

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
MAESTRÍA EN INGENIERÍA DE
CONTROL Y AUTOMATIZACIÓN



GENERACIÓN Y CONTROL DE TRAYECTORIA
PARA EMBARCACIONES MARINAS EN AGUAS
POCO PROFUNDAS Y ESPACIOS CONFINADOS

Tesis en opción al grado de Magister en Ingeniería de Control
y Automatización

Autor: Cayetano Juan Cáceres Mendoza

Asesor de Tesis: PhD Antonio Morán Cárdenas

2016

*Dedicado a mi esposa Evelyn y a mis Padres,
por su paciencia y apoyo incansable para lograr
este sueño, a mi asesor el Dr. Antonio Moran
por las enseñanzas y experiencias dadas
durante la maestría.*

Agradecimientos

Me gustaría agradecer a todos los miembros de mi familia, en especial a mi esposa Evelyn y a mi hija Alessia, puesto que la elaboración y redacción de esta tesis ha significado sacrificar tiempo valioso con ellas, agradezco por sus sonrisas y su buen ánimo siempre conmigo, siempre tengo claro que ustedes son las que motivan mi vida.

También quiero agradecer a mis padres Auria y Vidal, por su apoyo incondicional, sus consejos y sueños que también son los míos, ha sido muy grato compartir este tiempo con ustedes, espero que siempre tengan ese ánimo y esa fuerza progresista que los caracteriza. A mi hermano Jorge, decir o escribir mil veces gracias sería insuficiente para agradecer tu apoyo durante este proyecto.

Esta tesis no hubiera sido posible sin el apoyo de la Sra. Angelica y Wilson, muchas gracias por su ayuda, siempre estaré infinitamente agradecido con ustedes por su ayuda. A la Ing. Liliana Asalde por confiar en mi gracias por su apoyo desde el inicio de este sueño, siempre tendré presente su amistad y apoyo. A mis queridos amigos Rubén y Nelson, espero algún día poder ayudarlos, como ustedes lo han hecho a través de sus ánimos.

A todos mis amigos de la maestría, que siempre fueron de mucha ayuda para enfrentar algunos cursos complicados, a mi buen amigo Neil que siempre alentó seguir en la elaboración de esta tesis. A todos los profesores que nos enseñaron y guiaron a través de los cursos en el desarrollo de las diversas teorías de control. A los miembros del CONCYTEC y de la PUCP que confiaron en mi persona e hicieron posible cumplir este sueño.

Resumen

El capítulo 1, presenta una introducción y estado del arte del planeamiento y seguimiento de trayectorias en espacios confinados. En el capítulo 2, se presenta el desarrollo completo del modelo matemático general de una embarcación marina, considerando los 6 grados de libertad para posteriormente particularizar este modelo a un modelo de 3 grados de libertad que es el que se va analizar en esta tesis.

El propósito del capítulo 3, consiste en realizar el diseño mediante simulación de un sistema de planeamiento, guía y seguimiento de trayectoria para una embarcación marina subactuada con dos hélices, a partir del modelo matemático de barco de 3 grados de libertad, obtenido en el capítulo 2. En este capítulo se divide en 3 partes, el planeamiento de trayectoria, estrategia de control y síntesis de planeamiento y seguimiento de trayectoria para embarcaciones marinas en espacios confinados. En la primera parte, se da a conocer la estrategia utilizada para realizar el planeamiento de trayectoria partiendo de un espacio determinado caracterizado por su geometría y obstáculos. Para esta tarea se hace uso del diagrama de Voronoi, el algoritmo de Dijkstra, y un algoritmo de tal manera que filtramos algunos puntos para evitar los cambios de dirección innecesarios, también consideramos una tolerancia de tal manera que la embarcación pueda circular por lugares estrechos con cierta holgura. En la segunda parte, se presenta la teoría de control “*backstepping*”, que es aplicada para sistemas dinámicos no lineales, se comenta sus ventajas respecto a la técnica de “*feedback linearization*”. Se muestra el desarrollo de la técnica aplicada al control de embarcaciones marinas, considerando la dinámica del barco. Se desarrolla el sistema de guía del barco para lograr el seguimiento de trayectoria, con las consideraciones especiales que implican controlar un sistema subactuado.

En el capítulo 4, se desarrolla la simulación para dos entornos diferentes, el primero corresponde a un canal en forma senoidal por el cual el barco se desplazará, se realizará el planeamiento y control de la ruta, el segundo entorno corresponde a una aplicación más real, pues se toma como mapa referencia el mar Báltico, al cual se le aplican las técnicas desarrolladas en esta tesis.

Índice general

1. Introducción y estado del arte de generación y seguimiento de trayectorias para embarcaciones marinas en espacios confinados	1
1.1. Introducción	1
1.2. Antecedentes	2
1.3. Objetivo general	4
1.4. Objetivos específicos	4
1.5. Estado del Arte	6
1.6. Solución propuesta	15
1.6.1. Breve descripción de la solución propuesta	15
1.6.2. Contribución de la tesis	15
2. Modelo Matemático	18
2.1. Introducción	18
2.2. Dinámica	19
2.2.1. Sistemas de referencia:	19
2.2.2. Dinámica del centro de gravedad	21
2.2.3. Dinámica del punto CO	24
2.2.4. Representación del modelo en forma matricial	26
2.2.5. Fuerzas actuantes sobre la embarcación marina	27
2.2.5.1. Fuerzas de empuje por parte de los actuadores	28
2.2.5.2. Helice con timon para direccionamiento	28
2.2.5.3. Propulsor azimutal	28
2.2.5.4. Otras configuraciones	28
2.2.6. Fuerzas hidrodinámicas	30
2.2.6.1. Desplazamiento de masa de agua	30
2.2.6.2. Amortiguación	30
2.2.7. Fuerzas hidrostáticas	31
2.2.8. Fuerzas del ambiente	31
2.2.9. Otras fuerzas	32
2.3. Modelo de referencia	32
2.3.1. Modelo de simulación	33
2.3.1.1. Parámetros Dinámicos	33
2.3.2. Radio de giro del barco	35
2.3.3. Pruebas con el modelo de referencia	35
2.4. Conclusiones preliminares	37

3. Desarrollo de un sistema de planeamiento y seguimiento de ruta de un barco de dos hélices en espacio confinado	39
3.1. Introducción	39
3.2. Planeamiento de trayectoria	40
3.2.1. Terminología	40
3.2.2. Planeamiento de movimiento	42
3.2.2.1. Diagrama de Voronoi	43
3.2.2.2. Algoritmo de Dijkstra	43
3.2.2.3. Obstáculos	45
3.2.2.4. Eliminación de puntos no necesarios	45
3.2.3. Generación de ruta	46
3.2.3.1. Generación de ruta usando polinomio cúbico de Hermite	48
3.2.3.2. Generación de ruta usando Espiral de Fermat	49
3.3. Sistema de guiado	50
3.3.1. Línea de visión - Basado en 3 puntos	52
3.3.1.1. Cálculo de ángulo LOS ψ_{los} para embarcaciones marinas	53
3.3.1.2. Cálculo de posición LOS p_{los} para embarcaciones marinas	53
3.3.2. Sistema de referencia móvil	58
3.4. Sistema de control	59
3.4.1. Control Backstepping	59
3.4.2. Diseño de ley control MIMO backstepping para barcos	61
3.5. Conclusiones Preliminares	64
4. Simulación	66
4.1. Simulación en canal senoidal	66
4.1.1. Planeamiento de movimiento y trayectoria	66
4.1.2. Determinación de puntos de ruta "WayPoints".	67
4.1.3. Determinación de ruta paramétrica.	68
4.1.3.1. Polinomio cúbico de hermite	68
4.1.3.2. Espiral de Fermat	68
4.1.4. Seguimiento de trayectoria y control de embarcación	71
4.1.4.1. Resultados de simulación	71
4.2. Simulación en el mar Báltico	74
4.2.1. Planeamiento de movimiento y trayectoria en el mar Báltico	74
4.2.2. Determinación de puntos de ruta "WayPoints" en el mar Báltico	76
4.2.3. Determinación de ruta paramétrica en el mar Báltico	76
4.2.3.1. Polinomio cubico de hermite en el mar Báltico	79
4.2.3.2. Espiral de Fermat en el mar Báltico	79
4.2.4. Seguimiento de trayectoria y control de embarcación en el mar Báltico	82
4.2.4.1. Resultados de simulación en el mar Báltico	82
4.3. Conclusiones Preliminares	84
Conclusiones	87

Recomendaciones	88
Bibliografía	89
Anexo A. Script en MATLAB para control de barco	93
Anexo B. Script en MATLAB para eliminar puntos no necesarios	99
Anexo C. Script en MATLAB para generar Rutas con el Polinomio Cúbico de Hermite	101
Anexo D. Script en MATLAB para generar Rutas con la espiral de Fermat	105
Anexo E. Propuesta de implementación de barco para monitoreo de variables marinas	110
E.1. Introducción	111
E.2. Propuesta de instrumentación	112
E.2.1. Selección de sensores/transmisores	112
E.2.1.1. Sensor de Inercia (IMU)	113
E.2.1.2. Sensor de viento	113
E.2.1.3. Sensor de humedad y temperatura de ambiente	113
E.2.1.4. Sensor de temperatura del agua	113
E.2.1.5. Sensor de presión	115
E.2.1.6. Sensor de salinidad (Conductividad)	115
E.2.1.7. Sensor de PH	115
E.2.1.8. Módulo XBee	117
E.2.2. Actuadores	117
E.2.2.1. Motores y hélices	117
E.2.3. Sistema de Control	117
E.2.3.1. Raspberry PI	118
E.2.3.2. Arduino R3 uno	118
E.2.3.3. Computador de monitoreo	119
E.3. Propuesta de implementación del controlador desarrollado	120
E.4. Propuesta de sistema de control y monitoreo	121
E.5. Conclusiones	121
Anexo F. Echo Boat RC	123
Anexo G. Sensor de Inercia IMU	126
Anexo H. Sensor de Viento	131
Anexo I. Sensor de Humedad y temperatura de ambiente	134
Anexo J. Sensor de temperatura del agua	137
Anexo K. Sensor de Presión	140
Anexo L. Sensor de Salinidad	142

Anexo M. Sensor de PH	145
Anexo N. Módulo de comunicación inalámbrica Xbee	148
Anexo Ñ. Motor de hélices	151

Abreviaturas

κ_{max}	Curvatura máxima que puede realizar el barco
ψ	ángulo en dirección del eje Z del sistema body
ψ_{los}	Ángulo LOS
CG	Centro de gravedad
DOF	Grados de libertad (<i>Degrees of freedom</i>)
DP	Posición Dinámica (<i>Dynamic Positioning</i>)
E	Posición Este, asociada al eje Y
$ECDIS$	Visualizador de cartas de navegación electrónicas, (<i>Electronic Chart Display</i>).
GPS	Sistema de Posicionamiento Global
IMU	Unidad de Medición Inercial
L_{pp}	Longitud de embarcación marina.
LOS	Line of sight - Línea de visión
N	Posición Norte, asociada al eje X
p_{los}	Posición LOS
PP	Sistema de referencia paralelo a la ruta.
r	velocidad rotacional en dirección del eje Z del sistema body
u	velocidad en dirección del eje X del sistema body
v	velocidad en dirección del eje Y del sistema body
WP	Puntos de ruta (<i>Waypoint</i>).

Capítulo 1

Introducción y estado del arte de generación y seguimiento de trayectorias para embarcaciones marinas en espacios confinados

1.1. Introducción

En la actualidad, la navegación de navíos tiene su principal interés en el tema de la seguridad, por lo cual se vienen desarrollando proyectos de ciencia y tecnológica enfocados en este tema, se desarrollan investigaciones de planeamiento de trayectorias seguras, ó autopilotaje de navíos basado en sistemas inteligentes, para de esta manera frente a alguna perturbación se puedan tomar las mejores decisiones a fin de evitar algún incidente o accidente como por ejemplo colisión, encallamiento, etc. Además se debe tener especial cuidado en la navegación que se da cerca a archipiélagos, puertos o canales debido a que se presentan menos espacios para realizar maniobras seguras y el margen de error es menor que en alta mar. El tema de seguridad es mucho más estricto puesto que errores de maniobra pueden generar colisiones entre navíos, contra muelles y con alguna roca o isla, o frente a algún fenómeno climatológico. En la “Maritime Transport Conference” en el 2012 se presenta nuevos enfoques de programación de movimiento en buques (Mironenko, 2012b), basados en experiencias y estas representadas de forma aproximada por modelos matemáticos.

En el pasado, los sistemas de navegación de embarcaciones eran basadas en controlar la dirección del barco hacia una dirección deseada por medio de un giroscopio, Skjetne (2003). Con los nuevos sensores que se encuentran disponibles en el mercado y con el conocimiento de técnicas de control no lineal avanzado, hoy es posible realizar maniobras complicadas mediante control automático, Fossen (2011). Esto irá incrementando la funcionalidad y confiabilidad de los sistemas

comerciales de control automático de embarcaciones.

Por estos días, las embarcaciones marinas son equipadas con modernos dispositivos como sensores, propulsores y motores de alta eficiencia, Kurowski et al. (2013). Con estos nuevos avances en los actuadores y tecnología de control la maniobrabilidad de las embarcaciones se ha incrementado substancialmente. Maniobras especiales, como atracar el barco en el muelle o una serie de maniobras, pueden ser realizadas sin la necesidad de usar barcos de remolque, vientos o anclas. Además, se debe considerar que la confiabilidad de la información de posición especialmente en puertos o otras áreas críticas, que son de nuestro interés, han sido mejorados por sistemas auxiliares, Korte et al. (2015). Al tener el usuario datos de posición confiables, puede realizar el control con alta grado de seguridad de forma automática, en especial en caso de decremento de visibilidad.

La deficiencia de estos días en los sistemas de seguimiento de trayectorias es el método de planeamiento de rutas y en especial para lugares confinados, como canales o cercanía a puertos donde la cantidad de “obstáculos” es mayor, Mironenko (2013). La generación de valores de entrada para el sistema de control por parte del oficial de navegación, especialmente en modos de baja velocidad, donde el sistema se torna más inestable sigue siendo un problema, Mironenko (2012b). Con el propósito de hacer frente a esto, nuevos algoritmos de maniobra han sido desarrollados y agregados a entornos de simulación a fin de analizar sus resultados y encontrar los mejores, en esta tesis se presentaran algunos de ellos y se analizarán los resultados.

1.2. Antecedentes

El control de ruta de embarcaciones empezó con la invención del giroscopio, que fue el primer sensor de realimentación del sistema, esto se remonta a 1908 cuando el Dr. Kaempfe desarrolló el primer giroscopio, Skjetne (2003). Esto desencadenó en una serie de mejoras para que finalmente Elmer Sperry use este y genere un piloto automático, giropiloto, para el direccionamiento automático de un barco, recién en 1922 estuvo disponible comercialmente. El giropiloto, hoy conocido como autopiloto convencional, que es un sistema de control de simple entrada simple salida (SISO), donde la dirección es medida por un giroscopio y el error en dirección es compensado por acción del timón del barco. Debido a su fácil configuración este tipo de control ha sido ampliamente usado y aun sigue siendo usado en algunos casos. Sin embargo, con los avances tecnológicos y la introducción de nuevos sistemas de medición más precisos, confiables, en especial el sistema de posicionamiento global (GPS) que nos permite ubicar un cuerpo

alrededor del mundo, y la necesidad de mejorar la eficiencia y confiabilidad de las maniobras a fin de evitar accidentes sobre todo en espacios de difícil tránsito, donde la generación y seguimiento deben ser precisas, Mironenko (2013), hacen que se abran nuevas posibilidades y direcciones de investigación. Como resultado de esto en 1960 se logró realizar control de Posición dinámica (DP), esto se logró con la implementación de tres controladores PID desacoplados, Fossen (2002).

El sistema DP, es un sistema múltiple-entrada, múltiple-salida con tres grados de libertad, avance, deriva y guiñada (*surge, sway and yaw*), Skjetne (2003), que son controlados por propulsores. El modelo base para controlar DP, es un modelo hidrodinámico avanzando no lineal, derivado de los principios fundamentales, que posteriormente es simplificado a un sistema lineal aprovechando que la velocidad es casi cero para estas aplicaciones. Tomando como referencia la teoría desarrollada para lograr el control DP, ahora la nueva dirección de investigación está en el seguimiento de trayectorias a alta velocidad y la generación de rutas, Skjetne (2005).

La generación de trayectorias en espacios confinados es considerada como crucial y complicada, (Bowditch, 1995), por lo cual este campo de la ciencia marina se está orientando a trasladar ciertas tareas, que antes realizaba la tripulación, a sistemas de control y/o inteligencia artificial. Existen sistemas actuales que generan trayectorias pero durante esta generación obvian ciertas leyes de la física, no consideran el ángulo de deslizamiento del navío, cuando hay alguna curva se reemplaza directamente por arcos de círculos, Mironenko (2013), esto degenera en un salto de un ángulo de guiñada cero hacia un valor constante que a la postre se refleja en un pico en la señal de control. Luego de realizar la generación de ruta se tendrá que aplicar técnicas de control a fin de poder lograr que el navío cumpla con la trayectoria deseada.

Debido a que las embarcaciones circulan por lugares muy cercanos a zonas peligrosas como, zonas superficiales, con alto tráfico, bajo influencia de fuerte oleaje, lugares estrechos, etc. todos estos problemas han terminado en colisiones o encallamientos, como se puede ver todos estos problemas se pueden mejorar directamente con la generación de una ruta óptima y un seguimiento de trayectorias preciso, Mironenko (2013). La variedad de factores que pueden influenciar en la generación de trayectorias son diversos por lo cual para hallar las óptimas se deberán realizar pruebas. Básicamente las trayectorias se definen a partir del conocimiento de los tripulantes apoyados en sus instrumentos de navegación con los cuales corrigen posibles desviaciones, pero estos no son de alta precisión.

La importancia de la solución del problema radica en la mejora de la seguridad de navegación, actualmente se busca que los sistemas inteligentes puedan reem-

plazar la operación manual consiguiendo con esto eficiencia y mejores tiempos de respuesta frente a perturbaciones.

Anteriormente las maniobras por parte de las embarcaciones eran realizadas totalmente por los tripulantes, la respuesta y resultados dependían en gran manera de la experiencia del capitán y de la tripulación en sí. En el pasado, las soluciones venían dadas por capacitar a la tripulación con simuladores de estos fenómenos, para medir el tiempo de respuesta y las decisiones que se tomaban bajo ciertas circunstancias. Las simulaciones básicamente eran para mostrar a los operadores como manejar el barco en situaciones de mucha proximidad a los problemas planteados.

La solución que se quiere plantear es realizar las trayectorias aprovechando los nuevos dispositivos de medición que son más precisos y confiables, a fin de poder describir una trayectoria deseada segura, continua y suave. Además, se espera que este método pueda ser implementado en sistemas de navegación moderno y pueda servir para desarrollar nuevos esquemas de navegación así como centros de simulación.

En un sistema convencional de seguimiento basado en puntos de ruta, solamente el ángulo de guiñada (dirección) es controlado, tomando el barco desde un punto a otro, usando el algoritmo de línea de visión (LOS). La manera más fácil de convertir este problema en un problema de seguimiento de ruta es conectando las líneas rectas entre cada punto mediante círculos inscritos, como se muestran en la Figura 1.1. Como han señalado varios autores, la ruta con distancia más corta entre 2 puntos es una línea recta y segmentos de arco. Sin embargo, como la ruta no es factible porque al cambiar de recta a círculo se da un salto en el ángulo de guiñada deseado. Para tener un mejor resultado la ruta que una dos puntos debe ser una curva que sea diferenciable al menos dos veces.

1.3. Objetivo general

El presente trabajo tiene como objetivo general el desarrollo de un generador y seguidor de trayectorias para embarcaciones marinas en espacios confinados, canales y lugares de alto tráfico de tal forma que se obtengan trayectorias continuas, suaves y que se puedan seguir con un pequeño margen de error que no ponga en peligro la seguridad de la embarcación ni la de su tripulación.

1.4. Objetivos específicos

Los objetivos específicos de este proyecto son los siguientes:

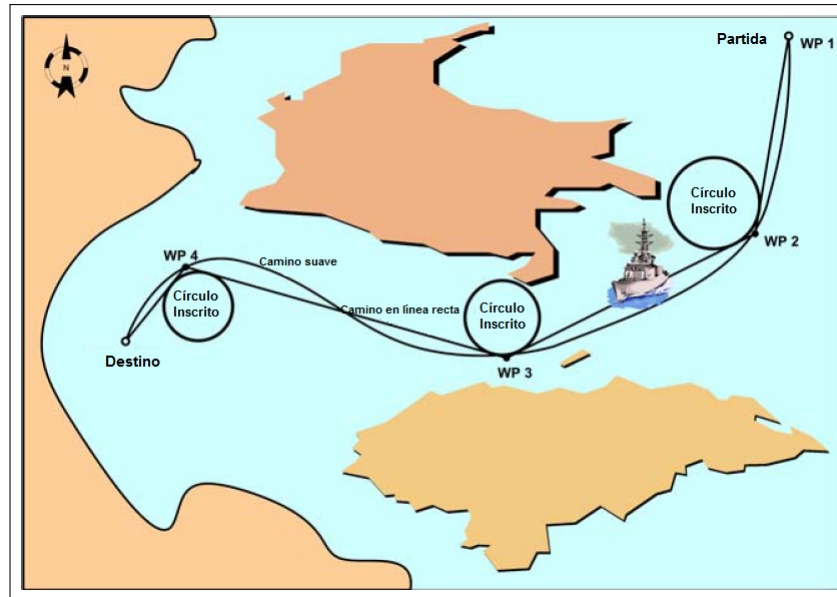


Figura 1.1: Mapa digital donde la ruta es construida a partir de puntos mediante líneas rectas y círculos, también se presenta una alternativa usando una curva suave, Skjetne (2003)

- Obtener un modelo matemático de una embarcación marina considerando las fuerzas dinámicas actuantes
- Realizar el planeamiento de movimiento a partir de un mapa digital considerando obstáculos o formas especiales
- Relacionar los parámetros cinemáticos a funciones que puedan dar continuidad y suavidad a las rutas generadas
- Relacionar los parámetros cinemáticos a funciones que puedan dar continuidad y suavidad a las rutas generadas.
- Generar rutas paramétricas al menos por dos métodos, comparar sus ventajas y desventajas frente a seguimiento de trayectoria basada en puntos de ruta.
- Generar rutas paramétricas deseadas a partir de coordenadas de posición, orientación y lograr su seguimiento a través de sistemas de control no lineal avanzado.
- Demostrar el mejoramiento en el control de trayectoria, comparando los índices respectivos para los casos: sistema de control considerando fuerzas dinámicas, sistema de control sin considerar fuerzas dinámicas.
- Realizar una propuesta de implementación económica para la aplicación de monitoreo de variables marinas.

Hipótesis, en este trabajo se parte de la hipótesis de que es posible generar y seguir trayectorias para embarcaciones marinas en espacios confinados.

1.5. Estado del Arte

En el presente, en el área de planeamiento de rutas para vehículos marinos, existen diferencias entre métodos de planeamiento de rutas navales que puedan ser usadas con los sistemas estándar de planeamiento de trayectoria respecto de los algoritmos de planeamiento de rutas para aplicaciones de mar abierto así como para vehículos de alta dinámica, como los vehículos autónomos superficiales y pequeñas embarcaciones, Kurowski et al. (2013). En la navegación comercial, la generación de la ruta de referencia está a cargo del oficial responsable. Por ello, idénticos métodos de planeamiento de ruta son usados para un amplio rango de vehículos desde los barcos largos y lentos hasta los más ágiles barcos multipropósito, con sus particulares requerimientos de maniobra, Breivik (2003). Es por esta razón que los métodos de planeamiento que son usados son muy conservadores, porque luego estos tendrán que ser adaptados con algunos cambios de parámetros al barco respectivo que puede ser grande o pequeño. Pero muy a pesar del cuidado con los algoritmos, es posible generar alguna que no sea ejecutable por cierto vehículo, esto porque cada vehículo tiene particularidades en su dinámica como su radio de giro, Korte et al. (2015). Esto deriva que el sistema de guiado para el seguimiento de ruta genere grandes valores en las leyes de control o genere riesgos en la embarcación. De ahí que muchos métodos en la actualidad son solo aplicables para realizar maniobras en áreas de mar abierto con suficiente espacio para realizar dichas maniobras.

Ahora revisaremos los métodos de planeamiento usados con los sistemas de guiado estándar que se manejan en la actualidad para el seguimiento de ruta. Esto se refiere al campo de los métodos de planeamiento por puntos de ruta, Fossen (2011). Por tanto, un número finito de puntos son indicados y conectados con líneas rectas. Los arcos circulares son integrados a la ruta automáticamente en cada punto, para definir así los círculos de giro para cada punto de ruta. Para estos casos el mínimo radio de giro es definido como radio de circunferencia en cada punto, esto con el fin de poder tener la maniobrabilidad suficiente y poder seguir la ruta sin problemas. Esta metodología puede ser exitosa debido a la multiplicidad de la mayoría de barcos que solamente cambian muy condicionalmente. Además, se debe tener en cuenta que la maniobrabilidad del vehículo marino cambia significativamente con la variación de la velocidad. Un parámetro entero sería necesario para mapear estos efectos, Kurowski et al. (2013).

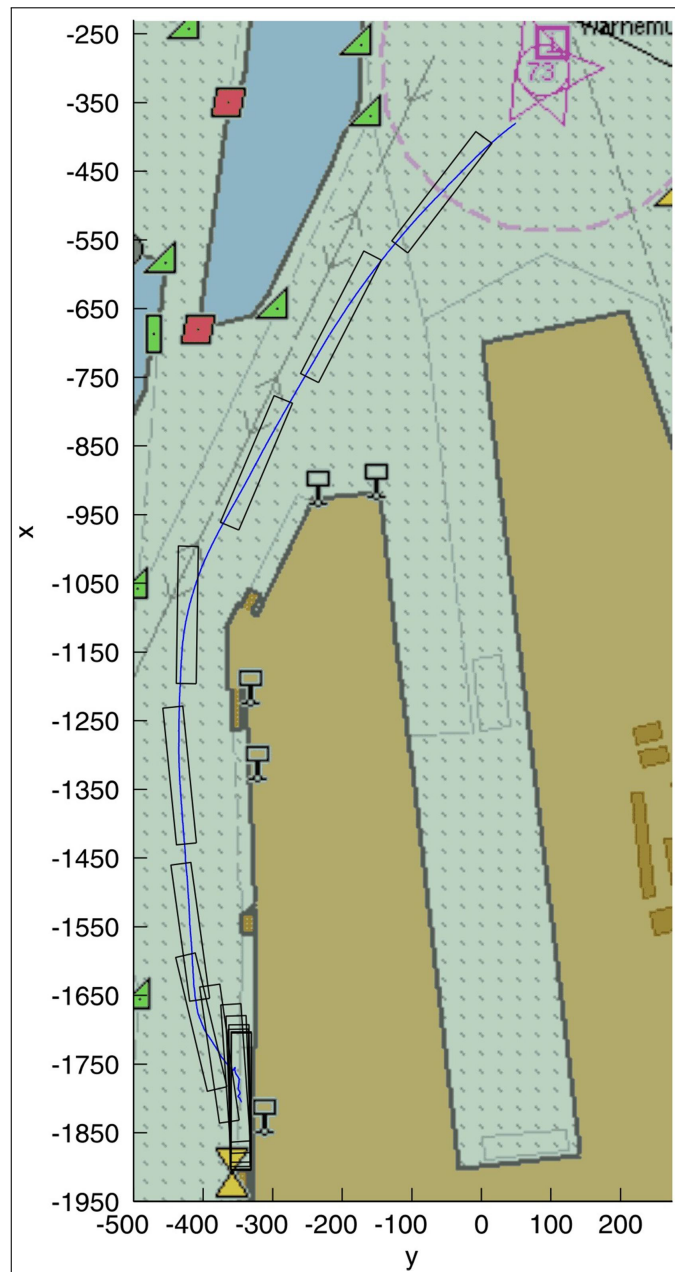


Figura 1.2: Reproducción de una maniobra manual de encallamiento del ferry M/V Mecklenburg-Vorpommern, (Kurowski et al., 2013)

Por esta razón los problemas se dan cuando uno hace planeamiento de rutas en canales o espacios confinados, considerando que los vehículos particularmente en estos lugares tienen baja velocidad y por ende su maniobrabilidad es menor. Para esclarecer esta parte vemos que en Kurowski et al. (2013), una ruta fue planeada con un sistema industrial de guía estándar para seguimiento de ruta. La ruta planeada muestra una rutina de maniobra para acercar el ferry al terminal de Rostock, como vemos en la Figura 1.2 que muestra la ruta generada, podemos apreciar que el radio mínimo de giro no fue lo suficiente para calcular la curvatura de la ruta, especialmente en la parte final de la aproximación al puerto como se ve en la Figura 1.3.

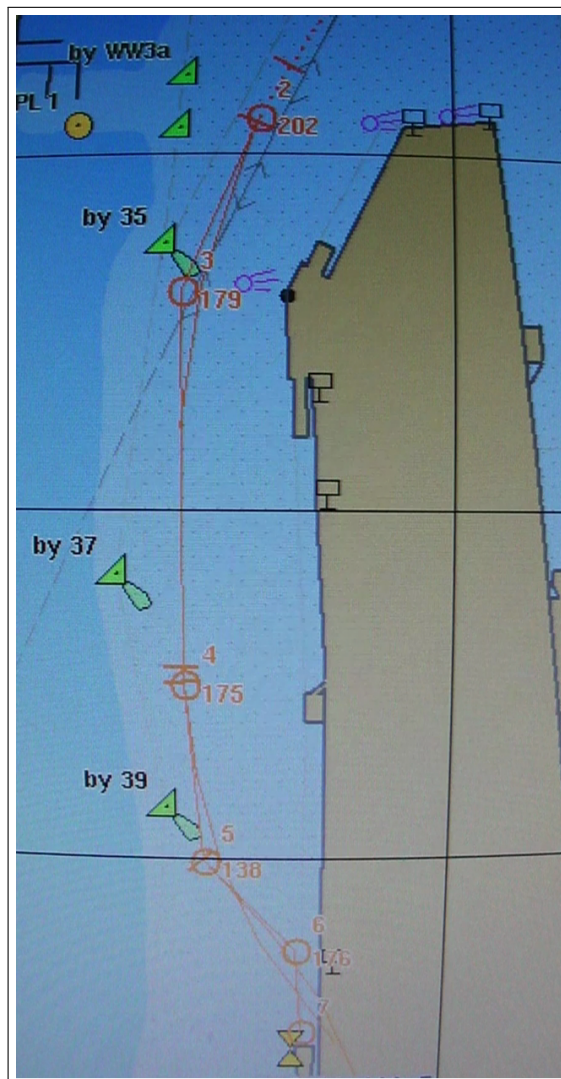


Figura 1.3: Maniobra de atracamiento en el puerto con un sistema de guiado estándar para seguimiento de ruta, (Kurowski et al., 2013)

Lo que se debió hacer, es un análisis para determinar si la maniobrabilidad del barco es suficiente, para seguir la ruta de referencia con la velocidad planeada y con los actuadores que se tiene.

El sistema de posición dinámica (DP) ofrece métodos de planeamiento más eficientes, debido a los modos de operación que presenta. Uno de estos modos es el modo autotrack, el cual es usado para seguir una ruta dada. La ruta también se basa en puntos de camino predefinidos, donde la velocidad y el ángulo de guiñada pueden ser seteados independientemente. Los arcos de circunferencia serán usados en caso de dirección en los puntos de ruta. Adicional a esto también el radio de giro es definido manualmente o calculado automáticamente.

En el caso de navegación dentro de canales o espacios confinados, la definición de ruta de referencia es realizada por la definición de par de puntos en un mapa electrónico de río. Estas coordenadas pueden ser asignadas con un parámetro de corrida tomado desde la curvatura del río. Los datos guardados dados serán interpolados usando funciones cúbicas. Estos resultados son generados fuera de línea, para luego ser cargados al sistema de guiado, (Wahl and Gilles, 2003). La arquitectura actual para este tipo de sistemas se presenta en la Figura 1.4.

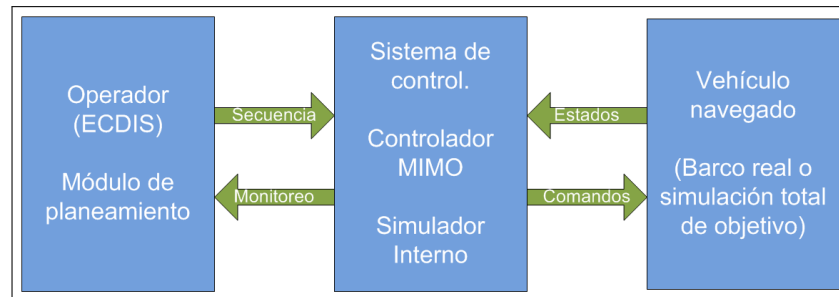


Figura 1.4: Arquitectura usada para el control de embarcaciones, (Korte et al., 2015)

Otro campo de la investigación son métodos capaces de generar rutas de forma automática, esto para ser aplicados a vehículos marinos no tripulados o un grupo de vehículos. El área de movimiento de estos vehículos, es mayormente regiones limitadas y sin considerable impacto de navegación comercial o de potencial peligro. Además las maniobras tienen lugar en el rango de visión de un operador, una esquema frecuentemente usado se muestra en la Figura 1.5, donde se pueden diferenciar los diversos procesos que interactúan, para lograr seguir una ruta deseada.

Un grupo de investigadores han desarrollado diferentes procedimientos para la generación de ruta. Zizzari et al. (2010), que desarrolla generación de ruta considerando restricciones dinámicas, mientras Pedersen and Fossen (2012), se basan en flujo de potencial para la generación de rutas, este método sirve para calcular rutas automáticamente en entornos desordenados. En todos estos casos el oficial responsable de la navegación del barco está incluido solo de forma condicional. De allí que, la realización práctica de estos métodos aun no son ampliamente acep-

tados. Debido al estado actual de la tecnología computacional, otros métodos de planeamiento de ruta obtienen cada vez más importancia, como el desarrollado por Benedict et al. (2010), donde se presenta un software que es capaz de simular y ser usado como una herramienta de predicción para visualizar la ruta obtenida inmediatamente. Con este tipo de algoritmos y software es posible predecir maniobras, estos métodos usan conocimiento experto de oficiales de navegación, pero de una manera que se requiere alto consumo de tiempo.

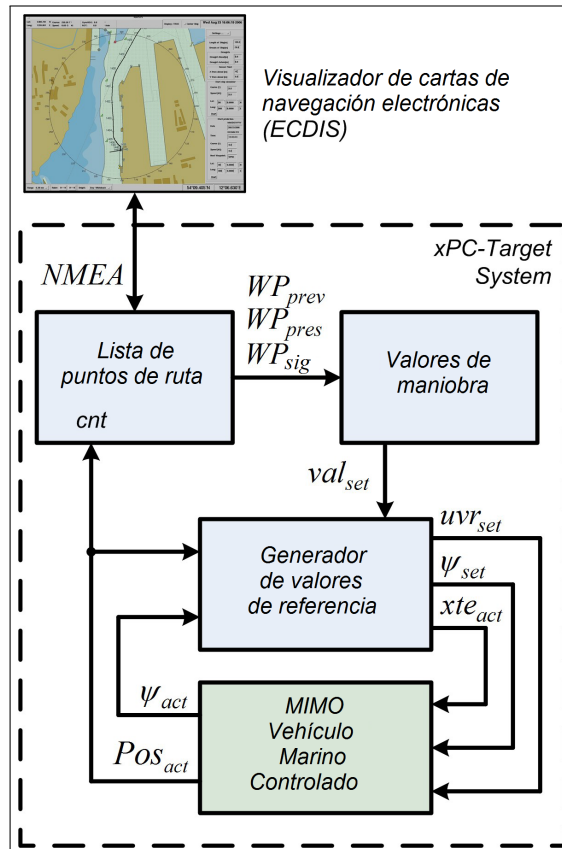


Figura 1.5: Diagrama esquemático del planeamiento de maniobra y sistema de guiado, (Kurowski et al., 2013)

Como hace referencia Breivik (2003), la factibilidad de realizar un ruta por parte de la embarcación es definida por propiedades particulares de cada barco, su mínimo radio de giro y su respuesta dinámica. Según Breivik, no tiene que ver la forma de la ruta con el proceso de diseño de control, por lo cual él define el objetivo de control como **el problema de maniobra** que se divide en dos:

Tarea geométrica: fuerza y torque necesaria para que el barco converja y siga la ruta deseada.

Tarea dinámica: hacer que el barco se mueva a la velocidad deseada, o dado como un perfil de velocidad a lo largo de la ruta, o ingresada por el piloto de la

embarcación.

Esta separación de tareas se hace referencia en Breivik (2003), Alessandretti et al. (2013), y nos hace ver claramente cuáles son los objetivos de control. Además, se observa claramente que hay una diferencia que existe entre seguimiento de ruta y seguimiento de trayectoria.

Problema seguimiento de trayectoria.- Tiene como entrada la trayectoria deseada en función del tiempo. $p_d(t)$ y el objetivo es calcular la ley de control de tal manera que $\|p(t) - p_d(t)\| = 0$, gráficamente se puede observar la Figura 1.6.

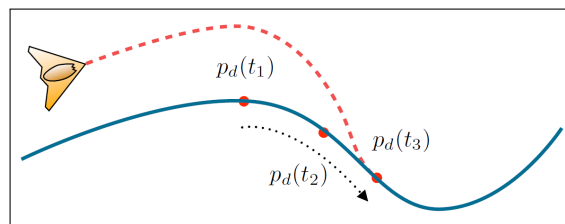


Figura 1.6: Problema de seguimiento de trayectoria, (Alessandretti et al., 2013)

Problema seguimiento de ruta.- Tiene como entrada una ruta deseada parametrizada $p_d(\theta)$, parámetro de velocidad θ y el objetivo es calcular la ley de control de tal manera que $\|p(t) - p_d(\theta(t))\| = 0$ y $\|\dot{\theta}(t) - \dot{\theta}_d\| = 0$, gráficamente se puede observar la Figura 1.7.

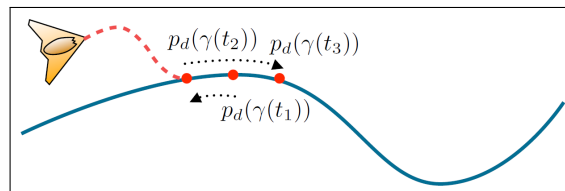


Figura 1.7: Problema de seguimiento de ruta,(Alessandretti et al., 2013)

Debido a que los problemas de maniobra significan controlar 3DOF, que es ambas posiciones y el ángulo de guiñada, para lograr esto se requieren actuadores que sean capaces de brindar fuerza y torque de tal manera que se pueda cumplir con los objetivos deseados. En Skjetne (2003), se explica el uso de las coordenadas de Serret-Frenet en estas aplicaciones, ver Figura 1.8, donde se observa que se pueden caracterizar dos tipos de error, uno perpendicular y otro tangencial a la trayectoria, además hacen referencia que el timón puede encargarse de compensar el error perpendicular mientras que el error de velocidad deseada puede ser compensando por las hélices o propulsores.

Fossen (2011), Skjetne et al. (2001) y Triantafyllou and Hover (2003), se hace referencia que uno de los modelos más conocidos y aplicados para realizar el

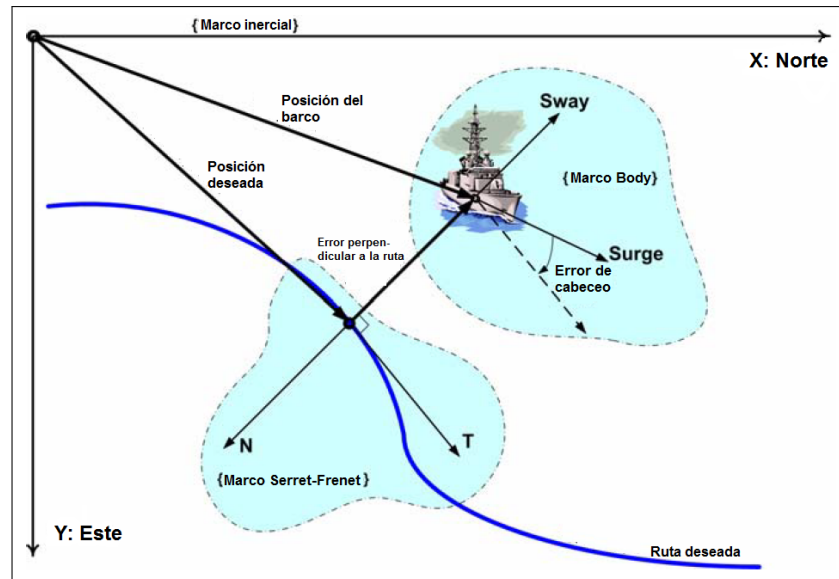


Figura 1.8: Barco con marco inercial y ruta deseada (Línea azul), los respectivos ejes de Serret-Frenet, (Skjetne, 2003)

control de embarcaciones (autopilotaje), son los modelos de Nomoto.

En Fossen (2011) y Triantafyllou and Hover (2002), se hace referencia a los modelos de embarcaciones de forma general, y se puede ver que los modelos dependen de una ecuación cinemática y dinámica, las cuales son derivadas del análisis de la dinámica de cuerpo rígido y de las fuerzas hidrodinámicas. Los modelos de alta velocidad presentan algunos problemas como por ejemplo:

- La matriz de inercia del sistema no es simétrica.
- No se ha especificado como modelar las fuerzas que se generan por navegar cerca de obstáculos o aguas poco profundas.
- La distribución de actuadores, es particular para cada embarcación. De ahí que, para realizar el control se asigna un nuevo caso a cada barco.

Además de estos problemas, se debe considerar que las ecuaciones cinemáticas y dinámicas son muy complicadas y además derivan en un análisis matemático muy difícil. Si la distribución entre actuadores se hace de tal manera que se generan fuerza y torque de tal manera que se puede realizar el control de cada uno de los grados de libertad, se dice que es un barco totalmente actuado. Por otro lado, si existen fuerzas y momentos cercanos al punto de operación que no pueden ser realizados por el sistema actuador del barco, se dice que el barco es sub-actuado.

Como se ve en Fossen (2011), Skjetne (2003) y Skjetne et al. (2001), para realizar el control se asume que la matriz de inercias es simétrica entre estribor

y babor, con esto se logrará desacoplar la velocidad de avance del barco, con lo cual un control independiente se encarga de mantener la velocidad en la velocidad deseada. Esta es una de las bases de diseño en Skjetne (2003).

Las consideraciones generales para el control de ruta de un barco, se presentan en Skjetne et al. (2001), donde se hace referencia que dado una ruta deseada realizable, los errores de tipo radial y en la orientación son descompuestas en el sistema coordinado Serret-Frenet. (Ver Figura 1.8). De acuerdo los objetivos de control son los siguientes: primero, que la velocidad deseada sea alcanzada y controlada por el sistema de control desacoplado, en segundo lugar, que el timan sea usado de tal manera que se elimine el error radial que se presente respecto de la ruta deseada. Como resultado se obtiene que la embarcación se mueva sobre la ruta deseada. En Skjetne et al. (2001), se plantea un esquema de control general donde se muestran los sistemas necesarios para el control de embarcaciones, se puede diferenciar los sistemas de guiado, control y navegación (Ver Figura 1.9). Además, se tiene un módulo de entradas externas, que podría ser reemplazado por un sistema de planeamiento de movimiento.

