

SISTEMA DE CONTROL DE UN CONVOY AUTOMÁTICO, CON BASE EXPERIMENTAL

ALINA-TEODORA GHEORGHITA

MÁSTER EN INVESTIGACIÓN EN INFORMÁTICA. FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID



Trabajo Fin de Máster en Ingeniería Informática para la Industria

Curso académico 2012-2013

Convocatoria: Septiembre de 2013
Calificación: Matrícula de Honor (10)

Director:
Dr. José María Girón Sierra

Autorización de difusión

Alina-Teodora Gheorghita

Septiembre de 2013

El/la abajo firmante, matriculado/a en el Máster en Investigación en Informática de la Facultad de Informática, autoriza 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 Máster: “Sistema de control de un convoy automático, con base experimental”, realizado durante el curso académico 2012-2013 bajo la dirección de Dr. Jose María Girón Sierra en el Departamento de Arquitectura, Control y Automática 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.

Resumen en castellano

Los sistemas de robots autónomos capaces de coordinarse entre sí presentan cada día un mayor interés. Algunas aplicaciones de estos sistemas pueden ser el rescate, la vigilancia, la limpieza de vertidos, defensa, etc.. Este Trabajo Fin de Máster se centra en la creación de un sistema de control de un grupo de robots móviles autónomos, terrestres y marinos, capaces de moverse en formación a lo largo de una trayectoria. Se ha empleado un control jerárquico *Null-Space Based*, curvas de Dubins para las trayectorias y una liebre virtual para el guiado. La demostración se realiza mediante varios escenarios experimentales. Los resultados son satisfactorios y el sistema se muestra prometedor.

Palabras clave

Robots móviles autónomos, formaciones, NSB, Dubins

Abstract

The systems of autonomous robots able to coordinate with each other, are gaining more interest every day. Some applications of these systems may be the rescue, surveillance, cleaning up oil spills, defense, etc.. This Master Thesis is focused on the creation of a control system of a group of autonomous mobile robots, terrestrial and marine surface vessels, able to move in formation along a path. It employs a hierarchical *Null-Space Based* control, Dubins curves for the paths and a “virtual target” for the guidance. The results are satisfactory and the system looks promising.

Keywords

Autonomous mobile robots, formations, NSB, Dubins

Índice general

Índice	I
List of Figures	III
List of Tables	VI
Agradecimientos	VII
1. Introducción	1
2. Estado del arte	5
2.1. Libros fundamentales	6
2.2. Aspectos de la Investigación a Nivel de Robot Móvil Individual	6
2.3. Aspectos de la Investigación a Nivel de Agrupaciones de Robots Móviles	20
2.4. Observaciones	34
3. Plataforma experimental	37
3.1. Robots	40
3.1.1. Robots terrestres	40
3.1.2. Robots marinos	42
3.1.3. Placa de control	44
3.1.4. Software de a bordo	54
3.2. Comunicación	56
3.2.1. Robot - Estación de tierra	57
3.2.2. Robot - Robot	58
3.2.3. Estación de tierra - Robot	58
3.3. Estación de tierra	58
3.3.1. Diseño	59
3.3.2. Interfaz de usuario	60
3.4. Simulación	65
3.4.1. Modelo Cinemático	65
3.4.2. Diseño	66
3.4.3. Interfaz Gráfica	68
4. Sistema de a bordo	72
4.1. Sistema de control de un robot	74
4.1.1. Navegación	75
4.1.2. Guiado	78

4.1.3. Control	80
4.2. Sistema de control de un convoy	84
4.3. Sistema de control de formaciones	86
4.3.1. Ruta y Seguimiento	88
4.3.2. Formaciones	90
4.3.3. Transiciones	92
4.3.4. Control jerárquico <i>Null-Space Based</i>	93
5. Trabajos Experimentales	98
5.1. Bases del estudio experimental	100
5.1.1. Preparación de un experimento	102
5.1.2. Realización de un experimento	103
5.1.3. Análisis de un experimento	103
5.2. Comportamiento individual	104
5.2.1. Robots terrestres	104
5.2.2. Robots marinos	108
5.2.3. Inserción en la pista	110
5.3. Convoy automático	112
5.3.1. Primer experimento	112
5.3.2. Segundo experimento	115
5.4. Formaciones	118
5.4.1. Primer experimento	119
5.4.2. Segundo experimento	121
5.4.3. Tercer experimento	123
6. Conclusiones y trabajos futuros	130
Bibliography	141

Índice de figuras

2.1. Arquitectura deliberativa	7
2.2. Arquitectura reactiva	8
2.3. Arquitecturas de control basadas en comportamientos	9
2.4. Arquitectura híbrida	10
2.5. Arquitectura reactiva Dynamic Expanding Zones	12
2.6. Arquitectura de control híbrida	14
2.7. Máquina de estados generada por el planificador	14
2.8. Eficacia de un robot en función del nivel de autonomía y el grado de desaten- ción por parte del usuario	18
2.9. Esquema del control comportamental NSB	19
2.10. Comparación entre los métodos competitivo, cooperativo y NSB	20
2.11. Formación esquivando un obstáculo con NSB	26
2.12. Ejemplos de formaciones	27
2.13. Ejemplo de transición	28
2.14. Clasificación de los tipos de interacción	29
2.15. Aplicaciones de alta prioridad cooperativa para la seguridad vial	33
3.1. Localización de los entornos experimentales	37
3.2. Plataforma experimental para los robots terrestres	38
3.3. Plataforma experimental para los robots marinos	39
3.4. Electromecánica de los robots terrestres	40
3.5. Robots terrestres	41
3.6. Electrónica de los robots terrestres	41
3.7. Electromecánica del barco	42
3.8. Robots marinos	43
3.9. Electrónica de los robots marinos	43
3.10. La placa electrónica	44
3.11. Interfaces de comunicación	45
3.12. Microcontrolador AT91SAM7S256	46
3.13. Receptor GPS Lassen iQ de Trimble	47
3.14. Magnetómetro HMC5883L de Honeywell	48
3.15. Interpretación de la brújula	49
3.16. Datos brújula originales y corregidos	50
3.17. Grados de declinación	50
3.18. Módulo radio XBee-Pro S1 de Digi	51
3.19. Actuadores: servo y variador+motor	52
3.20. Baterías de la electrónica y electromecánica	53

3.21. Software de a bordo: bucle principal	55
3.22. Red local Zigbee	57
3.23. Diagrama de clases	60
3.24. Estación de tierra - Pestaña Setup	61
3.25. Estación de tierra - Pestaña Controller	62
3.26. Estación de tierra - Pestaña Formation Experiment	63
3.27. Sistemas de referencia “mundo” y “robot”	66
3.28. Módulos del simulador	67
3.29. Simulador - Vistas Simulation y Experiment	69
3.30. Simulador - Vista Graphics	70
3.31. Simulador - Vista 3D	70
4.1. Sistema GNC	73
4.2. Sistema del robot autónomo	75
4.3. Sistemas de coordenadas	76
4.4. Cálculo de la orientación actual	78
4.5. Cálculo de la orientación deseada	79
4.6. Diagrama de flujo del seguimiento de una ruta basado en waypoints	80
4.7. Diagrama de un control PID	81
4.8. Diagrama del control PID de rumbo	82
4.9. Diagrama del control PID de velocidad	83
4.10. Cálculo de la distancia entre el líder y el seguidor	85
4.11. Diagrama del control PID de velocidad	85
4.12. Sistema del robot autónomo	87
4.13. Rutas de Dubins	88
4.14. Ejemplo de ruta	89
4.15. Cálculo de las posiciones deseadas	91
4.16. Transición	92
4.17. NSB: combinación de las velocidades de cada comportamiento	96
5.1. Metodología en espiral	99
5.2. Localización de la pista	101
5.3. Experimento cuadrado	106
5.4. Experimento barrido	107
5.5. Estanques	108
5.6. Experimento barco rojo	109
5.7. Experimento Pista	111
5.8. Experimento Seguidor	113
5.9. Experimento Seguidor	114
5.10. Experimento Seguidor - trazas	116
5.11. Experimento Seguidor - distancia	117
5.12. Experimento Seguidor - velocidad	117
5.13. Experimento Dubins NSB - traza	120

5.14. Experimento Dubins NSB - traza	122
5.15. Experimento Transiciones - ruta	124
5.16. Experimento Transiciones - trazas	125
5.17. Experimento Transiciones - errores	126
5.18. Experimento Transiciones - velocidades	126
5.19. Experimento Transiciones - errores de velocidad	127
5.20. Experimento Transiciones - errores de rumbo	128
5.21. Experimento Transiciones - imágenes	129
6.1. Temas abordados	132
6.2. Sistema experimental	132
6.3. Estudios experimentales	132

Índice de cuadros

2.1. Análisis de las arquitecturas de los sistemas de control	10
3.1. Medidas del robot terrestre	42
3.2. Medidas del robot marino	44
3.3. Componentes de la electrónica	45
3.4. Paquetes TSIP recibidos periódicamente desde el receptor GPS	48
4.1. Configuración del controlador PID de rumbo	82
4.2. Configuración del controlador PID de velocidad	83

Agradecimientos

Quiero dar las gracias a todas las personas del Grupo ISCAR-DACYA que han participado en la investigación realizada, por invertir su tiempo y conocimientos, para ayudarme a completar mi trabajo.

Un sincero agradecimiento a mi director de proyecto, Dr. José María Girón Sierra, por la ayuda ofrecida durante todo este tiempo y por haberme introducido en el campo de la investigación de la robótica. También le estoy agradecida por haberme facilitado los medios necesarios, los cuales me han permitido realizar este Proyecto Fin de Máster.

Muchísimas gracias a Dr. Juan Francisco Jiménez Castellanos por su disponibilidad, paciencia y la famosa letra roja. También les estoy agradecida a Dr. Segundo Esteban San Román y Dr. Santiago Cifuentes Costa por la ayuda que me han dado en varios momentos de la investigación.

Gracias a César Aguilar por la recomendación que hizo posible mi colaboración en este proyecto. También les agradezco a Fernando José Pereda Garcimartín y Hector García de Marina por el muy buen trabajo que han realizado y que fue mi base de partida. Muchas gracias también a Guillermo Ángulo Aragon por todas las ideas, consejos y ayuda que me ha ofrecido durante mi investigación. Sus conocimientos físicos y mecánicos han hecho posibles los trabajos experimentales realizados.

Los más sinceros agradecimientos a mis padres y a mi pareja sentimental por estar siempre a mi lado y por el apoyo y comprensión que me han ofrecido durante los momentos difíciles.

Capítulo 1

Introducción

Desde hace varios años, el Grupo ISCAR (Ingeniería de Sistemas, Control Automático y Robótica) del Departamento de Arquitectura de Computadores y Automática, Universidad Complutense de Madrid, viene preparando aspectos metodológicos y experimentales para participar en el Proyecto Europeo “NECSAVE”. Dicho Proyecto ha comenzado oficialmente en Abril del 2013, y tiene como objetivo la cooperación automatizada de grupos de robots móviles, en tierra, mar y aire. El Proyecto considera varios escenarios de demostración experimental. En nuestro caso nos interesa especialmente el de un grupo de embarcaciones capaces de moverse en formación sin intervención humana.

El papel del Grupo ISCAR en el Proyecto NECSAVE es el de crear plataformas de simulación y desarrollo que acompañen a las diversas etapas del Proyecto. Las simulaciones tienen lugar en dos formatos: uno de ellos sobre ordenador con diversas representaciones gráficas, el otro consiste en embarcaciones a escala para pruebas experimentales. El Grupo ISCAR también contribuye proponiendo y desarrollando metodologías para el control autónomo de robots marinos, terrestres y aéreos y para automatizar la cooperación entre ellos.

Puesto que los ensayos experimentales con barcos a escala presentan evidentes dificultades de logística, de realización y de solventar emergencias, se ha preferido insertar un tipo paralelo de ensayos experimentales, utilizando robots móviles terrestres. La idea es que tanto barcos como robots terrestres se vean gobernados por los mismos sistemas electrónicos, y con programas de ordenador muy similares. Evidentemente los experimentos con robots

terrestres son más fáciles de realizar, y conllevan menos dificultades y emergencias. Una vez que se haya podido probar una metodología con los robots terrestres, se comprobará con las embarcaciones a escala. Por ejemplo, haciendo que los vehículos móviles vayan como un convoy automático y efectúen por sí mismos un recorrido marcado con varios puntos de paso.

Para efectuar los trabajos descritos en esta Memoria, se ha puesto en marcha un escenario experimental. Dicho escenario incluye una estación de tierra o base de supervisión a distancia (BSD) y varios robots móviles, tanto terrestres como navales. En paralelo, se ha desarrollado un Sistema de Simulación con base Geo-referenciada (SSG) que permite realizar experimentos simulados y revisar los correspondientes experimentos reales.

A lo largo de los experimentos realizados se han podido detectar diversos problemas. Varios se refieren a la navegación autónoma de cada robot en un determinado lugar, de modo que sean capaces de moverse adecuadamente según los recorridos deseados, como por ejemplo una pista de atletismo. Después, la coordinación entre robots, de modo que no choquen, que mantengan la formación deseada, y que reaccionen bien ante dificultades. Solucionar estos problemas reales, que aparecen en los experimentos, ha supuesto un intenso trabajo.

Al comienzo de nuestro trabajo se contaba con un robot móvil terrestre, con dos embarcaciones a escala (pequeñas zodiac), y una primera generación de la BSD. Los autores del software ya desarrollado habían pasado a trabajar fuera de la Universidad. Pudimos comprobar que, si bien era posible realizar algunos experimentos con los barcos, muy poco se había hecho en cuanto a cooperación automatizada. En cuanto al robot terrestre, las pruebas experimentales mostraron evidentes problemas de control de navegación, que impedían poder hacer un recorrido especificado.

La primera fase de mi trabajo consistió en poner a punto un buen software de control autónomo de navegación para el robot terrestre. Durante los experimentos se vió también la necesidad de mejorar sustancialmente la BSD, adecuándose mejor a las necesidades que

surgen en el campo.

Tras un esfuerzo de familiarización con la tecnología y con el tipo de software que se maneja, realicé los nuevos desarrollos de software y se comprobaron experimentalmente, consiguiendo la navegación autónoma del robot.

Al cabo de un tiempo, se pusieron a punto dos nuevos robots terrestres, de modo que se hacía posible realizar experimentos conducentes al convoy automático. Esto requirió aumentar considerablemente la funcionalidad de la BSD.

En paralelo, fui desarrollando un entorno de simulación que permitiera probar software de navegación y de cooperación automatizada tanto en escenarios terrestres, como acuáticos. Esta simulación debe corresponderse bien con lo que sucede experimentalmente; de hecho, se emplea también para representar en forma animada, en pantalla, lo que ha sucedido en los experimentos (gracias a los datos recogidos y almacenados por la BSD en los experimentos reales).

Una vez que fue posible realizar experimentos de cooperación, comenzó una larga campaña experimental, realizando un desarrollo incremental de nuevas funcionalidades en paralelo con comprobaciones experimentales. Es en esta etapa en la que he introducido aportaciones metodológicas concernientes a la cooperación automatizada.

El contenido de la Memoria obedece a las diversas etapas del trabajo que acabamos de resumir: sistema experimental, simulación, funcionalidad, experimentación.

El Capítulo 2 hace un estudio del estado del arte con respecto al tema tratado.

El Capítulo 3 describe la plataforma experimental utilizada para desarrollar este Trabajo.

El Capítulo 4 ofrece una descripción del sistema de control implementado para el seguimiento de la ruta y la cooperación entre los robots.

En el Capítulo 5 se presentan los trabajos experimentales realizados tanto con los robots terrestres como navales.

El Capítulo 6 concluye este Trabajo.

Los resultados alcanzados son satisfactorios. El sistema experimental ha mostrado su

validez y eficacia. Las pruebas experimentales finales permiten comprobar que los robots móviles son capaces de seguir una formación prescrita, y recorrer usando un sistema de navegación autónoma las trayectorias requeridas.

En trabajos futuros se aumentará el número de robots móviles, su funcionalidad se verá incrementada, y se piensa en escenarios de cooperación entre robots aéreos, terrestres y navales.

Capítulo 2

Estado del arte

En el presente trabajo se plantea la realización experimental de agrupaciones de robots móviles autónomos, teniendo en cuenta diversos escenarios. Uno de los problemas concretos es el de los convoyes automáticos, que implica que los robots vayan en una formación.

De acuerdo con la línea de investigación del Grupo ISCAR del Departamento de Arquitectura de Computadores y Automática, nos interesan en particular las agrupaciones de barcos autónomos. En el caso de realizar formaciones, nos interesan la columna y el frente (línea).

Con el fin de facilitar la investigación, se han construido varios robots móviles terrestres, con ruedas (carros). La idea es poder probar los métodos de control primero en tierra, y cuando se tiene cierta garantía de buen funcionamiento poder probar esos métodos en barcos robot autónomos. Disponemos de tres carros robot, y de dos barcos robot. Ambos tipos de robot son de escala reducida, lo que los hace más manejables, facilitando los experimentos.

Dada la naturaleza de nuestro trabajo, podemos contar con varios tipos de fuentes conceptuales y de información pertinentes. Se citan en primer lugar los libros que sustentan los fundamentos de la temática abordada. Después, se hace una sucinta revisión de la literatura científica, considerando dos niveles principales: robot móvil individual, y agrupaciones de robots móviles.

2.1. Libros fundamentales

El libro⁶⁶ trata de las tecnologías para realizar robots móviles. Dentro de este aspecto, en⁵³ Mohammed Majid se centra en los robots con ruedas, considerando también aspectos de control.

Uno de los paradigmas más utilizados en cuanto a control de un robot móvil, es el que consiste en una jerarquía de dos bloques; el bloque en el nivel inferior corresponde al piloto de un barco, mientras que el bloque en el nivel superior corresponde al comandante. Hoy día se intenta combinar de forma adecuada, en el control de movimientos de un robot, un nivel deliberativo y un nivel reactivo. El piloto, que debe por ejemplo evitar obstáculos, realiza el nivel reactivo; mientras que el comandante, el deliberativo. Este tipo de cuestiones se trata en⁹.

En principio, el robot móvil debe ejecutar alguna tarea, como simplemente pasar por una serie de puntos (en inglés *'waypoints'*). Esto lleva consigo dos aspectos, uno es el de la planificación de recorridos, si es el caso, y otro el de navegación. También en principio, la planificación se realiza antes de llevar a cabo los experimentos, mientras que la navegación tiene lugar en el tiempo experimental. El tema de la planificación se trata de una forma exhaustiva en⁴⁵. En cuanto a aspectos de navegación pueden consultarse por ejemplo²³ o bien²⁸.

Con respecto a agrupaciones y formaciones de robots móviles, se han propuesto diversos enfoques y metodologías. Son dignos de mención los libros⁴⁹ y⁶⁹.

Finalmente, es interesante mencionar el libro⁶⁵, que trata sobre robots marinos autónomos.

2.2. Aspectos de la Investigación a Nivel de Robot Móvil Individual

Resumo a continuación aspectos de interés para nuestra investigación que aparecen en artículos científicos y en ponencias de congresos.

a) Arquitecturas de control de a bordo

Las arquitecturas de control de los robots autónomos se podrían clasificar en tres categorías: deliberativo, reactivo e híbrido. En ⁵⁶ se ofrece una revisión sobre estas arquitecturas de control analizando su importancia, sus ventajas y desventajas. A continuación se resumen las principales ideas.

Navegación deliberativa (centralizada) La arquitectura de navegación deliberativa es el esquema más antiguo en la inteligencia artificial. Ésta crea un modelo global del entorno a través del sistema de sensores o de la información proporcionada por el usuario. La planificación deliberativa busca la ruta óptima y genera el plan para alcanzar el objetivo. Teniendo en cuenta el modelo estático del entorno y el plan generado, se realiza una acción. Después de cada acción el robot para y actualiza la información para poder calcular el siguiente movimiento.

Esta arquitectura puede coordinar múltiples objetivos y restricciones en un entorno complejo. Sin embargo, se necesita disponer de un modelo del entorno correcto para poder planificar una ruta global factible. Desde el punto de vista del cómputo, para realizar los cálculos necesarios se necesitan una gran capacidad de procesamiento y de memoria.

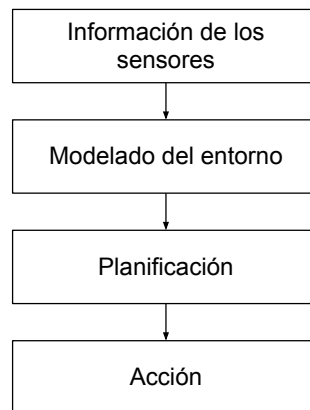


Figura 2.1: *Arquitectura deliberativa*

En la figura 2.1 está representado el diagrama de la arquitectura de navegación deliberativa. Este enfoque de planificación produce retardos en el proceso de navegación y si algún módulo no funciona correctamente, puede fallar el sistema entero.

Por tanto, esta arquitectura no es adecuada en presencia de incertidumbres en la dinámica o en el mundo real.

Navegación reactiva (basada en comportamientos) La arquitectura de navegación reactiva se desarrolló a partir de los años 1980 para afrontar los defectos del enfoque deliberativo en entornos dinámicos o desconocidos. Esta arquitectura genera comandos de control basados en el entorno actual percibido, por lo cual no es necesario construir un modelo entero del entorno. Los datos obtenidos por los sensores se acoplan a los actuadores del robot utilizando una serie de funciones de transferencia llamadas comportamientos.

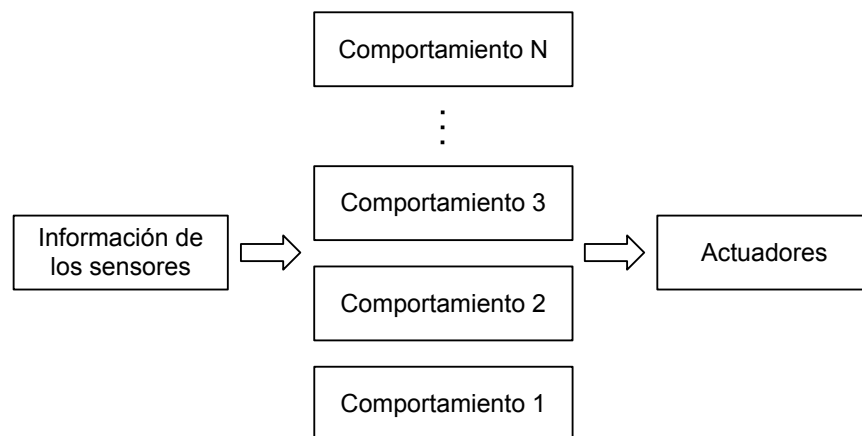


Figura 2.2: *Arquitectura reactiva*

En la figura 2.2 se presenta la arquitectura de los enfoques basados en comportamientos. En la primera capa se obtienen los datos de los sensores. A continuación la función de transferencia de un comportamiento recibe algunos de estos datos y los transforma en una respuesta predefinida. El siguiente paso consiste en ejecutar una acción basada en la salida del comportamiento activo en ese instante.

Con este método, las tareas de navegación complejas se dividen en varias tareas simples que mejoran el rendimiento total del sistema de navegación.

Dos arquitecturas básicas de control basado en comportamientos son la *competitiva* (*subsumption*) y la *cooperativa* (*motor schema*). En una arquitectura competitiva la coordinación de los comportamientos se basa en las prioridades de cada uno, activando solamente un comportamiento. En una arquitectura cooperativa cada comportamiento produce una salida en forma de vector que luego se combinan y se obtiene la salida final como el vector suma, controlando simultáneamente distintas actividades.

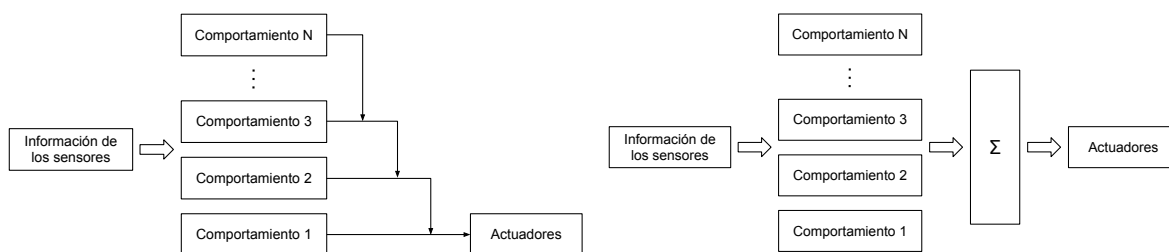


Figura 2.3: *Arquitecturas de control basadas en comportamientos: competitiva y cooperativa*

Navegación híbrida La arquitectura de navegación híbrida presenta las ventajas de la planificación en arquitecturas deliberativas y la respuesta rápida de las arquitecturas reactivas en entornos dinámicos o desconocidos. Las arquitecturas híbridas se han desarrollado en los últimos años aportando nuevas propuestas para sistemas de control de supervisión que utilizan arquitecturas de control reactivas y deliberativas.

En la figura 2.4 se pueden distinguir las tres capas de la arquitectura de navegación híbrida: capa deliberativa, ejecución del control y capa reactiva.

La tabla 2.1 muestra una comparación de los resultados del análisis de las distintas arquitecturas. Se puede concluir que el esquema híbrido es la arquitectura de control con mejor funcionamiento ya que es capaz de tratar con el problema del entorno desconocido o dinámico.

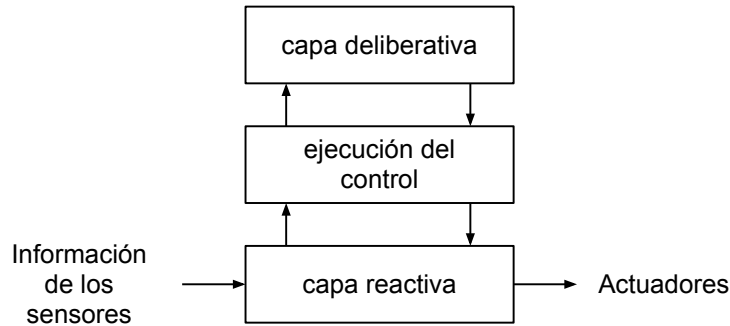


Figura 2.4: *Arquitectura híbrida*

Aspecto	Deliberativo	Reactivo	Híbrido
Orientado a objetivos	Muy bueno	Malo	Bueno
Flexibilidad	Muy malo	Muy bueno	Muy bueno
Facilidad de aplicación	Muy malo	Muy bueno	Bueno
Reacción	Muy malo	Muy bueno	Bueno
Operación óptima	Muy bueno	Muy malo	Bueno
Aprendizaje de tareas	Muy bueno	Moderado	Moderado
Robustez	Malo	Bueno	Muy bueno
Planificación	Muy bueno	Malo	Bueno
Eficiencia	Malo	Muy bueno	Muy bueno

Cuadro 2.1: *Análisis de las arquitecturas de los sistemas de control realizado en*⁵⁶

Referencias fundamentales En el contexto de las arquitecturas de control de robots, las referencias fundamentales, podemos decir clásicas, son Ronald C. Arkin⁷⁻⁹, R. Brooks^{17,18} y Maja J. Mataric^{51,52}.

Arkin propuso una arquitectura basada en comportamientos según la que el denominada *behaviour-based systems*, algo parecido a los instintos de los animales. Brooks por su parte propone la arquitectura *subsumption* que consiguió éxitos muy esperanzadores en cuanto al control efectivo de los robots. Mataric presenta distintos enfoques de las arquitecturas basadas en comportamientos y los aplica a robots móviles capaces de navegar, evitar obstáculos, planificar rutas.

Nuestro caso Estamos considerando inicialmente un entorno sin obstáculos salvo a los demás robots de la formación. Así pues, dentro de esta estructura nos situamos en la categoría de las arquitecturas híbridas donde la capa deliberativa presenta la ruta deseada y la reactividad se encarga de mantener la seguridad de la formación.

b) Perspectiva cognitiva

En¹⁴ se consideran facetas de la robótica que tienen que ver con inteligencia artificial. Concretamente se trata de emergencia de conductas al situar al robot en ciertos entornos que pueden ser colectivos de robots. En este contexto interesa considerar acciones situadas (*situated actions*).

Acciones situadas En la literatura científica cognitiva el lenguaje está visto como un canal de comunicación donde un significado está codificado en una señal física utilizando una gramática y luego decodificado por el oyente en una representación interna similar. Una perspectiva diferente propone Lucy A. Suchman bajo el nombre de *situated actions*⁶⁸.

Emergencia del comportamiento En el campo de la robótica inteligente la emergencia quiere decir que el comportamiento de un robot no se define explícitamente en el controlador si no que depende de la interacción entre el robot y su entorno.

Sistemas cognitivos Se plantea un sistema llamado “la mente extendida” (*the extended mind*). El concepto utiliza la noción de agente para describir el robot y se propone dividirlo en sub-agentes más pequeños con distintos sensores y actuadores. La función de cada sub-agente puede ser muy distinta con respecto a la función del agente completo, tal que ninguno por separado podría alcanzar el objetivo. La cuestión consiste en cómo se deben representar y comunicar los distintos sub-agentes.

Un aspecto importante, inspirado en la cognición distribuida, trata de ver los procesos cognitivos como algo que influye tanto en el entorno como en el individuo. En el caso de los

grupos de individuos, el sistema cognitivo está expandido no solamente para incorporar a una persona si no a más personas y artefactos en cooperación. La cognición distribuida se ha aplicado en mucho sistemas como navegación para barcos³⁹, control de aviones^{40,41}, etc.

c) Ejemplo de navegación de un robot autónomo

La tesis de máster⁶¹ presenta un sistema de navegación reactivo para un robot autónomo terrestre con el objetivo participar en la competición DARPA Grand Challenge. Los principales requisitos consisten en poder seguir la ruta a gran velocidad (entorno a $20m/s$) con distintas condiciones de terreno, evitar obstáculos, pasar por tramos cortos de túnel.

El equipamiento del robot consiste en una serie de sensores: un receptor GPS para la localización, una unidad IMU para estimar la posición dentro del túnel, tres módulos LADAR para detección de obstáculos y una cámara de visión estéreo para detección del camino. Además presenta tres módulos de procesamiento para el sistema de navegación, el sistema de visión y el sistema de control.

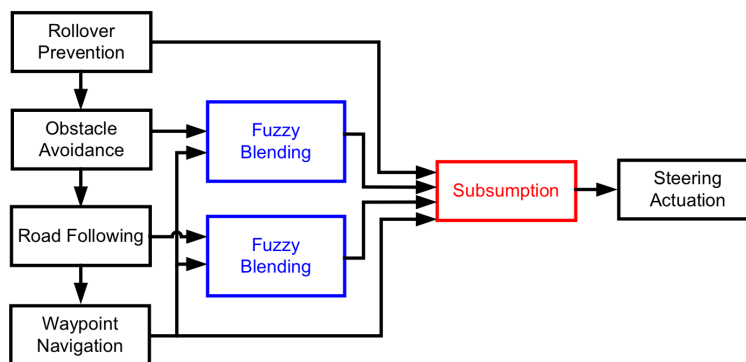


Figura 2.5: *Arquitectura reactiva Dynamic Expanding Zones*

Siguiendo la arquitectura de navegación reactiva el comportamiento del robot se ha dividido en cuatro tareas, utilizando distintos algoritmos para cada una. Por tanto, la navegación entre *waypoints* se lleva a cabo mediante campos potenciales (*Potential Fields*), el seguimiento del camino mediante seguimiento puro (*Pure Pursuit*) y la evasión de obstáculos mediante zonas de expansión dinámicas (*Dynamic Expanding Zones*). La cuarta tarea con-

siste en prevenir que el robot vuelque. En la figura 2.5 está representada la arquitectura del sistema reactivo *Dynamic Expanding Zones*. Se observa que para combinar las salidas de los distintos comportamientos se utiliza un control fuzzy *Hybrid Dominance-Fuzzy Control*.

En la competición DARPA Grand Challenge de 2005 los dos robots provistos con este sistema han conseguido los puestos 8º y 9º.

d) Arquitecturas de control para robots cooperativos

En⁷⁰ se describe el diseño, la implementación y evaluación de una arquitectura de control híbrida aplicada a equipos de robots cooperativos. El diseño emplea control reactivo, máquina de estados y planificación. El sistema presentado en esta tesis tiene como objetivo controlar un equipo de robots para la RoboCup de fútbol.

Para aprovechar las ventajas de los sistema reactivos y deliberativos el diseño presenta una arquitectura híbrida, representada en la figura 2.6. La capa deliberativa consiste en el planificador que genera la secuencia de acciones que activa los comportamientos reactivos necesarios en cada instante para cada uno de los robots. De esta forma se permite la coordinación compleja a nivel de equipo pero se reduce la complejidad del control reactivo para cada robot. El problema de planificar en un mundo dinámico no determinista (debido al equipo adverso) se resuelve por la habilidad de los comportamientos de reaccionar frente a los movimientos de los adversarios.

Los comportamientos atómicos utilizados para la capa reactiva son: errante, búsqueda, restringido, evitación, persecución, errante sin colisionar, espera el balón, defiende, ir a un punto X, ir detrás del balón, coger el balón, chutar a la posición X, chutar para liberar el robot, hacer espacio. En la figura 2.7 se puede observar la máquina de estados generada por el planificador para una jugada.

La evaluación del sistema se ha realizado mediante simulación. Se ha observado la importancia de la coordinación entre los diferentes robots mediante las maquinas de estados, puesto que al añadir más robots reactivos sin coordinación empeora el rendimiento.

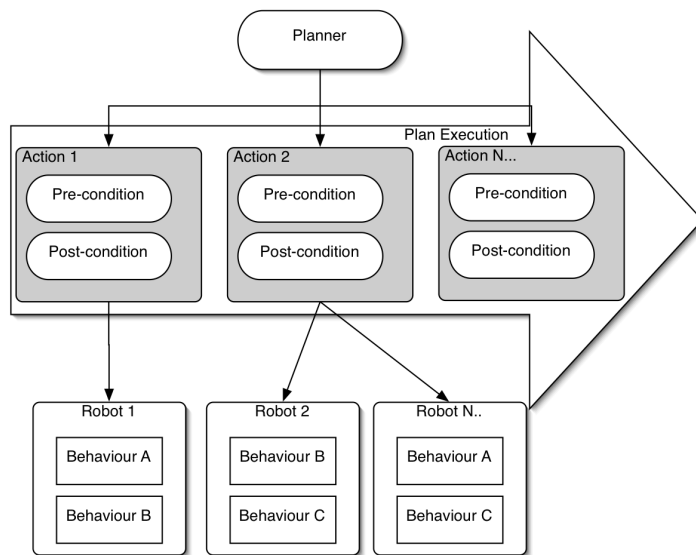


Figura 2.6: Arquitectura de control híbrida

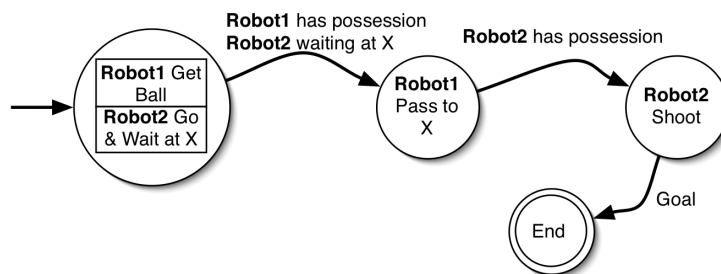


Figura 2.7: Máquina de estados generada por el planificador

e) Planificación de rutas

La planificación de rutas representa uno de los principales problemas de los robots móviles y se define como: “dado un robot móvil y la descripción de un entorno, planifica una ruta entre dos posiciones específicas que sea libre de obstáculos y cumpla todos los criterios de optimización”.

Con la aparición de sistemas de posicionamiento global GPS precisos se ha adoptado extensamente el sistema de navegación por waypoints. Un *waypoint* es una coordenada que identifica un punto de paso como longitud y latitud o bien longitud, latitud y altitud en el caso de los vehículos aéreos. Una forma de describir un entorno puede ser una red de

waypoints conectados entre sí por caminos libres de obstáculos y valorados con un cierto coste: distancia, tiempo, etc. Por planificación de rutas basada en *waypoints* (*Waypoint Based Path Planning*) se entiende seleccionar de esta red una serie de *waypoints* por los cuales tiene que pasar el robot para llegar al destino y que la ruta obtenida sea una solución al problema definido previamente.

El artículo⁶² presenta una comparación entre dos algoritmos de optimización para el problema de la planificación de rutas basada en waypoints: *Particle Swarm Optimization (PSO)* y el algoritmo genético *Artificial Immune System (GA-AIS)*.

El algoritmo GA-AIS es un algoritmo híbrido que combina GA (algoritmos genéticos) y AIS (sistema inmune artificial) de forma que AIS realiza una búsqueda local para ayudar al GA encontrar el camino factible de manera más eficiente. El algoritmo imita a los invasores y combina anticuerpos para obtener especificidad antigénica. Los cromosomas se dividen en dos partes: antígeno (representa la ruta factible) y anticuerpo (representa la ruta no factible).

La investigación realizada por los autores del artículo concluye que el algoritmo PSO es más rápido que GA-AIS para problemas pequeños, por lo cual es preferible en el caso de las aplicaciones en tiempo real que tienen recursos restringidos. Para problemas grandes se ha observado que GA-AIS funciona mejor.

Por cierto que en la ejecución de trayectorias tiene particular interés el empleo de curvas de Dubins^{15,27,33}.

Una de las maneras de simplificar y facilitar la ejecución de trayectorias complejas es descomponer estas trayectorias en primitivas de movimiento (*motion primitives*). El artículo³¹ ofrece una perspectiva actual del tema aplicado a aviones sin piloto.

