

UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE INFORMÁTICA

MÁSTER EN INVESTIGACIÓN EN INFORMÁTICA



Trabajo Fin de Máster en
Ingeniería Informática para la Industria

Guiado, navegación y control de una flota de barcos autónomos

Fernando José Pereda Garcimartín

Curso académico: 2010-2011

Director:
Dr. Jose María GIRÓN SIERRA

Autorización de difusión

Fernando José Pereda Garcimartín

Junio de 2011

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: “Guiado, navegación y control de una flota de barcos autónomos”, realizado durante el curso académico 2010-2011 bajo la dirección de D. Jose María Girón Sierra en el Departamento de Arquitectura de Computadores 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

Los sistemas que manejan distintos elementos autónomos tienen cada día mayor relevancia. En el caso de barcos autónomos, este tipo de sistemas pueden ser utilizados en tareas como persecución, rescate, vigilancia, limpieza de vertidos, defensa y desminado. Uno de los problemas de los sistemas multi robot es la inicialización y mantenimiento de formaciones rígidas a la vez que el grupo es guiado a lo largo de una trayectoria. Este Trabajo Fin de Máster presenta un sistema capaz de guiar a una flota de barcos robot a lo largo de una trayectoria; la flota mantiene una formación geométrica rígida. Las trayectorias a lo largo de las que se guía la flota se describen utilizando splines generados a partir de curvas de Bézier cúbicas. Para la inicialización y cambios de formación, se utiliza un sistema de subastas y una metáfora económica que permite resolver ambos problemas sin ningún tipo de casuística especial. Se hace un estudio en simulación, utilizando simulaciones realistas, del sistema completo. Adicionalmente, el sistema de navegación, basado en *Unscented Kalman Filter* (UKF), es validado de forma experimental. Todos resultados son satisfactorios y muestran la viabilidad del sistema propuesto.

Palabras clave: Barco autónomo, GNC, formación, UKF, NSB, subasta

Abstract

The relevance of autonomous systems that handle several entities is raising every day. In the case of autonomous vessels, this kind of systems can be used in tasks such as target tracking, security, safety, rescue, surveillance, spill confinement, defense and demining. One of the main problems of multi-robot systems is the initialization and maintenance of a rigid formation while the fleet is guided through a geometric path. This Master Thesis proposes a system able to guide a fleet of autonomous vessels along a geometric path; the fleet maintains a rigid formation. The trajectories are described using splines built from cubic Bézier curves. Both the initialization and the change of geometric formations are handled by an auction system and an economic metaphor; this methodology solves both problems without any special case handling. The system is studied through simulations, using realistic simulations of the experimental platform. Additionally, the navigation system, based upon *Unscented Kalman Filter* (UKF), is assessed through field experiments. The results are satisfactory and show the viability of the proposed system.

Keywords: Autonomous vessel, GNC, formation, UKF, NSB, auction

Índice general

1. Introducción	1
1.1. Motivación y objetivos	3
1.2. Estado del arte	5
1.3. Estructura del Trabajo Fin de Máster	6
2. Descripción general del sistema	9
2.1. Visión general	11
2.2. Alcance del sistema	14
2.3. Requisitos	15
3. Modelado matemático	17
3.1. Sistemas de referencia	19
3.2. Modelo de maniobra del barco	21
3.2.1. Modelo cinemático	21
3.2.2. Modelo dinámico	22
3.2.3. Barco experimental	23
4. El problema de la asignación y el algoritmo de subastas	27
4.1. El problema de la asignación	29
4.2. El algoritmo de subasta	29
4.2.1. Algoritmo	30
4.2.2. Coste computacional	31
4.2.3. Ejemplo del algoritmo	31
5. El filtro de Kalman <i>Unscented</i>	37
5.1. La transformada <i>Unscented</i>	39
5.2. UKF para procesos con ruido aditivo	40
5.2.1. Paso de predicción	40
5.2.2. Paso de corrección	41
6. Implementación del sistema	43
6.1. Navegación	45
6.1.1. Filtrado	45
6.1.2. Estimador	45
6.2. Guiado	48
6.2.1. Splines para trayectorias	48
6.2.2. Control jerárquico <i>Null-Space-Based</i> (NSB)	52
6.2.3. Subastas para formaciones	58
6.3. Control	60
6.3.1. Control de rumbo	61

6.3.2. Control de velocidad	66
7. Resultados en simulación	69
7.1. El simulador	71
7.1.1. Análisis y requisitos	71
7.1.2. Diseño e implementación	74
7.2. Estudio de escenarios simulados	78
7.2.1. Caso simple, dos barcos en paralelo	79
7.2.2. Efecto de las <i>resubastas</i>	84
7.2.3. Formaciones descentradas	86
7.2.4. Escenario completo	89
7.3. Simulaciones realistas	91
7.3.1. Caso simple, dos barcos en paralelo	91
7.3.2. Cambio de formación	93
7.3.3. Escenario completo	93
7.4. Conclusiones	93
8. Experimentos de Navegación	97
8.1. Descripción de los experimentos	99
8.2. Consideraciones	100
8.3. Resultados	101
8.4. Conclusiones	104
9. Conclusiones, contribuciones y trabajos futuros	105
A. Acrónimos y definiciones	111
Bibliografía	113

Índice de figuras

1.1. Barco autónomo desarrollado por el grupo de investigación durante un experimento.	4
1.2. Copias de la electrónica de a bordo de los barcos experimentales.	5
2.1. Diagrama conceptual de la solución propuesta.	11
2.2. Esquema del sistema propuesto.	13
2.3. Posible despliegue de los sistemas.	15
3.1. Ilustración de los sistemas de referencia ECI, ECEF y NED. Por cortesía de (Breivik, 2003)	20
3.2. Ilustración del sistema de referencia BARCO.	20
3.3. Relación entre los sistemas de referencia BARCO y NED.	22
3.4. Sistema experimental: despliegue.	24
4.1. Configuración inicial del algoritmo.	32
4.2. Configuración final del algoritmo.	34
4.3. Evolución del vector asignación.	34
4.4. Evolución del vector de precios.	35
5.1. Ejemplo de la aplicación de la <i>Unscented Transform</i> (UT)	40
6.1. Diagrama de la parte a implementar del sistema.	43
6.2. Sensores sin filtrar	46
6.3. Sensores filtrados	46
6.4. Diagrama del algoritmo estimador.	47
6.5. Ejemplo de curva de Bézier cúbica con puntos de control (6.13).	49
6.6. Derivada de la curva representada en la Figura 6.5.	50
6.7. Unión de las curvas con puntos de control (6.13) y (6.23).	51
6.8. Escenario de ejemplo del control NSB para la tarea de guiar el grupo.	55
6.9. Escenario de ejemplo del control NSB, tarea de mantener una formación.	57
6.10. Escenario de ejemplo del control NSB.	58
6.11. La propiedad de simetría no es deseable en la función h .	59
6.12. Esquema del sistema de Control.	60
6.13. Respuesta al escalón del control de rumbo.	62
6.14. Error y actuación del control de rumbo en la respuesta escalón.	63
6.15. Respuesta del control a varios cambios de rumbo.	63
6.16. Error y actuación del control de rumbo frente a varios cambios de consigna.	64
6.17. Respuesta del control a $\psi_d(t) = \sin\left(\frac{\pi}{5}t\right) + \sin\left(\frac{\pi}{20}t\right)$.	64
6.18. Error y actuación del control de rumbo frente a $\psi_d(t) = \sin\left(\frac{\pi}{10}t\right) + \sin\left(\frac{\pi}{40}t\right)$.	65
6.19. Respuesta al escalón del controlador.	67
6.20. Error y actuación del control de velocidad frente a una entrada escalón.	67

7.1. Componentes del software del simulador e interacciones entre los mismos.	73
7.2. UML	77
7.3. Captura de pantalla del simulador.	79
7.4. Escenario simple, dos barcos en paralelo. Sin subastas.	81
7.5. Escenario simple, dos barcos en paralelo. Empiezan cruzados. Sin subastas.	82
7.6. Escenario simple, dos barcos en paralelo. Empiezan cruzados. Con subastas.	83
7.7. Escenario para probar las <i>resubastas</i>	85
7.8. Fase inicial del escenario	85
7.9. Convergencia a formaciones centradas y descentradas.	87
7.10. Efecto de la normalización de las formaciones en $\ \tilde{\sigma}_f\ $	88
7.11. Convergencia a formaciones descentradas y luego normalizadas.	88
7.12. Escenario completo.	90
7.13. Resultados del estimador en un escenario simple.	92
7.14. Resultados del estimador en un escenario con cambios de formación.	94
7.15. Resultados del estimador en un escenario con cambios de formación.	95
8.1. Barco durante uno de los experimentos.	99
8.2. Resultados del estimador durante los <i>ochos</i>	102
8.3. Comparación de la velocidad medida por el sensor de presión y el <i>Global Positioning System</i> (GPS).	103
8.4. Orientación medida por la brújula.	103
8.5. Resultados del estimador utilizando UKF.	104

Índice de tablas

2.1. Requisitos funcionales del sistema de Guiado, Navegación y Control.	16
3.1. Datos físicos del barco experimental.	25
7.1. Requisitos funcionales del simulador.	72
7.2. Requisitos no funcionales del simulador.	72
9.1. Requisitos funcionales del sistema. Copia de la Tabla 2.1.	108

Capítulo 1

Introducción

El presente capítulo tiene como objetivo principal introducir el contenido del Trabajo de Fin de Máster. Adicionalmente, se enmarca el Trabajo tanto desde el punto de vista de la sociedad así como desde el punto de vista científico.

Por último, se da una visión general de los distintos capítulos y secciones del trabajo, de forma que su lectura sea más amena y sea fácil encontrar información concreta.

1.1. Motivación y objetivos

Existen, en la actualidad, numerosos escenarios donde los barcos autónomos pueden ser útiles. Los barcos autónomos tienen la ventaja de no requerir apenas intervención humana, reduciendo riesgos para las personas y haciendo el trabajo más eficiente. Algunas aplicaciones de los barcos autónomos son tareas de reconocimiento, persecución, protección, salvamento, despliegue de redes de pesca, recogida o confinamiento de vertidos, la creación de mapas batimétricos, ...

Muchos de estos escenarios, además, pueden ser tratados de una forma más eficiente por formaciones de estos barcos. El tema de la creación y mantenimiento de las formaciones es un problema recurrente en la robótica terrestre, así como cualquier tipo de escenario cooperativo. Hay escenarios en los que una solución cooperativa es más natural que la solución de un solo elemento aislado; el caso más representativo es el transporte de un elemento grande como una gran caja, un tablón, una mesa, ... el escenario de algún modo *pide* la cooperación de varios agentes. Escenarios marítimos con estas características también se encuentran con facilidad, por ejemplo, el confinamiento de un vertido de petróleo puede ser llevado a cabo por una pareja de barcos autónomos; o el remolque de un gran barco por parte de varios prácticos de puerto.

Desde el punto de vista de las formaciones, en los entornos marítimos también aparecen situaciones en las que aparecen las mismas. En el mundo militar, el desplazamiento de un grupo de fuerza aeronaval tiene unas reglas y normas sobre las formaciones. Las maniobras de suministro en alta mar también son un caso particular del control de formaciones.

Es importante tener como referencia todo el desarrollo realizado por la comunidad científica e industrial en la robótica terrestre a la hora de traer estos conceptos al mundo marino. Sin embargo, existen diferencias importantes entre ambos mundos:

- Los barcos, en principio, no frenan. Tampoco todos tienen facilidad de ir hacia atrás.
- Generalmente, cuentan con propulsión trasera, lo que hace que algunos sean inestables a rumbo.
- Son elementos muy lentos, pero que cuentan con grandes inercias.
- Tienen desplazamientos y movimientos de deriva (laterales).
- Las fuerzas de viento y corriente influyen mucho. No solo resisten el movimiento (como en el caso típico de un coche) si no que, además, son capaces de cambiar el rumbo y la trayectoria del sistema.
- El problema de evitar y sortear obstáculos tiene la particularidad de que, en algunos casos, también debe contar con la profundidad del fondo marino.
- Existen efectos hidrodinámicos difíciles de modelar y estimar; efectos como la atracción de dos barcos muy juntos y los efectos del agua poco profunda o *shallow water*.

Desde el punto de vista de la metodología, en este Trabajo se ha decidido utilizar una metáfora económica o de mercado para resolver algunos de los problemas y para explicar algunas de las soluciones. Este tipo de metáforas es común en la literatura y, se trata de una modesta contribución a las aplicaciones de la economía para la resolución de problemas de cooperación y coordinación.

Adicionalmente, el grupo de investigación del autor está ligado al estudio y desarrollo de vehículos autónomos como helicópteros cuatri-rotor, aviones, y barcos. En concreto, el autor ha

estado involucrado en el desarrollo de un barco autónomo que sirve como plataforma experimental, entre otras cosas, para el estudio del desminado automático. La Figura 1.1 muestra una foto del barco.



Figura 1.1: Barco autónomo desarrollado por el grupo de investigación durante un experimento.

Los barcos están equipados con una electrónica diseñada ex profeso para el control autónomo. Cuentan con distintos sensores para medir la orientación (brújula), la velocidad (sensor de presión y tubo de Pitot) y para posicionar globalmente al barco (receptor *Global Positioning System* (GPS)). Cuentan con un sistema de radio que permite tanto monitorizar el barco utilizando una estación de tierra como comunicarse con otros barcos de la flota. La Figura 1.2 muestra una fotografía de dos copias de la electrónica de a bordo de los barcos. La brújula es el cuadrado rojo de la parte inferior mientras el módulo azul a su izquierda es el sistema de radio. Justo al lado del módulo de radio se encuentra el sensor GPS. Adicionalmente, la electrónica cuenta con un sistema de toma de datos utilizando tarjetas SD como las que se pueden encontrar en las cámaras digitales de fotografía. Todos los sensores se gobiernan utilizando un microprocesador ARM7. El desarrollo de la parte de software de los barcos así como de la estación de tierra se encuentra en (Pereda, 2010).

En este punto, se pueden formular los objetivos del Trabajo. Más adelante, en la Sección 2.3 (Requisitos) se escriben de un modo más detallado. No obstante, a grandes rasgos, los objetivos son los siguientes:

- 1) Crear un sistema capaz de guiar a una flota de barcos autónomos a lo largo de una trayectoria.
- 2) Que los barcos de la flota mantengan una cierta configuración geométrica. Debe permitirse la inicialización de la formación desde cualquier configuración así como cambios en la forma de la misma.
- 3) Que el sistema sea capaz de funcionar con sensores y elementos comunes. Es decir, debe contar con algún tipo de filtrado y estimación.

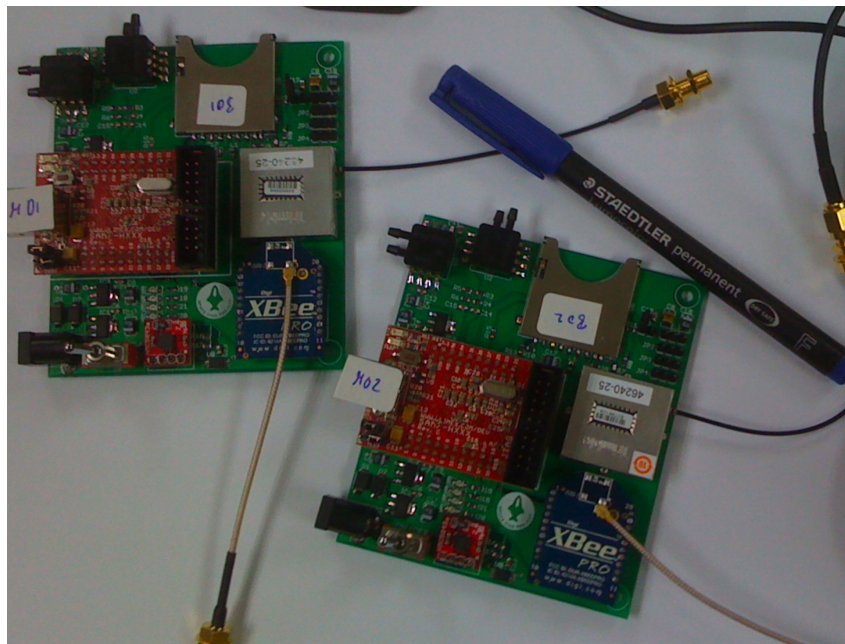


Figura 1.2: Copias de la electrónica de a bordo de los barcos experimentales.

- 4) Que los barcos y el sistema sea completamente autónomo y realimentado; una vez dada la configuración de la misión, debe llevarlo a cabo sin intervención humana.

El sistema de Guiado nace para cumplir el primer y segundo objetivos. Se utiliza el control jerárquico *Null-Space-Based Behavioral Control* (NSB) para el guiado y mantenimiento de la formación geométrica; un algoritmo de subastas para las fases de inicialización y cambio de formación; y un sistema de splines para describir las trayectorias del grupo. El sistema de Navegación, basado en *Unscented Kalman Filter* (UKF), aparece como producto del tercer objetivo, para fusionar las señales de los sensores del barco y proporcionar estimaciones razonables cuando estas señales tienen ruido. El Control ayuda al último objetivo, hace que los barcos sigan las directrices del Guiado.

1.2. Estado del arte

Como se ha comentado anteriormente, la investigación realizada en el Trabajo está ligada con la cooperación multi-robot; algunas referencias relevantes sobre este aspecto son (Hirsh et al., 2007), (Shima and Rasmusen, 2009) y (Vig, 2008). Desde el punto de vista de la cooperación (Hsu and Liu, 2005) ofrece una taxonomía y vocabularios utilizados en algunos puntos de este Trabajo.

Utilizar una metáfora económica, o de mercado, es común en el mundo de la distribución de tareas y carga. La referencia básica desde un punto de vista conceptual es (Davis and Smith, 1981); en ese informe se plantean los problemas que se pueden solucionar así como la metodología de base. Adicionalmente, se desarrolla un protocolo de comunicación y los conceptos utilizados a la hora de incluir la metáfora en la metodología de trabajo. Un ejemplo muy interesante de la aplicación de este tipo de metáforas se puede encontrar en (Gerkey and Mataric, 2002), donde se utiliza un sistema de subastas para que un conjunto de robots empujen una caja. En (Viguria et al., 2007) y (Viguria et al., 2008) se describe un algoritmo basado en subastas también, utilizado para, entre otras cosas, repartir tareas entre helicópteros autónomos para vigilancia

de equipos de bomberos (Maza et al., 2010). Uno de los algoritmos fundamentales en casi toda aplicación de esta metáfora es el algoritmo de subastas, presentado en (Bertsekas, 1989).

Desde el punto de vista de control y estimación en barcos, la referencia de carácter general más consultada durante la creación del Trabajo ha sido Fossen (2011). Sin embargo, algunas referencias de carácter específico con tienen mucha información interesante. Desde el punto de vista de la estimación de estados (Caccia et al., 2008) ofrece un sistema completo de navegación, guiado y control de un barco autónomo. En (Skejic et al., 2009) se hace el modelado de una maniobra de suministro en marcha de dos barcos. En (Breivik et al., 2008b) se hace un estudio teórico y experimental de una persecución utilizando un barco autónomo y se ofrecen datos reales del control del mismo.

En el ámbito del control por capas o jerárquico, la técnica utilizada en el Trabajo es NSB; cuyas referencias principales son (Antonelli et al., 2005), (Arrichiello et al., 2006) y (Antonelli et al., 2010). De esas tres referencias, la aplicación de este sistema a barcos se puede encontrar, únicamente en (Arrichiello et al., 2006). No obstante, otras técnicas existen para controlar una formación; por ejemplo (Dong, 2010) propone un sistema de control distribuido en el que no necesita que el grafo de comunicaciones sea completo. Un enfoque desde el punto de vista del seguimiento de trayectorias basado en (Breivik and Fossen, 2005) se puede encontrar en (Breivik et al., 2008a).

Para fusionar los datos de los distintos sensores del barco descrito en la sección anterior, se utiliza el UKF, que fue presentado en (Julier and Uhlmann, 1997) y más tarde desarrollado en (Wan and van der Merwe, 2001) y (Julier and Uhlmann, 2004). La utilización de UKF en entornos marinos no es muy habitual, siendo más común el uso de *Extended Kalman Filter* (EKF), por ejemplo, en (Caccia et al., 2008).

Dos referencias muy completas en temas de control de barcos son las Tesis de Máster y Doctorado de Morten Breivik (Breivik, 2003) y (Breivik, 2010), respectivamente.

1.3. Estructura del Trabajo Fin de Máster

El Trabajo se ha organizado en distintos capítulos, intentando que el orden vaya descubriendo los detalles a medida que sea necesario. Referencias internas a distintas partes del documento se han incluido para facilitar la navegación por el mismo.

Descripción general del sistema hace una propuesta de alto nivel de la solución planteada. Presenta una visión general, identifica el alcance del sistema; y, por último, resume los requisitos más importantes del mismo. Se identifican los tres grandes componentes: Guiado, Navegación y Control. De cada uno de ellos se detalla su alcance así como sus responsabilidades.

Modelado matemático formaliza los modelos matemáticos utilizados en el Trabajo. Explica los sistemas de referencia involucrados en la definición de los modelos, y desarrolla estos últimos con todo el nivel de detalle empleado. Por último, da los parámetros identificados y estimados del barco experimental; estos parámetros son los utilizados en las simulaciones del Trabajo.

El problema de la asignación y el algoritmo de subastas presenta formalmente el problema de la asignación y el algoritmo de la subastas. El problema de la asignación aparece en el Trabajo en dos momentos: en la inicialización de la formación y en los cambios de formación. En ambos casos, el algoritmo permite reducir el tiempo de convergencia de los

barcos a la formación deseada. El capítulo está ilustrado con un ejemplo práctico de los que se dan en el Trabajo.

El filtro de Kalman *Unscented* describe el algoritmo general del *Unscented Kalman Filter* (UKF). Este algoritmo es utilizado para la fusión sensorial en el sistema de Navegación.

Implementación del sistema es el capítulo central del Trabajo. Contiene el detalle de todos los algoritmos utilizados. Tiene tres grandes secciones:

Navegación contiene el detalle de los componentes del sistema de Navegación. La primera sección está dedicada al filtrado de las señales y la segunda a los modelos de propagación y corrección utilizados en el UKF.

Guiado detalla los elementos existentes en este componente. Por un lado, propone utilizar splines basados en curvas de Bézier cúbicas para describir las trayectorias del grupo. En la Sección 6.2.2 resume el control NSB y las tareas utilizadas en el Trabajo; se ha hecho un esfuerzo por incluir ejemplos relevantes relacionados con el Trabajo en esta sección. Por último, se hacen las apreciaciones necesarias a la hora de aplicar la metáfora de las subastas para hacer formaciones de barcos.

Control contiene el desarrollo matemático de la ley de control utilizada en el Trabajo. La ley se ha derivado utilizando la teoría de Lyapunov.

Resultados en simulación . Este capítulo tiene dos objetivos: por un lado mostrar el simulador como una pieza de software compleja desde el punto de vista de la Ingeniería de Software; y por otro lado, la explicación de los resultados obtenidos con dicho simulador. Se han probado escenarios completamente *ideales* y también realistas. Un escenario realista incluye modelos de los sensores y sus ruidos para hacer que la simulación parezca algo más real. Por último, el capítulo saca unas conclusiones sobre los escenarios estudiados. Los resultados obtenidos son muy buenos y se tiene confianza suficiente para llevar el sistema a experimentos de campo.

Experimentos de Navegación recoge los resultados experimentales del sistema de Navegación. En la propuesta, la Navegación es crucial; de forma que estos resultados permiten iniciar los trabajos experimentales del sistema al completo.

Conclusiones, contribuciones y trabajos futuros es el capítulo final. Resume las conclusiones y resultados más importantes obtenidos en el Trabajo y apunta a posibles líneas futuras.

Capítulo 2

Descripción general del sistema

El capítulo contiene los diagramas conceptuales y la explicación de los distintos conceptos del presente sistema. El detalle sobre cada uno de los elementos puede encontrarse en el Capítulo 6 (Implementación del sistema). Este capítulo está dedicado a presentar los distintos componentes, definiendo, de cada uno de ellos, el alcance y sus responsabilidades; sin entrar en detalles de implementación.

La primera sección se centra en dar una visión general del sistema. La segunda sección define el alcance del mismo así como de los distintos subsistemas. La tercera sección hace un pequeño resumen de los requisitos impuestos sobre los elementos a desarrollar.

2.1. Visión general

El objetivo del Trabajo es plantear un sistema capaz de conducir una formación de barcos a lo largo de una trayectoria. La forma geométrica de la formación no es estática durante toda la misión. Los cambios de formación deben hacerse de forma que las maniobras sean suaves y los barcos converjan a la formación deseada en el menor tiempo posible.

Es común dividir cualquier sistema de navegación autónoma en tres sistemas: Guiado, Navegación y Control (Caccia et al., 2008). Esta división en subsistemas corresponde a una división y separación de responsabilidades. Las responsabilidades de los distintos sistemas son las siguientes:

Guiado es el sistema que decide sobre la trayectoria del vehículo así como de las consignas de rumbo y velocidad que debe tomar el Control.

Navegación es el responsable de la determinación de la posición y actitud del vehículo.

Control es el sistema que manipula los actuadores de forma que se cumplan las consignas del sistema de Guiado.

Puede verse cómo esta separación de responsabilidades está en línea con el principio de *alta cohesión interna y bajo acoplamiento externo*. Que es un principio básico y fundamental en Informática. Este principio es la norma básica de todo buen diseño. Se debe tratar de maximizar la cohesión entre los componentes de un sistema mientras se minimizan las dependencias entre los distintos sistemas.

En la literatura existen muchas soluciones al problema de mantener una formación de barcos y guiarla a través de una trayectoria. Sin embargo, quizá por venir de ámbitos distintos a la Informática, no todas las soluciones propuestas respetan el principio de *alta cohesión, bajo acoplamiento*. Un claro ejemplo es (Dong, 2010), donde las tareas de guiado y control se solapan.

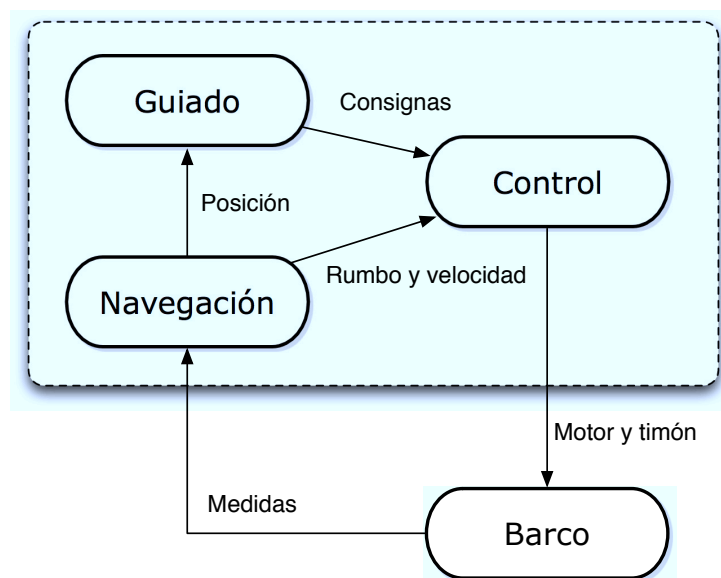


Figura 2.1: Diagrama conceptual de la solución propuesta.

La solución propuesta consta de tres grandes sistemas que se corresponden con las responsabilidades de Guiado, Navegación y Control descritas anteriormente. La Figura 2.1 muestra diagrama conceptual de la solución propuesta. Este capítulo sirve para introducir el sistema y

los distintos conceptos. Las formalización y descripciones detalladas de los procedimientos y técnicas utilizadas en cada uno de los sistemas se presentan en el Capítulo 6 (Implementación del sistema).

El sistema de Control se ha desarrollado por medio de funciones de Lyapunov para controlar el rumbo y la velocidad de cada barco. Recibe por un lado las consignas del sistema de Guiado, del sistema de Navegación su estado actual, y actúa en consecuencia. Su única responsabilidad es que el barco mantenga un cierto rumbo a una cierta velocidad de avance. Cada barco lleva a bordo un sistema de Control.

El sistema de Navegación es el responsable de procesar la información de los sensores para producir un vector de estados que contenga la posición del barco así como su rumbo y velocidad de avance. La solución tradicional en barcos es utilizar una brújula y un GPS. En este caso, además, se cuenta con un sensor de presión para medir la velocidad de avance del barco. Dado que el GPS proporciona una medida de posición cada segundo, se ha decidido integrar un estimador de posición utilizando un filtro de Kalman. Cada barco cuenta con su propio sistema de Navegación.

El sistema de Guiado es responsable de llevar los barcos a lo largo de una trayectoria. En este Trabajo se ha decidido utilizar curvas de Bézier cúbicas para representar la trayectoria que deben seguir los barcos. El sistema debe asegurar que los barcos no se chocan o cruzan muy cerca unos de otros así como organizarlos de la mejor forma posible en la formación. Las formaciones son rígidas, es decir, tienen una forma determinada durante su tiempo de vida; sin embargo, se debe permitir a la formación cambiar de forma. Durante un cierto tiempo los barcos pueden ir en fila para luego ir formando una delta. El sistema de Guiado es responsable de que estos cambios se hagan de la mejor forma posible. Se ha decidido utilizar una técnica de control denominada *Null-Space-Based Behavioral Control*, introducida en (Antonelli et al., 2005) y utilizada en barcos por primera vez en (Arrichiello et al., 2006). Para que los cambios de formación sean suaves, se ha decidido utilizar un algoritmo de subastas para decidir cuál es la mejor asignación de las posiciones en la formación. El sistema de Guiado es común a todos los barcos, es decir, puede llevarlo un solo barco o encontrarse en una estación en Tierra; pero solo existe una instancia de este sistema.

La Figura 2.2 muestra el sistema completo. Puede observarse claramente la realimentación del sistema así como el flujo de los datos y las actuaciones. Viendo detalladamente el sistema se puede apreciar un cierto retardo introducido por el módulo de estimación (Estimador). Sin embargo, a efectos prácticos, este retardo es despreciable.

Es importante hacer notar el hecho de que el diseño de los componentes de los sistemas de Navegación y Control también siguen los principios de *alta cohesión y bajo acoplamiento*. En el caso del sistema de Navegación, las señales en crudo se filtran por un componente de filtrado y se pasan tanto al sistema de Control como al componente de estimación. El componente de estimación recibe estos datos y utiliza un UKF para estimar la posición del barco entre las distintas medidas del GPS. En el sistema de Guiado, el módulo de NSB recibe las posiciones de todos los barcos del sistema y genera las consignas que son enviadas a los sistemas de Control de cada uno los barcos. Para generar las consignas se apoya en un componente de trayectorias basado en splines cúbicos y un componente de subastas para la gestión de las formaciones.

Hacer los elementos modulares tiene distintas ventajas. Desde el punto de vista del diseño se respeta el ya mencionado principio de *alta cohesión y bajo acoplamiento*. Desde un punto de vista práctico esto es muy deseable pues el desarrollo es mucho más fácil si las responsabilidades y el alcance de los distintos componentes está bien definido y acotado. Desde el punto de vista de la depuración y el mantenimiento de los sistemas involucrados, también es deseable que el sistema sea modular pues los errores suelen ser más fáciles de encontrar y reparar.

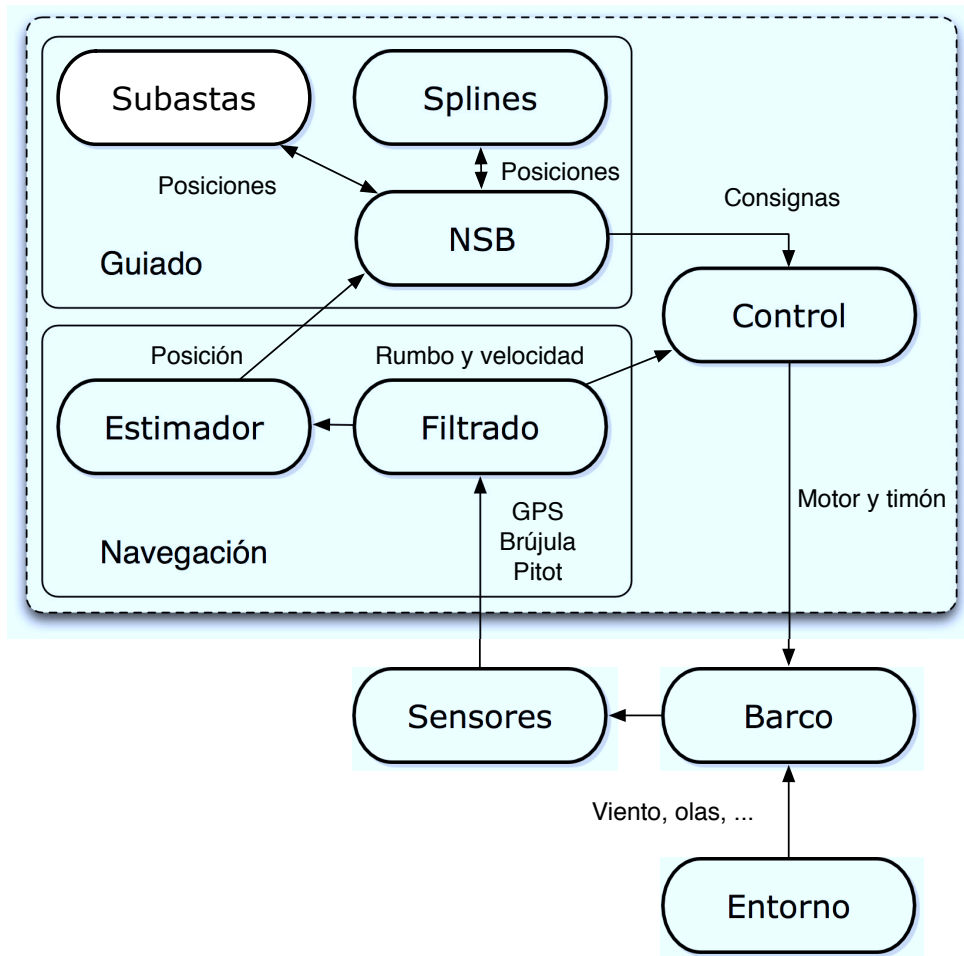


Figura 2.2: Esquema del sistema propuesto.

2.2. Alcance del sistema

El sistema está limitado a los algoritmos de Guiado, Navegación y Control del barco. Para el desarrollo de los mismos es interesante conocer la planta sobre la que trabajan así como las perturbaciones externas y sensores de la misma.

La Figura 2.2 representa el sistema frente al *mundo* utilizando una caja de línea discontinua. Dentro de la misma se representan los sistemas de Guiado, Navegación y Control así como los distintos componentes descritos en la sección *Visión general* de este mismo capítulo. No obstante, es imposible describir el sistema correctamente, ignorando los elementos externos al mismo. Al fin y al cabo, el control será muy dependiente de la planta que se controle y el sistema de navegación estará altamente ligado a los sensores y los ruidos presentes en sus medidas. Los sistemas de guiado y navegación, además, deben tener en cuenta que el sistema está sometido a perturbaciones externas. De modo que aunque el sistema tiene un alcance limitado, debe hacerse un estudio del entorno en el que se enmarca de cara a hacer una implementación real tanto en simulación como en una plataforma experimental.

Desde el punto de vista global, el sistema controla el timón y el motor de un barco del cual obtiene medidas de: posición, orientación y velocidad de avance. Teóricamente el sistema podría funcionar en cualquier barco con estas características; en la práctica, aunque esto es posible, habría que hacer bastantes ajustes. Sobre todo en los sistemas de navegación y control.

Desde el punto de vista de cada uno de los sistemas:

Navegación El sistema recibe las señales de los sensores. Filtra estas señales y utiliza un algoritmo de estimación. El sistema alimenta al sistema de Guiado con la posición del barco y al sistema de Control con la orientación y velocidad de avance.

Filtrado Recibe los datos de GPS, orientación y velocidad de avance y los filtra. Alimenta al sistema de Control con la orientación y la velocidad de avance. Envía todos los datos filtrados al componente Estimador.

Estimador Utiliza un UKF para estimar la posición del barco utilizando los datos que le entrega el componente de Filtrado. Alimenta con la estimación al sistema de Guiado.

Guiado El sistema recibe las posiciones de todos y cada uno de los barcos de la formación. Utilizando los sistemas y técnicas mencionadas anteriormente, genera las consignas de velocidad y rumbo de los barcos de la formación. El Guiado alimenta al sistema de Control con estas consignas.

NSB Recibe las posiciones de todos los barcos y decide sobre las consignas de cada uno de ellos. Utiliza el componente de Splines para definir la trayectoria que debe seguir la formación. Además, utiliza el componente de Subastas para decidir la mejor configuración de la formación en cada momento. Alimenta al sistema de Control con las consignas.

Splines Recibe la posición del centroide de la formación y envía el punto al que debe dirigirse la formación para seguir la trayectoria deseada.

Subastas Recibe la posición de todos y cada uno de los barcos así como de la configuración de la formación. Utilizando distintas métricas para decidir la mejor asignación. Resuelve el problema de la formación utilizando un algoritmo de subastas.

Control El sistema recibe las consignas de rumbo y velocidad así como la orientación y la velocidad de avance. Utiliza las consignas y el estado del barco para decidir las acciones de timón y motor que debe tomar el barco.

Un posible despliegue de los distintos sistemas está representado en la Figura 2.3. Varios barcos participan en una formación guiada a lo largo de una trayectoria. Todos los barcos cuentan con sus propios sistemas de Navegación y Control. Sin embargo, solo existe un sistema de Guiado. El sistema de Guiado puede estar en cualquiera de los barcos pertenecientes a la formación o incluso en una plataforma o barco externo a la formación; el único requisito es que todos los sistemas de Navegación y Control puedan comunicarse con el sistema de Guiado.

