XXVI Jornadas de Automática

CEA

Universitat d'Alacant
Universidad de Alicante

UNIVERSITAS Miguel Hernández

2005

ja05

ISBN: 84-689-0730-8

Editores: Fernando Torres, Oscar Reinoso

La presente publicación recoge las contribuciones realizadas a las XXVI Jornadas de Automática, celebradas en Alicante y Elche entre el 7 y el 10 de septiembre de 2005, organizadas de forma conjunta por la Universidad de Alicante y la Universidad Miguel Hernández de Elche, bajo la dirección del Comité Español de Automática CEA-IFAC.

Editores:

Fernando Torres
Universidad de Alicante
Departamento de Física, Ingeniería de Sistemas y Teoría de la Señal
e.mail: Fernando.Torres@ua.es

Oscar Reinoso
Universidad Miguel Hernández de Elche
Departamento de Ingeniería de Sistemas Industriales
e-mail: o.reinoso@umh.es

XXVI JORNADAS DE AUTOMÁTICA
ALICANTE – ELCHE, 2005

ISBN: 84-689-0730-8

ENTORNO DE SIMULACIÓN INTERACTIVA PARA EL CONTROL DE POSICIONAMIENTO DINÁMICO DE UNA PLATAFORMA MARINA AMARRADA

Joaquín Aranda Almansa
jaranda@dia.unes.es
UNED. ETS. Ingeniería Informática

Dictino Chaos García
dchaos@bec.unes.es
UNED. ETS. Ingeniería Informática

Sebastián Dormido Canto
sebas@dia.unes.es
UNED. ETS. Ingeniería Informática

José Manuel Díaz Martínez
josema@dia.unes.es
UNED. ETS. Ingeniería Informática

Rocío Muñoz Mansilla
rmunoz@dia.unes.es
UNED. ETS. Ingeniería Informática

Resumen

Se ha desarrollado un entorno de simulación interactiva de un modelo de plataforma marítima amarrada empleando conjuntamente Ejs (Easy Java Simulations) y Simulink.

Este problema de control tiene un gran interés dado que se está simulando el posicionamiento dinámico de un sistema subactuado y esta es una situación común en la industria naval.

El problema de posicionamiento dinámico se aborda mediante dos técnicas, por una parte se implementa un controlador basado en estructuras clásicas PID y por otra se diseña un controlador mediante la técnica de control robusto QFT. Ambas técnicas de control se comparan con un controlador robusto H_∞.

La simulación desarrollada permite visualizar el movimiento de la plataforma y el oleaje, las fuerzas y momentos que el oleaje ejerce sobre ella, la posición y orientación de la plataforma y la acción de control que se aplica. Asimismo permite modificar de forma interactiva el estado de la mar y seleccionar el tipo de controlador que se utiliza para posicionar la plataforma.

Palabras Clave: Plataforma marina, Ejs, Simulink, posicionamiento dinámico, control robusto, QFT.

1 INTRODUCCIÓN

En este trabajo se realiza una simulación interactiva de una plataforma flotante considerando dos grados de libertad en el entorno Ejs (Easy Java Simulations) [7] y [2]. Se usa como motor de simulación la plataforma Simulink de Matlab y los resultados son exportados por Ejs que se encarga de la presentación gráfica de los resultados y de añadir interactividad al modelo.

La simulación del modelo de plataforma flotante permite ensayar diversas estrategias de control para el posicionamiento dinámico de la misma. El tipo de controlador puede seleccionarse de forma interactiva en tiempo de simulación lo que permite comparar el rendimiento de cada uno de los controladores aplicados.

En este trabajo se presentan tres métodos: PID, H_∞ y QFT [1],[3],[4] y [6].

2 DESCRIPCIÓN DEL SISTEMA

2.1 MODELO FÍSICO

El sistema consiste en una plataforma flotante anclada al fondo del océano que posee dos motores para su posicionamiento [5]. La figura 1 muestra un esquema de la plataforma.

El modelo del sistema recoge dos grados de libertad para la plataforma, el desplazamiento horizontal "Y" y el ángulo de giro ϕ.