Nuestro caso A nivel básico, por el momento, hemos aplicado descomposición en primitivas de movimiento basadas en curvas de Dubins para la ejecución de determinadas trayectorias experimentales.

f) Niveles de autonomía

A lo largo de estos años se viene discutiendo sobre el nivel o grado de autonomía que deben tener los robots. En bastantes casos es conveniente una cierta mezcla entre iniciativa del robot y decisión humana. Por ejemplo, cuando se manda un robot a Marte, el robot debe evitar obstáculos por sí mismo, pero en ciertos casos es conveniente que consulte a la base terrestre.

La investigación realizada sobre la interacción humano-robot se puede clasificar en cinco categorías:

- robots autónomos
- teleoperación
- autonomía ajustable
- iniciativas mixtas
- interfaces avanzadas

En³⁸ se realiza un estudio experimental acerca de la autonomía ajustable considerando los siguientes niveles: autonomía total, autonomía con discriminación de objetivo (*autonomous with goal biases*), métodos con waypoints, teleoperación inteligente y nivel latente.

Los autores afirman que al diseñar una arquitectura que permite a múltiples usuarios comunicarse con múltiples robots, es deseable que estos dispongan de suficiente autonomía para permitir a un único usuario manejar a todos los robots.

Para comparar los diferentes niveles de autonomía utilizan dos robots Nomad SuperScout y una aplicación software que permite la supervisión, el cambio del nivel de autonomía y la comunicación con el robot.

Autonomía total A este nivel, el robot actúa de forma autónoma y su misión es realizar un mapa de su entorno. Se observa que el robot es poco eficaz, aunque los mapas son útiles,

pero puede tolerar un alto nivel de desatención por parte del operador.

Autonomía con discriminación de objetivo En este nivel el usuario puede especificar regiones de interés o regiones de peligro que ofrecen un bias al comportamiento autónomo del robot. De esta forma el usuario puede guiar al robot a ciertos objetivos lo que implica una mayor eficacia que en el caso del robot completamente autónomo.

Waypoints y Heurísticas En este nivel el usurario puede marcar waypoints que actúan como potenciales atractores, obstáculos que actúan como potenciales repulsores y restricciones heurísticas del campo potencial. Así, la eficacia del robot aumenta mientras decrementa su autonomía y la tolerancia con respecto a la desatención por parte de un humano.

Teleoperación inteligente A este nivel, el robot está controlado por un humano mediante un joystick. A la teleoperación clásica de tipo maestro-esclavo se le añade un control compartido que consiste en que el robot decide si la orden recibida es segura o no. Por ejemplo si el usuario dirige al robot hacia un obstáculo, este intenta evitarlo, aunque si el usuario insiste obedece la orden. En este caso el sistema puede tolera solamente un mínimo de desatención por parte del usuario.

En la figura 2.8 se representa la eficacia del robot con respecto a su nivel de autonomía y el grado de desatención que presenta por parte del usuario. Dependiendo de la aplicación que se le quiere dar al robot puede interesar aplicar un nivel de autonomía en concreto o bien una mezcla.

En³⁷ los mismos autores demuestran mediante cuatro experimentos los beneficios de la autonomía adaptativa en cuanto a la eficacia y a la atención que necesitan por parte de un humano.

Nuestro caso En nuestro caso los robots son capaces de seguir en formación una ruta deseada de forma autónoma y sin intervención humana. La función principal de la base de

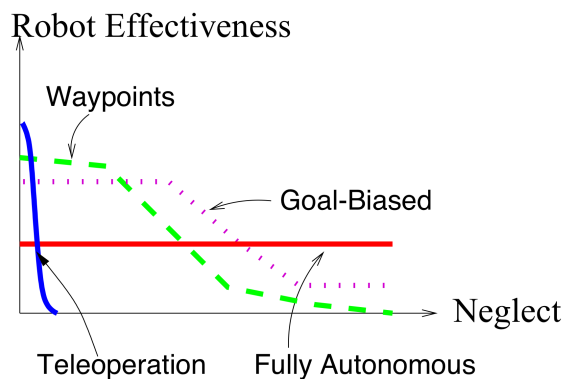


Figura 2.8: Eficacia de un robot en función del nivel de autonomía y el grado de desatención por parte del usuario ⁽³⁸⁾

tierra es permitir supervisar los robots, representando y guardando los datos de telemetría enviados por estos. También permite pasar a un control remoto en caso de emergencia. Además ofrece al usuario la posibilidad de cambiar distintos aspectos de la misión como son: la velocidad, la ruta o la geometría de la formación. Por tanto, la intervención del usuario se necesita solamente para configurar la misión.

g) Evitación de obstáculos

Desde los comienzos de la robótica, un tema fundamental ha sido la evitación de obstáculos. Se han propuesto diversas metodologías. En nuestro caso particular nos ha interesado la metodología reactiva basada en comportamientos y espacio nulo (*null-space* o NSB), porque hemos encontrado en la literatura que se ha propuesto recientemente para barcos. Esta metodología difiere de los demás métodos reactivos comentados previamente (competitivos y cooperativos) en la coordinación de los comportamientos: la forma de componer las salidas de los diferentes comportamientos para producir un comportamiento complejo.

La propuesta original acerca de *null-space* y la tarea de evitar obstáculos se hace en ⁵. Arrichiello junto con su equipo publican una serie de artículos en los que presentan distintas aplicaciones para las que se puede utilizar el NSB. En ³ se aplica a un robot móvil que golpea una pelota de forma que entre en la portería. En ¹⁰ y ¹² se aplica a una formación

de embarcaciones y se comprueba su funcionamiento en simulación. En⁴ se presenta un escenario de captura en el que una serie de robots rodean un objetivo que se encuentra en movimiento o bien estacionario. En¹¹ se aplica a un robot marino que realiza una navegación basada en waypoints. En¹³ presentan una misión de captura de un objeto flotante mediante dos robots cooperantes. En⁵⁰ se aplica el NSB a unos robots móviles vigilantes.

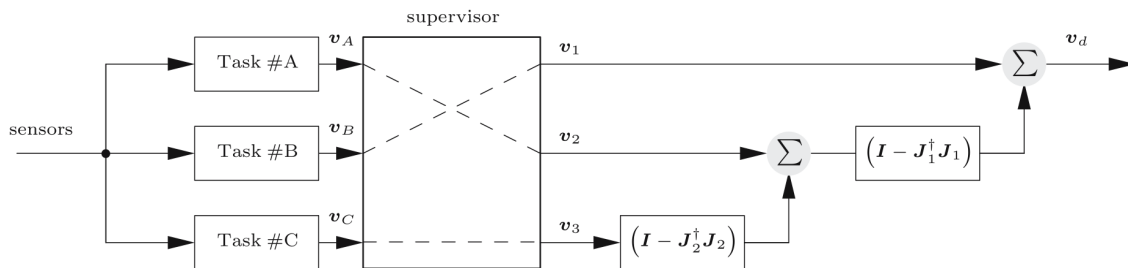


Figura 2.9: Esquema del control comportamental NSB ⁽⁵⁾

En la figura 2.9 se representa el esquema de control comportamental basado en espacio nulo. Se puede observar que a diferencia de los métodos competitivos la salida es una combinación de las salidas de cada comportamiento y a diferencia de los métodos cooperativos, las ganancias correspondientes a cada tarea no son fijas, si no que dependen del grado de cumplimiento de cada una. En concreto, la salida de cada tarea se proyecta en el espacio nulo de la tarea de mayor prioridad.

El control de comportamientos basado en espacio nulo (NSB) tiene la ventaja de asegurar el cumplimiento de la tarea de mayor prioridad sin verse afectada por la salida de las tareas de menor prioridad, igual que en el caso de los métodos competitivos. Pero además, explotando la proyección sobre el espacio nulo, puede cumplir simultáneamente dos o más tareas (dependiendo de la dimensión de su espacio) en lugar de solo una. La desventaja es que el NSB, debido a su carácter analítico, necesita la definición de una función para cada tarea que permita el cálculo de su Jacobiano.

En⁵ se presenta una comparación realizada de forma experimental entre las tres arquitecturas reactivas. Se observa que en los tres casos el robot consigue evitar el obstáculo, pero

solamente el control NSB es capaz de evitar completamente la zona de peligro. En la figura 2.10 se pueden observar las trazas del robot al esquivar un obstáculo y llegar al destino, utilizando cada uno de los tres métodos.

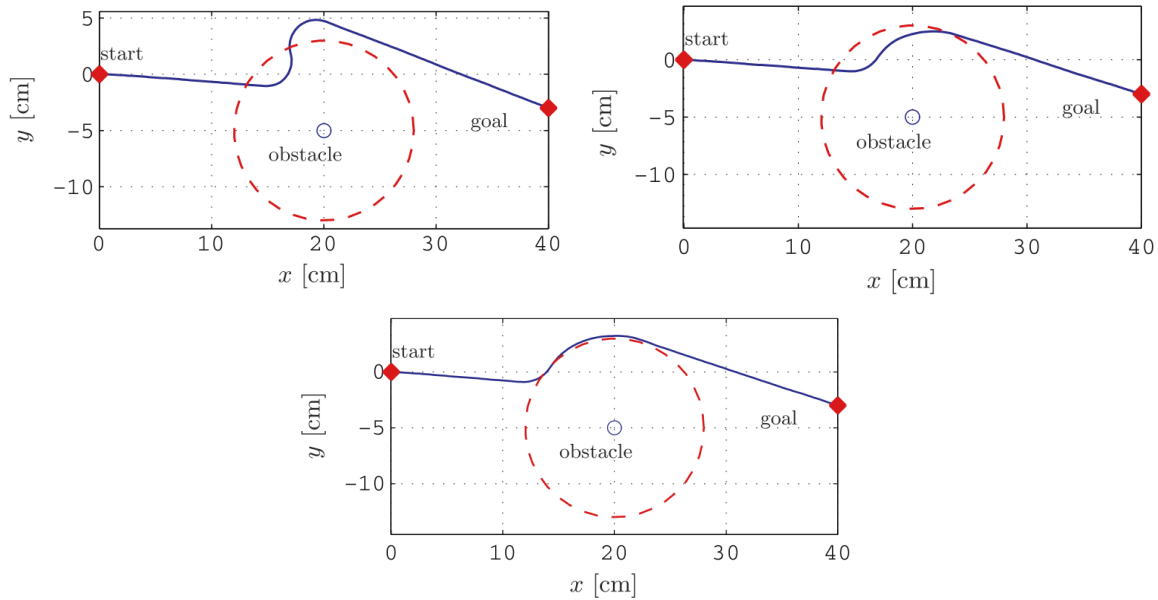


Figura 2.10: Ruta seguida para llegar al objetivo en presencia de un obstáculo utilizando los métodos: competitivo (izquierda), cooperativo (derecha), NSB (parte central-inferior) ⁽⁵⁾

Nuestro caso Para el control y combinación de los distintos comportamientos se ha elegido utilizar el control NSB porque presenta importantes ventajas frente a los métodos clásicos y se ha comprobado su funcionamiento en distintas aplicaciones tanto en simulación como experimentalmente.

2.3. Aspectos de la Investigación a Nivel de Agrupaciones de Robots Móviles

Una de las corrientes más influyentes en robótica se centra en la observación de lo que ocurre en la naturaleza. En ella observamos agrupaciones sin una geometría especial, como son los rebaños de ovejas. También podemos observar la aparición de geometrías como es en

el caso de formaciones de aves migratorias. De modo que podemos distinguir a grosso modo grupos o formaciones.

a) Agrupaciones de robots móviles

A nivel de agrupaciones de robots móviles, en⁶³ se trata sobre sistemas multi-robot. En⁵⁴ se ofrece una revisión de la investigación acerca de *swarms* y una revisión sobre formaciones se encuentra en²².

La robótica *swarm* es un nuevo enfoque sobre los sistemas de coordinación multi-robot que consisten en un gran número de robots relativamente simples inspirados en animales sociales: hormigas, abejas, peces, lobos e incluso personas. La característica más importante consiste en la habilidad de trabajar cooperativamente para alcanzar un objetivo común. Mediante la comunicación local limitada son capaces de obtener resultados comportamentales impresionantes: mantener la salud de la colonia, cuidando a los más jóvenes, respondiendo a una invasión, etc.

El problema de la generación de formaciones se define como la coordinación de un grupo de robots para alcanzar y mantener una formación con una cierta forma (círculo, línea, triángulo, etc). La coordinación se puede realizar de forma centralizada o distribuida. Se pueden adoptar distintas estrategias de control para mantener una formación:

- Basada en comportamientos
- Estructura virtual
- Líder-seguidor

Los principales problemas que deben considerarse a la hora de controlar una formación son: la estabilidad, la controlabilidad de distintos patrones de formación, la seguridad y las incertidumbres en la formación.

Los principales dominios de la investigación relacionada con las agrupaciones de robots son:

- La inspiración biológica
- La comunicación
- El control
- Mapas y localización
- Transporte de objetos y Manipulación
- Robótica reconfigurable
- Planificación de rutas
- Aprendizaje
- Asignación de tareas

A continuación se van a describir cada uno brevemente.

La comunicación en este tipo de sistemas puede ser *implícita/indirecta* o *explícita/directa*. La comunicación implícita en la naturaleza se lleva a cabo mediante feromonas. Así se ha introducido el concepto de “feromona virtual” empleado para realizar una comunicación y coordinación simple en áreas de vigilancia, reconocimiento, detección de peligros, búsqueda de camino. La comunicación explícita es el tipo de comunicación en la cual los robots intercambian mensajes entre ellos y/o el operador humano. Se ha comprobado que la comunicación entre los robots aumenta las capacidades y la eficiencia del grupo.

El control aplicado para estos grupos puede ser centralizado o distribuido.

- *Centralizado*: El sistema presenta un agente robot (o líder) que se encarga de organizar el trabajo de los demás robots. El líder se implica en proceso de decisión de todo el equipo, mientras los demás miembros actúan de acuerdo con las direcciones ofrecidas por éste.

- *Distribuido*: El sistema está formado por agentes robot completamente autónomos con respecto a los demás en el proceso de decisión. En este tipo de sistemas no existe un líder.

Los mapas son representaciones del entorno físico percibido a través de los sensores de los robots móviles. La localización consiste en encontrar la posición absoluta o relativa de un robot en el modelo espacial. Los problemas relacionados con estos conceptos son la localización y mapeo simultáneo (SLAM) o mapeo y localización concurrente (CML) y consisten en obtener un mapa de un entorno desconocido con un robot en movimiento, mientras simultáneamente se localiza el robot con respecto al mapa obtenido.

Sobre el transporte y manipulación de objetos se consideran tres tipos de métodos: agarrar, empujar y capturar. Estos métodos requieren distintos tipos de fuerzas y control.

Los robots auto-configurables son capaces de cambiar su forma deliberadamente, reordenando la conectividad de sus partes para adaptarse a nuevas circunstancias, llevar a cabo nuevas tareas o recuperarse tras un daño.

El problema de la planificación de rutas se ha definido en la sección 2.2. Los estudios sobre la planificación de rutas distinguen entre planificación de rutas local (dada por sensores instalados en el robot que ofrecen detalles sobre un entorno desconocido) y global (el modelo del entorno está definido con precisión y la navegación se realiza con la información conocida a priori). Se han introducido diferentes algoritmos para resolver este problema: lógica fuzzy, optimización partícula-swarm (PSO), optimización hormiga-colonia (ACO), etc.

El aprendizaje a nivel de grupos de robots se puede clasificar como aprendizaje supervisado y no supervisado. El aprendizaje supervisado requiere el uso de un supervisor externo y en la literatura se distinguen varios paradigmas: aprendizaje inductivo, basado en explicaciones, basado en casos, basado en memoria, etc. Los paradigmas del aprendizaje no supervisado son el aprendizaje evolutivo y por refuerzo. Estos métodos son útiles para permitir al robot adaptarse a situaciones donde la tarea o el entorno no es conocido de antemano o está cambiando constantemente.

El problema de la asignación de tareas consiste en asignar tareas a los robots del grupo de forma productiva y eficiente. Se tiene que asegurar que no solamente se cumple el objetivo si no que las tareas estén bien distribuidas entre los robots. El problema de la asignación de tareas multi-robot (MRTA) ha sido investigada utilizando varias técnicas como:

- El modelado físico.
- Planificación distribuida.
- Técnicas basadas en mercado.
- Técnicas basadas en subastas.
- ALLIANCE

b) Líder virtual

Una de las formas que se han propuesto para poder realizar movimientos de grupos como tales (navegación de grupos) es el empleo de un líder virtual, que en algunos casos puede estar localizado en un “centro de gravedad” de la agrupación.

Acerca de utilizar un líder virtual hay varias referencias que lo aplican tanto para embarcaciones como para vehículos submarinos [16,24,25,47,67](#).

El artículo [67](#) de Fossen, trata sobre el control de una formación de embarcaciones. El objetivo del control de cada barco es mantener su posición deseada en la formación mientras un punto de referencia virtual sigue una ruta predefinida. El comportamiento deseado de cada robot se divide en dos sub-problemas: 1) converger y seguir una ruta deseada y 2) satisfacer un comportamiento dinámico deseado a lo largo de la ruta.

En [24](#) se propone un el uso del método Líder-Seguidor para el control de una formación de robots móviles. Por tanto se tiene un robot líder que sigue la trayectoria mientras los demás robots, seguidores, intentan mantener una distancia y una orientación deseada con respecto a éste. A pesar de su simplicidad y escalabilidad este método depende fuertemente

de un único agente, lo que puede causar problemas en condiciones adversas. La convergencia de este sistema se demuestra matemáticamente.

En ¹⁶ y ²⁵ se utiliza también un método Líder-Seguidor para una formación de barcos y submarinos respectivamente. En estos casos la estructura de la formación se define relativa a un líder virtual. Mediante una simulación por ordenador se demuestra la funcionalidad de los sistemas.

En ⁴⁷ se presenta un sistema de control coordinado y distribuido para múltiples robots autónomos utilizando potenciales artificiales y líderes virtuales. Los potenciales artificiales definen las fuerzas de interacción entre robots vecinos y están diseñados para forzar un espaciado deseado entre los vehículos. Otros potenciales artificiales se utilizan para seguir a un líder virtual que se desplaza a lo largo de la ruta. Para comprobar la estabilidad del sistema en lazo cerrado se define una función de Lyapunov utilizando el sistema cinético de energías y la energía potencial artificial.

c) Evitación de obstáculos

Cuando se mueve un grupo de robots hay que tener en cuenta que desde la perspectiva de un robot los otros son obstáculos (móviles). Al mismo tiempo cabe considerar obstáculos como tales. El problema de la evitación de obstáculos se complica en la medida en que se quieran mantener ciertas características geométricas de la agrupación de robots.

Algunos trabajos realizados sobre grupos de carros o barcos que evitan obstáculos son: Arrichiello aplica *null-space* tanto para carros⁴ como para barcos^{10,13}; otros discípulos de Fossen³⁰ aportan trabajos en esta idea¹².

El artículo⁵⁹ propone crear polígonos libres de colisiones para un número N de pasos siguientes. La construcción del polígono la realizan mediante dos operaciones complejas logarítmico-lineales.

Previamente se ha mencionado el sistema de control de comportamientos basado en espacios nulos que permite definir una tarea de evitación de obstáculos. En¹² se presenta

un sistema centralizado NSB para el control de una formación en un entorno complejo realizando al mismo tiempo varias tareas: evitar obstáculos y mantener la formación. Con este sistema los obstáculos se esquivan a nivel individual, no de formación. Por tanto, durante la evitación del obstáculo la formación se intenta mantener mientras cada uno de los robots evita chocarse con los demás y con los obstáculos. Los resultados obtenidos en simulación se pueden ver en la figura 2.11.

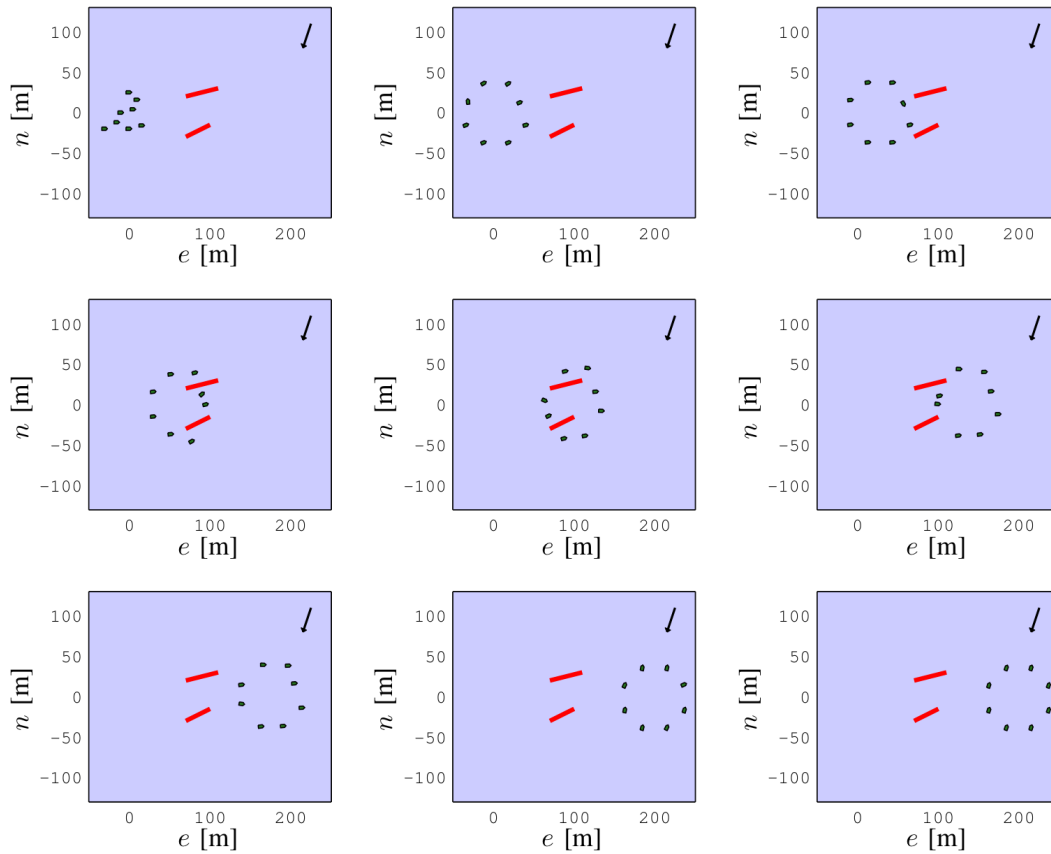


Figura 2.11: Nueve capturas de la ejecución de la misión. Una formación de 8 barcos se mueve a lo largo de una ruta rectilínea mientras mantiene una formación circular y evita dos obstáculos de forma lineal. La flecha indica la presencia de una fuerza en el entorno. ⁽¹⁰⁾

d) Modelado de formaciones

En²⁶ se propone el modelado de una formación de robots autónomos que se pueda controlar mediante simples leyes de control para mantener la geometría. Por tanto cada robot tiene que mantener una separación y un ángulo deseado con respecto a su líder y seguidor o bien con respecto a dos líderes, tal como se puede observar en la figura 2.12.

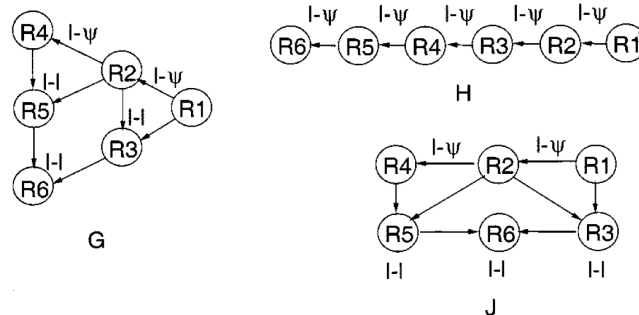


Figura 2.12: Ejemplos de formaciones (²⁶)

La estrategia de control utilizada por cada robot se describe mediante un grafo de control de la formación. Una transición de un de un grafo de control a otro se puede realizar mediante una matriz de transición. En la figura 2.13 se representa un cambio de transición y se muestra el resultado obtenido en simulación.

e) Cuestiones de comportamiento grupal

En⁵⁸ se trata sobre la inteligencia distribuida. Se definen tres paradigmas para este campo: el bioinspirado, el organizativo y social, y el paradigma basado en conocimiento. El desafío es determinar cual se adapta mejor a la aplicación, de forma que se obtenga coherencia global entre las interacciones locales de las entidades individuales. En la figura 2.14 se representan los cuatro tipos de interacción: colectiva, cooperativa, colaborativa y coordinativa que presentan los sistemas de inteligencia distribuida.

Colectiva Las entidades no son conscientes de las demás entidades del equipo pero comparten objetivos y sus acciones benefician a los demás. Por ejemplo los *swarms* donde

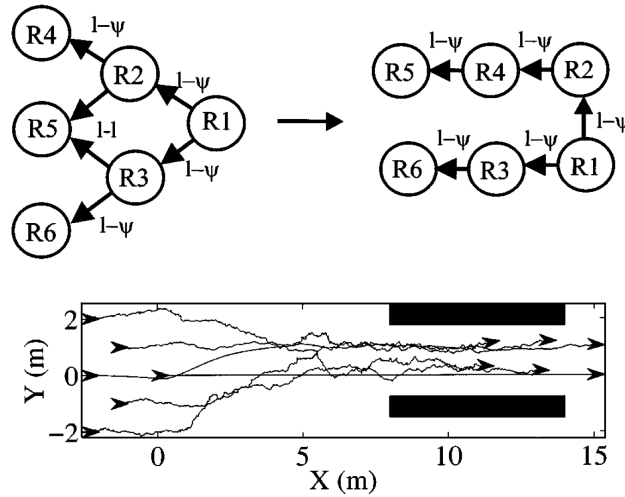


Figura 2.13: Ejemplo de transición ⁽²⁶⁾

los robots utilizan leyes de control local simple y un mayor número de robots consigue alcanzar el objetivo, normalmente como una propiedad emergente de las interacciones locales.

Cooperativa Las entidades son conscientes de las demás entidades, comparten objetivos y sus acciones benefician a los demás. Un ejemplo sería un grupo de robots que trabajan y razonan juntos para cumplir una tarea conjunta como empujar una caja, limpiar un terreno, buscar, rescatar, etc.

Colaborativa Los robots tienen objetivos individuales, son conscientes de los demás y sus acciones ayudan a avanzar los objetivos de estos. Un ejemplo podría ser un sistema en el cual cada robot tiene que alcanzar una posición en concreto.

Coordinativa Las entidades son conscientes de los demás pero no comparte un objetivo común y sus acciones no ayudan a los demás miembros del equipo. El caso se puede dar cuando los robots trabajan en el mismo entorno y coordinan sus acciones para minimizar las interferencias entre ellos mismos.

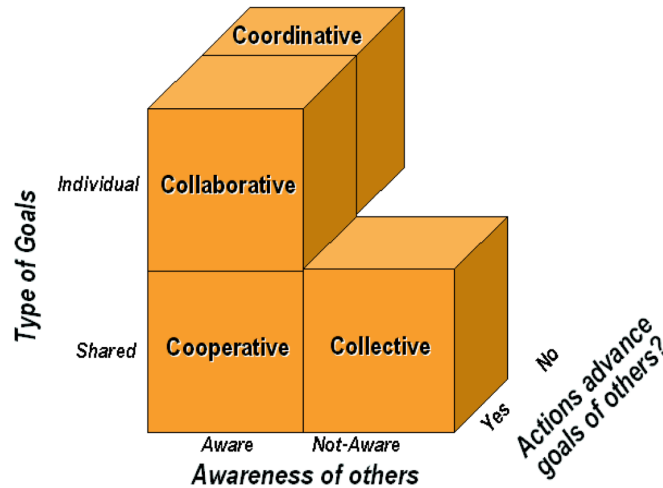


Figura 2.14: *Clasificación de los tipos de interacción* (58)

En el paradigma bioinspirado, caracterizado por interacciones colectivas, la necesidad de la comunicación entre los robots es muy reducida porque se asume la habilidad de cada uno de percibir el entorno. Por tanto, pueden ser idénticos y controlarse con simples leyes de control que conducen al comportamiento deseado del grupo.

El paradigma organizativo y social se basa en la teoría organizativa derivada de los sistemas humanos. En estos enfoques las interacciones agente/robot se diseñan modelando la dinámica individual y de grupo como parte de una organización. Se utiliza normalmente para los tipos de inteligencia distribuida cooperativa y colaborativa.

El paradigma basado en conocimiento utiliza el conocimiento compartido por los robots. A menudo este conocimiento se define como una ontología que especifica un vocabulario y semántica común al sistema de conocimiento. Este tipo de paradigma se puede utilizar para interacciones cooperativas, colaborativas y coordinativas.

En ⁵⁷ se propone una arquitectura software llamada ALLIANCE que facilita una cooperación tolerante a fallos de un equipo de robots móviles heterogéneos. Estos cooperan para realizar misiones compuestas por sub-tareas poco acopladas que pueden tener dependencia con respecto al orden de ejecución. ALLIANCE es completamente distribuido, basado en comportamientos y permite mediante motivaciones modeladas matemáticamente (impacien-

cia y aquiescencia) la elección adaptativa de acciones por parte de los robots.

Puesto que los robots son heterogéneos, no todos son capaces de realizar las mismas tareas, o bien no las realizan de la misma forma. La clave de este sistema es poder determinar qué tarea debe realizar cada robot de forma que se maximice la eficacia y la misión cumpla con el objetivo. Por tanto, cada robot determina de forma individual qué tarea realizar teniendo en cuenta los requisitos de la misión, la actividad de los demás robots, las condiciones del entorno y su propio estado interno. Puesto que mediante sensores es muy difícil percibir las acciones de los demás robots, cada robot informa a los demás mediante mensajes sobre la actividad que está realizando. La cooperación consiste por tanto en elegir las tareas a realizar o dejar de realizar una tarea si no se puede cumplir por uno mismo, permitiendo a otro continuar.

En ⁴² se trata sobre la cooperación entre robots marinos considerando varios escenarios de recogida de vertido. Se presentan cuestiones que implican la cooperación entre robots en estos escenarios.

En cuanto a la toma de decisiones se proponen las siguientes alternativas de organización:

- Un coordinador central: toma las decisiones principales, que se pueden basar en información compartida por el equipo.
- Cada individuo, en silencio: los individuos saben qué hacer y solamente miran al resto del equipo. La operación es planificada previamente.
- Individuos comunicativos: informan sobre sus decisiones.
- Estructura jerárquica: por ejemplo una flota de varias flotas

Durante la operación la organización puede cambiar. Esto implica la existencia de roles en el equipo: por ejemplo el coordinador puede cambiar.

En general, parte de la operación puede planificarse de antemano y la otra debe adaptarse a las circunstancias dinámicas.

En ³⁶ se consideran operaciones marinas complejas que involucran el movimiento coordinado entre varias embarcaciones semi-robot. Se propone un procedimiento genético para la planificación del control utilizando control “bang-bang” parcial.

En ⁶⁰ se propone un sistema de cooperación entre dos robots marinos para la recogida de vertido. Se utiliza un sistema de navegación basado en espacio nulo. Los diferentes comportamientos establecidos para la coordinación son:

- Mantener una distancia de seguridad.
- El centro de la formación debe seguir la ruta preestablecida.
- La línea que junta los dos robots debe ser perpendicular a la ruta que siguen.

El sistema se ha comprobado mediante simulaciones realistas y los resultados muestran la validez de la propuesta.

f) Subastas

Las subastas, un concepto inspirado en la economía de mercado, se propone para coordinar múltiples robots en la inicialización de formaciones y el reparto de tareas. En la literatura algunos ejemplos son ^{2,34,35,48,58}.

En ⁴⁸ se propone un método distribuido basado en subastas para la inicialización autónoma y la reorganización de formaciones. El método consiste en subastar cada posición de la formación y asignarla al robot ganador. La subasta de una celda la empieza el subastador que pide información a los demás sobre dicha posición. Si la posición no está ocupada y no hay una subasta abierta para esa posición se abre una subasta y se avisa a los demás robots. Estos empiezan a licitar si la posición es mejor que la que tienen asignada o bien no tienen una posición. La oferta consiste en la suma ponderada de la distancia a la posición y el número de vecinos actuales, ponderados por dos pesos E (*energy cost modifier*) y X (*relation cost modifier*). El ganador para esta posición será el que ha proporcionado la mejor oferta.

En ^{34,35} se propone el uso de las subastas para el reparto de tareas entre varios robots cooperativos. La negociación empieza con la introducción de una nueva tarea al sistema y sigue cinco pasos. El primer paso consiste en anunciar la tarea a los robots capaces de realizarla. El segundo paso consiste en la evaluación de dicha tarea por parte de los robots. En el tercer paso los robots presentan su evaluación calculada en el paso anterior. En el cuarto paso el subastador procesa las ofertas y determina al ganador que se encargará de realizar la tarea. En el quinto paso el subastador monitoriza el progreso del ganador hasta que la tarea se haya terminado. El sistema se ha validado mediante experimentación en un escenario en el cual dos robots empujan una caja hacia una posición deseada mientras que un tercer robot los vigila y dirige hacia el objetivo.

En ⁵⁸ se utilizan las subastas para el reparto de tareas entre robots móviles heterogéneos.

En ² se propone el uso de las subastas para extraer un número de robots de una formación minimizando la desorganización de la formación inicial.

g) Convoys

Sobre el caso especial del convoy de coches basado en cooperación²¹, está el proyecto SARTRE¹ que logró recorrer 125 millas en España, siendo el primero en probar una tecnología de este tipo en una carretera pública.

Para aumentar la seguridad y eficacia en el tráfico se ha identificado como posible solución el uso de sistemas de cooperación entre vehículos. Estos se basan en la comunicación entre vehículos y nodos de la infraestructura a través de una red inalámbrica ad-hoc.

En²⁰ se presentan distintas aplicaciones y un protocolo de comunicación en este tipo de sistemas. En la figura 2.15 se muestra una tabla en la que se describen estas aplicaciones, el tipo de comunicación, la frecuencia y los datos transmitidos.

El intercambio de información entre los coches reduce la necesidad de utilizar muchísimos sensores para la detección de los vehículos vecinos, reduciendo el coste del equipamiento necesario. Además, la información intercambiada entre los vehículos y entre el vehículo y

Table 5 High-priority cooperative vehicular safety applications as chosen by the National Highway Traffic Safety Administration and the Crash Avoidance Metrics Partnership [7]. Communication rates range from 1 Hz to 50 Hz with a maximum communication range span of 50–300 m. I2V represents infrastructure-to-vehicle, while V2V represents vehicle-to-vehicle.

Application	Comm. Type	Rate	Max. Latency	Data Transmitted	Range
Traffic signal violation	I2V	10 Hz	100 ms	Signal phase, timing, position, direction, road geometry	250 m
Curve speed warning	I2V	1 Hz	1000 ms	Curve location, curvature, slope, speed limit, surface	200 m
Emergency brake lights	V2V	10 Hz	100 ms	Position, heading, velocity, acceleration	200 m
Pre-crash sensing	V2V	50 Hz	20 ms	Vehicle type, position, heading, velocity, acceleration, yaw rate	50 m
Forward collision	V2V	10 Hz	100 ms	Vehicle type, position, heading, velocity, acceleration, yaw rate	150 m
Left turn assist	I2V or V2V	10 Hz	100 ms	Signal phase, timing, position, direction, road geometry	300 m
Lane-change warning	V2V	10 Hz	100 ms	Position, heading, velocity, acceleration, turn signal status	150 m
Stop sign assist	I2V or V2V	10 Hz	100 ms	Position, velocity, heading	300 m

Figura 2.15: *Aplicaciones de alta prioridad cooperativa para la seguridad vial* ⁽²⁰⁾

la infraestructura es más amplia y tiene más calidad que la medida o estimada a través de sensores. Otra ventaja consiste en que los sensores necesitan línea de vista para percibir el entorno, mientras que la comunicación radio alcanza grandes distancias y atraviesa obstáculos.

Las distintas aplicaciones tienen el objetivo de avisar al conductor de los posibles peligros en caso de una maniobra de frenado brusco por parte de un vehículo situado más adelante, en caso de una intersección en la que se podría producir una colisión o bien al cambiar de carril. Aunque estas aplicaciones solamente avisan al conductor de posibles colisiones, también es factible implementar el control en un vehículo autónomo.

En ⁶⁴ se definen los principales problemas de un sistema de conducción cooperativa. Estos consisten en la saturación de los canales de radio sin el uso de algoritmos de control y en encontrar un método distribuido adecuado. Además se debe asegurar la capacidad del vehículo de detectar y comunicarse con los vehículos vecinos relevantes. El artículo propone un sistema de control que adapte el tráfico de datos de forma que no se produzcan congestiones en la red.

En ²⁹ se presentan unos métodos para simplificar el diseño de los sistemas afectados por

la sobrecarga computacional y de comunicación .

2.4. Observaciones

Desde sus orígenes, la robótica móvil ha encontrado dificultades tecnológicas. Por ejemplo, la auto-localización: se ha empleado odometría (bastante tendente al error), también cámaras (pero eso implica trabajar en un entorno cerrado), empleo de balizas, etc. Desde hace algunos años se ha podido emplear GPS, más recientemente todavía este sistema se ha hecho más barato y compacto de forma que puede utilizarse en robots móviles pequeños. Buena parte de la literatura citada con base experimental no ha podido utilizar la reciente tecnología GPS. En nuestro caso esta es precisamente una de las ventajas que queremos aprovechar: el poder utilizar GPSs pequeños, con suficiente precisión y bajo precio.

Otros aspectos tecnológicos que son importantes para nuestra investigación es el de poder disponer de unidades de computación digital con suficiente potencia de cálculo y con pequeño tamaño; y también el empleo de brújulas digitales con resultados fiables y también con pequeño tamaño y bajo precio. Otro tanto cabe decir de los enlaces digitales vía radio para lo que se ha podido contar con recientes productos de bajo precio y pequeño tamaño.

En la mayor parte de los casos las publicaciones hacen uso de simulaciones y muy pocas se apoyan en resultados experimentales, que es donde nosotros hemos querido poner el acento.

