



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

TRABAJO FINAL DE GRADO

IMPLEMENTACIÓN DE ALGORITMOS DE CONTROL AVANZADOS
MPC EN PLC'S INDUSTRIALES

Curso 2018-2019

Autor:

Diego Ferri Beltrán (*dieferbe@etsid.upv.es*)

Tutor:

Sergio García-Nieto Rodríguez

Grado en Ingeniería Electrónica Industrial y Automática
Escuela Técnica Superior de Ingeniería del Diseño - ETSID
Universidad Politécnica de Valencia - UPV

Valencia, Julio de 2019

CONTENIDO DEL PROYECTO:

Este documento contiene por este orden los siguientes documentos:

- Memoria
- Pliego de condiciones
- Presupuesto



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

TRABAJO FINAL DE GRADO

IMPLEMENTACIÓN DE ALGORITMOS DE CONTROL AVANZADOS
MPC EN PLC'S INDUSTRIALES

MEMORIA

Curso 2018-2019

Autor:

Diego Ferri Beltrán (*dieferbe@etsid.upv.es*)

Tutor:

Sergio García-Nieto Rodríguez

Grado en Ingeniería Electrónica Industrial y Automática
Escuela Técnica Superior de Ingeniería del Diseño - ETSID
Universidad Politécnica de Valencia - UPV

Valencia, Julio de 2019

RESUMEN

El presente proyecto con título “Implementación de algoritmos de control avanzados MPC en PLC’s industriales” consiste en el diseño del software necesario para la implementación de un algoritmo de control avanzado como es el control predictivo por modelo en un PLC real, Modicon 241 de Schneider, y la posterior aplicación en el control de servomecanismos, Twin Rotor Mimo System de Feedback Instruments.

Dicha implementación se lleva a cabo mediante el diseño de un MPC multivariable con Matlab para el modelo linealizado del sistema TRMS y la posterior exportación del controlador en texto estructurado para la integración del mismo en el entorno de SoMachine, donde se recibirá la posición en ángulos de la inclinación y la rotación del sistema y proporcionará una determinada acción de control para alcanzar la posición requerida como referencia.

Palabras Clave: MPC, control, PLC’s, TRMS, Matlab, SoMachine

RESUM

El present projecte amb títol "Implementació d'algoritmes de control avançats MPC en PLCs industrials" consisteix en el disseny del software necessari per a la implementació d'un algoritme de control avançat com és el control predictiu per model en un PLC real, Modicon 241 de Schneider, i la posterior aplicació en el control de servomecanismes, Twin Rotor MIMO System de Feedback Instruments.

Aquesta implementació es du a terme mitjançant el disseny d'un MPC multivariable amb Matlab per al model linealitzat del sistema TRMS i la posterior exportació del controlador en text estructurat per a la integració del mateix en l'entorn de SoMachine, on es rebrà la posició en angles de la inclinació i la rotació del sistema i proporcionarà una determinada acció de control per aconseguir la posició requerida com a referència.

Paraules Clau: MPC, control, PLC's, TRMS, Matlab, SoMachine

ABSTRACT

The present project titled "Implementation of advanced control algorithms MPC in industrial PLCs" consists in the design of the software necessary for the implementation of an advanced control algorithm such as the predictive control by model in a real PLC, Modicon 241 of Schneider, and the subsequent application in the control of servomechanisms, Twin Rotor Mimo System of Feedback Instruments.

This implementation is carried out through the design of a multivariable MPC with Matlab for the linearized model of the TRMS system and the subsequent export of the structured text controller for its integration in the SoMachine environment, where the position will be received at angles of the inclination and rotation of the system and will provide a certain control action to reach the required position as a reference.

Keywords: MPC, control, PLC's, TRMS, Matlab, SoMachine

INDICE DE LA MEMORIA:

1.	Introducción	16
2.	Objetivo y alcance del proyecto	17
3.	Antecedentes.....	19
3.1.	Control predictivo	19
3.2.	Autómata M241CE40R.....	22
3.3.	NI USB-6001	24
3.4.	Twin Rotor Mimo System	25
4.	Estudio de Soluciones Alternativas	33
4.1.	C-Integration de CodeSys.....	33
4.2.	MPC Controller de Matlab	34
5.	Solución Adoptada.....	35
5.1.	Descripción General de la Solución	35
5.2.	Verificación del funcionamiento de la solución adoptada	35
5.2.1.	Generación de texto estructurado con PLC Coder de Matlab	38
5.2.2.	Generación de las POU's con CodeSys	39
5.2.3.	Creación del proyecto en SoMachine.....	40
5.2.4.	Comunicación con NI USB-6001	42
5.2.5.	Adquisición de datos en Simulink.....	43
5.3.	Implementación de la solución para TRMS	45
5.3.1.	Identificación de los parámetros de funcionamiento del TRMS.....	46
5.3.2.	Diseño controlador MPC	47
5.3.3.	Simulación del modelo linealizado.....	48
5.3.4.	Integración del código en SoMachine	53
6.	Resultados obtenidos	55
7.	Conclusiones.....	62
8.	Bibliografía.....	63

LISTA DE FIGURAS:

Fig. 1 Evolución Industrial en el tiempo.....	16
Fig. 2 Diagrama de flujo correspondiente al diseño software.....	17
Fig. 3 Diagrama de flujo correspondiente al diseño del MPC	18
Fig. 4 Metodología de un MPC	20
Fig. 5 Estructura básica de un MPC	21
Fig. 6 Autómata M241CE40R	22
Fig. 7 Estructura del M241 en el laboratorio.....	23
Fig. 8 Tarjeta de adquisición USB-6001	24
Fig. 9 Caja de conectores banana	25
Fig. 10 Modelo académico de un helicóptero	26
Fig. 11 Esquemático del sistema TRMS	27
Fig. 12 Twin Rotor MIMO System.....	27
Fig. 13 Esquema eléctrico de los motores de las hélices.....	28
Fig. 14 Plano Horizontal del TRMS.....	29
Fig. 15 Plano vertical del TRMS.....	30
Fig. 16 Funciones CodeSys con C-Integration.....	33
Fig. 17 Configuración Device CodeSys	34
Fig. 18 Error proporcionado tras compilar en CodeSys.....	34
Fig. 19 Esquema Simulink Ejemplo	37
Fig. 20 Respuesta del sistema de ejemplo.....	37
Fig. 21 Acción de control del controlador de ejemplo	38
Fig. 22 Configuración del bloque MPC.....	38
Fig. 23 Configuración del subsistema de control.....	39
Fig. 24 Configuración PLC Coder	39
Fig. 25 Dispositivos de SoMachine	41
Fig. 26 Bloque conversión INT a REAL	41
Fig. 27 Bloque de escalado de la señal	42
Fig. 28 Bloque principal de control	42
Fig. 29 Esquema Simulink comunicación con PLC	43
Fig. 30 Acción de control recibida del PLC.....	44
Fig. 31 Gráfica de la respuesta del sistema y la referencia	44
Fig. 32 Diagrama de flujo correspondiente a la implementación del MPC en el PLC real	46
Fig. 33 Experimento de simulación 1.....	48
Fig. 34 Esquema Simulink con el controlador MPC	49
Fig. 35 Gráfica de las salidas para observar el efecto de la inclinación sobre la rotación	49
Fig. 36 Gráfica de las acciones de control influencia inclinación	50

Fig. 37 Gráfica de las salidas para observar el efecto de la rotación sobre la inclinación	50
Fig. 38 Gráfica de las acciones de control para observar el efecto de la rotación sobre la inclinación.....	51
Fig. 39 Gráfica de las salidas del sistema.....	52
Fig. 40 Gráfica de las acciones de control del MPC en Simulink	52
Fig. 41 Experimento de simulación 2.....	53
Fig. 42 Esquema Simulink PLC-Modelo Lineal	54
Fig. 43 Configuración tarea MAST	55
Fig. 44 Gráfica de la respuesta para el controlador simulado.....	56
Fig. 45 Gráfica de la respuesta para el controlador real	57
Fig. 46 Gráfica de las acciones de control para el controlador real	57
Fig. 47 Gráfica de la respuesta para el controlador simulado.....	58
Fig. 48 Gráfica de las acciones de control para el controlador simulado.....	59
Fig. 49 Gráfica de la respuesta para el controlador real	59
Fig. 50 Gráfica de las acciones de control para el controlador real	60

LISTA DE TABLAS:

Tabla 1 Comparativa resultados experimento 2	60
Tabla 2 Comparativa resultados experimento 2	61

LISTA DE SIMBOLOS:

V_t	Voltaje mandado al motor principal
V_m	Voltaje mandado al motor de cola
φ	Ángulo de inclinación
Ψ	Ángulo de rotación
$\dot{\varphi}$	Velocidad angular en el eje horizontal
$\dot{\Psi}$	Velocidad angular en el eje vertical
$\ddot{\varphi}$	Aceleración angular en el eje horizontal
$\ddot{\Psi}$	Aceleración angular en el eje vertical
φ_0	Inclinación inicial
Ψ_0	Rotación inicial
ω_m	Velocidad angular del motor principal

ω_t	Velocidad angular del motor de cola
$\dot{\omega}_m$	Aceleración angular del motor principal
$\dot{\omega}_t$	Aceleración angular del motor de cola
$J_{p,t}$	Inercia causada por la hélice de cola
$J_{p,m}$	Inercia causada por la hélice principal
$k_{i,m}^t$	Constante del circuito eléctrico del motor
$\frac{R_t}{m}$	Resistencia del motor
$\frac{L_t}{m}$	Inductancia del motor
$\frac{i_t}{m}$	Corriente del motor
$k_{FM,m}^t$	Constante de la fricción del motor
$k_{air,t/m}$	Constante de la resistencia del aire
J_φ	Inercia en referencia al eje vertical
k_ζ	Constante de la fuerza centrífuga
$k_{p,\varphi}$	Constante de la propulsión del motor principal
$k_{p,t}$	Constante del par angular motivado por la inercia de la hélice de cola
$\tilde{k}_{air,t}$	Constante del efecto de acoplamiento cruzado motivado por la resistencia del aire de la hélice de cola
$k_{FC,\varphi}$	Constante de la fricción motivada por una rotación alrededor del eje horizontal
$k_{FS,\varphi}$	Constante correspondiente a la parte del par de fricción que depende del signo de la rotación alrededor del eje horizontal
J_ψ	Inercia en referencia al eje horizontal
$k_{p,\psi}$	Constante de propulsión del motor de cola
k_c	Constante del par causado por el cable de los sensores
$k_{FC,\psi}$	Constante de la fricción motivada por una rotación alrededor del eje vertical
$k_{FS,\psi}$	Constante correspondiente a la parte del par de fricción que depende del signo de la rotación alrededor del eje vertical

$k_{p,m}$	Constante del par angular motivado por la inercia de la hélice principal
$\tilde{k}_{air,m}$	Constante del efecto de acoplamiento cruzado motivado por la resistencia del aire de la hélice principal
$\tilde{k}_{p,m}$	Constante del efecto de acoplamiento cruzado motivado por el empuje de la rotación de la hélice principal
d	Longitud entre el punto de conexión de todas las vigas y el eje vertical.
g	Constante gravitacional

1. Introducción

El concepto de Industria 4.0 se utiliza para hacer referencia a la cuarta revolución industrial, el término fue emitido por el gobierno alemán para describir la fábrica inteligente. La aparición e introducción de nuevas tecnologías y nuevos métodos de trabajo han supuesto un cambio en la forma en la que se trabaja actualmente, permitiendo digitalizar y conectar entre si maquinas, personas y dispositivos.