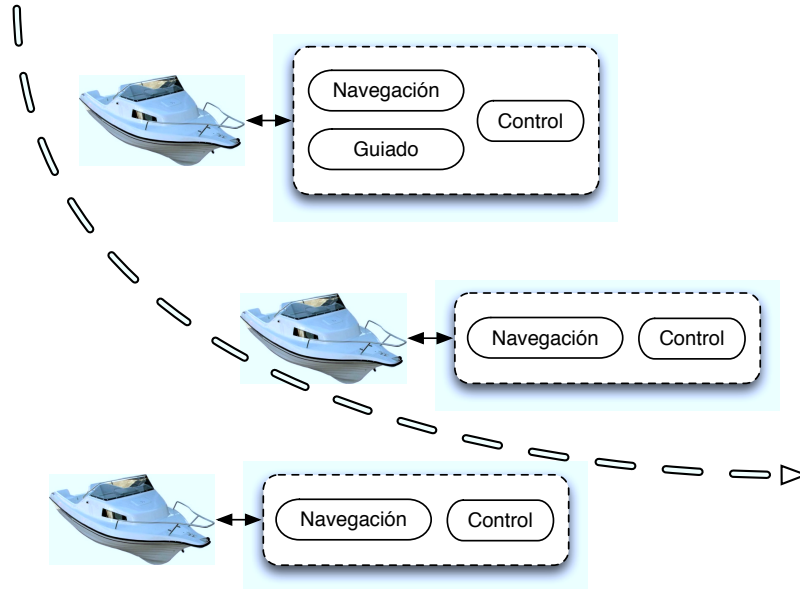


Figura 2.3: Posible despliegue de los sistemas.

Con esto finaliza la definición del alcance de los sistemas.

2.3. Requisitos

Los requisitos del sistema de Guiado, Navegación y Control de la flota deben cumplir, entre otros, los siguientes requisitos. Se recogen aquí por razones de completitud y para poder justificar las decisiones de diseño cuando sea necesario. La Tabla 2.1 recoge estos requisitos de alto nivel.

Tabla 2.1: Requisitos funcionales del sistema de Guiado, Navegación y Control.

Identificador	Descripción
GNC R.1	El sistema debe guiar el grupo a lo largo de una trayectoria geométrica. Es decir, no hace falta considerar restricciones temporales, solo espaciales.
GNC R.2	Debe ser suficientemente robusto para funcionar bajo los errores comunes de los sensores utilizados en barcos autónomos.
GNC R.3	El sistema debe permitir describir y mantener la forma geométrica del grupo. Es decir, la forma de la formación es un parámetro del sistema.
GNC R.4	El sistema debe evitar choques entre los barcos a toda costa.
GNC R.5	El sistema debe hacer que las trayectorias de los barcos sean lo más limpias posibles a lo largo de la misión. Esto es especialmente importante en la inicialización de la formación y entre los distintos cambios de formación.
GNC R.6	El sistema debe ser independiente del número de barcos de la flota. Este número es un parámetro al sistema y solo debe asumirse que será mayor que 0.
GNC R.7	El sistema debe ser completamente autónomo. Una vez decidida la trayectoria, el número de barcos y las formaciones, no debe requerir más intervención.

Capítulo 3

Modelado matemático

Este capítulo contiene las bases matemáticas para poder desarrollar y entender el contenido de este Trabajo. Los contenidos de este capítulo son estándares en la literatura relacionada con el modelado, guiado y control de vehículos en general y de barcos en particular. Se incluyen por razones de completitud.

La primera sección desarrolla los sistemas de referencia utilizados durante el Trabajo mientras que la segunda está dedicada a los modelos (cinemático y dinámico) empleados.

3.1. Sistemas de referencia

Para describir la posición y orientación de un barco a escala global, son necesarios cuatro sistemas de referencia. Dos de ellos con origen en el centro de la tierra, los otros dos son sistemas locales.

Earth-Centered Inertial (ECI) Realmente este es el único sistema de referencia inercial. En la Figura 3.1 el sistema de referencia ECI tiene subíndice I .

O_{ECI} es el centro de la Tierra.

X_{ECI} está contenido en el plano ecuatorial y apunta al equinoccio de verano. El punto de corte entre el plano eclíptico y el ecuatorial.

Z_{ECI} tiene la misma velocidad angular terrestre y apunta al Norte.

Y_{ECI} definido de tal forma que complete la triada.

Earth-Centered Earth-Fixed (ECEF) Es un sistema de referencia con el mismo origen que el sistema ECI, pero que rota con la Tierra. En la Figura 3.1 este sistema de referencia se indica con subíndice E . Este sistema de referencia, en la práctica, puede considerarse inercial para la navegación de barcos.

O_{ECEF} es el centro de la Tierra.

X_{ECEF} interseca la esfera terrestre en el punto de latitud 0° y longitud 0° .

Z_{ECEF} apunta al Norte.

Y_{ECEF} definido de tal forma que complete la triada.

North-East-Down (NED) El sistema de referencia representa a un plano tangente a la superficie de la Tierra. Al igual que con el sistema ECEF, el sistema puede considerarse inercial para la navegación marítima. En la Figura 3.1 este sistema de referencia aparece con subíndice N .

O_{NED} es algún punto en la superficie de la Tierra.

X_{NED} apunta al Norte.

Y_{NED} apunta al Este.

Z_{NED} apunta hacia el centro de la Tierra.

BARCO Sistema solidario al barco, con origen en su centro de masas. Representado en la Figura 3.2.

O_{BARCO} es el centro de masas del barco.

X_{BARCO} es longitudinal al barco. Tiene la misma dirección que el eje popa-proa y sentido hacia proa. La rotación sobre este eje se denomina *escora* (ϕ , en inglés *roll*).

Y_{BARCO} es transversal al barco. Tiene la misma dirección que el eje babor-estribor y sentido hacia estribor. La rotación sobre este eje se denomina *cabeceo* (θ , en inglés *pitch*).

Z_{BARCO} apunta hacia abajo, de forma que completa la triada. La rotación sobre este eje se denomina *guiñada*.

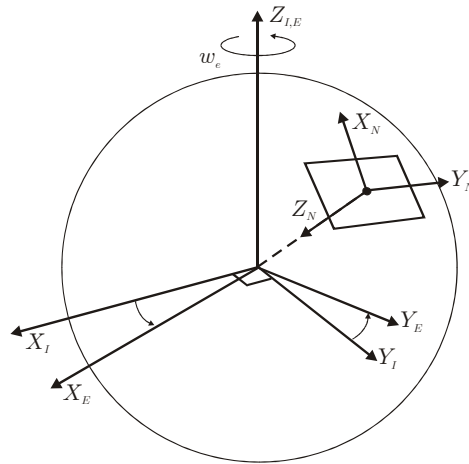


Figura 3.1: Ilustración de los sistemas de referencia ECI, ECEF y NED. Por cortesía de (Breivik, 2003)

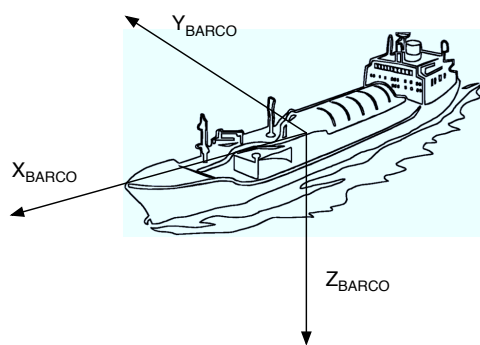


Figura 3.2: Ilustración del sistema de referencia BARCO.

3.2. Modelo de maniobra del barco

El sistema a modelar consta del agua que rodea al barco y al propio barco. El modelo completo de un barco consta del movimiento en sus 6 grados de libertad; sin embargo, para estudiar maniobras horizontales se puede utilizar un modelo reducido en 3 grados de libertad. Formalmente, el modelo presentado en esta sección asume ángulos de escora y cabeceo nulos o despreciables; es decir: $\phi \approx 0$ y $\theta \approx 0$.

El modelo presentado en esta sección es un resumen del contenido en (Fossen, 2011), para más información sobre el mismo, debe consultarse como referencia principal.

La sección está compuesta de dos partes, el objetivo de la sección es obtener unas ecuaciones que permitan simular maniobras de barcos. La primera sección trata el modelo cinemático del barco mientras que la segunda está dedicada al modelo dinámico.

3.2.1. Modelo cinemático

El modelo de maniobra solo considera 3 grados de libertad: la posición del barco en el sistema de referencia NED y su orientación (ángulo con respecto al Norte). El estado del barco puede describirse, por tanto, como un elemento perteneciente al grupo especial euclídeo de 2 dimensiones $SE(2)$. El grupo especial euclídeo $SE(2)$ es homeomorfo a $\mathbb{R}^2 \times \mathbb{S}^1$; es decir, el producto cartesiano de los puntos del plano euclídeo con las rotaciones del círculo unidad. De forma general, el grupo especial euclídeo de n dimensiones incluye a todas las matrices de la forma

$$\left\{ \begin{bmatrix} \mathbf{R} & \mathbf{v} \\ 0 & 1 \end{bmatrix} \mid \mathbf{R} \in SO(n), \mathbf{v} \in \mathbb{R}^n \right\} . \quad (3.1)$$

Un resumen práctico sobre topología orientado a la descripción de espacios de configuración es (LaValle, 2006).

La relación entre los sistemas de referencia BARCO y NED se puede expresar de la siguiente forma,

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\mathbf{v} \quad (3.2)$$

donde se identifican los siguientes elementos:

$\boldsymbol{\eta} = [n \ e \ \psi]^T \in SE(2)$ es la posición y orientación del barco en el sistema de referencia de navegación NED.

$\mathbf{v} = [u \ v \ r]^T \in SE(2)$ son las velocidades en ejes BARCO y la velocidad angular del sistema de referencia BARCO frente a NED. Es decir, u es la dirección de avance, v la velocidad de deriva y r la velocidad angular del barco.

$\mathbf{R}(\psi) \in SO(3)$ es la matriz de rotación del sistema de referencia BARCO al sistema de referencia NED. Esta matriz representa una rotación alrededor del eje D del sistema de referencia NED:

$$\mathbf{R}(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

La relación entre los sistemas de referencia BARCO y NED puede verse en la Figura 3.3.

La matriz de rotación $\mathbf{R}(\psi)$ tiene las siguientes propiedades:

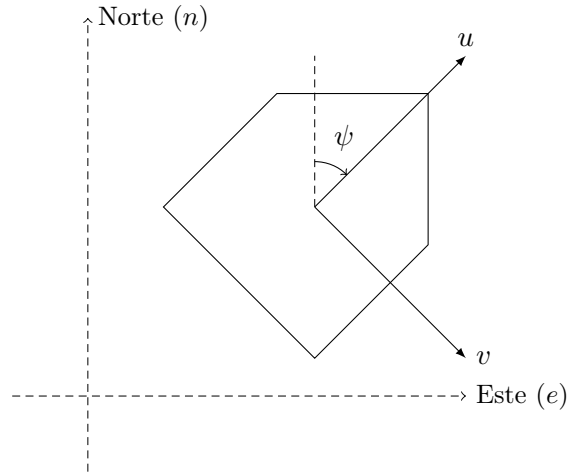


Figura 3.3: Relación entre los sistemas de referencia BARCO y NED.

- $\mathbf{R}(\psi)^{-1} = \mathbf{R}(\psi)^T \implies \mathbf{R}(\psi)^T \mathbf{R}(\psi) = \mathbf{I}$
- $\|\mathbf{R}(\psi)\| = 1$

Con esto concluye el modelo cinemático.

3.2.2. Modelo dinámico

La siguiente ecuación describe el movimiento de 3 grados de libertad del barco:

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} = \boldsymbol{\tau}_{\text{prop}} + \mathbf{R}(\psi)^T \boldsymbol{\tau}_{\text{pert}} \quad (3.4)$$

donde los diferentes términos son:

$\mathbf{M} \in \mathbb{R}^{3 \times 3}$ es la matriz de masas e inercia del barco así como las masas y momentos añadidos debido al agua que desplaza el barco a medida que se mueve en ella. A bajas velocidades esta matriz es simétrica y, además, $\mathbf{M} = \mathbf{M}^T > 0$. A altas velocidades, es posible hacer que esta matriz lo sea utilizando realimentación de la aceleración; los detalles se pueden encontrar en (Breivik, 2003).

$\mathbf{C}(\mathbf{v}) \in \mathbb{R}^{3 \times 3}$ representa las fuerzas centrífugas y de Coriolis que actúan en el barco. Es importante hacer notar que estas fuerzas no hacen trabajo sobre el sistema. No obstante, el término aparece porque el modelo dinámico está desarrollado en el sistema de referencia BARCO, que no es inercial y rota con respecto a NED.

La matriz $\mathbf{C}(\mathbf{v})$ tiene la propiedad de ser antisimétrica, es decir, $\mathbf{C}(\mathbf{v}) = -\mathbf{C}(\mathbf{v})^T$, lo que implica que $\mathbf{x}^T \mathbf{C} \mathbf{x} = 0$. Ambas propiedades confirman el hecho de que las fuerzas y momentos de esta matriz no contribuyen trabajo al sistema. La matriz $\mathbf{C}(\mathbf{v})$ puede escribirse de distintas formas según sea \mathbf{M} , una de ellas es la siguiente que puede encontrarse en (Fossen, 2011):

$$\mathbf{C}(\mathbf{v}) = \begin{bmatrix} 0 & 0 & -(m_{22}v + \frac{m_{23}+m_{32}}{2}r) \\ 0 & 0 & m_{11}u \\ (m_{22}v + \frac{m_{23}+m_{32}}{2}r) & -m_{11}u & 0 \end{bmatrix} \quad (3.5)$$

$$= \begin{bmatrix} 0 & 0 & -(m_{22}v + m_{23}r) \\ 0 & 0 & m_{11}u \\ (m_{22}v + m_{23}r) & -m_{11}u & 0 \end{bmatrix} \quad (3.6)$$

donde los m_{ij} son los elementos de la matriz \mathbf{M} , y el paso (3.6) es debido a la simetría de \mathbf{M} .

$\mathbf{D}(\mathbf{v}) \in \mathbb{R}^{3 \times 3}$ es la matriz de fuerzas y momentos hidrodinámicos. De forma general, se tiene que $\mathbf{D}(\mathbf{v}) > 0 \forall \mathbf{v} \neq 0$ por ser una matriz de amortiguamiento.

$\boldsymbol{\tau}_{\text{prop}} \in \mathbb{R}^3$ es el vector de fuerzas y momentos ejercidos por el sistema de propulsión del barco. Según sea el tipo de sistema de propulsión, este vector se modelará de distinta forma.

$\boldsymbol{\tau}_{\text{pert}} \in \mathbb{R}^3$ es el vector de fuerzas y momentos ejercidos sobre el sistema debidas a elementos ambientales como viento y corriente. Se asume que las fuerzas están representadas en el sistema de referencia NED, es por esto que, en el modelo, va multiplicado por $\mathbf{R}(\psi)^T$.

Aunque existen modelos más elaborados, algunos incluyendo el ángulo de escora o incluso también el de cabeceo, el modelo presentado en esta sección es más que suficiente para estudiar maniobras de barcos. Otros modelos utilizados para estudiar el comportamiento dinámico de elementos náuticos pueden encontrarse en (Fossen, 2011).

3.2.3. Barco experimental

El desarrollo del Trabajo está orientado a hacer pruebas experimentales en una plataforma experimental muy concreta. A continuación se describe el modelado matemático tanto de los propulsores del barco experimental así como de los distintos elementos de la Ecuación (3.4).

La plataforma experimental consta de un barco de radio control y una electrónica de Navegación y Control diseñada ex profeso para el mismo. El barco mide unos 80 cm de eslora y unos 35 cm de manga. El barco cuenta con un motor fuera borda que puede moverse utilizando un servo tradicional de radio control. La hélice del motor se controla utilizando un variador electrónico. Ambos sistemas se controlan con señales *Pulse-width Modulation* (PWM) en el rango $[0, 255]$.

La electrónica de a bordo cuenta con distintos sensores que serán las entradas al sistema de Navegación:

- Una brújula que permite medir la orientación del barco.
- Un sensor de presión conectado a un tubo de Pitot. Utilizando este sensor se puede obtener una medida de la velocidad del barco con respecto al agua.
- Sensor de GPS para posicionamiento global.

Adicionalmente, cuenta con un sistema de radio que permite interactuar con una estación de tierra y con otros barcos. Más detalles sobre el desarrollo de estos sistemas se pueden encontrar en (Pereda, 2010) y (Pereda et al., 2010). La Figura 3.4a muestra una fotografía del barco experimental durante uno de los experimentos, y la Figura 3.4b muestra el despliegue de los distintos elementos.

En el caso de este Trabajo, los barcos están equipados con un motor fuera borda, de modo que la ecuación (3.7) puede usarse como modelo para $\boldsymbol{\tau}_{\text{prop}}$. El modelo es equivalente al de un *waterjet*, y este trabajo se refiere siempre a este último tipo de propulsor. El término $F_{\text{wj}}(\mathbf{v})$ corresponde al empuje de los *waterjets* que deberá ser determinado analítica o experimentalmente:

$$\boldsymbol{\tau}_{\text{prop}} = F_{\text{wj}}(\mathbf{v}) \begin{bmatrix} \cos \theta_{\text{wj}} \\ \sin \theta_{\text{wj}} \\ l \cdot \sin \theta_{\text{wj}} \end{bmatrix} \quad (3.7)$$

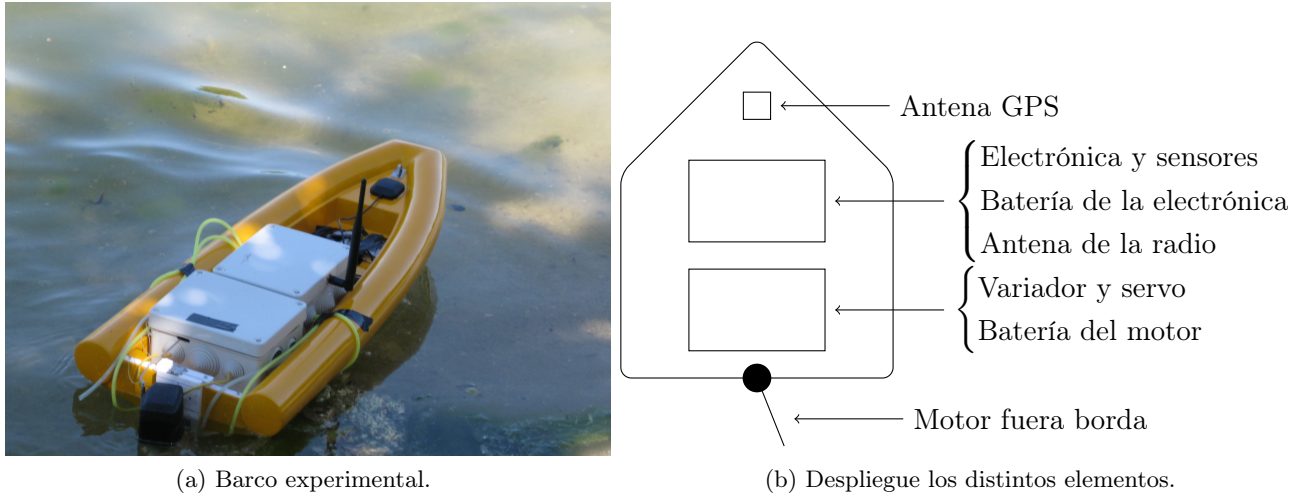


Figura 3.4: Sistema experimental: despliegue.

donde θ_{wj} corresponde al ángulo de los *waterjets* o del motor fuera borda y l es la distancia entre el centro de masas del barco y el punto de montaje del sistema de propulsión.

En general, los procesos existentes para determinar \mathbf{M} y $\mathbf{D}(\mathbf{v})$ son muy costosos tanto en tiempo como en recursos. Es por esto que, en la literatura es común ver algunas simplificaciones al modelo. En el caso de este Trabajo, se han hecho las siguientes:

- Se asume simetría total del barco. Es decir, se asume que el centro de masas del barco es el centro geométrico. La simetría en el eje babor-estribor es bastante común y aceptable. Aunque asumir la simetría del eje popa-proa es típico, por ejemplo en (Behal et al., 2002), algunas fuentes como (Breivik, 2003) la rechazan por completo. Sin embargo, en el caso de los modelos que se manejan en este Trabajo, se ha encontrado que es posible asumir tal simetría. De este modo, se ha decidido utilizar la siguiente matriz de masas:

$$\mathbf{M} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \quad (3.8)$$

donde m es la masa del barco, e I es su momento de inercia. Ambos valores son fáciles de determinar o estimar.

- Partiendo de la Ecuación (3.6) y, sustituyendo los m_{ij} con los valores de la Ecuación (3.8) se obtiene

$$\mathbf{C}(\mathbf{v}) = m \begin{bmatrix} 0 & 0 & -v \\ 0 & 0 & u \\ v & -u & 0 \end{bmatrix} \quad (3.9)$$

que es la forma matricial que se utiliza en este Trabajo. Puede comprobarse que, como es de esperar, en el caso de un barco que se mueve en línea recta $v = r = 0$, los términos de $\mathbf{C}(\mathbf{v})\mathbf{v}$ no contribuyen nada.

- Para los modelos del Trabajo se ha decidido modelar el amortiguamiento hidrodinámico del barco de la siguiente forma

$$\mathbf{D}(\mathbf{v}) = \begin{bmatrix} \mu_u(u) & 0 & 0 \\ 0 & \mu_v(v) & 0 \\ 0 & 0 & \mu_r(r) \end{bmatrix} \quad (3.10)$$

Experimentalmente, en (Pereda, 2010), se obtuvieron las expresiones para $\mu_*(\cdot)$:

$$\mu_u(u) = 4.49 \cdot u + 0.36 \quad (3.11)$$

$$\mu_v(v) = 14.81 \cdot v + 1.18 \quad (3.12)$$

$$\mu_r(r) = 0.5 \quad (3.13)$$

En (Pereda, 2010) también se obtuvo un modelo del propulsor del barco experimental, la expresión se recoge en la Ecuación (3.14):

$$\rho(n) = \begin{cases} 0 & \text{si } n < 30 \\ 0.087 \cdot n - 2.57 & \text{si } n \in [30, 200] \\ 14.94 & \text{si } n > 200 \end{cases} \quad (3.14)$$

Donde $n \in [0, 255]$ es la señal que recibe el variador del motor eléctrico. Adicionalmente, el ángulo de los *waterjet* puede modelarse utilizando la siguiente relación:

$$\theta_{\text{wj}}(d) = \frac{4}{9}(d - 115) \quad (3.15)$$

Adicionalmente, para hacer las maniobras más suaves se imponen las siguientes restricciones:

$$n \in [0, 200] \text{ unidades} \quad (3.16)$$

$$\dot{n} \in [-40, 10] \text{ unidades/s} \quad (3.17)$$

$$\theta_{\text{wj}} \in [-20, 20] \text{ grados} \quad (3.18)$$

$$\dot{\theta}_{\text{wj}} \in [-5, 5] \text{ grados/s} \quad (3.19)$$

Para completar los parámetros del modelo y poder realizar las simulaciones presentadas en el Trabajo se incluyen los parámetros físicos del barco experimental en la Tabla 3.1.

Tabla 3.1: Datos físicos del barco experimental.

Parámetro	Símbolo	Valor
Masa	m	4.8 kg
Distancia del propulsor al centro de masas	l	0.5 m
Momento de inercia	I	$1/12 \cdot m \cdot l^2 \text{ kg m}^2$

Combinando los modelos presentados en la sección es posible simular maniobras de barcos.

Capítulo 4

El problema de la asignación y el algoritmo de subastas

Este capítulo describe uno de los problemas que se presentan a la hora de cumplir con los objetivos del Trabajo. Cuando se hace un cambio de formación entre los barcos, es necesario asignar a cada barco una posición en la nueva formación. En caso de que varios barcos puedan ocupar distintas posiciones indistintamente es importante optimizar la decisión basándose en alguna función de coste.

Este problema se conoce como el problema de la asignación. El capítulo describe formalmente tanto el problema como el algoritmo utilizado para resolverlo.

4.1. El problema de la asignación

El problema de la asignación es uno de los problemas fundamentales de optimización combinatoria en las ramas de optimización e investigación operativa. Consiste en encontrar la asignación que maximiza los pesos de un grafo pesado.

Formalmente, dados dos conjuntos A y T , así como una función de pesos $C : A \times T \rightarrow \mathbb{R}$. Encontrar una función biyectiva $f : A \rightarrow T$ tal que la función de coste

$$J = \sum_{a \in A} C(a, f(a)) \quad (4.1)$$

es minimizada. En el caso de que $|A| = |T| = n$ se dice que el problema es simétrico. En el caso de tener más agentes que tareas, o viceversa, se dice que el problema es asimétrico. En esta sección se describe el caso simétrico.

Existen muchas técnicas para resolver el problema de la asignación. El algoritmo más típico es el llamado *algoritmo Húngaro* publicado por primera vez en (Kuhn, 1955).

Es posible reescribir el problema de la asignación como un problema de programación lineal donde la función objetivo es:

$$f_o = \min_{x_{ij}} \sum_{i \in A} \sum_{j \in T} C(i, j) x_{ij} \quad (4.2)$$

sujeto a las siguientes restricciones:

$$\sum_{j \in T} x_{ij} = 1, \text{ para } i \in A \quad (4.3)$$

$$\sum_{i \in A} x_{ij} = 1, \text{ para } j \in T \quad (4.4)$$

$$x_{ij} \in \{0, 1\} \text{ para } i, j \in A, T \quad (4.5)$$

Donde la matriz x es la matriz de asignación. Cada elemento x_{ij} tendrá valor 1 si el agente i ha sido asignado a la tarea j . Las restricciones aseguran que cada agente tenga asignada una única tarea, y que cada tarea esté asignada a un único agente.

En este Trabajo se ha decidido utilizar un algoritmo denominado *de subasta* para resolver el problema de la asignación.

4.2. El algoritmo de subasta

El algoritmo de subasta fue descrito por primera vez en (Bertsekas, 1979). El mismo autor lo extendió y explicó en más detalle en (Bertsekas, 1989). El algoritmo resuelve el problema de la asignación mediante una metáfora económica: una subasta. El algoritmo es capaz de resolver el problema simétrico y el asimétrico; sin embargo, la formulación presentada en el Trabajo se centra en el caso simétrico. No obstante, la adaptación al caso asimétrico es muy sencilla.

A lo largo de la presente sección asumiremos que los costes vienen dados como una matriz cuadrada de valores reales, es decir $C(i, f(i)) = c_{ij} \in \mathbb{R}^{n \times n}$. Y la asignación es un vector $s \triangleq \{s_i\} \in \mathbb{R}^n$ que contiene qué tarea tiene asignada cada agente.

En lugar de tener una matriz de coste que se debe minimizar, se tendrá una matriz de beneficios que hay que maximizar. Es decir, $b_{ij} \triangleq -c_{ij}$.

Es necesario introducir el concepto de *precio* mediante un vector de precios $\mathbf{p} \triangleq \{p_j\} \in \mathbb{R}^n$. Intuitivamente, se trata del precio que debe pagarse por asignar la tarea j a cualquier agente. En el caso de asignar agentes (barcos) a tareas (posiciones en una formación) el concepto de precio no tiene equivalente físico. Sin embargo, es necesario para el funcionamiento del algoritmo.

Decimos que una asignación proporciona un beneficio neto que es el resultado de sustraer del beneficio, el precio de la misma $b_{ij}^* \triangleq b_{ij} - p_j$. Concretando, el algoritmo busca la asignación que proporciona mayor beneficio neto.

Diremos que un agente i es *feliz* si se cumple que:

$$b_{i,s_i}^* = \max_j \{b_{ij}^*\} \quad (4.6)$$

es decir, si ninguna otra asignación le hace, a él, más feliz. El objetivo del algoritmo es hacer que todos los agentes sean *felices*. En ese punto, s representa la asignación óptima.

4.2.1. Algoritmo

El algoritmo se ejecuta por rondas y tiene los siguientes pasos:

- Si todos los agentes son *felices*, el algoritmo para.
- En caso contrario, se selecciona uno de los agentes *infelices*. El agente i es *infeliz* porque ha encontrado una tarea j que le proporciona un mayor beneficio neto, es decir, que no satisface la Ecuación (4.6). De modo que:
 - Intercambia tareas con aquel que tuviera asignada la tarea j .
 - Puja por la tarea j de forma que siga siendo la tarea que más beneficio neto le proporciona:

$$p'_j = p_j + \gamma_i \quad (4.7)$$

donde $\gamma_i = v_i - w_i$, v_i es el beneficio neto actual de i y w_i es el beneficio neto que le proporcionaría la segunda mejor asignación:

$$v_i = \max_j \{b_{ij}^*\} \quad (4.8)$$

$$w_i = \max_{j \neq s_i} \{b_{ij}^*\} \quad (4.9)$$

- En este punto, i es *feliz*. El algoritmo se repite hasta que todos los agentes son *felices*.

Debe hacerse notar que $\gamma_i \geq 0$ dado que, por definición, $v_i \geq w_i$. Sin embargo, en caso de que $v_i = w_i \implies \gamma_i = 0$, el algoritmo podría estancarse y no llegar nunca al óptimo. Para solventar este problema, es necesario introducir un factor de relajación en el problema. Se redefinen los siguientes conceptos utilizando el factor de relajación ε :

- Diremos que i es *casi feliz* si:

$$b_{i,s_i}^* = \max_j \{b_{ij}^*\} - \varepsilon \quad (4.10)$$

- La puja por una tarea en una de las rondas del algoritmo:

$$\gamma_i = v_i - w_i + \varepsilon \quad (4.11)$$

Si $\varepsilon > 0$ entonces el algoritmo obtendrá una asignación \mathbf{s} que estará en un entorno de radio $n \cdot \varepsilon$ con centro en la solución óptima. En (Bertsekas, 1989) se puede encontrar el razonamiento y la demostración, pero si los beneficios son números enteros y $\varepsilon < \frac{1}{n-1}$ entonces la solución será óptima.

4.2.2. Coste computacional

El número de rondas necesario para que el algoritmo llegue a una asignación óptima depende tanto de ε como del mayor valor absoluto de los beneficios:

$$C = \max_{i,j} |b_{ij}| \quad (4.12)$$

De hecho, tiende a ser proporcional a $\frac{C}{\varepsilon}$. Además, el número de rondas depende mucho de la asignación inicial así como del vector de precios iniciales. Si estos están cerca del óptimo, el algoritmo converge mucho más rápido. En (Bertsekas, 1989) se ofrecen distintas soluciones para hacer que el algoritmo converja razonablemente rápido en situaciones prácticas.

En este Trabajo se ha elegido la técnica de ε -relajación. Los beneficios se escalan por $n + 1$ y se resuelve el algoritmo iterativamente con valores cada vez más pequeños de ε . Dado que los beneficios se han escalado por $n + 1$, para que la solución sea óptima es suficiente con que $\varepsilon \leq 1$. La secuencia de valores utilizada es:

$$\varepsilon(k) = \max \left(1, \frac{\Delta}{\theta^k} \right) \quad (4.13)$$

donde k representa las ejecuciones del algoritmo. Se empieza con $k = 0$ y se resuelve la subasta hasta el óptimo. El resultado de la subasta se utiliza como estado inicial para la ejecución en $k = 1$ y así sucesivamente hasta que $\varepsilon \leq 1$. Siguiendo los criterios presentados en (Bertsekas, 1989), en este Trabajo se han elegido los valores $\Delta = 4$ y $\theta = 7$.

4.2.3. Ejemplo del algoritmo

Para ilustrar el funcionamiento del algoritmo se incluye un ejemplo de aplicación en el contexto del Trabajo. Para simplificar los resultados y la operativa del proceso se utiliza $\varepsilon = 0$ y no se utiliza la técnica de ε -relajación.

El ejemplo trata de asignar los barcos a la mejor posición dentro de una formación dada una cierta métrica. En este caso, para simplificar el ejemplo, la métrica considerada es la distancia euclídea. El objetivo es asignar los huecos

$$\text{hueco}_1 = \begin{bmatrix} 20 \\ 0 \end{bmatrix} \quad \text{hueco}_2 = \begin{bmatrix} -10 \\ 0 \end{bmatrix} \quad \text{hueco}_3 = \begin{bmatrix} -20 \\ 0 \end{bmatrix} \quad \text{hueco}_4 = \begin{bmatrix} 20 \\ 0 \end{bmatrix} \quad (4.14)$$

a los barcos en las siguientes posiciones

$$\text{barco}_a = \begin{bmatrix} -25 \\ 15 \end{bmatrix} \quad \text{barco}_b = \begin{bmatrix} -13 \\ 20 \end{bmatrix} \quad \text{barco}_c = \begin{bmatrix} -15 \\ -15 \end{bmatrix} \quad \text{barco}_d = \begin{bmatrix} 5 \\ 0 \end{bmatrix} \quad (4.15)$$

La Figura 4.1 muestra la configuración inicial, utilizando una asignación aleatoria que sirve de partida para el comienzo del algoritmo. Los huecos están caracterizados por los símbolos círculo, triángulo, aspa y asterisco; mientras que los barcos están representados por colores. La asignación se muestra dibujando cada hueco, además, del color del barco al que ha sido asignado.

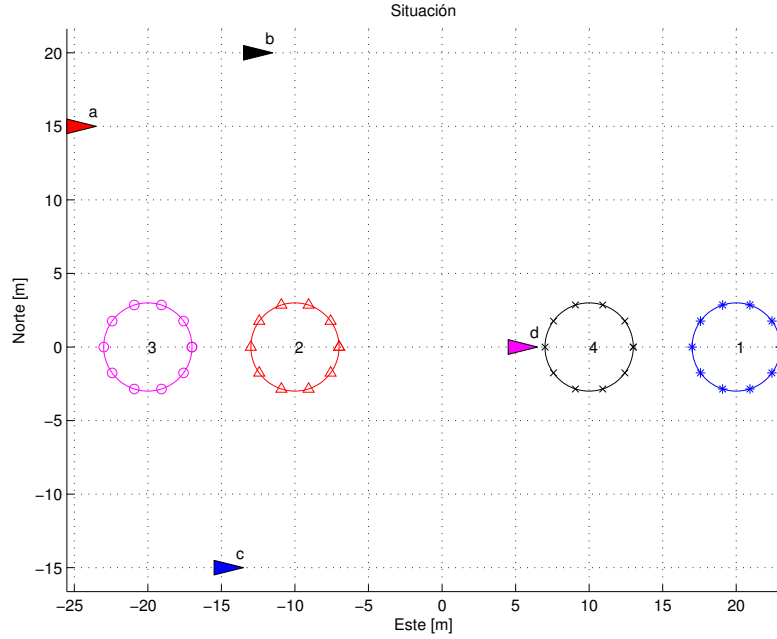


Figura 4.1: Configuración inicial del algoritmo.

El algoritmo comienza calculando la matriz de beneficios utilizando la métrica deseada. La matriz de beneficios es, por tanto, la distancia euclídea de cada uno de los barcos a cada uno de los huecos,

$$b = \begin{bmatrix} -47.4342 & -21.2132 & -15.8114 & -38.0789 \\ -38.5876 & -20.2237 & -21.1896 & -30.4795 \\ -38.0789 & -15.8114 & -15.8114 & -29.1548 \\ -15.0000 & -15.0000 & -25.0000 & -5.0000 \end{bmatrix} \quad (4.16)$$

La asignación inicial se decide aleatoriamente, y los pesos se inicializan a 0. Para este ejemplo:

$$s^{(0)} = [2 \quad 4 \quad 1 \quad 3] \quad (4.17)$$

$$p^{(0)} = [0 \quad 0 \quad 0 \quad 0] \quad (4.18)$$

Con los valores iniciales de s , p , y con la matriz b , se inician las rondas del algoritmo:

- Se encuentra que el barco a no es feliz. Y que sería más feliz con el hueco 3 que con el 2. Así que intercambia su hueco con el barco que tiene el hueco 3, que es el barco d . Automáticamente, puja por el hueco 3, para desincentivar al resto, $\gamma = 5.4018$, de forma que los precios y la asignación quedan:

$$s^{(1)} = [3 \quad 4 \quad 1 \quad 2] \quad (4.19)$$

$$p^{(1)} = [0 \quad 0 \quad 5.4018 \quad 0] \quad (4.20)$$

- En la siguiente ronda a es feliz por definición, sin embargo, b no es feliz; quiere el hueco 2. De modo que lo intercambia y puja por él usando $\gamma = 6.3677$. Los precios y la asignación en este punto quedan de la siguiente forma:

$$s^{(2)} = [3 \quad 2 \quad 1 \quad 4] \quad (4.21)$$

$$p^{(2)} = [0 \quad 6.3677 \quad 5.4018 \quad 0] \quad (4.22)$$

- Ahora es c , que tiene asignado el hueco 1, quien no es feliz. El barco c sería feliz si le asignaran el hueco 3 (o el 2). De modo que lo intercambia con su actual propietario, a , y puja por él. En este caso, puja usando $\gamma = 0.9659$. La asignación y precios quedan como sigue:

$$s^{(3)} = [1 \quad 2 \quad 3 \quad 4] \quad (4.23)$$

$$p^{(3)} = [0 \quad 6.3677 \quad 6.3677 \quad 0] \quad (4.24)$$

- Dado que en la iteración anterior se hizo feliz a c quitándole su asignación a a , a ahora no es feliz. Sigue queriendo el hueco 3, de forma que puja por él: $\gamma = 5.4018$. La asignación y precios son ahora:

$$s^{(4)} = [3 \quad 2 \quad 1 \quad 4] \quad (4.25)$$

$$p^{(4)} = [0 \quad 6.3677 \quad 11.7695 \quad 0] \quad (4.26)$$

- Debido a que tanto a como c están muy interesados en el hueco 3, el precio de este va a subir mucho; ya que ninguno de los dos está dispuesto a renunciar a él. El algoritmo converge en un total de 17 rondas.