Concretamente las publicaciones que tienen base experimental se van a comentar a continuación y se pondrá acento sobre su equipamiento hardware y el método utilizado para localización.

En⁶¹ utilizan dos carros robots preparados para DARPA Grand Challenge. El equipamiento del robot consiste en un receptor GPS, una unidad IMU, tres módulos LADAR y una cámara de visión estéreo. Además presenta tres módulos de procesamiento para el sistema de navegación, el sistema de visión y el sistema de control. Los robots recorren un trayecto de 130 millas sobre terreno en distintas condiciones y evita obstáculos.

En³ se utiliza un robot móvil Khepera II realizado por K-Team que se comunica a través

de un enlace radio con un ordenador. La plataforma experimental está instalada en un entorno controlado y dispone de un sistema de visión por computador para la localización del robot y de la pelota. En⁴ se utilizan varios robots móviles Khepera II y la misma plataforma experimental. Se realiza un escenario de captura en el que los robots tienen que rodear una pelota que se encuentra en movimiento.

En¹¹ se utiliza un robot marino diseñado por la University of Southern California. Mide $2,13m$ de largo, $0,71m$ de ancho y pesa $48kg$ con el equipamiento. Presenta un procesador dual-core Mini-ITX 2 GHz para los procesamientos principales y una placa Gumstix 400 MHz para operaciones auxiliares. Los sensores utilizados son: una IMU para la orientación, un receptor GPS para la localización, una DMU que se usa como giróscopo. Presenta también un sensor para la velocidad y dirección del viento, un sonar y un sensor de química del agua. El escenario experimental consiste en seguir una ruta marcada mediante *waypoints* y evitar obstáculos estáticos marcados de antemano en un estanque y con poco viento ($5m/s$).

En¹³ se utilizan dos robots marinos del tipo descrito anteriormente, en un escenario de recogida de un objeto flotante; la idea sería que dos barcos robot remolcaran una red flotante para capturar el objeto. El experimento se ha realizado con un objeto flotante virtual y sin red ninguna.

En⁵⁰ se utilizan tres robots móviles Pioneer-2Dx creado por ActivMedia Robotics que se emplean para un sistema de vigilancia en el pasillo de un edificio. Presenta 8 sonares dispuestos para cubrir 180° en la dirección de movimiento. La comunicación se realiza a través de ethernet, presenta una cámara y un telémetro láser. La localización se consigue mediante odometría (estimación en función de los movimientos de las ruedas) y mediante las lecturas del telémetro láser con un algoritmo de filtrado de partículas.

En⁵⁷ se utilizan tres robots móviles R-2 de IS Robotics en un escenario de limpieza de desechos peligrosos realizado en un laboratorio. El equipamiento sensorial incluye 8 sensores infra-rojos de proximidad que se utilizan para evitar colisiones, sensores piezoeléctricos para detectar colisiones y otros sensores de choque dentro de las pinzas para medir la fuerza de

sujeción. La localización de los robots se realiza mediante dos estaciones base sonar que se utilizan para obtener la posición del robot mediante triangulación.

En la mayoría de los artículos se realizan las comprobaciones mediante simulación.

Otras citas de interés son las siguientes:

- En ⁷⁰ se comprueba mediante simulación el comportamiento para robots móviles pensados para la RoboCup, una competencia de fútbol con robots autónomos.
- En ¹⁰ y ¹² se simula el comportamiento de formaciones de robots marinos que utilizan el control *Null-Space Based*.
- En ^{16,25,67} se simulan formaciones de barcos y submarinos que utilizan un líder virtual para la navegación.
- En ⁵⁹ se comprueba el método de los polígonos libres de colisiones.
- En ²⁶ se simulan formaciones de robots móviles y transiciones.

Capítulo 3

Plataforma experimental

La plataforma experimental utilizada en este Proyecto está desarrollada para permitir realizar las pruebas experimentales del sistema tanto en tierra como en agua. Por tanto se dispone por un lado de robots terrestres y por otro de robots marinos. Las dos configuraciones utilizan una misma estación de tierra para la supervisión y recogida de datos y el mismo software. La comunicación entre los robots y la estación de tierra se realiza a través de un enlace radio. Los ensayos experimentales se llevan a cabo en un entorno exterior (figura 3.1): una pista de atletismo (figura 3.2) situada en el campus Moncloa de la Universidad Complutense de Madrid y un estanque (figura 3.3) del parque Tierno Galván de Madrid.

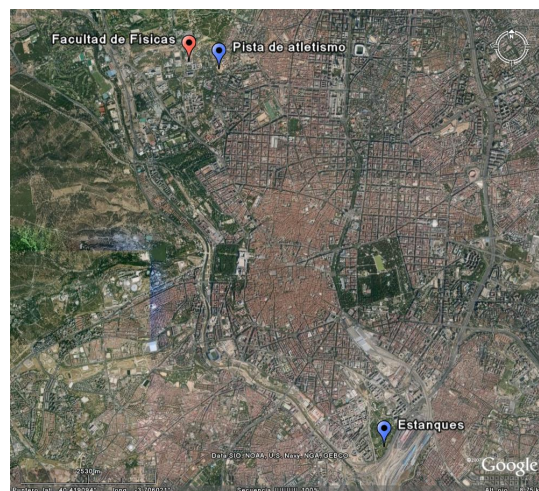


Figura 3.1: Localización de los entornos experimentales: facultad, pista y estanques.



Figura 3.2: *Plataforma experimental para los robots terrestres (foto y vista por satélite)*
Localización: latitud 40.449800°, longitud -3.719411°

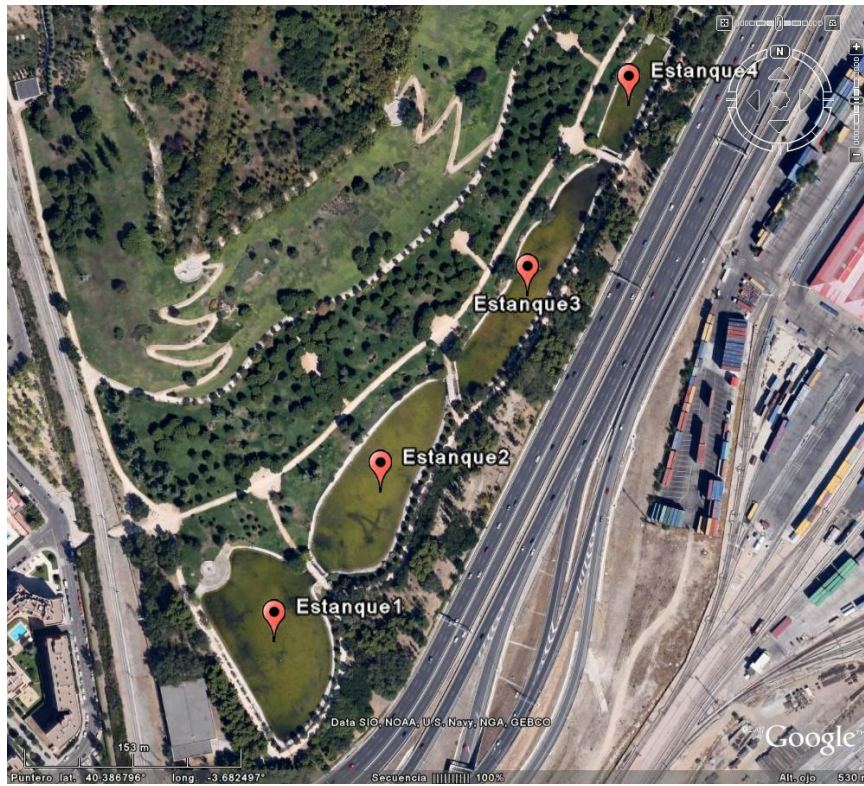
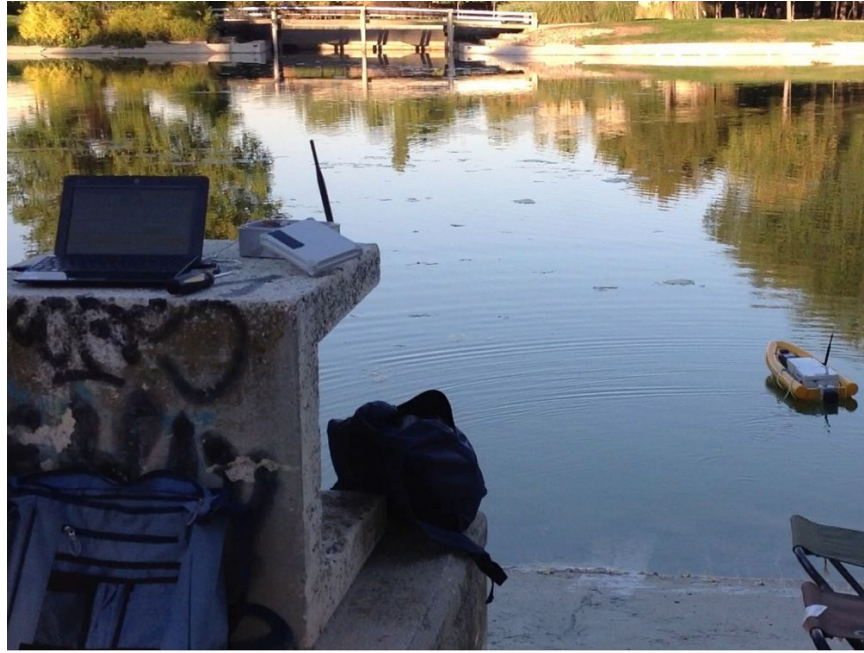


Figura 3.3: *Plataforma experimental para los robots marinos (foto y vista por satélite)*
Localización: latitud 40.386326°, longitud -3.682995°

3.1. Robots

En esta sección se van a describir los robots terrestres y marinos utilizados en este proyecto. Tanto los carros como los barcos se han equipado con las mismas placas de control, con la excepción de que los carros llevan la brújula montada fuera de la placa.

3.1.1. Robots terrestres

Los robots terrestres se han diseñado aprovechando el chasis y los actuadores de unos carros de radio-control Monster Truck Wheely King 4x4 de hpi-racing. Estos están realizados a escala de 1/12 y presentan los siguientes actuadores: un motor eléctrico Saturn 27T y un servo SF-10W. Los motores eléctricos necesitan el uso de un variador de velocidad, que recibe una señal PWM con la cual controla la velocidad rotatoria del motor. En los robots terrestres se utilizan dos tipos de variadores: el original de hpi-racing y un MC230CR de Futaba.

En la figura 3.4 se pueden ver la electromecánica del robot.



Figura 3.4: *Electromecánica de los robots terrestres*

En la figura 3.5 se muestra el robot con nuestra caja electrónica montada. En la figura 3.6 se distingue el interior de la caja, con la placa electrónica y los diferentes módulos.

Las medidas del robot terrestre se pueden consultar en la tabla 3.1.



Figura 3.5: *Robots terrestres*

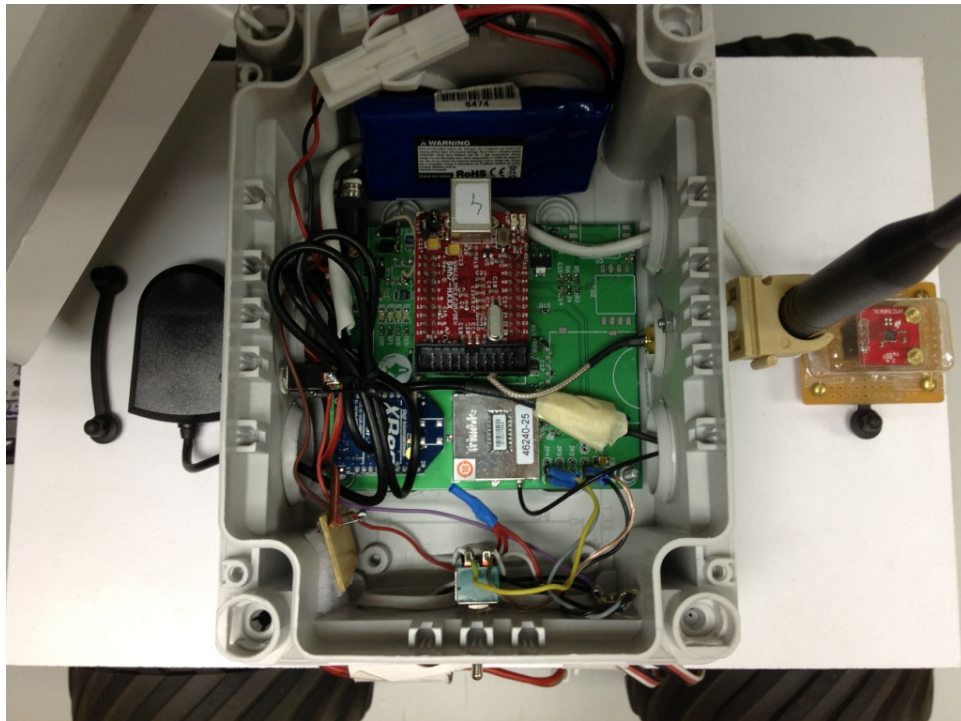


Figura 3.6: *Electrónica de los robots terrestres*

Medida	Dimensión
peso (con electrónica montada)	3.2 kg
longitud	431.8 mm
ancho	297.2 mm
altura (con electrónica montada)	280 mm
batalla (distancia entre los dos ejes)	238.8 mm
neumáticos	135 mm x 69 mm
recorrido suspensión	152.4 mm

Cuadro 3.1: *Medidas del robot terrestre*

3.1.2. Robots marinos

Las embarcaciones se han construido utilizando un casco Wiking de Kehrer Modellbau Berlin para barcos de radio-control, a una escala de 1/15. A nivel electromecánico disponen de un variador LRP IPC v3.1, un motor con hélice 700-Neodym fuera borda y un servo Toward Pro MG946R para girarlo.

En la figura 3.7 se muestra la electromecánica de los robots marinos.



Figura 3.7: *Vista de la electromecánica de los barcos*

En la figura 3.8 se muestran los robots con nuestra caja electrónica montada y el sistema de tracción.

En la figura 3.9 se distingue el interior de las cajas. A la izquierda se encuentra la placa electrónica y a la derecha el sistema mecánico.



Figura 3.8: *Robots marinos*

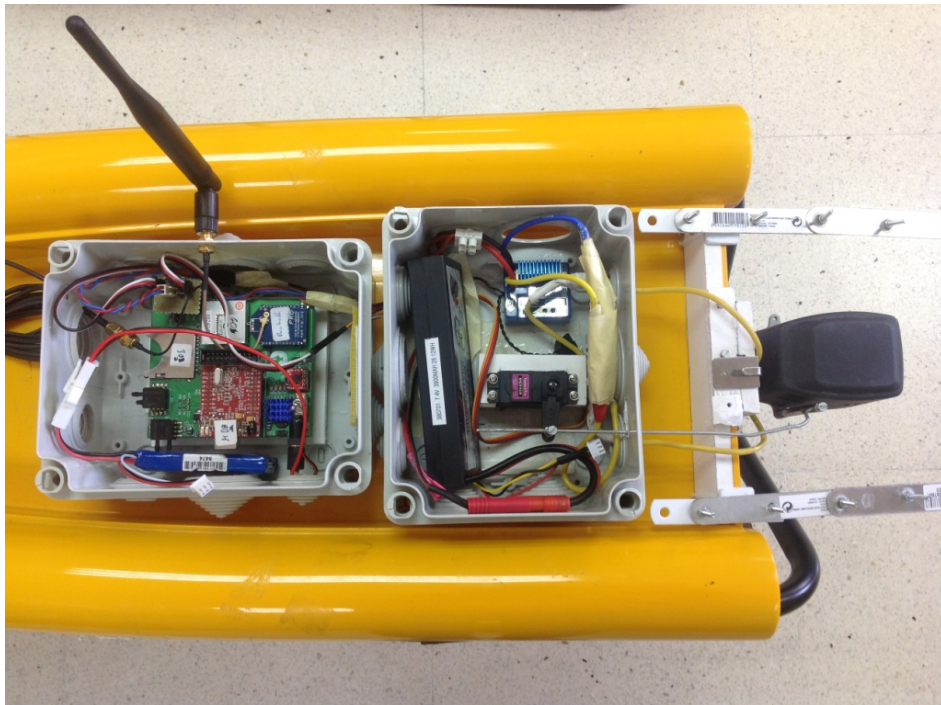


Figura 3.9: *Electrónica de los robots marinos*

Las medidas del robot marino se pueden consultar en la tabla 3.2.

Medida	Dimensión
peso (con electrónica montada)	3.9 kg
eslora	800 mm
manga	350 mm

Cuadro 3.2: *Medidas del robot marino*

3.1.3. Placa de control

Al inicio de mi Trabajo se disponía de una placa de a bordo diseñada y construida en nuestro laboratorio para el control de los robots. En la figura 3.10 se muestran dos copias de la electrónica y se pueden identificar los distintos elementos: el microcontrolador es la placa roja central, la brújula digital es la pequeña placa roja de la parte inferior, la radio es el módulo XBEE azul y el receptor GPS está conectado a la antena rectangular.

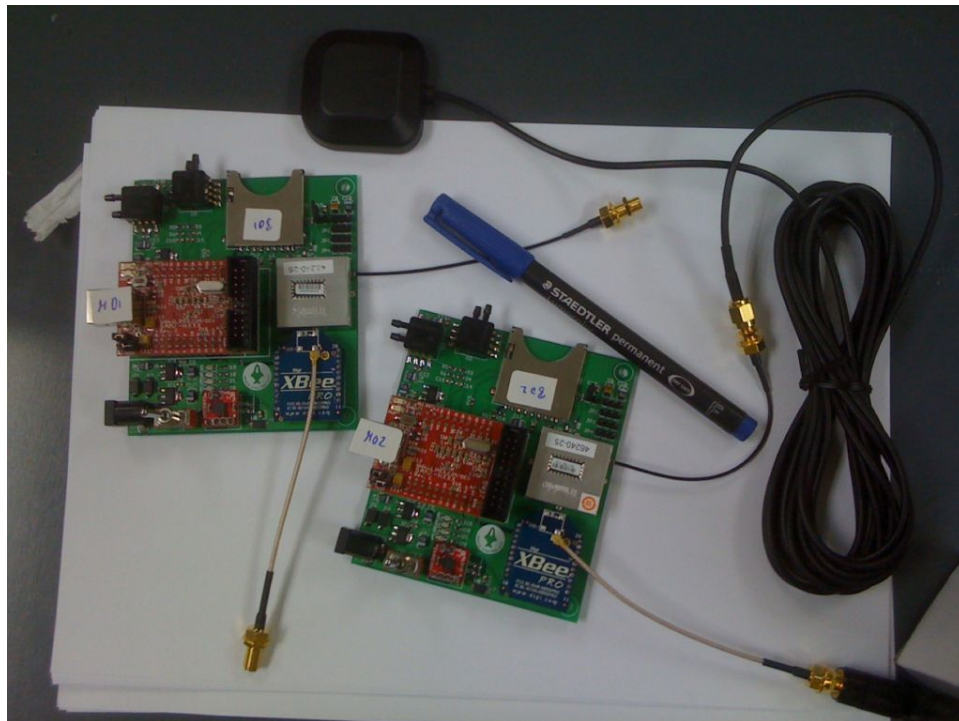


Figura 3.10: *Dos copias de la placa electrónica con sus distintos sensores*

La figura 3.11 muestra las distintas interfaces que se emplean para la comunicación entre el procesador y los distintos sensores y actuadores.

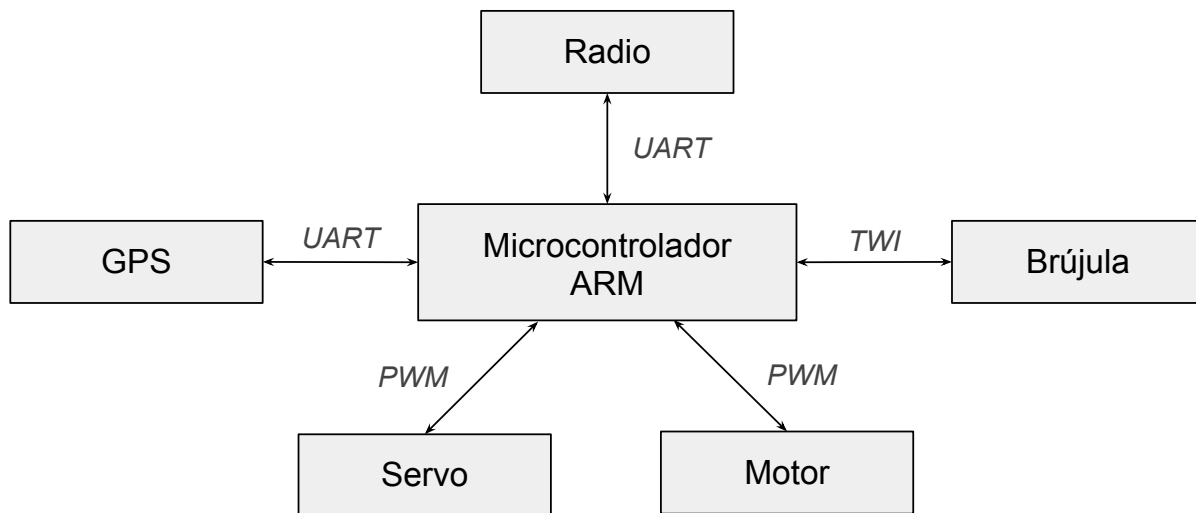


Figura 3.11: Interfaces de comunicación entre el microcontrolador y los sensores y actuadores

En el cuadro 3.3 se presentan los diferentes componentes de la placa, sus fabricantes y modelos.

Componente	Fabricante	Modelo
Microcontrolador	Atmel	AT91SAM7S256
GPS	Trimble	Lassen IQ DGPS
Brújula	Honeywell	HMC5843 y HMC5883L
Radio	Digi	XBee PRO S1

Cuadro 3.3: Componentes de la electrónica

A continuación se describen los elementos de la placa y la electromecánica con el objetivo de ofrecer una idea sobre las posibilidades y limitaciones que estos implican. Se proporcionan también enlaces web para poder consultar más detalles sobre cada módulo.

Microcontrolador El microcontrolador AT91SAM7S256 (figura 3.12) presenta un procesador ARM7 y es el encargado de ejecutar el software de a bordo y comunicarse con los distintos sensores y actuadores. Las características del microcontrolador son: 16/32 bit ARM7TDMI™, 256K Bytes de memoria flash, 64K Bytes de RAM, USB 2.0, RTT, 10 bit ADC 384 ksp/s, 2x UARTs, TWI (I2C), SPI, 3 temporizadores de 32bit, 4x PWM, SSC, WDT, PDC (DMA) para todos los periféricos, opera a una velocidad máxima de 60MHz. Viene montado en una tarjeta de desarrollo de Olimex que permite la programación y depuración a través de la interfaz JTag, aparte de un conector USB, un led de estado y un led indicador de alimentación, regulador de voltaje de 3.3V, botón de *RESET*, etc y tiene un tamaño de $50 \times 34mm$.

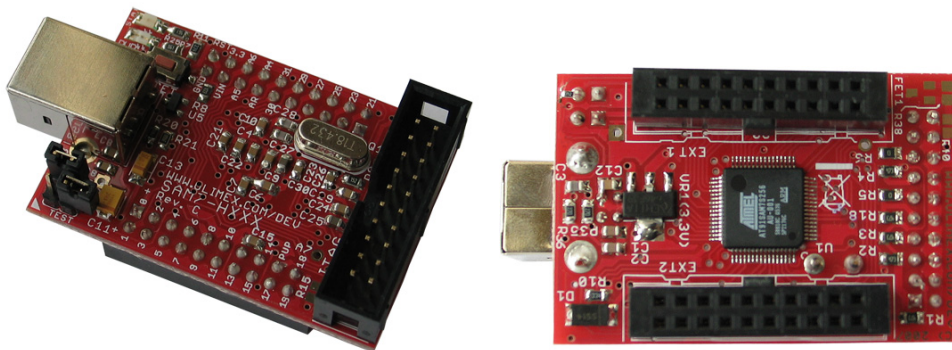


Figura 3.12: Microcontrolador AT91SAM7S256 (50 x 34 mm)

Se pueden consultar más detalles sobre este módulo en las siguientes páginas web: [Atmel](#) y [Olimex](#).

GPS El receptor GPS Lassen iQ de Trimble (figura 3.13) sirve para obtener la posición global y la velocidad del robot, ofreciendo medidas una vez por segundo. Se comunica con el procesador a través del puerto serie de forma asíncrona mediante *Direct Memory Access* (DMA) y utiliza el protocolo de comunicaciones propio del fabricante: *Trimble Standard Interface Protocol* (TSIP).

El funcionamiento de los receptores GPS se basa en calcular la posición global mediante *triangulación* a partir de las señales recibidas desde tres satélites. El problema consiste



Figura 3.13: *Receptor GPS Lassen iQ de Trimble (26 x 26 mm)*

en que las señales se ven afectadas al pasar por la ionosfera y troposfera, produciendo errores en el cálculo del posicionamiento. El módulo Lassen IQ de Trimble permite aplicar correcciones diferenciales y dispone de distintos filtros que se pueden utilizar para reducir los errores. En concreto, se pueden activar los siguientes filtros:

- Filtro Posición-Velocidad (PV): incrementa la precisión al suavizar las medidas tomadas cuando cambian los satélites de referencia. No afecta a la medida de la velocidad.
 - Filtro estático: es una característica del filtro PV que se aplica cuando el receptor se mueve muy despacio. Este filtro mejora la precisión en entornos urbanos.
 - Filtro de altitud: se recomienda su uso para aplicaciones marinas y en tierra.
- Filtro de Kalman: aplica un filtro de Kalman para la corrección de las medidas. Tiene en cuenta la dinámica que se puede configurar eligiendo entre las siguientes opciones: tierra, mar, aire, estacionario, automóvil.

En el cuadro 3.4 se describen los distintos paquetes TSIP que se reciben periódicamente desde el receptor GPS.

Para consultar más detalles sobre el módulo GPS Lassen Iq de Trimble se puede acceder al manual a través del siguiente enlace: [manual](#).

Paquete	Intervalo	Datos
Posición	1 s	latitud (rad), longitud (rad), altitud (m)
Velocidad	1 s	velocidad _{norte} (m/s), velocidad _{este} (m/s), velocidad _{arriba} (m/s)
Satélites	1 s	lista de satélites utilizados para el cálculo de la posición
Tiempo	5 s	tiempo GPS de la semana (s), número extendido de la semana
Estado	5 s	dos códigos de 1 byte que indican el estado del receptor: realizando correcciones de posición, todavía no se dispone de tiempo GPS, necesita inicialización, no hay satélites, hay sólo 1 satélite, hay solo 2 satélites, etc.

Cuadro 3.4: Paquetes TSIP recibidos periódicamente desde el receptor GPS

Brújula digital El módulo HMC5883L de Honeywell viene montado en una placa de Sparkfun (figura 3.14) que añade un interfaz TWI para la comunicación. El magnetómetro de 3 ejes sirve para medir las 3 componentes del campo magnético y obtener la orientación del robot. La comunicación con el procesador se realiza a través de la interfaz *Two Wire Interface* (TWI). La frecuencia de medición se puede configurar en función de las necesidades, siendo posible elegir distintas frecuencias entre 0.5 y 50 Hz. En nuestro caso, la brújula se consulta 10 veces por segundo.

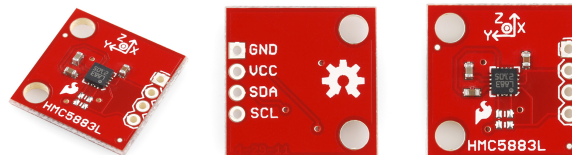


Figura 3.14: Magnetómetro HMC5883L de Honeywell (17.78 x 17.78 mm)

Después de montar una nueva brújula hace falta realizar una calibración para eliminar los ruidos creados por el motor y los diferentes componentes metálicos del robot. Un posible proceso de calibración consiste en mover el robot de forma que se obtengan medidas de todas las direcciones, por ejemplo realizando rutas circulares. A continuación se representan los datos obtenidos de los ejes x e y y se buscan las ganancias y los sesgos necesarios para cada eje de forma que al corregir los valores ofrecidos por la brújula, la circunferencia descrita por las medidas sea un círculo centrado en el punto

$(0,0)$. Una vez encontrados los bías correctos, después de cada lectura de la brújula se aplica la siguiente corrección:

$$eje_{corregido} = eje_{original} * ganancia - sesgo$$

La orientación del robot se calcula a partir de las componentes del campo magnético en los ejes x e y de la brújula, utilizando la función arcotangente, de forma que 0° representa el Norte, 90° el Este, 180° el Sur y -90° el Oeste.

$$\theta = atan2(eje_x, eje_y)$$

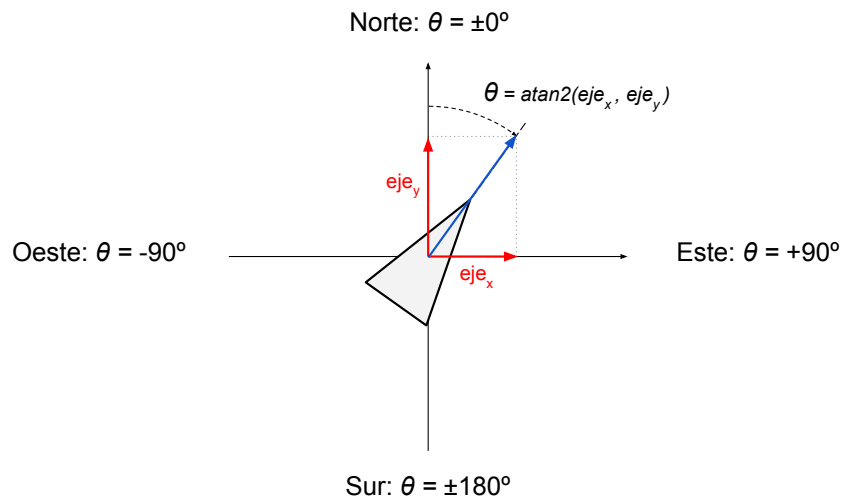


Figura 3.15: Interpretación de la brújula

La figura 3.16 muestra los datos de la brújula recogidos para realizar la calibración. Se representan los valores del campo magnético en los ejes x e y antes y después de aplicar la corrección. Se puede observar que la corrección consiste en fijar el sesgo del eje x en 45 y el sesgo del eje y en 515. La ganancia en los dos ejes es 1.

Se debe tener en cuenta que la brújula indica el polo norte magnético que no es el polo norte de la tierra, por lo que en realidad nuestro sistema vale donde la declinación magnética es cercana a 0, como en España. Para otros sitios del mundo habría que

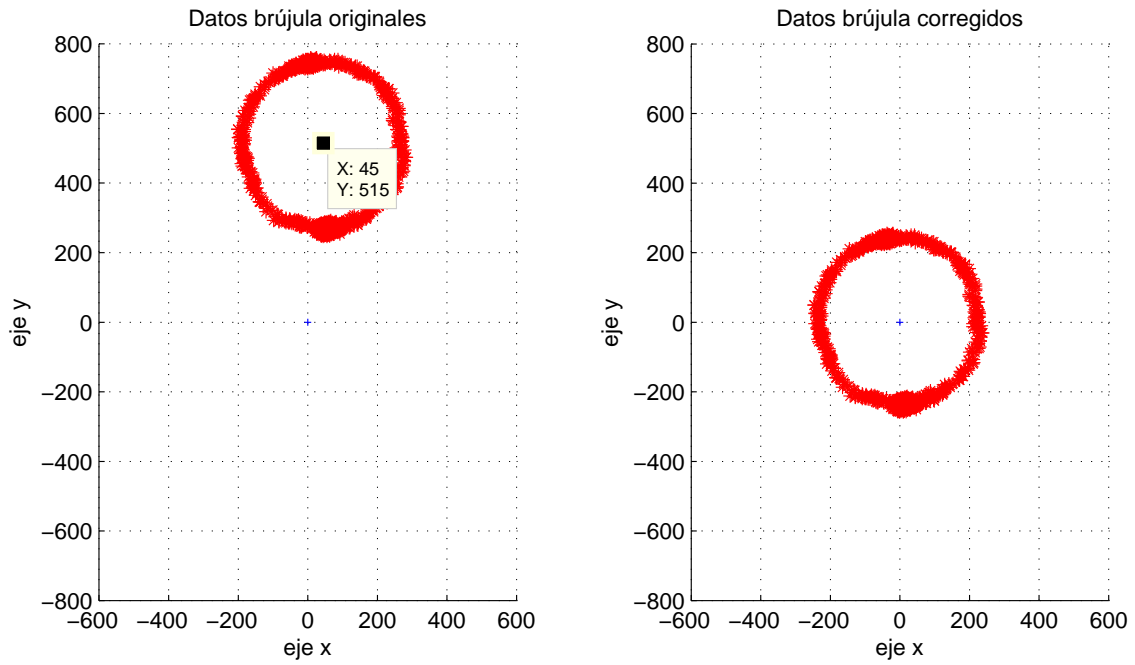


Figura 3.16: Datos brújula originales y corregidos

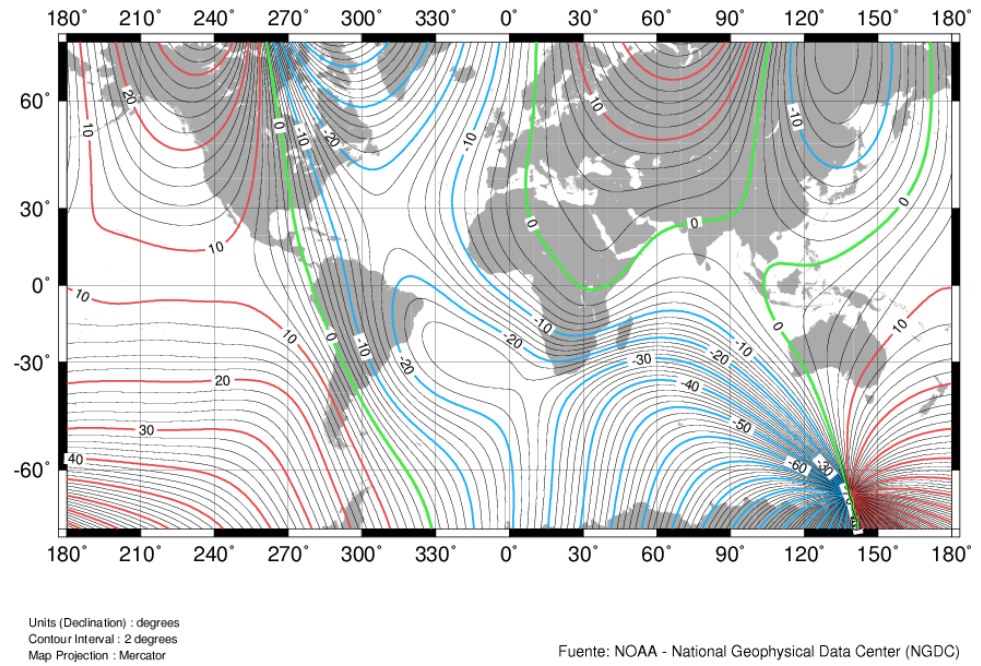


Figura 3.17: Grados de declinación para la corrección del norte magnético

revisar las fórmulas y añadir una corrección de la declinación magnética. En la figura 3.17 se pueden observar las declinaciones del campo magnético en la Tierra.

Las ventajas de este módulo son:

- Bajo consumo de potencia
- Precisión de orientación de 1° a 2°
- Interfaz TWI para comunicación a través del bus I²C

Se pueden consultar más detalles sobre este magnetómetro en las siguientes páginas web: [Honeywell](#) y [Sparkfun](#).

Radio El módulo de radio XBee PRO S1 de Digi, igual que el GPS, se comunica con el microprocesador a través de puerto serie, mediante DMA. En cada iteración del bucle principal de ejecución se comprueba si hay un paquete de datos recibidos. Cada paquete de datos se interpreta en función de un protocolo definido por nosotros.



Figura 3.18: *Módulo radio XBee-Pro S1 de Digi (32.94 x 24.38 mm)*

Algunas de las ventajas de los módulos XBee PRO S1 y los motivos por los cuales se han elegido son:

- Bajo coste
- Ultra-bajo consumo de potencia

- Uso de bandas de radio libres y sin necesidad de licencias
- Instalación barata y simple
- Redes flexibles y extensibles

Se pueden consultar más detalles sobre el módulo XBee PRO S1 en la siguiente página web: [enlace](#).

Servo y Motor Todos los actuadores de los robots (3.19) se controlan con señales *Pulse-Width Modulation* (PWM). Las señales tienen un periodo de 10 ms y el ancho del pulso está comprendido entre 0.636 ms y 1.273 ms. Para representarlo se utiliza un valor de 0 a 255 que se traduce a un ancho de pulso perteneciente a dicho intervalo en el caso de los servos. Para los motores se permite restringir el intervalo con el objetivo de eliminar la marcha atrás y las velocidades grandes no utilizadas.



Figura 3.19: Actuadores: servo y variador+motor

Los servos controlan el ángulo de las ruedas y del timón. El valor 0 representa el giro máximo a la derecha y el valor 255 el giro máximo a la izquierda.

En el caso del motor, el valor 0 detiene el motor y el valor 255 implica el máximo de revoluciones del motor (dentro de los límites establecidos). La señal PWM se envía al variador que procesa la señal y controla la velocidad de rotación del motor. Algunos variadores presentan distintos comportamientos:

- Una misma señal PWM recibida por dos variadores diferentes no se traduce en la misma velocidad de rotación del motor.
- Algunos variadores no permiten una subida brusca de velocidad si el motor está parado, esperando un aumento suave.
- Si se sobrecalientan, algunos variadores paran el motor hasta que se enfrían, mientras que otros limitan la velocidad durante el sobrecalentamiento.