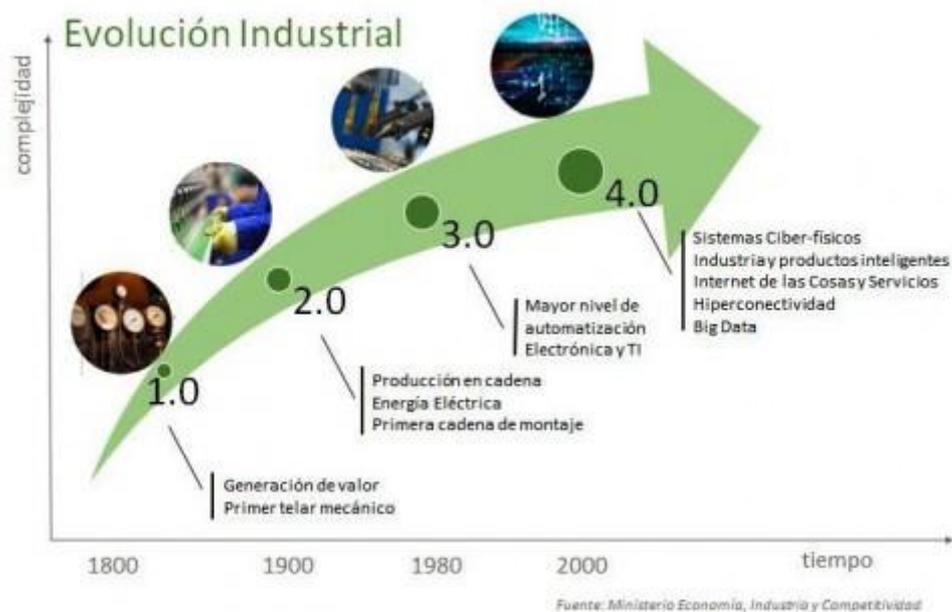


Fig. 1 Evolución Industrial en el tiempo

Nos encontramos frente a un transformación del modelo de producción tal y como lo conocemos. Sin embargo, no todos los procesos permiten un cambio de sistemas y equipos. Por lo que es necesario adaptar los equipos ya instalados o integrar nuevos dispositivos para permitir la adaptación necesaria para el desarrollo de la industria 4.0. En las actuales factorías se cuenta con PLC's industriales para la automatización de sus procesos industriales con un software que implementa, en su gran mayoría, algoritmos de control de tipo PID. No obstante, los avances y desarrollos tecnológicos hacen necesario que se puedan implementar estrategias de control más avanzadas y no lineales como pueden ser el control predictivo o inteligente en los PLC's industriales.

2. Objetivo y alcance del proyecto

El principal objetivo de este proyecto es el diseño del software necesario para la implementación de un algoritmo de control avanzado, como es el control predictivo por modelo, en un PLC real y la posterior aplicación en el control de servomecanismos.

El primer objetivo definido en el proyecto será explorar las posibles soluciones que permitan la correcta integración del controlador en el entorno del PLC real.

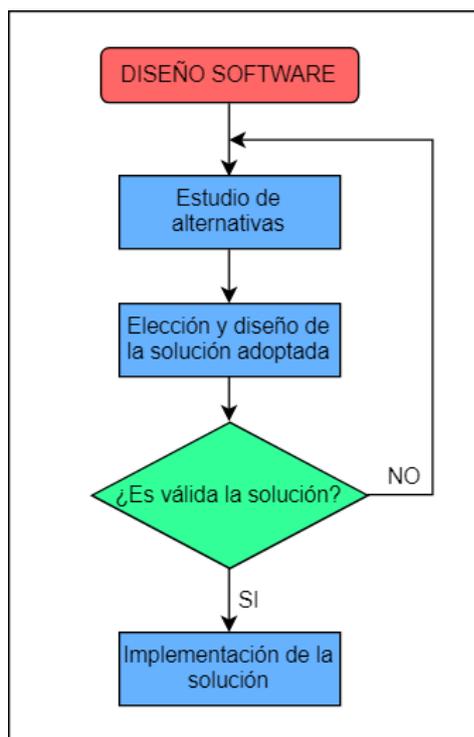


Fig. 2 Diagrama de flujo correspondiente al diseño software

El segundo objetivo del proyecto es obtener un modelo matemático que represente con precisión el comportamiento dinámico del sistema a controlar, para además de permitir el correcto diseño del controlador y la simulación.

El tercer objetivo es el diseño de un controlador MPC, siguiendo el proceso descrito mediante el diagrama de flujo de la figura 3, capaz de hacer frente a la dinámica del sistema, regulándola de forma que se minimice el error entre la salida del sistema y la trayectoria de referencia seleccionada.

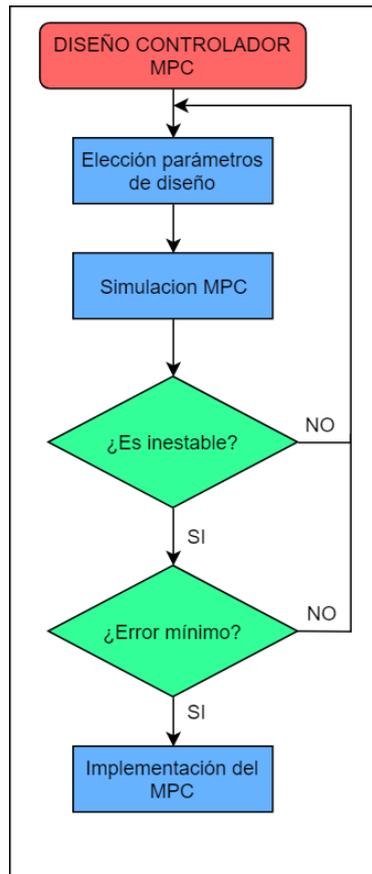


Fig. 3 Diagrama de flujo correspondiente al diseño del MPC

El cuarto objetivo del proyecto es el de realizar una correcta comunicación entre PLC y ordenador para que la ejecución sea en tiempo real, para ello se deben transformar las señales para adaptarlas al formato de trabajo del sistema al que se envía. Esto permitirá que el control del proceso en el PC mediante el PLC se realice de forma correcta.

Por último, se debe estudiar las características de las diferentes respuestas que se obtienen cambiando los parámetros de configuración de los controladores implementados. Esto permitirá analizar que conjunto obtiene la respuesta que mejor se adapta a las especificaciones.

3. Antecedentes

3.1. Control predictivo

El control predictivo por modelo (MPC) se desarrolló a finales de los años setenta y ha tenido una considerable evolución desde entonces. El concepto de control predictivo hace referencia a un conjunto de técnicas de control que mediante el uso del modelo del proceso predice las salidas futuras optimizando de esta manera las acciones de control. Para ello hace uso de los errores entre la referencia y las variables a controlar en los instantes tanto anteriores como futuros [2]. Para la realización de esta técnica de control se utilizan los siguientes parámetros:

- Horizonte de predicción (p): Es el número de instantes futuros que se tienen en cuenta a la hora de calcular las acciones de control. Debe cubrir las dinámica significativa del sistema.
- Horizonte de control (m): Es la acción de control en cada instante de tiempo. Cuanto más pequeño sea su valor menor son los cálculos a realizar, y cuanto mayor sea mejores son las predicciones proporcionadas, a costa de aumentar la complejidad.
- Tiempo de muestreo (T_s): Es la velocidad a la que el controlador ejecutará el algoritmo de control, no debe tener un valor demasiado alto porque si no el controlador no podrá hacer frente a las posibles perturbaciones sin embargo si es extremadamente pequeño supone una carga computacional excesiva.

La metodología en la que se basan los controladores MPC se caracteriza por seguir la siguiente estrategia, representada en la figura 4:

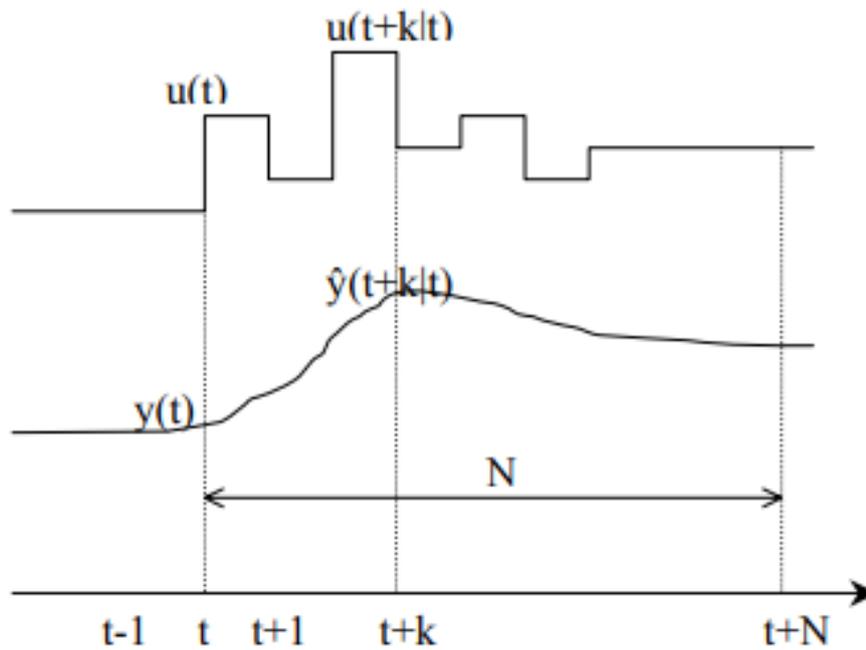


Fig. 4 Metodología de un MPC

1. Para cada instante t se predicen las salidas futuras utilizando el modelo del proceso para cierto horizonte de predicción N . Las predicciones de las salidas $\hat{y}(t+k|t)$ para $k=1\dots N$ dependen de los valores en el instante t y de las futuras acciones de control que se vayan a aplicar $u(t+k|t)$ para $k=1\dots N-1$ que han de ser calculadas.
2. Las acciones de control futuras se calculan de acuerdo con la optimización de un determinado criterio para mantener al proceso lo más cerca posible de la trayectoria de referencia $w(t+k)$. Este criterio tiene la forma de la función cuadrática del error entre la salida predicha y la trayectoria de referencias futuras.

$$J = \sum_{j=N_1}^{N_2} \delta(j) [\hat{y}(t+j|t) - w(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j) [\Delta u(t+j-1)]^2 \quad (1)$$

Donde: $\hat{y}(t+j|t)$ es la predicción óptima de la salida j pasos hacia delante, N_1 y N_2 son los horizontes mínimo y máximo de predicción y N_u es el horizonte de control, $\delta(j)$ y $\lambda(j)$ son secuencias de ponderación y $w(t+j)$ la futura trayectoria de referencia.

