

UNIVERSIDAD DE GRANADA

ESCUELA INTERNACIONAL DE POSTGRADO



TRABAJO DE FIN DE MÁSTER

Diseño e integración de sistemas de Guiado Navegación y Control
para un enjambre de vehículos autónomos

Design and integration of Guidance, Navigation and Control systems
for autonomous robot swarms

Supervisor: Héctor García de Marina

Jesús Bautista Villar

Máster Universitario en Ciencia de Datos e Ingeniería de Computadores

Curso académico 2022-2023

Convocatoria Julio

Considerando que la presentación de un trabajo hecho por otra persona o la copia de textos, fotos y gráficas sin citar su procedencia se considera plagio, el abajo firmante D. Jesús Bautista Villar con DNI 06294739F, que presenta el Trabajo Fin de Máster titulado "Diseño e integración de sistemas de Guiado Navegación y Control para un enjambre de vehículos autónomos", declara la autoría y asume la originalidad de este trabajo, donde se han utilizado distintas fuentes que han sido todas citadas debidamente en la memoria.

Y para que así conste firmo el presente documento en Granada a **8 de Julio del 2023**.



El autor:

Resumen:

En este trabajo, presentamos dos metodologías originales relacionadas con la robótica de enjambre. En primer lugar, se proponen una serie de resultados rigurosos que permiten garantizar la ausencia de colisiones durante el seguimiento de caminos con enjambres robóticos. Describimos los principios fundamentales de esta técnica y presentamos resultados rigurosos basados en el uso de "*control barrier functions*" y "*guidance vector fields*". Además, demostramos su efectividad en entornos reales a través de una colección de simulaciones y experimentos con una flota de *rovers* autónomos.

En segundo lugar, introducimos un conjunto de herramientas resilientes para buscar fuentes de campos escalares utilizando enjambres de robots. Analizaremos rigurosamente las características principales de estas herramientas, especialmente en términos de sensibilidad y observabilidad, y discutimos los mecanismos de adaptación y resistencia frente a perturbaciones o fallas del sistema. Además, presentamos una serie de simulaciones que validan la eficacia y robustez de estas herramientas en la búsqueda de fuentes de campos escalares.

Todos los resultados obtenidos demuestran la viabilidad y eficacia de ambas metodologías en entornos reales y sientan las bases para futuras investigaciones en el campo de la robótica de enjambres.

Palabras clave: Robótica de enjambre, control robusto, evasión de colisiones, búsqueda de fuentes en campos escalares.

Abstract:

In this work, we present two innovative swarm robotics methodologies. Firstly, we propose a collision-free path-following technique for robot swarms. We describe the fundamental principles of this methodology and provide rigorous results based on the utilization of "*control barrier functions*" and "*guidance vector fields*". Through a series of simulations and experiments with an autonomous rover fleet, we demonstrate its effectiveness in real-world environments.

Secondly, we introduce a resilient approach for source-seeking using robot swarms. We conduct a thorough analysis of this methodology, considering sensitivity & observability, and mechanisms for adaptation and resilience against disturbances or system failures. To validate its efficacy and robustness, we present a set of simulations that showcase the successful search for scalar field sources.

Overall, our results highlight the feasibility and effectiveness of both methodologies in real-world scenarios, providing a solid foundation for further research and advancements in swarm robotics.

Keywords: Swarm robotics, robust control, collision avoidance, source-seeking.

ÍNDICE

I.	Introducción	2
I-A.	Objetivos	3
I-B.	Organización	3
II.	Metodología para garantizar la ausencia de colisiones durante el seguimiento de caminos con enjambres robóticos	4
II-A.	Fundamentos teóricos	5
II-A.1.	Notación	5
II-A.2.	Dinámica de un rover con ruedas direccionales	6
II-A.3.	Seguimiento de caminos guiado por campos vectoriales	8
II-A.4.	Garantizando la seguridad de sistemas críticos con CBFs	11
II-A.5.	Diseño de una CBF que permita evita colisiones	13
II-B.	Formalización del problema	14
II-C.	Garantizando la ausencia de colisiones entre dos <i>rovers</i> con C3BFs	15
II-D.	Garantizando la ausencia de colisiones en enjambres de n <i>rovers</i> con C3BFs	18
II-E.	Simulaciones numéricas	20
II-F.	Experimentos	23
II-F.1.	Diseño y construcción de una flota autónoma de rovers	23
II-F.2.	Integración distribuida del algoritmo de seguimiento de caminos y resultados	25
II-G.	Conclusiones	27
III.	Metodología para localizar de forma resiliente la fuente de un campo escalar con enjambres robóticos	28
III-A.	Fundamentos teóricos	29
III-A.1.	Algoritmos existentes de source-seeking: Virtudes y defectos	29
III-A.2.	Notación general	30
III-B.	Teoría de Grafos	32
III-C.	Formalización del problema	32
III-D.	Herramientas de <i>source-seeking</i>	33
III-D.1.	La dirección de ascenso	34
III-D.2.	Ánalisis de observabilidad y sensibilidad	41
III-D.3.	Estimación distribuida del centroide	52
III-E.	Simulaciones	54
III-F.	Conclusiones	60
Conclusión final		61
Referencias		61

I. INTRODUCCIÓN

Existe un consenso generalizado dentro de la comunidad robótica de que los enjambres robóticos serán una herramienta fundamental en el futuro. Una vez esta tecnología alcance su madurez, permitirá llevar a cabo misiones críticas y persistentes, sobre todo en grandes espacios de terreno, o que requieran de una especial resiliencia y abaratamiento de costes, especialmente en términos logísticos y de mantenimiento.

Para entender correctamente a lo que nos referimos cuando hablamos de enjambres de robots, basta con fijarnos en la propia naturaleza. En ella existen numerosos organismos que colectivamente multiplican sus capacidades, generando comportamientos emergentes para resolver tareas que de forma individual pueden resultar muy complejas (ver Figura 1). Dichos comportamientos suelen ser muy beneficiosos para el grupo, siempre y cuando la interacción local entre vecinos sea adecuada.



Fig. 1: Esta fotografía ilustra cómo las hormigas son capaces de organizarse en una estructura entrelazada, utilizando sus cuerpos para formar un puente que les permite superar obstáculos y llegar a su destino de manera eficiente.

De forma muy similar, en robótica nos referimos a un sistema descentralizado compuesto por múltiples robots poco sofisticados que, trabajando en equipo, logran generar una serie de **comportamientos colectivos** que permiten mejorar el rendimiento individual de cada agente. Este tipo de sistemas son muy interesantes y poderosos, no obstante, plantean una dificultad logística lo suficientemente significativa como para ser considerados actualmente como uno de los 10 grandes retos actuales en robótica según *Science Robotics* [1].

En función de la finalidad para la que el sistema esté destinado, este gran reto se suele descomponer en numerosos problemas que también pueden llegar a ser bastante complejos. A nivel individual, podemos querer que cada individuo sea capaz percibir el medio, detectar la presencia de vecinos o seguir caminos. Mientras que a nivel de colectivo, se puede pensar en algoritmos distribuidos que permitan controlar movimientos en formación [2], realizar localización relativa [3], evitar colisiones o buscar la fuente de un campo escalar. Al igual que en la naturaleza, todas estas tareas tan fundamentales pueden ser llevadas a cabo mediante una *interacción local entre robots basada en las cuatro C's* [4]: **coordinación, cooperación, colaboración y competición**.

En este campo de la robótica siempre se habla sobre alcanzar la autonomía o **inteligencia de enjambre**, no obstante, no se suelen encontrar las técnicas más comunes de IA (Inteligencia Artificial), como lo son aquellas basadas en ML (*Machine Learning*) o DP (*Deep Learning*). Todas las metodologías que han demostrado rendir adecuadamente se basan reglas heurísticas, lógica difusa o teoría de control. Esto es algo que cobra bastante sentido cuando pensamos que en robótica, especialmente cuando hablamos de sistemas GNC (Guiado, Navegación y Control), lo que se suele buscar son algoritmos que aporten respuestas deterministas y altamente explicables. Este trabajo se centrará principalmente en técnicas de control basadas en análisis rigurosos y formales, pues por su naturaleza pueden llegar a ser más **robustas y resilientes** que otros métodos basados en reglas o comportamientos simples. Este tipo de garantías son necesarias en el caso de misiones consideradas "críticas" o que requieran de persistencia en el tiempo, conocidas como 24/7.

I-A. Objetivos

El fin principal de este trabajo es coordinar a un enjambre de robots mediante algoritmos GNC. Para lograrlo, se tendrán que alcanzar los siguientes objetivos específicos:

- 1) Desarrollar un algoritmo para la coordinación de vehículos terrestres garantizando la ausencia de colisiones.
- 2) Implementar el algoritmo anterior en los sistemas empotrados de un enjambre robótico, de manera que todos los robots sean capaces de coordinarse de forma autónoma, sin la intervención de sistemas externos.
- 3) Desarrollar un algoritmo de búsqueda de fuentes en campos escalares para enjambres de robots que sea resiliente, distribuido y escalable.

Dado que la presencia de garantías formales es uno de los puntos fuertes en teoría de control, todos los resultados matemáticos desarrollados en este trabajo serán demostrados y validados de forma numérica o experimental.

I-B. Organización

Una vez introducido el TFM y planteados sus objetivos, se presentarán dos metodologías distintas. Cada una de ellas, se focalizará en resolver un problema completamente distinto, por lo que ambas contarán con sus propias secciones de fundamentos teóricos, exposición de resultados, verificación numérica/experimental y conclusión. Una vez se haya terminado de hablar sobre ambas metodologías, en las últimas páginas de este trabajo se podrá encontrar la conclusión final y la sección de referencias.

II. METODOLOGÍA PARA GARANTIZAR LA AUSENCIA DE COLISIONES DURANTE EL SEGUIMIENTO DE CAMINOS CON ENJAMBRES ROBÓTICOS

Gran parte de las tareas de navegación con robots más fundamentales, como la monitorización, la vigilancia o el patrullaje, llevan la necesidad imperante de llevar a cabo un seguimiento preciso de ciertas rutas preestablecida.

Para este desafío fundamental, el estado actual de la robótica nos brinda una amplia variedad de algoritmos innovadores que ofrecen soluciones viables. Entre ellos, destacan dos enfoques principales. Por un lado, están las técnicas basadas en el **seguimiento de trayectorias** (*trajectory tracking*), donde se indican al robot una serie de puntos de referencia (*waypoints*) que debe recorrer en orden y alcanzar en un momento específico. Por otro lado, tenemos los algoritmos basados en el **seguimiento de caminos** (*path following*), donde lo que se proporciona al robot es la representación matemática del camino, sin ningún tipo de información temporal (ver Figura 2).

Las técnicas de seguimiento de trayectorias son muy rudimentarias y suelen encontrarse con bastantes dificultades, por ejemplo, a la hora de mantener una velocidad constante. Es por esta razón que, en este trabajo, nos centraremos en el seguimiento de caminos, ya que son algoritmos de navegación más adecuados para una amplia gama de aplicaciones.

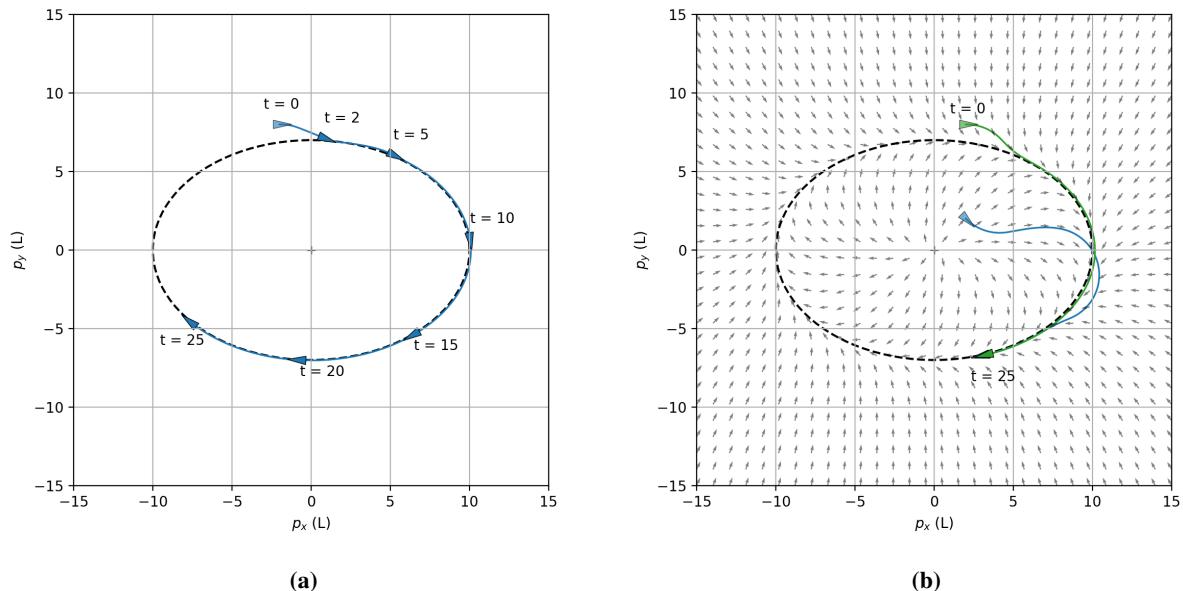


Fig. 2: En esta figura se ilustran dos tipos de algoritmos utilizados para el seguimiento de rutas en robots con dinámica de uniciclo. En (a) tenemos seguimiento de trayectorias, donde el robot cuenta con un controlador que le permite alcanzar los *waypoints* indicados en un tiempo determinado. En (b) se muestra un ejemplo de seguimiento de caminos, donde los robots generan y se alinean con un campo de guiado basado en la curva matemática proporcionada. Se observa que en \$t = 25\$, los dos robots del caso (b) han colisionado.

El problema que abordaremos en esta sección surge al intentar coordinar un enjambre de robots para que sigan caminos cerrados. Como se muestra en la Figura 2b, cuando tenemos a más de un robot moviéndose en un mismo entorno, la ausencia de colisión entre ellos nunca va a estar garantizada. Como es de esperar, esta situación afecta negativamente a la robustez y resiliencia de nuestros sistemas, por lo que resulta de gran interés y relevancia en el contexto de sistemas multi-robot.

En este TFM presentaremos, demostraremos y verificaremos una serie de **resultados originales** que, dados ciertos supuestos, permitirán aportar una primera solución al problema de garantizar la ausencia de colisiones entre *rovers* en el seguimiento de caminos.

II-A. Fundamentos teóricos

II-A.1. Notación

Cuando contemos con un enjambre de $N \in \mathbb{N}$ robots, asignaremos a cada uno de ellos una etiqueta $i \in \mathcal{N} := \{1, 2, \dots, N\}$, donde \mathcal{N} se conoce como el conjunto de robots. El estado de cada uno de estos robots vendrá descrito por el vector de estados $x_i \in \mathcal{D}_i \subset \mathbb{R}^m$, donde $m \in \mathbb{N}$ es el número de coordenadas que permiten describir de forma única dicho estado. Dentro de dicho vector x_i siempre se encontrará la componente $p_i \in \mathbb{R}^2$, que corresponde a la posición del robot i en el plano con respecto a un sistema de referencia inercial \mathcal{O}_N . Véase que, cuando no se haga referencia a un robot i en específico, el conjunto de estados permitidos \mathcal{D}_i se denotará de forma genérica por claridad como \mathcal{D} .

Todo vector unitario se denotará como $\hat{u} := u/\|u\|$, para $u \in \mathbb{R}^n$, donde $n \in \mathbb{N}$ es una dimensión arbitraria. El proyector asociado a una dirección paralela a \hat{u} se denotará como $P_u^\parallel = uu^T$, mientras que el asociado a la perpendicular como $P_u^\perp = I - uu^T$, donde $I \in \mathbb{R}^{n \times n}$ es la matriz identidad. También es importante tener en cuenta que $E := R(\pi/2)$ se traduce como la matriz rotación de 90° en 2D, donde

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \in \mathbb{R}^{2 \times 2}. \quad (\text{II.1})$$

Se dice que una función $\kappa : [0, d] \rightarrow [0, \infty)$, para un $d > 0$, pertenece a la **clase \mathcal{K}** cuando es estrictamente creciente y $\kappa(0) = 0$. Las funciones de **tipo \mathcal{C}^2** se caracterizan por ser continuamente diferenciables hasta, al menos, la segunda derivada. Se entiende que una función es **regular** en cierta región \mathcal{R} si se demuestra que es diferenciable y monoevaluada en \mathcal{R} .

La **continuidad de Lipschitz** permite garantizar de una forma mucho más fuerte la continuidad uniforme en funciones, mediante una restricción que limita su tasa de cambio. En particular, se dice que una función $f : \mathbb{R} \rightarrow \mathbb{R}$ es Lipschitz continua cuando existe un $K \in \mathbb{R} \geq 0$ tal que, para todos los números reales x_1 y x_2 , se cumple que

$$|f(x_1) - f(x_2)| \leq K|x_1 - x_2|$$

En teoría de control, existe un tipo de sistemas llamados **sistemas afines**:

$$\dot{x} = f(x) + g(x)u, \quad (\text{II.2})$$

donde $u \in \mathbb{U} \subseteq \mathbb{R}^n$ es la entrada al sistema dinámico. A la hora de diseñar controladores realimentados en lazo cerrado, tendremos $u = k(x)$, donde k es una función Lipchitz continua que se conoce como señal o ley de control. Véase que, en casos reales, el subconjunto \mathbb{U} rara vez va a poder ser igual a \mathbb{R}^n , pues las limitaciones físicas de la entrada al sistema dinámico de un robot suelen ser muy fuertes. En el caso de los *rovers*, ya veremos que u es una entrada que actúa directamente sobre el ángulo de giro de las ruedas, variable que evidentemente va a estar muy restringida por las características físicas del sistema.

II-A.2. Dinámica de un rover con ruedas direccionales

Esta sección se dedicará a analizar la dinámica de los robots que componen al enjambre. En nuestro caso particular se trata de unos vehículos terrestres con ruedas direccionales, muy similares al automóvil convencional. Lo primero que se ha de tener en cuenta cuando uno se encuentra ante este tipo de sistemas con ruedas móviles es la condición de rodar sin deslizar. Dicha ligadura provoca que este tipo de sistemas físicos sean **no holonómicos**, es decir, incapaces de modificar de forma instantánea su dirección de avance.

Comúnmente, se tiende a pensar que las dos ruedas frontales del automóvil giran con el mismo ángulo. No obstante, si queremos que realmente se cumpla la condición de no deslizamiento sobre ambas ruedas, esto no puede ser así. Tal y como se muestra en el primer esquema de la Figura 3, dado un centro instantáneo de rotación *CoR*, el eje longitudinal de cada rueda deberá de ser tangente a la circunferencia con centro en *CoR* que pasa por ella. Normalmente esto se consigue implementando una **dirección de Ackermann** en el eje frontal del automóvil, de modo que el ángulo de giro de cada rueda se puede modelar con

$$\psi_{in} = \tan^{-1} \left(\frac{l}{\rho - l_t/2} \right) \in \mathbb{R} \quad \text{y} \quad \psi_{out} = \tan^{-1} \left(\frac{l}{\rho + l_t/2} \right) \in \mathbb{R}, \quad (\text{II.3})$$

donde $l \in \mathbb{R}$ corresponde a la distancia entre el eje trasero y el frontal, $l_t \in \mathbb{R}$ a la distancia entre ruedas y $\rho \in \mathbb{R}$ al radio de giro.

El comportamiento físico de este tipo de direcciones es relativamente complejo. No obstante, siempre y cuando las condiciones sean las ideales, el automóvil se puede modelar de forma equivalente como el sistema que se muestra en el segundo esquema de la Figura 3, un vehículo con una única rueda direccional frontal (**uniciclo**). Dicha rueda es virtual y su ángulo de giro se puede modelar de forma muy similar al de las reales:

$$\psi = \tan^{-1} \left(\frac{l}{\rho} \right) \in \mathbb{R}. \quad (\text{II.4})$$

Ha de tenerse en cuenta que, dado un ψ , conocido l se puede determinar sin problemas el radio de giro ρ . Por lo tanto, sabiendo el valor de l_t y teniendo en cuenta (II.3), es directo calcular los ángulos ψ_{in} y ψ_{out} que corresponde a las ruedas reales.

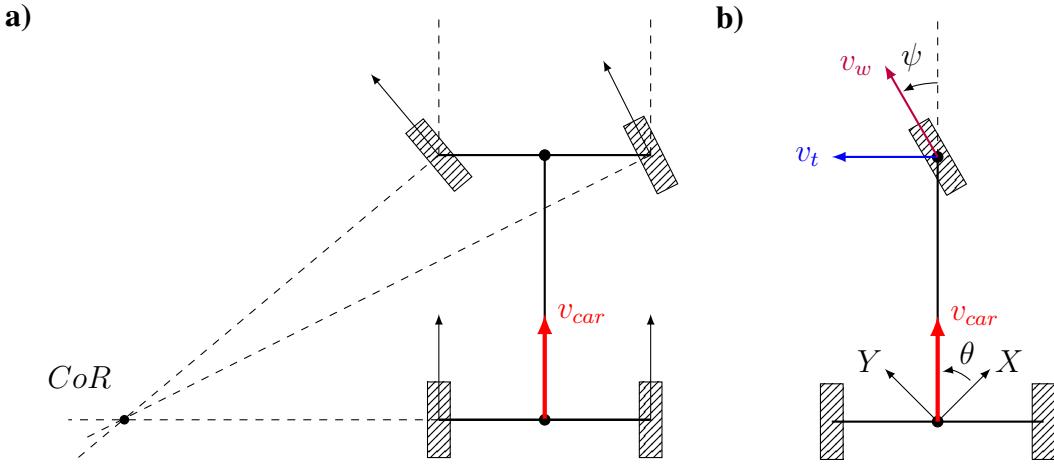


Fig. 3: En a) se muestra el esquema real de un *rover* con dirección de Ackermann, las dos ruedas frontales conforman ángulos distintos con respecto a la vertical. Por otro lado, en b) tenemos el esquema de un sistema simplificado con una rueda virtual que modela la dirección de Ackermann, junto a sus coordenadas generalizadas. Véase que ψ es el ángulo de giro de la rueda y θ corresponde al ángulo que forma v_{car} con respecto al eje X del sistema de referencia.

Teniendo en cuenta todo lo comentado anteriormente, las ecuaciones dinámicas del sistema que rigen el comportamiento nominal de nuestros *rovers* con ruedas direccionales se pueden representar en el espacio de estados como

$$\dot{x} = \begin{bmatrix} \dot{x}_p \\ \dot{y}_p \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = v \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ \tan(\psi)/l & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \omega_\psi, \quad (\text{II.5})$$

donde l corresponde a la distancia entre el eje trasero y el frontal. La velocidad de avance del vehículo, $v = \|v_{car}\| \in \mathbb{R}$, se interpreta como una señal de control, pues en todos los casos estudiados en este trabajo se podrá suponer constante. Hay que tener en cuenta que $\tan(\psi)$ presenta una discontinuidad en $\pm\pi/2$, no obstante, por regla general ψ siempre se suele encontrar limitado por un $\psi_{min} > -\pi/2$ y $\psi_{max} < \pi/2$. Si por alguna razón $\psi = \pm\pi/2$, entonces se dejaría de cumplir la condición de rodar sin deslizar, el vehículo saldría de su zona de comportamiento nominal y nuestro modelo físico dejaría de tener sentido.

A continuación, procederemos a explicar detalladamente cómo determinar exactamente (II.5). Si uno se fija en la Figura 3, puede observar que este sistema físico cuenta con 4 coordenadas generalizadas, lo cual indica que será necesario encontrar el mismo número de ecuaciones de estado. El caso de \dot{x} e \dot{y} es bastante sencillo, pues basta con proyectar el vector velocidad v_{car} sobre cada uno de los ejes X-Y. La componente $\dot{\psi}$, que hace referencia a la velocidad de giro del volante, también es bastante directa, pues corresponde a una señal de actuación sobre el sistema.

La ecuación más complicada de determinar es la de $\dot{\theta}$. No obstante, para un modelo tan simplificado como el que se muestra en la Figura 3 resulta relativamente intuitivo. Lo más importante es observar que el vector velocidad de la rueda v_w se puede descomponer en

$$\|v_{car}\| = \|v_w\| \cos \psi \quad y \quad \|v_t\| = \|v_w\| \sin \psi,$$

donde v_{car} es la celeridad del vehículo y v_t su velocidad lineal de giro. De este modo, tenemos $\|v_t\| = l\dot{\theta}$, que en combinación con las dos expresiones anteriores nos permite llegar directamente a la ecuación diferencial para la velocidad angular:

$$\dot{\theta} = \frac{\|v_t\|}{l} = \frac{\|v_w\| \sin \psi}{l} = \frac{\|v_{car}\|}{l} \tan \psi.$$

Finalmente, conviene destacar que el sistema dinámico que se muestra en (II.5) puede simplificarse intercambiando la señal de control ω_ψ por ψ . Esta es una aproximación que tiene bastante sentido cuando el valor máximo de $|\omega_\psi|$ es lo suficientemente alto como para cambiar el estado de ψ de forma prácticamente instantánea. Una vez realizada dicha aproximación, entonces intercambiaremos la señal de control ψ por la velocidad de rotación angular w_θ , pues nos permitirá linealizar el sistema dinámico. Para ello, invirtiendo en (II.5) la componente $\dot{\theta}(v, \psi)$ nos queda que

$$\psi = \tan^{-1} \left(\frac{l\omega_\theta}{v} \right). \quad (\text{II.6})$$

Por regla general, los sistemas reales no tienden a respetar ecuaciones lineales, por lo que esta técnica, conocida en teoría de control como **inversión dinámica no lineal**, se suele utilizar con bastante frecuencia. Una vez llegados a este punto, teniendo en cuenta todo lo anterior se pueden plantear unas **ecuaciones dinámicas del sistema** mucho más simplificadas que las de (II.5):

$$\dot{x} = \begin{bmatrix} \dot{x}_p \\ \dot{y}_p \\ \dot{\theta} \end{bmatrix} = v \underbrace{\begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{g(x)} \underbrace{\omega_\theta}_{u}. \quad (\text{II.7})$$

Véase que (II.7) ya si toma la forma de un sistema afín (II.2). No obstante, se ha de tener siempre en cuenta que, dado un w_θ , para actuar realmente sobre dicho sistema tendremos que computar un ψ con (II.6), pues esta es la coordenada que nos permitirá modular la señal de los actuadores del volante.

II-A.3. Seguimiento de caminos guiado por campos vectoriales

Entre los algoritmos más populares para el seguimiento de caminos disponibles en la literatura, se encuentran el rastreo de un punto de referencia que varía a lo largo del tiempo (*carrot*) [5], la minimización del error relacionado con la distancia euclídea al camino deseado [6], y

el seguimiento de un campo vectorial de guiado (GVF, *Guidance Vector Field*) [7]–[10]. Estas soluciones son consideradas confiables y factibles de implementar en sistemas con recursos hardware limitados. Sin embargo, al ser aplicadas en sistemas no cinemáticos, muchas de ellas están restringidas al seguimiento de líneas rectas o circunferencias, pues cuentan con una serie delimitaciones muy significativas.

Las técnicas basadas en el seguimiento de caminos mediante GVF han demostrado un rendimiento excepcional en numerosas tareas, permitiendo el seguimiento de caminos en sistemas dinámicos relativamente complejos. En la literatura, es común encontrar trabajos que proponen campos vectoriales diseñados a medida para caminos específicos [7]. No obstante, también existen soluciones que logran seguir caminos genéricos [8], aunque suelen estar limitadas por singularidades en el campo vectorial que pueden afectar a la convergencia para ciertos tipos de caminos. En este trabajo, nos centraremos en el enfoque propuesto en [9], pues plantea un algoritmo que trabaja directamente con ecuaciones implícitas en 2D, lo que lo hace muy sencillo e intuitivo. Cabe mencionar que los mismos autores proponen en [10] una solución más refinada que permite seguir caminos en 3D, abordando el problema de las intersecciones y resolviendo numerosas singularidades.

Una de las contribuciones más interesantes de [9] consiste en definir el camino $\mathcal{P} \subset \mathbb{R}^2$ mediante la ecuación implícita $\varphi(x) : \mathbb{R}^2 \rightarrow \mathbb{R}$, de tal modo que $\mathcal{P} := \{p : \varphi(p) = 0\}$. Esta representación es fascinante, ya que permite cubrir todo el plano \mathbb{R}^2 con las curvas de nivel $\varphi(p) = c$. De esta manera, el error entre el camino \mathcal{P} y el agente se puede definir como

$$e(p) := \varphi(p) \in \mathbb{R}. \quad (\text{II.8})$$

Basándose en este error, los autores proponen el siguiente campo vectorial de guiado:

$$\dot{p}_d := \tau(p) - k_e e(p) n(p), \quad (\text{II.9})$$

donde $k_e \in \mathbb{R}^+$ representa una ganancia que modula la agresividad del campo, mientras que $\tau \in \mathbb{R}^2$ y $n \in \mathbb{R}^2$ son respectivamente las componentes tangencial y normal a \mathcal{P} . Como se puede apreciar en la Figura 4, el vector n permite acercarse progresivamente al camino, mientras que τ hace que sea posible seguirlo.

En [9] demuestran formalmente que (II.9) es un campo vectorial de guiado adecuado, siempre y cuando $\varphi(p)$ pertenezca a C^2 (dos veces continuamente derivable) y es regular en un entorno de \mathcal{P} . Esto último siempre se cumplirá si, por ejemplo,

$$\nabla \varphi(p) \neq 0, \quad p \in \mathcal{N}_{\mathcal{P}}, \quad (\text{II.10})$$

donde $\mathcal{N}_{\mathcal{P}} := \{p : |\varphi(p)| \leq c^*\}$, con $c^* \in \mathbb{R}^+$. Además, los mismos autores también diseñan un controlador que, actuando sobre una velocidad de giro $\omega \in \mathbb{R}$, que en el caso de nuestros rovers será ω_θ , permite alinear la dirección del vehículo con (II.9). Si se consulta el artículo, se puede observar que la agresividad de convergencia al GVF vendrá modulada por una constante $k_d \in \mathbb{R}^+$.

