UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE CIENCIAS FÍSICAS

DEPARTAMENTO DE ARQUITECTURA DE COMPUTADORES Y AUTOMÁTICA

TRABAJO DE FIN DE GRADO

Código TFG: [Código TFG]

[Desarrollo de sistemas de control cooperativos para USVs en tareas de bioinspección]

[Cooperative Control Systems for USVs in Bio-surveillance tasks]

[Ulises Alejandro Ardizzi Rodríguez]

Supervisor/es: [Nombre del/os supervisores]

Grado en Ingeniería Electrónica de Comunicaciones
Curso académico 20[XX-XX]

Convocatoria XXXX

Autorización de difusión

Apellidos, Nombre

Madrid, a XX de XX de XX

Los abajo firmantes, matriculados en el Grado de XX de la Facultad de XX, autorizan a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Grado: "Título", realizado durante el curso académico XX-XX bajo la dirección de XX y la co-dirección de XX en el Departamento de XX, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

"Dedicatoria, si es necesaria."

Edward Tufte

Agradecimientos

Agradecimientos, si son necesarios.

Índice general

Índice de figuras

[Título extendido del TFG (si procede)]

Resumen

Breve resumen de contenidos.

Palabras clave:

Separadas, por, comas.

Abstract

Key works:

[Nota: el título extendido (si procede), el resumen y abstract deben estar en una misma página y su extensión no debe superar la página. Tamaño mínimo 11 pto.
Extensión máxima 50 páginas sin contar portada ni resumen (sí se incluye índice, introducción, conclusiones y bibliografía)]

Capítulo 1

Introducción

1.1. Motivación

Actualmente los sistemas robóticos representan una ventaja al otorgar un mayor rango de acción, flexibilidad y operar en situaciones riesgosas.

Un robot realmente no posee una definición precisa y universal dado que existe bastante discrepancia entre los expertos. Por lo tanto, podrían considerarse como un sistema autónomo y programable capaz de realizar tareas. Además, están dotados por la integración de tres capacidades claves:

1. **Sensores** para reunir datos del entorno.
2. La **toma de decisiones** para convertir dichos datos en acciones.
3. Al ya tener definida su labor deben extenderlas al mundo real a través de sus efectores finales y/o actuadores.

Si juntas dichos aspectos con el comportamiento de los organismos sociales, en donde los individuos no han de tener un alto conocimiento para producir un comportamiento colectivo complejo, ni existir un líder que guía al resto para completar un objetivo, como en los bancos de peces, un panal de abejas o una bandada de pájaros, se tiene la **robótica de enjambre**.

1.1.1. Robótica de enjambre

Hoy en día, la robótica de enjambre conforma un grupo de investigación muy activo por su versatilidad en diferentes ámbitos, tales como militar o industrial. En contraposición de tener un único robot realizando una labor compleja se tiene la robótica de enjambre, en la que varios individuos simples forma un comportamiento colectivo para realizar la misma tarea traduciendo a su vez en una reducción de costes. Las características principales con las que se pueden definir los enjambres son:

1. El número optimo de agentes varia en función de la tarea asignada pudiendo ir desde tan pocos como una simple pareja hasta miles de unidades.

2. Presenta gran **diversidad**, es decir, en ocasiones se mezclan robots simples o complejos, sistemas tripulados o no tripulados, e incluso con dominio cruzado.

3. Para poder diferenciarlos de los sistemas multi-robots, en el que cada robot individualmente tiene una tarea asignada de antemano, los de tipo enjambre han de tener un **comportamiento colectivo** que involucre colaboración entre los propios agentes y estos con su entorno.

4. Se necesita establecer una forma de comunicación entre los agentes para permitir el intercambio de información, esta puede ser implícita o explícita

5. El hecho de que se puede definir su modo de operar no implica que se controle a cada robot individualmente, es decir, cada uno ellos han de poseer un comportamiento **autónomo** y **descentralizado**.

1.1.2. Sistemas multiagentes

Un agente se puede definir como una entidad software que es capaz de realizar una tarea definida de forma autónoma y con cierto grado de complejidad por el hecho de estar dotado de cierto grado de inteligencia.