Estos aspectos se han tenido en cuenta a la hora de configurar los intervalos de los anchos de pulso para señal PWM de los motores y el control de velocidad.

Alimentación Todos los robots son eléctricos. Para su funcionamiento necesitan dos baterías (figura 3.20): una para la parte electrónica (Zippy LiPo, 2 celdas 7.4V, 1000 mAh) y otra más potente para la electromecánica (Turnigy LiPo, 2 celdas 7.4V, 3800 mAh). Se puede realizar un experimento de más de dos horas sin necesidad de cambiar las baterías.



Figura 3.20: *Baterías de la electrónica y electromecánica*

3.1.4. Software de a bordo

El software de a bordo controla el robot de manera que este navegue de forma autónoma a lo largo de una trayectoria deseada y manteniendo una formación junto con los demás robots. El control se realiza en base al algoritmo de guiado y a las medidas ofrecidas por los distintos sensores. Además comunica datos sobre su estado actual a la estación de tierra y coopera con los demás elementos de la formación.

El bucle principal del programa de a bordo, sigue el flujo ilustrado en la figura 3.21. Los distintos eventos periódicos se sincronizan mediante uno de los relojes del procesador, que genera una interrupción cada 100 Hz.

- Con una frecuencia de 10 Hz se ejecuta el algoritmo de guiado que devuelve las consignas de rumbo y velocidad, se lee la brújula y se aplica la corrección de rumbo.
- Con una frecuencia de 5 Hz se envían a la estación de tierra los datos del estado actual que han cambiado con respecto a los últimos datos enviados.
- Con una frecuencia de 1 Hz se aplica la corrección de velocidad, puesto que el receptor GPS ofrece medidas una vez por segundo. Además se envían datos de cooperación a los demás robots y se comprueba el estado de la formación. En caso de haberse incluido o retirado elementos de la formación se procede al ajuste de la nueva formación, asignándose nuevas posiciones.
- Se comprueba constantemente si hay un nuevo paquete recibido del receptor GPS, actualizando la posición y la velocidad actual del robot. En nuestro caso se reciben medidas de este sensor una vez por segundo.
- Se comprueba constantemente si hay un nuevo paquete radio recibido. Si es un paquete de cooperación con otro robot, se actualiza la información correspondiente al mismo. Si es un paquete recibido de la estación de tierra, se analiza el comando recibido y se ejecuta la orden.

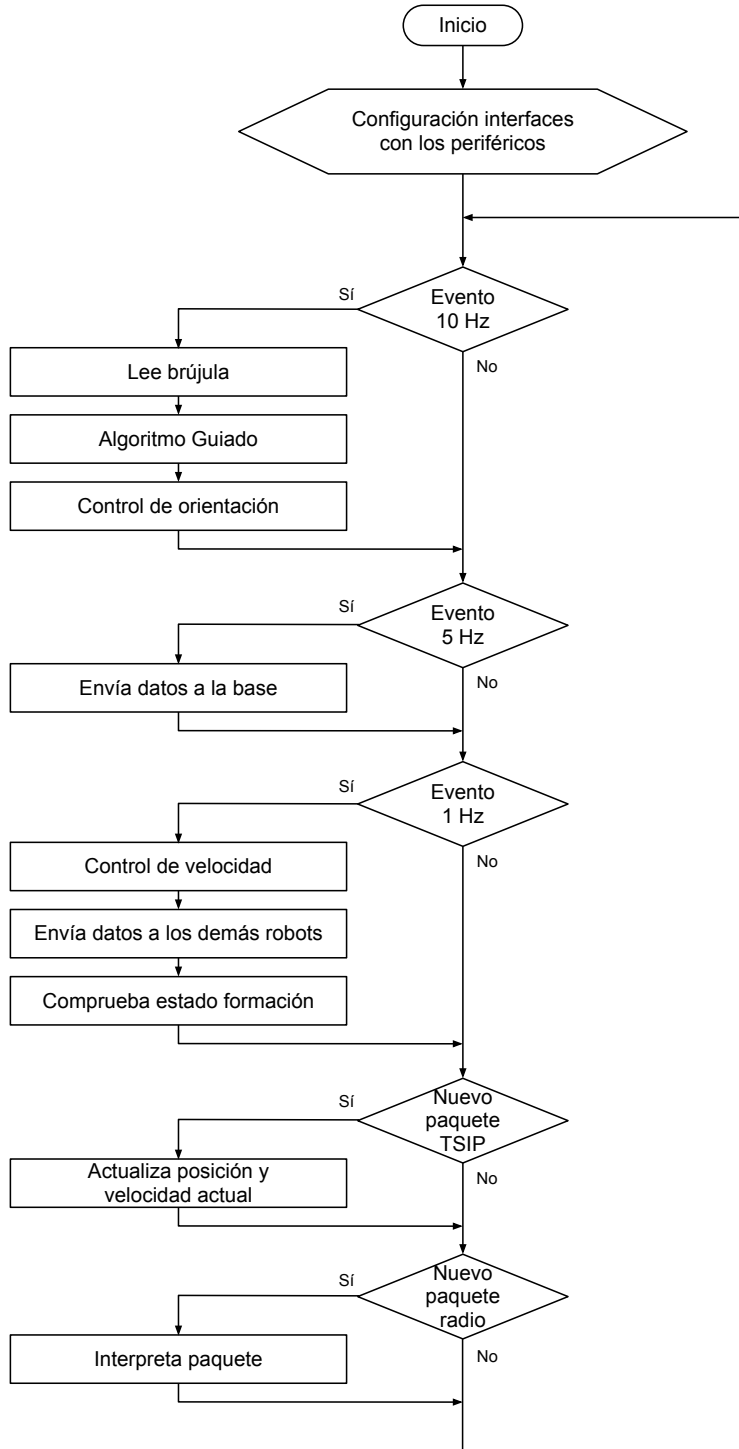


Figura 3.21: *Software de a bordo: bucle principal*

El programa está implementado en el lenguaje C99 y se ejecuta directamente en el microprocesador ARM7 de la placa electrónica, sin ningún sistema operativo que gestione los procesos. La compilación del programa se realiza mediante un compilador cruzado *GNU ARM* y la programación del microcontrolador se lleva a cabo a través de OpenOCD y la interfaz JTag.

Puesto que los robots no son idénticos al compilar el código se tiene que especificar para qué robot se va a utilizar el programa. Las diferencias consisten en la configuración por defecto de algunos parámetros que dependen del robot. Por ejemplo no son iguales las ganancias y los sesgos encontrados al calibrar las brújulas, hay distintos parámetros para la configuración de los variadores y las configuraciones de los sistemas de control de rumbo y velocidad son diferentes para los carros y los barcos.

3.2. Comunicación

La comunicación entre los robots y la estación de tierra se realiza a través de una red local Zigbee *peer-to-peer* creada mediante módulos XBee PRO S1. En la figura 3.22 se ilustra la topología y se puede ver que los componentes de la red pueden comunicarse con todos los demás, recibiendo y enviando información.

Zigbee es un protocolo de comunicaciones inalámbrico basado en el estándar de comunicaciones para redes inalámbricas *IEEE 802.15.4* realizadas en la banda libre de 2.4 GHz, también utilizada por los estándares de Bluetooth, Wi-Fi y algunas emisoras de radiocontrol. Según el protocolo, en esta banda se dispone de 16 canales para la comunicación, en nuestro caso se utiliza el quinto canal de frecuencias 2.4225 - 2.4275 GHz. El alcance normal con antena dipolo en línea vista es de aproximadamente 1.6 km y en interiores de unos 90 m. La velocidad de datos es de hasta 256 kbps y una red Zigbee la pueden formar teóricamente hasta 65535 dispositivos.

Los módulos XBee presentan unos *buffers* de recepción y envío de 100 bytes. Por tanto, los paquetes de datos no deben superar este tamaño. Para ello, en el caso de algunos datos

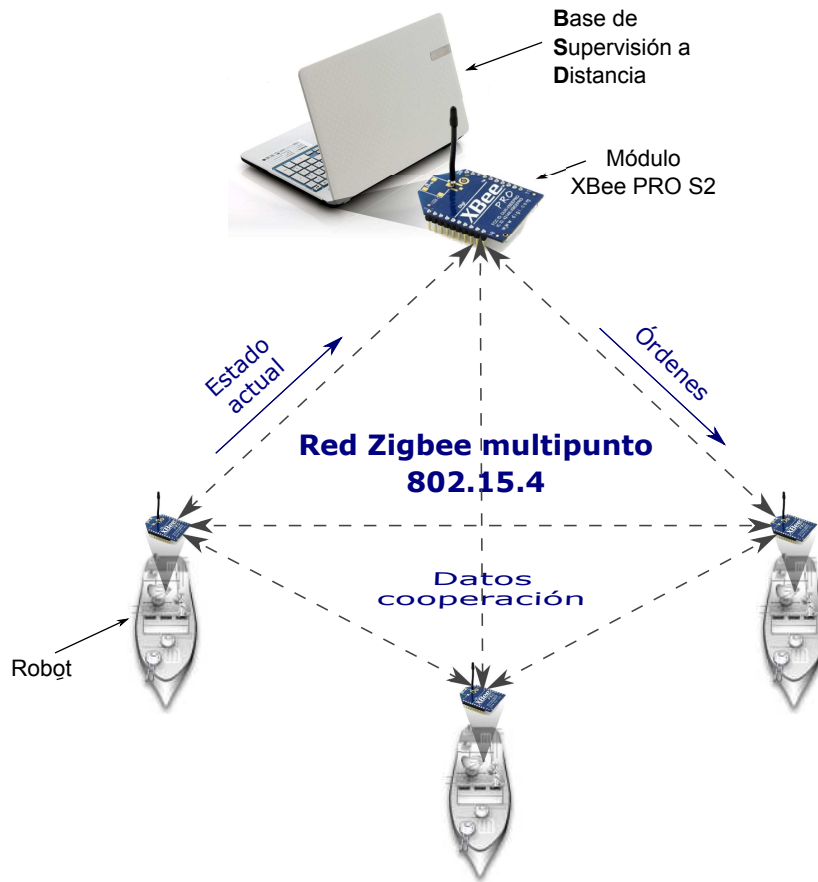


Figura 3.22: Red local Zigbee peer-to-peer

de punto flotante que no necesitan mucha precisión se ha decidido pasarlos a punto fijo con un precisión de 8 bits y un tamaño total de 2 bytes, reduciendo de esta forma a la mitad el tamaño de los valores en punto flotante.

3.2.1. Robot - Estación de tierra

La comunicación entre los robots y la estación de tierra implica un gran volumen de datos, necesarios para la supervisión y el análisis posterior. Los datos a enviar se agrupan en paquetes y se envían solamente si ha cambiado alguno de sus datos desde el último envío. Cada paquete presenta un identificador que le permite a la estación de tierra saber interpretar los demás bytes recibidos. A continuación se describen los distintos paquetes de datos:

- Brújula: contiene 3 valores de que representan las tres componentes del campo magnético medidas por el sensor.
- Actuadores: contiene 2 bytes que representan los valores del servo y del motor.
- GPS: contiene la latitud, longitud, altitud y velocidad.
- Estado: contiene información sobre qué subsistemas están activos, y los parámetros de control.
- Formación: contiene información sobre el subsistema de las formaciones.

3.2.2. Robot - Robot

La comunicación entre los distintos robots consiste en un paquete que contiene el identificador del emisor, su posición actual y un indicador de pertenencia a la formación.

3.2.3. Estación de tierra - Robot

La comunicación entre la estación de tierra y los robots presenta paquetes de datos más cortos, correspondientes a las distintas órdenes y configuraciones. Cada paquete contiene un identificador de la orden y los parámetros necesarios para cada una. Estas órdenes se van a describir en la siguiente sección, dedicada a la estación de tierra.

3.3. Estación de tierra

La estación de tierra se encarga de representar los datos enviados por los robots y por otro lado permite enviarles órdenes. Es una herramienta muy importante a la hora de realizar ensayos experimentales con los robots. Sus principales funcionalidades son:

- Mostrar los datos recibidos en formato textual y gráfico
- Guardar los datos en ficheros log
- Permitir activar, desactivar y configurar los subsistemas de los robots

- Configurar los escenarios de experimentación: elegir una ruta, una velocidad, el tipo de formación
- Controlar manualmente el robot mediante un joystick

3.3.1. Diseño

El software está escrito en C++ y utiliza las librerías Gtkmm y Cairo para la interfaz de usuario y para las representaciones gráficas. El programa presenta distintos hilos de ejecución en paralelo. Uno de ellos se encarga de monitorizar la radio, otro de leer los valores del Joystick y otro de la interfaz gráfica.

La aplicación actúa como un *Productor-Consumidor* donde los productores y los consumidores se ejecutan en distintos hilos. Los productores son los hilos encargados de monitorizar la radio y el joystick, mientras que el consumidor es el hilo encargado de representar los datos y enviar órdenes.

El patrón de diseño utilizado es el *Observer*, donde los productores son los *observables* y los consumidores *observers*. Conforme a este patrón de diseño, siempre que un objeto *observable* cambia, se notifica a todos sus *observers*. Este comportamiento se ajusta bien a nuestras necesidades, puesto que los visualizadores *observan* a los productores de información. De esta forma se desacoplan los productores de los consumidores y se evita la espera activa.

La figura 3.23 muestra el esqueleto del diseño de la estación de tierra. A continuación se describen las diferentes clases representadas:

Observable y Observer Son las clases que implementan el patrón Observer.

SerialPort Clase envoltura que permite manejar un puerto serie

XBeeManager Clase que permite interactuar con la radio de la estación de tierra. Proporciona una interfaz de alto nivel al driver de la radio.

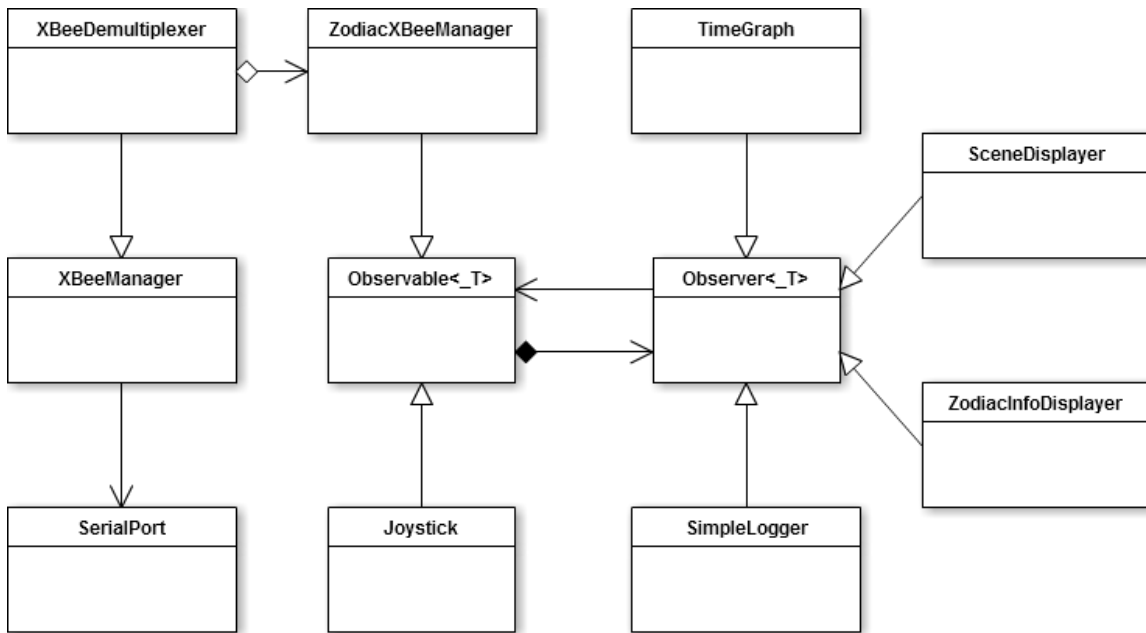


Figura 3.23: Diagrama de clases

XBeeManagerDemultiplexer Clase que permite generar distintos *Observables* para poder distinguir entre diferentes vehículos.

ZodiacXBeeManager Clase que permite decodificar los datos que envía el robot.

Joystick Productor de datos que lee de un joystick.

SimpleLogger Consumidor que guarda los datos que recibe en ficheros de texto.

TimeGraph Consumidor que muestra una gráfica con respecto al tiempo

ZodiacInfoDisplayer Esta clase muestra el estado de los distintos subsistemas del robot.

SceneDisplayer Esta clase representa gráficamente en 2D el escenario completo desde dos puntos de vista.

3.3.2. Interfaz de usuario

La interfaz de usuario presenta por cada robot una pestaña de configuración y una pestaña de control. Además se tiene una pestaña común para todos, en la que se puede

configurar un experimento y visualizar en tiempo real el escenario completo. A continuación se analizan cada una de las pestañas y se describen las distintas funcionalidades que tienen.

Pestaña de configuración: *Setup* # Se dispone de una pestaña de *Setup* para cada robot y está prevista para realizar la configuración previa al experimento.

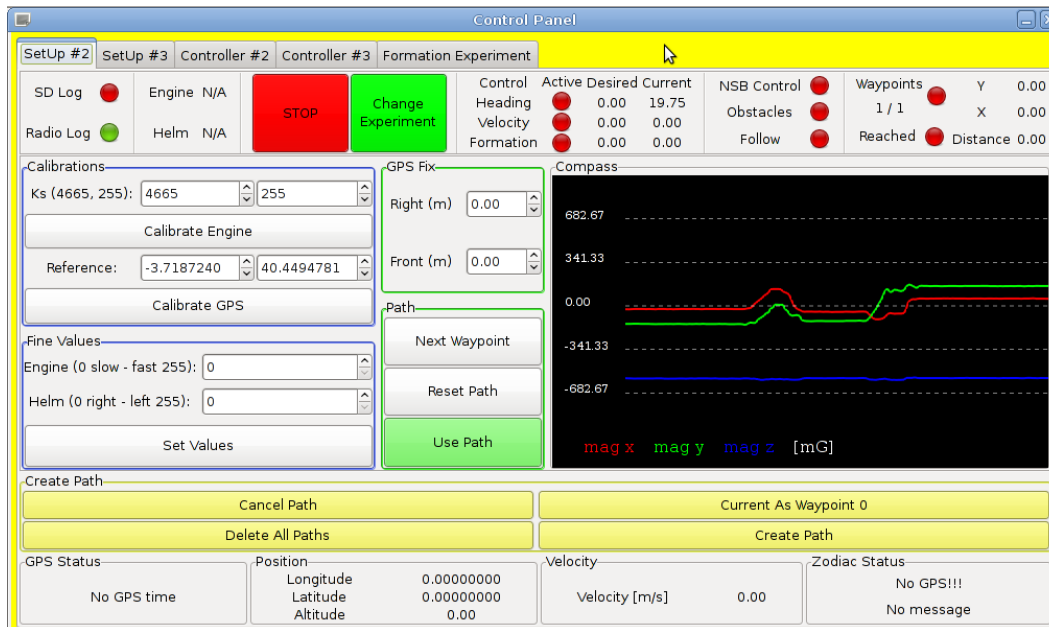


Figura 3.24: Pestaña *Setup*

En la sección *Calibrations* se permite limitar la señal de PWM enviada al variador. De esta forma se pueden eliminar la marcha atrás y las velocidades muy altas no utilizadas, obteniéndose un comportamiento más suave. Al mismo tiempo se realiza la calibración de los actuadores, recorriendo sucesivamente todos los valores de las señales PWM del servo y del motor.

La sección *Fine Values* permite fijar unos valores concretos para modular las señales PWM de los actuadores. Se utilizan para controlar de forma manual el robot, desactivando los subsistemas de control de rumbo y velocidad.

A la derecha, en la sección *Compass* se representan los 3 ejes del campo magnético y sirve para comprobar el funcionamiento y la calibración de la brújula.

En la parte superior e inferior se muestra la información relacionada con el estado actual del robot: configuración de los subsistemas de control, las consignas correspondientes y los datos de los sensores. Además hay un botón *STOP* de parada de emergencia y un botón *Change Experiment* que implica la creación de una nueva carpeta de ficheros de datos log, útil para el análisis posterior de la información guardada.

Pestaña de control: *Controller #* Se dispone de una pestaña de *Controller* para cada robot y está prevista para activar, desactivar y configurar los distintos subsistemas de control.

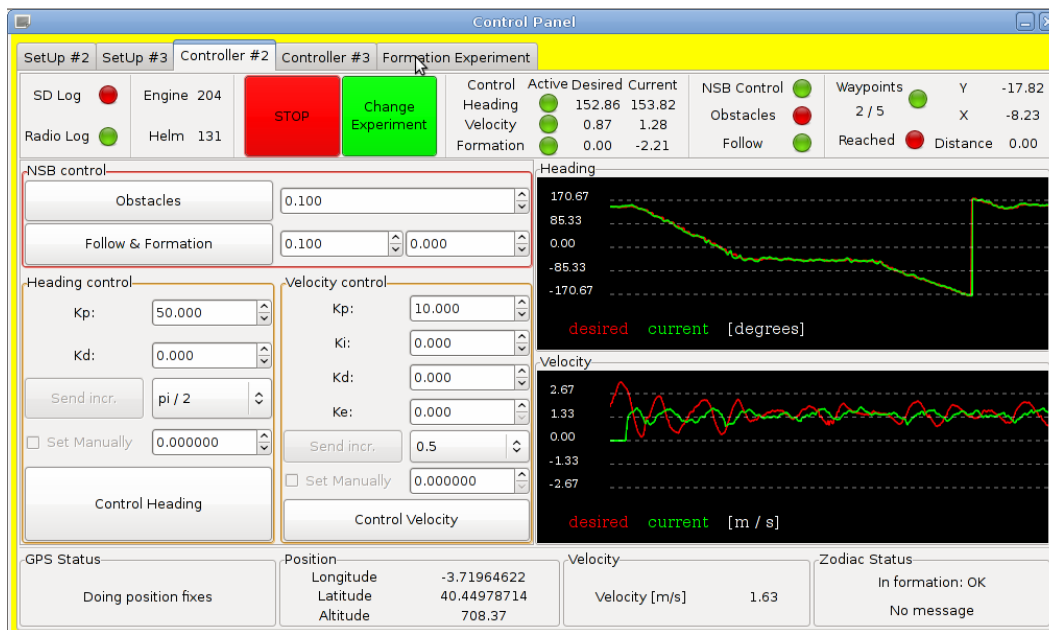


Figura 3.25: Pestaña *Controller*

La sección *NSB Control* permite activar o desactivar las tareas de evasión de obstáculos, de seguimiento de la ruta y de ajuste a la formación deseada. Además permite cambiar los valores de las ganancias correspondientes a cada tarea, con el fin de encontrar la configuración correcta del control.

La sección *Heading Control* permite activar o desactivar el control de rumbo. Además se le puede dar una consigna de orientación concreta y se permite cambiar las ganancias del

controlador PID.

La sección *Velocity Control* permite activar o desactivar el control de velocidad. Igual que en la sección del control de rumbo, se le puede dar una consigna de velocidad y se permite cambiar las ganancias del controlador PID.

Las gráficas *Heading* y *Velocity* representan en tiempo real las consignas y los valores actuales de rumbo y velocidad respectivamente. Estas sirven a la hora de configurar los subsistemas de control para comprobar el comportamiento del robot con respecto al deseado.

Igual que en la pestaña *Setup*, en la parte superior e inferior se muestra la información relacionada con el estado actual del robot, configuración de los subsistemas de control, las consignas correspondientes, los datos de los sensores y los botones *STOP* y *Change Experiment*.

Pestaña de configuración de experimento: *Formation Experiment* Esta pestaña es común para todos los robots y se utiliza para configurar los experimentos individuales o de formaciones.

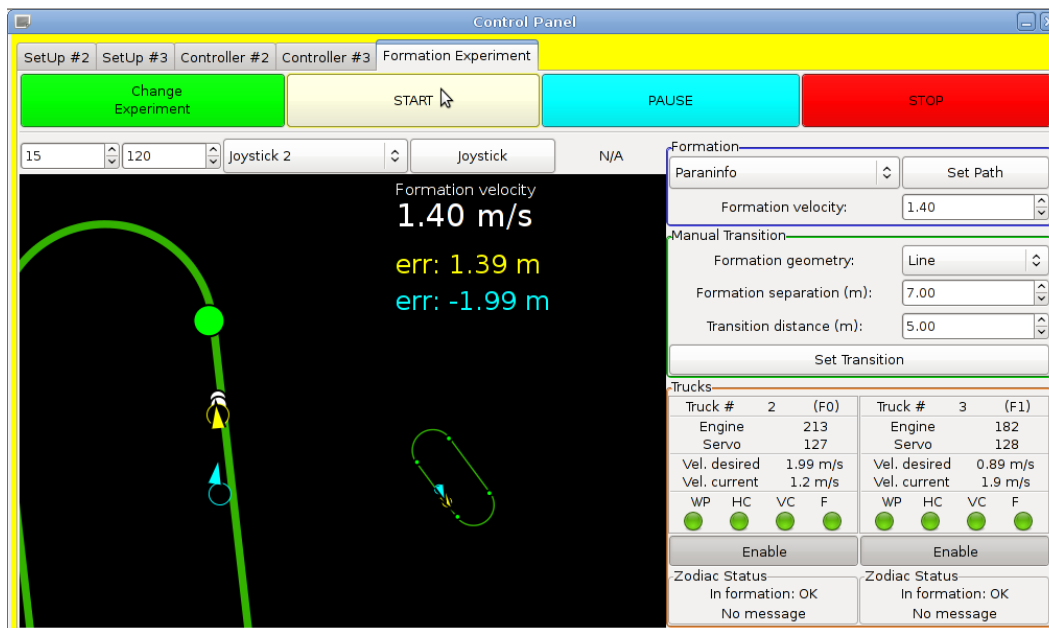


Figura 3.26: Pestaña *Formation Experiment*

En la parte superior presenta una sección de botones:

- *Change Experiment*: Igual que en las demás pestañas implica la creación de una nueva carpeta de ficheros de datos log.
- *START*: Arranca el experimento, activando los subsistemas de control de los robots elegidos para la formación.
- *STOP*: Termina un experimento, desactivando los subsistemas de control de los robots.

La sección de configuración del Joystick permite activarlo o desactivarlo, limitar los valores de PWM de motor enviados al robot y elegir qué robot se quiere controlar de forma manual.

En la parte derecha hay distintas secciones que permiten configurar:

- La ruta deseada y la velocidad de la formación.
- Un cambio de formación manual: geometría (línea, columna, triángulo), separación (metros), distancia en la que se tiene que realizar la transición de la antigua formación a la nueva.

También hay una sección por robot que permite visualizar su estado y presenta un botón para incluir o excluir el robot de la formación.

En la parte izquierda hay un gráfico bidimensional que permite visualizar en tiempo real el escenario completo. Así pues, se representan todos los robots con la orientación actual, la formación y la trayectoria deseada. El escenario se visualiza desde dos puntos de vista:

- Escenario completo visto desde arriba. Representa el escenario completo en el sistema de coordenadas ENU (Este-Norte-Arriba).
- Escenario parcial, representado en el sistema de coordenadas de la formación deseada. De esta forma las posiciones deseadas están siempre fijas, permitiendo enfocar la atención en el comportamiento de los robots.

3.4. Simulación

Durante el Trabajo se ha implementado un simulador geo-referenciado para poder validar y depurar los distintos algoritmos de forma rápida y sin riesgo de que se produzcan daños materiales. Además permite analizar a nivel básico el comportamiento de los robots en un entorno ideal, sin ruidos en las medidas de los sensores o causados por diversos obstáculos.

El simulador presenta dos principales funcionalidades:

- configurar un escenario experimental y realizar la simulación
- reproducir un experimento real a partir de los ficheros log guardados por la estación de tierra

3.4.1. Modelo Cinemático

La cinemática de un robot móvil se puede describir utilizando de forma combinada variables-robot y variables-mundo.

La figura 3.27 representa un robot, sus variables-robot $\langle x, y, \theta \rangle$ y sus variables-mundo $\langle X, Y, \varphi \rangle$. La velocidad del robot v sigue el ángulo θ marcado por el guiado (p.ej. en el caso del barco, el guiado se debe al timón).

Las ecuaciones del movimiento en variables-robot son las siguientes:

$$\dot{x} = v \cos \theta$$

$$\dot{y} = v \sin \theta$$

Mediante una proyección sobre los ejes-mundo se obtienen las ecuaciones del movimiento en variables-mundo. El ángulo φ (orientación del carro) se mide con respecto al Norte.

$$\dot{X} = \dot{x} \cdot \sin \varphi - \dot{y} \cdot \cos \varphi$$

$$\dot{Y} = \dot{x} \cdot \cos \varphi + \dot{y} \cdot \sin \varphi$$

$$\varphi = \text{atan2}(\dot{X}, \dot{Y})$$

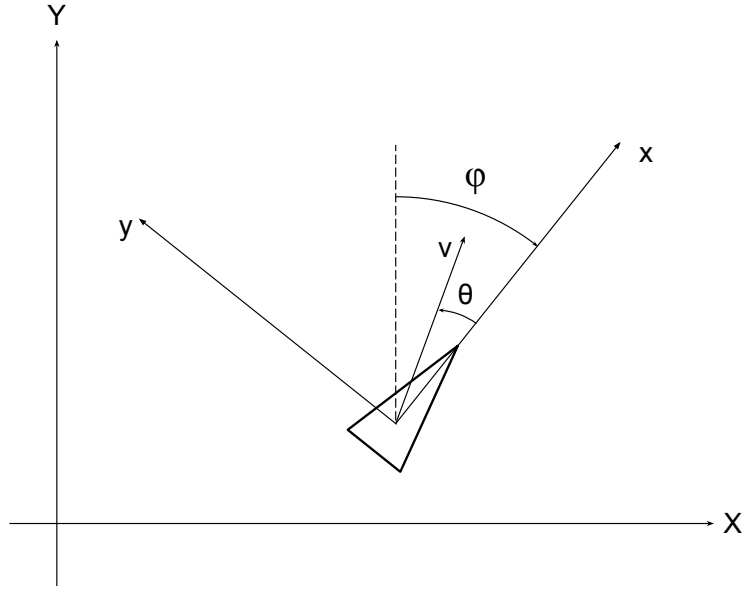


Figura 3.27: *Sistemas de referencia “mundo” y “robot”*

En la versión actual, la simulación no considera aspectos de la dinámica de los robots. Se incorporarán una vez que se disponga de suficientes datos experimentales (por el momento los sensores -en especial el GPS- no nos dan precisión y ritmo de muestreo suficientes para medidas directas, de modo que pensamos deducir los aspectos dinámicos a partir de resultados en lazo cerrado). Hay que considerar además un factor que complica la situación, y es que el ataque a los motores de los robots se efectúa mediante variadores (componentes estándar de radio control). Estos variadores impiden de forma automática cambios bruscos en el PWM enviado al motor, transformando entradas en escalón en entradas en rampa. De modo que la dinámica del robot no es puramente mecánica, sino que engloba el comportamiento interno de los variadores.

3.4.2. Diseño

El simulador está escrito en C99 para permitir una mayor similitud con el código del software de a bordo. Para la representación gráfica en 2D y 3D se utiliza la librería OpenGL.

En la figura 3.28 se pueden identificar los distintos módulos que presenta el software y a

continuación se va a describir la interacción y la funcionalidad de cada uno.

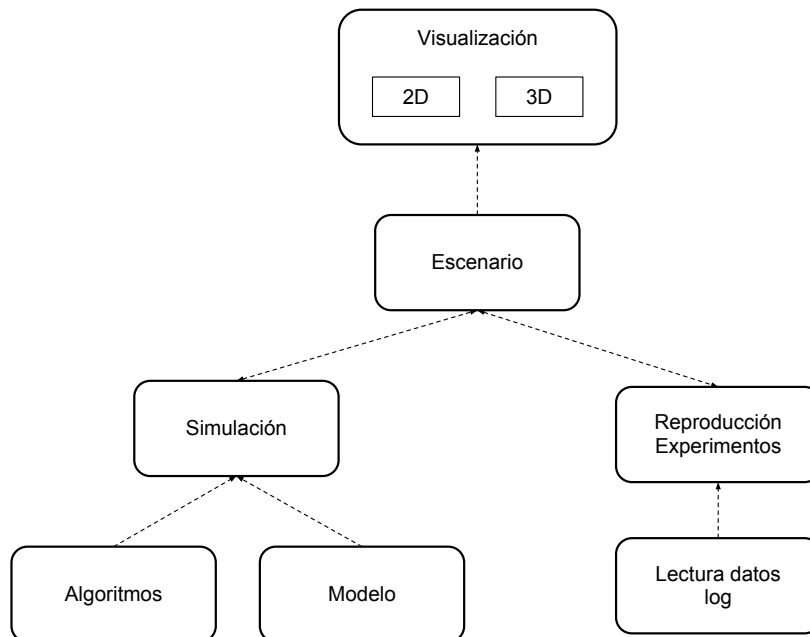


Figura 3.28: *Módulos del simulador*

Visualización El módulo de visualización se encarga de representar gráficamente el interfaz de usuario y el escenario, tanto en 2D como en 3D.

Escenario Este módulo guarda la información del escenario en cada instante de tiempo. Consiste en la información relacionada con cada robot y con la ruta que estos tienen que realizar. Esta información se actualiza o bien por el módulo *Simulación* o *Reproducción de experimento*.

Simulación Módulo que controla la simulación de un experimento. Tiene en cuenta la frecuencia a la que se leen los sensores y se ejecutan los algoritmos, así como el modelo de los vehículos simulados.

Algoritmos Este módulo engloba los distintos algoritmos utilizados por los robots en el software de a bordo.

Modelo Actualiza el estado de un vehículo cada instante de tiempo, en función de un modelo cinemático.

Reproducción Experimentos Este módulo se encarga de consumir de forma síncrona los datos leídos desde los ficheros log.

Lectura datos log Módulo que lee los siguientes datos log, una vez que los anteriores han sido consumidos por el módulo *Reproducción Experimentos*.

3.4.3. Interfaz Gráfica

La interfaz gráfica presenta 4 vistas diferentes:

- *Simulation* - escenarios simulados
- *Experiment* - reproducción de experimentos
- *Graphics* - representación de señales
- *3D View* - vista 3D

Simulation y Experiment Permiten visualizar el escenario completo mediante una representación en 2D sobre una malla de $5m \times 5m$ o bien sobre una imagen de satélite de *Google Maps*. En la parte superior se presentan la velocidad y el error de formación actual. En la parte izquierda se muestran las distintas opciones disponibles mediante el teclado y el ratón. La vista *Experiment* se activa al reproducir un experimento a partir de los ficheros log, mientras que la vista *Simulation* queda desactivada.

Graphics Esta vista presenta aparte del escenario completo una gráfica con las evoluciones de las velocidades y los errores de formación. En la parte izquierda se muestran las distintas opciones disponibles mediante el teclado y el ratón.

3D View Permite visualizar el escenario en 3D eligiendo una de las dos *cámaras*:

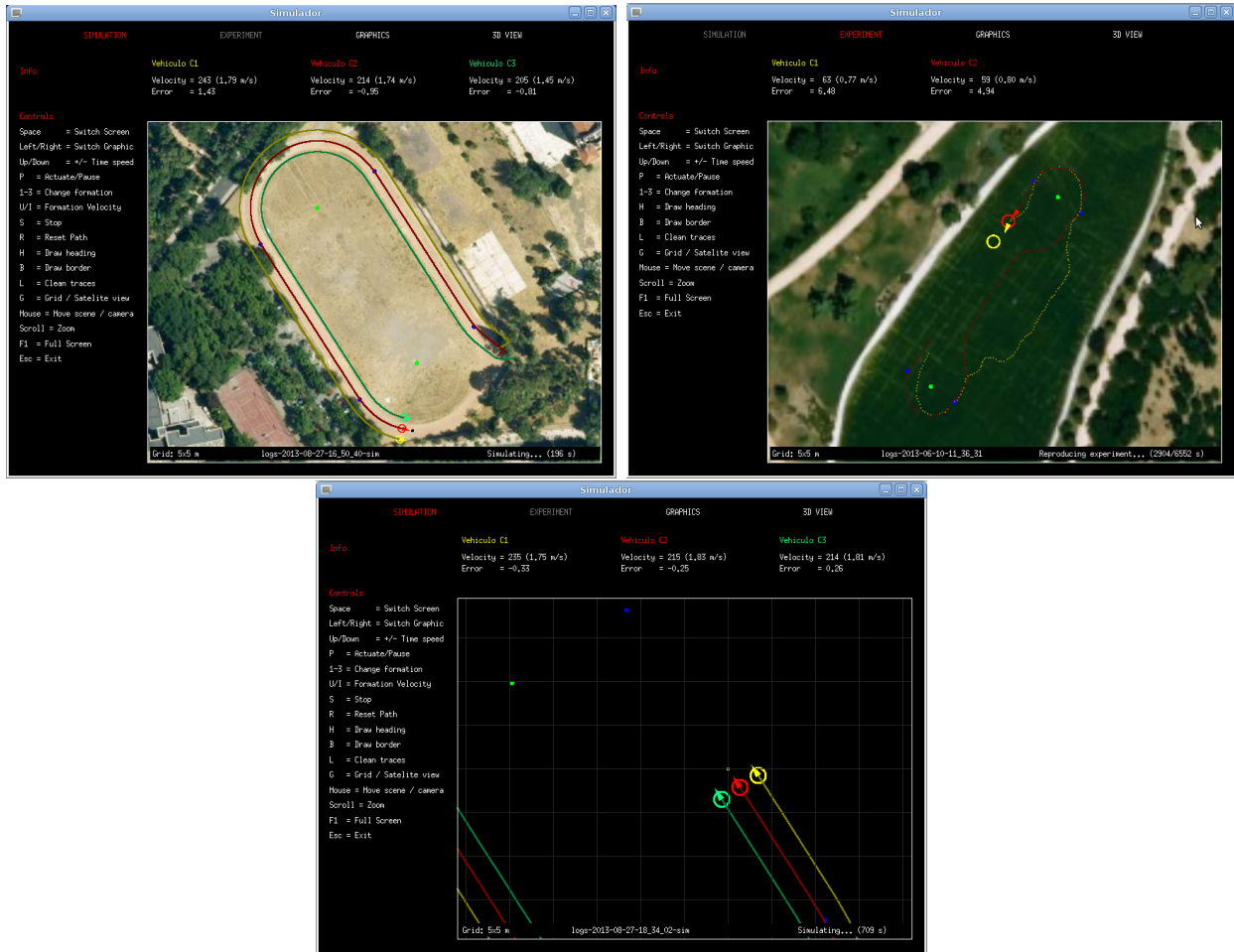


Figura 3.29: Vista Simulation (izquierda), Experiment (derecha) y simulación sobre malla (abajo).