A la hora de poder usar este algoritmo, únicamente nos preocupan dos aspectos clave. En primer lugar, dado un camino a seguir, tendremos que calcular e introducirle al modelo la expresión explícita de $\varphi(p)$, $\nabla\varphi(p)$ y $H(\varphi(p))$. Por otro lado, también es importante conocer las ganancias k_e y k_d , ya que nos permitirán modular el comportamiento de nuestro sistema de navegación.

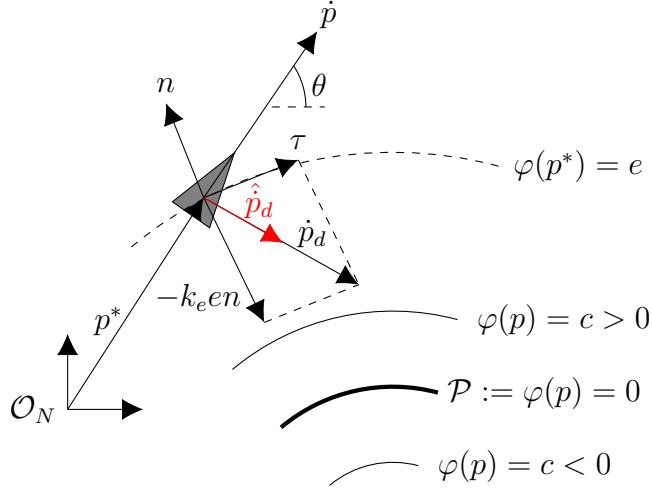


Fig. 4: Figura adaptada de [9]. El vector con el que el *rover* se tiene que alinear en p^* viene dado por \hat{p}_d . Las componentes normal n y tangencial τ se calculan con $\nabla\varphi(p^*)$, y el error e viene dado por $\varphi(p^*)$.

Supongamos que queremos que nuestro robot siga un camino cerrado con forma de elipse. En primer lugar, podemos partir de la ecuación general de la elipse en coordenadas cartesianas:

$$\frac{x_{el}^2}{a^2} + \frac{y_{el}^2}{b^2} = 1, \quad \text{con} \quad \begin{bmatrix} x_{el} \\ y_{el} \end{bmatrix} = R(\alpha) \begin{bmatrix} x - h_x \\ y - h_y \end{bmatrix},$$

donde, $h = [h_x \ h_y]^T$ es el centro de la elipse con respecto a \mathcal{O}_N , $R(\alpha)$ es una matriz de rotación del tipo (II.1), α es el ángulo de rotación de la elipse con respecto a \mathcal{O}_N y a y b son las longitudes de los dos ejes de la elipse. En base a esta expresión, es muy sencillo determinar la ecuación implícita que define \mathcal{P} :

$$\varphi(p) = \frac{x_{el}^2}{a^2} + \frac{y_{el}^2}{b^2} - 1$$

Véase que esta $\varphi(p)$ cumple (II.10), por lo que podrá generar un campo de guiado válido. Una vez conocida la ecuación implícita del camino, solo queda determinar el gradiente $\nabla\varphi(p)$ y su hessiano $H(\varphi(p))$:

$$\nabla\varphi(p) = \begin{bmatrix} \frac{2x_{el}}{a^2} \cos(\alpha) + \frac{2y_{el}}{b^2} \sin(\alpha) \\ \frac{2y_{el}}{b^2} \cos(\alpha) - \frac{2x_{el}}{a^2} \sin(\alpha) \end{bmatrix} \quad \text{y} \quad H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix},$$

donde

$$\begin{aligned} H_{11} &= 2 \left(\frac{\cos(\alpha)^2}{a^2} + \frac{\sin(\alpha)^2}{b^2} \right), \\ H_{12} = H_{21} &= 2 \sin(\alpha) \cos(\alpha) \left(\frac{1}{b^2} - \frac{1}{a^2} \right), \\ H_{22} &= 2 \left(\frac{\sin^2(\alpha)}{a^2} + \frac{\cos^2(\alpha)}{b^2} \right). \end{aligned}$$

Introduciendo toda esta información al algoritmo propuesto en [9], y modulando el controlador con unas ganancias $k_e = 4$ y $k_d = 2$, se puede replicar el resultado que se muestra en la Figura 2b, donde dos robots tratan de seguir una elipse con $a = 10$, $b = 7$ y $\alpha = 0$ grados.

II-A.4. Garantizando la seguridad de sistemas críticos con CBFs

Todos estamos de acuerdo en que cualquier sistema ingenieril debe ser diseñado para ser seguro, garantizando que no ocurran situaciones indeseables. En este contexto, una herramienta que nos permite verificar y asegurar la seguridad de los controladores más críticos son las CBFs (*Control Barrier Functions*).

Este tipo de algoritmos se basan en la siguiente definición matemática de seguridad, introducida por Nagumo en los años cuarenta. Dado un sistema dinámico $\dot{x} = f(x)$ y una función continua y derivable $h(x) : \mathcal{D} \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$, con $\frac{\partial h}{\partial x}(x) \neq 0$ para todo x tal que $h(x) = 0$, se puede definir el **conjunto seguro** como

$$\mathcal{C} := \{x \in \mathcal{D} \subseteq \mathbb{R}^n : h(x) \geq 0\}.$$

En relación a dicho conjunto, el Teorema de Nagumo demuestra lo siguiente:

$$\mathcal{C} \text{ es } \mathbf{invariante} \Leftrightarrow \dot{h}(x) \geq 0 \quad \forall x \in \partial\mathcal{C}. \quad (\text{II.11})$$

Véase que, mientras el conjunto seguro sea invariante, se podrá garantizar que el sistema dinámico es seguro. Para trasladar esta definición al contexto de la teoría de control, es necesario extender el resultado de Nagumo a sistemas afines (II.2). Para este tipo de sistemas, tendremos un conjunto seguro \mathcal{C} definido por una función h . Estas funciones son las que se conocen como CBF, y para saber si son válidas o no tendremos que tener en cuenta la siguiente definición.

Definición II.1 (CBF válida). — *Dada una CBF $h(x)$, un sistema afín (II.2) y cierto conjunto de estados $\mathcal{B} \subset \mathcal{D}$, se entiende que la CBF es válida para dicho sistema en \mathcal{B} cuando*

$$\mathcal{L}_f h(x) + \mathcal{L}_g h(x) u \geq -\kappa(h(x)), \quad (\text{II.12})$$

donde $\mathcal{L}_f h(x) = \frac{\partial h}{\partial x} f(x)$ y $\mathcal{L}_g h(x) = \frac{\partial h}{\partial x} g(x)$ son las derivadas de Lie, y $\kappa : \mathbb{R} \rightarrow \mathbb{R}$ es una función de clase \mathcal{K} .

La condición (II.12) es una variante mucho más relajada de la condición orginal de Nagumo (II.11), de tal modo

$$\mathcal{C} \text{ es \textbf{invariante}} \Leftrightarrow \underbrace{\mathcal{L}_f h(x) + \mathcal{L}_g h(x)u}_{\dot{h}(x)} \geq -\kappa(h(x)).$$

La función κ se encargará de modular la agresividad de (II.12) en función del valor de $h(x)$. Cuanto mayor sea el valor de κ más sencillo será cumplir la condición. Véase que, cuando $h(x) = 0$, entonces $\kappa(h(x)) = 0$, por lo que se recupera la condición (II.11), que es la más restrictiva posible. En la Figura 5 se muestra un ejemplo que permite visualizar este comportamiento.

Haciendo uso de esta condición, se pueden utilizar métodos de control basados en optimización para modificar a un controlador de referencia de la forma menos invasiva posible. Más concretamente, si suponemos que sobre el sistema afín (II.2) actua un controlador realimentado con una señal de control $u_{ref}(x, t)$, se puede proponer el siguiente problema de programación cuadrática (QP, *Quadratic Program*):

$$\begin{aligned} u^*(x, t) = \min_{u \in \mathbb{U}} & \|u - u_{ref}(x, t)\|^2 \\ \text{t.q. } & \mathcal{L}_f h(x) + \mathcal{L}_g h(x)u + \kappa(h(x)) \geq 0 \end{aligned} \tag{II.13}$$

Ha de tenerse en cuenta que, cuando u se encuentre acotada, pueden existir ciertos casos para los que este CBF-QP no tenga una solución contenida en \mathbb{U} . Únicamente podremos asegurar que es factible cuando $\mathbb{U} = \mathbb{R}^m$, en estos casos su solución explícita será

$$u^*(x, t) = u_{ref}(x, t) + u_{safe}(x, t),$$

donde $u_{safe}(x, t)$ viene dado por

$$u_{safe}(x, t) = \begin{cases} 0 & , \text{ para } \Psi(x, t) \geq 0 \\ -\frac{\mathcal{L}_g h(x)^T \Psi(x, t)}{\mathcal{L}_g h(x)^T \mathcal{L}_g h(x)} & , \text{ para } \Psi(x, t) < 0 \end{cases}, \tag{II.14}$$

con $\Psi(x, t) := \dot{h}(x, u_{ref}(x, t)) + \kappa(h(x))$. Véase que el signo de esta función Ψ indicará si se ha de activar o no el controlador CBF u_{safe} .

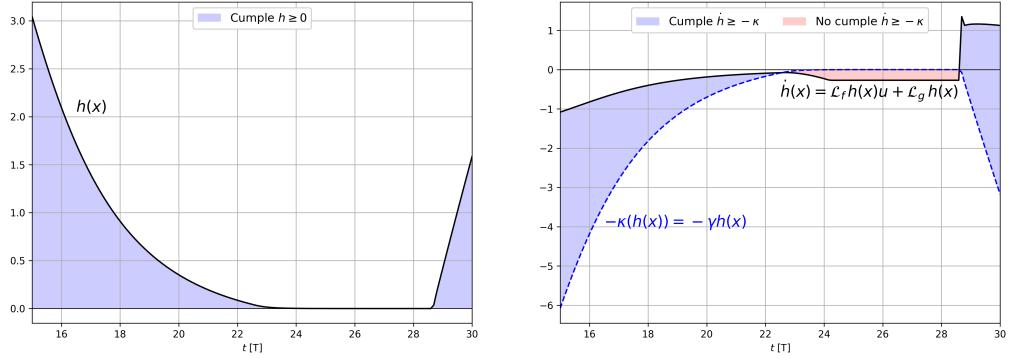


Fig. 5: En el gráfico de la izquierda tenemos la evolución de una CBF $h(x)$ a lo largo del tiempo, mientras que a la derecha se muestra su derivada temporal $\dot{h}(x)$. En esta figura, resulta interesante visualizar cómo varía la función $\kappa(h(x))$ según el valor que toma $h(x)$. Cuando el valor de la CBF se encuentra muy próximo a 0, la condición (II.12) se deja de cumplir para $u = 0$ porque κ sobre pasa a \dot{h} , de modo que el controlador (II.14) entra en acción para tratar de mantener $h(x) \geq 0$.

II-A.5. Diseño de una CBF que permite evita colisiones

Si queremos utilizar las CBFs para evitar colisiones entre robots, es fundamental desarrollar una CBF válida que se adapte a este contexto. Es decir, necesitamos diseñar una función que identifique las situaciones que podrían llevar al sistema a entrar en estado de colisión.

Inicialmente, uno podría comenzar asumiendo que los obstáculos tienen forma circular. De este modo, el conjunto resultante de vectores velocidad no deseados resulta ser un cono como el que se muestra en rojo en la Figura 6. Con esta idea es como nacen las C3BFs (*Collision Cone Control Barrier Functions*), comúnmente utilizadas en aplicaciones aeroespaciales, como el guiado de misiles [11] o la detección y resolución de conflictos entre aviones [12]. Una posible C3BF capaz de generar estas regiones de colisión cónicas es

$$h(x, t) = p_{rel}v_{rel}^T + \|p_{rel}\| \|v_{rel}\| \cos \phi, \quad (\text{II.15})$$

donde $p_{rel} = (p_{obs} - p_{rbt}) \in \mathbb{R}^2$ y $v_{rel} = (v_{obs} - v_{rbt}) \in \mathbb{R}^2$ son los vectores de posición y velocidad relativa, y ϕ es la mitad del ángulo del cono. Visualizando la Figura 6 es sencillo comprobar que

$$\cos \phi = \frac{\sqrt{\|p_{rel}\|^2 - r^2}}{\|p_{rel}\|},$$

donde $r \in \mathbb{R}$ es el radio del obstáculo virtual con forma circular.

Aplicando esta C3BF a la condición (II.12) podemos generar un conjunto seguro \mathcal{C} que excluya aquellos casos en los que el ángulo entre p_{rel} y v_{rel} es menor que $\pi - \phi$. En la Figura 6 se muestra un ejemplo donde el estado del sistema se encuentra fuera de la zona segura.

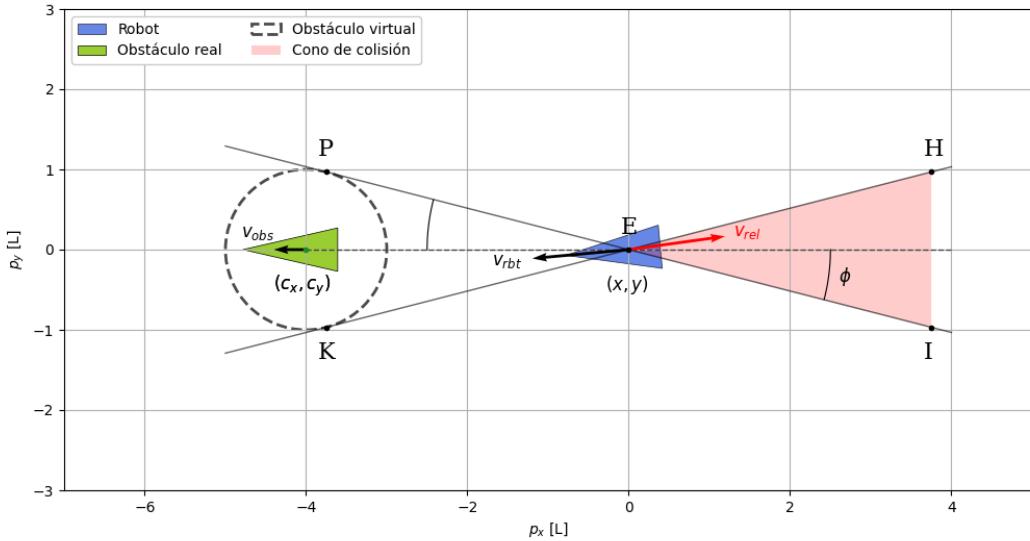


Fig. 6: Esquema, adaptado de [13], donde se visualiza la construcción del cono de colisión para un obstáculo virtual de radio r . Este es un caso donde el sistema se encuentra en rumbo de colisión, pues v_{rel} está dentro de la región EHI. Véase que v_{obs} y v_{rbt} corresponden a los vectores velocidad del obstáculo y el robot, y que $p_{obs} = [c_x \ c_y]^T$.

II-B. Formalización del problema

La correcta formalización de los problemas que se proponen en ciencia e ingeniería es esencial, ya que proporciona una base sólida para su posterior estudio y análisis. Un problema presentado de forma clara, facilita la comunicación, fomenta la reproducción de resultados y contribuye al avance del conocimiento. Es por esta razón que dedicaremos una sección a formalizar y explicar el problema de garantizar la ausencia de colisiones entre *rovers* en el seguimiento de caminos de forma coordinada.

En primer lugar, hemos de dejar claro lo que entendemos por colisión:

Definición II.2 (Colisión entre dos robots). — *Dado un conjunto de robots \mathcal{N} , se entiende que un robot i ha colisionado con otro robot j cuando $\|p_j - p_i\| < r$, donde $r \in \mathbb{R}^+$ es el radio de un obstáculo circular.*

En base a este enunciado, se puede entender el subconjunto de estados de colisión como $\mathcal{X}_i := \{x_i : \|p_j - p_i\| < r, \forall j \neq i \in \mathcal{N}\} \subset \mathcal{D}_i$. El conjunto seguro $\mathcal{C}_{ij} \subseteq \mathcal{C}_i$ se define como un conjunto seguro \mathcal{C} que excluye al subconjunto de estados de colisión entre un robot i y otro robot j , denotado por $\mathcal{X}_{ij} := \{x_i : \|p_j - p_i\| < r, j \neq i \in \mathcal{N}\} \subseteq \mathcal{X}_i$. Véase que, dado un i fijo, \mathcal{C}_i es el resultado de la intersección de todos los \mathcal{C}_{ij} . Teniendo en cuenta toda la notación introducida hasta llegar a este punto, ya nos encontramos en disposición de formalizar nuestro problema.

Problema II.1 (Garantizar ausencia de colisiones entre *rovers* en el seguimiento de caminos de forma coordinada). — *Supongamos un conjunto \mathcal{N} de N robots, situados en un mismo plano y con una misma dinámica de estados x_i descrita por (II.7), donde $i \in \mathcal{N}$. Dada una serie de subconjuntos de estados de colisión \mathcal{X}_i , y teniendo en cuenta que u_{ref}^i es una señal de control encargada de coordinar al robot i con el resto del enjambre para seguir el camino \mathcal{P} , el problema de control consiste en diseñar para todo i una u_{safe}^i tal que se cumplan las siguientes condiciones:*

- 1) *Si $x_i(0) \notin \mathcal{X}_i$, entonces $x_i(t) \notin \mathcal{X}_i$ para todo tiempo $t \in \mathbb{R}^+$.*
- 2) *Dada una constante $\epsilon > 0$, se cumple que $|e_i(t)| < \epsilon, \forall t > T$, para algún tiempo $T \in \mathbb{R}^+$.*

Resulta pertinente destacar la importancia de que $x_i(0) \notin \mathcal{X}_i$. Si esta condición inicial no se cumple, entonces es imposible asegurar que CBF vaya a llevar al sistema a un estado seguro, pues necesariamente $x_i(0) \in \mathcal{C}_i$. También es interesante tener en cuenta que T y ϵ son constantes que van depender de la agresividad de u_{ref} y u_{safe} , modulada por los parámetros k_e o k_d en el caso de u_{ref} y $\kappa(h(x))$ en el caso de u_{safe} .

Véase que en el Problema II.1 no se exige en ningún momento que el sistema sea siempre seguro, es decir, que $x_i \in \mathcal{C}_i$. Esta es una condición suficiente, pero no necesaria, para evitar colisiones en un enjambre de robots.

II-C. Garantizando la ausencia de colisiones entre dos *rovers* con C3BFs

Una vez introducidas las C3BFs, comenzaremos a explorar la forma de aplicarlas para resolver el Problema II.1. En primer lugar, necesitamos definir los vectores posición relativa p_{ij} y velocidad relativa v_{ij} . El cálculo del vector posición relativa entre el robot i y el obstáculo j es bastante directo:

$$p_{ij} := p_j - p_i = \begin{bmatrix} x_p^j - x_p^i \\ y_p^j - y_p^i \end{bmatrix}. \quad (\text{II.16})$$

Teniendo en cuenta (II.7), se tiene que la derivada temporal de p_{ij} corresponde a

$$v_{ij} := \begin{bmatrix} v_j \cos \theta_j - v_i \cos \theta_i \\ v_j \sin \theta_j - v_i \sin \theta_i \end{bmatrix}, \quad (\text{II.17})$$

cuya evolución va a venir marcada a su vez por

$$\begin{aligned} \dot{v}_{ij} &= \frac{\partial v_{ij}}{\partial \theta_j} \dot{\theta}_j + \frac{\partial v_{ij}}{\partial \theta_i} \dot{\theta}_i = \dot{v}_j - \dot{v}_i \\ &= \begin{bmatrix} v_j \sin \theta_j \\ -v_j \cos \theta_j \end{bmatrix} \omega_\theta^j + \underbrace{\begin{bmatrix} -v_i \sin \theta_i \\ v_i \cos \theta_i \end{bmatrix}}_u \omega_\theta^i. \end{aligned} \quad (\text{II.18})$$

A la hora introducir estas definiciones en la C3BF (II.15), tendremos que tener en cuenta que $p_{rel} = p_{ij}$ y $v_{rel} = v_{ij}$. Es interesante volver a la Figura 6 para visualizar cómo se van a comportar estos vectores p_{ij} y v_{ij} . En ella se muestra un ejemplo donde el robot está en rumbo de colisión con el obstáculo, por lo que v_{ij} apunta a la espalda del robot y entra dentro del cono de colisión, lo que se traduce en que $h(x) < 0$.

A continuación enunciaremos una serie de supuestos que serán necesarios más adelante:

Suposición II.1. Sea un conjunto de robots \mathcal{N} , todo robot $i \in \mathcal{N}$ se va a encontrar conectado y podrá compartir información sin interrupción con todo $j \neq i \in \mathcal{N}$.

Suposición II.2. Todas aquellas situaciones en las que el vector $u_{ij} = p_{ij} + \hat{v}_{ij}\sqrt{\|p_{ij}\|^2 - r^2}$ sea paralelo a v_i son inestables. Si el sistema cae en ellas podrá salir rápidamente, siempre y cuando exista al menos una acción de control actuando sobre el robot i o el robot j .

Teniendo en cuenta todas las definiciones que acabamos de introducir, e inspirados por [13, Teorema 1], presentaremos el primer resultado original de este trabajo.

Teorema II.1. — *Dado un par de robots i, j con una dinámica descrita por (II.7), la CBF propuesta en (II.15), con $p_{rel} = p_{ij}$ y $v_{rel} = v_{ij}$ definidos por (II.16) y (II.17), es válida para el conjunto \mathcal{D} si se cumplen las suposiciones II.1 y II.2.*

Demostración. En primer lugar, teniendo en cuenta (II.18) y que $\partial_x\|v_{ij}\| = v_{ij}^T(\partial_x v_{ij})/\|v_{ij}\|$, calculamos la derivada de (II.15):

$$\begin{aligned} \dot{h} &= \dot{p}_{ij}^T v_{ij} + p_{ij}^T \dot{v}_{ij} + p_{ij}^T \dot{p}_{ij} \frac{\|v_{ij}\|}{\sqrt{\|p_{ij}\|^2 - r^2}} + v_{ij}^T \dot{v}_{ij} \frac{\sqrt{\|p_{ij}\|^2 - r^2}}{\|v_{ij}\|} \\ &= \underbrace{p_{ij}^T v_{ij} + p_{ij}^T \dot{p}_{ij} \frac{\|v_{ij}\|}{\sqrt{\|p_{ij}\|^2 - r^2}} + p_{ij}^T \dot{v}_j + v_{ij}^T \dot{v}_j \frac{\sqrt{\|p_{ij}\|^2 - r^2}}{\|v_{ij}\|}}_{\mathcal{L}_f h(x)} \\ &\quad \underbrace{- p_{ij}^T \dot{v}_i - v_{ij}^T \dot{v}_i \frac{\sqrt{\|p_{ij}\|^2 - r^2}}{\|v_{ij}\|}}_{\mathcal{L}_g h(x)u}. \end{aligned} \tag{II.19}$$

Siempre y cuando se cumpla la Suposición II.1, el robot j podrá compartir con i su información sobre v_j y \dot{v}_j , de modo que i dispondrá de todas las variables necesarias para calcular \dot{h} . Dado que $\dot{p}_{ij} = v_{ij}$, entonces se puede identificar $\mathcal{L}_g h(x)$ directamente en (II.19), basta con tomar todos aquellos términos en los que aparece \dot{v}_i , pues es el que introduce la acción de control $u = \omega_\theta^i$:

$$\mathcal{L}_g h(x) = \left(p_{ij} + \hat{v}_{ij} \sqrt{\|p_{ij}\|^2 - r^2} \right)^T \begin{bmatrix} -v \sin \theta \\ v \cos \theta \end{bmatrix} \tag{II.20}$$

Una vez llegado a este punto, conviene analizar los escenarios para los que $\mathcal{L}_g h(x) = 0$:

- 1) $p_{ij} + \hat{v}_{ij} \sqrt{\|p_{ij}\|^2 - r^2} = 0$, no es posible. En primer lugar, no podemos tener $p_{ij} = 0$, pues necesariamente se tiene que cumplir que $\|p_{ij}\| > r$, si no fuera así, entonces el robot se encontraría dentro del obstáculo. Por otro lado, tenemos que la magnitud de $\|p_{ij}\|$ siempre va a ser mayor que $\sqrt{\|p_{ij}\|^2 - r^2}$, por lo que es imposible que $\hat{v}_{ij} = -p_{ij} / \sqrt{\|p_{ij}\|^2 - r^2}$.
- 2) $p_{ij} + \hat{v}_{ij} \sqrt{\|p_{ij}\|^2 - r^2}$ es perpendicular a \hat{v}_i , lo cual no puede suceder si se tiene en cuenta la Suposición II.2.

Con todo esto, se puede llegar a la conclusión de que $\mathcal{L}_g h(x)$ nunca se va a anular, por lo que la $h(x)$ propuesta en (II.15) es una CBF válida en \mathcal{D} . □

Resulta interesante destacar que Teorema II.1 tiene en cuenta que la dirección del vector velocidad del obstáculo puede variar. Esta es una de las principales virtudes de nuestro resultado, pues permite diferenciarlo con respecto a otras soluciones de la literatura [13], [14]. Si no se hubiera tenido en cuenta que $\dot{v}_j \neq 0$, entonces las condiciones del Teorema II.1 no se cumplirían para enjambres de robots en el seguimiento de caminos, pues el \dot{h} calculado no representaría la evolución real de h .

Dado que el Teorema II.1 se cumple para todo $x_i \in \mathcal{D}$, aunque inicialmente el estado del sistema se encuentre fuera de \mathcal{C}_{ij} , siempre se podrá diseñar una función κ que le haga tender a la región segura de forma exponencial (consultar [13]). No obstante, hay que tener en cuenta que, si nuestro sistema no cumple la Suposición II.2, entonces el Teorema II.1 no tiene ninguna validez. Si dicha suposición no se cumpliera, dado que $\mathcal{L}_g h$ se podría anular, en aquellas situaciones donde $\dot{h} < 0$ entonces el C3BF-QP (II.13) no sería factible, ya que la entrada al sistema u no tendría ningún efecto sobre la condición (II.12).

Incluso en los casos donde la Suposición II.2 se cumple, si el estado del sistema se encuentra en un entorno próximo a la singularidad, el valor de $\mathcal{L}_g h$ será muy pequeño, de modo que se podrían demandar acciones de control tan agresivas que $u^* \notin \mathbb{U}$, por lo que el problema C3BF-QP volvería a no ser factible.

En las siguientes secciones, se demostrará de forma numérica y experimental la validez de la Suposición II.2. Comprobaremos que el propio controlador encargado de alinear a los robots con el camino también se encarga de alejar rápidamente a los robots de la singularidad.

No obstante, hay que tener en cuenta que un buen diseño de κ nos puede mantener alejados de este tipo de singularidades. Cuando $L_g h = 0$, tenemos que

$$\begin{aligned} \dot{h}(x) + \kappa(h(x)) &= \mathcal{L}_f h(x) + \kappa(h(x)) \\ &= \frac{\|v_{ij}\|}{\sqrt{\|p_{ij}\|^2 - r^2}} (h(x_i) + \kappa(h(x_i))) + \left(p_{ij}^T + \hat{v}_{ij} \sqrt{\|p_{ij}\|^2 - r^2} \right) \dot{v}_j. \end{aligned}$$

Siempre que $x_i \in \mathcal{C}_{ij}$, entonces $h > 0$, de modo que únicamente nos tendremos que preocupar por los términos que van multiplicados por v_j . Existirán muchas situaciones donde $p_{ij}^T v_j > 0$ por

lo que, dado $\|p_{ij}\| > \sqrt{\|p_{ij}\|^2 - r^2}$, tendremos directamente que $\dot{h} > 0$. Cuando $p_{ij}^T v_j < 0$, si suponemos que $\|v_j\|$ nunca será excesivamente grande, como $h > 0$, siempre podremos diseñar una función κ que opague a los términos negativos, de modo que $\dot{h} + \kappa > 0$.

II-D. Garantizando la ausencia de colisiones en enjambres de n rovers con C3BFs

Una vez contamos con un controlador que nos permite evitar colisiones entre dos *rovers*, llega el momento de empezar a pensar en cómo adaptarlo para enjambres de robots.

Cuando aparece más de un obstáculo en el plano, el C3BF-QP (II.13) ha de reformularse añadiendo una condición más para cada uno de los nuevos obstáculos. En estos casos, la C3BF que hemos diseñado para (II.7) puede dar lugar a situaciones en las que el C3BF-QP no sea factible. Si, por ejemplo, uno de las condiciones exigiera una señal de control $u_{safe} > 0$ y otra una $u_{safe} < 0$, entonces el problema de optimización no tendría solución.

En esta sección, nos dedicaremos a diseñar una señal de control u_{safe}^i , para todo robot i contenido en el plano, que nos permita solucionar el Problema II.1. A continuación introduciremos una serie de enunciados que, posteriormente, darán paso al segundo resultado importante de este trabajo.

Definición II.3 (Robot alineado con un camino \mathcal{P}). — *Dado un robot i siguiendo un camino \mathcal{P} , entendemos que se encuentra alineado con \mathcal{P} cuando $\hat{v}_i^T \hat{\tau}(p_i) > \delta \in (0, 1]$.*

Suposición II.3 (Saturación no alcanzable). Supongamos cierto robot i con una dinámica descrita por (II.7). Dada una señal de control u_{ref}^i que permite seguir cierto camino \mathcal{P} , y una u_{safe}^{ij} dada por II.14 que permite a i esquivar a otro robot j . Siempre ocurrirá que la señal de control $u^* = u_{ref}^i + u_{safe}^{ij}$ se encuentra contenida en \mathbb{U} .

