

## Trabajo Fin de Máster

Modelado y control no lineal de un puente grúa

Modeling and non-linear control of an overhead  
crane

Autora

Carla Cabrejas Escosa

Director

Édgar Ramírez Laboreo

Titulación del autor

Máster en Ingeniería Electrónica

Escuela de Ingeniería y Arquitectura  
Año 2023 - 2024

# Tabla de contenidos

Resumen .....	6
1. Introducción .....	7
1.1. Objetivos y alcance .....	7
1.2. Descripción .....	8
1.2.1. Física y dinámica de un puente grúa .....	8
1.2.2. El puente grúa de INTECO .....	10
1.3. Estado del arte .....	12
1.4. Estructura de la memoria .....	13
1.5. Cronograma .....	14
2. Modelado .....	15
2.1. Modelo de referencia .....	15
2.2. Modelo no lineal .....	22
2.2.1. Estimación de parámetros .....	26
2.3. Linealización .....	29
2.3.1. Descripción .....	29
2.3.2. El modelo .....	30
2.4. Cálculo del error .....	34
3. Planitud diferencial .....	35
3.1. Definición .....	35
3.2. El modelo .....	35
3.3. Diseño de trayectorias .....	36
4. Control .....	39
4.1. Control con prealimentación .....	39
4.2. Control con realimentación .....	40
4.3. Control con prealimentación y realimentación .....	43
4.4. Cálculo del error .....	44
5. Conclusiones y trabajo futuro .....	46
5.1. Conclusiones .....	46
5.2. Líneas futuras .....	46
Bibliografía .....	48
Anexo A. Código C .....	51
Anexo B. Función de Matlab con modelo no lineal .....	55
Anexo C. Función de Matlab con modelo lineal .....	57

# Lista de figuras

Figura 1. Partes que conforman el puente grúa .....	9
Figura 2. Puente grúa de INTECO .....	10
Figura 3. Interfaz de Simulink .....	11
Figura 4. Driver principal .....	11
Figura 5. Ejes de movimiento .....	12
Figura 6. Ejes y ángulos .....	12
Figura 7. Cronograma .....	14
Figura 8. Esquema del Puente grúa .....	15
Figura 9. Modelo de referencia .....	19
Figura 10. Bloque de conversión a coordenadas de la carga .....	20
Figura 11. Fuerza aplicada al modelo .....	20
Figura 12. Resultados del modelo de referencia .....	21
Figura 13. Modelo no lineal con parámetros iniciales .....	22
Figura 14. Movimiento en el eje Z .....	23
Figura 15. Cálculo de la fuerza de fricción .....	23
Figura 16. Resultados del modelo no lineal con parámetros iniciales .....	25
Figura 17. Resultados del modelo lineal con estimación de parámetros .....	28
Figura 18. Modelo lineal .....	31
Figura 19. Resultados del modelo lineal .....	32
Figura 20. Ángulos Alpha y Beta .....	33
Figura 21. Modelo con movimiento en una única dirección .....	35
Figura 22. Trayectoria deseada y fuerza necesaria .....	38
Figura 23. Control con prealimentación .....	39
Figura 24. Resultados del control con prealimentación .....	39
Figura 25. Control con realimentación .....	41
Figura 26. Resultados del control con realimentación para las primeras constantes .....	41
Figura 27. Resultados del control con realimentación para las segundas constantes .....	42
Figura 28. Control con prealimentación y realimentación .....	43
Figura 29. Resultados para el control combinado para las primeras constantes .....	43
Figura 30. Resultados para el control combinado para las primeras constantes .....	44

# Lista de tablas

Tabla 1. Parámetros iniciales.....	26
Tabla 2. Parámetros estimados .....	27
Tabla 3. Variables de estado y punto de equilibrio.....	30
Tabla 4. Comparación de los errores de los modelos .....	34
Tabla 5. Constantes iniciales del control con realimentación .....	41
Tabla 6. Constantes finales del control con realimentación .....	42
Tabla 7. Comparación de los errores de los distintos tipos de control .....	45

# Lista de ecuaciones

Ecuación (1). Cálculo de X de la carga.....	16
Ecuación (2). Cálculo de Y de la carga.....	16
Ecuación (3). Cálculo de Z de la carga.....	16
Ecuación (4). Componente X de la tensión en la cuerda .....	16
Ecuación (5). Componente Y de la tensión en la cuerda.....	16
Ecuación (6). Componente z de la tensión en la cuerda .....	16
Ecuación (7). Fuerzas que actúan sobre el carro en el eje X .....	16
Ecuación (8). Fuerzas que actúan sobre el carro en el eje Y.....	16
Ecuación (9). Componente x de la tensión en la cuerda.....	17
Ecuación (10). Componente Y de la tensión en la cuerda .....	17
Ecuación (11). Componente z de la tensión en la cuerda .....	17
Ecuación (12). Aceleración de la carga en x .....	17
Ecuación (13). Aceleración de la carga en Y .....	17
Ecuación (14). Aceleración de la carga en Z .....	17
Ecuación (15). Aceleración del carro en X .....	17
Ecuación (16). Aceleración del carro en Y .....	17
Ecuación (17). Aceleración de la carga en X .....	17
Ecuación (18). Aceleración de la carga en Y .....	17
Ecuación (19). Aceleración de la carga en Z .....	18
Ecuación (20). Aceleración del carro en X.....	18
ecuación (21). Aceleración del carro en Y.....	18
Ecuación (22). Aceleración de la cuerda en Z.....	18
Ecuación (23). Aceleración de la variación de Alpha.....	18
Ecuación (24). Aceleración de la variación de Beta.....	19
Ecuación (25). Fuerza que se opone al movimiento en X.....	19
Ecuación (26). Fuerza que se opone al movimiento en Y .....	19
Ecuación (27). Fuerza que se opone al movimiento en Z.....	19
Ecuación (28). Ecuación de estados en un modelo de estados.....	29
Ecuación (29). Ecuación de salida en un modelo de estados.....	29
Ecuación (30). Raíz del error cuadrático medio.....	34
Ecuación (31). Fuerza en el eje Y.....	36
Ecuación (32). Componente Y de la tensión en la cuerda.....	36
Ecuación (33). Componente Z de la tensión en la cuerda.....	36
Ecuación (34). Relación entre Sy y Sz.....	36
Ecuación (35). Tensión en la cuerda.....	36
Ecuación (36). Ángulo Alpha.....	36
Ecuación (37). Aceleración del carro en el eje Y.....	36
Ecuación (38). Posición del carro en el eje Y.....	36
Ecuación (39). Trayectoria deseada definida a trozos.....	37
Ecuación (40). Posición en Y en el instante inicial.....	37
Ecuación (41). Posición en Y en el instante medio.....	37
Ecuación (42). Velocidad en Y en el instante inicial.....	37
Ecuación (43). Velocidad en Y en el instante medio.....	37
Ecuación (44). Aceleración en Y en el instante inicial.....	37
Ecuación (45). Aceleración en Y en el instante medio.....	37

Ecuación (46). Ecuación del controlador.....	40
Ecuación (47). Transformada de Laplace de la ecuación del controlador.....	40

# Resumen

Este trabajo aborda el modelado y control no lineal de un puente grúa, una estructura vital en la industria para el movimiento seguro de cargas pesadas. Utilizando una maqueta a escala INTECO, uno de los objetivos principales es desarrollar un modelo preciso que describa el movimiento del puente grúa de la forma más exacta posible. Otro objetivo clave es implementar un control que permita ejecutar trayectorias específicas con la mínima oscilación de la carga posible. Esto es fundamental para evitar accidentes que puedan dañar tanto objetos como a operarios, o incluso provocar derrames de sustancias peligrosas transportadas por la carga.

En este estudio se profundiza en la mecánica y física del movimiento de los puentes grúa, explorando conceptos como la descripción detallada de la maqueta utilizada y los fundamentos teóricos detrás de su funcionamiento.

Se realiza una comparación entre el modelo de referencia proporcionado por el fabricante y un modelo mejorado, cuyos parámetros son estimados para minimizar el error cometido durante las simulaciones y pruebas experimentales.

Además, se lleva a cabo la linealización del modelo con el propósito de optimizar los recursos computacionales requeridos para su simulación, manteniendo una representación aproximada del comportamiento del sistema.

Se introduce el concepto de planitud diferencial, diseñando una trayectoria de referencia que sirve como base para el control preciso del movimiento de la carga a lo largo del puente grúa.

Posteriormente, se desarrollan y comparan tres estructuras de control diferentes para evaluar su rendimiento y determinar cuál de ellas ofrece los mejores resultados en términos de precisión.

Finalmente, se describen las conclusiones del trabajo. Por un lado, el modelo de referencia no representa fielmente la realidad, por lo que es necesario llevar a cabo algunos cambios. Linealizar el modelo hace que este pierda bastante precisión, aunque a grandes rasgos puede ser útil para intuir la respuesta del sistema aproximada. Por último, la carga puede controlarse para llevar a cabo una trayectoria deseada sin apenas oscilación.

Este trabajo no solo contribuye al avance en el modelado y control de puentes grúa, sino que también establece bases para futuras investigaciones que busquen mejorar la seguridad y eficiencia en el manejo de cargas pesadas en entornos industriales.

# 1. Introducción

En el ámbito industrial, la manipulación eficiente y segura de cargas pesadas es esencial para garantizar la productividad y la seguridad en los procesos de producción y almacenamiento. En este contexto, el puente grúa constituye una herramienta clave, una estructura resistente y versátil que permite el levantamiento y movimiento de cargas en entornos industriales, almacenes y otros espacios de trabajo.

Un puente grúa típico está compuesto por una viga metálica horizontal, conocida como puente, que se desplaza a lo largo de rieles montados en los extremos de una estructura elevada. Esta viga puede desplazarse a lo largo del área de trabajo. En la viga principal del puente grúa se instala un mecanismo de elevación, que puede desplazarse verticalmente para levantar y bajar cargas con precisión y control.

El funcionamiento óptimo del puente grúa depende en gran medida de la comprensión de los principios físicos y dinámicos por los que se rige, así como del uso de tecnologías electrónicas para controlar y optimizar su rendimiento.

Por ello, en el presente trabajo fin de máster se va a llevar a cabo el modelado y control no lineal de una maqueta de un puente grúa. A través del uso de herramientas de simulación y control, como Matlab y Simulink, se busca desarrollar estrategias de control que permitan optimizar el funcionamiento del sistema, permitiéndole desarrollar trayectorias específicas con el menor error y oscilación posibles. El estudio y análisis de este caso práctico proporcionará una mejor comprensión de los principios de control aplicados a sistemas industriales y contribuirá al avance en el campo de la ingeniería electrónica y la automatización industrial.

Este capítulo aborda diversos temas relacionados con los objetivos y alcance del trabajo, así como el estudio de la física y dinámica de los puentes grúa. Se incluye la descripción de la maqueta proporcionada por INTECO, junto con un análisis del estado del arte en este campo.

## 1.1. Objetivos y alcance

El objetivo principal de este trabajo es realizar el modelado y el control de un puente grúa con movimiento 3D, un sistema altamente no lineal. A partir de lo anterior, se hablará del diseño de trayectorias específicas, así como del diseño de controladores, pensados para obtener la mínima oscilación de la carga.

El proyecto se llevará a cabo utilizando Matlab, un software de programación y cálculo, donde se implementarán las ecuaciones físicas que rigen el modelo. También se hará uso de la herramienta de simulación y modelado Simulink, mediante la cual se realizarán simulaciones antes de llevar a cabo cualquier prueba con la maqueta real.



El objetivo es conseguir un modelo cuyas trayectorias sean lo más similares posibles a la trayectoria que la carga realiza en la maqueta y, además, conseguir un control que permita realizar diferentes trayectorias con la mínima oscilación de la carga posible.

Este proyecto puede servir como base para futuras investigaciones sobre diferentes formas de control del movimiento de cargas en puentes grúa.

En cuanto a las tareas a realizar para conseguir los objetivos, son las que se citan a continuación:

- Estudio de la documentación aportada por el fabricante del puente grúa.
- Comprensión del modelo de referencia.
- Corrección de errores que presenta dicho modelo y creación de uno nuevo.
- Estimación de parámetros del nuevo modelo.
- Linealización del modelo.
- Estudio de la técnica de planitud diferencial y modificación del modelo para hacerlo más simple.
- Diseño de un control por prealimentación.
- Diseño de un control por realimentación.
- Diseño de un control por prealimentación y realimentación.
- Elaboración de la memoria.

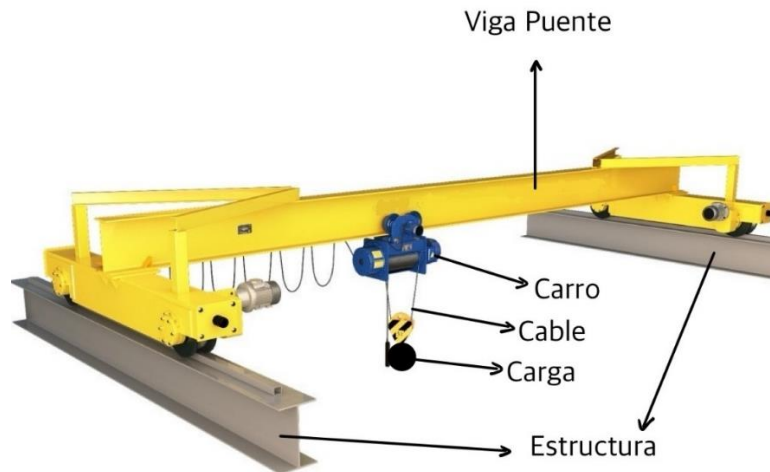
## 1.2. Descripción

### 1.2.1. Física y dinámica de un puente grúa

Como se ha comentado anteriormente, un puente grúa es una máquina empleada en entornos industriales que se utiliza para mover y/o levantar cargas pesadas. El uso de estos equipos mejora la seguridad en la industria, ya que los operarios no tienen que mover cargas pesadas entre los diferentes puestos de una instalación. Además, con un buen control del sistema, se pueden realizar trayectorias específicas, evitando obstáculos y oscilaciones.

Para comprender el funcionamiento de la estructura, que se basa en los principios de la mecánica clásica como la estática y la dinámica [1], es esencial conocer sus componentes principales [2]. Un puente grúa está compuesto, en primer lugar, por los rieles, que constituyen la estructura principal y permiten que el puente grúa se desplace a lo largo de ellos. El puente es una viga metálica de gran tamaño que se

desplaza a lo largo de los rieles. Sobre el puente se encuentra el carro, que se traslada a lo largo del mismo. La carga está unida al carro mediante un mecanismo de elevación y un cable, lo que permite su ascenso y descenso. Todos estos componentes se mueven por la acción de unos motores, que pueden ser eléctricos o hidráulicos. En la Figura 1 se representan las diferentes partes de un puente grúa.



*Figura 1. Partes que conforman el puente grúa*

En cuanto a los aspectos físicos, existen numerosas fuerzas que actúan sobre un puente grúa y que dependen de su configuración específica y los movimientos que realice. Algunas de las fuerzas que se abordarán en apartados y capítulos posteriores son las siguientes:

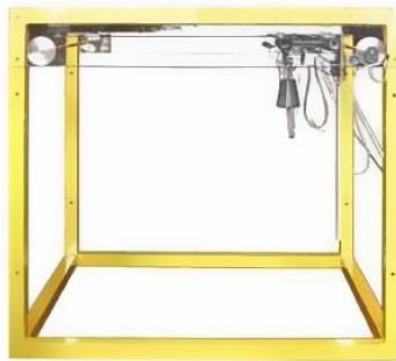
- Efecto de la gravedad: la carga, que cuelga del carro y está sujeta por un cable pendular, es afectada por la fuerza de la gravedad, la cual actúa verticalmente hacia abajo debido a la masa de la carga.
- Fuerzas de control generadas por los motores: Hay tres motores, cada uno de los cuales aplica la fuerza deseada sobre un eje de movimiento. Esta fuerza se transmite como una señal modulada por ancho de pulso (PWM, por sus siglas en inglés) que luego se amplifica y se convierte en corriente para ser aplicada al motor.
- Fuerzas de tensión: Al mover el puente grúa en cualquiera de los tres ejes, se generan unas fuerzas de tensión que se oponen al movimiento lineal de los objetos en reposo. Para la viga puente, estas fuerzas se producen por el peso del carro y la carga. En el caso del carro, las fuerzas de tensión se originan en las ruedas del carro al moverse a lo largo de la viga. Para la carga, se generan en el cable al subirla o bajarla.
- Fuerzas de fricción: Estas fuerzas, que se oponen al movimiento, surgen del contacto entre dos superficies y dependen principalmente de los materiales en contacto. La fricción estática debe ser superada para que un objeto en reposo

comience a moverse. Una vez en movimiento la fricción dinámica actúa oponiéndose al movimiento y provocando una pérdida de energía cinética.