Los sistemas multiagentes como bien su nombre indica, se basan en un grupo de dos o más agentes que interaccionan entre si para lograr un objetivo común en un mismo entorno. Dicha comunicación puede darse entre vecinos sin necesidad de recurrir a una entidad central, es decir, cada uno de ellos va a poseer un comportamiento autónomo y aun así conocer la existencia del resto.

Por tal motivo, la información va a estar distribuida en cada uno de los agentes con una rol distinto, además, se añade la posibilidad de fallo en cualquiera de ellos. Esto se traduce en un sistema más eficaz, flexible y fiable.

1.2. Planteamiento del problema

Uno de los problemas principales es la coordinación de los agentes para que adopten una simetría concreta, en donde, se van a tener en cuenta la velocidad de cada uno de ellos, su posición con respecto al mundo y a los agentes vecinos o la posibilidad de colisión.

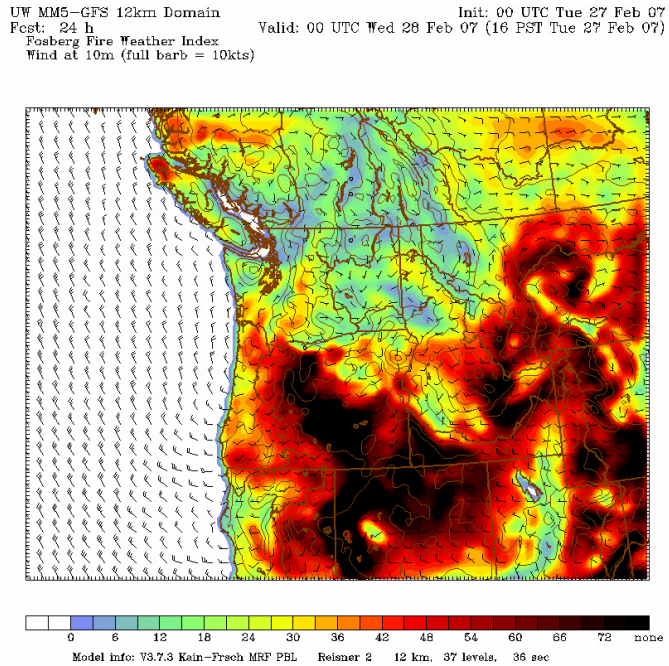


Figura 1.1: *Curvas de nivel descritas por un incendio en Forsberg [Referencia bibliográfica]*

Por otro lado, los agentes han de ser capaces de cumplir su rol asignado una vez dispuestos alrededor de la forma simétrica adoptada, dicha misión consiste en avanzar desde un punto cualquiera a uno de interés que puede ser descrito mediante las **curvas de nivel**.

En la figura 1.1 se aprecian las curvas de nivel de la intensidad de la flama en un incendio, esta información puede ser útil para indicarle al enjambre hacia donde debe moverse para realizar misiones de rescate. El modelado que posteriormente se hará para la resolución del problema se va a basar en dichas curvas.

En base a todo lo descrito anteriormente se va a desarrollar a lo largo de la memoria un sistema multiagente de tipo enjambre que sea capaz de estimar un gradiente a partir de 3 o más robots dispuestos en una formación circular, además de poder desplazarse a una zona de interés.

1.3. Organización de la memoria

En el capítulo dos se dará una idea general del problema global haciendo uso de un diagrama de bloques, además, de describir brevemente diferentes aspectos necesarios para el desarrollo del problema. En el tercero se realiza la estimación previamente descrita. En el cuarto se aporta un algoritmo de control para la coordinación de los agentes de manera simétrica a lo largo de una formación circular. Finalmente, en el quinto y ultimo capítulo se dan los diferentes resultados obtenidos mediante la acción conjunta de ambos algoritmos.