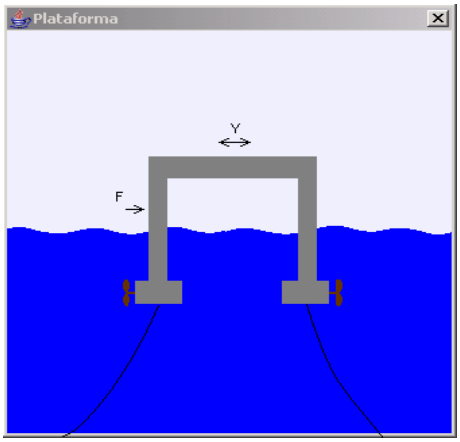

Figura 1: Plataforma flotante amarrada.

Sobre el sistema actúa una fuerza F y un momento ejercido por el oleaje M.
La salida Y del sistema es medida por el sensor w cuya función de transferencia es $w(s)=1+0.1/s$. Por otro lado el actuador produce una fuerza $F_u=\frac{u}{0.7s+1}$, ver Figura 2.

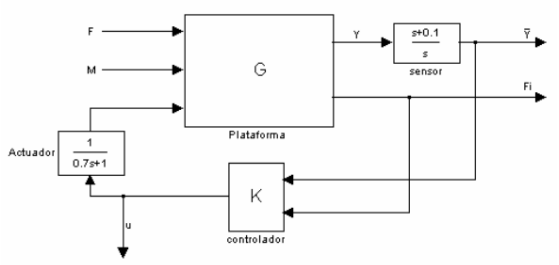

Figura 2: Diagrama de bloques de la plataforma.

El modelo en ecuaciones de estado del sistema es el siguiente

$$\dot{x}=Ax+B\begin{pmatrix}F\\M\\F_u\end{pmatrix}\quad\begin{pmatrix}Y\\\phi\end{pmatrix}=Cx \quad (1)$$

Donde las matrices A, B y C son las mostradas a continuación:

$$A=\begin{pmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ -0.1010 & -0.1681 & -0.04564 & -0.01075\\ 0.06082 & -2.1407 & -0.05578 & -0.1273\end{pmatrix}(2)$$

$$B=\begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0.1179 & 0.1441 & 0.1476\\ 0.1441 & 1.7057 & -0.7557\end{pmatrix}(3)$$

$$C=\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{pmatrix}(4)$$

2.2 MODELO SIMULINK

Se ha implementado en Simulink tanto el modelo de la plataforma, como los actuadotes, como un modelo de las perturbaciones causadas por el oleaje.

El oleaje se simula mediante un bloque Simulink que genera las señales de fuerza y momento.

La fuerza F se descompone en la suma de 3 señales sinusoidales F11, F12, F13 y una componente constante c=0.03. El espectro del las señales depende del tipo de marea, por lo que para la simulación se utilizan tres espectros que representan los estados de la mar Marejada, Fuerte Marejada y Mar Gruesa cuyas características se resumen en las Tablas 1, 2 y 3 respectivamente:

Fuerza	Frecuencia Rad./s	Amplitud
F11	0,45	0,00467
F12	0,9	0,00467
F13	1,5	0,03736

Tabla 1: Espectro utilizado para la simulación de la Marejada.

Fuerza	Frecuencia Rad./s	Amplitud
F11	0,3	0,01
F12	0,6	0,01
F13	1	0,08

Tabla 2: Espectro utilizado para la simulación de la Fuerte Marejada.

Fuerza	Frecuencia Rad./s	Amplitud
F11	0,222	0,01733
F12	0,444	0,01733
F13	0,74	0,13864

Tabla 3: Espectro utilizado para la simulación de la Mar Gruesa.

El momento M ejercido sobre la plataforma se modela como una única señal sinusoidal cuya frecuencia y amplitud depende del estado de la mar como se resume en la Tabla 4

Marejada	6,75	0,4203
F. Marejada	4,5	0,9
Gruesa	3,33	1,5597

Tabla 4: Espectro de momentos ejercidos por el oleaje en función del estado de la mar

Cada espectro de marea es generado por un bloque similar al que se muestra en la Figura 3.