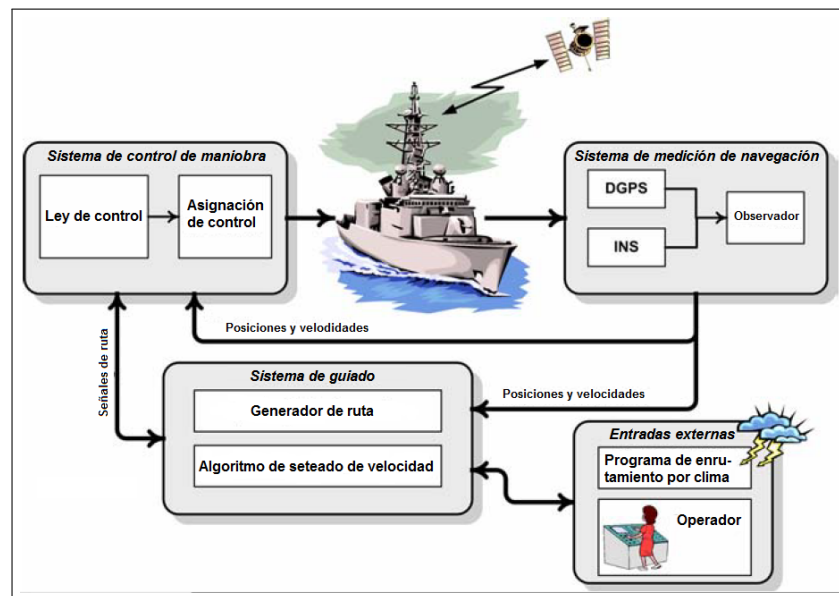


Figura 1.9: Sistemas que interactúan para el control de embarcaciones, (Skjetne, 2003).

Líneas arriba se dijo que los sistemas pueden ser tratados como totalmente actuados, así como sub-actuados, en Zhang et al. (2015) se explica con mayor detalle lo referido a sub-actuados, que es lo que se aplicará en esta tesis. Además de esto se desarrolla la idea de “Barco virtual”, que se muestra en la Figura 1.10, y a partir de esta se desarrolla una ley de orientación original, partiendo de que la referencia de ruta generada son puntos de camino dados, además se observa

que el diseño del control adaptivo es independiente de la planta.

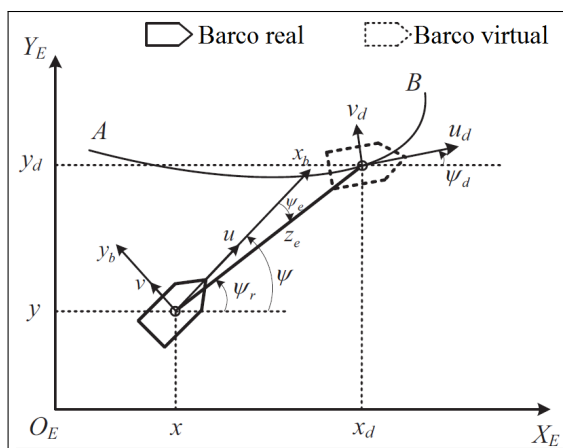


Figura 1.10: Esquema de barco virtual, (Zhang et al., 2015).

En Lekkas et al. (2014b), se muestra como minimizar los errores en los ejes radiales y tangenciales respecto a una ruta deseada, considerando que la embarcación es sub-actuada. Esto a través del desarrollo del sistema de guiado, para esto usan la técnica línea de visión (LOS), que se usa para minimizar el error perpendicular a la ruta. En Lekkas et al. (2014a), se muestra la generación de ruta usando los polinomios, teniendo como entrada puntos de camino, obtenidos por el planeamiento de movimiento, luego muestran el sistema de guiado que usa la ley de guiado LOS, se puede ver que esta técnica trae como resultado una maniobra de comportamiento más flexible que contribuye a que se alcance la ruta deseada de forma más rápida y segura. En Bibuli et al. (2014), se presentan resultados experimentales y además se presentan la modificación del método LOS para calcular la ley del sistema de guiado, la modificación viene dada por la integral que se agrega, siendo esta para compensar cualquier perturbación o corriente no deseada.

Para la generación de ruta paramétricas, si bien se muestran diferentes técnicas y herramientas, lo que se debe garantizar al momento de generar las ecuaciones de ruta paramétricas es que tenga suavidad en los puntos de camino, por ello generalmente se busca que las ecuaciones de ruta tenga segunda derivada continua, en Lekkas et al. (2013) se muestra la generación de ruta paramétrica haciendo uso de la espiral de Fermat, se aprovecha de esta técnica por sus características parte con una curvatura igual a cero y va creciendo gradualmente a medida que va avanzando, esta es una propiedad muy provechosa si lo que se requiere es conectar con un línea recta sin generar discontinuidades. Candeloro et al. (2013), presenta la metodología para la generación de rutas a partir de un mapa, usando líneas rectas y curvas definidas por la espiral de Fermat, el planeamiento de movimiento

se da mediante el diagrama de Voronoi.

1.6. Solución propuesta

1.6.1. Breve descripción de la solución propuesta

La solución para el problema de generación y seguimiento de trayectoria, partirá de realizar el planeamiento de movimiento, tomando como punto de partida un mapa, este se pasará a una simplificación poligonal, luego se aplicará la técnica de Diagrama de Voronoi para obtener como resultado los posibles puntos que definirán la trayectoria. Luego para obtener la trayectoria más corta se aplicará el algoritmo de Dijkstra, que es siempre capaz de dar proporcionar la mejor solución en un tiempo razonable y además no se requiere una configuración especial para aplicarlo, seguidamente se realizará un filtrado de los puntos que hacen que el curso cambie indebidamente. Finalmente con los puntos obtenidos se realizará la generación de ecuaciones paramétricas de ruta a seguir.

Con la ruta definida se realiza el seguimiento de la misma aplicando el sistema de guiado “línea de visión” (LOS) ó de marco de referencia móvil se definiría el tema de ángulo de guiñada deseado para poder alcanzar la ruta de referencia. Para lograr seguir la ruta deseada se implementará un controlador no lineal avanzado de tal manera que se pueda seguir la ruta a la velocidad deseada.

1.6.2. Contribución de la tesis

La enfoque principal de esta tesis esta en el planeamiento y seguimiento de rutas en espacios confinados como archipiélagos, canales, ríos y zonas cercanas a puertos.

Una contribución importante será es el uso de un modelo y controlador no lineal para lograr el seguimiento de ruta deseada por la interpolación polinómica de Hermite y la espiral de Fermat, su comparación y su comportamiento.

Otras contribuciones vienen dadas por:

- Uso de técnica de planeamiento de movimiento usada en la robótica, en aplicaciones marítimas.
- Revisión de técnicas de planeamiento de ruta y temas de continuidad de curvas paramétricas.

-
- Desarrollo y explicación de los algoritmos de generación de rutas, sistema de guiado y control no lineal.
 - Simulación y resultados de la evaluación de las rutas propuestas y comparación del performance que se consigue con cada método.

Capítulo 2

Modelo Matemático

2.1. Introducción

A continuación se presenta el procedimiento para obtener el modelo matemático, el cual nos servirá para determinar, simular el comportamiento dinámico del barco, y también para realizar el diseño del controlador.

El modelamiento matemático es importante para el diseño de control debido a que los controladores son implementados en software.

El diseño del controlador es posible a través de aproximaciones una vez que el modelo matemático está disponible. Para aplicaciones marinas es obtener la fuerza y torque necesario, que se traduce en velocidad de hélices y diferencia de velocidades, para que el barco tenga el comportamiento deseado. El comportamiento deseado se puede reflejar a través de posición, velocidad o aceleración deseada ó la combinación de estos.

Prueba de simulación de controladores, para su realización también se hace uso de modelos matemáticos. La simulación permite darle al controlador las entradas para realizar el cálculo de ley de control, de forma segura, sin consumo de tiempo, sin costo ni peligro innecesario. Otra de las ventajas es la reproductibilidad que se puede alcanzar.

El modelo es siempre una simplificación, es prácticamente imposible e innecesario describir cada fuerza o fenómeno que interactúa con la planta que se está analizando. Además, se conoce que algunas leyes o ecuaciones físicas pueden ser simplificadas, lo único que se requiere es representar el comportamiento, por ello muchas veces los comportamientos son determinados solamente a través de datos experimentales a partir de lo cual se aprovecha sus particularidades para poder traducirlos en formas matemáticas conocidas.

La precisión necesaria del modelo depende naturalmente del uso que se le va a dar. Para realizar el diseño de los controladores es solamente necesario incluir las

principales propiedades físicas, Sorensen 2012 (22), caso diferente se da cuando la necesidad es realizar una simulación del comportamiento de la planta, en ese caso si se requiere la mayor precisión y fidelidad posible para poder conocer el comportamiento del modelo.

Teniendo en cuenta que, nuestro objetivo es generar una ruta segura y hacer que se siga con escaso margen de error en aplicaciones de espacios confinados, es necesario realizar un correcto diseño de controlador y poder simular de tal manera que se pueda mostrar el comportamiento más real posible a fin de evitarnos riesgos innecesarios al hacer pruebas reales.

Cuando se habla de modelos matemáticos de embarcaciones marinas, se tienen que considerar la aplicaciones a las que va a ser sometido el modelo, hay aplicaciones como *Dynamic Position* que ameritan tener solo modelos para baja velocidad, Fossen (2011) y Fossen et al. (2003), mientras otras como en embarcaciones de alta velocidad donde sí debería considerarse todas las ecuaciones dinámicas y tener un modelo completo.

Para nuestro caso se ha considerado tener el modelo de 3 grados de libertad considerando todos los parámetros dinámicos y cinemáticos, a partir del cual se realizarán acorde a pruebas la deducción de las aproximaciones necesarias a fin de no restringir algún comportamiento dinámico importante.

2.2. Dinámica

Para estudiar la dinámica dividimos dicha tarea en dos partes, según (Fossen, 2011)

- Cinemática, que trata solamente aspectos geométricos y de movimiento.
- Cinética, que analiza las fuerzas que causan el movimiento.

En esta sección se resume la cinemática y cinética necesaria para determinar un modelo matemático según la referencia vista en Fossen (2011, Cap. 2 y Cap. 3).

2.2.1. Sistemas de referencia:

Para definir el movimiento de una embarcación marina, es necesario definir seis coordenadas independientes que definan la posición y orientación, las tres primeras corresponder a su posición y traslación a lo largo de los ejes x , y y z , mientras que los tres últimos corresponden a su orientación y rotación, para el caso de embarcaciones marinas se ha desarrollado un estandar que se presenta en la tabla 2.1.

EIC: es un sistema de referencia inercial, generalmente se usa para navegación terrestre, en las cuales las leyes de Newton se pueden considerar y trabajar. El origen de este sistema está en el centro de la tierra y por lo general lo usan los sistemas de navegación inercial.

ECEF: Tiene origen en el centro de la tierra, pero los ejes rotan relativamente al sistema inercial. La velocidad de rotación es $\omega_e = 7.2921105 \frac{rad}{s}$. Para sistemas con baja velocidad relativa se puede considera r como inercial, este sistema se muestra en la Figura 2.1.

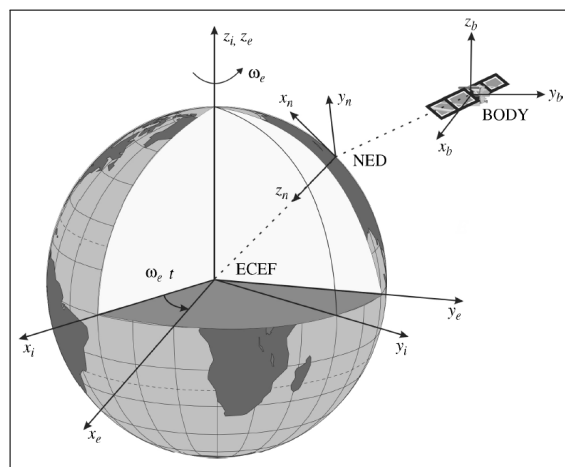


Figura 2.1: Diferentes sistemas inerciales, (Fossen, 2011)

NED: Es un sistema definido en base a la referencia de elipsoide terrestre. Se encuentra relacionado con el sistema ECEF mediante la longitud y latitud y también se muestra en la Figura 2.1.

BODY: Es un marco referencial móvil que está fijado a la embarcación. La posición y orientación de la embarcación son descritas relativamente respecto del marco de referencia inercial. El origen OB es elegido usualmente con el punto medio de las embarcaciones en la línea del agua, esta se puede referenciar con el punto CO, ver Figura 2.2.

Donde:

- **CG** - Centro de gravedad.
- **CB** - Centro de flotabilidad.
- **CF** - Centro de flotation (localizada a una distancia LCF desde el CO en la dirección X).

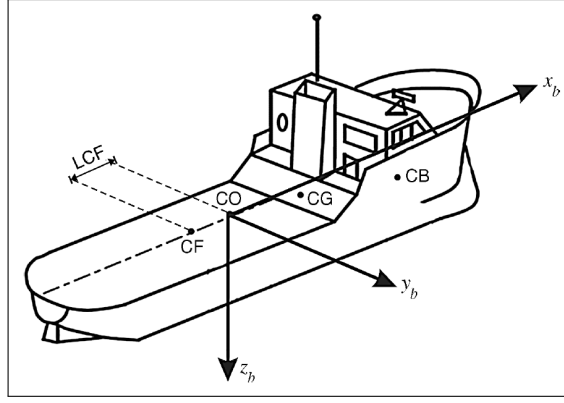


Figura 2.2: Punto de referencia fijados al cuerpo

Como la gran mayoría de cuerpos dinámicos, se consideran seis grados de libertad los cuales serán 3 de posición y 3 de rotación, estos definen 6 movimientos en el barco los cuales se pueden ver en la Tabla 2.1.

Tabla 2.1: Tabla que contiene los estados de un vehículo marino (η y ν) y sus descripciones,(Moe, 2013).

Vector	Vector	Var	Nombre	Definición
$\eta \left(\eta_{b/n}^n \right)$	$P_{b/n}^n$	x	Surge	Posición en la dirección X en NED
		y	Sway	Posición en la dirección Y en NED
		z	Heave	Posición en la dirección Z en NED
	Θ_{nb}	ϕ	Roll	Rotación en la dirección Y en NED
		θ	Pitch	Rotación en la dirección Y en NED
		ψ	Yaw	Rotación en la dirección Z en NED
$\nu \left(\nu_{b/n}^n \right)$	$v_{b/n}^b$	u	Vel. surge	Velocidad en la dirección x en BODY
		v	Vel. sway	Velocidad en la dirección y en BODY
		w	Vel. heave	Velocidad en la dirección z en BODY
	$\omega_{b/n}^b$	p	Vel. roll	Velocidad angular sobre el eje-x en BODY
		q	Vel. pitch	Velocidad angular sobre el eje-y en BODY
		r	Vel. yaw	Velocidad angular sobre el eje-z en BODY

Para nuestro caso consideramos 3 grados de libertad o 3 movimientos que serán *surge*, *sway* y *yaw*, en base a esto realizaremos el modelo matemático.

2.2.2. Dinámica del centro de gravedad

Para lograr tener la dinámica del barco realizamos la dinámica del centro de gravedad para luego realizar un traslado y tener la dinámica de cualquier punto dentro del mismo. Tomando como vectores unitarios del sistema coordenado NED $\hat{I}, \hat{J}, \hat{K}$ y del sistema coordenado BODY $\hat{i}, \hat{j}, \hat{k}$, se tiene que:

$$\begin{cases} \hat{i} = \cos(\psi) \hat{I} + \sin(\psi) \hat{J} \\ \hat{j} = -\sin(\psi) \hat{I} + \cos(\psi) \hat{J} \end{cases} \rightarrow \begin{bmatrix} \hat{i} \\ \hat{j} \end{bmatrix} = \begin{bmatrix} \cos(\psi) & \sin(\psi) \\ -\sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} \hat{I} \\ \hat{J} \end{bmatrix} \quad (2.1)$$

$$\rightarrow \begin{bmatrix} \hat{I} \\ \hat{J} \end{bmatrix} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} \hat{i} \\ \hat{j} \end{bmatrix} \quad (2.2)$$

Luego simplemente calculamos la razón de cambio de los vectores:

$$\begin{cases} \frac{d\hat{i}}{dt} = -\sin(\psi) \dot{\psi} \hat{I} + \cos(\psi) \dot{\psi} \hat{J} = \hat{j} \dot{\psi} \\ \frac{d\hat{j}}{dt} = -\cos(\psi) \dot{\psi} \hat{I} - \sin(\psi) \dot{\psi} \hat{J} = -\hat{i} \dot{\psi} \end{cases} \quad (2.3)$$

Aplicamos las leyes de Newton para un sistema de referencia inercial:

Primera condición de equilibrio dinámico:

$$\sum \vec{F} = m \vec{a} \quad (2.4)$$

Segunda condición de equilibrio dinámico:

$$\sum \vec{\tau} = I \vec{\alpha} \quad (2.5)$$

Para la primera condición ecuación (2.4), es la sumatoria de fuerzas aplicadas sobre el cuerpo en el centro de gravedad sobre el cuerpo, m es la masa del cuerpo y \vec{a} es la aceleración lineal del centro de gravedad del cuerpo; para la ecuación (2.5) es la sumatoria de torques aplicados en el centro de gravedad del cuerpo, I es la matriz de inercias del cuerpo respecto al centro de gravedad y $\vec{\alpha}$ es la aceleración angular del cuerpo. Se considera que I y m no varían ya que estamos trabajando con un cuerpo rígido. Para el caso de tres dimensiones que se está planteando, las condiciones de equilibrio dinámico quedarían expresadas por:

$$\vec{F} = m \vec{a} = m \underbrace{\begin{bmatrix} \ddot{N} \\ \ddot{E} \end{bmatrix}}_{\ddot{p}_g} = m \ddot{p}_g \quad (2.6)$$

$$\tau_G = I_G \alpha = I_G \ddot{\psi} \quad (2.7)$$

Donde \vec{F} es un vector libre de la suma de fuerzas, \ddot{p}_G es el vector de aceleración del centro de gravedad respecto al sistema inercial, τ_G es la suma de torques sobre el centro de gravedad (considerando que se encuentra paralela al eje z), I_G la inercia respecto al centro de gravedad y $\ddot{\psi}$ la aceleración angular del cuerpo. Tomando el vector de posición del centro de gravedad en el sistema inercial:

$$p_g = N\hat{I} + E\hat{J} \quad (2.8)$$

Derivando respecto del tiempo:

$$\dot{p}_g = \dot{N}\hat{I} + \dot{E}\hat{J} \quad (2.9)$$

La velocidad puede ser descompuesta en cualquier sistema coordenado ya que es un vector libre. Escogiendo el sistema BODY.

$$\dot{p}_g = \dot{N}\hat{I} + \dot{E}\hat{J} = p_{g_x}\hat{i} + p_{g_y}\hat{j} \quad (2.10)$$

Llamando $p_{g_x} = u_g$, $p_{g_y} = v_g$, se tiene:

$$\dot{p}_g = \dot{N}\hat{I} + \dot{E}\hat{J} = p_{g_x}\hat{i} + p_{g_y}\hat{j} = u_g\hat{i} + v_g\hat{j} \quad (2.11)$$

Derivando respecto al tiempo:

$$\ddot{p}_g = \dot{u}_g\hat{i} + u_g\frac{d\hat{j}}{dt} + \dot{v}_g\hat{j} + v_g\frac{d\hat{i}}{dt} = \dot{u}_g\hat{i} + u_g\hat{j}\psi + \dot{v}_g\hat{j} + v_g(-\hat{i}\dot{\psi}) \quad (2.12)$$

Llamando a $\dot{\psi} = r$:

$$\ddot{p}_g = (\dot{u}_g - v_g r)\hat{i} + (\dot{v}_g + u_g r)\hat{j} \quad (2.13)$$

De donde las ecuaciones de equilibrio dinámico quedarían como:

$$\vec{F} = m\ddot{p}_g = m(\dot{u}_g - v_g r)\hat{i} + m(\dot{v}_g + u_g r)\hat{j} \quad (2.14)$$

$$\tau_G = I_G \dot{r} \quad (2.15)$$

Proyectando el vector de fuerza \vec{F} en en los ejes x e y , las ecuaciones quedarían:

$$\begin{aligned} F_{G_x} &= m(\dot{u}_g - v_g r)\hat{i} \\ F_{G_y} &= m(\dot{v}_g + u_g r)\hat{j} \end{aligned} \quad (2.16)$$

$$\tau_G = I_G \dot{r}$$

Luego de tener este resultado y a fin de tener el control en el punto CO (Ver Figura 2.3), prácticamente se realizará un traslado de puntos el cual estará afectado por todas las derivadas y demás, esto se desarrollará en la siguiente sección a considerar.

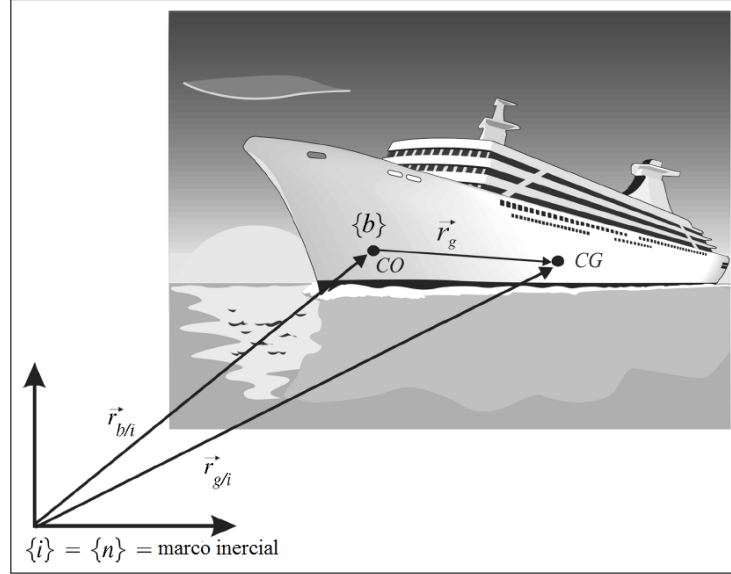


Figura 2.3: Definición de origen de coordenadas de centro y gravedad, (Fossen, 2011).

2.2.3. Dinámica del punto CO

La posición del centro de gravedad puede ser expresada en función a la posición del centro CO con respecto al sistema inercial:

$$p_g = p_o = x_g \hat{i} \quad (2.17)$$

Donde x_g es la distancia medida desde el centro CO hacia el centro de gravedad y p_o es un vector libre de posición del centro O con valor $p_o = x_g \hat{I} + y_g \hat{J}$ (x e y no son los mismos referidos a los ejes coordenados).

Derivando respecto al tiempo:

$$\dot{p}_g = \dot{p}_o + \dot{x}_g \hat{I} + \dot{x}_g \hat{J} + x_g \frac{d\hat{i}}{dt} = \dot{p}_o + x_g \hat{J} \dot{\psi} \quad (2.18)$$

Considerando, $\dot{x}_g = 0$ ya que el barco es un cuerpo rígido y sin variaciones en sus distancias internas. Se puede descomponer la velocidad del centro O (p_o) en el sistema BODY, teniéndose $\dot{p}_o = u_0 \hat{i} + v_0 \hat{j}$ usando la explicación análoga a la descomposición de la velocidad del centro de gravedad.

$$\dot{p}_g = u_0 \hat{i} + v_0 \hat{j} + x_g \hat{j} \dot{r} \quad (2.19)$$

Derivando respecto al tiempo:

$$\ddot{p}_g = \dot{u}_0 \hat{i} + \dot{v}_0 \hat{j} + u_0 \frac{d\hat{i}}{dt} + v_0 \frac{d\hat{j}}{dt} + x_g \frac{d\hat{j}}{dt} \dot{r} + x_g \hat{j} \ddot{r} \quad (2.20)$$

$$\ddot{p}_g = (\dot{u}_0 - v_0 r - x_g r^2)\hat{i} + (\dot{v}_0 + u_0 r + x_g \dot{r})\hat{j} \quad (2.21)$$

Reemplazando \ddot{p}_g en las ecuaciones de equilibrio dinámico:

$$\begin{aligned} \vec{F} = m\ddot{p}_g &= m(\dot{u}_0 - v_0 r - x_g r^2)\hat{i} + m(\dot{v}_0 + u_0 r + x_g \dot{r})\hat{j} \\ \tau_G &= I_G \dot{r} \end{aligned} \quad (2.22)$$

Descomponiendo el vector de fuerza en sus proyecciones en los ejes x e y, las ecuaciones quedarían como:

$$\begin{aligned} F_{G_X} &= m(\dot{u}_0 - v_0 r - x_g r^2) \\ F_{G_Y} &= m(\dot{v}_0 + u_0 r + x_g \dot{r}) \\ \tau_G &= I_G \dot{r} \end{aligned} \quad (2.23)$$

Dado que se trabajará directamente sobre el centro O, se quiere tener las fuerzas y el torque referidos al mismo. Trasladando las fuerzas al centro O:

$F_{O_X} = F_{G_X}$, ya que la fuerza lineal no cambia según sea el punto de aplicación. $F_{O_Y} = F_{G_Y}$, de la misma forma que el punto anterior. $\tau_O = \tau_G + \tau_{F_{O_Y}}$, ya que el torque τ_G se traslada sin modificaciones, la fuerza $F_{O_X} = F_{G_X}$ no genera torque ya que pasa por el centro O y la fuerza $F_{O_Y} = F_{G_Y}$ genera un torque $\tau_{F_{O_Y}} = x_g F_{G_Y} = x_g F_{O_Y}$ al ser trasladada al punto O.

Expresando las fuerzas y torque respecto al centro O, de la ecuación (2.22) :

$$\begin{aligned} \tau_G + \tau_{F_{O_Y}} &= I_G \dot{r} + (x_g F_{O_Y}) \rightarrow \tau_O = I_G \dot{r} + x_g m(\dot{v}_0 + u_0 r + x_g \dot{r}) \\ &= (I_G + x_g^2 m)\dot{r} + x_g m(\dot{v}_0 + u_0 r) \end{aligned} \quad (2.24)$$

Por el teorema de Steiner, la inercia respecto al centro O sería la inercia respecto al centro de gravedad sumada con la masa del cuerpo por el cuadrado de la distancia entre el centro O y el centro de gravedad, esto es $I_O = I_G + x_g^2 m$. Entonces:

$$\tau_O = I_O \dot{r} + x_g m(\dot{v}_0 + u_0 r) \quad (2.25)$$

Renombrando a las variables, $F_{G_X} = F_u$, $F_{G_Y} = F_v$, $u_O = u$, $v_O = v$, se tiene:

$$\begin{cases} F_u = m(\dot{u} + vr + x_g r^2) \\ F_v = m(\dot{v} + ur + x_g \dot{r}) \\ \tau_o = I_o \dot{r} + x_g m(\dot{v} + ur) \end{cases} \quad (2.26)$$

Como las ecuaciones están referidas a la velocidad del barco en el sistema

BODY, se deben calcular las velocidades en el sistema NED para que puedan ser usadas para el cálculo posterior de posiciones (mediante las ecuaciones diferenciales obtenidas), para eso hacemos uso de la ecuación (2.2), la cual nos permitirá expresar un vector del sistema BODY hacia el sistema NED. Utilizando $p_O = x\hat{I} + y\hat{J} \rightarrow \dot{p}_O = \dot{x}\hat{I} + \dot{y}\hat{J}$ se tiene:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \quad (2.27)$$

Añadiendo la igualdad de velocidad rotacional entre el sistema BODY y el sistema NED, $\dot{\psi} = r$, se tiene:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{J(\psi)} \begin{bmatrix} u \\ v \\ r \end{bmatrix} \quad (2.28)$$

Donde $J(\psi)$ vendría a ser la matriz de rotación para pasar del sistema BODY al sistema NED.

2.2.4. Representación del modelo en forma matricial

A efectos de realizar los cálculos respectivos así como las simulaciones se hace de suma importancia lograr poner las ecuaciones de forma matricial, por lo cual a continuación se reordenará y dará la forma de tal manera que se logró tener la forma matricial del modelo.

Del sistema:

Partiendo de la ecuación (2.26) se reordenarán las ecuaciones para realizar un agrupamiento. Posteriormente se agruparán aceleraciones y velocidades como sigue:

$$\begin{bmatrix} F_u \\ F_v \\ \tau_o \end{bmatrix} = \begin{bmatrix} m\dot{u} \\ m\dot{v} + x_g m\dot{r} \\ I_o\dot{r} + x_g m\dot{v} \end{bmatrix} + \begin{bmatrix} -mvr - mx_g rr \\ mur \\ mvu + x_g mru - muv \end{bmatrix} \quad (2.29)$$

Reordenando las velocidades y aceleraciones para formar matrices:

$$\begin{aligned}
\underbrace{\begin{bmatrix} X \\ Y \\ N \end{bmatrix}}_{F_{RB}} &= \underbrace{\begin{bmatrix} F_u \\ F_v \\ \tau_o \end{bmatrix}}_{F_{RB}} = \underbrace{\begin{bmatrix} m & 0 & 0 \\ 0 & m & x_g m \\ 0 & x_g m & I_O \end{bmatrix}}_{M_{RB}} \underbrace{\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{r} \end{bmatrix}}_{\dot{\nu}} + \\
\underbrace{\begin{bmatrix} 0 & 0 & -mv - mx_g r \\ 0 & 0 & mu \\ mv + mx_g r & -mu & 0 \end{bmatrix}}_{C_{RB}(\nu)} \underbrace{\begin{bmatrix} u \\ v \\ r \end{bmatrix}}_{\nu} & \quad (2.30)
\end{aligned}$$

A partir de la ecuación (2.30) se obtiene el modelo:

$$F_{RB} = \tau = M_{RB}\dot{\nu} + C_{RB}(\nu)\nu \quad (2.31)$$

Se considera que el subíndice RB (Rigid Body) denota las ecuaciones que se obtuvieron considerando al barco como un cuerpo rígido y que no se incluyeron componentes de otra naturaleza, tales como parámetros inducidos por la interacción con el mar y otros, todos se verán en la siguiente sección.

Estas matrices tienen interesantes propiedades para el diseño de control.

- M_{RB} es simétrica, $M_{RB} = M_{RB}^T$,
- M_{RB} es también positiva definida, cuando su determinante es $\det(M_{RB}) = m^2(I - m(x_o^2 + y_o^2))$, es positiva.
- C_{RB} es simétrica $C_{RB} = -C_{RB}^T$.

El modelo basado en la ecuación (2.31) va ser usado en esta tesis.

2.2.5. Fuerzas actuantes sobre la embarcación marina

En esta sección se trabajará la explicación de las fuerzas actuantes sobre el barco representadas por el lado derecho luego de reordenar la ecuación (2.31):

$$M_{RB}\dot{\nu} + C_{RB}(\nu)\nu = F_{RB} = \tau \quad (2.32)$$

Las fuerzas de empuje son el principal medio para llevar el barco al comportamiento deseado, mientras la principal fuerza disipadora es la fuerza amortiguadora hidrodinámica, también se presentará las fuerzas hidrostáticas, así como fuerzas que se ejercen por fenómenos ambientales, y otras fuerzas más que se puedan sumar.

$$\tau = \tau_{propuls} + \tau_{hidrodin} + \tau_{hidrost} + \tau_{ambiente} + \tau_{otros} \quad (2.33)$$

2.2.5.1. Fuerzas de empuje por parte de los actuadores

Las fuerzas de empuje, son las fuerzas de propulsión y direccionamiento combinado, es la forma principal de poder controlar una embarcación marina, es parte fundamental al momento de hacer el diseño del sistema de control.

La controlabilidad del barco depende claramente de la configuración de actuadores. Existen diferencias claras entre los barcos totalmente actuados y subactuados, las diferencias se dan en la configuración de actuadores. Existen diferentes tipos de configuración entre las cuales podemos destacar:

2.2.5.2. Hélice con timón para direccionamiento

Es la configuración más simple y común que se puede encontrar, para un ejemplo se puede ver la Figura 2.4:

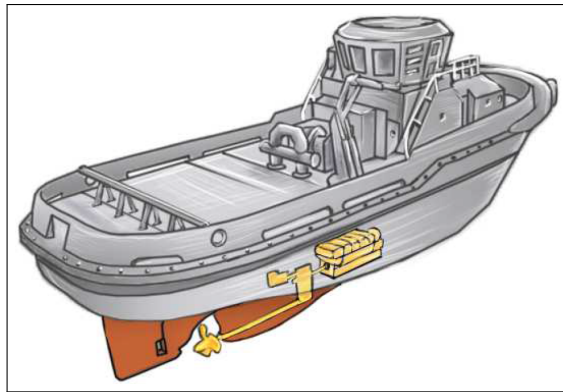


Figura 2.4: Hélice con timón para direccionamiento, (Dahl, 2013)

2.2.5.3. Propulsor azimutal

Esta es constituido por una hélice, que se puede orientar girando sobre un eje vertical y generar impulso en cualquier dirección perpendicular al eje. Este giro puede ser hasta de 360° , lo que mejora la maniobrabilidad haciendo incluso innecesario el timón, ver la Figura 2.5.

2.2.5.4. Otras configuraciones

Adicional a estas dos presentadas, existen otras configuraciones como las impulsadas por agua, hélices aéreas, dos propulsores con tunel de propulsión, ver Figura 2.6 y modernas versiones de veleros aéreos como se muestra en la Figura 2.7.

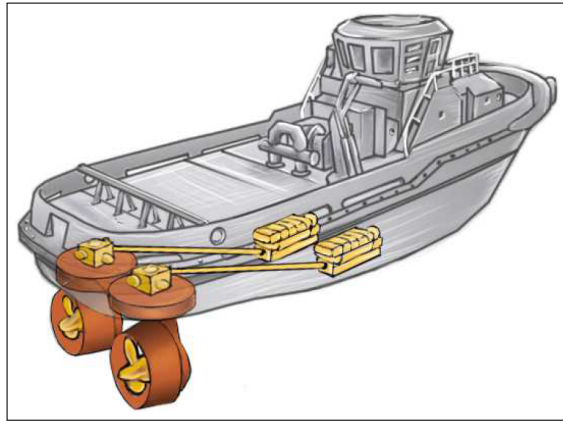
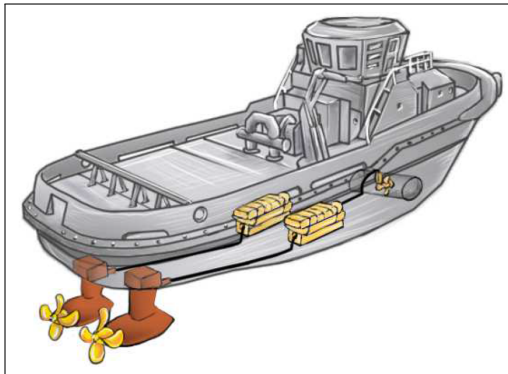
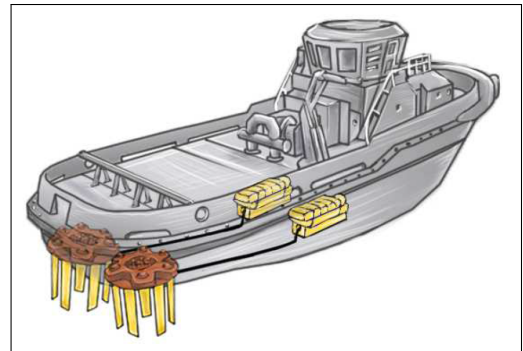


Figura 2.5: Configuración de propulsor azimutal, (Dahl, 2013)



(a) Dos propulsores y tunel



(b) Hélices tipo Voith-Schneider

Figura 2.6: Otras configuraciones de propulsores,, (Dahl, 2013)

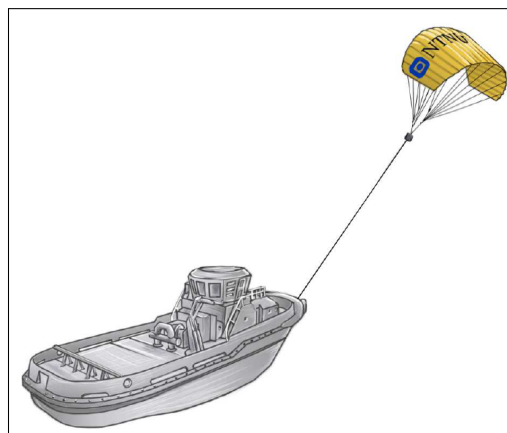


Figura 2.7: Propulsor moderno usando paracaídas, (Dahl, 2013)

2.2.6. Fuerzas hidrodinámicas

Dentro de la navegación, muchos fenómenos hidrodinámicos afectan al barco, inclusive muchos de ellos aun no han podido ser explicados del todo. Considerando un modelo simplificado tenemos lo siguiente:

$$\tau_{\text{Hidrodinamica}} = -(M_A \dot{\nu}_r + C_A(\nu_r)\nu_r) - D(\nu_r)\nu_r \quad (2.34)$$

Este modelo es lo suficientemente preciso para diferentes aplicaciones donde ν_r es la velocidad relativa del barco respecto de la corriente, generalmente se asume que la componente giratoria no presenta velocidad en la corriente.

2.2.6.1. Desplazamiento de masa de agua

El movimiento del barco sobre un fluido, causa el movimiento del mismo. Como se indica en Fossen (2011). El fluido debe moverse al lado y luego cerca al barco. La energía cinética es aquella agregada al fluido. La correspondiente energía debe ser disipada desde el barco.

Para representar este fenómeno, Fossen hace uso del concepto de masa agregada y se representa a través de esta matriz.

$$M_A = \begin{bmatrix} -X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -N_{\dot{v}} & -N_{\dot{r}} \end{bmatrix} \quad (2.35)$$

Para representar el mismo fenómeno, pero correspondiente a la matriz de masa agregada por fuerzas de Coriolis y centrifugas.

$$C_A(\nu_r) = \begin{bmatrix} 0 & 0 & Y_{\dot{v}}\nu_r + Y_{\dot{r}}r \\ 0 & 0 & -X_{\dot{u}}u_r \\ -(Y_{\dot{v}}\nu_r + Y_{\dot{r}}r) & X_{\dot{u}}u_r & 0 \end{bmatrix} \quad (2.36)$$

Los coeficientes de estas matrices representan las derivadas parciales efectos hidrodinámicas respecto a las aceleraciones respectivas.

Se debe tener en cuenta que las derivadas hidrodinámicas no se encuentran disponibles directamente, diferente de los componentes de masa del cuerpo rígido, que usan los parámetros físicos propios del barco. Para obtenerlos se hacen aproximaciones numéricas así como técnicas de identificación a fin de tenerlos.

2.2.6.2. Amortiguación

La energía es disipada del barco de muchas formas diferentes a la vista en la sección anterior. Fossen (2011), hace una lista de potencial de amortiguación,

fricción, fuerza de amortiguamiento por olas, debido al efecto vortex y fuerzas de elevación. Como resultado de todas estas fuerzas se obtiene un matriz de amortiguamiento $D(\nu_r)$.

Tomando como referencia a Fedyaevsky and Sobolev (1964) y Norrbin , proponen el uso de modelos de segundo orden:

$$D(\nu_r) = D_l + D_q(\nu_r) \quad (2.37)$$

Donde:

$$D_l = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix} \quad (2.38)$$

La ecuación (2.38) representa el modelo lineal de amortiguamiento.

$$D_q(\nu_r) = \begin{bmatrix} -X_{|u|u} |u_r| & 0 & 0 \\ 0 & -Y_{|v|v} |v_r| - Y_{|r|v} |r| & -Y_{|v|r} |v_r| - Y_{|r|r} |r| \\ 0 & -N_{|v|v} |v_r| - N_{|r|v} |r| & -N_{|v|r} |v_r| - N_{|r|r} |r| \end{bmatrix} \quad (2.39)$$

La ecuación (2.38), representa el amortiguamiento cuadrático. Todos los coeficientes presentados son derivadas hidrodinámicas. Como se dijo anteriormente estas no son obtenidas directamente.

2.2.7. Fuerzas hidrostáticas

Las fuerzas hidrostáticas vienen dadas por el peso y el empuje (flotabilidad) propio del cuerpo. Juntas estas fuerzas son llamadas las fuerzas de restauración según Fossen (2011).

$$\tau_{Hidrostatico} = -(\mathbf{g}(\boldsymbol{\eta}) + \mathbf{g}_0) \quad (2.40)$$

Donde el primer término representa las fuerzas de restauración mientras el segundo término es una constante dada por la fuerza debido al lastre del equipo. Para el modelo de barco con tres grados de libertad, ó planar, las fuerzas hidrostáticas pueden ser ignoradas ya que actúan en la dirección vertical.

2.2.8. Fuerzas del ambiente

El ambiente actúa de muchas formas sobre el barco, por ejemplo a través de las olas, viento y corriente marina, etc. Incluir todas las fuerzas no es algo práctico, porque lo único que se haría es agregar más variables que quizás no tengan tanta

incidencia en la dinámica. Comúnmente se incluyen las fuerzas dadas por las olas, corriente marina y viento.

$$\tau_{ambiente} = \tau_{olas} + \tau_{corriente} + \tau_{viento} \quad (2.41)$$

2.2.9. Otras fuerzas

Algunas fuerzas de control, no vienen dadas por los actuadores comunes como hélices o propulsores, sino más bien por algunos particulares como aletas, que sirven para aplicaciones particulares como estabilización del cabeceo para mejorar el confort y seguridad de un barco de pasajeros, Perez (2006).

2.3. Modelo de referencia

Para realizar las simulaciones y diseño del controlador no lineal, se tomará como referencia los parámetros de un modelo que ha sido usado para diferentes simulaciones en los artículos científicos revisados, Fossen et al. (2003), Moe (2013), Breivik et al. (2004), Skjetne (2005), Breivik (2003), además de ser un modelo validado y basado en la teoría de modelamiento desarrollado en el presente capítulo. Este modelo es de un barco prototipo de pruebas y muy parecido a los usados para el tema de monitoreo de variables marinas, por tanto es una referencia ideal para poder desarrollar el planeamiento y control del mismo. El modelo de referencia será el CyberShip II (Figura 2.8), que es una réplica de 1:70 de un barco real. La longitud de este es de 1.25m de largo, tiene una configuración de actuadores de tal manera que resulta un barco totalmente actuado.

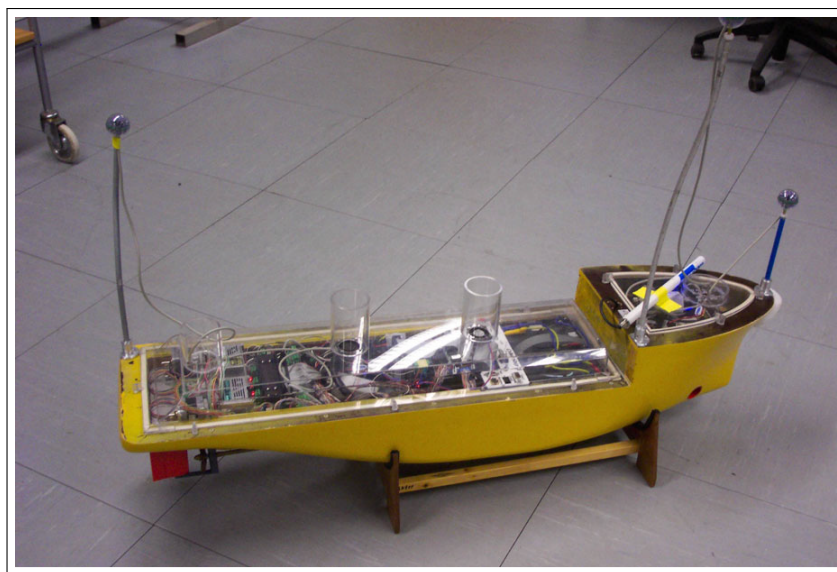


Figura 2.8: Barco a escala CyberShip II, (Corneliussen, 2003)

Este barco ha sido extensivamente estudiado y usado en experimentos en la Universidad Noruega de Ciencia y Tecnológica. En esta sección se resumirá los parámetros del CyberShip II, modelo basado en las siguientes referencias, Lindgaard (2003), Corneliussen (2003), Skjetne et al. (2004). Además se analizará el valor máximo de radio de curvatura que se puede alcanzar, que será una restricción y entrada para el sistema de generación de ruta.