Capítulo 2

Sistema de manera global

Sea una función $f(x)$ con $x \in \mathbb{R}^n$. Además de ser continua y derivable para todo n. Aplicando el desarrollo en serie de Taylor siendo n = 1.

$$f(x) = f(a) + f^{'}(a)(x-a) + \frac{1}{2!} \cdot f^{''}(a)(x-a)^2$$

Donde $f^{'}(a)$ y $f^{''}(a)$ se corresponden con la primera y segunda derivada de la función en torno a un punto cualquiera en el espacio "a", si en lugar de ello se hace con x_*, estando x lo suficientemente cerca de dicho punto.

$$f(x) = f(x_*) + f^{'}(x_*)(x-x_*) + \frac{1}{2!} \cdot f^{''}(x_*)(x-x_*)^2$$

Si $f^{'}(x_*) = 0$ se tiene un máximo, mínimo o un punto de inflexión, teniendo esto en cuenta y despejando de la ecuación anterior.

$$f(x) - f(x_*) = \frac{1}{2!} \cdot f^{''}(x_*)(x-x_*)^2$$

Se dan diferentes situaciones:

- Si $f''(x_*) < 0 \rightarrow f(x) - f(x_*) < 0 \rightarrow f(x) < f(x_*) \rightarrow f(x_*)$ es un máximo.
- Si $f''(x_*) > 0 \rightarrow f(x) - f(x_*) > 0 \rightarrow f(x) > f(x_*) \rightarrow f(x_*)$ es un mínimo.
- Si $f''(x_*) = 0 \rightarrow f(x) - f(x_*) = 0 \rightarrow f(x) = f(x_*) \rightarrow f(x_*)$ es un punto de inflexión.

Si se hace el mismo desarrollo y se expande el dominio para $n \geq 2$, se obtiene:

$$f(x) = f(x_*) + \nabla f(x_*)^T (x - x_*) + \frac{1}{2!} \cdot (x - x_*)^T \cdot H(f(x_*)) \cdot (x - x_*) \qquad (2.1)$$

Donde:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \qquad \text{y} \qquad H(f) = \nabla^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \cdot \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \cdot \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \cdot \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \cdot \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \cdot \partial x_1} & \frac{\partial^2 f}{\partial x_n \cdot \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

En este caso lo que interesa es que el gradiente de la función sea 0, es decir, que "$\nabla f(x_*) = 0$". Por lo tanto, los casos particulares previamente descritos adoptan un significado similar.

$$f(x) - f(x_*) = \frac{1}{2!} \cdot H(f) \cdot (x - x_*)^2$$

- Si $H(f) < 0$ (definida negativa) $\rightarrow f(x) - f(x_*) < 0 \rightarrow f(x_*)$ es un máximo.
- Si $H(f) > 0$ (definida positiva) $\rightarrow f(x) - f(x_*) > 0 \rightarrow f(x_*)$ es un mínimo.

- Si $H(f) = 0$ es indefinida es un punto silla.

En caso de las funciones para dos o más dimensiones, la condición necesaria para ser optimo es estar semidefinido, es decir, si $\nabla f(x_*) = 0$ y $H(f)$ es semidefinida, se tiene:

- Es máximo si esta semidefinida negativa $\rightarrow y^T \cdot H(f) \cdot y \leq 0$
- Es mínimo si esta semidefinida positiva $\rightarrow y^T \cdot H(f) \cdot y \geq 0$

2.1. Algoritmo de estimación del gradiente/de busqueda de fuentes (decidir luego

2.1.1. Descripción general

Se pretende describir un procedimiento para estimar el gradiente de una función $\widehat{\nabla} f(c)$, basándose en mediciones locales de múltiples robots situados de manera simétrica en un espacio de 2D. En dicho procedimiento, se consideran N robots distribuidos uniformemente a lo largo de una formación circular con un radio D y un punto central c definido en dos dimensiones. [poner referencia]

Partiendo de la ecuación 2.1 pero haciendo la expansión únicamente hasta el termino de primer orden sobre cada una de las medidas r_i pertenecientes a la función $f(r_i)$ y despejando el gradiente se llega a:

$$\frac{2}{D^2 \cdot N} \cdot \sum_{i=1}^{N} f(r_i) \cdot (r_i - c) = \underbrace{\nabla f(c) + \varphi(D, c)}_{:=\hat{\nabla} f(c)}$$

Donde se tiene un termino de error $\varphi(D, c)$ que va a corresponder al remanente de orden 2 de la expasión de Taylor..