Al final, el algoritmo converge a:

$$s^{(16)} = [3 \quad 1 \quad 2 \quad 4] \quad (4.27)$$

$$p^{(16)} = [0 \quad 22.2675 \quad 23.7656 \quad 10.0000] \quad (4.28)$$

El estado final está representado en la Figura 4.2. La evolución del vector de asignación s se representa en la Figura 4.3, mientras que la evolución de los precios se muestra en la Figura 4.4. Puede verse cómo el precio de los dos huecos más cercanos a los tres barcos de la izquierda crece. La intuición económica de este hecho es que a mucha demanda y poca oferta, el precio crece. De la misma forma, nadie está realmente interesado en el hueco 1, ya que queda bastante lejos de todos los barcos; esto queda contrastado con el hecho de que nadie puja por el hueco 1. Asimismo el hueco 4 es atractivo para todos los barcos, pero sobre todo, es atractivo para el barco d . En el momento en el que el algoritmo identifica d como infeliz, éste puja por el hueco 4; la puja es tan alta que hace que ningún barco vuelva a interesarse por el hueco 4.

El ejemplo ilustra los símiles y analogías económicas y de mercado presentes en el algoritmo. Los conceptos de oferta y demanda quedan patentes en la gráfica de evolución de precios (Figura 4.4). El algoritmo tiene la particularidad de solucionar tanto el problema primal como el dual, es decir, maximiza beneficios a la vez que minimiza los precios. Por completitud, se incluye la matriz de beneficios netos b^* donde puede observarse que el algoritmo ha hecho felices a todos los barcos.

$$b^* = \begin{bmatrix} -47.4342 & -43.4807 & -39.5770 & -48.0789 \\ -38.5876 & -42.4912 & -44.9552 & -40.4795 \\ -38.0789 & -38.0789 & -39.5770 & -39.1548 \\ -15.0000 & -37.2675 & -48.7656 & -15.0000 \end{bmatrix} \quad (4.29)$$

La matriz de asignación correspondiente al vector $s^{(16)}$ que resuelve el problema de programación lineal es, por tanto,

$$x_{ij} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.30)$$

que respeta las restricciones del problema (4.3), (4.4) y (4.5).

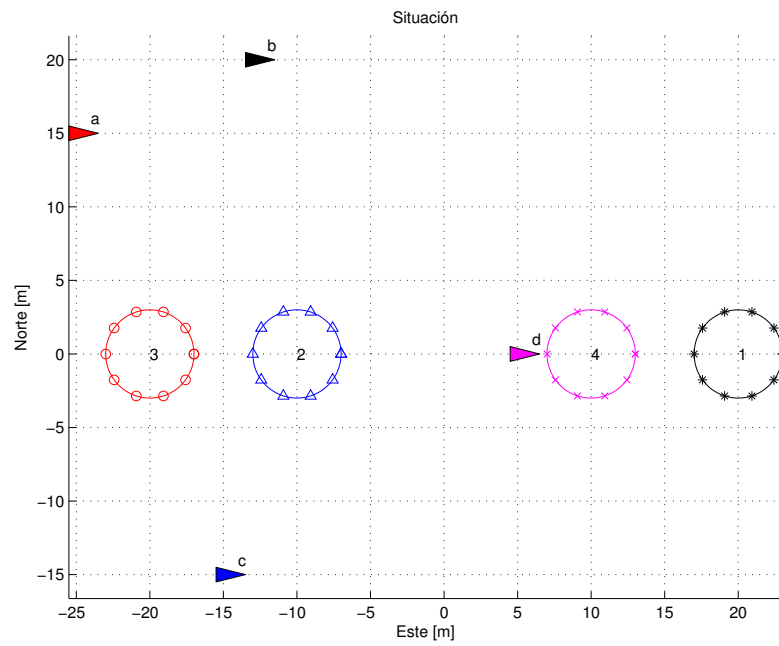


Figura 4.2: Configuración final del algoritmo.

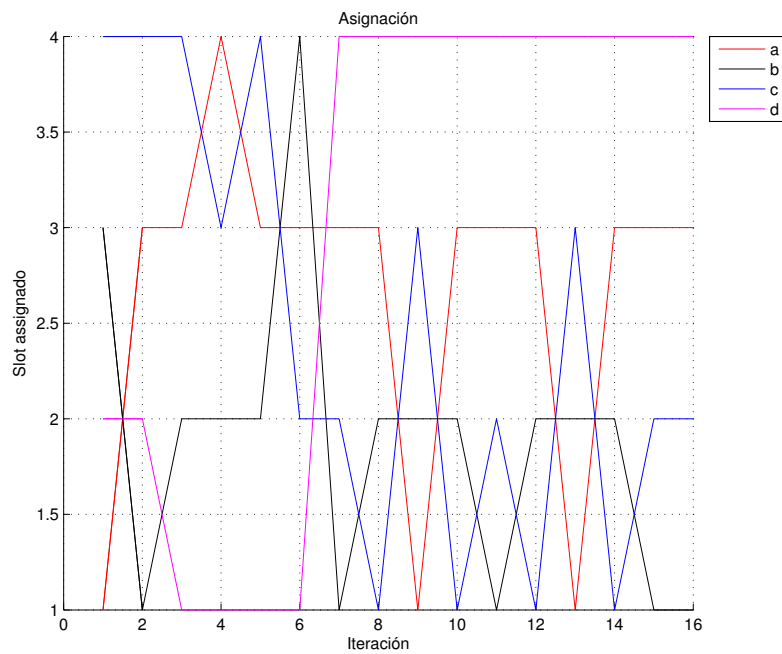


Figura 4.3: Evolución del vector asignación.

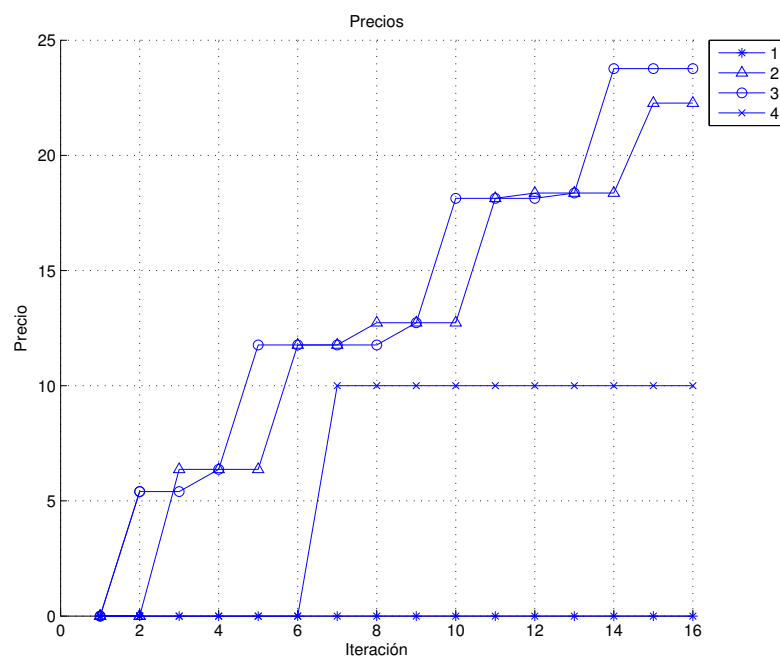


Figura 4.4: Evolución del vector de precios.

Capítulo 5

El filtro de Kalman *Unscented*

Este capítulo presenta la formulación matemática y los algoritmos utilizados para la estimación de la posición del barco entre cada medida GPS utilizando una técnica muy conocida de fusión sensorial llamada Filtro de Kalman.

El capítulo comienza con la descripción de la *Unscented Transform* (UT) para después describir el algoritmo UKF para tiempo discreto.

5.1. La transformada *Unscented*

La UT es un método para calcular la estadística de una variable aleatoria sobre la que se aplica una transformación no lineal (Julier and Uhlmann, 1997). Consideramos la propagación de un vector $\mathbf{x} \in \mathbb{R}^L$ a través de una función no lineal $\mathbf{y} = f(\mathbf{x})$. Se asume que \mathbf{x} tiene media $\bar{\mathbf{x}}$ y covarianza $\mathbf{P}_{\mathbf{x}} \in \mathbb{R}^{L \times L}$. Para calcular la estadística de \mathbf{y} se crea una matriz $\mathcal{X} \in \mathbb{R}^{L \times 2L+1}$ de *puntos sigma*:

$$\begin{aligned}\mathcal{X}_0 &= \bar{\mathbf{x}} \\ \mathcal{X}_i &= \bar{\mathbf{x}} + \left(\sqrt{(L + \lambda) \mathbf{P}_{\mathbf{x}}} \right)_i & i = 1, \dots, L \\ \mathcal{X}_i &= \bar{\mathbf{x}} - \left(\sqrt{(L + \lambda) \mathbf{P}_{\mathbf{x}}} \right)_{i-L} & i = L + 1, \dots, 2L\end{aligned}$$

donde $\lambda = \alpha^2(L + \kappa) - L$ es un parámetro de escala. La constante α determina la dispersión de los *puntos sigma* alrededor $\bar{\mathbf{x}}$ con un valor positivo (típicamente $1 \cdot 10^{-4} \leq \alpha \leq 1$). La constante κ es otro parámetro de escala, usualmente 0 o $3 - L$; para más información, ver (Wan and van der Merwe, 2001). El término $\left(\sqrt{(L + \lambda) \mathbf{P}_{\mathbf{x}}} \right)_i$ corresponde a la columna *i-ésima* de la raíz cuadrada de la matriz $\sqrt{(L + \lambda) \mathbf{P}_{\mathbf{x}}}$. Esta raíz puede calcularse, por ejemplo, utilizando la factorización de Cholesky.

Estos *puntos sigma* son propagados a través de la función no lineal,

$$\mathcal{Y}_i = f(\mathcal{X}_i) \quad i = 0, \dots, 2L \quad (5.1)$$

y la media y la covarianza para \mathbf{y} son aproximadas utilizando una media ponderada y la covarianza de los *puntos sigma* propagados,

$$\begin{aligned}\bar{\mathbf{y}} &\approx \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Y}_i \\ \mathbf{P}_{\mathbf{y}} &\approx \sum_{i=0}^{2L} W_i^{(c)} (\mathcal{Y}_i - \bar{\mathbf{y}}) (\mathcal{Y}_i - \bar{\mathbf{y}})^T\end{aligned} \quad (5.2)$$

con pesos W_i dados por

$$\begin{aligned}W_0^{(m)} &= \lambda / (L + \lambda) \\ W_0^{(c)} &= \lambda / (L + \lambda) + (1 - \alpha^2 + \beta) \\ W_i^{(m)} &= W_i^{(c)} = 1 / (2(L + \lambda)) \quad i = 1, \dots, 2L\end{aligned} \quad (5.3)$$

La constante β puede utilizarse para añadir conocimiento *a priori* sobre la distribución de \mathbf{x} . Para distribuciones Gaussianas, según (Wan and van der Merwe, 2001), $\beta = 2$ es óptimo.

El uso de la UT muestra unos resultados equivalentes a una aproximación de tercer orden para entradas Gaussianas a un sistema no lineal. Para entradas no Gaussianas, la UT es, al menos, tan precisa como una aproximación de segundo orden, y con una correcta selección de α y β es posible llegar a órdenes incluso superiores a tres (Wan and van der Merwe, 2001).

Un ejemplo de este proceso para un vector de estados de dos dimensiones es mostrado en la Figura 5.1. La gráfica izquierda muestra la media y covarianza verdadera después del procesado no lineal; la gráfica central hace referencia a la aproximación lineal como puede ser un EKF; la gráfica de la derecha finalmente muestra el rendimiento de aplicar la UT. Dado que se trata de un vector de dos variables, se utilizan cinco *puntos sigma*. Claramente, el rendimiento de la UT es superior.

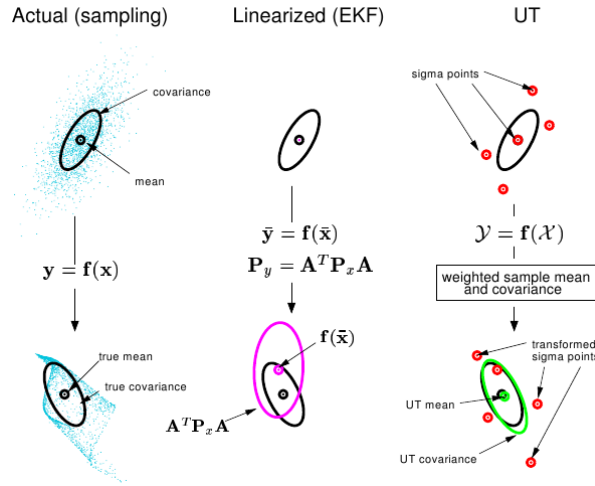


Figura 5.1: Ejemplo de la aplicación de la UT

5.2. UKF para procesos con ruido aditivo

Sea \mathbf{x} un vector de variables aleatorias x_k , en un UKF el vector de estados queda definido por $x_k^a = [x_k^T \ v_k^T \ n_k^T]^T$. Donde v_k y n_k son los ruidos de proceso y medida respectivamente. Hay que destacar que, contrariamente al EKF, no existe en el proceso ningún cálculo de Jacobianos o Hessianos, de forma que el UKF es mucho más fácil de implementar.

En el caso de que el ruido en el proceso y la medida sean aditivos, es posible reducir el coste computacional del UKF, ya que no es necesario ampliar el vector de estados con sus ruidos (Wan and van der Merwe, 2001). Por lo tanto, reduce la dimensión del vector de *puntos sigma*. Además, la complejidad del algoritmo en este caso es $O(L^3)$. Incluso es posible hacerlo $O(M \cdot L^2)$ donde M es la dimensión de la salida y_k .

El UKF en tiempo discreto con ruido aditivo se puede describir con los dos pasos comunes en un filtro de Kalman, *predicción* y *corrección*. Estos pasos se describen en las dos siguientes subsecciones.

5.2.1. Paso de predicción

El algoritmo comienza con las siguientes condiciones iniciales:

$$\hat{\mathbf{x}}_0 = E[\mathbf{x}_0] \quad (5.4)$$

$$\mathbf{P}_0 = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T] \quad (5.5)$$

, donde el operador E es el operador valor esperado o esperanza. Para los siguientes instantes de tiempo $k \in \{1, \dots, \infty\}$ se empieza calculando el vector de *puntos sigma*:

$$\mathcal{X}_{k-1} = [\hat{\mathbf{x}}_{k-1} \quad \hat{\mathbf{x}}_{k-1} + \gamma\sqrt{\mathbf{P}_{k-1}} \quad \hat{\mathbf{x}}_{k-1} - \gamma\sqrt{\mathbf{P}_{k-1}}] \quad (5.6)$$

donde $\gamma = \sqrt{(L + \lambda)}$.

El siguiente paso es propagar los *puntos sigma* a través del proceso no lineal:

$$\mathcal{X}_{k|k-1}^* = F[\mathcal{X}_{k-1}, \mathbf{u}_{k-1}] \quad (5.7)$$

donde \mathbf{u}_{k-1} es el vector entrada al modelo del sistema.

La predicción del vector de estados es el cálculo de la media ponderada, resultado de su UT:

$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{X}_{i,k|k-1}^* \quad (5.8)$$

La predicción de la covarianza se calcula a partir de la ponderación entre los *puntos sigma* y la media ponderada.

$$\mathbf{P}_k^- = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{X}_{i,k|k-1}^* - \hat{\mathbf{x}}_k^-][\mathcal{X}_{i,k|k-1}^* - \hat{\mathbf{x}}_k^-]^T + Q \quad (5.9)$$

donde Q es la matriz de covarianza del ruido de proceso.

Esto concluye el paso de propagación.

5.2.2. Paso de corrección

Una vez existen medidas disponibles para corregir el estado predicho, se lleva a cabo el proceso de corrección.

El proceso comienza con un recálculo de los *puntos sigma*. Según (Wan and van der Merwe, 2001) este recálculo es necesario para incorporar el efecto de haber añadido la matriz Q en el paso anterior.

$$\mathcal{X}_{k|k-1} = \left[\hat{\mathbf{x}}_k^- \quad \hat{\mathbf{x}}_k^- + \gamma \sqrt{\mathbf{P}_k^-} \quad \hat{\mathbf{x}}_k^- - \gamma \sqrt{\mathbf{P}_k^-} \right] \quad (5.10)$$

Se continúa con la UT de los *puntos sigma* con el observador H deseado.

$$\mathcal{Y}_{k|k-1} = H[\mathcal{X}_{k|k-1}] \quad (5.11)$$

$$\hat{\mathbf{y}}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Y}_{i,k|k-1} \quad (5.12)$$

Para el cálculo de la matriz de ganancia de Kalman \mathcal{K} , además de la matriz de covarianza de la observación, es necesario el cálculo de la matriz de covarianzas cruzadas entre el estado y la observación.

$$\mathbf{P}_{\hat{\mathbf{y}}_k \hat{\mathbf{y}}_k} = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_k^-][\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_k^-]^T + R \quad (5.13)$$

$$\mathbf{P}_{\mathbf{x}_k \mathbf{y}_k} = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{X}_{i,k|k-1} - \hat{\mathbf{x}}_k^-][\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{y}}_k^-]^T \quad (5.14)$$

$$\mathcal{K}_k = \mathbf{P}_{\mathbf{x}_k \mathbf{y}_k} \mathbf{P}_{\hat{\mathbf{y}}_k \hat{\mathbf{y}}_k}^{-1} \quad (5.15)$$

donde R es la matriz de covarianza del ruido de medida.

Para finalizar, se corrige el vector de estados y su covarianza con la ganancia de Kalman \mathcal{K} :

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathcal{K}_k (\mathbf{y}_k - \hat{\mathbf{y}}_k^-) \quad (5.16)$$

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathcal{K}_k \mathbf{P}_{\hat{\mathbf{y}}_k \hat{\mathbf{y}}_k} \mathcal{K}_k^T \quad (5.17)$$

Esto concluye el paso de corrección del UKF.

Capítulo 6

Implementación del sistema

En este punto, todo lo necesario para implementar el sistema ha sido descrito. El Capítulo 3 (Modelado matemático) ha descrito el modelo matemático del barco y los actuadores. Los capítulos 4 (El problema de la asignación y el algoritmo de subastas) y 5 (El filtro de Kalman *Unscented*) han descrito las herramientas necesarias para el subsistema de Navegación.

La Figura 6.1 muestra las distintas partes del sistema. El primer sistema que se describe es el de Filtrado. Luego se describe el diseño del Estimador, basado en el UKF. El capítulo continúa con una descripción de los sistemas de Guiado, Navegación y Control.

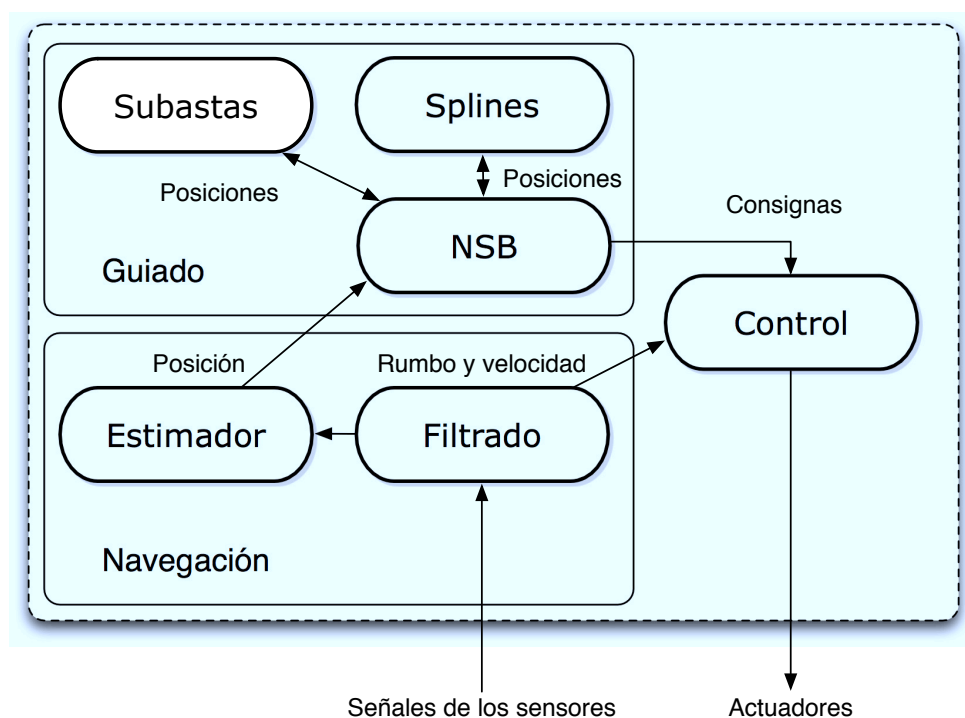


Figura 6.1: Diagrama de la parte a implementar del sistema.

Como muestra la figura, el sistema tiene una entrada, que son las señales de los distintos sensores, y una salida, que son las señales que reciben los actuadores. El Trabajo desarrolla estas interfaces tanto desde el punto de vista analítico (simulación) como experimental en posteriores capítulos.

6.1. Navegación

El sistema de Navegación tiene como responsabilidad principal localizar, situar y/o estimar el estado del barco en el entorno. Para llevar a cabo su tarea utiliza las señales de los sensores así como algoritmos y conocimiento experto del sistema. En concreto, el sistema debe medir y estimar el estado del barco (posición y orientación) para alimentar tanto al sistema de Guiado como al de Navegación.

La primera parte de la sección está dedicada al filtrado de las señales. Luego, sigue una subsección dedicada al estimador de posición utilizando UKF.

6.1.1. Filtrado

Las señales recibidas de los sensores tienen bastante ruido, de forma que la primera de las tareas del sistema de navegación es filtrar estas señales. Los sensores considerados son una brújula y un sensor de presión para medir la velocidad con respecto al agua. La brújula obtiene la medida del campo magnético en los tres ejes del barco, de modo que la medida de la orientación puede obtenerse como

$$\hat{\psi} = \text{atan2} \left(\frac{m_y}{m_x} \right) \quad (6.1)$$

donde m_x y m_y son las medidas del sensor en los ejes X_{BARCO} e Y_{BARCO} , y atan2 es la versión en cuatro cuadrantes de la función arcotangente. A partir de la medida del sensor de velocidad Δp podemos obtener una medida de la velocidad de avance del barco \hat{u} utilizando

$$\Delta p = \frac{1}{2} \cdot \rho \cdot \hat{u}^2 \implies \hat{u} = \sqrt{\frac{2 \cdot \Delta p}{\rho}} \quad (6.2)$$

donde Δp es la presión de estancamiento leída por el sensor y $\rho \approx 1000$ es la densidad del agua.

Dado que los sensores tienen ruido, es necesario hacer algún tipo de filtrado sobre las señales obtenidas. La Figura 6.2 muestra los valores de estos sensores antes del filtrado. Se puede observar que el ruido es de alta frecuencia y que, además, tiene un cierto carácter impulsivo. En el caso de la Figura 6.2b el ruido impulsivo ha sido truncado para poder apreciar el resto de la señal. De forma que interesa un filtro paso baja que, además, elimine el ruido impulsivo.

Uno de los problemas a la hora de filtrar señales es que es posible mermar la acción derivativa del control, de forma que el filtro debe filtrar bien las altas frecuencias y el ruido impulsivo pero que no *engañe* mucho al control. El filtro no lineal de la mediana elimina las altas frecuencias y, además, es capaz de eliminar el ruido impulsivo. Adicionalmente, no desvirtúa mucho la acción derivativa del control (Li et al., 2006). Por tanto, se ha decidido mantener el mismo filtro utilizado por el autor en (Pereda, 2010), un filtro de la mediana de 6 posiciones. Las figuras 6.3a y 6.3b muestran el resultado de filtrar las señales anteriores.

Las señales obtenidas de los sensores conforman los datos con los que se alimenta el estimador de posición descrito a continuación así como el sistema de Control.

6.1.2. Estimador

El estimador tiene como objetivo proporcionar estimaciones razonables de la posición del barco entre las distintas medidas GPS. Los GPS, generalmente, dan medidas de posición cada

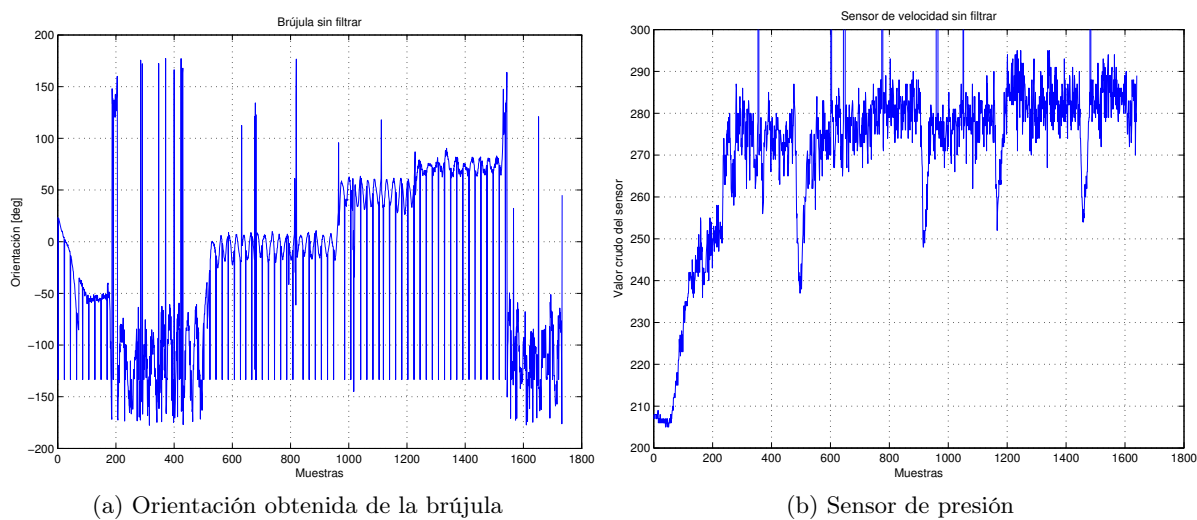


Figura 6.2: Sensores sin filtrar

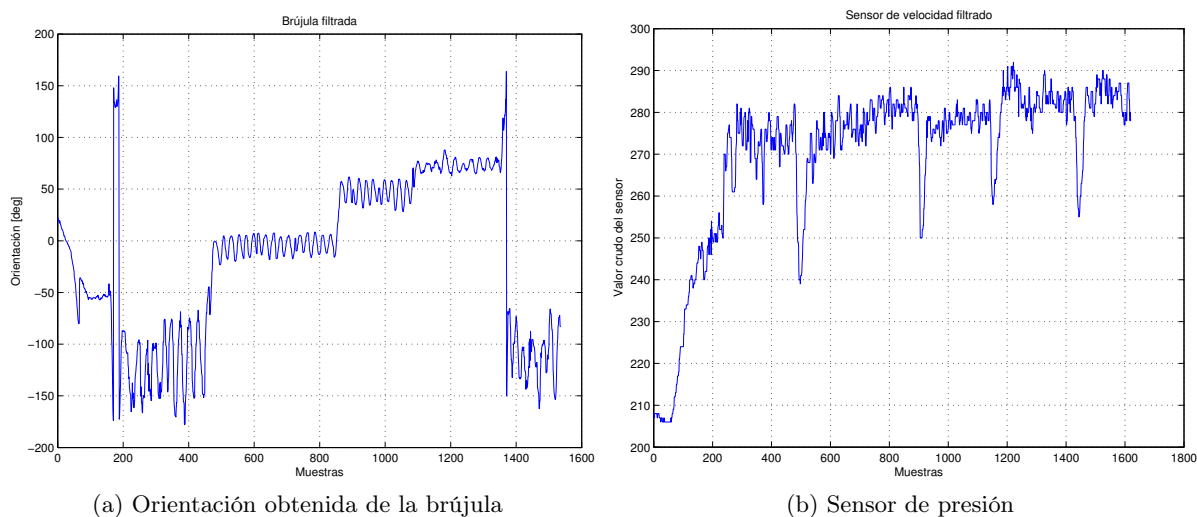


Figura 6.3: Sensores filtrados

segundo. Entre dos medidas, es necesario estimar la posición utilizando las medidas de los sensores disponibles. En el caso de un barco, se utiliza la orientación y la velocidad de avance para estimar el estado. La Figura 6.4 muestra el diagrama de funcionamiento del estimador.

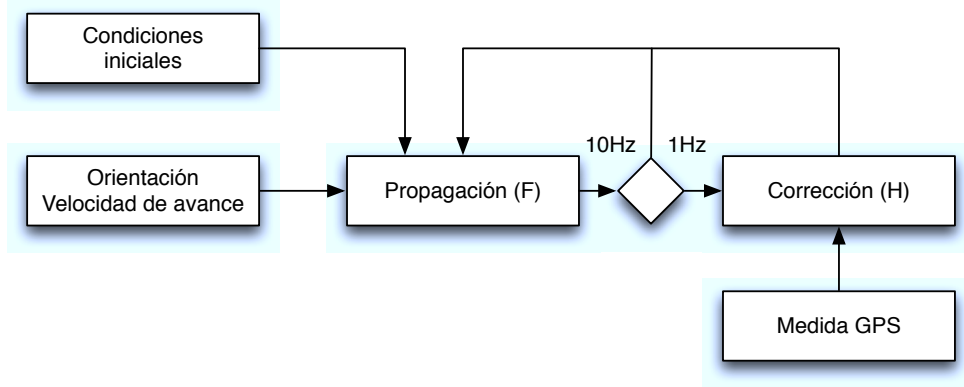


Figura 6.4: Diagrama del algoritmo estimador.

El algoritmo utilizado para la estimación es el *Unscented Kalman Filter*, descrito en el Capítulo 5 (El filtro de Kalman *Unscented*). En el diagrama de la Figura 6.4 pueden verse los pasos de *Propagación* y *Corrección* descritos en dicho capítulo. Cada uno de ellos se ejecuta en un bucle a distinta frecuencia. Faltan por definir, sin embargo, el vector de estados \mathbf{x}_k , la función de propagación $F[\mathbf{x}, \mathbf{u}]$ y la función de observación $H[\mathbf{x}]$.

El vector de estados está compuesto de las posiciones Este y Norte del barco en el sistema de coordenadas NED:

$$\mathbf{x}_k \triangleq [n \ e]^T \quad (6.3)$$

Como se muestra en la Figura 6.4, la función de propagación recibe como entradas las medidas de orientación y velocidad de avance. Es decir, el vector \mathbf{u} de entradas al sistema es:

$$\mathbf{u}_k \triangleq [\hat{u}_k \ \hat{\psi}_k]^T \quad (6.4)$$

donde \hat{u} es la medida de la velocidad de avance y $\hat{\psi}$ es la medida de la orientación. De modo que la función de propagación queda definida de la siguiente forma:

$$F[\mathbf{x}_k, \mathbf{u}_k] = \mathbf{x}_k + \begin{bmatrix} \cos \hat{\psi} & -\sin \hat{\psi} \\ \sin \hat{\psi} & \cos \hat{\psi} \end{bmatrix} \begin{bmatrix} \hat{u} \\ 0 \end{bmatrix} \Delta t \quad (6.5)$$

donde $\Delta t = 0.1s$. La función de propagación no es nada más que una particularización del modelo cinemático del barco, presentado en el Capítulo 3 (Modelado matemático) como la Ecuación 3.2. El modelo asume $\hat{v} = 0$ porque en el barco experimental no existe forma de medir esta velocidad.

El sensor GPS es el que alimenta el paso de corrección. En este paso se corrigen los errores de la estimación debidos tanto a imprecisiones del modelo como errores en las medidas. El vector de observaciones \mathbf{y}_k es, por tanto, directamente la medida del GPS en el sistema de referencia NED.

$$\mathbf{y}_k \triangleq [n \ e]^T \quad (6.6)$$

y la función de observación del sistema devuelve directamente el estado:

$$H[\mathbf{x}_k] = \mathbf{x}_k \quad (6.7)$$

Las matrices de covarianza de los ruidos de proceso y medida deben elegirse utilizando conocimiento experto sobre los distintos procesos. Dado que el modelo del proceso es muy fiable, basándose en los ruidos de \mathbf{u}_k , se ha decidido utilizar

$$\mathbf{Q} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \quad (6.8)$$

El ruido de medida depende, únicamente, del ruido del sensor. El GPS, generalmente, tiene muy buena precisión relativa entre sus medidas; sin embargo, la precisión absoluta es algo menor. De cara a hacer el sistema todo lo robusto posible y no confiar mucho en el GPS, se ha decidido no otorgar mucha confianza al mismo. Por tanto, la covarianza del ruido de medida utilizada es

$$\mathbf{R} = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \quad (6.9)$$

El algoritmo de estimación utilizando UKF queda completamente descrito utilizando las ecuaciones descritas en el Capítulo 5 (El filtro de Kalman *Unscented*) y las descritas en esta sección. Las estimaciones de posición en el sistema de referencia NED de cada uno de los barcos de la flota serán entradas del sistema de Guiado. Este sistema se describe en la siguiente sección.

6.2. Guiado

Es importante recordar el diagrama de despliegue presentado en la Figura 2.3 en el Capítulo 2 (Descripción general del sistema). El sistema de Guiado es único para todo el grupo; puede encontrarse en uno de los barcos del grupo o en un sistema externo, pero es único. Todos los sistemas de Navegación alimentan al único sistema de Guiado, y éste, alimenta a los sistemas de Control de cada uno de los barcos.

El sistema de Guiado tiene como objetivo calcular las referencias de rumbo y velocidad que serán entregadas a los sistemas de Control. El sistema de Guiado es alimentado con la posición y actitud de todos los barcos del grupo por parte de los distintos sistemas de Navegación. El sistema de Guiado debe permitir guiar al grupo a lo largo de una cierta trayectoria, hacer que el grupo tome una cierta forma geométrica y evitar que los barcos choquen. Además, debe permitir hacer cambios en la configuración geométrica del grupo y que esos cambios sean lo tan rápidos y *limpios* como sea posible.

La primera parte de la presente sección está centrada en explicar el formalismo utilizado para describir las trayectorias del centroide de la flota. La segunda parte explica en detalle el control utilizado, que no es más que una aplicación geométrica de la arquitectura por capas de Brooks, presentada en (Brooks, 1986). La tercera sección une el Capítulo 4 (El problema de la asignación y el algoritmo de subastas) con el Trabajo; está dedicado a aplicar los conceptos que subyacen al utilizar la metáfora económica para resolver uno de los problemas planteados en los cambios de formación. Todas las partes tienen, además de la formulación matemática, un ejemplo dirigido para facilitar la introducción de los conceptos que se exponen.

6.2.1. Splines para trayectorias

Existen muchas formas de representar trayectorias. Una de las más comunes en robótica es utilizar algún tipo de curva paramétrica:

$$\begin{bmatrix} x & y \end{bmatrix}^T = c(\varpi) \quad \varpi \in [0, 1] \quad (6.10)$$

En el caso de este Trabajo se ha decidido utilizar curvas de Bézier cúbicas para describir las trayectorias del centroide de la flota. La elección de este tipo de curvas viene motivada por el hecho de que permiten definir la orientación de los puntos iniciales y finales de la curva. Una curva de Bézier genérica viene dada por la siguiente ecuación:

$$\mathbf{C}(\varpi) = \sum_{i=0}^n B_{n,i}(\varpi) \mathbf{P}_i \quad (6.11)$$

donde \mathbf{P}_i son los puntos de control de la curva, y los $B_{n,i}(\varpi)$ son polinomios de Bernstein de la forma

$$B_{n,i}(\varpi) = \binom{n}{i} \varpi^i (1 - \varpi)^{n-i} = \frac{n!}{i!(n-i)!} \varpi^i (1 - \varpi)^{n-i} \quad (6.12)$$

La Figura 6.5 muestra una curva de Bézier cúbica de ejemplo con puntos de control:

$$\mathbf{P}_0 = \begin{bmatrix} 10 \\ 5 \end{bmatrix} \quad \mathbf{P}_1 = \begin{bmatrix} 20 \\ 5 \end{bmatrix} \quad \mathbf{P}_2 = \begin{bmatrix} 30 \\ 15 \end{bmatrix} \quad \mathbf{P}_3 = \begin{bmatrix} 25 \\ 20 \end{bmatrix} \quad (6.13)$$

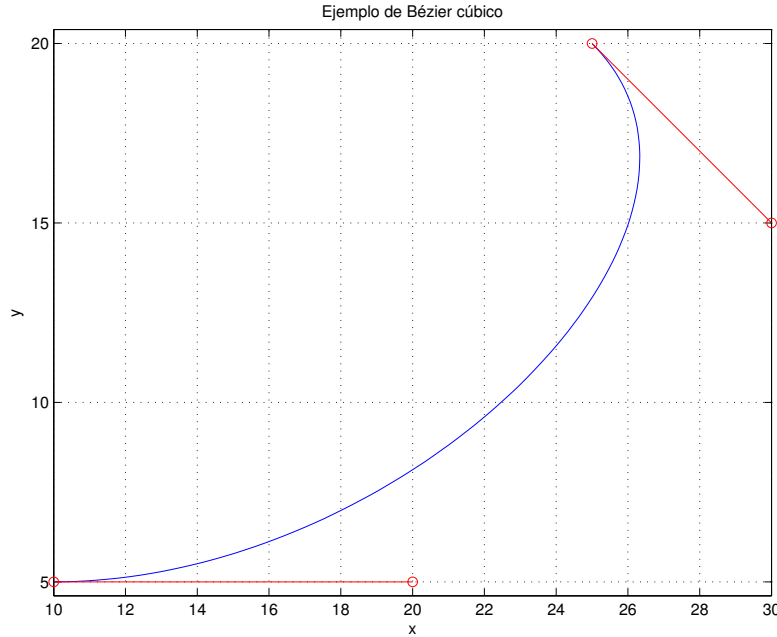


Figura 6.5: Ejemplo de curva de Bézier cúbica con puntos de control (6.13).