Conocer los aspectos físicos es fundamental para comprender el funcionamiento del puente y poder realizar el modelado del sistema.

### 1.2.2. El puente grúa de INTECO

La maqueta utilizada a lo largo del trabajo para realizar el modelado y las simulaciones de los distintos controladores es la 3D Crane de Inteco [3], mostrada en la Figura 2.



*Figura 2. Puente grúa de INTECO*

La estructura del puente es cúbica, con dimensiones de 1x1x1 metros. Esta maqueta cuenta con tres motores de corriente continua de 12 voltios, controlados por PWM y 5 encoders para medir la posición. El control de los motores y la lectura de los encoders se realiza mediante la placa E/S RT-DAC, un dispositivo electrónico que permite la comunicación en tiempo real entre un ordenador y el entorno externo. Esta placa incluye un chip XILINX programable, que almacena la lógica necesaria para controlar los motores mediante pulsos PWM y leer los encoders.

Además, la maqueta incluye un modelo escrito en lenguaje C, compatible con Matlab, y una interfaz desarrollada en Simulink (Figura 3).

A partir de esta interfaz, es posible realizar diversas operaciones como testeo, prueba de *drivers*, prueba del modelo de referencia, ajuste de la posición inicial de los ángulos o pruebas sencillas sobre el movimiento del sistema. También incluye una serie de controladores con los que se puede experimentar y observar los diferentes movimientos de la carga.

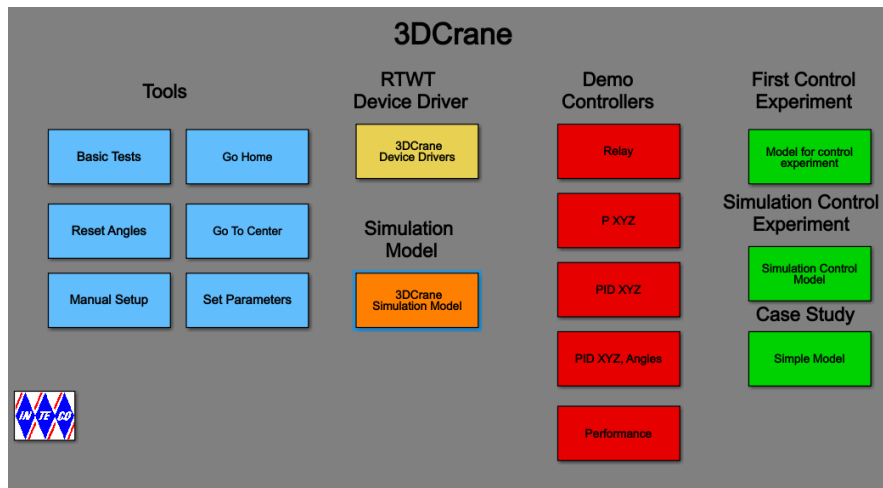


Figura 3. Interfaz de Simulink

## Crane3D Device Driver (USB)

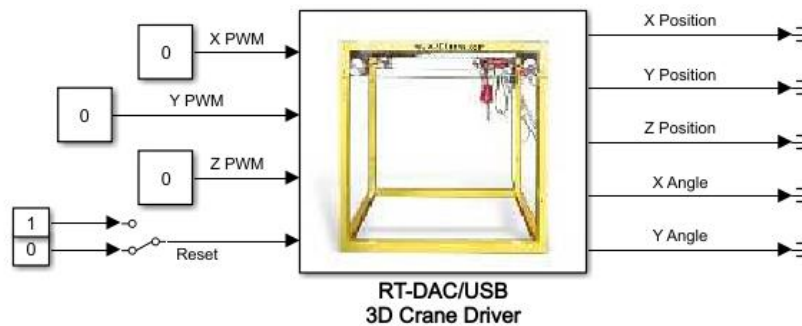


Figura 4. Driver principal

El *driver* principal (Figura 4) actúa como intermediario software entre el entorno real de la grúa y la tarjeta de adquisición RT-DAC/USB. Su función es enviar las señales de control y recibir las señales de los encoders.

El interruptor de *reset* mostrado en la Figura 4 es de seguridad y puede detener los motores automáticamente.

Los movimientos que se pueden realizar con esta maqueta se ilustran en la Figura 5, donde se presentan los 3 ejes de movimiento.

Según las partes descritas anteriormente, el puente se desplaza en el eje X, el carro en el eje Y, y la carga en el eje Z. Conociendo la posición del carro en los ejes X e Y, se puede determinar la posición de la carga mediante dos ángulos medidos con los encoders. Estos ángulos  $\alpha$  y  $\beta$ , representan los movimientos del puente y el carro en los ejes X e Y, respectivamente. En la Figura 6 se observan estos ángulos, donde para una posición de reposo de la carga, el ángulo  $\alpha$  es de 90 grados y el  $\beta$  de 0 grados.

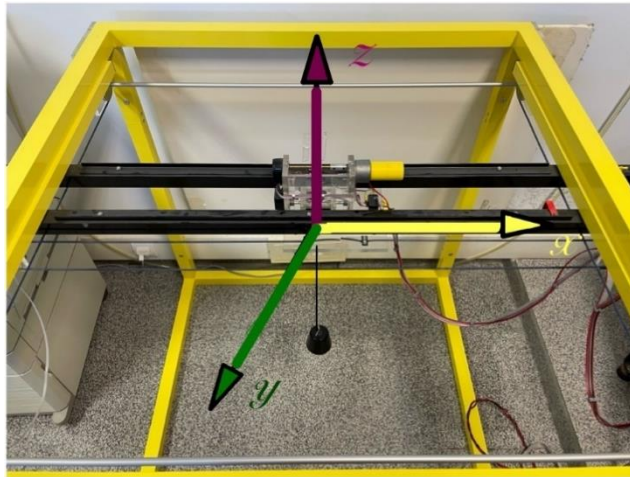


Figura 5. Ejes de movimiento

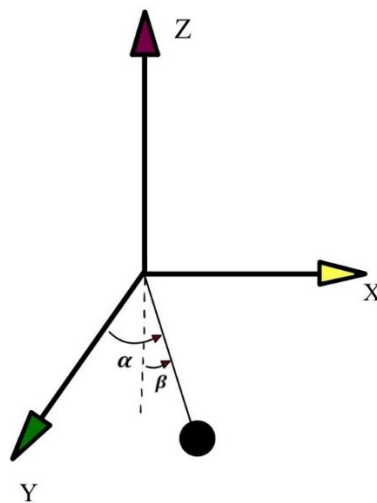


Figura 6. Ejes y ángulos

### 1.3. Estado del arte

Debido a los avances tecnológicos y a las nuevas necesidades surgidas en la industria a lo largo de los años, la maquinaria industrial ha experimentado una gran evolución [4].

En particular, el concepto de grúas se remonta a la Edad Media [5], cuando se utilizaban mecanismos con poleas para transportar objetos pesados. Las primeras grúas motorizadas aparecieron durante la revolución industrial, marcando el inicio de una rápida evolución en la búsqueda de mejores soluciones para la tarea diaria de transportar cargas pesadas en la industria.

En la actualidad, con los avances tecnológicos y la instalación permanente de estas estructuras en las industrias, el objetivo es lograr la máxima eficiencia y seguridad. Para ello, es fundamental crear modelos precisos que representen el sistema de manera fiable. Hoy en día, somos capaces de simular el funcionamiento de los puentes grúa en entornos como Simulink, lo que permite comprobar su correcto funcionamiento, corregir fallos y evitar desastres. Además, existe una continua investigación en métodos de control de estas estructuras, que ha abarcado desde técnicas de control clásico [6] hasta controladores adaptativos [7], robustos e incluso el uso de técnicas de inteligencia artificial [8] y realidad aumentada. Estas técnicas buscan alcanzar la máxima eficiencia y convertir los entornos industriales en lugares seguros para los operarios.

En este trabajo, nos centraremos en la creación de un modelo lo más representativo posible de la realidad para diseñar un control que minimice la oscilación de la carga al realizar una trayectoria específica. Para ello, se utilizará los entornos de trabajo Matlab y Simulink, que ofrecen numerosas ventajas [9]:

- El entorno gráfico es fácil de usar e intuitivo.
- Son adecuados para pruebas en tiempo real debido a su buena integración con otro hardware.
- Permiten realizar simulaciones precisas en tiempo real, facilitando la observación del funcionamiento del modelo.
- Posibilitan realizar análisis paramétricos para optimizar los diseños.
- Cuentan con una gran cantidad de documentación.

## 1.4. Estructura de la memoria

Esta memoria contiene 5 capítulos.

- El capítulo 1 introduce el tema del trabajo fin de máster, así como los objetivos y el estado del arte.
- El capítulo 2 explica el modelado del sistema, englobando el modelo de referencia y el modelo con correcciones.
- El capítulo 3 introduce el término de planitud diferencial y en él se diseña la trayectoria de referencia.
- El capítulo 4 describe el control implementado y compara los diferentes tipos.
- El capítulo 5 está dedicado a las conclusiones y a posibles mejoras que pueden ser realizadas en el futuro.

# 1.5. Cronograma

A continuación, en la Figura 7 se muestra un cronograma por semanas en el que se ilustra el tiempo que ha requerido cada fase del trabajo.

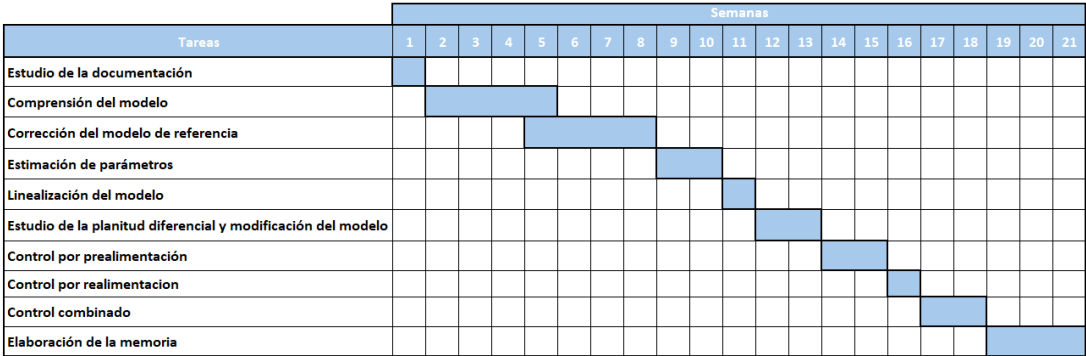


Figura 7. Cronograma

## 2. Modelado

En el capítulo actual, se abordará el tema del modelado del sistema. Comienza con el modelo de referencia proporcionado por el fabricante de la maqueta, sigue con el desarrollo de un modelo no lineal y la estimación de parámetros para mejorar la precisión respecto al modelo de referencia. Finalmente, se presenta el modelo lineal y se realiza un análisis del error.

### 2.1. Modelo de referencia

Como se ha mencionado anteriormente, el modelo de la maqueta del puente grúa, es un sistema MIMO altamente no lineal. La documentación de la maqueta incluye un código desarrollado en lenguaje C que contiene las ecuaciones de un modelo no lineal que describe el sistema (ver Anexo A). Para comprender estas ecuaciones, es necesario realizar un estudio detallado de las leyes físicas que rigen el funcionamiento del puente.

En la Figura 8, se puede observar un esquema del puente grúa y algunas de las fuerzas que actúan sobre él, además de los ejes de movimientos X, Y, Z.

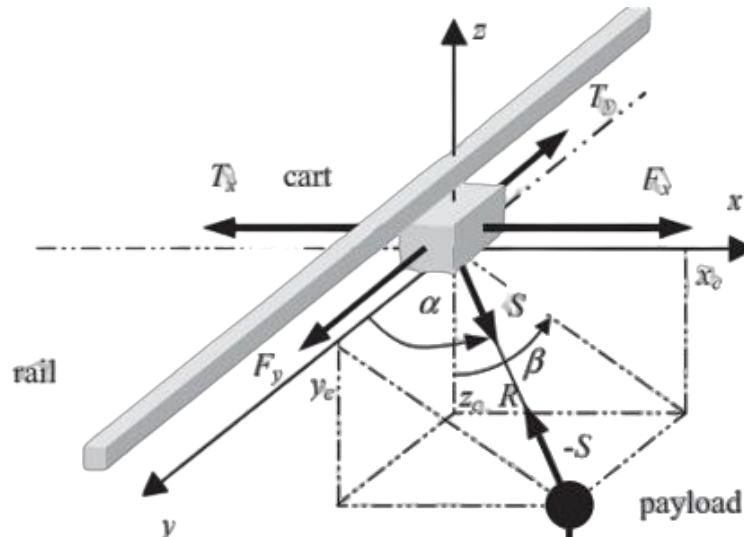


Figura 8. Esquema del Puente grúa

La viga puente (*rail*) y el carro (*cart*), se desplazan en el eje X, impulsados por la fuerza  $F_x$ , que es proporcionada por el motor en esa dirección. El carro se mueve a lo largo de la viga puente en el eje Y, impulsado por la fuerza  $F_y$ , también generada por el motor. Finalmente, la carga (*payload*) se mueve en el eje Z, subiendo y bajando mediante la fuerza  $F_z$  (no mostrada en la Figura 8, pero proporcionada por el motor). Las fuerzas  $T_x$ ,



$T_y$  y  $S(T_R)$ , representan las fuerzas que se oponen al movimiento en los ejes X, Y, Z, respectivamente. La carga está conectada al carro por medio de un cable que se balancea, por lo que con el movimiento del carro en los ejes x e y, la carga adquiere cierto ángulo con respecto a los ejes Y y Z,  $\alpha$  y  $\beta$  respectivamente.

Las cinco variables que miden los encoders son las siguientes:

- $x_w$ : Distancia del carro respecto del centro de la estructura en el eje X.
- $y_w$ : Distancia del carro respecto del centro de la estructura en el eje Y.
- $R$ : Longitud variable del cable del que cuelga la carga.
- $\alpha$ : Ángulo de la carga respecto al eje Y.
- $\beta$ : Ángulo de la carga respecto al eje Z.