Para dicho calculo, adopta vital importancia los **algoritmos de tipo consenso** siendo estos un mecanismo que permite a maquinas coordinarse en un entorno distribuido, es decir, encuentran la solución al problema de la comunicación entre diferentes entes aislados con el objetivo de ponerse de acuerdo para realizar una tarea concreta.

Adicionalmente, las funciones sobre las que se basa el algoritmo han de cumplir ser **lipzchiana** cuya definición se describe a continuación. Posterior a ello, en el capitulo 3 se dará un desarrollo más profundo del algoritmo, en donde, se verá toda la base matemática sobre la que se sustenta.

2.1.2. Función lipzchiana

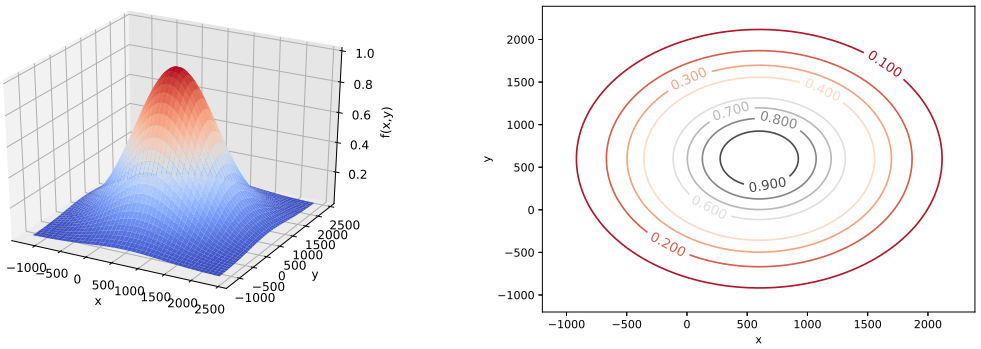
Dada una función $f : U \subset \mathbb{R}^n \to \mathbb{R}^m$, se dice que f es globalmente Lipschitz en el conjunto U cuando existe una constante $L > 0$ tal que

$$|f(x) - f(y)| \leq L|x - y| \iff L \geq |f^{'}(x)| \qquad \forall_{x,y} \in U[referenciabibliografica]$$

La importancia de uso de dicho tipo de función recae en el criterio de convergencia utilizado por el algoritmo, es decir, se garantiza que siempre y cuando la función sobre la que se desplazan los agentes sea lipzchiana estos van a llegar de manera exitosa al punto de interés.

Se destaca que toda función lipzchiana a su vez debe ser cuadrática. No obstante, toda función cuadrático no es lipzchiana, un ejemplo de ello es $x^2 = 0$ la cual no esta acotada en un intervalo de la recta. Por ello, se propone la utilización de distribuciones gaussianas para evaluar la eficacia del algoritmo.

Por ende, se define una distribución gaussiana de la siguiente forma:



(a) Función definida en 3D

(b) Curvas de nivel

Figura 2.1: *Representación de una función lipzchiana*

$$f(x,y) = ke^{-H} \qquad con \;\; H = PSP^T$$

Donde $P = [x, y]$ son sus coordenadas definidas $\forall_{x,y} \in R$ y $S = [S_x, S_y]$ es su desviación siendo una matriz cuadrada y diagonal que mientras más grande sean sus valores más plana será y consecuentemente sus curvas de nivel cubrirán un espacio más amplio, además, su centro va a estar definido como $c = [c_x, c_y]$.

Finalmente, para la simulación del algoritmo se propuso situar el centro en $c = [600, 600]$, tener una desviación $S = \left[\frac{1000}{sqrt2}, \frac{1000}{sqrt2}\right]$ y finalmente un valor de $k = 1$ tal como puede apreciarse en la figura 2.1.

2.2. Algoritmo de control de formación circular

2.2.1. Control de formación

El control de la formación tiene como objetivo evaluar la cooperación y la coordinación en sistemas con múltiples agente, donde su uso recae en impulsarlos a lograr unas restricciones

prescritas en sus estados conllevando a formar y mantener una forma geométrica deseada.