La derivada de una curva de Bézier es, también, una curva de Bézier de un grado menos. La expresión de la curva derivada puede calcularse derivando $B_{n,i}(\varpi)$ con respecto a ϖ ya que los puntos de control son independientes del parámetro ϖ . De forma que:

$$\frac{d}{d\varpi} B_{n,i}(\varpi) = B'_{n,i}(\varpi) = n (B_{n-1,i-1}(\varpi) - B_{n-1,i}(\varpi)) \quad (6.14)$$

de lo que se sigue que

$$\frac{d}{d\varpi} \mathbf{C}(\varpi) = \mathbf{C}'(\varpi) = \sum_{i=0}^{n-1} B_{n-1,i}(\varpi) [n (\mathbf{P}_{i+1} - \mathbf{P}_i)] \quad (6.15)$$

que puede reducirse a

$$\mathbf{C}'(\varpi) = \sum_{i=0}^{n-1} B_{n-1,i} \mathbf{Q}_i \quad (6.16)$$

donde $\mathbf{Q}_{n-1} \triangleq n(\mathbf{P}_n - \mathbf{P}_{n-1})$. La Figura 6.6 muestra la derivada de la curva representada en la Figura 6.5, los puntos de control de esta curva son:

$$\mathbf{Q}_0 = \begin{bmatrix} 30 \\ 0 \end{bmatrix} \quad \mathbf{Q}_1 = \begin{bmatrix} 30 \\ 30 \end{bmatrix} \quad \mathbf{Q}_2 = \begin{bmatrix} -15 \\ 15 \end{bmatrix} \quad (6.17)$$

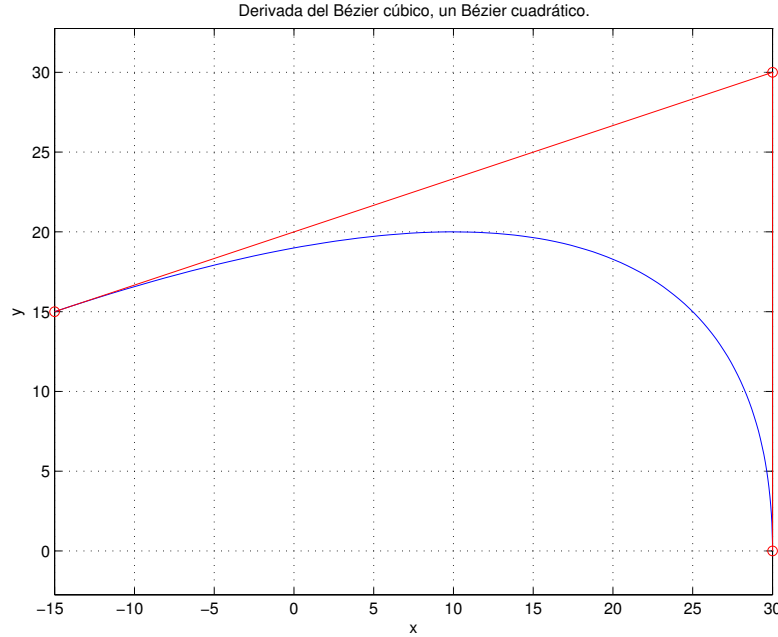


Figura 6.6: Derivada de la curva representada en la Figura 6.5.

Las curvas cúbicas tienen, sin embargo, un problema, que solo pueden representar ciertas trayectorias. De forma que se ha decidido utilizar un método de unir varias curvas cúbicas de forma que generar una trayectoria larga sea cuestión de definir unos puntos de paso y la orientación deseada en esos puntos de paso.

El método de unión debe permitir unir dos curvas $\mathbf{C}(\varpi)$ (con m puntos de control \mathbf{P}_i) y $\mathbf{D}(\varpi)$ (con n puntos de control \mathbf{Q}_i) con continuidad C^1 . Para conseguir continuidad C^0 es necesario hacer que los puntos de fin e inicio coincidan:

$$\mathbf{C}(1) = \mathbf{D}(0) \implies \mathbf{P}_m = \mathbf{Q}_0 \quad (6.18)$$

Además, para garantizar C^1 es necesario que

$$\mathbf{C}'(1) = \mathbf{D}'(0) \quad (6.19)$$

De modo que, partiendo de (6.19) se puede derivar un método para unir dos curvas de Bézier cualesquiera:

$$\mathbf{C}'(1) = m(\mathbf{P}_m - \mathbf{P}_{m-1}) = \mathbf{D}'(0) = n(\mathbf{Q}_1 - \mathbf{Q}_0) \quad (6.20)$$

reescribiendo los términos como $\mathbf{X} \triangleq \mathbf{Q}_0$, $\mathbf{Y} \triangleq \mathbf{P}_{m-1}$ y $\mathbf{Z} \triangleq \mathbf{Q}_1$ y aplicando (6.18) obtenemos

$$m(\mathbf{X} - \mathbf{Y}) = n(\mathbf{Z} - \mathbf{X}) \quad (6.21)$$

de donde podemos obtener una expresión para \mathbf{Z} despejando:

$$\mathbf{Z} = \frac{m}{n}(\mathbf{X} - \mathbf{Y}) + \mathbf{X} \quad (6.22)$$

La Figura 6.7 muestra un ejemplo de unión de dos curvas de Bézier cúbicas. La primera curva es la que tiene puntos de control (6.13). Se une con una curva que tiene los siguientes puntos de control, los dos primeros obtenidos usando (6.18) y (6.22):

$$\mathbf{Q}_0 = \mathbf{P}_3 = \begin{bmatrix} 25 \\ 20 \end{bmatrix} \quad \mathbf{Q}_1 = 2 \cdot \mathbf{P}_3 - \mathbf{P}_2 = \begin{bmatrix} 20 \\ 25 \end{bmatrix} \quad \mathbf{Q}_2 = \begin{bmatrix} 14 \\ 15 \end{bmatrix} \quad \mathbf{Q}_3 = \begin{bmatrix} 10 \\ 10 \end{bmatrix} \quad (6.23)$$

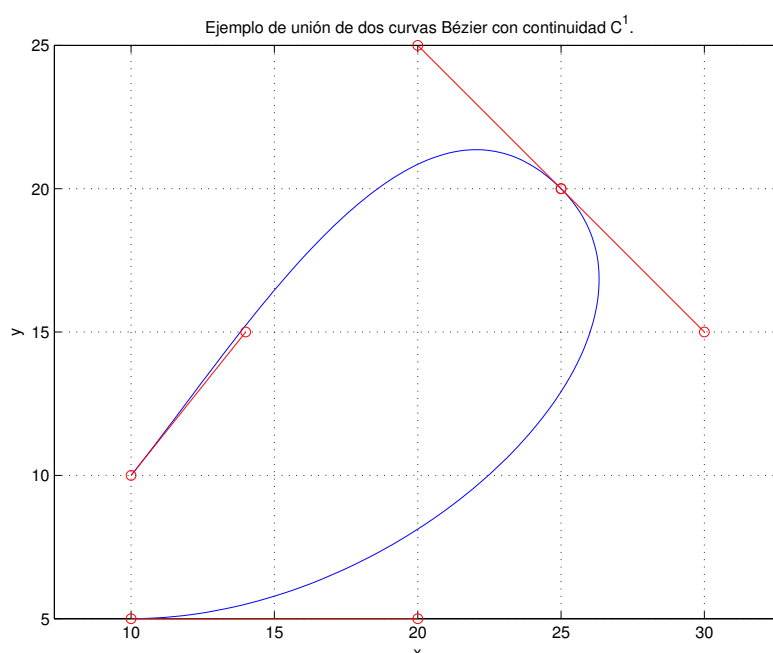


Figura 6.7: Unión de las curvas con puntos de control (6.13) y (6.23).

Un aspecto importante, que aparece al definir formaciones a lo largo de una trayectoria, es conocer la orientación que debe tomar la trayectoria en un punto de la misma. La orientación de la curva en algún punto de la trayectoria, se puede obtener de la siguiente forma:

$$c_\chi(\varpi) = \text{atan2}(c'_y(\varpi), c'_x(\varpi)) \quad (6.24)$$

donde

$$c'(\varpi) = \frac{d}{d\varpi} c(\varpi) \quad (6.25)$$

Además de ser capaz de representar trayectorias utilizando curvas de Bézier, el sistema de splines tiene la responsabilidad de obtener la posición deseada de la formación así como la orientación de la misma. La orientación deseada de la formación puede obtenerse analíticamente utilizando la Ecuación (6.24). Este ángulo es una de las salidas del subsistema de splines.

La otra salida del subsistema es la posición a la que debe dirigirse la formación. Numéricamente se encarga de buscar un punto $\mathbf{p}_d \in c(\varpi)$ que esté siempre a una determinada distancia d de la posición actual del centroide de la formación. Además, se impone $\dot{\varpi} > 0$ de forma que la formación siempre avance en la dirección de la trayectoria. La salida de este proceso, $\mathbf{p}_d \triangleq [n_d \ e_d]^T$, es la consigna de una de las tareas del control jerárquico explicado en la siguiente subsección.

6.2.2. Control jerárquico *Null-Space-Based* (NSB)

La tarea del control jerárquico es organizar la formación de barcos, guiarla a lo largo de una trayectoria y, a su vez, evitar que los barcos se choquen. La responsabilidad puede dividirse, por tanto, en esas tres tareas. El control jerárquico NSB permite la especificación de tareas en un cierto orden de prioridad. Las referencias relevantes sobre el control y su aplicación tanto a robots terrestres como a barcos son (Antonelli et al., 2005; Arrichiello et al., 2006; Antonelli et al., 2010).

Sea $\boldsymbol{\sigma} \in \mathbb{R}^m$ la variable a controlar y $\mathbf{p} \in \mathbb{R}^n$ el estado del sistema, la función que define la tarea $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ es:

$$\boldsymbol{\sigma} = f(\mathbf{p}) \quad (6.26)$$

dado que \mathbf{p} es, de hecho, $\mathbf{p}(t)$, se obtiene la siguiente relación diferencial

$$\dot{\boldsymbol{\sigma}} = \frac{\partial f(\mathbf{p})}{\partial \mathbf{p}} \mathbf{v} = \mathbf{J}(\mathbf{p}) \mathbf{v} \quad (6.27)$$

donde $\mathbf{J} \in \mathbb{R}^{m \times n}$ es la matriz Jacobiana dependiente de la configuración del sistema, y $\mathbf{v} \in \mathbb{R}^n$ es la velocidad del mismo. Según (Arrichiello et al., 2006), una forma de generar referencias suaves es utilizar la solución denominada *Closed Loop Inverse Kinematics* (CLIK) de forma que la referencia es

$$\mathbf{v}_d = \mathbf{J}^\dagger (\dot{\boldsymbol{\sigma}}_d + \boldsymbol{\Lambda} \tilde{\boldsymbol{\sigma}}) \quad (6.28)$$

donde $\mathbf{J}^\dagger \triangleq \mathbf{J}^T (\mathbf{J}\mathbf{J}^T)^{-1}$ es la pseudo-inversa de la matriz Jacobiana, $\boldsymbol{\Lambda}$ es una matriz de ganancias definida positiva y $\tilde{\boldsymbol{\sigma}}$ es el error de la tarea, definido como $\tilde{\boldsymbol{\sigma}} \triangleq \boldsymbol{\sigma}_d - \boldsymbol{\sigma}$. Esto indica que el control jerárquico NSB requiere que las funciones de tareas, f_i , sean derivables de modo que la matriz Jacobiana pueda calcularse.

En el caso de varias tareas, la expresión de la referencia de la velocidad de la tarea *i-ésima* es,

$$\mathbf{v}_{i,d} = \mathbf{J}_i^\dagger (\dot{\boldsymbol{\sigma}}_{i,d} + \boldsymbol{\Lambda}_i \tilde{\boldsymbol{\sigma}}_i) \quad (6.29)$$

Para cumplir la prioridad de cada una de las tareas, las de menor prioridad son proyectadas sobre el espacio nulo de las de mayor prioridad, de modo que un sistema con tres tareas obtendría la referencia de velocidad

$$\mathbf{v}_{NSB} = \mathbf{v}_{1,d} + \mathbf{null}_1(\mathbf{v}_{2,d} + \mathbf{null}_2 \mathbf{v}_{3,d}) \quad (6.30)$$

donde $\mathbf{null}_i \triangleq \mathbf{I} - \mathbf{J}_i^\dagger \mathbf{J}_i$ es la matriz de proyección en el espacio nulo de la tarea *i*.

El final de la presente sección contiene un ejemplo de aplicación donde se observa claramente el efecto de proyectar las tareas de prioridad inferior sobre el espacio nulo de las de mayor prioridad. El objetivo es evitar que las consignas que generan las tareas sean contradictorias. En

caso de que lo fueran, la proyección sobre el espacio nulo de la tarea inmediatamente superior elimina las contradicciones.

En (Arrichiello et al., 2006) se demuestra que el sistema es capaz de llevar el error $\tilde{\sigma}$ a cero de tantas tareas como grados de libertad tenga el sistema, siempre y cuando estas no entren en conflicto. Para el sistema de guiado de este Trabajo se utilizan tres tareas: 1) evitar choques entre barcos 2) guiar al grupo 3) crear y mantener una formación. La referencia básica sobre estas tareas es (Arrichiello et al., 2006).

La presente sección recoge la formulación de cada una de las tareas y comenta las peculiaridades de cada una de ellas. Al final de la sección se muestra un ejemplo de aplicación del algoritmo para dos barcos en formación. Las tareas tanto de guiado del grupo como de crear y mantener la formación se calculan y ejecutan de manera global; sin embargo, la tarea de evitar choques (u obstáculos) debe calcularse de forma individual para cada elemento.

Evitar choques

Cuando se trabaja con maniobras entre distintos robots es importante garantizar la seguridad de los elementos involucrados. El control NSB permite describir una tarea para forzar la distancia máxima permitida entre dos elementos.

La tarea de evitar choques es la que se ejecuta con mayor prioridad, pero solo debe activarse cuando se está en una cierta vecindad del obstáculo. Para evitar choques se hace que los barcos estén a una cierta distancia d los unos de los otros. La variable que se quiere controlar es, por tanto,

$$\sigma_c = \|\mathbf{p} - \mathbf{p}_o\| \quad (6.31)$$

donde $\mathbf{p} = [n \ e]^T$ es la posición del barco y $\mathbf{p}_o = [n_o \ e_o]^T$ es la posición del otro barco (u obstáculo). Dado que se quiere que la distancia sea d , se tiene que $\sigma_{c,d} = d$. Y, por tanto,

$$\mathbf{J}_c = \hat{\mathbf{r}}^T \quad (6.32)$$

donde

$$\hat{\mathbf{r}} = \frac{\mathbf{p} - \mathbf{p}_o}{\|\mathbf{p} - \mathbf{p}_o\|} \quad (6.33)$$

y

$$\|\mathbf{p} - \mathbf{p}_o\| = \sqrt{(n - n_o)^2 + (e - e_o)^2} \quad (6.34)$$

Finalmente, la matriz de proyección sobre el espacio nulo de la tarea de evitar choques es

$$\mathbf{null}_c = \mathbf{I} - \hat{\mathbf{r}}\hat{\mathbf{r}}^T \quad (6.35)$$

Utilizando todo lo anterior se puede obtener la expresión de la consigna de velocidad de esta tarea

$$\mathbf{v}_{c,d} = \mathbf{J}_c^\dagger \lambda_c \tilde{\sigma}_c = \lambda_c (d - \|\mathbf{p} - \mathbf{p}_o\|) \hat{\mathbf{r}} \quad (6.36)$$

A continuación se ilustra un ejemplo de aplicación de esta tarea. En el caso de un barco en la posición $\mathbf{p} = [5 \ 0]^T$ y un obstáculo (u otro barco) en la posición $\mathbf{p}_o = [10 \ 0]^T$. Si el radio de seguridad deseado es $\sigma_{o,d} = d = 5$ metros, se tiene que

$$\tilde{\sigma}_c = d - \|\mathbf{p} - \mathbf{p}_o\| = -2 \quad (6.37)$$

$$\hat{\mathbf{r}} = [-1 \ 0]^T \quad (6.38)$$

por tanto,

$$\mathbf{v}_{c,d} = \lambda_c \begin{bmatrix} 2 \\ 0 \end{bmatrix} \quad (6.39)$$

$$\mathbf{null}_c = \mathbf{I} - \hat{\mathbf{r}}\hat{\mathbf{r}}^T = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad (6.40)$$

La consigna de velocidad y rumbo obliga al barco a 'acercarse' hasta estar a d metros del obstáculo. La matriz \mathbf{null}_c solo permite que las componentes tangenciales al círculo contribuyan a la consigna final. De forma que las tareas de menor prioridad nunca podrían llevar al sistema a colisionar con el obstáculo.

Guiar al grupo

Una forma válida de guiar a un grupo es hacer que su centroide siga una cierta trayectoria. Esto casa bastante bien con el sistema de trayectorias usando splines descrito anteriormente. La idea es que el centroide del grupo siga la trayectoria descrita por un conjunto de splines generados consecutivamente siguiendo las reglas descritas en la Sección 6.2.1 (Splines para trayectorias).

La variable a controlar es, por tanto, la posición del centroide

$$\boldsymbol{\sigma}_g = f_g(\mathbf{p}) = \frac{1}{n} \sum_{i=1}^n \mathbf{p}_i \quad (6.41)$$

donde $\mathbf{p}_i = [n_i \ e_i]^T$ es la posición del barco i -ésimo en el sistema de coordenadas NED. El valor deseado de la variable característica de la tarea será un cierto punto $\boldsymbol{\sigma}_{g,d} \in c(\varpi)$ calculado por el sistema de trayectorias. Derivando la expresión (6.41) se obtiene

$$\dot{\boldsymbol{\sigma}}_g = \sum_{i=1}^n \frac{\partial f_g(\mathbf{p})}{\partial \mathbf{p}_i} \mathbf{v}_i = \mathbf{J}_g(\mathbf{p}) \mathbf{v}_i \quad (6.42)$$

donde

$$\mathbf{J}_g = \frac{1}{n} \begin{bmatrix} 1 & 0 & \cdots & 1 & 0 \\ 0 & 1 & \cdots & 0 & 1 \end{bmatrix} \in \mathbb{R}^{2 \times 2n} \quad (6.43)$$

de forma que la consigna de velocidad debido a esta tarea es

$$\mathbf{v}_{g,d} = \mathbf{J}_g^\dagger (\dot{\boldsymbol{\sigma}}_{g,d} + \boldsymbol{\Lambda}_g \tilde{\boldsymbol{\sigma}}_g) \quad (6.44)$$

Un ejemplo de esta tarea se incluye a continuación. Supongamos dos barcos situados en $\mathbf{p}_1 = [-5 \ 3]^T$ y $\mathbf{p}_2 = [3 \ -5]^T$ y un punto $\boldsymbol{\sigma}_{g,d} = [10 \ 0]^T$. El centroide del grupo y el error de la tarea son, por tanto,

$$\boldsymbol{\sigma}_g = \frac{1}{2} (\mathbf{p}_1 + \mathbf{p}_2) = [-1 \ -1]^T \quad (6.45)$$

$$\tilde{\boldsymbol{\sigma}}_g = \boldsymbol{\sigma}_{g,d} - \boldsymbol{\sigma}_g = [11 \ 1]^T \quad (6.46)$$

El escenario de ejemplo está representado en la Figura 6.8. Los barcos están representados con triángulos de color (rojo y azul) mientras que el centroide se muestra como un triángulo transparente de borde negro. El círculo de línea negra discontinua representa la posición deseada del centroide del grupo, $\boldsymbol{\sigma}_{d,g}$. Las flechas representan los comandos de velocidad de cada uno de los barcos. Los comandos de velocidad son iguales para cada uno de los barcos ya que todos comparten, en el mismo grado, la responsabilidad de llevar el centro al punto deseado.

La tarea de guiar el grupo es la de mayor prioridad o la segunda de mayor prioridad, según se active o no la tarea de evitar choques.

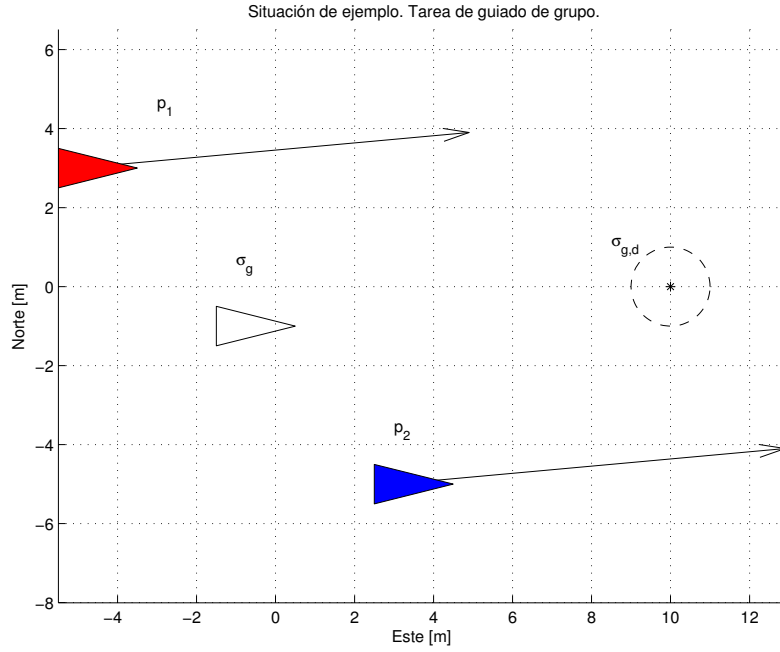


Figura 6.8: Escenario de ejemplo del control NSB para la tarea de guiar el grupo.

Crear y mantener una formación

El control NSB puede utilizarse para que un grupo mantenga una formación rígida. Esto fue presentado por primera vez en (Arrichiello et al., 2006) para una formación que se mueve en línea recta. Aquí se muestra el caso general para una formación que se mueve a lo largo de una trayectoria cualquiera, representada por una curva paramétrica $c(\varpi)$. En concreto, solo necesita la orientación de la curva para un valor particular del parámetro ϖ . En secciones anteriores se ha utilizado $c_\chi(\varpi)$ para referirse a esta orientación, esta convención se mantiene a lo largo del Trabajo.

Se define un sistema de referencia llamado CENTROIDE con origen en el centroide del grupo en el sistema de referencia NED, $\mathbf{c}^n = [n_c \ e_c \ 0]^T$. La orientación¹ de este sistema de referencia es, según la Ecuación (6.24), $c_\chi(\varpi)$. Una posición en este sistema de referencia \mathbf{p}^c puede representarse en el sistema NED mediante la siguiente relación,

$$\mathbf{p}^n = \mathbf{R}(c_\chi(\varpi)) \mathbf{p}^c + \mathbf{c}^n \quad (6.47)$$

Por definición, $\mathbf{c}^c \triangleq [0 \ 0 \ 0]^T$.

La forma de representar una formación rígida es especificar las posiciones de la formación en el sistema CENTROIDE, por ejemplo, un cuadrado con cinco barcos, sería

$$\mathbf{cuadrado} = [\mathbf{p}_{f,d_1}^c \ \cdots \ \mathbf{p}_{f,d_5}^c] = \begin{bmatrix} 0 & -d & -d & d & d \\ 0 & -d & d & -d & d \end{bmatrix} \quad (6.48)$$

donde $2d$ es la longitud del lado del cuadrado.

La variable característica de la tarea se define como la distancia de cada uno de los barcos a

¹Su orientación, es decir, su rotación con respecto al sistema de referencia NED.

su posición deseada en la formación en el sistema de referencia CENTROIDE,

$$\boldsymbol{\sigma}_f = \begin{bmatrix} \mathbf{p}_1^c \\ \vdots \\ \mathbf{p}_n^c \end{bmatrix} \quad \text{y} \quad \boldsymbol{\sigma}_{f,d} = \begin{bmatrix} \mathbf{p}_{f,d_1}^c \\ \vdots \\ \mathbf{p}_{f,d_n}^c \end{bmatrix} \quad (6.49)$$

donde los \mathbf{p}_{f,d_i}^c son la posición deseada de cada barco en la formación. Por ejemplo, como en (6.48).

En aras de simplificar el cálculo de \mathbf{J}_f se escribe $\tilde{\boldsymbol{\sigma}}_f$ agrupando primero los errores en la coordenada x del sistema de referencia CENTROIDE y luego los errores en y . De forma que quede

$$\tilde{\boldsymbol{\sigma}}_f = [x_{d,1} - x_1 \quad \cdots \quad x_{d,n} - x_n \quad \cdots \quad y_{d,1} - y_1 \quad \cdots \quad y_{d,n} - y_n]^T \quad (6.50)$$

donde $x_{d,i}$ e $y_{d,i}$ son las posiciones de la formación y x_i e y_i son las posiciones de los barcos en el sistema de referencia CENTROIDE.

La matriz Jacobiana queda

$$\mathbf{J}_f = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} \end{bmatrix} \in \mathbb{R}^{2n \times 2n} \quad (6.51)$$

donde

$$\mathbf{A} = \mathbf{I}_{n \times n} - n^{-1} \mathbf{1}_n \mathbf{1}_n^T \quad (6.52)$$

donde $\mathbf{1}_n \triangleq [1 \quad \cdots \quad 1]^T \in \mathbb{R}^n$. Dado que \mathbf{A} es singular e idempotente, la matriz \mathbf{J}_f también lo es. De forma que se puede utilizar $\mathbf{J}_f^\dagger = \mathbf{J}_f$ ya que, por ser idempotente, cumple

$$\mathbf{J}\mathbf{J}^\dagger\mathbf{J} = \mathbf{J} \quad \mathbf{J}^\dagger\mathbf{J}\mathbf{J}^\dagger = \mathbf{J}^\dagger \quad (6.53)$$

La justificación de esta asunción puede encontrarse en (Arrichiello et al., 2006).

En el caso de una formación fija, $\dot{\boldsymbol{\sigma}}_{f,d} = 0$, de modo que la consigna de velocidad de esta tarea puede escribirse de la siguiente forma

$$\mathbf{v}_{d,f} = \mathbf{J}_f^\dagger \boldsymbol{\Lambda}_f \tilde{\boldsymbol{\sigma}}_f \quad (6.54)$$

A continuación se muestra un ejemplo en el cual los barcos deben ir en línea perpendicular a la trayectoria, a una distancia $2d$ entre ellos, es decir,

$$\boldsymbol{\sigma}_{d,f} = \begin{bmatrix} 0 & 0 \\ d & -d \end{bmatrix} \quad (6.55)$$

supongamos que las posiciones de los barcos en el sistema CENTROIDE son

$$\mathbf{p}_1^c = \begin{bmatrix} -5 \\ 3 \end{bmatrix} \quad \text{y} \quad \mathbf{p}_2^c = \begin{bmatrix} 5 \\ -3 \end{bmatrix} \quad (6.56)$$

La Figura 6.9 muestra el resultado de ejecutar la tarea en el escenario de ejemplo. El sistema de coordenadas es, de hecho, el sistema de referencia CENTROIDE. Los círculos muestran las posiciones deseadas en este sistema de referencia. Las flechas son el resultado de aplicar el algoritmo a los datos del ejemplo. Puede observarse como la responsabilidad de mantener la formación se distribuye equitativamente entre los elementos del grupo.

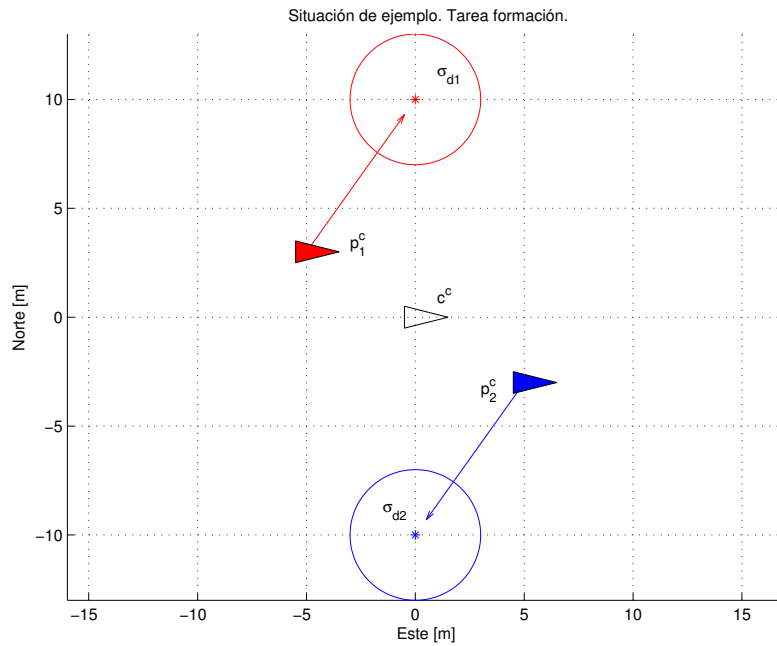


Figura 6.9: Escenario de ejemplo del control NSB, tarea de mantener una formación.

Es importante hacer notar que el sistema de referencia CENTROIDE no es un sistema de referencia Serret-Frenet. En un sistema de referencia Serret-Frenet el vector N (normal) apunta siempre hacia dentro de la curva, de forma que existe un cambio de signo en los puntos de inflexión de la curvatura. El cambio de signo no es deseable en el sistema de referencia CENTROIDE ya que invertiría las posiciones de la formación y la desvirtuaría. El sistema de referencia CENTROIDE está rotado siempre por la tangente a la curva que define la trayectoria, $c_\chi(\varpi)$.

Ejemplo de las tareas de guiado de grupo y formación

Las tareas descritas en esta sección permiten:

- Crear y mantener una formación rígida con un grupo de barcos robot.
- Guiar la formación a lo largo de una trayectoria cualquiera.
- Evitar que los barcos choquen entre si.

El control NSB consigue cumplir estas tareas calculando las consignas de velocidad y proyectando las consignas de menor prioridad sobre el espacio nulo de las tareas de mayor prioridad. Durante el desarrollo de las tareas se ha mostrado un ejemplo de aplicación de cada una de ellas. Queda por tanto, presentar la consigna resultante tras proyectar la consigna de formación sobre el espacio nulo de la tarea de guiado del grupo.

La Figura 6.10 muestra un ejemplo completo. Los triángulos de colores azul y rojo muestran la posición de los barcos en el sistema de referencia NED. El triángulo negro muestra el centroide de la formación, origen del sistema de referencia CENTROIDE; el círculo de línea discontinua muestra el punto al que debe dirigirse el grupo, es decir, $\sigma_{g,d}$. Las flechas negras muestran las consignas de la tarea de guiado del grupo $\mathbf{v}_{g,d}$. Las flechas de color rojo y azul muestran las consignas de la tarea de formación, $\mathbf{v}_{f,d}$. Las flechas magenta muestran la resultante de proyectar las consignas de la tarea de formación sobre el espacio nulo de la tarea de guiado de grupo.

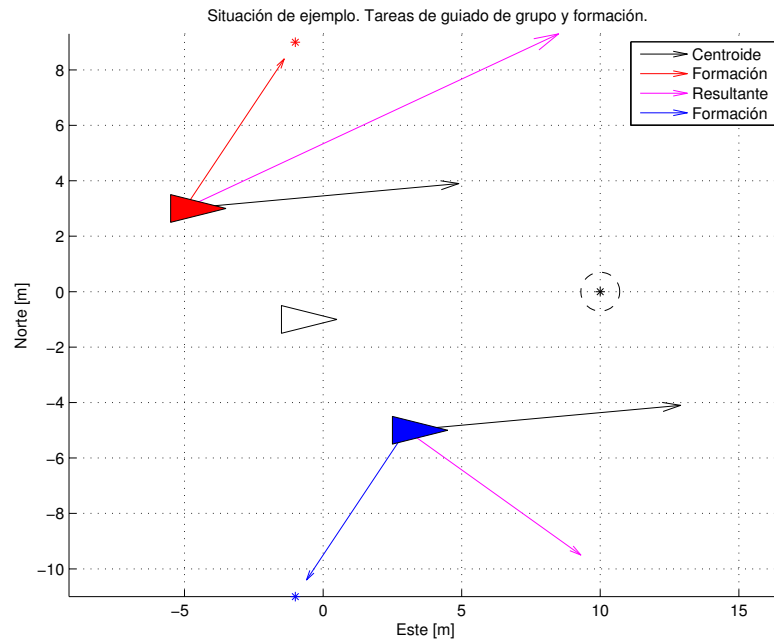


Figura 6.10: Escenario de ejemplo del control NSB.

Se puede observar que aunque, individualmente, las tareas obtienen consignas que reparten la responsabilidad de forma equitativa entre los elementos del grupo, las consignas resultantes no son simétricas; esto es efecto de la proyección sobre el espacio nulo de la tarea de mayor prioridad, el guiado del grupo.

6.2.3. Subastas para formaciones

El control NSB es una herramienta muy potente para el control por tareas de grupos de robot. En la gestión de formaciones, cada robot tiene su posición fija y es ahí donde debe ir. Sin embargo, no siempre es posible saber qué hueco debe ocupar cada barco en la formación. Ser capaz de asignar las posiciones a cada barco es importante, sobre todo, en la inicialización de la formación y en los cambios de formación.

Especificar, a priori, qué puesto debería ocupar cada barco es prácticamente imposible, ya que la mejor configuración depende tanto de la dinámica y configuración inicial del grupo así como de los efectos ambientales (corrientes, viento, ...) y perturbaciones y ruidos. En lugar de hacer una asignación a priori, es posible decidir en cada momento qué barco debe ocupar qué hueco. Este problema se conoce como *El problema de la asignación*, y ha sido descrito formalmente en el Capítulo 4 (El problema de la asignación y el algoritmo de subastas). En el mismo capítulo se describe un algoritmo llamado *de subasta* para resolver el problema. El algoritmo utiliza la metáfora de la negociación para resolver el problema de la asignación (Bertsekas, 1989; Davis and Smith, 1981). El mismo capítulo incluye un ejemplo de aplicación del algoritmo en el contexto del Trabajo donde pueden verse los conceptos económicos de la oferta y la demanda, producto de la metáfora del mercado en la que se basa el algoritmo. La misma metáfora se utiliza a lo largo de esta sección para describir algunos de los conceptos y resultados.

El sistema de Guiado monitoriza el error de la formación $\|\tilde{\sigma}_f\|$ y, si este es mayor de un cierto umbral $\tilde{\sigma}_{f,\text{máx}}$, lanza el proceso de la subasta. El proceso de la subasta comienza con la publicación, por parte del subastador, de una oferta por cada una de las posiciones disponibles en la formación. Cada posición se caracteriza por *a*) las coordenadas de la posición en el sistema de

referencia CENTROIDE b) la orientación del sistema de referencia CENTROIDE² en el sistema de referencia NED c) la métrica con la que evaluar la oferta . Los barcos actúan como postores en la subasta y puján por cada una de las posiciones en función de la métrica establecida por el subastador.

Cuando finaliza la subasta, el subastador otorga un contrato de servicios a los barcos. Contrato por el cual los barcos se comprometen a reducir el error de formación $\|\tilde{\sigma}_f\|$ por debajo del umbral $\tilde{\sigma}_{f,\text{máx}}$ antes de que venza el plazo de revisión del cumplimiento del contrato $t_{\text{revisión}}$. Esta cláusula actúa como un acuerdo de nivel de servicio que los barcos deben cumplir. Una vez vencido el plazo, el subastador evalúa si se han cumplido los términos del contrato. Si los barcos han cumplido, el subastador proroga automáticamente el contrato durante otros $t_{\text{revisión}}$ segundos. Por el contrario, si los barcos no han cumplido el contrato, el subastador lo termina unilateralmente y vuelve a convocar una subasta. Cuando convoca la subasta, sin embargo, parte de la asignación actual en lugar de iniciarla aleatoriamente; esto permite que la subasta converja más rápidamente. La intuición detrás de esta decisión es que la solución será la misma o muy parecida, de forma que la última solución debería ser un buen punto de partida para la nueva. Este modelo de asignación temporal tiene la ventaja de compensar las causas de elementos externos (ambientales, ruidos, perturbaciones, ...) que no puedan ser modelados en la métrica.

La métrica es una función $h : SE(2) \times SE(2) \rightarrow \mathbb{R}$ que evalúa el coste de que un barco sea asignado un cierto hueco. La función métrica comparte las propiedades deseables de las funciones heurísticas, en concreto, es deseable que sea consistente (Russell and Norvig, 2003). Cualquiera de las métricas mencionadas en (LaValle, 2006, sección 5.1) es una buena candidata para h . Sin embargo, todas ellas pecan de lo mismo cuando se aplican a este problema en particular.

Cualquier función ρ sobre el espacio topológico X que defina un cierto espacio métrico (X, ρ) debe cumplir:

No negatividad $\rho(a, b) \geq 0$

Reflexividad $\rho(a, b) = 0 \iff a = b$

Simetría $\rho(a, b) = \rho(b, a)$

Desigualdad triangular $\rho(a, b) + \rho(b, c) \geq \rho(a, c)$

Sin embargo, en este caso, la propiedad de *Simetría* no es deseable. La Figura 6.11 ilustra este hecho. Claramente, es más fácil que un barco vaya de la configuración A a la configuración B que viceversa. Es, por tanto, necesario introducir ese conocimiento en el diseño de la función h . Esto no quiere decir que obligatoriamente la función h deba ser asimétrica, de hecho, la distancia euclídea (ignorando la orientación) da muy buenos resultados; sobre todo si se utiliza con un $t_{\text{revisión}}$ bajo.

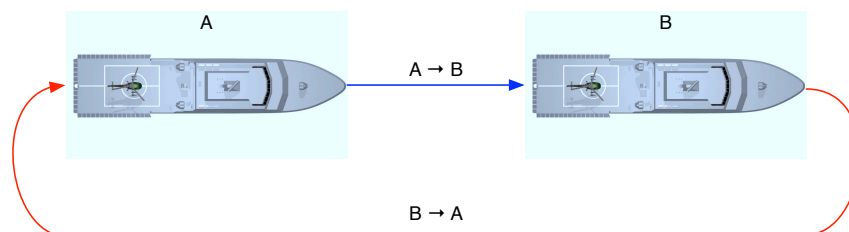


Figura 6.11: La propiedad de simetría no es deseable en la función h .

²La orientación del sistema de referencia CENTROIDE es $c_\chi(\varpi)$.

En caso de no querer usar un $t_{\text{revisión}}$ bajo, o si fuera deseable desactivar las revisiones de los contratos otorgados por el subastador, una forma de encontrar una heurística consistente es partir de la longitud de las curvas de Dubins, (Dubins, 1957). Una consideración a tener en cuenta es que los barcos tienen una cierta capacidad de reducir su velocidad. Dado que la formación avanza a lo largo de $c(\varpi)$, un barco puede decidir frenar para llegar a un punto que tiene por detrás, en lugar de darse la vuelta. Esto es posible porque, a su vez, la formación se mueve a lo largo de la trayectoria y el hueco *alcanza* al barco.

