

Sistema Inteligente de apoyo a maniobras de grandes buques en puertos

G. N. Marichal ^{*}, A. Hernández, J.A. Rojas, E. Melón, J.A. Rodríguez, I. Padrón

*Escuela Politécnica Superior de Ingeniería. Sección de Náutica, Máquinas y Radioelectrónica Naval, Universidad de La Laguna.
C/ Avda. Francisco Larroche s/n. 38071.Santa Cruz de Tenerife, España.*

Resumen

En este trabajo se propone un sistema de abordaje de apoyo al capitán o al práctico que se encuentre realizando una maniobra dentro de puerto. Concretamente, el elemento clave del sistema es la presencia de un sistema Neuro-Fuzzy genético que una vez entrenado permitirá su utilización como una herramienta de ayuda, sugiriendo cómo se debe maniobrar según las circunstancias. Concretamente se centra en la maniobra de desatraque, indicando cuál es la maniobra más apropiada a realizar. La indicación de la manera de maniobrar se proporciona a través de tres parámetros de control presentes en los grandes buques: el ángulo de giro del timón, el nivel de potencia del motor principal y el de hélices de maniobra. Se han realizado varios ensayos en simulación, a partir de los datos adquiridos en donde se han mostrado una serie de resultados satisfactorios.

Palabras Clave: Algoritmos genéticos, Neuro-Fuzzy, buque, maniobra

1. Introducción

El atraque y desatraque son las maniobras que más riesgo suponen en el manejo de una embarcación, de hecho la mayoría de los abordajes de buques se producen en el puerto (Pérez, 2012). Existen muchos factores que se deben tener en cuenta durante el transcurso de esta operación, como las dimensiones del buque y su comportamiento dinámico (Santos, 2011) (Herrera, 2012) localización de las embarcaciones cercanas, condiciones meteorológicas, como los vientos predominantes, las corrientes, altura de la marea, las estructuras del propio muelle o sus luces y marcas (Marrero *et al.*, 93). Esta diversidad de factores influyentes hace que sea una maniobra compleja que muchas veces requiere de ayuda externa. Los apoyos mayoritarios que se prestan dentro del recinto portuario están formados por prácticos y remolcadores, lo que supone un gasto, tanto económico como temporal, para la empresa propietaria de la embarcación. Por otro lado el capitán cuenta con sistemas a bordo que le proporcionan información, como el ARPA, el AIS (Colin *et al.*, 2004), anemómetro o el ECDIS (Zhang *et al.*, 2007, Pillich *et al.*, 2001), pero lo que finalmente lleva al éxito en la maniobra es la pericia y experiencia del capitán.

La presencia de la automatización en los buques es cada vez más evidente. Actualmente se están desarrollando herramientas más novedosas que proporcionan a los oficiales información para realizar las maniobras dentro del muelle (Espósito 2014, Gucma, *et al.*, 2008). Por un lado están los dispositivos PPU (Portable

Pilot Unit). Se trata de un par de antenas GPS, separadas por una barra de longitud fija. Los receptores GPS calculan la posición, la dirección del vector que los une y la tasa de giro, obtenida por la ayuda de un pequeño sensor inercial. Dichos datos son transmitidos mediante señal de radio (Wi-Fi), lo que permite en su propio equipo informático y con el software apropiado, ver en tiempo real, en su propia cartografía, la ubicación del barco dentro del entorno portuario, su velocidad, dirección y tasa de giro.

Por otro lado se están comercializando sistemas de asistencia al atraque consistente en un conjunto de láser (Gucma, *et al.*, 2010) dispuestos en la estructura vertical del muelle, dirigidos hacia el barco. Cada conjunto de láser envía la información a través de un ordenador situado en tierra, que muestra a través de un gran display la distancia a la que se encuentra el buque, su inclinación y su velocidad. La distancia es la medición más importante puesto que en la mayoría de las embarcaciones se obtiene de manera inexacta dado que se proporciona directamente por el personal, lo que está sujeto a fallos de factor humano.