Suposición II.4 (v_{ij} siempre sobre un mismo lado del cono). Supongamos un conjunto de robots \mathcal{N} siguiendo cierto camino \mathcal{P} en el mismo sentido de giro. Para todo $i, j \in \mathcal{N}$ alineado con \mathcal{P} , siempre que $\|v_i\|$ sea mayor que $\|v_j\|$, se cumplirá que

- $(p_{ij}E)^T v_{ij} < 0$ si \mathcal{P} se recorre en sentido horario.
- $(p_{ij}E)^T v_{ij} > 0$ si \mathcal{P} se recorre en sentido antihorario.

Proposición II.1. — *Sea un conjunto de robots \mathcal{N} y una señal de control $u_{ref}^i \neq 0$, encargada de coordinar al robot $i \in \mathcal{N}$ con el resto del enjambre para seguir el camino \mathcal{P} en un mismo sentido. Dado un u_{safe}^{ij} generado por (II.14), asumiendo que las Suposiciones II.3 y II.4 se cumplen, y que se cumplen las condiciones del Teorema II.1, entonces*

$$u_{safe}^i = \begin{cases} \max & (\{u_{safe}^{ij} : (p_{ij}E)^T v_{ij} < 0 \wedge j \neq i \in \mathcal{N}\}) & \text{si } u_{ref}^i < 0 \\ \min & (\{u_{safe}^{ij} : (p_{ij}E)^T v_{ij} > 0 \wedge j \neq i \in \mathcal{N}\}) & \text{si } u_{ref}^i > 0 \end{cases} \quad (\text{II.21})$$

es una señal de control válida para resolver el Problema (II.1).

Demostración. Dado un robot $i \in \mathcal{N}$, si $N > 2$ el C3BF-QP que se muestra en (II.13) pasará a tener la siguiente forma:

$$\begin{aligned} u^*(x, t) = \min_{u \in \mathbb{U}} \|u - u_{ref}(x, t)\|^2 \\ t.q. \quad \mathcal{L}_f h_{ij}(x) + \mathcal{L}_g h_{ij}(x) u_{ij} + \kappa(h_{ij}(x)) \geq 0, \quad \forall j \neq i \in \mathcal{N}. \end{aligned} \quad (\text{II.22})$$

Cabe destacar que este problema dista mucho de (II.13). Al contar con más de una condición, se pueden dar situaciones en las que una condición exija $u > 0$, mientras que otra $u < 0$. De forma general, no se puede asegurar que este problema sea factible, incluso cuando $\mathbb{U} = \mathbb{R}$. No obstante, para todos aquellos casos en los que la Suposición (II.4) se cumpla, nos encontraremos únicamente con dos tipos de situaciones, ambas con solución explícita:

- 1) Dado un $u_{ref}^i < 0$ entonces tendremos que $(p_{ij}E)^T v_{ij} < 0$, para todo $j \neq i \in \mathcal{N}$ tal que $\|v_j\| < \|v_i\|$. Si nos quedamos únicamente con estos casos, entonces todas las condiciones de (II.22) exigirán un $u_{ij} = u_{safe}^{ij} \geq 0$, donde u_{safe}^{ij} viene dado por (II.14), de modo que $u_{safe}^i = \max(u_{ij})$ es una solución factible al problema de optimización.
- 2) Cuando $u_{ref}^i > 0$ tenemos lo mismo, pero esta vez todas las condiciones seleccionadas exigirán $u_{ij} = u_{safe}^{ij} \leq 0$, pues $(p_{ij}E)^T v_{ij} > 0$. En este otro caso, basta con proponer un $u_{safe}^i = \min(u_{safe}^{ij})$.

Siempre y cuando se cumpla la Suposición II.3, esta u_{safe}^i será una solución factible de (II.22).

A continuación, analizaremos si dicha u_{safe}^i cumple con las condiciones que se enuncian en el Problema (II.1):

- 1) Dado un par de robots i, j , la u_{safe}^i propuesta no puede garantizar que $x_i \in \mathcal{C}_i$. No obstante, siempre ocurrirá que $x_i \in \mathcal{C}_{ij}$ o $x_j \in \mathcal{C}_{ji}$, dependiendo de si $\|v_j\| < \|v_i\|$ o $\|v_j\| > \|v_i\|$. En función del caso en el que nos encontremos, el Teorema II.1 nos asegura que $x_i \notin \mathcal{X}_{ij}$ o que $x_j \notin \mathcal{X}_{ji}$. No obstante, según definición de colisión, si $x_i \notin \mathcal{X}_{ij}$ entonces necesariamente $x_j \notin \mathcal{X}_{ji}$. Es decir, no importa si $\|v_j\| < \|v_i\|$ o $\|v_j\| > \|v_i\|$, siempre ocurrirá para todo j que $x_i \notin \mathcal{X}_{ij}$. En base a todo este razonamiento, se puede concluir que u_{safe}^i garantiza, para todo tiempo $t \geq 0$, que $x_i(t) \notin \mathcal{X}_i$.
- 2) Dado que la solución de II.22 es factible, dado un tiempo $T \geq 0$, se puede asegurar que $|e_i(t)| < \epsilon$ para todo $t > T$.

Con todo esto, llegamos a la conclusión de que el u_{safe}^i propuesto es una solución válida para resolver el Problema (II.1). □

Para que la Suposición II.3 sea factible en el caso de los *rovers*, la velocidad de todos los vehículos que adelantan tendrá que ser mayor que cierto umbral. De no ser así, el ángulo de giro de las ruedas ψ , dado por (II.6), alcanzará rápidamente su valor máximo ψ_{max} . Véase que los valores de $u = \omega_\theta$, que exijan un $|\psi| > \psi_{max}$, no se encontrarán contenidos en \mathbb{U} .

Es muy importante destacar que la Suposición II.4 no exige únicamente que todos los robots se encuentren recorriendo un camino \mathcal{P} en el mismo sentido, sino que también es necesario que todo el enjambre de encuentre alineado con \mathcal{P} , con un δ bastante alto. Cuando mayor sea este valor δ , menor es la probabilidad de caer en situaciones donde la suposición no se cumple.

Cuando todos los robots se encuentren alineados con \mathcal{P} , los únicos casos que activarán el controlador CBF serán aquellos en los que un robot i se aproxime **por detrás** a otro robot j , donde siempre se cumplirá que $\|v_i\| > \|v_j\|$. En estas circunstancias, $(p_{ij}E)^T v_{ij}$ es un valor que nos permitirá discernir si el vector v_{ij} se encuentra a un lado u otro del cono. Dicha información es muy valiosa porque, si nosotros podemos asegurar que dicho valor va a ser < 0 cuando el controlador se active, también podemos garantizar que $u_{safe}^i > 0$ para todo robot i . En la sección de simulaciones se podrá visualizar muy bien lo que esta suposición nos quiere decir.

II-E. Simulaciones numéricas

La verificación numérica de los resultados en un trabajo científico es de vital importancia. Proporciona confianza en los resultados, detecta posibles errores, facilita la optimización y el diseño adecuado, y valida experimentalmente los resultados. Al invertir tiempo y recursos en la verificación numérica, los investigadores y científicos pueden asegurarse de que sus contribuciones sean sólidas y fiables, lo que a su vez beneficia a la comunidad científica y a la sociedad en general.

En esta sección, presentaremos un total de tres simulaciones distintas, con robots que seguirán las ecuaciones dinámica de un uniciclo (II.7). Cada una de ellas, se centrará en ilustrar y verificar numéricamente algunos de los resultados originales propuestos en esta metodología. Todo el código con el que han sido generado estas simulaciones puede encontrarse en [15], un repositorio de GitHub creado y mantenido por el autor de este TFM.

Sim. I: Un Rover adelanta a otro que se encuentra parado

La simulación de la Figura 7 tiene como objetivo verificar el Teorema II.1 para un caso entre dos robots trivial, donde la celeridad de unos de los agentes es nula y la del otro es constante. Resulta muy pertinente empezar con este caso concreto, pues nos permitirá ilustrar muy bien el comportamiento de las C3BFs analizando únicamente tres instantes de tiempo concretos:

- $t = 10$: El robot azul comienza a aproximarse al obstáculo, pero el controlador CBF todavía no se encuentra activo porque el valor de Ψ es muy alto. Véase que el vector v_{rel} todavía se encuentra muy lejos del cono de colisión.
- $t = 22$: En este caso, v_{rel} está lo suficientemente cerca del cono como para proyectar un valor de $h(x)$ muy cercano a 0, lo que finalmente se traduce en un $\Psi < 0$ que activa el controlador CBF.
- $t = 35$: El robot azul ha vuelto al camino tras conseguir adelantar al robot verde sin colisionar con el obstáculo virtual (zona gris). Dado que el robot acaba de sobreponerse al obstáculo, v_{rel} se encuentra muy lejos del cono de colisión.

En conclusión, esta simulación verifica, para el caso de dos robots, con uno de ellos estático, que el controlador u_{safe}^i (II.14) logra mantener $x_i \notin \mathcal{X}_i$ haciendo uso de una C3BF. Además, dado que $\dot{h}(x, u^*) + \kappa(h(x)) > 0$ en todo momento, también se verifica el Teorema II.1.

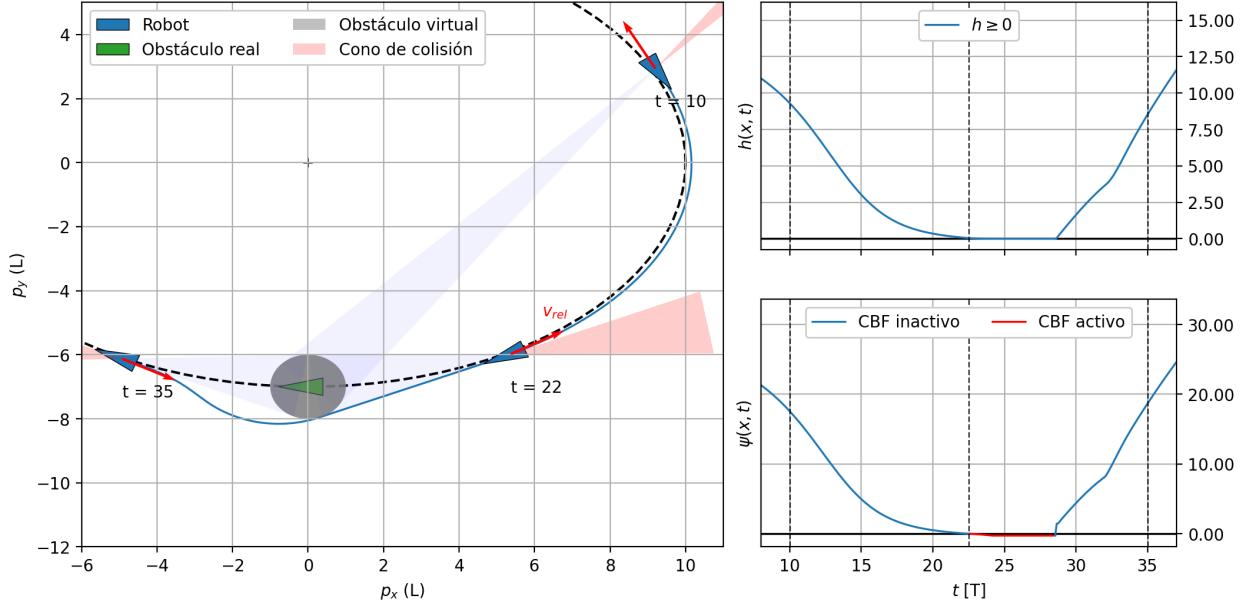


Fig. 7: En esta simulación, tenemos a un robot con dinámica de uniciclo (azul) siguiendo un camino elíptico en el que otro robot (verde) se encuentra parado. En el gráfico de la izquierda se visualiza la simulación, mientras que en los de la derecha se muestra la evolución de $h(x, t)$ y $\Psi(x, t)$.

Sim. II: Un Rover adelanta a otro que se encuentra en movimiento

En la segunda simulación de esta sección (Figura 8), tenemos a dos robots siguiendo una misma elipse, uno con mayor celeridad que el otro. Esta situación es muy pertinente porque visualiza lo que sucede cuando dos robots con celeridades no nulas se adelantan, permitiendo verificar numéricamente el Teorema II.1 para el caso más general posible.

También resulta muy interesante fijarse en la relación entre el vector p_{ij} rotado 90° ($p_{ij}E$) y v_{ij} . En los gráficos inferiores de la observa en la Figura 8 se puede comprobar que el producto escalar entre estos dos vectores varía en función del sentido de giro, lo cual se debe a que vector v_{ij} se encuentra en lados distintos del cono. Dado que el producto escalar entre los dos vectores mencionados anteriormente es siempre mayor que cero o menor que cero, se verifica numéricamente que la Suposición II.4 es factible para los casos con los que estamos trabajando.

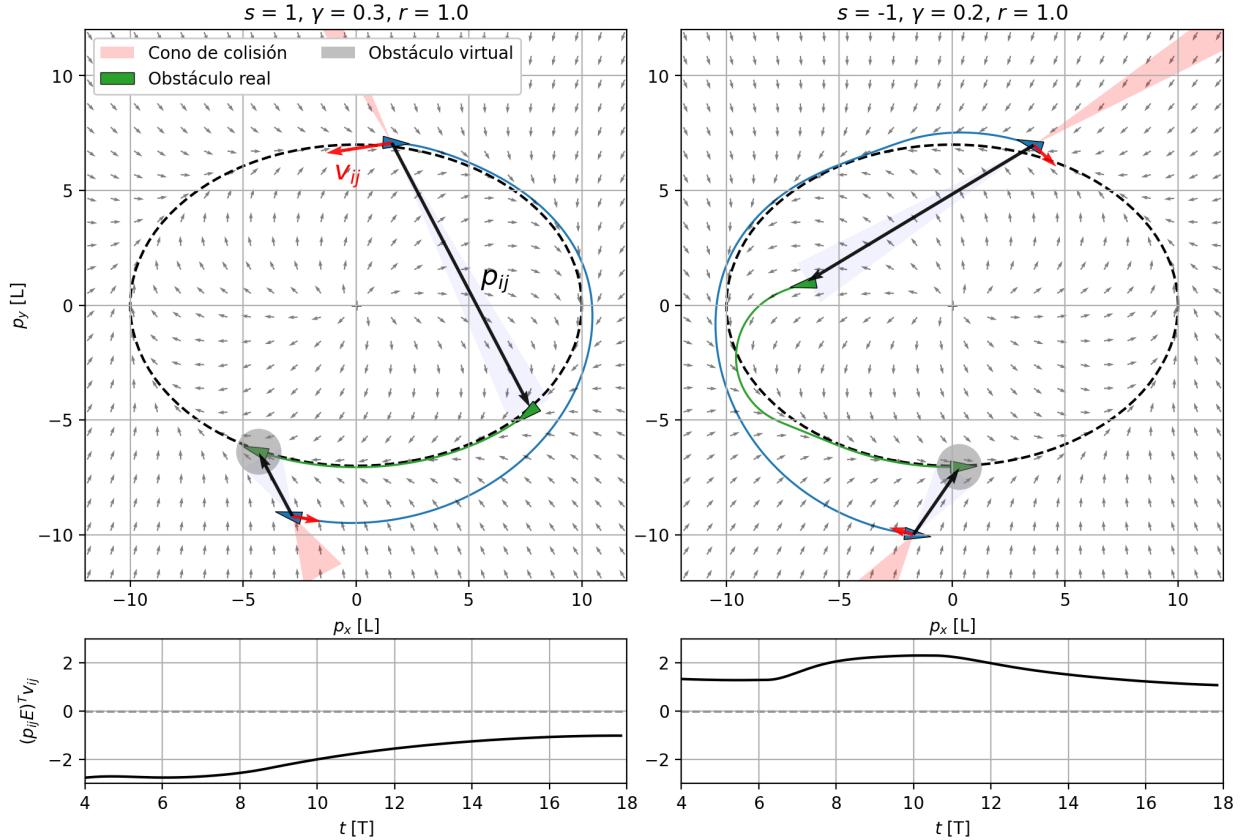


Fig. 8: En estas dos simulaciones, tenemos a un *rover* (azul) adelantando a otro *rover* (verde) que se encuentra en movimiento. En el caso de la izquierda, los rovers se encuentran siguiendo una elipse en sentido horario, mientras que en el caso de la derecha lo hacen en sentido antihorario. En los gráficos inferiores se visualiza el producto escalar entre $p_{ij}E$ y v_{ij} .

Sim. III: Tres rovers siguiendo un mismo camino sin colisionar

En esta última simulación (Figura 9), tenemos a una flota de tres *rovers* con celeridades distintas siguiendo una mismo camino elíptico. Esta misión, en la que ya tenemos más de dos robots involucrados, nos permite ir un paso más lejos, explorando lo que sucede cuando ya empezamos con enjambres de robots.

El instante más interesante de esta simulación se encuentra $t = 19$, pues precisamente los tres robots se están adelantando simultáneamente. Véase, que en este instante de tiempo todos los controladores se encuentran activos, de modo que el robot 2 se encuentra esquivando al 1 y el 3, que es el más rápido, al 1 y 2. Véase que, aunque ambos $\Psi_{1j} < 0$ durante todo el adelantamiento, el robot 1 no activa en ningún momento su controlador CBF, pues es consciente de que el resto de robots ya se están encargando de esquivarle. En conclusión, dado que ninguno de los robots llega a colisionar (véase el gráfico de $\|p_{ij}\|$ en la Figura 9), esta simulación nos permite verificar numéricamente la Proposición II.1.

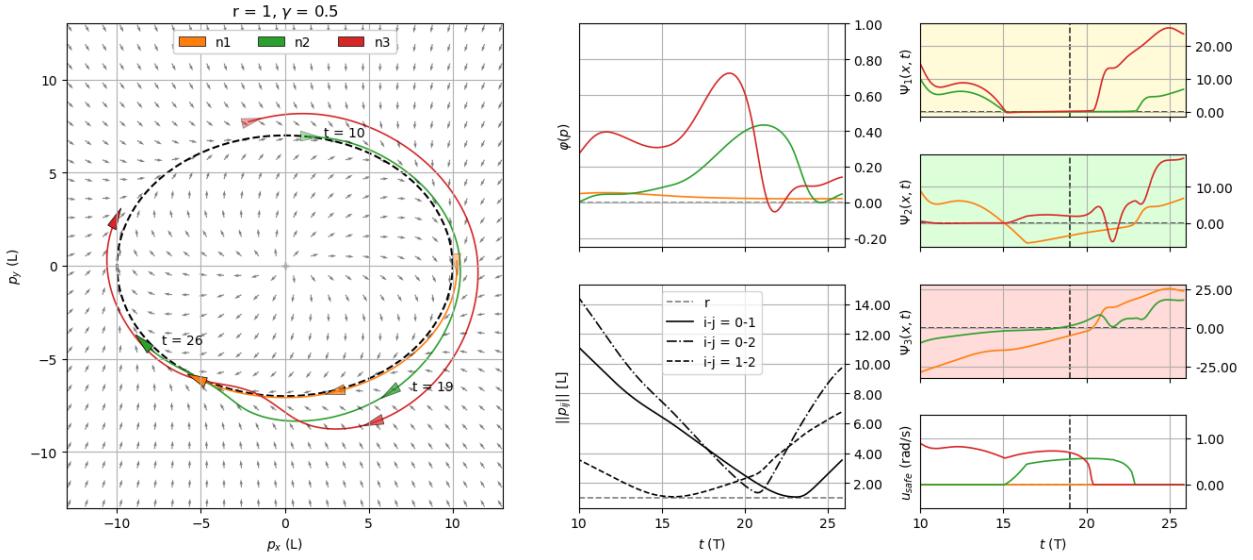


Fig. 9: En esta simulación, se visualiza el adelantamiento simultáneo entre tres *rovers* con distintas celeridades no nulas. El gráfico de la izquierda nos muestra la evolución de la posición y orientación de los robots a lo largo de la simulación, mientras que en los gráficos de la derecha tenemos el error del algoritmo de seguimiento de caminos $\varphi(p)$, la distancia entre los distintos agentes $\|p_{ij}\|$, todos los ψ_{ij} y la acción de control u_{safe}^i que actúa sobre cada robot. Véase que $\|p_{ij}\|$ es siempre mayor que el radio de colisión r .

II-F. Experimentos

Con el fin de verificar experimentalmente todos los resultados originales propuestos en esta metodología, el autor de este TFM construyó desde cero una flota de robots, compuesta por tres *rovers* de radio control con una dinámica que se puede modelar de forma simplificada por (II.7). En esta sección, detallaremos el diseño de los robots y discutiremos sobre los resultados obtenidos en una prueba de campo.

II-F.1. Diseño y construcción de una flota autónoma de rovers

Los *rovers* que hemos diseñado son bastante sencillos a nivel de hardware (ver Figuras 10 y 11). Cada robot cuenta con un controlador principal, encargado de procesar todo el código y gestionar las entradas y salidas del resto de dispositivos. Dicho controlador, además del microprocesador y de todas las interfaces E/S, también cuenta con una IMU y una etapa de potencia capaz de alimentar a varios servos y modular distintas señales PWM.

El mismo controlador, también alimentará y administrará varios dispositivos externos: GPS para geolocalizar a los robots y poder calcular la distancia relativa entre ellos; radio para que los robots se comuniquen entre ellos y, además, manden telemetría a la estación de tierra; y una receptora RC que permitirá al operador mandar comandos manuales desde una emisora RC. A toda esta lista de componentes, hay que sumar el chasis, que trae un motor con su variador y un servo de dirección. Para alimentar a la etapa de potencia del controlador y al variador del motor principal, utilizaremos una batería LiPo.

A continuación, listaremos el nombre concreto de todos los componentes utilizados:

- Controlador: Matek Systems - F405-WMN
- Emisoras RC: Futaba - T16SZ SBUS & Free Sky - X20
- Receptoras RC: Futaba - R6303SBE 3 Channel 2.4Ghz & Free Sky - ACCESS RX6R
- Radio: XBee Pro S1
- GPS: Matek Systems - M10Q-5883 (Ublox chip)
- Batería: LiPo Gens ACE Soaring 2s 7.4V 2200mAh 30C
- Chasis: HPI Racing - Crawler King Ford Raptor 110 Crawler
- Motor: HPI Racing - 55T Crawler motor
- Servo de dirección: HPI Racing - SS-20WR Waterproof Servo - 6.5kg/6.0V



Fig. 10: En esta imagen se muestra a la flota completa de *rover*. A la izquierda tenemos la estación de tierra (PC + radio XBee) y a la derecha los tres *rovers*, junto a sus emisoras de radio control.

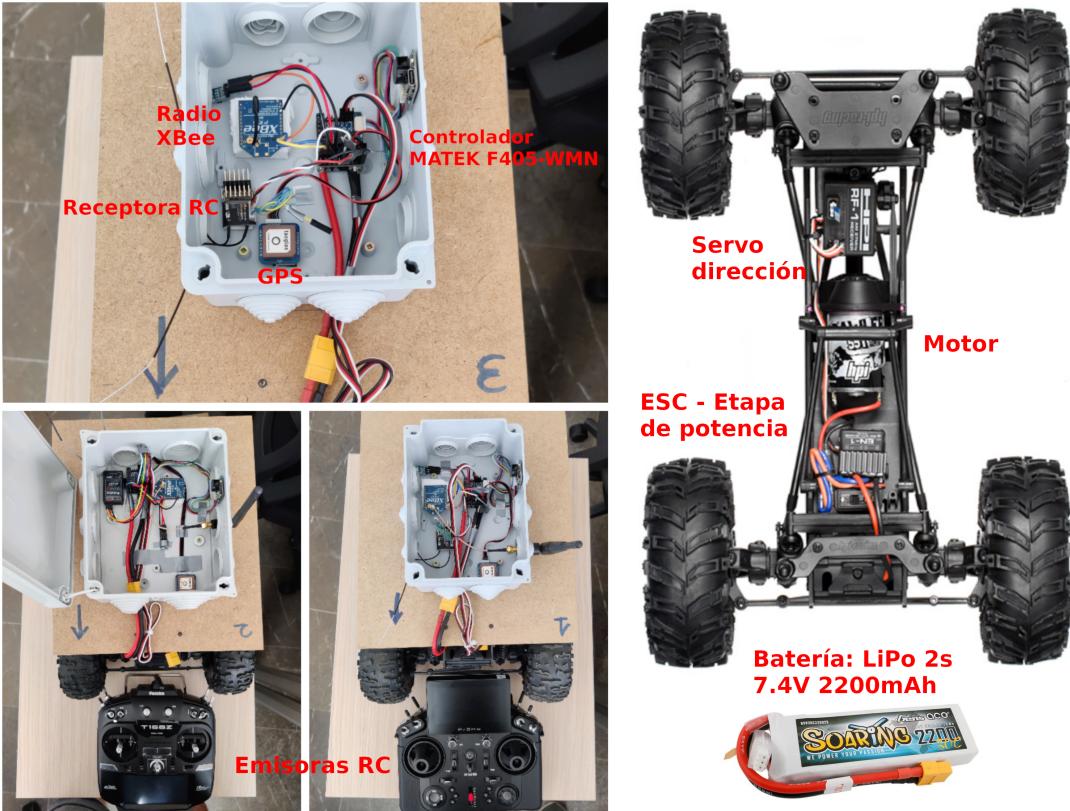


Fig. 11: En esta figura se muestra la electrónica de todos los *rovers*. A la izquierda, tenemos el contenido de la caja estanca, mientras que a la derecha se pueden visualizar todos los componentes de chasis. Cada uno de estos *rover* consta de un controlador principal al que se conecta el variador del motor, receptor RC, GPS, servo de dirección y, por supuesto, batería.

II-F.2. Integración distribuida del algoritmo de seguimiento de caminos y resultados

Todo el software de los robots se ha desarrollado dentro del marco de Paparazzi UAS [16], un proyecto de hardware y software libre que cuenta con gran cantidad de drivers y herramientas para programar y trabajar con robots no tripulados.

Trabajar dentro de este entorno nos facilitó enormemente la tarea de integrar todos los subsistemas de cada *rover*. No obstante, el proyecto no disponía de drivers para el controlador (Matek F405 WMN) y la IMU (ICM 42688-P) que nosotros habíamos seleccionado, por lo que el autor de este TFM se encargó de desarrollarlos. Además de estos driver, el autor también contribuyó a mejorar la estación de tierra implementando una herramienta para visualizar los campos vectoriales de guiado [17] y diseñó los sistemas de guiado y navegación para rovers con volante [18].

El controlador CBF se ha integrado en los robots de forma completamente distribuida. Es decir, son los propios robots los que se encargan de compartir entre ellos toda la información necesaria, haciendo uso de sus radios. La estación de tierra (ver Figura 10) únicamente se utiliza para visualizar la telemetría y mandar órdenes a los robots, por lo que no es fundamental para que el enjambre pueda llevar a cabo su misión de forma autónoma.

En las Figuras 12, 13 y 14 se pueden visualizar los instantes más relevantes de la prueba de campo realizada con la flota de *rover*. En este experimento, a todos los robots se les ha indicado que sigan el mismo camino elíptico en sentido horario. Una vez que todos los robots se encuentren en modo asistido, el ángulo de giro de las ruedas vendrá modulado automáticamente por el controlador de a bordo, mientras que la celeridad será controlada manualmente desde una emisora RC. Téngase en cuenta que, en este modo asistido, tanto el controlador GVF (con $k_e = k_d = 1$) como el CBF (con $r = 2$ y $\gamma = 0.4$) se encontrarán activos.

Por regla general, en la telemetría recogida por la estación de tierra podemos observar un comportamiento muy similar al de las simulaciones numéricas. El controlador CBF logra mantener a los *rovers* fuera de la zona de colisión siempre que la acción de control u_{safe} exigida es alcanzable. No obstante, en la realidad hay que tener muy en cuenta la precisión y desviación del GPS, pues no todos los robots tienen por qué estar donde realmente dicen estar. Dada una medida GPS, siempre se tiene que tener en cuenta un radio de incertidumbre de entorno a 1 o 2 metros. Es por esta razón, que al controlador CBF le indicamos un radio de colisión de 2 metros, para intentar lidiar con la imperfección inherente en la medida GPS. No obstante, a pesar de tomar esta precaución, la imprecisión de las medidas reales nos sigue pudiendo llevar a situaciones en las que, aunque la telemetría nos diga que todo va bien, los *rovers* hayan colisionado. Esto es precisamente lo que sucede en el último instante de la Figura 14, donde *rover* 3 ha colisionado en la realidad con el 1 al intentar esquivarlo.

Resulta muy interesante fijarse especialmente en el instante $t = 287$ de la Figura 12, donde el *rover* 2, que se encuentra esquivando al 1, tiene que adaptarse rápidamente para evitar colisionar con el *rover* 3, pues éste ha decidido realizar un giro muy brusco para esquivar al *rover* 1. Es un caso muy bonito, pues además de verificar experimentalmente el Teorema II.1 y la Proposición II.1, también permite ilustrar la robustez de nuestro algoritmo ante casos extremos.