En cuanto a las coordenadas de la carga, se denominan  $x_c$ ,  $y_c$  y  $z_c$ . Estas son las coordenadas de mayor interés para el trabajo, ya que el objetivo es controlar la carga. Las coordenadas de la carga no se miden directamente con los encoders, pero pueden calcularse fácilmente a partir de los datos que proporcionan y la relación entre ellos, como se muestra a continuación en las Ecuaciones (1) - (3).

$$x_c = x_w + R \sin\alpha \sin\beta \quad (1)$$

$$y_c = y_w + R \cos\alpha \quad (2)$$

$$z_c = -R \sin\alpha \cos\beta \quad (3)$$

Las ecuaciones que describen la dinámica del puente son las siguientes ((4) - (8)), donde  $m_c$ ,  $m_w$ ,  $m_s$ , representan las masas de la carga, el carro y la viga puente, respectivamente.

$$m_c \ddot{x}_c = -S_x \quad (4)$$

$$m_c \ddot{y}_c = -S_y \quad (5)$$

$$m_c \ddot{z}_c = -S_z - m_c g \quad (6)$$

$$(m_w + m_s) \ddot{x}_w = F_x - T_x + S_x \quad (7)$$

$$m_w \ddot{y}_w = F_y - T_y + S_y \quad (8)$$

$S_x, S_y, S_z$  son los componentes en X, Y y Z, respectivamente, del vector que indica la tensión del cable, y se calculan de la forma siguiente:

$$S_x = (F_R - T_R) \sin\alpha \sin\beta \quad (9)$$

$$S_y = (F_R - T_R) \cos\alpha \quad (10)$$

$$S_z = -(F_R - T_R) \sin\alpha \cos\beta \quad (11)$$

A partir de todas las ecuaciones anteriores, que provienen de la dinámica del puente teniendo en cuenta el esquema de la Figura 7 y de relaciones trigonométricas entre las diferentes componentes, sustituyendo las Ecuaciones (9) - (11) en las (4) - (8), obtenemos lo siguiente:

$$\ddot{x}_c = -\left(\frac{F_R}{m_c} - \frac{T_R}{m_c}\right) \sin\alpha \sin\beta \quad (12)$$

$$\ddot{y}_c = -\left(\frac{F_R}{m_c} - \frac{T_R}{m_c}\right) \cos\alpha \quad (13)$$

$$\ddot{z}_c = \left(\frac{F_R}{m_c} - \frac{T_R}{m_c}\right) \sin\alpha \cos\beta - g \quad (14)$$

$$\ddot{x}_w = \frac{F_x}{m_w + m_c} - \frac{T_x}{m_w + m_c} + \frac{m_c}{m_w + m_s} \left(\frac{F_R}{m_c} - \frac{T_R}{m_c}\right) \sin\alpha \sin\beta \quad (15)$$

$$\ddot{y}_w = \frac{F_y}{m_w} - \frac{T_y}{m_w} + \frac{m_c}{m_w} \left(\frac{F_R}{m_c} - \frac{T_R}{m_c}\right) \cos\alpha \quad (16)$$

Además, derivando dos veces las Ecuaciones (1) - (3) y teniendo en cuenta que la longitud de la cuerda R, es variable, obtenemos:

$$\begin{aligned} \ddot{x}_c = \ddot{x}_w + (\ddot{R} - R\dot{\alpha}^2 - R\dot{\beta}^2) \sin\alpha \sin\beta + 2R\dot{\alpha}\dot{\beta} \cos\alpha \cos\beta + \\ + (2\dot{R}\dot{\alpha} + R\ddot{\alpha}) \cos\alpha \sin\beta + (2\dot{R}\dot{\beta} + R\ddot{\beta}) \sin\alpha \cos\beta \end{aligned} \quad (17)$$

$$\ddot{y}_c = \ddot{y}_w + (\ddot{R} - R\dot{\alpha}^2) \cos\alpha - (2\dot{R}\dot{\alpha} + R\ddot{\alpha}) \sin\alpha \quad (18)$$

$$\begin{aligned}\ddot{z}_c = & (-\ddot{R} + R\dot{\alpha}^2 + R\dot{\beta}^2) \sin\alpha \cos\beta + 2R\dot{\alpha}\dot{\beta} \cos\alpha \sin\beta - \\ & -(2\dot{R}\dot{\alpha} + R\ddot{\alpha}) \cos\alpha \cos\beta + (2\dot{R}\dot{\beta} + R\ddot{\beta}) \sin\alpha \sin\beta\end{aligned}\quad (19)$$

A partir de todo lo anterior, obtenemos un sistema de orden 10 con las siguientes ecuaciones:

$$\ddot{x}_w = \frac{F_x}{m_w + m_c} - \frac{T_x}{m_w + m_c} + \frac{m_c}{m_w + m_s} \left( \frac{F_R}{m_c} - \frac{T_R}{m_c} \right) \sin\alpha \sin\beta \quad (20)$$

$$\ddot{y}_w = \frac{F_y}{m_w} - \frac{T_y}{m_w} + \frac{m_c}{m_w} \left( \frac{F_R}{m_c} - \frac{T_R}{m_c} \right) \cos\alpha \quad (21)$$

$$\begin{aligned}\ddot{R} = & - \left( \frac{F_y}{m_w} - \frac{T_y}{m_w} \right) \cos\alpha - \\ & - \left( \frac{F_x}{m_w + m_c} - \frac{T_x}{m_w + m_c} \right) \sin\alpha \sin\beta - \\ & - \left( \frac{F_R}{m_c} - \frac{T_R}{m_c} \right) \left( 1 + \frac{m_c}{m_w} \cos^2\alpha + \frac{m_c}{m_w + m_s} \sin^2\alpha \sin^2\beta \right) \\ & + R\dot{\beta}^2 \sin^2\alpha \\ & + g \sin\alpha \cos\beta + \dot{\alpha}^2 R\end{aligned}\quad (22)$$

$$\begin{aligned}\ddot{\alpha} = & \frac{1}{R} \left( \left( \frac{F_y}{m_w} - \frac{T_y}{m_w} \right) \sin\alpha - \left( \frac{F_x}{m_w + m_c} - \frac{T_x}{m_w + m_c} \right) \cos\alpha \sin\beta \right. \\ & + \left( \frac{F_R}{m_c} - \frac{T_R}{m_c} \right) \left( \frac{m_c}{m_w} - \frac{m_c}{m_w + m_s} \sin^2\beta \right) \cos\alpha \sin\alpha \\ & \left. + R\dot{\beta}^2 \cos\alpha \sin\alpha - 2\dot{R}\dot{\alpha} + g \cos\alpha \cos\beta \right)\end{aligned}\quad (23)$$

$$\begin{aligned}
\ddot{\beta} = & -\frac{1}{R \sin \alpha} \left( \left( \frac{F_x}{m_w + m_c} - \frac{T_x}{m_w + m_c} \right) \cos \beta \right. \\
& + \left( \frac{F_R}{m_c} - \frac{T_R}{m_c} \right) \frac{m_c}{m_w + m_s} \sin \alpha \cos \beta \sin \beta \\
& \left. + 2\dot{\beta} (R\dot{\alpha} \cos \alpha + \dot{R} \sin \alpha) + g \sin \beta \right)
\end{aligned} \tag{24}$$

Los términos  $T_x$ ,  $T_y$ ,  $T_z$  (Ecuaciones (25) – (27)), representan las fuerzas que se oponen al movimiento, y están formadas por un término constante  $T_{x\_cte}$ ,  $T_{y\_cte}$ ,  $T_{z\_cte}$ , con valores de 100, 82, 75, respectivamente y otro dependiente de la velocidad (concretamente del signo de la velocidad) en cada eje  $T_{sx}$ ,  $T_{sy}$ ,  $T_{sz}$ , cuyas constantes tienen valores de 5, 7.5 y 10, respectivamente,  $m_s$ ,  $m_w$ ,  $m_c$  hacen referencia a las masas de la viga puente, el carro y la carga.

$$T_x = T_{x\_cte} - T_{sx} \text{signo}(\dot{x}_w) \tag{25}$$

$$T_y = T_{y\_cte} - T_{sy} \text{signo}(\dot{y}_w) \tag{26}$$

$$T_z = T_{z\_cte} - T_{sz} \text{signo}(\dot{z}_w) \tag{27}$$

En las ecuaciones anteriores aparecen unos términos que incluyen unas constantes cuyo cálculo es desconocido y que multiplican al signo de las velocidades en cada uno de los ejes. Esto se puede interpretar como una fuerza de fricción, ya que se suma a la tensión en oposición a la fuerza de entrada producida por los motores.

La implementación de este modelo (cuyas ecuaciones están escritas en código c) en Simulink se puede ver en la Figura 9. Existen unas constantes por las que se multiplica la entrada (PWM) antes de aplicar la fuerza al modelo, que están relacionadas con las fricciones y tensiones en los raíles.

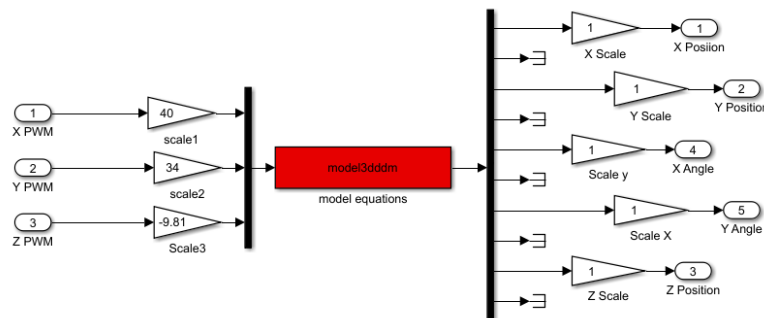


Figura 9. Modelo de referencia

Las salidas de este modelo son las coordenadas del carro y los dos ángulos. Sin embargo, también es crucial conocer las coordenadas de la carga. Por ello, se ha incluido un bloque de Simulink (Figura 10), que calcula dichas coordenadas a partir de las existentes, y que se utilizará a lo largo del trabajo, en todos los modelos.

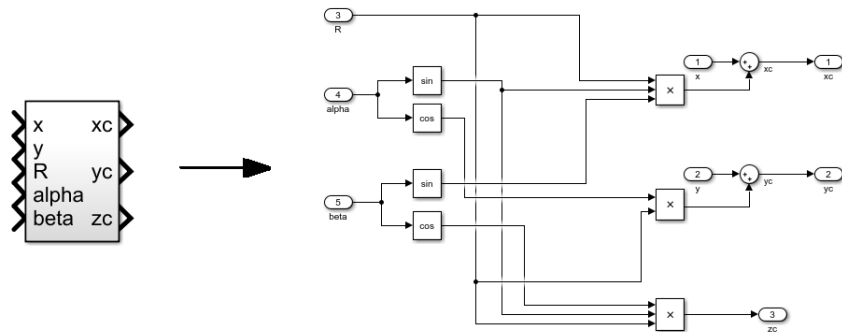


Figura 10. Bloque de conversión a coordenadas de la carga

La fuerza aplicada al sistema es la que se observa en la Figura 11.

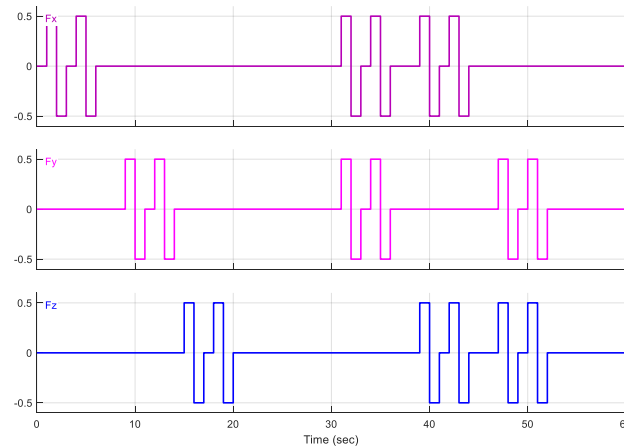


Figura 11. Fuerza aplicada al modelo

En la Figura 12 se representa el movimiento de la carga al aplicar la fuerza anterior sobre el sistema. En el eje X vemos el tiempo en segundos y en el Y la distancia con respecto al centro de la estructura, en metros. Se muestra la comparación entre el movimiento del modelo de referencia y el del sistema real. Se pueden apreciar diferencias significativas, lo que indica que el modelo no es completamente preciso. Al analizar estas discrepancias, se observa que existe cierta fricción, que además no es igual en todas las direcciones y que no está considerada en el modelo de referencia. Además, el cálculo del movimiento en el eje Z es incorrecto, ya que al aplicar fuerza en este eje, no se produce ningún movimiento.

Por todo lo anterior se decidió realizar un modelo basado en el de referencia, solucionando algunos de los errores comentados.

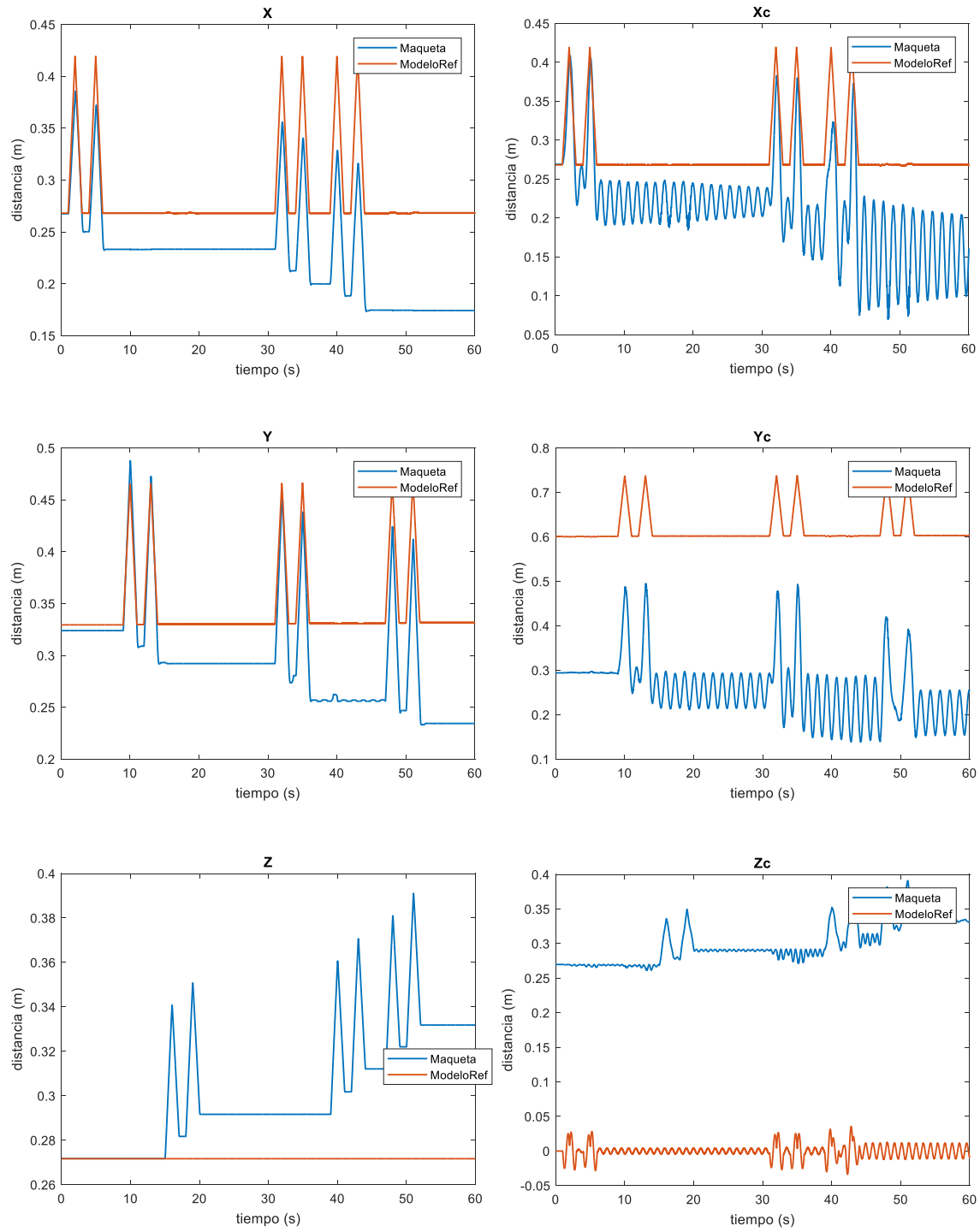


Figura 12. Resultados del modelo de referencia

## 2.2. Modelo no lineal

En este apartado se aborda la corrección de errores del modelo de referencia y la obtención del modelo final que represente con la mayor precisión posible el sistema real [10].

Inicialmente, se desarrolló el modelo en Simulink, utilizando el bloque 'Matlab Function' [11]. Dentro de esta función se escribieron las ecuaciones de las 10 variables de estado del sistema (ver Anexo B), las cuales fueron integradas con bloques de integración con condición inicial, como se muestra en la Figura 13. La letra 'p' que sigue al nombre de cada variable de estado indica su derivada; por ejemplo, xp representa la primera derivada de x y xpp la segunda.

