

UNIVERSIDAD COMPLUTENSE DE MADRID

DEPARTAMENTO DE ARQUITECTURA DE COMPUTADORES Y
AUTOMÁTICA



Modelado y Simulación de Sistemas Dinámicos – Barco entre las olas

TRABAJO FIN DE GRADO DEL GRADO EN CIENCIAS FÍSICAS

Autor: Juan Serrano Sanz

Tutores: Carlos García Sanchez, Jose Manuel Velasco Cabo

Fecha: 3 de Julio de 2019

Agradecimientos

Quisiera agradecer en primer lugar a mis tutores por la ayuda prestada para realizar este Trabajo Fin de Grado, resolviendo y asesorandome en cualquier duda que me pudiese surgir a lo largo del desarrollo de este TFG. A mi familia, por ser un constante apoyo durante todos estos años, a la Universidad Complutense de Madrid por ofrecerme la posibilidad de estudiar el Grado que quería, a mis amigos por ayudarme a desconectar y descansar la cabeza cuando más lo he necesitado, a todos los profesores y compañeros que han hecho que este proceso de aprendizaje no solo sea un crecimiento académico sino también humano y por último a mi pareja por estar en los buenos momentos, pero sobretodo por ayudarme en los no tan buenos. Muchas gracias a todos.

Resumen

El presente Trabajo de Fin de Grado se divide en dos partes, la primera consiste en adquirir los conceptos básicos de la modelización y simulación de sistemas dinámicos en MATLAB, para ello usaremos como guía el capítulo 7 de [1].

La segunda parte consiste en una vez ya familiarizados con el entorno y partiendo de lo aprendido en la primera parte, desarrollar una simulación de un barco que será controlado mediante un mando por el usuario y estará sometido a las corrientes del océano.

Abstract

This Degree Project is divided in two phases, first one is acquiring basic knowledge of modeling and simulation of dynamic in MATLAB, we will use chapter 7 from [1] as guide.

Second's phase purpose is using our recently acquired knowledge to develop a boat simulation where the user will be able to control the boat with a controller while oceanic currents will affect the ship.

Índice general

Índice de Figuras	VIII
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	1
2. Introducción al Modelado y Simulación de SSDD	2
2.1. Introducción	2
2.2. Creación de la Escena Virtual para el barco y el mar	2
2.3. Ecuaciones de Movimiento del Barco y las Olas de Mar	3
2.4. Script de MATLAB® para Animar la Escena Virtual	3
2.5. Aplicación: Controlador de orientación del barco	4
2.6. Resultados	6
3. Simulación de un Barco entre las Olas Utilizando un Joystick	7
3.1. Introducción	7
3.2. Modificación del entorno creado en 2.2	7
3.3. Script de MATLAB® para Animar la Escena Virtual	8
3.4. Resultados	10
3.5. Posibles mejoras	12

3.6. Conclusiones	12
A. Código utilizado	13
Bibliografía	14

Índice de figuras

2.1. Referencia de los ángulos del barco	4
2.2. Esquema del controlador principal del Fuzzy Logic Controller	5
2.3. Reglas del Fuzzy Logic Controller	5
2.4. Funciones de pertenencia de e	5
2.5. Ángulos ψ , ψ_{target} y α frente al tiempo	6
3.1. Partes de un barco	9
3.2. Velocidad del barco durante la simulación ejemplo	10
3.3. Trayectoria del barco	11

Capítulo 1

Introducción

1.1. Motivación

Desde el primer curso del Grado en Ciencias Físicas, que fue mi primera toma de contacto con el mundo de la programación con la asignatura Laboratorio de Computación Científica, me llamó la atención aprender a programar, es por ello que en tercero decidí cursar la asignatura de Estadística y Análisis de Datos y finalmente realizar un Trabajo de Fin de Grado que perteneciese al departamento de Arquitectura de Computadores y Automática.