Otra opción como función h es utilizar una simulación en lazo cerrado del propio barco y usar como métrica el tiempo de simulación que tarda el modelo en llegar a la configuración de consigna. Esta métrica es, claramente, consistente. Sin embargo

A lo largo de este trabajo se utiliza la métrica de la distancia euclídea,

$$h(\mathbf{a}, \mathbf{b}) = \|\mathbf{b} - \mathbf{a}\| = \sqrt{(b_x - a_x)^2 + (b_y - a_y)^2} \quad , \quad (6.57)$$

que, aunque es simétrica, da buenos resultados si se utiliza la revisión de contratos por parte del subastador.

6.3. Control

El sistema de control recibe las consignas de velocidad y rumbo del sistema de Guiado y la velocidad actual y la orientación del sistema de Navegación. Su responsabilidad es que el barco adquiera un cierto rumbo y velocidad en poco tiempo. Se ha decidido separar el control en dos modos, uno de velocidad de avance y otro de rotación. Esto es posible debido a las simplificaciones comentadas en el Capítulo 3 (Modelado matemático).

El Sistema de control se divide en dos subsistemas, el control de rumbo y el control de velocidad. Cada uno de ellos recibe la variable del vector de estados y la consigna correspondiente. La Figura 6.12 muestra un diagrama que sitúa el sistema de control frente al resto de sistemas especificando las distintas entradas y salidas.

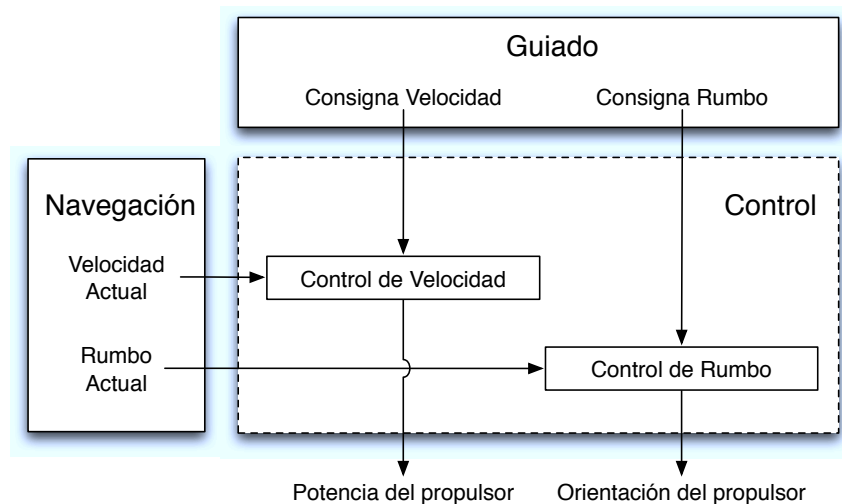


Figura 6.12: Esquema del sistema de Control.

Las dos partes de esta sección desarrollan dos controladores utilizando la técnica de Lyapunov y basándose en el modelo dinámico, Ecuación 3.4. Es importante tener presente dicho modelo durante el desarrollo de los controladores. La primera sección está dedicada al control de rumbo,

contiene la derivación matemática y formal así como distintos ensayos de simulación para el ajuste. La segunda sección trata el control de velocidad, del mismo modo que la primera sección, presenta la derivación matemática así como un ajuste. Ambos controladores se diseñan siguiendo la teoría de Lyapunov, un resumen sobre la estabilidad de sistemas no lineales puede encontrarse en (Fossen, 2011, Apéndice A).

Se ha decidido desarrollar ambos sistemas de control como si los modos de avance y giro estuvieran completamente desacoplados. Esto no es cierto en la realidad ya que al girar, el barco escora un poco que, a su vez, modifica el rozamiento lateral y longitudinal. Adicionalmente, al mover el propulsor, la fuerza en la dirección de avance disminuye, de forma que el control de velocidad tendrá que hacer esfuerzo adicional. Sin embargo, es posible y común, diseñar el control separando ambos modos, y es lo que se ha hecho en esta sección.

6.3.1. Control de rumbo

El control de rumbo tiene como objetivo cumplir las consignas de rumbo que entrega el sistema de Guiado. Formalmente, si se define el error de rumbo como $\tilde{\psi}(t) \triangleq \psi_d(t) - \psi(t)$, queda claro que el objetivo del control de rumbo es llevar este error a cero, es decir,

$$\lim_{t \rightarrow \infty} \tilde{\psi}(t) = 0 \quad (6.58)$$

Se define la siguiente función candidata de Lyapunov:

$$V \triangleq \frac{1}{2}e^2 \quad (6.59)$$

donde $e \triangleq \dot{\tilde{\psi}} + \alpha\tilde{\psi}$ y $\alpha > 0$ es una ganancia que habrá que ajustar. La función es continua y definida positiva, $V > 0$, $\forall e \neq 0$, además, $V = 0$ para $e = 0$. Derivando con respecto al tiempo se obtiene

$$\dot{V} = e\dot{e} = \left(\dot{\tilde{\psi}} + \alpha\tilde{\psi}\right) \left(\ddot{\tilde{\psi}} + \alpha\dot{\tilde{\psi}}\right) \quad (6.60)$$

Para que \dot{V} sea definida negativa, es decir, $\dot{V} < 0$, $e \neq 0$, se puede imponer que $\dot{V} = -\kappa V$ con $\kappa > 0$. De este modo,

$$\left(\ddot{\tilde{\psi}} + \alpha\dot{\tilde{\psi}}\right) = -\frac{\kappa}{2} \left(\dot{\tilde{\psi}} + \alpha\tilde{\psi}\right) \rightarrow \quad (6.61)$$

$$\rightarrow \left(\ddot{\psi}_d - \ddot{\psi} + \alpha\dot{\tilde{\psi}}\right) = -\frac{\kappa}{2} \left(\dot{\tilde{\psi}} + \alpha\tilde{\psi}\right) \quad (6.62)$$

Reagrupando términos y sustituyendo $\ddot{\psi}$ utilizando el modelo dinámico de la Ecuación (3.4) se obtiene

$$\tau_3 = I \left[\frac{\kappa}{2} \left(\dot{\tilde{\psi}} + \alpha\tilde{\psi}\right) + \alpha\dot{\tilde{\psi}} + \ddot{\psi}_d \right] + \mu_r(r)r \quad (6.63)$$

donde τ_3 es el par que debe ejercer el sistema de propulsión.

Utilizando la Ecuación (3.14) se obtiene que $\tau_3 = F_{\text{wj}} \cdot l \cdot \sin \theta_{\text{wj}}$. Despejando se obtiene la expresión para θ_{wj} y de la señal del servo d ,

$$\theta_{\text{wj}}(d) = \text{asin} \left(\frac{\tau_3}{l \cdot F_{\text{wj}}} \right) \implies d = \theta_{\text{wj}}^{-1} \left(\text{asin} \left(\frac{\tau_3}{l \cdot F_{\text{wj}}} \right) \right) \quad (6.64)$$

Dado que V es positiva definida y $\dot{V} = -\kappa V$ con $\kappa > 0$, utilizando argumentos estándar de Lyapunov, se puede demostrar que el error de control $\tilde{\psi}$ decae exponencialmente a cero.

Ajustar la ley de control es la tarea de buscar los valores de κ y α que hacen que el sistema se comporte como se desea. El ajuste se ha hecho buscando intuitivamente los mejores valores basándose en ensayos en simulación. Los valores obtenidos no serán óptimos pero permitirán conseguir un controlador razonablemente bueno.

La ley de control (6.63) depende de las derivadas primera y segunda de ψ_d y de la primera derivada de ψ . Esto hace que sea bastante sensible a los valores de κ y α . Al depender de las derivadas de ψ_d , el control tendrá cierto carácter predictivo frente a cambios en la referencia de rumbo. Para hacer el ajuste del control se van a hacer tres ensayos en simulación. Cada ensayo corresponde al estudio de la respuesta del control frente a una función de consigna. Los ensayos estudian el control frente 1) al escalón 2) a varios cambios de consigna 3) a una consigna que varía de forma continua. A continuación se muestran todos los ensayos y el análisis y conclusiones de los mismos.

La Figura 6.13 muestra la respuesta escalón del control para distintos valores de κ y de α . Se observa el efecto de las constantes en el rumbo del barco y que, como era de esperar, es muy sensible a los valores de éstas. En este caso, se ha decidido que el control no debe ser muy lento, aunque sobreoscile un poco. De forma que se descartan los pares $\{\kappa = 5, \alpha = 1\}$ y $\{\kappa = 3, \alpha = 3\}$, estos pares ya no aparecerán en los distintos ensayos. Por razones de completitud, la Figura 6.14 muestra tanto el error como la actuación de cada uno de los pares de constantes.

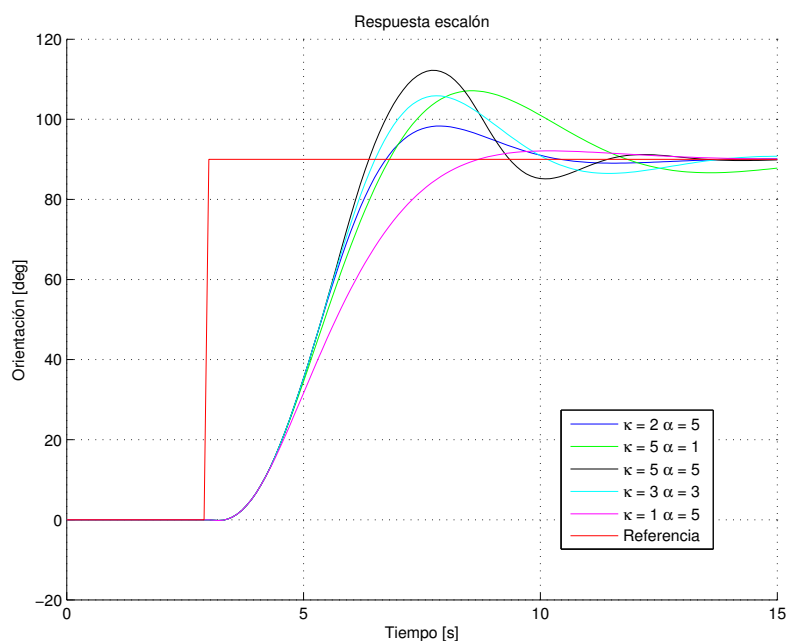


Figura 6.13: Respuesta al escalón del control de rumbo.

El siguiente ensayo consta de una consigna compuesta de la suma de varios escalones. La Figura 6.15 muestra la respuesta a los distintos escalones, mientras que las figuras 6.16a y 6.16b muestran el error y el ángulo del propulsor durante el ensayo.

A la vista de los dos primeros ensayos en simulación, el par más prometedor es $\{\kappa = 2, \alpha = 5\}$. No obstante, es necesario hacer el último ensayo para decidir qué par de ganancias utilizar. Para facilitar la visualización del último ensayo se descarta el par $\{\kappa = 5, \alpha = 5\}$ por tener el mayor tiempo de asentamiento sin ofrecer menor error.

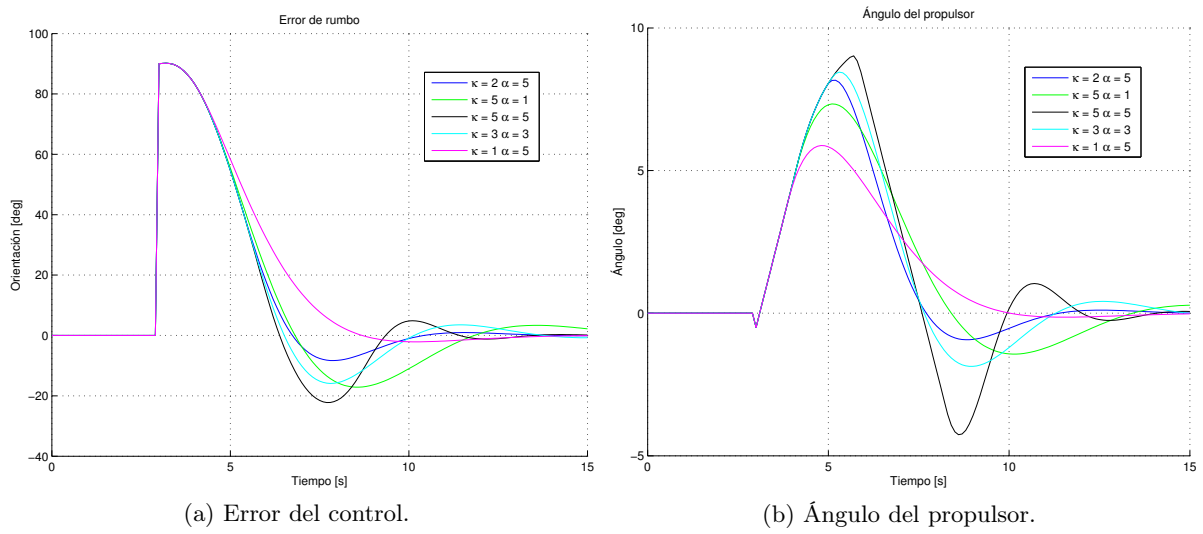


Figura 6.14: Error y actuación del control de rumbo en la respuesta escalón.

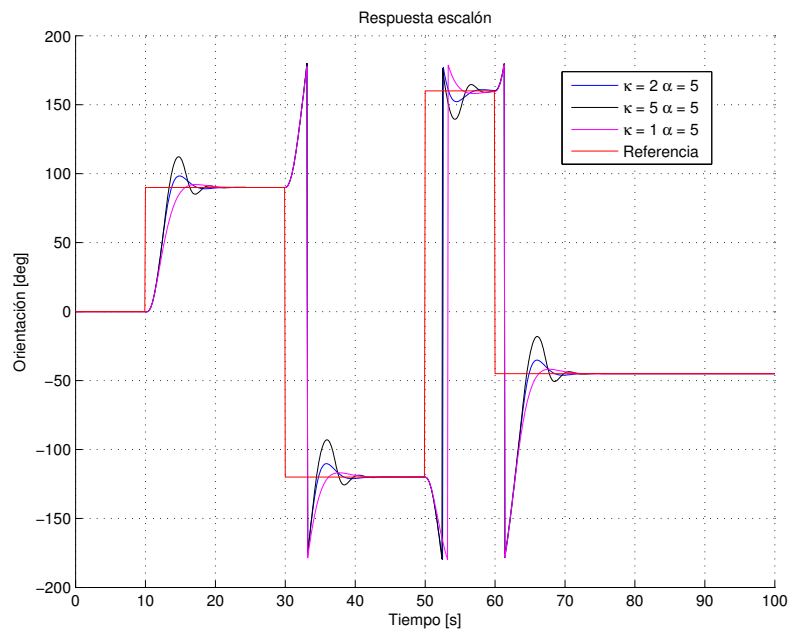


Figura 6.15: Respuesta del control a varios cambios de rumbo.

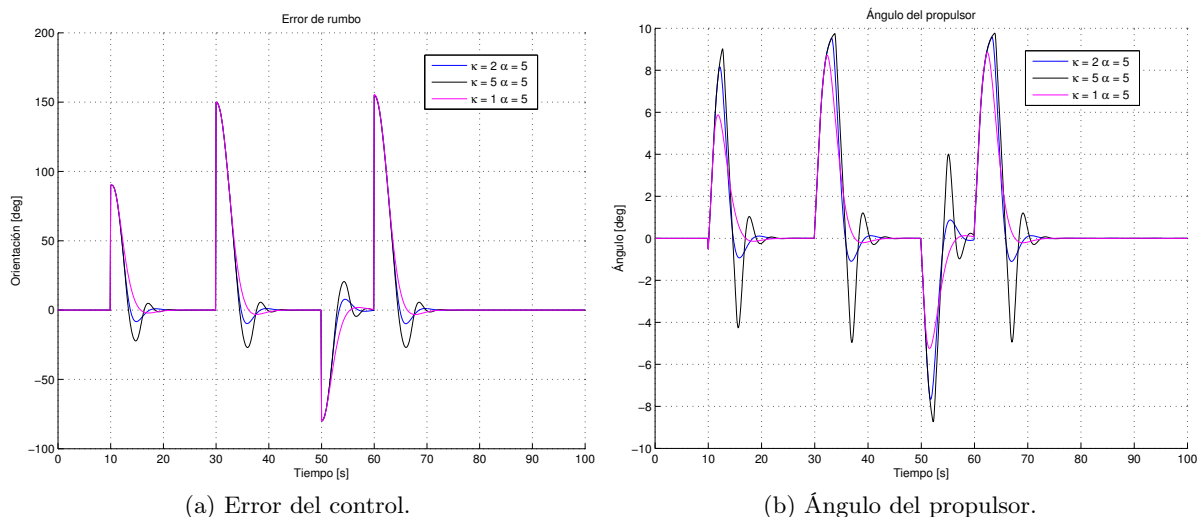


Figura 6.16: Error y actuación del control de rumbo frente a varios cambios de consigna.

En el último ensayo se hace que la consigna del sistema varíe de forma continua en el tiempo. En este ensayo la consigna es $\psi_d(t) = \sin\left(\frac{\pi}{10}t\right) + \sin\left(\frac{\pi}{40}t\right)$. La Figura 6.17 muestra la respuesta del control a esta consigna. Puede observarse que, aunque ambos pares de constantes son razonablemente buenos, el par $\{\kappa = 2, \alpha = 5\}$ tiene menos error. No obstante, cualquiera de los dos pares es una buena aproximación. La experimentación en situaciones reales permitiría mejorar el ajuste. Por completitud se incluyen tanto el error como la señal del actuador de ambos pares de ganancias en las figuras 6.18a y 6.18b respectivamente.

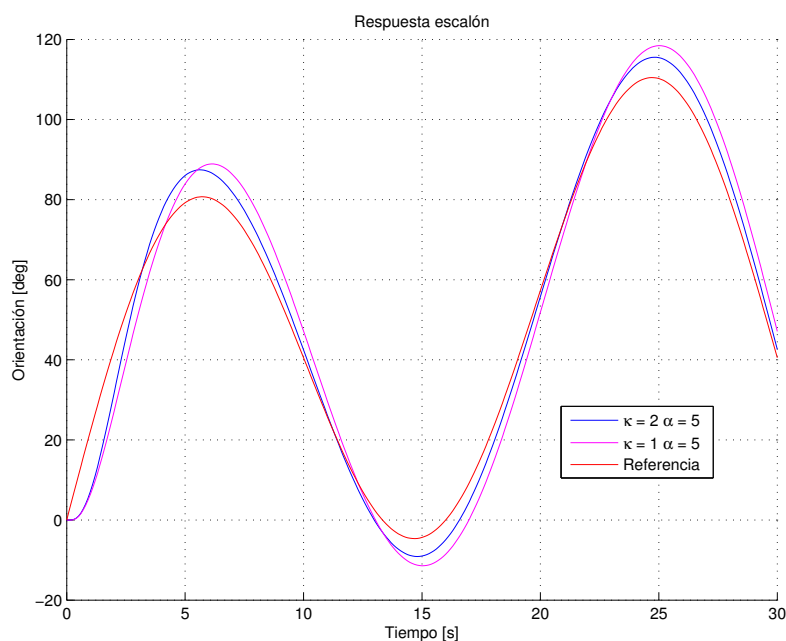


Figura 6.17: Respuesta del control a $\psi_d(t) = \sin\left(\frac{\pi}{5}t\right) + \sin\left(\frac{\pi}{20}t\right)$.

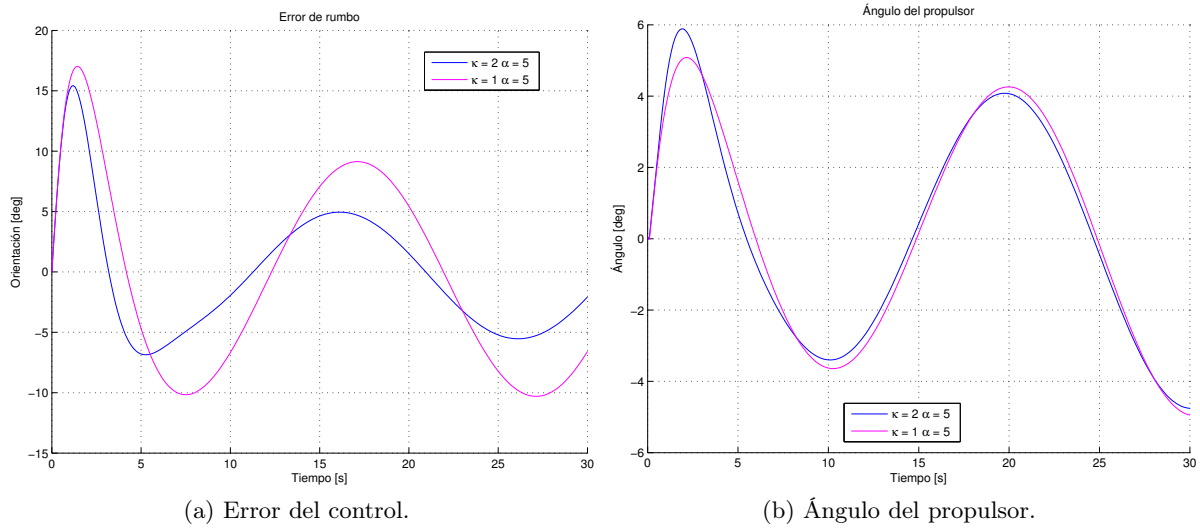


Figura 6.18: Error y actuación del control de rumbo frente a $\psi_d(t) = \sin\left(\frac{\pi}{10}t\right) + \sin\left(\frac{\pi}{40}t\right)$.

6.3.2. Control de velocidad

El control de velocidad tiene como objetivo cumplir las consignas de velocidad de avance. Se define el error en velocidad como $\tilde{u}(t) \triangleq u_d(t) - u(t)$, de forma que el objetivo del control es intentar que ese error sea cero:

$$\lim_{t \rightarrow \infty} \tilde{u}(t) = 0 \quad (6.65)$$

Se define la siguiente función candidata de Lyapunov:

$$V \triangleq \frac{1}{2} \tilde{u}^2 \quad (6.66)$$

que es continua y definida positiva para cualquier valor de $\tilde{u} \neq 0$. Es decir $V > 0$, $\forall \tilde{u} \neq 0$. Derivando con respecto al tiempo se obtiene que

$$\dot{V} = \tilde{u} \dot{\tilde{u}} \quad (6.67)$$

Dado que el objetivo es que \dot{V} sea definida negativa, se puede reescribir como:

$$\dot{V} = \tilde{u} \dot{\tilde{u}} = -\kappa V \quad (6.68)$$

donde $\kappa > 0$. De modo que debe cumplirse que

$$\dot{\tilde{u}} = \dot{u}_d - \dot{u} = -\frac{\kappa}{2} \tilde{u} \quad (6.69)$$

Reagrupando términos y sustituyendo \dot{u} usando la Ecuación 3.4 se obtiene

$$\dot{u} = \frac{\kappa}{2} \tilde{u} + \dot{u}_d \quad \text{debido a (6.69)} \quad (6.70)$$

$$= \frac{\tau_1 - \mu_l(u)u}{m} \quad \text{debido a (3.4)} \quad (6.71)$$

donde τ_1 es la fuerza que debe ejercer el motor para conseguir esa velocidad. De modo que, despejando y reagrupando:

$$\tau_1 = m \left(\frac{\kappa}{2} \tilde{u} + \dot{u}_d \right) + \mu_l(u)u \quad (6.72)$$

Adicionalmente, utilizando las ecuaciones (3.7) y (3.14) se puede obtener la señal del variador del motor eléctrico n :

$$\tau_1 = \rho(n) \cos \theta_{wj} \implies n = \rho^{-1} \left(\frac{\tau_1}{\cos \theta_{wj}} \right) \quad (6.73)$$

Dado que $V > 0$, $\forall \tilde{u} \neq 0$ y $\dot{V} = -\kappa V < 0$, $\forall \tilde{u} \neq 0$, $\kappa > 0$, utilizando argumentos estándar de Lyapunov, se puede demostrar que el error de control \tilde{u} decrece exponencialmente a 0.

Cabe hacer una interpretación más detallada de la ley de control (6.72). Reagrupando los términos se obtiene:

$$\tau_1 = \underbrace{m \frac{\kappa}{2} \tilde{u} + \mu_l(u)u}_{\tau_{FB}} + \underbrace{m \dot{u}_d}_{\tau_{FF}} \quad (6.74)$$

donde τ_{FF} es una entrada de *feed-forward* o alimentación hacia delante y τ_{FB} es una entrada de realimentación (o *feedback*) proporcional al error de control.

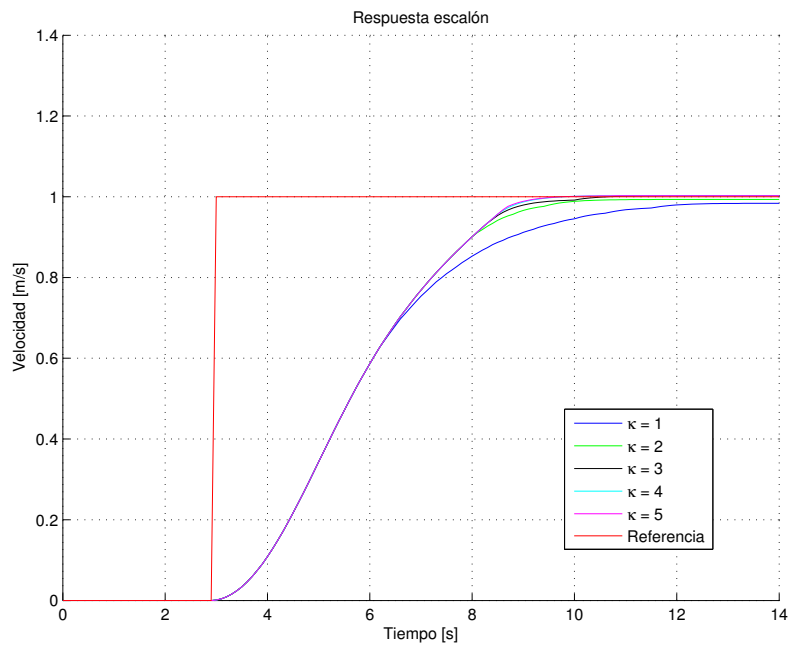
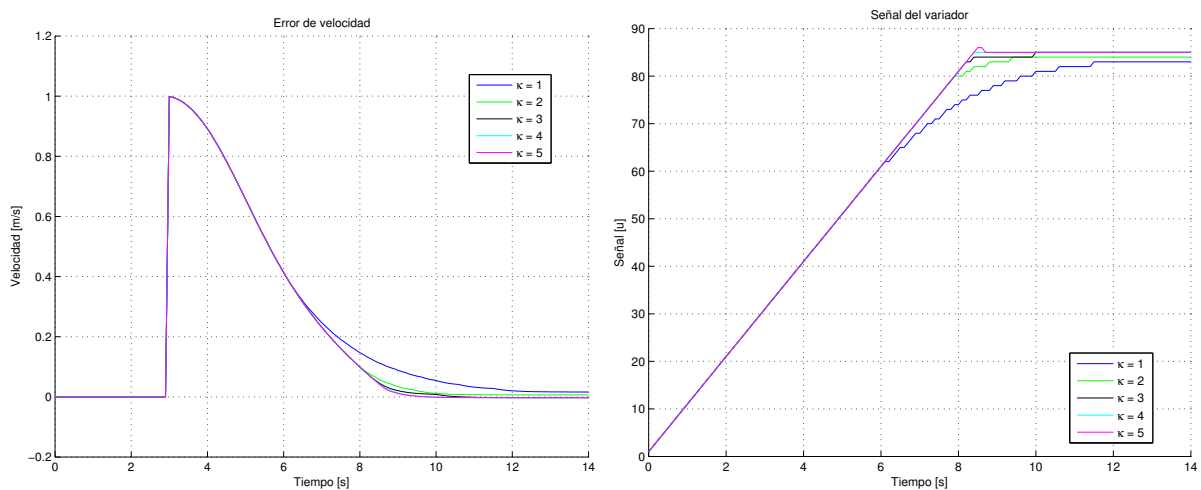


Figura 6.19: Respuesta al escalón del controlador.



(a) Error del control de velocidad, \tilde{u} .

(b) Señal enviada al motor.

Figura 6.20: Error y actuación del control de velocidad frente a una entrada escalón.

Para el ajuste de κ solo se hace un ensayo en simulación correspondiente a la respuesta al escalón. Las figuras 6.19 y 6.20a muestran la respuesta del control frente a una entrada escalón para distintos valores de κ . El valor de κ debe decidirse de forma que las consignas de control no excedan los límites de agilidad del propulsor. En el caso de este Trabajo se ha decidido utilizar $\kappa = 3$, aunque casi todos los valores mostrados dan un comportamiento razonable. Puede verse cómo el error decae exponencialmente en todos los casos.

Con esto concluye el diseño del control de velocidad.

Capítulo 7

Resultados en simulación

Una vez que el sistema ha sido propuesto, los fundamentos matemáticos han sido explicados y se han descrito los detalles de implementación, la siguiente fase es evaluar la viabilidad del mismo.

Una técnica muy común para validar sistemas de guiado, navegación y control es la simulación. En el caso de un sistema relativamente complejo como el propuesto en este Trabajo, el simulador es un importante componente software que debe ser diseñado e implementado siguiendo los principios y técnicas relativos a la Ingeniería Informática.

El simulador debe permitir estudiar los distintos elementos del sistema, así como el comportamiento de los distintos algoritmos propuestos. Además, debe generar los datos de forma que sean fáciles de analizar y estudiar.

El capítulo comienza con la descripción del simulador como una pieza de software. Se describen los requisitos del mismo así como el resultado de las fases de análisis y diseño. Posteriormente se realizan simulaciones para estudiar el sistema propuesto. Se dedica una sección completa a realizar simulaciones realistas en las que se incluyen ruidos y modelado de los sensores y actuadores. Finalmente, se describen las conclusiones obtenidas del análisis de los resultados.

7.1. El simulador

El simulador es una parte muy importante del presente Trabajo. Se trata de una pieza de software que permite evaluar el sistema propuesto y estudiar los distintos algoritmos y métodos que lo componen. Esta sección presenta el simulador desde el punto de vista de la Ingeniería del Software. Se pretende que sirva como nexo de unión entre los planteamientos teóricos del sistema y la realización como producto software; es decir, sirve como documento de trazabilidad de los elementos software, los requisitos del sistema y los planteamientos teóricos.

La primera parte está dedicada a la descripción de requisitos y al análisis del sistema, se estudia el sistema como un conjunto de grandes componentes o subsistemas y se describen los requisitos del mismo. La segunda parte está dedicada al diseño del software así como a los detalles de implementación del mismo; se muestra el diseño de clases y se hace una discusión sobre los patrones de diseño empleados en el desarrollo.

7.1.1. Análisis y requisitos

El sistema debe permitir simular distintos tipos de escenarios. Cada escenario estará compuesto de un número de barcos autónomos, una trayectoria y unos valores para los distintos parámetros configurables del sistema. Estos parámetros son muy numerosos, de forma que un sistema de ficheros de configuración debe ser desarrollado para poder tener un catálogo de escenarios que puedan ser simulados. Estos ficheros de configuración deben permitir configurar, entre otras cosas:

- Si se activa el sistema de subastas. En caso de que esté activado, si se activa la revisión de los contratos según los acuerdos de nivel de servicio. Deben poder configurarse los parámetros de estos acuerdos: el tiempo de revisión y el umbral de error permitido. También se deben poder configurar los parámetros del algoritmo de subastas.
- Si se usan los modelos de los sensores o si se obtienen las señales directamente desde la dinámica. Debe permitir especificar, además, los parámetros de los sensores, a saber: cada cuánto tiempo actualizan sus medidas y magnitud del ruido.
- Configurar las ganancias del control jerárquico NSB así como los distintos parámetros de cada una de las tareas. También se debe poder configurar la frecuencia con la que se calculan y envían las distintas consignas.
- Si se utiliza el estimador basado en UKF o no.
- El número de barcos del escenario y su estado inicial.
- Las ganancias de los distintos sistemas de control.
- Los puntos de control de la trayectoria asociada al escenario.

La Tabla 7.1 muestra los principales requisitos funcionales del simulador. Se recogen aquellos que definen la funcionalidad básica: simular uno o varios barcos, poder simular sensores para hacer simulaciones realistas, generar datos que sean fáciles de estudiar, ...

Al igual que con los requisitos funcionales, solo se han incluido los requisitos no funcionales más importantes. Existen muchos más, pero por razones de brevedad solo se describen las principales restricciones. La Tabla 7.2 recoge estos requisitos.

Tabla 7.1: Requisitos funcionales del simulador.

Identificador	Descripción
Req F.1	El software debe permitir simular varios barcos a la vez.
Req F.2	El software debe poder simular sensores.
Req F.3	Se deben poder configurar los parámetros pertinentes con arreglo a la lista anterior.
Req F.4	El software debe generar datos que sean fáciles de utilizar y cargar en Matlab.
Req F.5	Los modelos físicos de los barcos deben ser fácilmente reemplazables y configurables.
Req F.6	Se debe permitir la simulación en tiempo real.
Req F.7	La simulación en tiempo real debe permitir ver distintos aspectos del sistema propuesto como las consignas generadas por el control NSB.
Req F.8	Se debe permitir la simulación en modo <i>batch</i> .
Req F.9	El software debe implementar todas las técnicas y algoritmos de los sistemas de Guiado, Navegación y Control propuestos.

Tabla 7.2: Requisitos no funcionales del simulador.

Identificador	Descripción
Req N.1	Se debe permitir la visualización en 3D y en 2D utilizando OpenGL.
Req N.2	Si se usa el modo <i>batch</i> , el software debe ser lo más rápido posible.
Req N.3	El software debe funcionar tanto en Linux como en Mac OS X.
Req N.5	El software debe estar escrito en C++.
Req N.6	Los componentes realizados no deben asumir que están funcionando en un entorno de simulación. Esto hará la transición a una plataforma experimental mucho más sencilla.

Las tablas 7.1 y 7.2 muestran los principales requisitos del sistema, han sido resumidos por razones de brevedad ya que este capítulo no pretende ser un estudio exhaustivo sobre el software del simulador. Utilizando los requisitos de ambas tablas se puede hacer una primera subdivisión del sistema en los componentes más grandes. Cada componente tiene una responsabilidad única y clara:

Configuración es el responsable de facilitar la configuración del software. Es el encargado de leer los ficheros de configuración y ofrecer esos datos.

Física y sensores es el componente encargado de simular la dinámica de los barcos así como los distintos sensores y actuadores. Incluye los elementos descritos en el Capítulo 3 (Modelado matemático).

Visualización es el componente responsable de visualizar la simulación en tiempo real. Según los requisitos, debe utilizar OpenGL y permitir la visualización 2D y 3D de la simulación.

Algoritmos GNC es el componente que contiene los algoritmos y técnicas que componen la propuesta del presente Trabajo. Incluye el algoritmo del Capítulo 4 (El problema de la asignación y el algoritmo de subastas), el del Capítulo 5 (El filtro de Kalman *Unscented*), así como los descritos en el Capítulo 6 (Implementación del sistema).

Simulación es el componente que implementa el integrador numérico y gestiona el tiempo de simulación. Además, es el encargado de hacer que los distintos componentes interactúen los unos con los otros.

La Figura 7.1 muestra las interacciones entre los distintos componentes del sistema. Se puede observar que el diseño de alto nivel del software sigue los principios de alta cohesión interna y bajo acoplamiento externo.

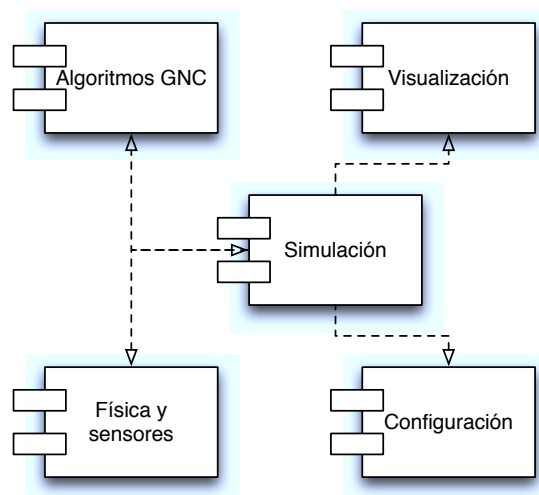


Figura 7.1: Componentes del software del simulador e interacciones entre los mismos.

El diseño de alto nivel (descrito en esta sección) así como el de bajo nivel (descrito en la siguiente sección) cumplen los principios de Orientación a Objetos denominados *S.O.L.I.D.* (*robusto*). Estos principios fueron enunciados por Robert C. Martin a principios de los años 2000, y forman la base de cualquier sistema fácil de desarrollar, mantener y extender. Pueden resumirse de la siguiente forma:

- S** *Single Responsibility Principle (SRP)*, dicta que un elemento del sistema debe tener una única responsabilidad clara.
- O** *Open/Closed Principle (OCP)*, que dice que los elementos software deben poder modificar su comportamiento sin necesidad de cambiar su implementación.
- L** *Liskov Substitution Principle (LSP)*, enuncia que todas las apariciones de un cierto tipo T deben poder ser reemplazadas por alguno de sus subtipos S sin que esto comprometa las propiedades del software. Alternativamente, se puede enunciar de la siguiente forma: *Si $q(x)$ es una propiedad demostrable de objetos x de tipo T . Entonces $q(y)$ debe ser cierto para objetos y de tipo S donde S es un subtipo de T .* Este principio impone restricciones sobre el uso de la herencia durante las fases de diseño e implementación.
- I** *Interface Segregation Principle (ISP)*, dicta que muchas pequeñas interfaces (o contratos) son más deseables que una única interfaz monolítica.
- D** *Dependency Inversion Principle (DIP)*, dice que las entidades software deben depender siempre de abstracciones y nunca de concreciones o detalles de implementación.