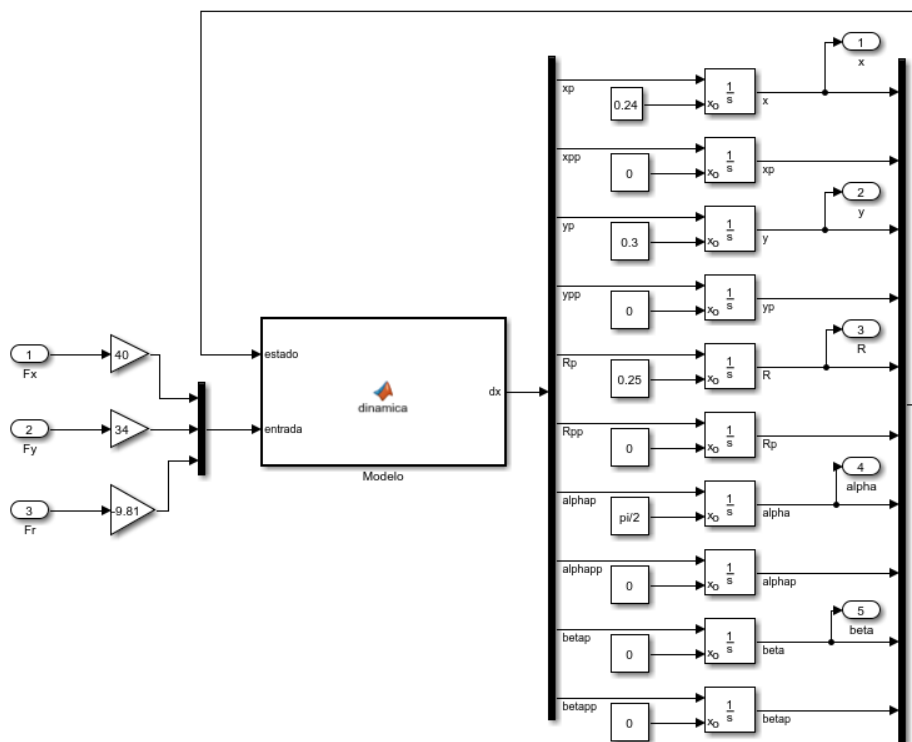


Figura 13. Modelo no lineal con parámetros iniciales

Uno de los principales errores se comete en el cálculo de la longitud de la cuerda R (movimiento en el eje Z). Como se observó en la Figura 12, la longitud de la cuerda no variaba al aplicar fuerza sobre dicho eje. Tras un análisis detallado del modelo de referencia, se concluyó que el efecto de la gravedad no se está teniendo en cuenta adecuadamente. Aunque la gravedad influye en el movimiento del cable, su efecto no coincide con la descripción en las ecuaciones. Lo que ocurre es que el término de la gravedad en las ecuaciones  $g \sin \alpha \cos \beta$  (de la Ecuación (22)) se está anulando con la fuerza de entrada multiplicada por la ganancia de ese eje (que es 9.8), lo que conlleva

ausencia de movimiento en Z a pesar de aplicar fuerza en ese motor. Al eliminar este término, como se muestra en la Figura 14, el movimiento del cable en el eje Z representa mejor lo que ocurre en la realidad.

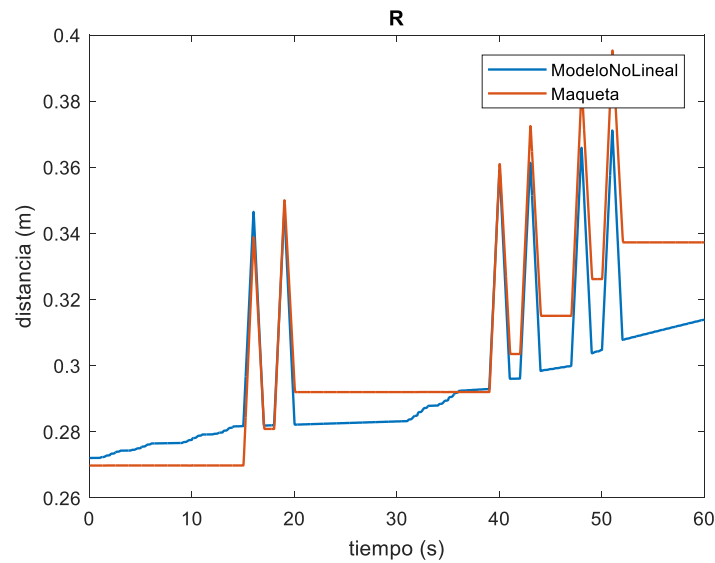


Figura 14. Movimiento en el eje Z

Observábamos en la Figura 12 cómo los movimientos en los ejes X e Y tampoco representaban fielmente la realidad. Se puede deducir que existe un término de fricción [12] que no se está considerando adecuadamente en las ecuaciones del modelo de referencia. Además, esta fricción no es uniforme en el movimiento en ambas direcciones. Como se ha mencionado previamente, existen valores numéricos asociados a esta fuerza que podrían ser incorrectos. Por esta razón, la fuerza de fricción se ha expresado de la forma en la que se observa en la Figura 15.

```

if x(4)>0
    tanhx = Tsx*tanh(x(4));
elseif x(4)<0
    tanhx = 0;
else
    tanhx = p*tanh(x(4));
end

if x(2)>0
    tanhy = Tsy*tanh(x(2));
elseif x(2) ==0
    tanhy = 0;
else
    tanhy = p1*tanh(x(2));
end

if x(10)>0
    tanhz = Tsz*tanh(x(10));
elseif x(10) ==0
    tanhz = 0;
else
    tanhz = p2*tanh(x(10));
end

```

Figura 15. Cálculo de la fuerza de fricción



Donde  $x(4)$  es la velocidad en el eje X,  $x(2)$  la velocidad en el eje Y, y  $x(10)$  la velocidad en el eje Z. Se ha utilizado la función tangente hiperbólica para determinar el signo de las velocidades, dado que es derivable, a diferencia de la función signo. Además, la constante por la que se multiplica el signo de la velocidad varía dependiendo del sentido del movimiento del puente. Este método de cálculo de las fuerzas de fricción resulta en 6 parámetros ( $T_{sx}$ ,  $T_{sy}$ ,  $T_{sz}$ ,  $p$ ,  $p_1$ ,  $p_2$ ) que se estimarán en el siguiente apartado.

Después de implementar las modificaciones mencionadas anteriormente, los resultados previos a la estimación de parámetros obtenidos son los que se observan en la Figura 16. Aunque existen todavía ciertos errores significativos, se observa que en general, el comportamiento del modelo se asemeja más a la realidad.

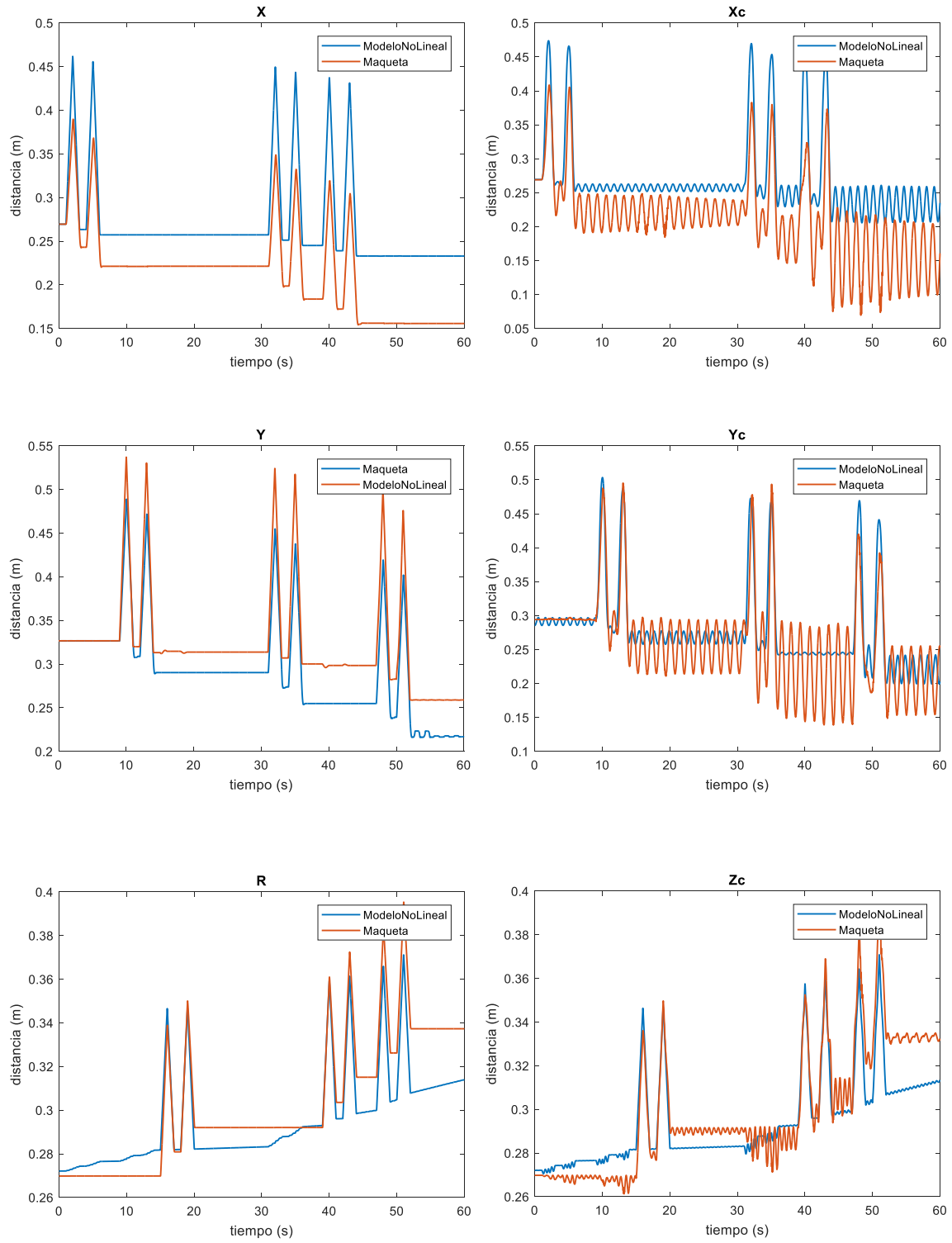


Figura 16. Resultados del modelo no lineal con parámetros iniciales

### 2.2.1. Estimación de parámetros

Como se ha visto en el apartado anterior, hay varios parámetros que se deben estimar, especialmente aquellos relacionados con las fuerzas de fricción, que en el modelo de referencia son constantes numéricas sin justificación explícita. Además, también existen constantes que dan valor a las fuerzas de tensión. Debido a que estos parámetros dependen de las propiedades de los materiales del puente grúa y sus características de deformación, no es posible saber con exactitud si los valores proporcionados en el modelo de referencia son adecuados.

En total, hay 9 parámetros que se van a estimar con la herramienta '*Parameter Estimation*', que proporciona Simulink [13]. Se trata de una herramienta que permite el estudio de modelos dinámicos en función de sus parámetros. Para utilizar la herramienta, primero se definen los parámetros a estimar y se seleccionan las señales cuyo comportamiento queremos observar en función de estos parámetros, así como la referencia que deseamos que sigan. Es crucial considerar todos los escenarios posibles al plantear el análisis. En este trabajo, esto implica aplicar fuerza en los tres motores, abarcando todas las posibles combinaciones (un motor, dos motores, todos los motores simultáneamente...).

Las señales de interés para el estudio son las salidas del modelo, que en la maqueta son leídas por los encoders. Se utilizarán estas como referencia ya que resulta fácil acceder a ellas, y se aplicará a la maqueta las mismas entradas que se aplican al modelo.

Para iniciar el proceso de estimación, se deben escoger valores iniciales para los parámetros. Cuanto mejor sea este valor inicial, menos iteraciones tardará el algoritmo en hallar la mejor solución. Se han seleccionado los valores que vemos en la Tabla 1, que son los que venían dados en el modelo de referencia.

$T_{sx}$	$T_{sy}$	$T_{sz}$	$p$	$p_1$	$p_2$	$T_x$	$T_y$	$T_z$
5	7.5	10	1	1	1	100	82	75

Tabla 1. Parámetros iniciales

La herramienta de análisis de parámetros calcula el error cuadrático entre las señales que están estimando y las de referencia y busca minimizar lo máximo posible este error utilizando un algoritmo u optimizador. El método que utiliza el optimizador es el de mínimos cuadrados para sistemas no lineales [14]. Este método itera para encontrar los valores de parámetros que minimizan la suma de los cuadrados de las diferencias observados y los predichos por el modelo. El método iterativo utilizado para realizar un barrido por diferentes valores de parámetros es el algoritmo '*Trust Region Reflective*'

[15], que busca los valores óptimos dentro de una región limitada alrededor del punto actual, proporcionando robustez y rapidez en la optimización.

Una vez llevada a cabo la estimación, se obtienen los parámetros óptimos que el algoritmo ha encontrado, los cuales se presentan en la Tabla 2.

$T_{sx}$	$T_{sy}$	$T_{sz}$	$p$	$p_1$	$p_2$	$T_x$	$T_y$	$T_z$
25.05	9.974	97.65	-0.3	1.801	1	148.9	80.68	46.97

*Tabla 2. Parámetros estimados*

Tras realizar las simulaciones y compararlas con el movimiento real de la maqueta, observamos en la Figura 17 que el error ha disminuido significativamente en comparación con el ejemplo anterior.

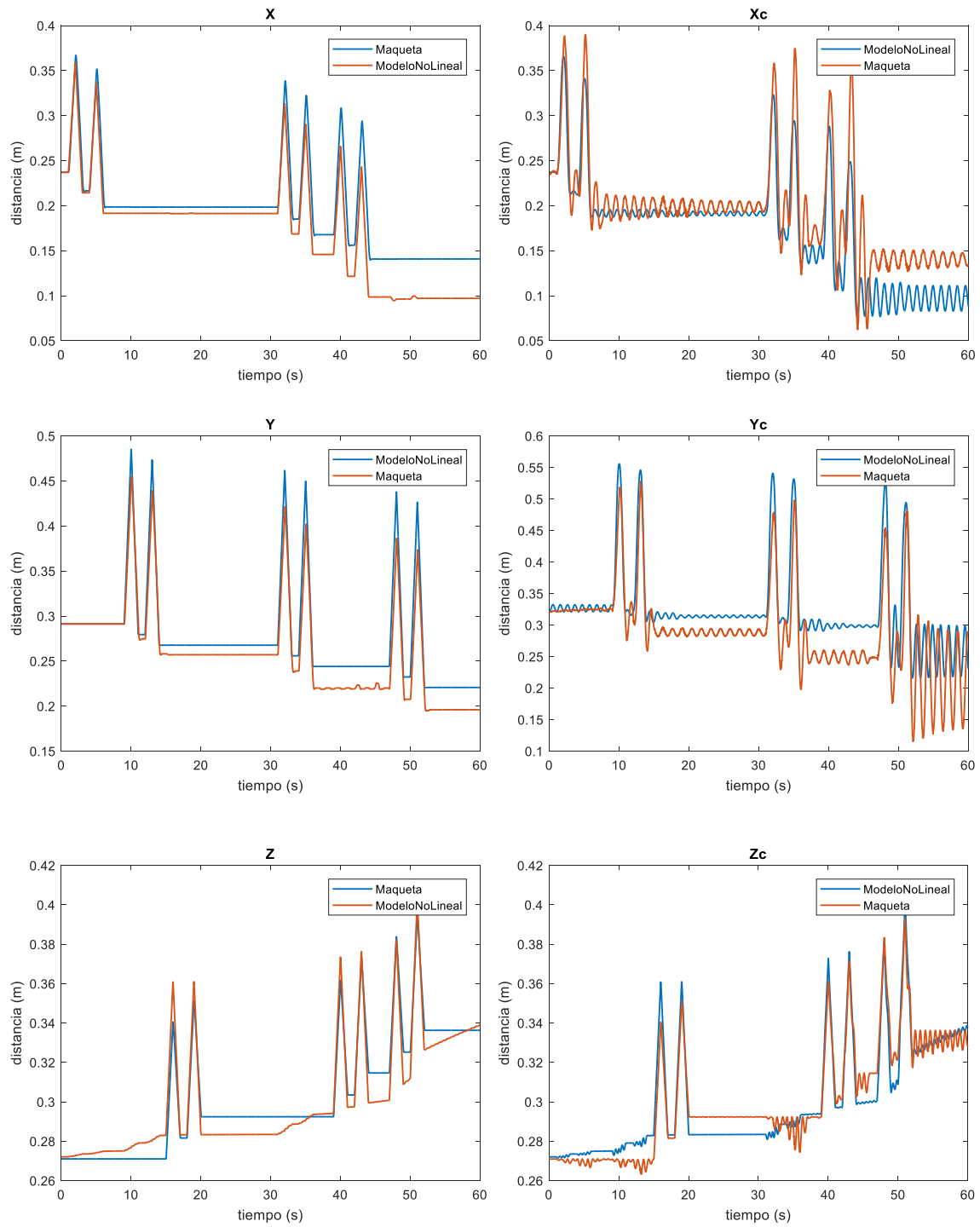


Figura 17. Resultados del modelo lineal con estimación de parámetros

## 2.3. Linealización

### 2.3.1. Descripción

Un modelo lineal [16] se caracteriza por relaciones lineales entre sus variables, donde cada término de las ecuaciones es una constante o el producto de una constante con una variable.

El proceso de linealización [17] de un modelo no lineal consiste en aproximar dicho modelo por su equivalente lineal en torno a un punto de operación. Este punto de operación describe condiciones bajo las cuales el sistema opera en una región donde puede considerarse aproximadamente lineal. Estas condiciones suelen ser las que llevan al sistema a encontrarse en un estado de equilibrio.