Además creo que en mi futuro laboral programar va a ser muy importante, por lo que todo lo que pueda aprender mientras esté en la Universidad será un conocimiento muy útil y que me abrirá muchas puertas.

1.2. Objetivos

El objetivo de este TFG es familiarizar al alumno con los conceptos básicos de la modelización y simulación de sistemas dinámicos para luego realizar una simulación de un barco navegando en un entorno creado con la herramienta V-Realm Builder. El movimiento del barco estará determinado en todo momento por las instrucciones dadas por el usuario mediante un mando de consola que permitirá acelerar, frenar y controlar el timón del barco como se desee. Además se ha introducido una corriente que desplazará al barco y le afectará en mayor o menor medida dependiendo del ángulo que forme el barco con la corriente.

Capítulo 2

Introducción al Modelado y Simulación de SSDD

2.1. Introducción

En esta primera parte vamos a aprender los conceptos básicos de la modelización y simulación de sistemas dinámicos, para ello empezaremos creando el entorno en el que realizaremos nuestra simulación, posteriormente introduciremos las ecuaciones que darán vida a nuestro barco y a las olas sobre las que navega.

2.2. Creación de la Escena Virtual para el barco y el mar

En este capítulo se explica el proceso para crear el Entorno Virtual utilizando la herramienta V-Realm Builder.

- 1- Abrimos V-Realm Builder e iniciamos un nuevo proyecto.
- 2- Añadimos un fondo y un grid que actuará como superficie del mar, además le daremos textura con una imagen previamente descargada. Cambiaremos las propiedades por defecto del mar para colocarlo de la forma deseada.
- 3- Creamos un punto desde el cual veremos la simulación, este Viewpoint puede ser estático o móvil como veremos más adelante.

- 4- Con el fin de crear un paisaje introduciremos una caja en nuestro entorno y cambiaremos sus dimensiones para que ocupe todo el fondo de la escena y le daremos la textura de montañas de forma similar a lo que hicimos en el paso 2.
- 5- Ahora introduciremos el barco utilizando la librería de objetos que nos ofrece VRML.
- 6- Por último iluminamos la escena para que todo se vea con claridad y guardamos nuestro proyecto.

2.3. Ecuaciones de Movimiento del Barco y las Olas de Mar

Las ecuaciones de esta primera parte serán muy sencillas:
La altura de las olas irá cambiando sinusoidalmente obedeciendo la fórmula (2.1)

$$h(t, z) = 2\sin(\omega t + z) \quad (2.1)$$

El barco solo se moverá en la dirección x, empezando en la posición $x(0) = -100$

$$x(t) = -100 + 10t \quad (2.2)$$

2.4. Script de MATLAB® para Animar la Escena Virtual

El código utilizado para realizar esta primera simulación se puede encontrar completo en el Apéndice A, sin embargo en esta sección comentaremos las partes más importantes de dicho código, ya que nos ayudarán a un mayor entendimiento del proceso que ocurre al ejecutar el script.

- 1- Primero hay que cargar el entorno que hemos construido en la sección 2.2 e introduciremos las variables que irán cambiando a lo largo de la simulación.
- 2- Creamos un bucle que vaya modificando los parámetros de la escena virtual en cada iteración, dentro de este bucle una vez cambiados los parámetros los actualizaremos en la escena para poder ver la forma en la que va evolucionando nuestra simulación. También forzamos una pequeña pausa entre cada iteración para aproximar así la animación a un movimiento más real.
- 3- Por último se puede recoger los valores que han ido tomando las variables para elaborar unas gráficas y obtener un mayor entendimiento del proceso que ha sido realizado. En este apartado ignoraremos este paso pero lo usaremos más adelante.

2.5. Aplicación: Controlador de orientación del barco

El principal objetivo de esta sección es diseñar un controlador para que el barco se gire hacia un determinado ángulo ψ , será exclusivamente un movimiento de rotación. Utilizaremos el Fuzzy Logic Controller, que es una herramienta proporcionada por MATLAB®, con dos inputs y un output, los inputs son el error normalizado y su derivada, mientras que el output es la acción que provoca que el timón se gire un ángulo α para acercarse al ángulo ψ . Para mejor comprensión de qué representa cada ángulo ver Figura: 2.1. El código utilizado en esta sección está en Apéndice A.