- Una cámara situada en el centro del escenario y que apunta a la formación.
- Una cámara que sigue a la formación a lo largo de su trayectoria.

En cualquier momento, mediante el ratón se puede acercar, alejar o desplazar el escenario para obtener una mejor visualización. Mediante el teclado dependiendo de la vista actual, se permiten distintas acciones:

- Mostrar la siguiente vista.
- Cambiar la gráfica actual (vista *Graphics*).

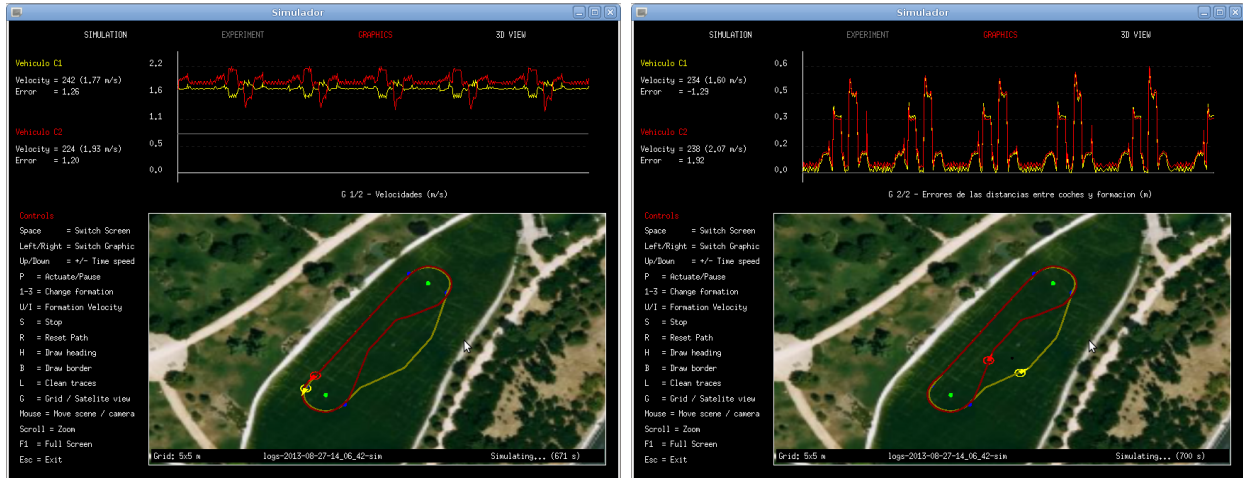


Figura 3.30: Vista Graphics: velocidades (izquierda) y errores (derecha)

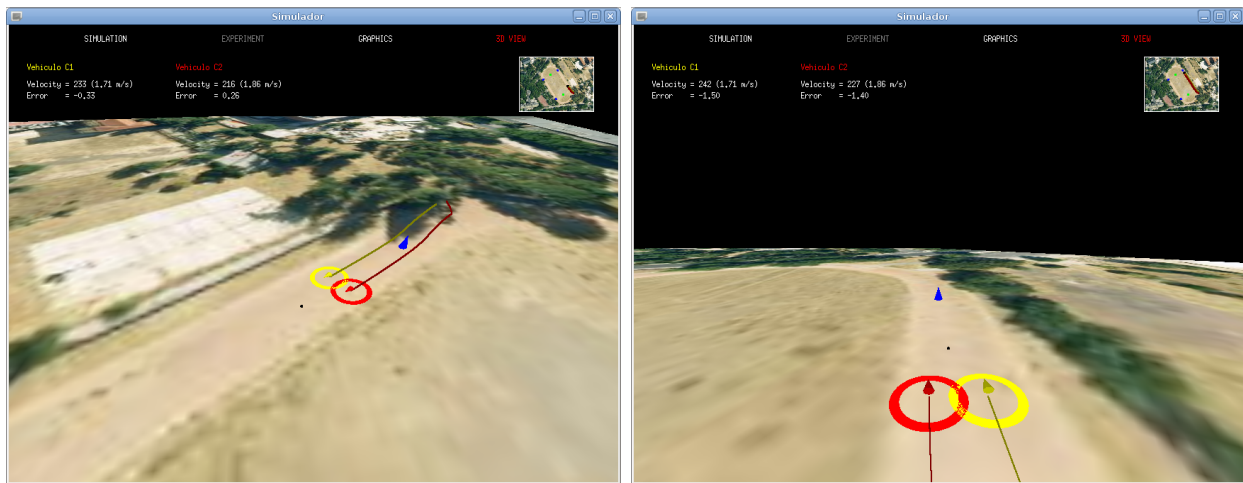


Figura 3.31: Vista 3D: cámara superior (izquierda) y cámara seguidora (derecha)

- Acelerar el tiempo de simulación o de reproducción.
- Pausar la simulación o la reproducción de un experimento.
- Cambiar la geometría de la formación actual.
- Parar un experimento simulado.
- Reiniciar un experimento simulado.
- Representar la orientación actual y la deseada de los robots.

- Limpiar las trazas anteriores.
- Representar el suelo como una malla o la imagen de satélite.
- Mostrar la aplicación en pantalla completa.
- Salir.

Durante la simulación o la reproducción de un experimento se puede acelerar o disminuir la velocidad del tiempo. Esto permite visualizar con mas detalle el comportamiento de los robots, o bien reproducir más rápido partes del experimento que no interesan.

Al simulador se le ha añadido conocimiento experimental. Esto consiste en relacionar los distintos valores que codifican las señales PWM de motor y servo con la velocidad y el ángulo de las ruedas o del timón.

Capítulo 4

Sistema de a bordo

El guiado, navegación, y control (GNC) es una rama de la ingeniería que se ocupa del diseño de sistemas de control de movimiento de vehículos, en concreto, automóviles, embarcaciones, aeronaves, naves espaciales. En muchos casos estas funciones pueden llevarse a cabo por gente capacitada, pero en algunos casos, debido a la velocidad de la dinámica, la reacción humana es muy lenta para poder controlar el movimiento. Incluso en los casos en los cuales los humanos pueden realizar estas funciones, es habitual que sistemas GNC proporcionen beneficios tales como el alivio de la carga de trabajo del operador, el ahorro de combustible, etc. Además, las aplicaciones GNC pueden habilitar el control automático o manual del vehículo.

Los sistemas GNC se encuentran en casi todos los sistemas autónomos o semi-autónomos:

- Pilotos automáticos
- Coches sin conductor
- Misiles guiados
- Sistemas de lanzamiento aéreo de precisión guiada
- Sistemas de control de reacción de naves espaciales
- Vehículos de lanzamiento de naves espaciales

En la literatura se ha empleado para vehículos autónomos aéreos⁴³, submarinos³², marinos^{19,55} y terrestres²⁰.

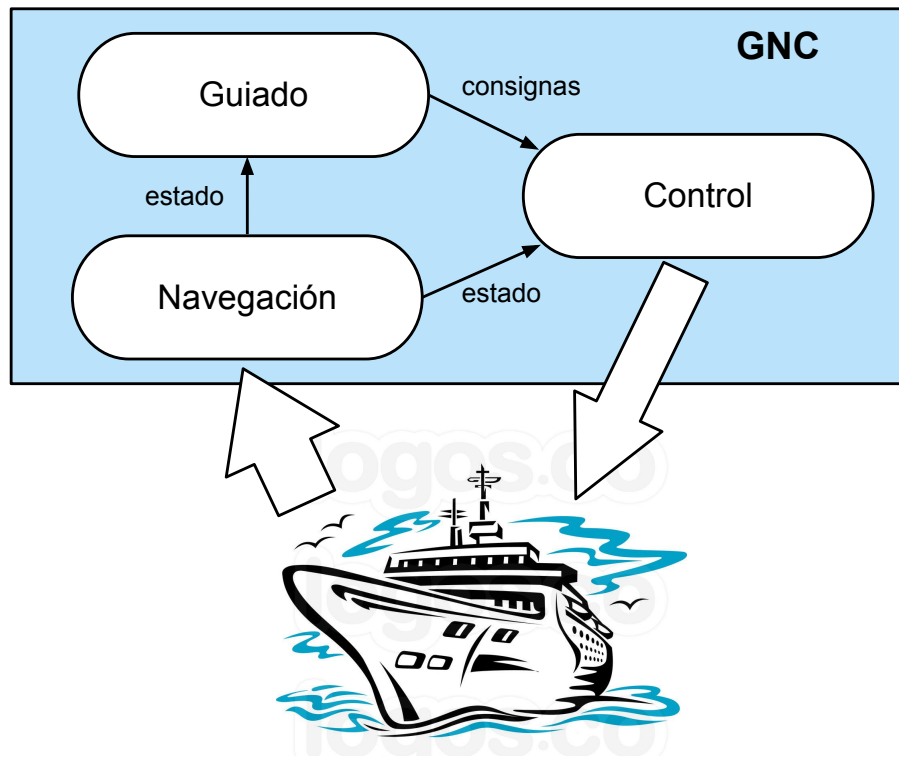


Figura 4.1: *Sistema GNC*

En la figura 4.1 se representa un diagrama del sistema GNC. A continuación se describe la funcionalidad general de cada módulo.

Guiado Normalmente se encarga de la producción de trayectorias para el vehículo que reflejan conocimiento del terreno, las capacidades del robot, la ruta deseada y la colaboración con otros robots.

Navegación Construye el conocimiento sobre el estado del robot, el entorno, y la posición del robot en el entorno. Este conocimiento se puede obtener a través de los sensores con el que está equipado y a partir de la información recibida de los demás robots o del usuario.

Control Recibe las trayectorias del subsistema de Guiado y las modifica, si hace falta, para asegurar el movimiento libre de colisiones del robot. Además controla los actuadores para seguir la ruta segura.

En nuestro caso se ha utilizado la arquitectura GNC para la implementación del sistema de a bordo de los robots marinos y terrestres. A lo largo del desarrollo del proyecto se ha pasado por tres etapas. Se ha aprovechado la modularidad de esta arquitectura, permitiendo en cada etapa el cambio de un módulo sin afectar el funcionamiento de los demás.

En las siguientes secciones se van a describir las tres etapas por las que se ha pasado durante el Trabajo:

1. **Sistema de control de un robot:** presenta la implementación del sistema GNC para el control de un robot autónomo.
2. **Sistema de control de un convoy:** con respecto al sistema GNC anterior presenta cambios al nivel del módulo de control para permitir el seguimiento de la ruta en un convoy formado por varios robots autónomos.
3. **Sistema de control de una formación:** con respecto al primer sistema GNC presenta cambios en el módulo de guiado para permitir el seguimiento de la ruta manteniendo una formación rígida y permitiendo transiciones de formaciones.

4.1. Sistema de control de un robot

La primera etapa de este Trabajo consistió en poner en marcha el sistema existente para un robot, de forma que realice una ruta rectangular marcada por *waypoints* a una velocidad constante.

En la figura 4.2 se representa el sistema y las comunicaciones entre los distintos bloques del sistema GNC. Se puede observar que el subsistema de Navegación se encarga de localizar al robot mediante la información percibida por los sensores. El subsistema de Guiado se

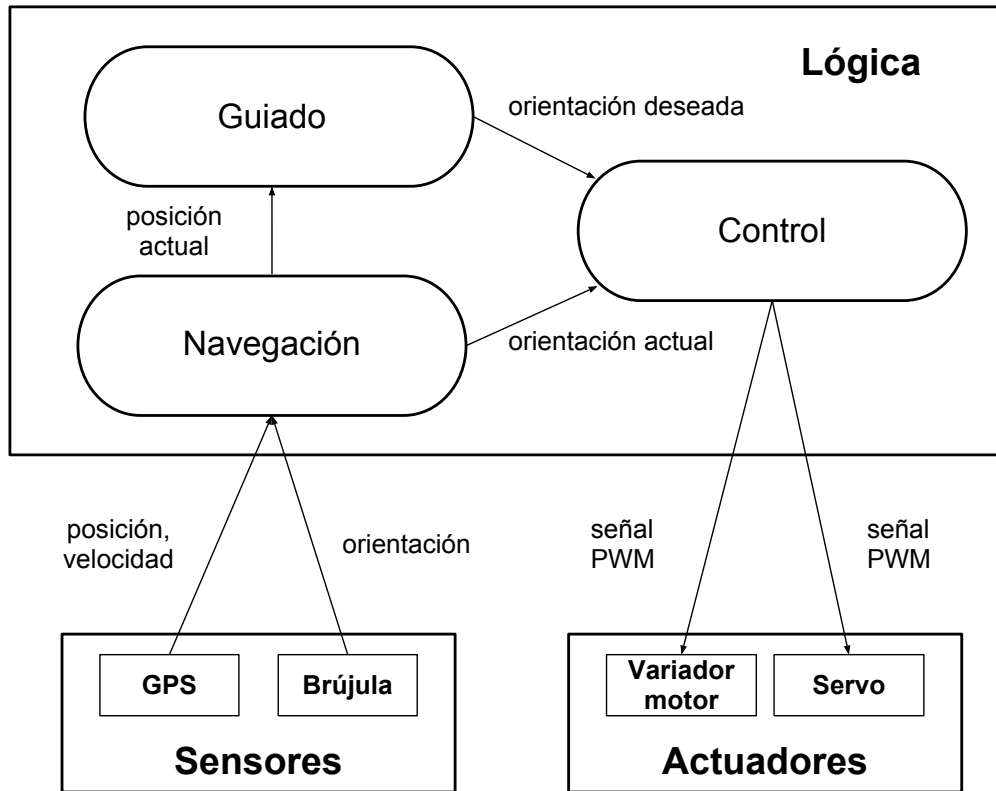


Figura 4.2: Sistema del robot autónomo

encarga de guiar al robot calculando las consigna de orientación. Finalmente el subsistema de Control intenta cumplir las consignas dadas por el subsistema de Guiado controlando los actuadores del robot.

4.1.1. Navegación

El subsistema de Navegación se encarga de construir el estado del robot. Este consiste en la localización y orientación del robot, aspecto imprescindible para la navegación autónoma. Para ello, recibe las señales de los diferentes sensores y los filtra, corrige o estima con el objetivo de ofrecer datos coherentes a los demás subsistemas y evitar comportamientos erróneos.

Localización

En la figura 4.3 están representados los principales sistemas de coordenadas que se pueden utilizar para localizar un punto en la superficie terrestre.

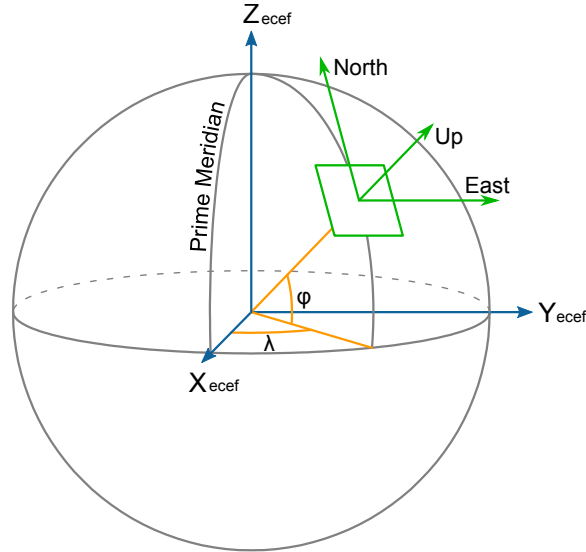


Figura 4.3: *Sistemas de coordenadas*

World Geodetic System 1984 (WGS84) Este sistema geodésico de coordenadas, aproxima la superficie de la Tierra a una elipsoide y las localizaciones se describen en términos de latitud φ , longitud λ , altura h .

Earth-centered earth-fixed (ECEF) Llamado también sistema de coordenadas convencional terrestre, rota con la Tierra y tiene su origen en el centro de la Tierra. El eje X pasa por el ecuador y el primer meridiano. El eje Z pasa por el polo norte. El eje Y puede ser determinado por la *regla de la mano derecha* y pasa por el ecuador a los 90° de longitud.

Local east, north, up (ENU) Es un sistema de coordenadas cartesiano mucho más intuitivo y práctico que ECEF y el sistema Geodésico. Las coordenadas locales ENU está en un plano tangente a la superficie de la Tierra fijado en una localización concreta. Por convención la notación del eje este es x , el norte y y el arriba z .

En nuestro caso los robots disponen de un receptor GPS que devuelve la posición de robot en el sistema de referencia geodésico WGS84, así como su velocidad. La frecuencia con la que se reciben nuevas medidas es de $1Hz$.

Para simplificar los cálculos necesarios para la navegación, la posición se convierte al sistema de referencia local ENU. El origen de este sistema se ha fijado en el centro del área experimental, siendo común para todos los robots.

La conversión de las coordenadas geodésicas a coordenadas locales ENU se realiza en dos pasos:

1. Conversión de coordenadas WGS84 a coordenadas ECEF.

$$X = (N(\varphi) + h) \cdot \cos \varphi \cdot \cos \lambda$$

$$Y = (N(\varphi) + h) \cdot \cos \varphi \cdot \sin \lambda$$

$$Z = (N(\varphi)(1 - e^2) + h) \cdot \sin \varphi$$

donde

$$N(\varphi) = \frac{a}{\sqrt{1 - e^2 \cdot \sin^2 \varphi}}$$

donde a y e^2 son el semi-eje mayor y el cuadrado de la primera excentricidad. Los valores de estos parámetros están definidos en el estándar WGS84:

$$a = 6378137,0m$$

$$e^2 = 6,69437999014 \cdot 10^{-3}$$

2. Conversión de coordenadas ECEF a coordenadas ENU.

Para calcular las coordenadas de un punto R en el sistema ENU de origen O se utiliza la ecuación 4.1. Se necesitan las coordenadas en el sistema ECEF del punto R (X_r, Y_r, Z_r) y las coordenadas ECEF y geodésicas del origen: (X_o, Y_o, Z_o), ($\varphi_o, \lambda_o, h_o$).

$$\begin{bmatrix} E_r \\ N_r \\ U_r \end{bmatrix} = \begin{bmatrix} -\sin \lambda_o & \cos \lambda_o & 0 \\ -\sin \varphi_o \cdot \cos \lambda_o & -\sin \varphi_o \cdot \sin \lambda_o & \cos \varphi_o \\ \cos \varphi_o \cdot \cos \lambda_o & \cos \varphi_o \cdot \sin \lambda_o & \sin \varphi_o \end{bmatrix} \cdot \begin{bmatrix} X_r - X_o \\ Y_r - Y_o \\ Z_r - Z_o \end{bmatrix} \quad (4.1)$$

Orientación

La brújula digital la leemos con una frecuencia de 10 Hz. Una vez obtenidos los valores del campo magnético en los ejes x e y se filtran y se les aplica la corrección en función de los bias encontrados durante el proceso de calibración, tal como se ha comentado en el capítulo anterior.

En la figura 4.4 se vuelve a representar la forma de calcular la orientación del robot a partir de los componentes corregidos del campo magnético.

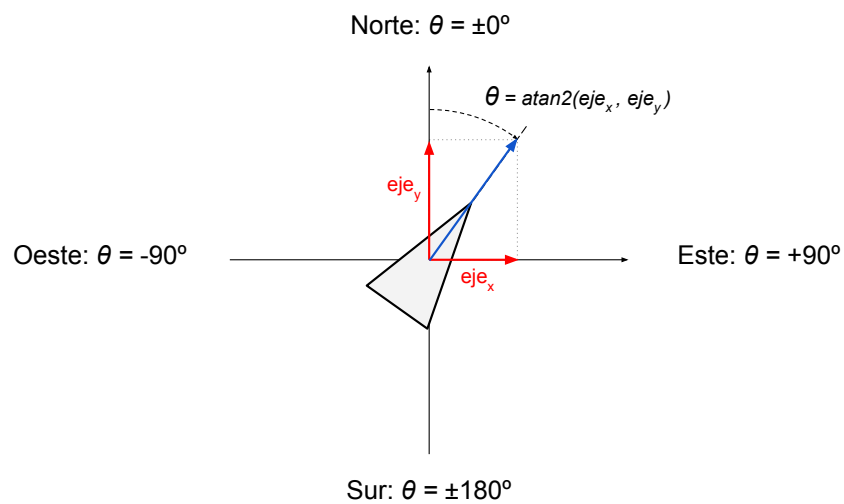


Figura 4.4: Cálculo de la orientación deseada

4.1.2. Guiado

El subsistema de Guiado contiene la lista de *waypoints* en WGS84 que marcan la ruta deseada y un contador que marca el *waypoint* actual al que se dirige el robot. Por tanto, una vez localizado el robot por el subsistema de Navegación, teniendo en cuenta el *waypoint* actual, se calcula la consigna de orientación θ_d , donde E y N son la diferencia entre la posición actual del robot y la del *waypoint*.

$$\theta_d = \text{atan2}(E, N)$$

Puesto que el sistema está propenso a pequeños errores debidos a los sensores empleados y al sistema de control, un *waypoint* se considera alcanzado si el robot se encuentra a una distancia menor que un valor umbral r . Por tanto, la condición de alcance de un *waypoint* es:

$$\sqrt{E^2 + N^2} \leq r$$

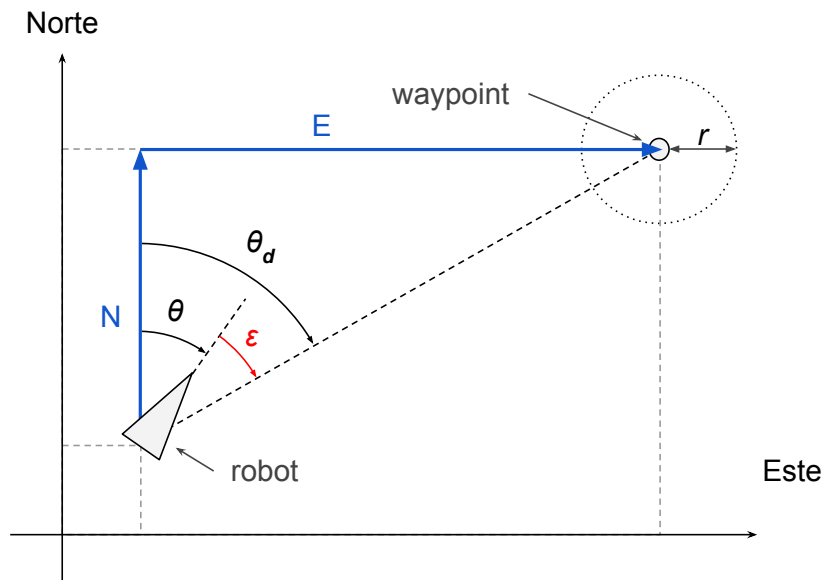


Figura 4.5: *Cálculo de la orientación deseada*

En la figura 4.5 se representa el cálculo de la orientación deseada en función del *waypoint* actual.

En la figura 4.6 se muestra el diagrama de flujo del algoritmo de guiado basado en *waypoints* para el control de un robot autónomo.

La orientación deseada se le pasa como consigna al subsistema de Control, encargado de corregir el rumbo del robot y acercarlo al *waypoint* actual.

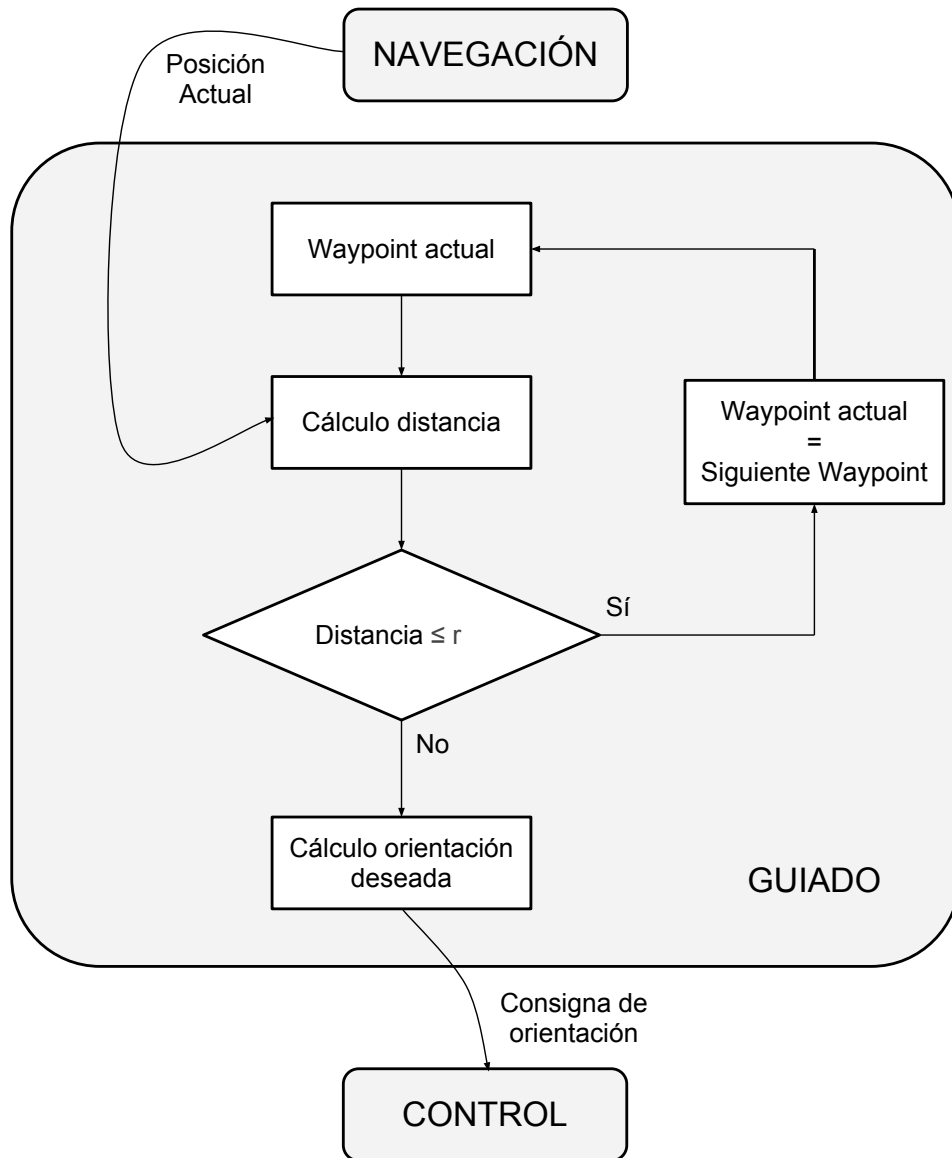


Figura 4.6: Diagrama de flujo del seguimiento de una ruta basado en waypoints

4.1.3. Control

El subsistema de control se encarga de controlar la velocidad y el rumbo del robot, en función de las consignas recibidas desde el sistema de guiado o desde la estación de tierra. Su salida consiste en las señales PWM que van al variador del motor y al servo.

Se ha utilizado en los dos casos la técnica de control por realimentación de tipo PID

(Proporcional Integral Derivativo). El algoritmo del controlador PID implica tres parámetros constantes para las tres componentes del controlador: K_p , K_i , K_d . Desde el punto de vista del tiempo la componente proporcional depende del error presente, la componente integral de la acumulación de los errores pasados y la componente derivativa es una predicción de los errores futuros. El controlador PID calcula un error ε como diferencia entre el valor medido y la consigna, e intenta minimizarlo mediante el ajuste de la señal de control. En la figura 4.7 se puede ver el diagrama del controlador PID.

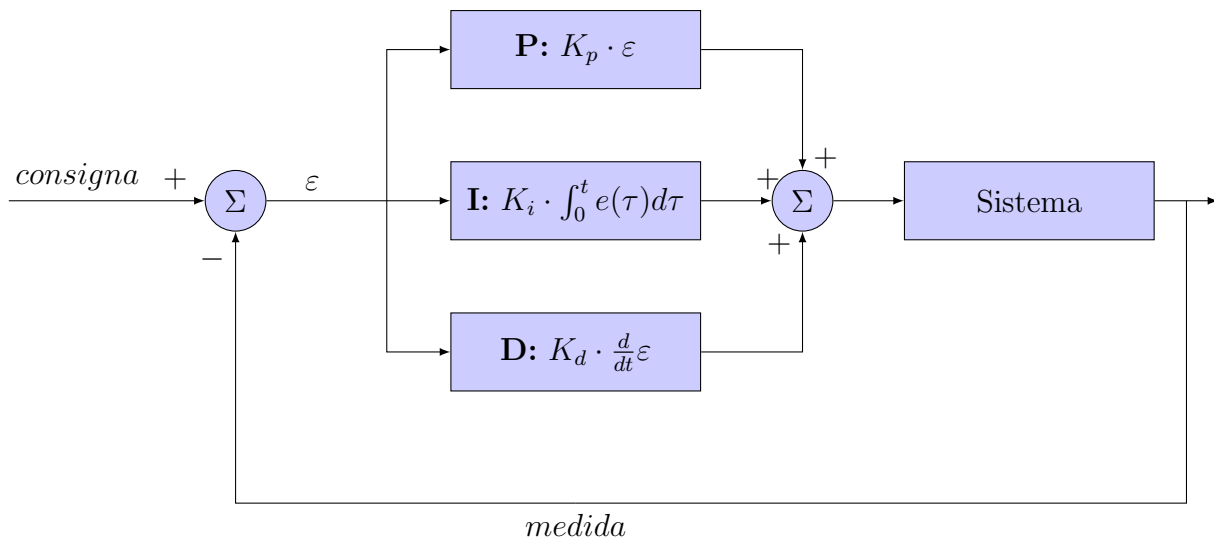


Figura 4.7: Diagrama de un control PID

La configuración del controlador PID tanto para rumbo, como para la velocidad se ha realizado de forma experimental. Mediante la estación de tierra se han cambiado de forma razonable las constantes de control K_p , K_i , K_d hasta que se ha obtenido un comportamiento aceptable del robot.

Control de rumbo

El control de rumbo tiene como objetivo alcanzar y mantener la consigna de orientación recibida desde el nivel superior o fijada desde la estación de tierra.

En la figura 4.8 está representado el diagrama del controlador PID implementado para el control de rumbo. Se puede observar que el controlador calcula el desplazamiento del servo

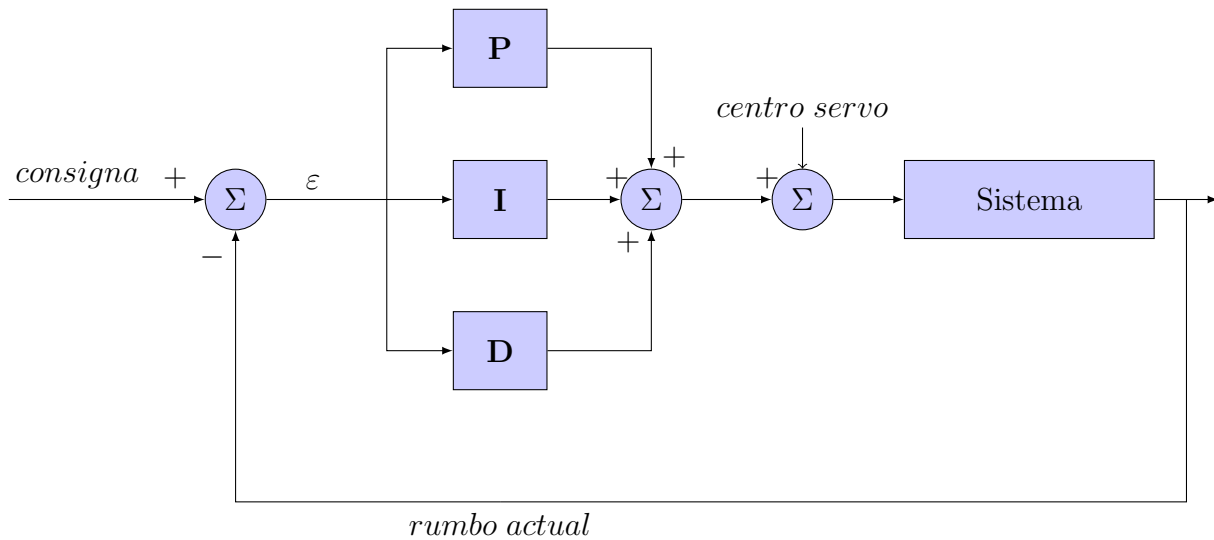


Figura 4.8: Diagrama del control PID de rumbo

con respecto al valor de PWM que lo mantiene en su posición central. De esta forma, si el error es 0, el servo se mantiene en su posición central y por tanto, sigue con la misma orientación.

Se tiene que tener en cuenta que el servo centrado no implica la posición centrada del timón o de las ruedas, por lo cual se debe corregir de forma mecánica este desajuste. En caso contrario, sería imposible que el robot mantuviera la orientación correcta.

Presentamos en la tabla 4.1 las constantes que hemos establecido por el momento, a expensas de realizar más experimentos, para el control de rumbo de los robots terrestres y marinos.

Constante	Robot terrestre	Robot marino
K_p	50	30
K_i	0	0
K_d	0	0

Cuadro 4.1: Configuración del controlador PID de rumbo

Control de velocidad

Igual que en el caso del controlador de rumbo, el control de velocidad tiene como objetivo alcanzar y mantener la consigna de velocidad recibida desde el nivel superior o fijada desde la estación de tierra.

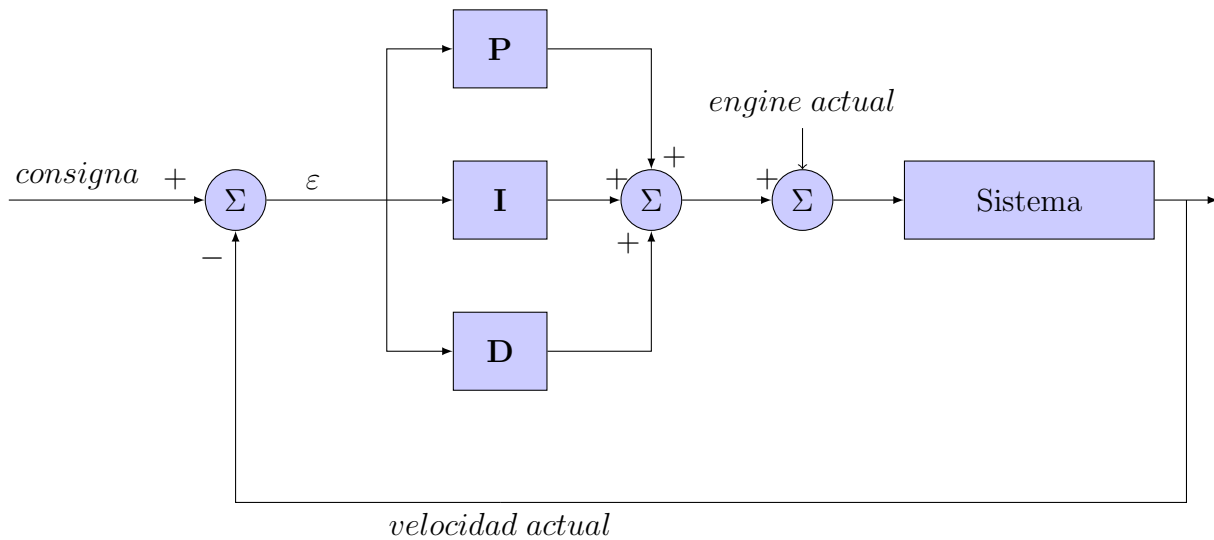


Figura 4.9: Diagrama del control PID de velocidad

En la figura 4.9 está representado el diagrama del controlador PID implementado para el control de velocidad. Se puede observar que el controlador calcula el incremento o decremento de la señal de PWM del motor con respecto a su valor actual. De esta forma, si el error es 0, se mantiene la velocidad actual.

Presentamos en la tabla 4.2 las constantes que hemos establecido por el momento, a expensas de realizar más experimentos, para el control de velocidad de los robots terrestres y marinos.

Constante	Robot terrestre	Robot marino
K_p	5	10
K_i	0	0
K_d	0	0

Cuadro 4.2: Configuración del controlador PID de velocidad

4.2. Sistema de control de un convoy

La siguiente etapa se ha centrado en crear un sistema multi-robot de forma que la ruta se siga en convoy. Se ha elegido una arquitectura Líder-Seguidor, en la que el líder del convoy sigue de forma autónoma y a velocidad constante la ruta mientras que los demás robots se encargan de seguir la ruta y mantener una distancia concreta con respecto al líder.

Por tanto, cada robot tiene asignado un rol en el convoy: líder, primer seguidor, segundo seguidor, etc tal que el seguidor i -ésimo tiene que mantener una distancia $i \cdot d$ con respecto al líder. Cada robot conoce la ruta y cada seguidor sabe quien es el líder del convoy. Por tanto, para mantener la distancia deseada controla su velocidad de forma adecuada.

Para calcular la distancia al líder se utiliza la cooperación a través de la comunicación radio. Con una frecuencia de $5Hz$ el líder envía a todos los seguidores su posición, el índice del *waypoint* al que se dirige y las coordenadas del punto en el cual ha alcanzado el último *waypoint*.