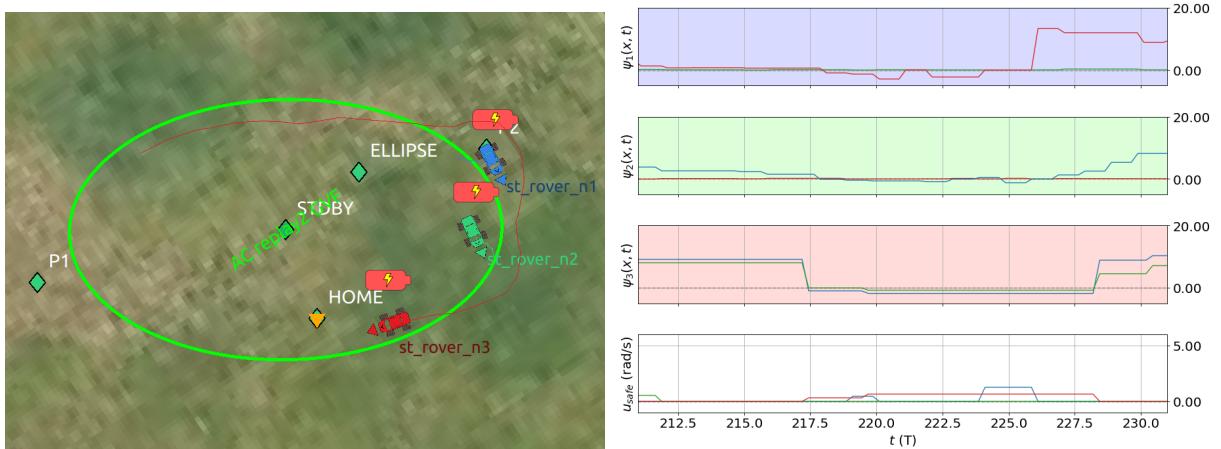


Fig. 12: En esta figura se visualizan los resultados del experimento entre $t = 201$ y $t = 231$. En este intervalo de tiempo, el *rover* 3 se encuentra adelantando al 1 y 2, que se encuentran parados. A la izquierda se visualiza el camino a seguir y la trazada los robots, mientras que en los gráficos de la derecha se visualizan todos los ψ_{ij} y la acción de control u_{safe}^i que actúa sobre cada robot.

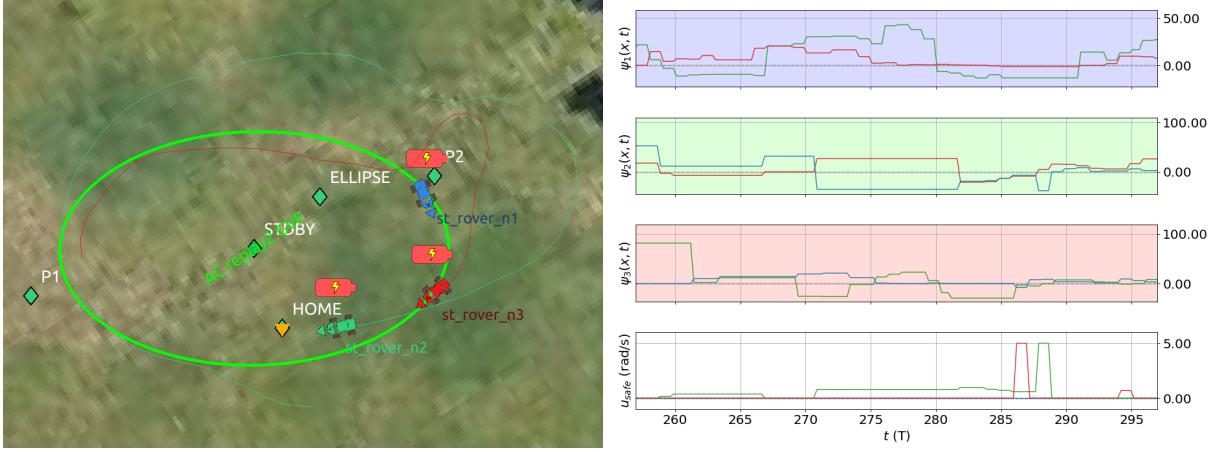


Fig. 13: En esta figura se visualizan los resultados del experimento entre $t = 297$ y $t = 257$. En este intervalo de tiempo, el *rover* 2 se encuentra adelantando al 1 y 3, al mismo tiempo que el *rover* 3 adelanta al 1, que se entra parado. Véase que el *rover* 2 tiene que hacer un giro bastante brusco en $t = 288$ para esquivar al *rover* 3. A la izquierda se visualiza el camino a seguir y la trazada los robots, mientras que en los gráficos de la derecha se visualizan todos los ψ_{ij} y la acción de control u_{safe}^i que actúa sobre cada robot.

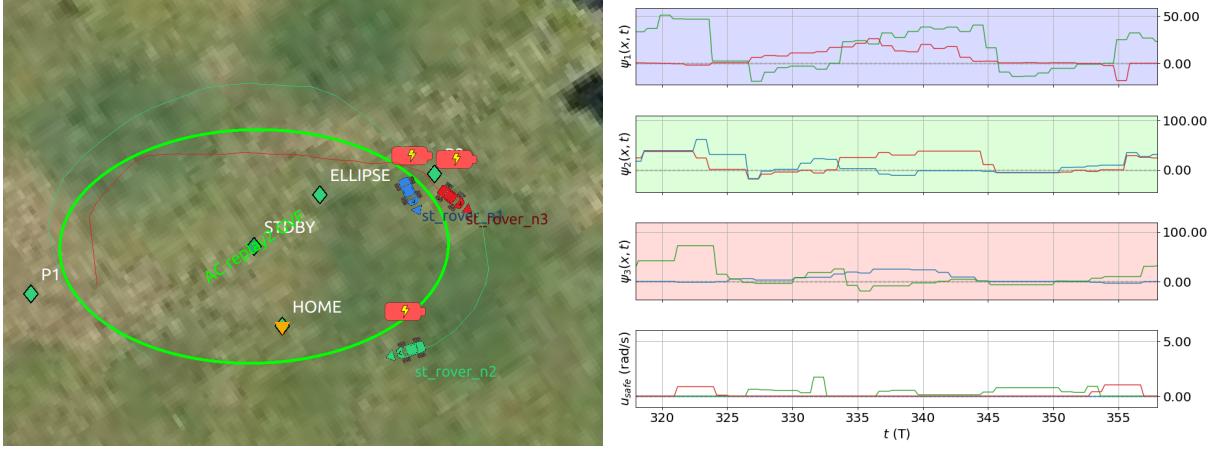


Fig. 14: En esta figura se visualizan los resultados del experimento entre $t = 358$ y $t = 318$. En este intervalo de tiempo, el *rover* 2 se encuentra adelantando al 1 y 3, al mismo tiempo que el *rover* 3 adelanta al 1, que se entra parado. A la izquierda se visualiza el camino a seguir y la trazada los robots, mientras que en los gráficos de la derecha se visualizan todos los ψ_{ij} y la acción de control u_{safe}^i que actúa sobre cada robot.

II-G. Conclusiones

A lo largo de esta metodología, hemos presentado una serie de resultados originales que permiten resolver de forma robusta y efectiva el problema de garantizar la ausencia de colisiones en el seguimiento de caminos con enjambres robóticos. Además de verificar numéricamente todos estos resultados teóricos, se ha construido toda una flota de *rover* que han permitido verificar experimentalmente la viabilidad práctica de nuestro algoritmo.

De cara a un trabajo futuro, se podría pensar en tratar de generalizar nuestros resultados lo máximo posible, intentando deshacernos de todas las suposiciones que limitan las capacidades prácticas de nuestro algoritmo. Si se lograra alcanzar este punto, estaríamos hablando de una solución completamente novedosa y con una enorme cantidad de aplicaciones reales.

III. METODOLOGÍA PARA LOCALIZAR DE FORMA RESILIENTE LA FUENTE DE UN CAMPO ESCALAR CON ENJAMBRES ROBÓTICOS

La búsqueda de fuentes en campos escalares (*Source-Seeking*) se considera un problema fundamental dentro del campo de la robótica de enjambres, debido a su enorme potencial para resolver algunos de los desafíos más actuales [19]. El objetivo consiste en que un equipo de robots sea capaz de coordinarse efectivamente para detectar y rodear fuentes de, por ejemplo, productos químicos, contaminación o señales de radio. Resolver este problema permitiría llevar a cabo misiones de persistencia (24/7) en áreas extensas, cambiando drásticamente la forma que tenemos actualmente de monitorizar el medio ambiente, realizar operaciones de búsqueda y rescate, o ejecutar agricultura de precisión [20]–[24].

Como ya comentamos en la introducción, gracias a su alta resiliencia, los enjambres de robots se encuentran hoy en día entre los sistemas multiagente más prometedores. Su naturaleza les permite preservar la funcionalidad frente a condiciones adversas inesperadas y perturbaciones desconocidas. En el contexto de búsqueda de fuentes, con un enjambre se podría asegurar, por ejemplo, que el equipo de robots encuentre la fuente aunque ciertos agentes desaparezcan durante la misión. No obstante, aunque inicialmente todo esto suena muy bien, garantizar formalmente el rendimiento de este tipo de soluciones, mientras se mantiene su viabilidad práctica, no es tarea sencilla. En el mundo real, todo algoritmo aplicable a la robótica de enjambres debe de tener en cuenta dinámicas de robots realistas, dispositivos de comunicación y la escalabilidad [25].

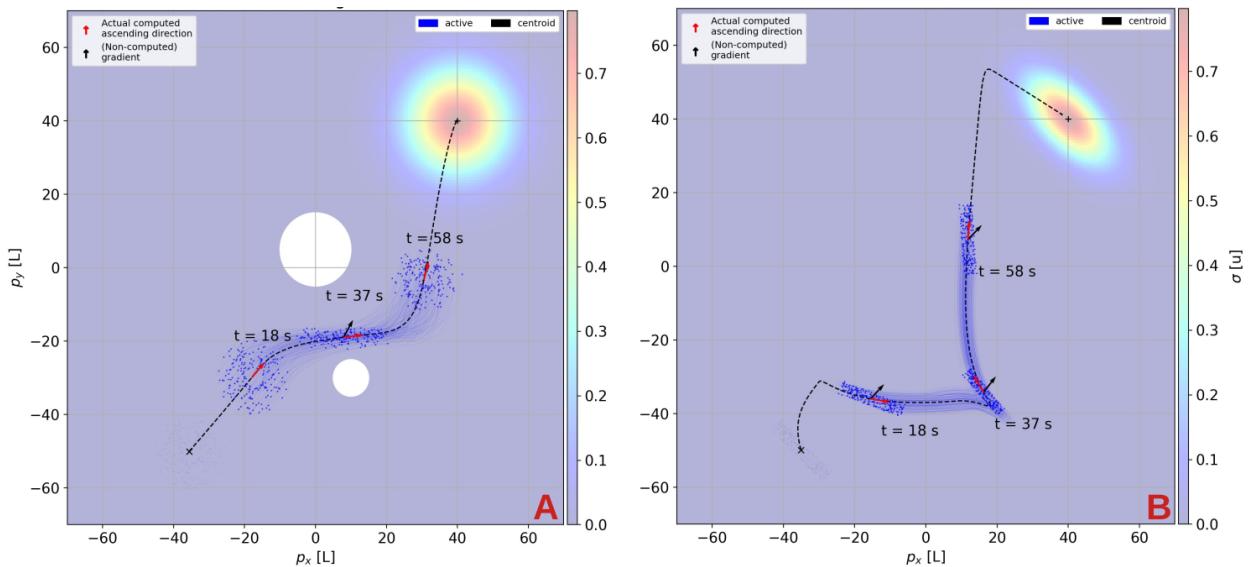


Fig. 15: En esta figura tenemos a dos enjambres de 200 robots con dinámica de integrador simple utilizando nuestro algoritmo de *source-seeking* para ajustar su dirección de movimiento mediante modificaciones en la geometría de su formación. En (A), el enjambre **evita los obstáculos** al pasar de una formación circular a otra rectangular. En (B), el enjambre **rota 90° su dirección de movimiento** realizando una rotación equivalente de su formación rectangular.

En este trabajo, proponemos un algoritmo resiliente y escalable capaz de coordinar a un equipo de robots para alcanzar la fuente de un campo escalar. Demostraremos, de forma rigurosa, cómo el enjambre determina con medidas locales de intensidad una **dirección de ascenso**, en lugar del gradiente, con la que guiar al centroide del enjambre hacia la fuente. Seguir esta estrategia, nos permite dotar al algoritmo de una gran **flexibilidad**, pues no se requiere de una formación específica para el enjambre de robots. Veremos que, gracias a esta característica, la dirección de ascenso calculada reacciona a la relación entre la geometría de la formación y el campo escalar; por lo tanto, actuando únicamente sobre la distribución del enjambre o la forma del campo, seremos capaces de maniobrar al equipo de robots mientras continúan aproximándose simultáneamente a la fuente. Dicho de otro modo, podremos modular indirectamente la dirección de movimiento del enjambre durante la misión para, por ejemplo, evitar obstáculos o adaptarnos a otros factores ambientales (ver Figura 15).

Dicha flexibilidad, será también la encargada de aportar **resiliencia y robustez** al algoritmo. En caso de que ciertos robots desaparezcan o se unan al enjambre, el algoritmo seguirá funcionando, ya que podrá manejar perfectamente la formación resultante. Por esta misma razón, tampoco será un problema gestionar la fusión y división de enjambres de robots, lo que permitirá, por ejemplo, fusionar dos equipos de robots en rumbo de colisión y dividirlos una vez sus direcciones de ascenso individuales ya no estén en conflicto.

III-A. Fundamentos teóricos

III-A.1. Algoritmos existentes de source-seeking: Virtudes y defectos

Los algoritmos del estado del arte que abordan el problema de *source-seeking* ofrecen una serie de virtudes y defectos. No existe la solución perfecta, y por supuesto nuestro algoritmo no es una excepción. Por esta razón, antes de comentar a analizarlo detalladamente, procederemos a contextualizarlo revisando algunas las soluciones existentes. Es decir, resumiremos las posibilidades que otros algoritmos de la literatura tienen para ofrecer, de modo que el lector pueda evaluar sus virtudes y defectos en comparación con nuestra solución. Particularmente, nos centraremos en hablar sobre la adaptabilidad de la formación de robots, la consideración de dinámicas que se ajusten a robots realistas, el rendimiento anticipado en términos de distancia recorrida o trayectorias previstas suaves, o los requisitos de comunicación, como la topología y el ancho de banda.

El primer método ampliamente utilizado se centra en la **estimación del gradiente y el Hessiano**, donde podemos encontrar algunas variaciones. Por ejemplo, los autores en [26], [27] estiman el gradiente de los robots en sus posiciones de forma distribuida, y el acuerdo sobre la dirección común a seguir se obtiene aplicando un controlador de formación basado en la distancia que mantiene la cohesión del enjambre. Sus resultados se demuestran con robots modulados como puntos cinemáticos (integradores simples o dobles), y dejan ver que la estimación del gradiente se vuelve poco confiable si al menos un grupo de robots vecinos muestra una forma degenerada, como una línea, en el plano 2D.

Los autores en [28], [29] estiman el gradiente y el Hessiano de un enjambre de robots en el centroide de una *formación circular*. Su técnica permite el uso de robots con dinámica de uniciclo; no obstante, el algoritmo es muy rígido, ya que necesariamente la geometría de la

formación ha de ser circular, o una esfera en 3D. La misma formación circular resulta de aplicar el algoritmo propuesto en [30], donde los robots también se modelan como uniciclos; sin embargo, en este caso los agentes no necesitan medir sus posiciones relativas, sino sus orientaciones relativas. Finalmente, relacionado con la técnica del gradiente, pero asumiendo que se conoce el campo, en [20] podemos encontrar uno de los algoritmos más pioneros.

La **búsqueda de extremos** es otra técnica ampliamente utilizada para el problema de *source-seeking*, como se analiza en [31], [32]. En estos trabajos, se fuerza a los robots (normalmente solo uno) a realizar movimientos periódicos para realizar una estimación del gradiente no basada en modelos. El método tiene éxito con dinámicas no holonómicas; no obstante, en caso de varios robots, es posible que necesiten intercambiar parámetros estimados y, según la acción de control resultante, las trayectorias finales de los robots podrían alejarse de las deseadas, ya que suelen ser largas y tortuosas con giros bruscos.

Todos los algoritmos mencionados anteriormente para la búsqueda de fuente requieren comunicación (principalmente distribuida), es decir, los robots comparten la intensidad del campo escalar. No obstante, los autores en [33] ofrecen una solución elegante que no requiere compartir la intensidad del campo entre los robots. Sin embargo, es necesario conocer la posición del centroide del enjambre de robots, y un algoritmo para su estimación distribuida podría requerir comunicación dentro del equipo. El método que proponen utiliza un algoritmo de **análisis de componentes principales basado en consenso**, y los robots modulan sus movimientos centrados únicamente en sus mediciones instantáneas del campo; por lo tanto, la formación (variable en el tiempo) de los robots durante la misión está determinada por sus posiciones iniciales y el campo escalar. En otras palabras, la formación no está bajo control.

Concluimos esta breve revisión de la literatura haciendo referencia a un problema muy relacionado, el problema de **cobertura con sensores móviles** [34]. Los algoritmos que abordan este escenario son particularmente buenos para rodear más de una fuente simultáneamente con un enjambre de robots. Mayoritariamente, se basan en explotar la partición de un área con celdas de Voronoi. Sin embargo, hasta ahora dicho rendimiento conlleva un precio más alto que las soluciones típicas de *source-seeking*. En estos algoritmos, el robot (o enjambre) necesita la información (densa o continua) del campo escalar dentro de su celda correspondiente y, además, su configuración eventual depende de la forma de dicha celda, que ha de ser fija o conocida de antemano.

III-A.2. Notación general

Consideremos un equipo de $N \in \mathbb{N}$ robots, donde la posición del robot $i \in \{1, \dots, N\}$ en el espacio cartesiano se representa por $p_i \in \mathbb{R}^m$, donde $m \in \{2, 3\}$ es la dimensión de dicho espacio. Definimos $p \in \mathbb{R}^{mN}$ como el vector apilado que contendrá las posiciones de todos los robots que conforman al enjambre.

Cuando trabajemos con múltiples enjambres, definiremos $N_c \in \mathbb{N}$ como el número de **equipos de robots**. En estos casos, denotaremos al número de robots en el k -ésimo equipo como N_k , y la posición del i -ésimo robot en el k -ésimo grupo como $p_i^k \in \mathbb{R}^m$. Téngase en cuenta que omitiremos el superíndice k por claridad si solo hay un equipo. En cada enjambre, habrá una **unidad de cálculo** que recopilará la información de todos agentes dentro del equipo.

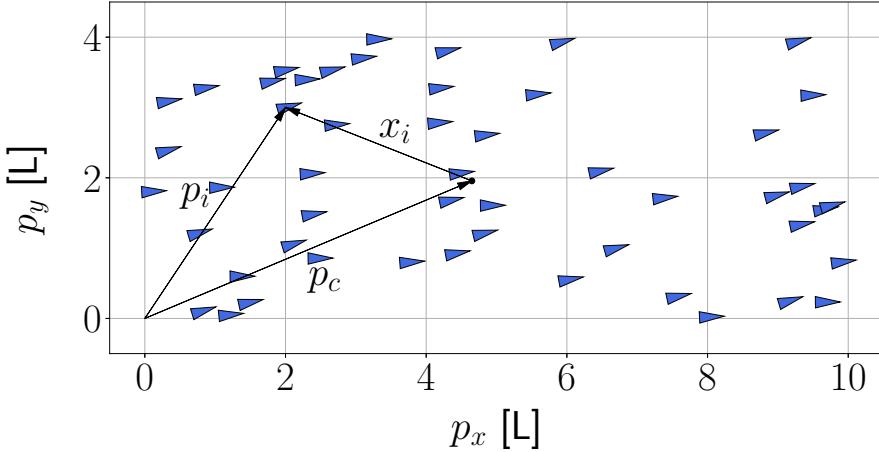


Fig. 16: Despliegue x de un enjambre de robots con centroide en p_c . El vector p_i representa la posición de cada agente con respecto al sistema de referencia y x_i su posición con respecto al centroide.

Nuestro algoritmo de *source-seeking* lo contrastaremos con robots que simularán la dinámica de integrador simple y uniciclo. En el primer caso, los agentes pueden seguir directamente la velocidad guía en 2D y 3D. Nos obstante, para el caso de los uniciclos, será necesario utilizar un controlador secundario para alinear la dirección del robot con la dirección de ascenso. Se puede demostrar que dicho controlador puede ser muy similar al utilizado en la metodología anterior para alinear a los agentes con el GVF. Para más información, consultar [35].

Definimos el centroide de un equipo como $p_c := \frac{1}{N} \sum_{i=1}^N p_i$. Por lo tanto, podemos escribir $p_i = p_c + x_i$, donde $x_i \in \mathbb{R}^m$ para todo $i \in \{1, \dots, N\}$ describe cómo se distribuyen los robots alrededor de su centroide (ver Figura 16).

Definición III.1 (Geometría del enjambre). — *El vector apilado $x := [x_1^T, \dots, x_N^T]^T \in \mathbb{R}^{mN}$ denotará la geometría o formación del enjambre. Dicha geometría será **no degenerada** si los vectores $\{x_1^T, \dots, x_N^T\}$ generan el espacio \mathbb{R}^m .*

Teniendo en cuenta esta definición, entonces es necesario imponer que $N > m \geq 2$ para que la geometría del enjambre no sea degenerada. La intensidad de una señal en todo el espacio puede describirse mediante un campo escalar de la siguiente manera.

Definición III.2 (Intensidad de un señal). — *La intensidad de una señal es un campo escalar $\sigma : \mathbb{R}^m \rightarrow \mathbb{R}^+$, dos veces diferenciable y con todas sus derivadas hasta segundo orden acotadas globalmente. En nuestro caso, dicho σ cuenta únicamente con un **máximo** en p_σ y su gradiente en $a \in \mathbb{R}^m$ satisface que $\nabla \sigma(a) \neq 0 \iff a \neq p_\sigma$ y $\lim_{||a|| \rightarrow \infty} \sigma(a) \rightarrow 0$.*

Algunas de las señales que se ajustan a esta definición son las distribuciones Gaussianas, o todas aquellas señales que disminuyen desde la fuente de acuerdo con la ley de potencia $x^{-\alpha}$, con $2 \leq \alpha \leq 3$, modelo que comienza a ser aplicable después de considerar una distancia mínima fija x_{\min} desde el origen [36]. La intensidad de este tipo de señales se puede representar, por ejemplo, como el módulo del campo electromagnético, la concentración de un contaminante o la radiación de calor. En particular, resulta interesante tener en cuenta las distribuciones cuadráticas (leyes de potencia x^{-2}), o el logaritmo de una distribución Gaussiana, pues son muy relevantes dentro del mundo físico que observamos.

En este trabajo, definiremos el gradiente como un vector columna $\nabla \sigma(\cdot) \in \mathbb{R}^m$. Véase que,

según nuestra definición de señal, tenemos que

$$\|\nabla\sigma(a)\| \leq K \quad \text{y} \quad \|H_\sigma(a)\| \leq 2M, \quad \forall a \in \mathbb{R}^m, \quad (\text{III.1})$$

donde $K, M \in \mathbb{R}^+$, y H_σ es el Hessiano del campo escalar σ , es decir, $H_\sigma(a) \in \mathbb{R}^{m \times m}$; por lo tanto, a partir de la expansión de la serie de Taylor de σ en $a \in \mathbb{R}^m$ [37, Teorema 5.15], se sigue que

$$|\sigma(a) - \sigma(b) - \nabla\sigma(a)^T(a - b)| \leq M\|a - b\|^2, \quad \forall b \in \mathbb{R}^m. \quad (\text{III.2})$$

III-B. Teoría de Grafos

Los algoritmos que proponemos para resolver el problema de *source-seeking* y estimar el centroide de un equipo van a ser implementados de forma distribuida. Por lo tanto, necesitamos introducir algunas nociones de Teoría de Grafos [38] para poder definir las relaciones entre los robots de manera precisa. Consideremos un grupo de N robots, entonces un **grafo** $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consta de dos conjuntos no vacíos: el conjunto de nodos $\mathcal{V} = \{1, \dots, N\}$ donde cada nodo i corresponde al robot i , y el conjunto ordenado de aristas $\mathcal{E} \subseteq (\mathcal{V} \times \mathcal{V})$ que define las comunicaciones o percepciones entre pares de robots diferentes. Para una arista arbitraria $\mathcal{E}_k = (\mathcal{E}_k^{\text{tail}}, \mathcal{E}_k^{\text{head}})$, llamamos a su primer y segundo elemento la **cola** y la **cabeza** respectivamente. El conjunto \mathcal{N}_i que contiene a los vecinos del nodo i se define como $\mathcal{N}_i := \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$.

EN este trabajo, únicamente abarcaremos el caso especial de grafos **no dirigidos**, donde todas las aristas \mathcal{E}_k se consideran **bidireccionales**; es decir, si $(i, j) \in \mathcal{E}$ entonces necesariamente $(j, i) \in \mathcal{E}$. Para un grafo no dirigido, elegimos solo una de estas dos direcciones arbitrarias entre los nodos i y j , lo que nos permitirá construir la **matriz de incidencia** $B \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}|}$ de \mathcal{G} de la siguiente manera:

$$b_{ik} := \begin{cases} +1 & \text{si } i = \mathcal{E}_k^{\text{tail}} \\ -1 & \text{si } i = \mathcal{E}_k^{\text{head}} \\ 0 & \text{en otro caso.} \end{cases} \quad (\text{III.3})$$

Para un grafo no dirigido, la **matriz Laplaciana** $L \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ [38, Capítulo 6] se puede calcular como

$$L = BB^T. \quad (\text{III.4})$$

Si el grafo \mathcal{G} es **conectado** [38, Capítulo 3], entonces la matriz Laplaciana L tiene un único valor propio igual a cero, cuyo vector propio asociado es $\mathbf{1}_N$, ya que $B^T \mathbf{1}_N = 0$; por lo tanto, L es semidefinida positiva.

III-C. Formalización del problema

Una vez introducida toda la notación necesaria, ya nos encontramos en disposición de formalizar el problema de *source-seeking* con enjambres de robots.

Problema III.1 (*Source-seeking* con enjambres de robots). — *Dada una señal σ , N_c equipos de robots y una constante $\epsilon > 0$, encontrar acciones de control para todos los robots de manera que $\|p_c^k(t) - p_\sigma\| < \epsilon$, $\forall t \geq T$, $\forall k \in \{1, \dots, N_c\}$ para algún tiempo finito $T \in \mathbb{R}^+$.*

III-D. Herramientas de *source-seeking*

En esta sección, presentaremos una serie de herramientas que nos permitirán diseñar una solución autónoma al problema de *source-seeking*. Las herramientas que introduciremos en esta sección nos permitirán: calcular de forma distribuida la dirección de ascenso, analizar la observabilidad y sensibilidad de la dirección de ascenso con respecto a una formación x , coordinar a múltiples equipos para seguir una misma dirección de ascenso y, por último, realizar una estimación distribuida del centroide de un equipo utilizando a los robots que lo componen.

A continuación, resumiremos la información necesaria para que los robots implementen nuestra solución autónoma:

- El robot i mide la intensidad de la señal $\sigma(p_i)$.
- Dado un grupo codificado como un grafo \mathcal{G} , el robot i mide en su sistema de coordenadas local la posición relativa $(p_i - p_j) = (x_i - x_j)$, $(i, j) \in \mathcal{E}$. Ya veremos que, siempre y cuando \mathcal{G} esté conectado, esta información es suficiente para estimar el centroide de forma distribuida.
- La unidad de cálculo del k -ésimo equipo necesita todas las $\sigma(p_i^k)$ y todas las posiciones relativas de los robots con respecto al centroide, es decir, $x_i^k = (p_i^k - p_c^k)$, $\forall k \in 1, \dots, N_k$. Por lo tanto, hay una topología en estrella para el cálculo de la dirección de ascenso. Si el número de robots de un grupo es muy grande, entonces éste se podrá dividir en una red arbitraria pero conectada de subgrupos con topología en estrella, para así calcular una dirección de ascenso de manera distribuida con un algoritmo de consenso estándar.

Cabe destacar que al enjambre nunca se le va a requerir conocer su posición con respecto a un sistema de referencia absoluto. Este es un factor muy importante, pues en la práctica hace que no sea necesario hacer uso de sistemas de geolocalización.

Dado el requisito de medir posiciones relativas entre robots, la formación geométrica deseada x se puede lograr con controladores de formación basados en desplazamiento [39]–[41]. No entraremos mucho en detalle sobre cómo funcionan este tipo de controladores, pues no resultan ser una contribución original de este trabajo. No obstante, hay que tener en cuenta que será necesario su implementación para poder llevar ciertas herramientas de nuestro algoritmo autónomo de *source-seeking* al mundo real.

A continuación, definiremos formalmente lo que entendemos por dirección de ascenso en un punto relativo a una señal σ .

Definición III.3 (Dirección de ascenso). — *Un vector no nulo $v \in \mathbb{R}^m$ es una dirección de ascenso en un punto $p \in \mathbb{R}^m$ con respecto a una señal $\sigma : \mathbb{R}^m \rightarrow \mathbb{R}^+$ si y solo si $\nabla\sigma(p)^T v > 0$.*

III-D.1. La dirección de ascenso

Los autores en [28] demuestran que

$$\hat{\nabla}\sigma(p_c) = \frac{m}{ND^2} \sum_{i=1}^N \sigma(p_i)(p_i - p_c), \quad (\text{III.5})$$

es una estimación del gradiente de la señal, en el espacio 2D, en el centro de una circunferencia con radio D , si al menos tres robots están igualmente espaciados angularmente en la circunferencia. La misma expresión, también se emplea en [29] para una distribución similar en una esfera 3D. A lo largo de esta subsección, veremos esta distribución simétrica particular es solo una condición suficiente para estimar el gradiente.