Esta nueva tecnología proporciona información relativa al buque, pero finalmente el capitán es quien debe decidir la maniobra a realizar. En este trabajo se propone un sistema de abordaje de asistencia al capitán, sugiriéndole cómo debe maniobrar según las circunstancias. Se trata de un algoritmo que en función de las distancias a los distintos obstáculos que tiene alrededor, como las estructuras verticales (paredes) del muelle o las embarcaciones atracadas en su entorno, sugiere cuál es la maniobra apropiada a realizar. Concretamente se proporciona el ángulo de giro del timón y el nivel de potencia del motor principal y el de hélices laterales.

^{*} Autor en correspondencia.

Correos electrónicos: ni.comar@ul1.edu.es

La estructura del trabajo incluye en la sección 2 una exposición del simulador empleado para la realización de las maniobras, además de la explicación del proceso seguido con los datos recogidos. Por su parte, en la sección 3 se presenta una descripción de la técnica basada en una aproximación híbrida de un algoritmo genético y un sistema neuro-fuzzy, empleada para el entrenamiento de los datos. Para finalmente presentar los resultados y conclusiones en la sección 4 y 5, respectivamente.

2. Toma de datos

La idea de este trabajo es estudiar las decisiones que toma el capitán a bordo del buque a la hora de realizar una maniobra en función de los obstáculos que hay a su alrededor. Este estudio no es posible hacerlo mientras se ejecuta la maniobra a bordo de un barco real, ya que no está permitido permanecer en el puente de mando, salvo al personal encargado de la maniobra, como el capitán, el práctico y el timonel. Por esta razón se ha decidido utilizar entornos virtuales. Se ha empleado el programa Shpissimulator, que es un programa de simulación, en el que se puede realizar una multitud de tareas dentro de un entorno 3D, tomando el control de una amplia gama de tipos de buques, desde grandes buques de carga hasta lanchas o yates. Tanto la calidad gráfica de los buques y su dinámica responden al comportamiento esperado de cada acción, haciendo que el programa reproduzca con cierto nivel de detalle el comportamiento que tendría el barco. Dentro de la gran gama de posibilidades que ofrece Shpissimulator, se ha decidido fijar para el desarrollo de este trabajo el mismo tipo de buque y además realizar la maniobra en el mismo entorno portuario. Particularmente se ha elegido el buque portacontenedores, puesto que es uno de los barcos más grandes y por lo tanto, con un mayor nivel de dificultad a la hora de maniobrar dentro de un puerto. Por otro lado se ha elegido que se estudiará la maniobra de desatraque, a realizar en el puerto de Rotterdam. Otra variable modificable en el entorno de simulación es el viento. Sin embargo a efectos de simplificación en el estudio, las maniobras se han realizado tomando en consideración unas condiciones de viento constante. Es importante señalar que los métodos propuestos en este artículo son directamente extrapolables a otras situaciones en donde las condiciones de viento sean diferentes. De igual forma a efecto de las simulaciones se ha considerado la situación de corrientes constantes ó de poca magnitud, suposición mantenida en este caso al realizar las operaciones dentro de un entorno portuario en presencia de condiciones meteorológicas no adversas. La figura 1 muestra una captura de un momento de una de las maniobras de desatraque realizadas en el simulador. En este estudio, se ha considerado un buque cuyo manejo depende de tres variables de control. Concretamente, se ejecutan a través de la rueda de timón mostrada en el lado derecho de la imagen, la palanca de potencia del motor principal, a la izquierda de la imagen, y la palanca de acción del motor de la hélice de proa, situado a la derecha del anterior.

Es importante señalar respecto a la hélice de proa que ésta por lo general se coloca dentro de un túnel que atraviesa la proa de babor a estribor estando la hélice al centro de dicho túnel. al accionarla en un sentido u otro moverá la proa a babor o a estribor. Sin embargo, en este artículo al objeto de simplificar la exposición nos referiremos a hélice lateral izquierda y hélice lateral derecha, indicando que se realiza un desplazamiento a babor o estribor.



Figura 1: Captura de una maniobra en Shpissimulator.