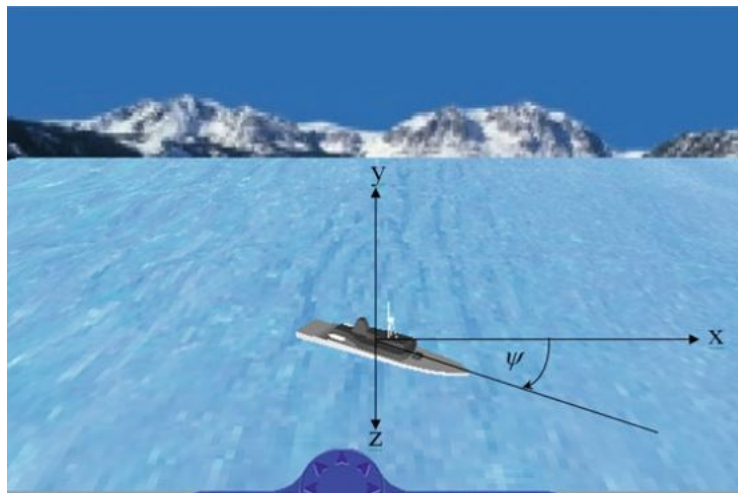


Figura 2.1: Referencia de los ángulos del barco

Para que el Fuzzy Logic Controller funcione adecuadamente necesita que introduzcamos unos valores por defecto o "gains" que regirán el movimiento que realice el barco. Estos gains serán el ángulo final y la diferencia entre el ángulo actual ψ y ψ_{target} , que se irá actualizando a medida que se realiza la simulación. También será necesario introducir K_e , $K_{\dot{e}}$ y K_{α} que son factores que acercan la simulación a la realidad.

Para obtener el ángulo α tras todo el proceso es necesario además multiplicarlo por K_{α} . El esquema del Fuzzy Logic Controller se puede ver en la Figura 2.2 y su lógica está representada en las Figuras 2.3 y 2.4