En los trabajos [28], [29] se ha estudiado con detalle que (III.5) para distribuciones uniformes, en la circunferencia, no obstante, todavía no está claro si ésta sigue siendo una buena estimación del gradiente para distribuciones genéricas. De hecho, sin aparentes cambios importantes, la expresión (III.5) se puede extender significativamente admitiendo cualquier distribución genérica x . En particular, a lo largo de este trabajo demostraremos que bajo algunas condiciones, el vector

$$L_\sigma(p_c, x) = \frac{1}{ND^2} \sum_{i=1}^N \sigma(p_i)(p_i - p_c) = \frac{1}{ND^2} \sum_{i=1}^N \sigma(p_c + x_i)x_i, \quad (\text{III.6})$$

es una dirección de ascenso en p_c , donde ahora $D = \max_{1 \leq i \leq N} |x_i|$. Esta expresión para la dirección de ascenso fue originalmente propuesta en [35], donde se presentaron una serie de resultados muy interesantes que han motivado en gran medida los análisis que llevaremos a cabo en este TFM.

Véase que L_σ puede ser calculada por la unidad de cálculo de cada enjambre, ya que dispone de toda la información necesaria para ello. También conviene destacar que (III.6), a diferencia de (III.5), es una función de una distribución genérica x , y que la dirección de ascenso no necesariamente aproxima el gradiente de la señal (es decir, posiblemente no es paralela al gradiente). Más adelante, veremos que esta propiedad es precisamente una ventaja que nos permitirá maniobrar al enjambre de robots manipulando únicamente x .

A continuación, aproximemos $\sigma(p_c + x_i)$ a una serie de Taylor de orden dos para $|x_i|$ **pequeños**, es decir, $\sigma(p_c + x_i) \approx \sigma(p_c) + \nabla\sigma(p_c)^T x_i$, de modo que $L_\sigma(p_c, x) \approx L_\sigma^0(p_c, x) + L_\sigma^1(p_c, x)$, con

$$\begin{aligned} L_\sigma^0(p_c, x) &= \frac{1}{ND^2} \sum_{i=1}^N \sigma(p_c)x_i, \\ L_\sigma^1(p_c, x) &= \frac{1}{ND^2} \sum_{i=1}^N (\nabla\sigma(p_c)^T x_i) x_i, \end{aligned} \quad (\text{III.7})$$

donde es fácil comprobar que $L_\sigma^0(p_c, x) = 0$ debido a la definición del centroide p_c .

Observación III.1. La elección del factor $\frac{1}{ND^2}$ es irrelevante, ya que la propiedad importante de L_σ^1 es su dirección y no su magnitud. Sin embargo, D^2 hace que L_σ^1 tenga las mismas unidades que el gradiente, de modo que significan lo mismo físicamente [*unidades de señal / unidades de longitud*] = [U/L]. N es simplemente un factor de promediación.

El vector L_σ^1 es interesante porque **siempre** es una dirección de ascenso en el centroide p_c (ver Figura 17). Es decir, (III.7) no requiere que se cumplan condiciones particulares, solo que la distribución x no sea degenerada, tal y como se muestra en el siguiente resultado de [35]:

Lema III.1 ($L_\sigma^1(p_c, x)$ siempre es dirección de ascenso). — *Si la distribución x es no degenerada y $p_c \neq p_\sigma$, entonces $L_\sigma^1(p_c, x)$ siempre es una dirección de ascenso en p_c hacia el máximo p_σ del campo escalar σ .*

Demostración. Si L_σ^1 es siempre una dirección de ascenso, entonces debe cumplir $\nabla\sigma(p_c)^T L_\sigma^1(p_c, x) > 0$, y eso es fácil de comprobar ya que

$$\nabla\sigma(p_c)^T L_\sigma^1(p_c, x) = \frac{1}{ND^2} \sum_{i=1}^N |\nabla\sigma(p_c)^T x_i|^2$$

siempre es positivo si las posiciones relativas x_i en la distribución x abarcan \mathbb{R}^m y $p_c \neq p_\sigma$. \square

Observación III.2. Otra demostración comienza considerando $0 \neq b \in \mathbb{R}^m$ y el siguiente operador $\mathcal{L}(b, x) := \sum_{i=1}^N (b^T x_i) x_i = \sum_{i=1}^N (x_i x_i^T) b$, y dado que $P(x) := \sum_{i=1}^N x_i x_i^T$ es definida positiva para x no degenerada, entonces $b^T \mathcal{L}(b, x) = b^T P Z b > 0$. Reemplazando b por $\nabla\sigma(p_c)$, se puede demostrar el lema anterior. Además, si el resultado es $P = \lambda I_m$, donde $\lambda \in \mathbb{R}^+$, entonces $\mathcal{L}(b, x) = Pb = \lambda b$, lo que significa que produce un vector paralelo a la entrada b . Veremos que si x consta de los vértices de polígonos o poliedros, estas distribuciones tienen esta propiedad (es decir, Lema III.1).

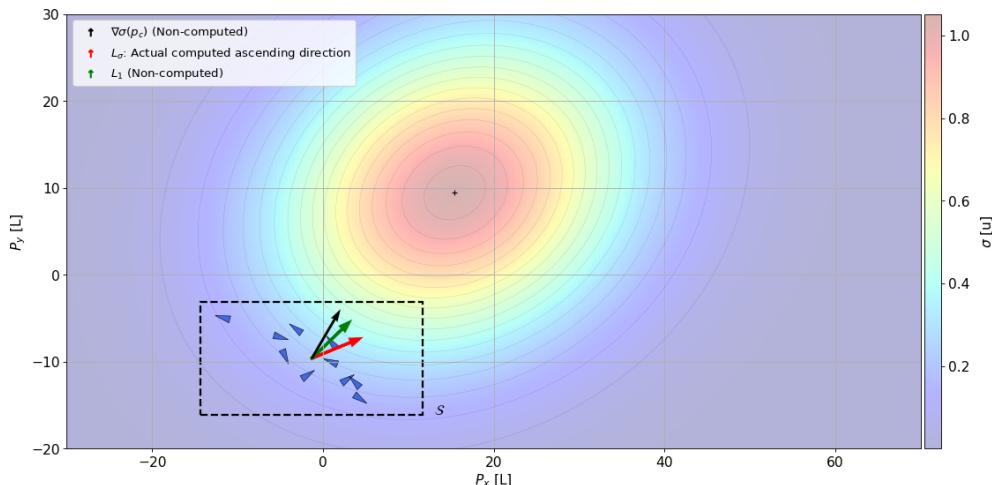


Fig. 17: Para una formación genérica x y señal gaussiana σ , se visualiza $\nabla\sigma(p_c)$, un posible conjunto S , y el cómputo de L_σ y L_σ^1 . Resulta interesante visualizar que ninguno de estos vectores son paralelos.

Recordemos que L_σ^1 no es la aproximación de primer orden de L_σ con respecto a x_i , sino que considera la aproximación de primer orden de σ . Teniendo esto en cuenta, el Lema III.2 conviene contrastarlo con (III.6), que como ya veremos únicamente es una dirección de ascenso cuando se impone cierta condición conservadora. No obstante, dicha condición se derivará teniendo en cuenta que L_σ^1 es siempre una dirección de ascenso. La idea del Lema III.1, motivó al autor de [35] a analizar cómo de rápido diverge L_σ^1 de L_σ en relación a $\|x\|$.

Lema III.2. — *Para una señal σ , la divergencia entre $L_\sigma^1(p_c, x)$ y $L_\sigma(p_c, x)$ depende linealmente de $D := \max_{1 \leq i \leq N} |x_i|$ (es decir, la magnitud máxima de la distribución x). Es decir,*

$$\|L_\sigma(p_c, x) - L_\sigma^1(p_c, x)\| \leq MD,$$

donde M es el límite superior en (III.1). Por lo tanto, $\lim_{D \rightarrow 0} (L_\sigma(p_c, x) - L_\sigma^1(p_c, x)) = 0$.

Demostración. A partir de (III.2), (III.6) y (III.7), es fácil ver que

$$\begin{aligned} \|L_\sigma - L_\sigma^1\| &= \frac{1}{ND^2} \left\| \sum_{i=1}^N \left(\sigma(p_c + x_i) - \sigma(p_c) - \nabla \sigma(p_c)^T x_i \right) x_i \right\| \\ &\leq \frac{1}{ND^2} \sum_{i=1}^N M \|x_i\|^3 \leq MD. \end{aligned}$$

□

De hecho, si D es lo suficientemente pequeño, entonces es seguro que L_σ es una dirección de ascenso, al igual que L_σ^1 . Sin embargo, ¿qué se considera como **pequeño** para señales y despliegues genéricos? Definamos $E := L_\sigma - L_\sigma^1$, de modo que

$$\nabla \sigma(p_c)^T L_\sigma = \nabla \sigma(p_c)^T (L_\sigma^1 + E), \quad (\text{III.8})$$

y consideremos un conjunto compacto $\mathcal{S} \subset \mathbb{R}^m$ con $p_\sigma \notin \mathcal{S}$. Por la definición de σ y su gradiente acotado, sabemos que $\nabla \sigma(p_c)^T L_\sigma^1$ tiene un mínimo $F_{\mathcal{S}}(x)$ que depende del despliegue elegido x y es positivo si x no es degenerado, para todo $p_c \in \mathcal{S}$. Por lo tanto, tenemos $\nabla \sigma(p_c)^T L_\sigma \geq F_{\mathcal{S}}(x) - K_{\mathcal{S}}^+ M_{\mathcal{S}} D$, donde $K_{\mathcal{S}}^+$ y $M_{\mathcal{S}}$ son las normas máximas del gradiente y Hessiano de la señal en \mathcal{S} , respectivamente; por lo tanto, si

$$F_{\mathcal{S}}(x) - K_{\mathcal{S}}^+ M_{\mathcal{S}} D > 0, \quad (\text{III.9})$$

entonces L_σ es una dirección de ascenso en \mathcal{S} . Encontrar el mínimo $F_{\mathcal{S}}(x)$ numéricamente puede ser una tarea ardua, pero podemos acotarlo de manera conservadora utilizando el siguiente resultado de [35].

Lema III.3. — Si x no es degenerado, entonces existe $C(x) > 0$, que solo depende del despliegue del conjunto de robots, tal que

$$\frac{1}{C(x)} \|\nabla \sigma(p_c)\|^2 \leq L_\sigma^1(p_c, x)^T \nabla \sigma(p_c) \leq C(x) \|\nabla \sigma(p_c)\|^2.$$

Demostración. En primer lugar, vemos que el caso trivial $\nabla \sigma(p_c) = 0$ satisface la afirmación. En cualquier otro caso, sabemos por el Lema III.1 que $\nabla \sigma(p_c)^T L_\sigma^1(p_c, x) = \frac{1}{ND^2} \sum_{i=1}^N |\nabla \sigma(p_c)^T x_i|^2 = \frac{1}{ND^2} \nabla \sigma(p_c)^T P(x) \nabla \sigma(p_c) > 0$ para alguna matriz definida positiva $P(x) = \sum_{i=1}^N x_i x_i^T$ ya que x no es degenerado. Por lo tanto,

$$\frac{\lambda_{\min} P(x)}{ND^2} \|\nabla \sigma\|^2 \leq \nabla \sigma^T L_\sigma^1 \leq \frac{\lambda_{\max} P(x)}{ND^2} \|\nabla \sigma\|^2,$$

donde $\lambda_{\min, \max} P(x)$ son los valores propios mínimo y máximo de $P(x)$. Elegimos $C(x) = \max \frac{\lambda_{\max} P(x)}{ND^2}, \frac{ND^2}{\lambda_{\min} P(x)}$, y solo depende del despliegue x .

□

Ahora ya estamos listos para establecer una condición más fácil de verificar que (III.9), utilizando el primer resultado original que presentamos en esta metodología:

Proposición III.1. — Sea \mathcal{S} un conjunto compacto con $p_\sigma \notin \mathcal{S}$. Entonces, si

$$\lambda_{\min}\{P(x)\} K_{\mathcal{S}}^- - NM_{\mathcal{S}} D^3 > 0,$$

donde $K_{\mathcal{S}}^-$ es la norma mínima del gradiente en el conjunto compacto \mathcal{S} , entonces $L_\sigma(p_c, x)$ es una dirección de ascenso en $p_c \in \mathcal{S}$.

Demostración. El Lema III.3 acota inferiormente $F_{\mathcal{S}}(x)$ en (III.9); por lo tanto, tenemos que

$$\begin{aligned} \nabla \sigma(p_c)^T L_\sigma &= \nabla \sigma(p_c)^T L_\sigma^1 + \nabla \sigma(p_c)^T E \\ &= \frac{1}{ND^2} \nabla \sigma(p_c)^T P(x) \nabla \sigma(p_c) + \nabla \sigma(p_c)^T E \\ &\geq \frac{\lambda_{\min}\{P(x)\}}{ND^2} \|\nabla \sigma(p_c)\|^2 + \nabla \sigma(p_c)^T E. \end{aligned}$$

Por lo tanto, para garantizar que $L_\sigma(p_c)$ sea una dirección de ascenso en $p_c \in \mathcal{S}$, es suficiente satisfacer $\frac{\lambda_{\min}\{P(x)\}}{ND^2} \|\nabla \sigma(p_c)\|^2 + \nabla \sigma(p_c)^T E > 0$. Es decir, si

$$\begin{aligned}
\nabla \sigma(p_c)^T L_\sigma &= \frac{\lambda_{\min}\{P(x)\}}{ND^2} \|\nabla \sigma(p_c)\|^2 - \|\nabla \sigma(p_c)\| M_S D \\
&= \|\nabla \sigma(p_c)\| \left(\frac{\lambda_{\min}\{P(x)\} \|\nabla \sigma(p_c)\|}{ND^2} - M_S D \right) \\
&\propto \lambda_{\min}\{P(x)\} K_S^- - NM_S D^3 > 0,
\end{aligned}$$

entonces $\nabla \sigma(p_c)^T L_\sigma > 0$; por lo tanto, se satisface la condición (III.9) y $L_\sigma(p_c)$ es una dirección de ascenso en $p_c \in \mathcal{S}$. \square

En la Proposición III.1, la dependencia con N es un poco engañosa. Recordemos que $P(x) = \sum_{i=1}^N x_i x_i^T$, de modo que para algún $a \in \mathbb{R}^m$ no nulo, la expresión $a^T P(x) a > 0$ puede ser acotada inferiormente por $\min\{\lambda_{\min}\{x_i x_i^T + x_j x_j^T\}\} \frac{1}{2} N \|a\|^2$, lo que haría desaparecer a N de la condición. Podemos decir algo similar sobre la dependencia en D , ya que $P(x)$ incorpora D^2 debido a que consiste en los elementos $x_i x_i^T$; por lo tanto, la dependencia efectiva de la Proposición III.1 en D es lineal, no cúbica. Esta última observación no es sorprendente, ya que coincide con el resultado del Lema III.2.

Aunque la señal σ sea desconocida, gracias a la Proposición III.1, siempre podemos diseñar D en función de los escenarios esperados. Por ejemplo, supongamos la liberación de un contaminante; los científicos pueden estimar los valores de M_S , K_S^+ y K_S^- para los umbrales mínimos/máximos de contaminación en el **área de patrullaje** \mathcal{S} , donde el equipo de robots necesita reaccionar de manera confiable. También resulta interesante destacar que, al diseñar L_σ^1 de forma paralela al gradiente $\nabla \sigma(p_c)$, también se hace que el producto escalar $\nabla \sigma(p_c)^T L_\sigma > 0$ más robusto en relación a D .

Hasta ahora, nuestra metodología ha requerido de una topología en estrella, lo que significa que todo el conjunto de robots deja de funcionar si la unidad de cómputo que actúa como núcleo desaparece. Si bien seleccionar robots de respaldo podría ser una solución, un enfoque alternativo es distribuir los cálculos en una red arbitraria de subequipos dentro del enjambre de robots. Dicha red mejoraría la resiliencia del enjambre de robots, ya que los distintos subequipos podrían unirse o abandonar al enjambre sin afectar significativamente el rendimiento de la misión, al mismo tiempo que reduciría las necesidades en infraestructura de comunicación. Proponemos entonces una red arbitraria de equipos en una topología en estrella (ver Figura 18), donde la **dirección de ascenso común** \tilde{L}_σ se logra mediante un algoritmo estándar de consenso [42], es decir, las unidades de cómputo de los equipos vecinos compartirán sus direcciones de ascenso para alcanzar el consenso de que

$$\tilde{L}_\sigma = \frac{1}{N_c} \sum_{k=1}^{N_c} L_{\sigma_k}(p_c^k, x^k). \quad (\text{III.10})$$

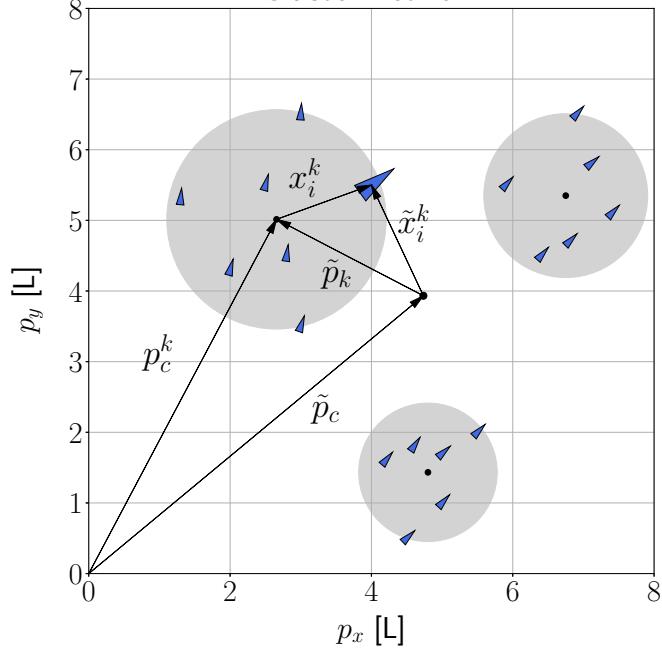


Fig. 18: En esta figura se visualiza toda la notación para un caso en el que tres equipos coordinados computan una dirección de ascenso común \tilde{L}_σ .

La pregunta que nos hacemos ahora es si \tilde{L}_σ es una dirección de ascenso en el centroide de centroides \tilde{p}_c . Dado que el sumatorio de productos escalares positivos siempre va a ser estrictamente mayor que cero, está claro que si $L_{\sigma_k}(p_c^k, x^k)$ es una dirección de ascenso en \tilde{p}_c para todos los $k \in \{1, \dots, N_c\}$, entonces \tilde{L}_σ también lo será (ver Figura 19). Inicialmente, puede parecer un trabajo redundante, ya que si se encuentra una dirección de ascenso, el resto de los subequipos podrían adoptarla. Sin embargo, en la práctica, el algoritmo de consenso va a ofrecer una gran robustez, ya que elimina la necesidad de diseñar una lógica adicional para determinar qué $L_k(p_c^k, x^k)$ son admisibles en cada p_c^k , al tiempo que permite a los subequipos un modo seguro de fusionarse o salir de la red. Es interesante señalar que cualquier robot puede pertenecer a múltiples subequipos simultáneamente, lo que permite la superposición y mejora la resiliencia del enjambre de robots.

También conviene destacar que el algoritmo de consenso, donde todas las unidades de cómputo vecinas comparten sus $L_{\sigma_k}(p_c^k, x^k)$, converge rápidamente al valor común de forma exponencial. Esta es otra característica importante en la práctica, ya que $L_{\sigma_k}(p_c(t)^k, x^k)$ varía con el tiempo y la velocidad de los robots determinará el ancho de banda para ejecutar el algoritmo de consenso. En conclusión, la convergencia exponencial a L_{σ_k} aliviará esta necesidad de ancho de banda.

A continuación, mostraremos cómo garantizar que $L_{\sigma_k}(p_c^k, x^k)$ sea una dirección de ascenso en \tilde{p}_c . Sabemos que si el k -ésimo clúster elige su D (o D_k) de acuerdo con la Proposición III.1, entonces hay un margen para variar p_c^k de manera que L_{σ_k} sea una dirección de ascenso en $\tilde{p}_c = p_c^k + \tilde{p}_k$. Veamos cómo hacerlo.

Consideremos la siguiente expansión en serie de Taylor entorno a p_c^k :

$$\nabla \sigma(p_c^k + \tilde{p}_c^k) = \nabla \sigma(p_c^k) + H(p_c^k)\tilde{p}_c^k + \mathcal{O}(\|\tilde{p}_c^k\|^2),$$

y definamos $\tilde{E} := \nabla\sigma(p_c^k + \tilde{p}_c^k) - \nabla\sigma(p_c^k)$. Entonces tenemos que

$$\|\tilde{E}\| \leq M\|\tilde{p}_c^k\|.$$

Al igual que en la demostración de la Proposición III.1, debemos asegurarnos de que el producto escalar $L_{\sigma_k}^T \nabla\sigma(p_c^k + \tilde{p}_c^k) > 0$ para que L_{σ_k} sea una dirección de ascenso en \tilde{p}_c . Esto es equivalente a verificar si

$$L_{\sigma_k}^T \nabla\sigma(p_c^k + \tilde{p}_c^k) = L_{\sigma_k}^T \nabla\sigma(p_c^k) + L_{\sigma_k}^T \tilde{E} > 0,$$

lo que nos lleva a presentar nuestro segundo resultado original para esta metodología, un corolario de la Proposición III.1.

Coloratio III.1. — *Dado un enjambre de robots formado por N_c (sub)equipos, tendremos que \tilde{L}_σ es una dirección de ascenso en $\tilde{p}_c \in \mathcal{S}$ si $\forall k \in \{1, \dots, N_c\}$*

$$\lambda_{\min}\{P(x^k)\}K_{\mathcal{S}}^- - M_{\mathcal{S}}(N_k D_k^3 + \|\tilde{p}_c^k\|) > 0.$$

Cuando contemos con más de una unidad de cómputo en un mismo enjambre, el Corolario III.1 nos dice que será más sencillo satisfacer la condición para que \tilde{L}_σ sea dirección de ascenso cuanto menor sea D_k y la distancia de los equipos al centroide de centroides $\|\tilde{p}_c^k\|$. Es decir, siempre será conveniente minimizar el tamaño de los equipos y la distancia entre ellos. Véase que la dependencia de N también es un poco engañoso en esta condición, pues cuanto mayor sea el número de agentes del equipo k , menor será el valor de $\|\tilde{p}_c^k\|$ según la definición de centroide.

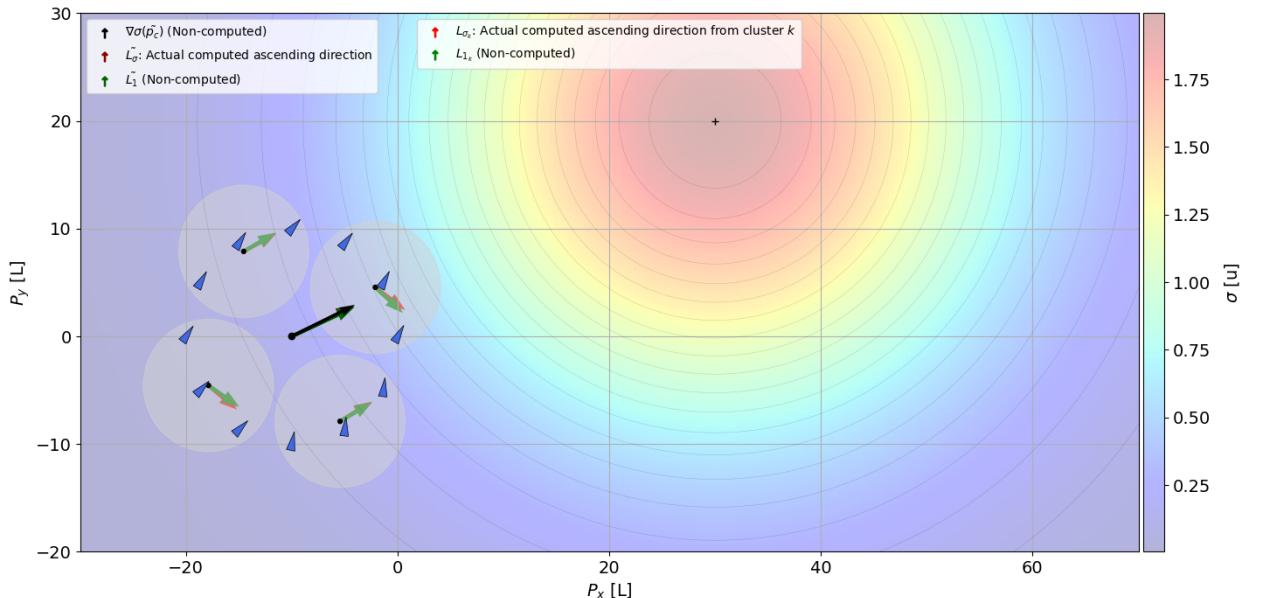


Fig. 19: En esta figura tenemos a un enjambre formado por cuatro (sub)equipos. La unidad de cómputo de cada uno de los equipos calcula su L_{σ_k} y la comparte con el resto para calcular de forma distribuida \tilde{L}_{σ_k} .

Concluimos esta subsección mencionando que el Corolario III.1 también se puede emplear para diseñar la implementación de diferentes equipos que patrullan un mismo área. Si están en riesgo de colisión, podrán fusionarse con garantías y tener una dirección de ascenso común para el nuevo centroide creado.

III-D.2. Análisis de observabilidad y sensibilidad

El análisis de cómo ciertos despliegues x generan una dirección L_σ^1 paralela al gradiente $\nabla\sigma(p_c)$, y la sensibilidad de dicho paralelismo ante modificaciones en el despliegue, es crucial para obtener enjambres de robots más resilientes en la resolución del problema de *souce-seeking*; por ejemplo, si consideramos desplazamientos erróneos de los robots o transformaciones de la formación a lo largo de la misión. Al fin y al cabo, las magnitudes de L_σ^1 y L_σ no son tan importantes, especialmente si queremos mantener independiente la velocidad de los robots de dichas magnitudes, es decir, simplemente rastrear $\frac{L_\sigma}{|L_\sigma|}$. Además, veremos que para algunos despliegues tenemos que $L_\sigma^1 = L_\sigma$ cuando aproximamos la señal $\sigma(p_c)$ hasta segundo orden alrededor de p_c .

Con el fin de ser ilustrativos, comenzaremos el análisis proponiendo el caso discreto de cuatro robots formando un rectángulo ($x^{4\text{rect}}$ en la Figura 20), luego un polígono regular, y aplicamos algunas transformaciones afines para comprobar la sensibilidad de las direcciones de ascenso. Posteriormente, desplegamos *un enjambre de robots* dentro de una forma siguiendo una distribución de densidad (x^{rect} en la Figura 20), es decir, consideramos $N \rightarrow \infty$, y mostramos condiciones de simetría suficientes para que L_σ^1 sea paralelo al gradiente.

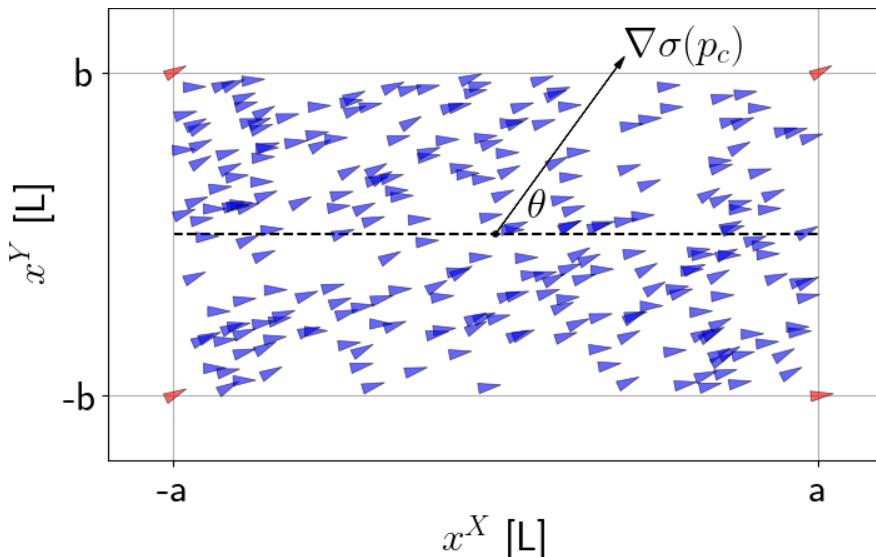


Fig. 20: La formación rectangular de 4 robots $x^{4\text{rect}}$ se encuentra en las esquinas del rectángulo en color rojo. Los 250 robots en color azul pertenecen a la formación x^{rect} y están distribuidos uniformemente dentro del rectángulo. El gradiente $\nabla\sigma(p_c)$ es arbitrario y forma un ángulo θ con el eje horizontal de la formación rectangular.