El primer paso por tanto, es ejecutar la maniobra de desatraque del portacontenedores desde un amarre del puerto de Rotterdam modificando las posiciones de los buques que se encuentran a su alrededor, unos atracados y otros no, tal y como muestra la figura 1. De esa forma podremos obtener un conjunto de datos necesario para el ajuste del sistema propuesto en este artículo y la comprobación de su funcionamiento. Para que la ejecución de la maniobra sea realizada de manera óptima, se ha pedido la colaboración de un capitán con más de cinco años de experiencia para que maniobre la embarcación. Los controles que se pueden manipular en el programa son los que tendría en una situación real: rueda de timón (u otro mando de control del timón), mando de gestión del motor principal y de la hélice de proa. Entonces, se le proporcionan al capitán diferentes situaciones, en lo que a obstáculos se refiere, y se graban las maniobras realizadas, así como los cambios en los controles que se han ejecutado. La figura 2 muestra una fotografía del capitán realizando una maniobra a través del simulador. Es importante remarcar que un capitán experto sería necesario en la fase de aprendizaje del sistema, tal como indica la figura 2.



Figura 2: Fotografía del capitán realizando maniobras en el simulador.

De hecho, esta aportación por parte de un capitán experto en el pilotaje del buque objeto de estudio es vital para que el sistema sea capaz de incorporar su dinámica particular al sistema propuesto y de esa forma poder extender los métodos expuestos a otros buques.

Una vez se disponen de estos datos se debe construir una tabla que relacione la forma de realizar la maniobra, a través de los controles, con las distancias a los obstáculos que hay alrededor. En el caso presentado en este trabajo se ha simulado que el buque tiene un grupo de 10 sensores distribuidos uniformemente a lo largo de toda su cubierta, de manera que cada uno de ellos mide la distancia al obstáculo más cercano. Entonces, puesto que cada maniobra dura algo más de 4 minutos, se recopila la información cada 10 segundos, es decir, el ángulo de giro del timón, la potencia del motor principal, la potencia del motor de hélice de proa, y las distancias que detectan los 10 sensores en ese instante. A la hora de recoger estos datos se ha establecido como criterio la discretización de los valores de potencia de los motores, de manera que el motor principal, puede tomar los valores 2, 1, 0, -1, -2, donde los valores negativos indican marcha atrás y el 0 motor apagado. Por otro lado el motor de hélice de proa toma los valores 1, 0, -1, donde el valor positivo indica que actúa la hélice izquierda, el negativo que actúa la hélice derecha, y 0 apagado. Finalmente, para el giro de timón, se tomarán valores angulares positivos cuando el giro es en sentido horario, y negativo en sentido anti-horario. En cuanto a los valores que puede tomar esta variable es necesario hacer hincapié en que en el entorno simulado, se han tomado valores del giro de la rueda del timón, variando en el rango de 0° a 360° en un giro en el sentido horario y en el rango de 0° a -360° en sentido anti-horario. Sin embargo, el giro del timón se sitúa aproximadamente en el rango de 0° a 30° grados en un caso y de 0° a -30° en el otro, dependiendo en cada caso del buque tomado en consideración. Obsérvese que esta forma de actuar sobre el timón implica la realización de giros suaves en las diferentes actuaciones. Por tanto, la obtención del giro real del timón se realiza a través de la aplicación de un factor de multiplicación según el buque.

3. Control

Puesto que el propósito final de este trabajo es construir un sistema de apoyo que sugiera la maniobra que debe realizar el capitán en función de los obstáculos que tenga a su alrededor, se debe diseñar un algoritmo inteligente que proporcione los valores de los controles. Para que la modificación de estos valores sea automática se ha escogido un método basado en la capacidad de aprendizaje. Por esto, es necesario que sea una técnica basada en el entrenamiento. En este trabajo se emplea un método Neuro-Fuzzy (Mitra et al., 2000) (Marichal et al., 2009) similar al ANFIS (*Adaptive-Neuro-based Fuzzy Inference Systems*) de Jang (Jang, 1993) (Marichal et al., 2011). Lo que se ha hecho es construir un sistema Neuro-Fuzzy para cada uno de los controles disponibles, es decir, se tendrán tres sistemas. Cada Neuro-Fuzzy, una vez entrenado, proporcionará, en función de sus entradas, que son los valores de distancia proporcionados por los 10 sensores, el ángulo de giro de timón, la potencia del motor principal y la del motor de hélices más convenientes según cada situación particular. En este punto hay que mencionar que en los valores de distancia de los sensores no se han considerado errores. Esto se justifica por varios motivos. En primer lugar, las distancias que se detectan son de cientos o decenas de metros, y el posible error que se cometa no será relevante en estos rangos. En segundo lugar, durante la maniobra nunca se va a acercar a las estructuras del muelle a menos de tres o cuatro metros, puesto que desde estas distancias se lanzan las sigas al personal del puerto, encapillando las gazas de los cabos en los respectivos Norays, para finalmente acercar el buque mediante el equipo de amarre.