3. La acción de control $u(t|t)$ es enviada al proceso mientras que el resto de las señales calculadas son desechadas, ya que $u(t+1|t+1)$ es calculada

con una información diferente. Se aplica según una política denominada horizonte móvil.

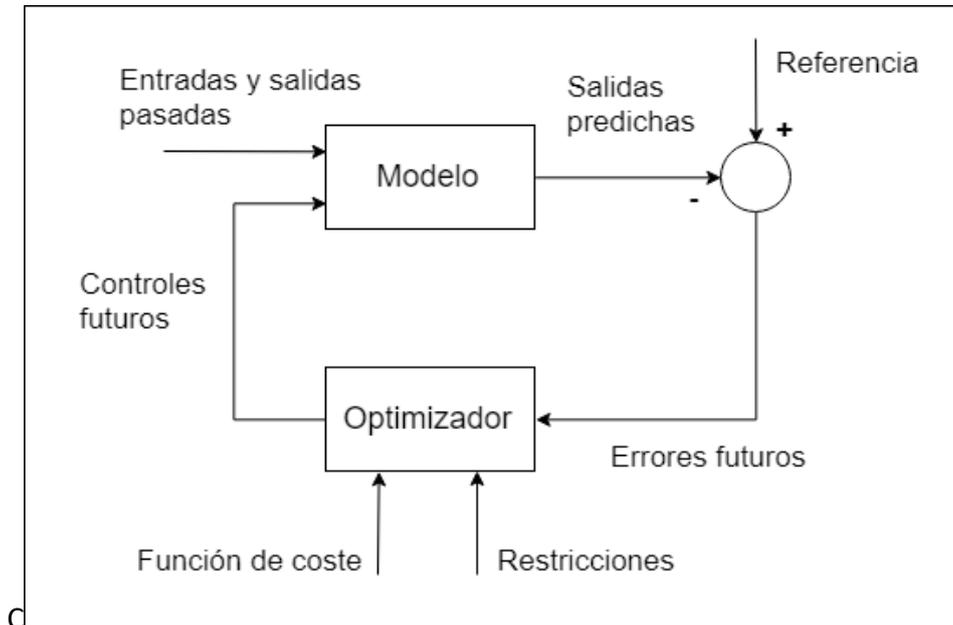


Fig. 5 Estructura básica de un MPC

La figura 5 muestra la estructura básica que siguen este tipo de controladores, en la que se observan dos partes fundamentales: el modelo del proceso y el optimizador. El modelo del proceso debe ser capaz de capturar la dinámica del sistema para predecir como evolucionarán las salidas de forma precisa. Por otro lado, el optimizador es el que permite obtener las señales de control a aplicar al sistema. Si el controlador a diseñar es muy sencillo (modelo lineal, sin restricciones, etc.) se puede obtener una solución explícita, de no ser así se debe acudir a un algoritmo numérico de optimización que requiere mayor capacidad de cálculo. El tamaño del cálculo depende del número de variables y restricciones que tenga el sistema, así como del horizonte de predicción y control.

El control predictivo presenta las siguientes ventajas respecto a otros métodos de control:

- Es posible controlar gran cantidad de procesos, desde procesos muy simples hasta procesos con dinámicas complejas, de procesos monovariantes a multivariantes.
- Introduce un control anticipativo y, de forma natural, se compensan las perturbaciones medibles.

- Su carácter predictivo hace que sea capaz de compensar los tiempos muertos.
- Cuando se conocen las referencias futuras como en el caso de la robótica, se puede emplear esa información.
- Puede incorporar restricciones tanto a las entradas como a las salidas, pudiendo ser de tipo duras o blandas [7]. Las restricciones fuertes no pueden ser infringidas en cambio las suaves sí. Si se establecen fuertes restricciones a la entrada como a la salida, estas pueden entrar en conflicto provocando que no se obtenga una solución viable para el problema de optimización.

3.2. Autómata M241CE40R

El autómata M241CE40R es un controlador lógico (PLC) que pertenece a la familia Modicon M241 de Schneider Electric, están diseñados para ofrecer un alto rendimiento. La característica más destacada es la comunicación incluyendo 5 puertos integrados (ethernet, CANopen, 2 puertos serie y un puerto USB) [6].



Fig. 6 Autómata M241CE40R

Están equipados con E/S y una interfaz de salidas de control por pulso/dirección como estándar.

La programación del PLC se realiza con un software propio de Schneider, SoMachine, diseñado para poder programar, poner en marcha y realizar tareas de mantenimiento en sus máquinas con un solo software.

Este software está basado en CodeSys, que es un entorno de desarrollo diseñado para solucionar las necesidades de los proyectos de automatización actuales proporcionando soluciones integradas y sencillas, aunque personalizado para los dispositivos de Schneider. Integra los 5 tipos de lenguaje

de programación del estándar industrial más implementado en Europa IEC 61131-3 [4]:

- IL: Lista de instrucciones (Instruction List)
- ST: Texto estructurado (Structures Text)
- LD: Diagrama de contactos (Ladder Diagram)
- FBD: Diagrama de bloques funcionales (Function Block Diagram)
- SFC: Bloques de función secuenciales (Sequential function chart)



Fig. 7 Estructura del M241 en el laboratorio

En la figura 7 se muestra la estructura que se ha construido en el laboratorio que permite el acceso a los módulos cartridge de entradas y salidas analógicas mediante conexiones de tipo banana (amarillos para las entradas y verdes para las salidas, además de uno negro para la conexión a masa). El autómata tiene una etiqueta que indica la dirección IP del autómata, necesaria para conectar el programa en SoMachine al PLC.

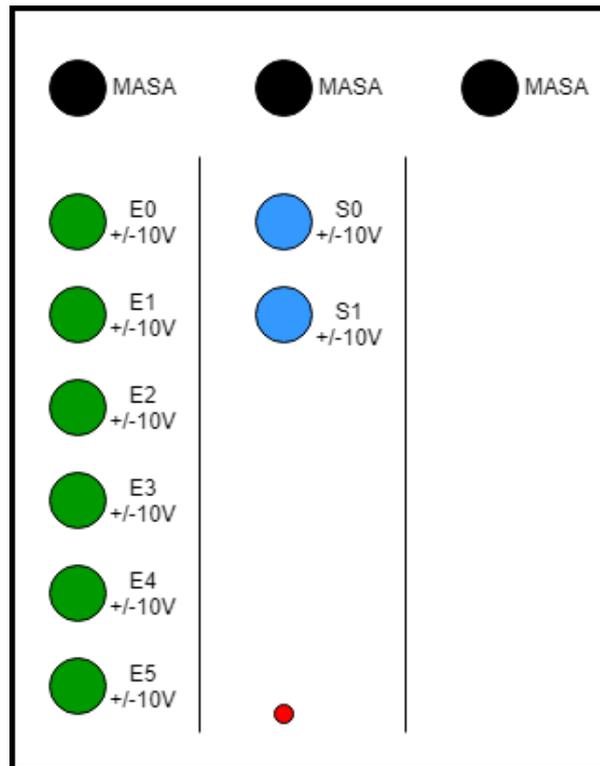


Fig. 9 Caja de conectores banana

3.4. Twin Rotor Mimo System

El Twin Rotor Mimo System (TRMS) es un sistema de laboratorio para experimentos de control que replica las características de un helicóptero, fabricado por Feedback Instruments Limited Company. Está constituido por un eje pivotado en la base permitiendo el libre movimiento tanto en el plano vertical como en el horizontal. En los extremos de la barra hay dos rotores, el principal y el de cola, accionados por dos motores de corriente continua. En la articulación de la base tiene fijado una barra con un contrapeso en el extremo para estabilizar el sistema con un ángulo de inclinación de aproximadamente 17 grados.

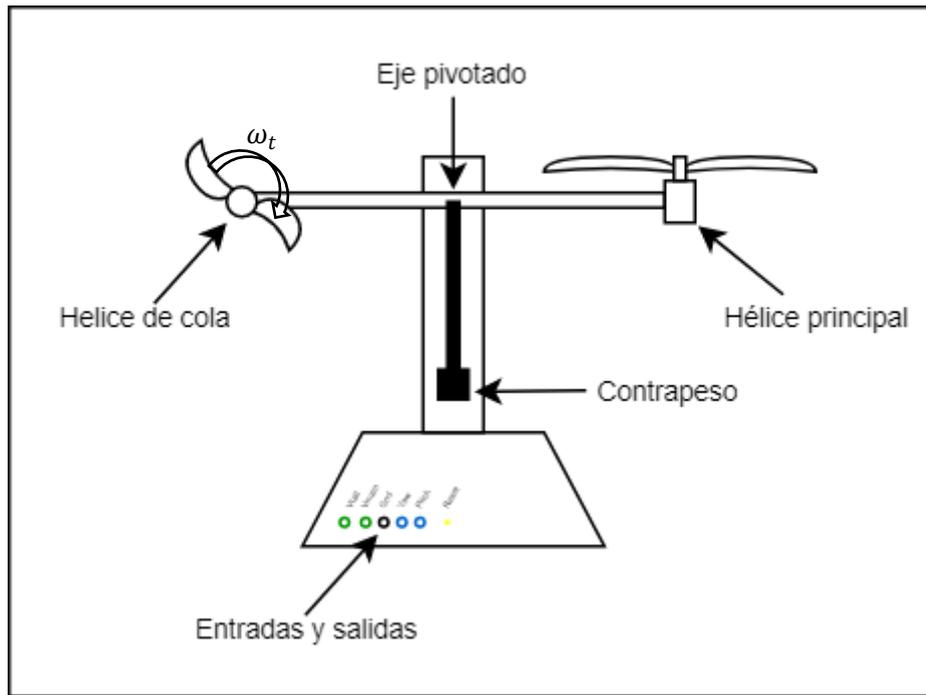


Fig. 10 Modelo académico de un helicóptero

En un helicóptero convencional la fuerza aerodinámica es controlada cambiando el ángulo de ataque. Sin embargo, en el TRMS dicho ángulo es fijo por lo que la fuerza aerodinámica se controla modificando la velocidad y la dirección de los rotores. Esto se produce al variar los voltajes de entrada del sistema, siendo las entradas de control V_{main} (V_m) y V_{tail} (V_t) con un rango de funcionamiento de -2.5 V a 2.5 V. Las salidas proporcionadas por el sistema son el ángulo de elevación (φ , pitch) y el ángulo de rotación (Ψ , yaw) con un rango de trabajo de -10 V a 10 V. El modelo posee un acoplamiento cruzado significativo por lo que cada rotor influye en ambas posiciones de ángulo como se muestra en la figura 11.