Consideremos cuatro robots en las esquinas de un rectángulo y, sin pérdida de generalidad, supongamos que el lado largo del rectángulo es paralelo al eje horizontal, de modo que el gradiente $\nabla\sigma(p_c) = \|\nabla\sigma(p_c)\| [\cos(\theta) \sin(\theta)]^T$ tiene una norma y un ángulo arbitrarios (ver Figura 20). Entonces, la dirección de ascenso L_σ^1 se puede escribir como

$$L_\sigma^1(p_c, x) = \frac{\|\nabla\sigma(p_c)\|}{ND^2} \sum_{i=1}^N \left([\cos(\theta) \sin(\theta)] \begin{bmatrix} x_i^X \\ x_i^Y \end{bmatrix} \right) \begin{bmatrix} x_i^X \\ x_i^Y \end{bmatrix}, \quad (\text{III.11})$$

donde los superíndices X e Y denotan las coordenadas horizontales y verticales de x_i con respecto a un sistema de referencia arbitrario, como se muestra en la Figura 20. Por lo tanto, para los cuatro robots en las esquinas del rectángulo con despliegue $x^{4\text{rect}}$ tenemos que

$$\begin{aligned} L_\sigma^1(p_c, x^{4\text{rect}}) &= \frac{\|\nabla\sigma(p_c)\|}{4(a^2 + b^2)} \left((a \cos(\theta) + b \sin(\theta)) \begin{bmatrix} a \\ b \end{bmatrix} + \right. \\ &\quad + (-a \cos(\theta) + b \sin(\theta)) \begin{bmatrix} -a \\ b \end{bmatrix} + (-a \cos(\theta) - b \sin(\theta)) \begin{bmatrix} -a \\ -b \end{bmatrix} + \\ &\quad \left. + (a \cos(\theta) - b \sin(\theta)) \begin{bmatrix} a \\ -b \end{bmatrix} \right) = \frac{\|\nabla\sigma(p_c)\|}{(a^2 + b^2)} \begin{bmatrix} a^2 \cos(\theta) \\ b^2 \sin(\theta) \end{bmatrix}, \end{aligned}$$

donde podemos observar que para el cuadrado ($a = b$), tenemos que $L_\sigma^1(p_c, x^{4\text{rect}}) = \frac{1}{2}\nabla\sigma(p_c)$. Finalmente, si a o b es igual a cero (pero no ambos) para la formación rectangular $x^{4\text{rect}}$, en $L_\sigma^1(p_c, x^{4\text{rect}})$ únicamente se observará la proyección de $\nabla\sigma(p_c)$ en la línea descrita por la formación x degenerada. No obstante, hay que tener en cuenta que en este caso los robots colapsarían por parejas en dos puntos coincidentes. Esta situación indeseable es la que nos motivó para llegar al siguiente resultado.

Proposición III.2 (Observabilidad con formaciones en cruz). — *Consideremos dos formaciones degeneradas en 2D (segmentos) y perpendiculares con longitudes iguales; es decir, x^\parallel y x^- con N_\parallel y N_- números de robots, que están distribuidos simétricamente alrededor del centroide, de modo que ambos despliegues comparten su centroide. Considerando uno de los segmentos, su robot más lejano 1 se encontrará a una distancia $a_1 \in \mathbb{R}$, mientras que el resto de sus robots $i \in \{1, \dots, \frac{N}{2}\}$ a distancias $0 \leq a_{i+1} < a_i$ tal que $\sum_{i=2}^{\frac{N}{2}} a_i^2 = \frac{N}{4}a_1^2$. Entonces, $\lim_{N_\parallel \rightarrow \infty} L_\sigma^1(p_c, x^\parallel) + \lim_{N_- \rightarrow \infty} L_\sigma^1(p_c, x^-) = \frac{1}{2}\nabla\sigma(p_c)$.*

Demostración. En primer lugar, observemos que un robot en $x_i = 0$ no contribuye en absoluto al cómputo de L_σ ; por lo tanto, dado que requerimos simetría reflectante alrededor de $x = 0$, podemos suponer que N_\parallel y N_- son pares para que sus mitades también sean números enteros. Además, dado que los dos segmentos tienen la misma longitud, podremos asumir que $a_{1\parallel} = a_{1-} = a_1$. Entonces, de acuerdo con (III.7), tenemos que

$$L_\sigma^1(p_c, x^\parallel) = \frac{\|\nabla\sigma(p_c)\| \cos(\theta)}{N_\parallel a_1^2} 2 \sum_{i=1}^{\frac{N_\parallel}{2}} \begin{bmatrix} a_i^2 \\ 0 \end{bmatrix}.$$

Aplicando la condición $\sum_{i=2}^{\frac{N}{2}} a_i^2 = \frac{N}{4} a_1^2$ del enunciado, que se puede verificar como factible para una serie geométrica $a_{i+1} = \alpha a_i$, $i \in \{1, \dots, (\frac{N}{2} - 1)\}$ para algún $0 < \alpha < 1$, y $a_{\frac{N}{2}} = -\left(\sum_{i=2}^{\frac{N}{2}-1} a_i^2 - \frac{N}{4}\right) < a_1^2$, tenemos que

$$\begin{aligned} L_\sigma^1(p_c, x^\parallel) &= \frac{2\|\nabla\sigma(p_c)\| \cos(\theta)}{N_\parallel a_1^2} \begin{bmatrix} a_1^2 + \frac{N_\parallel}{4} a_1^2 \\ 0 \end{bmatrix} \\ &= \frac{1}{2}\|\nabla\sigma(p_c)\| \cos(\theta) \begin{bmatrix} \frac{4+N_\parallel}{N_\parallel} \\ 0 \end{bmatrix}. \end{aligned}$$

De manera similar, podemos obtener que

$$L_\sigma^1(p_c, x^-) = \frac{1}{2}\|\nabla\sigma(p_c)\| \sin(\theta) \begin{bmatrix} 0 \\ \frac{4+N_-}{N_-} \end{bmatrix}.$$

Por lo tanto, si las dos unidades de cómputo combinan sus resultados, tendremos que

$$\lim_{N_\parallel \rightarrow \infty} L_\sigma^1(p_c, x^\parallel) + \lim_{N_- \rightarrow \infty} L_\sigma^1(p_c, x^-) = \frac{1}{2}\nabla\sigma(p_c).$$

□

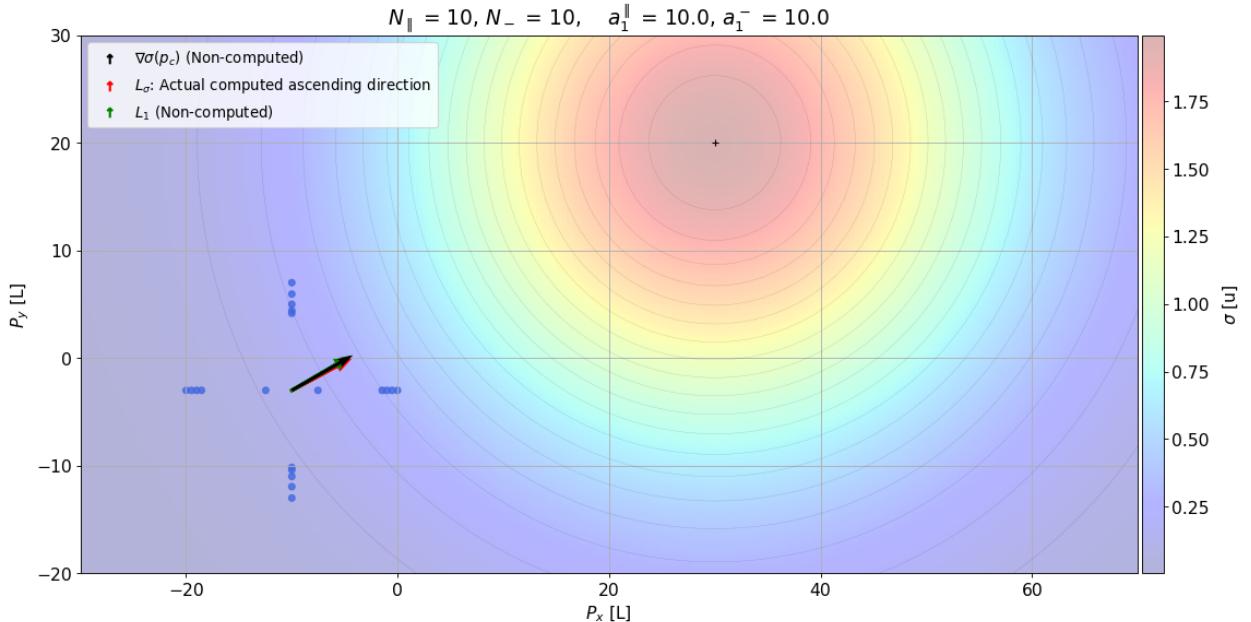


Fig. 21: Para una señal gaussiana σ y una formación x^{cross} , como la que se introduce en la Proposición III.2, en esta figura se muestra el cómputo de $L_\sigma^1(p_c, x^{\text{cross}})$ y $L_\sigma(p_c, x^{\text{cross}})$. Dado que $N_\parallel = N_-$ y $a_{1\parallel} = a_{1-}$, podemos comprobar que $L_\sigma^1(p_c, x^{\text{cross}})$ es paralelo al gradiente.

La demostración de la Proposición III.2 nos permite intuir que si $N_{\parallel} = N_-$, entonces la dirección de ascenso resultante es paralela al gradiente $\nabla\sigma(p_c)$ (ver Figura 21), un resultado que también puede obtener en 3D para una distribución equivalente. Por supuesto, el procedimiento de diseño se puede modificar para que, cuando $N_{\parallel} \neq N_-$ y $a_{1\parallel} \neq a_{1-}$, el resultado sea el mismo que en la Proposición III.2. La serie geométrica propuesta para la distribución de los robots es solo una sugerencia. Dado que lo que importa es la dirección de L_{σ}^1 , la suma $\sum_{i=2}^{\frac{N}{2}} a_i^2$ siempre podrá ser ajustada en cada segmento de forma independiente para compensar aquellos casos en los que $N_{\parallel} \neq N_-$ o $a_{1\parallel} \neq a_{1-}$.

También es bastante directo comprobar que el grupo de 3 robots $x^{3\text{tri}}$ formando un triángulo equilátero con centroide en p_c calcula $L_{\sigma}^1(p_c, x^{3\text{tri}})$ de forma paralela al gradiente $\nabla\sigma(p_c)$. En general, cualquier formación que forme un **polígono regular** tendrá su $L_{\sigma}^1(p_c, x^{N\text{poly}})$ paralelo al gradiente $\nabla\sigma(p_c)$. Este resultado ya se ha descubierto en [28], donde los robots están distribuidos de manera equitativa en una circunferencia, es decir, describen un polígono regular. Sin embargo, ¿cuál es la sensibilidad de $L_{\sigma}^1(p_c, x^{N\text{poly}})$ en relación a posibles desplazamientos incorrectos o ligeras modificaciones en la formación regular?

Aunque ya hemos demostrado en el Lema III.1 que, independientemente de x , $L_{\sigma}^1(p_c, x)$ es siempre una dirección de ascenso, proporcionaremos más detalles al respecto, estudiando lo que sucede para distintas formaciones. Aunque el enjambre de robots calcula L_{σ} en lugar de L_{σ}^1 , este análisis es relevante porque, cuanto más paralelo sea L_{σ}^1 al gradiente $\nabla\sigma$, más robusta será la condición de ascenso $L_{\sigma}(p_c, x)^T \nabla\sigma(p_c) > 0$.

En primer lugar, formalizaremos un resultado técnico que ya introducimos en la Observación III.2, pues será necesario más adelante.

Lema III.4 (Proporcionalidad con polígonos regulares). — *Consideremos la distribución $x^{N\text{poly}}$ (o $x^{3D-N\text{poly}}$) de N robots formando un polígono regular (poliedro), y $b \in \mathbb{R}^m$, entonces*

$$\sum_{i=1}^N (b^T x_i) x_i \propto b.$$

Demostración. Partimos del echo de que

$$\sum_{i=1}^N (b^T x_i) x_i = \sum_{i=1}^N (x_i x_i^T) b.$$

Para que esta expresión sea proporcional a b , es necesario que todos los elementos diagonales de la matriz semidefinida positiva $P = \sum_{i=1}^N (x_i x_i^T)$ sean iguales y que los elementos no diagonales sean nulos. Dado que todas las distribuciones consideradas se encuentran en la esfera m -dimensional, entonces todas las normas $\|x_i\|$ son iguales. Además, todos los ángulos diedros de las distribuciones consideradas son iguales, por lo que podemos encontrar ejes XYZ donde el conjunto de todas las proyecciones de los vértices x_i en los planos XY, XZ e YZ exhibe una simetría par; por lo tanto, $\sum_i^N (x_i^X)^2 = \sum_i^N (x_i^Y)^2 = \sum_i^N (x_i^Z)^2$. Finalmente, un polígono regular tiene simetría de reflexión para un eje XY; por lo tanto, $\sum_i^N (x_i^X)(x_i^Y) = 0$. Esto también es cierto para un poliedro regular en los mismos planos XY, XZ e YZ.

□

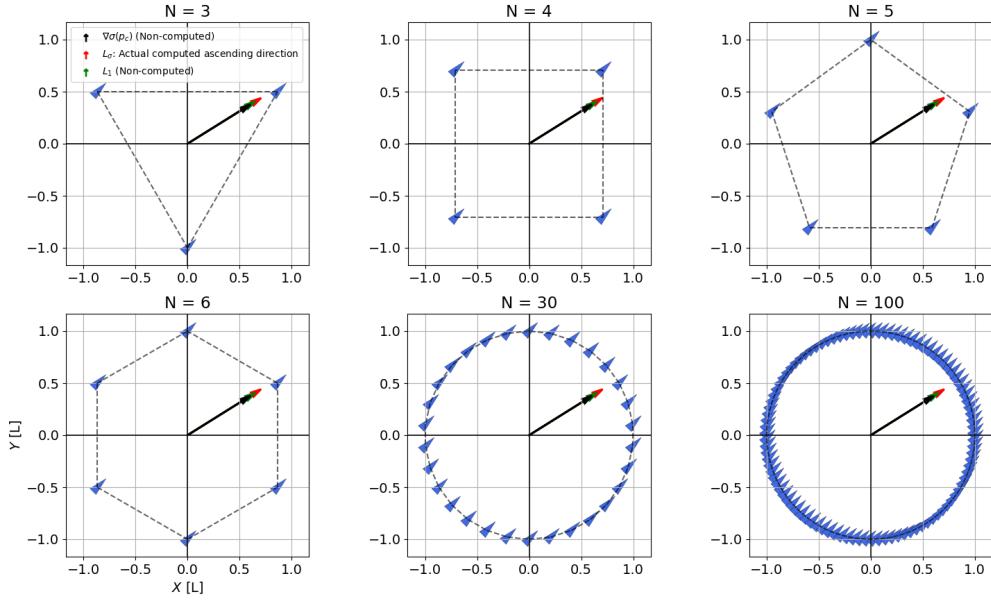


Fig. 22: En esta figura se muestra el cálculo de $L_\sigma^1(p_c, x^{N\text{poly}})$ y $L_\sigma(p_c, x^{N\text{poly}})$ para distintos valores de N . En todos los casos, se puede observar que $L_\sigma^1(p_c, x^{N\text{poly}})$ es paralelo al gradiente.

Para polígonos regulares en 2D, si alineamos el primer vértice con la línea horizontal de uno de los ejes de simetría y normalizamos la distancia de dicho vértice al centroide a 1, entonces la suma de los cuadrados de todas las componentes X de los vértices es $\sum_{i=1}^N (x_i^X)^2 = \sum_{i=0}^{N-1} \cos^2(\frac{i2\pi}{N})$, y debido a la simetría del polígono regular, la misma igualdad se cumple para $\sum_{i=1}^N (x_i^Y)^2$.

Ahora es fácil ver que $L_\sigma^1(p_c, x^{N\text{poly}}) \propto [\cos \theta \sin \theta]^T$, es decir, es paralelo al gradiente $\nabla \sigma(p_c)$ (ver Figura 22), y para poliedros regulares $L_\sigma^1(p_c, x^{3\text{D-Npoly}}) \propto [\sin \theta \cos \varphi \sin \theta \sin \varphi \cos \theta]^T$ con θ y φ siendo la inclinación y azimut del gradiente en coordenadas esféricas, con origen en el centroide del poliedro regular 3D.

De hecho, no necesitamos robots en todos los vértices del poliedro regular, sino solo un subconjunto. Por ejemplo, tomemos dos caras paralelas de un cubo; desde la primera, nos enfocamos en una diagonal y seleccionamos dos robots opuestos; desde la segunda cara, seleccionamos los robots en la otra diagonal perpendicular, entonces tendremos que $\sum_{i=1}^4 (b^T x_i) x_i \propto b, b \in \mathbb{R}^3$ para los cuatro robots en esos vértices específicos de un cubo. Por lo tanto, estos cuatro robots son suficientes para calcular un $L_\sigma^1(p_c, x^{4\text{robots-cube}})$ paralelo al gradiente 3D $\nabla \sigma(p_c)$. En 2D, necesitaremos al menos tres robots formando un triángulo equilátero. El resultado presentado aquí en 3D también coincide con el trabajo presentado en [29]; sin embargo, hemos llegado a la misma conclusión sin la necesidad de un uso intensivo de la trigonometría. De hecho, incluso podríamos extender los resultados presentados a $m > 3$ dimensiones.

A continuación, analizaremos la sensibilidad de $L_\sigma^1(p_c, x^{N\text{poly}})$ ¹ cuando la forma de la distribución está bajo una **transformación afín**; por ejemplo, escalado, rotación y cizalla, que podrían ser necesarias para explorar de forma efectiva un entorno desconocido para los robots.

¹Para una notación más clara y concisa, no indicamos explícitamente si el polígono regular es 2D o 3D, ambos casos los denotaremos como $x^{N\text{poly}}$.

En primer lugar, conviene apreciar que dicha transformación afín se puede aplicar formalmente con $(I_N \otimes A)x^{N\text{poly}}$, donde $A \in \mathbb{R}^{m \times m}$ es el operador lineal para la transformación. Dicho operador, siempre se podrá descomponer como un escalado y una rotación aplicando la descomposición en valores singulares (SVD, *singular value decomposition*) $A = U\Sigma V^T$. La matriz V , que codifica las rotaciones, ya veremos que será irrelevante, pues $L_\sigma^1(p_c, x^{N\text{poly}})$ es paralelo al gradiente en p_c para cualquier rotación de la distribución (ver Figura 23), tal y como se muestra en la siguiente proposición.

Proposición III.3 (Invarianza ante rotaciones). — $L_\sigma^1(p_c, x^{N\text{poly}})$ es invariante para todas las distribuciones $(I_N \otimes R)x^{N\text{poly}}$ con $R \in SO(m)$ siendo una matriz de rotación.

Demostración. En lo siguiente, definimos $r \in \mathbb{R}^m$ como el vector unitario que marca la dirección del gradiente $\nabla\sigma(p_c)$, y definimos su rotación por R^T como $\tilde{r} = R^T r$. Luego, aplicando la transformación R a cada x_i , tenemos que

$$\begin{aligned} L_\sigma^1(p_c, (I_N \otimes R)x^{N\text{poly}}) &= \frac{|\nabla\sigma(p_c)|}{ND^2} \sum_{i=1}^N (r^T Rx_i) Rx_i = \frac{|\nabla\sigma(p_c)|}{ND^2} R \sum_{i=1}^N (\tilde{r}^T x_i) x_i \\ &= \frac{c|\nabla\sigma(p_c)|}{ND^2} R \tilde{r} = \frac{c|\nabla\sigma(p_c)|}{ND^2} RR^T r = L_\sigma^1(p_c, x^{N\text{poly}}), \end{aligned}$$

donde $c > 0$ es la constante proporcional después de aplicar el Lema III.4, que es igual a la constante de la suma sin aplicar la rotación. \square

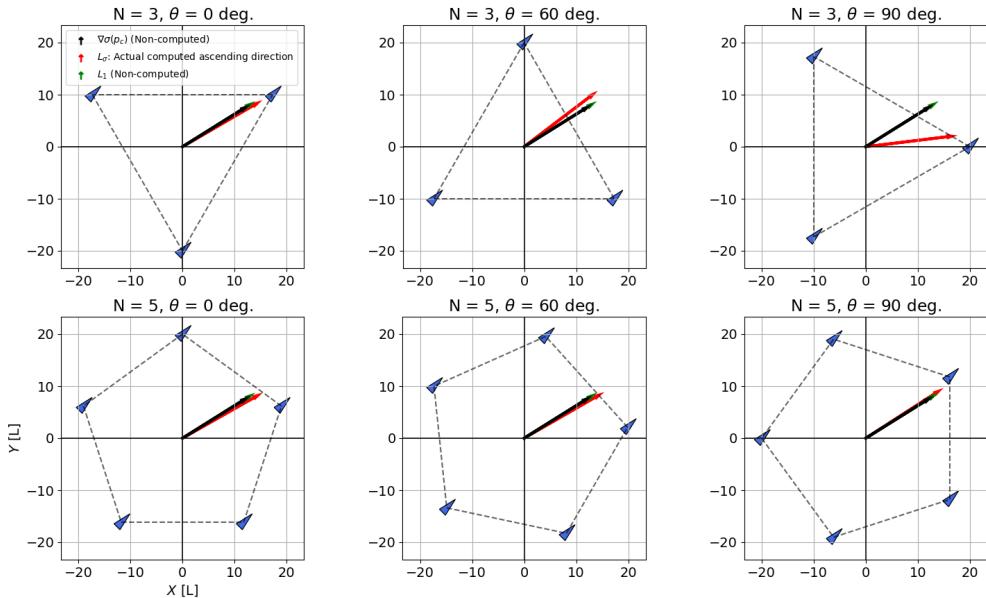


Fig. 23: Dadas dos formaciones $x = x^{N\text{poly}}$ sometidas a distintas rotaciones con respecto a p_c , en esta figura se muestra el cómputo de $L_\sigma^1(p_c, x^{N\text{poly}})$ y $L_\sigma(p_c, x^{N\text{poly}})$. Véase que, para ambas formaciones, $L_\sigma^1(p_c, x^{N\text{poly}})$ es paralelo al gradiente y completamente invariante.

Esta proposición es trivial si nos fijamos en que el ángulo θ que aparece en (III.11) es completamente arbitrario, pues depende del sistema de referencia seleccionado. No obstante, es importante haberla formalizado para poder seguir adelante con nuestro razonamiento lógico.

Con este resultado en mente, ahora solo necesitamos enfocarnos en los efectos de U y Σ de la transformación afín sobre la distribución $x^{N\text{poly}}$.

Proposición III.4. — *Consideremos la descomposición SVD $A = U\Sigma V^T$, entonces*

$$L_\sigma^1(p_c, (I_N \otimes A)x^{N\text{poly}}) \propto U\Sigma^2 U^T r,$$

donde $r \in \mathbb{R}^m$ es el vector unitario que marca la dirección del gradiente $\nabla\sigma(p_c)$.

Demostración. Dado el siguiente cambio de coordenadas $\tilde{r} = \Sigma U^T r$ y sabiendo que $L_\sigma^1(p_c, x^{N\text{poly}})$ es invariante bajo rotaciones $(I_N \otimes V^T)x$, nos enfocaremos en aplicar la transformación $(I_N \otimes U\Sigma)x$:

$$\begin{aligned} L_\sigma^1(p_c, (I_N \otimes (U\Sigma))x^{N\text{poly}}) &\propto \sum_{i=1}^N \left(r^T U\Sigma x_i \right) U\Sigma x_i \\ &\propto U\Sigma \sum_{i=1}^N \left(\tilde{r}^T x_i \right) x_i \propto U\Sigma \tilde{r} = U\Sigma^2 U^T r, \end{aligned}$$

donde hemos aplicado nuevamente el Lema III.4. □

Observación III.3. Véase que $P = U\Sigma^2 U^T$ es la descomposición unitaria de una matriz semidefinida positiva (algo que no nos sorprende debido al Lema III.1) que, para formaciones $x^{N\text{poly}}$, depende únicamente de A . No obstante, hay que tener en cuenta que para un x genérico, P también dependerá de la matriz de rotación (desconocida a priori) entre r y los ejes donde se define A .

La Proposición III.4 tiene una interesante aplicación práctica. Dado que el L_σ^1 para formaciones $x^{N\text{poly}}$ sigue la dirección del gradiente, podemos decir que el enjambre dispone de un rango simétrico ($\pm\frac{\pi}{2}$ radianes en 2D) para maniobrar mientras se acerca a la fuente. En particular, según la Proposición III.4, la formación solo necesita estirar su distribución en la dirección en la que desea maniobrar, es decir, elegir una matriz diagonal Σ genérica y establecer $U = I_m$. De este modo, la dirección resultante seguirá el estiramiento $\xi\Sigma^2 \xi\nabla\sigma(p_c)$, donde ξ denota la representación en los ejes perpendiculares elegidos.

También conviene recordar que esta propiedad no es cierta para una distribución genérica, pues no se cumplen las condiciones para poder aplicar el Lemma III.4, ni las Proposiciones III.3 y III.4. En 2D, podría ocurrir que un estiramiento genérico de x sea precedido por una matriz de rotación que impida independizar el escalado de las componentes de L_σ^1 en dos ejes perpendiculares.

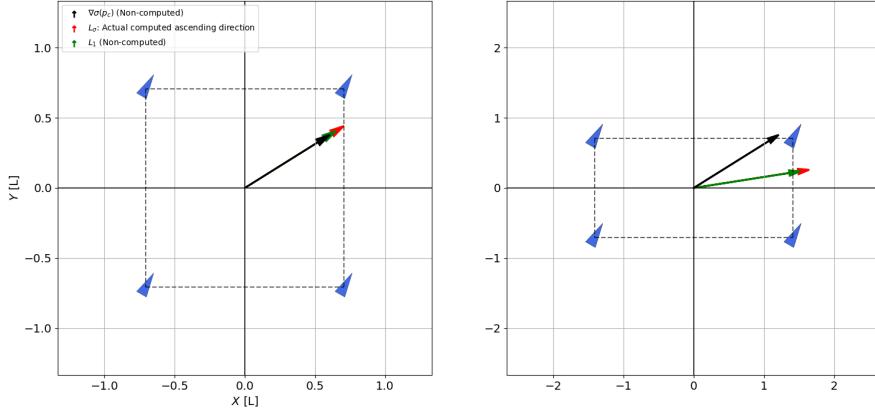


Fig. 24: En esta figura, se muestra el cómputo de $L_\sigma^1(p_c, x)$ y $L_\sigma(p_c, x)$ para dos formaciones distintas. Una es $x^{4\text{poly}}$ y la otra es una formación rectangular que puede ser obtenida aplicando una transformación afín de escalado sobre $x^{4\text{poly}}$. Véase que, cuando $x^{4\text{poly}}$ es ensanchada con respecto a un único eje, las proyecciones de $L_\sigma^1(p_c, x)$ y $L_\sigma(p_c, x)$ sobre ese mismo eje también crecen proporcionalmente.

A continuación, nos preguntamos cuál sería L_σ^1 si desplegamos un enjambre compuesto por una gran cantidad de agentes, encerrado dentro de cierto área/volumen y con x siguiendo una distribución genérica ρ . Con este análisis, podremos dejar de enfocarnos únicamente en polígonos/poliedros regulares, y comenzar a explorar **formaciones genéricas** con ciertas simetrías que, como ya veremos más adelante, nos permitirán enunciar una serie de condiciones suficientes para garantizar que L_σ^1 sea paralelo al gradiente $\nabla\sigma$.

Cuando $N \rightarrow \infty$, nos acercamos al continuo, donde la suma discreta en (III.7) se puede interpretar como una integral si tenemos en cuenta la aproximación de una integral definida con sumas de Riemann. En estos casos, dado que los robots se interpretan como puntos diferenciales, el N de (III.7) se convertirá en el área/volumen encerrado por el perímetro de x , es decir, $N = \iiint_{\mathcal{A}} \rho(X, Y, Z) dX dY dZ$, donde \mathcal{A} es la superficie/volumen correspondiente, y $\rho : \mathbb{R}^m \rightarrow \mathbb{R}^+$ es la función de densidad de probabilidad [43] de los robots. Dicha función ρ puede ser, por ejemplo, igual en todas partes para distribuciones uniformes, o similar a una serie geométrica como se ilustra en la Proposición III.2. Conviene resaltar que las coordenadas x tendrán su origen en la media de $\rho(X, Y, Z)$ en \mathcal{A} .

Para ser más concisos, nos centraremos únicamente en el caso 2D, y para mayor claridad en la notación, denotaremos x_i^X y x_i^Y simplemente como X e Y . En consecuencia, para el caso de un enjambre de robots siguiendo una función de densidad de probabilidad $\rho(X, Y)$ dentro de una forma/superficie genérica \mathcal{A} , la dirección de ascenso $L_\sigma^1(p_c, x)$ se puede calcular como

$$\begin{aligned} L_\sigma^1(p_c, x) &= \frac{\|\nabla\sigma(p_c)\|}{A} \iint_{\mathcal{A}} \rho(X, Y) \left(\begin{bmatrix} \cos(\theta) & \sin(\theta) \end{bmatrix} \begin{bmatrix} x_i^X \\ x_i^Y \end{bmatrix} \right) \begin{bmatrix} x_i^X \\ x_i^Y \end{bmatrix} dX dY \\ &= \frac{\|\nabla\sigma(p_c)\|}{A} \iint_{\mathcal{A}} \rho(X, Y) \begin{bmatrix} X^2 \cos(\theta) + XY \sin(\theta) \\ Y^2 \sin(\theta) + XY \cos(\theta) \end{bmatrix} dX dY, \end{aligned}$$

donde, en lugar de D , tenemos $A = \iint_{\mathcal{A}} \rho(X, Y) dX dY$, que es el área de \mathcal{A} en unidades [*longitud* \times *longitud*]. De modo que será suficiente tener las integrales $\iint_{\mathcal{A}} \rho(X, Y) XY dX dY$ y $\iint_{\mathcal{A}} \rho(X, Y) (X^2 - Y^2) dX dY$ iguales a cero para que $L_{\sigma}^1(p_c, x)$ sea paralelo al gradiente $\nabla \sigma(p_c)$. Teniendo esto en mente, se puede llegar al siguiente resultado.