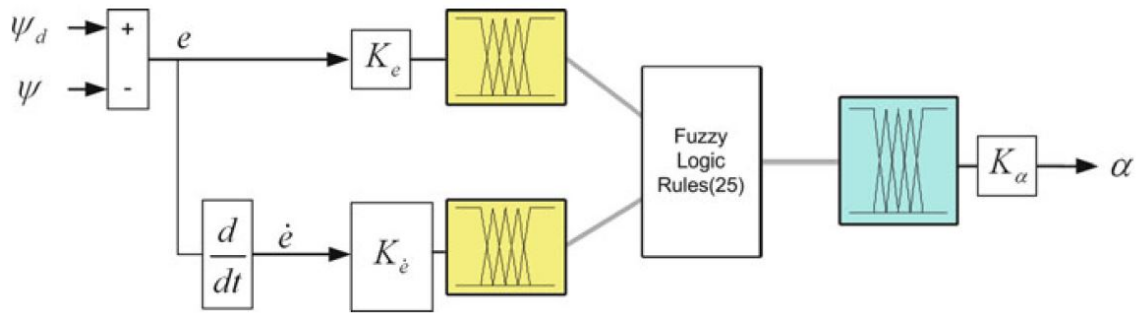
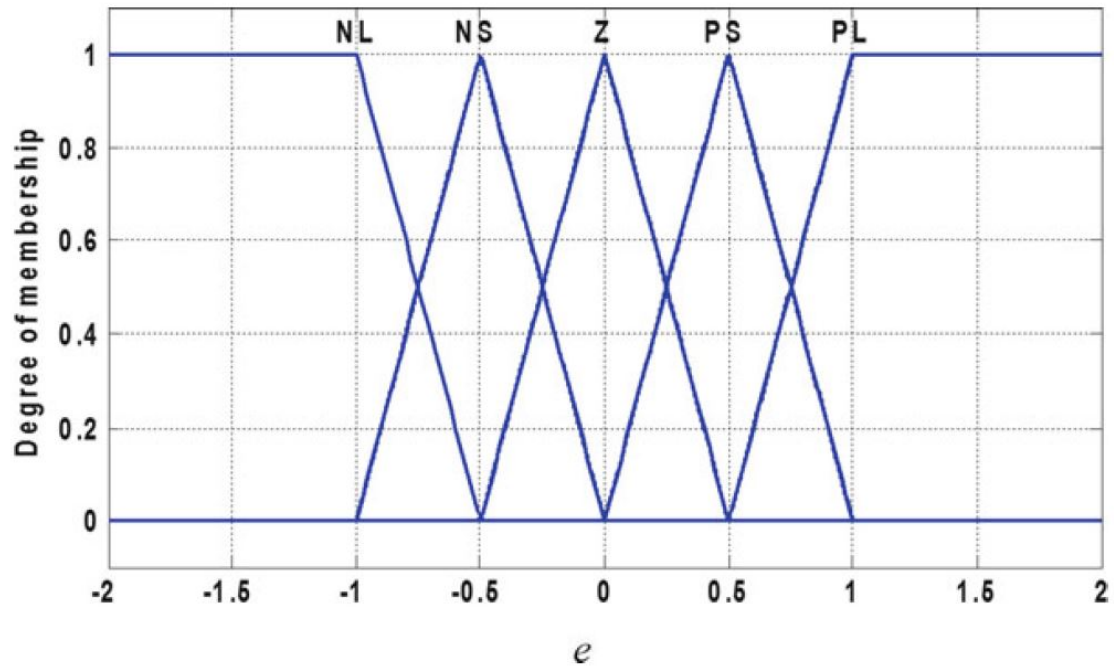


Figura 2.2: Esquema del controlador principal del Fuzzy Logic Controller

$\begin{matrix} e \\ \dot{e} \end{matrix}$	NL	NS	Z	PS	PL
NL	NL	NL	NL	NL	PL
NS	NL	NS	NS	Z	PL
Z	NL	NS	Z	PS	PL
PS	NL	Z	PS	PS	PL
PL	NL	PL	PL	PL	PL

Figura 2.3: Reglas del Fuzzy Logic Controller


Figura 2.4: Funciones de pertenencia de e

2.6. Resultados

Como el objetivo de la simulación realizada en 2.4 es simplemente ver que somos capaces de crear el entorno y hacer que el barco se mueva, no nos detendremos en ello. Sin embargo si almacenamos los datos de los ángulos de la simulación de 2.5 y los representamos gráficamente podemos ver cómo el barco tiende al ángulo objetivo ψ_{target} , una vez que se acerca y debido a la inercia del movimiento empieza a oscilar en torno a él como podemos observar en la Figura: 2.5. También podemos ver cómo el timón cambia continuamente de posición para intentar corregir su movimiento anterior para volver a colocar el barco con el ángulo ψ_{target}

