum_{k=1}^{N} |e^{i\omega_0 t_k} - i\omega_0 \tilde{r}_k|^2
\]

which has a minimum zero for circular motions around the center of mass with radius \( \rho_0 = 1/\omega_0 \), with the potential function \( U \), i.e.

\[
V(r, u) = \kappa S(r, u) + U(u)
\]

\footnote{This control law has been derived in connection with the kinematic model (32)-(33), but we have rewritten this law to accommodate the simplified model (34)-(35)}
Note that as the control law $u$ is dynamical, the variable $u_k$ can be interpreted as an internal control state, and hence need to be included in the Lyapunov function $V$.

The control law (40) does not necessarily allows of a evenly spaced location of the vehicles. For instance, it is possible to reach equilibrium so that the vehicles superimposes. To avoid this, authors [3], and [4], proposed a new form of potential including “high-order harmonics”, i.e.

$$U_m(u) = \frac{N}{2} |\mathbf{p}_m|^2, \quad \mathbf{p}_m = \frac{1}{mN} \sum_{k=1}^{N} e^{imu_k}$$

with $\langle \nabla U_m, \mathbf{1} \rangle = 0$

![Figure 15](image)

*Figure 15: Numerical simulation of the splay state formation starting from random initial conditions. Each vehicle and its velocity is illustrated by a black circle and an arrow. Note that the center of mass of the group, illustrated by a crossed circle, is fixed at steady-state. Figure extracted of [3].*

There is a couple of extensions to this approach that are worth to mention. These are done by introducing the suitable variables (reflecting the desired objective), and then redefining the potential function $U$, and perhaps also the function $S$ as a function of these new error coordinates.

**Coordinated subgroups.** The extension to the coordination of sub-groups allows to design control laws to coordinate vehicles in sub-groups using *block all-to-all* interconnections. The idea can be of interests, if we decide that we should split the vehicle fleet motion into a subsets of smaller subgroups. Or if it is interesting to split the gradient source search in 3D, into cuts of subgroups at different deeps.

**Shape control: Elliptical Beacon control laws.** Other possible extension is to modify the circular motions and to stabilize a single vehicle on an elliptical trajectory about a fixed beacon. In addition, it was also shown by the same authors that it is possible to couple several vehicles via their relative heading in order to synchronize the vehicle phase of each ellipse.
10.3.2 Laplacian-based consensus algorithms

Laplacian-based consensus control algorithms are the control counterpart to the Gossip-based algorithms in computer sciences, that we review first before introduce the control discussion.

Gossip protocols. A gossip protocol is a protocol designed to mimic the way that information spreads when people gossip about some fact.

For example, imagine a group of office workers. Person A comments to person B that he believes person C is starting to dye his mustache. B tells D, while A repeats the news to E. Notice that as people move about and share the news, the number of individuals who know about the rumor doubles "round by round". In computer systems, it is common to implement such type of protocols with some form of random "peer selection": at some wired-in frequency,
each machine picks some other machine at random and shares any hot rumors. Expressing these ideas in more technical terms, a gossip protocol is one that satisfies the following conditions:

- The core of the protocol involves periodic, pairwise, inter-process interaction
- The information exchanged during these interactions is of (small) bounded size,
- When agents interact, the state of one or both changes in a way that reflects the state of the other. For example, if A pings B just to measure the round-trip time for messages from A to B and back, it isn’t a gossip interaction.
- Reliable communication is not assumed
- The frequency of the interactions is low compared to typical message latencies, so that the protocol costs are negligible
- There is some form of randomness in the peer selection. Peer selection might occur within the full node set, or might be performed in a smaller set of neighbors.

Inspired by these ideas Laplacian-based consensus control algorithms has been derived to deal with the control of multi-agent systems with limited information. The discussion what follows is based in the complete review recently provided in the paper [5], which is based on the five-key papers [6, 7, 8, 9, 10]. The following is an extract of Section A. from [5]

**Consensus in Networks.** The interaction topology of a network of agents is represented using a directed graph $G = (V, E)$ with the set of nodes $V = \{1, 2, \ldots, n\}$ and edges $E \subseteq V \times V$. The neighbors of agent $i$ are denoted by

$$N_i = \{j \in V : (i, j) \in E\}$$

![Figure 18](image-url)

Figure 18: Form of consensus algorithms: a network of integrator agents in which agent $i$ receives the state $x_j$ of its neighbor, agent $j$, if there is a link $(i, j)$ connecting the two nodes. Figure extracted of [6].
This set describes the information that is accessible to the agent $i$, either all the time (fixe network topology), or at particular time instant (variable network topology). Assuming now that the control law for each agent is designed on the basis of limited information, i.e.