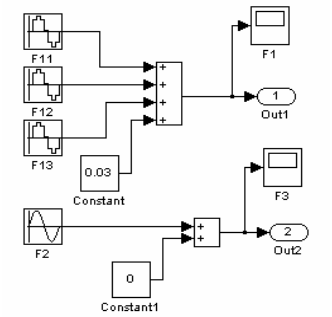

Figura 3: Simulador del espectro de oleaje simulink.

Con este bloque se construye un generador de oleaje con selector del estado de la mar mediante un bloque "multiport swhitch".

El modelo completo de la plataforma se muestra en la Figura 4.

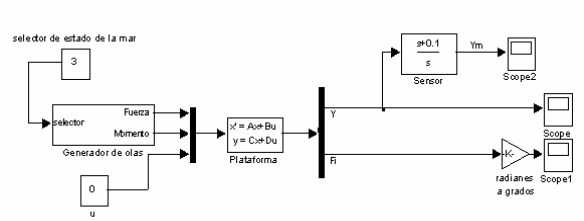

Figura 4: Modelo Simulink de la plataforma.

3 ESTRATEGIAS DE CONTROL

Sobre el modelo anterior se ensayan tres estrategias de control, la primera de ellas está basada en un controlador con dos PID's, uno para el ángulo y otro para el desplazamiento.

La segunda estrategia de control consiste en un controlador H_∞ diseñado en [5].

La tercera es un controlador basado en la técnica de diseño QFT.

3.1 CONTROLADOR PID

El controlador basado en PID's tiene la estructura de la Figura 5.

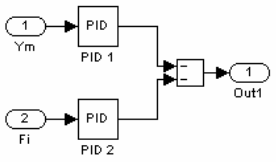

Figura 5: Control PID.

El controlador se diseña desde una perspectiva de control multivariable centralizado y se aplica una técnica de síntesis que tiene como objetivo la optimización de los parámetros de los PID's de modo que se obtenga el máximo rechazo ante las perturbaciones externas que produce el oleaje.

3.1.1 Sintonía.

La sintonía de los controladores se trata como un problema global de optimización para el cual la función de coste es el rechazo ante las perturbaciones. La Función de coste empleada para la optimización se implementa mediante el bloque Simulink de la Figura 6.

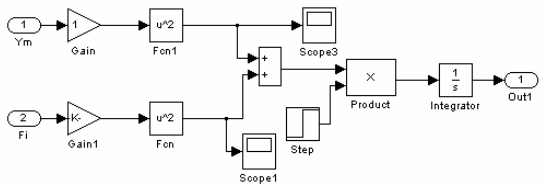

Figura 6: función de coste.

Los parámetros que se obtienen mediante un proceso de optimización no lineal empleando la toolbox de optimización de Matlab son los siguientes:

Kp1=1.64 Kp2=0.10
Ki1=0.687 Ki2=0.030
Kd1=1.82 Kd2=-0.24

3.1.2 Implementación final.

El controlador final debe permitir interactividad con Ejs, por tanto el controlador se implementa de tal modo que sus parámetros sean tomados como señales externas al bloque.

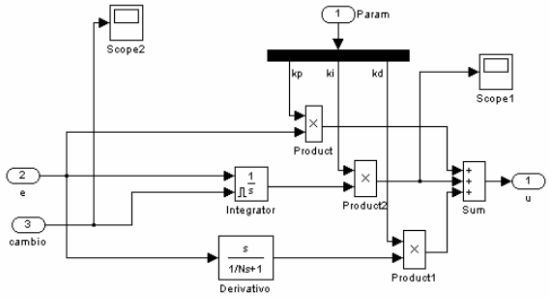

Figura 7: Implementación del bloque PID de parámetros externos.