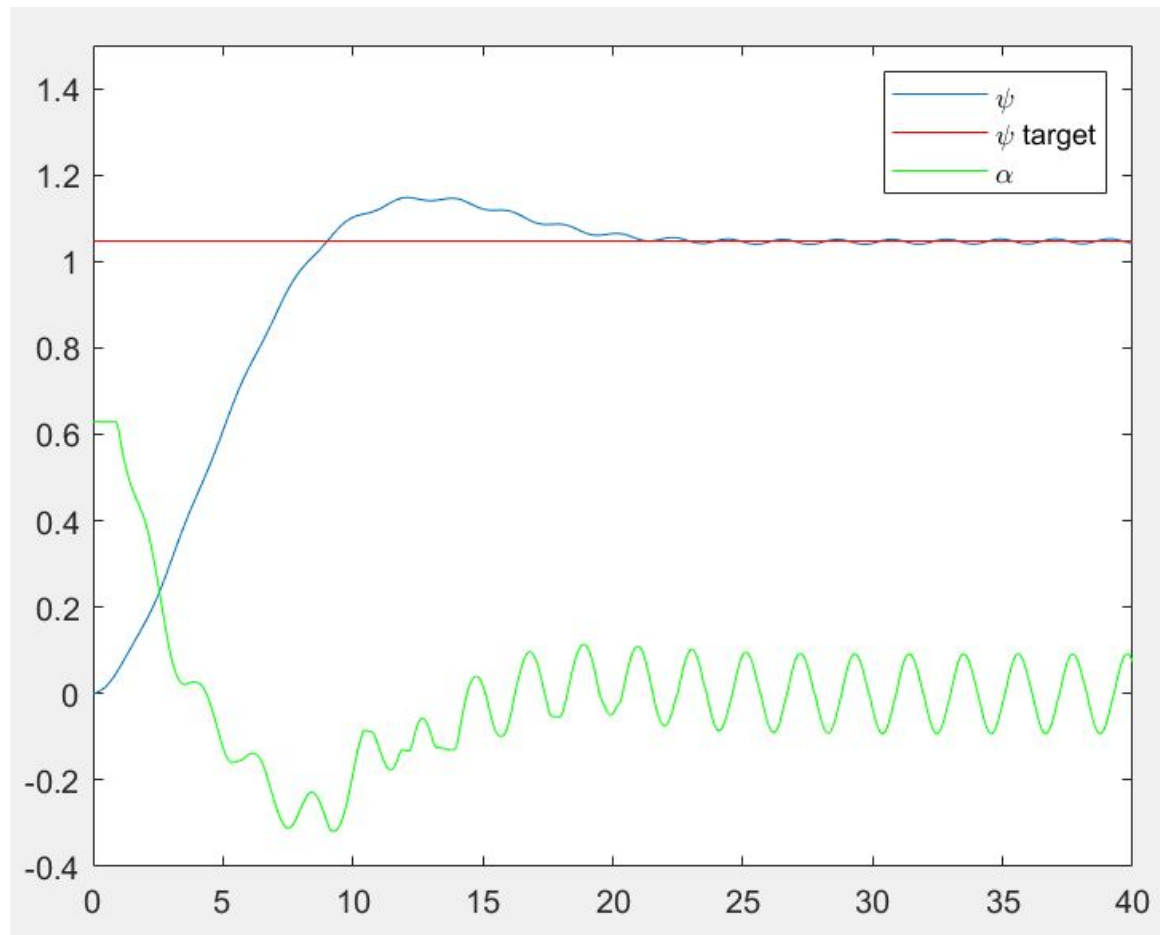


Figura 2.5: Ángulos ψ , ψ_{target} y α frente al tiempo

Capítulo 3

Simulación de un Barco entre las Olas Utilizando un Joystick

3.1. Introducción

En esta segunda etapa del TFG vamos a poner en práctica todo el conocimiento adquirido en el Capítulo 2, para ello tendremos que modificar el entorno creado en 2.2. Usando como ayuda el Capítulo 6 de [1] implementaremos un mando de consola como input de nuestra simulación.

3.2. Modificación del entorno creado en 2.2

La simulación realizada en 2 muestra un barco que se desplaza en el eje x , nosotros queremos que nuestro barco se pueda desplazar ahora en 2D y para ello tenemos que actualizar el grid del océano y dotarle de un mayor tamaño en el eje y , antes ignorado. Debido a las limitaciones de MATLAB® solo podemos crear un grid con 10001 polígonos como máximo.

Modificamos el Viewpoint y lo convertimos en una cámara con "vista de águila" para poder ver el desplazamiento del barco en el plano x, y y añadimos otro viewpoint que se sitúe encima del barco y se desplace con él. Por último repetiremos el paso 4 de 2.2 y añadimos los otros 3 lados del océano que nos faltan para crear así un entorno cerrado.