$$u_i = \sum_{j \in N_i} (x_j(t) - x_i(t))$$

A simple consensus algorithm to reach an agreement regarding the state of $n$ integrator agents with dynamics $\dot{x}_i = u_i$ can be then expressed as an $n$th-order linear system on a graph

$$\dot{x}_i = \sum_{j \in N_i} (x_j(t) - x_i(t))$$

(41)

hence, the collective dynamics of the group of agents can be compactly written as

$$\dot{x} = Lx$$

where $L$ is graph Laplacian of the network and its elements are defined as follows:

$$l_{ij} = \begin{cases} -1 & j \in N_i \\ \frac{1}{|N_i|} & j = i \end{cases}$$

where $|N_i|$ denotes the number of neighbors of node $i$ (or out-degree of node $i$). With this control law, the authors in [5] had shown that the states converge to the same equilibrium, $x \to x^*$, where

$$x^* = \alpha 1_n, \quad \alpha = \frac{1}{n} \sum_i x_i(0)$$

which shows that all states (nodes) agrees (consent). It is also shown that $x^*$ is an unique equilibrium as long as the graph is connected.

**Distributed Formation Control**. The above controller is not able to impose a particular formation. This can be reached by using vectors of relative positions of neighboring vehicles, and then using consensus based-controllers with input bias [7]. For this, the authors considered the problem of minimizing locally the cost function

$$U(x) = \sum_{j \in N_i} ||x_j - x_i - r_{ij}||^2$$

via a distributed algorithm, where $r_{ij}$ is the desired inter-vehicle relative position vector.

If the agents use the gradient decent algorithm to minimize the cost $U(x)$, this leads to

$$\dot{x}_i = \sum_{j \in N_i} (x_j(t) - x_i(t) - r_{ij}) = \sum_{j \in N_i} (x_j(t) - x_i(t)) + b_i$$

(42)

with input bias $b_i = -\sum_{j \in N_i} r_{ij}$. This is equivalent to the consensus problem mentioned before, with the bias term added. Although this bias does not play any role in the stability of the system, it helps to modify the equilibrium of the state. In that way a particular formation can be obtained by modifying each of the components of $r_{ij}$.
Switched Networks In some scenarios, network systems may have a dynamic (time-varying) network topology, due to node and link failures/creations.

The so-called switching networks may be modeled with the help of parametrized dynamic graphs $G_{s(t)}$ parametrized with a switched signal $s(t)$. The consensus mechanism on a network with a variable topology becomes a linear switching system,

$$\dot{x} = -L(G_{s(t)})x,$$

under certain conditions (balance digraph network topology), an average consensus can be found.
10.3.3 Formation control with collision avoidance

Flocking is a form of collective behavior of large number of interacting agents with a common group objective. This kind of agents can be found in diverse disciplines as biophysics or social sciences and the flocking algorithms are inspired in the clouding animal behavior.

The three rules of Reynolds generate the main principles for the flocking method control. This rules, known as cohesion, separation and alignment rules, are gathered in [11]:

- **Flock Centering**: attempt to stay close to nearby flockmates;
- **Collision Avoidance**: avoid collision with nearby flockmates;
- **Velocity Matching**: attempt to match velocity whit nearby flockmates.

Following theses rules, the gradient-based algorithm equipped with velocity consensus protocol and flocking with obstacle avoidance [12] achieve methods for formation control.

A graph model is used to tackle the flocking problem. Thus each node represents one agent, called α-agent, and the set of neighbors for each α-agent is defined as an open ball with radius r [12]:

\[
N_i = \{ j \in V : \| q_j - q_i \| < r \}
\]

where \( \| \cdot \| \) is the Euclidean norm in \( \mathbb{R}^m \), and \( q_i \) describes the Euclidean generalized coordinated of agent \( i \).

![Figure 21: Agent and its neighbors in a spherical neighborhood. Figure extracted of [12].](image)

Distributed algorithms for flocking in free-space, or free-flocking, use α-agents to represent each agent in the network system. The flocking method deals with each α-agent form an α-lattice with its neighbors.

We introduce virtual agents called β-agents and γ-agents which model the effect of obstacles and collective objective of a group, respectively [12].
In free-flocking each $\alpha$-agent applies a control input than consists of three terms:

$$u_i = f^g_i + f^d_i + f^\gamma_i$$

where $f^g_i = -\nabla q_i V(q)$ is a gradient-based term, $f^d_i$ is a *velocity consensus term* that acts as a damping force, and $f^\gamma_i$ is a navigational feedback due to group objective.

**Flocking with obstacle avoidance.** The $\beta$-agents have been defined previously due to the necessity to avoid the obstacles. We use agent-based representation of all nearby obstacles; therefore, we defined the set of $\beta$-neighbors of an $\alpha$-agent as follows:

$$N^\beta_i = \{k \in V^\beta : \| q_{i,k} - q_i \| < r \}$$

where $r > 0$ is *interaction range* of an $\alpha$-agent with neighboring $\beta$-agents.