En la figura 4.10 se muestra la forma de calcular la distancia entre el líder y el seguidor. Se debe tener en cuenta si el líder se dirige al mismo *waypoint*. Se ha decidido estimar la distancia como la suma entre la distancia entre el seguidor y el punto donde el líder ha alcanzado el *waypoint* y la distancia entre ese punto y el líder. Si hay más de un *waypoint* de diferencia, se suman los tramos entre los distintos puntos de alcance.

Con respecto al sistema de seguimiento individual descrito anteriormente el líder sigue el mismo comportamiento y los seguidores utilizan el mismo subsistema de Navegación, Guiado y Control de orientación.

Para controlar la velocidad, los seguidores utilizan un controlador PID representado en la figura 4.11. La consigna es la distancia deseada con respecto al líder y el error es:

$$\varepsilon = distancia_{deseada} - distancia_{actual}$$

Por tanto, ante una distancia mayor a la consigna, se aumenta la velocidad del robot y ante una distancia menor, la velocidad se disminuye.

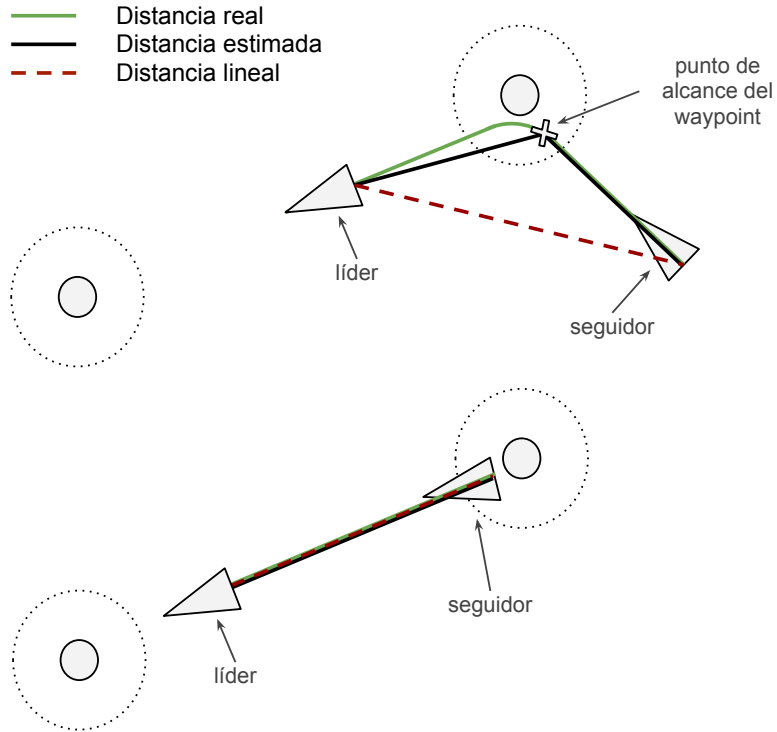


Figura 4.10: Cálculo de la distancia entre el líder y el seguidor

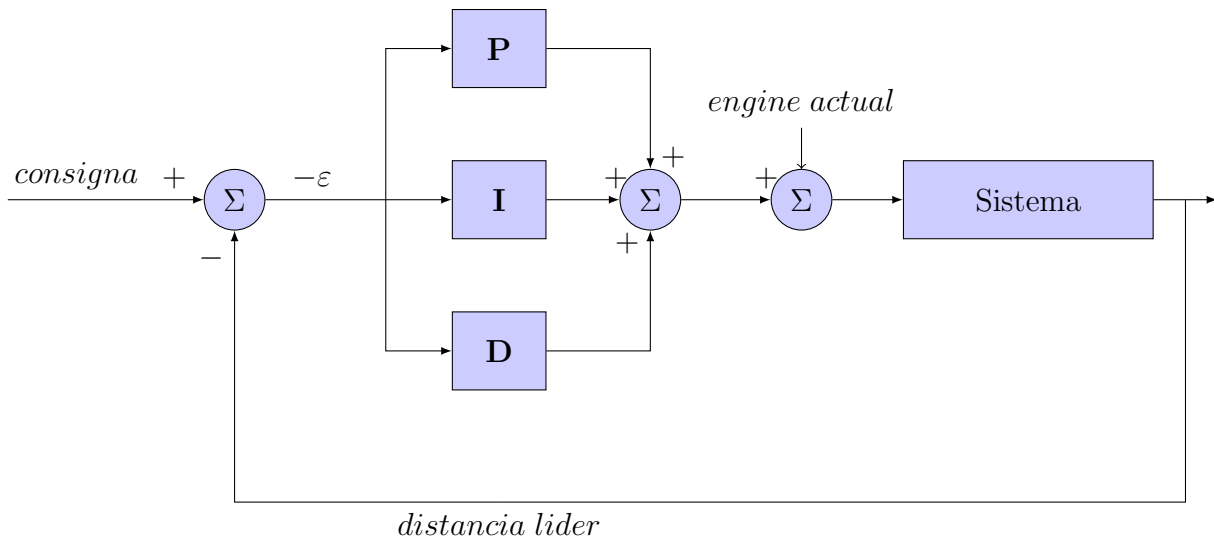


Figura 4.11: Diagrama del control PID de velocidad

Este sistema se ha probado con dos robots terrestres en varios escenarios experimentales: una ruta rectangular de 4 *waypoints* y una pista de atletismo marcada con 14 *waypoints*.

El funcionamiento con más robots se ha comprobado en simulación.

4.3. Sistema de control de formaciones

El objetivo del presente trabajo es crear un sistema que permita mantener y guiar una formación de robots autónomos a lo largo de una ruta preestablecida. La plataforma experimental descrita en el capítulo anterior sirve para probar el sistema tanto en simulación como experimentalmente en tierra y en agua. Teniendo en cuenta los robots y el hardware del que disponen se pueden establecer las comunicaciones entre el sistema y el exterior.

Por tanto, las entradas del sistema son:

- Las medidas obtenidas desde los distintos sensores: posición, velocidad, orientación.
- La información recibida desde los demás robots: su posición y el indicador de pertenencia a la formación.

Las salidas del sistema consisten en las dos señales PWM enviadas a los actuadores que permiten el desplazamiento de los robots.

En la figura 4.12 se presenta un esquema de los distintos módulos y subsistemas encargados de aportar la autonomía al robot. Se distinguen el subsistema de navegación que recoge los datos de los sensores, el subsistema de guiado que se encarga de calcular las consignas de posición en función de la ruta deseada y del estado de la formación, y el subsistema de control que activa los actuadores.

Se observa que el sub-sistema de control presenta dos niveles: NSB y control de velocidad y rumbo. De esta forma se desacoplan los problemas de coordinación y maniobra permitiendo tratarlos por separado.

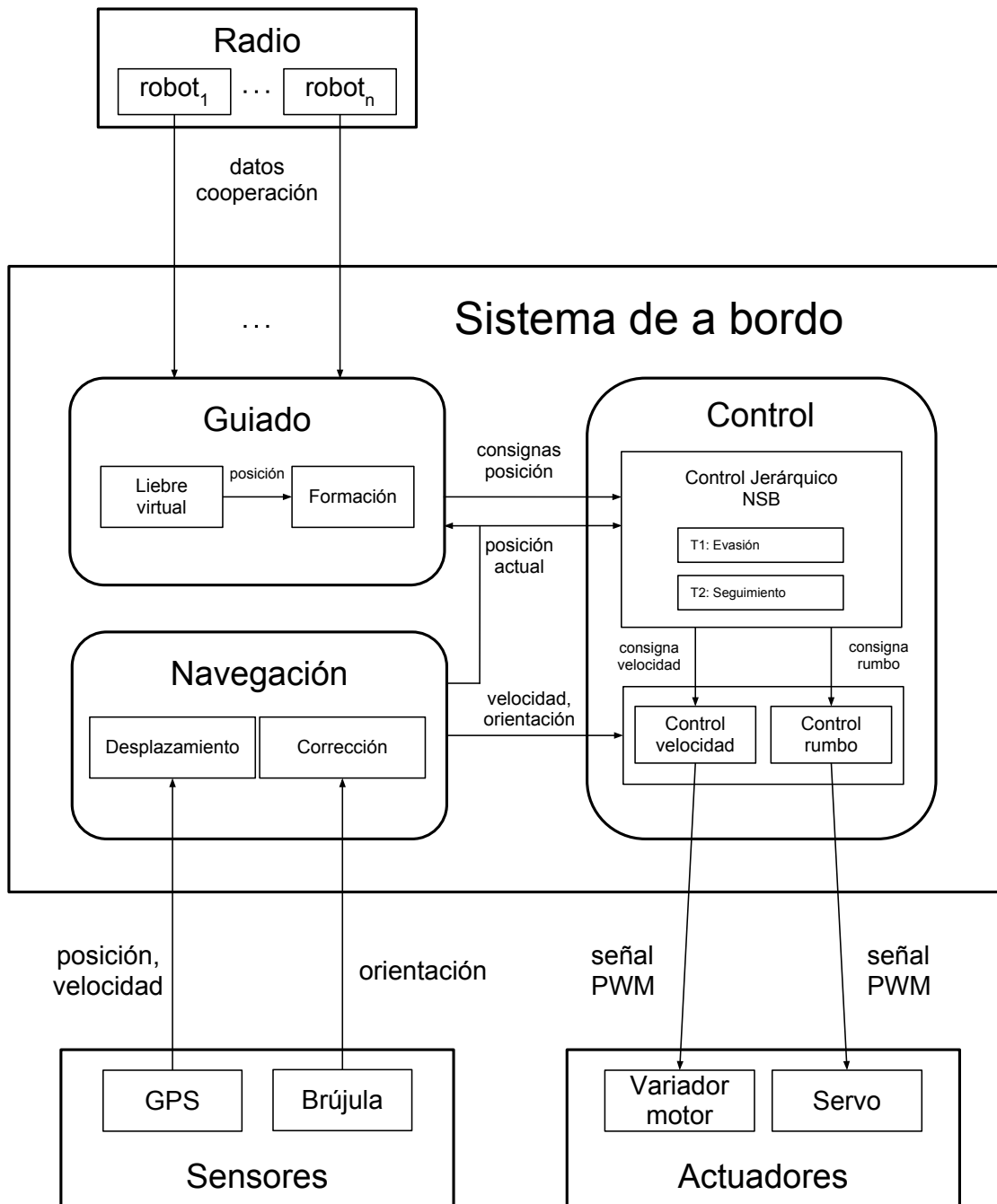


Figura 4.12: Sistema del robot autónomo

4.3.1. Ruta y Seguimiento

Hoy en día, tal como se ha presentado en el capítulo del Estado del Arte, hay distintas técnicas de planificación de rutas mediante visión por computador, una lista de puntos de paso, distintos tipos de curvas, etc.

El seguimiento de la ruta por los robots se realiza mediante la técnica *follow-the-rabbit* ^{16,44,47}. Este método consiste en tener una liebre virtual que sigue la ruta deseada a una velocidad prefijada y controlar los robots de forma que centrado una formación virtual deseada en la posición de la liebre y dándole su orientación, los robots converjan a dicha formación.

Puesto que el objetivo principal del trabajo es crear formaciones, se ha decidido utilizar rutas preestablecidas y definidas de forma sencilla mediante tramos rectos y tramos de curvas circulares. Este tipo de rutas lo propuso L.E. Dubins ^{15,27,33} para definir la ruta más corta entre dos posiciones para un “coche Dubins”; este coche presenta un radio de giro mínimo y no tiene marcha atrás.

La ruta de Dubins viene determinada por el radio de giro mínimo del coche, su posición inicial y la posición final deseada. Dubins demuestra que la ruta más corta es una ruta de tipo $LRL, RLR, LSL, LSR, RSL, RSR$ donde L representa giro a la izquierda, R representa giro a la derecha y S representa un tramo recto. En la figura 4.13 se ilustran dos ejemplos de rutas de Dubins.

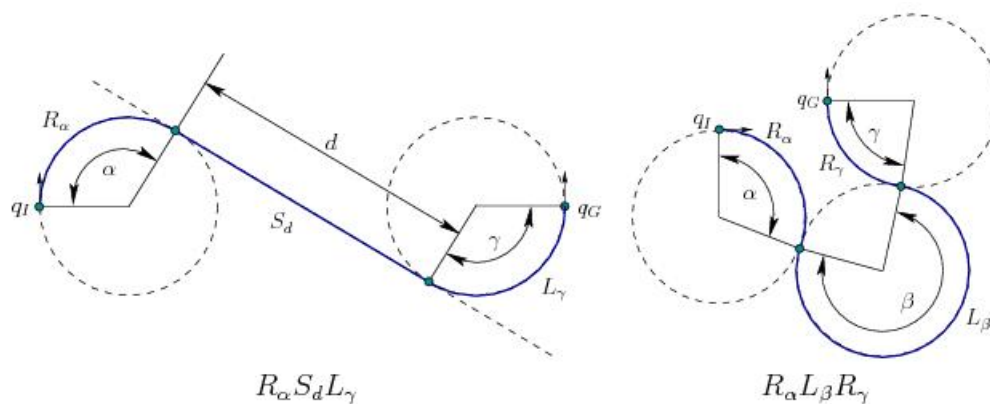


Figura 4.13: *Rutas de Dubins* ⁽⁴⁶⁾

En este trabajo se ha utilizado el concepto de “rutas de Dubins” para definir las rutas deseadas de una forma más sencilla. Así se puede describir la ruta para recorrer la pista de carreras de forma fácil y nos ofrece suficiente flexibilidad a la hora de crear otros tipos de ruta. Para ello, hace falta proporcionar una serie de puntos que representan los centros de giro, los radios de giro y los sentidos de giro. Además se tiene que especificar si la ruta es cíclica o no. En la figura 4.14 se muestra un ejemplo de ruta aleatoria cíclica definida mediante este método.

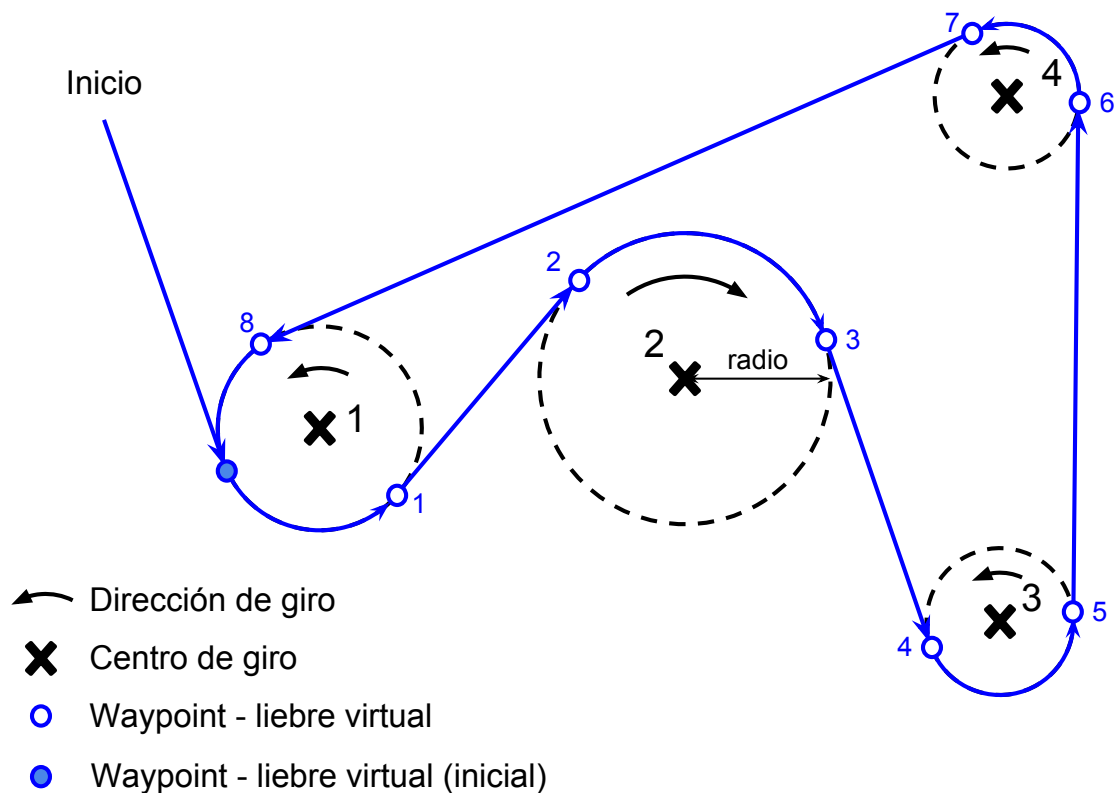


Figura 4.14: *Ejemplo de ruta utilizando curvas de Dubins*

Al asignarle una ruta al robot, se calculan de forma automática los *waypoints - liebre virtual* marcados en la figura 4.14. Estos puntos representan las intersecciones de las tangentes correspondientes entre cada dos circunferencias consecutivas, en función del sentido de giro de cada una.

Una vez obtenidos los *waypoints*, al iniciar un experimento, se calcula el *waypoint - liebre*

virtual (inicial) como la intersección entre la tangente a la primera circunferencia y el centro de formación. Además se le tiene que asignar una velocidad a la liebre virtual. A partir de este punto, la liebre recorre la ruta pasando por cada *waypoint* y parando en el último, o bien siguiendo la ruta de forma cíclica hasta que se detiene el experimento desde la estación de tierra.

Considerando la posición de la liebre como $\langle x_{liebre}, y_{liebre}, \theta \rangle$, su movimiento en cada paso se calcula de la siguiente manera, dependiendo del tramo en el que se encuentra.

- En tramo recto:

$$x = x_{waypoint} - x_{liebre}$$

$$y = y_{waypoint} - y_{liebre}$$

$$\theta = atan2(y, x)$$

- En curva ($c = -1$ si el sentido de la curva es derecha o $c = 1$ si el sentido de la curva es izquierda.):

$$dH = \frac{v \cdot dt}{r}$$

$$\theta = \theta + c \cdot dH$$

Una vez calculada la nueva orientación, el desplazamiento de la posición es:

$$x_{liebre} \dot{=} v \cdot \cos \theta \cdot dt$$

$$y_{liebre} \dot{=} v \cdot \sin \theta \cdot dt$$

4.3.2. Formaciones

Para definir una formación en el plano bidimensional se utiliza una serie de coordenadas o posiciones relativas con respecto a la liebre virtual. Cada robot, en función de la geometría de la formación y del índice de la posición que se le ha asignado, calcula automáticamente su coordenada en la formación. Por el momento el índice en la formación de cada robot está

prefijado. Solamente cambia si mediante la cooperación detecta que está solo en la formación y entonces ocupa la primera y única posición. Esta situación se puede dar cuando se utiliza solamente un robot o bien se retira el otro robot de la formación.

En la figura 4.15 se representa la formación deseada, calculada en función de su posición relativa, de la posición de la liebre virtual y de la orientación dada por la liebre virtual.

Las posiciones locales deseadas de cada robot se calculan pasando del sistema de coordenadas de la formación, al sistema de coordenadas ENU.

$$p_{local}.x = p_{relativa}.y * \cos(\theta) - p_{relativa}.x * \sin(\theta) + p_{liebre}.x$$

$$p_{local}.y = p_{relativa}.y * \sin(\theta) + p_{relativa}.x * \cos(\theta) + p_{liebre}.y$$

Donde p_{liebre} es la posición de la liebre y θ es su orientación.

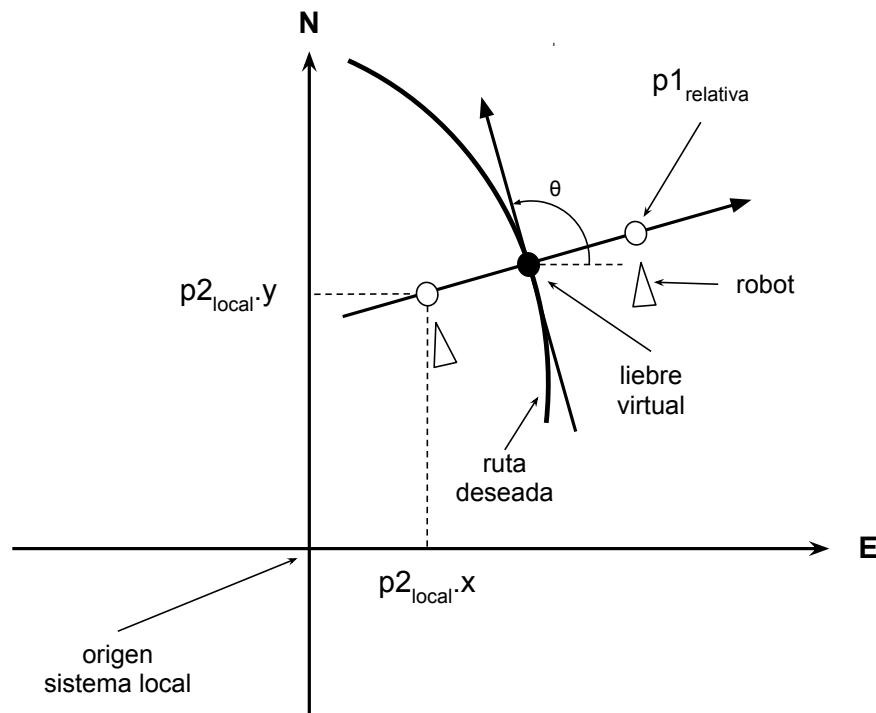


Figura 4.15: *Cálculo de las posiciones deseadas*

4.3.3. Transiciones

El sistema experimental permite realizar cambios de formación pasando por una etapa de transición que modifica las coordenadas de todos los robots de forma suave. La duración de esta etapa está restringida por una distancia en la que debe completarse la transición. De esta forma se puede configurar una transición más suave o más brusca.

Las transiciones se pueden configurar y activar de forma manual a través de la BSD. También se le puede asignar a cada ruta una lista de transiciones automáticas y puntos de activación. En nuestro caso, cuando la liebre virtual se encuentra a menos de 3 metros del punto de activación empieza la etapa de transición. Cada robot calcula los desplazamientos en X e Y que tiene que aplicar a su coordenada o posición relativa en cada paso.

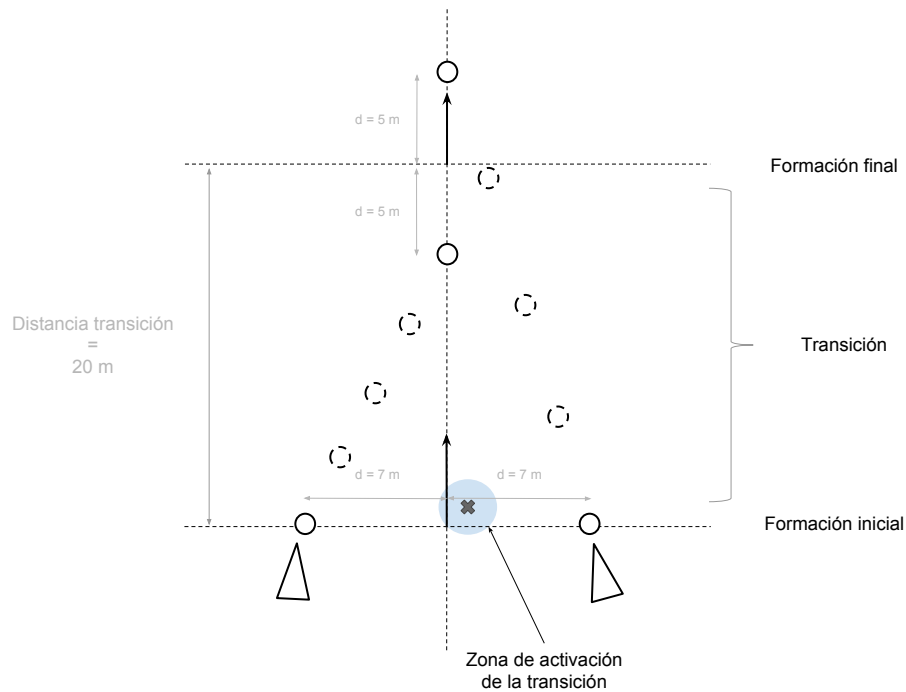


Figura 4.16: Representación de una transición automática de formación línea a columna en 3 pasos, con una restricción de 20 metros

En la figura 4.16 se ha representado una transición automática a partir de una formación línea a una columna con distintas separaciones y una restricción de 20 metros para realizar

el cambio.

Los desplazamientos en los ejes X e Y del sistema liebre para cada paso de la transición y cada posición virtual se calcula teniendo en cuenta la distancia deseada, la velocidad actual de la liebre y el diferencial de tiempo en el cual se actualizan las posiciones de la formación. En cada paso se suman a las posiciones actuales. Esto garantiza que si no se cambia la velocidad de la liebre, al haber recorrido la distancia impuesta como restricción, las posiciones virtuales marcarán la geometría de la formación deseada.

$$\begin{aligned}dt &= 0,1s \\duracion &= \frac{distancia_transicion}{velocidad_centroide} \\pasos &= duracion/dt \\dX &= \frac{X_{final} - X_{inicial}}{pasos} \\dY &= \frac{Y_{final} - Y_{inicial}}{pasos}\end{aligned}$$

Teniendo la posibilidad de calcular en cada momento la posición deseada se quiere por tanto que el robot vaya hacia la posición objetivo y que no choque con los demás miembros de la formación. Estas dos tareas describen distintos comportamientos que se tienen que coordinar para obtener una única salida adecuada en cada momento.

4.3.4. Control jerárquico *Null-Space Based*

El modo de control reactivo *Null-Space Based*^{6,10} es un método de coordinación de comportamientos. La diferencia con otros métodos de este tipo consiste en la manera de combinar las salidas de los distintos comportamientos, siempre que estos no entran en conflicto. Se necesita asignar un orden de prioridad a cada comportamiento. El algoritmo consiste en proyectar las salidas de los comportamientos de menor prioridad sobre el espacio nulo de los comportamientos de mayor prioridad. De esta forma se obtiene una única salida que

tiene en cuenta todos los comportamientos y siempre lleva a cabo el comportamiento más prioritario.

A continuación se presenta el modelo matemático del método NSB y se describen los distintos comportamientos definidos en nuestro caso.

Modelo matemático

El sistema considera como $p \in \mathbb{R}^n$ la configuración del robot móvil y define una función de comportamiento $\sigma \in \mathbb{R}^m$ como:

$$\sigma = f(p) \quad (4.2)$$

Y la relación diferencial:

$$\dot{\sigma} = \frac{\partial f(p)}{\partial p} v = J(p)v \quad (4.3)$$

donde $J \in \mathbb{R}^{m \times n}$ es la matriz Jacobiana del comportamiento σ y $v \in \mathbb{R}^n$ es la velocidad del robot. Utilizando la solución denominada *Closed Loop Inverse Kinematics* (CLIK) la referencia es

$$V_d = J^\dagger(\dot{\sigma}_d + \Lambda \tilde{\sigma})$$

donde $J^\dagger = J^T(JJ^T)^{-1}$ es la pseudo-inversa de la matriz Jacobiana, Λ es una matriz de ganancias definida positiva y $\tilde{\sigma}$ es el error del comportamiento, definido como $\tilde{\sigma} = \sigma_d - \sigma$.

En el caso de varios comportamientos, el comportamiento i -ésimo es,

$$v_i = J_i^\dagger(\dot{\sigma}_{i,d} + \Lambda_i \tilde{\sigma}_i) \quad (4.4)$$

Los comportamientos de menor prioridad se proyectan sobre el espacio nulo de los comportamientos de mayor prioridad, de forma que con tres comportamientos se obtendría

$$v_d = v_1 + N_1(v_2 + N_2 v_3) \quad (4.5)$$

En nuestro caso los comportamientos que tienen los distintos robots son:

1. **Evasión:** no chocar con los demás miembros de la formación.
2. **Seguimiento en formación:** seguir la ruta en formación.

El modelado de las funciones de comportamiento utilizadas se describe a continuación.

Comportamiento de Evasión

El comportamiento de evasión es el más prioritario para garantizar la seguridad de los robots. Se quiere mantener una distancia mínima d entre los robots. Por tanto, este comportamiento solamente se activa si la distancia entre el robot y otro miembro de la formación es menor que d . La función de este comportamiento se define como:

$$\sigma_1 = \|p - p_0\| \in \mathbb{R} \quad (4.6)$$

$$\sigma_{1,d} = d \quad (4.7)$$

$$\hat{r} = \frac{p - p_0}{\|p - p_0\|} \quad (4.8)$$

$$J_1 = \hat{r}^T \in \mathbb{R}^{1 \times 2} \quad (4.9)$$

Donde p_0 es la posición del obstáculo, $\sigma_{1,d}$ es el valor deseado para este comportamiento (la distancia mínima que debe mantenerse entre el robot y el obstáculo).

Utilizando las ecuaciones anteriores se llega a la consigna de velocidad para este comportamiento:

$$v_1 = J_1^\dagger \lambda_1 (d - \|p - p_0\|) \quad (4.10)$$

$$N_1 = I - \hat{r} \hat{r}^T \quad (4.11)$$

Donde λ_1 es la ganancia para este comportamiento.

Comportamiento de Seguimiento en formación

El comportamiento de seguimiento trata de acercar al robot a una posición deseada p_d . Esta posición se calcula en función de la formación deseada y de la posición y orientación de la liebre virtual.

Por tanto el comportamiento de seguimiento controla la posición del robot:

$$\sigma_2 = p \in \mathbb{R}^{2 \times 1} \quad (4.12)$$

$$\sigma_{2,d} = p_d \in \mathbb{R}^{2 \times 1} \quad (4.13)$$

$$J_2 = I \in \mathbb{R}^{2 \times 2} \quad (4.14)$$

La consigna de velocidad para este comportamiento es:

$$v_2 = \Lambda_2(p_d - p) \quad (4.15)$$

Donde Λ_2 es una matriz de ganancias.

Combinación de comportamientos

Una vez calculadas las consignas de velocidad para cada comportamiento, se combinan para obtener el vector velocidad resultante \vec{v} .

$$v = v_1 + N_1 \cdot v_2$$

En la figura 4.17 se representa la forma de combinar las velocidades de cada comportamiento para obtener el vector velocidad final mediante NSB. v_1 representa el vector velocidad para la evasión y v_2 es el vector velocidad para el seguimiento en formación. v_2 se proyecta sobre el espacio nulo del Jacobiano del comportamiento de mayor prioridad (evasión).

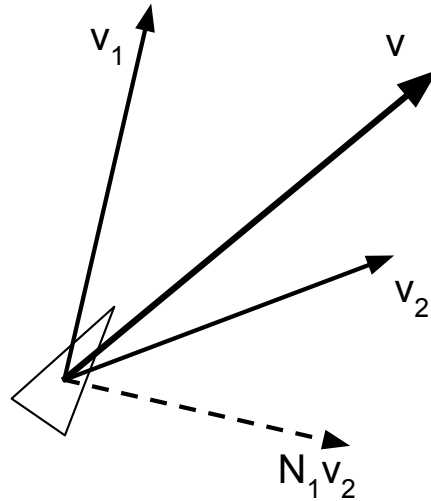


Figura 4.17: NSB: combinación de las velocidades de cada comportamiento

A partir del vector velocidad resultante se calcula la consigna de velocidad v_d y la consigna de rumbo θ_d .

$$v_d = \|\vec{v}\|$$

$$\theta_d = \text{atan2}(\vec{v}.x, \vec{v}.y)$$

Puesto que v_d es la velocidad necesaria para alcanzar la posición deseada, una vez alcanzada sería $0m/s$, pero nosotros necesitamos que se siga la ruta a la velocidad de la liebre. Por tanto, la consigna de velocidad la calculamos de la forma:

$$v_d = v_{liebre} + \|\vec{v}\|$$

donde v_{liebre} es la velocidad de la liebre virtual y las ganancias del NSB se ajustan de forma que el robot tenga el comportamiento deseado.

Estas consignas se pasan al módulo de control de rumbo y velocidad que finalmente controla los distintos actuadores, tal como se ha descrito anteriormente.

Capítulo 5

Trabajos Experimentales

Existen diversas metodologías para la ejecución de proyectos de desarrollo e investigación. Por ejemplo, un caso típico es el proceso estandar en V, el cual empieza con una etapa de requerimientos, sigue con etapas de desarrollo, y sólo al final realiza experimentos de comprobación. Si bien esta metodología es adecuada para algunos casos, tiene graves inconvenientes si al final los experimentos no dan buen resultado.

En el caso de la robótica, es más adecuado ir construyendo comportamientos por etapas, empezando por las más simples, y comprobando en cada etapa mediante experimentos si las soluciones parciales propuestas van bien. De alguna manera es como construir un edificio planta a planta, empezando por los cimientos. Desde el punto de vista de la ingeniería de procesos, se trata un desarrollo en espiral, que comienza por un prototipo básico y lo va enriqueciendo en sucesivas vueltas de la espiral.

Por ejemplo, si queremos realizar un experimento en el que dos robots tienen que avanzar en paralelo, debemos tener garantizado que cada uno de los robots es capaz de controlar de modo autónomo su propio movimiento. Pero para ello, se han de realizar otros experimentos, anteriores, con robots individuales. Es decir, el estudio experimental debe seguir una secuencia apropiada, por pasos.

Así pues, para el desarrollo de este trabajo se ha utilizado la metodología en espiral, representada en la figura 5.1.

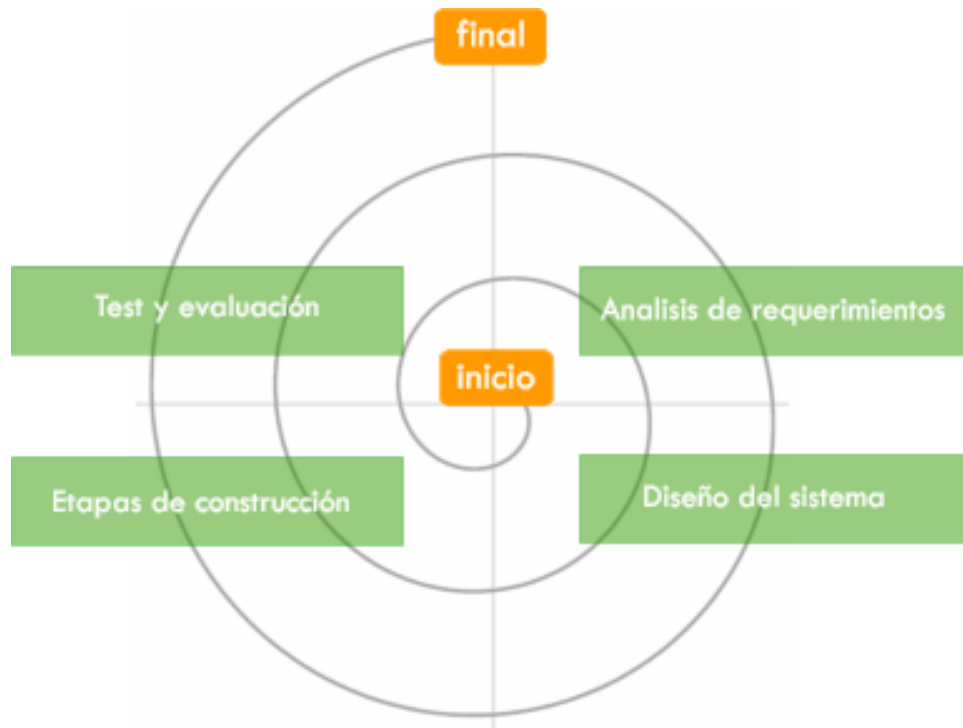


Figura 5.1: Metodología en espiral, con las distintas etapas y las fases de cada una

Las distintas etapas que se han realizado son:

1. **Comportamiento individual:** seguimiento individual de una ruta.
2. **Comportamiento en convoy:** convoyes automáticos.
3. **Formaciones:** mantener formaciones, con posibilidad de transiciones.

Para seguir un orden académico se han expuesto en los capítulos anteriores los aspectos de control en un orden sistemático, el seguimiento de la espiral ha supuesto la introducción gradual y con diversas versiones de diversos aspectos funcionales. En este capítulo se describen según el orden cronológico las diversas aportaciones realizadas, a medida que los estudios experimentales han ido ganando en complejidad.

5.1. Bases del estudio experimental

Uno de los aspectos cruciales que se plantean a la hora de realizar estudios experimentales de robótica es la escala y el alcance tecnológico.

En los estudios ya realizados por el Grupo ISCAR acerca de la estabilización de barcos, se utilizaron barcos a escala de unos 3 metros de longitud. La escala era adecuada para el estudio experimental en una piscina de olas, de 150 metros de longitud, en el Canal de Experiencias Hidrodinámicas de El Pardo (CEHIPAR). Sin embargo, cuando al cabo de unos años se empezaron trabajos sobre maniobras de barcos, se vio que es una escala demasiado grande para las instalaciones del CEHIPAR. Quedan dos opciones, o continuar con escalas del orden de varios metros navegando en pantanos (si el tiempo lo permite), o bien bajar a escalas más reducidas que permitan estudios experimentales en estanques o similares. La ventaja de escalas más pequeñas es que los experimentos se pueden hacer más rápidamente, a lo largo de casi todo el año, y sobre todo que si hay problemas (averías, choques entre barcos o con obstáculos, etc.) éstos no revisten especial gravedad y normalmente se pueden solventar sin grandes dificultades. De hecho, se han podido desarrollar con relativa rapidez los métodos de control que se presentan en esta Memoria, y ahora es interesante utilizarlos en escalas más grandes.

Otro de los aspectos que se han considerado es el del empleo de carros y de barcos. Se decidió utilizar carros para poder acelerar los estudios experimentales. La idea es probar los diversos algoritmos primero en carros, porque esto se puede hacer con rapidez en una pista de atletismo cercana a nuestra Facultad. Una vez comprobados los algoritmos, el paso siguiente es probarlos en barcos, lo cual requiere ir al parque Tierno Galván, transportando todo el material experimental y de soporte necesario. Este procedimiento, carros-barcos, se ha demostrado muy efectivo.