2.3.1. Modelo de simulación

El CyberShip II, es implementado como modelo de simulación de la siguiente forma:

$$\dot{\eta} = \mathbf{R}(\psi)\boldsymbol{\nu} \quad (2.42)$$

$$\mathbf{M}\dot{\boldsymbol{\nu}} = \boldsymbol{\tau} - \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} - \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} \quad (2.43)$$

Donde las matrices \mathbf{M} de masa, $\mathbf{C}(\boldsymbol{\nu})$ de coreolis y $\mathbf{D}(\boldsymbol{\nu})$ de amortiguamiento, serán presentadas con mayor detalle más adelante.

La entrada para el sistema viene dada por el vector de fuerzas y torque $\boldsymbol{\tau}$. Se asume que el sistema de control es implementando de tal manera que el empuje $\boldsymbol{\tau}$, se consiga instantáneamente.

Se debe considerar que ninguna fuerza ambiental o de alguna corriente de viento, es considerada en el modelo.

2.3.1.1. Parámetros Dinámicos

La matriz de masa es:

$$\mathbf{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ 0 & mx_g - N_{\dot{v}} & I_z - N_{\dot{r}} \end{bmatrix} \quad (2.44)$$

Esto corresponde a:

$$\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A \quad (2.45)$$

Donde \mathbf{M}_{RB} es la matriz de inercia de cuerpo rígido y \mathbf{M}_A la matriz de masa agregada.

La matriz de centrifuga–coriolis, viene dada por:

$$\mathbf{C}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & c_{13} \\ 0 & 0 & c_{23} \\ -c_{13} & -c_{23} & 0 \end{bmatrix} \quad (2.46)$$

Donde:

$$\begin{aligned} c_{13} &= -(m - Y_{\dot{v}})v - (mx_g - Y_{\dot{r}})r \\ c_{23} &= (m - X_{\dot{u}})u \end{aligned} \quad (2.47)$$

Este corresponde a:

$$\mathbf{C}(\boldsymbol{\nu}) = \mathbf{C}_{RB}(\boldsymbol{\nu}) + \mathbf{C}_A(\boldsymbol{\nu}) \quad (2.48)$$

Donde análogamente a la matriz de masa, aparecen las matrices asociadas al cuerpo rígido y a la masa agregada.

La matriz de amortiguamiento es:

$$\mathbf{D}(\boldsymbol{\nu}) = \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & d_{23} \\ 0 & d_{32} & d_{33} \end{bmatrix} \quad (2.49)$$

Donde los elementos de la matriz vienen dados por:

$$\begin{aligned} d_{11} &= -X_u - X_{|u|u} |u| - X_{uuu} u^2 \\ d_{22} &= -Y_v - Y_{|v|v} |v| - Y_{|r|v} |r| \\ d_{23} &= -Y_r - Y_{|v|r} |v| - Y_{|r|r} |r| \\ d_{32} &= -N_v - N_{|v|v} |v| - N_{|r|v} |r| \\ d_{33} &= -N_r - N_{|v|r} |v| - N_{|r|r} |r| \end{aligned} \quad (2.50)$$

Esto corresponde a los términos de amortiguamiento, pero con un término extra de tercer orden en surge, agregado por Skjetne.

Parámetros usados, algunos se obtienen directamente a través de las propiedades físicas como la masa, longitud, ancho, otros parámetros son estimados. Lindegaard (2003), obtiene una primera estimación para los parámetros de amortiguamiento para la aplicación de posición dinámica (DP), aunque todos estos son términos lineales. Este modelo considerando amortiguamiento lineal se puede utilizar para trabajos a baja velocidad, para casos de maniobra a mayores velocidades se requiere modelos más completos. A continuación se presentan los parámetros usados en el modelo de referencia obtenido por, Lindegaard (2003)

Parámetros directamente obtenidos:

$$m = 23.8000 \quad I_z = 1.7600 \quad x_g = 0.0460$$

Parámetros, primera estimación:

$$\begin{aligned} X_{\dot{u}} &= -2.0 & Y_{\dot{v}} &= -10.0 & N_{\dot{v}} &= -0.0 \\ & & Y_{\dot{r}} &= -0.0 & N_{\dot{r}} &= -1.0 \end{aligned}$$

Parámetros obtenidos por identificación:

$$\begin{aligned} X_u &= -0.72253 & Y_v &= -0.88965 & N_v &= 0.03130 \\ X_{|u|u} &= -1.32742 & Y_{|v|v} &= -36.47287 & N_{|v|v} &= 3.95645 \\ X_{uuu} &= -5.86643 & & & & \end{aligned}$$

Parámetros obtenidos por estimación adaptativa:

$$\begin{aligned} Y_r &= -7.250 & N_r &= -1.900 \\ Y_{|r|v} &= -0.805 & N_{|r|v} &= 0.130 \\ Y_{|v|r} &= -0.845 & N_{|v|r} &= 0.080 \\ Y_{|r|r} &= -3.450 & N_{|r|r} &= 0.750 \end{aligned}$$

2.3.2. Radio de giro del barco

Uno de los temas centrales cuando se hace planeamiento de trayectorias es el radio de giro. Los sistemas tienen un mínimo radio de giro que pueden alcanzar (La Valle, 2006), teniendo el mínimo radio de giro se puede obtener la máxima curvatura posible.

$$\kappa_{max} = \frac{1}{R} \quad (2.51)$$

Donde: κ_{max} : Curvatura máxima que puede describir la embarcación. R : Radio de giro de barco.

La restricción de curvatura es un tema central, cuando se diseñan rutas mediante técnicas que usan círculos como referencias al momento de hacer los giros. Se tiene como referencia que el CyberShip II alcanza un radio de giro mínimo de aproximadamente 2.5 m, por tanto tiene una curvatura máxima de $0.4m^{-1}$.

2.3.3. Pruebas con el modelo de referencia

Para probar el modelo matemático realizaremos algunas pruebas de simulación, considerando entradas que nos den resultados lógicos, por ejemplo si le

damos solo fuerza en *surge* y torque en *yaw* sin fuerza en la dirección *sway* deberíamos conseguir, una trayectoria circular. Luego de realizar pruebas para obtener los torques aproximados que generen esta trayectoria circular los resultados se muestran en la Figura 2.10 y los torques se muestran en la Figura 2.9.

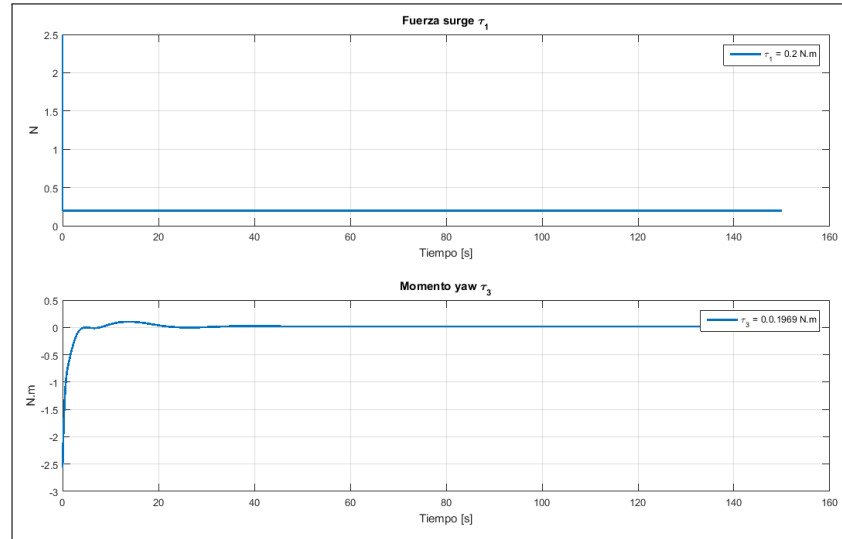


Figura 2.9: Fuerza y torque aplicado, $\tau_1=0.2\text{N}$ y $\tau_3=0.01\text{N.m}$

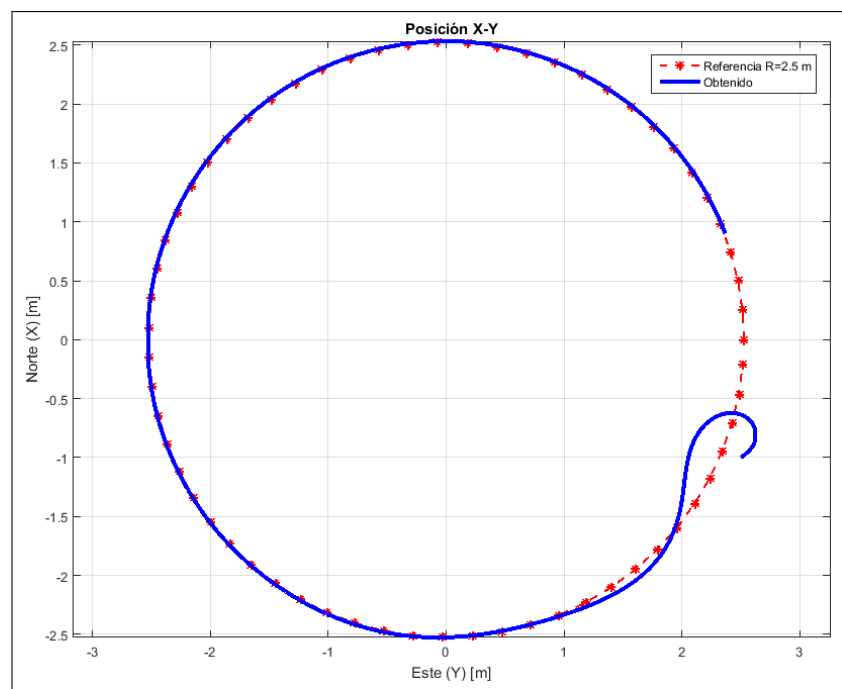


Figura 2.10: Respuesta del sistema a entradas (Línea azul).

Como era lógico al usar la fuerza y el torque mencionadas líneas arriba se ha conseguido una trayectoria circular, de tal manera que se puede, de alguna manera, comprobar que el modelo está correcto, ya que se ha obtenido un resultado lógico.

2.4. Conclusiones preliminares

- Se obtuvo un modelo matemático de tres grados de libertad considerando la dinámica del cuerpo rígido y fuerzas que interactúan con el barco, donde se puede observar que los parámetros mecánicos como longitud, masa, distancia al origen de referencia son determinantes para el comportamiento dinámico.
- Se pudo ver que existen fuerzas que no se pueden considerar para el modelado, así que sólo se consideran las que tienen mayor influencia en la dinámica del sistema.
- Al buscar implementar un barco autónomo a fin de probar los algoritmos, se investigó y determinó que el CyberShip II, es el prototipo más cercano al deseado, por lo cual se tomará sus parámetros dinámicos como referencia. Este prototipo es ampliamente usado en muchas investigaciones.
- Se comprobó que el modelo de referencia elegido trabaja correctamente, al obtener un resultado lógico frente a entradas conocidas de fuerza $\tau_1 = 0.2N$ y torque $\tau_1 = 0.01N.m$.
- La curvatura máxima $\kappa_{max} = 0.4m^{-1}$ será tomada en cuenta en la generación de trayectoria, a fin de que el barco no sea forzado a realizar trayectorias imposibles.

Capítulo 3

Desarrollo de un sistema de planeamiento y seguimiento de ruta de un barco de dos hélices en espacio confinado

3.1. Introducción

El propósito del presente capítulo consiste en realizar el diseño mediante simulación de un sistema de planeamiento, guía y seguimiento de trayectoria para una embarcación marina subactuada con dos hélices a partir del modelo matemático de barco de 3 grados de libertad, obtenido en el capítulo anterior.

Este capítulo se divide en 3 partes, el planeamiento de trayectoria, estrategia de control y síntesis de planeamiento y seguimiento de trayectoria para embarcaciones marinas en espacios confinados. En la primera parte se da a conocer la estrategia utilizada para realizar el planeamiento de trayectoria partiendo de un espacio determinado caracterizado por su geometría y obstáculos que se puedan presentar. Para esta tarea se hace uso del diagrama de Voronoi y con un algoritmo de filtrado se consigue las posibles trayectorias a seguir, luego con estos puntos y posibles trayectorias obtenidas aplicamos algoritmos de optimización de trayectorias a fin de obtener la más corta, aquí aplicamos el algoritmo de Dijkstra, a partir del cual se obtienen puntos que unen la trayectoria más corta. Finalmente aplicamos un algoritmo de tal manera que filtramos los puntos que no sean necesarios para evitar los cambios de dirección innecesarios, también consideramos una tolerancia de tal manera que la embarcación pueda circular por lugares estrechos, con cierta holgura.

En la segunda parte, se presenta la teoría de control “backstepping”, que fue

desarrollada por Petar Kokotovic en los 1990, y es aplicada para sistemas dinámicos no lineales, se comenta sus ventajas respecto a la técnica de “*feedback linearization*”. Se muestra el desarrollo de la técnica aplicada al control de embarcaciones marinas, considerando la dinámica del barco. Se desarrolla el sistema de guía del barco para lograr el seguimiento de trayectoria, con las consideraciones especiales que implican controlar un sistema subactuado.

En la tercera parte, se desarrolla la simulación de todo el sistema, partimos del espacio de navegación definido por el mapa que posteriormente simplificamos mediante puntos, luego aplicamos todo lo desarrollado en la parte de planeamiento para finalmente obtener puntos o trayectorias de acuerdo a la técnica de guía que se requiera usar. Se presenta el modelo matemático usado para las simulaciones, con los parámetros del modelo conocidos se desarrolla el controlador y se presentan las simulaciones respectivas.

3.2. Planeamiento de trayectoria

Para realizar el control o guía de una embarcación es necesario tener como entrada una descripción matemática del movimiento deseado. El movimiento deseado puede ser obtenido de diferentes maneras, puede ser calculado automáticamente basado en tareas de alto nivel, o podrían ser descritas manualmente con mayores detalles. Estos, podrían ser líneas que deberían ser seguidas o puntos de interés que deben ser alcanzados a fin de evitar obstáculos o tener la trayectoria más óptima posible. En esta sección se presentarán las herramientas matemáticas usadas a fin de obtener el planeamiento de trayectoria deseado.

3.2.1. Terminología

Como se describe en LaValle (2006), el planeamiento es inherentemente interdisciplinario y comprende los campos de la robótica, ciencias computacionales, teoría de control y matemáticas entre otras. Para nuestra aplicación que viene a ser referida a control de embarcaciones, Fossen (2011) define: La configuración del espacio n -dimensional es el espacio de posibles posiciones y orientaciones que una embarcación puede tener, sujeta posiblemente a restricciones externas. En consecuencia la configuración puede ser descrita:

$$n_{3DOF} = \begin{bmatrix} N \\ E \\ \psi \end{bmatrix} \quad n_{6DOF} = \begin{bmatrix} N \\ E \\ D \\ \phi \\ \theta \\ \psi \end{bmatrix} \quad (3.1)$$

Donde: n_{3DOF} : espacio de 3 dimensiones. n_{6DOF} : espacio de 6 dimensiones.

Esto se da cuando operamos en el plano horizontal, 3 grados de libertad ó cuando operamos en los 6 grados de libertad, Ecuación (3.1).

La mayoría de las embarcaciones marinas son subactuadas y sus componentes de configuración, N , E , D , ϕ , θ y ψ no puede ser variados independientemente. La terminología ajustada para aplicaciones marinas es propuesta. La Figura 3.1 resume la terminología y relación entre pasos a seguir hasta obtener la trayectoria deseada.

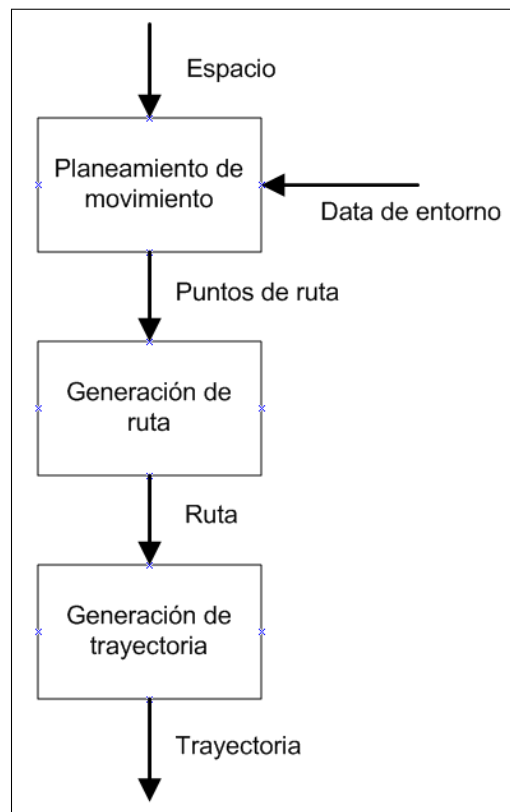


Figura 3.1: Pasos para la generación de trayectoria.

Planeamiento de movimiento, Encuentra una secuencia de posiciones permitidas, de aquí se obtienen waypoints. En el caso de barcos subactuados, es necesario dejar algún grado de libertad libre debido a que no se pueden controlar todos los grados de libertad. Para una embarcación subactuada por ejemplo el ángulo

de cabeceo ψ , no puede ser controlado en un eje fijo y el objetivo de control es comúnmente limitado a la posición (N,E) .

Generación de ruta, como resultado nos da una ruta. Esta ruta es un conjunto continuo de posiciones en el espacio geométrico.

Generación de trayectoria, convierte la ruta a una función del tiempo. Según Fossen (2011), Una trayectoria describe el movimiento de un objeto moviéndose a través del espacio como una función del tiempo.

La diferencia entre planeamiento y generación, es separada a propósito para enfatizar que ruta y trayectoria mayormente se consiguen implícitamente a partir de los resultados del planeamiento de movimiento y la aplicación de restricciones. La principal decisión elaborada es hecha en el planeamiento del movimiento.

3.2.2. Planeamiento de movimiento

Frecuentemente como salida del planeamiento de movimiento se obtienen puntos. Estos son usualmente puntos en el espacio físico. Estos puntos implícitamente describen líneas rectas que el barco debe seguir, por ejemplo hoja de ruta para dejar el puerto, en estos casos la embarcación no pasa necesariamente por los puntos obtenidos. En algunos casos no es necesario algoritmos ni uso de las matemáticas, esto se da cuando un operador humano define los puntos de ruta a seguir, desde el inicio hasta el final, considerando que no haya riesgo de colisión. Para lograr el planeamiento de movimiento a partir de datos de mapas, se debe realizar algunas simplificaciones a polígonos como sugiere LaValle (2006) y como se puede observar en la Figura 3.2. Depende del grado de simplificación del mapa, que la creación de obstáculos poligonales pueda perder mucha data.



Figura 3.2: Izquierda mapa y a la derecha su simplificación poligonal.

A continuación se presenta el método de mapeo usado en esta tesis para la obtención de los posibles puntos de ruta.

3.2.2.1. Diagrama de Voronoi

El diagrama de Voronoi es llamado así debido a su creador el matemático Voronoi en 1908. Básicamente este método puede verse con la partición de una región determinada en base a unos conjuntos de puntos, su visualización en su forma simple se ve en la Figura 3.3, donde se ha aplicado para un número de puntos aleatorios.

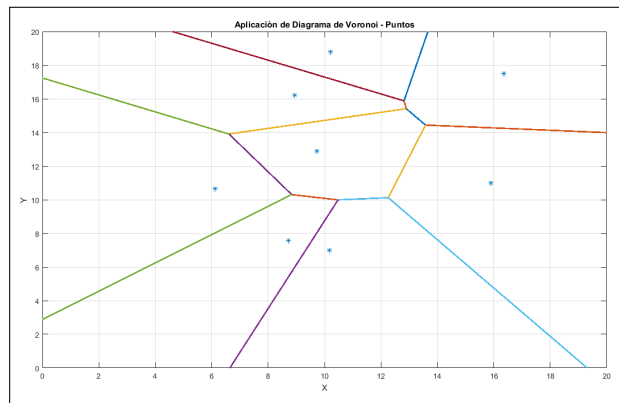


Figura 3.3: Diagrama de Voronoi para puntos aleatorios.

El diagrama de Voronoi se basa en un conjunto finito de puntos contenidos en un espacio de trabajo que definimos, de tal manera que la distancia del diagrama hacia los puntos es la mínima, respecto de otro punto.

$$d(x, p) = \inf \{ d(x, p) \mid p \in \mathcal{P} \} \quad (3.2)$$

La idea del diagrama de Voronoi es expandida hacia obstáculos poligonales: cada obstáculo tiene un subespacio perteneciente de todos los puntos que están más cerca de este obstáculo. Una conectividad gráfica es creada, donde los ejes son los bordes de estos subespacios. Los vértices son la intercepción de dos o más bordes. Una ruta a través del gráfico es elegida, como se puede ver en la Figura 3.4.

3.2.2.2. Algoritmo de Dijkstra

Este algoritmo es muy conocido por sus diversas aplicaciones en el campo de la busca de la ruta más corta en los métodos gráficos. Este algoritmo fue desarrollado en 1959. Este método asigna a cada nodo en la gráfica un valor de distancia, inicialmente es seteada cada distancia a infinito y el nodo de inicio es seteado a cero. El algoritmo considera todos los nodos excepto el inicial en una condición de no-vistado, y configura el nodo de inicio como el nodo actual. El algoritmo consiste en calcular la distancia tentativa, entre el nodo inicial y demás nodos no-visitados de la vecindad, y compararlo con el valor previo y actualizar

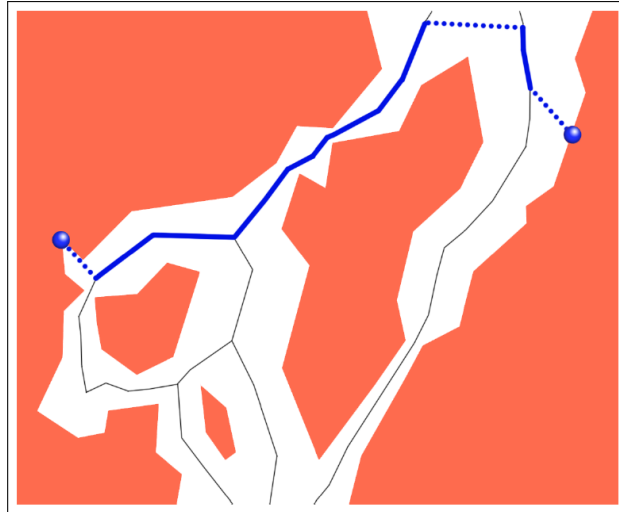


Figura 3.4: Diagrama de Voronoi aplicado.

si el valor calculado es menor y el nodo que genere este valor menor debe ser tomado como el nuevo nodo actual. Esto se va desarrollando hasta que el nodo de destino (Punto final) es visitado o no existe ninguna conexión entre los nodos visitados y no-visitados.

En la Figura 3.5, vemos que tenemos un punto inicial a y un punto objetivo b , que están unidos a través de líneas que simbolizan el peso o distancia entre cada punto de ruta, el algoritmo de Dijkstra inicia una búsqueda de la menor ruta según el procedimiento descrito líneas arriba, al iniciar tiene 3 opciones las cuales se prueban y continúan hasta obtener la mejor ruta, en este caso vendría dada por los nodos 1-3-6-5.

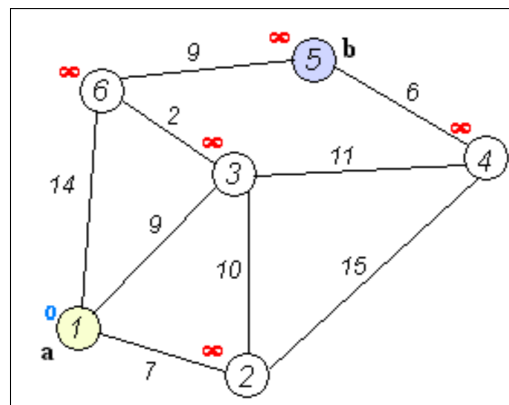


Figura 3.5: Algoritmo de Dijkstra.

Una vez obtenidos los puntos, gracias al diagrama de Voronoi, se procede a utilizar el algoritmo de Dijkstra para poder obtener los puntos que hagan menor el recorrido desde un punto hacia otro evitando los obstáculos.

3.2.2.3. Obstáculos

La representación de obstáculos en este trabajo será hecha por polígonos en dos dimensiones, siendo fácilmente extensible esto hacia obstáculos 3D. Naturalmente lo que se busca es evitar los obstáculos para llegar del punto inicial hacia el punto final. Una de las ventajas de trabajar con polígonos es que son fáciles de incrementar en tamaño y se puede fácilmente asignar una distancia de tolerancia. (Ver Figura 3.6b)

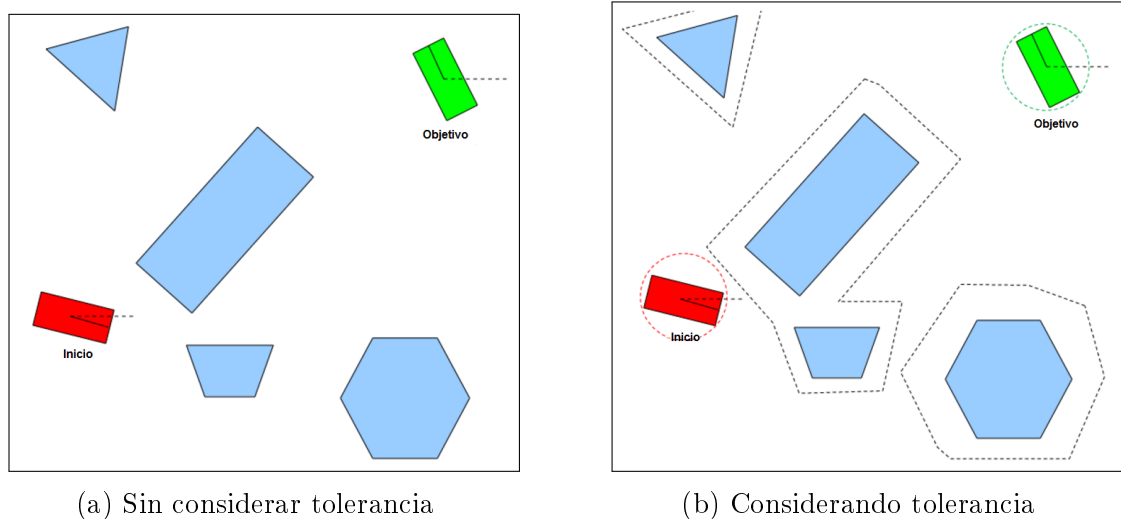


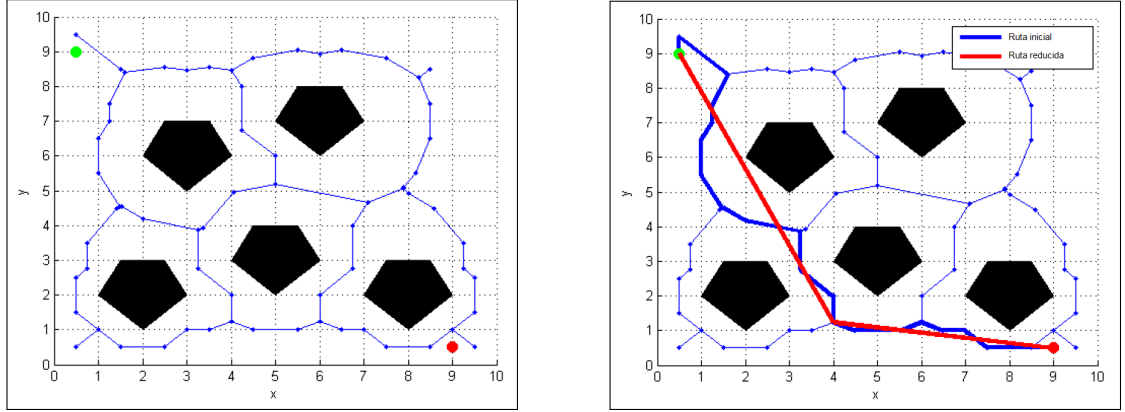
Figura 3.6: Representación de obstáculos gráficamente y consideración de tolerancia

Al tener los obstáculos como figuras geométricas se pueden aprovechar algunas funciones del software de simulación MATLAB. Para nuestro caso hemos usado las funciones `polyxpoly`, la cual comprueba si existen intersecciones entre polígonos y la función `inpolygon`, que determina que valores de los puntos están dentro y fuera o sobre un eje de una región poligonal.

3.2.2.4. Eliminación de puntos no necesarios

Luego de tener la ruta con menor recorrido por la aplicación del diagrama Voronoi y el algoritmo de Dijkstra, a se procede a realizar la implementación de un algoritmo que consiste simplemente en remover los puntos que causan cambio de ruta innecesaria y que esta no implique la colisión, también se toma en cuenta un valor de tolerancia de distancia respecto a los obstáculos. En la Figura 3.7a, se muestra un entorno al cual se le aplicó el diagrama de Voronoi, teniendo como partida el punto verde y como llegada el punto rojo, a partir de este diagrama se aplicará el algoritmo de Dijkstra y se obtendrá la mejor ruta considerando los punto del dados por el diagrama de Voronoi, luego se aplicará el algoritmo de eliminación de puntos innecesarios, para obtener una mejor ruta y

evitar los cambios de orientación (ángulo del cabeceo) innecesarios, este algoritmo se implementa en MATLAB, ver Anexo B



(a) Diagrama de Voronoi con obstáculos

(b) Ruta inicial y ruta reducida libre de obstáculos

Figura 3.7: Eliminacion de puntos no necesarios

Finalmente como resultado del planeamiento del movimiento tenemos un conjunto de puntos de ruta que nos garantizan una trayectoria libre de obstáculos y con el menor cambio orientación.

3.2.3. Generación de ruta

Como resultado de la generación de ruta tendremos una ruta con consideraciones geométricas, estas consideraciones deben estar relacionadas con la dinámica del barco. La entrada para obtener la ruta es el conjunto de puntos obtenidos en el planeamiento de movimiento.

Debido a la complejidad de la dinámica del barco, la ruta solamente respeta la dinámica del barco hasta cierto punto. Esto puede ser trasladado en consideraciones geométricas y otros criterios de evaluación.

Antes de seguir debemos definir de una forma precisa a que se refiere con ruta.

Una ruta es un conjunto que puede ser expresado de la siguiente manera:

$$\mathcal{P} \triangleq \{\mathbf{p} \in \mathbb{R}^n \mid \mathbf{p} = \mathbf{p}_p(\varpi) \forall \varpi \in X\} \quad (3.3)$$

Donde ϖ es la variable escalar de parametrización y $p_p(\varpi)$ es la posición de un punto correspondiente sobre la ruta. El intervalo X define el dominio de $p_p(\varpi)$. Para naves de superficie, al menos 2 dimensiones ($n=2$) son necesarias.

$$\mathbf{p}_p(\varpi) = \begin{bmatrix} N_p(\varpi) \\ E_p(\varpi) \end{bmatrix} \in \mathbb{R}^2 \quad (3.4)$$

La notación dada es hecha en Skjetne et al. (2001), Skjetne (2005), Breivik (2003) y Fossen (2011). Otras notaciones son $P_d(q)$ usadas en Breivik et al. (2004).

Para muchas formas de curva, las ecuaciones paramétricas aproximadas pueden ser derivadas, por ejemplo por la serie de Taylor, Fourier o por interpolación polinomial. Sin embargo, una curva simple no puede representar formas complejas sin alta complejidad, como si dice en Haugen and Breivik (2010).

Haugen and Breivik (2010), propone construir un ruta a partir de una combinación de sub-rutas. La ruta puede ser expresada.

$$\mathcal{P} = \bigcup_{i=1}^m \mathcal{P}_i \quad (3.5)$$

Donde las sub-rutas serán definidas:

$$\mathcal{P}_i \triangleq \{\mathbf{p}_i \in \mathbb{R}^n \mid \mathbf{p}_i = \mathbf{p}_{p,i}(\varpi) \forall \varpi \in X_i\} \quad (3.6)$$

Donde \mathcal{P}_i , es la ecuación paramétrica para la curva correspondiente a la sub-ruta i . Un mínimo requerimiento es que las trayectorias subsecuentes sean conectadas. Es importante notar que si una ruta se parametiza respecto a una variable w esta no necesita tener algún significado físico. En general, estas no denotan la distancia a lo largo de la ruta ni el tiempo el cual se llega a un punto $\mathbf{p}_p(\varpi)$.

Antes de presentar algunos de los algoritmos de creación de trayectorias, es bueno revisar cuales son las consideraciones que se deben tener en cuenta al momento de crear estas curvas.

Estos criterios son explicados en Dahl (2013), entre los cuales tenemos los siguientes:

- Suavidad de la ruta paramétrica: Descripción de la velocidad a la cual la curva es trazada.
- Suavidad geométrica: Descripción del comportamiento geométrico de la ruta.
- Longitud de la ruta: La longitud de la ruta a ser recorrida por el barco.
- Tolerancia de ruta: Término conocido como la desviación respecto del deseado.
- Trazabilidad: el grado en el cual la ruta puede ser cambiada localmente.

- Complejidad del algoritmo: Consideraciones sobre los recursos necesarios para correr el algoritmo y su arquitectura lógica.

3.2.3.1. Generación de ruta usando polinomio cúbico de Hermite

La construcción de rutas usando polinomios no es un tema reciente, y se han considerado para crear formas geométricas primitivas. Sin embargo, es también posible usar estos polinomios para describir rutas complejas mediante el uso de funciones, Lekkas et al. (2014a). El número de sub-rutas es reducido a $n - 1$, siendo n el número de puntos que definen la trayectoria.

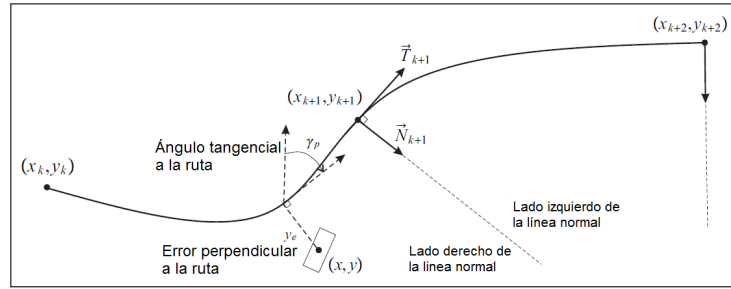


Figura 3.8: Curvas de generación de ruta entre puntos, (Lekkas, 2014).

La función de una curva polinomial debe soportar una inflexión. El polinomio de menor grado que cumple con estas condiciones es un polinomio cúbico, además la curva puede ser parametrizada como sigue:

$$\mathbf{P}_{\text{cubica}} = \begin{bmatrix} a_N \varpi^3 + b_N \varpi^2 + c_N \varpi + d_N \\ a_E \varpi^3 + b_E \varpi^2 + c_E \varpi + d_E \end{bmatrix} \quad (3.7)$$

Donde a , b , c y d son los coeficientes del polinomio. El problema de diseño de sub-ruta se convierte en la obtención de estos coeficientes.

Los coeficientes son determinados de tal manera que satisfacen las condiciones de final de punto, y las condiciones de finales de derivadas (tangentes). Por tanto los coeficientes del polinomio cubico de Hermite son:

$$\begin{bmatrix} a_i^T \\ b_i^T \\ c_i^T \\ d_i^T \end{bmatrix} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{wpt}_i^T \\ \mathbf{wpt}_{i+1}^T \\ \mathbf{t}_i^T \\ \mathbf{t}_{i+1}^T \end{bmatrix} \quad (3.8)$$

En la Figura 3.9, se muestran los resultados obtenidos cuando se aplica la técnica de polinomio cúbico de hermite, las dos primeras curvas nos muestran las posiciones X y Y obtenidas a partir de los puntos dados, la tercera curva muestra el ángulo de curso, mientras la última gráfica corresponde a la curvatura se observa ligeros saltos que nos dicen que la curva no es totalmente suave.

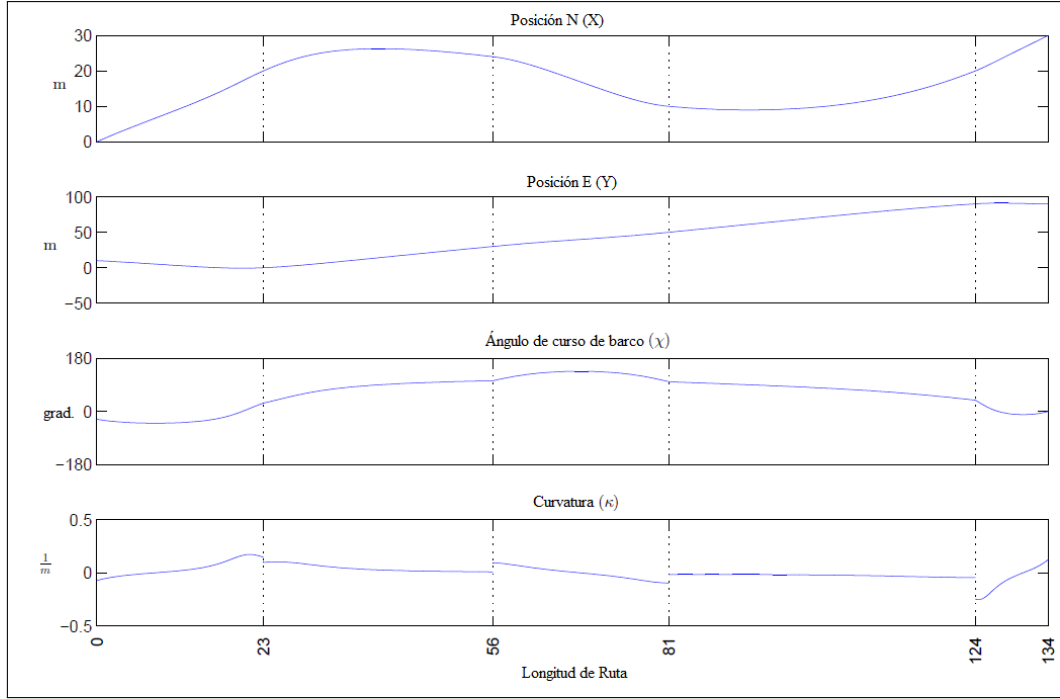


Figura 3.9: Propiedades geométricas de la ruta generada por polinomio cúbico de Hermite.

3.2.3.2. Generación de ruta usando Espiral de Fermat

La espiral de Fermat presenta una curvatura suave que va desde cero hasta un valor deseado. Esta puede usarse para suavizar y reflejar una curva, como una opción alternativa a otras como las polinómicas ó Clotoide, Dahl (2013).

Para realizar la curva mediante el espiral de Fermat, se debe trazar 2 gráficas donde una es el reflejo de la otra. Para la primera mitad de la curva, se elige la siguiente parametrización:

$$\mathbf{P}_{\text{Fermat}} = \begin{bmatrix} N_0 + \kappa \sqrt{\varpi \theta_{\text{final}}} \cos(\rho \varpi \theta_{\text{final}} + \chi_0) \\ E_0 + \kappa \sqrt{\varpi \theta_{\text{final}}} \sin(\rho \varpi \theta_{\text{final}} + \chi_0) \end{bmatrix} \quad (3.9)$$

Donde $\mathbf{p}_0 = [N_0, E_0]^T$, es el punto de inicio para la trayectoria, κ es una escala constante, θ_{final} corresponde al ángulo polar correspondiente al final de la curva, ρ es la dirección de rotación y χ_0 es el ángulo de curso inicial.

La curva simétrica tiene la siguiente parametrización:

$$\overline{\mathbf{P}}_{\text{Fermat}} = \begin{bmatrix} N_{\text{final}} + \kappa \sqrt{\theta_{\text{final}} - \varpi \theta_{\text{final}}} \cos(\rho(\varpi \theta_{\text{final}} - \theta_{\text{final}}) + \chi_0) \\ E_{\text{final}} + \kappa \sqrt{\theta_{\text{final}} - \varpi \theta_{\text{final}}} \sin(\rho(\varpi \theta_{\text{final}} - \theta_{\text{final}}) + \chi_0) \end{bmatrix} \quad (3.10)$$

Donde $P_{final} = [N_{final}, E_{final}]^T$, es el punto final.

Se debe tener en cuenta que la espiral de Fermat, nos sirve para unir las partes donde hay cambio de ruta y las demas deben ser unidas por lineas rectas. Tomando como base esta parametrización evaluamos los puntos dados y obtenemos el resultado que se muestra en la Figura 3.10, de igual manera que en los resultados mostrados en la Sección 3.2.3.1. Las dos primeras curvas corresponden a la posición X-Y, donde se ve claramente la parte definida por lineas y curvas de Fermat, podemos ver que se presenta curvas mas suaves, respecto a la gráfica de cambios de curso no tienen cambios abruptos, mientras la gráfica de curvatura muestra claramente un mejor comportamiento respecto a la curvatura dada en la Sección 3.2.3.1.

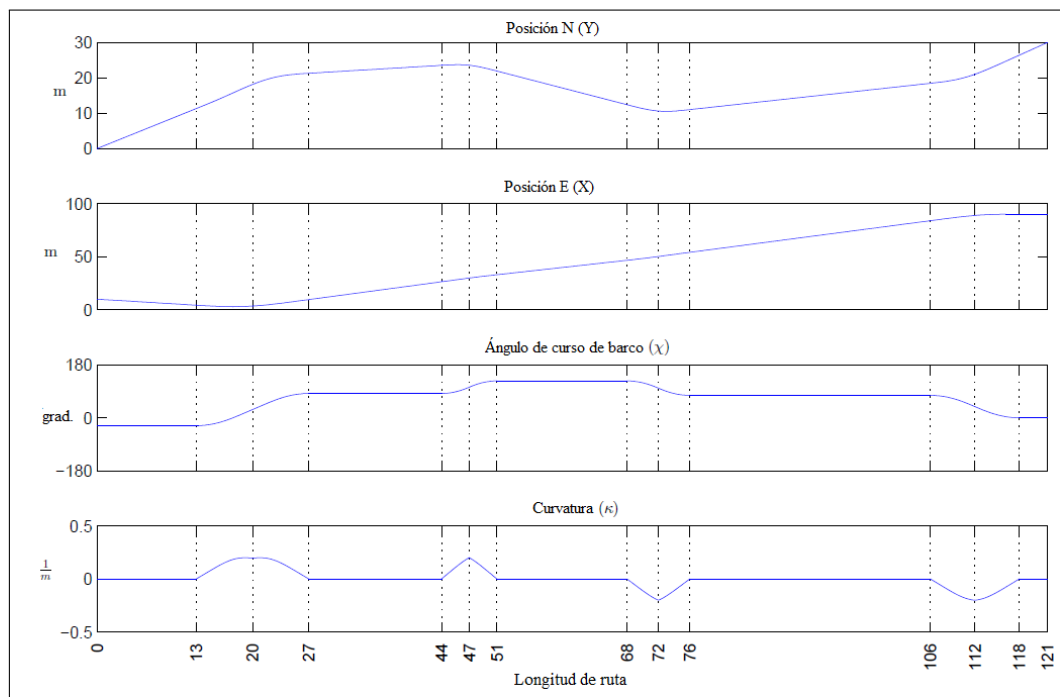


Figura 3.10: Espiral de Fermat y sus características geométricas.