Finalmente, destacar que el uso de 10 sensores permite que si alguno da valores erróneos, esto sea minimizado por los demás datos recibidos. Si bien el uso de sensores redundantes resultaría de interés a la hora de ayudar en la determinación de posibles lecturas erróneas.

La figura 3 muestra la arquitectura de uno de los sistemas Neuro-Fuzzy, en particular el que proporciona el ángulo del timón. La primera capa representa las funciones de pertenencia y está compuesta por neuronas de Base Radial, cuyas entradas son las entradas del sistema neuro-fuzzy genético y la salida de los nodos viene expresada por (1) donde N_1 es el número de entradas, N_2 el número de nodos en la capa intermedia, mientras que U_i la entrada i -ésima, m_{ij} , σ_{ij} y p_{ij} son el centro de la función de pertenencia, el ancho de la función de pertenencia y la salida de la neurona con entrada i -ésima y salida conectada con el nodo j -ésimo de la capa intermedia.

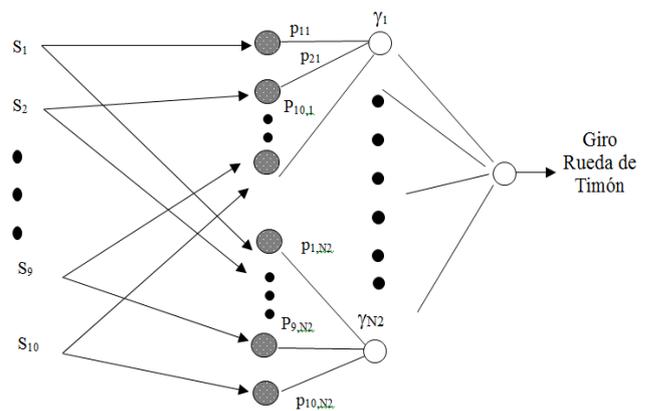


Figura 3: Arquitectura del sistema neuro-fuzzy genético utilizado.

$$p_{ij} = \exp\left(-\frac{(S_i - m_{ij})^2}{2\sigma_{ij}^2}\right) \quad \begin{matrix} j=1,2,\dots,N_2 \\ i=1,2,\dots,N_1 \end{matrix} \quad (1)$$

La segunda capa representa el sistema de reglas y sus salidas se obtienen mediante (2).

$$\lambda_j = \min\left\{p_{1j}, p_{2j}, \dots, p_{N_1j}\right\} \quad j=1, \dots, N_2 \quad (2)$$

En la tercera capa se realiza la defuzzificación y se obtiene la salida global del sistema por (3), siendo N_3 el número de salidas del sistema neuro-fuzzy genético y v_{jk} es el valor estimado de la salida k -ésima proporcionada por el nodo j -ésimo.

$$\Psi_k = \frac{\sum_{j=1}^{N_2} v_{jk} \lambda_j}{\sum_{j=1}^{N_2} \lambda_j} \quad k=1, \dots, N_3 \quad (3)$$

Se comprueba en las ecuaciones presentadas que el sistema descrito depende de tres parámetros que serán obtenidos mediante

un algoritmo de entrenamiento. En particular, nos referimos a los centros de las funciones de pertenencia, su ancho y las salidas estimadas.

El algoritmo de entrenamiento se divide en tres fases, donde las dos primeras son las encargadas de proporcionar los valores iniciales a algunos parámetros y de optimizar el número de nodos de la capa oculta, es decir, el número de reglas. La última fase reajusta los valores de los parámetros obtenidos en las fases anteriores.