Como se trata de desplegar varios robots móviles, aparece un problema de costes. A esto se añade que a pequeña escala no todo es posible en cuanto a tecnología; por ejemplo, un GPS diferencial es demasiado grande y pesado; y tampoco podemos colocar todo un PC en

un pequeño carro o barco. Esto da razón de las elecciones tecnológicas que se han descrito en el Capítulo 3. Nuestros robots móviles no son excesivamente caros, tienen un GPS de una precisión aceptable (aunque evidentemente limita algunas posibilidades) y una brújula también aceptable en cuanto a precisión y fiabilidad; y en cuanto al procesador de a bordo, la tecnología actual permite buenas prestaciones y sin una excesiva limitación de memoria.

Al comentar estos aspectos aparece de modo natural el porqué se han elegido determinados entornos experimentales. Para los carros un pista de atletismo cercana, y para los barcos un conjunto de estanques públicos en los que no se necesita permiso para experimentar.

A continuación, se detallan cuestiones prácticas asociadas a la realización de estudios experimentales. Cada sesión experimental al aire abierto requiere una cierta disciplina, con tres etapas principales: preparación, ejecución, y análisis de resultados.



Figura 5.2: *Localización de la pista.*

5.1.1. Preparación de un experimento

A la hora de preparar un experimento se realizan una serie de pasos y se utilizan varias herramientas:

Google Earth se utiliza para obtener las coordenadas GPS de los distintos *waypoints* que se quieren elegir para la ruta.

Simulador se simula el experimento con la nueva ruta y se valida la ruta elegida y los cambios de formación si durante la simulación, los robots no se acercan demasiado a los obstáculos del entorno: bordes, orillas, fuentes, etc.

Base de Supervisión a Distancia (BSD) se instala la última versión del software de la estación de tierra y se copia la última versión del software de a bordo, para la posible necesidad de hacer un cambio rápido y volver a programar los robots.

Los robots necesitan una serie de preparaciones

Carros y Embarcaciones en el laboratorio, se compila el programa de a bordo y se comprueba el funcionamiento de la comunicación con la estación de tierra.

Baterías se conectan las baterías cargadas y se comprueba el funcionamiento de los servos y los motores.

Mecánica a veces hace falta volver a calibrar las brújulas o centrar los timones.

Complementos aparte de los carros y el portátil con la BSD hace falta llevar otras cosas:

catamarán controlado a través de un mando a distancia, se utiliza para rescatar los robots en el caso en el cual se quedan atascados.

red se engancha a los sistemas de tracción que presentan las embarcaciones para los experimentos de recogida de vertidos.

herramientas destornillador, cinta, tijeras, etc para casos de emergencia.

5.1.2. Realización de un experimento

Pasos para la realización de un experimento:

calibración en el caso de los carros se realiza una configuración automática del variador para eliminar la marcha atrás y obtener un rango de velocidades más amplio.

posición global se espera a la estabilización de los receptores GPS. En la BSD se puede observar que al encender los robots y al recibir datos de posición del satélite, transcurre un tiempo largo hasta que estos sean correctos. Conviene esperar, porque de lo contrario, si los robots se ponen en marcha antes de tener una posición correcta, conservan el error durante mucho tiempo.

configuración experimento a través de la BSD se eligen la ruta y la formación deseada, se eligen los robots que van a participar en la formación y se inicia el experimento

supervisión a través de la BSD se pueden observar en tiempo real los distintos parámetros de los robots y se les pueden configurar los sistemas de control.

flasheo en algunas situaciones se detectan pequeños problemas en el programa de a bordo que se corrigen en el momento y se vuelven a programar los robots.

5.1.3. Análisis de un experimento

Después del experimento se recuperan los datos y se analizan:

Base de Supervisión a Distancia se recuperan los datos enviados por los robots y guardados durante el experimento.

Google Earth se representan las trazas de los robots.

Simulador se reproduce el experimento a partir de los datos registrados observando mejor algunos aspectos que durante el experimento pasan desapercibidos.

Matlab se representan los datos de interés de los ficheros registrados. Por ejemplo la evolución de ciertos parámetros a lo largo del experimento.

5.2. Comportamiento individual

El primer objetivo que se debe cumplir es que cada robot individual sea capaz de gobernar su velocidad y rumbo de forma autónoma. Esto implica encontrar valores adecuados de los controladores PID.

Para separar problemas, al principio se asignó a la velocidad un valor fijo (control en lazo abierto), y se centró la experimentación en el control de rumbo, lo que implica seguir trayectorias conocidas, con rectas y con cambios de rumbo.

Si el robot no fuera capaz de mantener un rumbo, sería imposible estudiar experimentalmente el control de velocidad.

La forma básica con la que se especificaron trayectorias fue mediante varios *waypoints*. Se comenzó marcando un cuadrado.

Se realizó un primer ajuste de parámetros del PID de rumbo, quedándonos por el momento con control proporcional y de modo que el transitorio posterior a un giro de 90° fuera breve (desaparece en menos de 2 metros).

Como se verá a continuación, aparecen unos arcos cuando se va de un *waypoint* a otro, tanto en los carros como en los barcos. El hecho de que aparezcan en los carros descarta que puedan deberse a viento o corriente. Nótese que en el caso de los barcos sí cabe esperar algún efecto debido a viento, corriente, y a que no tengamos control directo sobre movimientos laterales. Hay que decir al respecto que tanto los carros como los barcos son no-holónomos.

5.2.1. Robots terrestres

Según lo que se ha dicho antes, los primeros ensayos experimentales se han realizado utilizando los robots terrestres, con el objetivo de establecer el control de rumbo.

Los estudios experimentales se han hecho en la pista de atletismo ya referida en el

Capítulo 3. Se trata de una pista cercana a nuestra Facultad. La longitud total de la pista periférica es de 400 metros. La pista está en mal estado. Hay matas sin cortar en todas partes, de modo que los robots no pueden dar una vuelta completa sin chocar con las matas. La pista periférica tiene bordillos poco definidos, con piedras. En la parte central aparece una zona lisa, arenosa, con varios objetos abandonados, baches, alguna mata, etc. En primavera las matas cubren casi completamente la parte central.

Hemos podido comprobar que los edificios cercanos, los árboles, y la climatología afectan al GPS.

A continuación presentamos los resultados de una serie de experimentos para comprobar el control de rumbo y la posibilidad de seguir rutas marcadas por *waypoints*. Mediante *waypoints* marcamos un cuadrado en la zona central de la pista.

La figura 5.3 presenta los primeros resultados para la ruta cuadrada. Se dibujan mediante símbolos las posiciones de los *waypoints*, y se dibuja la traza GPS del movimiento del carro. Esta traza se ha creado a partir de ficheros con formato KML (*Keyhole Markup Language*) con las medidas GPS guardadas por la estación de tierra.

Como se ve en la figura, tenemos éxito en cuanto al control de rumbo y el seguimiento de *waypoints*.

En cuanto al uso de *waypoints*, se debe especificar un radio adecuado para reconocer que un robot ha alcanzado un *waypoint* (es decir, ha entrado en una región circular alrededor del *waypoint*). Si el radio fuese demasiado pequeño, cabría esperar que el robot diera vueltas alrededor del *waypoint* hasta conseguir entrar en el círculo. Una vez alcanzado un *waypoint*, el robot debe ir al siguiente. Se recuerda que al robot se le pasa una lista de *waypoints*.

Según se ve en la figura, el robot comienza a moverse en una posición cercana a <1> (marcamos los *waypoints* mediante <>) y vuelve hacia atrás dando la vuelta a <1>, después va de <1> a <2> llegando a <2> sin problemas; el tramo <2> a <3> también sin problemas; en el tramo <3> a <4> se observa algo relativamente extraño, y es que al entrar en el círculo de <4> se detecta la entrada antes de dar una vuelta completa y entonces se decide ir a <1>;



Figura 5.3: *Traza de un robot terrestre durante la ruta cuadrada.*

al llegar a <1> da una vuelta completa y seguiría hacia <2>; etc.

Este recorrido es sistemático, habiéndose comprobado en vueltas completas sucesivas al cuadrado.

Además de la ruta cuadrada se ha especificado otra ruta que correspondería a una exploración en barrido de un área, como por ejemplo si se estuviera intentando limpiar un campo de minas.

La figura 5.4 presenta el resultado del experimento. De nuevo hay éxito en cuanto a seguir una ruta relativamente complicada.

La figura representa la traza GPS correspondiente a dos vueltas completas. Es notable observar lo parecidas que son las trazas de una vuelta y de la siguiente.

Es interesante observar en la figura que el comportamiento del robot cerca de <8> es muy parecido al que teníamos en el experimento de la ruta cuadrada cerca de <4>. Igualmente, en los dos experimentos se produce la vuelta completa alrededor de <1>.



Figura 5.4: *Traza de un robot terrestre durante la realización de una ruta en forma de barrido marcado con ocho waypoints.*

5.2.2. Robots marinos

Tras comprobar con los carros que el control de rumbo y la utilización de *waypoints* funcionan bien, se decidió probar los mismos algoritmos con nuestros robots marinos de superficie.

El lugar experimental está situado en el Parque Tierno Galván. Allí existe un conjunto de estanques interconectados a diversos niveles. La siguiente figura muestra los 3 estanques finales de la cadena.

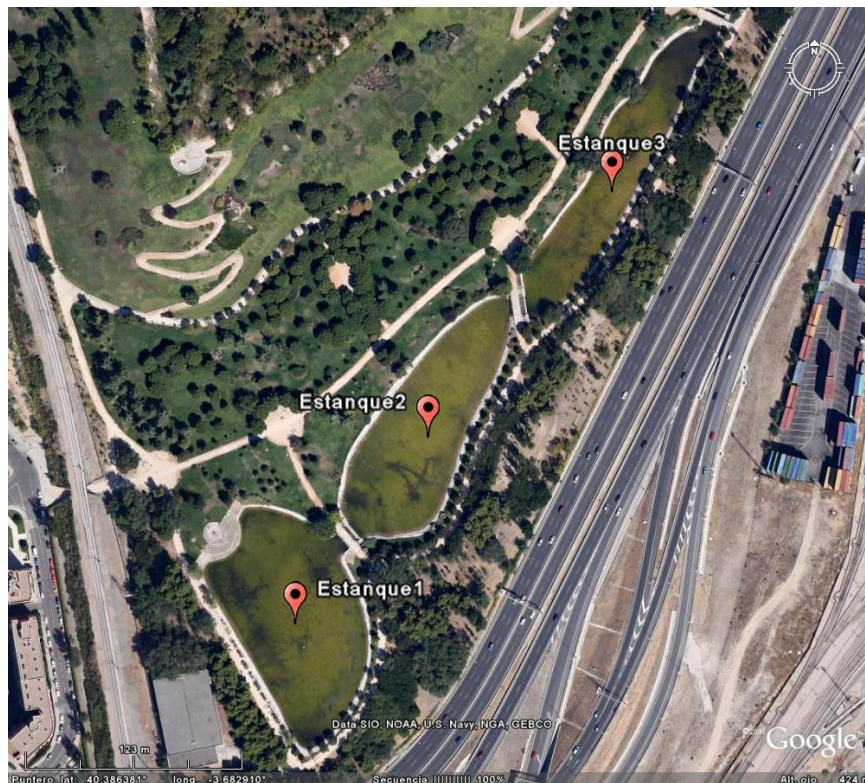


Figura 5.5: *Estanques*

Debido a la diferencia de niveles entre estanques, existe una corriente que va en la secuencia estanque 3, estanque 2, estanque 1. Los primeros experimentos los hemos realizado en el estanque 1, que es el más parecido a un cuadrado.

El primer experimento ha seguido la misma pauta del primero con carros. Se ha marcado un cuadrado con *waypoints*, y el barco debe ir de *waypoint* en *waypoint*. La 5.6 muestra la

traza GPS del movimiento obtenido en el experimento.

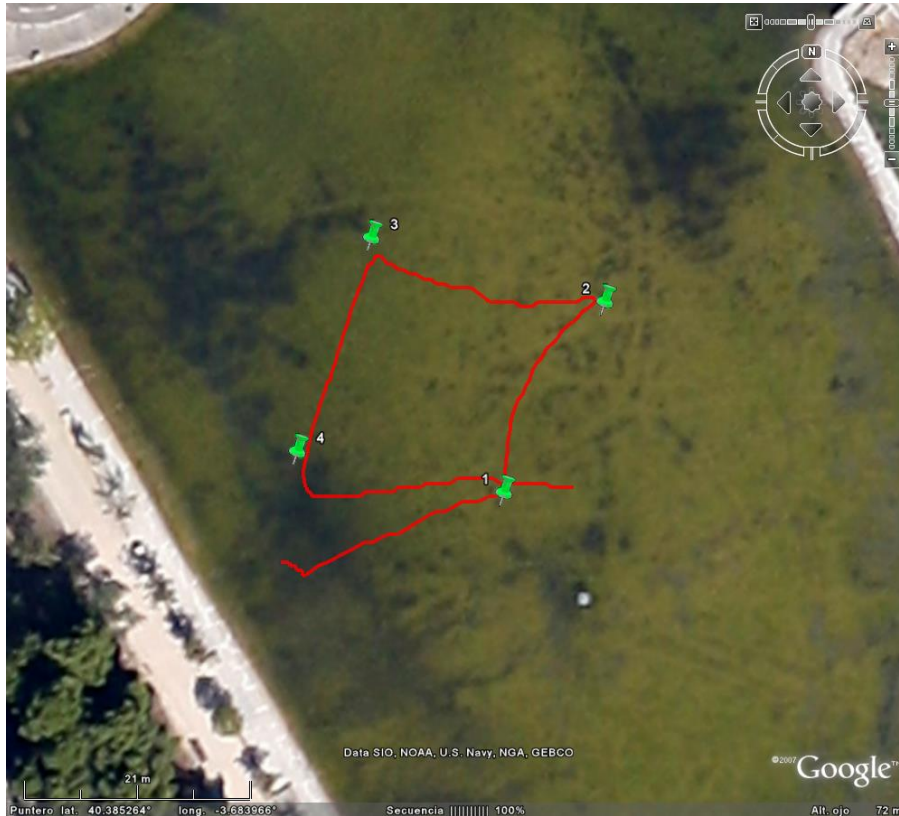


Figura 5.6: *Traza del barco rojo realizando una ruta con 4 waypoints*

En primer lugar hay que resaltar que el experimento tiene buen resultado, confirmando además el buen acuerdo entre experimentos con carros y experimentos con barcos.

Es importante destacar que existen diferencias entre carros y barcos. En particular, el barco se ve influenciado por viento y corriente. Además, hemos observado oscilaciones (transitorio) de rumbo más acusadas, en los giros de 90° , en los barcos. Este efecto se hace notar porque en los carros no hay movimiento lateral, y en los barcos sí aunque atenuados por el rozamiento del agua.

5.2.3. Inserción en la pista

Tras cumplir con un primer objetivo, relativo al control de rumbo (mientras se mantiene la velocidad fija), hay que atacar también el control de velocidad. Esto es necesario si se quiere pasar a investigar la realización de convoys automáticos.

Con las rutas cuadradas definidas en los anteriores experimentos, las rectas son relativamente cortas y no permiten apreciar adecuadamente los transitorios de la velocidad. Por este motivo, nos planteamos utilizar la pista de carreras, que marca la periferia de la pista de atletismo, como ruta a seguir por el carro robot. De este modo, se puede experimentar con rectas más largas.

Al mismo tiempo, esta decisión supone un desafío, porque la garantía de poder permanecer dentro de la pista viene dada por el GPS.

En cuanto a marcar la ruta, esto se hizo mediante bastantes *waypoints* a lo largo de las curvas. En concreto, mediante 7 *waypoints* en cada una de las semicircunferencias.

La figura 5.7 muestra la traza GPS obtenida en uno de los experimentos de recorrer la pista. Se observa la situación de los *waypoints*.

Como muestra la figura, el resultado es plenamente satisfactorio, de modo que el carro robot permanece dentro de la pista, y recorre bien las curvas.

En cuanto al control de velocidad, se introdujo control proporcional y se observó un comportamiento satisfactorio con un transitorio sin oscilaciones.



Figura 5.7: *Experimento pista*

5.3. Convoy automático

La segunda etapa de la investigación se ha centrado en la creación de convoys automáticos. Para ello se ha planteado un control adecuado de la velocidad de los robots para que mantengan entre sí una distancia especificada.

Se ha implementado el sistema de seguimiento de rutas en convoy utilizando nuestra versión del método *Líder-Seguidor*.

Las pruebas experimentales se han realizado utilizando dos carros robots terrestres. Se intentó después hacer este mismo estudio con barcos, pero aparecieron serios problemas en el estanque 1, que estaba lleno de verdina de modo que se atoraron las hélices y hubo que proceder a un rescate bastante complejo; para poder hacer más experimentos hubo que esperar a que pasara el invierno.

5.3.1. Primer experimento

Con el fin de centrar la investigación en aspectos por separado, se ha preferido plantear para los primeros experimentos de convoy automático una ruta simple de forma rectangular situada en la parte central de la pista de atletismo.

Los dos robots han recibido la misma ruta.

En las figuras 5.8 se pueden observar las trazas realizadas durante el experimento a través de Google Earth con la vista por satélite. En la figura 5.9 se representan las mismas trazas mediante Matlab.

Los experimentos se han tomado en vídeo. Esto es conveniente, porque hay aspectos del comportamiento difícilmente apreciables sólo con las trazas. Lo que hemos podido ver es que efectivamente el convoy funciona bien, de modo que los dos robots siguen aceptablemente la ruta marcada y mantienen la distancia especificada (con oscilaciones).



Figura 5.8: *Experimento seguidor (vista satélite)*

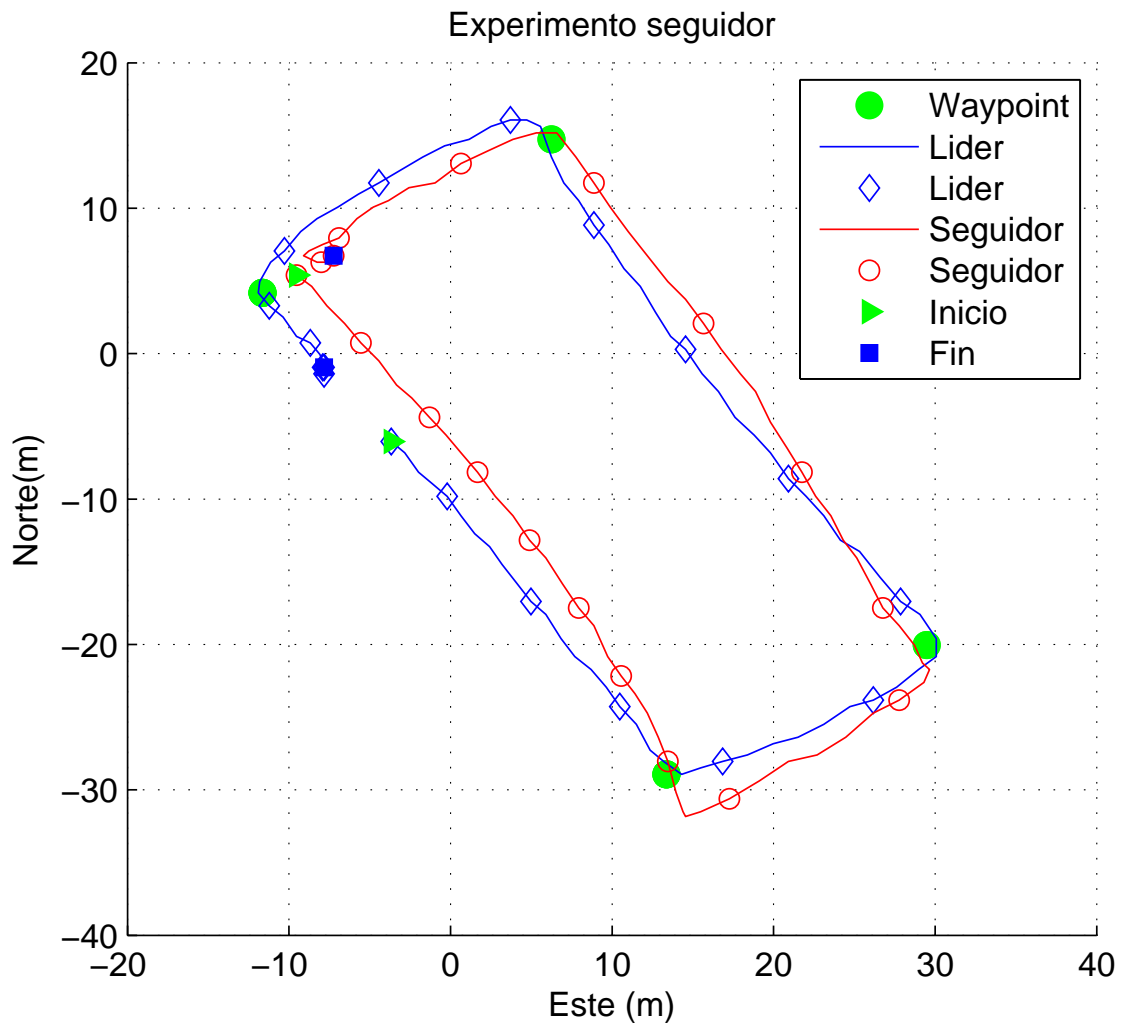


Figura 5.9: *Experimento seguidor (representación en Matlab)*

5.3.2. Segundo experimento

Para el estudio experimental se ha continuado con el uso de la pista de carreras, para poder prestar especial atención a lo que suceda con la distancia entre robots.

En este contexto se han evidenciado problemas y dificultades que no aparecen (lógicamente) en nuestras simulaciones. Por ejemplo, si un carro se acerca demasiado al borde de la pista de carreras, entonces tropieza con el bordillo y una fila de matas que lo paran y entonces hay que hacer una intervención manual para continuar con el experimento.

En la figura 5.10 se han representado las trazas correspondientes a una vuelta a la pista, manteniendo un convoy de dos robots con una velocidad de $1,5m/s$ y una separación de $7m$. Se puede observar que los dos carros al recorrer las rectas describen arcos; sucede entonces (sobretudo en la recta inferior-izquierda de la figura) que se tropieza con el bordillo. De hecho se producen salidas de la pista que o bien frenan al líder o bien al seguidor, generando mayores oscilaciones en la distancia entre los dos, según aparece en la figura 5.11. Esto implica también caídas bruscas y discontinuidades en los gráficos de velocidad de los dos carros, figura 5.12. Se puede ver que en la segunda recta, la más afectada por el arco, que el líder ha llegado a pararse, obligando al seguidor a hacer lo mismo. En el primer y último tramo no afectados por paradas inesperadas del líder, las oscilaciones tienen menos amplitud.

Interesa comentar con más detalle los primeros 90 segundos, que anteceden al incidente se acaba de mencionar. Dentro de este intervalo, los primeros 50 segundos son la recta, y después se entra en la curva. Al fijarse en los primeros 50 segundos en la figura 5.12 se ve con claridad el transitorio de la velocidad del seguidor (cuya referencia es la distancia entre robots); se observan también las oscilaciones de esta distancia en la figura 5.11. La entrada en la curva supone una nueva excitación del transitorio. Actualmente se está pensando en utilizar otro escenario experimental con mejores condiciones, incluyendo alguna recta más larga para poder observar mejor el transitorio.

Es conveniente mencionar un aspecto de la representación e interpretación de los datos. Hemos podido comprobar visualmente que estando los dos carros juntos, sus medidas de

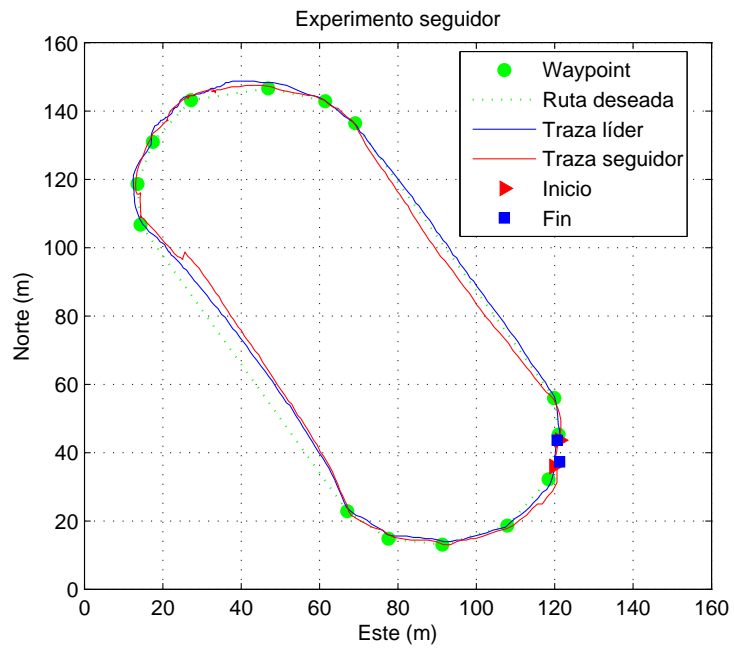


Figura 5.10: Experimento seguidor

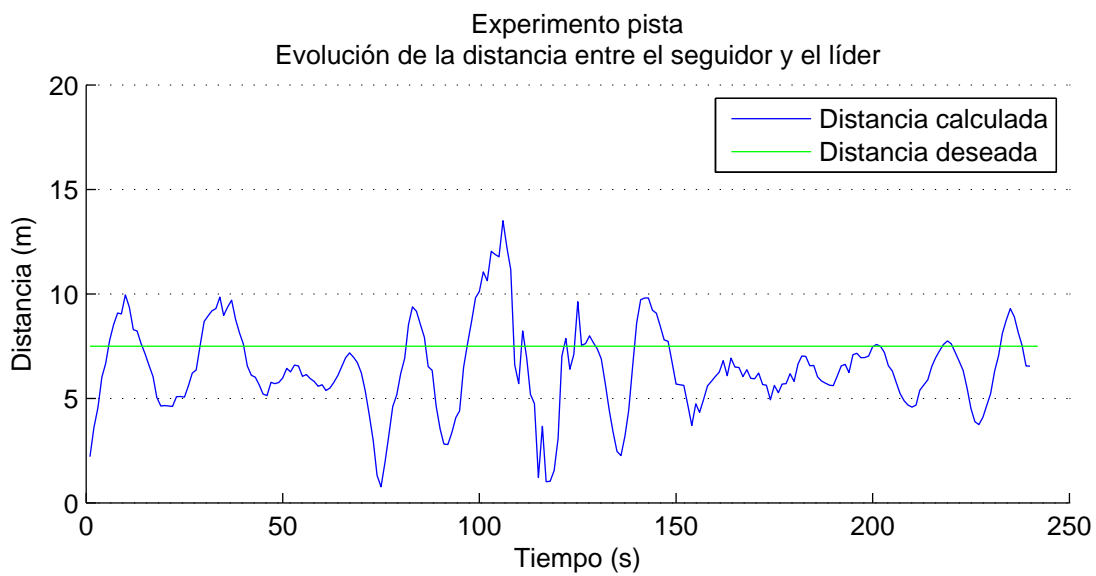


Figura 5.11: *Evolución de la distancia durante una vuelta*

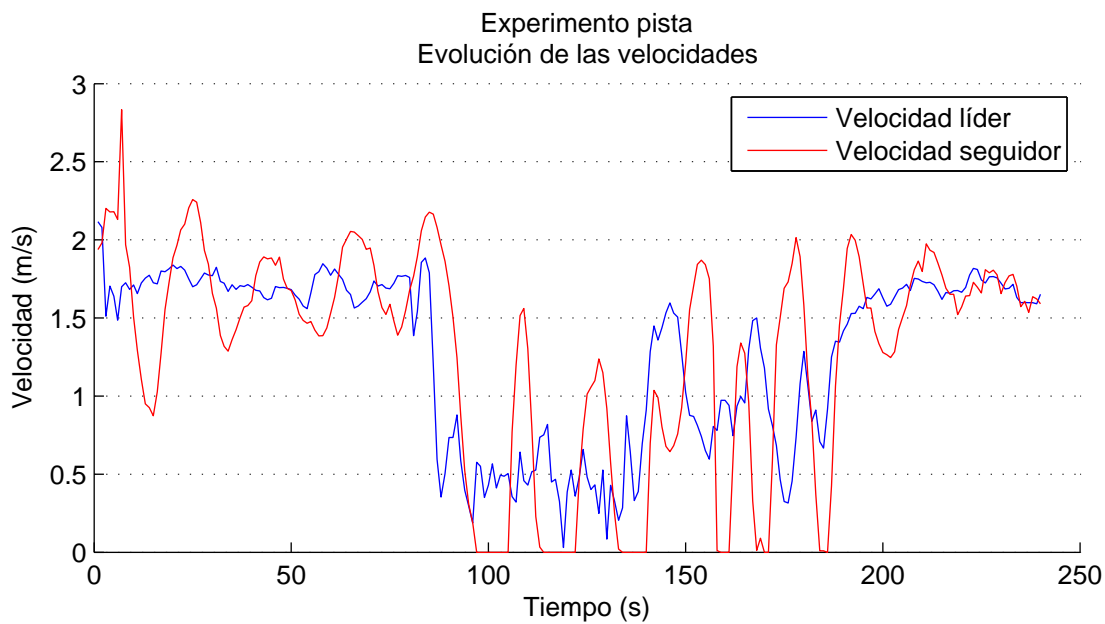


Figura 5.12: *Evolución de las velocidades durante una vuelta*

GPS señalaban distancias mutuas de metros y viceversa, aunque los carros están separados, puede suceder que sus medidas GPS coincidan. Por esto es interesante registrar en vídeo lo que sucede.

También sucede que los GPS utilizan internamente un filtro Kalman, de modo que al estar en marcha los robots, la precisión de las medidas de posición mejora notablemente (se puede apreciar una precisión en el orden del metro).

Se ha podido observar, dentro de las limitaciones que ofrece la pista, que el convoy funciona aceptablemente y a buena velocidad.

5.4. Formaciones

El avance natural de nuestro trabajo es pasar a considerar otras formaciones, y no sólo el convoy (que es una formación en columna).

Si se piensa en una formación en línea, formando un frente con robots avanzando en paralelo, se aprecia que ya no basta con mantener el módulo de una distancia como la única variable; aquí debemos considerar también ángulos entre robots. Si se consideran otras formaciones, con más robots, aparecen cuestiones de geometría (que consisten en distancias y ángulos).

Suponiendo que una formación en línea deba atravesar un pasillo estrecho, entonces se debería pasar a una formación en columna. Aparece así la problemática de manejar geometrías y transiciones entre ellas.

Como durante las transiciones podrían producirse colisiones, tiene sentido incluir algoritmos de navegación como por ejemplo el NSB.

En esta tercera etapa del trabajo, se comienza a tratar de formaciones. Se introduce NSB y una liebre virtual. Dicha liebre virtual lleva asociada una formación virtual a la que tiende la formación real.

También en esta fase del trabajo se ha introducido un cambio importante en la forma de describir las rutas mediante *waypoints*. La pista de carreras consiste en dos rectas y dos

semicircunferencias, lo cual se corresponde directamente con las curvas de Dubins. Estas curvas forman parte importante de la tradición establecida en robótica, y consisten en combinaciones de rectas y arcos de circunferencia. Usando este concepto, la descripción de la pista de carreras se simplifica radicalmente: basta con marcar los centros de las dos semicircunferencias e indicar su radio. Esta es la información que se le pasa a los robots, junto con el sentido de la marcha (a favor o en contra del reloj). De esta manera se evita tener que utilizar 14 *waypoints*.

Si las rutas fueran más complejas no se descartaría considerar “splines” u otras curvas paramétricas.

5.4.1. Primer experimento

Para el primer experimento se vuelve a utilizar la pista de carreras y un carro robot.

La ruta se ha descrito proporcionando la latitud y longitud de los dos centros de giro, el radio de giro de $31m$ y la dirección de giro como izquierda.

En el robot se incorporan los nuevos niveles software con el NSB y la liebre virtual. A la liebre virtual se le especifica una velocidad constante, tal que pueda ser seguida por el carro robot. Así pues, el experimento equivale a un convoy virtual: una liebre seguida por un carro. El NSB se encarga de dar al robot las consignas de rumbo y velocidad.

En la figura 5.13 se representa la traza del robot recorriendo la pista con este nuevo sistema. Se puede observar una mejora considerable con respecto a los experimentos anteriores en la pista.

El desvío de la ruta en la curva inferior (un pequeño bulto) fue causado por un “salto” en las medidas del receptor GPS. Este problema está mencionado también en⁶¹ que denomina este fenómeno como *GPS pop*. Esta situación suele aparecer cuando se pierde la comunicación con algún satélite.

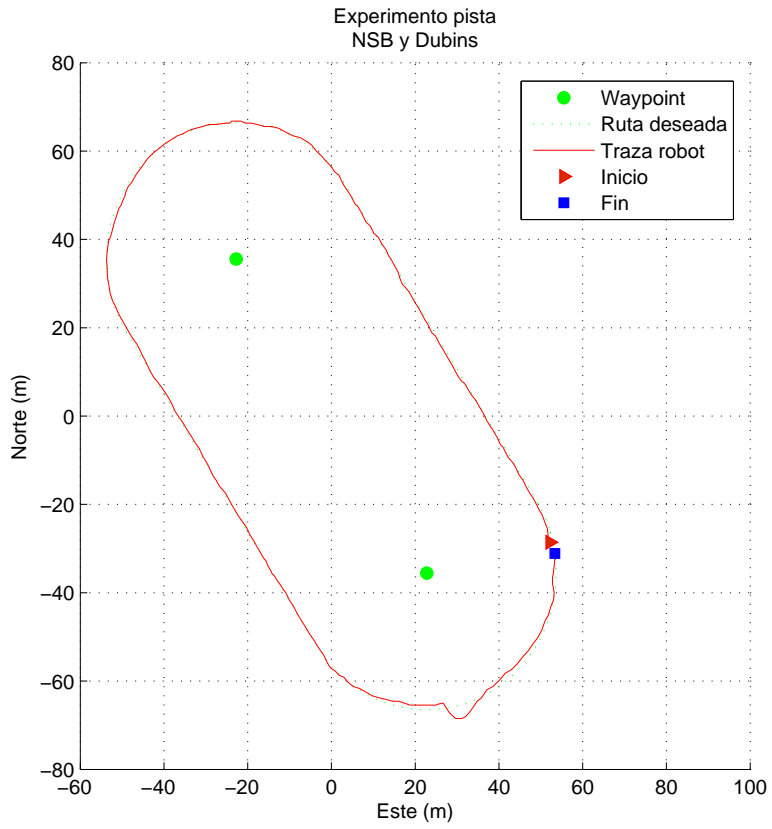


Figura 5.13: Traza durante el experimento con trayectorias de Dubins y control NSB.

5.4.2. Segundo experimento

Una vez comprobado el nuevo sistema de navegación y control en tierra se han realizado experimentos acuáticos equivalentes.

Hay que tener en cuenta que, a diferencia de los carros, los barcos sufren bastante rozamiento al avanzar sobre el agua. Esto implica que los transitorios en el control de velocidad se atenúan más rápidamente, de modo que se puede decir que en cuanto al control de velocidad el comportamiento de los barcos es mejor, y esto es lo que de hecho hemos podido comprobar visualmente en los experimentos.

Puesto que la verdina en el estanque 1 no permite hacer experimentos, hemos pasado a utilizar el estanque 2. En este estanque la corriente (desde el estanque 3 al estanque 1) es apreciable.

La forma del estanque 2 nos permite insertar una ruta parecida a la de la pista de carreras. Para aprovechar más el estanque, hemos definido un radio menor ($10m$) en la curva más al Norte, y otro radio mayor ($15m$) en la curva más al Sur. En la figura 5.16 se puede observar la traza GPS que hemos obtenido con uno de los barcos durante dos vueltas. Esta traza se ajusta bastante bien a la ruta deseada, y permite también apreciar cómo es esta ruta (con los dos radios diferentes).

Es probable que las divergencias entre dos pasadas sucesivas que aparecen en la curva al Sur se deban a cambios en la corriente o en el viento.

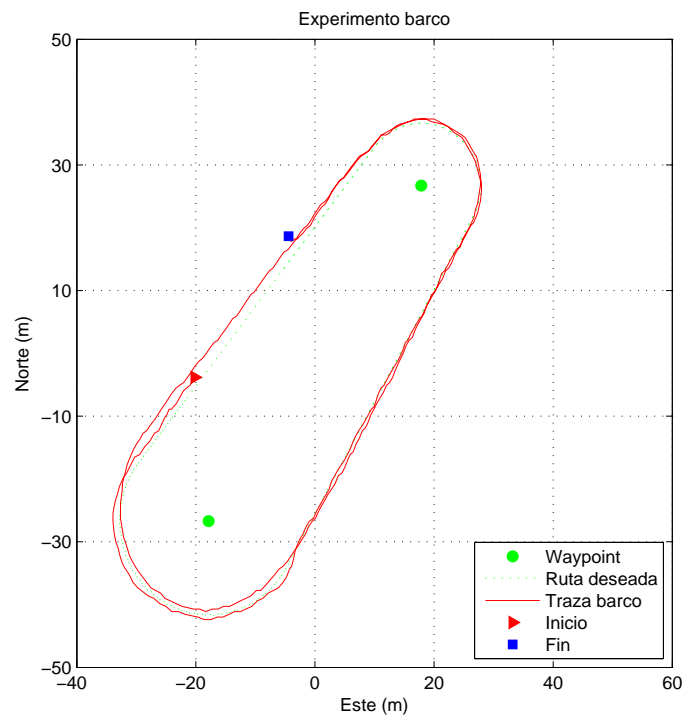


Figura 5.14: Traza durante el experimento con trayectorias de Dubins y control NSB

5.4.3. Tercer experimento

Contando con la base del buen comportamiento observado en los dos experimentos que se acaban de describir, se ha procedido a explorar el uso de otras formaciones e, incluso, intentar un primer ejemplo de transiciones entre formaciones. Para empezar del modo más sencillo, se han utilizado sólo dos barcos.