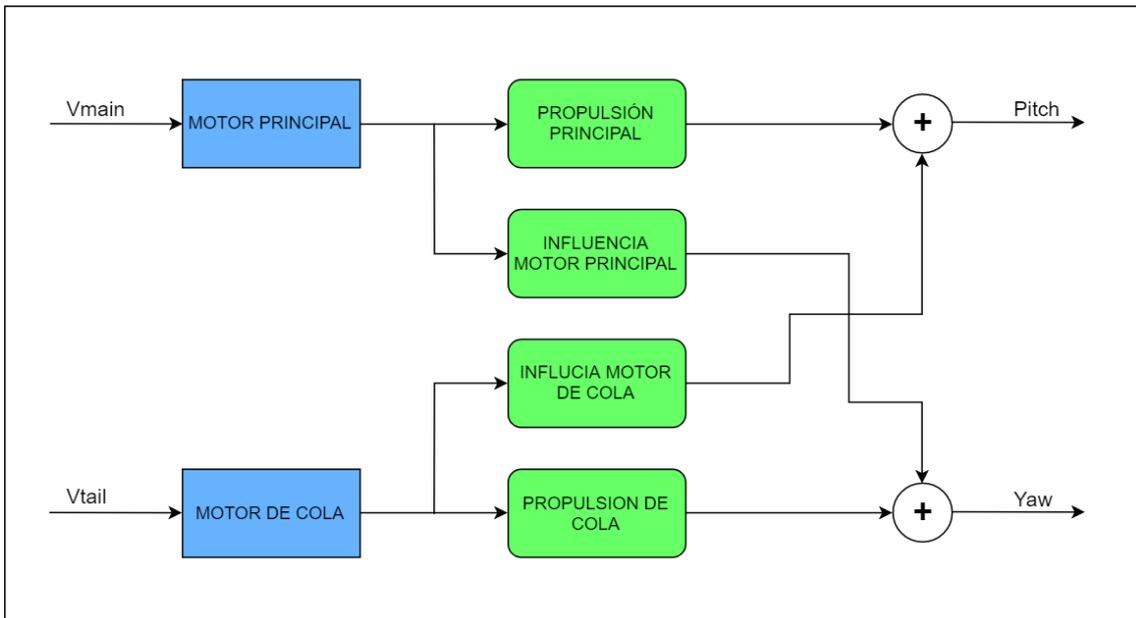


Fig. 11 Esquemático del sistema TRMS

Además el sistema posee un botón de reset, con el que poder situar el punto de equilibrio (0V) de las salidas en la posición deseada. Para el manejo del sistema existe una caja metálica donde hay un botón verde para la activación y un pulsador rojo, parecido a una seta de emergencia para la detención de este.

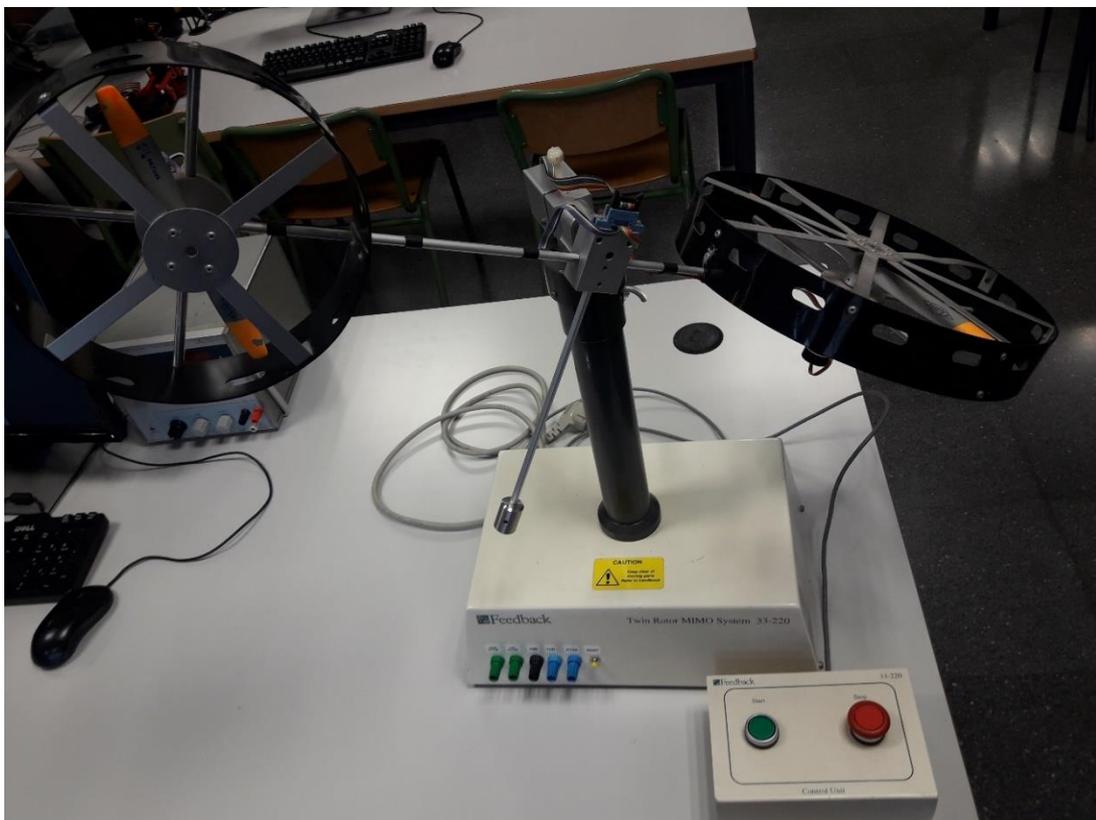


Fig. 12 Twin Rotor MIMO System

Las ecuaciones que describen el modelo físico del proceso se dividen entre las que describen los componentes eléctricos, obteniéndose mediante el análisis de un circuito eléctrico bastante simple (Figura 13), y las que describen la dinámica mecánica, obteniéndose mediante el principio de conservación de los momentos como en la siguiente ecuación:

$$J_{\varphi} \cdot \ddot{\varphi} = \sum_i^N M_i \quad (2)$$

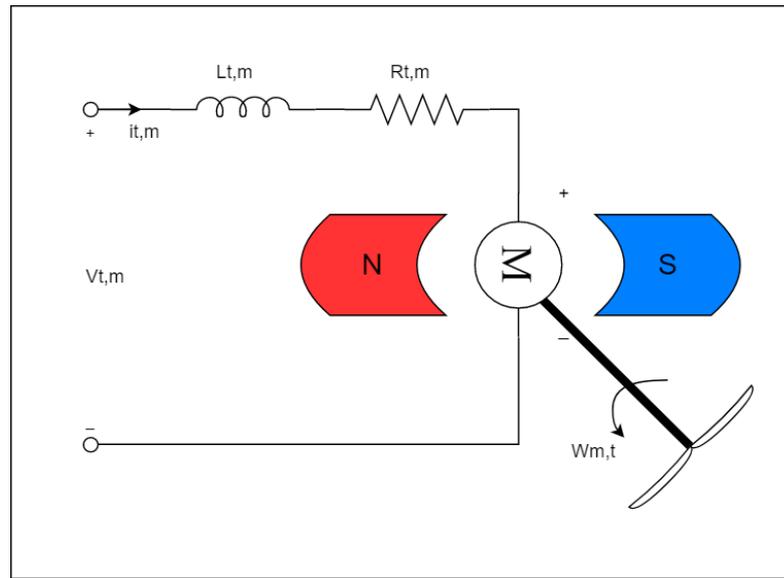


Fig. 13 Esquema eléctrico de los motores de las hélices

- Modelado de la dinámica eléctrica de los motores de corriente continua:

Mediante la aplicación de Ley de Ohm en el circuito eléctrico de la figura 13 se obtiene la relación entre cargas mecánicas y el par electromagnético causado por la corriente del motor.

La carga que se asume está compuesta por el par debido a la resistencia del aire en las palas de la hélice ($M_{air,t/m}$), el par debido a la inercia de la hélice ($M_{J,t/m}$) y la fricción del motor ($M_{FM,t/m}$).

Combinando las anteriores relaciones se obtienen las ecuaciones de la aceleración angular para los dos respectivos motores:

$$\dot{\omega}_m = \frac{1}{J_{p,t}} \cdot \left[\frac{k_{i,t}}{R_{t/m}} \cdot V_t - \left(\frac{k_{i,t}^2}{R_{t/m}} + k_{FM,t/m} \right) \cdot \omega_m - k_{air,t} \cdot |\omega_m| \cdot \omega_m \right] \quad (3)$$

$$\dot{\omega}_t = \frac{1}{J_{p,m}} \cdot \left[\frac{k_{i,m}^t}{R_t} \cdot V_m - \left(\frac{k_{i,m}^2}{R_t} + k_{FM,m}^t \right) \cdot \omega_t - k_{air,m} \cdot |\omega_t| \cdot \omega_t \right] \quad (4)$$

- Modelado de la dinámica mecánica del TRMS:

Para llevar a cabo este modelado se dividen las principales partes rígidas del cuerpo (viga, motores y contrapeso), agrupándolas en tres masas y tres longitudes diferentes, m_m y l_m para el principal, m_t y l_t para la cola y m_g y l_g para el contrapeso.