3.1. Primera fase

En la primera fase se obtienen los valores iniciales de los centros de las funciones de pertenencia y de las salidas estimadas a través de la aplicación de un algoritmo basado en una Red Neuronal Artificial Kohonen (Kohonen *et al.*, 1989). El vector inicial de pesos del mapa auto-organizativo se calcula como la media entre los valores máximo y mínimo de las entradas proporcionadas por el usuario, y su dimensión será la del número de entradas N_1 más el de salidas N_3 . Por otro lado se determina el nodo 'ganador' para actualizar los pesos según el algoritmo auto-organizativo de Kohonen. Particularmente se emplea un mapa auto-organizativo monodimensional, es decir que únicamente se considera una secuencia de nodos. Además, el número de vectores de pesos escogido es el mismo que el número de neuronas de la capa oculta del sistema neuro-fuzzy genético. Una vez se ha determinado el nodo 'ganador' se calculan los nuevos pesos. Hay que destacar que cada vez que los pesos se actualizan, un nuevo par de entrada-salida deseada se presenta. El siguiente paso sería asignar los valores a los centros de las funciones de pertenencia y a las salidas estimadas. Cada vector de pesos se asocia con cada neurona de la capa oculta, de manera que las funciones de pertenencia y las salidas estimadas se determinan por los componentes del vector de pesos asociados. Los nodos de la capa oculta están directamente relacionados con el número de reglas, por lo que es más apropiado optimizar el número de nodos previamente. Esto hace necesario un proceso de optimización para obtener un número mínimo de reglas.

3.2. Segunda fase

En la fase anterior se asoció cada nodo de la capa oculta o intermedia con una regla generada por el sistema neuro-fuzzy genético, por lo que se tienen N_2 reglas. Teniendo en cuenta la primera fase de aprendizaje donde se han fijado valores para dos de los parámetros del sistema neuro-fuzzy genético, concretamente los centros de las funciones de base radial y los valores estimados, indicados en los párrafos anteriores como m_{ij} , y v_{jk} . Sin embargo, faltan por determinar los parámetros σ_{ij} , correspondientes a las anchuras de las funciones de base radial. De hecho sería conveniente ajustar el número de nodos en la capa intermedia, es decir, decidir que reglas asociadas a esta capa serían de interés en la resolución del problema y de cuales podríamos prescindir. En este sentido en esta fase del entrenamiento se introducen técnicas de algoritmos genéticos (Goldberg *et al.*, 1989). Se plantea como individuo una posible solución, entendida tal como la elección de un cierto conjunto de reglas y un cierto conjunto de valores σ_{ij} , tomando los valores calculados para m_{ij} , y v_{jk} de la fase previa. A estos individuos se les aplican operadores de cruce y mutación, obteniendo nuevas posibles soluciones (individuos). En este caso, se elige como función de costo, la evaluación del error a partir de los valores reales obtenidos en las mediciones realizadas. Una vez concluida

está fase de aplicación de un algoritmo genético podemos tener todos los valores necesarios para configurar un sistema neuro-fuzzy genético final.

3.3. Tercera fase

En esta última fase del entrenamiento se ajustan los parámetros del sistema obtenidos en las fases anteriores. Se emplea el algoritmo de aprendizaje de mínimos cuadrados debido a la similitud de este sistema con una red de base radial (Chen *et al.*, 1991). Este algoritmo tratará de minimizar una función criterio basada en la diferencia entre las salidas proporcionadas y las deseadas. Es importante indicar que en las tres fases de entrenamientos se han utilizado tres paradigmas de la inteligencia artificial, bastante extendidos. Concretamente, en la primera fase, se ha hecho uso de un algoritmo no supervisado, en la segunda fase se ha recurrido a los algoritmos genéticos, mientras que la última fase se ha centrado en la aplicación de un algoritmo de aprendizaje supervisado. Si bien podríamos designar al sistema resultante como sistema neuro-fuzzy, dado que está constituido básicamente por neuronas y una lectura de conjunto nos permitiría extraer una cierta similitud con la estructura de un sistema fuzzy, se ha preferido el término sistema neuro-fuzzy genético, en tanto en cuanto estos algoritmos forman una parte esencial del aprendizaje, determinando las reglas más adecuadas y por tanto incidiendo de forma significativa en las propiedades de generalización del sistema resultante.