Las características esenciales de este tipo de control están relacionadas con la capacidad de detección y la topología de interacción de la red formada por los agentes. Esto conlleva a diferenciar dos tipos de variables:

- **Detectadas:** Especifican la capacidad de detección de los agentes individuales, un ejemplo de ello sería la velocidad u orientación que posee cada agente.
- **Controladas:** Son aquellas relacionadas con la topología de interacción, es decir, sirve para detallar la mejor formación deseada posible que puedan lograr los agentes.

2.2.2. Descripción general del algoritmo

El problema de control considera vehículos tipo monociclo con velocidad constante, es decir, solo se actúa sobre la dirección del vehículo a través de giros coordinados actuando sobre el ángulo de inclinación lateral de la aeronave. A esto se le añade la velocidad del viento que altera la velocidad constante propia llevaba por cada agente cuyo marco de coordenadas se encuentra fijado en el centro de una formación circular. Sin embargo, se asumirá que la velocidad del viento es mucho menor que la velocidad deseada, por lo que la velocidad respecto al suelo puede considerarse casi constante durante la misión del vehículo. [2].

Se llega a un **algoritmo distribuido para controlar formaciones circulares** de UAV de ala fija con velocidades constantes que actúa sobre el radio del circulo a ser rastreado no sobre la dirección de cada uno de los agentes, es decir, altera la velocidad angular que poseen en torno a un punto central (centro de la circunferencia).

En resumidas cuentas, cada uno de los robots se encontrarán en un punto p^* cualquiera y ha de converger a un centro definido C_r, donde $p^* \cap C_r \in \mathbb{R}^2$. Consecuentemente, mediante

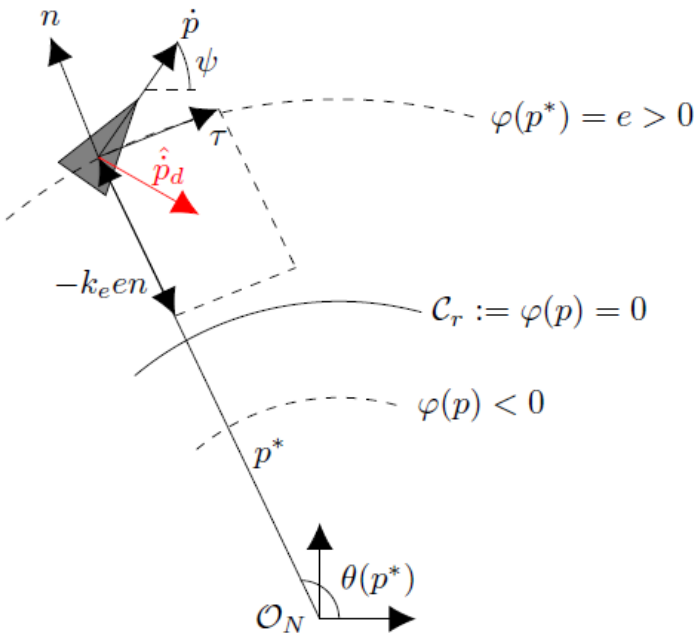


Figura 2.2: *Trayectoria descrita por cada uno de los agentes [Referencia bibliográfica]*

la siguiente ecuación se puede describir los comportamientos adoptados por los agentes.

$$\varphi\left(p\right) = c_i \Longrightarrow x^2 + y^2 - r^2 = c_i \tag{2.2}$$

En donde, $c_i \in \mathbb{R}$ siendo la señal de control que se diseñará y el superindice i denota cada uno de los vehículos/agentes/robots (ver que poner). Se definen dos casos particulares:

- Si c_i se hace muy pequeño, el radio tenderá a aumentar y con ello se reduce la velocidad angular.
- Si c_i se hace muy grande, el radio tenderá a disminuir y con ello se aumenta la velocidad angular.

Ambos casos tienen como objetivo converger a la formación circular definida previamente. Finalmente, en esta sección únicamente se trato al algoritmo de manera simplificada dado

que más adelante, en el capitulo 4, se dará una descripción más detallada de este, en donde se verán dos formas de consenso entre los agentes o el comportamiento propio descrito para cada uno de ellos.