3.3. Script de MATLAB® para Animar la Escena Virtual

- 1- Lo primero que hay que hacer para realizar la simulación es cargar el entorno que hemos creado en 3.2 e introducir los valores iniciales de posición, tiempo etc, los valores fijos como las dimensiones del barco o el tamaño del mar.
- 2- Creamos también la variable del joystick que vamos a usar para controlar el barco. Para asegurarnos de que el programa está leyendo correctamente lo que indicamos con el joystick se ha añadido un gráfico que irá indicando en todo momento la posición del mismo.
- 3- Una vez hechos los pasos iniciales procedemos a crear el bucle que iteración tras iteración irá actualizando la posición del barco y su rotación. Se ha elegido hacer un bucle infinito utilizando el comando *while* 1 para no tener así ninguna limitación con el tiempo que podemos estar realizando la simulación, para salir de este bucle infinito se puede programar que al pulsar un botón del mando se active el comando *break* y así poner fin al bucle.
- 4- El barco se puede mover marcha adelante y marcha atrás dependiendo de si apretamos \bigcirc o \times , la fórmula utilizada será:

$$ac = delta.t * (\bigcirc - \frac{3}{4} * \times) \quad (3.1)$$

en donde \bigcirc es el botón de acelerar en nuestro mando y \times el de frenar y solo pueden valer 1 o 0, *delta.t* es el incremento temporal entre iteraciones. La velocidad del barco será la velocidad previa mas la aceleracion multiplicada por el diferencial temporal.

- 5- Acotamos la velocidad del barco para que no supere un valor máximo y la multiplicamos por una matriz de rotación, ya que esta simulacion no es como la de 2, aquí trabajamos en el plano x,z . La matriz de rotación dependerá del ángulo que tenga el barco respecto al eje x . Cuando decimos el ángulo del barco lo que estamos haciendo es convertir el barco a un vector que va desde su proa a popa.
- 6- Como el mar no está en calma tendremos que tener en cuenta este movimiento a la hora de calcular la velocidad del barco, también será importante la orientación del barco respecto a la corriente, ya que no es lo mismo que el barco se desplace paralelamente a la corriente que transversalmente, pues la corriente le afectará mas si incide por babor o estribor que por proa o popa.
- 7- La nueva posición del barco vendrá determinada por:

$$Position_i = Position_{i-1} + V_{Ship} \quad (3.2)$$

Y la actualizaremos en cada iteración en nuestro entorno virtual.



Figura 3.1: Partes de un barco

- 8- La orientación del barco es importante ya que el ángulo de la velocidad depende de ella. Lo primero que hacemos es introducir la variable que irá moviendo el timón con el joystick usando la fórmula:

$$\theta_i = \theta_{i-1} + 0.2 * a(1) \quad (3.3)$$

Donde $a(1)$ corresponde al eje x del joystick. Después limitaremos el ángulo de giro del timón entre $-\frac{\pi}{4}$ y $\frac{\pi}{4}$ y modificaremos el ángulo del barco:

$$\beta_i = \beta_{i-1} + 0.05 * \theta_i \quad (3.4)$$

Y por último actualizamos el entorno virtual con el nuevo ángulo β_i .

- 9- El final del bucle será actualizar nuestras variables iniciales con las variables que han sido calculadas desde que hemos comenzado el *while*. Los datos calculados los vamos almacenando en vectores para representarlos una vez terminada la simulación.

3.4. Resultados

En nuestra simulación vamos a ver que el barco parte con una velocidad inicial distinta de 0, esto es porque estando en reposo el barco tiene la velocidad de la corriente. El mar en el que se hace la simulación es de tamaño limitado como ya se explicó en 3.2 aunque solo sea una limitación visual, ya que el barco se comportará de igual manera independientemente de si está dentro o fuera del recinto dibujado con V-Realm Builder.