4. Resultados

Todas estas fases se aplican a los datos de la tabla construida en el apartado 2. En cada una de las fases se han realizado varios ensayos y en función de las curvas de error que se obtiene en cada caso, se escoge aquel conjunto de parámetros que ofrecen unos resultados adecuados. Con el conjunto elegido se acomete la siguiente fase, en la que nuevamente se realizan varios ensayos y observando las curvas de error se escogen los parámetros. Siguiendo el mismo proceso se acometen las fases de entrenamiento restantes. En cada uno se realizaron 5000 iteraciones, partiendo de 100 como número de reglas iniciales y fijando la velocidad de aprendizaje del algoritmo a 0.01. En la figura 4 se muestra un ejemplo de una gráfica de la evolución de la curva de error de la aplicación del algoritmo genético. Se puede observar que tras 100 generaciones se obtiene un error medio de 0.01697, mientras que el mejor individuo tiene un error de 0.016581

Una vez se han concluido todas las fases de los tres entrenamientos, es decir, el que proporciona el giro del timón, el de la potencia del motor principal y el de la potencia del motor de hélices laterales, se introducen valores que el sistema desconoce. En la tabla 1 se muestran las salidas que proporciona el Neuro-Fuzzy entrenado para indicar el ángulo de giro de la rueda del timón. Hay que hacer hincapié en que los valores de las entradas son desconocidos, lo que quiere decir que el sistema no ha utilizado para entrenar ese conjunto de diez valores de distancia a los obstáculos. La columna que indica la salida real hace referencia al valor que el capitán dio a la variable de control del buque en cada situación, en este caso es el ángulo de giro de la rueda del timón.

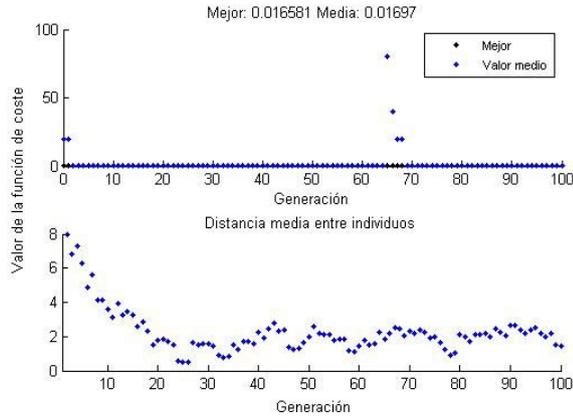


Figura 4: Curva de evolución de error en la aplicación del algoritmo genético.

De esta manera en las siguientes tablas se comparan los valores que indicó el experto con los que proporcionan los sistemas cuando existen diferentes condiciones en su entorno. Para el sistema mostrado en la tabla 1, los mejores resultados se obtuvieron en la fase del algoritmo genético, con un error absoluto medio de entrenamiento de 13.9664 y un error absoluto medio de generalización de 5.1525.

Tabla 1: Comparación entre salida real y la salida del Neuro-Fuzzy en ángulo de giro del timón.

Neuro-Fuzzy ángulo timón	
Salida real	Salida proporcionada
32	32.0012
32	32.0012
-48	-45.1601
0	6.4565
-28	-30.0389
-100	-98.6614
-340	-339.376
0	-8.8795
0	0.93167
0	0.57876
0	1.9076
-48	-46.9214
0	-0.30619

La tabla 2 muestra el caso del Neuro-Fuzzy entrenado para proporcionar la potencia del motor principal. En este sistema se obtuvo en la fase supervisada el mejor resultado, dando un error absoluto medio de entrenamiento de 0.068788 y un error absoluto medio de generalización de 0.051307.

Finalmente en la tabla 3 se presentan los resultados con el Neuro-Fuzzy entrenado para indicar la potencia del motor de hélices laterales. Igual que en el sistema anterior, el mejor resultado aparece en la fase supervisada, con un error absoluto medio de entrenamiento de 0.1361 y un error absoluto medio de generalización de 0.076072. Si se comparan los errores de las tablas 2 y 3 con los de la primera tabla, se observa que son mucho menores. Esto se debe a que el rango de valores con los que se trabaja en el giro de la rueda del timón es mucho mayor, puesto

que puede variar entre 360° y -360° , dependiendo si el giro es horario o antihorario, respectivamente.

Tabla 2: Comparación entre salida real y la salida del Neuro-Fuzzy en potencia del motor principal.