El movimiento de inclinación (φ) alrededor del eje vertical es inducido por varios pares de torsión:

- $M_{P,\varphi}$, par de propulsión proporcionado por el rotor principal, este par es el principal para controlar la inclinación.
- $M_{Cr,\varphi}$, influencia de la dinámica del motor de cola en la inclinación.
- M_g , par producido por la fuerza gravitatoria.
- M_z , par producido por las fuerzas centrífugas, esta fuerza dirigida hacia afuera surge debido a la rotación del cuerpo.
- $M_{F,\varphi}$, par producido por las fricciones.

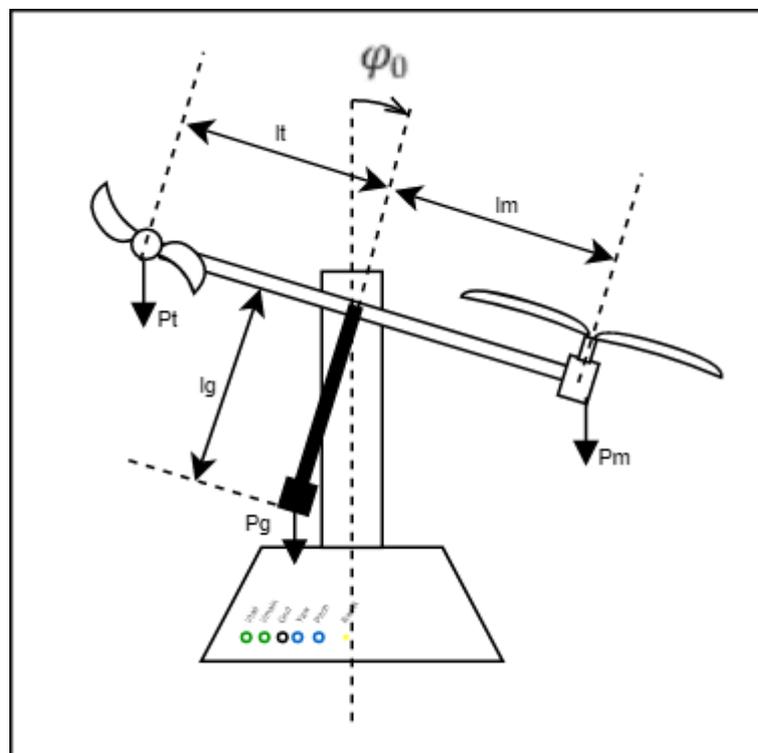


Fig. 14 Plano Horizontal del TRMS

Combinando todos los momentos descritos de acuerdo con la ecuación (2), junto con la inercia J_φ , cuyo valor es una constante se obtiene la siguiente ecuación:

$$\ddot{\varphi} = \frac{1}{J_\varphi} \left[\begin{aligned} & \underbrace{[\cos(-\varphi) \cdot (m_t l_t - m_m l_m) + \text{sen}(-\varphi) \cdot m_g l_g] \cdot g}_{M_g} + \\ & + \underbrace{k_\zeta \cdot \dot{\Psi}^2 \cdot \cos(-\varphi) \text{sen}(-\varphi) \cdot (m_t l_t^2 - m_m l_m^2 - m_g l_g^2)}_{M_z} + \underbrace{k_{p,\varphi} \cdot |\omega_m| \cdot \omega_m \cdot l_m}_{M_{p,\varphi}} + \\ & + \underbrace{\frac{k_{i,t}}{R \frac{t}{m}} \cdot k_{p,t} \cdot V_t - \left(\frac{k_{i,t}^2}{R \frac{t}{m}} + k_{FM,\frac{t}{m}} \right) \cdot k_{p,t} \cdot \omega_t - (k_{p,t} k_{air,t} - \tilde{k}_{air,t}) \cdot |\omega_t| \cdot \omega_t}_{M_{Cr,\varphi}} + \\ & + \underbrace{k_{Fc,\varphi} \cdot \dot{\varphi} - k_{Fs,\varphi} \cdot \text{sgn}(\dot{\varphi})}_{M_{F,\varphi}} \end{aligned} \right] \quad (5)$$

El movimiento de rotación (Ψ) alrededor del eje horizontal es similar al de inclinación excepto que en este caso las fuerzas gravitatorias y centrífugas no influyen, además existe un par generado por el cable de los sensores, por tanto este movimiento es inducido por los siguientes pares de torsión:

- $M_{P,\psi}$, par de propulsión proporcionado por el rotor de cola, este par es el principal para controlar la rotación.
- $M_{Cr,\psi}$, influencia de la dinámica del motor principal en la rotación.
- M_c , fuerza generada por el cable de conexiones.
- $M_{F,\psi}$, par producido por las fricciones.

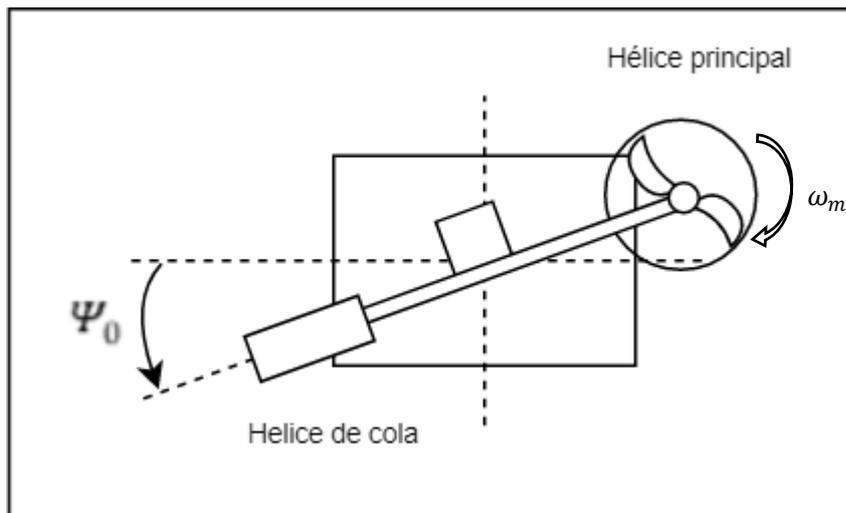


Fig. 15 Plano vertical del TRMS

Combinando todos los momentos descritos anteriormente de acuerdo con la ecuación (2), junto con la inercia J_ψ , cuyo valor varía dependiendo del ángulo de inclinación, φ , se obtiene la siguiente ecuación:

$$\ddot{\Psi} = \frac{1}{k_{J,\psi} \cdot J_\psi} + \left[\underbrace{\frac{k_{p,\psi} \cdot |\omega_t| \cdot \omega_t \cdot \cos(\varphi) l_t}{M_{p,\psi}} - \frac{k_c \cdot (\Psi - \Psi_0)}{M_c} - \frac{k_{Fc,\psi} \cdot \Psi - k_{Fs,\psi} \cdot \text{sgn}(\Psi)}{M_{F,\psi}}}_{M_{c,\psi}} \right] + \underbrace{\cos(\varphi) \cdot \left[\frac{k_{i,\frac{t}{m}}}{R_{\frac{t}{m}}} \cdot k_{p,m} \cdot V_m - \left(\frac{k_{i,\frac{t}{m}}^2}{R_{\frac{t}{m}}} + k_{FM,\frac{t}{m}} \right) \cdot k_{p,m} \cdot \omega_m - (k_{p,m} k_{air,m} - \tilde{k}_{air,m}) \cdot |\omega_m| \cdot \omega_m \right]}_{M_{Cr,\psi}} + \underbrace{\text{sen}(\varphi) \cdot \tilde{k}_{p,m} \cdot |\omega_m| \cdot \omega_m \cdot d}_{M_{Cr,\psi}} \right] \quad (6)$$

Para el diseño del controlador es necesario introducir un modelo lineal, por lo que hay que linealizar el sistema de ecuaciones diferenciales no lineales descrito por las ecuaciones 4 y 5, para ello se va a utilizar la representación en espacio de estados del proceso, mostraba mediante las ecuaciones (7) y (8), diseñada por Tim Zieger [1] en la que se linealiza el modelo no lineal utilizando la expansión de Taylor según un punto de trabajo.

El espacio de estados es una forma de representación de sistemas multivariados constituido por ecuaciones diferenciales matriciales en el que se relacionan las entradas, las salidas y las variables de estado. La ecuación general de un sistema continuo e invariante en el tiempo es la siguiente:

$$\begin{cases} \dot{x}(t) = A \cdot x(t) + B \cdot u(t) \\ y(t) = C \cdot x(t) + D \cdot u(t) \end{cases}$$

Donde:

- $x(t)$: variables de estado, es el conjunto mínimo de variables necesarias para representar la dinámica del sistema por completo.
- $\dot{x}(t)$: $\frac{dx(t)}{dt}$
- $y(t)$: salidas del sistema.
- $u(t)$: acciones de control.
- A : matriz de estados, expresa la forma en que varía el sistema si no hay entradas, evolución dinámica.
- B : matriz de entrada, expresa como actúan las entradas a la dinámica del sistema (variables de estado).

- C : matriz de salida, expresa como se reflejan los estados en las salidas.
- D : matriz de acoplamiento directo.

$$\underbrace{\begin{pmatrix} \dot{\varphi} \\ \ddot{\varphi} \\ \dot{\Psi} \\ \ddot{\Psi} \\ \dot{\omega}_m \\ \dot{\omega}_t \end{pmatrix}}_{\dot{x}(t)} = \underbrace{\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -5,709 & -0,046 & 0 & 0 & 0,039 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0,072 & 0 & -0,12 & -0,271 & 0,001 & 0,011 \\ 0 & 0 & 0 & 0 & -0,409 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0,83 \end{pmatrix}}_A \cdot \underbrace{\begin{pmatrix} \varphi \\ \dot{\varphi} \\ \Psi \\ \dot{\Psi} \\ \omega_m \\ \omega_t \end{pmatrix}}_{x(t)} + \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & 0,069 \\ 0 & 0 \\ 0,262 & 0 \\ 14,68 & 0 \\ 0 & 77,932 \end{pmatrix}}_B \cdot \underbrace{\begin{pmatrix} V_m \\ V_t \end{pmatrix}}_{u(t)} \quad (7)$$

$$\underbrace{\begin{pmatrix} \varphi \\ \dot{\varphi} \\ \Psi \\ \dot{\Psi} \\ \omega_m \\ \omega_t \end{pmatrix}}_{y(t)} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}}_C \cdot \underbrace{\begin{pmatrix} \varphi \\ \dot{\varphi} \\ \Psi \\ \dot{\Psi} \\ \omega_m \\ \omega_t \end{pmatrix}}_{x(t)} + \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}}_D \cdot \underbrace{\begin{pmatrix} V_m \\ V_t \end{pmatrix}}_{u(t)} \quad (8)$$

4. Estudio de Soluciones Alternativas

4.1. C-Integration de CodeSys

Una solución que se barajo era crear un proyecto en CodeSys y mediante el plug-in C-Integration importar el código c del MPC diseñado para controlar el modelo en espacio de estados del TRMS. De esta manera se crea la interfaz IEC, es decir, se convierte las funciones que contiene el archivo c en funciones de texto estructurado, siendo fundamental para poder usar el PLC. A continuación se exportaría el proyecto con las POU's necesarias para implementar todas las funciones a SoMachine.