Aquí solo se han presentado dos métodos que se pueden usar para la generación de rutas, se debe tener en cuenta que existen más técnicas como los que usan rectas y círculos, spline, espirales de clothoid, funciones sigma, etc.

3.3. Sistema de guiado

En esta sección se describe los dos métodos para el diseño de sistema de guiado de embarcaciones marinas, el sistema de guiado es definido por Shneydor (1998) como “El proceso de guiado de la ruta de un objeto a través de un punto dado, que generalmente puede estar moviéndose”. De acuerdo a Fossen (2011), el sistema de guiado en su forma simple, en lazo abierto son usados para generar

trayectorias de referencia variantes en el tiempo ó, alternativamente, rutas para hacer seguimiento de rutas invariantes en el tiempo. El sistema de control de movimiento del barco trabajar de forma muy cercana con el sistema de guiado a fin de que el móvil cumpla con lo deseado, la estructura se puede ver en la Figura 3.11.

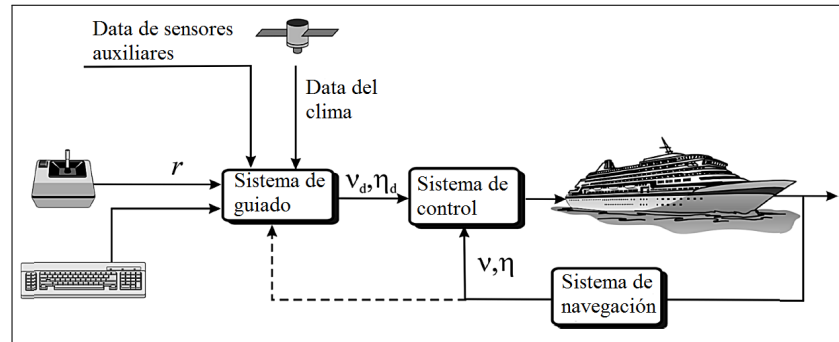


Figura 3.11: Sistema de guiado en lazo cerrado (Líneas punteadas) los estados se realimentan mientras en lazo abierto solo usan señales de sensores y referencias de entrada Fossen (2011)

Para cualquier barco o submarino, el sistema de guiado y control usualmente constan de dos subsistemas:

- Sistema de control de posición.
- Sistema de control de seguimiento de ruta.

De forma sencilla se puede decir que, el sistema control de posición es un autopiloto de curso, donde los ángulos de roll y pitch son regulados a cero o dejados sin control. La principal función de sistema de control de curso es mantener al barco dentro de la ruta deseada sin importar aspectos dinámicos como velocidad o aceleración. Mientras que la tarea del sistema de control de seguimiento, es mantener a la embarcación sobre la ruta preestablecida a una determinada velocidad.

Una ley de control es necesaria para seguir una ruta. De acuerdo a lo estudiado en secciones anteriores, la ruta puede o no ser estrictamente geométrica o puede ser una trayectoria parametrizada por el tiempo. Equivalentemente la ley de control variará de acuerdo al objetivo.

El seguimiento de ruta demanda solamente que el controlador asegure que la embarcación se mantenga sobre su ruta geométrica. No existen requerimientos sobre cuando el barco deba estar en una posición dada. Asimismo, la velocidad a lo largo de la ruta podría variar independientemente, por ejemplo el barco podría ser controlado a una velocidad a lo largo de la ruta “surge” acorde a las necesidades de maniobra.

A continuación se describirá dos métodos de guiado, el primero será basado en tres puntos de ruta a partir del algoritmo de línea de visión, teniendo en cuenta que para cambiar y avanzar al siguiente punto de ruta se utiliza el criterio de círculo de aceptación, el segundo será basado en la generación de ruta usando la técnica de líneas rectas unidas a través de la espiral de Fermat, la cual nos facilita la continuidad hasta la segunda derivada, para de esta manera poder tener rutas y señales de control mas suaves.

3.3.1. Línea de visión - Basado en 3 puntos

Es la aproximación menos precisa para seguir una ruta implícita definida por un conjunto de puntos de ruta, el curso del barco es controlado a través de un punto en el tiempo. Se aplica un criterio de aceptación de tal forma que se pueda cambiar de referencia y avanzar hacia el siguiente punto de ruta, una representación gráfica de este método se puede ver en la Figura 3.12.

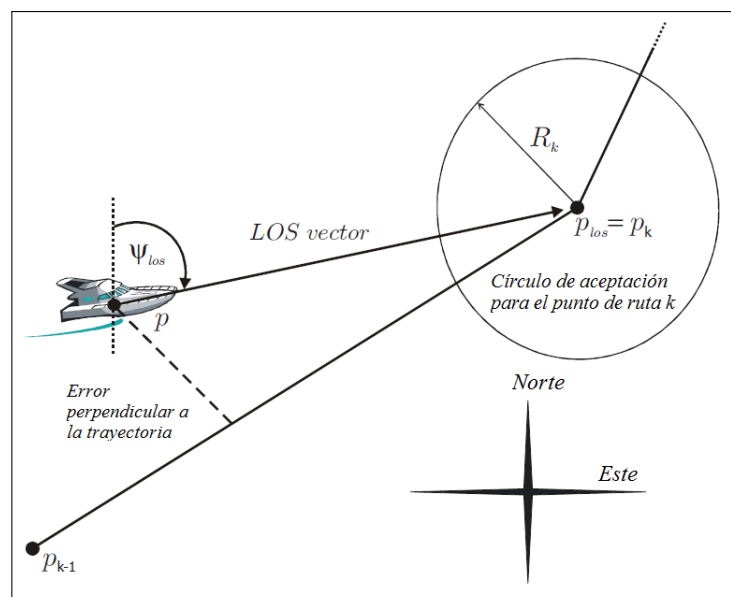


Figura 3.12: Representación de algoritmo, con círculo de aceptación (Breivik, 2003)

El guiado por principio de línea de visión es un forma intuitiva de entender el comportamiento del barco y de la acción del piloto. El problema principal de este método viene dado por si el barco es capaz de mantener su curso alineado con el ángulo de línea de visión ψ_{los} , convergiendo a la posición deseada y alcanzada. Toda estas propiedades se hacen mucho mas interesantes cuando trabajamos controlando barcos subactuados, debido a que se debe reducir las variables de salida, una posición con 3 GDL (Posición XY y curso ψ) partiendo de 2 GDL (curso ψ y velocidad surge u).

El concepto de ángulo LOS (ψ_{los}), puede ser explicado utilizando los términos vector LOS y posición LOS. El vector LOS es un vector que apunta desde la posición actual hacia la posición deseada a lo largo de la ruta deseada, también llamada posición LOS ($p_{LOS} = [x_{los}, y_{los}]^T$), se puede verificar los términos en la Figura 3.12. La obtención de la posición se puede hacer de muchas maneras.

3.3.1.1. Cálculo de ángulo LOS ψ_{los} para embarcaciones marinas

Considerando que la posición actual es $p = [x, y]^T$ y la posición LOS viene dada por $p_{LOS} = [x_{los}, y_{los}]^T$, esto sugiere que el ángulo LOS deseado deberá ser seleccionado como:

$$\psi_{los} = \text{atan2}(y_{los} - y, x_{los} - x) \quad (3.11)$$

Donde $[x, y]^T$ es la posición obtenida a través de satélites de navegación o GPS. Al usar la función atan2 , aseguramos que el dominio de la función tangente varíe entre:

$$-\pi \leq \text{atan2}(x, y) \leq \pi \quad (3.12)$$

Podemos ver que la Ecuación (3.11) se aplica independientemente del cálculo de la posición LOS (p_{los}). De acuerdo a lo expuesto por Breivik (2003) existen tres diferentes métodos para obtener p_{los} .

3.3.1.2. Cálculo de posición LOS p_{los} para embarcaciones marinas

En esta sección, se revisan los tres enfoques dados por Breivik (2003), de forma breve.

1. La ruta a seguir consiste únicamente de puntos de ruta en una tabla, la posición p_{los} , es simplemente el punto de ruta actual denotado por p_k . Por tanto, no se realiza cálculo alguno.
2. De igual manera, solo se cuenta con una tabla de puntos de ruta, con la diferencia que en este método la posición p_{los} se ubica a lo largo de una recta que une dos puntos de la tabla denotados por p_{k-1} y p_k . La Figura 3.13 muestra este principio. El principio se basa en considerar una circunferencia de radio n veces la longitud del barco (nL_{pp}) alrededor de la posición. Este círculo interceptará a la línea que une dos puntos de ruta, se selecciona el punto que está más cerca al siguiente punto de ruta. Se requieren 2 ecuaciones que se deducen fácilmente de la Figura 3.13. A continuación se presentan las ecuaciones.

$$\frac{y_{los} - y_{k-1}}{x_{los} - x_{k-1}} = \frac{y_k - y_{k-1}}{x_k - x_{k-1}} = \tan(\alpha_{k-1}) \quad (3.13)$$

$$(x_{los} - x)^2 + (y_{los} - y)^2 = (nL_{pp})^2 \quad (3.14)$$

Se debe tener en cuenta que el sub-índice k representa el actual punto de ruta mientras $k - 1$ representa el punto de ruta anterior. Vemos que la primera Ecuación (3.13) representa la pendiente constante que se da entre dos puntos de ruta, mientras la Ecuación (3.14)

3. La ruta deseada es una ruta geométrica parametrizada, donde la variable de parametrización es un escalar denotado por θ , la ruta deseada es dada por $p_d(\theta)$, la representación de este enfoque se puede ver Figura 3.14.

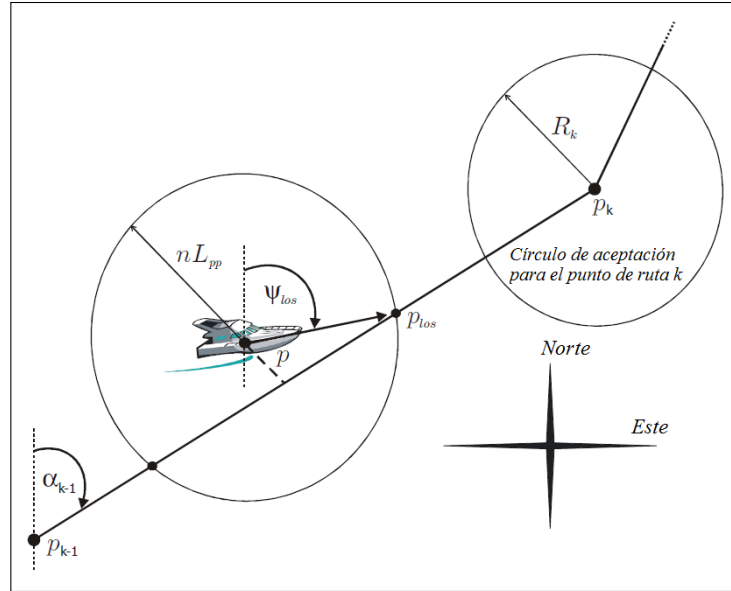


Figura 3.13: Ángulo LOS obtenido a partir de resolución de dos ecuaciones. (Breivik, 2003)

Para los dos primeros enfoques, es necesario tener un criterio de cambio de punto de ruta. Un criterio para la selección del siguiente punto de ruta $p_{k+1} = [x_{k+1}, y_{k+1}]^T$ viene dado por considerar un círculo de aceptación, en la que en algún instante se satisfaga la siguiente ecuación.

$$(x_k - x)^2 + (y_k - y)^2 \leq R_k^2 \quad (3.15)$$

Este radio de aceptación puede ser único para todos los puntos de ruta o puede variar para cada punto, de acuerdo a las restricciones geométricas necesarias.

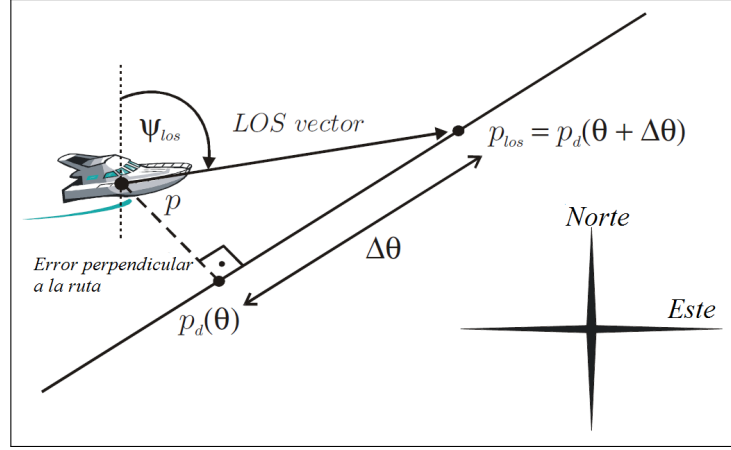


Figura 3.14: Ángulo LOS usando la teoría de maniobra. (Breivik, 2003)

En esta tesis se aplica la técnica basada en el segundo enfoque, por lo cual se hace necesario desarrollar las ecuaciones que nos permitan obtener los valores desconocidos $p_{los} = [x_{los}, y_{los}]^T$, por lo cual las Ecuaciones (3.13) y (3.14). Notar que, la diferencia entre la posición x e y de dos puntos de ruta viene dada por $\Delta x = x_{k+1} - x_k$ y $\Delta y := y_{k+1} - y_k$, respectivamente. Las ecuaciones son resueltas de forma analítica considerando como primer caso $\Delta x > 0$, y como segundo caso $\Delta x = 0$.

Caso 1: $\Delta x > 0$

De la Ecuación (3.13) tenemos que:

$$y_{los} = \left(\frac{\Delta x}{\Delta y} \right) (x_{los} - x_k) + y_k \quad (3.16)$$

Para evitar tantas variables, hacemos los siguientes reemplazos:

$$d = \left(\frac{\Delta y}{\Delta x} \right) \quad e = x_{k-1} \quad f = y_{k-1}$$

Insertamos esto en la parte izquierda de la Ecuación (3.14), obtenemos:

$$(x_{los} - x)^2 + (y_{los} - y)^2 = x_{los}^2 - 2xx_{los} + x^2 + y_{los}^2 - 2yy_{los} + y^2 \quad (3.17)$$

Primero se considera el término y_{los}^2 , y se inserta en la Ecuación (3.16), y

tenemos:

$$\begin{aligned}
 y_{los}^2 &= \left(\left(\frac{\Delta y}{\Delta x} \right) (x_{los} - x_{k-1}) + y_{k-1} \right)^2 \\
 &= (dx_{los} + (f - de))^2 \\
 &= (dx_{los} + g)^2 \\
 &= d^2 x_{los}^2 + 2dgx_{los} + g^2
 \end{aligned} \tag{3.18}$$

Donde la siguiente abreviación ha sido usada.

$$g = f - de = y_{k-1} - \left(\frac{\Delta y}{\Delta x} \right) x_{k-1}$$

Se considera el termino $2yy_{los}$, del cual obtenemos:

$$2yy_{los} = 2y(dx_{los} + g) = 2dyx_{los} + 2gy \tag{3.19}$$

De aquí utilizando las Ecuaciones (3.18) y (3.19) en (3.17), obtenemos:

$$(1 + d^2)x_{los}^2 + 2(dg - dy - x)x_{los} + (x^2 + y^2 + g^2 - (nL_{pp})^2 - 2gy) = 0 \tag{3.20}$$

Que es una ecuación general de segundo orden, que fácilmente puede ser resuelta. A fin de hacer mas clara la ecuación, realizamos algunos cambios de variables convenientemente.

$$\begin{aligned}
 a &= 1 + d^2 \\
 b &= 2(dg + dy - x) \\
 c &= x^2 + y^2 + g^2 - (nL_{pp})^2 - 2gy
 \end{aligned}$$

Siendo la solución general a esta ecuación, x_{los} la siguiente:

$$x_{los} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{3.21}$$

Para elegir cual de las posibles soluciones tomar, se usa el siguiente criterio.

- Si $\Delta x > 0$ entonces $\frac{-b + \sqrt{b^2 - 4ac}}{2a}$
- Si $\Delta x < 0$ entonces $\frac{-b - \sqrt{b^2 - 4ac}}{2a}$

Una vez obtenido x_{los} , es fácil obtener y_{los} , a partir de la Ecuación (3.16). Se debe tener en cuenta que cuando $\Delta y = 0$, $y_{los} = y_{k-1} (= y_k)$.

Caso 2: $\Delta x = 0$

Si $\Delta x = 0$, solamente la Ecuación (3.14) es valida, por tanto:

$$(y_{los} - y)^2 = (nL_{pp})^2 \quad (3.22)$$

Teniendo como resultado:

$$y_{los} = y \pm nL_{pp} \quad (3.23)$$

En base a esto se escoge entre dos posibles soluciones:

- Si $\Delta y > 0$ entonces $y_{los} = y + nL_{pp}$
- Si $\Delta y < 0$ entonces $y_{los} = y - nL_{pp}$

En este caso adquirimos x_{los} de $x_{los} = x_{k-1} (= x_k)$ ya que $\Delta x = 0$.

Ahora pondremos a prueba el sistema de guiado, basado en el algoritmo de linea de visión, para lo cual se tomaran puntos arbitrarios, y se aplicara dicha técnica, vale decir que la etapa de control aun no esta ajustada plenamente por ello no se logra un seguimiento con error cero, como se puede ver en la Figura 3.15 donde vemos la trayectoria de la embarcación.

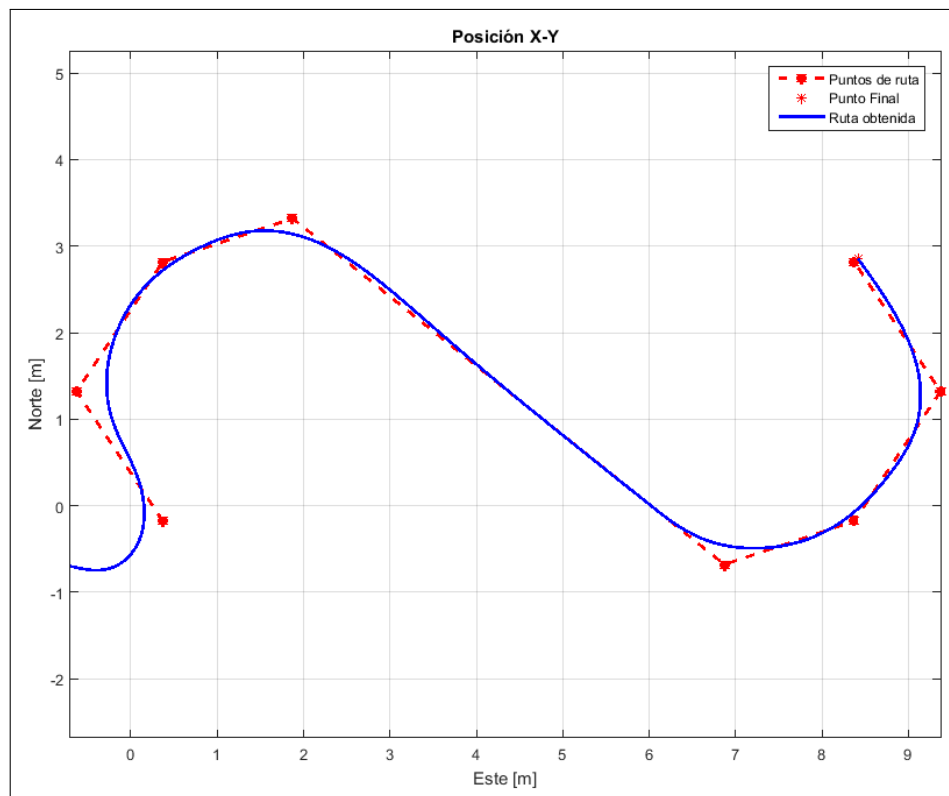


Figura 3.15: Resultado de aplicación de algoritmo linea de visión.

En la Figura 3.16 se puede ver el ángulo LOS obtenido, que es el curso que sigue el barco a fin de cumplir la ruta deseada.

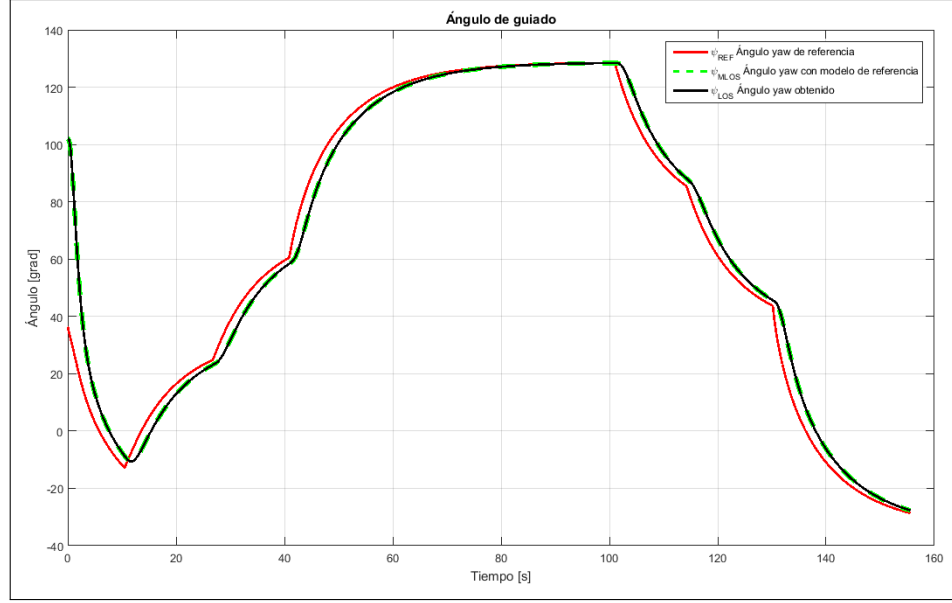


Figura 3.16: Curso de barco obtenido, ψ_{los}

3.3.2. Sistema de referencia móvil

Esta sección se desarrolla la ley de guiado, requerida para resolver problema de seguimiento de ruta planar, caso 2D. Tomando como referencia a Breivik et al. (2004), definimos un sistema de referencia local en $p_d(x, y)$ y lo llamamos sistema paralelo a la ruta (PP). Acorde a la Figura 3.17, este sistema es rotado un ángulo χ_t , relativo al sistema de referencia inercial. Debido a esto, la trayectoria deseada se convierte en una línea recta y no en una ruta curva, con esto tenemos que χ_t es una constante.

El sistema de referencia PP, es tangente a la ruta geométrica deseada y además p_d es la posición deseada del vehículo. El error expresado en un vector 2D a partir de la diferencia entre la posición actual del vehículo p respecto a la posición deseada p_d expresada en el sistema PP es dado por:

$$\varepsilon = R_p^T(\chi_t) (p - p_d) \quad (3.24)$$

Donde:

$$R_p^T = \begin{bmatrix} \cos(\chi_t) & -\sin(\chi_t) \\ \sin(\chi_t) & \cos(\chi_t) \end{bmatrix} \in \mathbf{SO}(2) \quad (3.25)$$

Como resultado del desarrollo de esta técnica que se puede ver en Breivik et al. (2004), y realizando las consideraciones propias de nuestro modelo y técnica de control se plantea una ruta paramétrica senoidal obtienen los resultados mostrados en las Figura 3.18 y Figura 3.19.

En la Figura 3.18, podemos observar que el barco logra seguir la ruta deseada

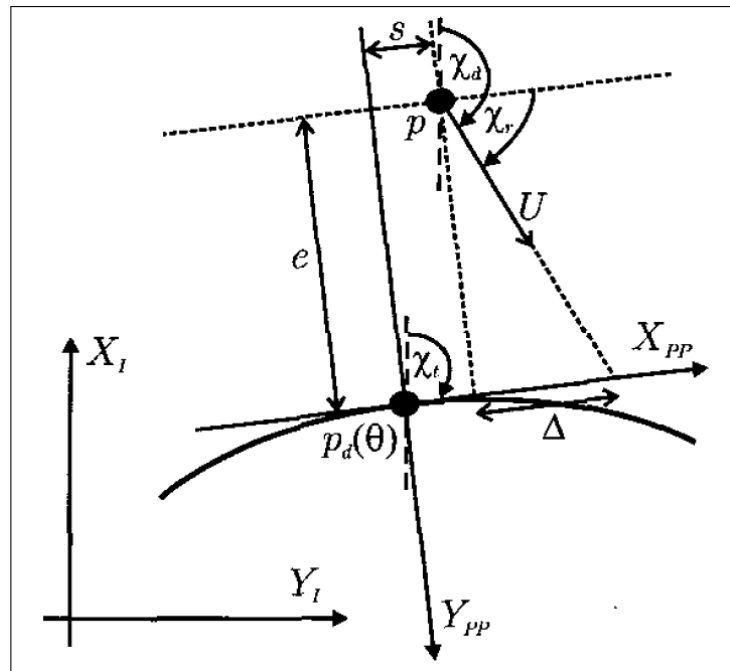


Figura 3.17: Representación de marco móvil, (Breivik et al., 2004)

sin problemas a pesar de tener una geometría especial como es la senoidal, al no tener perturbaciones no se generan cambios de dirección forzados. El sistema de guiado es presentado en la Figura 3.19, donde los ángulos presentados son los que se muestran en la Figura 3.17, vemos que $\beta \approx 0$ debido a que al no presentarse perturbaciones no hay deslizamiento considerable, también se puede observar el ángulo generado por la ruta paramétrica donde se ve a $t \approx 70$ segundos, el ángulo del barco logra converger a este valor y luego permanece estable y realizando el correspondiente seguimiento a pesar de los cambios de dirección, finalmente vemos que χ_r que esta asociado al ángulo de rotación del marco móvil, vemos que cuando los ángulos χ_p y ψ_{LOS} se igualan, entonces $\chi_r \approx 0$ debido a que no existe rotación entre el los sistemas del barco y la ruta.

3.4. Sistema de control

A continuación se presentará la estrategia de control aplicada para lograr el seguimiento de la ruta obtenida en la sección anterior.

3.4.1. Control Backstepping

Backstepping es una metodología de diseño para la construcción de una ley de control por realimentación a través de la construcción recursiva de una función de control Lyapunov. Los diseños por backstepping no lineal son fuertemente relacionados con la técnica de “feedback linearization”, pero ambos tienen diferencias

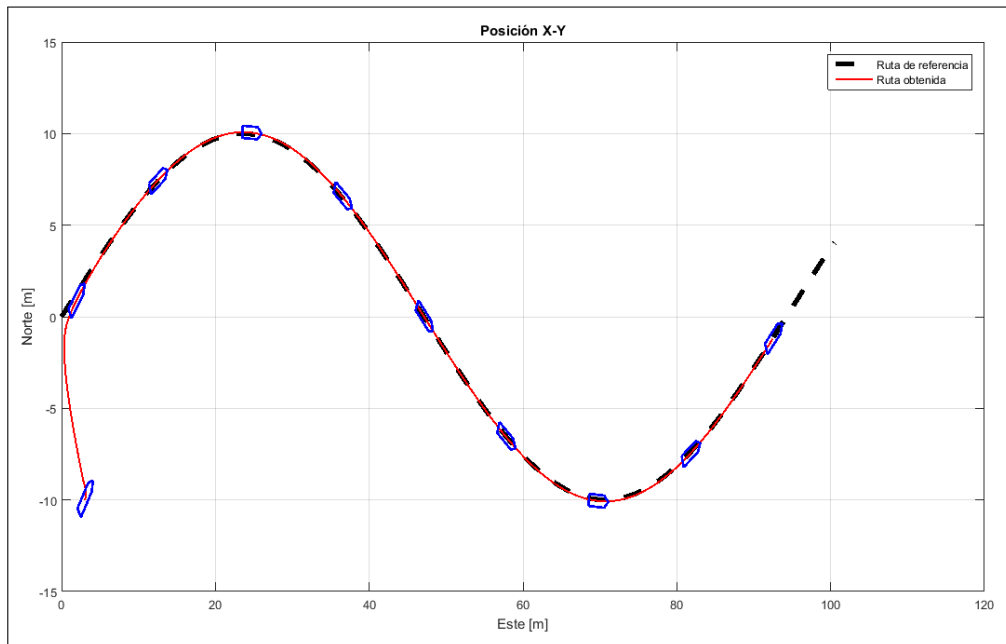


Figura 3.18: Resultado de prueba aplicando Marco móvil, Referencia (líneas punteadas), Resultado obtenido (Línea continua).

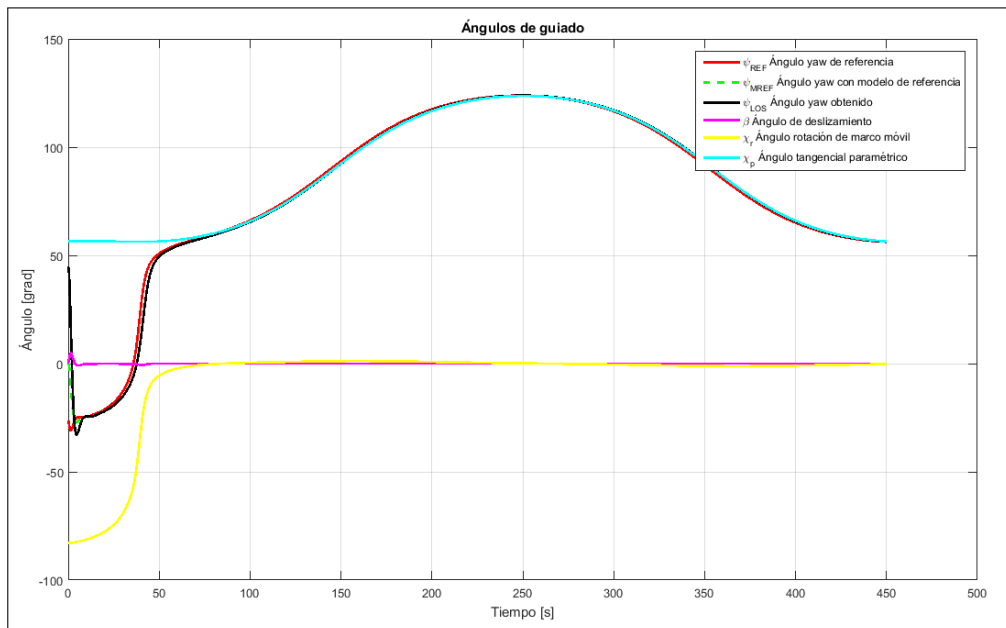


Figura 3.19: Curso de barco obtenido, ψ_{los} , por el método Marco Móvil.

marcadas. Mientras el método de “feedback linearization”, cancela todas las no linealidades en el sistema se mostrará que cuando se aplica backstepping se tendrá más flexibilidad en el diseño. En particular el diseñador podrá aprovechar las “buenas” no linealidades del sistema, mientras las “malas” no linealidades pueden ser dominadas. Por ello, una robustez adicional es obtenida, las cuales son muy importantes en los sistemas de control industrial, debido a que para conseguir la cancelación de todas las no linealidades es necesario tener un modelo muy preciso que difícilmente es conseguido en la práctica.

3.4.2. Diseño de ley control MIMO backstepping para barcos

Los sistemas convencionales de control de barcos, trabajan bajo la presunción de que las ecuaciones cinemáticas pueden ser linealizadas de tal manera que se puedan aplicar técnicas de ganancia programada y teoría de control óptimo Fossen (1994). Esta no es una buena asunción puesto que la dirección *surge*, *sway* y *yaw* deben ser controladas simultáneamente. Para lograr esto según Fossen (2002), es necesario linealizar la matriz de yaw en 36 puntos de operación para lograr una precisión adecuada.

Consideramos que la señal de control en la dirección sway es subactuada.

$$\tau = \begin{bmatrix} \tau_1 \\ 0 \\ \tau_3 \end{bmatrix} \quad (3.26)$$

Que representa la configuración de actuador para nuestro caso. Debido a que τ_2 no es posible de implementar, los términos requeridos para la dirección sway, la dinámica puede ser impuesta sobre la correspondiente función de estabilización de tal manera que se cumpla la ley de control.

Partimos del modelo de una embarcación marina de superficie. La cinemática y cinética para 3 grados de libertad pueden ser representadas como:

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\boldsymbol{\psi})\boldsymbol{\nu} \quad (3.27)$$

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau} + \mathbf{R}(\boldsymbol{\psi})^T \mathbf{b} \quad (3.28)$$

Donde $\boldsymbol{\eta} = [x \ y \ \psi]^T$, representa la posición y orientación del barco, representan las velocidades. $\mathbf{R}(\boldsymbol{\psi})$, representa la matriz de rotación desde el marco fijo hacia el marco de referencia del cuerpo de la embarcación. \mathbf{M} es la matriz

de inercia, \mathbf{C} la matriz centrífuga de coreolis, \mathbf{D} la matriz de amortiguamiento hidrodinámico, $\boldsymbol{\tau}$ la matriz de fuerza y torque por parte del barco, \mathbf{b} describe todas las fuerzas y momentos del ambiente, por ejemplo efectos causados por el viento, corrientes del océano y carga por olas de segundo orden, se asume que estos efectos varían de forma muy lenta en comparación con la dinámica del barco de tal manera que, se puede considerar que la derivada es cero, $\dot{\mathbf{b}} \approx \mathbf{0}$, esta es una suposición válida.

Empezamos por definir la proyección del vector \mathbf{h} :

$$\mathbf{h} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \quad (3.29)$$

Luego definimos las variables de error $z_1 \in \mathbb{R}$ y $\mathbf{z}_2 \in \mathbb{R}^3$, de acuerdo a:

$$z_1 = \psi - \psi_d = \mathbf{h}^T \boldsymbol{\eta} - \psi_d \quad (3.30)$$

$$\mathbf{z}_2 = \begin{bmatrix} z_{2,1} & z_{2,2} & z_{2,3} \end{bmatrix} = \boldsymbol{\nu} - \boldsymbol{\alpha} \quad (3.31)$$

Donde $\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \end{bmatrix} \in \mathbb{R}^3$ es un vector de la función de estabilización a ser especificado después.

Paso 1:

Se define la primera función de control de Lyapunov como:

$$V_1 = \frac{1}{2} z_1^2 > 0 \quad (3.32)$$

Diferenciando V_1 con respecto del tiempo considerando la dinámica de z_1 , se obtiene:

$$\dot{V}_1 = z_1 \dot{z}_1 \quad (3.33a)$$

$$\dot{V}_1 = z_1 (\mathbf{h}^T \dot{\boldsymbol{\eta}} - \dot{\psi}_d) \quad (3.33b)$$

$$\dot{V}_1 = z_1 (\mathbf{h}^T \boldsymbol{\nu} - \dot{\psi}_d) \quad (3.33c)$$

Dado que $\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{\nu}$ y $\mathbf{h}^T \dot{\boldsymbol{\eta}} = \mathbf{h}^T \mathbf{R}(\psi)\boldsymbol{\nu}$, de donde obtenemos:

$$\dot{V}_1 = z_1 (\mathbf{h}^T (\mathbf{z}_2 + \boldsymbol{\alpha}) - \dot{\psi}_d) \quad (3.34a)$$

$$\dot{V}_1 = z_1 \mathbf{h}^T \mathbf{z}_2 + z_1 (\mathbf{h}^T \boldsymbol{\alpha} - \dot{\psi}_d) \quad (3.34b)$$

$$\dot{V}_1 = z_1 \mathbf{h}^T \mathbf{z}_2 + z_1 (\alpha_3 - \dot{\psi}_d) \quad (3.34c)$$

Con esto escogemos la función de estabilización α_3 como:

$$\alpha_3 = -k_1 z_1 + \dot{\psi}_d \quad (3.35)$$

Donde $k_1 > 0$, lo cual resulta en:

$$\dot{V}_1 = -k_1 z_1^2 + z_1 \mathbf{h}^T \mathbf{z}_2 \quad (3.36)$$

Con esto se concluye la parte 1.

Paso 2:

Aumentando la primera función de control de lyapunov obtenemos:

$$V_2 = V_1 + \frac{1}{2} \mathbf{z}_2^T \mathbf{M} \mathbf{z}_2 + \frac{1}{2} \bar{\mathbf{b}}^T \Gamma \bar{\mathbf{b}} > 0 \quad (3.37)$$

Donde $\bar{\mathbf{b}} \in \mathbb{R}^3$ es un error de adaptación definido como: $\bar{\mathbf{b}} = \hat{\mathbf{b}} - \mathbf{b}$, considerando al vector $\hat{\mathbf{b}}$ como el estimado del vector de perturbación ambiental \mathbf{b} , además consideramos una variación muy lenta de tal manera que $\dot{\mathbf{b}} = 0$ para efectos prácticos, esto quiere decir que $\dot{\bar{\mathbf{b}}} = \dot{\hat{\mathbf{b}}}$ y $\Gamma = \Gamma^T > 0$, que es la matriz de ganancia de adaptación.

De igual manera derivamos V_2 , respecto del tiempo sobre las trayectorias z_1 , z_2 y $\bar{\mathbf{b}}$, obtenemos:

$$\dot{V}_2 = -k_1 z_1^2 + z_1 \mathbf{h}^T \mathbf{z}_2 + \mathbf{z}_2^T \mathbf{M} \dot{\mathbf{z}}_2 + \bar{\mathbf{b}}^T \Gamma^{-1} \dot{\hat{\mathbf{b}}} \quad (3.38)$$

A partir de: $\mathbf{M} = \mathbf{M}^T$ y $\dot{\bar{\mathbf{b}}} = \dot{\hat{\mathbf{b}}}$, tenemos que:

$$\mathbf{M} \dot{\mathbf{z}}_2 = \mathbf{M}(\dot{\boldsymbol{\nu}} - \dot{\boldsymbol{\alpha}}) \quad (3.39a)$$

$$\mathbf{M} \dot{\mathbf{z}}_2 = \boldsymbol{\tau} + \mathbf{R}^T \mathbf{b} - \mathbf{C}(\boldsymbol{\nu}) \boldsymbol{\nu} - \mathbf{D}(\boldsymbol{\nu}) \boldsymbol{\nu} - \mathbf{M} \dot{\boldsymbol{\alpha}} \quad (3.39b)$$

Reemplazando esta equivalencia en la Ecuación (3.38) tenemos lo siguiente:

$$\dot{V}_2 = -k_1 z_1^2 + \mathbf{z}_2^T (\mathbf{h} z_1 + \boldsymbol{\tau} + \mathbf{R}^T \mathbf{b} - \mathbf{C}(\boldsymbol{\nu}) \boldsymbol{\nu} - \mathbf{D}(\boldsymbol{\nu}) \boldsymbol{\nu} - \mathbf{M} \dot{\boldsymbol{\alpha}}) + \bar{\mathbf{b}}^T \Gamma^{-1} \dot{\hat{\mathbf{b}}} \quad (3.40)$$

Reescribiendo $\mathbf{C}(\boldsymbol{\nu}) = \mathbf{C}$ y $\mathbf{D}(\boldsymbol{\nu}) = \mathbf{D}$ para abreviar la notación, utilizando $\boldsymbol{\nu} = \mathbf{z}_2 + \boldsymbol{\alpha}$ y $\mathbf{b} = \hat{\mathbf{b}} - \bar{\mathbf{b}}$ podemos obtener:

$$\dot{V}_2 = -k_1 z_1^2 - \mathbf{z}_2^T \mathbf{C} \mathbf{z}_2 - \mathbf{z}_2^T \mathbf{D} \mathbf{z}_2 + \mathbf{z}_2^T (\mathbf{h} z_1 + \boldsymbol{\tau} + \mathbf{R}^T \hat{\mathbf{b}} - \mathbf{C} \boldsymbol{\alpha} - \mathbf{D} \boldsymbol{\alpha} - \mathbf{M} \dot{\boldsymbol{\alpha}}) + \bar{\mathbf{b}}^T \Gamma^{-1} (\dot{\hat{\mathbf{b}}} - \Gamma \mathbf{R} \mathbf{z}_2) \quad (3.41)$$

Donde $\mathbf{z}_2^T \mathbf{C} \mathbf{z}_2 = 0$ debido a que \mathbf{C} es skew-simétrica, de donde tenemos:

$$\boldsymbol{\tau} = \mathbf{M}\dot{\boldsymbol{\alpha}} + \mathbf{C}\boldsymbol{\alpha} + \mathbf{D}\boldsymbol{\alpha} - \mathbf{R}^T\hat{\mathbf{b}} - \mathbf{h}z_1 - \mathbf{K}_2\mathbf{z}_2 \quad (3.42)$$

Donde: $\mathbf{K}_2 = \text{diag}(k_{2,1}, k_{2,2}, k_{2,3}) > 0$, y seleccionando:

$$\dot{\hat{\mathbf{b}}} = \boldsymbol{\Gamma}\mathbf{R}\mathbf{z}_2 \quad (3.43)$$

Finalmente obtenemos:

$$\dot{V}_2 = -k_1z_1^2 - \mathbf{z}_2(\mathbf{D} + \mathbf{K}_2)\mathbf{z}_2 \leq 0 \quad (3.44)$$

De donde tenemos que las funciones de estabilización $\boldsymbol{\alpha} = \begin{bmatrix} u_d & \alpha_2 & -k_1z_1 + \dot{\psi}_d \end{bmatrix}$
Donde α_2 , es hallado a partir de la dinámica para el sistema subactuado.

3.5. Conclusiones Preliminares

- Se estudió el planeamiento de movimiento basado en el algoritmo de Voronoi para luego aplicar el Dijkstra y de esta manera obtener los puntos de ruta necesarios para que el barco logre el objetivo.
- Al generar una ruta con los métodos estudiados en la Sección 3.2.3.1 y 3.2.3.2, se puede apreciar que el primero genera un mayor recorrido al tener una longitud total de ruta de 134 metros mientras el segundo 121 metros, esto debido que el segundo solo usa la curvatura en los cambios de curso, luego usa líneas rectas que hacen más eficiente la ruta.

Tabla 3.1: Longitud de ruta generada por ambos métodos

	Polinomio Cúbico de Hermite	Espiral de Fermat
Longitud de Ruta [m]	134	121

- El control “Backstepping” propuesto, presenta una metodología que podría ser aplicada a otros sistemas dinámicos subactuados. Puesto que se controlan tres variables con únicamente 2 variables de control
- La forma planteada para procesar planos reales, es a través de la simplificación poligonal, se debe tener en cuenta que mientras esta simplificación sea más detallada se obtendrán mejores resultados con el diagrama de Voronoi.

Capítulo 4

Simulación

En el presente capítulo se desarrolla la simulación y aplicación de técnicas estudiadas, para lo cual hacemos uso del software *MATLAB*[®], en el cual se implementan los algoritmos de planeamiento, guiado y control de rutas. Asimismo, se presentan dos simulaciones, una corresponde a un canal senoidal mientras que el otro corresponde a un mapa tomado de Google (2015), que corresponde al mar Báltico, en ambos se aplican las desarrolladas en esta tesis. Cabe resaltar que al momento de realizar la simulación del mar Báltico no se consideran medidas reales, puesto que, lo que se requiere en las simulaciones, son las formas particulares de un entorno real.