![Agent-based representation of obstacles. Figure extracted of [12].](image)

To achieve flocking in presence of obstacles, in [12] the authors use the following multi-species *collective potential function* for the particle system:

$$V(q) = c^\alpha_1 V^\alpha(q) + c^\beta_1 V^\beta(q) + c^\gamma_1 V^\gamma(q)$$

where the $c^\alpha_1$, $c^\beta_1$, $c^\gamma_1$ are positive constants and $V^\alpha(q)$, $V^\beta(q)$, $V^\gamma(q)$ interaction potentials. The main flocking algorithm capable to avoid the obstacles consists of three terms:

$$u_i = u^\alpha_i + u^\beta_i + u^\gamma_i$$

where each term $u^\alpha_i$, $u^\beta_i$ denote the $\alpha$-neighbors interaction, $\beta$-neighbors interaction respectively, and $u^\gamma_i$ is a distributed navigational feedback [12].

Each terms is based in the interaction potentials $V^\alpha(q)$, $V^\beta(q)$, $V^\gamma(q)$ used before, and they attempt to relate the control functions with *attractive/repulsive* pairwise potential [13].
10.4 Source Seeking control

There are several works dealing with the problem of source seeking in different scenarios. They rate from the source seeking for signal fields with are static (light sources, magnetic fields) with respect to the target emitting them [14], to the cases where signals are diffusive [15]. In this latter case the source signal is governed by a diffusion equation of the form,

$$\frac{\partial C}{\partial t} = \frac{\partial^2 C}{\partial^2 x} + \frac{\partial^2 C}{\partial^2 y} + \delta(x - x_g, y - y_g)$$

where $C = C(x, y, t)$ is the concentration at time $t$, and at the point $(x, y)$, and $(x - x_g, y - y_g)$ is the position of the source.

Application concerning plume tracking and source seeking using multiple sensors on a single agent, or multiple agents communicating between each other are reporting in works [16, 17, 18, 19].

10.4.1 Single agent with a single sensor

Works in [14], and [15] concern a single agent modeled as a pointless mass similar to the model (34) – (35). In particular [15] treat the case of diffusive sources and show how this source tracking problem can be tackled with a simple control law of the form

$$\dot{u} = a\omega \cos(\omega t) + b \frac{s}{s + h}[C] \sin(\omega t)$$

where $a, b, h$ are constants, the $C$ is the concentration measurements, and $\frac{s}{s + h}$ is a high-pass filter.

The first term in the control law is used to produce an excitation such as to always collect sufficient rich concentration information, while the second term is used to direct the vehicle heading towards the minimum (or maximum) source concentration. Stability and convergence toward the source focus are shown in [14].
Figure 24: Sequence of images from a vehicle tracking a diffusive signal. The top row shows the target has created a curving footprint which the vehicle (arrow) follows successfully. The middle row of images picture a source by itself without a following vehicle moving along and releasing a diffusive contaminant. The bottom row of images show the reduced footprint resulting from a vehicle tracking the source and cleaning as it moves along. Figure extracted of [15].

Figure 25: Trajectory of vehicle with coupled center and sensor dynamics. Figure extracted of [15].

Note that this controller does not use any particular information on the function $c(t, x, y)$, but only uses the sensor reading of such information. The control can also be modified to track specific isoclines of the source concentration.
10.4.2 Source seek using multiple sensors

Contour-shape formations. In [19] general curve evolutionary theory has been used to the decentralized control of contour-shape formations of underwater vehicles commissioned to adapt to a certain level set of the source concentration. Several interesting ideas are here proposed. In particular the concept of flexible (or deformable) shape formations, where the shape of the formation is dictated by the environment. The forces used for that purpose need to be in adequation to those needed to preserve the formation.

Adaptive gradient climbing. The authors in [17] present a stable control strategy for groups of vehicles to move and reconfigure cooperatively in response to a sensed, distributed environment. The underlying coordination framework uses virtual bodies and artificial potentials. This work address the problem of a gradient climbing missions in which the mobile sensor network seeks out local maxima or minima in the environmental field.

To accomplish the decoupling of the formation stabilization problem from the overall performance of the network mission, the authors introduce to the group a virtual body. The virtual body is a collection of linked, moving reference points. The vehicle group moves (and reconfigures) with the virtual body by means of forces that derive from artificial potentials between the vehicles and the reference points on the virtual body. The virtual body can translate and rotate in three-dimensional space, expand and contract. The dynamics of the virtual body are designed in two steps. In one step, the speed of the virtual body is regulated using a feedback formation error function to ensure stability and convergence properties of the formation. In the other step, the direction of motion of the virtual body is prescribed so as to accomplish the desired mission, e.g., adaptive gradient climbing in a distributed environment.

For the gradient climbing tasks, the gradient of the measured field is approximated at the virtual body’s position using the (noisy) data available from all vehicles. Centralized computa-
tion is used. A least-squares approximation of the gradient is sued and study the problem of the optimal formation that minimizes estimation error. We also design a Kalman filter and use measurement history to smooth out the estimate.
References