7.1.2. Diseño e implementación

Partiendo de la descomposición en componentes principales representada en la Figura 7.1 y de los requisitos descritos en la sección anterior, es posible seguir con el diseño de bajo nivel del sistema. El diseño se ha hecho siguiendo los principios *SOLID*. La Figura 7.2 muestra el diseño de clases del simulador. A continuación se describen los distintos elementos y clases así como sus responsabilidades principales. Se han añadido referencias a las distintas secciones del documento en aquellas clases donde procede.

- En el componente *Algoritmos GNC* se encuentran las siguientes clases. Los elementos de este componente son producto del requisito *Req F.9*.

Auctioneer Es la clase responsable de implementar todo lo relativo a las subastas para las formaciones de la flota. El principal método de la clase, `auction`, implementa tanto el algoritmo básico como las extensiones propuestas en este Trabajo relativas a las *resubstas*. La documentación relativa a esta clase se encuentra en el Capítulo 4 (El problema de la asignación y el algoritmo de subastas) y en la Sección 6.2.3 (Subastas para formaciones).

UT Implementa la transformada *Unscented* que se utiliza en el UKF. Está descrita en el la Sección 5.1 (La transformada *Unscented*).

UKF Implementa la base matemática del filtro UKF. Permite utilizar distintos modelos de propagación y corrección. Los métodos de la clase, `propagate` y `correct`, implementan los pasos del algoritmo cuyo funcionamiento se ha descrito en la Sección 5.2 (UKF para procesos con ruido aditivo).

UKFVessel Es la clase que implementa el algoritmo estimador de la posición en cada uno de los barcos de la flota. Los métodos `propagate` y `correct` utilizan los modelos descritos en la Sección 6.1.2 (Estimador).

MedianFilter Implementa el filtro de la mediana descrito en la Sección 6.1.1 (Filtrado).

NSB Implementa el control jerárquico NSB descrito en la Sección 6.2.2 (Control jerárquico *Null-Space-Based* (NSB)). Esta clase es la responsable de calcular las distintas consignas del sistema de Guiado. Los métodos `set_desired_*` permiten cambiar las

consignas de grupo $\sigma_{g,d}$ y formación $\sigma_{f,d}$. El método `update` permite actualizar la información de los distintos barcos de la flota y `velocities` calcula los comandos de velocidad para cada uno de los barcos.

NSBData Esta clase sirve para guardar el estado de cada uno de los barcos de la flota a ojos del control NSB.

MultiCubicSpline Esta clase implementa la unión de varias curvas de Bézier cúbicas mediante el método propuesto en la Sección 6.2.1 (Splines para trayectorias). Permite obtener el punto en el plano correspondiente a un valor del parámetro usando el método `xy`, así como su derivada con el método `ddt`.

CubicSpline Esta clase representa una curva Bécier cúbica. Las ecuaciones se describen en la Sección 6.2.1 (Splines para trayectorias). Los métodos `xy` y `ddt` son análogos a los de la clase anterior.

ShipController Esta clase implementa una abstracción de un control de rumbo y velocidad para un barco. La clase permite cambiar las consignas de rumbo y velocidad por medio de los métodos `set_desired_*` y obtener las distintas acciones de control por medio de `get_rudder_angle` y `get_engine_esc`.

LyapunovShip Esta clase implementa las leyes de control descritas en la Sección 6.3 (Control).

PIDShip Es una clase que especializa la abstracción de control de rumbo y velocidad utilizando dos controladores Proporcional Integral Derivativo (PID).

PID Implementa un control PID genérico. La clase ofrece un método para cambiar la referencia del controlador (`set_reference`) y uno para calcular la salida del controlador en función de un valor nuevo de la señal (`pid`).

- En el componente de *Física y sensores*:

Ship Implementa el modelo de maniobra tanto dinámico como cinemático de un barco. Ambos modelos están descritos en la Sección 3.2 (Modelo de maniobra del barco). Los elementos más importantes de la clase son las velocidades en el sistema de referencia BARCO (`vel_b`), la posición en el sistema de referencia NED (`pos_w`). El método `step` permite dar un paso de simulación.

SimulatedShip Es una abstracción de un barco con sensores. Permite obtener los valores de los sensores por medio de distintos métodos. Para el sensor de velocidad `surge_velocity`, para la brújula `heading` y para el GPS `position`. El método `step` permite dar un paso de simulación.

ControlledShip Es una abstracción de un barco con control de rumbo y velocidad. Permite cambiar las consignas de rumbo y velocidad por medio de los métodos `set_desired_*` así como dar un paso de simulación mediante el método `step`.

SensorNoise Permite describir y modelar el ruido de un sensor. Es un producto del requisito *Req F.2*. El método `noise` devuelve el valor de ruido que hay que sumar al sensor. Modela, por tanto, ruido aditivo.

- El componente de *Simulación* contiene las siguientes clases:

Simulation Es la clase que orquesta toda la simulación. Gestiona el tiempo así como la integración numérica de los modelos matemáticos. El método más importante, `step`, permite ejecutar un paso de simulación.

SimulationShip Es una clase que permite guardar información sobre cada barco de la flota. Entre otras cosas, guarda los datos de la última vez que se ejecutó el control NSB en la propiedad `nsb_data`.

- El componente de *Visualización* cuenta con las siguientes clases:

Renderer Esta clase es una abstracción de un representador de información de simulación. La clase tiene un único método, `run` que ejecuta la simulación.

GLUTGlue Es una clase que permite utilizar la librería GLU de forma simple desde C++. GLU es una librería que permite crear y gestionar una ventana y un contexto OpenGL.

NullRenderer Es una clase que simula un escenario pero no genera ninguna información sobre los elementos simulados. Solo genera información sobre la simulación, por ejemplo, el tiempo de simulación.

TextRenderer Es una clase que simula un escenario y guarda los datos de la simulación en ficheros de texto para poder ser estudiados a posteriori. Esta clase implementa el modo *batch* correspondiente al requisito *Req F.8*.

GLRenderer Es una clase que simula un escenario en tiempo real y permite la visualización del mismo en un entorno 2D y 3D. Esta es fruto de los requisitos *Req F.6*, *Req F.7* y *Req N.1*.

- Las clases del componente de *Configuración* no se han incluido en el diagrama por razones de brevedad, no obstante, se describen a continuación:

SimpleParser Implementa un analizador léxico y sintáctico de gramáticas libres de contexto.

Configuration Utiliza el analizador léxico y sintáctico para analizar la gramática de los ficheros de configuración. Además, analiza la línea de comandos en busca de elementos de configuración. Permite ofrecer valores por defecto para los elementos de configuración en caso de que no sean especificados. Esta clase es producto del requisito *Req F.3*.

El diseño tanto de alto como de bajo nivel presenta varios patrones de diseño software. A continuación se describen los casos más importantes que ilustran las capacidades y características del simulador. La referencia clásica en patrones de diseño software es (Gamma et al., 1994).

Una inspección detallada de las clases de la Figura 7.2 y de sus responsabilidades permite ver algunos patrones utilizados en la fase de diseño. La simulación de cada uno de los barcos completos es la simulación de distintos elementos en cadena. La clase **Ship** proporciona la dinámica básica, el resto de clases van añadiendo responsabilidades y complementando la funcionalidad. Responsabilidades como modelar sus sensores y añadir control automático. La jerarquía formada por las clases **Ship**, **SimulatedShip**, **ControlledShip** y **SimulationShip** es un caso particular del patrón *Decorator* desde el punto de vista estructural. Desde el punto de vista de comportamiento, es un patrón *Chain-Of-Responsibility* ya que las distintas clases atienden las operaciones correspondientes a su nivel de responsabilidad y las que no pueden atender, las pasan al siguiente nivel.

Una de las responsabilidades de la clase **ControlledShip** es hacer que el barco pueda ser controlado de forma automática. La clase **ShipController** ofrece una interfaz única para cualquier control en rumbo y velocidad de un barco. Las distintas subclases son las distintas estrategias y leyes de control. Desde el punto de vista del comportamiento de las clases, estas forman un patrón *Strategy* ya que permiten cambiar completamente el control sin cambiar nada en la clase **ControlledShip**. Este patrón también está presente en la clase **Auctioneer** que permite elegir y modificar, fácilmente, la métrica utilizada para evaluar el coste de cada barco a cada hueco de la formación.

Como detalle de implementación, la forma de generar los datos en ficheros de texto también sigue un patrón *Chain-Of-Responsibility*. Cada clase que es susceptible de generar datos implementa el método `std::ostream & operator<< (std::ostream &, const T &)` que es la forma común en C++ para volcar los datos internos de una clase a un flujo de datos de salida. La llamada empieza en lo alto de la jerarquía, `SimulationShip`, y cada clase escribe sus datos y llama al mismo método de la siguiente clase. De este modo se escriben los datos internos de cada una de las clases que forman el *Decorator*.

Queda descrito el software del simulador que sirve como herramienta para validar los planteamientos teóricos y las asunciones realizadas. Las siguientes secciones se centran en el análisis de los resultados obtenidos con el simulador.

7.2. Estudio de escenarios simulados

El simulador permite describir los escenarios de una forma simple, utilizando ficheros de configuración. Se han planteado distintos escenarios que van comprobando y validando uno a uno los elementos de la propuesta planteada en el Capítulo 2 (Descripción general del sistema) y desarrollada en el Capítulo 6 (Implementación del sistema).

Durante todo el análisis de resultados se siguen las siguientes convenciones:

- Las consignas o valores deseados se representan con trazos discontinuos.
- Los barcos se representan como triángulos sólidos que indican la posición y orientación de los mismos.
- Las posiciones deseadas en la formación se representan con círculos (con trazos discontinuos).
- Los trazos continuos representan valores de los barcos.
- El rumbo se representa en el rango $(-180, 180]$ de forma que pueden aparecer *discontinuidades ficticias*.

La Figura 7.3 muestra una captura de pantalla del simulador en uno de los escenarios. Se pueden observar distintas características directamente relacionadas con los requisitos de las tablas 7.1 y 7.2. La línea verde claro muestra la trayectoria deseada del centroide del grupo, es decir, $c(\varpi)$; mientras que el círculo blanco representa el valor de $\sigma_{g,d}$. El sistema de referencia CENTROIDE se representa con un triángulo verde transparente y dos líneas del mismo color. Este sistema de referencia es el que se utiliza para calcular las posiciones deseadas de cada barco en la formación.

De cada uno de los barcos se representa su posición y orientaciones actuales, así como la trayectoria que ha seguido. Adicionalmente, las líneas amarillas representan la velocidad deseada por la tarea de guiado del grupo, es decir, $\mathbf{v}_{g,d}$; las líneas de color cian muestran el comando de velocidad de la tarea de formación, $\mathbf{v}_{f,d}$. La velocidad resultante del control jerárquico NSB se representa como una línea morada.

Se han incluido cuatro escenarios distintos para probar y estudiar distintas partes del sistema. Durante el desarrollo se han hecho innumerables pruebas que no pueden ser incluidas en este Trabajo. De cada escenario, el simulador genera varios ficheros con los datos de cada barco y con datos generales de la simulación. Cada columna de cada fichero es un conjunto de datos digno

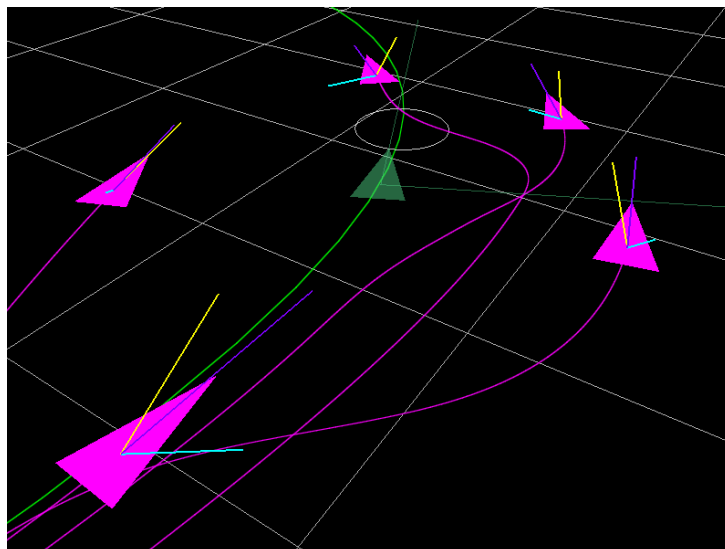


Figura 7.3: Captura de pantalla del simulador.

de estudio y cuyo análisis podría ser parte de la memoria; sin embargo, por razones de brevedad, solo se ha incluido el subconjunto de datos que tiene más relevancia en cada situación.

El primer escenario es el más simple: se trata de dos barcos a lo largo de una curva de Bézier cúbica. Los barcos tienen una única formación. No existen cambios de formaciones y la curva es suficientemente *suave* como para que mantenerse perpendiculares a la misma sea una tarea, a priori, simple. En el mismo escenario se estudian asuntos de inicialización y aflora la utilidad del algoritmo de subastas.

El segundo escenario añade otros tres barcos, hasta un total de cinco, y se centra en el estudio de las *resubastas* y la revisión de los contratos otorgados por el subastador. Aparece uno de los problemas clásicos en el control de formaciones: la inicialización de la formación. En este escenario se produce un cambio de formación para analizar cómo responde el sistema.

El siguiente escenario estudia la estabilidad del método cuando las tareas entran en conflicto. En concreto, se centra en un caso importante, que es la descripción de formaciones cuyo centroide es $\mathbf{c} = [0 \ 0]^T$. Se trata de una trayectoria en línea recta con distintos cambios de formación.

El último escenario intenta poner a prueba todo el sistema con una trayectoria que contiene dos grandes giros. Los cambios de formación se hacen coincidir justo con ambos giros, en aras de hacer la tarea un poco más complicada.

7.2.1. Caso simple, dos barcos en paralelo

El primero de estos escenarios pretende demostrar y comprobar la implementación de los sistemas de Guiado y Control. Desde el punto de vista del sistema de Guiado, se pretende comprobar que las trayectorias utilizando splines funcionan bien con el control jerárquico NSB. El sistema de trayectorias con splines está descrito en la Sección 6.2.1, mientras que el control NSB en la Sección 6.2.2. El escenario está compuesto de dos barcos y la trayectoria determinada por los puntos de control

$$\mathbf{P}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{P}_1 = \begin{bmatrix} 70 \\ 0 \end{bmatrix} \quad \mathbf{P}_2 = \begin{bmatrix} 60 \\ 60 \end{bmatrix} \quad \mathbf{P}_3 = \begin{bmatrix} 80 \\ 80 \end{bmatrix} \quad (7.1)$$

El centroide de ambos barcos debe seguir esa trayectoria. La formación deseada es que los barcos vayan uno al lado del otro, en una línea perpendicular a la trayectoria. El parámetro para el control NSB es

$$\sigma_{f,d} = \begin{bmatrix} d & -d \\ 0 & 0 \end{bmatrix} . \quad (7.2)$$

donde $2d$ es la distancia entre los barcos, en este caso 12 m. En este escenario se han desactivado las subastas. La Figura 7.4a muestra las trayectorias seguidas por ambos barcos en rojo y azul, en negro la trayectoria del centro del grupo (trazo continuo) así como la trayectoria deseada (trazo discontinuo). Cualitativamente, se observa que los dos barcos son capaces de cumplir sus objetivos tanto con la tarea de formación como la de guiado del grupo. La Figura 7.4b muestra la norma de los errores de formación a lo largo de la simulación, es decir, $\|\tilde{\sigma}_f\|$. El error decrece y el sistema consigue mantenerlo en valores inferiores a dos metros durante el resto de la simulación.

Aunque pueda parecer que la convergencia de la formación es lenta, es importante hacer notar el hecho de que los barcos parten del reposo y deben acelerar con arreglo a la agilidad de su propulsor. Los valores del control de velocidad están representados en la Figura 7.4c; donde se observa que durante los primeros 18 segundos el barco azul está intentando conseguir la consigna impuesta por el sistema de Guiado. Finalmente, la Figura 7.4d muestra los valores del control de rumbo. Se observa que las consignas de rumbo son, generalmente, paralelas, como cabe esperar debido a la configuración del escenario.

En las figuras mencionadas anteriormente se puede observar un fenómeno que ya fue discutido en la Sección 6.2.2 (Control jerárquico *Null-Space-Based* (NSB)). Las consignas de rumbo y velocidad que genera el control NSB reparten las tareas de forma equitativa. Si es necesario que un barco vaya más rápido, hace que el otro vaya más lento también.

Es fácil ver la necesidad del sistema de subastas si intercambiamos las matrices de la formación. Supongamos que ahora

$$\sigma_{f,d} = \begin{bmatrix} -d & d \\ 0 & 0 \end{bmatrix} . \quad (7.3)$$

Desde el punto de vista semántico, la formación es la misma: los barcos deben formar una línea perpendicular a la trayectoria. Lo que se conoce, en el ámbito militar, como una formación en columna. Si volvemos a ejecutar la simulación, aún con el sistema de subastas desactivado, obtenemos los resultados representados en la Figura 7.5. Se puede ver como, al empezar los barcos *cruzados*, el comportamiento es claramente subóptimo.

La Figura 7.5a muestra las posiciones de este escenario. Puede verse cómo al principio la trayectoria del centroide se separa de la trayectoria deseada. Al haber intercambiado las posiciones de la formación, los barcos se cruzan. Como se asume que la flota es homogénea, es decir, que todos los barcos son iguales, los cruces de trayectorias deberían evitarse si existe una solución mejor. La Figura 7.5b muestra cómo el error de formación al principio crece, mientras los barcos se cruzan, y luego el comportamiento del sistema es igual que antes.

Una curiosidad de este escenario es que, al cruzarse, el barco azul activa su tarea de evitar obstáculos durante el intervalo de tiempo [15, 18). El resultado de esto puede verse tanto en la Figura 7.5c como en la Figura 7.5d. En ambas se puede apreciar como las consignas de velocidad y rumbo dan un cambio brusco para evitar el choque entre los barcos.

Para evitar este problema, se puede activar el sistema de subastas. Este sistema ha sido descrito de forma general en el Capítulo 4 (El problema de la asignación y el algoritmo de subastas) y de forma más específica en la Sección 6.2.3 (Subastas para formaciones). Cuando

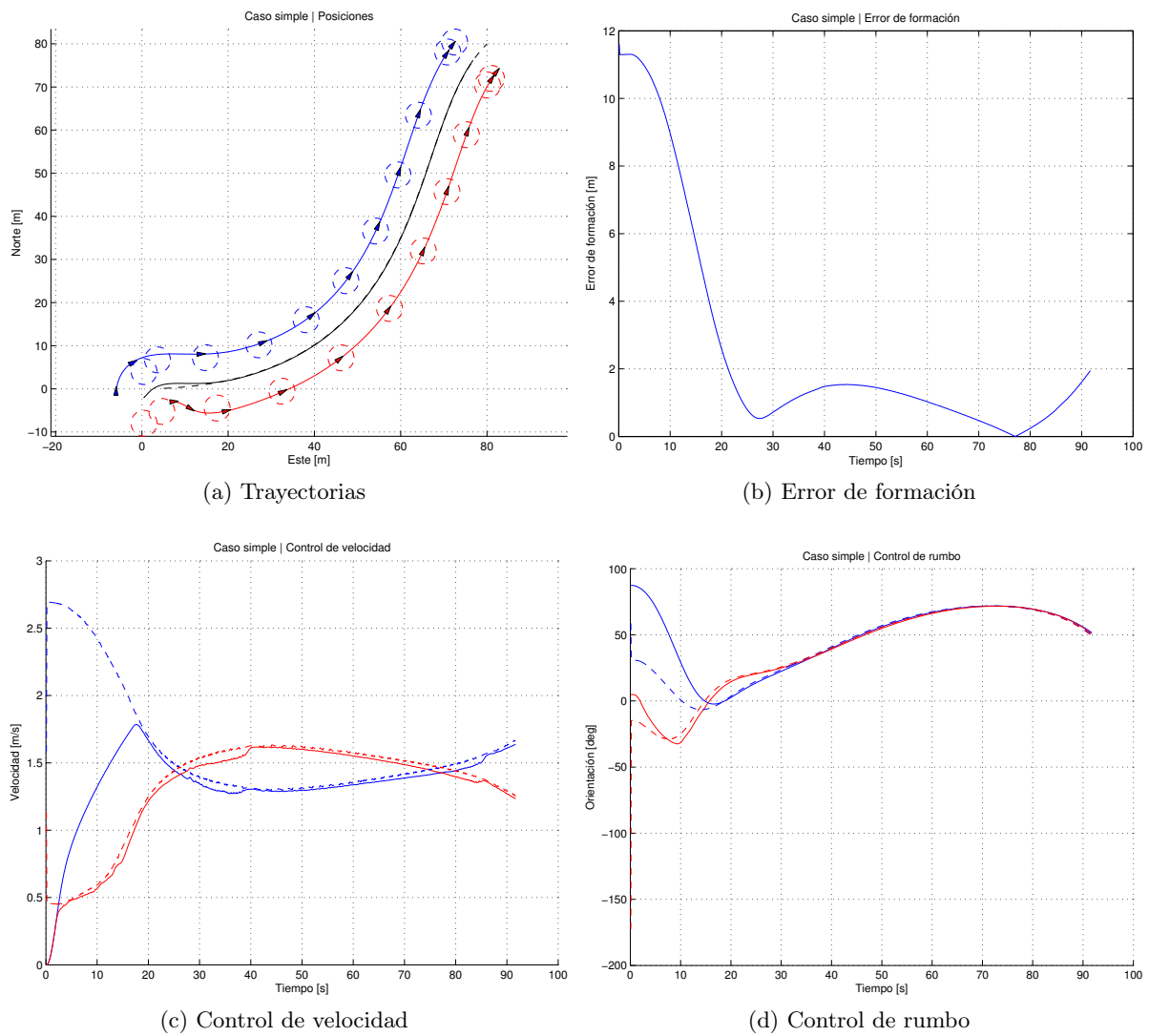


Figura 7.4: Escenario simple, dos barcos en paralelo. Sin subastas.

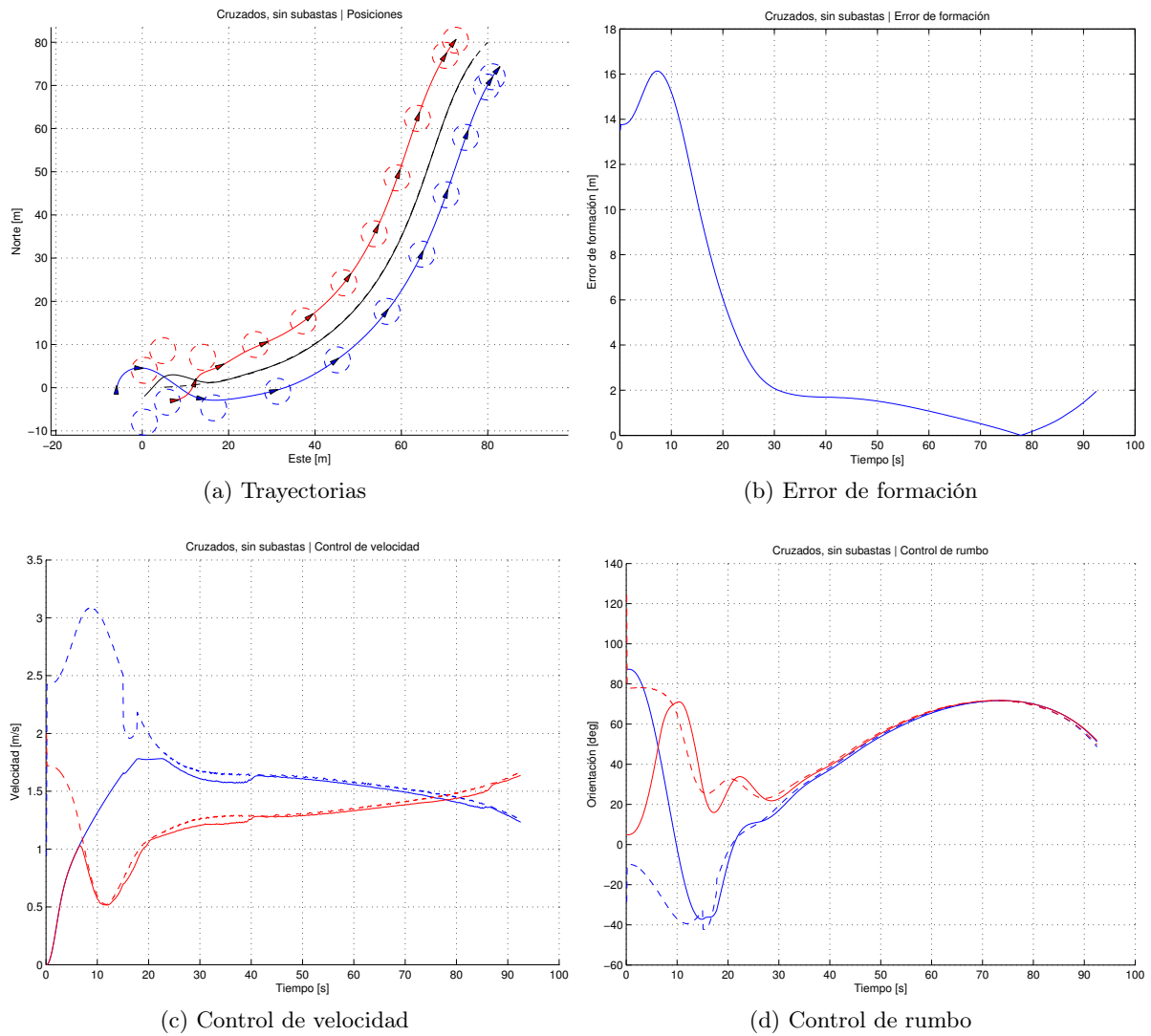


Figura 7.5: Escenario simple, dos barcos en paralelo. Empiezan cruzados. Sin subastas.

se activa una formación se lleva a cabo el proceso de la subasta y los barcos acuerdan la mejor configuración.

Los resultados del escenario con las subastas activadas se muestra en la Figura 7.6. Los resultados son idénticos a los del primer caso, en el que no se habían intercambiado las posiciones de la formación. De hecho, los dos casos tardan exactamente 91.8 segundos de simulación. Introducir el algoritmo de las subastas ha independizado al sistema del orden en el que se escriban las posiciones a la hora de diseñar $\sigma_{f,d}$. En la Figura 7.6a se puede ver como, en el primer momento, la formación forzaría al cruce de los barcos, sin embargo, en cuanto se lleva a cabo la subasta, el sistema de Guiado decide intercambiar las posiciones de la formación.

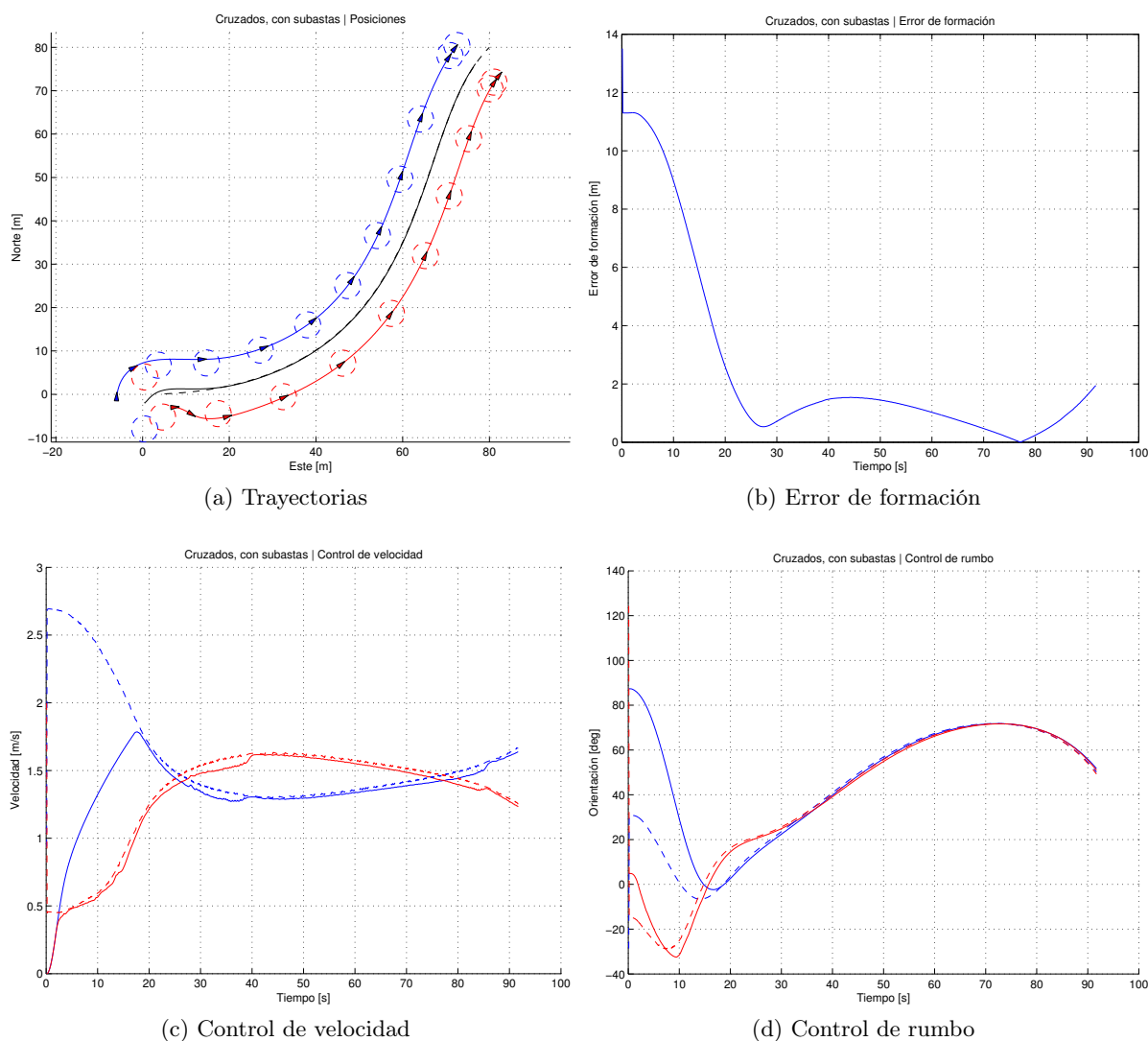


Figura 7.6: Escenario simple, dos barcos en paralelo. Empiezan cruzados. Con subastas.

Aunque el escenario no deja de ser bastante simple, captura toda la esencia del sistema propuesto. No obstante, queda por evaluar la utilidad de las *resubastas* y estudiar, en un poco más de detalle, las iteraciones que necesita el algoritmo para llegar a una solución.

7.2.2. Efecto de las *resubastas*

El siguiente escenario tiene como objetivo evaluar el efecto de las *resubastas* en el sistema de Guiado. La *resubasta* es un concepto producto de la metáfora económica utilizada en una parte del sistema; se explican en detalle en la Sección 6.2.3 (Subastas para formaciones). El escenario consta de una flota de cinco barcos que deben moverse a lo largo de una línea recta. La formación inicial es que todos los barcos vayan en fila india a lo largo de la trayectoria. En el ámbito militar se conoce a esta formación como *hilera*. Al minuto de comenzar el escenario, se comanda un cambio de formación para que los barcos formen en columna. Las formaciones pueden describirse de la siguiente forma:

$$\mathbf{hilera}_5 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -2d & -d & 0 & d & 2d \end{bmatrix} \quad \mathbf{columna}_5 = \begin{bmatrix} -2d & -d & 0 & d & 2d \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (7.4)$$

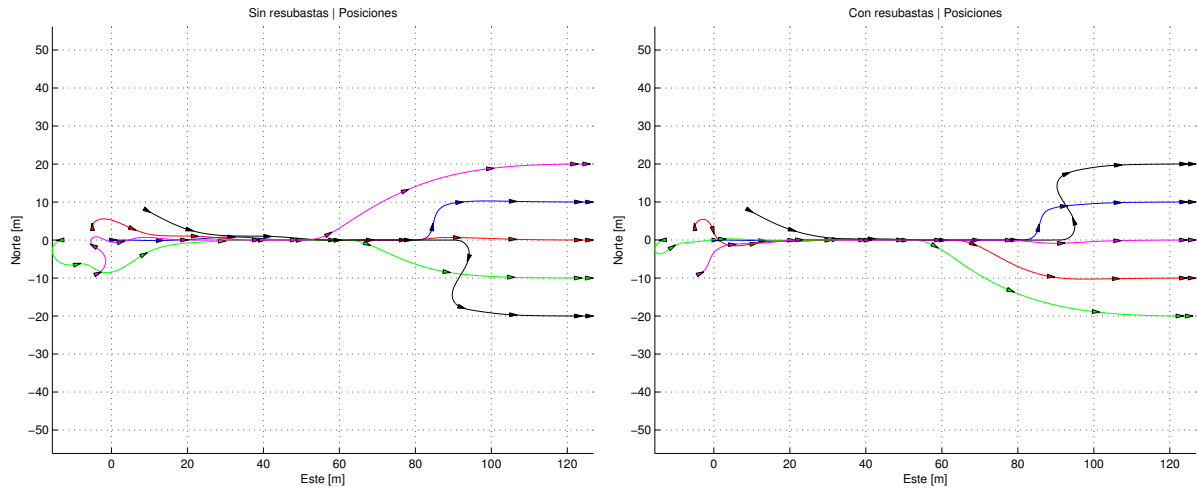
El escenario tiene dos momentos críticos desde el punto de vista del control de la formación: el inicio y el cambio de forma geométrica. La Figura 7.7 muestra las trayectorias tanto en el caso de tener las *resubastas* activadas como desactivadas. Puede observarse que las trayectorias son muy parecidas en ambos casos, a excepción del inicio de la misión. En el caso de la simulación con *resubastas* se ha utilizado un tiempo de revisión de contratos de 5 segundos y un umbral de error de formación de 10 metros. En la Figura 7.7c se incluye la comparación del error de formación en ambos casos. Se puede ver perfectamente como a los cinco segundos tiene lugar una revisión del contrato tras la cual se encuentra un acuerdo mejor.

La convergencia de la segunda formación es igual de buena en ambos casos, aunque las trayectorias resultantes no sean exactamente las mismas. Sin embargo, la fase inicial es distinta. Se puede observar que a los 20 segundos hay una diferencia de unos 7 metros en los errores de formación. La Figura 7.8 muestra la fase inicial ampliada para que sea más fácil ver el efecto de la *resubasta*.

La Figura 7.7d muestra el número de iteraciones de la última vez que se ejecutó el algoritmo de subastas. Se observa que en el caso de las *resubastas* desactivadas, línea azul, solo hay dos subastas en toda la simulación, una al inicio y una al cambio de formación. Sin embargo, en la línea roja, se observa cómo el algoritmo de subastas se ejecuta varias veces; durante la fase inicial, se ejecuta en el segundo 5 y en el 10. Al revisar el estado de la formación en $t = 15$, el error de la formación $\|\tilde{\sigma}_f\|$ es menor que el umbral, de forma que no se lleva a cabo la *resubasta*. Al cambiar la formación (tras el segundo 60), se llevan a cabo distintas revisiones del error de formación. Sin embargo, el método no consigue ninguna solución que mejore la configuración previamente acordada.

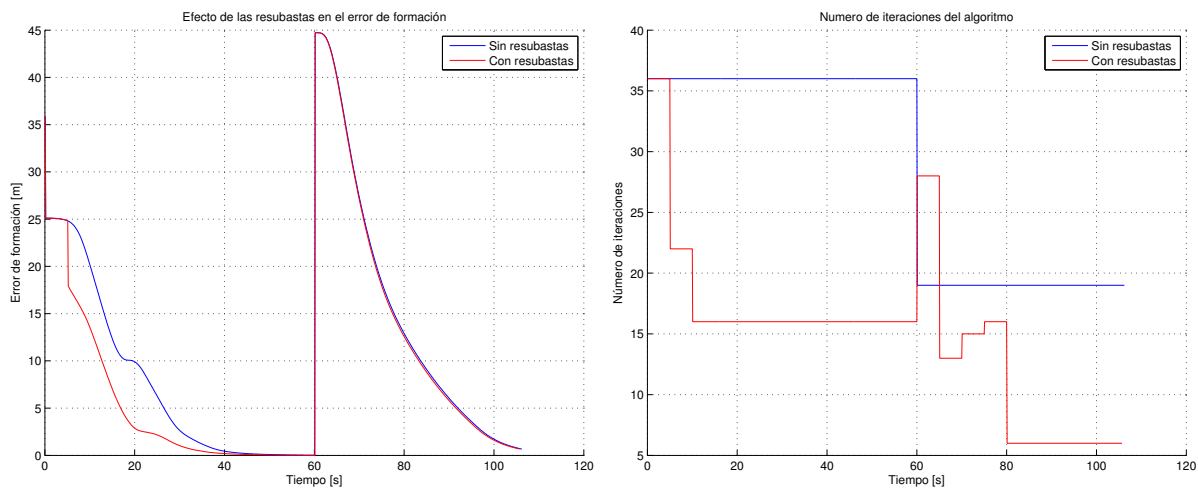
Se observa que las diferencias están entre los barcos rojo, verde y magenta. En ambos casos, los barcos azul y negro siguen la misma trayectoria y son siempre asignados a los mismos huecos de la formación. Para los otros tres barcos la solución inicial puede estar sesgada por las orientaciones de los barcos verde y rojo. Realmente el problema está en la métrica, pero dada la dificultad de encontrar una métrica que cumpla todos los requisitos necesarios¹, las *resubastas* son un método elegante de reducir el impacto de las deficiencias de la métrica así como de los elementos externos. En este escenario, la *resubasta* encuentra una configuración que mejora la convergencia global del error de formación.

¹Esto se describe en más detalle en la Sección 6.2.3 (Subastas para formaciones).