2.3. Acción conjunta de ambos algoritmos

El diagrama de fluyo ira acá

2.3.1. Algoritmo de ascenso de gradiente

Capítulo 3

Estimación del gradiente

Acá la idea es describir el algoritmo con su desarrollo matemático para que no salga nada de la nada (mas o menos 10-12 paginas en este capitulo).

Me quede aca, desde aca para abajo es sucio.

En donde $r_i =$ es la posición del robot i, $\boldsymbol{\phi_i} = \frac{\boldsymbol{2 \cdot \pi \cdot i}}{\boldsymbol{N}}$ es el ángulo de rotación, R_ϕ es la matriz de rotación definida como $\begin{bmatrix} \boldsymbol{c_\phi} & \boldsymbol{-s_\phi} \\ \boldsymbol{s_\phi} & \boldsymbol{c_\phi} \end{bmatrix}$, finalmente $e = [1, 0]^T$, por simplicidad no se considera la dinámica de los robots.

La señal está definida según una función cuadrática $\boldsymbol{\sigma(r\ ) = r^T \cdot S \cdot r + p^T \cdot r + q}$ si se tiene una formación de más de 4 robots se asume que la estimación es el gradiente de la función.

$$\phi_i = \phi_o + \frac{2 \cdot \pi \cdot i}{N}$$

Con $\phi_o(t) = w_o \cdot t$la formación propuesta es adecuada para robots que se mueven en formación circular como vehículos aéreos no tripulados de área.

Problema en cuestión:

Se puede poner de tres formas:

Primera forma:

$$\hat{\nabla} f\left(c\right) := \frac{2}{D^2 \cdot N} \cdot \sum_{i=1}^{N} f(r_i) \cdot (r_i - c)$$

Donde:

$$\hat{\nabla} f\left(c\right) = \nabla f\left(c\right) + \varphi\left(D, c\right)$$

Segunda forma:

$$\frac{2}{D^2 \cdot N} \cdot \sum_{i=1}^{N} f(r_i) \cdot (r_i - c) = \nabla f\left(c\right) + \varphi\left(D, c\right)$$

Tercera forma:

$$\frac{2}{D^2 \cdot N} \cdot \sum_{i=1}^{N} f(r_i) \cdot (r_i - c) = \underbrace{\nabla f\left(c\right) + \varphi\left(D, c\right)}_{:=\hat{\nabla} f(c)}$$

Se tiene una función $f\left(r\right)$, donde r definida en 2 dimensiones, que el gradiente en el punto máximo es 0 ($\nabla f\left(r^*\right) = 0$), pero en el punto del campo escalar será distinto de 0 ($\left(\nabla \sigma\left(r\right) \neq 0\right)$, obviamente se ha de dar con "situaciones espaciales" diferentes lugares ($\forall r \neq r^*$) y finalmente el hessiano estará definido negativamente dado que es un máximo local, es decir, $H_{\sigma(r^*)} < -a \cdot I_p$ (con a > 0 e I_p es una matriz identidad perteneciente al espacio R^{pxp}.

Parrafos que me pueden servir en algun momento:

Se tienen dos casos $\left[A\right]_\alpha = A$ si $A \leq -\alpha \cdot I_p$ sino seria $\left[A\right]_\alpha = -I_p$, el primero de los casos se utilizará si el centro de formación c está muy alejado de la fuente así se evita que la matriz se

defina como semipositiva y tienda a alejar a los robots del punto de interés, además, cuando dicho punto "c" está cerca de la fuente se asume entonces que $A < -\alpha \cdot I_p$

Capítulo 4

Coordinación

Acá la idea es explicar de manera breve el algoritmo de control de Hector, por ejemplo poner una figura del paper y decir que influiría en el calculo del gradiente estimado (mas o menos 5-6 paginas acá).

En la actualidad, uno de los principales problemas de los sistemas multiagente

El objetivo principal de este algoritmo consiste en mantener a los agentes alrededor de un circulos, es decir,

Capítulo 5

Conclusiones

Acá irían las gráficas de las simulaciones que estoy haciendo actualmente (llevaría de momento 1 de indice + 3-4 introducción + 18-20 (en dos capítulos), 6-8 el capitulo 4 y 6-8 el capitulo de conclusiones, mas la bibliográfica, darían en torno a 40-48).