Neuro-Fuzzy potencia motor principal	
Salida real	Salida proporcionada
1	1
1	1.0001
1	1.0463
2	1.9991
1	0.7534
1	0.9285
1	0.9998
1	1.0007
0	0.0005
2	1.8002
1	1
1	1.0005

Tabla 3: Comparación entre salida real y la salida del Neuro-Fuzzy en potencia del motor de hélices laterales.

Neuro-Fuzzy potencia motor lateral	
Salida real	Salida proporcionada
0	0.0025705
1	0.55444
-1	-0.96265
1	0.97941
0	0.0035644
0	-0.0004678
1	0.99998
0	$-2.006 \cdot 10^{-5}$
0	$-8.0844 \cdot 10^{-5}$
0	-0.039058
-1	-1.0025
0	-0.01538

5. Conclusiones.

Los ensayos realizados muestran que los sistemas híbridos propuestos basados en un algoritmo genético y un sistema genético neuro-fuzzy han presentado un entrenamiento satisfactorio y que además han conseguido una buena generalización. Los valores de distancia a los obstáculos que se encuentran alrededor del buque, se pueden emplear como factor externo que indique cómo se debe realizar la maniobra. En el trabajo presentado se toman las distancias de diez sensores distribuidos por el barco, como parámetro externo que hace que el desatraso se realice de determinada forma. Concretamente se trata del ángulo de giro de la rueda del timón, la potencia del motor principal y la potencia del motor de la hélice de proa, como valores de actuación modificables que serán proporcionados por cada sistema neuro-fuzzy genético. Los errores obtenidos son considerablemente bajos, siendo el sistema que proporciona el ángulo de giro de la rueda del timón el que obtiene errores mayores. Esto se debe a que el rango de trabajo es mucho más amplio que en los otros sistemas, donde se han discretizado los posibles valores y en la práctica real sólo tomaron valores de -1, 0 y 1.

Los sistemas construidos no pretenden ser un sistema de pilotaje automático en esta primera fase, sino ser un apoyo a la navegación. Se trataría de proporcionar información extra al piloto sobre la situación que tiene a su alrededor y aconsejarle cómo debería realizar la maniobra. Sobre todo en caso de tratarse de capitanes con menor experiencia en la maniobra, donde el sistema propuesto puede servir para la mejora de sus habilidades en cuanto al pilotaje se refiere. Por otra parte, la metodología expuesta en este artículo nos permitiría extraer las reglas borrosas propias de una buena maniobra a partir de los conocimientos del capitán o práctico, lo que nos permitirá elevar el nivel de automatización de estas operaciones, con lo que ello representa respecto a la disminución de errores humanos, si bien dichos sistemas siempre han de estar bajo la supervisión de estos profesionales. Si bien es cierto que cada buque tiene un comportamiento diferente, el trabajo muestra que la metodología planteada es eficaz para el buque portacontenedores, y se podría hacer extensible a otro tipo de buques, siempre y cuando se realice la fase de entrenamiento a partir de los datos proporcionados por un piloto experto para ese buque concreto, pudiendo generar sistemas entrenados que se le proporcionarían al capitán en función de las características propias del buque (Obra muerta, potencia de las máquinas marinas, arqueado bruto y neto, elementos de maniobra, etc.). Por otra parte, las metodologías expuestas permiten realizar un análisis más exhaustivo de las formas de operar de los diferentes capitanes y prácticos estableciendo reglas de comportamiento a la hora de realizar el pilotaje de un determinado barco que garantice la seguridad y la eficiencia en todo momento de la maniobra de desatracque.

English Summary

Intelligent onboard assistant for maneuvering in port.

Abstract

An assistant onboard system is proposed in order to provide help to the captain. It is a Genetic Neuro-Fuzzy system which is able to advise about how the maneuver should be done depending on each particular situation. Particularly the casting-off is taking into account. In this way, the obstacles situated around the vessel like other boats or the port structures are considered when the maneuver is in process. Depending on the distances the system provides adequate values to the three control parameters: angle turn of tiller, principal engine power and lateral engine power.

Keywords:

Neuro-Fuzzy, genetics algorithm, vessel, maneuver.