Por otra parte el cambio en el tipo de controlador produce saltos debidos a la acción integral acumulada en periodos en los que el controlador no

está actuando. Para solucionar este problema y tener un control bumpless se pone a cero la acción integral mientras el controlador no está activo. Esto se lleva a cabo mediante la señal de reset externo por nivel "cambio".

3.2 CONTROLADOR H_∞

El controlador H_∞ es el diseñado en [5].
La estructura del lazo de control es la siguiente:

$$u = k_1(s)Y + k_2(s)\phi \quad (5)$$

Donde las funciones de transferencia son las siguientes:

$$k_1(s) = -20.36\frac{(s+1.43)(s+338.36)(s^2+254.03s+3.472x10^4)(s^2+0.329s+3.67x10^{-2})}{s(s+3.25)(s^2+172.11s+2.722x10^4)(s^2+37.37s+1.051x10^3)} \quad (6)$$

$$k_1(s) = 64.37\frac{(s+1.43)(s+0.23)(s+411.41)(s^2+51.38s+2.065x10^3)}{(s+3.25)(s^2+172.11s+2.722x10^4)(s^2+37.37s+1.051x10^3)} \quad (7)$$

3.3 CONTROLADOR QFT

La técnica QFT es una técnica de diseño en el dominio de la frecuencia de sistemas de control robusto, que fue introducida por Horowitz [1]. El fundamento de QFT reside en la realimentación que se requiere en los casos en los que la respuesta en lazo abierto no encuentra el comportamiento adecuado, debido a incertidumbres en la dinámica y/o incertidumbres en las señales de entrada ó salida (perturbaciones). Gracias a la realimentación, se consigue que el sistema tenga la respuesta requerida. Nuestro caso particular presenta un modelo bien definido de la planta y unas perturbaciones en la señal de salida, que corresponden con las respuestas de fuerza F y momento M generadas por el oleaje.

El procedimiento de diseño QFT envuelve tres pasos básicos:

-Cálculo de las curvas de restricción (bounds)
-Diseño del controlador (loop shaping), y
-Análisis del diseño.

3.3.1 Curvas de restricción

QFT convierte las especificaciones en lazo cerrado en restricciones de una *función en lazo abierto L(jω).* La función lazo abierto es entonces diseñada para satisfacer simultáneamente sus restricciones y a la vez la estabilidad en lazo cerrado. Esta función lazo se define como el producto de la función de transferencia del controlador y de la planta

$$L(j\omega) = G_{control}(j\omega)\cdot P_{planta}(j\omega) \quad (8)$$

Las especificaciones QFT se dan en términos de respuesta en frecuencia. Para este caso, se van a emplear los márgenes de estabilidad de fase y ganancia (9), reducción de sensibilidad ó rechazo a perturbaciones de salida (10), y esfuerzo de control (11). Por tanto, las restricciones originales que se tienen en el dominio temporal se deben traducir al dominio de la frecuencia.

- márgenes de fase y ganancia

$$\left|\frac{G_{plant}G_{control}}{1+G_{plant}G_{control}}\right| \le W_{s1} \quad (9)$$

- reducción de sensibilidad

$$\left|\frac{1}{1+G_{plant}G_{control}}\right| \le Ws_2 \quad (10)$$

- esfuerzo de control

$$\left|\frac{G_{control}}{1+G_{plant}G_{control}}\right| \le Ws_3 \quad (11)$$

3.3.2 Diseño del controlador

Para realizar el diseño QFT, se utiliza el diagrama de Nichols. En este diagrama se representa la función de lazo abierto L(jω) y las curvas de restricción en el dominio de la frecuencia. El diseño del control consiste en el diseño de una función lazo L(jω) que satisfaga el peor caso (intersección) de todas las restricciones.

El ejemplo de la plataforma es un sistema SIMO, con dos salidas (Y, ϕ) y una entrada de control (u = k1·Y + k2·ϕ). El problema de diseño de los controladores se resuelve mediante un proceso iterativo ó multifase transformando el problema en el diseño de dos sistemas SISO secuenciales.