5.1. Resultados experimentales

5.2. Futuras investigaciones

Capítulo 6

Parrafor en sucio

Introducción:

En base a todo lo descrito anteriormente se va a desarrollar a lo largo de los siguientes capítulos un sistema multiagente de tipo enjambre que sea capaz de estimar un gradiente a partir de 3 o más robots dispuestos en una formación circular y estos sean capaces de desplazarse a una zona de interés definido como un máximo de una determinada función.

Capitulo 2: Según lo dicho en la charla de diciembre y en el pdf del campus aca va todo lo explicativo, tipo el optimo de una funcion, las funciones, bases de algoritmos....

Acá habría que definir los consensus algorithms e introducciones generales a varios aspectos (no se si meter lo de abajo al inicio acá), también un diagrama de flujo del control+estimador (este capitulo y el siguiente osea el 2+3 me deberían llevar como 18-20 paginas, llevaría de momento 1 de indice + 3-4 introducción + 18-20 (en dos capítulos))

Parrafos en la parte de control:

Específicamente, si las posiciones de los agentes individuales se controlan activamente, los agentes pueden moverse a sus posiciones deseadas sin interactuar entre sí.

En el caso de que las distancias entre agentes se controlen activamente, la formación de agentes puede tratarse como un cuerpo rígido. Luego, los agentes deben interactuar entre sí para mantener su formación como un cuerpo rígido. En resumen, los tipos de variables controladas especifican la mejor formación deseada posible que pueden lograr los agentes, lo que a su vez prescribe el requisito sobre la topología de interacción de los agentes.

Definir breves conceptos, problemas en la coordinación, descripción de las curvas de nivel para usar la estimación de gradiente.

• Control basado en la posición: los agentes detectan sus propias posiciones con respecto a un sistema de coordenadas global. Controlan activamente sus propias posiciones para lograr la formación deseada, que está prescrita por las posiciones deseadas con respecto al sistema de coordenadas global.

• Control basado en el desplazamiento: Los agentes controlan activamente los desplazamientos de sus agentes vecinos para lograr la formación deseada, que se especifica mediante los desplazamientos deseados con respecto a un sistema de coordenadas global bajo el supuesto de que cada agente es capaz de detectar las posiciones relativas de sus agentes vecinos con respecto al sistema de coordenadas global. Esto implica que los agentes necesitan conocer la orientación del sistema de coordenadas global. Sin embargo, los agentes no requieren conocimiento del sistema de coordenadas global en sí ni de sus posiciones con respecto al sistema de coordenadas.

• Control basado en la distancia: las distancias entre agentes se controlan activamente para lograr la formación deseada, que viene dada por las distancias entre agentes deseadas. Se supone que los agentes individuales pueden detectar las posiciones relativas de sus agentes vecinos con respecto a sus propios sistemas de coordenadas locales. Las orientaciones de los sistemas de coordenadas locales no están necesariamente alineadas entre sí.

Parte de la estimación del gradiente:

$$f(r_i) - f(c) = \nabla f(c)^T (r_i - c) + \varphi_i(D, c)$$

Bibliografía

[1] *INA219 - DataSheet*. http://www.ti.com/lit/ds/symlink/ina219.pdf.

[2] *NodeMCU - Documentación*. https://nodemcu.readthedocs.io/en/dev/.

[3] *MQTT - Documentación*. http://mqtt.org/documentation.

[4] *Mosquitto - Broker*. http://mosquitto.org/.

[5] *Mosca - Broker*. https://github.com/mcollina/mosca.

[6] Node-Red. http://nodered.org/.

[7] I^2C - Module. https://nodemcu.readthedocs.io/en/dev/en/modules/i2c/.

[8] MQTT - Module. https://nodemcu.readthedocs.io/en/dev/en/modules/mqtt/.

[9] Emoncms - Dashboard. https://emoncms.org/.

[10] Freeboard - Dashboard. https://freeboard.io/.

Apéndice A

Introduction

Apéndice B

Conclusions