Para comprobar que esta solución es factible se empieza desarrollando un función simple en código c, como es una suma de dos variables. Además del archivo c también se debe añadir el archivo .h, de cabecera, para que el programa pueda acceder a la función que se identifica en este archivo y pueda llevar a cabo la creación de la función en texto estructurado. Posteriormente se intenta realizar el proceso de exportación a SoMachine.

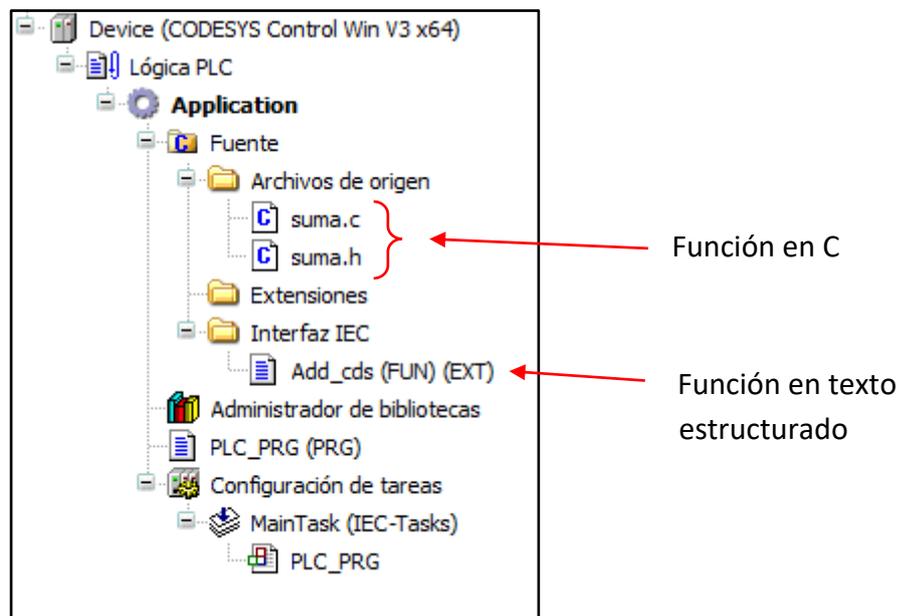


Fig. 16 Funciones CodeSys con C-Integration

Para poder crear la interfaz IEC con C-Integration es necesario tener instalado en el ordenador de trabajo las versiones adecuadas de Visual Studio y la .NET framework de Microsoft, además de proporcionar la ruta de la ubicación de los programas como se muestra en la figura 17.

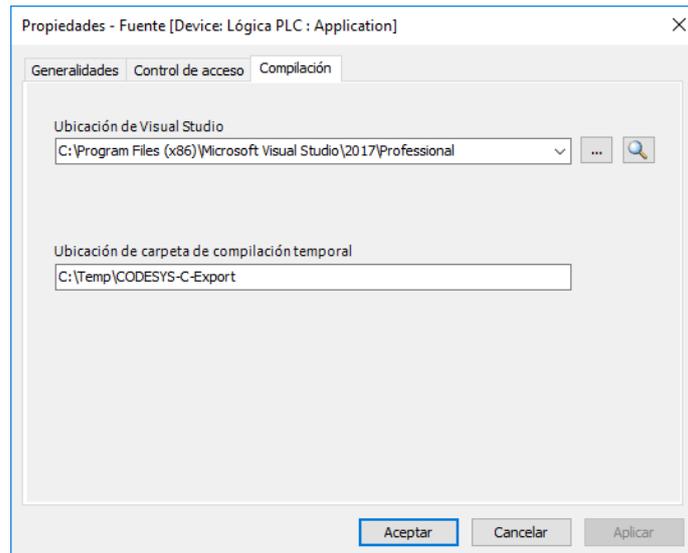


Fig. 17 Configuración Device CodeSys

Sin embargo, cuando se compila el proyecto se produce un error de compilación (figura 18) y tras investigar se observa que para utilizar por completo la integración c de CodeSys y de esta forma poder compilar el proyecto de forma correcta sin errores, hay que ser desarrollador software o tener una licencia especial, por lo que se desecha esta solución.

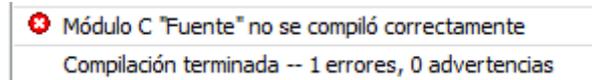


Fig. 18 Error proporcionado tras compilar en CodeSys

4.2. MPC Controller de Matlab

Esta alternativa se basa en el diseño del controlador mediante el Model Predictive Control Toolbox de Matlab, que mediante la introducción de los parámetros de diseño (horizonte de control, horizonte de predicción y tiempo de muestreo), la planta del sistema en espacio de estados y las restricciones deseadas para las variables manipulables y las salidas, se obtiene el controlador. Que se implementa en un esquema de Simulink junto a la planta del TRMS y después de simular el esquema, es posible la generación de texto estructurado del bloque del controlador mediante PLC Coder de Matlab. Generando un archivo de exportación que se puede usar en SoMachine.

5. Solución Adoptada

5.1. Descripción General de la Solución

La solución finalmente adoptada consiste en diseñar el controlador MPC en Matlab y finalmente exportarlo a SoMachine. Para llevar a cabo este proceso es necesario tener instaladas diferentes plug-in de Matlab:

- Simulink PLC Coder: genera texto estructurado desde bloques de Simulink o funciones de Matlab. Se generan en formato PLCOpen XML y otros formatos de archivo compatibles con los entornos de desarrollo.
- Model Predictive Control Toolbox: proporciona funciones, una aplicación y bloques de Simulink para el diseño y la simulación de controladores MPC.
- Data Acquisition Toolbox: proporciona apps y funciones para configurar hardware de adquisición de datos, leer y escribir datos en canales DAQ analógicos y digitales.

5.2. Verificación del funcionamiento de la solución adoptada

Para verificar que esta solución es válida, se utiliza el ejemplo de Matlab “Simulation and Structured Text Generation Using PLC Coder” [3] modificando los parámetros del controlador para los límites físicos existentes.

Dicho ejemplo se estructura de la siguiente manera:

1. Comprobar que están instalados Simulink y PLC Coder

```
if ~MPCchecktoolboxinstalled('simulink')
    disp('Simulink(R) is required to run this example.')
    return
end
if ~MPCchecktoolboxinstalled('PLCcoder')
    disp('Simulink(R) PLC Coder (TM) is required to run this
example. ');
    return
end
```

2. Se cambia el directorio actual a uno temporal antes de iniciar la simulación y la generación de código, para tener permiso de escritura para generar los archivos relevantes y el ejecutable.

```

cwd = pwd;
tmpdir = tempname;
mkdir(tmpdir);
cd(tmpdir);

```

3. Se define una planta SISO

```

plant = ss (tf ([3 1], [1 0,6 1]));

```

4. Se definen los parámetros del controlador

```

Ts = 0.1;    %Sampling time
p = 10;     %Prediction horizon
m = 2;     %Control horizon
Weights = struct('MV',0,'MVRate',0.01,'OV',1); % Weights
MV = struct('Min',-10,'Max',10,'RateMin',-
100,'RateMax',100); % Input constraints
OV = struct('Min',-2,'Max',2); % Output constraints
MPCobj = MPC(plant,Ts,p,m,Weights,MV,OV);

```

5. Simulación

```

mdl = 'MPC_PLCDemo';
open_system(mdl)
close_system([mdl '/Control System/MPC Controller'])
open_system ([mdl '/Outputs//References'])
open_system ([mdl '/Inputs'])
sim(mdl)

```

El esquema Simulink siguiente “MPC_PLCDemo” muestra los bloques necesarios para realizar la simulación. En el subsistema llamado “Control System” se introduce el bloque del controlador MPC, al que se le introducen la referencia y las salidas del sistema realimentadas. La referencia se introduce en el MPC mediante un generador de señales, siendo esta una señal senoidal de amplitud 1 y frecuencia 0,4Hz. Además se incluyen dos bloques “scope” para representar la acción de control que se le introduce al modelo y la salida del sistema frente a la referencia.

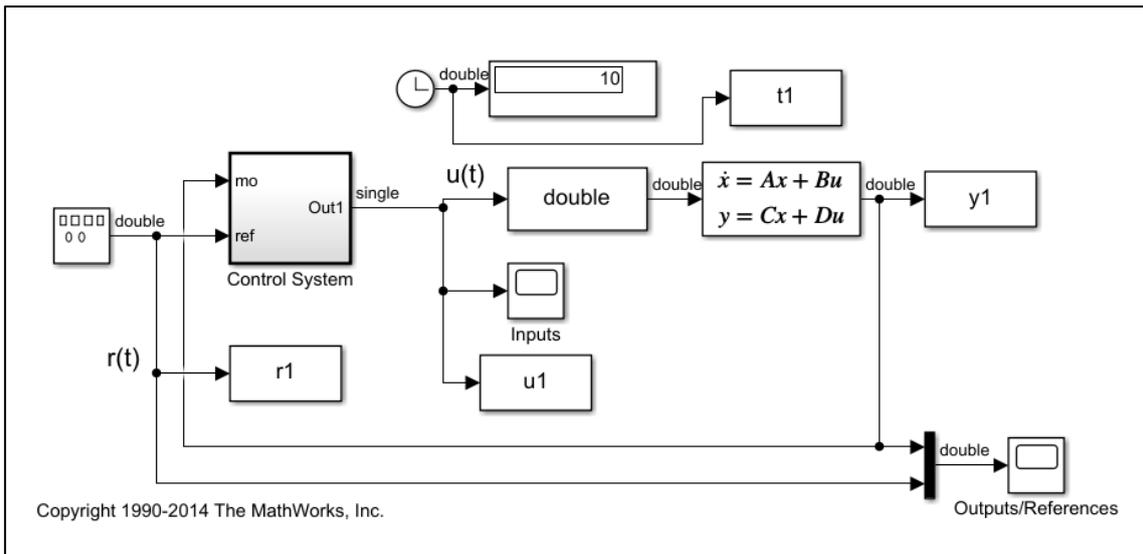


Fig. 19 Esquema Simulink Ejemplo

Representando la salida del sistema junto a la referencia se obtiene la siguiente gráfica:

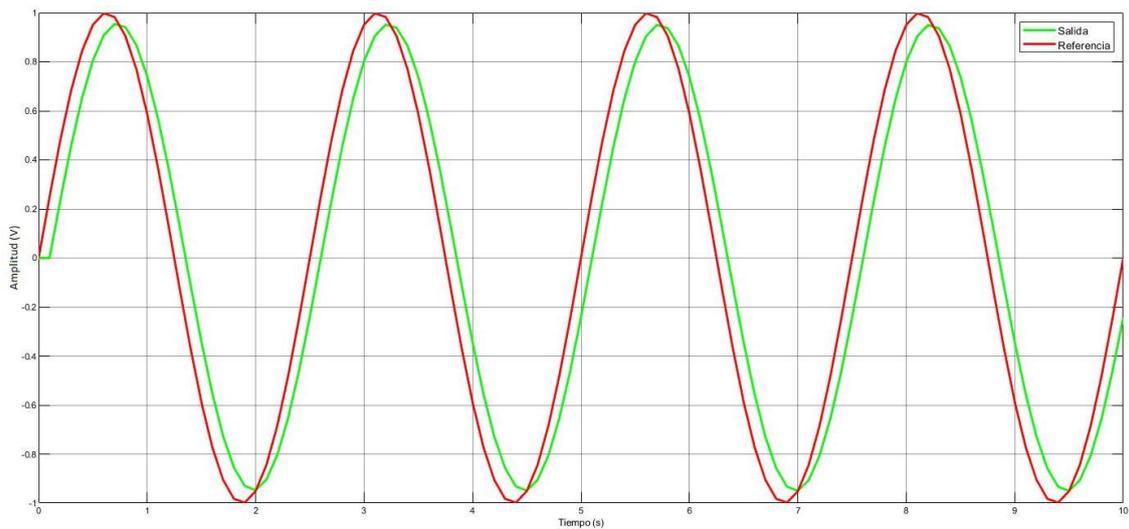


Fig. 20 Respuesta del sistema de ejemplo

En la respuesta del sistema se observa como la salida sigue la trayectoria senoidal de referencia con un pequeño retardo de 100ms y una pequeña disminución en la amplitud de esta.

En la siguiente gráfica se muestra la acción de control, generada por el regulador, que se introduce al modelo para alcanzar la referencia. Como se observa en la figura 21, la señal de control esta escalonada, esto es debido a que el tiempo de muestreo del controlador es de 0.1s, por lo que se puede observar a simple vista cada periodo de muestreo, con un valor más pequeño del mismo se obtendría

una respuesta mejor, sin embargo para comprobar el correcto funcionamiento de la solución no es necesario cambiar dicho valor.

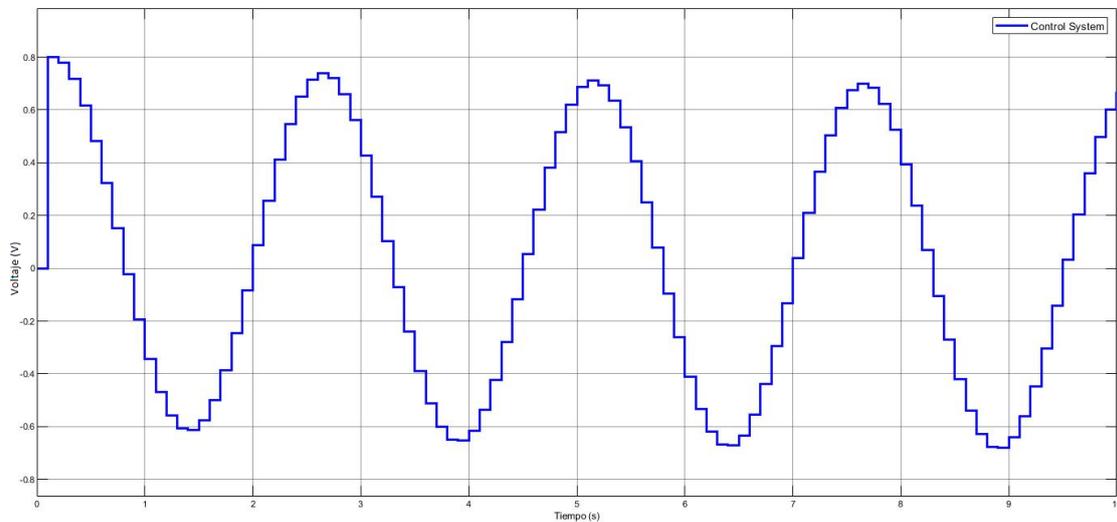


Fig. 21 Acción de control del controlador de ejemplo

5.2.1. Generación de texto estructurado con PLC Coder de Matlab

Para poder generar texto estructurado para el bloque de control donde se ubica el MPC se deben realizar los siguientes pasos:

- Configurar el bloque MPC para que el tipo de salida que proporciona sea “single” (Figura 22).

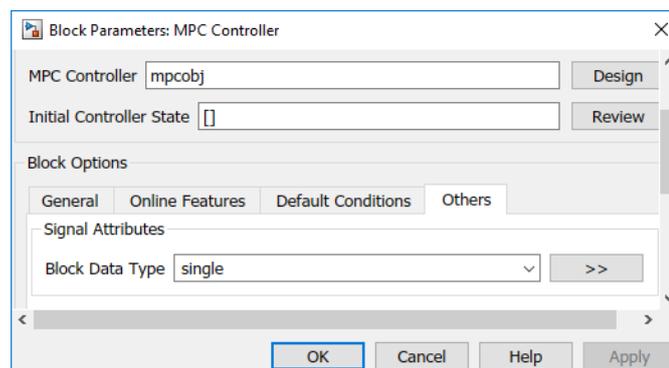


Fig. 22 Configuración del bloque MPC

- Seleccionar “Treat as atomic unit” dentro de la configuración de parámetros del subsistema de control (Figura 23).

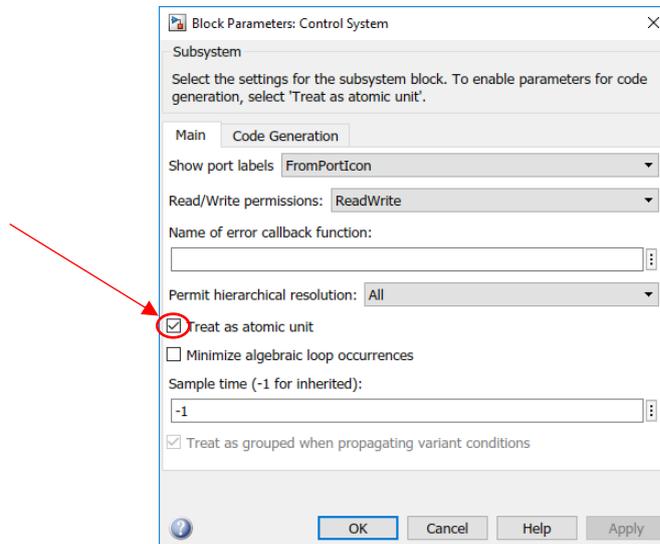


Fig. 23 Configuración del subsistema de control

- Elegir dentro de las opciones generales, en el apartado de PLC Code Generation, como target IDE 3S: CoDeSys 3.5. (Figura 24).

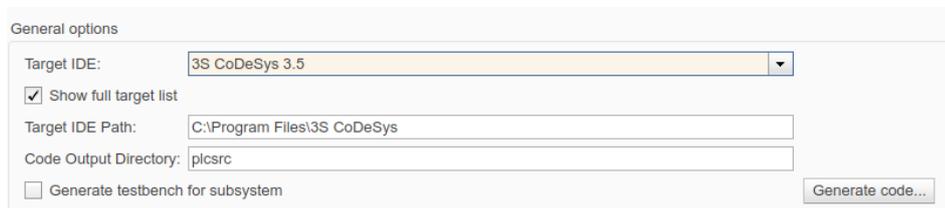


Fig. 24 Configuración PLC Coder

Mediante el comando “PLCgeneratecode”, indicando el subsistema del cual se quiere generar código “Control System”, se genera un archivo de exportación de texto estructurado. Para guardar el archivo le ponemos la extensión “.xml” para que pueda ser importado correctamente, ya que si se guarda en otro formato el programa no lo detecta.

5.2.2. Generación de las POU’s con CodeSys

Para poder importar el texto estructurado, generado anteriormente, en SoMachine hay que hacer un paso previo ya que el programa no encuentra funciones para las que crear un POU dentro del archivo de importación. En

cambio, en CodeSys sí que es admitido, ya que mediante la instalación del plugin “Simulink PLC Coder Import”, que es específico para importar el código generado con PLC Coder. De esta manera se generan automáticamente las siguientes POU’s:

- Abs_o (FB)
- b_abs (FB)
- Control (FB)
- DropConstraint (FB)
- KWIKfactor (FB)
- norm (FUN)
- qpkwik (FB)
- qr (FB)
- rdivide_helper (FUN)
- Unconstrained (FB)
- xgemv (FB)
- xgerc (FB)
- xnrm2 (FUN)

Finalmente, se exporta el proyecto seleccionando todos los POU’s existentes, además de las variables y constantes globales. Se debe seleccionar la versión CoDeSys V3.3. Es necesario guardar el archivo generado con la extensión “.export”.

5.2.3. Creación del proyecto en SoMachine

Lo primero que se debe realizar al crear el proyecto es seleccionar el controlador lógico que se va a usar en: configuración → Gestionar dispositivos, en este caso es el M241 (TM241CE40R). A continuación, se deben añadir al dispositivo los módulos de entrada (TMC4AI2) y salida (TMC4AQ2) analógicas de tipo cartridge, como se muestra en la figura 25. Además hay que activar las entradas y salidas que se van a utilizar, eligiendo el rango de trabajo, en este caso se elige el de 0 – 10V (0 – 10000).