A continuación se detallan las consideraciones hechas para simular:

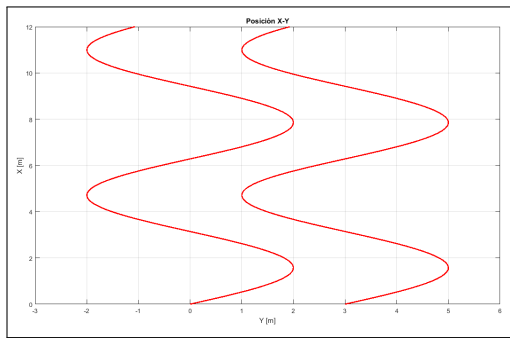
- Los parámetros y modelos dinámicos del sistema son conocidos.
- Para realizar el control, se considera que todos los estados son medidos. La posición puede ser medida por un sistema de posicionamiento global (GPS). Las velocidades se pueden obtener a partir de la integración de velocidades obtenidos con la unidad de medición inercial (IMU).
- Los valores medidos de posición, velocidad y aceleración, son considerados libres de ruido.
- Se asume que se puede dar un control ideal.

4.1. Simulación en canal senoidal

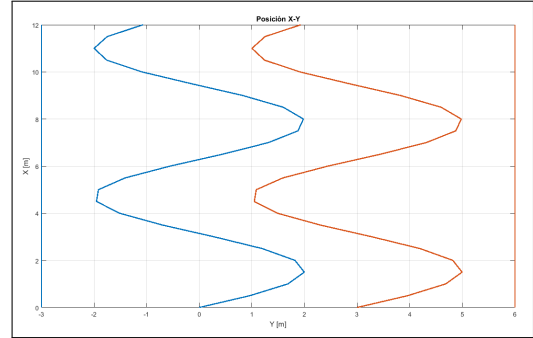
4.1.1. Planeamiento de movimiento y trayectoria

En esta sección se trabajará el planeamiento partiendo de un mapa, al cual se aplicará una simplificación poligonal y a partir del cual se realizará el trabajo de planeamiento de movimiento.

A partir de este mapa lo que se hará es la simplificación poligonal del mismo para de esta manera aplicarle las técnicas revisadas respecto al planeamiento de movimiento. Para poder visualizar esto se plantea un canal senoidal que se presenta en la Figura 4.1a antes de aplicar la simplificación y después de aplicar la simplificación, Figura 4.1b.

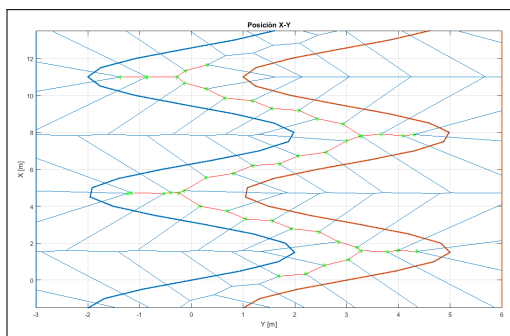


(a) Mapa de canal en forma senoidal.

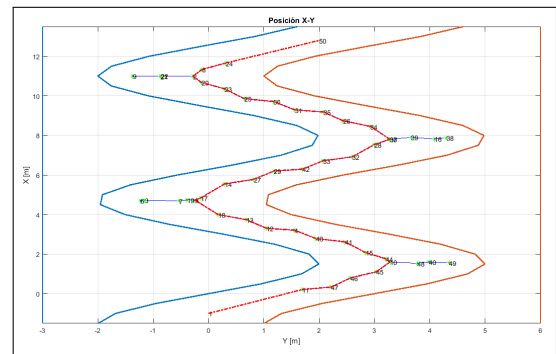


(b) Mapa de canal con simplificación

Figura 4.1: Mapa antes y después de la aplicar la simplificación poligonal.



(a) Aplicando diagrama de Voronoi.



(b) Aplicando el algoritmo de Dijkstra.

Figura 4.2: Se obtiene la ruta deseada continuando con el planeamiento de movimiento.

Luego aplicamos el diagrama de Voronoi a fin de obtener los posibles puntos para generar la ruta deseada.(ver Figura 4.2a). A fin de obtener la mejor ruta partiendo del diagrama de Voronoi, aplicamos el algoritmo de Dijkstra (ver Figura 4.2b)

4.1.2. Determinación de puntos de ruta “WayPoints”.

A fin de evitar los cambios de dirección innecesarias aplicamos un algoritmo de tal forma que obtenemos los puntos de ruta necesarios para trazar la trayectoria, teniendo como resultado la Figura 4.3.

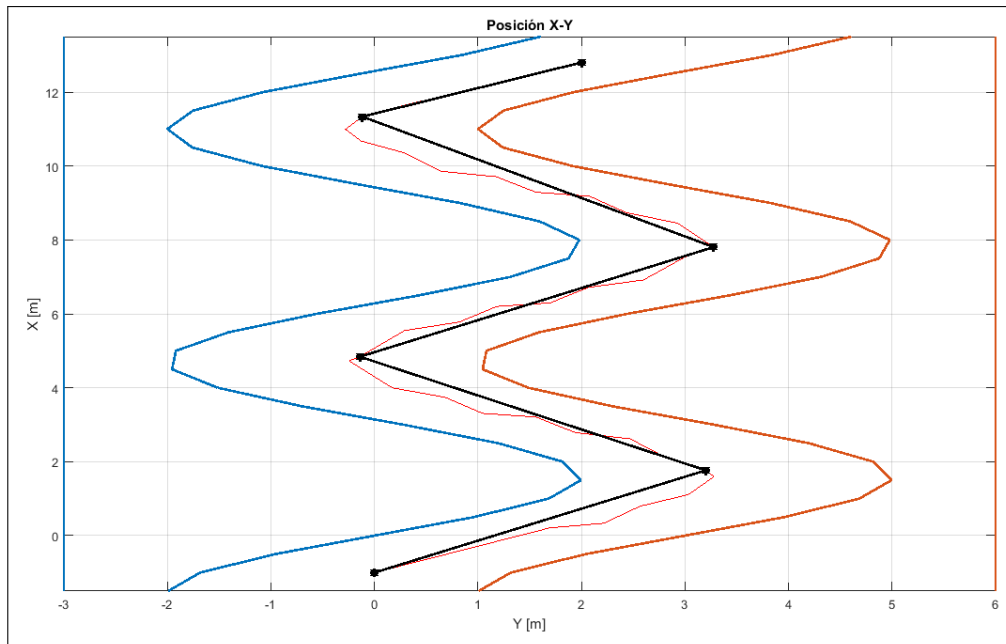


Figura 4.3: A partir de los puntos obtenidos por el algoritmo, determinamos los puntos que producen menor ángulo de cabeceo.

4.1.3. Determinación de ruta paramétrica.

Tomando como base los puntos de ruta, se procede a la generación de ruta paramétrica usando las técnicas de polinomios de Hermite y el espiral de Fermat revisadas en las Secciones 3.2.3.2 y 3.2.3.1, para esto se implementa un programa generador de trayectoria en MATLAB.

A continuación se presentará los resultados obtenidos, para nuestro canal de forma senoidal, se mostrará la ruta obtenida así como la gráfica de cada componente respecto a la longitud recorrida.

4.1.3.1. Polinomio cúbico de hermite

Aplicando la Ecuación (3.7), a partir de los puntos obtenidos en la sección anterior obtenemos los siguientes resultados, la ruta en X-Y, Figura 4.4, y los componentes de la ruta respecto de la longitud recorrida, Figura 4.5:

4.1.3.2. Espiral de Fermat

Aplicando las Ecuaciones (3.9) y (3.10), a partir de los puntos de ruta obtenidos en la sección anterior tenemos como resultado la ruta en X-Y, Figura 4.6, y los componentes de ruta respecto de la longitud recorrida, Figura 4.7.

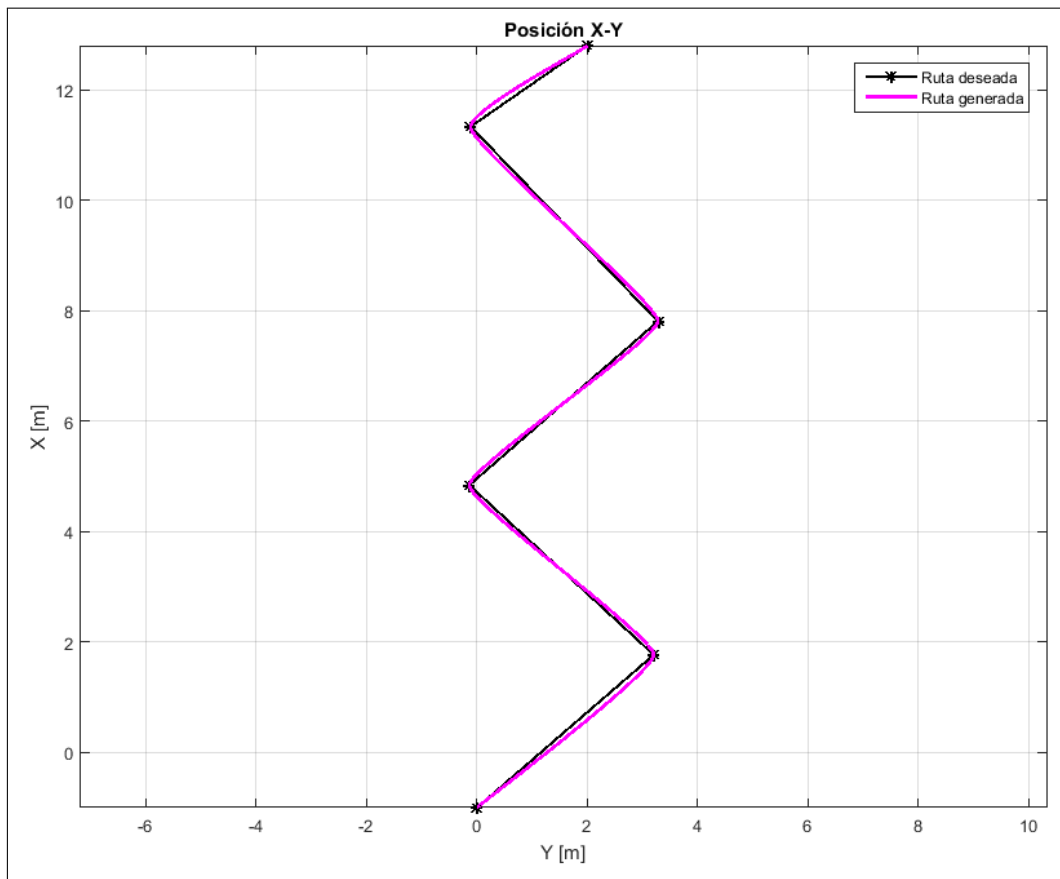


Figura 4.4: Ruta generada por polinomio de Hermite (magenta)

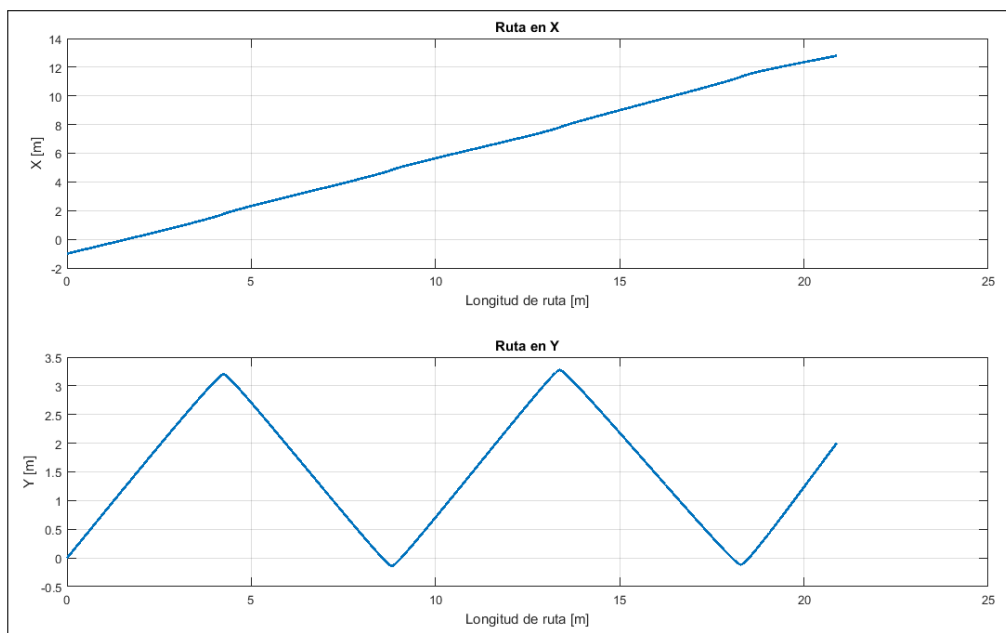


Figura 4.5: Componentes de ruta vs longitud recorrida

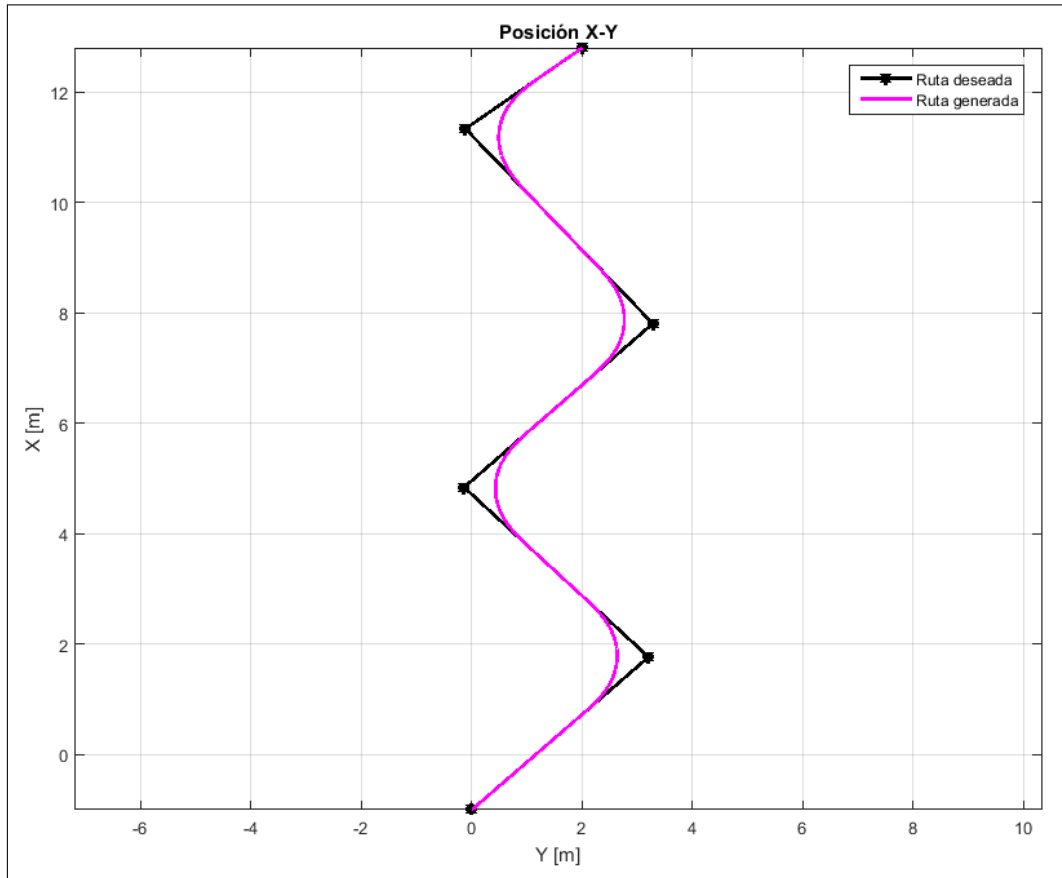


Figura 4.6: Ruta paramétrica generada por espiral de Fermat

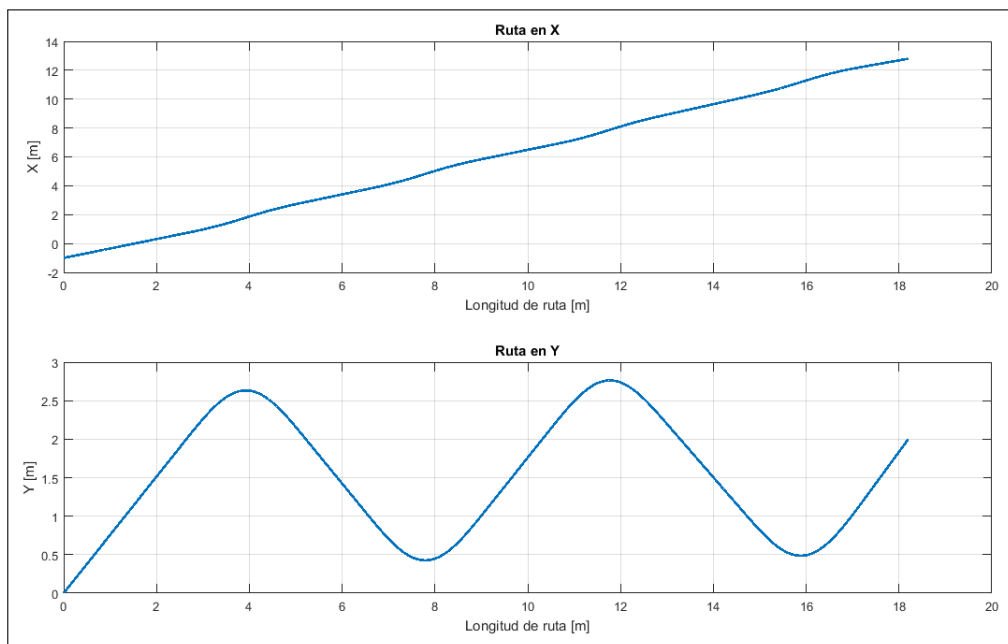


Figura 4.7: Componentes de la ruta paramétrica por espiral de Fermat vs longitud recorrida

4.1.4. Seguimiento de trayectoria y control de embarcación

A partir de los resultados de la sección anterior se obtuvieron puntos de ruta, estos son los que se usarán para la simulación obtener la guía y control de la embarcación.

4.1.4.1. Resultados de simulación

Luego de realizar la simulación a partir de los puntos de ruta obtenidos, sistema de guiado y control, además considerando la dinámica del barco, por un lapso de a lo largo de 200 segundos obtenemos la trayectoria de barco mostrada en la Figura 4.8.

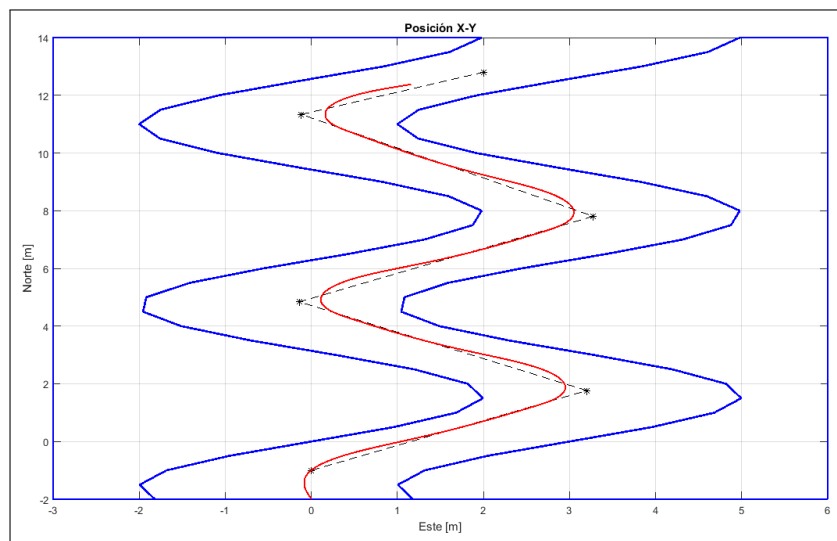


Figura 4.8: Posición de barco incluyendo la dinámica a través del canal (Trayectoria en rojo)

Se puede apreciar claramente que el barco logra atravesar el canal sin problemas de colisión, todo esto a partir del buen planeamiento de movimiento y del funcionamiento correcto del sistema de guiado y control que realiza el seguimiento de trayectorias sin problemas.

En la Figura 4.9 se presenta la ley de control dada por fuerza y torque, vemos que el valor de fuerza necesario es constante e igual a $\tau_1 = 0.2N$ y que se presentan picos en el torque cuando se hace necesario realizar el cambio de dirección, $\tau_{3pico} = 0.12N.m$.

Las velocidades alcanzadas a lo largo de la ruta se pueden ver en la Figura 4.10, vemos que la velocidad a lo largo de la ruta *surge*, se mantiene constante en un valor $u_d = 0.1m/s$, valor asignado por el capitán del navío o dado por un perfil de velocidad predeterminado, mientras que la velocidad *sway*, tiene valores cercanos

a cero $v_{max} \approx 2.10^{-3}$, lo cual es lógico, para garantizar que el barco siga la ruta deseada y no tenga deslizamiento respecto de la ruta deseada, además podemos comprobar que la velocidad de giro cambia cada vez que se presenta un cambio de dirección, alcanzando valores pico aproximados de $r_{pico} \approx .015rad/seg$.

Los errores de estado se presentan en la Figura 4.11, el error z_1 que corresponde al error en el ángulo yaw, vemos que presenta un tiempo de estabilización de aproximadamente 10 segundos cuando luego de realizar un cambio de dirección, y tiene un error en estado estacionario $z_1 \approx 0$, hasta que se produce un cambio en el curso del navío. Respecto al error de la variable de velocidad surge z_{21} , podemos ver que el error en estado estacionario es cero $z_{21} = 0$ quiere decir que a lo largo de la ruta se ha logrado alcanzar y mantener la velocidad deseada, el error de velocidad sway presenta valores pico de $z_{22pico} \approx 0.5 \times 10^{-6}$ prácticamente cero y finalmente el error de la variable de velocidad rotacional yaw tiene valores pico de $z_{23pico} \approx 2.5 \times 10^{-5}$ muy próximo al cero.

El ángulo de cabeceo o curso del navío, es presentado en la Figura 4.12, que presenta tres curvas. La primera curva es el ángulo ψ_{LOS} obtenido a través del algoritmo línea de visión, la segunda curva es este mismo ángulo ψ_{LOS} obtenido a partir de un modelo de referencia de segundo orden, con el fin de poder alimentar al sistema de control, puesto que este requiere la segunda derivada para el cálculo de la ley de control, finalmente la tercera curva es el ángulo yaw obtenido $\psi_{obtenido}$.

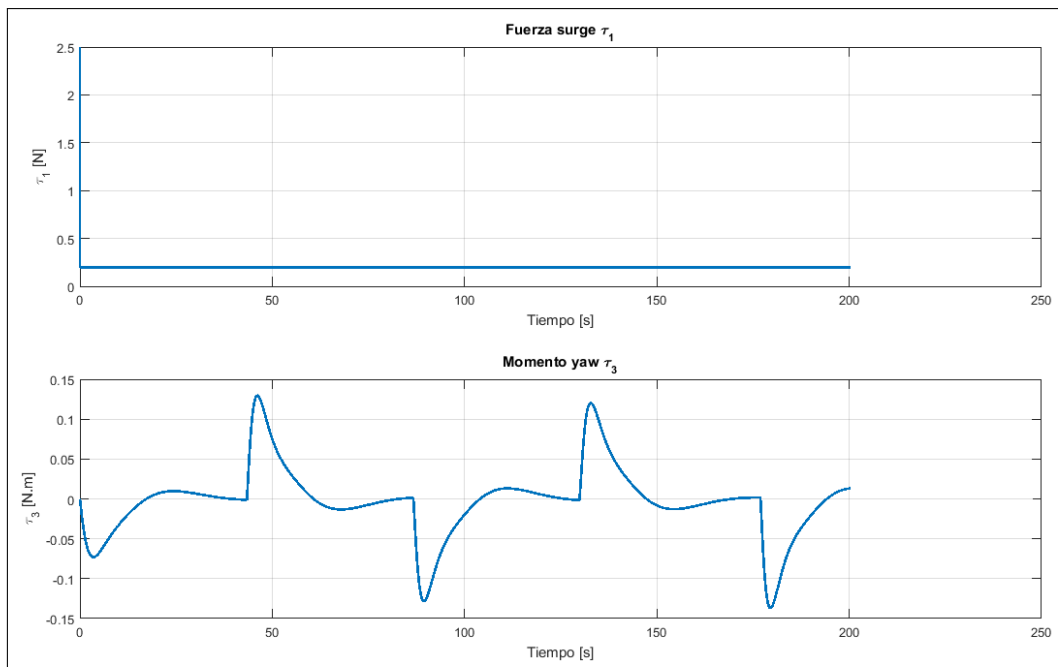


Figura 4.9: Ley de control del sistema subactuado, fuerza y torque

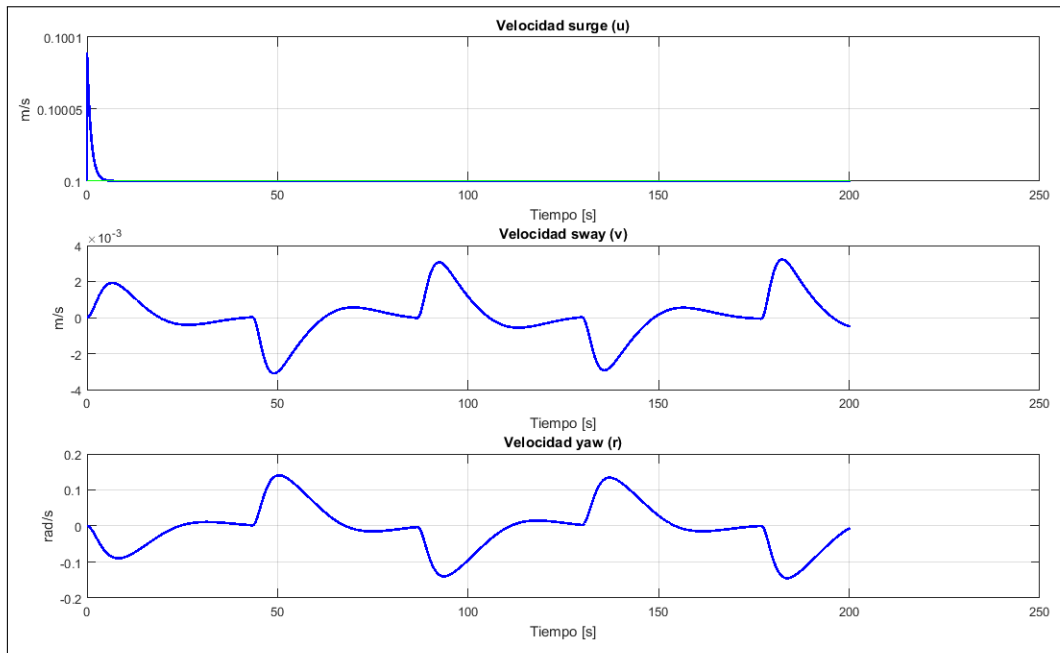


Figura 4.10: Velocidad del sistema body (surge, sway and yaw) durante la simulación

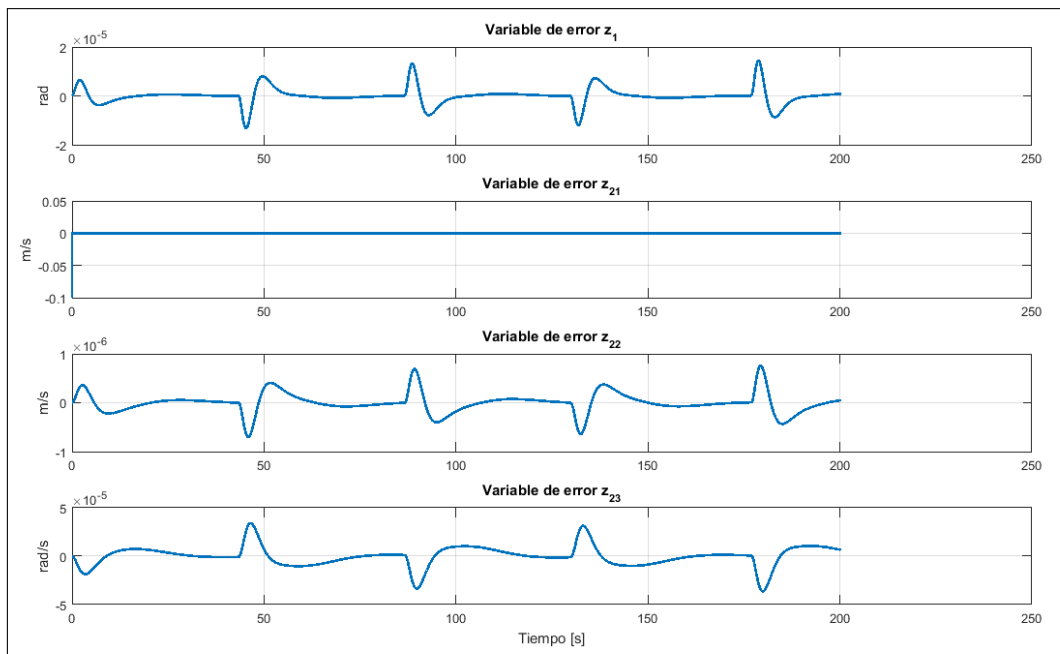


Figura 4.11: Error de estados, cabeceo, velocidad en surge, velocidad en sway y velocidad en yaw

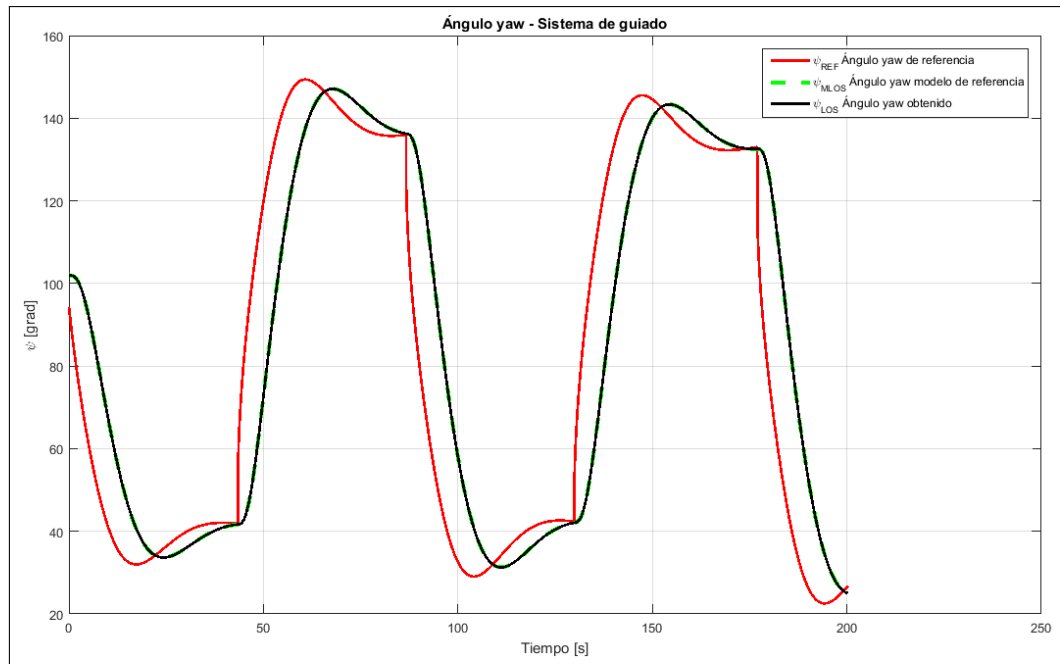


Figura 4.12: Resultado de ángulo de cabeceo deseado a partir del sistema de guía basado en Línea de Visión

4.2. Simulación en el mar Báltico

En esta sección, se realizará la simulación del sistema con el fin de poner a prueba lo desarrollado, para lo cual se usó un mapa real, obtenido de Google Maps, Google (2015), con las características de espacio confinado, toda la simulación se ha realizado en MATLAB. El mapa corresponde al mar Báltico que se encuentra al norte de Europa, se muestra en detalle en la Figura 4.13, este será el punto de partida para realizar la simplificación poligonal, para posteriormente aplicar las demás herramientas desarrolladas.

4.2.1. Planeamiento de movimiento y trayectoria en el mar Báltico

Luego de aplicar simplificación poligonal se procede a aplicar el diagrama de Voronoi, para de esta manera tener el mar caracterizado por posibles puntos de ruta. A continuación lo que se debe realizar es elegir un punto de partida y llegada, en el primer caso se plantea que el barco se dirija desde Alemania hacia Finlandia, con estos datos aplicamos el algoritmo de Dijkstra, como se puede ver en la Figura 4.14 se obtiene la posible ruta representada en color verde.

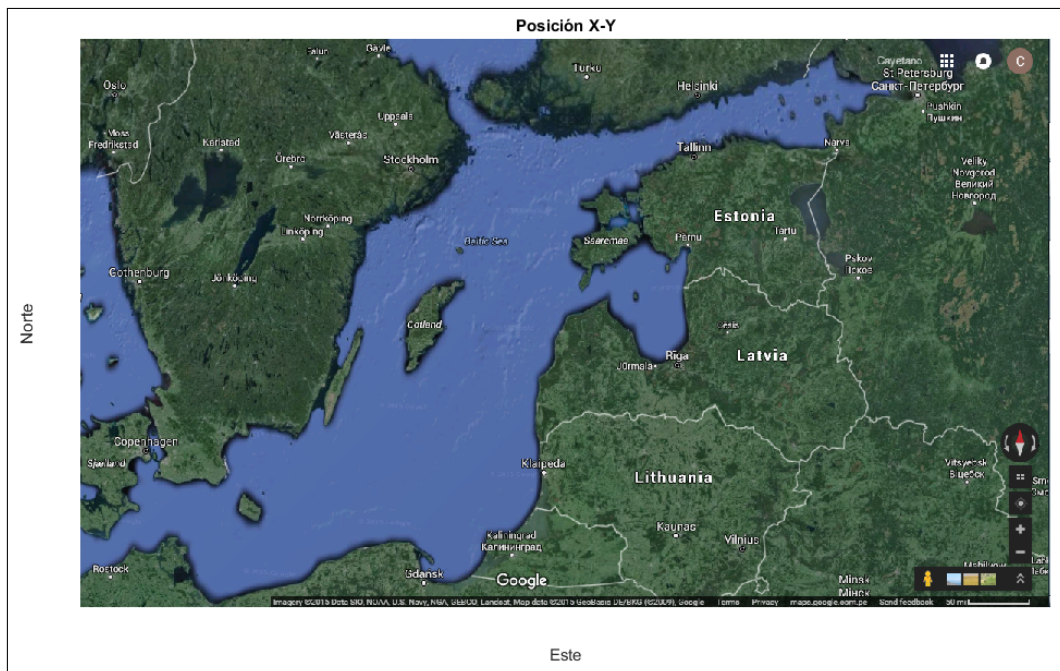


Figura 4.13: Mapa real Navegación localizado el norte de Europa,(Google, 2015)

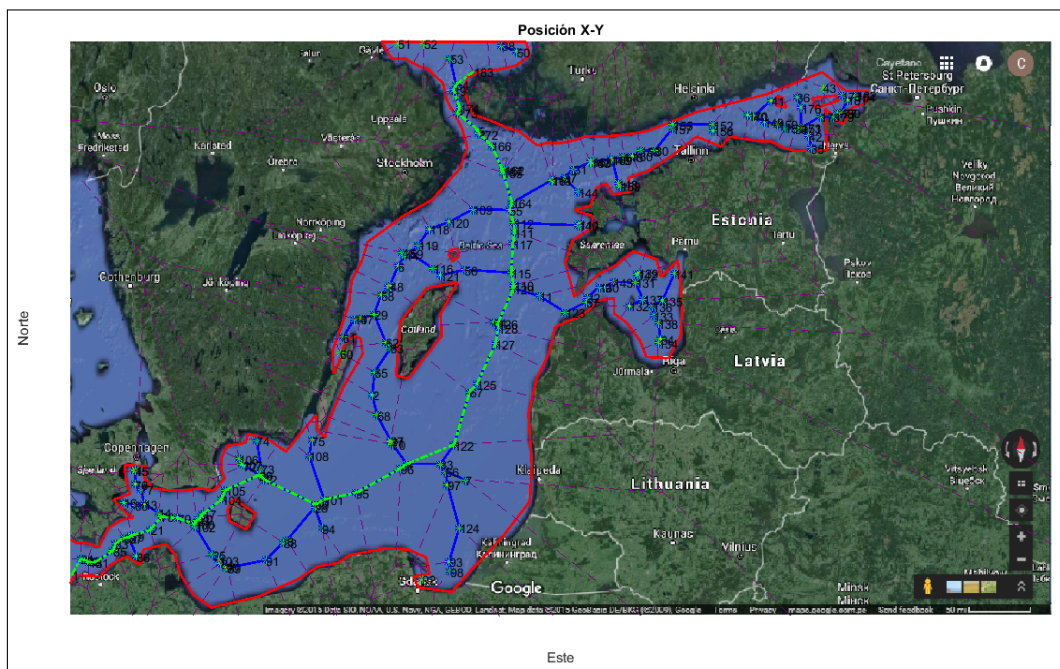


Figura 4.14: Aplicación de herramientas para navegación de Alemania hacia Finlandia

4.2.2. Determinación de puntos de ruta “WayPoints” en el mar Báltico

Luego de aplicar Dijkstra, tenemos una posible ruta, la cual en ciertos casos hace cambios de dirección innecesarios, para evitar esfuerzos de control innecesarios se plantea un algoritmo de filtrado, y este a la vez considera una tolerancia respecto a lo cerca que se puede pasar de un obstáculo, en la Figura 4.15 se pueden ver rectángulos de color cyan, que representan los puntos de ruta donde no se cumple ni con el filtrado ni la tolerancia respecto a la orilla, por lo cual en estos puntos se da un cambio de orientación de forma necesaria.

4.2.3. Determinación de ruta paramétrica en el mar Báltico

También se pueden ver dos curvas, una amarilla y una magenta que están prácticamente superpuestas, estas corresponden a las rutas generadas, la primera a usando la técnica de polinomio cúbico de Hermite, y la segunda usando la técnica de espiral de Fermat, se puede ver que ambas logran el objetivo, con la diferencia de que una demanda mayor recorrido que la otra, en este caso la que se basa en el espiral de Fermat es la que usa mayor longitud de recorrido.

Se presenta un cambio en la ruta, ahora la designada será hacia un puerto en Estonia, para realizar nuevamente obtener los resultados lo único que se debe hacer es cambiar el punto de llegada, y hacer una corrida del algoritmo, obteniendo el resultado presentado en la Figura 4.16.

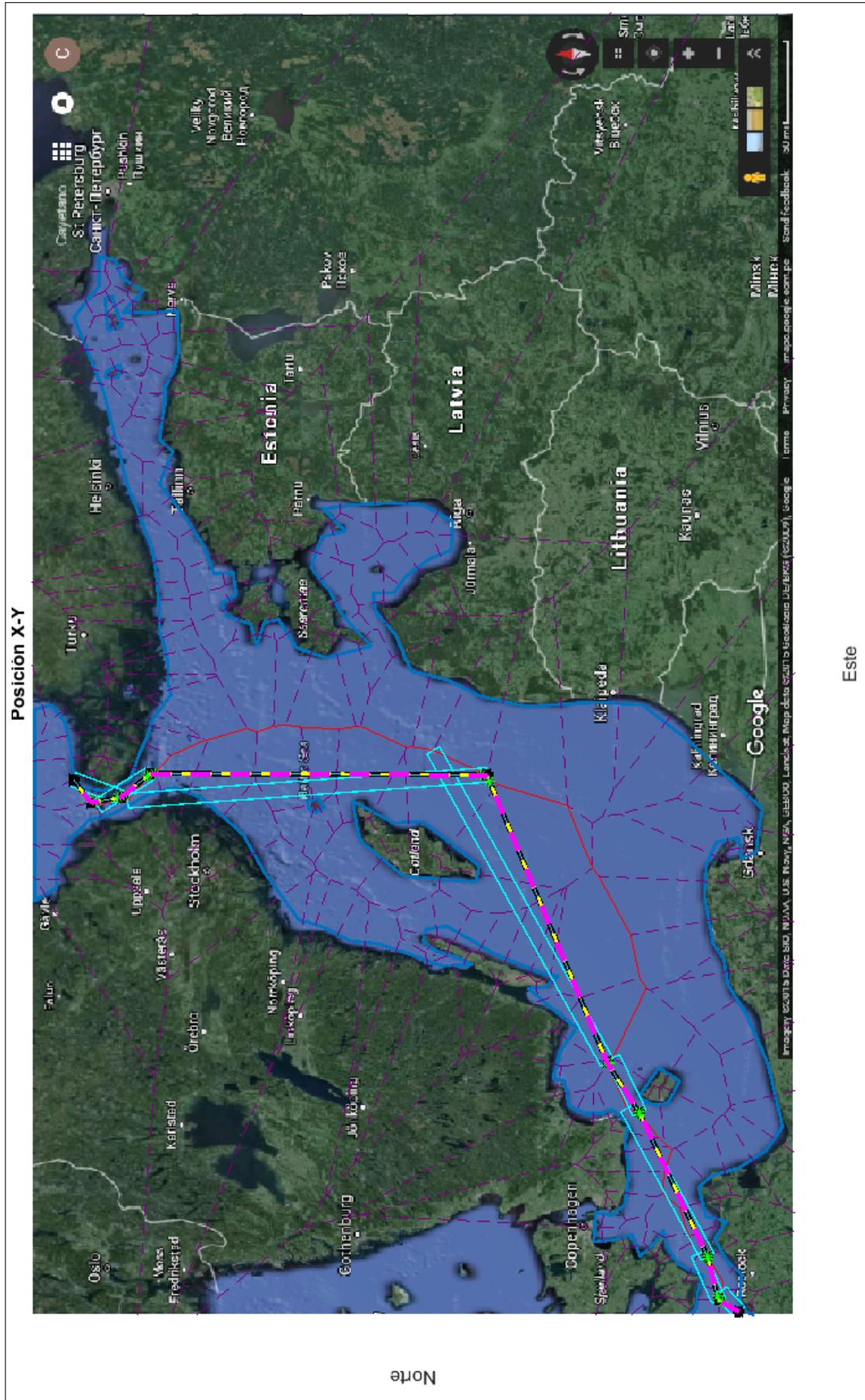


Figura 4.15: Generación de ruta, lugares donde ocurre no se cumple la tolerancia (cyan), referencia (Negro), ruta por polinomio cúbico (Amarillo) y ruta por espiral de Fermat (Magenta)

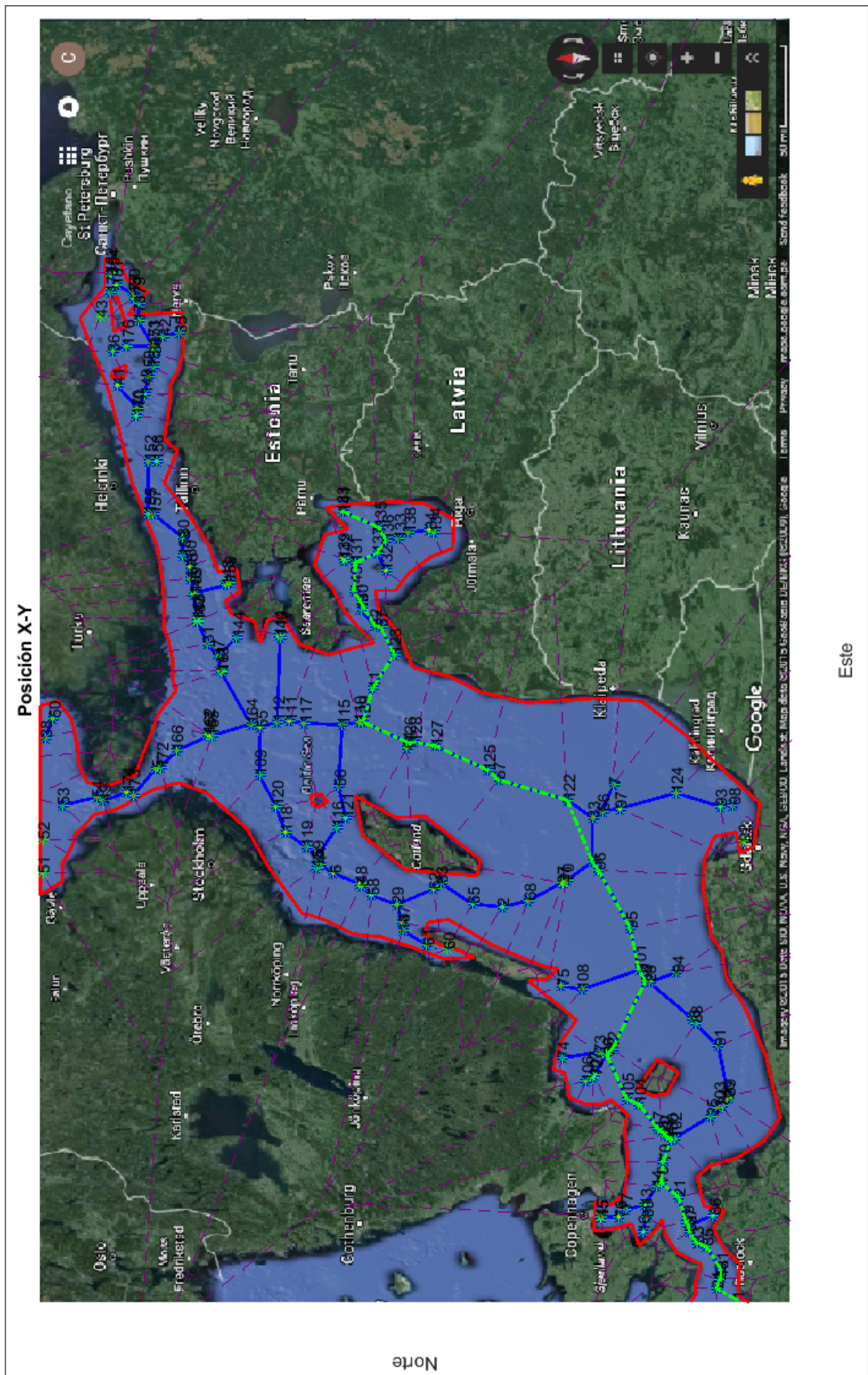


Figura 4.16: Generación de ruta para viaje de Alemania hacia Estonia

En el espacio confinado cerca a Estonia se realiza un movimiento especial, el cual se muestra en detalle en la Figura 4.17. Se puede observar que para lograr mejor disposición de los puntos de camino, la simplificación poligonal del mapa debe ser lo más precisa posible, porque de esta dependen los puntos generados por el diagrama de Voronoi. Como se puede ver en la Figura 4.17 se hace un recorrido algo ineficiente para llegar al punto final, precisamente el antepenúltimo punto es el que trae problemas, si se plantea un sistema híbrido de control de ruta manual-automático, el capitán podría corregir este error a fin de tener una longitud de ruta eficiente.