Sucede que el estanque tiene una fuente en el centro (se puede apreciar con cierta dificultad en la figura 5.16 y también en la figura 5.15). Se va a tomar como referencia este obstáculo. La idea del experimento es pasar de formación en columna a formación en línea unos metros antes de la fuente. Con la formación en línea, los dos barcos avanzan a izquierda y derecha respectivamente de la fuente, de modo que se evita este obstáculo. Una vez superada la fuente, se realiza otra transición para volver a la formación en columna.

En este caso, por sencillez, se ha especificado una pista con radios de giro iguales al Norte y al Sur. Los centros de giro de la pista están alineados dejando la fuente a un lado. Los puntos de activación y la fuente están en la misma recta.

En la figura 5.15 se representan los centros de giro con marcadores verdes y los puntos de activación de las transiciones con marcadores amarillos. Los radios de giro son de $7,5m$.

El estado inicial de este experimento es una formación columna con una separación de $7m$. La primera transición automática lleva a una formación en línea con una separación de $15m$ y la segunda vuelve a una formación columna con $7m$ de separación. La zona de activación de las transiciones está comprendida en un radio de $3m$ entorno a los puntos de activación. Las etapas de transición están configuradas para realizarse a lo largo de $7m$ desde la activación.

En la figura 5.15 aparecen los resultados del experimento. La figura se ha obtenido a partir de los datos GPS y de la brújula, y se han representando con MATLAB. Los triángulos representan a los robots; estos triángulos están orientados según los datos experimentales (brújula). Los circulitos indican la posición deseada en cada momento. La trayectoria a trazos indica la ruta deseada, y la trayectoria continua marca el resultado experimental.

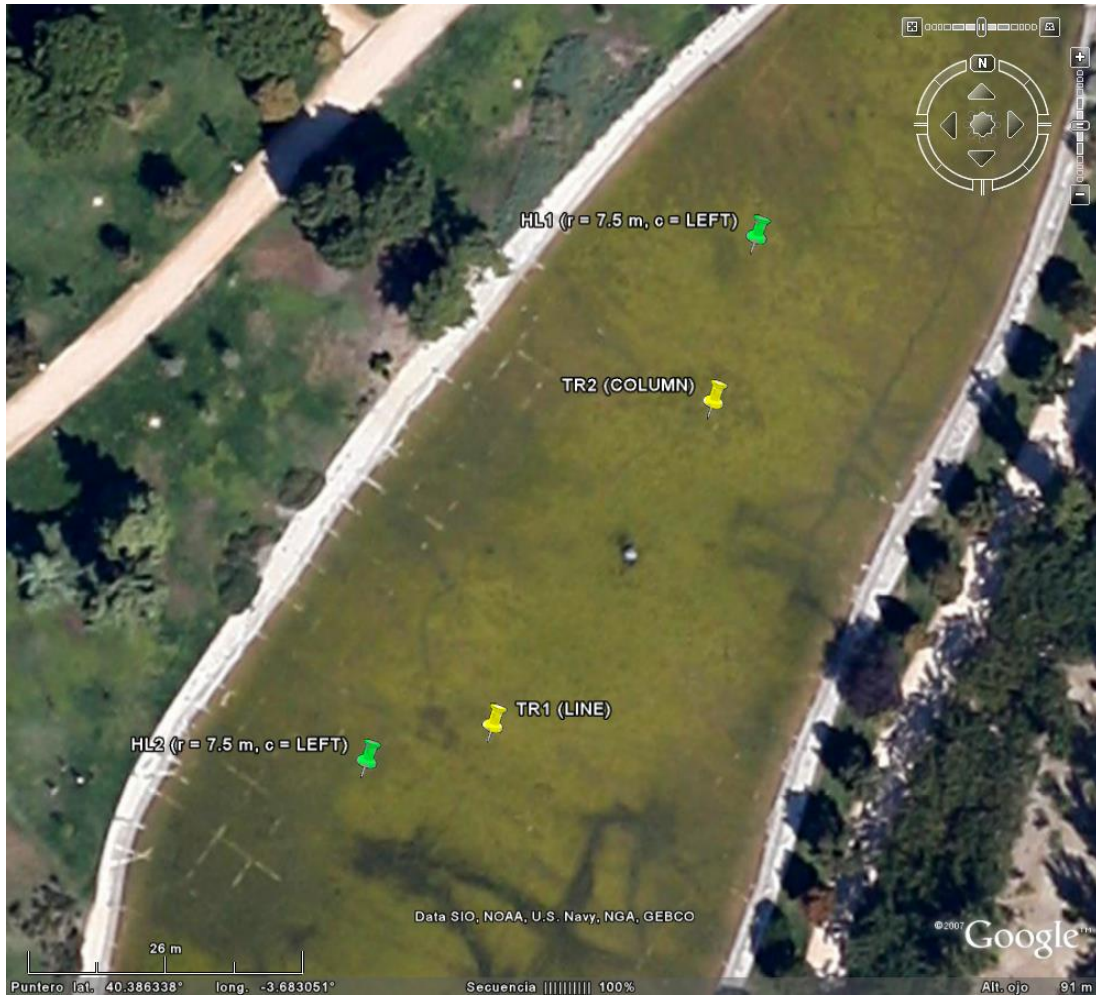


Figura 5.15: Configuración de la ruta para el ensayo con transiciones automáticas. Los marcadores verdes indican los centros de giro y los amarillos los puntos de activación automática de las transiciones.

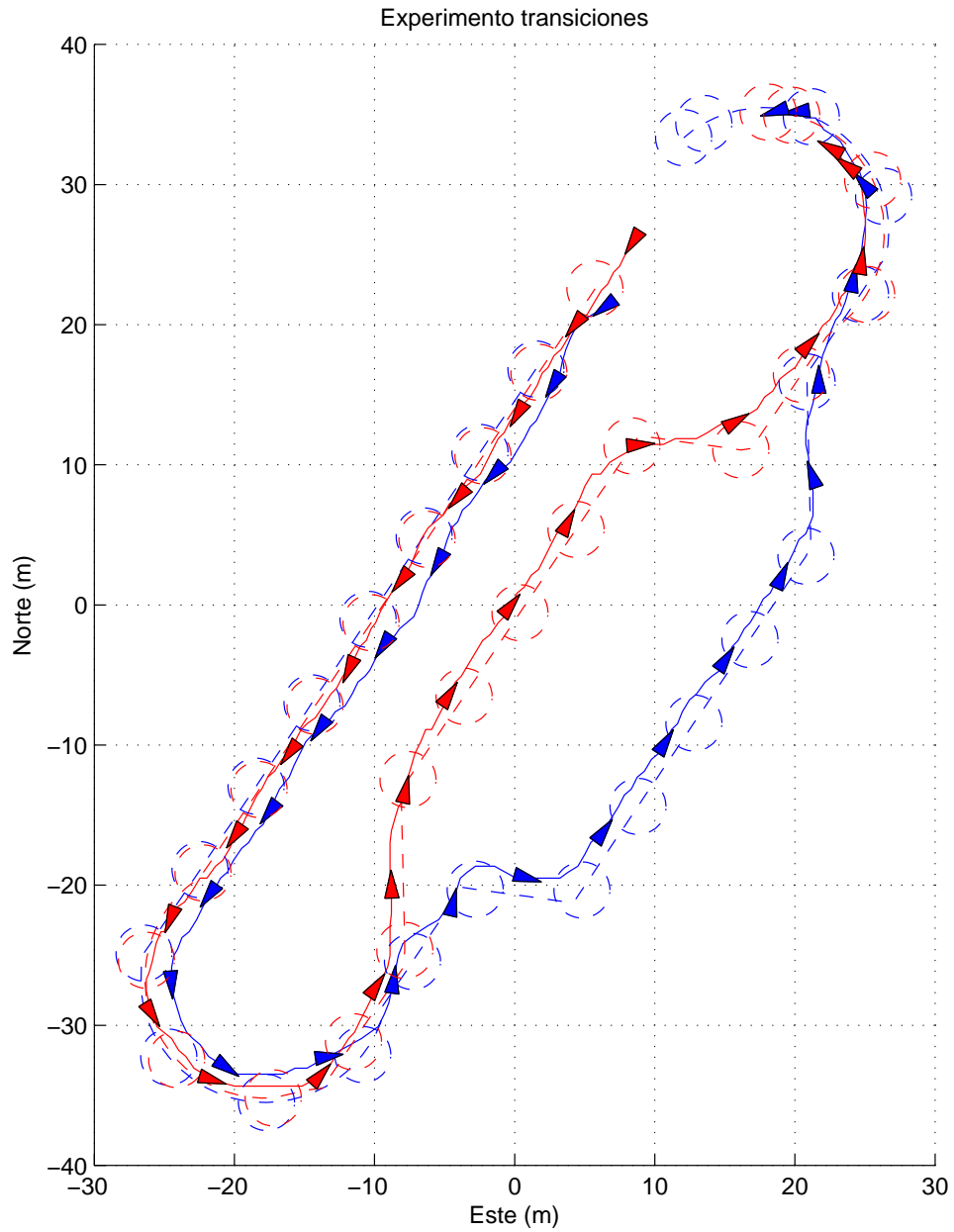


Figura 5.16: *Trazas durante el experimento de transiciones automáticas (columna - línea - columna) con los robots marinos. Con línea continua están representadas las trazas y con triángulos los robots. Las trayectorias deseadas están marcadas con línea discontinua y las posiciones deseadas con círculos.*

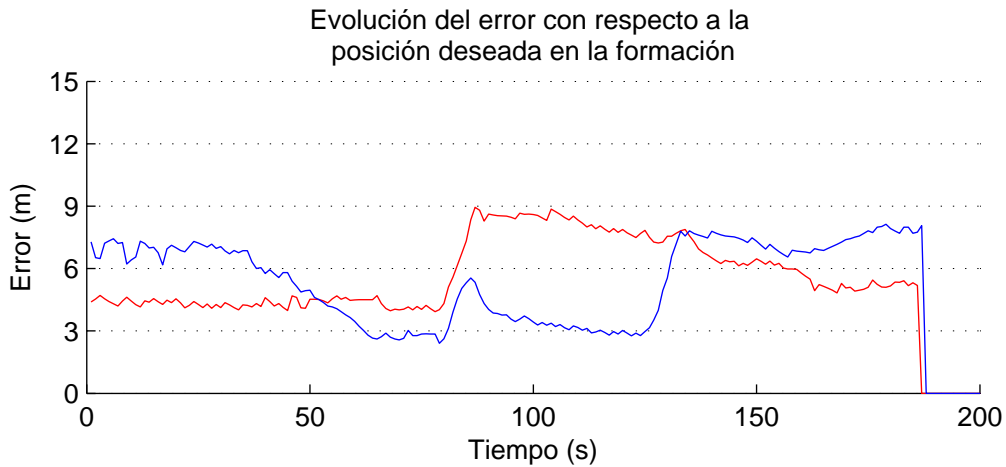


Figura 5.17: Errores de posición durante el experimento con dos transiciones automáticas

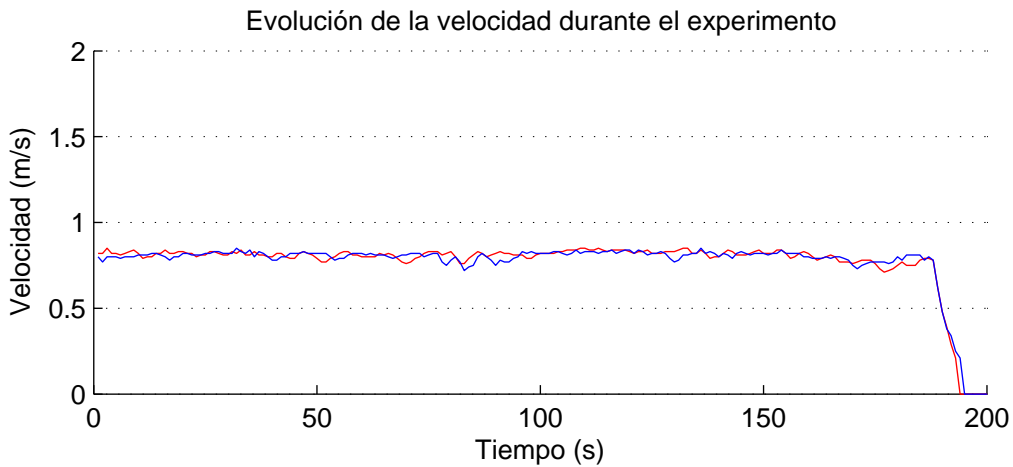


Figura 5.18: Velocidades de los barcos durante el experimento con dos transiciones automáticas

Otra forma de representar los errores de posición es dibujar el módulo del error de posición a lo largo del tiempo. Esto es lo que se representa en la figura 5.17. Los cambios bruscos en este error se corresponden a las transiciones.

La figura 5.18 muestra la evolución de la velocidad durante el experimento. Se observa una respuesta muy débil con respecto a los errores de posición.

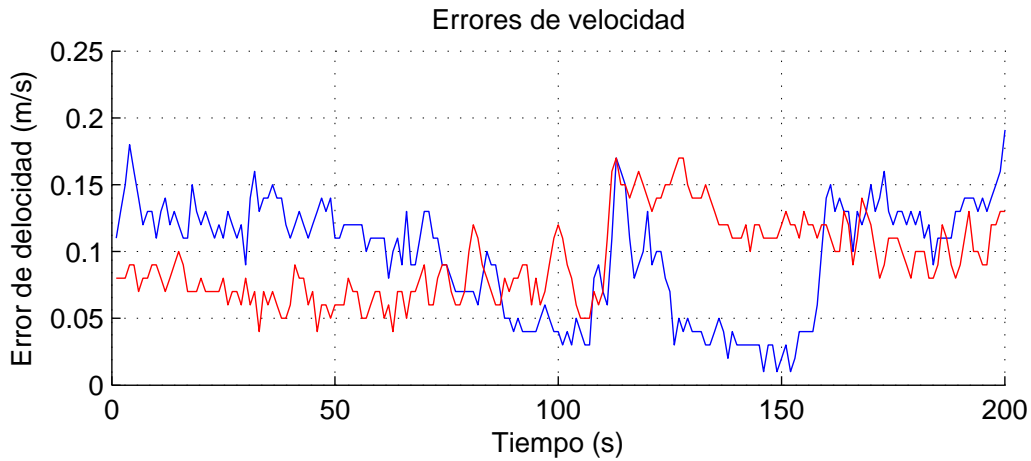


Figura 5.19: Errores de velocidad de los barcos durante el experimento con dos transiciones automáticas

En las figuras 5.19 y 5.20 se representan los errores de velocidad y rumbo como las diferencias entre las consignas ofrecidas por el sistema de control y las medidas tomadas por los sensores.

En cuanto a los errores de velocidad se observa que son muy pequeños en comparación con el error de posición. Esto indica que las consignas de velocidad no son lo suficientemente adecuadas, debido a las ganancias del controlador *Null-Space Based* y justifica por qué no aumentaba lo suficiente la velocidad del robot.

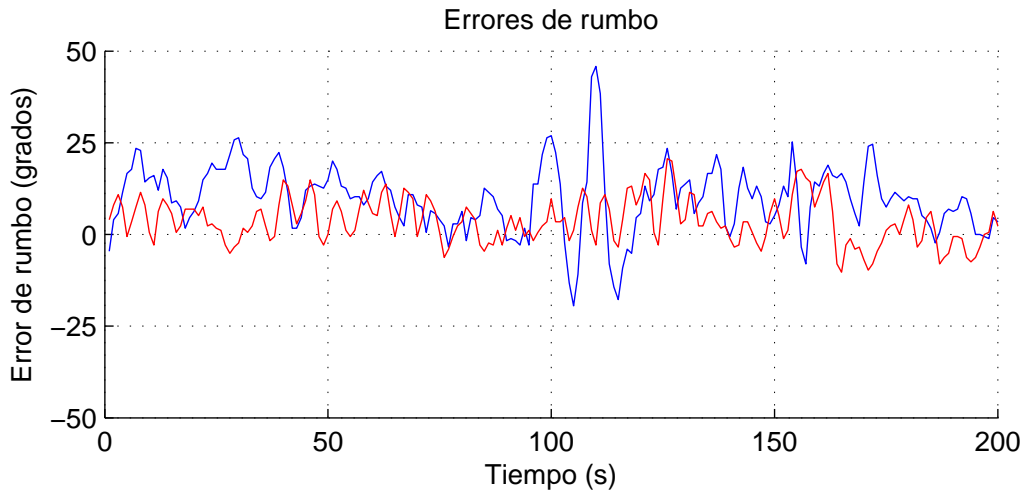


Figura 5.20: Errores de rumbo de los barcos durante el experimento con dos transiciones automáticas

Los errores de rumbo indican un transitorio pronunciado (cerca de la mitad del gráfico) que corresponde al efecto de un golpe de timón para cambiar de formación. Se aprecia una rápida recuperación tanto en esta figura, como en la figura 5.16.

En conjunto el experimento es especialmente satisfactorio. Se ha tomado vídeo de todo el experimento.

Con el fin de dar una información más completa, se presentan unas imágenes tomadas en distintos momentos del vídeo, que corresponden a la transición de línea a columna. Figura 5.21.

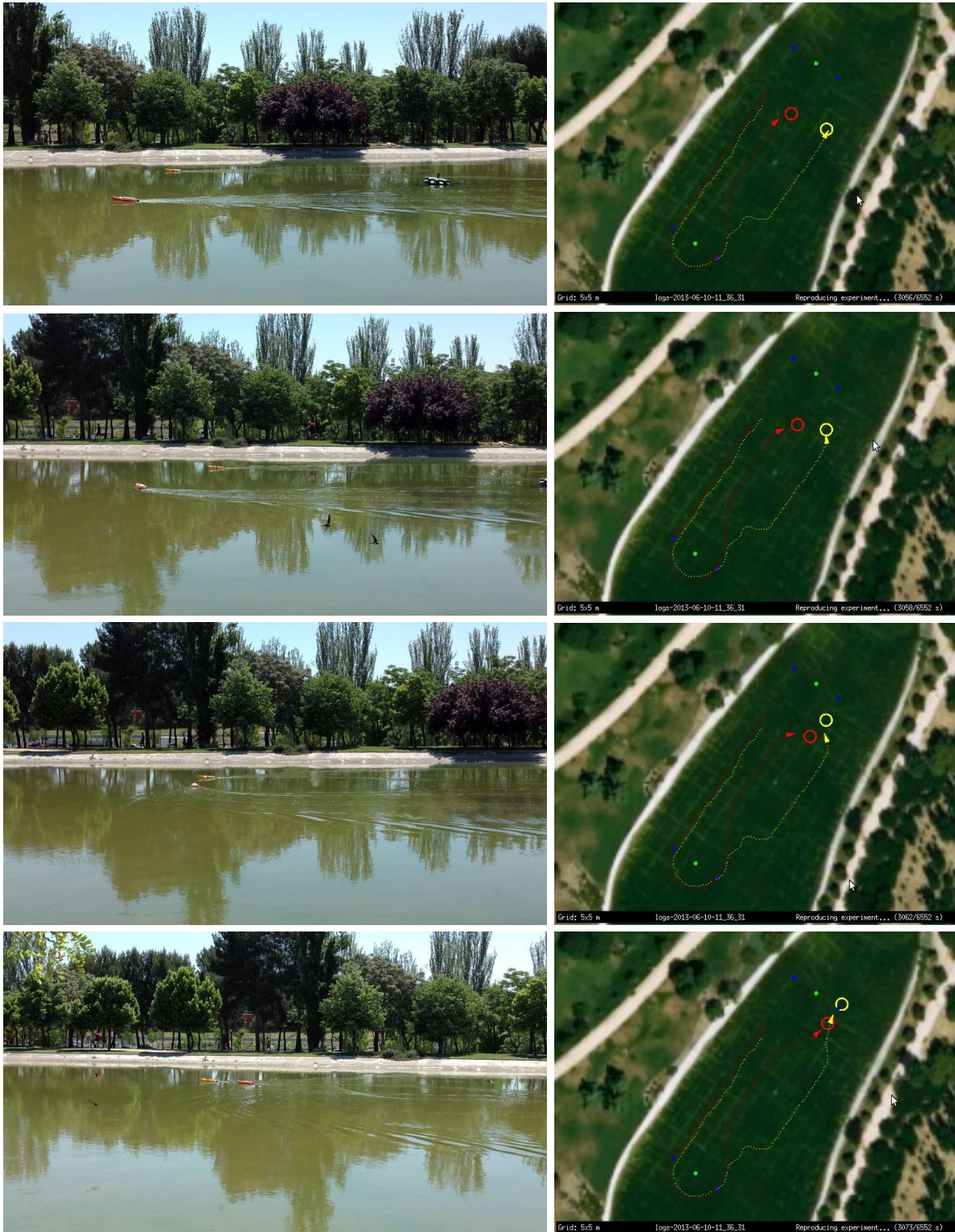


Figura 5.21: Capturas de vídeo y de la reproducción del experimento

Capítulo 6

Conclusiones y trabajos futuros

A lo largo del trabajo presentado en esta Memoria se ha creado una plataforma experimental que permite la prueba de métodos de control para conseguir ciertos comportamientos de grupos de robots, como por ejemplo ir en formación. Se trata de una versión de plataforma con suficiente potencial para el momento actual de nuestra investigación; por supuesto se piensa ya en ciertas ampliaciones interesantes.

También a lo largo de este trabajo se han realizado primeros estudios de métodos de control orientados a formaciones de robots móviles. Estos estudios se han podido hacer tanto en simulación como experimentalmente.

Ya que la plataforma experimental debe ser una herramienta para pruebas, se ha planteado modularmente de modo que se simplifique su uso para diversas aplicaciones, según los intereses de la investigación de que se trate. Por ejemplo, el control de los robots móviles se hace con una “caja universal”, que se puede utilizar en cualquier otra plataforma móvil. A su vez, el software que se ha creado está en C99, y está estructurado con claridad en módulos.

Los estudios experimentales se han realizado a una escala adecuada para simplificar las operaciones a campo abierto. Esto tiene especial sentido porque la investigación se centra en maniobras (p.ej., no se trata de estabilización de barcos). Pensamos que no es complicado pasar a otras escalas, puesto que se basaría en las mismas “cajas univesales”.

Los resultados experimentales han sido bastante satisfactorios, según se describe en el Capítulo 5. En adelante, nos gustaría encontrar entornos con mejores condiciones y más

grandes en algunos casos.

Las perspectivas de posterior investigación son amplias. Debemos profundizar en experimentos ya realizados, y debemos también extender el tipo de casos a estudiar, incluyendo nuevas tecnologías.

Este trabajo ha continuado una línea abierta hace algunos años. Es conveniente indicar cuales son mis aportaciones y qué parte he recibido del trabajo anterior. Por este motivo, presento algunos gráficos que ayudan a visualizar el engarce entre las etapas anteriores y la mía.

Los diagramas son meramente conceptuales, de modo que no reflejan en tamaño la cantidad de tiempo que ha requerido el trabajo anterior. Por ejemplo, el mero desarrollo de las placas de circuito impreso con los controladores, supuso mucho tiempo de desarrollo y pruebas. Por tanto, no se trata de minusvalorar lo que han hecho los que me han precedido.

En particular, he de agradecer el esfuerzo de Fernando Pereda y de Héctor García de Marina para hacerme entender con rapidez y eficacia todo lo que habían hecho.

A continuación presento los diagramas que resumen la labor que he efectuado (bloques violeta), engarzada con la parte final de los trabajos anteriores (bloques verdes). En la figura 6.1 se representan los temas abordados durante el trabajo. En la figura 6.2 se marcan las principales aportaciones en cuanto al sistema experimental. La figura 6.3 muestra los estudios experimentales realizados.

Temas Abordados						
Control Individual y Waypoints	Versión Básica			Versión Mejorada, Curvas de Dubins		
Convoys	Versión Elemental	Nuevo Planteamiento				
Formaciones	Primeras Ideas					

Figura 6.1: Temas abordados

Sistema Experimental						
Carros robot	Primer carro	3 carros con "caja universal"				
Barcos robot	2 barcos		Mejoras y más accesorios			
Base de tierra	Primera Versión	Versión ampliada Para formaciones				
Simulador	Simulador básico	Simulador más realista, con interfaz experimental				

Figura 6.2: Sistema experimental

Estudios Experimentales						
Terrestres	Rutas simples	1 robot, zona central	1 robot, pista	2 robots, convoy, Dubins, Null-Space		
En el agua	Rutas simples	Seguidor elemental	1 barco, Dubins, Null-space	2 barcos, formaciones y transiciones		

Figura 6.3: Estudios experimentales

En resumen, las aportaciones de esta Memoria son las siguientes:

- Ampliaciones del programa de a bordo para el control de carros y de barcos, de modo que puedan ir en formación.
- Una estación de tierra con capacidad de interactuar en experimentos con formaciones.
- Un simulador para la validación y depuración de los algoritmos.
- Una especificación de trayectorias basada en curvas de Dubins.
- Un primer método de control para convoys automáticos, con una modificación del “follow the leader”.
- Un segundo método de control para formaciones basado en “Null-space” y que engloba un “virtual-target”. Se ha aplicado tanto en convoys como en la formación en línea.
- Una ampliación de la infraestructura experimental, aumentando el número de robots para los experimentos.
- Estudios experimentales para comprobar los algoritmos, tanto para robots individuales, como para convoys y formaciones en línea, y para transiciones entre formaciones.

Bibliografía

- [1] Sartre project website. <http://www.sartre-project.eu>, 2013.
- [2] Noa Agmon, Gal A Kaminka, Sarit Kraus, and Meytal Traub. Task reallocation in multi-robot formations. 2010.
- [3] Gianluca Antonelli, Filippo Arrichiello, and Stefano Chiaverini. The null-space-based behavioral control for soccer-playing mobile robots. In *Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on*, pages 1257–1262. IEEE, 2005.
- [4] Gianluca Antonelli, Filippo Arrichiello, and Stefano Chiaverini. The entrapment/escorting mission. *Robotics & Automation Magazine, IEEE*, 15(1):22–29, 2008.
- [5] Gianluca Antonelli, Filippo Arrichiello, and Stefano Chiaverini. The null-space-based behavioral control for autonomous robotic systems. *Intelligent Service Robotics*, 1(1):27–39, 2008.
- [6] Gianluca Antonelli, Filippo Arrichiello, and Stefano Chiaverini. The nsb control: a behavior-based approach for multi-robot systems. *Paladyn*, 1(1):48–56, 2010.
- [7] Ronald C Arkin. Towards cosmopolitan robots: Intelligent navigation in extended man-made environments. 1987.
- [8] Ronald C Arkin. Motor schema-based mobile robot navigation. *The International journal of robotics research*, 8(4):92–112, 1989.
- [9] Ronald C Arkin. *Behavior-based robotics*. MIT press, 1998.

- [10] F. Arrichiello, S. Chiaverini, and T.I. Fossen. Formation control of underactuated surface vessels using the null-space-based behavioral control. pages 5942–5947, 2006.
- [11] F Arrichiello, J Das, H Heidarsson, S Chiaverini, and GS Sukhatme. Experiments in autonomous navigation with an under-actuated surface vessel via the null-space based behavioral control. In *Advanced Intelligent Mechatronics, 2009. AIM 2009. IEEE/ASME International Conference on*, pages 362–367. IEEE, 2009.
- [12] Filippo Arrichiello, Stefano Chiaverini, and Thor I Fossen. Formation control of marine surface vessels using the null-space-based behavioral control. In *Group coordination and cooperative control*, pages 1–19. Springer, 2006.
- [13] Filippo Arrichiello, Hordur Heidarsson, Stefano Chiaverini, and Gaurav S Sukhatme. Cooperative caging using autonomous aquatic surface vehicles. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4763–4769. IEEE, 2010.
- [14] Erik Billing. Cognitive perspectives on robot behavior. In *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence: Special Session on Computing Languages with Multi-Agent Systems and Bio-Inspired Devices*, pages 373–382. INSTICC, 2010.
- [15] Jean-Daniel Boissonnat, André Cérézo, and Juliette Leblond. *Shortest paths of bounded curvature in the plane*. Springer, 1993.
- [16] Morten Breivik, Vegard E Hovstein, and Thor I Fossen. Ship formation control: A guided leader-follower approach. In *Proc. IFAC World Congress*, 2008.
- [17] Rodney Brooks. A robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of*, 2(1):14–23, 1986.

- [18] Rodney A Brooks. Intelligence without representation. *Artificial intelligence*, 47(1):139–159, 1991.
- [19] Massimo Caccia, Marco Bibuli, Riccardo Bono, and Gabriele Bruzzone. Basic navigation, guidance and control of an unmanned surface vehicle. *Autonomous Robots*, 25(4):349–365, 2008.
- [20] Derek Caveney. Cooperative vehicular safety applications. *Control Systems, IEEE*, 30(4):38–53, 2010.
- [21] Eric Chan, Peter Gilhead, Pavel Jelínek, Petr Krejčí, and Tom Robinson. Cooperative control of satre automated platoon vehicles. 2012.
- [22] Yang Quan Chen and Zhongmin Wang. Formation control: a review and a new consideration. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3181–3186. IEEE, 2005.
- [23] Howie M Choset. *Principles of robot motion: theory, algorithms, and implementation*. Bradford Books, 2005.
- [24] Luca Consolini, Fabio Morbidi, Domenico Prattichizzo, and Mario Tosques. Leader-follower formation control of nonholonomic mobile robots with input constraints. *Automatica*, 44(5):1343–1349, 2008.
- [25] Rongxin Cui, Shuzhi Sam Ge, Bernard Voon Ee How, and Yoo Sang Choo. Leader-follower formation control of underactuated autonomous underwater vehicles. *Ocean Engineering*, 37(17):1491–1502, 2010.
- [26] Jaydev P Desai, James P Ostrowski, and Vijay Kumar. Modeling and control of formations of nonholonomic mobile robots. *Robotics and Automation, IEEE Transactions on*, 17(6):905–908, 2001.

- [27] Lester E Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of mathematics*, 79(3):497–516, 1957.
- [28] Gregory Dudek and Michael Jenkin. *Computational principles of mobile robotics*. Cambridge university press, 2010.
- [29] Yaser P Fallah, ChingLing Huang, Raja Sengupta, and Hariharan Krishnan. Design of cooperative vehicle safety systems based on tight coupling of communication, computing and physical vehicle dynamics. In *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems*, pages 159–167. ACM, 2010.
- [30] Thor I Fossen. *Marine control systems: Guidance, navigation and control of ships, rigs and underwater vehicles*. Marine Cybernetics Trondheim, 2002.
- [31] Emilio Frazzoli, Munther A Dahleh, and Eric Feron. Maneuver-based motion planning for nonlinear systems with symmetries. *Robotics, IEEE Transactions on*, 21(6):1077–1091, 2005.
- [32] D Fryxell, P Oliveira, A Pascoal, C Silvestre, and I Kammer. Navigation, guidance and control of auvs: an application to the marius vehicle. *Control Engineering Practice*, 4(3):401–409, 1996.
- [33] Andrei A Furtuna and Devin J Balkcom. Generalizing dubins curves: Minimum-time sequences of body-fixed rotations and translations in the plane. *The International Journal of Robotics Research*, 29(6):703–726, 2010.
- [34] Brian P Gerkey and Maja J Matarić. Sold!: Auction methods for multirobot coordination. *Robotics and Automation, IEEE Transactions on*, 18(5):758–768, 2002.
- [35] Brian P Gerkey and Maja J Matarić. A formal analysis and taxonomy of task allocation

- in multi-robot systems. *The International Journal of Robotics Research*, 23(9):939–954, 2004.
- [36] Jose M Giron-Sierra, Santiago Cifuentes, and Juan F Jimenez. Genetic trajectories control planning for teams of quasi-robot ships. In *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, pages 1206–1211. IEEE, 2008.
- [37] Michael A Goodrich, Timothy W McLain, Jeffrey D Anderson, Jisang Sun, and Jacob W Crandall. Managing autonomy in robot teams: observations from four experiments. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 25–32. ACM, 2007.
- [38] Michael A Goodrich, Dan R Olsen, Jacob W Crandall, and Thomas J Palmer. Experiments in adjustable autonomy. In *Proceedings of IJCAI Workshop on Autonomy, Delegation and Control: Interacting with Intelligent Agents*, pages 1624–1629, 2001.
- [39] Edwin Hutchins. *Cognition in the Wild*, volume 262082314. MIT press Cambridge, MA, 1995.
- [40] Edwin Hutchins and Barbara Holder. Conceptual models for understanding an encounter with a mountain wave. *HCI-Aero 2000*, 2000.
- [41] Edwin Hutchins and Tove Klausen. Distributed cognition in an airline cockpit. *Cognition and communication at work*, pages 15–34, 1996.
- [42] JF Jimenez, JM Giron-Sierra, A Dominguez, JM De la Cruz, and JM Riola. Ships confining an oil spill over: a scenario for automatized cooperation. In *Oceans 2005-Europe*, volume 2, pages 1226–1231. IEEE, 2005.
- [43] Jong-Hyuk Kim, Salah Sukkarieh, and Stuart Wishart. Real-time navigation, guidance,

- and control of a uav using low-cost sensors. In *Field and Service Robotics*, pages 299–309. Springer, 2006.
- [44] L Lapierre, D Soetanto, and A Pascoal. Coordinated motion control of marine robots. In *Proceedings of the 6th IFAC Conference on Manoeuvring and Control of Marine Craft (MCMC2003)*, 2004.
- [45] Steven Michael LaValle. *Planning algorithms*. Cambridge university press, 2006.
- [46] Steven Michael LaValle. *Planning algorithms*. Cambridge university press, 2006.
- [47] Naomi Ehrich Leonard and Edward Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, volume 3, pages 2968–2973. IEEE, 2001.
- [48] Rob Long, Ross Mead, Jerry B. Weinberg, Mobile Robotics, Lab Interaction, Lab Mobile, and Robotics Lab. Distributed auction-based initialization of mobile robot formations. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*, 1949.
- [49] Edward A Macdonald. Multi-robot assignment and formation control. 2011.
- [50] Alessandro Marino. *A Null-Space-based Behavioral Approach to Multi-Robot Patrolling*. PhD thesis, 2004.
- [51] Maja J Matarić. Integration of representation into goal-driven behavior-based robots. *Robotics and Automation, IEEE Transactions on*, 8(3):304–312, 1992.
- [52] Maja J Matarić. Behaviour-based control: Examples from navigation, learning, and group behaviour. *Journal of Experimental & Theoretical Artificial Intelligence*, 9(2-3):323–336, 1997.

- [53] Mohammed Zeki Al-Faiz Mohammed Majid Al Khalidy, Rami A. Mahir. *Wheeled Mobile Robot: Design and Implementation of Wheeled Mobile Robot Using Intelligent Controllers*. LAP Lambert Academic Publishing, 2012.
- [54] Yogeswaran Mohan and SG Ponnambalam. An extensive review of research in swarm robotics. In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, pages 140–145. IEEE, 2009.
- [55] W Naeem, T Xu, R Sutton, and A Tiano. The design of a navigation, guidance, and control system for an unmanned surface vehicle for environmental monitoring. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 222(2):67–79, 2008.
- [56] D Nakhaeinia, SH Tang, SB Mohd Noor, and O Motlagh. A review of control architectures for autonomous navigation of mobile robots. *Int. J. Phys. Sci*, 6(2):169–174, 2011.
- [57] Lynne E Parker. Alliance: An architecture for fault tolerant multirobot cooperation. *Robotics and Automation, IEEE Transactions on*, 14(2):220–240, 1998.
- [58] Lynne E Parker. Distributed intelligence: Overview of the field and its application in multi-robot systems. *Journal of Physical Agents*, 2(1):5–14, 2008.
- [59] Satish Pedduri and K Madhava Krishna. Collision avoidance for multiple robots till next waypoints through collision free polygons. 2007.
- [60] Fernando J Pereda, Hector Garcia de Marina, Jose M Giron-Sierra, and Juan Jimenez. Towards automatic oil spill confinement with autonomous marine surface vehicles. In *OCEANS, 2011 IEEE-Spain*, pages 1–6. IEEE, 2011.
- [61] Joseph Satoru Putney. *Reactive navigation of an autonomous ground vehicle using dy-*

- namic expanding zones*. PhD thesis, Virginia Polytechnic Institute and State University, 2006.
- [62] S Veera Ragavan, SG Ponnambalam, and Cony Sumero. Waypoint-based path planner for mobile robot navigation using pso and ga-ais. In *Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE*, pages 756–760. IEEE, 2011.
- [63] Alan C Schultz and Lynne E Parker. *Multi-robot systems: from swarms to intelligent automata*, volume 1. Springer, 2002.
- [64] Miguel Sepulcre, Jens Mittag, Paolo Santi, Hannes Hartenstein, and Javier Gozalvez. Congestion and awareness control in cooperative vehicular systems. *Proceedings of the IEEE*, 99(7):1260–1279, 2011.
- [65] Mae L Seto. *Marine Robot Autonomy*. Springer, 2013.
- [66] Roland Siegwart and Illah Reza Nourbakhsh. *Intro to Autonomous Mobile Robots*. MIT press, 2004.
- [67] Roger Skjetne, Sonja Moi, and Thor I Fossen. Nonlinear formation control of marine craft. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 2, pages 1699–1704. IEEE, 2002.
- [68] Lucille Alice Suchman. *Plans and situated actions: the problem of human-machine communication*. Cambridge university press, 1987.
- [69] Lovekesh Vig. *Multiple Robot Teaming: Coalition Formation: From Software Agents to Robots*. VDM Verlag, 2008.
- [70] Iain A. Wallace. *A Hybrid Reactive and Plan-based Agent Architecture for Robotic Soccer*. PhD thesis, The University of York, Department of Computer Science, 2006.