En la primera etapa se considera inicialmente la función de transferencia k2 =0. Se plantea un primer problema SISO, en el que se diseña un control *k1* a partir las especificaciones siguientes: W_{s1Y}= 3, W_{s2Y}=3, W_{s3Y}= 15.

En una segunda etapa, considerando el diseño de k1, se plantea un segundo sistema SISO, en el que se diseño un control k2 con las especificaciones $W_{s1\phi}$= 2.8, $W_{s2\phi}$=16, $W_{s3\phi}$= 15

En las etapas consecutivas, a partir del diseño del control de uno de los sistemas, se procede al re-diseño de un nuevo control para el otro sistema, y

así de forma iterativa hasta que se consiga que las especificaciones iniciales sean satisfechas.

El caso particular de la plataforma consta de 5 etapas, en las que las funciones de transferencia finales son (12) y (13).

$$k1(s) = -0.28 \cdot \frac{\left(\frac{1}{0.52}s+1\right)}{\left(\frac{1}{1.95}s+1\right)} \quad (12)$$

$$k2(s) = \frac{\left(\frac{1}{0.26^2}s^2 + \frac{2\cdot 0.02}{0.26}s+1\right)}{\left(\frac{1}{2.4^2}s^2 + \frac{2\cdot 0.7}{2.4}s+1\right)} \quad (13)$$

3.3.3 Análisis del diseño

Una vez diseñada la ley de control, se debe comprobar que el sistema en lazo cerrado cumple los objetivos de control fijados inicialmente. En este caso el sistema es verificado en Matlab/Simulink y Ejs.

4 SISTEMA COMPLETO Y CONEXIÓN CON EJS

Easy Java Simulations es una herramienta software creada, específicamente, para el desarrollo de simulaciones interactivas en el campo de la física [7].

La arquitectura de *Ejs* deriva del paradigma modelo-vista-control. Este paradigma expresa que una simulación interactiva se compone de tres partes perfectamente diferenciadas:

1. el **modelo**, que describe el fenómeno bajo estudio en término de variables y relaciones entre éstas expresadas mediante algoritmos en un computador.

2. el **control**, que define las acciones que un usuario puede ejecutar sobre una simulación,

3. la **vista**, que recoge una representación de los diferentes estados que el fenómeno puede presentar.

La simulación realizada consta de dos partes. En primer lugar existe un **modelo** Simulink que realiza la simulación externa de la plataforma. En la segunda parte se desarrolla un proyecto de simulación en Ejs cuyas variables se conectan con las del sistema en Simulink y en la cual se realiza la parte de la **vista** y el **control**.

Una vez definido el modelo y la vista de la simulación interactiva, *Ejs* genera el código Java, compila el programa, comprime todos los ficheros resultantes en un único archivo, y genera un conjunto de páginas HTML que permiten ejecutar la simulación como un applet.

4.1 MODELO SIMULINK COMPLETO

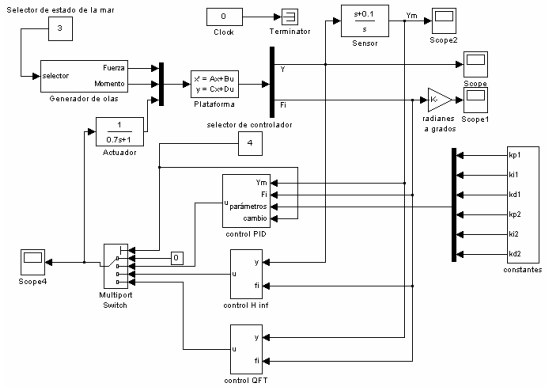

Figura 8: Modelo completo Simulink de la plataforma y los controladores.

El modelo de la Figura 8 contiene todas las variables que van a ser accesibles desde Ejs. El modelo consta de los bloques que simulan la planta a los cuales se les ha añadido los tres bloques de control: "control PID", "control H inf" y "control QFT".

Los parámetros del controlador PID son establecidos por Ejs en la simulación, de modo que los valores que proporciona el bloque "constantes" solo se utilizan cuando se hace funcionar el modelo de forma autónoma en Simulink durante la fase de depuración.