Para sistemas dinámicos con múltiples entradas y salidas, como el abordado en este trabajo, el modelo lineal se representa en el espacio de estados. Este enfoque matemático permite describir el sistema en términos de sus variables de estado, las cuales ofrecen información sobre el estado global del sistema en cada instante.

Una vez identificado el punto de operación, se calculan las derivadas parciales de las funciones con respecto a cada variable de estado para construir las matrices del sistema [18]. Las ecuaciones que definen un modelo en espacio de estados se muestran en las Ecuaciones (28) – (29), donde:

- 'y' representa el vector de salidas.
- 'x' es el vector de variables de estado.
- 'u' es el vector de entradas.
- 'A' es la matriz de estados, calculada a partir de las derivadas parciales de las ecuaciones en función de las variables de estado.
- 'B' es la matriz de entrada, calculada a partir de las derivadas parciales de las ecuaciones en función de las entradas.
- 'C' es la matriz de salida.
- 'D' es la matriz de transmisión directa.

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (28)$$

$$y(t) = Cx(t) + Du(t) \quad (29)$$

Es importante destacar que la linealización de un modelo puede resultar en una pérdida de precisión en su respuesta, dado que implica una aproximación válida únicamente si el sistema opera en las proximidades del punto de operación seleccionado. No obstante, la linealización también presenta una serie de ventajas:

- Ecuaciones más sencillas: simplificar las ecuaciones permite realizar un análisis más sencillo, lo que conlleva una menor capacidad computacional requerida y

por lo tanto un menor tiempo de simulación. También se ahorra en recursos de memoria y almacenamiento.

- Controladores: al linealizar el modelo es posible diseñar controladores lineales, los cuales resultan más fáciles de diseñar e implementar.
- Herramientas software: existen herramientas que permiten implementar de forma sencilla modelos lineales, como por ejemplo Matlab y Simulink.
- Análisis: al tener un modelo lineal, existen métodos matemáticos para analizar su estabilidad, observabilidad y controlabilidad.

### 2.3.2. El modelo

Tras comprender el concepto de linealización y evaluar sus posibles ventajas y desventajas, en este apartado se va a realizar la linealización del modelo (ver Anexo C).

En la Tabla 3 se representan las 10 variables de estado que definen el modelo, indicando sus unidades y los valores correspondientes cuando el sistema se encuentra en el punto de equilibrio.

$x_w$ (m)	$\dot{x}_w$ (m/s)	$y_w$ (m)	$\dot{y}_w$ (m/s)	$R$ (m)	$\dot{R}$ (m/s)	$\alpha$ (°)	$\dot{\alpha}$ (°/s)	$\beta$ (°)	$\dot{\beta}$ (°/s)
0.24	0	0.3	0	0.25	0	90°	0	0°	0

Tabla 3. Variables de estado y punto de equilibrio

Las entradas al sistema, como se ha mencionado a lo largo de la memoria, son las tres fuerzas PWM que controlan los motores:  $F_x$ ,  $F_y$ ,  $F_z$ .

Respecto a las salidas del modelo, como se ha discutido anteriormente, son las coordenadas de la carga. Pero a la hora de linealizar el modelo, se ha decidido que sean las 3 coordenadas del carro ( $x_w$ ,  $y_w$ ,  $R$ ) y los ángulos ( $\alpha$  y  $\beta$ ), ya que son variables de estado. Posteriormente se calcularán las coordenadas de la carga con el bloque descrito en capítulos anteriores.

La linealización se ha llevado a cabo utilizando la función '*jacobian*' [19] de Matlab, que calcula la matriz jacobiana a partir de las funciones del modelo.

$$\begin{pmatrix} \dot{x}_w \\ \ddot{x}_w \\ \dot{y}_w \\ \ddot{y}_w \\ \dot{R} \\ \ddot{R} \\ \dot{\alpha} \\ \ddot{\alpha} \\ \dot{\beta} \\ \ddot{\beta} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{T_x}{m_w + m_c} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{T_y}{m_w} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{T_z}{m_c} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{T_y}{0.25m_w} & 0 & 0 & -\frac{g}{0.25} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & \frac{T_x}{(m_w + m_c) 0.25} & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{g}{0.25} & 0 \end{pmatrix} \begin{pmatrix} x_w \\ \dot{x}_w \\ y_w \\ \dot{y}_w \\ R \\ \dot{R} \\ \alpha \\ \dot{\alpha} \\ \beta \\ \dot{\beta} \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & \frac{1}{0.25} & 0 \\ 0 & 0 & 0 \\ \frac{1}{0.25} & 0 & 0 \end{pmatrix} \begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix}$$

$$\begin{pmatrix} x_w \\ y_w \\ R \\ \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{x}_w \\ \dot{y}_w \\ \dot{R} \\ \dot{\alpha} \\ \dot{\beta} \end{pmatrix}$$

Figura 18. Modelo lineal

Una vez calculadas todas las matrices, el modelo lineal resultante se muestra en la Figura 18.

Al simular el modelo lineal y compararlo con el movimiento real realizado por la maqueta, observamos los resultados representados en la Figura 19.



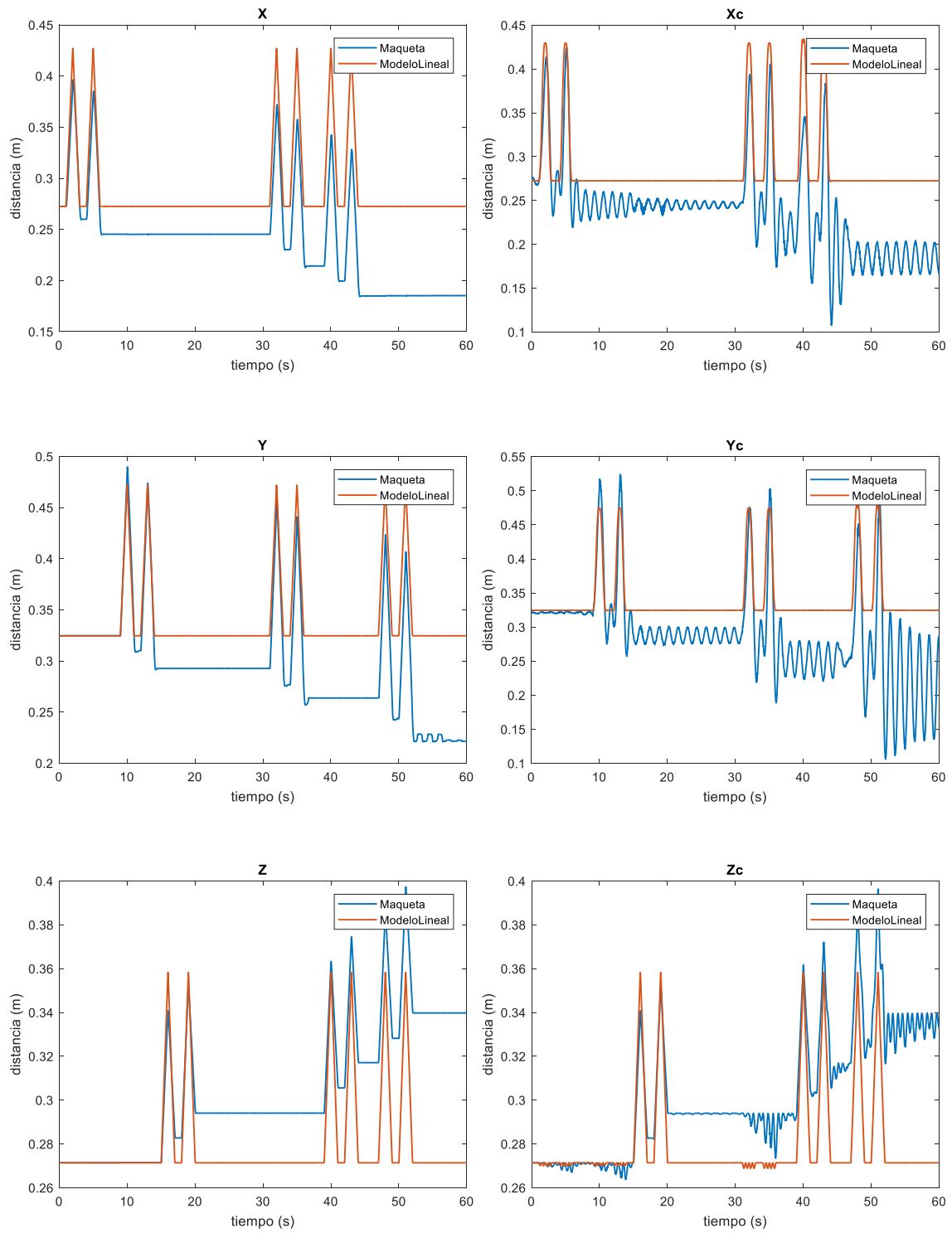


Figura 19. Resultados del modelo lineal

Se ve claramente en las gráficas una pérdida de precisión al linealizar el modelo, ya que el sistema original es altamente no lineal, como se discutió previamente.

A simple vista, se aprecia que la respuesta del modelo linealizado se asemeja a la del modelo de referencia. Esto se debe a que, como se comentó anteriormente, en el modelo de referencia las fuerzas de fricción no se están considerando adecuadamente. Al linealizar el modelo, las fuerzas de fricción (Figura 15) no se incluyen en las ecuaciones porque dependen de las velocidades, que en el punto de equilibrio son nulas. Sin embargo, esta aproximación no es completamente coherente con el movimiento real, que se aleja lo suficiente del punto de equilibrio como para que las fuerzas de fricción influyan.

Además, se observa un error significativo en las coordenadas de la carga. Esto se debe a que las ecuaciones, especialmente las relacionadas con los ángulos, son muy complejas y presentan relaciones altamente no lineales, por lo que al linealizar se pierde información esencial, como las oscilaciones producidas por la carga al aplicar fuerza. En el sistema real, si aplicamos fuerza en un eje, por ejemplo el eje Y, la carga oscilará con ángulo  $\alpha$ . Al dejar de aplicar esta fuerza, el ángulo  $\alpha$  no se reduce a cero inmediatamente, sino que la carga sigue oscilando en esa dirección. Este efecto se pierde al linealizar el sistema. Los ángulos  $\alpha$  (que en el modelo no lineal depende del movimiento en x y en z, además de en y) y  $\beta$  (que en el modelo no lineal depende del movimiento en z, además de en x), se reducen a cero en cuanto no se aplica fuerza en los ejes Y o X, respectivamente, como se muestra en la Figura 20. Por esta razón, no se aprecian las oscilaciones en la carga en el modelo linealizado.

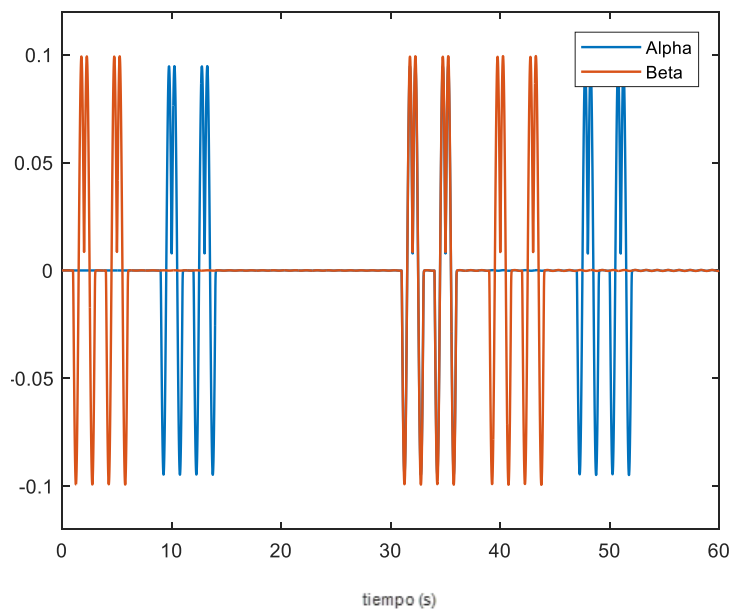


Figura 20. Ángulos Alpha y Beta

## 2.4. Cálculo del error

A pesar de que en las gráficas se observa la mejora del modelo con los cambios, es útil obtener alguna medida cuantitativa que indique la magnitud de esta mejora.

Se ha decidido calcular la raíz del error cuadrático medio (RMSE) [20], que se define en la Ecuación (30), donde  $n$  representa el número de datos tomados de las trayectorias.

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(TrayectoriaReal - TrayectoriaModelo)^2}{n}} \quad (30)$$

Una vez calculado, para evaluar la precisión de los modelos respecto al comportamiento real, se calcula el error relativo con respecto a la amplitud total del movimiento observado.

	Modelo de Referencia	Modelo no lineal sin estimación de parámetros	Modelo no lineal con estimación de parámetros	Modelo lineal
X	32.56%	25.34%	12.71%	26,83%
X <sub>c</sub>	34.28%	12.83%	7.91%	19,84%
Y	34.10%	10.92%	6.32%	27,01%
Y <sub>c</sub>	25.02%	20.80%	10.63%	19,18%
Z	100.48%	11.62%	11.88%	16,62%
Z <sub>c</sub>	231.66%	9.48%	6.16%	23,53%

Tabla 4. Comparación de los errores de los modelos

Los resultados se presentan en la Tabla 4, donde los valores más bajos indican una mejor precisión del modelo en la predicción del comportamiento del sistema. Se observa que el menor error (columna marcada en verde) corresponde al modelo no lineal en el que se ha realizado la estimación de parámetros. En contraste, los errores más altos se encuentran en el modelo de referencia proporcionado por el fabricante (columna marcada en rojo).

### 3. Planitud diferencial

En este apartado se introduce el concepto de planitud diferencial y se procede a simplificar el modelo. Se concluye con el cálculo detallado de una trayectoria específica que será empleada como referencia en el posterior control del sistema.

#### 3.1. Definición

La planitud diferencial [21] es una propiedad de algunos sistemas no lineales que permite representar las entradas del sistema de manera plana, es decir, que dependan únicamente de las salidas de este y sus derivadas.

En el contexto del control, la planitud diferencial es ampliamente utilizada para el diseño de controladores robustos en sistemas no lineales. Con esta técnica es posible el diseño de trayectorias deseadas. Una vez se conoce la trayectoria que se desea seguir, es posible calcular la fuerza necesaria que debe aplicarse al modelo para seguir dicha trayectoria. Esto permite implementar un control en lazo abierto sobre el sistema, aplicando directamente la fuerza requerida para ejecutar la trayectoria diseñada.

#### 3.2. El modelo

El modelo del puente grúa es plano, lo que implica que el concepto de planitud diferencial puede ser aplicado directamente. Debido a la complejidad matemática de las ecuaciones, en este trabajo fin de máster se ha aplicado la técnica de planitud diferencial al control de trayectorias que no incluyen movimientos en el eje X.

Esto simplifica el desarrollo, de forma que la única entrada al modelo será la fuerza del motor en el eje Y,  $F_y$ . Aunque el movimiento está restringido al eje Y, la carga se balanceará con un ángulo  $\alpha$ , lo que significa que la coordenada en el eje Z de la carga,  $z_c$ , también variará y será relevante en las ecuaciones. El ángulo  $\beta$  se aproxima a  $0^\circ$ , dado que el movimiento en el eje X es nulo. Con estas consideraciones, las salidas del sistema serán  $y_c$  y  $\alpha$ .

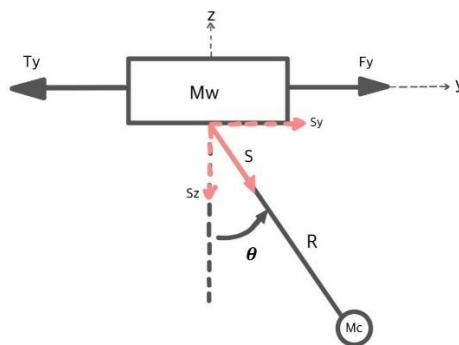


Figura 21. Modelo con movimiento en una única dirección

El esquema de las fuerzas y movimientos puede ahora representarse en dos dimensiones, como se observa en la Figura 21.

$$F_y = m_w \ddot{y}_w + \tan h y + T_y \dot{y}_w - S_y \quad (31)$$

$$S_y = -m_c \ddot{y}_c = (F_R - T_R) \sin \theta \quad (32)$$

$$S_z = m_c g + m_c \ddot{z}_c = (F_R - T_R) \cos \theta \quad (33)$$

$$\frac{S_y}{S_z} = \frac{(F_R - T_R) \sin \theta}{(F_R - T_R) \cos \theta} = \tan \theta = \frac{-m_c \ddot{y}_c}{m_c g + m_c \ddot{z}_c} = \frac{-\ddot{y}_c}{g + \ddot{z}_c} \quad (34)$$

$$S = \sqrt{S_y^2 + S_z^2} = m_c \sqrt{(\ddot{y}_c)^2 + (g + \ddot{z}_c)^2} \quad (35)$$

$$\alpha = 90^\circ - \theta = \frac{\pi}{2} - \arcsin \left( \frac{-m_c \ddot{y}_c}{S} \right) \quad (36)$$

$$\ddot{y}_w = \frac{1}{m_w} (F_y - \tan h y - T_y \dot{y}_w + S \cos \alpha) \quad (37)$$

$$y_w = y_c - z_c \frac{\ddot{y}_c}{g + \ddot{z}_c} = -R \sin \alpha \frac{\ddot{y}_c}{g + \ddot{z}_c} \quad (38)$$

Las Ecuaciones (31) – (38) describen la nueva dinámica del modelo. Derivando la Ecuación (38) y sustituyendo en la (31), se puede expresar la única entrada existente ( $F_y$ ) en función de las entradas ( $y_c$  y  $\alpha$ ) y sus derivadas.