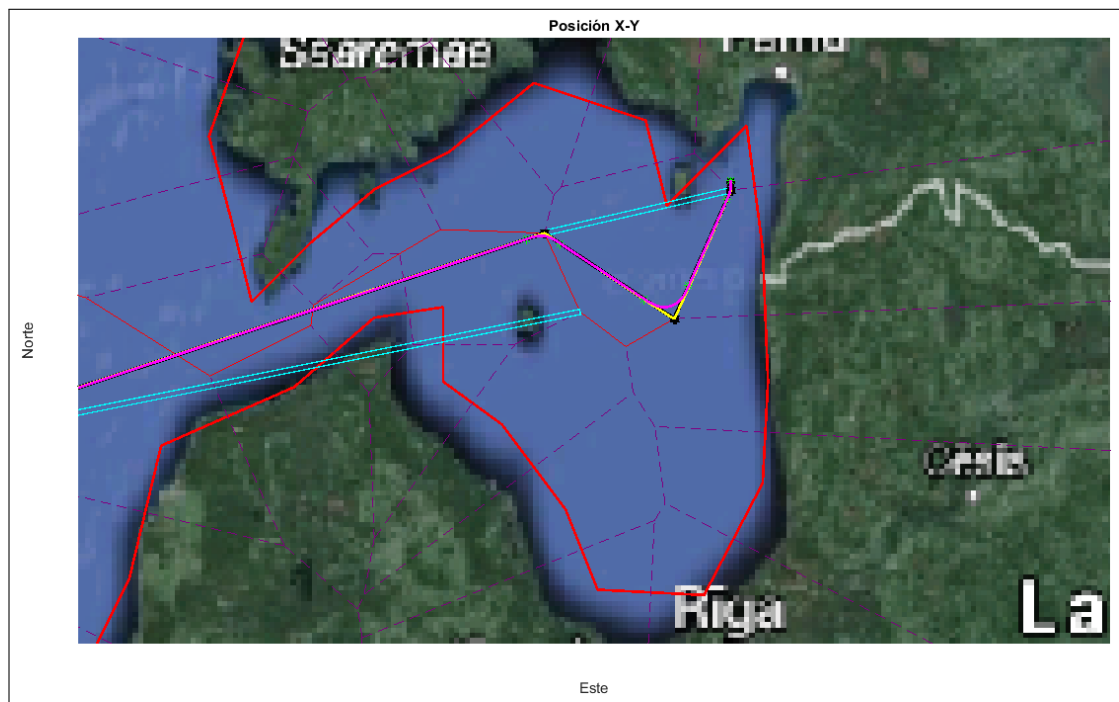


Figura 4.17: Detalle de ruta deseada cerca a Estonia, ruta por polinomio cúbico (Amarillo) y ruta por espiral de Fermat (Magenta)

4.2.3.1. Polinomio cubico de hermite en el mar Báltico

Aplicando la Ecuación (3.7), a partir de los puntos obtenidos en la sección anterior obtenemos los siguientes resultados, la ruta en X-Y, Figura 4.18, y los componentes de la ruta respecto de la longitud recorrida, Figura 4.19:

4.2.3.2. Espiral de Fermat en el mar Báltico

Aplicando las Ecuaciones (3.9) y (3.10), a partir de los puntos de ruta obtenidos en la sección anterior tenemos como resultado la ruta en X-Y, Figura 4.20, y los componentes de ruta respecto de la longitud recorrida, Figura 4.21.

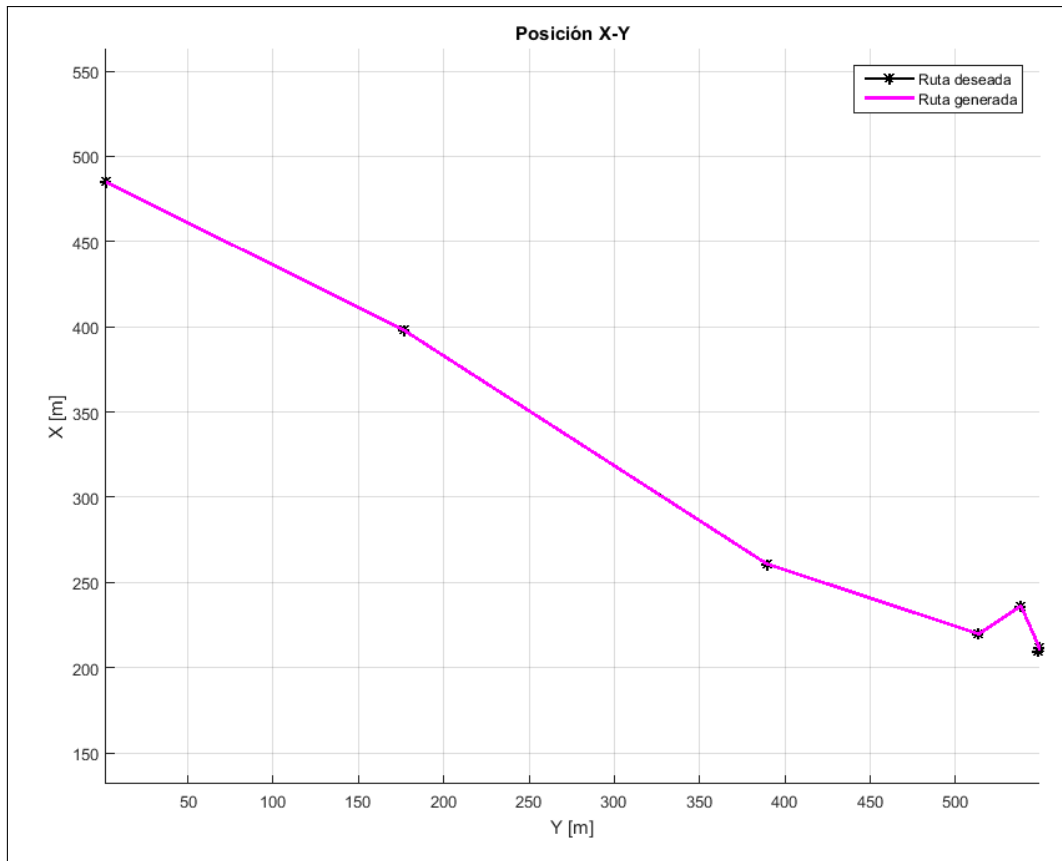


Figura 4.18: Ruta generada por polinomio de Hermite (magenta) en el mar Báltico

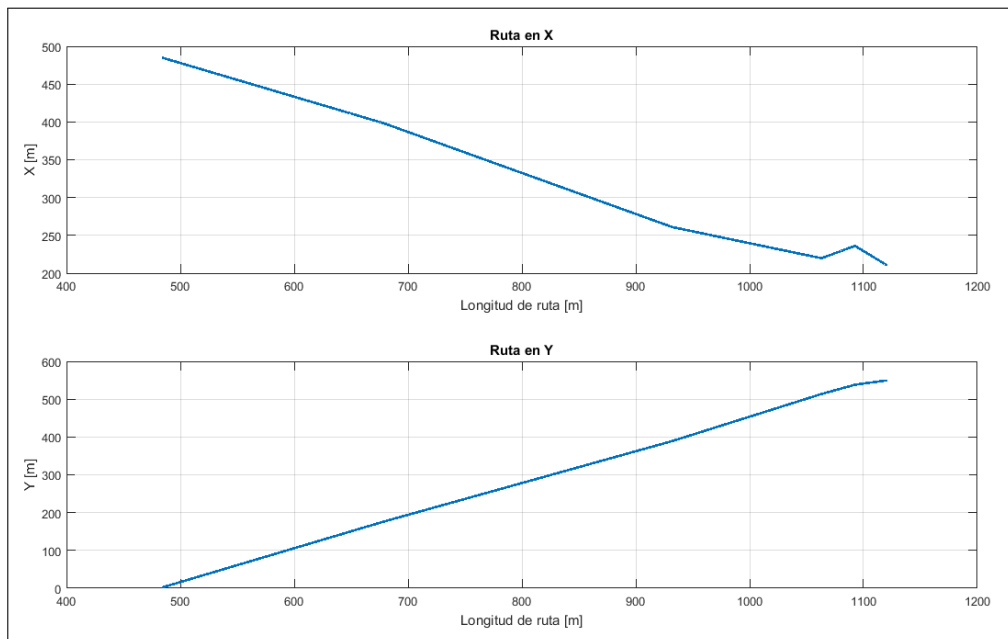


Figura 4.19: Componentes de ruta vs longitud recorrida en el mar Báltico

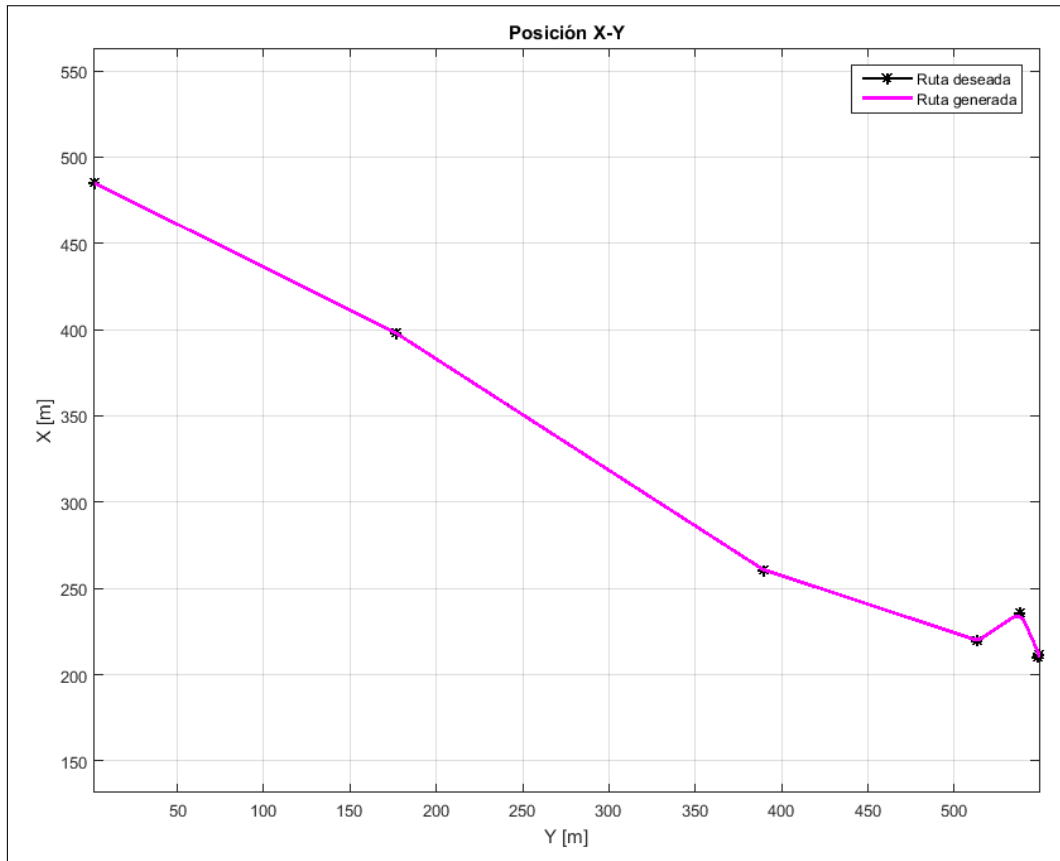


Figura 4.20: Ruta paramétrica generada por espiral de Fermat en el mar Báltico

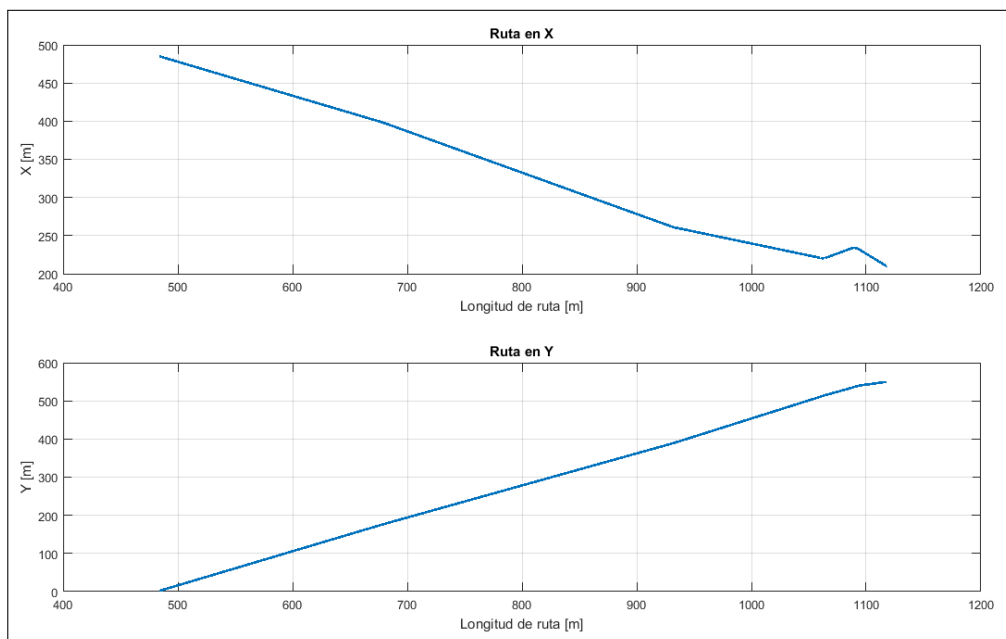


Figura 4.21: Componentes de la ruta paramétrica por espiral de Fermat vs longitud recorrida en el mar Báltico

Tabla 4.1: Longitud de ruta en el mar Báltico

	Polinomio Cúbico de Hermite	Espiral de Fermat
Longitud de Ruta [m]	1121	1118

Con esto vemos claramente que la longitud de ruta desarrollada por el uso del Polinomio de Hermite es mayor a la usada con la espiral de Fermat.

4.2.4. Seguimiento de trayectoria y control de embarcación en el mar Báltico

A partir de los resultados de la sección anterior se obtuvieron puntos de ruta, estos son los que se usarán para la simulación obtener la guía y control de la embarcación.

4.2.4.1. Resultados de simulación en el mar Báltico

Una vez determinada la ruta deseada, se procede a realizar el seguimiento de la misma haciendo uso del sistema de guiado y control, todo esto a partir de los puntos obtenidos, teniendo como resultado la ruta mostrada en la Figura 4.22. El tiempo total de simulación hasta que se llega al punto final es de aproximadamente 650 segundos. Los parámetros considerados en esta simulación vienen dados por dos velocidades deseadas, una cerca a los puertos $u_{d1} = 0.5m/s$ y otra cuando se esta en mar abierto $u_{d2} = 1m/s$, ver Figura 4.25. A partir de estas consideraciones se puede ver claramente en la Figura 4.24, que a los 100 segundos la fuerza τ_1 cambio de un valor generándose un pico de aproximadamente $\tau_1 = 5N$, mientras que a los 510 segundos la fuerza se reduce al valor inicial, esto esta relacionado claramente con el cambio de velocidad que se hace cerca a los puerto de partida y llegada. Todo lo contrario ocurre con el momento τ_3 , que tiene valores cercanos a cero $\tau_3 \approx 0$.

El resultado del sistema de guiado se muestra en la Figura 4.23, de donde se observa la curva deseada obtenida en rojo ψ_d , la curva por parte del sistema de referencia verde, esta sirve para tener las derivadas necesarias a fin de ingresar los valores al sistema de control, Fossen (2011), finalmente se muestra la curva del ángulo obtenido como resultado de la simulación, en negro.

A fin de obtener un seguimiento de trayectoria más preciso se procede a realizar un cambio de velocidad en *surge*, a fin de partir y llegar con baja velocidad, esto se puede mostrar en la Figura 4.25, respecto a las velocidad *sway* se puede ver que su tendencia es cero, esto garantiza que no hay un desplazamiento por parte de la embarcación respecto a la ruta deseada y la velocidad de rotación

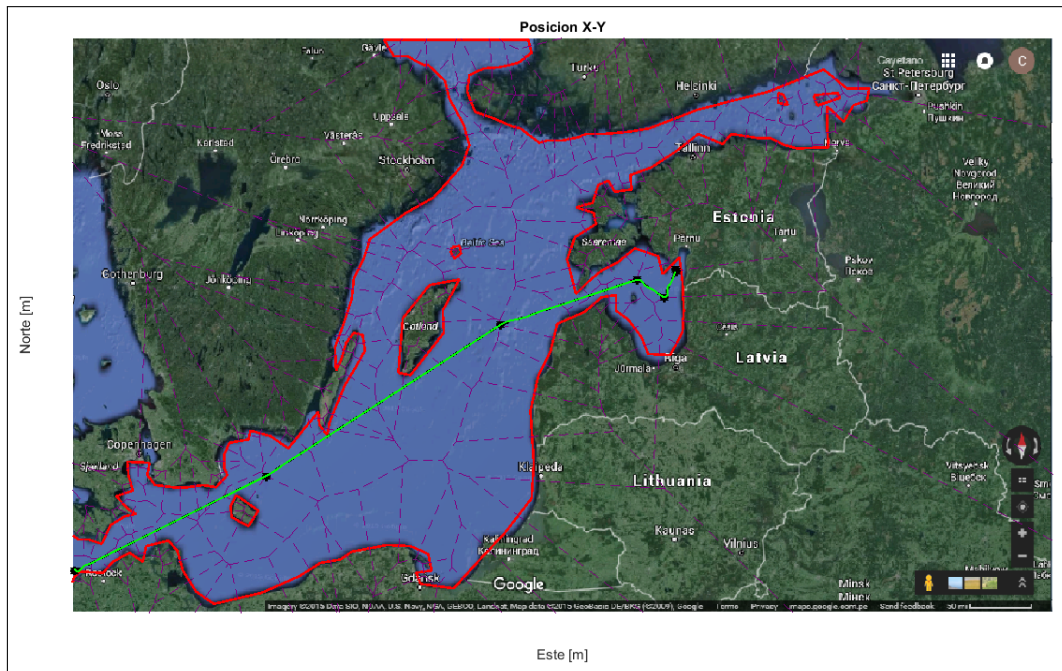


Figura 4.22: Ruta obtenida luego de aplicar el control a la embarcación

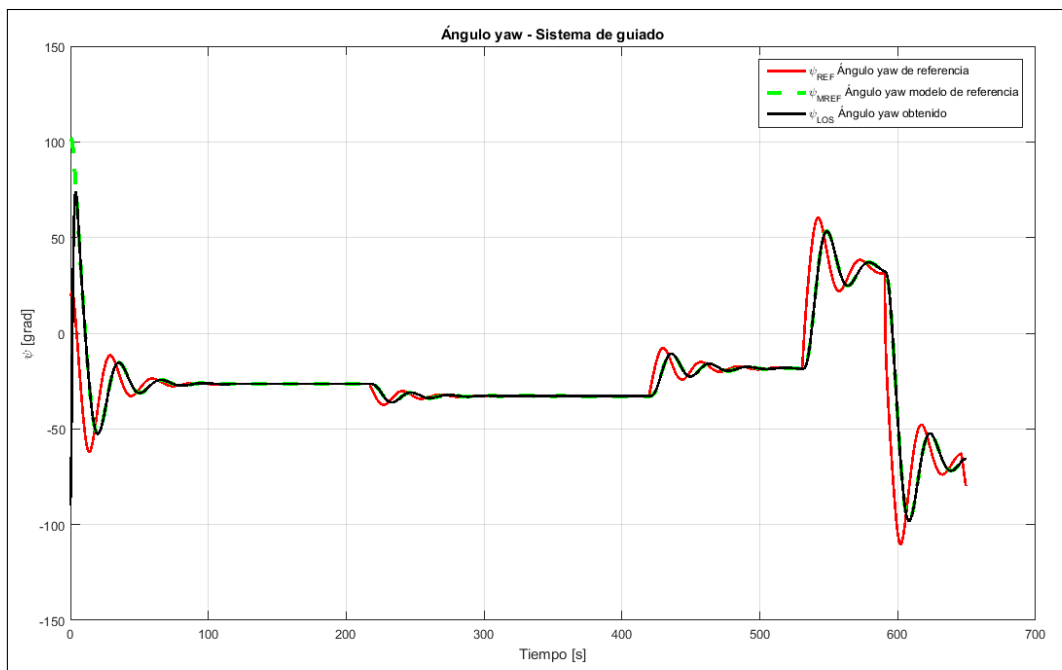


Figura 4.23: Resultado de ángulo de cabeceo deseado a partir del sistema de guía basado en Línea de Visión



Figura 4.24: Ley de control del sistema subactuado, fuerza y torque

yaw, que de igual manera presenta ciertos cambios en los puntos donde hay un cambio de dirección pero de igual manera su valor es prácticamente cero.

En la Figura 4.26, se muestran las variables de error, si analizamos z_1 que es el error del ángulo *yaw* vemos que es prácticamente cero $z_1 \approx 0$ durante toda la simulación, z_{21} que es el error de la velocidad *surge* se puede observar que a lo largo de la simulación el valor es cero $z_{21} \approx 0$, inclusive en los momentos donde ocurre el cambio de velocidad, finalmente z_{22} y z_{23} que son los errores de la velocidad *sway* y velocidad de rotación *yaw* respectivamente, ambos errores tienen valores prácticamente cero, lo que nos dice que no ocurre desplazamiento respecto de la ruta deseada y que el curso del barco siempre es muy cercano al deseado.

4.3. Conclusiones Preliminares

- Acorde a la Tabla 4.1, podemos concluir que el método de polinomio cubico de hermite desarrolla mayor longitud de recorrido para llegar al mismo objetivo, por lo cual se podría concluir que el método de espiral del Fermat es mas eficiente.
- En la simulación en el mar Báltico se realiza un cambio de velocidad *surge* de $u = 0.5m/s$ hasta $u = 1.25m/s$, demostrando lo robusto del control ya que este cambio no afecto el seguimiento de ruta, esto se puede ver claramente en las curvas de variables de error, respecto a la ley de control

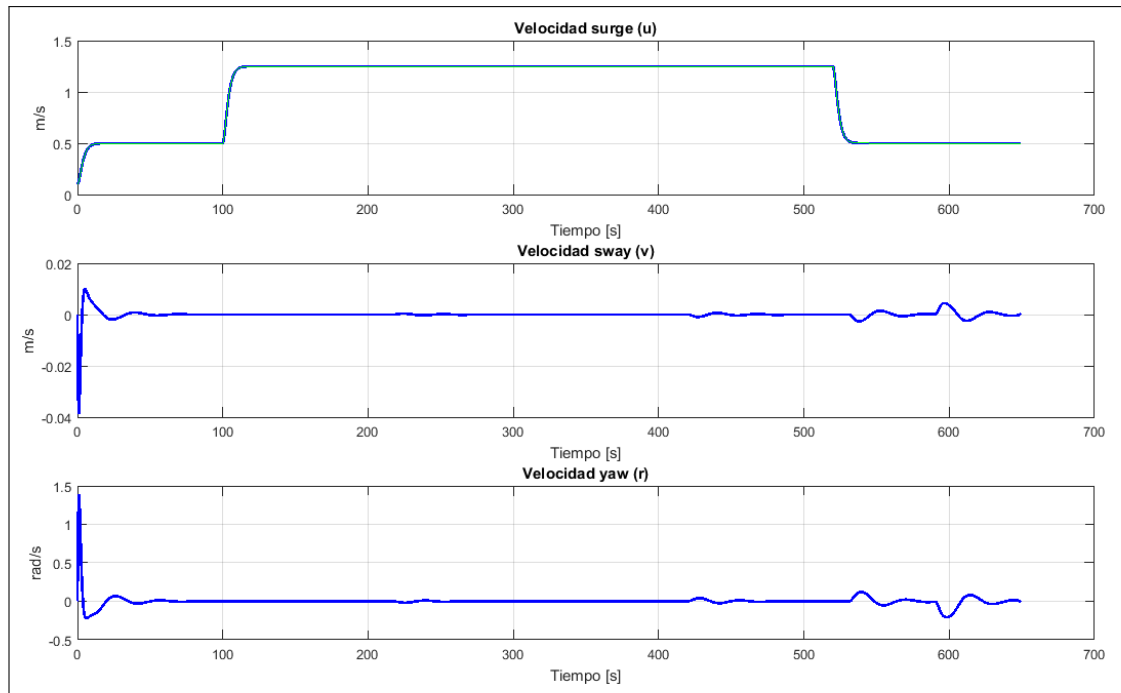


Figura 4.25: Velocidad del sistema body (surge, sway and yaw) durante la simulación

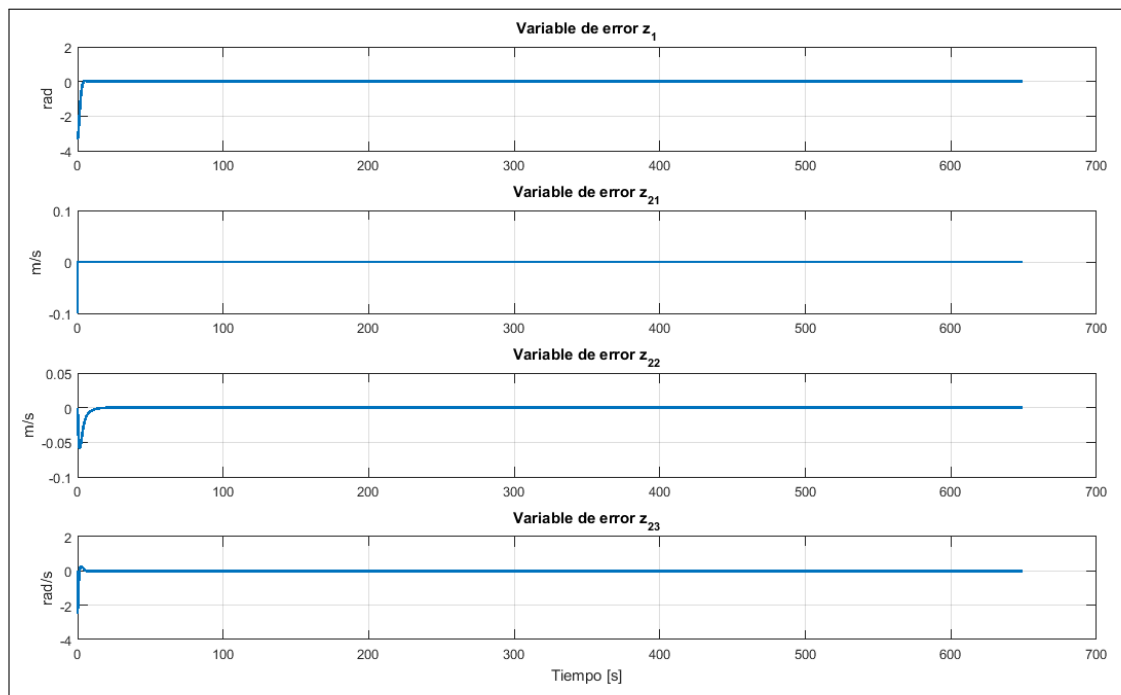


Figura 4.26: Error de estados, cabeceo, velocidad en surge, velocidad en sway y velocidad en yaw

τ_1 vemos que llega a un valor aproximado de $\tau_1 \approx 5N$ cuando se eleva la velocidad y $\tau_1 \approx -1.25N$ cuando la velocidad se reduce.

- En la simulación del mar Báltico los valores mostrados en la Figura 4.26, muestran que los valores de los errores prácticamente son cero en todos los casos, a pesar de haber existido un cambio de velocidad.

Conclusiones

- Se obtuvo un modelo matemático no lineal para desarrollar las simulaciones, en el modelo se consideraron las fuerzas que interactúan con la embarcación, respecto a los parámetros se tomó como referencia el barco CyberShip II.
- Se estudiaron dos métodos diferentes a fin de obtener la ruta partiendo de puntos, concluyendo que el método más eficiente, es el que usa la espiral de Fermat y líneas rectas, para las simulaciones realizadas se pudo comprobar esto, puesto que como resultado siempre genera menor ruta a recorrer.
- Las variables de error, en todas las simulaciones siempre tienen valores ≈ 0 , garantizando un adecuado seguimiento de ruta.
- El sistema de guiado basado en la estrategia línea de visión, muestra un comportamiento adecuado toda vez que la variable de error asociada z_1 siempre tiende a cero $z_1 \approx 0$.
- Se realizó el diseño del controlador aplicando la técnica de “*backstepping*” para el sistema subactuado (3 variables a controlar, 2 variables de control). La ley de control es mostrada a través de τ_1 y τ_3 , respecto al primero que es la fuerza de empuje, se puede concluir que esta directamente relacionada con la velocidad surge, pues al aumentar dicha velocidad desde $u = 0.5m/s$ hasta $u = 1.25m/s$ presenta un sobreimpulso llegando hasta un valor pico de $\tau_{1pico} = 5N$.
- En la Sección 4.2.3, se muestra un cambio de ruta, donde comprobamos que el sistema implementado cambia de inmediato el curso, para esto se ha cambiado el punto de destino y se ha tomado como punto inicial el punto actual.

Recomendaciones

- Considerar el modelo de 6 DOF, a fin de poder evaluar el efecto dinámico que causa navegar por lugares cercanos a la orilla ó de poca profundidad.
- Simular el sistema considerando perturbaciones, como fuerzas debido a corrientes de aire o marinas, analizar el comportamiento de los mismos.
- Implementar el sistema de guiado usando métodos diferentes a línea de visión, a fin de tener otros resultados en ambientes con perturbaciones.
- Simular situaciones donde no se puedan medir todas las variables de estado y sea necesario desarrollar observadores no lineales, implementar y probar su eficacia en alta velocidad y baja velocidad.
- Implementar el sistema desarrollado en un barco autónomo para aplicaciones en aguas peruanas, inclusive se podría pensar realizar transporte autónomo en los ríos de la selva, donde se transporta carga por medios fluviales.
- Desarrollar el control aplicando otras técnicas para probar la eficiencia de la técnica backstepping y ver si se puede mejorar el tiempo de procesamiento.
- Revisar otras técnicas de planeamiento de movimiento, a fin de establecer la más eficiente para la aplicación.
- Investigar otras técnicas para generación de rutas, con el fin de reducir la carga computacional y obtener el mejor comportamiento dinámico del barco.
- Aplicar técnicas de indentificación a fin de obtener los parámetros reales del barco autónomo a implementar.

Bibliografía

- A. Alessandretti, A. P. Aguiar, and C. Jones. Trajectory-tracking and path-following controllers for constrained underactuated vehicles using model predictive control. In *2013 European Control Conference (ECC)*, number EPFL-CONF-199227, pages 1371–1376. IEEE, 2013.
- K. Benedict, M. Gluch, M. Kirchhoff, S. Fischer, S. Klaes, and M. Baldauf. Application of fast time simulation technologies for enhanced ship manoeuvring operation. In *Control Applications in Marine Systems*, pages 79–84, 2010.
- M. Bibuli, W. Caharija, K. Y. Pettersen, G. Bruzzone, M. Caccia, and E. Zereik. Ilos guidance-experiments and tuning. In *World Congress*, volume 19, pages 4209–4214, 2014.
- N. Bowditch. *The american practical navigator*, washington, 1995.
- M. Breivik. *Nonlinear maneuvering control of underactuated ships*, 2003.
- M. Breivik, T. Fossen, et al. Path following for marine surface vessels. In *OCEANS'04. MTS/IEEE TECHNO-OCEAN'04*, volume 4, pages 2282–2289. IEEE, 2004.
- M. Candeloro, A. Lekkas, A. Soerensen, and T. I. Fossen. Continuous curvature path planning using voronoi diagrams and fermat's spirals. In *Control Applications in Marine Systems*, volume 9, pages 132–137, 2013.
- J. Corneliussen. *Implementation of a guidance system for cybership II*. PhD thesis, Master's thesis, NTNU, Trondheim, 2003.
- A. R. Dahl. *Path planning and guidance for marine surface vessels*. 2013.
- K. Fedyaevsky and G. Sobolev. *Control and stability in ship design*. 1964.

- T. I. Fossen. *Guidance and control of ocean vehicles*. John Wiley & Sons Inc, 1994.
- T. I. Fossen. *Marine control systems: guidance, navigation and control of ships, rigs and underwater vehicles*. 2002.
- T. I. Fossen. *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.
- T. I. Fossen, M. Breivik, and R. Skjetne. Line-of-sight path following of underactuated marine craft. *Proceedings of the 6th IFAC MCMC, Girona, Spain*, pages 244–249, 2003.
- Google. Google maps, 2015. URL <https://www.google.com.pe/maps/place/Baltic+Sea/@57.542676,15.8017743,6z/data=!4m2!3m1!1s0x46f4d7d988201b2b:0xb43097ae8474cb3?hl=en&hl=en>. Accedido 17-12-2015.
- J. Haugen and M. Breivik. A speed control algorithm for planar path maneuvering. In *Proc. 8th IFAC Conference on Control Applications in Marine Systems*, pages 219–224, 2010.
- L. Kobylinski. 15. capabilities of ship handling simulators to simulate shallow water, bank and canal effects. *Navigational Systems and Simulators: Marine Navigation and Safety of Sea Transportation*, page 115, 2011.
- H. Korte, O. Koeckritz, and M. Kurowski. Precise maneuver planning for berth-to-berth navigation of modern ships. *at-Automatisierungstechnik*, 63(5):368–379, 2015.
- M. Kurowski, H. Korte, et al. Full-state manoeuvre planning system for marine vehicles. In *Control Applications in Marine Systems*, volume 9, pages 144–149, 2013.
- S. M. LaValle. *Planning algorithms*. Cambridge university press, 2006.
- A. M. Lekkas. *Guidance and path-planning systems for autonomous vehicles*. 2014.
- A. M. Lekkas and T. I. Fossen. Trajectory tracking and ocean current estimation for marine underactuated vehicles. In *Control Applications (CCA), 2014 IEEE Conference on*, pages 905–910. IEEE, 2014.
- A. M. Lekkas, A. R. Dahl, M. Breivik, and T. I. Fossen. Continuous-curvature path generation using fermat. 2013.

- A. M. Lekkas, T. Fossen, et al. Integral los path following for curved paths based on a monotone cubic hermite spline parametrization. *Control Systems Technology, IEEE Transactions on*, 22(6):2287–2301, 2014a.
- A. M. Lekkas, T. Fossen, et al. Minimization of cross-track and along-track errors for path tracking of marine underactuated vehicles. In *Control Conference (ECC), 2014 European*, pages 3004–3010. IEEE, 2014b.
- K.-P. Lindegaard. Acceleration feedback in dynamic positioning. 2003.
- Z. Liu, L. Liu, and J. Zhang. Nonlinear path tracking control of underactuated surface vessel by lssvm-based identification. In *Industrial Electronics Society, IECON 2014-40th Annual Conference of the IEEE*, pages 209–214. IEEE, 2014.
- A. Mironenko. Construction of the program trajectory of maneuvering of the vessel at mooring. *Expluatatsiya morskogo transporta*, 4:25–29, 2009.
- A. Mironenko. The dynamic synthesis of ship’s programmed track. *Expluatatsiya morskogo transporta*, 67:7–11, 2012a.
- A. Mironenko. The new approach to vessel’s program. *International conference on maritime transport*, 5:1235–1244, 2012b.
- A. Mironenko. Methodology of vessel’s route program motion on the basis of sigma function. In *Control Applications in Marine Systems*, volume 9, pages 138–143, 2013.
- S. Moe. Path following of underactuated marine vessels in the presence of ocean currents. 2013.
- M. D. Pedersen and T. I. Fossen. Marine vessel path planning and guidance using potential flow. In *Manoeuvring and Control of Marine Craft*, volume 9, pages 188–193, 2012.
- T. Perez. *Ship motion control: course keeping and roll stabilisation using rudder and fins*. Springer Science & Business Media, 2006.
- N. A. Shneydor. *Missile guidance and pursuit: kinematics, dynamics and control*. Elsevier, 1998.
- R. Skjetne. Ship maneuvering: The past, the present and the future. *Sea Technology*, 2003.
- R. Skjetne. The maneuvering problem. *These de doctorat, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway*, 2005.

- R. Skjetne, T. Fossen, et al. Nonlinear maneuvering and control of ships. In *OCEANS, 2001. MTS/IEEE Conference and Exhibition*, volume 3, pages 1808–1815. IEEE, 2001.
- A. J. Sørensen. Marine control systems propulsion and motion control of ships and ocean structures lecture notes. 2012.
- B. S. Thomas and P. D. Sclavounos. Optimal-control theory applied to ship maneuvering in restricted waters. *Journal of Engineering Mathematics*, 58(1-4):301–315, 2007.
- M. S. Triantafyllou and F. S. Hover. Maneuvering and control of marine vehicles. *Department of Ocean Engineering, Massachusetts Institute of Technology, Cambridge, USA*, 2002.
- M. S. Triantafyllou and F. S. Hover. Maneuvering and control of marine vehicles. *Cambridge, Massachusetts, USA: edited by authors*, 2003.
- A. Wahl and E.-D. Gilles. Optimale regelverfahren zur automatischen bahnführung von binnenschiffen (automatic track-keeping on waterways using optimal control). *at-Automatisierungstechnik/Methoden und Anwendungen der Steuerungs-, Regelungs-und Informationstechnik*, 51(6/2003):255–264, 2003.
- G. Zhang, X. Zhang, and Y. Zheng. Adaptive neural path-following control for underactuated ships in fields of marine practice. *Ocean Engineering*, 104:558–567, 2015.
- A. A. Zizzari, P. Pedone, and G. Indiveri. A path following controller for the dynamic model of a marine surface vessel. In *Intelligent Autonomous Vehicles*, volume 7, pages 157–162, 2010.