Agradecimientos

Este trabajo ha sido financiado por la beca de la Agencia Canaria de Investigación, Innovación y Sociedad de la Información del Gobierno de Canarias cofinanciada con fondos sociales europeos.

Referencias

- Chen, C. F. N., and Grant P. M., "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Networks*, vol. 2, no. 2, pp. 302-309, Mar. 1991
- Colin R. Pratt and Geoff Taylor (2004). AIS – A Pilot's Perspective. *Journal of Navigation*, 57, pp 181-188. doi:10.1017/S0373463304002772.
- Esposito, J.M.; Graves, M., "An algorithm to identify docking locations for autonomous surface vessels from 3-D LiDAR scans," *Technologies for Practical Robot Applications (TePRA), 2014 IEEE International Conference on*, vol., no., pp.1,6, 2014 doi: 10.1109/TePRA.2014.6869160
- Goldberg D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Publishing Company, January 1989.
- Gucma, S. Bąk, A. Jankowski, S. Gućma, M. Pilot Navigation System - a new tool for handling vessels in ports and confined areas. 2008. *Problemy Eksploatacji*, ISSN 1232-9312
- Gucma, L., Bak, A., Gućma, M., Jankowski, S., Zalewski, P., & Perkovic, M. (2010). Laser docking system integrated with Pilot Navigation Support System. Background to high precision, fast and reliable vessel docking. 17th St. In *Petersburg International Conference on Integrated Navigation Systems, St. Petersburg*.
- Herrero, E. R., González, F. J. V., García, E. L., & Pérez, E. M. (2012). Diseño de Experimentos para la Estimación de Parámetros de Modelos de Maniobra Lineales de Buques. *Revista Iberoamericana de Automática e Informática Industrial IIAI*, 9(2), 123-134.
- Hsueh W. J. and Lee Y. J., "A Hydraulic Absorber for Wideband Vibration Reduction in Ship Hulls" *J. Offshore Mech. Arct. Eng.* 116, 43-48 (1994) (6 pages). doi:10.1115/1.2920127
- Hsueh Wen -Jeng "Vibration reduction of main hulls using semiactive absorbers" *Journal of Marine Science and Technology* 1998, Volume 3, Issue 1, pp 50-60 DOI 10.1007/BF01239806
- Jang, J. S. R., (1993), "ANFIS: Adaptive-Network-based Fuzzy Inference Systems" *IEEE Trans. On Systems, Man, and Cybernetics*, vol. 23, pp. 665-685.
- Kohonen T., *Self-Organizing Maps*. Springer Series in Information Sciences, Vol. 30, Springer, Berlin, Heidelberg, New York, 2001.
- Marichal, G. N., Hernández A., Acosta L., González E.(2009). A Neuro-Fuzzy System for Extracting Environment Features Based on Ultrasonic Sensors. *Sensors*. 9, pp. 10,023 - 10,043. ISSN 1424-8220
- Marichal, G., Artés, M., García Prada, J.C., Casanova O.,(2011), "Extraction of rules for faulty bearing classification by a neuro-fuzzy approach" *Mechanical Systems and Signal Processing*, 25, 2073-2082
- Marrero, M^a M., Valero, García, E., Bermejo, A. C. Planificación de la derrota. 1993 Depósito legal tf-1604/92
- Mitra S., Hayashi Y., 2000. Neuro-fuzzy rule generation: Survey in soft computing framework, *IEEE Transactions on Neural Networks*, 11 (3) pp. 748-768.
- Pillich, B.; Buttgenbach, G. ECDIS-the intelligent heart of the hazard and collision avoidance system, *Intelligent Transportation Systems, 2001. Proceedings.* 2001 IEEE, vol., no., pp.1116,1119, 2001 doi: 10.1109/ITSC.2001.948818
- Perez, N. Operaciones de atraque y desatracque, fondeo y recalada de buques en los puertos canarios. Tesis 2012
- Santos, M. (2011). Aplicaciones Exitosas de Control Inteligente a Casos Reales. Science Direct, *Revista Iberoamericana de Automática e Informática Industrial*.
- Shipsimulator. www.shipsim.com
- Zhang Li-hua¹, Zhu Qing², LIU Yan-chun¹, LI Shu-jun. A method for automatic routing based on ECDIS. *Journal of Dalian Maritime University* 2007-03.