### 3.3. Diseño de trayectorias

Con la técnica de planitud diferencial, es posible diseñar una trayectoria y calcular la fuerza necesaria a aplicar sobre el carro para realizarla sin necesidad de resolver ninguna ecuación diferencial. Aunque la trayectoria deseada se calcula para la carga, la fuerza calculada a partir de las Ecuaciones (31) – (38) se aplica al carro.

Al diseñar la trayectoria es crucial considerar diversos factores que influirán en el tipo de movimientos que queremos que la carga realice [22]. Normalmente, los puentes

grúa reales mueven cargas muy pesadas y operan en espacios de trabajo amplios. Para garantizar la máxima seguridad tanto de la carga como de la instalación y los operarios, el movimiento debe ser suave. Por ello, se ha optado por utilizar el concepto ‘arranque y aterrizaje suave’ (‘soft start/landing’). Esta técnica implica movimientos suaves al inicio y final de la trayectoria: al comienzo, la fuerza aplicada y la velocidad son mínimas, aumentando gradualmente a medida que la carga avanza, y luego disminuyen nuevamente al acercarse al destino final. Este enfoque no solo asegura que la carga se mueva con baja velocidad desde y hacia los puntos extremos, sino que también ayuda a prevenir oscilaciones bruscas, siendo especialmente adecuado para cargas frágiles o que contengan sustancias peligrosas que pudieran derramarse.

La trayectoria deseada se ha definido por tramos utilizando un polinomio de grado 5, ya que existen 6 condiciones para la trayectoria, como se muestra en la Ecuación (39), En esta ecuación,  $y_i$  representa la posición inicial de la carga en el eje Y,  $y_f$  la posición final,  $t_i$  el tiempo de inicio del movimiento,  $t_m$  el tiempo en el que la carga deja de moverse y  $t_f$  el tiempo final de la trayectoria.

$$y(t) = \begin{cases} y_i & \text{si } 0 \leq t \leq t_i \\ a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0 & \text{si } t_i < t < t_m \\ y_f & \text{si } t_m \leq t \leq t_f \end{cases} \quad (39)$$

Para garantizar que la trayectoria sea derivable y se pueda calcular la fuerza necesaria, la función definida por tramos debe ser derivable dos veces en todos los puntos. Además, en los instantes  $t_i$  y  $t_m$ , la velocidad y la aceleración deben ser nulas para aplicar la técnica de arranque y aterrizaje suave. Las condiciones necesarias para calcular los coeficientes se presentan en las Ecuaciones (40) – (45).

$$y(t_i) = y_i \quad (40)$$

$$y(t_m) = y_f \quad (41)$$

$$\dot{y}(t_i) = 0 \quad (42)$$

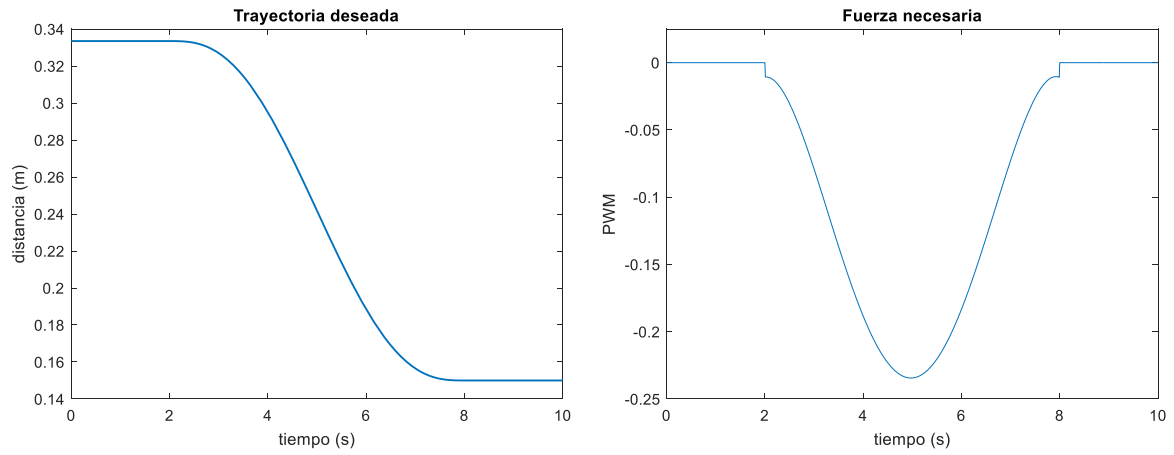
$$\dot{y}(t_m) = 0 \quad (43)$$

$$\ddot{y}(t_i) = 0 \quad (44)$$

$$\ddot{y}(t_m) = 0 \quad (45)$$

Estas ecuaciones se han implementado en una función de Matlab para calcular trayectorias con diferentes tiempos y posiciones.

La trayectoria obtenida, así como la fuerza necesaria para llevarla a cabo, se muestran en la Figura 22.



*Figura 22. Trayectoria deseada y fuerza necesaria*

## 4. Control

En el siguiente apartado se presentan tres esquemas de control distintos y se analiza su eficacia. Se realiza además una comparación entre los tres para determinar cuál es el más adecuado.

### 4.1. Control con prealimentación

Para el diseño de control con prealimentación (*feedforward*) [23] se utilizará la técnica de planitud diferencial explicada anteriormente. Con esta técnica, se calculará la fuerza necesaria para seguir una trayectoria deseada, aplicando dicha fuerza directamente al sistema. A diferencia del control realimentado, que corrige el error después de que ocurra, el control con prealimentación lo previene. Esto requiere conocer de antemano el comportamiento deseado de la variable controlada. La estructura utilizada para este tipo de control se muestra en la Figura 23.

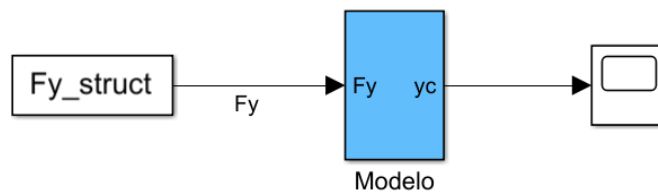


Figura 23. Control con prealimentación

Tras aplicar la fuerza calculada al modelo y a la maqueta, se obtienen los resultados que se presentan en la Figura 24.

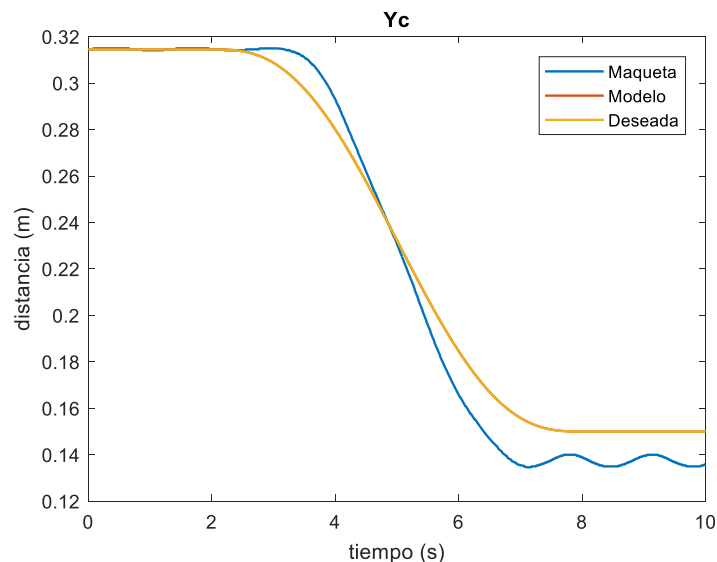


Figura 24. Resultados del control con prealimentación



La respuesta del modelo es exactamente igual a la trayectoria deseada. Esto se debe a que se han utilizado las mismas ecuaciones del modelo para calcular la fuerza necesaria. Sin embargo, observamos que el modelo no representa con total exactitud la dinámica de la maqueta, ya que esta presenta cierta oscilación y error. Aun así, al tratarse de un control únicamente prealimentado, el error cometido no es demasiado grande y puede ser corregido mediante otras técnicas de control, como veremos a continuación.

## 4.2. Control con realimentación

El control con realimentación (*feedback*) implica utilizar la información de la salida del sistema para ajustar la entrada mediante un controlador, intentando que la salida se aproxime lo máximo posible a la referencia deseada. Para lograrlo, se calcula inicialmente el error como la diferencia entre la salida deseada y la salida obtenida en cada instante, y luego se aplica una acción de control que determinará la entrada necesaria para el sistema.

En el presente trabajo, la acción de control se ha obtenido mediante un controlador proporcional integral (PI). La acción proporcional al error corrige las oscilaciones y mejora el tiempo de respuesta del sistema, pero no elimina el error en estado estacionario. La acción integral integra el error y logra eliminarlo en estacionario. Una desventaja de este tipo de control es su respuesta ante errores muy grandes, lo que puede saturar la acción integral. Esto podría ser perjudicial si el sistema experimenta paradas y arranques bruscos. Sin embargo, como se ha demostrado anteriormente, al utilizar la técnica de arranque y aterrizaje suaves, este efecto se minimiza.

Así, la ecuación del controlador se muestra en la Ecuación (46), en tiempo continuo y en la Ecuación (47) su transformada de Laplace, donde  $K_p$  es la constante proporcional y  $K_i$  la constante integral.

$$acción(t) = K_p \text{ error} + K_i \int_0^t \text{error} dt \quad (46)$$

$$acción(s) = \text{error} \left( K_p + \frac{K_i}{s} \right) \quad (47)$$

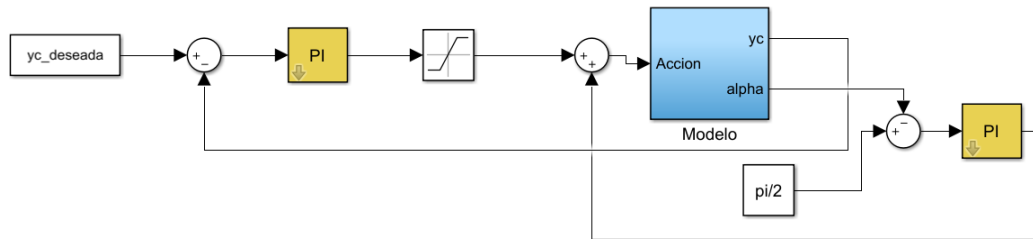


Figura 25. Control con realimentación

El esquema de control realimentado se muestra en la Figura 25, donde se observa que la acción de control es la suma de la acción de regular el movimiento de la carga en el eje y la de controlar ángulo  $\alpha$ . Esto tiene como objetivo ejecutar la trayectoria con la mínima oscilación posible. Una vez establecido el esquema, se procederá a determinar experimentalmente los parámetros óptimos del controlador.

Los valores iniciales elegidos son los que se muestran en la Tabla 5, mientras que los resultados obtenidos se presentan en la Figura 26.

$K_p$ de $y_c$	$K_i$ de $y_c$	$K_p$ de $\alpha$	$K_i$ de $\alpha$
5	5	2	2

Tabla 5. Constantes iniciales del control con realimentación

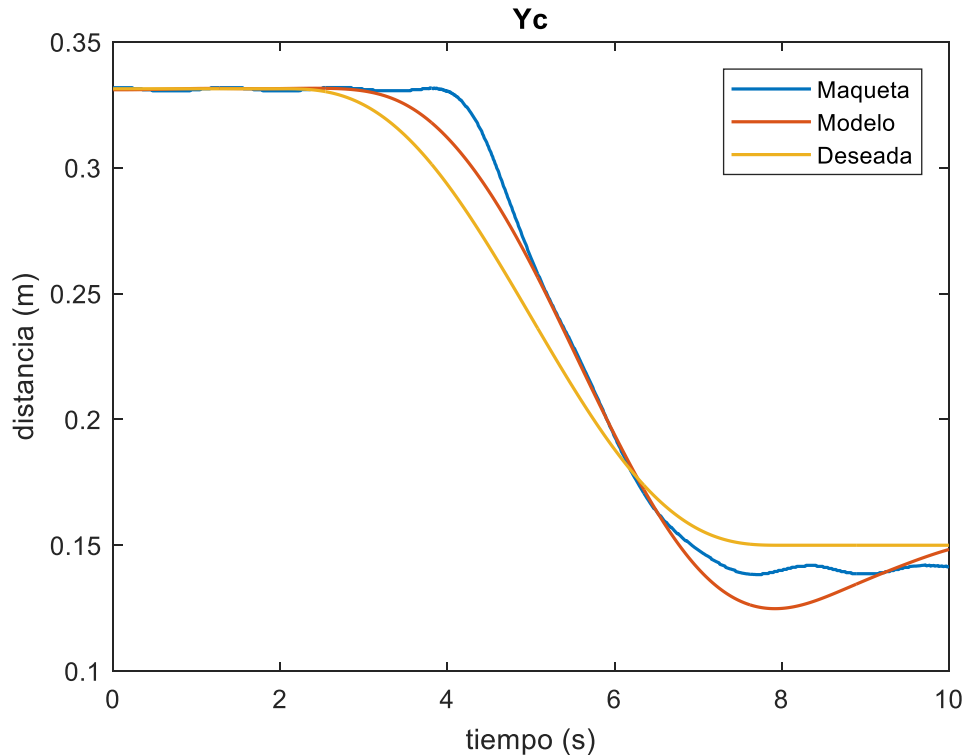


Figura 26. Resultados del control con realimentación para las primeras constantes

Al analizar el gráfico anterior, vemos que el sistema exhibe una respuesta lenta y un error significativo. Para abordar estos problemas, pueden aumentarse las ganancias del controlador.

Aumentar la ganancia proporcional mejorará el tiempo de respuesta del sistema, aunque podría provocar oscilaciones, por lo que también será necesario aumentar la ganancia proporcional del control del ángulo. Además, será necesario aumentar la ganancia integral para eliminar el error en estado estacionario presente en el movimiento de la maqueta. También se observan oscilaciones al finalizar el movimiento, lo cual justifica la necesidad de ajustar las ganancias del control del ángulo.

Después de realizar varias pruebas, se determinan los siguientes parámetros que ofrecen la mejor respuesta, detallados en la Tabla 6. Los resultados alcanzados se muestran en la Figura 27.