Anexo A

Script en MATLAB para control de barco

```

%Programa para control segun la
tesis
clear all, close all, clc
%Parametros del barco
n=1;
%Parametros Disponibles
directamente
L=1.255;
R_k=n*L;%criteria de aceptacion
m=23.8;%Masa
x_g=0.0425;
Iz=1.76;
%Estimados aproximados
X_du=-2;
Y_dv=-10;
Y_dr=0;
N_dv=0;
N_dr=-1;
%Identificados experimentalmente
X_u=-2;
Y_v=-7;
Y_r=-0.1;
N_v=-0.1;
N_r=-0.5;
h=[0,0,1]';
ud=0.25;

%Tiempo
dt=0.001;
ttt=0:dt:450;
tt=0;

%%
%Generacion de trayectoria
A=10;
lambda=15;
bb=0*[-1;1;0];
bb_hat=[0;0;0];
R=5;
x0=0;
y0=0;
theta= 0;
deltatheta_max=3*L;%Coeficiente
de amortiguamiento
deltatheta_min=L;
K_deltatheta=1;
%We=[1 0
%    0 1];
%gamma=10.0;

%%
%Condiciones iniciales de
simulacion
%Condiciones iniciales
x=-10;
y=3;
psi=0.785;

u=0.25;
v=0;
r=0;

nu=[u
    v
    r];
eta=[x
    y
    psi];

alpha_1=ud;
alpha_2=0;
alpha_3=0;

R_psi=[cos(psi) -sin(psi) 0
        sin(psi) cos(psi) 0
        0          0      1];

%%
%Algoritmo Angulo
%psi=0;
accumulation=0;
psi_last=0;
state=0;

%%
%Matrices de modelo de barco
m11=m-X_du;
m22=m-Y_dv;
m23=m*x_g-Y_dr;
m32=m*x_g-Y_dr;
m33=Iz-N_dr;

c13=- (m-Y_dv) *v- (m*x_g-Y_dr) *r;
c23=(m-X_du) *u;

d11=-X_u
d22=-Y_v
d23=-Y_r
d32=-N_v
d33=-N_r

u=0.000;

n11=-X_u;
n22=-Y_v;
n23=m*u-Y_r;
n32=m*u-Y_r;
%n32=-N_v;
n33=m*x_g*u-N_r;

```

```

% u=0.1;

M=[m11 0 0
   0 m22 m23
   0 m32 m33]

C=[0 0 c13
   0 0 c23
  -c13 -c23 0]

D=[d11 0 0
   0 d22 d23
   0 d32 d33]

N_nu=[n11 0 0
       0 n22 n23
       0 n32 n33];

i=1;

%%
%Parametros del modelo de
referencia
%Angulo PSI
wc=1.5;
dddpsi_d=0;
ddpsi_d=0;
dpsi_d=0;
psi_d=0;

%Velocidad surge
wcu=0.5;
ddu_d=0;
du_d=0;
u_d=0.1;

%%
%Parametros del controlador
cc=0.75;
k_1=20;
k_2=20;
k_3=1;

K=diag([k_1;k_2;k_3]);

GAMMA_MTR=0.5*eye(3);

Mz11=-cc;
Mz12=h';
Mz21=-inv(M)*h;
Mz22=-inv(M)*K;

Mz=[Mz11 Mz12
     Mz21 Mz22];

%%
%Bucle de control
for i=1:length(ttt)

    %Angulos implicados
    beta=atan2(v,u);
    betav(i)=beta;

    theta_ast=newtonr2(0,theta,1e-
6,1000,x,y);

    xd_ast=A*sin(theta_ast/lambda);
    yd_ast=theta_ast;

    dxd_ast=(A/lambda)*cos(theta_ast/
lambda);
    dyd_ast=1;

    ddx_d_ast=-
(A/lambda^2)*sin(theta_ast/lambda
);
    ddy_d_ast=0;

    Chip=atan2(dyd_ast,dxd_ast);
    Chipv(i)=Chip;

    %
    R_chip=[cos(Chip) -sin(Chip)
            sin(Chip) cos(Chip)];

    % varepsilon=Rp_chit'*[x-
xd;y-yd];

    Alon_Cros_e = R_chip'*[x-
xd_ast;y-yd_ast];

    % s=varepsilon(1);
    % e=varepsilon(2);
    %
    x_e=Alon_Cros_e(1);
    y_e=Alon_Cros_e(2);

    deltatheta=(deltatheta_max-
deltatheta_min)*exp(-
K_deltatheta*y_e^2)+deltatheta_mi
n;

```

```

%Chir=atan2(-e,deltatheta);
Chir=atan2(-y_e,deltatheta);
Chirv(i)=Chir;

tt=tt+dt;
t(i)=tt;
XYPSI(:,i)=eta;
nuv(:,i)=nu;

psi_los=Chip+Chir-beta;
psi_now=psi_los;
psidv(i)=psi_los;

%%
%REFERENCE MODEL

%Angulo PSI
dddpsi_d=-3*wc*ddpsi_d-
3*wc^2*dpsi_d-
wc^3*psi_d+wc^3*psi_los;
ddpsi_d=ddpsi_d+dt*dddpsi_d;
drd=ddpsi_d;
dpsi_d=dpsi_d+dt*ddpsi_d;
rd=dpsi_d;
psi_d=psi_d+dt*dpsi_d;
psi_dv(i)=psi_d;

%Velocidad surge
ddu_d=-2*wcu*du_d-
wcu^2*u_d+wcu^2*ud;
du_d=du_d+dt*ddu_d;
u_d=u_d+dt*du_d;
u_dv(i)=u_d;

%%
%Mapping angle from <??;??> to
<pi;-pi>
if( sign(psi_d)>0 ) %
sign(psi)==1
n = floor(psi_d/pi);
elseif( sign(psi_d)<0 ) %
sign(psi)==-1
n = ceil(psi_d/pi);
else
n = 0;
end

remainder = rem(n,2);

if( remainder==0 )
psi_mapped = psi_d - n*pi;
else % remainder~=0
if( sign(remainder)>0 ) %
sign(remainder)==1
psi_mapped = psi_d -
(n+1)*pi;
else % sign(remainder)<0,
sign(remainder)==-1
psi_mapped = psi_d - (n-
1)*pi;
end
end

psi_d=psi_mapped;

%%
%Mapping angle from <pi;-pi> to
<0;2pi>
%Para el controlador

if(psi_d<0)
psi_mapped = psi_d + 2*pi;
else
psi_mapped = psi_d;
end

%psi_d=psi_mapped;

%%
%Controlador
z1=psi-psi_d;
z21=u-u_d;
z22=v-alpha_2;
z23=r-alpha_3;

Z=[z1
z21
z22
z23];
Zv(:,i)=Z;

% dZ=Mz*Z;
% Z=Z+dt*dZ;

%Vector de control Alpha
alpha_1=u_d;
alpha_3=-cc*z1+rd;
dalalpha_3=-cc*(r-rd)+drd;

dalalpha_2=(1/m22)*(-
n22*alpha_2+(k_2-n22)*z22-
m23*dalalpha_3-n23*r-

```

```

(bb_hat(1)*sin(psi)-
bb_hat(2)*cos(psi));
alpha_2=alpha_2+dt*dalpha_2;

tau_1= m11*du_d+n11*u-k_1*(u-
u_d)-
(bb_hat(1)*cos(psi)+bb_hat(2)*sin
(psi));
tau_3=
m32*dalpha_2+m33*dalpha_3+n32*v+n
33*r-k_3*(r-alpha_3)-z1-
bb_hat(3);

dbb_hat=GAMMA_MTR*R_psi*Z(2:4,1);
bb_hat=bb_hat+dt*dbb_hat;

tau=[tau_1;0;tau_3];

tauv(:,i)=tau;

%Modelo del barco
dnu=inv(M)*(-
N_nu*nu+tau+R_psi'*bb);
nu=nu+dt*dnu;
%nuv(:,i)=nu;

R_psi=[cos(psi) -sin(psi) 0
        sin(psi) cos(psi) 0
        0          0       1];

deta=R_psi*nu;
eta=eta+dt*deta;

u=nu(1);
v=nu(2);
r=nu(3);

x=eta(1);
y=eta(2);
psi=eta(3);

end

%%
%%Graficos

figure(2)
plot(0:0.1:40*pi,A*sin((0:0.1:40*
pi)/lambda),'--k','LineWidth',4)
hold on
%plot(y_losv,x_losv,'--r'),hold
on
%axis([-7 7 -2 10])

```

```

title('Posición X-Y')
xlabel('Este [m]')
ylabel('Norte [m]')
grid on
plot(XYPSI(2,:),XYPSI(1:,:),'r','L
ineWidth',1.5)
legend('Ruta de referencia','Ruta
obtenida')

%Parametros del barco
La=2;%metros
L1=La/3;%metros
L2=La/6;%metros

nk=length(t);
for k = 1:50000:nk
    x = XYPSI(2,k);    y =
XYPSI(1,k);
    fil = pi-XYPSI(3,k);
    x1 = x - L2*cos(fil);    y1 = y
- L2*sin(fil);
    x2 = x1 + L1*cos(fil);    y2 =
y1 + L1*sin(fil);

    xA = x2 - La/2*sin(fil);
    yA = y2 + La/2*cos(fil);
    xB = x2 + La/2*sin(fil);
    yB = y2 - La/2*cos(fil);
    xC = x1 + La/2*sin(fil);
    yC = y1 - La/2*cos(fil);
    xD = x1 - La/2*sin(fil);
    yD = y1 + La/2*cos(fil);
    xE = x + 1.5*La/2*sin(fil);
    yE = y - 1.5*La/2*cos(fil);

    xcab = [ xA; xB; xE; xC; xD;
xA; ];
    ycab = [ yA; yB; yE; yC; yD;
yA; ];
    %plot(xE,yE,'gx');
    plot(xcab,ycab,'-
b','LineWidth',2)
    %pause(0.5);
    %plot(xcab,ycab,'-
w','LineWidth',2)

end

figure(3)
plot(t,(180/pi)*psidv,'r','LineWi
dth',2),hold on
plot(t,(180/pi)*psi_dv,'--
g','LineWidth',2),hold on
plot(t,(180/pi)*XYPSI(3,:),'k','L
ineWidth',2)

```

```

%plot(t, (180/pi)*Chitv, 'b', 'LineW
idth', 2)
plot(t, (180/pi)*betav, 'm', 'LineWi
dth', 2)
plot(t, (180/pi)*Chirv, 'y', 'LineWi
dth', 2)
plot(t, (180/pi)*Chipv, 'c', 'LineWi
dth', 2)
title('Ángulos de guiado')
%legend('PSI-LOS', 'PSI-LOS
REFERENCE MODEL', 'PSI-
OBTENIDO', 'Chit', 'SideSlip', 'Chir
', 'Chip')
legend('\psi_{REF} Ángulo yaw de
referencia', '\psi_{MREF} Ángulo
yaw con modelo de
referencia', '\psi_{LOS} Ángulo
yaw obtenido', '\beta Ángulo de
deslizamiento', '\chi_r Ángulo
rotación de marco móvil', '\chi_p
Ángulo tangencial paramétrico')
%legend('PSI-LOS REFERENCE
MODEL', 'PSI-
OBTENIDO', 'Chit', 'SideSlip', 'Chir
')
grid on
xlabel('Tiempo [s]')
ylabel('Ángulo [grad]')

figure(4)
subplot(411)
plot(t, Zv(1, :)), grid on
title('Z (z1, z21, z22, z23)')
subplot(412)
plot(t, Zv(2, :)), grid on
subplot(413)
plot(t, Zv(3, :)), grid on
subplot(414)
plot(t, Zv(4, :)), grid on

figure(5)
subplot(211)
plot(t, tauv(1, :)), grid on
title('\tau (tau1, tau3)')
subplot(212)
plot(t, tauv(3, :)), grid on

figure(6)
subplot(311)
plot(t, nuv(1, :)), grid on, hold on
plot(t, u_dv, 'g')
title('\nu (u, v, r)')
subplot(312)
plot(t, nuv(2, :)), grid on
subplot(313)
plot(t, nuv(3, :)), grid on

```

Anexo B

Script en MATLAB para eliminar
puntos no necesarios

```
Q=[Vx(p),Vy(p)];%Puntos X-Y
Pf(k,:)=Pini;%Punto inicial
for i=1:length(p)-3
    Q1=Q(k,:);
    Q2=Q(i+2,:);
    %P=Q(i+1,:);
    %if(norm(Q1-Q2))
    [xrr,yrr]=LineOffset([Q1(1) Q2(1)], [Q1(2) Q2(2)],1);%Tolerancia
    if(polyxpoly(xrr',yrr',x,y,'unique') ~= 0)
        k=k+1;
        plot(xrr,yrr,'c','LineWidth',1)
    else
        Q(k,:)=Q1;
        Q(k+1,:)=Q2;
    end
end
end
Q(k+1,:)=Q2;
Q(k+2,:)=Pfin;%Punto Final
```

Anexo C

Script en MATLAB para generar
Rutas con el Polinomio Cúbico de
Hermite

```

%POLINOMIO CUBICO DE HERMITE
clear all;
close all;
clc;
load('WayPoints')
yh=Wpt(:,1)';
xh=Wpt(:,2)';
nWpt=length(Wpt(:,1));
figure(2)
hold on
plot(yh,xh,'-*k','LineWidth',1.5);
t=xh(1):0.1:xh(length(xh));
PP=pchip(xh,yh,t);
PPs=spline(xh,yh,t);
%Matriz Hermita
MH=[2 -2 1 1
     -3 3 -2 -1
     0 0 1 0
     1 0 0 0];
MHa=[2*eye(2) -2*eye(2) 1*eye(2) 1*eye(2)
     -3*eye(2) 3*eye(2) -2*eye(2) -1*eye(2)
     0*eye(2) 0*eye(2) 1*eye(2) 0*eye(2)
     1*eye(2) 0*eye(2) 0*eye(2) 0*eye(2)];

for k=1:(nWpt-2)
    vin=(Wpt(k+1,:)-Wpt(k,:))/norm(Wpt(k+1,:)-Wpt(k,:),2);
    vinv(k,:)=vin;
    vout=(Wpt(k+2,:)-Wpt(k+1,:))/norm(Wpt(k+2,:)-Wpt(k+1,:),2);
    voutv(k,:)=vout;
end
tannv(1,:)=vinv(1,:);
tannv(nWpt,:)=voutv(nWpt-2,:);
for k=1:(nWpt-2)
    tann=(vinv(k,:)+voutv(k,:))/norm(vinv(k,:)+voutv(k,:),2);
    tannv(k+1,:)=tann;
end
omegav=0:0.001:1;
thetav=0:0.001:(nWpt-1);
inte=0;x1=0;x2=0;y1=0;y2=0;

```

```

for k=1:(nWpt-1)
    coef=MHa*[Wpt(k,:)';Wpt(k+1,:)';tannv(k,:)';...
    tannv(k+1,:)'];
    coefy(k,:)=(MH*[Wpt(k,1)';Wpt(k+1,1)';tannv(k,1)';...
    tannv(k+1,1)']')';
    coefx(k,:)=(MH*[Wpt(k,2)';Wpt(k+1,2)';tannv(k,2)';...
    tannv(k+1,2)']')';
end
for k=1:length(thetav)
    theta=thetav(k);
    if theta==0
        i=1;
    else
        i=ceil(theta);
    end
    omega=theta-(i-1);
    yp(k)=polyval(coefy(i,:),omega);
    xp(k)=polyval(coefx(i,:),omega);
    x2=xp(k);
    y2=yp(k);
    inte=inte+sqrt((x2-x1)^2+(y2-y1)^2);
    intev(k)=inte;
    x1=x2;
    y1=y2;
end
intev=intev-1;
%%
plot(yp,xp,'m','LineWidth',2)
title('Posici3n X-Y')
xlabel('Y [m]');
ylabel('X [m]');
legend('Ruta deseada','Ruta generada')
grid on
axis equal

figure(4)
subplot(211)
plot(intev,xp,'LineWidth',2)
title('Ruta en X')

```

```
ylabel('X [m]');  
xlabel('Longitud de ruta [m]')  
grid on  
subplot(212)  
plot(intev,yp,'LineWidth',2)  
title('Ruta en Y')  
ylabel('Y [m]');  
xlabel('Longitud de ruta [m]')  
grid on
```

Anexo D

Script en MATLAB para generar
Rutas con la espiral de Fermat

```

%Espiral de Fermat
clear all;
close all;
clc;
load('WayPoints')
yh=Wpt(:,1)';
xh=Wpt(:,2)';
nWpt=length(Wpt(:,1));
R1=4;
kappa_max=1/R1;
w_theta=0:0.1:1;
w_theta1=0:0.1:1;
figure(2)
plot(yh,xh,'-k','LineWidth',1.5);
hold on
for k=1:(length(Wpt)-2)
    vin=(Wpt(k+1,:)-Wpt(k,:))/norm(Wpt(k+1,:)-Wpt(k,:),2);
    vinv(k,:)=vin;
    Chi_ini=atan2(vin(1),vin(2));
    Chi_iniv(k)=Chi_ini;
    vout=(Wpt(k+2,:)-Wpt(k+1,:))/norm(Wpt(k+2,:)-Wpt(k+1,:),2);
    voutv(k,:)=vout;
    Chi_end=atan2(vout(1),vout(2));
    Chi_endv(k)=Chi_end;
    delta_Chi=acos(dot(vin,vout));
    delta_Chiv(k)=delta_Chi;
    rho=-sign(vin(1)*vout(2)-vin(2)*vout(1));
    rhov(k)=rho;
    tann=(vin+vout)/norm(vin+vout,2);
    tannv(k,:)=tann;
    tannL=rho*[tann(2) -tann(1)];
    tannLv(k,:)=tannL;
    cent=Wpt(k+1,:)+R1*tannL;
    centv(k,:)=cent;
    %plotcirc(cent(1),cent(2),R1)
    tic
    theta_end=halley(delta_Chi/2,pi/8,1e-5,100);
    toc
    theta_endv(k)=theta_end;

```

```

    theta_kmax=min(theta_end,sqrt(sqrt(7)/2-5/4));
K_param=(1/kappa_max)*2*sqrt(theta_kmax)*(3+4*theta_kmax^2)...
/(1+4*theta_kmax^2)^(3/2);
    K_paramv(k)=K_param;
    alpha=(pi-delta_Chi)/2;
    LL1=K_param*sqrt(theta_end)*cos(theta_end);
    hh=K_param*sqrt(theta_end)*sin(theta_end);
    LL2=hh/tan(alpha);
    LL=LL1+LL2;
    PP1=Wpt(k+1,:)-LL*vin;
    PP2=Wpt(k+1,:)+LL*vout;
    Wptff(3*k-1,:)=PP1;
    Wptff(3*k,:)=PP2;
    Wptff(3*k+1,:)=PP2;
    %plot(PP1(1),PP1(2),'g*')
    theta=theta_end*w_theta;
    theta1=theta_end*w_theta1;
    FS1=[PP1(1)+K_param*sqrt(theta).*sin(rho*theta+Chi_ini);
    PP1(2)+K_param*sqrt(theta).*cos(rho*theta+Chi_ini)];
    %plot(FS1(1,:),FS1(2,),'-k')
    FS2=[PP2(1)-K_param*sqrt(theta_end-theta1).*sin(rho*...
    (theta1-theta_end)+Chi_end);
    PP2(2)-K_param*sqrt(theta_end-theta1).*cos(rho*(theta1...
    -theta_end)+Chi_end)];
    %plot(FS2(1,:),FS2(2,),'-k')
end
nWptff=3*nWpt-4;
Wptff(1,:)=Wpt(1,:);
Wptff(nWptff,:)=Wpt(nWpt,:);
for k=1:(nWptff-1)
    vlinff=(Wptff(k+1,:)-Wptff(k,:));
    vlinffv(k,:)=vlinff;
end
thetav=0:0.001:(nWptff-1);
inte=0;x1=0;x2=0;y1=0;y2=0;
for k=1:length(thetav)
    theta=thetav(k);
    if theta==0
        i=1;

```

```

else
    i=ceil(theta);
end
omega=theta-(i-1);
dpar=(i+2)/3-floor((i+2)/3);
dtri=(i+1)/3-floor((i+1)/3);
if dpar==0
    yp(k)=Wptff(i,1)+omega*vlinffv(i,1);
    xp(k)=Wptff(i,2)+omega*vlinffv(i,2);
elseif dtri==0
    omega=omega*theta_endv((i+1)/3);
    ufe=sqrt(omega);
yp(k)=Wptff(i,1)+K_paramv((i+1)/3)*ufe*sin(rhov((i+1)/3)*ufe^2...
+Chi_iniv((i+1)/3));
xp(k)=Wptff(i,2)+K_paramv((i+1)/3)*ufe*cos(rhov((i+1)/3)*ufe^2...
+Chi_iniv((i+1)/3));
else
    omega=omega*theta_endv(i/3);
yp(k)=Wptff(i,1)-K_paramv(i/3)*sqrt(theta_endv(i/3)-omega)...
*sin(rhov(i/3)*(omega-theta_endv(i/3))+Chi_endv(i/3));
xp(k)=Wptff(i,2)-K_paramv(i/3)*sqrt(theta_endv(i/3)-omega)...
*cos(rhov(i/3)*(omega-theta_endv(i/3))+Chi_endv(i/3));
end
x2=xp(k);
y2=yp(k);
%inte=inte+(x2-x1)*(y1+y2)/2;
inte=inte+sqrt((x2-x1)^2+(y2-y1)^2);
intev(k)=inte;
x1=x2;
y1=y2;
end
intev=intev-1;
plot(yp,xp,'m','LineWidth',2)
%legend('lineas','pchip','spline','cubic hermite')
title('PosiciÃn X-Y')
xlabel('Y [m]');
ylabel('X [m]');
legend('Ruta deseada','Ruta generada')
grid on

```

```
axis equal
figure(4)
subplot(211)
plot(intev,xp,'LineWidth',2)
title('Ruta en X')
ylabel('X [m]');
xlabel('Longitud de ruta [m]')
grid on
subplot(212)
plot(intev,yp,'LineWidth',2)
title('Ruta en Y')
ylabel('Y [m]');
xlabel('Longitud de ruta [m]')
grid on
```

Anexo E

Propuesta de implementación de barco
para monitoreo de variables marinas

E.1. Introducción

En este anexo se analizará una propuesta de implementación, para realizar las pruebas de los algoritmos y darle una aplicación real al sistema, que vendría a ser monitorear las variables marinas del litoral en todo momento (régimen 24/7), como primera fase se realizará las pruebas de funcionamiento en un ambiente controlado, por ejemplo una piscina, por lo cual el alcance de este anexo será únicamente implementar esta primera fase, una fase posterior sería integrar el barco a un sistema satelital para hacer un mapeo de todo el litoral y realizar todo el procesamiento en un computadora dentro de la embarcación. Las ventajas de tener un sistema autónomo, tienen que ver con la seguridad, tener data constante y ahorros económicos, debido a que actualmente para la toma de datos se usan barcos grandes y al menos dos personas conforman la tripulación por embarcación, personal que claramente no puede cumplir con el régimen 24/7. Además por estos tiempos la seguridad en las operaciones ha tomado un gran papel en las decisiones, por lo que frente a un fenómeno que tenga cierto grado de peligrosidad será poco probable enviar un equipo de personas a revisar las áreas en mención, diferente a enviar un robot autónomo, que a lo más puede causar gastos materiales pero no exponer al peligro vidas humanas. Para lograr este objetivo y luego de un análisis económico/técnico se ha seleccionado un barco prototipo de nombre Echo Boat RC (Ver Anexo F). El Echo Boat RC suministrado por la compañía SEAFLOOR, es un barco básico que solo cuenta con los actuadores, por lo que la implementación de los sensores y sistema control para monitoreo serán seleccionados, instalados y probados in-house.

A continuación se muestra una propuesta básica-práctica para la primera fase de implementación del barco autónomo, para control y monitoreo de variables marinas, Echo-Boat RC. (Ver Figura E.1). Los objetivos de la primera fase son los siguientes:

- Dar a conocer la lógica de funcionamiento del sistema.
- Proponer sensores y actuadores necesarios para el trabajo de monitoreo-autónomo.
- Proponer sistemas embebidos capaces de realizar el control y lectura de los sensores.

Para la implementación se establecerá comunicación a bajo nivel entre los sistemas embebidos y el barco de monitoreo. Se tendrá un acceso a los datos a través de la computadora principal, que estará en la estación remota de control.



Figura E.1: Echo boat RC, barco de monitoreo (Seafloor)

E.2. Propuesta de instrumentación

Para realizar la instrumentación del barco de monitoreo, se requiere la identificación de variables que requieren monitorear. En tal sentido el sistema deberá tener los siguientes componentes:

- Un sensor inercial, para determinar la aceleración del sistema.
- Un sensor de viento, para monitorear la velocidad del viento en determinado punto.
- Un sensor de humedad y temperatura.
- Un sensor de presión barométrica.
- Un sensor de PH.
- Controladores embebidos Arduino y Raspberry.
- Una antena Wireless, para la comunicación remota entre el barco y la estación de control. (Xbee)
- Motores.

E.2.1. Selección de sensores/transmisores

Como sabemos la utilidad de los sensores en la embarcación a implementar será para realizar el control de posición y monitoreo de variables marinas, por lo cual se determinó los rangos dentro de los que debía trabajar así como su compatibilidad con las tarjetas embebidas que servirán para realizar el control. Los sensores que se usarán para la implementación serán los siguientes:

E.2.1.1. Sensor de Inercia (IMU)

Este sensor nos servirá para realizar el control del barco de monitoreo. El sensor de inercia, mide aceleración y velocidad angular. Este sensor está compuesto por acelerómetros, giroscopios y magnetómetros. Cabe resaltar que para utilizar este sensor se debe pasar por una etapa de calibración. Para mayores detalles de presenta la ficha técnica en el Anexo G y una foto referencial, Figura E.2.

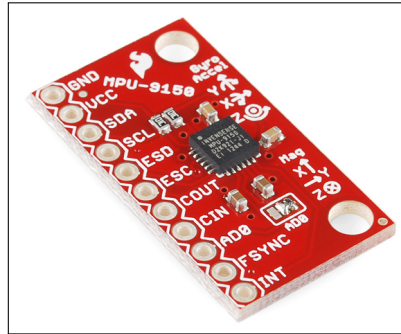


Figura E.2: Foto sensor inercial IMU

E.2.1.2. Sensor de viento

Sensor meteorológico que mide la velocidad y dirección del viento, este será un dato recolectado por el barco. La hoja técnica del sensor seleccionado se puede ver en el Anexo H.

E.2.1.3. Sensor de humedad y temperatura de ambiente

Una de las variables que se deben monitorear es la humedad relativa del área. Para esto se usará el sensor RHT03 que es un sensor de humedad y temperatura de muy bajo costo con un solo cable de interfaz. El sensor viene calibrado por lo cual no es necesario realizar alguna acción adicional. Para mayores detalles se deja la hoja técnica del mismo en el Anexo I y una foto referencial, Figura E.3.

E.2.1.4. Sensor de temperatura del agua

Para realizar esta labor se usará el sensor DS18B20. Este sensor es un sensor que se comunica en forma digital, para lo cual consta de 3 terminales, 2 de alimentación y 1 para los datos. Su interacción con los sistemas embebidos es muy sencilla. Para mayores detalles revisar la hoja técnica que se presenta en el Anexo J y una foto referencial se muestra en la Figura E.4.

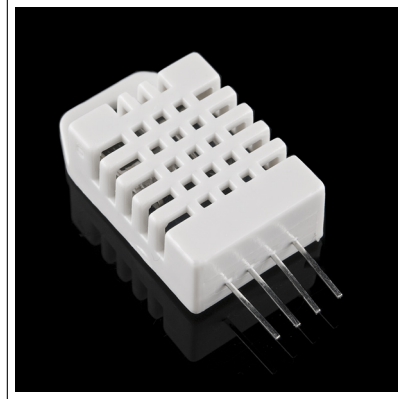


Figura E.3: Sensor de humedad seleccionado RHT03 (Sparkfun)



Figura E.4: Sensor de temperatura DS18B20-Spark Fun

E.2.1.5. Sensor de presión

Otra de las variables que se requieren medir es la presión para lo cual se usara el sensor de presión MPL3115A2, sensor de bajo costo y de alta resolución, la salida del sensor nos muestra el valor de presión en pascales, también nos puede entregar un valor de altura en metros. Además, cuenta con comunicación I2C que hace más fácil su integración con Arduino, se presenta una foto referencial en la Figura E.5 y las principales características se muestran en su hoja técnica mostrada en el Anexo K .

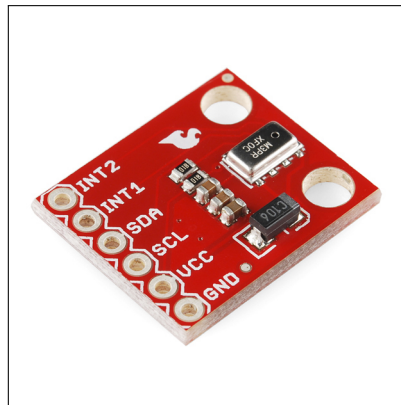


Figura E.5: Sensor de presión MPL3115A2-Sparkfun

E.2.1.6. Sensor de salinidad (Conductividad)

Una de las variables de monitoreo importante es la salinidad del agua, para ello se usará un sensor de conductividad, este sensor nos medirá la conductividad en microSiemens. Siemens es una unidad internacional que nos sirve para medir conductancia que es simplemente la inversa de la resistencia de un elemento, esto puede decirnos mucho de una sustancia en particular. Para poner en funcionamiento este kit, mostrado en la Figura E.6, solo basta con realizar las conexiones indicadas en la hoja de datos que se presenta en el Anexo L.

E.2.1.7. Sensor de PH

Para el registro del PH, se usará un kit, muestra en la Figura E.7, el cual contará con un punta de prueba y soluciones buffer para la calibración del sensor, además se tendrá una tarjeta que brindará la libertad de conectar el sensor con cualquier microcontrolador de forma directa. Para ver las principales características y conexiones, se sugiere de revisar el Anexo M.



Figura E.6: Kit de sensor de conductividad-Sparkfun



Figura E.7: Kit de sensor de PH-Sparkfun

E.2.1.8. Módulo XBee

Es un módulo de comunicación inalámbrica hecha por la empresa Digi. Este módulo es una solución embebida que provee conexión libre de cables hacia dispositivos, usa el protocolo de red IEEE 802.15.4 para comunicación punto-multipunto ó punto a punto. Esta desarrollado para aplicaciones de alto rendimiento que requieren baja latencia y tiempo de comunicación confiable, son ideales para aplicación de bajo consumo o bajo costo, las conexiones con el raspberry se muestran en la Figura E.8 y las principales características y especificaciones son presentadas en el Anexo N

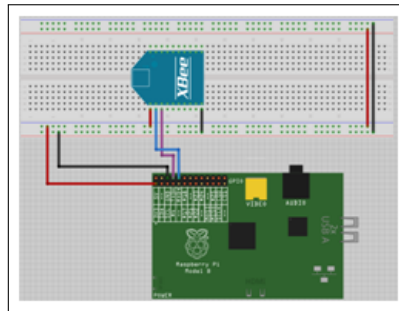


Figura E.8: Conexión típica, Raspberry PI + Xbee

E.2.2. Actuadores

E.2.2.1. Motores y hélices

El barco de monitoreo consta de 2 motores idénticos, los motores serán motores DC sin escobillas, de modelo LBP56110 Æ 640KV, ver Anexo Ñ. Estos motores conectados mediante un eje a hélices servirán como actuadores en el barco de monitoreo, Figura E.9, de acuerdo a la configuración propia estos servirán para que el barco se desplace hacia adelante dándole la misma velocidad a los motores, mientras que para girar el barco se tendrá que asignar velocidades diferentes a los motores. Cabe resaltar que la etapa de potencia viene con los motores, por lo cual lo único que se requiere para activarlos es enviar señal de control del tipo ppm.

E.2.3. Sistema de Control

Respecto al controlador, para este caso al necesitar un sistema de bajo consumo de energía y capaz de realizar el control y adquisición de datos, se usará un computador de placa reducida (Sistema embebido), este será conectado a un sistema de radiocomunicación inalámbrica Xbee para tener comunicación con una computadora maestra que será el centro de control. Para la comunicación de bajo



Figura E.9: Actuadores de barco de monitoreo

nivel que será para los sensores y actuadores se trabajará con la placa de desarrollo Arduino.

A continuación se detallará más sobre los elementos asociados al sistema de control.

E.2.3.1. Raspberry PI

Es un computador de placa reducida, de bajo costo creado con el fin de estimular la enseñanza en ciencias de la computación. Esta placa cuenta con puertos de entrada, salida digitales, puertos usb, además puede soportar comunicación CAN, I2C. Es de fácil programación y por su gran difusión, se pueden encontrar muchos programas y librerías desarrolladas para aplicaciones como comunicación con sensores ó comunicación con las PC, en la Figura E.10, se muestra la versión Raspberry PI B+.

Para nuestra aplicación el Raspberry será conectado a un computador, que hará de estación de remota de control, a través de un Xbee. Además, el Raspberry será conectado a dos arduinos a fin de poder recibir la información de los sensores y enviar acciones con los actuadores a través de comunicación y no de cableado duro, esto con el fin de distribuir las funciones y no sobrecargar el sistema en un solo sistema embebido.

E.2.3.2. Arduino R3 uno

Es una plataforma de hardware libre, que posee un micro-controlador Atmel AVR, que tiene un entorno capaz de conectarse a diferentes sensores o actuadores, y hacer posible desarrollar proyectos electrónicos multidisciplinarios. En

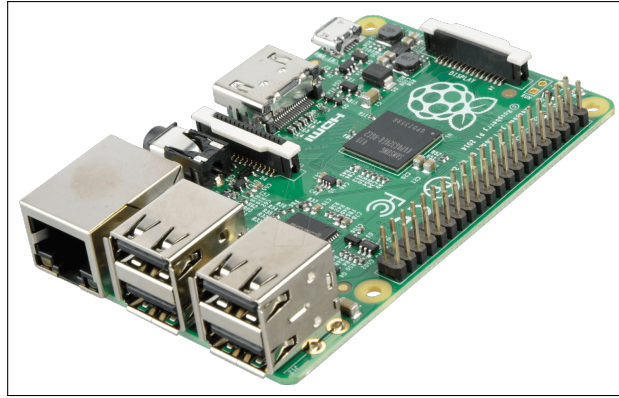


Figura E.10: Tarjeta Raspberry PI +

nuestra aplicación se usarán 2 placas arduino, la primera será para conectarse con los sensores de monitoreo (Temperatura, Presión, IMU, etc.) y poder tomar datos que serán enviados al Raspberry PI, la segunda será para accionar los actuadores (motores) a través de una señal de tipo ppm, que le dirá a los motores la velocidad a la que deben moverse. En la Figura E.11, se puede ver un Arduino 1.



Figura E.11: Placa arduino

E.2.3.3. Computador de monitoreo

Se implementará una estación remota de control. Para enviar y recibir datos desde el controlador Raspberry a través del Xbee. Con este se podrá leer y almacenar los valores de las variables, además de poder tomar acción sobre el barco de monitoreo. Será posible frente a una situación de emergencia pasar a modo manual, y poder maniobrar desde el computador el barco de monitoreo.

E.3. Propuesta de implementación del controlador desarrollado

La implementación del controlador desarrollado se realizará en la tarjeta Raspberry PI, para esto se realizará la programación del controlador no lineal teniendo como entradas las señales adquiridas por el sensor inercial para determinar las aceleraciones y un sistema mediante cámaras para determinar la posición, para lo demás será necesario implementar un observador capaz de estimar la velocidad. Luego de realizar el cálculo a partir de estos valores se procederá a obtener una señal de control que será enviada como referencia a la tarjeta Arduino, está a su vez enviar dicha referencia de velocidad a los motores de tal manera que el barco logre alcanzar los estados adecuados, el esquema se muestra en la Figura E.12.

El sistema de mayor nivel jerárquico, computadora, se encargara de enviar la referencia de posición y velocidad al barco, a través del modulo Xbee.

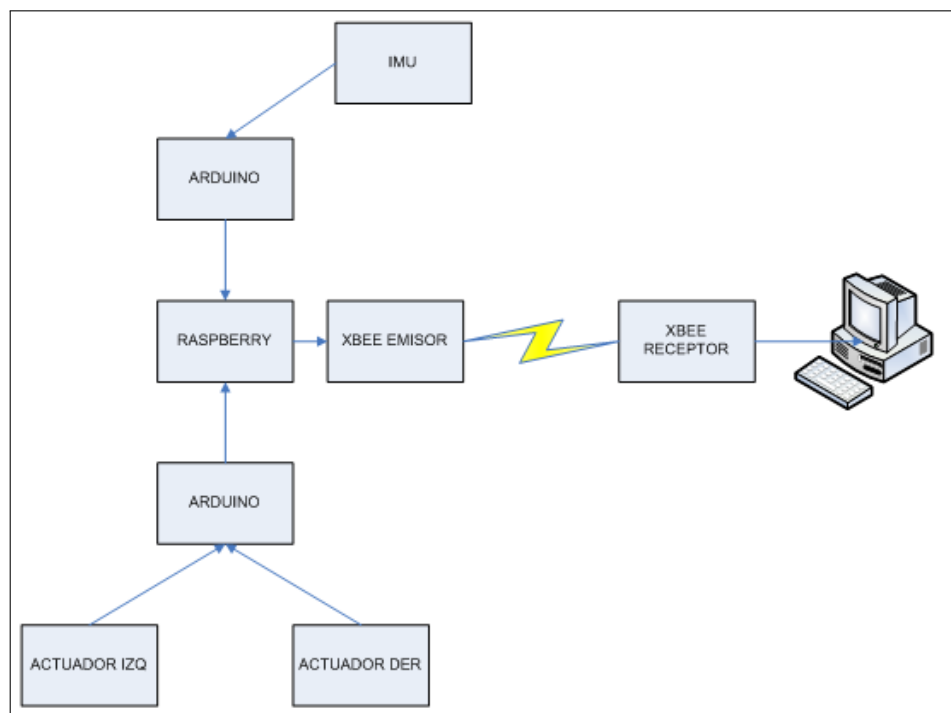


Figura E.12: Sistemas embebidos, sensores y actuadores que son partes del control autónomo del barco

E.4. Propuesta de sistema de control y monitoreo

Para la implementación de sistema de control, se hará uso de los sistemas embebidos Raspberry PI y Arduino, así como de un computador que hará de sistema de monitoreo. La arquitectura que se seguirá será la presentada en la Figura E.13).

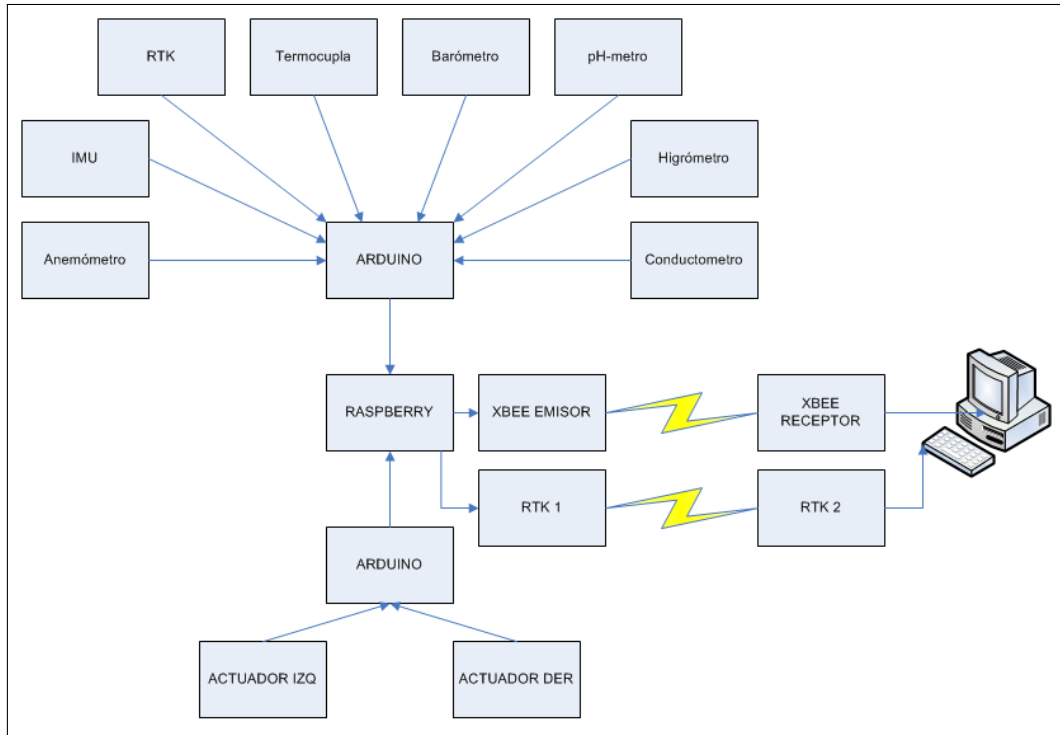


Figura E.13: Arquitectura del sistema de control y monitoreo

E.5. Conclusiones

- Se seleccionaron los sensores necesarios para realizar el control de la embarcación y el monitoreo de las variables marinas deseadas. Se detallaron uno a uno y se presentaron los sistemas embebidos que se usaran para el control y recolección de datos del sistema de monitoreo.
- Para la implementación práctica se propuso sensores y sistemas embebidos de bajo costo y de bajo consumo de energía a fin de hacer más viable la implementación. Además se debe considerar que hay librerías y programas desarrollados para integrar los sensores con los sistemas embebidos uno a uno, lo que se debe hacer es replantear a fin de lograr la conexión del sistema integral.

- Se propone que el monitoreo del barco sea mediante una PC remota, además se debe considerar que el sistema debe constar de una opción para control manual, para poder guiar al barco por si ocurre algún problema con el control autónomo del barco.

Anexo F

Echo Boat RC



Hoja de datos

Beneficios

- Instrumentación personalizada a requisito del cliente.
- Acceso a creas remotas
- Operación llave en mano.

Acerca de

El sistema EchoBoat-RCV es una plataforma de estudios hidrográficos controlada remotamente. El control remoto de la embarcación de levantamiento es sencillo, utilizando la posición GPS y rumbo en tiempo real, visible en la orilla a través de la aplicación de escritorio remoto. El control remoto de alto poder tiene hasta 1 Km de alcance y la embarcación proporciona hasta 8 horas de operación con un solo paquete de baterías.

Para estudios hidrográficos profesionales, el EchoBoat-RCV puede cumplir con las especificaciones particulares del cliente. La embarcación se puede adquirir con la ecosonda preinstalada, o ser preparada para recibir uno de los equipos con los que

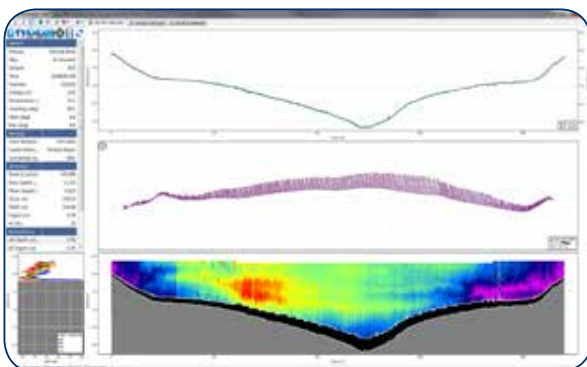
EchoBoat-RCV™



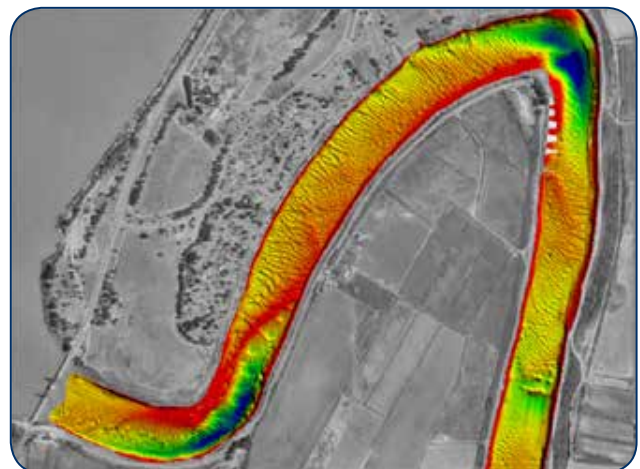
Embarcación EchoBoat levantando en un río.

cuenta el cliente. Asimismo se puede incluir un cableado personalizado, aceptando equipos existentes de GPS, GNSS, y RTK. Para un sistema llave en mano con calidad hidrográfica, es el EchoBoat-RCV puede ser equipado con haz sencillo, multihaz, y sistemas de sonar de barrido lateral.

El EchoBoat-RCV es compatible con software de adquisición de datos hidrográficos como HYPACK, HYDROpro, y OINSY, y utiliza el formato de datos NMEA estándar para interfaz de datos.



Datos de ADCP (Perfilador Doppler) de levantamiento en río.



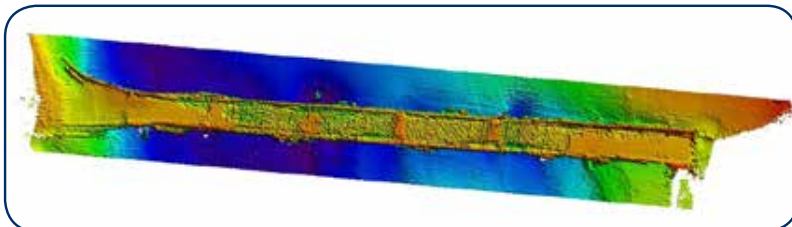
Levantamiento de río con ecosonda multihaz

Especificaciones

Velocidad típica de levantamiento	3 kts (1.5 m/s)
Velocidad máxima	10 kts (5 m/s)
Eslora	180 cm
Manga	90 cm
Duración de la batería (Velocidad máxima)	90 min = aprox 3 millas
Duración de la batería (velocidad de crucero)	240 min = approx 5 millas
Carga útil	40 libras/18 kg
Energía de batería	12v LiFe P04 12v (1)
Motor	DC sin escobillas outdrive
Material de casco de la embarcación	ABS resistente a UV
Peso del casco	23 kg / 46 libras
Herrajes	Acero inoxidable
Control remoto	Futaba 2.4 GHZ
Antena	Omni-direccional
Rango de control remoto.....	1,000 m
GPS	Especificado por el clienteo
Comunicaciones	Radio modem Bluetooth/900 MHZ radio modem de amplio espectro
Montaje del transductor	A través del casco

Instrumentación

SONDA	GPS	AUXILIAR
Ecosonda Multihaz	RTK/GNSS	SVS
Ecosonda de haz sencillo	DGPS/ADCP	SVP
ADCP		CTD
Sonar de barrido lateral		



Levantamiento de ecosonda multihaz de Puente sumergido.