Proposición III.5. — *Sea una señal σ , un enjambre con una formación x que consta de $N \rightarrow \infty$ robots siguiendo una función de densidad de probabilidad $\rho(X, Y)$ dentro de una superficie \mathcal{A} , y un sistema de coordenadas cartesianas $(X - Y)$ arbitrario con origen en el centroide de la formación p_c . Diremos que la dirección de ascenso $L_{\sigma}^1(p_c, x)$ es paralela al gradiente $\nabla \sigma(p_c)$ si la función de densidad de probabilidad $\rho(X, Y)$ para las posiciones de los robots y la superficie \mathcal{A} cumplen las siguientes simetrías:*

- S0) *La función de densidad de probabilidad $\rho(X, Y)$ tiene simetría de reflexión (función par) con respecto a al menos uno de los ejes $(X - Y)$, por ejemplo, $\rho(X, Y) = \rho(-X, Y)$.*
- S1) *La superficie \mathcal{A} tiene simetría de reflexión respecto al mismo eje que en S0.*
- S2) *Para cada cuadrante de $(X - Y)$, la función de densidad $\rho(X, Y)$ tiene simetría de reflexión con respecto a la bisectriz del cuadrante.*
- S3) *La superficie \mathcal{A} tiene simetría de reflexión respecto al los mismos ejes que en S2.*

Demostración. En primer lugar, abordamos el caso de $h_1(X, Y) = XY$ y posteriormente el de $h_2(X, Y) = X^2 - Y^2$. Sin pérdida de generalidad, asumimos que la simetría de reflexión es en el eje Y ; por lo tanto, tenemos que $h_1(X, Y) = -h_1(-X, Y)$ y $\rho(X, Y) = \rho(-X, Y)$. Gracias a esta simetría, también es factible asumir que los límites de integración de \mathcal{A} son simétricos para el eje X , tal y como se muestra en la Figura 25(a). De este modo, tenemos que

$$\begin{aligned} \iint_{\mathcal{A}} \rho(X, Y) XY dX dY &= \int_{-t_{\beta}}^{t_{\beta}} \int_0^{\beta(X)} \rho(X, Y) XY dX dY + \int_{-t_{\alpha}}^{t_{\alpha}} \int_{\alpha(X)}^0 \rho(X, Y) XY dX dY \\ &= \int_{-t_{\beta}}^{t_{\beta}} XF(X, \beta(X)) dX - \int_{-t_{\alpha}}^{t_{\alpha}} XF(X, \alpha(X)) dX = 0, \end{aligned}$$

siempre y cuando $\alpha(X)$ y $\beta(X)$ sean funciones pares. Expliquemos este paso con más detalle; en primer lugar, hay que tener en cuenta que

$$F(X, Y) = \int \rho(X, Y) Y dY$$

siempre va a cumplir que $F(X, Y) = F(-X, Y)$, pues $\rho(X, Y) = \rho(-X, Y)$. Lo que no se tiene por qué cumplir es $F(X, Y) = F(X, -Y)$; no obstante, teniendo en cuenta que la composición de una función par con otra impar es siempre par, dado $g(X) = F(X, f(X))$, si $f(X)$ es par entonces $g(X) = g(-X) \implies F(X, f(X)) = F(-X, f(-X))$. Es decir, aplicado a nuestro caso particular, si $\alpha(X)$ y $\beta(X)$ son pares entonces $F(X, \alpha(X))$ y $F(X, \beta(X))$ también van a ser pares. Finalmente, teniendo en cuenta que el producto de una función por otra impar da como resultado una función impar, y que la integral de una función impar en un rango de integración simétrico siempre es nula. Es directo llegar a la conclusión de que $\iint_{\mathcal{A}} \rho(X, Y) XY dX dY = 0$

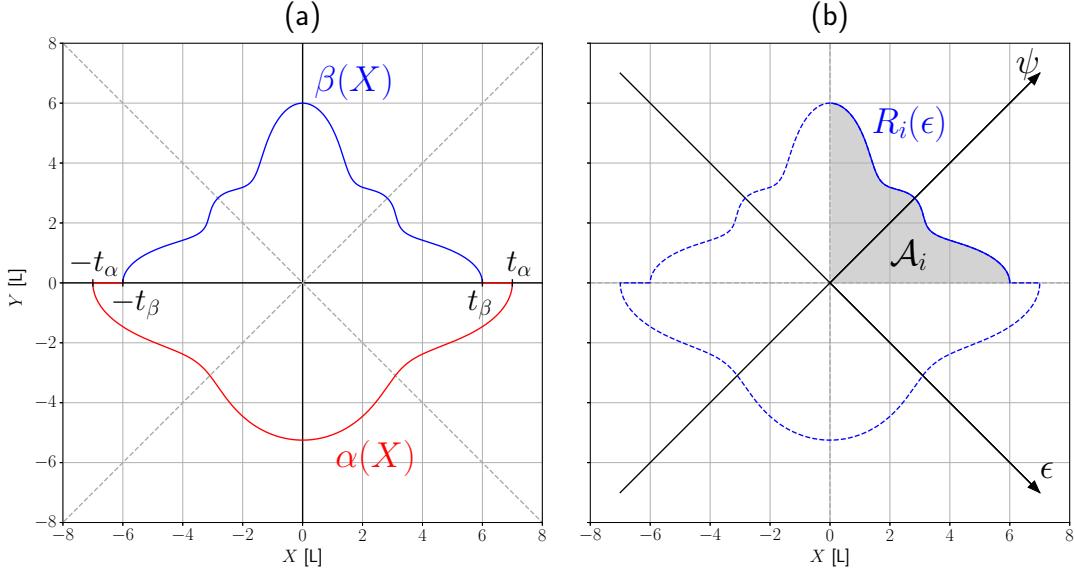


Fig. 25: Ilustración de las dos simetrías referentes al área \mathcal{A} , S1 a la izquierda y S3 a la derecha. Dicha \mathcal{A} hace referencia a la superficie que encierra a un enjambre compuesto por $N \rightarrow \infty$ robots, distribuidos siguiendo una función de densidad de probabilidad $\rho(X, Y)$ que debería respetar S0 y S2.

A continuación, analizaremos la integral de $h_2(X, Y)$, sumando a todas las simetrías anteriores S2 y S3. En primer lugar, hay que tener en cuenta que

$$\iint_{\mathcal{A}} \rho(X, Y) h_2(X, Y) dX dY = \sum_{i=1}^4 \iint_{\mathcal{A}_i} \rho(X, Y) h_2(X, Y) dX dY,$$

donde \mathcal{A}_i denota el área del cuadrante $i \in \{1, 2, 3, 4\}$. Es decir, si conseguimos demostrar que la integral de $h_2(X, Y)$ sobre \mathcal{A}_i es nula para todos los cuadrantes, entonces la integral de $h_2(X, Y)$ sobre \mathcal{A} también lo será. Con esto en mente, aplicaremos el cambio de variable $g(\epsilon, \psi) = (\psi + \epsilon, \psi - \epsilon)/\sqrt{2}$, que corresponde a una rotación de $+\pi/2$ radianes del plano $(X - Y)$. De este modo, sabiendo que $\int_A f(x, y) = \int_B (f \circ g)|J_g|$, donde $|J_g| = |\begin{bmatrix} \nabla g_1 & \nabla g_2 \end{bmatrix}^T| = \sqrt{2}$, y que gracias a S2 y S3 tenemos límites de integración simétricos en ϵ y $\rho(\epsilon, \psi) = \rho(-\epsilon, \psi)$, entonces

$$\begin{aligned} \iint_{\mathcal{A}_i} \rho(X, Y) h_2(X, Y) dX dY &= 2\sqrt{2} \iint_{\mathcal{B}_i} \rho(\epsilon, \psi) \epsilon \psi d\epsilon d\psi \\ &= 2\sqrt{2} \int_{-t_i}^{t_i} \int_{|\epsilon|}^{R_i(\epsilon)} \rho(\epsilon, \psi) \epsilon \psi d\epsilon d\psi \\ &= 2\sqrt{2} \int_{-t_i}^{t_i} \epsilon [F(\epsilon, R_i(\epsilon)) - F(\epsilon, |\epsilon|)] d\epsilon = 0, \end{aligned}$$

siempre y cuando $R_i(\epsilon) = R_i(-\epsilon)$; lo que equivale a decir que $\alpha(X)$ tenga reflexión especular con respecto respecto al bisector dentro de los dos cuadrantes superiores, y que $\beta(X)$ cumpla los mismo pero para los dos cuadrantes inferiores. Téngase en cuenta que evidentemente $|\epsilon|$ también es una función par.

□

Es interesante tener en cuenta que $\iint_{\mathcal{A}} \rho(X, Y) XY dX dY$ se puede separar en $\iint_{\mathcal{A}} \rho(X, Y) X dX = X_c$ y $\iint_{\mathcal{A}} \rho(X, Y) Y dY = Y_c$, donde X_c e Y_c son las coordenadas del centroide de la distribución en el plano $(X - Y)$; y que $\iint_{\mathcal{A}} \rho(X, Y) (X^2 - Y^2) dX dY = \text{VAR}_{\mathcal{A}}[X] - \text{VAR}_{\mathcal{A}}[Y]$, donde $\text{VAR}_{\mathcal{A}}[X]$ es la varianza de la distribución con respecto a un eje arbitrario X . Es decir, cuando la integral en $h_1(X, Y)$ sea nula, significará que la coordenada del centroide en el eje perpendicular al eje de simetría es nula; mientras que si la integral en $h_2(X, Y)$ es nula, tendremos que $\text{VAR}_{\mathcal{A}}[X] = \text{VAR}_{\mathcal{A}}[Y]$.

La Proposición III.5 es un resultado muy potente, pues nos permite asegurar que la dirección de ascenso $L_{\sigma}^1(p_c, x)$ es siempre paralela al gradiente $\nabla \sigma(p_c)$ para una familia de formaciones mucho más amplia que $x^{N_{\text{poly}}}$ (ver Figura 26). Es más, resulta interesante observar que si tomamos el área que encierra cualquier formación $x^{N_{\text{poly}}}$ y suponemos, por ejemplo, una distribución uniforme (ver x^{rect} en la Figura 20), entonces la Proposición III.5 nos dice lo mismo que el Lema III.4. Este echo nos motiva a pensar que, en cierto modo, siempre que una formación discreta cumpla unas simetrías similares a las de la Proposición III.5, tendremos que la dirección de $L_{\sigma}^1(p_c, x)$ es muy próxima a la del gradiente $\nabla \sigma(p_c)$.

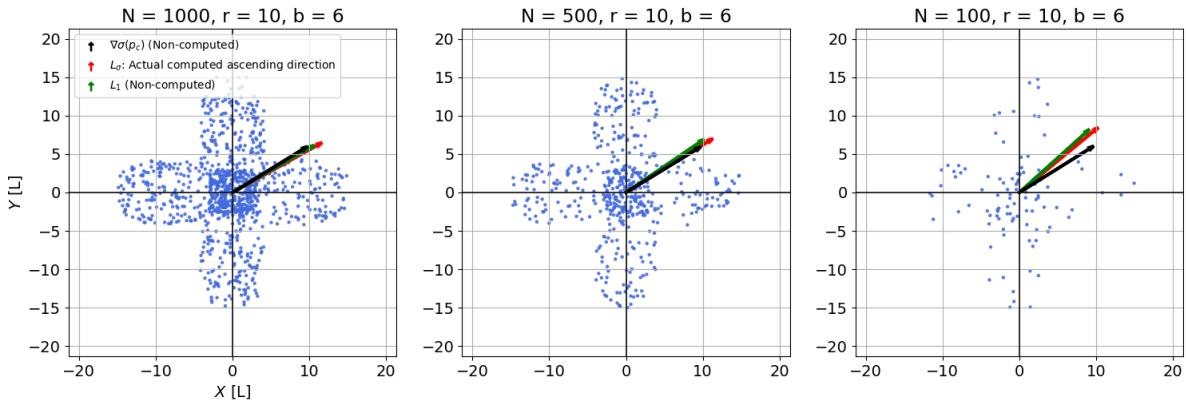


Fig. 26: En esta figura tenemos tres formaciones, con grandes diferencias en cuanto a número de agentes N , pero encerradas en un mismo perímetro \mathcal{A} y siguiendo una misma función de densidad $\rho(X, Y)$. Dado que se cumplen todas las simetrías de la Proposición III.5, podemos visualizar que según N aumenta, es decir N comienza a tender a ∞ , $L_{\sigma}^1(p_c, x)$ es cada vez más paralelo al gradiente $\nabla \sigma(p_c)$.

Para finalizar esta subsección, conviene analizar lo que sucede cuando únicamente se cumplen las simetrías S0 y S1, es decir $\text{VAR}_{\mathcal{A}}[X] \neq \text{VAR}_{\mathcal{A}}[Y]$. En estos casos, tendremos que

$$L_{\sigma}^1(p_c, x) = \frac{\|\nabla \sigma(p_c)\|}{A} \begin{bmatrix} \text{VAR}_{\mathcal{A}}[X] \cos(\theta) \\ \text{VAR}_{\mathcal{A}}[Y] \sin(\theta) \end{bmatrix},$$

de modo la varianza en un eje modulará la proyección de $L_\sigma^1(p_c, x)$ sobre ese mismo eje. Suponiendo una formación de este estilo, como la que se muestra en la Figura 27, podemos comprobar que esta característica es muy interesante, pues escalando la formación en direcciones paralelas o perpendiculares al eje de simetría podemos maniobrar al enjambre mientras se garantiza que L_σ^1 sigue siendo una dirección de ascenso. Esta es una herramienta que, por ejemplo, permitirá al enjambre evitar obstáculos y adaptarse a escenarios desconocidos mientras p_c sigue tendiendo a la fuente de σ .

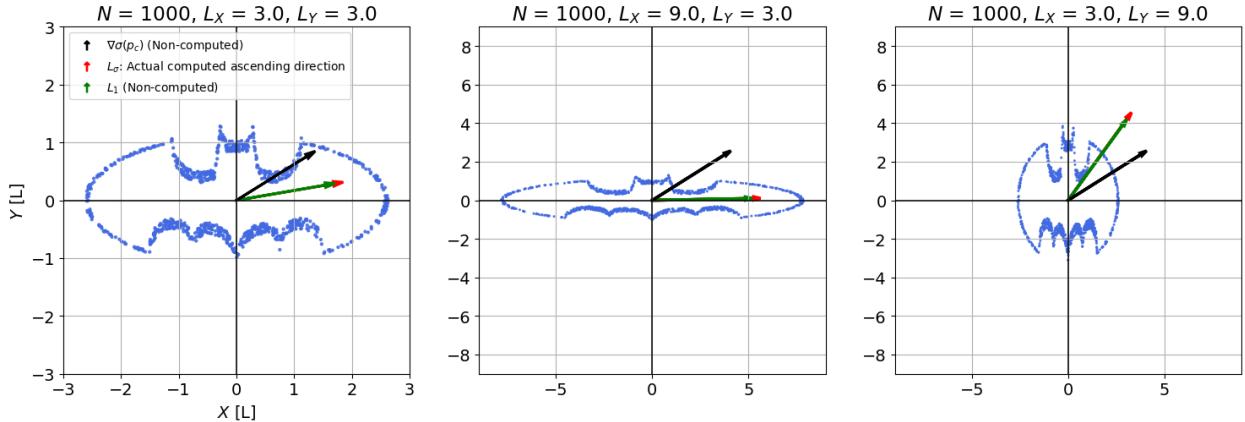


Fig. 27: Tenemos a un enjambre con un número fijo de agentes N que trata de dibujar el logo de Batman, generando una formación con simetría especular respecto a Y . En esta figura, se computa L_σ^1 y L_σ para observar cómo varían dichos vectores ante distintos escalados de la formación en los ejes X e Y . Notamos que cuando la varianza aumenta en un eje, la componente paralela a dicho eje de L_σ^1 también lo hace de forma proporcional y, por supuesto, L_σ le acompaña.

III-D.3. Estimación distribuida del centroide

Terminamos la sección de herramientas de *source-seeking* con un algoritmo que permitirá a los robots estimar el centroide de la formación. El objetivo final es presentar un resultado que permita calcular todo x_i , es decir, la posición del centroide p_c desde el sistema de referencia propio de cada robot.

El algoritmo que nosotros propondremos surge de una colección de resultados bien establecidos en el control de formación, pues este tipo de técnicas tienen su dual en el problema de localización relativa [39]. En particular, suponiendo que algunos robots del enjambre (más adelante detallaremos a qué nos referimos con "algunos") puedan medir sus distancias relativas, entonces podemos considerar la siguiente ley de localización:

$$\frac{d}{dt} \hat{x}_i(t) = - \sum_{j \in \mathcal{N}_i} \left((\hat{x}_i(t) - \hat{x}_j(t)) - (x_i - x_j) \right), \quad \forall i \in \mathcal{V}, \quad (\text{III.12})$$

donde $(x_i - x_j)$ es el término fijo de la medición hardware, que es equivalente a la posición relativa $(p_i - p_j)$; y $\hat{x}_i \in \mathbb{R}^m$ es el término dinámico es el valor software, que corresponderá a la estimación de la posición del centroide desde el robot i . Téngase en cuenta que, dado que \hat{x}_j es un valor de software, entonces necesita ser comunicado al robot i para todo $(i, j) \in \mathcal{E}$. A continuación, propondremos un resultado que estudia todas aquellas soluciones de (III.12) que nos permitirán estimar x_i .

Proposición III.6. — *La estimación \hat{x}_i en (III.12) converge de forma exponencialmente rápida al valor actual x_i a medida que $t \rightarrow \infty$ si y solo si $\sum_i^N \hat{x}_i(0) = 0$.*

Demostración. En primer lugar, recordemos que el tipo de grafo \mathcal{G} considerado para los siguientes resultados es no dirigido y conectado. Con esto en mente, escribamos la forma compacta de (III.12) como

$$\frac{d}{dt} \hat{x}(t) = -\bar{L}\hat{x}(t) + \bar{L}x \quad \Rightarrow \quad \frac{d}{dt} \hat{x}(t) + \bar{L}\hat{x}(t) = \delta, \quad (\text{III.13})$$

donde L es la matriz Laplaciana (III.4), y $\delta = \bar{L}x = \bar{L}p \in \mathbb{R}^{m|\mathcal{E}|}$ es el vector apilado de las posiciones relativas $(x_i - x_j) = (p_i - p_j)$, $(i, j) \in \mathcal{E}$. Notamos que la dinámica de $\sum_i^N \hat{x}_i(t)$ es estacionaria, es decir, $\sum_i^N \frac{d}{dt} \hat{x}_i(t) = 0$, ya que

$$\mathbf{1}_{mN}^T (-\bar{L}\hat{x}(t) + \bar{L}x) = 0, \quad (\text{III.14})$$

en vista de (III.4) y teniendo en cuenta que $B^T \mathbf{1}_N = 0$. Por lo tanto, si $\sum_i^N \hat{x}_i(0) = 0$, el centroide de $\hat{x}(t)$ permanecerá en cero para todo t . Tenga en cuenta que esta es una condición suficiente y también necesaria para garantizar la invarianza del centroide de $\hat{x}(t)$, ya que $\mathbf{1}_N^T$ es el autovector de L asociado a su único autovalor nulo. También sabemos que la trayectoria eventual $\hat{x}^h(t)$ de la parte homogénea de la ecuación diferencial (III.13) está dada por

$$\lim_{t \rightarrow \infty} \hat{x}^h(t) = c_1 \otimes \mathbf{1}_N, \quad (\text{III.15})$$

ya que $\mathbf{1}_N$ es el autovector asociado al único autovalor cero de L , c_1 depende de la condición inicial $\hat{x}(0)$, y el resto de los términos de $\hat{x}^h(t)$ desaparecen rápidamente de forma exponencial, ya que L es semidefinida positiva².

Ahora, dado $\bar{L}x^p = \delta$, verifiquemos que $\hat{x}^p = x + (c_2 \otimes \mathbf{1}_N)$ es una potencial familia de soluciones particulares para (III.13), dependientes de $c_2 \in \mathbb{R}^m$. Sumando a la solución homogénea este nuevo término x^p , nos queda que

$$\lim_{t \rightarrow \infty} \hat{x}(t) = ((c_1 + c_2) \otimes \mathbf{1}_N) + x.$$

²De hecho, sabemos, a partir del resultado estándar de un protocolo de consenso para grafos no dirigido, que $c_1 = (\sum_{i=1}^N \hat{x}_i(0)) / N$ [42]; no obstante, este resultado no será necesario para nuestro análisis.

Sin embargo, dado que el centroide de x es cero por definición, y el centroide de $\hat{x}(t)$ es cero e invariante debido a (III.14), debe ser cierto que $(c_1 + c_2) = 0$, es decir, x y $\hat{x}(t)$ comparten el mismo marco de coordenadas; por lo tanto, llegamos a $\lim_{t \rightarrow \infty} \hat{x}(t) = x$ exponencialmente rápido.

□

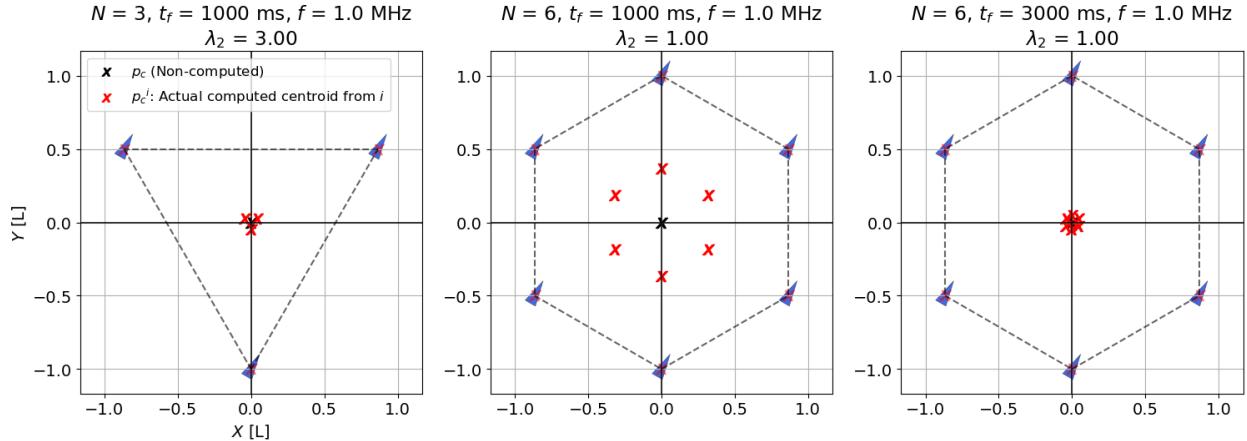


Fig. 28: En esta figura se visualiza, para dos formaciones distintas, la estimación del centroide ("x" roja) realizada por cada uno de los agentes del enjambre. El algoritmo de estimación del centroide se está ejecutando en ambos casos a 1 MHz. Resulta interesante observar que, cuando mayor es la conectividad algebraica del grafo λ_2 , más rápido convergen las estimaciones al valor real del centroide.

La condición $\sum_{i=1}^N \hat{x}_i(0) = 0$ se puede satisfacer estableciendo todos los valores iniciales del software $\hat{x}_i(0) = 0$. Después de que los robots alcancen su formación deseada, si queremos estimar cuándo un grupo calcula su primera dirección ascendente confiable, es importante saber cómo de rápido $\hat{x}(t)$ tiende a su valor asintótico. Esto está determinado por la conectividad algebraica λ_2 en \mathcal{G} , es decir, el menor autovalor distinto de cero del Laplaciano L [38], y por una posible ganancia positiva para (III.12) (ver Figura 28). Mientras que λ_2 está relacionado únicamente con la topología del grafo no dirigido considerado [38], la ganancia positiva está relacionada con el ancho de banda disponible y la implementación en tiempo discreto de (III.12) en hardware; es decir, cuanto más rápido iteremos (III.12), más rápido será la convergencia $\hat{x}(t) \rightarrow x$.

III-E. Simulaciones

A lo largo de toda la sección anterior, hemos ido mostrando pequeñas simulaciones que han permitido verificar numéricamente cada uno de los resultados expuestos. No obstante, en esta sección dedicada exclusivamente a simulaciones, presentaremos una serie de escenarios mucho más complejos que harán uso de forma simultánea de varias de las herramientas presentadas. Todos los gráficos y simulaciones relacionados con la metodología de *source-seeking* pueden encontrarse en [44], un repositorio de GitHub creado y mantenido por el autor de este TFM.

Sim. I: Un enjambre con una cantidad variable de agentes tratando de encontrar la fuente

En esta primera misión (Figura 29), un enjambre de 200 robots con dinámica de integrador simple, distribuidos uniformemente en una formación rectangular, ha de hallar el origen de un campo cuadrático. En su camino hacia la fuente, el equipo se encuentra en $t = 15$ con un conjunto de 15 nuevos agentes que se unen al enjambre. Sin embargo, en $t = 20$ algunas baterías empiezan a fallar, lo que produce que un total de 20 agentes se desconecten de la red.

El objetivo de esta misión es demostrar la **resiliencia** del enjambre ante situaciones en las que, debido a variaciones drásticas el número de agentes, la geometría de la formación se hace muy irregular. A pesar de estas adversidades, el enjambre consigue encontrar la fuente de la señal en $t = 41$.

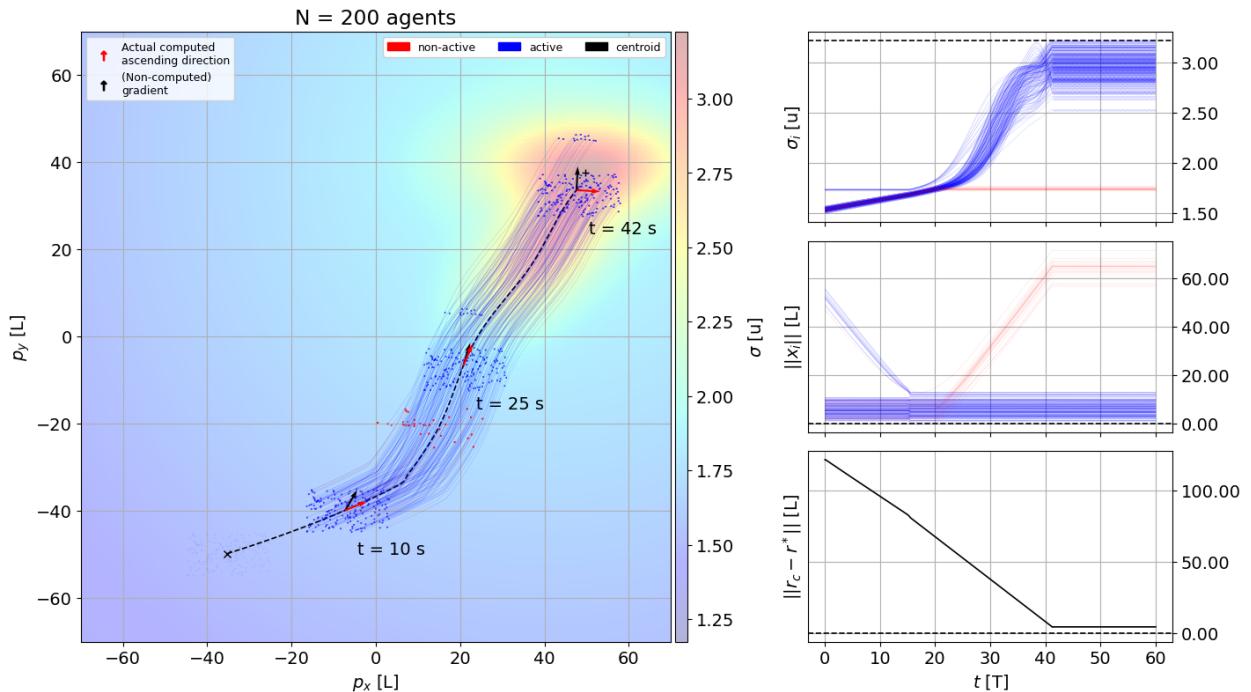


Fig. 29: En esta figura tenemos la simulación (I) de un enjambre de robots ejecutando nuestro algoritmo de *source-seeking*. En el gráfico de la izquierda tenemos una representación del escenario virtual que permite visualizar cómo se aproxima el enjambre a la fuente; mientras que en los gráficos de la derecha se visualiza la distancia del centroide a la fuente y , para todo agente i , la medición σ_i y su distancia al centroide x_i .

Sim. II: La geometría del enjambre cambia para esquivar obstáculos

En esta segunda misión (Figura 30), otro enjambre de 200 agentes, inicialmente distribuidos de forma uniforme en una circunferencia, debe de encontrar la fuente de una señal gaussiana. A lo largo de su camino, el enjambre se encontrará con una serie de obstáculos, de modo que deberá de aplicar algunos de nuestros resultados para lograr esquivarlos mientras se sigue aproximando a la fuente.

El principal objetivo de esta simulación es demostrar el potencial de nuestro algoritmo en términos de **maniobrabilidad**. Como ya comentados al discutir sobre la Proposición III.5, la varianza de la formación permite modular indirectamente L_σ , vector que al fin y al cabo se puede traducir como una **dirección de guiado** para el enjambre. En este caso concreto, al pasar de una formación circular, donde la varianza es igual en todos los ejes (L_σ^1 paralelo a $\nabla\sigma$), a una formación rectangular, el enjambre logra aumentar drásticamente la proyección de L_σ sobre la horizontal, lo que le permite seguir aproximándose a la fuente mientras esquiva a los obstáculos.