$K_p$ de $y_c$	$K_i$ de $y_c$	$K_p$ de $\alpha$	$K_i$ de $\alpha$
25	20	10	10

Tabla 6. Constantes finales del control con realimentación

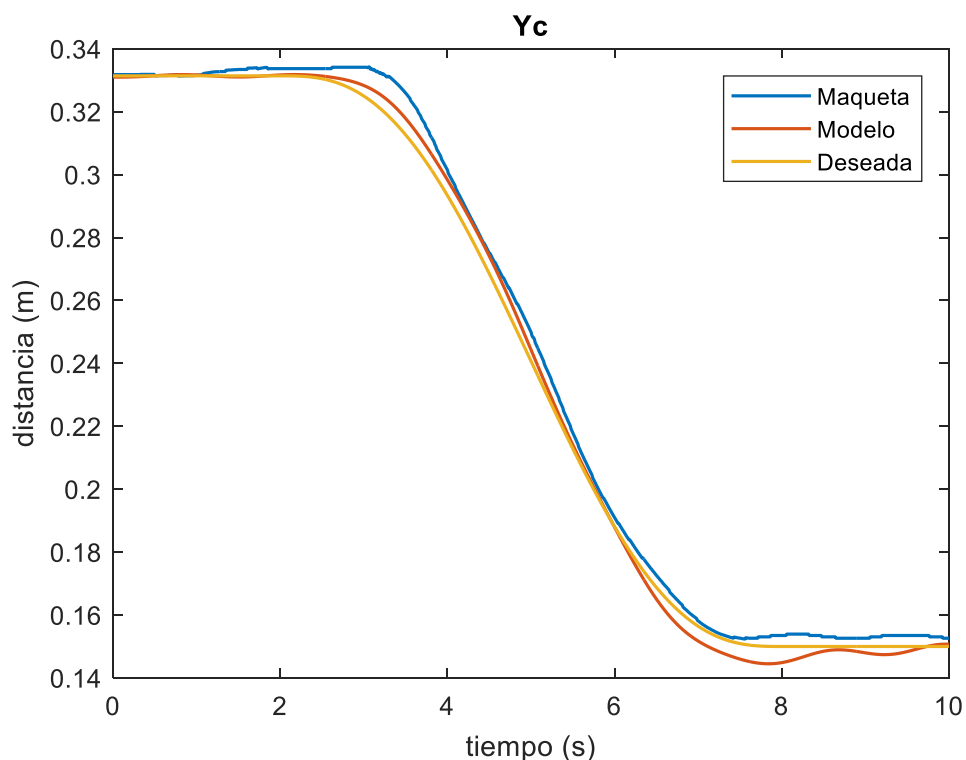


Figura 27. Resultados del control con realimentación para las segundas constantes

A simple vista, se aprecia una notable reducción en el error, aunque persiste cierto error de posición en estado estacionario. Además, aunque se observa una ligera oscilación al inicial el movimiento, es significativamente menor que en el caso anterior.

Finalmente, vemos cómo la carga sigue la trayectoria prácticamente sin oscilaciones, incluso al concluir la misma.

A pesar de que este control satisface los objetivos iniciales, se va a analizar a continuación una última estructura de control.

### 4.3. Control con prealimentación y realimentación

Este esquema de control, ilustrado en la Figura 28, combina los dos tipos mencionados anteriormente [24]. La acción total consiste en la suma de la fuerza calculada a partir de la trayectoria deseada y la acción de control del regulador PI. De esta manera, cuando la fuerza calculada no sea suficiente para que la seguir la trayectoria deseada, el controlador contribuirá proporcionando una mayor acción, mientras que cuando esta fuerza sea adecuada, el controlador no intervendrá. Al integrar la prealimentación y la realimentación en el control, se mejora a precisión, estabilidad y capacidad de respuesta del sistema.

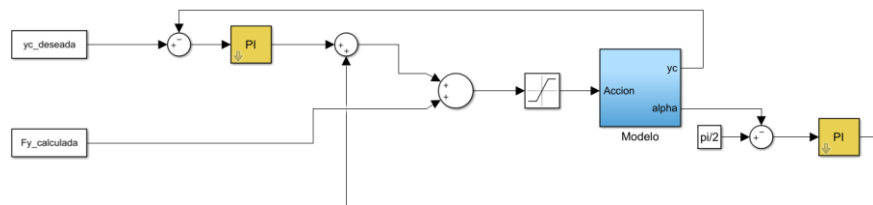


Figura 28. Control con prealimentación y realimentación

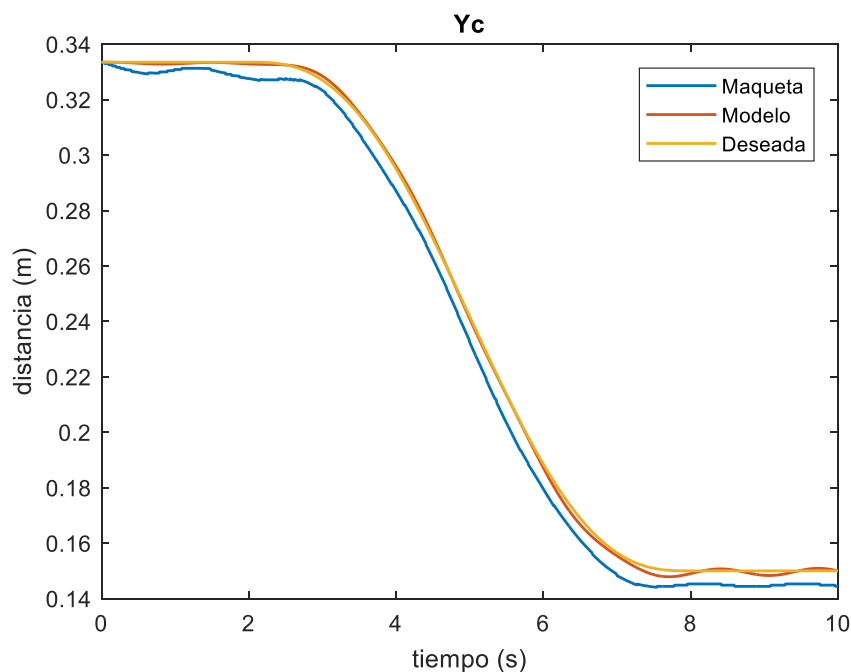


Figura 29. Resultados para el control combinado para las primeras constantes

Inicialmente se probarán los mismos parámetros utilizados en el caso anterior, que se detallaban en la Tabla 6. Los resultados obtenidos se muestran en la Figura 29.

Se observan ciertos errores en el transitorio y el estacionario. Aunque en la gráfica anterior parecen considerables, equivalen a menos de un centímetro de error. Vemos cómo apenas existe oscilación y que el tiempo de respuesta es adecuado. Por lo tanto, se ha decidido aumentar únicamente la constante integral a 30, con la intención de intentar minimizar el error, obteniendo lo que vemos a continuación, en la Figura 30.

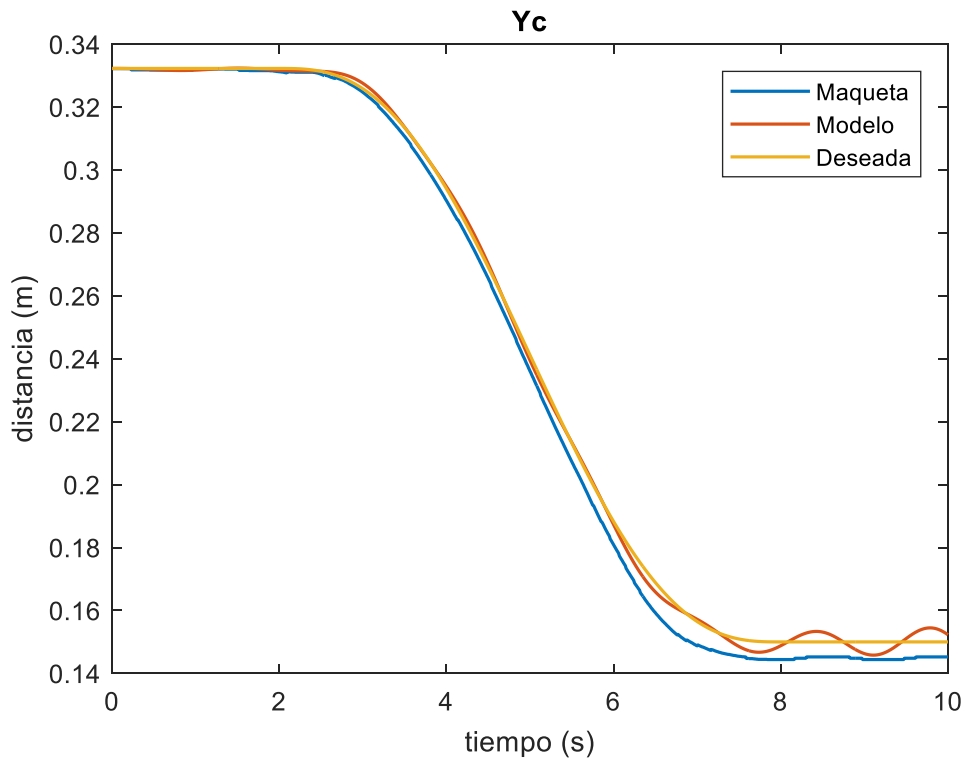


Figura 30. Resultados para el control combinado para las primeras constantes

Ahora vemos cómo se minimiza al máximo el error de posición a lo largo de toda la trayectoria, manteniéndose la misma libre de oscilación.

#### 4.4. Cálculo del error

A continuación se presenta un análisis de los errores asociados a las diferentes formas de control explicadas en el apartado anterior. En la Tabla 7 se muestran los errores cuadráticos medios raíz (RMSE) que comete la carga de la maqueta real, calculados según la Ecuación (30), en función de la amplitud de la trayectoria deseada.

Prealimentación	Realimentación		Combinación de ambos	
6,32%	Constantes iniciales	Constantes finales	Constantes iniciales	Constantes finales
	8,49%	3,12%	3.49%	2,63%

*Tabla 7. Comparación de los errores de los distintos tipos de control*

Se observa que el error es mínimo cuando se emplea una combinación de control con realimentación y prealimentación. Se puede concluir que se ha conseguido controlar la carga para que ejecute una trayectoria deseada con el mínimo error y oscilación posibles.

## 5. Conclusiones y trabajo futuro

En el último apartado, se presentan las conclusiones obtenidas durante la elaboración del trabajo fin de máster, junto con propuestas de mejoras o investigaciones futuras.

### 5.1. Conclusiones

Tras la elaboración de este trabajo, cabe destacar que el objetivo propuesto (modelado y control no lineal de un puente grúa), ha sido alcanzado satisfactoriamente.

El mundo de los puentes grúa es muy amplio, con una gran variedad de configuraciones y tipos de controladores que se adaptan a diferentes necesidades y aplicaciones industriales. Al tratarse de una estructura tan utilizada hoy en día en todo tipo de entornos y que además, mueve cargas muy pesadas, es esencial realizar un buen control sobre ellos. Controlar bien las cargas transportadas por los puentes grúas se traduce en una mayor seguridad y eficiencia.

En este contexto, es importante entender no solo las características físicas de los puentes grúa, sino también los principios dinámicos que rigen su comportamiento. En el presente trabajo se ha realizado un estudio exhaustivo del funcionamiento de un puente grúa, investigando las fuerzas y movimientos involucrados. Se ha profundizado en las ecuaciones matemáticas que describen su comportamiento dinámico, desde las más simples hasta las más complejas, para desarrollar un modelo que represente fielmente la realidad, consiguiendo reducir el error inicial en más de un 40%. Este análisis integral permite diseñar sistemas de control más efectivos, capaces de optimizar el rendimiento y minimizar los riesgos asociados con la operación de puentes grúa. Para optimizar los recursos computacionales en simulaciones, se procedió a la linealización del modelo. Aunque este proceso resultó en una pérdida significativa de precisión, el modelo linealizado sigue siendo útil para obtener una idea aproximada de la forma de la trayectoria cuando se aplica fuerza a la maqueta. Además, se propusieron tres esquemas de control diferentes, cuyo principal objetivo era seguir una trayectoria deseada con la carga, minimizando las oscilaciones. Estos esquemas de control fueron implementados y evaluados, logrando cumplir con los objetivos planteados con errores pequeños.

En resumen, a través de este trabajo se ha logrado un entendimiento profundo del modelado y control de puentes grúa, así como la implementación de soluciones prácticas y efectivas para su control. Este trabajo no solo cumple con los objetivos iniciales, sino que también sienta las bases para futuras investigaciones y mejoras en el control de estas importantes estructuras industriales.

### 5.2. Líneas futuras

En cuanto a las líneas futuras en el desarrollo de modelos matemáticos para el modelado y control de puentes grúa, es crucial la continua investigación en ese campo para seguir avanzando hacia soluciones más efectivas y precisas.

Una posible mejora a considerar en el futuro, específicamente para la maqueta utilizada, sería la implementación de una botonera física, externa al puente, que permita controlar los movimientos de manera rápida y eficiente con tan solo pulsar un botón. Esta botonera podría ofrecer desde movimientos simples hasta trayectorias más complejas preprogramadas, optimizando así la operación del puente grúa.

Además, podrían explorarse distintos tipos de controladores, como por ejemplo los Controladores Lógicos Programable (PLC) [25], ideales para automatizar procesos industriales y gestionar movimientos repetitivos de manera precisa y confiable.

Otra opción prometedora sería la integración de visión por computador en el control de la carga [26]. Utilizando cámaras para captar continuamente la posición de la carga y compararla con la posición deseada, se podrían ajustar automáticamente los movimientos de los motores para mantener una trayectoria precisa y segura.

La visión por computador también podría desempeñar un papel crucial en la seguridad, mediante la instalación de una cámara de 360 grados en el gancho que sujeta la carga. Esto permitiría detectar y evitar colisiones con operarios u otros objetos, proporcionando una capa adicional de protección en caso de fallas en el control o cálculos incorrectos de la trayectoria.

Además, la inclusión de inteligencia artificial en el control del puente grúa [27] podrían revolucionar su operación. Por ejemplo, mediante comandos de voz se podría permitir que el operador dicte los movimientos del carro, mejorando la accesibilidad y la eficiencia operativa.

Finalmente, una combinación avanzada de visión por computador e inteligencia artificial podría automatizar por completo las operaciones del puente grúa. La carga equipada con cámaras de 360 grados podría identificar objetos a levantar, navegar hacia ellos y determinar el punto de destino de manera autónoma reduciendo la necesidad de intervención humana y aumentando la seguridad y precisión en las operaciones industriales.

Estas propuestas representan áreas prometedoras para la innovación futura en el control de puentes grúa, aprovechando tecnologías emergentes para optimizar su rendimiento, seguridad y eficiencia operativa.



# Bibliografía

- [1] KORYTOV, Mikhail S.; SHCHERBAKOV, Vitaly S.; TITENKO, Vladimir V. Effect of the payload mass on forces acting from the overhead crane drives during movement in the mode of suppressing uncontrolled oscillations. En Journal of Physics: Conference Series. IOP Publishing, 2020. p. 012134.
- [2] COLLADO, Ricardo Esteban. Diseño de puente grúa de 5 Toneladas. 2010. Tesis Doctoral. Hochschule Mittweida.
- [3] INTECO, 3D crane System User's Manual. [En línea]  
Disponible: [https://www.inteco.com.pl/Docs/3DCrane\\_um.pdf](https://www.inteco.com.pl/Docs/3DCrane_um.pdf) 3
- [4] PITRÉ, Augustin. Estudio, diseño y mejora de un puente-grúa. 2024. Tesis de Licenciatura. Universitat Politècnica de Catalunya.
- [5] Pimeg. (2020, abril 17). Historia y origen de las grúas. PIMEG - Puentes Grúa. [En línea]  
Disponible: <https://puentegrúa.es/historia-y-origen-de-las-gruas>
- [6] JUANG, Jih-Gau; LIN, Ren-Wei; LIU, Wen-Kai. Comparison of classical control and intelligent control for a MIMO system. *Applied Mathematics and computation*, 2008, vol. 205, no 2, p. 778-791.
- [7] SASTRY, Sosale Shankara; ISIDORI, Alberto. Adaptive control of linearizable systems. *IEEE Transactions on Automatic Control*, 1989, vol. 34, no 11, p. 1123-1131.
- [8] LIU, Jiaying, et al. Artificial intelligence in the 21st century. *Ieee Access*, 2018, vol. 6, p. 34403-34421.
- [9] CHATURVEDI, Devendra K. Modeling and simulation of systems using MATLAB and Simulink. CRC press, 2017.
- [10] SCHETZEN, Martin. Nonlinear system modeling based on the Wiener theory. Proceedings of the IEEE, 1981, vol. 69, no 12, p. 1557-1573.
- [11] Mathworks. Implement MATLAB functions in Simulink with MATLAB Function blocks. Mathworks.com. [En línea]  
Disponible: <https://es.mathworks.com/help/simulink/ug/what-is-a-matlab-function-block.html?lang=en>
- [12] AL-BENDER, Farid; SWEVERS, Jan. Characterization of friction force dynamics. *IEEE Control Systems Magazine*, 2008, vol. 28, no 6, p. 64-81.

[13] Mathworks. Estimate Parameters from Measured Data. Mathworks.com. [En línea]

Disponible: <https://es.mathworks.com/help/slido/guide/estimate-parameters-from-measured-data-using-the-gui.html>

[14] BRULS, Johan, et al. Linear and non-linear system identification using separable least-squares. European Journal of Control, 1999, vol. 5, no 1, p. 116-128.

[15] CONN, Andrew R.; GOULD, Nicholas IM; TOINT, Philippe L. Trust region methods. Society for Industrial and Applied Mathematics, 2000.