Imagen de sonar de barrido lateral.

Seafloor Systems, Incorporated
 3941 Park Drive, Suite 20-218 · El Dorado Hills, CA 95762 · USA
 (530) 677-1019 · info@seafloorsystems.com · www.seafloorsystems.com

Anexo G

Sensor de Inercia IMU

5 Features

5.1 Gyroscope Features

The triple-axis MEMS gyroscope in the MPU-9150 includes a wide range of features:

- Digital-output X-, Y-, and Z-Axis angular rate sensors (gyroscopes) with a user-programmable full-scale range of ± 250 , ± 500 , ± 1000 , and $\pm 2000^\circ/\text{sec}$
- External sync signal connected to the FSYNC pin supports image, video and GPS synchronization
- Integrated 16-bit ADCs enable simultaneous sampling of gyros
- Enhanced bias and sensitivity temperature stability reduces the need for user calibration
- Improved low-frequency noise performance
- Digitally-programmable low-pass filter
- Factory calibrated sensitivity scale factor
- User self-test

5.2 Accelerometer Features

The triple-axis MEMS accelerometer in MPU-9150 includes a wide range of features:

- Digital-output 3-Axis accelerometer with a programmable full scale range of $\pm 2g$, $\pm 4g$, $\pm 8g$ and $\pm 16g$
- Integrated 16-bit ADCs enable simultaneous sampling of accelerometers while requiring no external multiplexer
- Orientation detection and signaling
- Tap detection
- User-programmable interrupts
- High-G interrupt
- User self-test

5.3 Magnetometer Features

The triple-axis MEMS magnetometer in MPU-9150 includes a wide range of features:

- 3-axis silicon monolithic Hall-effect magnetic sensor with magnetic concentrator
- Wide dynamic measurement range and high resolution with lower current consumption.
- Output data resolution is 13 bit ($0.3 \mu\text{T}$ per LSB)
- Full scale measurement range is $\pm 1200 \mu\text{T}$
- Self-test function with internal magnetic source to confirm magnetic sensor operation on end products

5.4 Additional Features

The MPU-9150 includes the following additional features:

- 9-Axis MotionFusion via on-chip Digital Motion Processor (DMP)
- Auxiliary master I²C bus for reading data from external sensors (e.g., pressure sensor)
- Flexible VLOGIC reference voltage supports multiple I²C interface voltages
- Smallest and thinnest package for portable devices: 4x4x1mm LGA
- Minimal cross-axis sensitivity between the accelerometer, gyroscope and magnetometer axes
- 1024 byte FIFO buffer reduces power consumption by allowing host processor to read the data in bursts and then go into a low-power mode as the MPU collects more data
- Digital-output temperature sensor
- User-programmable digital filters for gyroscope, accelerometer, and temp sensor
- 10,000 g shock tolerant



- 400kHz Fast Mode I²C for communicating with all registers
- MEMS structure hermetically sealed and bonded at wafer level
- RoHS and Green compliant

5.5 MotionProcessing

- Internal Digital Motion Processing™ (DMP™) engine supports 3D MotionProcessing and gesture recognition algorithms
- The MPU-9150 collects gyroscope, accelerometer and magnetometer data while synchronizing data sampling at a user defined rate. The total dataset obtained by the MPU-9150 includes 3-Axis gyroscope data, 3-Axis accelerometer data, 3-Axis magnetometer data, and temperature data.
- The FIFO buffers the complete data set, reducing timing requirements on the system processor by allowing the processor burst read the FIFO data. After burst reading the FIFO data, the system processor can save power by entering a low-power sleep mode while the MPU collects more data.
- Programmable interrupt supports features such as gesture recognition, panning, zooming, scrolling, tap detection, and shake detection
- Digitally-programmable low-pass filters.
- Low-power pedometer functionality allows the host processor to sleep while the DMP maintains the step count.

5.6 Clocking

- On-chip timing generator $\pm 1\%$ frequency variation over full temperature range
- Optional external clock inputs of 32.768kHz or 19.2MHz



6 Electrical Characteristics

6.1 Gyroscope Specifications

Typical Operating Circuit of Section 7.2, VDD = 2.375V-3.465V, VLOGIC= 1.8V±5% or VDD, T_A = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
GYROSCOPE SENSITIVITY						
Full-Scale Range	FS_SEL=0 FS_SEL=1 FS_SEL=2 FS_SEL=3		±250 ±500 ±1000 ±2000		°/s °/s °/s °/s	
Gyroscope ADC Word Length			16		bits	
Sensitivity Scale Factor	FS_SEL=0 FS_SEL=1 FS_SEL=2 FS_SEL=3		131 65.5 32.8 16.4		LSB/(°/s) LSB/(°/s) LSB/(°/s) LSB/(°/s)	
Sensitivity Scale Factor Tolerance	25°C	-3		+3	%	
Sensitivity Scale Factor Variation Over Temperature	-40°C to +85°C		±0.04		%/°C	
Nonlinearity	Best fit straight line; 25°C		0.2		%	
Cross-Axis Sensitivity			±2		%	
GYROSCOPE ZERO-RATE OUTPUT (ZRO)						
Initial ZRO Tolerance	Component level (25°C)		±20		°/s	
ZRO Variation Over Temperature	-40°C to +85°C		±20		°/s	
GYROSCOPE NOISE PERFORMANCE	FS_SEL=0					
Total RMS Noise	DLPFCFG=2 (92Hz)		0.06		°/s-rms	
Rate Noise Spectral Density	At 10Hz		0.005		°/s/√Hz	
GYROSCOPE MECHANICAL FREQUENCIES						
X-Axis		30	33	36	kHz	
Y-Axis		27	30	33	kHz	
Z-Axis		24	27	30	kHz	
LOW PASS FILTER RESPONSE						
	Programmable Range	5		256	Hz	
OUTPUT DATA RATE						
	Programmable	4		8,000	Hz	
GYROSCOPE START-UP TIME						
ZRO Settling	DLPFCFG=0 to ±1% of Final		30		ms	



6.2 Accelerometer Specifications

Typical Operating Circuit of Section 7.2, VDD = 2.375V-3.465V, VLOGIC= 1.8V±5% or VDD, T_A = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
ACCELEROMETER SENSITIVITY						
Full-Scale Range	AFS_SEL=0 AFS_SEL=1 AFS_SEL=2 AFS_SEL=3		±2 ±4 ±8 ±16		g g g g	
ADC Word Length	Output in two's complement format		16		bits	
Sensitivity Scale Factor	AFS_SEL=0 AFS_SEL=1 AFS_SEL=2 AFS_SEL=3		16,384 8,192 4,096 2,048		LSB/g LSB/g LSB/g LSB/g	
Initial Calibration Tolerance			±3		%	
Sensitivity Change vs. Temperature	AFS_SEL=0, -40°C to +85°C		±0.02		%/°C	
Nonlinearity	Best Fit Straight Line		0.5		%	
ZERO-G OUTPUT						
Initial Calibration Tolerance	X and Y axes Z axis		±80 ±150		mg mg	
Change over specified temperature – Component level -25°C to 85°C	X & Y Axis Z Axis		±0.75 ±1.50		mg/°C mg/°C	
NOISE PERFORMANCE						
Power Spectral Density	X, Y & Z Axes, @10Hz, AFS_SEL=0 & ODR=1kHz		400		μg/√Hz	
Total RMS Noise	AFS = 0 @100Hz		4		mg-rms	
LOW PASS FILTER RESPONSE						
	Programmable Range	5		260	Hz	
OUTPUT DATA RATE						
	Programmable Range	4		1,000	Hz	
INTELLIGENCE FUNCTION INCREMENT			32		mg/LSB	

Anexo H

Sensor de Viento



Weather Sensor Assembly p/n 80422

Imported by Argent Data Systems

Usage Notes

This kit includes a wind vane, cup anemometer, and tipping bucket rain gauge, with associated mounting hardware. These sensors contain no active electronics, instead using sealed magnetic reed switches and magnets to take measurements. A voltage must be supplied to each instrument to produce an output.

Assembly

The wind sensor arm mounts on top of the two-piece metal mast and supports the wind vane and anemometer. A short cable connects the two wind sensors. Plastic clips on the underside of the arm hold this cable in place. Screws are provided to secure the sensors to the arm.

The rain gauge may be mounted lower on the mast using its own mounting arm and screw, or it may be mounted independently.

Rain Gauge

The rain gauge is a self-emptying tipping bucket type. Each 0.011" (0.2794 mm) of rain causes one momentary contact closure that can be recorded with a digital counter or microcontroller interrupt input. The gauge's switch is connected to the two center conductors of the attached RJ11-terminated cable.

Anemometer

The cup-type anemometer measures wind speed by closing a contact as a magnet moves past a switch. A wind speed of 1.492 MPH (2.4 km/h) causes the switch to close once per second.

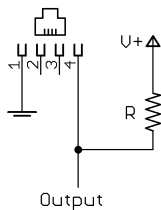
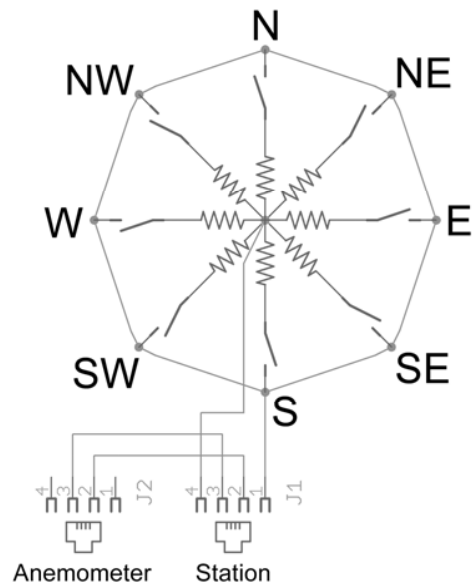
The anemometer switch is connected to the inner two conductors of the RJ11 cable shared by the anemometer and wind vane (pins 2 and 3.)

Wind Vane

The wind vane is the most complicated of the three sensors. It has eight switches, each connected to a different resistor. The vane's magnet may close two switches at once, allowing up to 16 different positions to be indicated. An external resistor can be used to form a voltage divider, producing a voltage output that can be measured with an analog to digital converter, as shown below.

The switch and resistor arrangement is shown in the diagram to the right. Resistance values for all 16 possible positions are given in the table.

Resistance values for positions between those shown in the diagram are the result of two adjacent resistors connected in parallel when the vane's magnet activates two switches simultaneously.



Example wind vane interface circuit. Voltage readings for a 5 volt supply and a resistor value of 10k ohms are given in the table.

Direction (Degrees)	Resistance (Ohms)	Voltage (V=5v, R=10k)
0	33k	3.84v
22.5	6.57k	1.98v
45	8.2k	2.25v
67.5	891	0.41v
90	1k	0.45v
112.5	688	0.32v
135	2.2k	0.90v
157.5	1.41k	0.62v
180	3.9k	1.40v
202.5	3.14k	1.19v
225	16k	3.08v
247.5	14.12k	2.93v
270	120k	4.62v
292.5	42.12k	4.04v
315	64.9k	4.78v
337.5	21.88k	3.43v

Anexo I

Sensor de Humedad y temperatura de ambiente



Digital relative humidity & temperature sensor RHT03

1. Feature & Application:

- *High precision
- *Capacitive type
- *Full range temperature compensated
- *Relative humidity and temperature measurement
- *Calibrated digital signal
- *Outstanding long-term stability
- *Extra components not needed
- *Long transmission distance, up to 100 meters
- *Low power consumption
- *4 pins packaged and fully interchangeable

2. Description:

RHT03 output calibrated digital signal. It applies exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements are connected with 8-bit single-chip computer.

Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory.

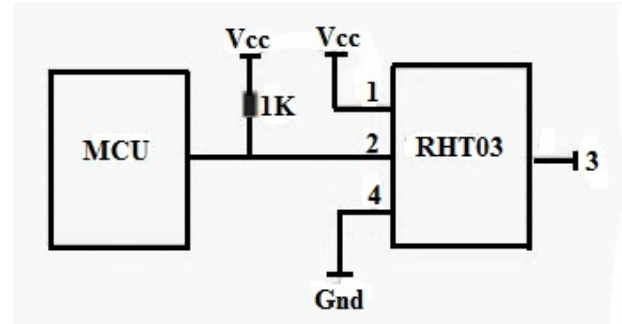
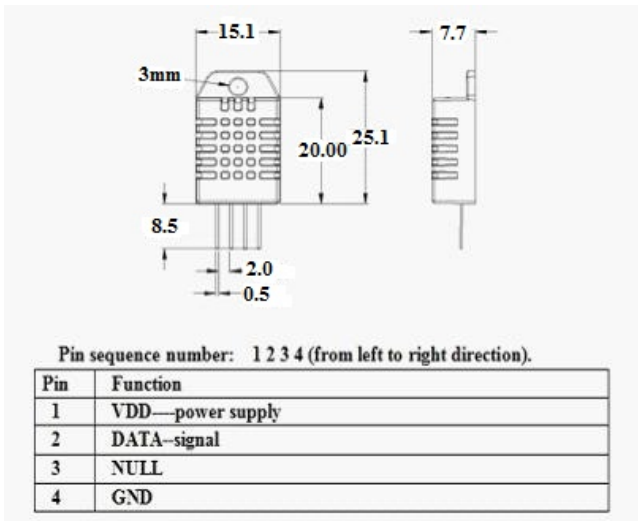
Small size & low consumption & long transmission distance(100m) enable RHT03 to be suited in all kinds of harsh application occasions. Single-row packaged with four pins, making the connection very convenient.

3. Technical Specification:

Model	RHT03	
Power supply	3.3-6V DC	
Output signal	digital signal via MaxDetect 1-wire bus	
Sensing element	Polymer humidity capacitor	
Operating range	humidity 0-100%RH;	temperature -40~80Celsius
Accuracy	humidity +2%RH (Max +5%RH);	temperature +0.5Celsius
Resolution or sensitivity	humidity 0.1%RH;	temperature 0.1Celsius
Repeatability	humidity +-1%RH;	temperature +-0.2Celsius

Humidity hysteresis	+/-0.3%RH
Long-term Stability	+/-0.5%RH/year
Interchangeability	fully interchangeable

4. Dimensions: (unit----mm)



5. Electrical connection diagram:

6. Operating specifications:

(1) Power and Pins

Power's voltage should be 3.3-6V DC. When power is supplied to sensor, don't send any instruction to the sensor within one second to pass unstable status. One capacitor valued 100nF can be added between VDD and GND for wave filtering.

(2) Communication and signal

MaxDetect 1-wire bus is used for communication between MCU and RHT03. (MaxDetect 1-wire bus is specially designed by MaxDetect Technology Co., Ltd. , it's different from Maxim/Dallas 1-wire bus, so it's incompatible with Dallas 1-wire bus.)

Illustration of MaxDetect 1-wire bus:

Data is comprised of integral and decimal part, the following is the formula for data.

DATA=8 bit integral RH data+8 bit decimal RH data+8 bit integral T data+8 bit decimal T data+8 bit check-sum

Anexo J

Sensor de temperatura del agua



DS18B20

Programmable Resolution 1-Wire[®] Digital Thermometer

www.dalsemi.com

FEATURES

- Unique 1-Wire interface requires only one port pin for communication
- Multidrop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line. Power supply range is 3.0V to 5.5V
- Zero standby power required
- Measures temperatures from -55°C to +125°C. Fahrenheit equivalent is -67°F to +257°F
- $\pm 0.5^\circ\text{C}$ accuracy from -10°C to +85°C
- Thermometer resolution is programmable from 9 to 12 bits
- Converts 12-bit temperature to digital word in 750 ms (max.)
- User-definable, nonvolatile temperature alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

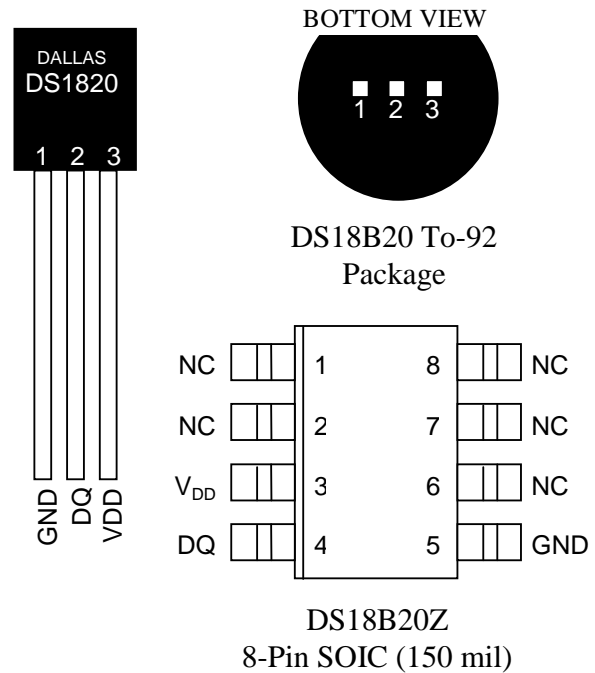
DESCRIPTION

The DS18B20 Digital Thermometer provides 9 to 12-bit (configurable) temperature readings which indicate the temperature of the device.

Information is sent to/from the DS18B20 over a 1-Wire interface, so that only one wire (and ground) needs to be connected from a central microprocessor to a DS18B20. Power for reading, writing, and performing temperature conversions can be derived from the data line itself with no need for an external power source.

Because each DS18B20 contains a unique silicon serial number, multiple DS18B20s can exist on the same 1-Wire bus. This allows for placing temperature sensors in many different places. Applications where this feature is useful include HVAC environmental controls, sensing temperatures inside buildings, equipment or machinery, and process monitoring and control.

PIN ASSIGNMENT



PIN DESCRIPTION

- GND - Ground
 DQ - Data In/Out
 V_{DD} - Power Supply Voltage
 NC - No Connect

DETAILED PIN DESCRIPTION Table 1

PIN 8PIN SOIC	PIN TO92	SYMBOL	DESCRIPTION
5	1	GND	Ground.
4	2	DQ	Data Input/Output pin. For 1-Wire operation: Open drain. (See “Parasite Power” section.)
3	3	V _{DD}	Optional V_{DD} pin. See “Parasite Power” section for details of connection. V _{DD} must be grounded for operation in parasite power mode.

DS18B20Z (8-pin SOIC): All pins not specified in this table are not to be connected.

OVERVIEW

The block diagram of Figure 1 shows the major components of the DS18B20. The DS18B20 has four main data components: 1) 64-bit lasered ROM, 2) temperature sensor, 3) nonvolatile temperature alarm triggers TH and TL, and 4) a configuration register. The device derives its power from the 1-Wire communication line by storing energy on an internal capacitor during periods of time when the signal line is high and continues to operate off this power source during the low times of the 1-Wire line until it returns high to replenish the parasite (capacitor) supply. As an alternative, the DS18B20 may also be powered from an external 3 volt - 5.5 volt supply.

Communication to the DS18B20 is via a 1-Wire port. With the 1-Wire port, the memory and control functions will not be available before the ROM function protocol has been established. The master must first provide one of five ROM function commands: 1) Read ROM, 2) Match ROM, 3) Search ROM, 4) Skip ROM, or 5) Alarm Search. These commands operate on the 64-bit lasered ROM portion of each device and can single out a specific device if many are present on the 1-Wire line as well as indicate to the bus master how many and what types of devices are present. After a ROM function sequence has been successfully executed, the memory and control functions are accessible and the master may then provide any one of the six memory and control function commands.

One control function command instructs the DS18B20 to perform a temperature measurement. The result of this measurement will be placed in the DS18B20's scratch-pad memory, and may be read by issuing a memory function command which reads the contents of the scratchpad memory. The temperature alarm triggers TH and TL consist of 1 byte EEPROM each. If the alarm search command is not applied to the DS18B20, these registers may be used as general purpose user memory. The scratchpad also contains a configuration byte to set the desired resolution of the temperature to digital conversion. Writing TH, TL, and the configuration byte is done using a memory function command. Read access to these registers is through the scratchpad. All data is read and written least significant bit first.

Anexo K

Sensor de Presión

I²C Precision Altimeter

The MPL3115A2 employs a MEMS pressure sensor with an I²C interface to provide accurate Pressure/Altitude and Temperature data. The sensor outputs are digitized by a high resolution 24-bit ADC. Internal processing removes compensation tasks from the host MCU system. Multiple user-programmable, power saving, interrupt and autonomous data acquisition modes are available, including programmed acquisition cycle timing, and poll-only modes. Typical active supply current is 40 μ A per measurement-second for a stable 30 cm output resolution. Pressure output can be resolved with output in fractions of a Pascal, and Altitude can be resolved in fractions of a meter.

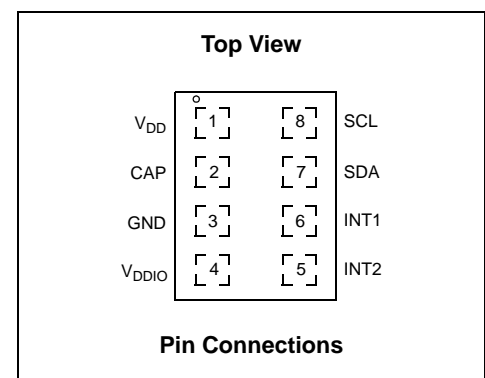
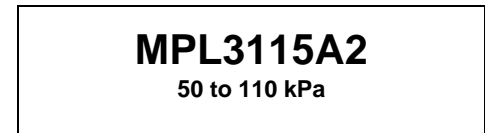
The MPL3115A2 is offered in a 5 mm by 3 mm by 1.1 mm LGA package and specified for operation from -40°C to 85°C. Package is surface mount with a stainless steel lid and is RoHS compliant.

Features

- 1.95V to 3.6V Supply Voltage, internally regulated by LDO
- 1.6V to 3.6V Digital Interface Supply Voltage
- Fully Compensated internally
- Direct Reading, Compensated
 - Pressure: 20-bit measurement (Pascals)
 - Altitude: 20-bit measurement (meters)
 - Temperature: 12-bit measurement (degrees Celsius)
- Programmable Events
- Autonomous Data Acquisition
- Resolution down to 1 ft. / 30 cm
- 32 Sample FIFO
- Ability to log data up to 12 days using the FIFO
- 1 second to 9 hour data acquisition rate
- I²C digital output interface (operates up to 400 kHz)

Application Examples

- High Accuracy Altimetry
- Smartphones/Tablets
- Personal Electronics Altimetry
- GPS Dead Reckoning
- GPS Enhancement for Emergency Services
- Map Assist, Navigation
- Weather Station Equipment



ORDERING INFORMATION									
Device Name	Package Options	Case No.	# of Ports			Pressure Type			Digital Interface
			None	Single	Dual	Gauge	Differential	Absolute	
MPL3115A2	Tray	2153	•					•	•
MPL3115A2R1	Tape & Reel (1000)	2153	•					•	•

This document contains information on a new product. Specifications and information herein are subject to change without notice.

© 2011-2012 Freescale Semiconductor, Inc. All rights reserved.

Anexo L

Sensor de Salinidad

EZOTM class embedded electrical conductivity circuit

v1.6

This is an evolving document
check back for updates.

Features

Reads

- Conductivity
- Total dissolved solids
- Practical salinity units
- Specific gravity of sea water

- E.C. readings +/- 2 $\mu\text{s}/\text{cm}$
- Full E.C. range from 0.55 $\mu\text{s}/\text{cm}$ to 500,000+ $\mu\text{s}/\text{cm}$
- Temperature dependent or temperature independent readings
- Flexible calibration protocol supports single point or dual point calibration
- Calibrate to any E.C. value

Two data protocols

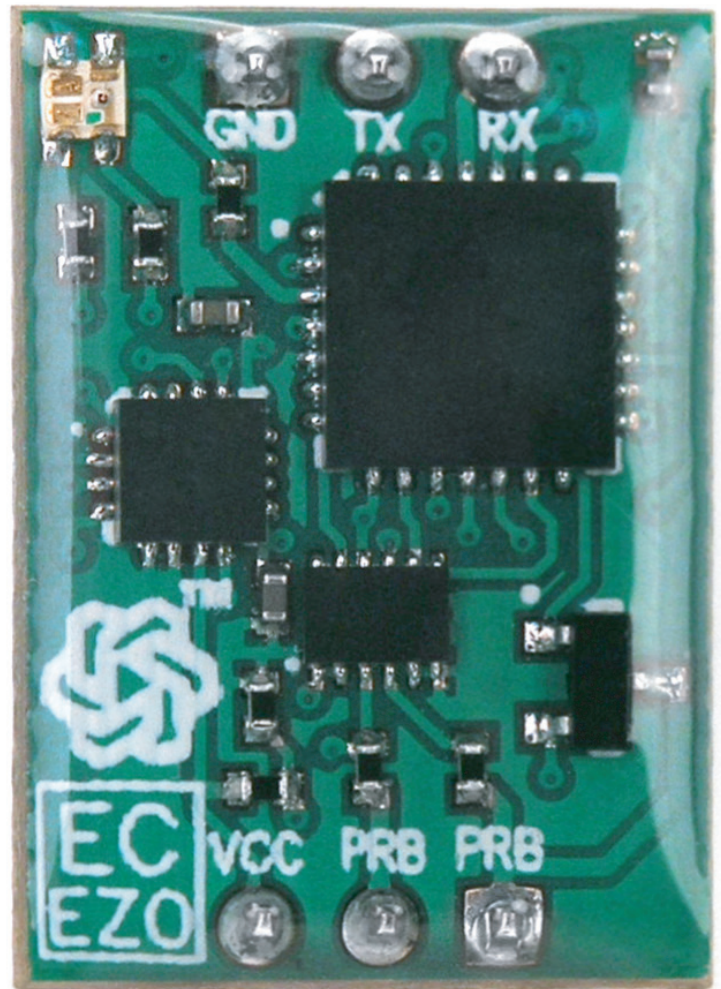
- UART asynchronous serial connectivity (RX/TX voltage swing 0-VCC)
- I2C (default I2C address 0x64)

- Operating voltage: 3.3V to 5V

- Works with any off-the-shelf two conductor conductivity probe
- Works with any K value from K=0.1 to K=10

Sleep mode power consumption

- 0.4mA* at 3.3V



Patent pending

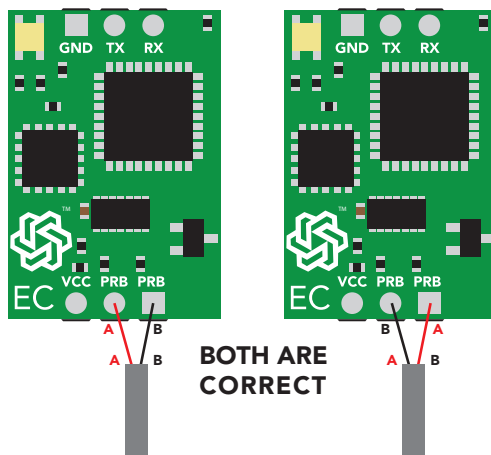
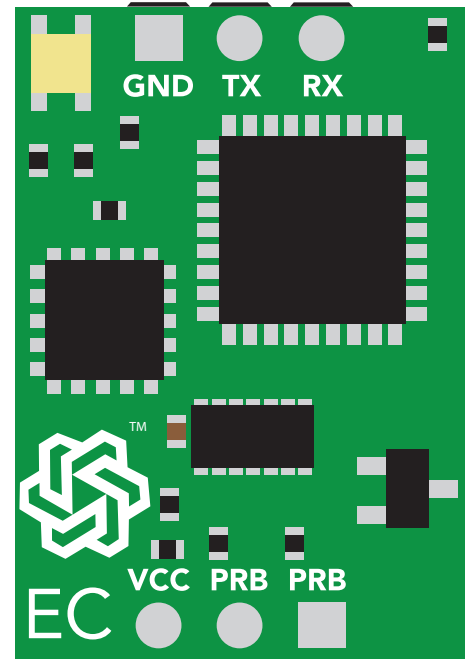


Description

The Atlas ScientificTM EZOTM class embedded electrical conductivity circuit is our 6th generation embedded electrical conductivity circuit. The EC-EZOTM electrical conductivity circuit can work with any off-the-shelf two conductor conductivity probe from K=0.1 to K=10. This includes any value between K=0.1 and K=10 such as K=0.66 or K=4.78. **This device reads electrical conductivity from an E.C. probe/sensor/electrode. This device does not include an E.C. probe/sensor/electrode.**

Pin out

- GND** Return for the DC power supply
- Vcc** Operates on 3.3V – 5.5V
- TX / SDA** All EZO™ class circuits can operate in either UART mode or I²C mode
- The default state is UART mode.
In UART mode, this pin acts as the transmit (TX) line.
The default baud rate is 38400, 8 bits, no parity, no flow control, one stop bit. If standard RS232 voltage levels are desired, connect an RS232 converter such as a MAX232.
If the device is in I²C mode, this pin acts as the Serial Data Line (SDA). The I²C protocol requires an external pull up resistor on the SDA line (resistor not included).
- RX / SCL** All EZO™ class circuits can operate in either UART mode, or I²C mode.
- The default state is UART mode.
In UART mode, this pin acts as the receive (RX) line.
If the device is in I²C mode, this pin acts as the Serial Clock Line (SCL). The I²C protocol requires an external pull up resistor on the SCL line (resistor not included).
- PRB** Two pins are marked PRB. These pins are to be connected to an E.C. probe. It makes **no difference** which lead of the E.C. probe is connected to the two probe pins.



Anexo M

Sensor de PH

EZOTM class embedded pH circuit

V 1.3

This is an evolving document
check back for updates.

Features

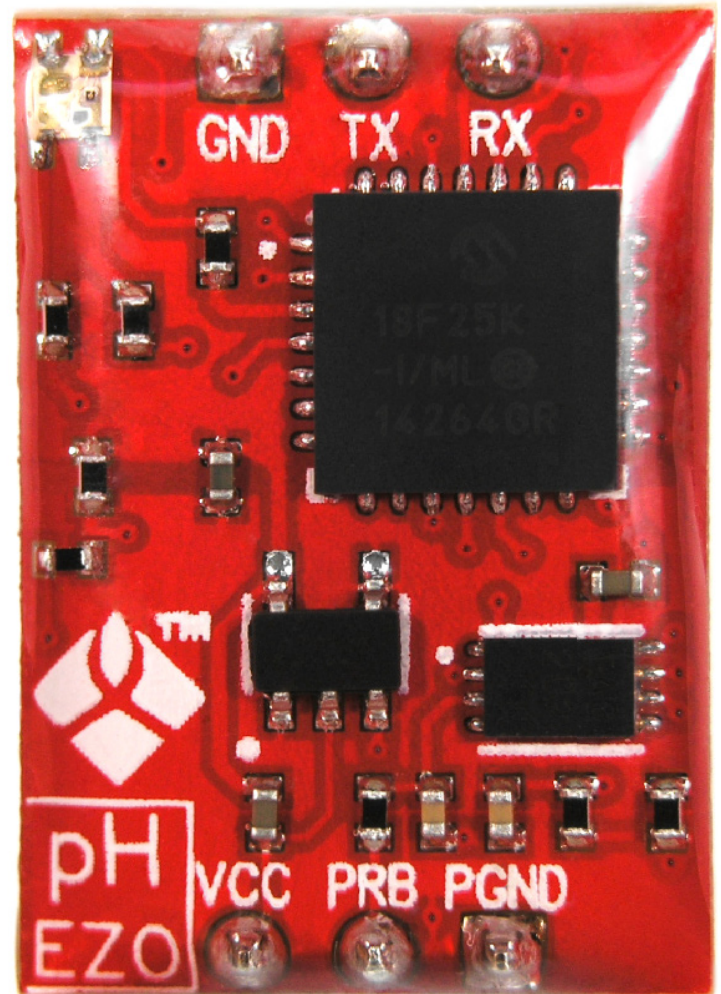
- Full range pH reading from .001 to 14.000
- Accurate pH readings down to the thousands place (**+/- 0.02**)
- Temperature dependent or temperature independent readings
- Flexible calibration protocol supports single point, 2 point, or 3 point calibration
- Calibration required only once per year with Atlas Scientific pH probe
- Single reading or continuous reading modes
- **Data format is ASCII**

Two data protocols

- UART asynchronous serial connectivity
- (RX/TX voltage swing 0-VCC)
- I²C (default I²C address 0x63)
- Compatible with any microprocessor that supports UART, or I²C protocol
- Operating voltage: 3.3V to 5V
- Works with any off-the-shelf pH probe

Sleep mode power consumption

- 0.995mA at 3.3V



Patent pending

Description

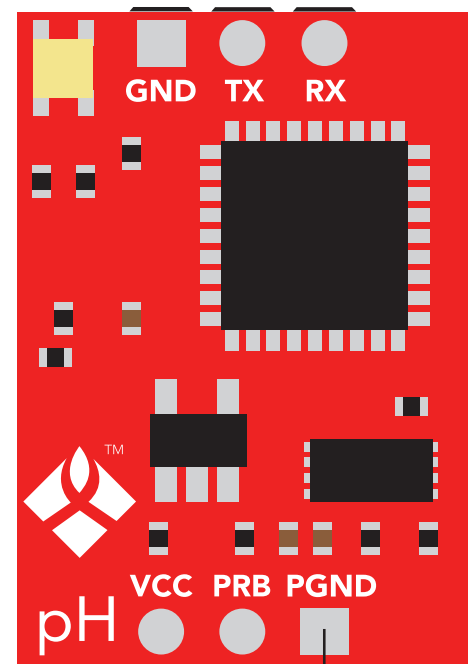
The Atlas ScientificTM EZOTM class embedded pH circuit, is our 6th generation embedded pH circuit. This EZO class pH circuit, offers the highest level of stability and accuracy. With proper configuration the EZO class pH circuit, can meet, or exceed the accuracy and precision found in most bench top laboratory grade pH meters. The pH-EZOTM pH circuit, can work with any off-the-shelf pH probe/sensor/electrode. **This device reads pH from a pH probe/sensor/electrode. This device does not include a pH probe/sensor/electrode.**



Pin Out

- GND** Return for the DC power supply
- Vcc** Operates on 3.3V – 5.5V
- TX / SDA** All EZOTM class circuits can operate in either UART mode, or I²C mode
- The default state is UART mode.
In UART mode, this pin acts as the transmit (TX) line. The default baud rate is 38400, 8 bits, no parity, no flow control, one stop bit. If standard RS232 voltage levels are desired, connect an RS232 converter such as a MAX232. If the device is in I²C mode, this pin acts as the Serial Data Line (SDA). The I²C protocol requires an external pull up resistor on the SDA line (resistor not included).
- RX / SCL** All EZOTM class circuits can operate in either UART mode, or I²C mode.
- The default state is UART mode.
In UART mode, this pin acts as the receive (RX) line. If the device is in I²C mode, this pin acts as the Serial Clock Line (SCL). The I²C protocol requires an external pull up resistor on the SCL line (resistor not included).
- PRB** This pin connects to the output lead of a pH probe/sensor/electrode
- PGND** This pin connects to the ground lead of a pH probe/sensor/electrode

This pin is not ground.
Do not tie this pin to system ground



Anexo N

Módulo de comunicación inalámbrica Xbee

1. XBee®/XBee-PRO® RF Modules

The XBee and XBee-PRO RF Modules were engineered to meet IEEE 802.15.4 standards and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between devices.

The modules operate within the ISM 2.4 GHz frequency band and are pin-for-pin compatible with each other.



Key Features

Long Range Data Integrity

XBee

- Indoor/Urban: up to 100' (30 m)
- Outdoor line-of-sight: up to 300' (90 m)
- Transmit Power: 1 mW (0 dBm)
- Receiver Sensitivity: -92 dBm

XBee-PRO

- Indoor/Urban: up to 300' (90 m), 200' (60 m) for International variant
- Outdoor line-of-sight: up to 1 mile (1600 m), 2500' (750 m) for International variant
- Transmit Power: 63mW (18dBm), 10mW (10dBm) for International variant
- Receiver Sensitivity: -100 dBm

RF Data Rate: 250,000 bps

Advanced Networking & Security

Retries and Acknowledgements
DSSS (Direct Sequence Spread Spectrum)
Each direct sequence channels has over 65,000 unique network addresses available
Source/Destination Addressing
Unicast & Broadcast Communications
Point-to-point, point-to-multipoint and peer-to-peer topologies supported

Low Power

XBee

- TX Peak Current: 45 mA (@3.3 V)
- RX Current: 50 mA (@3.3 V)
- Power-down Current: < 10 µA

XBee-PRO

- TX Peak Current: 250mA (150mA for international variant)
- TX Peak Current (RPSMA module only): 340mA (180mA for international variant)
- RX Current: 55 mA (@3.3 V)
- Power-down Current: < 10 µA

ADC and I/O line support

Analog-to-digital conversion, Digital I/O
I/O Line Passing

Easy-to-Use

No configuration necessary for out-of box RF communications
Free X-CTU Software (Testing and configuration software)
AT and API Command Modes for configuring module parameters
Extensive command set
Small form factor

Worldwide Acceptance

FCC Approval (USA) Refer to Appendix A [p64] for FCC Requirements. Systems that contain XBee®/XBee-PRO® RF Modules inherit Digi Certifications.

ISM (Industrial, Scientific & Medical) **2.4 GHz frequency band**

Manufactured under **ISO 9001:2000** registered standards

XBee®/XBee-PRO® RF Modules are optimized for use in the United States, Canada, Australia, Japan, and Europe. Contact Digi for complete list of government agency approvals.



Specifications

Table 1-01. Specifications of the XBee®/XBee-PRO® RF Modules

Specification	XBee	XBee-PRO
Performance		
Indoor/Urban Range	Up to 100 ft (30 m)	Up to 300 ft. (90 m), up to 200 ft (60 m) International variant
Outdoor RF line-of-sight Range	Up to 300 ft (90 m)	Up to 1 mile (1600 m), up to 2500 ft (750 m) international variant
Transmit Power Output (software selectable)	1mW (0 dBm)	63mW (18dBm)* 10mW (10 dBm) for International variant
RF Data Rate	250,000 bps	250,000 bps
Serial Interface Data Rate (software selectable)	1200 bps - 250 kbps (non-standard baud rates also supported)	1200 bps - 250 kbps (non-standard baud rates also supported)
Receiver Sensitivity	-92 dBm (1% packet error rate)	-100 dBm (1% packet error rate)
Power Requirements		
Supply Voltage	2.8 – 3.4 V	2.8 – 3.4 V
Transmit Current (typical)	45mA (@ 3.3 V)	250mA (@3.3 V) (150mA for international variant) RPSMA module only: 340mA (@3.3 V) (180mA for international variant)
Idle / Receive Current (typical)	50mA (@ 3.3 V)	55mA (@ 3.3 V)
Power-down Current	< 10 µA	< 10 µA
General		
Operating Frequency	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960" x 1.297" (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip or U.FL Connector, RPSMA Connector	Integrated Whip, Chip or U.FL Connector, RPSMA Connector
Networking & Security		
Supported Network Topologies	Point-to-point, Point-to-multipoint & Peer-to-peer	
Number of Channels (software selectable)	16 Direct Sequence Channels	12 Direct Sequence Channels
Addressing Options	PAN ID, Channel and Addresses	PAN ID, Channel and Addresses
Agency Approvals		
United States (FCC Part 15.247)	OUR-XBEE	OUR-XBEEPRO
Industry Canada (IC)	4214A XBEE	4214A XBEEPRO
Europe (CE)	ETSI	ETSI (Max. 10 dBm transmit power output)*
Japan	R201WW07215214	R201WW08215111 (Max. 10 dBm transmit power output)*
Australia	C-Tick	C-Tick

* See Appendix A for region-specific certification requirements.

Antenna Options: The ranges specified are typical when using the integrated Whip (1.5 dBi) and Dipole (2.1 dBi) antennas. The Chip antenna option provides advantages in its form factor; however, it typically yields shorter range than the Whip and Dipole antenna options when transmitting outdoors. For more information, refer to the "XBee Antennas" Knowledgebase Article located on Digi's Support Web site

Mechanical Drawings

Figure 1-01. Mechanical drawings of the XBee®/XBee-PRO® RF Modules (antenna options not shown)

Anexo Ñ

Motor de hélices

LEOPARD Brushless Motor for LBP56110

Thank you for purchasing one of Leopard Hobby's products. Leopard motors represent the latest brushless motor technology – all the latest innovations combined with the best materials available on the market. This motor is perfectly matched with Leopards range of ESCs. Please kindly visit our website www.leopardhobby.com for more details about Leopard Hobby's range of products. All motors have six weeks limited warranty from the date of purchase that does not cover crash damage, improper use or overheating.

◆ Features:

- ◆ High efficiency, high power, high torque
- ◆ Sintered neodymium rotor magnets
- ◆ 12 slot stator
- ◆ Powerful four pole design
- ◆ Minimal motor maintenance is required
- ◆ Excellent heat dissipation, a high overload ability
- ◆ Handwound
- ◆ long service life
- ◆ Low current, low temperature
- ◆ Low electromagnetic interference
- ◆ Precision ball bearings
- ◆ Capable of high overloads for short periods

◆ Specifications:

LBP56110 Series											
Model	Max Amps	Max voltage	Max Power	KV (RPM /Volt)	Resis-tance	No-load Current (15V)	Diameter × Length (mm)	Mounting hole depth	Length of extend Shaft	Shaft Diameter (mm)	Weight
LBP56110 /2Y	170A	44V	7600W	780	0.0072	3.7A	Φ55.8×110	8mm	30mm	Φ8.0	1160g
LBP56110 /3Y	113A	67V	7600W	520	0.0152	2.1A	Φ55.8×110	8mm	30mm	Φ8.0	1160g
LBP56110 /4Y	82A	92V	7600W	380	0.0265	1.8A	Φ55.8×110	8mm	30mm	Φ8.0	1160g
LBP56110 /2D	300A	24V	7600W	1430	0.0027	7.4A	Φ55.8×110	8mm	30mm	Φ8.0	1160g
LBP56110 /3D	210A	36V	7600W	950	0.0055	4.3A	Φ55.8×110	8mm	30mm	Φ8.0	1160g
LBP56110 /4D	158A	48V	7600W	720	0.0092	3.1A	Φ55.8×110	8mm	30mm	Φ8.0	1160g
LBP56110 /4.5D	140A	54V	7600W	640	0.0149	3.0A	Φ55.8×110	8mm	30mm	Φ8.0	1160g

◆ Caution:

- ◆ Do not immerse the motor in Acid/Alkaline solvent.
- ◆ Do not bend wires excessively. Secure wires to prevent breakage due to vibration.
- ◆ Avoid over gearing by monitoring the temperature.
- ◆ Be sure to use the proper size mounting screws to avoid a short circuit or damage to other components.
- ◆ Do not apply full throttle if motor is not installed, this can cause severe motor damage (high RPMs without a load).
- ◆ Do not cut motor wires.