Cuando terminemos de mover nuestro barco por el entorno nos saldrán dos gráficas: una de ellas representando el camino recorrido por el barco y la otra indicando la velocidad en valor absoluto que tenía el barco en cada momento, por lo que nuestra gráfica de velocidad no diferenciará si el barco va hacia delante o marcha atrás.

Realizamos una simulación ejemplo y obtenemos las gráficas 3.2 y 3.3, en donde observamos que las zonas con mayor densidad de puntos coinciden con las zonas de menor velocidad y así podríamos extrapolar en dónde ha estado el barco en cada momento, pero como el objetivo principal es estudiar la trayectoria se ha decidido dejar el gráfico en 2D y no introducir el tiempo como otra variable más.

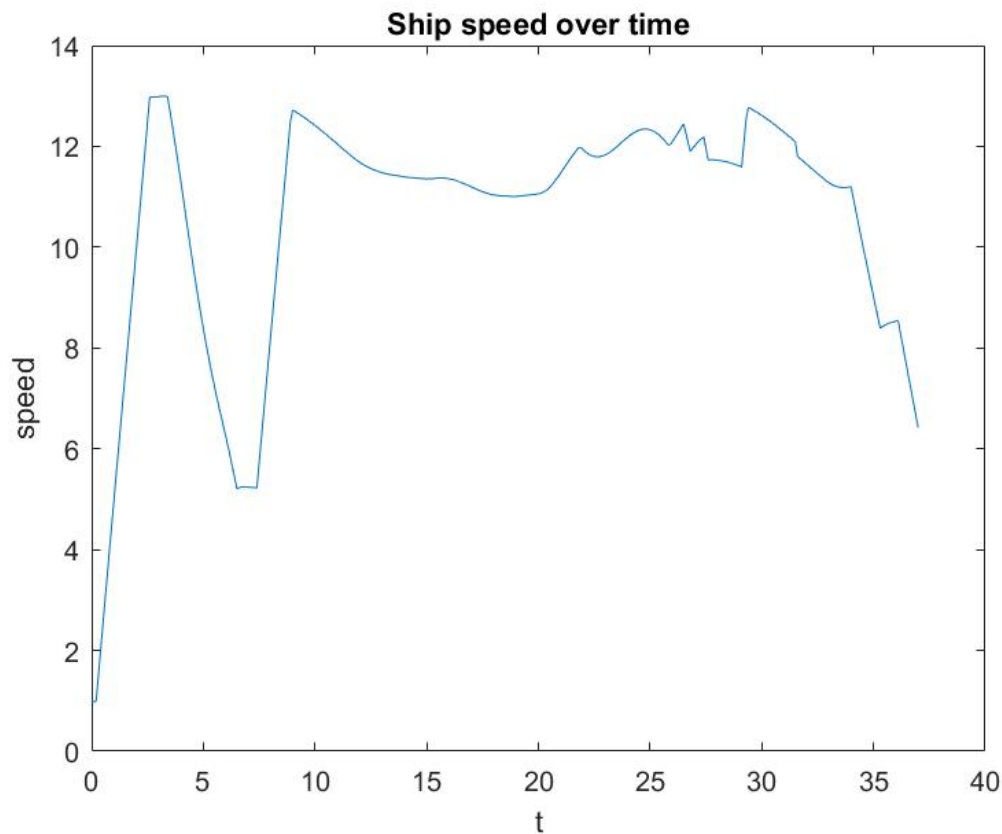


Figura 3.2: Velocidad del barco durante la simulación ejemplo

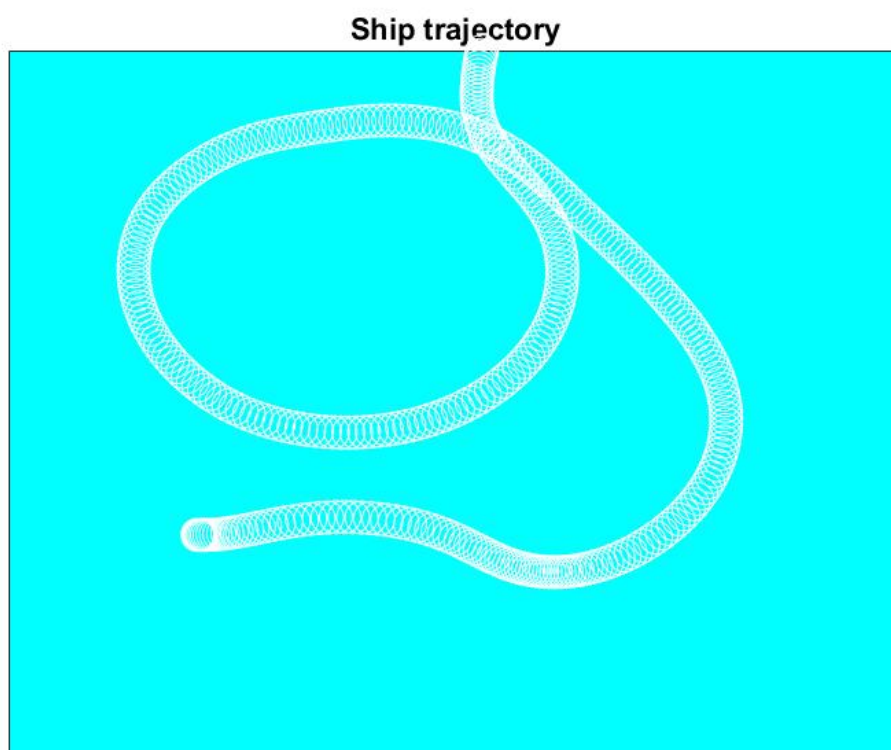


Figura 3.3: Trayectoria del barco

3.5. Posibles mejoras

En este proyecto hemos intentados simular un barco, pero debido a la escasez de práctica, conocimientos y tiempo entre otras cosas la simulación no es todo lo real que podría ser. El barco se desplaza sobre las olas sin que estas afecten al movimiento, algo que una buena simulación sí debería tener en cuenta. El entorno virtual no es todo lo gráficamente agradable que nos gustaría, debido en parte a la antigüedad del programa utilizado y a que está mas centrado en ofrecer una buena funcionalidad que en la estética. Otra posible mejora sería hacer tender el timón a alinearse con el movimiento del barco, de forma que