(a) Trayectorias sin *resubastas*

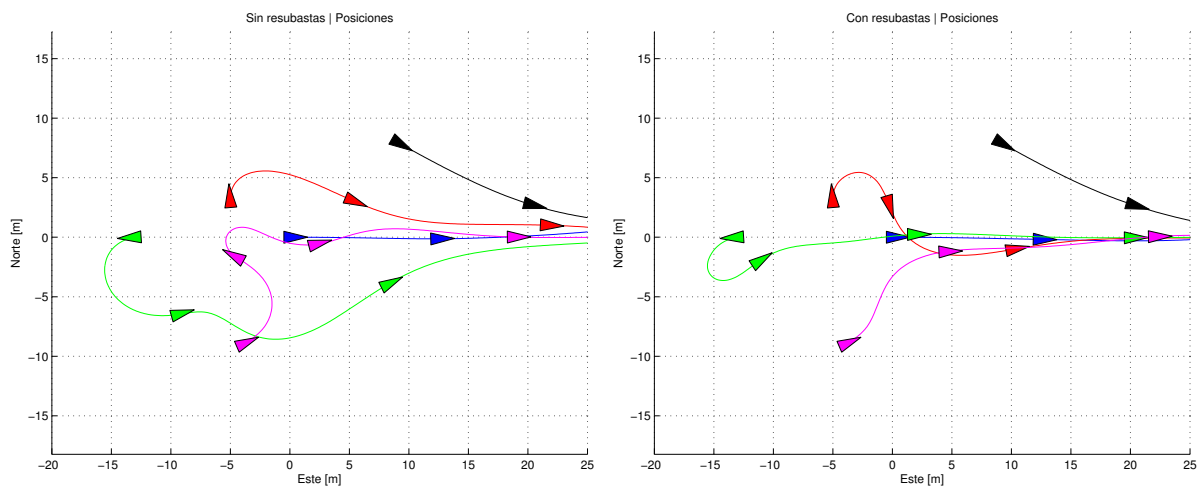
(b) Trayectorias con *resubastas*



(c) Error de formación

(d) Iteraciones del algoritmo

Figura 7.7: Escenario para probar las *resubastas*.



(a) Trayectorias sin *resubastas*

(b) Trayectorias con *resubastas*

Figura 7.8: Fase inicial del escenario

7.2.3. Formaciones descentradas

Una de las afirmaciones que se hizo en la descripción del control NSB, en la Sección 6.2.2 (Control jerárquico *Null-Space-Based* (NSB)), es que es capaz de cumplir las tareas siempre y cuando estas no entren en conflicto entre si. Es importante estudiar mediante qué ocurre cuando las tareas de guiar al grupo y formación no son completamente ortogonales. Es posible hacer que las dos tareas entren en conflicto si se utilizan formaciones que no estén centradas, es decir, que cuyo centroide no corresponda con el origen del sistema de coordenadas en el que se describen. Este es un problema común ya que no siempre es fácil describir las formaciones de forma centrada si lo hace un operador humano.

El escenario consta de cinco barcos navegando en línea recta. Se utilizan distintas formaciones, algunas centradas, y otras descentradas. Al inicio se hace que los barcos formen en hilera, en $t = 40$ se les hace ir en cuadrado, en $t = 90$ se cambia a una formación de tipo delta. Un minuto después, en $t = 150$ se invierte la delta, formando la configuración de tipo tenaza, en $t = 200$ forman en círculo y, por último, en $t = 250$ forman en escuadra o *ele*. Las nuevas formaciones se pueden describir de la siguiente forma:

$$\mathbf{cuadrado}_5 = \begin{bmatrix} 0 & d & d & -d & -d \\ 0 & d & -d & d & -d \end{bmatrix} \quad \mathbf{delta}_5 = \begin{bmatrix} 0 & -d & d & -2d & 2d \\ d & 0 & 0 & -d & -d \end{bmatrix} \quad (7.5)$$

$$\mathbf{tenaza}_5 = \begin{bmatrix} 2d & -2d & -d & 0 & d \\ d & d & 0 & -d & 0 \end{bmatrix} \quad \mathbf{círculo}_5 = \begin{bmatrix} -d & d & 2d & 0 & -2d \\ 2d & 2d & 0 & -d & 0 \end{bmatrix} \quad (7.6)$$

$$\mathbf{escuadra}_5 = \begin{bmatrix} d & d & d & 0 & -d \\ -d & 0 & d & d & d \end{bmatrix} \quad (7.7)$$

Dos de las formaciones (hilera y cuadrado) se han descrito de forma centrada, de forma que se espera que el control haga converger las posiciones de los barcos a las posiciones deseadas de la formación. Por otro lado, las formaciones delta, tenaza y círculo han sido desplazadas en el eje de avance de la formación. Por último, la formación de escuadra ha sido desplazada en ambos ejes. En estos casos, las tareas de guiado de grupo y formación entran en conflicto. La Figura 7.10a muestra el error de formación $\|\tilde{\sigma}_f\|$ durante la simulación. El sistema es capaz de llevar el error a 0 en las formaciones centradas, pero no así en aquellas que hacen que las dos tareas entren en conflicto. En estos casos, el sistema simplemente busca el punto de estabilidad y lo mantiene. En el caso de la última formación el error en el que se estabiliza el sistema es mayor porque el desplazamiento de la formación es mayor.

La Figura 7.9 muestra el detalle de la convergencia de los barcos a la formación deseada. Se puede observar en las figuras 7.9a y 7.9b que la convergencia de las formaciones centradas. En las figuras 7.9b, 7.9c y 7.9d los barcos quedan algo desfasados con respecto a sus posiciones de referencia en la formación. De hecho, quedan un poco atrasados o adelantados, según si el centroide de la formación queda a un lado o al otro del sistema de referencia en el que se describe la misma. Por último, la Figura 7.9f muestra el caso de la formación desfasada en ambos ejes. En este caso, se puede ver cómo los barcos quedan desplazados tanto en Este como en Norte.

En la práctica, esto no es nada más que un problema *estético*, es posible hacer que el error sea cero si se centra o *normaliza* la formación introducida por el usuario. Esto es tan simple como restar el centroide a todas las posiciones de la formación. En caso de pedirle al sistema que normalice las formaciones por nosotros, se obtiene el comportamiento deseado, como puede verse en la Figura 7.10b. Por razones de completitud, se incluye la convergencia de las formaciones círculo y escuadra, una vez que las normaliza el sistema, en las figuras 7.11a y 7.11b, respectivamente.

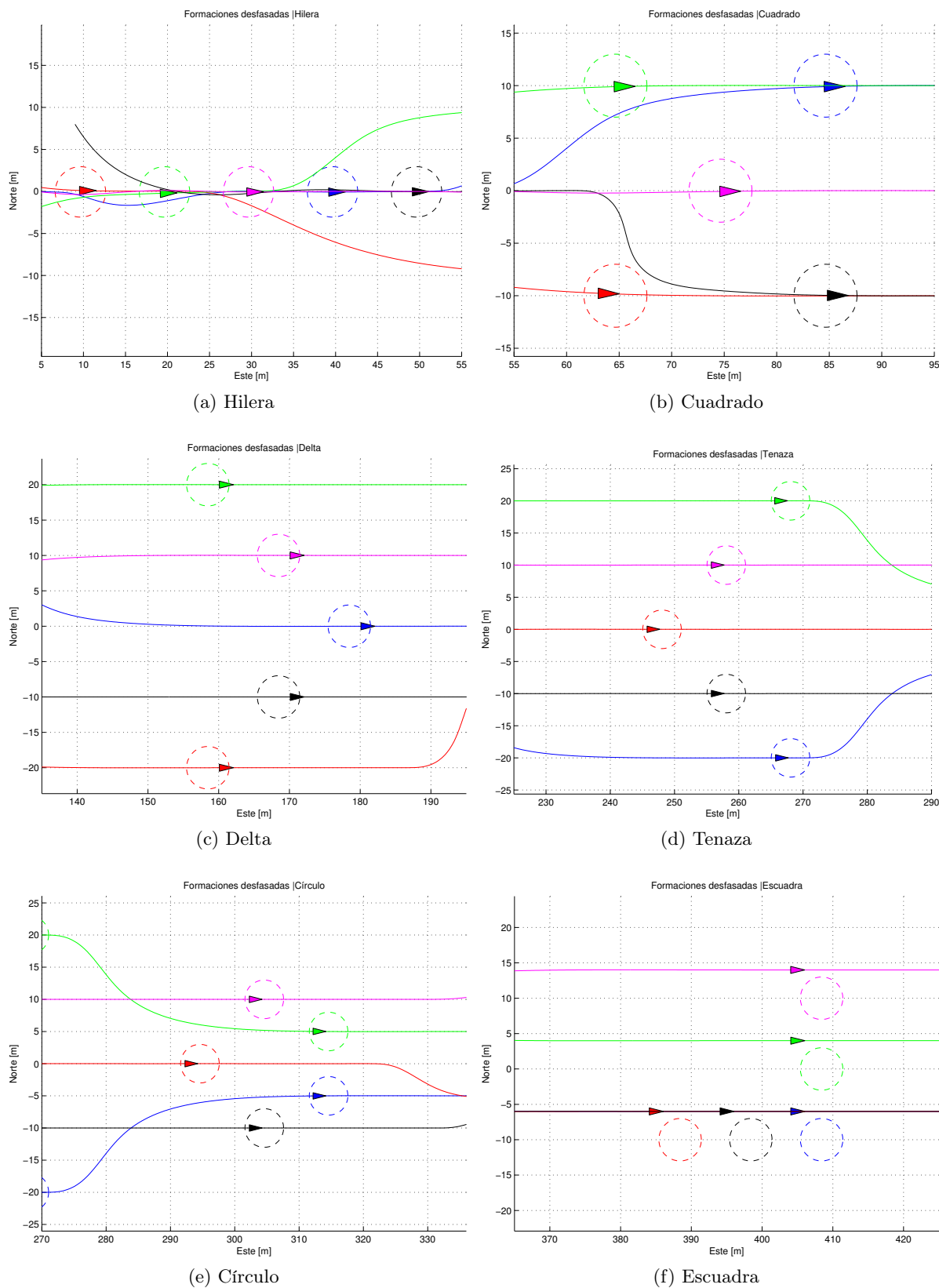


Figura 7.9: Convergencia a formaciones centradas y descentradas.

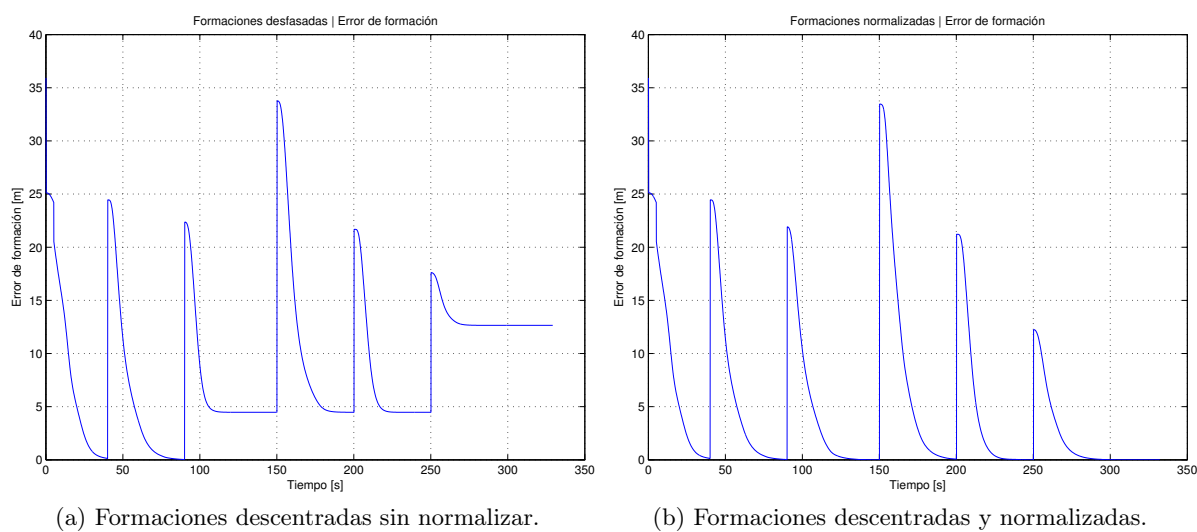


Figura 7.10: Efecto de la normalización de las formaciones en $\|\tilde{\sigma}_f\|$.

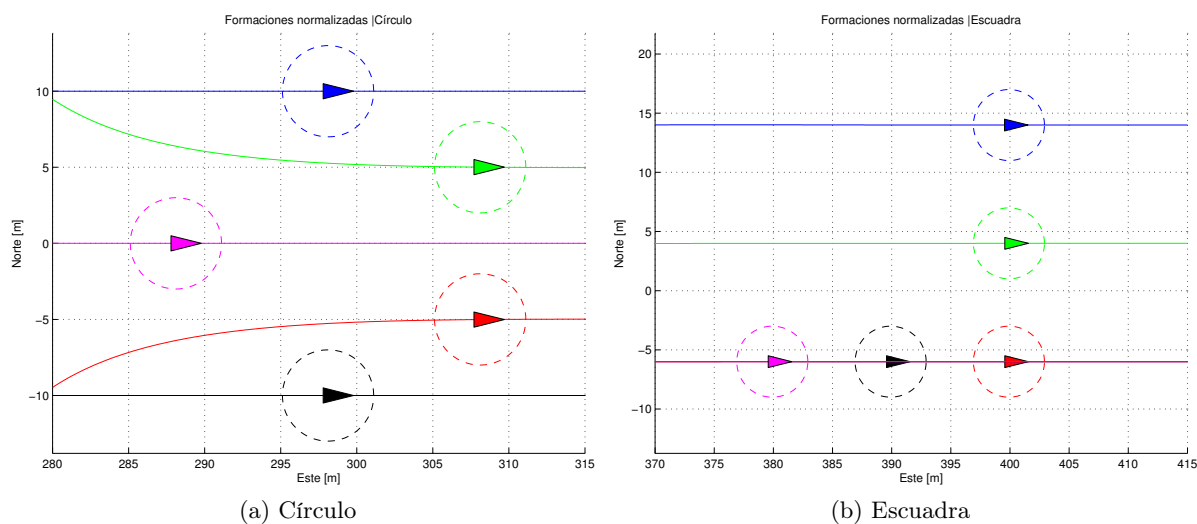


Figura 7.11: Convergencia a formaciones descentradas y luego normalizadas.

7.2.4. Escenario completo

Los tres primeros escenarios han servido para validar y estudiar distintos aspectos de los sistemas de Guiado, Navegación y Control. El siguiente escenario tiene como objetivo comprobar el sistema con todos los elementos a la vez. Está compuesto de cinco barcos que cambian de formación a lo largo de la trayectoria definida por los siguientes puntos de control

$$\mathbf{P}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{P}_1 = \begin{bmatrix} -120 \\ 30 \end{bmatrix} \quad \mathbf{P}_2 = \begin{bmatrix} 210 \\ 300 \end{bmatrix} \quad \mathbf{P}_3 = \begin{bmatrix} 30 \\ 300 \end{bmatrix} \quad \mathbf{P}_4 = \begin{bmatrix} -60 \\ 225 \end{bmatrix} \quad \mathbf{P}_5 = \begin{bmatrix} -60 \\ 150 \end{bmatrix} \quad (7.8)$$

La curva utiliza el método de unión propuesto en la Sección 6.2.1 (Splines para trayectorias). Los barcos hacen las siguientes formaciones: hilera, delta, hilera, círculo y cuadrado. Los tiempos de cambio entre las formaciones se han ajustado para que coincidan con las zonas de mayor curvatura de la trayectoria, para hacer la tarea un poco más difícil al sistema.

La Figura 7.12a muestra la trayectoria completa de los barcos a lo largo de la simulación. El error de formación $\|\tilde{\sigma}_f\|$ está representado en la Figura 7.12b. Se puede observar que el sistema no siempre consigue mantener el error a cero debido a que la consigna de formación está constantemente cambiando debido a la curvatura de la trayectoria. Sin embargo, se puede apreciar que el error es muy bajo, casi nulo. El efecto de la curvatura en la tarea de mantener la formación se puede observar en las figuras 7.12c y 7.12d. En ambas se puede ver cómo los barcos van en la formación deseada, pero con una leve rotación. Dado que el movimiento lateral no es controlable en los barcos², la convergencia es mucho más lenta y difícil.

El escenario utiliza el método de extensión de curvas de Bézier cúbicas propuesto anteriormente para representar la trayectoria de la flota. La unión entre las curvas ocurre en la parte más al Norte de la trayectoria. Se puede ver que la transición es completamente suave, como es de esperar dado que el método garantiza, matemáticamente, la continuidad y suavidad de la unión.

²La velocidad lateral no es controlable porque el modelo es subactuado, el movimiento lateral y longitudinal están fuertemente acoplados.

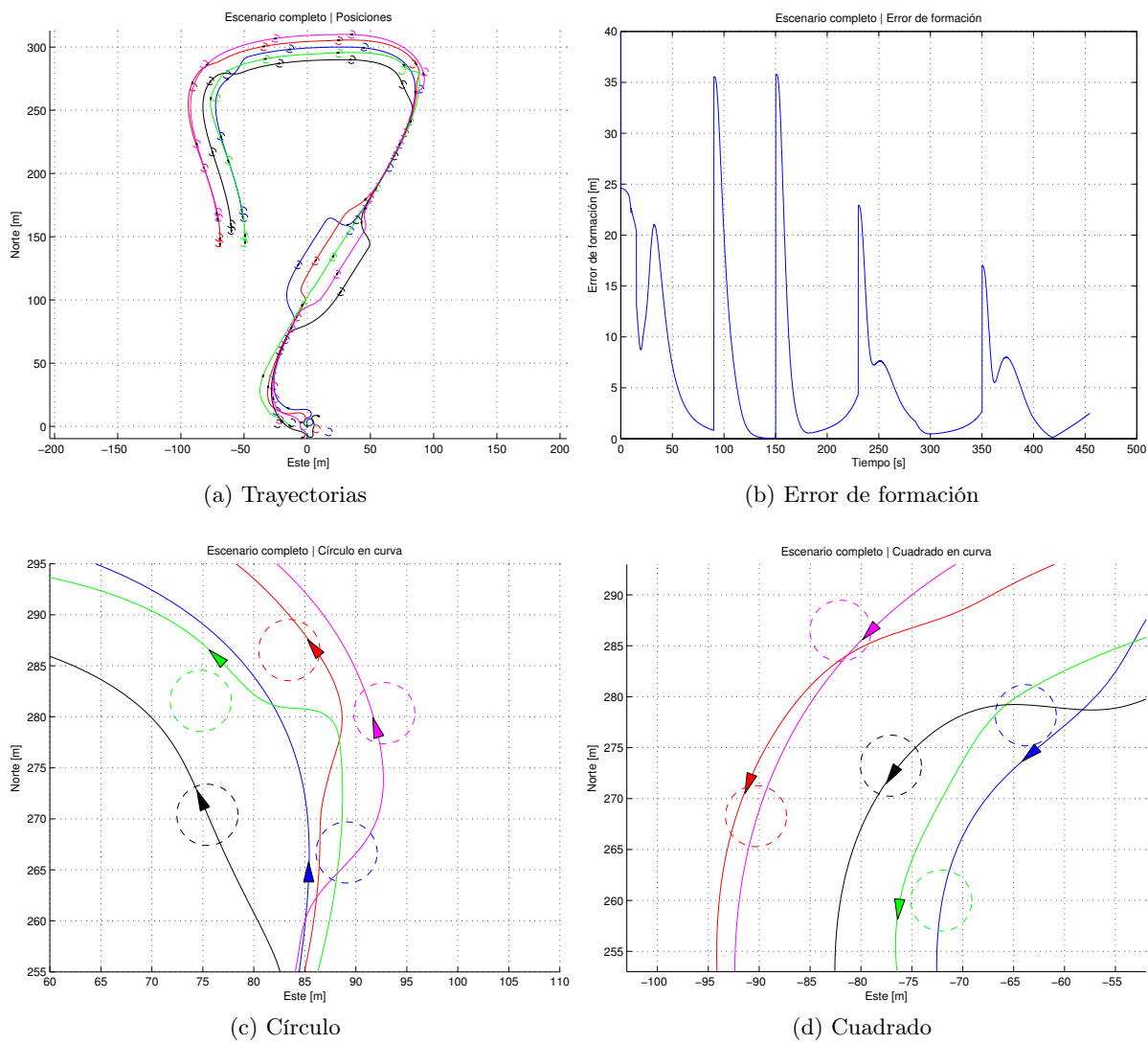


Figura 7.12: Escenario completo.

7.3. Simulaciones realistas

Los escenarios simulados y analizados en la sección anterior han permitido comprobar la viabilidad del sistema propuesto. Adicionalmente, han permitido estudiar algunos aspectos y consecuencias del sistema. Sin embargo, el sistema ha trabajado en un mundo ideal. Las señales no tienen ruido, y los sistemas tienen los datos con la frecuencia que los necesitan. En un experimento real, esto no es así. Los sensores tienen ruido y solo ofrecen datos con una frecuencia determinada. Para obtener simulaciones más fiables se han realizado *simulaciones realistas*.

Esta sección pretende mostrar resultados en simulación donde los sistemas no tienen toda la información que querrían. La brújula se actualiza con una frecuencia de 10Hz y tiene un ruido medio de 1° . El sensor de velocidad de avance se actualiza también con una frecuencia de 10Hz y su ruido medio es de $0.1m/s$. El GPS da una medida nueva cada segundo con un error de $5m$. Aunque los ruidos tanto de la brújula como del sensor de velocidad puedan parecer pequeños, realmente son de más del doble de lo que muestran los datos experimentales en (Pereda, 2010); el ruido del GPS es muchísimo más grande de lo que podría ser, ya que el error dominante en este tipo de sensores es sistemático de forma que el error entre dos medidas es bajísimo. Sin embargo, se ha intentado hacer que el sistema tenga suficiente holgura para que, a la hora de llevarlo a una plataforma experimental los ruidos no sean un gran problema.

En los escenarios anteriores el sistema de Navegación estaba desactivado, dado que los sistemas tenían conocimiento exacto de los estados. Estas simulaciones en escenarios realistas pondrán a prueba el sistema de Navegación: tanto el subsistema de filtrado como el del estimador de posición. Los escenarios reales permiten validar este sistema.

Las clases del diseño que habilitan la simulación realista pertenecen a los componentes *Algoritmos GNC* y *Física y sensores*. La descripción de estos sistemas se encuentra en la Sección 6.1 (Navegación). En concreto, estas clases son:

- `SimulatedShip` es la clase que actúa como un barco real, utiliza el resto de clases para simular los sensores y utiliza las clases de filtros y estimador.
- `SensorNoise` permite generar el ruido para un sensor.
- `MedianFilter` implementa el filtro de la mediana.
- `UT`, `UKF` y `UKFVessel` implementan el estimador de posición basado en UKF.

En todos los escenarios realistas están activadas tanto las subastas, como las resubastas, como la normalización de formaciones. Esto permite estudiar el sistema de Navegación. El filtro estimador ha sido descrito de forma general en el Capítulo 5 (El filtro de Kalman *Unscented*) y desarrollado de manera particular en la Sección 6.1.2 (Estimador).

Los escenarios son análogos y similares a los presentados en la sección anterior. El primer escenario está enfocado a comprobar la inicialización del estimador y el funcionamiento general en un caso sencillo. El segundo escenario tiene como objetivo observar el efecto de un cambio de formación en el estimador. Por último, se repite el escenario completo para comprobar cómo se comporta el estimador a lo largo de una simulación completa.

7.3.1. Caso simple, dos barcos en paralelo

El primer escenario realista es muy similar al primer escenario en simulación. Se trata de dos barcos que tienen que navegar en paralelo, formando una columna perpendicular a la dirección de la trayectoria.

Para que la situación sea más realista, el filtro estimador de cada barco³ se inicializa a cero, utilizando los parámetros

$$\hat{\mathbf{x}}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{P}_0 = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix} \quad (7.9)$$

De este modo, el UKF tiene incertidumbre total sobre la posición inicial. Evidentemente, necesitará hacer algunos pasos de corrección para refinar la estima; de forma que las estimas iniciales serán erróneas. Al ser erróneas para todos los barcos, las primeras consignas del control NSB no tendrán mucho sentido hasta que el filtro converja a unos valores razonables.

La Figura 7.13a muestra las trayectorias reales de los dos barcos y las medidas del sensor GPS simulado como una nube de puntos. Se puede observar cómo los ruidos son bastante grandes y, sin embargo, el sistema de Navegación es capaz de dar buenas señales a los sistemas de Guiado y Control.

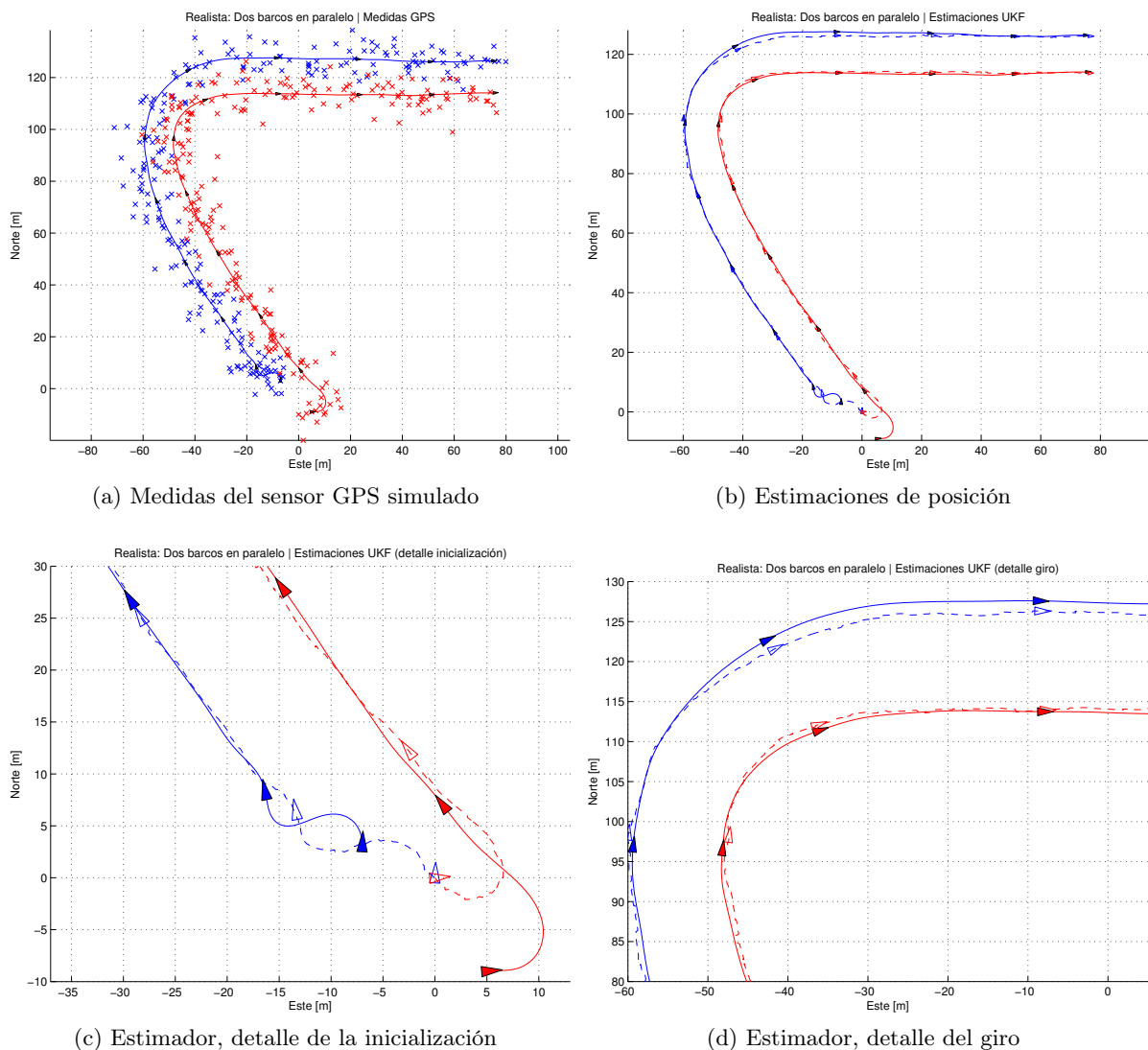


Figura 7.13: Resultados del estimador en un escenario simple.

³Es importante recordar que existe un sistema de Navegación por cada barco. Estos sistemas son completamente independientes y no se comunican los unos con los otros.

La Figura 7.13b muestra la trayectoria real de cada barco con un trazo continuo y la trayectoria estimada utilizando un trazo discontinuo. Se observa que las estimaciones tienen bastante poco error durante casi toda la maniobra. Sin embargo, hay dos partes de la simulación que presentan problemas para el estimador: la inicialización y el giro. El error en la fase de inicialización es normal por lo comentado anteriormente, el estimador se inicia sin ningún conocimiento *a priori* de la posición y necesita un tiempo de convergencia hasta que las estimas empiezan a parecerse a la realidad. Durante el giro, el modelo de propagación del estimador es menos preciso, ya que carece de información sobre el desplazamiento lateral; es normal que las estimas generadas se ajusten menos a la posición real del barco. La Figura 7.13c muestra el detalle de la fase de la inicialización, mientras que la Figura 7.13d muestra el detalle del giro.

7.3.2. Cambio de formación

Este escenario es análogo al presentado en la Sección 7.2.2 (Efecto de las *resubastas*), sin embargo, en este caso se va a utilizar para estudiar el efecto de un cambio de formación en un escenario real. La inicialización de la formación es un problema importante, ya que los barcos, literalmente, no saben donde están. El sistema de Guiado tiene, por tanto, entradas erróneas, y puede generar comandos no válidos.

La Figura 7.14a muestra las medidas del sensor GPS durante la simulación, mientras la Figura 7.14b muestra las estimaciones del UKF. Se observa cómo el filtro es capaz de ajustarse bastante bien a los estados simulados aún con los ruidos de los sensores. La inicialización, reproducida en la Figura 7.14c, es un poco caótica, debido a los errores en las estimas de los barcos; en concreto, las peores son las de los barcos negro, magenta y verde. Por otro lado, la estima del barco azul es muy buena debido a que realmente empieza en la posición $[0 \ 0]$. No obstante, una vez que el estimador ha convergido, las estimaciones son muy buenas. Esto queda patente en el cambio de formación, representado en la Figura 7.14d, se observa que no supone un gran problema para el estimador, que es capaz de mantener unas estimas razonables y no diverge.

7.3.3. Escenario completo

Por razones de completitud, se incluye el escenario completo presentado en la Sección 7.2.4 (Escenario completo), esta vez, utilizando simulaciones realistas. La Figura 7.15a muestra las medidas del sensor GPS simulado. Se observa cómo, en ciertas zonas, las medidas de los barcos se confunden completamente entre sí. Sin embargo, tal y como se aprecia en la Figura 7.15b, el estimador hace unas estimas buenas a lo largo de toda la trayectoria.

La Figura 7.15c muestra el detalle de la inicialización de la primera formación. Se observa que las estimas de los barcos negro y verde son particularmente malas, mientras que las de los barcos azul y rojo convergen en poco tiempo. El barco magenta casi ha convergido en la última *instantánea* mostrada en la figura. Adicionalmente, se incluye el error de formación en la Figura 7.15d; este error es muy parecido al de la simulación no realista. Es interesante ver que la configuración final de los barcos no es la misma a la de la simulación no realista; esto es efecto de las *resubastas*.

7.4. Conclusiones

Se ha comprobado la viabilidad del sistema tanto en escenarios simulados como en simulaciones realistas. Se ha obtenido la confianza necesaria en el mismo, de cara a llevar a cabo pruebas

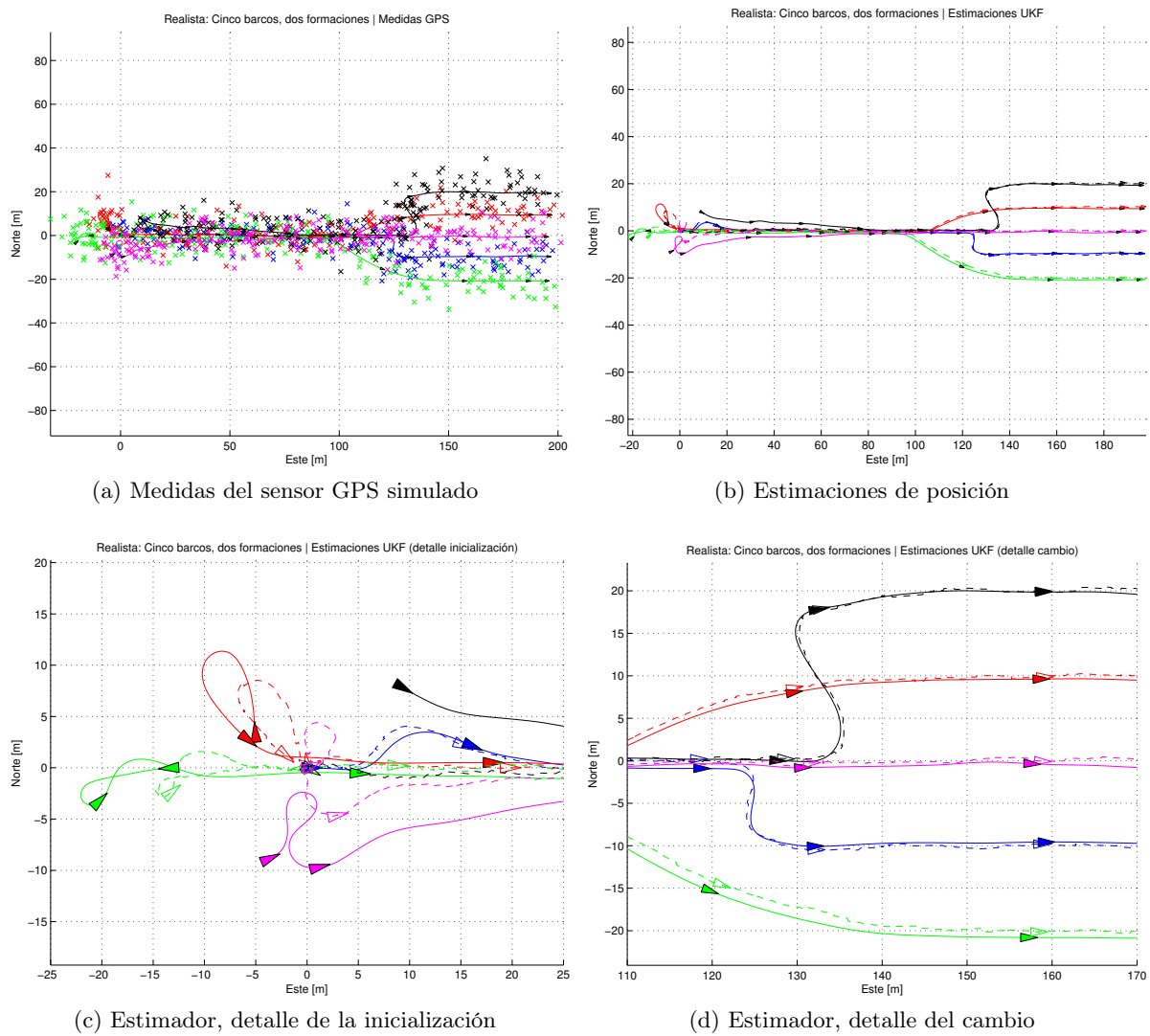


Figura 7.14: Resultados del estimador en un escenario con cambios de formación.

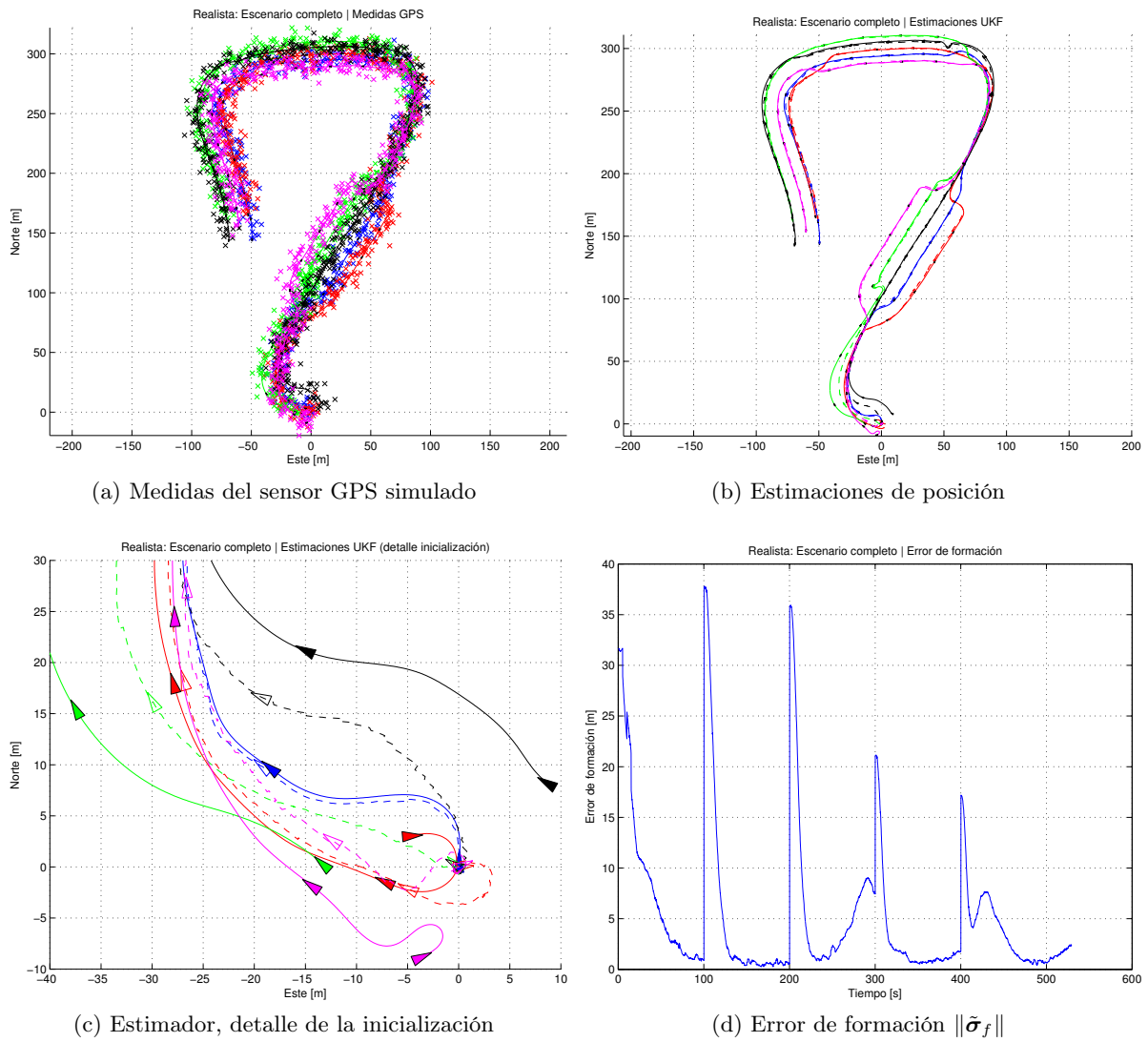


Figura 7.15: Resultados del estimador en un escenario con cambios de formación.

experimentales de los distintos sistemas de alto nivel: Guiado, Navegación y Control. Adicionalmente, haber utilizado modelos reales (medidos y estimados) tanto del barco experimental como de los sensores, ha permitido hacer un primer ajuste de las distintas ganancias involucradas en el Guiado y el Control.