La señal de control puede ser nula (plataforma libre o en lazo abierto) o bien se selecciona de entre los 3 controladores disponibles. Esta acción la realiza el bloque "multiport switch" y se sincroniza con Ejs a través de la variable que procede del bloque "selector de controlador".

4.2 MODELO EJS

Ejs desarrolla la vista y el control de la simulación empleando como motor de simulación el modelo Simulink descrito anteriormente. El esquema del modelo Ejs y la conexión con Matlab/Simulink se muestra en la Figura 9.

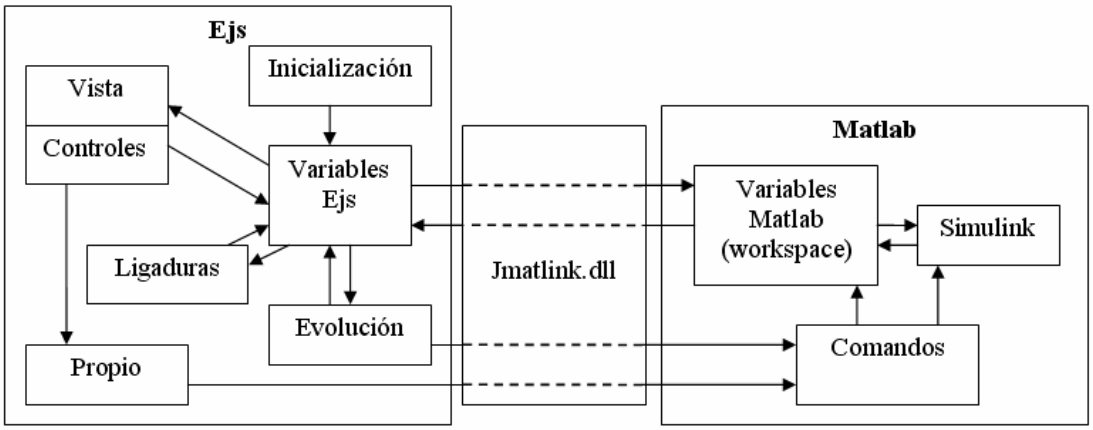

Figura 9: esquema de conexión de las partes del modelo y EJS.

Cada una de las partes que se muestran en el diagrama de la Figura 9 se describe a continuación.

4.2.1 Variables

El modelo Ejs define tres ventanas de variables. Las variables del dibujo y de ventanas se utilizan internamente en Ejs para el control y la vista mientras que las variables compartidas se encuentran conectadas al modelo Simulink.

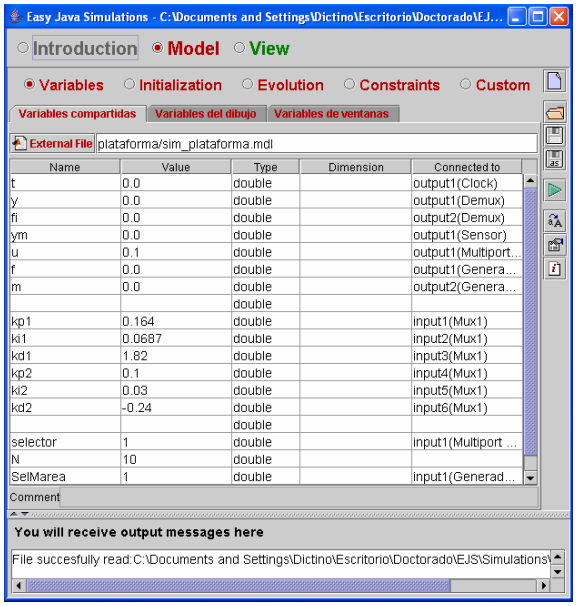

Figura 10: Ventana de variables compartidas entre Matlab y Ejs

4.2.2 Inicialización

En esta parte se inicializan la forma de onda que se utilizará para realizar la simulación del oleaje.