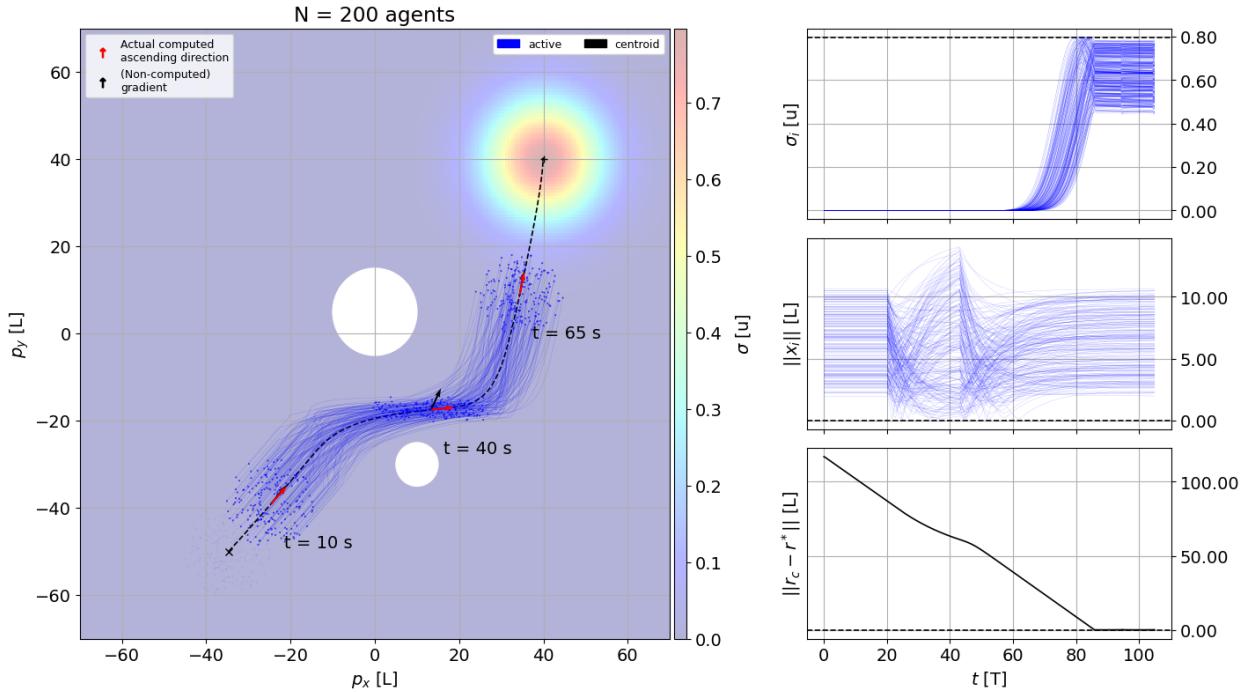


Fig. 30: En esta figura tenemos la simulación (II) de un enjambre de robots ejecutando nuestro algoritmo de *source-seeking*. En el gráfico de la izquierda tenemos una representación del escenario virtual que permite visualizar cómo se aproxima el enjambre a la fuente a la vez que esquiva una serie de obstáculos (círculos blancos); mientras que en los gráficos de la derecha se visualiza la distancia del centroide a la fuente y, para todo agente i , la medición σ_i y su distancia al centroide x_i .

Sim. III: El enjambre rota para adaptarse a un terreno desconocido

En esta tercera misión (Figura 31), tenemos nuevamente a un enjambre de 200 robots, uniformemente distribuidos en una formación rectangular, que ha de encontrar la fuente de una señal gaussiana. En su camino a la fuente, el enjambre se percata de que debe de rotar drásticamente su dirección de movimiento para, por ejemplo, poder adaptarse al entorno mientras se sigue aproximando a la fuente.

El objetivo de esta simulación vuelve a ser ilustrar las capacidades de nuestro algoritmo en cuanto a maniobrabilidad. En este caso, se parte de una formación con una gran varianza en el eje horizontal, para posteriormente pasar a otra formación que traslada esta varianza superior al eje vertical. Es decir, mediante una rotación de 90° en la formación, el enjambre logra rotar también en la misma magnitud angular L_σ , que como ya hemos comentado anteriormente, se puede interpretar como un vector de guiado.

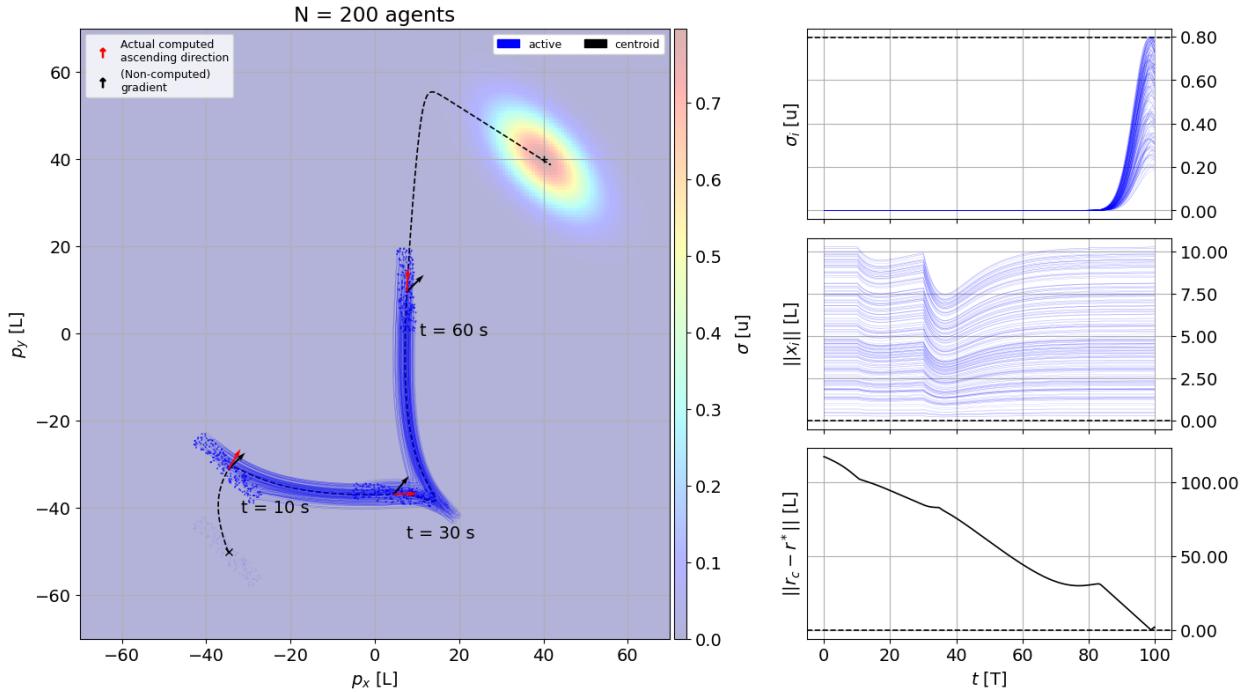


Fig. 31: En esta figura tenemos la simulación (III) de un enjambre de robots ejecutando nuestro algoritmo de *source-seeking*. En el gráfico de la izquierda tenemos una representación del escenario virtual que permite visualizar cómo se aproxima el enjambre a la fuente a la vez que rota su formación; mientras que en los gráficos de la derecha se visualiza la distancia del centroide a la fuente y, para todo agente i , la medición σ_i y su distancia al centroide x_i .

Sim. IV: Enjambre con formación tipo Batman controlado por un campo escalar que rota

La cuarta misión (Figura 32) que proponemos en esta sección es muy peculiar. Tenemos a un enjambre de 150 agentes, distribuido uniformemente en los bordes de una formación que se asemeja al logo de Batman (igual que en la Figura 27), que debe encontrar la fuente de una señal gaussiana con simetría elíptica variante en el tiempo.

En esta simulación, se intenta enfatizar más todavía en el potencial de maniobrabilidad del que dispone nuestro algoritmo. En este caso, el campo es generado y modulado por un agente externo, por lo que se puede interpretar como un **campo de control**; es decir, una herramienta de control sobre el enjambre. A partir de $t = 20$, se puede observar cómo la dirección de movimiento del enjambre rota según lo hace el propio campo.

Este es un resultado numérico con un enorme potencial, pues puede llegar a ser el origen de una nueva metodología para guiar a enjambres de robots de forma controlada, resiliente y robusta. Por ejemplo, imaginemos que tenemos una red de equipos capaces de generar un campo de control sobre sus vecinos; además, supongamos que el equipo 1 puede comunicarse con el 2 y el 2 con el 3. En este escenario, dada una situación en la que el equipo 1 se encuentra en rumbo de colisión con el 3, el equipo 2 podría generar un campo de control para guiar al equipo 1 hasta una zona segura.

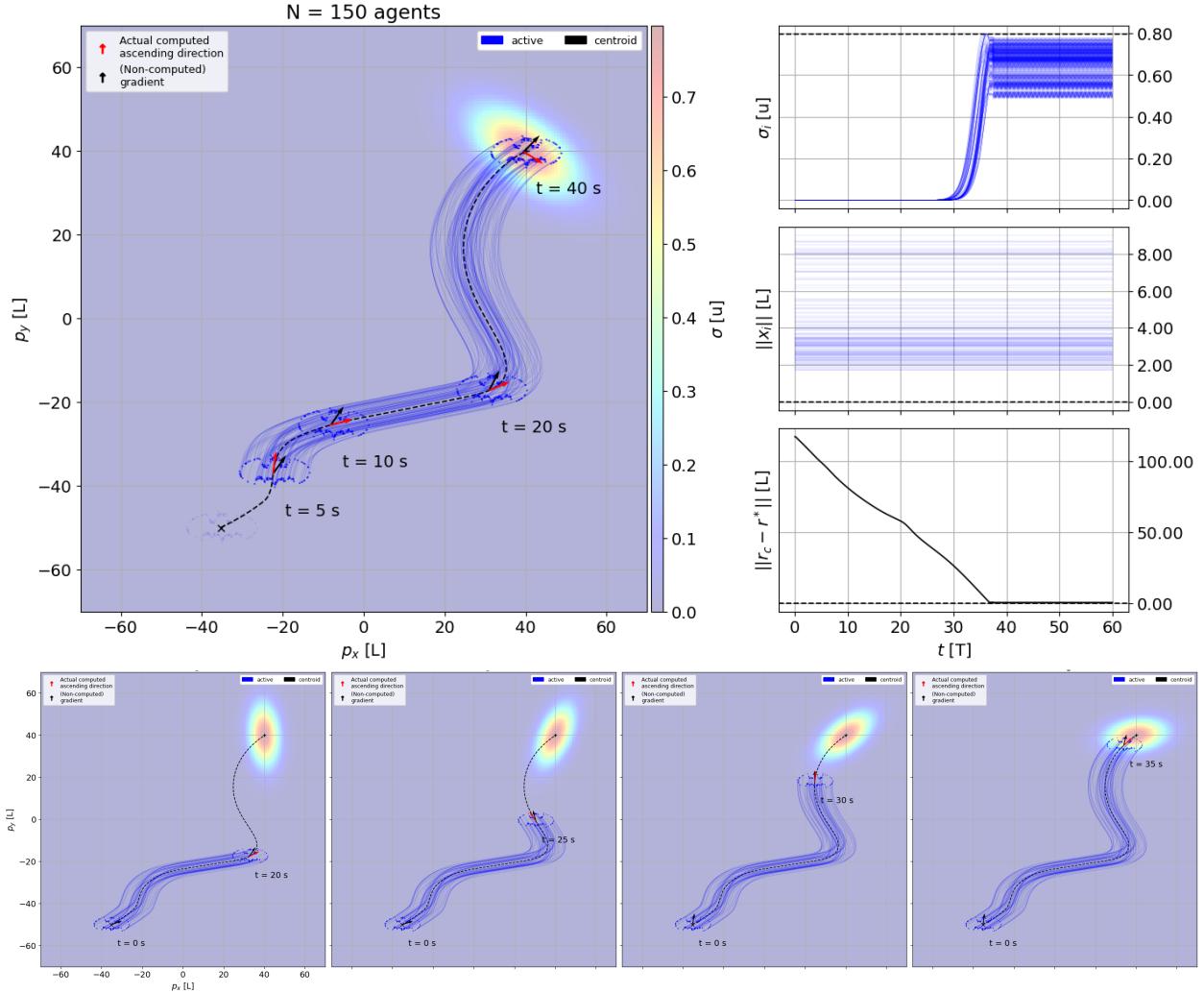


Fig. 32: En esta figura tenemos la simulación (IV) de un enjambre de robots ejecutando nuestro algoritmo de *source-seeking*. En el gráfico de la izquierda tenemos una representación del escenario virtual que permite visualizar cómo se aproxima el enjambre a la fuente de un campo que está rotando; mientras que en los gráficos de la derecha se visualiza la distancia del centroide a la fuente y, para todo agente i , la medición σ_i y su distancia al centroide x_i . Las cuatro capturas de la simulación que aparecen en la parte inferior de la figura intentan visualizar cómo rota entre $t = 20$ y $t = 35$.

Sim. V: Un enjambre de uniciclos en busca de la fuente de un campo cuadrático

En esta quinta misión (Figura 33), contamos con un enjambre, compuesto esta vez por 20 **uniciclos**, que debe de encontrar la fuente de un campo escalar cuadrático. Este caso es mucho más particular que los anteriores, pues ahora la dinámica de los robots no nos permite tomar directamente $\dot{x} = \hat{L}_\sigma$. No obstante, en [35] se demuestra que es posible implementar un controlador similar al utilizado en [9] para alinear a los robots con L_σ .

El objetivo de esta simulación es demostrar numéricamente que nuestro algoritmo de *source-seeking*, también funciona para dinámicas más realistas. Este caso es tan interesante porque dinámica de muchos sistemas reales pueden aproximarse a la de un uniciclo, como los aviones de ala fija (*fixed-wing*) o los vehículos con volante (Subsubsección II-A.2).

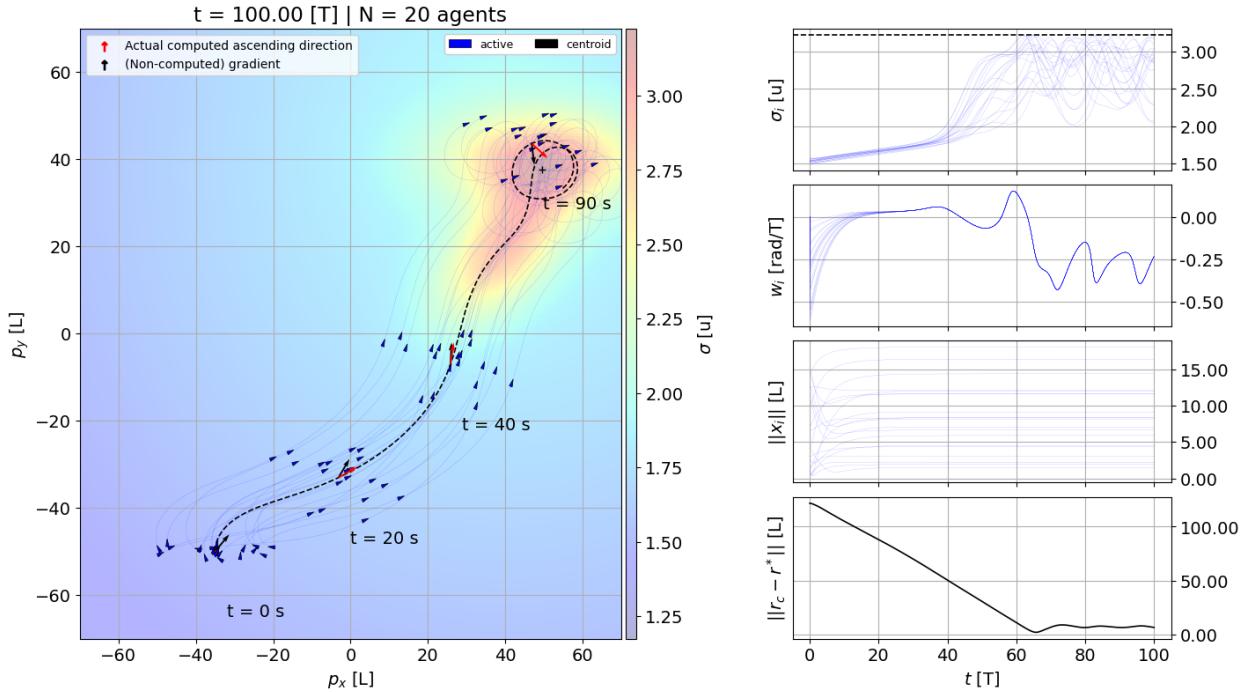


Fig. 33: En esta figura tenemos la simulación (V) de un enjambre de unicilos ejecutando de forma coordinada nuestro algoritmo de *source-seeking*. En el gráfico de la izquierda tenemos una representación del escenario virtual que permite visualizar cómo se aproximan los enjambres a la fuente; mientras que en los gráficos de la derecha se visualiza la distancia del centroide a la fuente y, para todo agente i , la medición σ_i , su velocidad angular ω_i y su distancia al centroide x_i .

Sim. VI: Dos equipos se fusionan para evitar colisionar mientras se aproximan a la fuente

En esta sexta y última misión (Figura 34), tenemos a dos enjambres de uniciclos, cada uno compuesto por 15 agentes distribuidos de forma aleatoria, que deben encontrar la fuente de un mismo campo cuadrático. Dado que ambos equipos se van a dirigir al mismo punto, llegará un momento en el que tengan que fusionarse para evitar colisionar entre ellos.

El objetivo de esta simulación es demostrar que nuestro algoritmo es capaz de gestionar la fusión de varios equipos en un solo enjambre ($t = 60$). A efectos prácticas, esta última misión es equivalente a la primera (Figura 29), ya que la fusión y separación de varios clusters es equivalente a introducir o desconectar agentes de la red. La mayor complicación reside en coordinar a las unidades de cómputo para que calculen una dirección de ascenso común; no obstante, en el Colorario III.1 ya demostramos que es posible y ahora ha quedado verificado numéricamente.

En el contexto de utilizar la fusión de enjambres para evitar colisiones, resulta pertinente pensar en cómo identificar que los enjambres se encuentran en rumbo de colisión. En este caso, sería interesante retomar las herramientas introducidas en la metodología anterior, y diseñar una CBF válida capaz de identificar de forma robusta cuándo dos direcciones de ascenso L_σ están en conflicto.

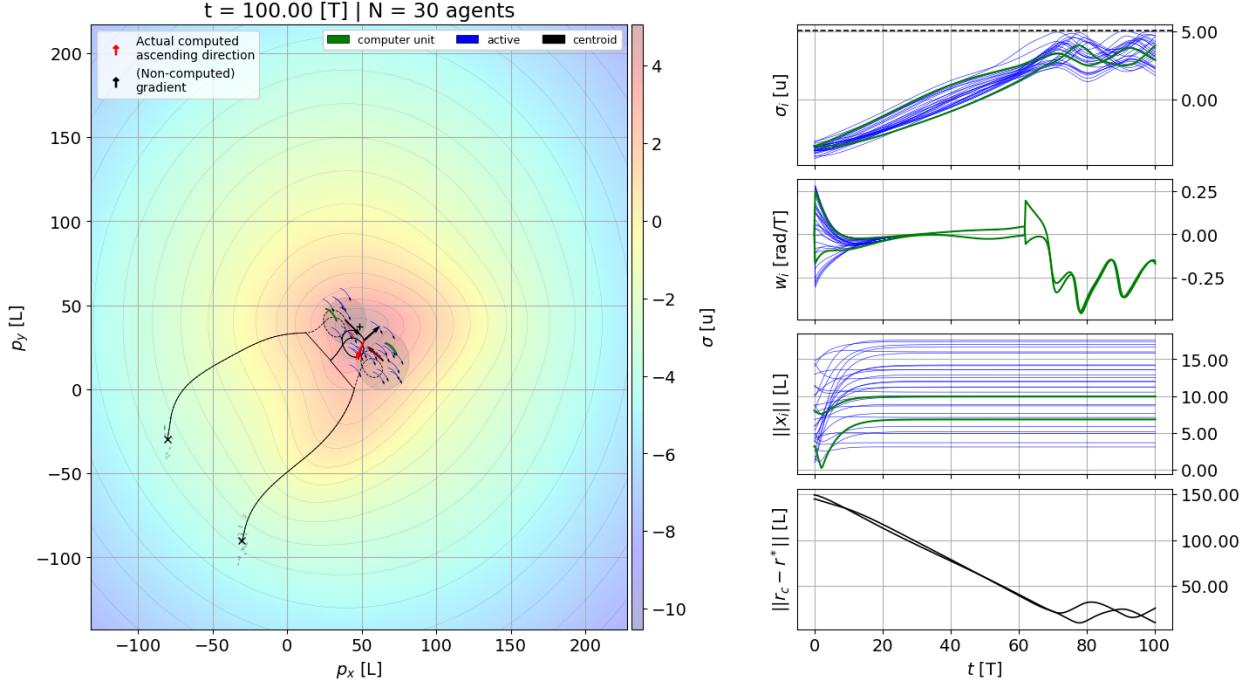


Fig. 34: En esta figura tenemos la simulación (V) de dos uniciclos de robots ejecutando de forma coordinada nuestro algoritmo de *source-seeking*. En el gráfico de la izquierda tenemos una representación del escenario virtual que permite visualizar cómo se aproximan los enjambres a la fuente; mientras que en los gráficos de la derecha se visualiza la distancia del centroide a la fuente y, para todo agente i , la medición σ_i , la velocidad angular ω_i y su distancia al centroide x_i .

III-F. Conclusiones

En esta metodología, nos hemos centrado en analizar una nueva forma de resolver el problema de *source-seeking* con enjambres de robots, propuesta originalmente en [35]. Una solución que, en contraposición con los métodos ya existentes, es capaz de trabajar con geometrías genéricas de la formación, gracias al cómputo una dirección de ascenso. En este trabajo, hemos analizado en profundidad la observabilidad y sensibilidad de dicha dirección de ascenso, proponiendo una serie de resultados teóricos que han permitido identificar varias herramientas para maniobrar al enjambre mientras éste se sigue aproximando a la fuente. Además, en vista a implementar estas técnicas en una flota real de robots, también se ha propuesto un algoritmo capaz de estimar el centroide de forma distribuida, lo cual respeta la filosofía de enjambre.

De cara a un trabajo futuro, se puede pensar en diseñar un experimento que nos permita verificar todos estos resultados sobre un enjambre real. Adicionalmente, aprovechando las propiedades de nuestro algoritmo, se podrían explorar nuevas soluciones para algunos de los grandes problemas actuales dentro de la robótica de enjambre, como la integración local de múltiples equipos basada en las cuatro C's (coordinación, cooperación, colaboración y competición) o la evasión de colisiones.

CONCLUSIÓN FINAL

A lo largo de este trabajo, se han presentado dos metodologías que tratan de lidiar con unos de los problemas más actuales en la robótica de enjambre, como lo son el seguimiento de caminos, la evasión de colisiones y la búsqueda de fuentes en campos escalares. Además, se ha diseñado y construido a nivel de hardware y software una flota de *rovers* autónomos de radio control, los cuales han permitido verificar experimentalmente todos los resultados propuestos en la primera metodología.

REFERENCIAS

- [1] G.-Z. Yang, J. Belligham, P. E. Dupont, P. Fischer, L. Floridi, R. Full, N. Jacobstein, V. Kumar, M. McNutt, R. Merrifield, B. J. Nelson, B. Scassellati, M. Taddeo, R. Taylor, M. Veloso, Z. L. Wang, and R. Wood, “The grand challenges of Science Robotics,” *Science Robotics*, vol. 3, no. 14, p. eaar7650, 2018.
- [2] Z. Lin, L. Wang, Z. Han, and M. Fu, “Distributed Formation Control of Multi-Agent Systems Using Complex Laplacian,” *IEEE Transactions on Automatic Control*, vol. 59, no. 7, pp. 1765–1777, 2014.
- [3] K.-K. Oh, M.-C. Park, and H.-S. Ahn, “A survey of multi-agent formation control,” *Automatica*, vol. 53, pp. 424–440, 2015.
- [4] A. Prorok, M. Malencia, L. Carlone, G. S. Sukhatme, B. M. Sadler, and V. Kumar, “Beyond Robustness: A Taxonomy of Approaches towards Resilient Multi-Robot Systems,” 2021.
- [5] A. Micaelli and C. Samson, “Trajectory tracking for unicycle-type and two-steering-wheels mobile robots,” PhD dissertation, INRIA, 1993.
- [6] A. Ratnoo, P. Sujit, and M. Kothari, “Adaptive optimal path following for high wind flights,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 12 985–12 990, 2011, 18th IFAC World Congress.
- [7] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, “Vector field path following for miniature air vehicles,” *Trans. Rob.*, vol. 23, no. 3, p. 519–529, jun 2007.
- [8] V. M. Goncalves, L. C. A. Pimenta, C. A. Maia, B. C. O. Dutra, and G. A. S. Pereira, “Vector fields for robot navigation along time-varying curves in n -dimensions,” *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 647–659, 2010.
- [9] H. Garcia de Marina, Y. Kapitanyuk, M. Bronz, G. Hattenberger, and M. Cao, “Guidance algorithm for smooth trajectory tracking of a fixed wing uav flying in wind flows,” 05 2017.
- [10] W. Yao, H. G. de Marina, B. Lin, and M. Cao, “Singularity-free guiding vector field for robot navigation,” *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1206–1221, 2021.
- [11] D. Bhattacharjee, A. Chakravarthy, and K. Subbarao, “Nonlinear model predictive control and collision-cone-based missile guidance algorithm,” *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 8, pp. 1481–1497, 2021. [Online]. Available: <https://doi.org/10.2514/1.G005879>
- [12] J. Goss, R. Rajvanshi, and K. Subbarao, “Aircraft conflict detection and resolution using mixed geometric and collision cone approaches,” 08 2004.
- [13] P. Thontepu, B. G. Goswami, N. Singh, S. P. I. au2, S. S. M. G, S. Sundaram, V. Katewa, and S. Kolathaya, “Control barrier functions in ugv’s for kinematic obstacle avoidance: A collision cone approach,” 2023.
- [14] M. Tayal and S. Kolathaya, “Control barrier functions in dynamic uavs for kinematic obstacle avoidance: A collision cone approach,” 2023.
- [15] Jesús B. V., “Multi_C3BF,” https://github.com/jesusBV20/multi_C3BF, 2023.
- [16] Paparazzi Team, “Paparazzi UAS,” <https://github.com/paparazzi/paparazzi>, 2023.
- [17] Jesús B. V., “[GVF Viewer] New PprzGCS widget to paint GVF parametric trajectories and vector fields,” <https://github.com/paparazzi/PprzGCS/pull/15>, 2023.
- [18] Jesús B. V., “Rover steering wheel,” <https://github.com/paparazzi/paparazzi/pull/2845>, 2023.
- [19] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, “Swarm robotics: a review from the swarm engineering perspective,” *Swarm Intelligence*, vol. 7, pp. 1–41, 2013.
- [20] P. Ogren, E. Fiorelli, and N. E. Leonard, “Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment,” *IEEE Transactions on Automatic control*, vol. 49, no. 8, pp. 1292–1302, 2004.
- [21] V. Kumar, D. Rus, and S. Singh, “Robot and sensor networks for first responders,” *IEEE Pervasive computing*, vol. 3, no. 4, pp. 24–33, 2004.
- [22] K. McGuire, C. De Wagter, K. Tuyls, H. Kappen, and G. C. de Croon, “Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment,” *Science Robotics*, vol. 4, no. 35, p. eaaw9710, 2019.

- [23] W. Li, J. A. Farrell, S. Pang, and R. M. Arrieta, "Moth-inspired chemical plume tracing on an autonomous underwater vehicle," *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 292–307, 2006.
- [24] J. N. Twigg, J. R. Fink, L. Y. Paul, and B. M. Sadler, "Rss gradient-assisted frontier exploration and radio source localization," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 889–895.
- [25] M. Dorigo, G. Theraulaz, and V. Trianni, "Swarm robotics: Past, present, and future [point of view]," *Proceedings of the IEEE*, vol. 109, no. 7, pp. 1152–1165, 2021.
- [26] E. Rosero and H. Werner, "Cooperative source seeking via gradient estimation and formation control," in *2014 UKACC International Conference on Control (CONTROL)*. IEEE, 2014, pp. 634–639.
- [27] S. A. Barogh and H. Werner, "Cooperative source seeking with distance-based formation control and non-holonomic agents," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 7917–7922, 2017.
- [28] L. Briñón-Arranz, L. Schenato, and A. Seuret, "Distributed source seeking via a circular formation of agents under communication constraints," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 2, pp. 104–115, 2015.
- [29] L. Briñón-Arranz, A. Renzaglia, and L. Schenato, "Multirobot symmetric formations for gradient and hessian estimation with application to source seeking," *IEEE Transactions on Robotics*, vol. 35, no. 3, pp. 782–789, 2019.
- [30] R. Fabbiano, F. Garin, and C. Canudas-de Wit, "Distributed source seeking without global position information," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, pp. 228–238, 2016.
- [31] Z. Li, K. You, and S. Song, "Cooperative source seeking via networked multi-vehicle systems," *Automatica*, vol. 115, p. 108853, 2020.
- [32] J. Cochran and M. Krstic, "Nonholonomic source seeking with tuning of angular velocity," *IEEE Transactions on Automatic Control*, vol. 54, no. 4, pp. 717–731, 2009.
- [33] S. Al-Abri and F. Zhang, "A distributed active perception strategy for source seeking and level curve tracking," *IEEE Transactions on Automatic Control*, vol. 67, no. 5, pp. 2459–2465, 2021.
- [34] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.
- [35] A. A. Huertos, "Control de sistemas multiagente y sus aplicaciones," 2022.
- [36] A. Clauset, C. R. Shalizi, and M. E. Newman, "Power-law distributions in empirical data," *SIAM review*, vol. 51, no. 4, pp. 661–703, 2009.
- [37] W. Rudin, *Principles of mathematical analysis*, 3rd ed. McGraw-hill New York, 1976.
- [38] F. Bullo, *Lectures on network systems*. Kindle Direct Publishing, 2020, vol. 1.
- [39] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.
- [40] Z. Sun, H. G. de Marina, G. S. Seyboth, B. D. Anderson, and C. Yu, "Circular formation control of multiple unicycle-type agents with nonidentical constant speeds," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 1, pp. 192–205, 2018.
- [41] H. Garcia de Marina, "Distributed formation maneuver control by manipulating the complex Laplacian," *Automatica*, vol. 132, p. 109813, 2021.
- [42] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on automatic control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [43] W. Mendenhall, R. J. Beaver, and B. M. Beaver, *Introduction to probability and statistics*. Cengage Learning, 2012.
- [44] Jesús B. V., "MRS-SS_souce_seeking," https://github.com/jesusBV20/MRS-SS_souce_seeking, 2023.