estuviese teniendo en cuenta el empuje que realizaría el timón sobre el mar, esto se podría hacer utilizando el Fuzzy Logic Controller forma análoga a lo hecho en 2.5 pero cambiando el ψ_{target} por el ángulo del barco respecto al eje x en cada instante (β_i)

3.6. Conclusiones

A pesar de todas las posibles mejoras que se pueden añadir o modificar a este sistema dinámico, hemos sido capaces de desarrollar una simulación adecuadamente en la que somos perfectamente capaces de navegar el barco utilizando un mando de consola, llevar un control de que todo está funcionando correctamente, sacar datos de la simulación para poder estudiarlos y comprobar que cada simulación dependerá de cómo decidamos manejar el barco.

En el Capítulo 1 se explicaba algunos de los motivos que me habían llevado a escoger el tema de este TFG, y una vez terminado puedo concluir que el principal objetivo, el de aprender, ha sido cumplido satisfactoriamente. A programar se aprende programando, pero no solo he aprendido más sobre programación si no que he seguido mejorando mi habilidad para buscar posibles soluciones a errores que iban saliendo a medida que desarrollaba el script, para ello la página [2] y la biblioteca de MATLAB® [3] han sido fundamentales al igual que la ayuda de mis dos tutores.

Apéndice A

Código utilizado

<https://github.com/JSerranoS/MySSD-Barco-entre-las-olas>

Bibliografía

- [1] Nassim Khaled, et al.: 'Virtual Reality and Animation for MATLAB® and Simulink® Users'
- [2] <https://www.mathworks.com/>
- [3] <https://stackoverflow.com/>