4.2.3 Evolución

En este apartado se realiza la llamada a la simulación externa en simulink

Se simula el sistema en N instantes de tiempo y se muestrea los valores de las variables.
El parámetro N permite de este modo alterar la velocidad de refresco de la vista y por tanto de la simulación.

También se modifica el valor de la variable booleana imagen. Este valor se utiliza durante la simulación para conmutar entre dos imágenes de la plataforma de modo que se produce el efecto del giro de las hélices.

4.2.4 Ligaduras

En la parte de restricciones se recalcula el dibujo de las olas tomando como referencia el valor de la fuerza y el momento.

4.2.5 Propio

En esta parte se definen las funciones que permiten mostrar y ocultar el modelo durante la simulación.

4.3 VISTA

La vista del modelo se organiza en 5 ventanas:

-Controles simulación.
-Ventana plataforma.
-Ventana posiciones.
-Ventana control.
-Ventana fuerzas.

4.3.1 Controles simulación

Es la ventana principal de la aplicación, contiene los botones básicos de control de simulación "play", "pause", "paso" y "reset" en el cuadro simulación. También posee un panel de selectores que costa de una barra deslizante para la selección del número de pasos N que se ejecutan entre dos actualizaciones consecutivas de la vista y un panel de separadores con cuatro apartados:

-El apartado general selecciona que partes del modelo serán visibles.

-El apartado plataforma muestra u oculta las olas en la simulación de la plataforma.

-El apartado control permite seleccionar el tipo de controlador aplicado.

-El apartado oleaje sirve para modificar el estado de la mar.

El aspecto gráfico de la ventana se muestra en la Figura 11.

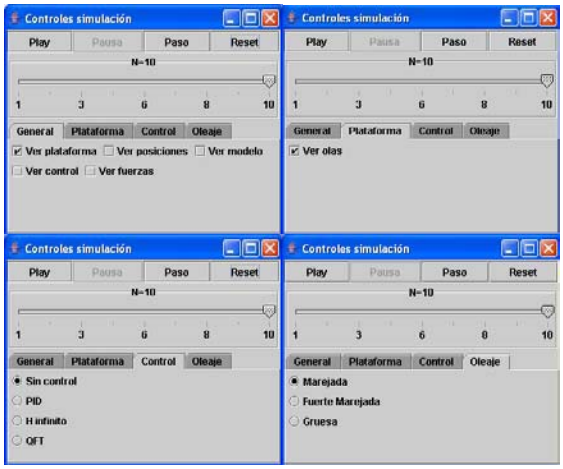

Figura 11: Aspecto gráfico de la ventana "controles Simulación"

4.3.2 Ventana Plataforma

Esta es la ventana en la que se ve gráficamente el movimiento de la plataforma, contiene un panel de dibujo y en su interior tres objetos.

-Olas: Consiste en un polígono que se utiliza para representar el oleaje.

-Plataforma 1 y 2: son dos imágenes jpg de la plataforma que se muestran en la posición definida por Ym y giradas un ángulo ϕ.

El aspecto final de la ventana se muestra en la Figura 12.

Figura 12: Aspecto gráfico de la ventana "Plataforma"

4.3.3 Ventana Posiciones

Costa de tres paneles con ejes en los cuales se representa la posición de la plataforma. La diferencia esencial entre el primer y segundo panel es que mientras el primero tiene escala fija el segundo es auto ajustable permitiendo ver simultáneamente el comportamiento de la posición a gran escala y al detalle como puede apreciarse en la Figura 13.

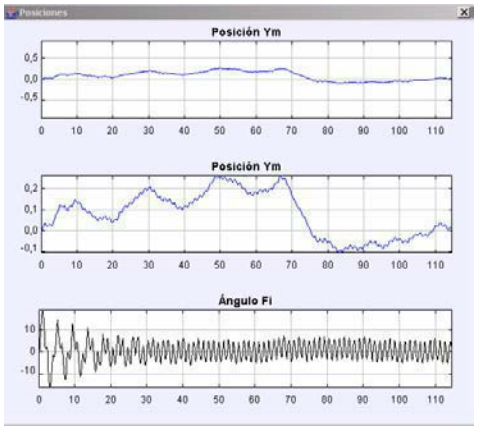

Figura 13: Aspecto gráfico de la ventana "Posiciones"

4.3.4 Ventana Control

Se trata de una ventana simple con un panel con ejes que a su vez contiene un trazo que se muestra la señal de control como se muestra en al Figura 14.