[16] IBM. (2021, Agosto 17). Modelos lineales. Ibm.com. [En línea]

Disponible: <https://www.ibm.com/docs/es/spss-modeler/saas?topic=node-linear-models>

[17] Etchehoury, M. del R., & Muravchik, C. H. (1991). Linealización local de sistemas no lineales de tiempo continuo. IX Congreso Chileno de Ingeniería Eléctrica (Chile, 14 al 18 de octubre de 1991).

[18] SCHEI, Tor Steinar. A finite-difference method for linearization in nonlinear estimation algorithms. Automatica, 1997, vol. 33, no 11, p. 2053-2058.

[19] Mathworks. Jacobian. Mathworks.com. [En línea]

Disponible: <https://es.mathworks.com/help/symbolic/sym.jacobian.html>

[20] WANG, Weijie; LU, Yanmin. Analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model. En IOP conference series: materials science and engineering. IOP Publishing, 2018. p. 012049.

[21] LÉVINE, Jean. On necessary and sufficient conditions for differential flatness. Applicable Algebra in Engineering, Communication and Computing, 2011, vol. 22, no 1, p. 47-90.

[22] SUN, Ning, et al. A novel kinematic coupling-based trajectory planning method for overhead cranes. IEEE/ASME Transactions on Mechatronics, 2011, vol. 17, no 1, p. 166-173.

[23] ELLIOTT, Stephen J.; SUTTON, Trevor J. Performance of feedforward and feedback systems for active control. IEEE Transactions on Speech and Audio Processing, 1996, vol. 4, no 3, p. 214-223.

[24] POLYCARPOU, Marios M.; IOANNOU, Petros A. A robust adaptive nonlinear control design. En 1993 American control conference. IEEE, 1993. p. 1365-1369.

[25] BOLTON, William. Programmable logic controllers. Newnes, 2015.

[26] NILSSON, Roger; NORDGREN, Mikael. Vision-Based Control of the Overhead Crane. MSc Theses, 2003.

[27] NEMCIK, Jakub, et al. The use of neural network for nonlinear predictive control design for and overhead crane. En 2021 21st international conference on control, automation and systems (ICCAS). IEEE, 2021. p. 725-730.

## Anexo A. Código C

```
#include <conio.h>
#include <math.h>

#define S_FUNCTION_NAME model3dddm

#include "simstruc.h" /* Where simulation structure, S, is defined */

#ifdef MATLAB_MEX_FILE
#include "mex.h"
#endif

/* Input Arguments */
/*#define MODEL_PARAMETERS_ARG    ssGetArg(S,1) */

#define XWOZEK                ssGetArg(S,0)
#define VXWOZEK                ssGetArg(S,1)
#define YWOZEK                ssGetArg(S,2)
#define VYWOZEK                ssGetArg(S,3)
#define ALFA                  ssGetArg(S,4)
#define VALFA                  ssGetArg(S,5)
#define BETA                  ssGetArg(S,6)
#define VBETA                  ssGetArg(S,7)
#define RLINKI                ssGetArg(S,8)
#define VRLINKI                ssGetArg(S,9)

#define SAMPLE_TIME_ARG        ssGetArg(S,10)
#define Rstale                  ssGetArg(S,11)
#define NUMBER_OF_ARGS        (12)
#define NSAMPLE_TIMES          (1)

static double mc = 1;
static double mw = 1.155;
static double ms = 2.20;
static double g = 9.81;
static double Tx=100;
static double Ty=82;
static double Tz=75;

static double R;
static double My,M,mi1,mi2,T1,T2,T3,Tsx,Tsy,Tsz;

static void mdlInitializeSizes(S)
    SimStruct *S;{

#ifdef MATLAB_MEX_FILE
    if (ssGetNumArgs(S) != NUMBER_OF_ARGS)
        mexErrMsgTxt("Wrong number of input arguments passed.\nTwelve
arguments are expected.\n");
#endif
    ssSetNumContStates(S, 10);
    ssSetNumDiscStates(S, 0);
    ssSetNumInputs(S,3);
    ssSetNumOutputs(S, 10);
    ssSetDirectFeedThrough(S, 0);
    ssSetNumInputArgs(S, NUMBER_OF_ARGS);
    ssSetNumSampleTimes(S, NSAMPLE_TIMES);
}
```

```

/* Function to initialize sample times */
static void mdlInitializeSampleTimes(S)
    SimStruct *S; {

#ifdef MATLAB_MEX_FILE
    ssSetSampleTimeEvent(S, 0, mxGetPr(SAMPLE_TIME_ARG)[0]);
    ssSetSampleTimeEvent(S, 0, CONTINUOUS_SAMPLE_TIME);
    ssSetOffsetTimeEvent(S, 0, 0);
#endif

}

void ustaw() {

    My = mw+ms;
    M=mw*mw+mc*ms+mw*ms+mc*mw;

    m1=mc/mw;
    m2=mc/(mw+ms);

    T1 = Ty/mw;
    T2 = Tx/(mw+mc);
    T3 = Tz/mc;

    Tsx = 5/(mw+ms);
    Tsy = 7.5/mw;
    Tsz = 10/mc;

}

static void mdlInitializeConditions(x0, S)
    double *x0;
    SimStruct *S; {

    double xw = (double)mxGetPr(XWOZEK)[0];
    double vxw = (double)mxGetPr(VXWOZEK)[0];
    double yw = (double)mxGetPr(YWOZEK)[0];
    double vyw = (double)mxGetPr(VYWOZEK)[0];
    double al = (double)mxGetPr(ALFA)[0];
    double val = (double)mxGetPr(VALFA)[0];
    double be = (double)mxGetPr(BETA)[0];
    double vbe = (double)mxGetPr(VBETA)[0];
    double r = (double)mxGetPr(RLINKI)[0];
    double vr = (double)mxGetPr(VRLINKI)[0];

    x0[0]=xw;
    x0[1]=vxw;
    x0[2]=yw;
    x0[3]=vyw;
    x0[4]=(3.1415926/2)-al; //La posicion de equilibrio se considera alpha
                          // = 0 aunque en la realidad es alpha = pi/2
    x0[5]=val;
    x0[6]=be;
    x0[7]=vbe;
    x0[8]=r;
    x0[9]=vr;
    ustaw();

}

```

```

double sign(x)
double x; {

if (x==0) {return 0;}
if (x>0) {return 1;}
if (x<0) {return -1;}
return 0;

}

/* Function to compute outputs */
static void mdlOutputs(y, x, u, S, tid)
double *y;
double *x;
double *u;
SimStruct *S;
int tid; {

y[0]=x[0];
y[1]=x[1];
y[2]=x[2];
y[3]=x[3];
y[4]=(3.1415926/2)-x[4];
y[5]=x[5];
y[6]=x[6];
y[7]=x[7];
y[8]=x[8];
y[9]=x[9];

}

/* Function to compute derivatives */
static void mdlDerivatives(dx, x, u, S, tid)
double *dx;
double *x;
double *u;
SimStruct *S;
int tid; {

int i;
double un[3];
double RR = (double)mxGetPr(Rstale)[0];

un[0]=u[0]/mw;
un[1]=u[1]/(mw+mc);
un[2]=u[2]/mc;

dx[0] = x[1];

dx[1] = un[0]-x[1]*T1-Tsy*sign(x[1])+mi1*cos(x[4])*un[2]-
mi1*cos(x[4])*(-x[9]*T3-Tsz*sign(x[9]));

dx[2] = x[3];

dx[3] = un[1]-x[3]*T2-Tsx*sign(x[3])+mi2*sin(x[4])*sin(x[6])*un[2]-
mi2*sin(x[4])*sin(x[6])*(-x[9]*T3-Tsz*sign(x[9]));

dx[4] = x[5];

```

```

dx[5] = (-
(x[1]*T1+Tsy*sign(x[1]))*sin(x[4])+un[0]*sin(x[4])+cos(x[4])*x[8]*sin(
x[4])*x[7]*x[7]-cos(x[4])*un[1]*sin(x[6])+cos(x[4])*cos(x[6])*g-
cos(x[4])*mi2*sin(x[4])*sin(x[6])*sin(x[6])*un[2]+cos(x[4])*mi2*sin(x[
4])*sin(x[6])*sin(x[6])*(-x[9]*T3-
Tsz*sign(x[9]))+cos(x[4))*(x[3]*T2+Tsx*sign(x[3]))*sin(x[6])+cos(x[4])
*mi1*un[2]*sin(x[4])-cos(x[4])*mi1*(-x[9]*T3-
Tsz*sign(x[9]))*sin(x[4])-2*x[9]*x[5])/x[8];

dx[6] = x[7];

dx[7] = -
(cos(x[6])*un[1]+g*sin(x[6])+2*x[8]*cos(x[4])*x[5]*x[7]+cos(x[6])*mi2*
sin(x[4])*sin(x[6])*un[2]-cos(x[6])*mi2*sin(x[4])*sin(x[6])*(-x[9]*T3-
Tsz*sign(x[9]))+2*x[9]*sin(x[4])*x[7]-
cos(x[6))*(x[3]*T2+Tsx*sign(x[3])))/(x[8]*sin(x[4]));

if (RR==1) {dx[8]=0;dx[9]=0;}
else {
dx[8] = x[9];

dx[9] = cos(x[4))*(x[1]*T1+Tsy*sign(x[1]))-
cos(x[4])*un[0]+x[8]*sin(x[4])*sin(x[4])*x[7]*x[7]-
un[1]*sin(x[4])*sin(x[6])+sin(x[4])*cos(x[6])*g-
mi2*sin(x[4])*sin(x[4])*sin(x[6])*sin(x[6])*un[2]+mi2*sin(x[4])*sin(x[
4])*sin(x[6])*sin(x[6])*(-x[9]*T3-
Tsz*sign(x[9]))+(x[3]*T2+Tsx*sign(x[3]))*sin(x[4])*sin(x[6])-
mi1*un[2]+mi1*un[2]*sin(x[4])*sin(x[4])+mi1*(-x[9]*T3-Tsz*sign(x[9]))-
mi1*(-x[9]*T3-Tsz*sign(x[9]))*sin(x[4])*sin(x[4])+x[8]*x[5]*x[5]-
un[2]-x[9]*T3-Tsz*sign(x[9]));
}
}

#ifdef MATLAB_MEX_FILE */
#include "simulink.c"
#else
#include "cg_sfun.h"
#endif

```

## Anexo B. Función de Matlab con modelo no lineal

```
function dx = dinamica(estado,entrada,Tx, Ty1, Tr, Tsx1, Tsyl,  
Tsz1,p,p1,p2)  
  
%%Masas  
mc = 1; mw = 1.155; ms = 2.20; %MasaCarga MasaCarro MasaRail  
mi1=mc/mw;  mi2=mc/(mw+ms);  
  
%%Tensiones  
T1 = Ty1/mw; T2 = Tx/(mw+mc); T3 = Tr/mc;  
Tsx = Tsx1/(mw+ms); Tsy = Tsyl/mw;  Tsz = Tsz1/mc;  
  
%%Gravedad  
g = 9.81;  
  
x = [estado(3) %y  
      estado(4) %yp  
      estado(1) %x  
      estado(2) %xp  
      estado(7) %alpha  
      estado(8) %alphap  
      estado(9) %beta  
      estado(10) %betap  
      estado(5) %R  
      estado(6)]; %Rp  
  
un = [entrada(2)/mw entrada(1)/(mw+mc) entrada(3)/mc];  
  
%Fuerzas de friccion  
if x(4)>0  
    tanhx = Tsx*tanh(x(4));  
elseif x(4)<0  
    tanhx = 0;  
else  
    tanhx = p*tanh(x(4));  
end  
  
if x(2)>0  
    tanhy = Tsy*tanh(x(2));  
elseif x(2) ==0  
    tanhy = 0;  
else  
    tanhy = p1*tanh(x(2));  
end  
  
if x(10)>0  
    tanhz = Tsz*tanh(x(10));  
elseif x(10) ==0  
    tanhz = 0;  
else  
    tanhz = p2*tanh(x(10));  
end  
  
dx = [x(4)
```



```

un(2)-x(4)*T2-tanhx+mi2*sin(x(5))*sin(x(7))*un(3)-
mi2*sin(x(5))*sin(x(7))*(-x(10)*T3-tanhz)

x(2)
un(1)-x(2)*T1-tanhy+mi1*cos(x(5))*un(3)-mi1*cos(x(5))*(-
x(10)*T3-tanhz)

x(10)
cos(x(5))*(x(2)*T1+tanhy)-
cos(x(5))*un(1)+x(9)*sin(x(5))*sin(x(5))*x(8)*x(8)-
un(2)*sin(x(5))*sin(x(7))-
mi2*sin(x(5))*sin(x(5))*sin(x(7))*sin(x(7))*un(3)+mi2*sin(x(5))*
sin(x(5))*sin(x(7))*sin(x(7))*(-
x(10)*T3)+(x(4)*T2+tanhx)*sin(x(5))*sin(x(7))-
mi1*un(3)+mi1*un(3)*sin(x(5))*sin(x(5))+mi1*(-x(10)*T3)-mi1*(-
x(10)*T3)*sin(x(5))*sin(x(5))+x(9)*x(6)*x(6)-un(3)-x(10)*T3

x(6)
(-
(x(2)*T1+tanhy)*sin(x(5))+un(1)*sin(x(5))+cos(x(5))*x(9)*sin(x(5))
)*x(8)*x(8)-cos(x(5))*un(2)*sin(x(7))+cos(x(5))*cos(x(7))*g-
cos(x(5))*mi2*sin(x(5))*sin(x(7))*sin(x(7))*un(3)+cos(x(5))*mi2*
sin(x(5))*sin(x(7))*sin(x(7))*(-x(10)*T3-
tanhz)+cos(x(5))*(x(4)*T2+tanhx)*sin(x(7))+cos(x(5))*mi1*un(3)*s
in(x(5))-cos(x(5))*mi1*(-x(10)*T3-tanhz)*sin(x(5))-
2*x(10)*x(6))/x(9)

x(8)
-
(cos(x(7))*un(2)+g*sin(x(7))+2*x(9)*cos(x(5))*x(6)*x(8)+cos(x(7))
)*mi2*sin(x(5))*sin(x(7))*un(3)-
cos(x(7))*mi2*sin(x(5))*sin(x(7))*(-x(10)*T3-
tanhz)+2*x(10)*sin(x(5))*x(8)-
cos(x(7))*(x(4)*T2+tanhx))/(x(9)*sin(x(5)))
];

```

**end**

## Anexo C. Función de Matlab con modelo lineal

```
function Xp = Modelo(vec)

%%Masas
mc = 1; mw = 1.155; ms = 2.20; %Carga Carro Rail

%%Tensiones
Tx = 138.0356468143783; Ty = 77.559849304614400; Tr = 55.710193797799110;
Tsx1= 40.343033652015780; Tsyl= 4.837007135965244; Tsz1= 4.925295564634344e+02;
T1 = Ty/mw; T2 = Tx/(mw+mc); T3 = Tr/mc;

%%Gravedad
g = 9.81;

%%Longitud del cable inicial
R = 0.25;

%%Modelo lineal

A = [0 1 0 0 0 0 0 0 0 0
      0 -T2 0 0 0 0 0 0 0 0
      0 0 0 1 0 0 0 0 0 0
      0 0 0 -T1 0 0 0 0 0 0
      0 0 0 0 0 1 0 0 0 0
      0 0 0 0 0 -T3 0 0 0 0
      0 0 0 0 0 0 0 1 0 0
      0 0 0 -T1/R 0 0 -(g)/R 0 0 0
      0 0 0 0 0 0 0 0 0 1
      0 T2/R 0 0 0 0 0 0 -(g)/R 0];

B = [0 0 0
      1 0 0
      0 0 0
      0 1 0
      0 0 0
      0 0 -1
      0 0 0
      0 1/R 0
      0 0 0
      -1/R 0 0];

C = [1 0 0 0 0 0 0 0 0 0 %x
      0 0 1 0 0 0 0 0 0 0 %y
      0 0 0 0 1 0 0 0 0 0 %R
      0 0 0 0 0 0 1 0 0 0 %alpha
      0 0 0 0 0 0 0 0 1 0 %beta];

u = [vec(1); vec(2); vec(3)]; %Fx Fy Fr

x = [vec(4)
      vec(5)
      vec(6)
      vec(7)]
```

```
vec(8)
vec(9)
vec(10)
vec(11)
vec(12)
vec(13)];

Xp = A*x + B*u;

end
```