Adicionalmente, las simulaciones permiten ganar confianza no solo en la viabilidad matemática o formal, también en la calidad de la implementación. Los *bugs* en el software se han podido identificar y arreglar antes de tener que hacerlo en el campo experimental.

Un problema que ha aparecido en las simulaciones es la inicialización de la formación en los escenarios realistas. Se acoplan dos cosas: por un lado, no se tiene conocimiento *a priori* sobre las posiciones y orientaciones iniciales; por otro lado, durante los primeros segundos, las estimaciones del UKF están en la fase de convergencia. El efecto de ambos problemas es que las inicializaciones son un poco caóticas. El sistema de las *resubastas* cumple bien su función, disminuyendo el tiempo de convergencia del error de formación.

Los sistemas de Guiado, Navegación y Control cumplen los requisitos impuestos en la Tabla 2.1 de la Sección 2.3 (Requisitos). De este modo, desde el punto de vista de la simulación, se han alcanzado los objetivos propuestos en el Trabajo.

Capítulo 8

Experimentos de Navegación

Este capítulo recoge los datos experimentales del sistema de Navegación. Dado que este es el sistema que trabaja directamente con las señales de los sensores, es el primer candidato a ser probado experimentalmente.

El trabajo experimental consta de la implementación de los algoritmos descritos en el Capítulo 5 (El filtro de Kalman *Unscented*) y en la Sección 6.1 (Navegación). Ambos algoritmos se implementan en C/C++ para poder ser utilizados en el microcontrolador del barco experimental.

La primera sección describe los experimentos y los encuadra geográficamente. Después, una sección discute distintas apreciaciones y consideraciones previas a la presentación de los resultados. La siguiente sección muestra un conjunto de los resultados y su análisis. Por último, se exponen las conclusiones obtenidas de los ensayos experimentales.

8.1. Descripción de los experimentos

El objetivo de los experimentos es validar el sistema de Navegación y hacerse una idea del tipo de señales que recibirá el Guiado. En concreto, se pretende comprobar la utilidad y el rendimiento del estimador de la Navegación. Se han realizado varios experimentos y se han obtenido muchos datos de telemetría. Sin embargo, en aras de la brevedad y la facilidad de lectura, solo se han incluido los resultados más representativos.

El sistema de Navegación se ha descrito en la Sección 6.1 (Navegación). El algoritmo utilizado en el estimador ha sido descrito en el Capítulo 5 (El filtro de Kalman *Unscented*), mientras que los modelos utilizados en el mismo se han discutido en la Sección 6.1.2 (Estimador).

Los experimentos se han llevado a cabo en el *Parque Enrique Tierno Galván* en Madrid. Para representar los datos tanto del sensor GPS como del estimador, se ha utilizado un plano NED con origen en $40.385053^\circ N$, $3.684129^\circ E$ y $630m$ de altitud. La Figura 8.1 muestra el barco experimental en uno de los estanques del parque.



Figura 8.1: Barco durante uno de los experimentos.

Los datos experimentales se han tomado de distintos tipos de maniobras. En general, las maniobras incluyen grandes tramos rectos y giros. Estas maniobras son las más comunes en barcos a escala real. Los experimentos incluidos en la memoria son una trayectoria de *pasillos* y un *ocho*. El barco experimental tiene unos sistemas de guiado y control básicos. No se han utilizado los sistemas descritos en este Trabajo porque el objetivo es validar, de forma aislada, el sistema de Navegación propuesto.

El guiado del barco se hace utilizando puntos de paso o *waypoints* y guiado por línea de visión. Sea el vector de posición y orientación del barco $\boldsymbol{\eta} \triangleq [n \ e \ \psi]^T$ y la posición del punto de paso $\mathbf{w} \triangleq [n_w \ e_w]^T$, entonces se puede obtener el rumbo deseado

$$\psi_d = \text{atan2}(\tilde{e}, \tilde{n}) \quad (8.1)$$

donde $\tilde{e} \triangleq e_w - e$ es el error en la dirección Este y $\tilde{n} \triangleq n_w - n$ es el error en la dirección Norte.

La consigna de velocidad de avance u_d se fija de forma manual y se mantiene constante a lo largo del experimento.

El control del barco utiliza control PID para obtener los valores del ángulo del timón y la señal del motor eléctrico. Para controlar el rumbo se tiene que

$$d(t) = k_{r,p}\tilde{\psi}(t) - k_{r,d}\dot{\psi}(t) \quad (8.2)$$

donde $d(t)$ es la señal del servo, $\tilde{\psi}(t) \triangleq \psi_d(t) - \psi(t)$ y $k_{r,*} > 0$ son las constantes del control de rumbo. Para la velocidad se usa

$$n(t) = k_{v,p}\tilde{u}(t) + k_{v,i} \int_0^t \tilde{u}(\tau) d\tau \quad (8.3)$$

donde $n(t)$ es la señal del motor, $\tilde{u}(t) \triangleq u_d(t) - u(t)$ y $k_{v,*} > 0$ son las constantes del control de velocidad. Ambas leyes de control son muy simples pero son suficientes para los experimentos que se van a realizar; el ajuste de las distintas constantes fue realizado de forma experimental y está descrito en (Pereda, 2010).

Durante los experimentos, el sistema de Navegación está completamente aislado, es decir, no está alimentando a los sistemas de guiado y control descritos en esta sección. Adicionalmente, se hizo que, aleatoriamente, el sistema de Navegación no recibiera datos del GPS. Al no recibir estos datos, el algoritmo no puede corregir y solo puede confiar en el modelo de propagación hasta que llegue la siguiente medida. Esto se ha hecho así para ver cómo de fiable es el modelo y el estimador en ausencia de medidas GPS.

8.2. Consideraciones

Antes de entrar en el detalle de los resultados, es importante tener en cuenta algunos de los efectos que van a observarse, y qué los propicia. La siguiente lista contiene las consideraciones más importantes que deben hacerse a la hora de analizar los resultados de este capítulo.

Retardo del GPS Los GPS, además de actualizar la posición con una frecuencia baja, la señal que dan tiene retardo. Generalmente, un GPS que ofrezca una medida por segundo dará esas medidas con un segundo de retardo. A la hora de utilizar un algoritmo como el UKF, este retardo puede ser demoledor. El filtro de Kalman y sus variantes requieren que la medida con la que se ejecuta el paso de corrección corresponda con el estado en ese mismo instante. Sin embargo, esto no es así en el caso de un sensor con retardo. En (van der Merwe and Wan, 2004) se ofrecen varias alternativas para lidiar con este problema. En estos experimentos se ha asumido que las medidas no tienen retardo; esto no introduce mucha distorsión en el filtro, debido a que la dinámica del barco es bastante lenta con respecto al retardo.

Precisión del sensor GPS En el algoritmo estimador propuesto, se asume que la covarianza del error de las medidas (la matriz \mathbf{Q} del UKF) es constante. Sin embargo esta no es una restricción del UKF y mucho menos es real. Los sensores GPS ofrecen información sobre la precisión de la solución¹ y esta información se puede incorporar al UKF por medio de la matriz \mathbf{Q} .

¹Técnicamente, esta información, se denomina 'disolución del punto'.

Efecto de la escora en la brújula Cuando se utiliza una brújula de tres ejes, es posible medir la orientación utilizando la función `atan2`; sin embargo, esto tiene un problema. Aunque el modelo matemático puede ignorar los movimientos de escora y cabeceo para simular maniobras, el modelo real si tiene esos movimientos. En el caso de la brújula, el ángulo de escora falsea la medida de la orientación ya que parte de la información necesaria está ahora en el campo medido en el eje Z_{BARCO} , y no solo en X_{BARCO} e Y_{BARCO} . En caso de tener una estima del ángulo de escora, se puede corregir el efecto fácilmente. En el Trabajo, se ha ignorado este efecto; esto no es un gran problema pero introduce un poco de ruido adicional en la brújula.

Tubos de Pitot La velocidad que se mide utilizando los tubos de Pitot es local, con respecto al agua alrededor del barco. Es decir, cuando se interpreta la medida del sensor como una velocidad en NED, se está incluyendo, también la velocidad del agua en NED. En el caso de que el barco vaya a favor de corriente, la velocidad medida será menor que la real; mientras que cuando va a contra corriente, la velocidad medida será mayor que la real. En el Trabajo se ha decidido asumir que la velocidad del agua es baja y despreciable. Adicionalmente, la medida de la velocidad de avance está ligada al ángulo de cabeceo; el barco adquiere un ligero ángulo de cabeceo cuando acelera y por efecto de la corriente. Este ángulo hace que el tubo se desalinee de la dirección de avance y su medida sea menor.

Velocidad de deriva El barco experimental no tiene ninguna forma de medir la velocidad lateral. Durante su desarrollo se probó a poner tubos de Pitot laterales, pero no resultó. El barco tiene velocidad en el eje Y_{BARCO} , es decir, v no es necesariamente 0. Sin embargo, a la hora de propagar el modelo en el UKF, se asume que $\hat{v} = 0$. El efecto que esto puede generar es que el GPS haga correcciones notables en la dirección de deriva del barco ya que el modelo no es preciso en esa dirección.

Es importante hacer notar que los modelos de propagación y corrección utilizados en el Trabajo son los más simples posibles. La formulación del filtro estimador es la más directa. Existen formulaciones que permitirían mejorar el rendimiento en caso de que fuera necesario; una de estas formulaciones es (van der Merwe and Wan, 2004). También es posible hacer que el filtro estime la velocidad de deriva utilizando las medidas del GPS. Introduciendo unos valores de sesgo en las medidas de velocidad, es posible que el filtro estime la velocidad del agua utilizando las medidas de velocidad GPS. Todas estas mejoras se pueden implementar si de la formulación simple se obtiene un rendimiento no satisfactorio.

Los efectos creados por los ángulos de escora y cabeceo no deberían ser muy significativos en este barco experimental; sin embargo, pueden serlo en barcos de mayor tamaño. Para solucionar estos problemas lo mejor es introducir una unidad de medición inercial, *Inertial Measurement Unit* (IMU), y un sistema de determinación de actitud como el propuesto en (de Marina et al., 2011). Utilizando las medidas de estos sensores se pueden estimar los ángulos de escora y cabeceo y corregir ambos efectos.

8.3. Resultados

Uno de los experimentos era hacer que el barco hiciera una trayectoria de *ocho*. Se hizo que el barco repitiera esta maniobra varias veces para comprobar la repetibilidad del algoritmo. Se espera que si el barco pasa dos veces por el mismo punto, el estimador también haga lo mismo.

La Figura 8.2 muestra la trayectoria estimada por el sistema de Navegación frente a las medidas del sensor GPS. Se observa cómo las estimas tienen un poco de sesgo pero siempre

están dentro de la precisión asignada a los valores del GPS. Se observa cómo tanto los valores de las medidas como los del estimador son muy repetibles; y las distintas vueltas al *ocho* son muy parecidas las unas a las otras.

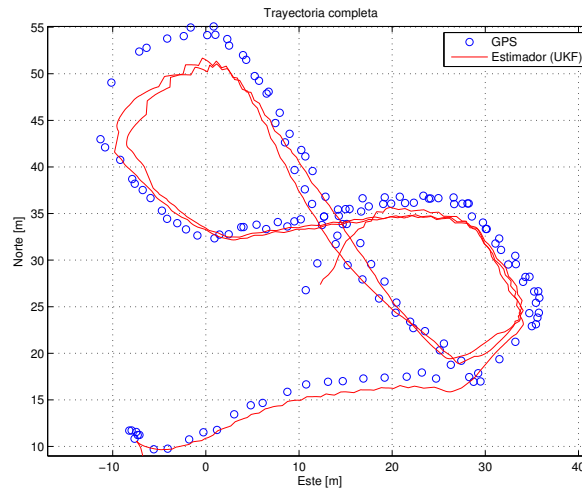


Figura 8.2: Resultados del estimador durante los *ochos*.

En la misma figura también se aprecia el efecto de las correcciones por no tener información de v en el modelo de propagación. Como era de esperar, estas correcciones ocurren cuando $v \neq 0$, es decir, en los giros; en los tramos rectos apenas se aprecian correcciones ya que, en esos casos, $v \approx 0$.

El siguiente experimento consiste en una trayectoria de *pasillos*: tramos rectos y cambios de sentido. La Figura 8.3a muestra las velocidades medidas con el GPS y con el sensor de presión. Se observa cómo hasta el segundo 600 ambos valores son muy parecidos mientras cerca de los 730, existe un error constante entre ambos; probablemente debido a la velocidad del agua. La Figura 8.3b muestra, en detalle, una parte del experimento. Se observa cómo la calibración del sensor es buena y las medidas, durante ese tiempo, se parecían mucho a las medidas por el GPS. La Figura 8.3a muestra la orientación del barco a lo largo del experimento. Pueden verse los distintos cambios de rumbo correspondientes a los giros, y zonas de rumbo más o menos constante, correspondientes a los tramos rectos.

La Figura 8.5a muestra el experimento completo. El trazo continuo rojo representa las estimaciones del UKF mientras los círculos azules muestran las medidas del GPS. Se vuelve a observar que el estimador ofrece buenas estimas en el movimiento longitudinal, pero sufre correcciones en la dirección de deriva por los efectos comentados anteriormente.

La Figura 8.5b muestra el primer giro del experimento. En este giro, se han eliminado varias medidas GPS para simular una pérdida de señal del mismo. En este caso, un sistema que no cuente con un estimador está, literalmente, perdido; no sabe dónde está hasta que se recupere la señal GPS y se reciba una medida. Sin embargo, el estimador es capaz de predecir la posición del barco muy bien en ambos casos; esto se puede deducir del hecho de que las correcciones realizadas por el algoritmo cuando llega la siguiente medida son muy pequeñas. En concreto, es curioso el caso del giro (parte derecha de la Figura 8.5b). En caso de perder el GPS, se perdería el punto de paso y la trayectoria sería mucho peor; sin embargo, el UKF es capaz de estimar bien el giro y, por tanto, no habría perdido el punto de paso.

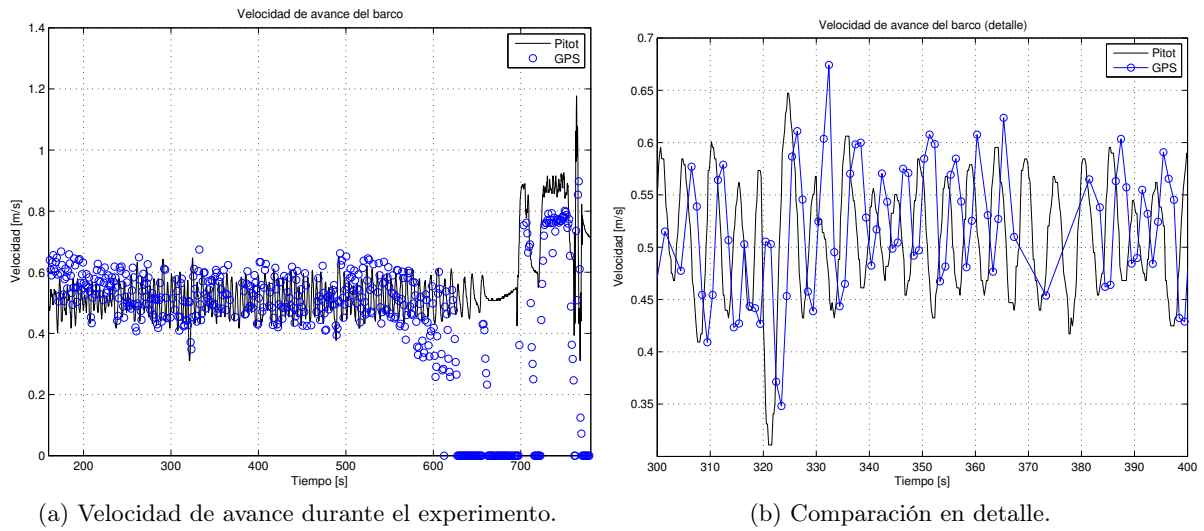


Figura 8.3: Comparación de la velocidad medida por el sensor de presión y el GPS.

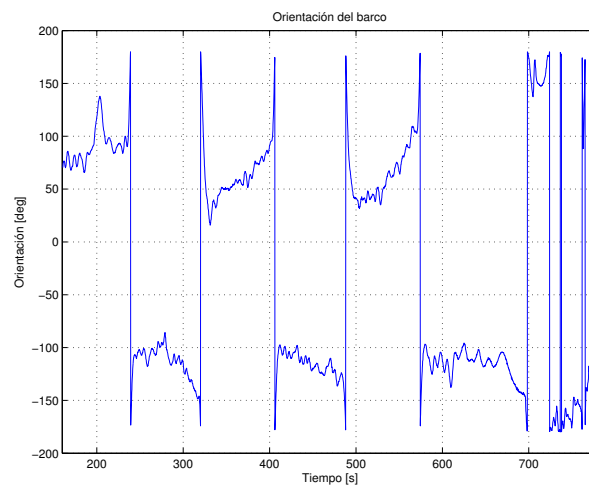


Figura 8.4: Orientación medida por la brújula.

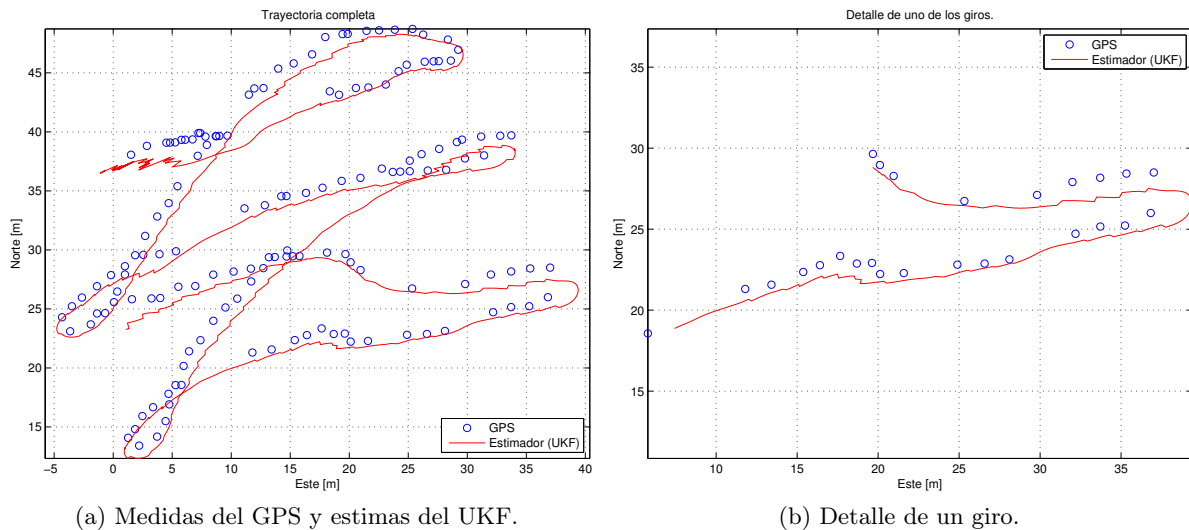


Figura 8.5: Resultados del estimador utilizando UKF.

8.4. Conclusiones

Los experimentos dan confianza en el sistema de Navegación tal cual está. En caso de que se detectaran problemas es posible utilizar alguna de las técnicas propuestas en la Sección 8.2 (Consideraciones) de este mismo capítulo. Aún así, la formulación directa proporciona muy buenas estimas incluso cuando se simula que el GPS ha perdido señal y no puede entregar medidas. También se ha comprobado que ignorar el retardo de las medidas GPS no produce ningún problema en el estimador. Una vez más se comprueba lo que, intuitivamente, es evidente: sería muy interesante tener medidas de la velocidad de deriva del barco. En caso de tener estas medidas, las correcciones serían mucho menos agresivas y las estimas del paso de propagación serían mejores.

Habiendo probado el sistema de Navegación, es posible seguir con los experimentos del sistema de Control. Una vez ambos sistemas hayan sido probados se pueden iniciar los experimentos de la propuesta completa. Estos experimentos, sin embargo, no son parte de este Trabajo.

Capítulo 9

Conclusiones, contribuciones y trabajos futuros

Los capítulos anteriores han sido dedicados al estudio del problema; a la propuesta de una solución; a la descripción en detalle de la solución así como de los algoritmos y metodologías empleados; y al estudio de resultados tanto en simulación como experimentales. Este capítulo enlaza la primera parte, los objetivos, con los dos últimos capítulos, los resultados. Recoge las conclusiones y contribuciones que nacen como producto del desarrollo de la propuesta y el sistema final. Adicionalmente, se hacen unos comentarios sobre trabajos y líneas futuras que pueden seguirse para refinar, mejorar y extender el sistema.

El objetivo principal de este Trabajo Fin de Máster es la creación de un sistema autónomo de Guiado, Navegación y Control de una flota de barcos.

El sistema propuesto cumple con el objetivo principal de este Trabajo. El sistema se ha dividido en tres grandes componentes:

Guiado que está encargado de generar las órdenes de alto nivel (rumbo y velocidad) de cada uno de los barcos de la flota. Este sistema utiliza splines para describir la trayectoria del grupo, es un sistema de subastas para mejorar la inicialización y cambios de formación; y un control jerárquico para guiar el grupo a la vez que se mantiene la formación evitando colisiones entre los barcos. El problema de la asignación y el algoritmo de las subastas se han descrito en el Capítulo 4 (El problema de la asignación y el algoritmo de subastas); mientras que los detalles concretos del sistema de Guiado se encuentran en la Sección 6.2 (Guiado).

Navegación que se encarga de localizar y estimar a cada barco en el espacio. Está compuesto de un sistema de filtrado básico de las señales y de estimación de la posición utilizando *Unscented Kalman Filter* (UKF). El algoritmo de estimación UKF se ha descrito en el Capítulo 5 (El filtro de Kalman *Unscented*) mientras que la Sección 6.1 (Navegación) discute los detalles del sistema de Navegación.

Control que tiene como responsabilidad principal ejecutar las órdenes de bajo nivel comandadas por el sistema de Guiado. Las leyes de control se han derivado utilizando la teoría de Lyapunov. Este sistema se ha escrito en la Sección 6.3 (Control).

Para el estudio de la viabilidad de la propuesta se ha realizado un estudio en simulación y el estudio experimental del sistema de Navegación. El estudio en simulación ha consistido en:

- Desarrollo y descripción de modelos matemáticos del barco y sus actuadores. Estos modelos se pueden encontrar en el Capítulo 3 (Modelado matemático).
- Implementación de los algoritmos y métodos descritos en el Capítulo 6 (Implementación del sistema).
- Diseño e implementación del simulador como componente software, descrito en la Sección 7.1 (El simulador).
- La tarea de obtención y análisis de resultados en una simulación *ideal* ha sido descrita en la Sección 7.2 (Estudio de escenarios simulados). Para evaluar mejor el sistema, se han planteado varios escenarios utilizando simulaciones realistas, estos resultados se han expuesto y presentado en la Sección 7.3 (Simulaciones realistas). Las conclusiones sobre la simulación se han comentado en la Sección 7.4 (Conclusiones).

Los resultados del estudio en simulación dan suficiente confianza en el sistema como para llevarlo al campo experimental. Este estudio experimental consta de los siguientes apartados:

- La plataforma experimental ha sido descrita en la Sección 3.2.3 (Barco experimental).
- Se han hecho algunas consideraciones importantes y que deben tenerse en cuenta en los experimentos en la Sección 8.2 (Consideraciones).
- Los resultados y el análisis de los mismos se presentan en la Sección 8.3 (Resultados). Los resultados son bastante buenos incluso en el caso de pérdida de señal del *Global Positioning System* (GPS).

- Las conclusiones sobre el estudio experimental de la Navegación se incluyen en la Sección 8.4 (Conclusiones). Dado que los resultados son razonablemente buenos, se comenta la necesidad de probar el siguiente sistema de la cadena de forma experimental, el Control.

Los resultados obtenidos demuestran la viabilidad de la propuesta. El siguiente paso es llevar a cabo experimentos del sistema de Control y del sistema de Guiado, en ese mismo orden.

Tabla 9.1: Requisitos funcionales del sistema. Copia de la Tabla 2.1.

Identificador	Descripción
GNC R.1	El sistema debe guiar el grupo a lo largo de una trayectoria geométrica. Es decir, no hace falta considerar restricciones temporales, solo espaciales.
GNC R.2	Debe ser suficientemente robusto para funcionar bajo los errores comunes de los sensores utilizados en barcos autónomos.
GNC R.3	El sistema debe permitir describir y mantener la forma geométrica del grupo. Es decir, la forma de la formación es un parámetro del sistema.
GNC R.4	El sistema debe evitar choques entre los barcos a toda costa.
GNC R.5	El sistema debe hacer que las trayectorias de los barcos sean lo más limpias posibles a lo largo de la misión. Esto es especialmente importante en la inicialización de la formación y entre los distintos cambios de formación.
GNC R.6	El sistema debe ser independiente del número de barcos de la flota. Este número es un parámetro al sistema y solo debe asumirse que será mayor que 0.
GNC R.7	El sistema debe ser completamente autónomo. Una vez decidida la trayectoria, el número de barcos y las formaciones, no debe requerir más intervención.

Se han cumplido, además, los siguientes objetivos y restricciones de diseño. A lo largo del Trabajo, siempre se ha valorado de forma muy importante el bajo acoplamiento externo de los distintos componentes. Los requisitos mencionados a continuación se encuentran en la Tabla 2.1. En aras de facilitar la lectura, se ha incluido una copia en este mismo capítulo como la Tabla 9.1.

Independencia de la trayectoria del grupo El sistema es capaz de guiar la flota a lo largo de cualquier trayectoria paramétrica. Efectivamente, el sistema encargado de generar las distintas consignas de posición del grupo, podría generarlas *al vuelo*. Por supuesto, el sistema solo podrá guiar al grupo a lo largo de trayectorias factibles para la dinámica del mismo.

Independencia del tamaño de la flota El sistema es capaz de guiar una flota de cualquier tamaño. Esto corresponde con el requisito *GNC R.6*.

Independencia de las formaciones El sistema es independiente de la forma geométrica de las formaciones de la flota. Es posible especificar formaciones completamente centradas o descentradas y el sistema llevará la flota a esa configuración; siempre dentro de los límites dinámicos de la misma. Este objetivo corresponde con el requisito *GNC R.3*.

No requiere intervención durante su funcionamiento Tal y como describe el requisito *GNC R.7*, el sistema es completamente autónomo. No requiere de intervención humana. Una vez se dan todos los parámetros, el sistema no requiere nada más para funcionar.

Inicialización de la formación sin casuística especial Al utilizar la metáfora económica a lo largo de toda la misión, el caso de la inicialización de una formación a partir de una

configuración aleatoria de los barcos, se resuelve sin ningún tipo de casuística especial. El mecanismo por el que se consigue esto es la revisión de los contratos por parte del subastador. Este concepto se describe en la Sección 6.2.3 (Subastas para formaciones) y se analizan los resultados en la Sección 7.2.2 (Efecto de las *resubastas*). Se comprueba que el algoritmo es capaz de disminuir el tiempo de convergencia de la flota a la formación. Esta parte del sistema también está ligada al requisito *GNC R.5*.

La lista anterior ha descrito los objetivos secundarios que se han conseguido y los ha relacionado con los requisitos del sistema de Guiado, Navegación y Control. Los requisitos que no estaban relacionados con objetivos secundarios se incluyen en la siguiente lista que especifica cómo se han cumplido y, si es preciso, el método de validación empleado.

GNC R.1: El sistema de Guiado permite la descripción de la trayectoria geométrica del grupo utilizando cualquier curva paramétrica. Las curvas utilizadas en este Trabajo son splines uniendo curvas de Bézier cúbicas. Este mecanismo se describe en la Sección 6.2.1 (Splines para trayectorias).

GNC R.2: Este requisito se ha cumplido y validado mediante simulaciones realistas en la Sección 7.3 (Simulaciones realistas); y, en parte, mediante experimentos del sistema de Navegación en el Capítulo 8 (Experimentos de Navegación).

GNC R.4: Utilizando el control *Null-Space-Based Behavioral Control* (NSB) la tarea de mayor prioridad siempre se cumple. En el caso de este Trabajo, esta tarea es la de evitar choques entre los barcos. Esto se describe en la Sección 6.2.2 (Control jerárquico *Null-Space-Based* (NSB)).

Se plantean las siguientes ideas como posibles líneas futuras de investigación:

- Desde el punto de vista del control, se plantea la posibilidad de extender el control diseñado en la Sección 6.3 (Control) sin separar el modo de avance del modo de deriva y giro.
- En el área del modelado, la identificación y la simulación, queda abierta la posibilidad de caracterizar el modelo con los términos cruzados de las matrices \mathbf{M} y $\mathbf{D}(\mathbf{v})$.
- Desde el punto de vista de la navegación, es posible desarrollar los modelos de propagación y corrección del algoritmo estimador presentado en la Sección 6.1.2 (Estimador). Utilizar modelos más elaborados, con técnicas como las presentadas en (de Marina et al., 2011), mejoraría las estimaciones y permitiría calibrar los sensores de forma dinámica durante la misión.
- Desde el punto de vista de la cooperación, es posible plantear escenarios en los que los barcos deban navegar en formación. Una posibilidad son los escenarios de recogida y confinamiento de vertidos como en (Jimenez et al., 2005). Otro escenario es la creación de mapas batimétricos utilizando una columna de barcos autónomos como se propone en (Breivik et al., 2008b); posiblemente utilizando técnicas de *Simultaneous Localization And Mapping* (SLAM) para la localización y guiado de la formación.
- Desde el punto de vista de las formaciones, se ha abierto la posibilidad de utilizar distintas métricas. En concreto, sería deseable una métrica asimétrica que tuviera en cuenta la orientación de los barcos. Algunas ideas para desarrollar la métrica se han expuesto en la Sección 6.2.3 (Subastas para formaciones).

- Desde el punto de vista de la Ingeniería, el paso evidente es continuar con los experimentos y extender el sistema a otro tipo de vehículos como aviones no tripulados.

El sistema propuesto cumple los objetivos del Trabajo: es capaz de guiar al grupo a lo largo de una trayectoria mientras la flota mantiene una formación rígida. El método se ha evaluado mediante simulaciones tradicionales y simulaciones realistas; además, se han llevado a cabo experimentos del sistema de Navegación, con muy buenos resultados. Todos los resultados han sido muy satisfactorios y el siguiente paso es la experimentación con el sistema completo.

Apéndice A

Acrónimos y definiciones

ECEF *Earth-Centered Earth-Fixed*

ECI *Earth-Centered Inertial*

EKF *Extended Kalman Filter*

GNC *Guiado Navegación y Control*

GPS *Global Positioning System*

IMU *Inertial Measurement Unit*

NED *North-East-Down*

NSB *Null-Space-Based Behavioral Control*

PID *Proporcional Integral Derivativo*

PWM *Pulse-width Modulation*

SD *Secure Digital*

SLAM *Simultaneous Localization And Mapping*

UKF *Unscented Kalman Filter*

UT *Unscented Transform*

Bibliografía

- Antonelli, G., Arrichiello, F., and Chiaverini, S. (2005). The null-space-based behavioral control for soccer-playing mobile robots. In *Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.*, pages 1257–1262.
- Antonelli, G., Arrichiello, F., and Chiaverini, S. (2010). The nsb control: a behavior-based approach for multi-robot systems. *Paladyn. Journal of Behavioral Robotics*, 1:48–56.
- Arrichiello, F., Chiaverini, S., and Fossen, T. I. (2006). Formation control of marine surface vessels using the null-space-based behavioral control. In Pettersen, K., Gravdahl, J., and Nijmeijer, H., editors, *Group Coordination and Cooperative Control*, volume 336 of *Lecture Notes in Control and Information Sciences*, pages 1–19. Springer Berlin / Heidelberg.
- Behal, A., Dawson, D. M., Dixon, W. E., and Fang, Y. (2002). Tracking and regulation control of an underactuated surface vessel with nonintegrable dynamics. *IEEE Transactions on Automatic Control*, 47(3):495–500.
- Bertsekas, D. P. (1979). A distributed algorithm for the assignment problem. Technical report, Massachusetts Institute of Technology, Laboratory for Information and Decision Systems.
- Bertsekas, D. P. (1989). The auction algorithm for assignment and other network flow problems. Technical report, Massachusetts Institute of Technology, Laboratory for Information and Decision Systems.
- Breivik, M. (2003). Nonlinear maneuvering control of underactuated ships.
- Breivik, M. (2010). *Topics in Guided Motion Control of Marine Vehicles*. PhD thesis, Norwegian University of Science and Technology. Faculty of Information Technology. Mathematics and Electrical Engineering Department of Engineering Cybernetics.
- Breivik, M. and Fossen, T. I. (2005). Principles of guidance-based path following in 2d and 3d. In *44th IEEE Conference on Decision and Control, and European Control Conference.*, pages 627–634.
- Breivik, M., Hovstein, V. E., and Fossen, T. I. (2008a). Ship formation control: A guided leader-follower approach. In *IFAC World Congress*, volume 17.
- Breivik, M., Hovstein, V. E., and Fossen, T. I. (2008b). Straight-line target tracking for unmanned surface vehicles. *Modeling, Identification and Control*, 29(4):131–149.
- Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23.
- Caccia, M., Bibuli, M., Bono, R., and Bruzzone, G. (2008). Basic navigation, guidance and control of an unmanned surface vehicle. *Autonomous Robots*, 25:349–365. 10.1007/s10514-008-9100-0.

- Davis, R. and Smith, R. G. (1981). Negotiation as a metaphor for distributed problem solving. Technical report, Massachusetts Institute of Technology, Computer Science and Artificial Intelligence Lab.
- de Marina, H. G., Pereda, F. J., Giron-Sierra, J. M., and Espinosa, F. (2011). Uav attitude estimation using ukf and triad. *IEEE Transactions on Industrial Electronics*, page Pending Publication.
- Dong, W. (2010). Cooperative control of underactuated surface vessels. *IET, Control Theory Applications*, 4(9):1569–1580.
- Dubins, L. E. (1957). 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.
- Fossen, T. I. (2011). *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons Ltd.
- Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- Gerkey, B. P. and Mataric, M. J. (2002). Sold!: auction methods for multirobot coordination. *IEEE Transactions on Robotics and Automation*, 18(5):758–768.
- Hirsh, M. J., Pardalos, P. M., Murphey, R., and Grundel, D. (2007). *Cooperative Control and Optimization*. Springer Verlag.
- Hsu, H. C.-H. and Liu, A. (2005). Applying a taxonomy of formation control in developing a robotic system. *IEEE International Conference on Tools with Artificial Intelligence*, 0:3–10.
- Jimenez, J., Giron-Sierra, J. M., Dominguez, A., de la Cruz, J. M., and Riola, J. M. (2005). Ships confinings an oil spill over: A scenario for automatized cooperation. In *IEEE Conference Oceans*, pages 1226–1231.
- Julier, S. J. and Uhlmann, J. K. (1997). A new extension of the kalman filter to nonlinear systems. In *Proc. Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, Orlando, FL*, pages 182–193.
- Julier, S. J. and Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. In *Proceedings of the IEEE*, volume 92, pages 401–422.
- Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.
- LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press, Cambridge, U.K.
- Li, Y., Ang, K., and Chong, G. (2006). PID control system analysis and design. *IEEE Control Systems Magazine*, 26(1):32–41.
- Maza, I., Caballero, F., Capitan, J., de Dios, J. M., and Ollero, A. (2010). Firemen monitoring with multiple uavs for search and rescue missions. In *IEEE Workshop on Security, Safety and Rescue Robotics (SSRR)*.
- Pereda, F. J. (2010). Sistema de telemetría y control de un barco autónomo.

- Pereda, F. J., de Marina, H. G., Jimenez, J. F., and Giron-Sierra, J. M. (2010). A development project of autonomous marine surface vehicles for sea demining. In *Proceedings of the 11th International Conference on Control Automation Robotics Vision (ICARCV)*, pages 573–578.
- Russell, S. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Second edition.
- Shima, T. and Rasmussen, S. (2009). *UAV Cooperative Decision and Control*. SIAM.
- Skejjic, R., Breivik, M., Fossen, T. I., and Faltinsen, O. M. (2009). Modeling and control of underway replenishment operations in calm water. In *Proceedings of the 8th IFAC International Conference on MCMC*, pages 78–85.
- van der Merwe, R. and Wan, E. (2004). Sigma-point kalman filters for integrated navigation. In *Proceedings of the 60th Annual Meeting of The Institute of Navigation*, pages 641–654.
- Vig, L. (2008). *Multiple Robot Teaming*. VDM Verlag.
- Viguria, A., Maza, I., and Ollero, A. (2007). Set: An algorithm for distributed multirobot task allocation with dynamic negotiation based on task subsets. In *IEEE International Conference on Robotics and Automation*, pages 3339–3344.
- Viguria, A., Maza, I., and Ollero, A. (2008). S+t: An algorithm for distributed multirobot task allocation based on services for improving robot cooperation. In *IEEE International Conference on Robotics and Automation*, pages 3163–3168.
- Wan, E. and van der Merwe, R. (2001). The unscented kalman filter. In Haykin, S., editor, *Kalman Filtering and Neural Networks*, chapter 7. Wiley.