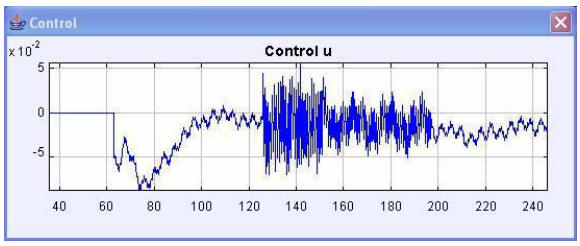

Figura 14: Aspecto gráfico de la ventana "control"

4.3.5 Ventana Fuerzas

En esta ventana se representan dos gráficas, una con la fuerza y otra con el y el momento aplicado a la plataforma. Su aspecto gráfico se muestra en la Figura 15.

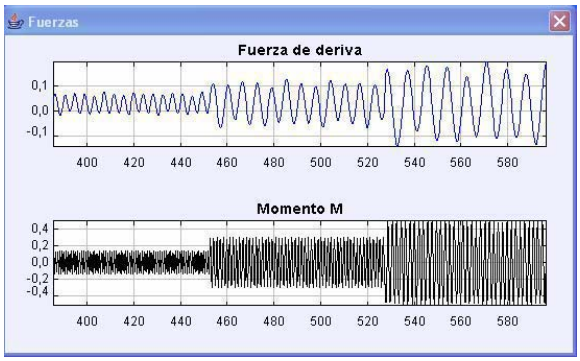

Figura 15: Aspecto gráfico de la ventana "Fuerzas"

5 CONCLUSIONES

En este artículo se ha mostrado un entorno para la simulación del posicionamiento dinámico de una plataforma flotante amarrada empleando Ejs y Simulink.

Se pone de manifiesto la gran facilidad que tiene Ejs para generar simulaciones interactivas ya que permite generar las vistas y el control de la simulación de forma simple y eficiente. Además se explota la potencia de Simulink como entorno de modelado aprovechando la conexión entre Simulink y Ejs.

Dentro de este entorno de simulación se han implementado tres controladores distintos. Esto permite usar la simulación como plataforma de prueba de diversas estrategias de control para el posicionamiento dinámico bajo diferentes situaciones de operación. Además el modelo es fácilmente ampliable para incluir cualquier otro tipo de controlador que se desee poner a prueba.

Agradecimientos

Se agradece al Ministerio de Educación y Ciencia la financiación y apoyos recibidos dentro del proyecto DPI2003-09745-C04-01.

Referencias

[1] Borguesani, C., Chait, Y., Yaniv, O.(1995). "Quantitative Feedback Theory Toolbox - for use with MATLAB". The Mathworks Inc.,Natick, MA.

[2] Esquembre, F. (2004). "Creación de simulaciones para la enseñanza de la Física". Pearson Prentice Hall. ISBN 84-205-4009-9.

[3] Horowitz, I.M. (2001). "Survey of Quantitative feedback design theory (QFT), In Int. J. of Robust and Nonlinear control". 2001, 11:887-921.

[4] Horowitz, I.M. (2003). "Some ideas for QFT research". In: Int. J. Robust Nonlinear Control, 2003; 13:599-605.

[5] Scherer, C., Gahinet, P. y Chilai, M. (Julio 1997) "Multiobjetive Ounput-Feedback Control via LMI Optimization" IEEE Transactions on Automatic Control, Vol. 42 Nº7.

[6] Yaniv, D. (1999). "Quantitative Feedback Design of Linear and Nonlinear Control Systems". Kluwer Academic Publishers.

[7] Página web oficial de Easy Java Simulations http://fem.um.es/Ejs.