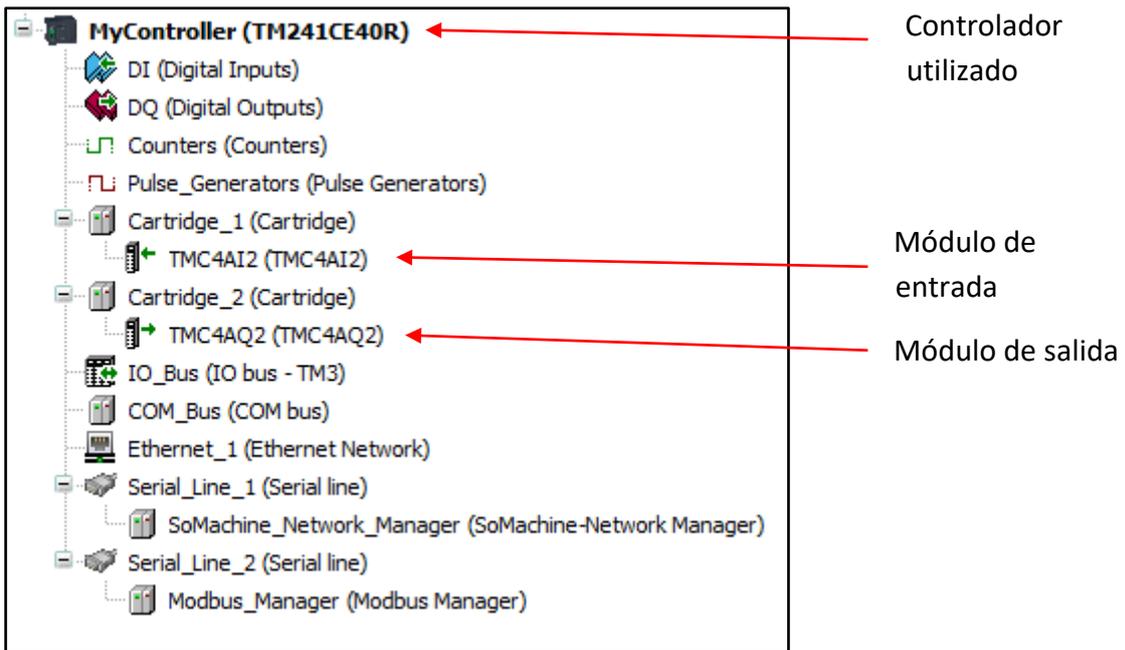


Fig. 25 Dispositivos de SoMachine

A continuación, se importan las POU's generadas anteriormente en CodeSys, y se agrupan en la carpeta "global". En el programa principal, de tipo diagrama de contactos, se incluyen los siguientes bloques:

- Los bloques de conversión, para transformar las señales de entrada de tipo int a tipo real (Figura 26) y las señales de salida de tipo real a tipo int.

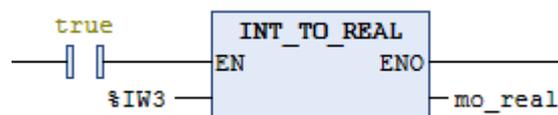


Fig. 26 Bloque conversión INT a REAL

- Los bloques "FB_Scaling" incluidos en la biblioteca "Toolbox 2.2.0.0". Para escalar las señales de entrada de 0 – 10000 a -10 – 10 y las de salida de -10 – 10 a 0 – 10000 (Figura 27). Para poder hacer uso de estos bloques hay que añadir la biblioteca al proyecto, ya que no se encuentra añadida por defecto.

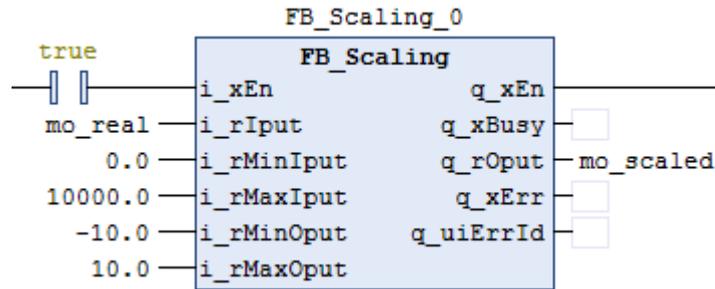


Fig. 27 Bloque de escalado de la señal

- El bloque de control, que incluye el POU “Control (FB)”, al que se introducen la salida del sistema (mo_scaled), el estado del bloque (state: 0 modo inicialización y 1 modo run) y la referencia (RefVector), y proporciona la acción de control (u_mpc), que se envía al sistema (Figura 28).

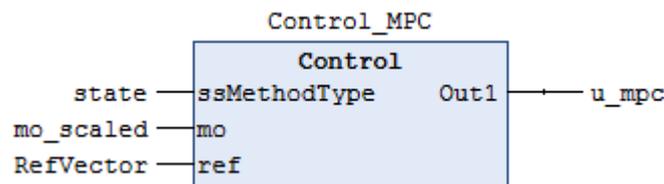


Fig. 28 Bloque principal de control

Además, para que el programa principal se ejecute de forma cíclica debe añadirse a la tareas MAST e introducir el tiempo de ciclo adecuado para que este sincronizado con el regulador diseñado, en este caso 0,1s.

5.2.4. Comunicación con NI USB-6001

Después de conectar la tarjeta de adquisición a un terminal USB del ordenador, se debe iniciar el programa NI Device Monitor para que el ordenador detecte la tarjeta de adquisición, y poder utilizarla con Simulink. Este programa proporciona el nombre del canal físico al que se ha conectado la tarjeta “Dev7”, que deberá concordar con el del bloque usado en Simulink.

Las conexiones que se deben realizar entre el PLC y la tarjeta de adquisición son las siguientes:

- El terminal de masa del PLC con cualquier terminal de masa de la tarjeta.
- La salida 1 del PLC con la entrada E0 de la tarjeta.
- La entrada 1 del PLC con la salida S0 de la tarjeta (Salida del modelo).

- La entrada 2 del PLC con la salida S1 de la tarjeta (Señal senoidal de referencia).

5.2.5. Adquisición de datos en Simulink

Se utiliza Data Acquisition Toolbox de Matlab para poder acceder a la tarjeta de adquisición. Para ello se eligen los bloques de Analog Input (Single Sample) y Analog Output (Simple Sample), que reciben y envían una sola muestra cada intervalo de tiempo, de forma síncrona desde y hacia el dispositivo durante todo el tiempo de ejecución que dure la simulación.

Como el M241 solo trabaja de 0 a 10 V, se deben escalar las señales para adecuarlas a este rango, tanto las que se van a recibir como las que se van a enviar, para ello se utilizan los bloques de función, donde se introducen las siguientes ecuaciones lineales:

- (Entradas) $\frac{x}{10} = \frac{y+10}{10+10} \rightarrow 20x = 10y + 100 \rightarrow y = 2x - 10$ (9)
- (Salidas) $\frac{x+10}{10+10} = \frac{y}{10} \rightarrow 20y = 10x + 100 \rightarrow y = 0,5x + 5$ (10)

También, se incluye para las señal de entrada un limitador del rate para que no se sobrepase los valores del rate con los que se ha diseñado el controlador (-100 y 100). Y para la señal de salida un saturador de la señal a -10 y 10, para que no se sobrepasen los valores límite que se pueden enviar por la tarjeta de adquisición, lo que provocaría la detención de la simulación.

En el siguiente esquema de Simulink (figura 29) se muestran todos los bloques nombrados anteriormente que son necesarios para comunicar de forma correcta el ordenador con el PLC.

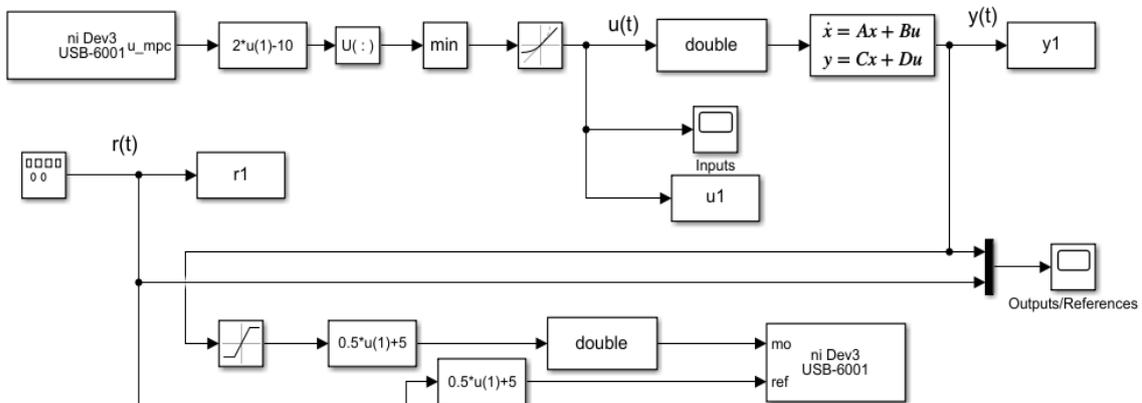


Fig. 29 Esquema Simulink comunicación con PLC

La acción de control que se introduce al sistema, generadas en el PLC, se muestra en la siguiente gráfica (figura 30):

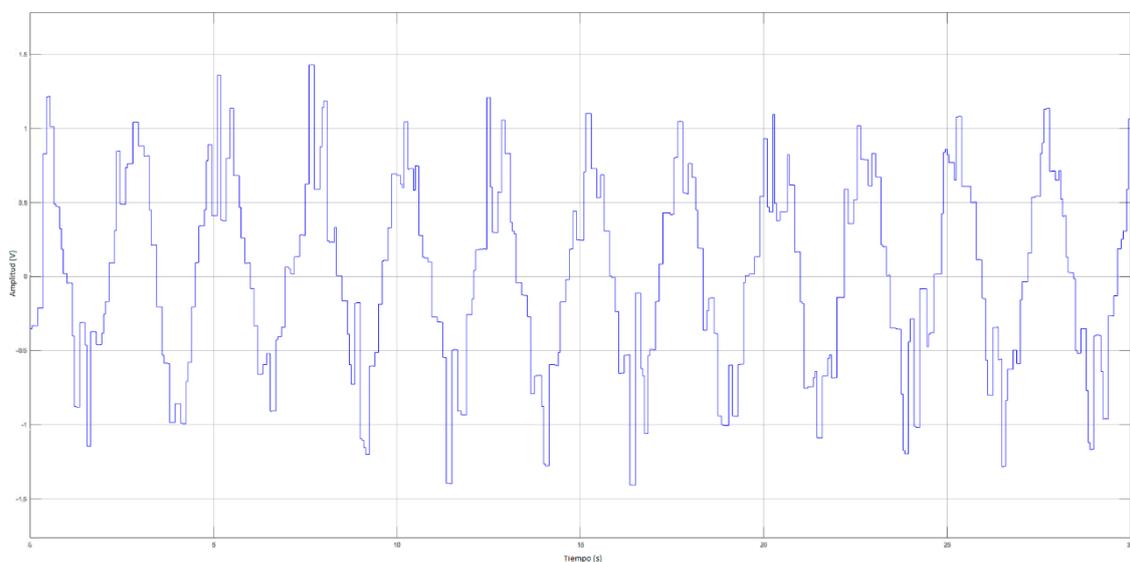


Fig. 30 Acción de control recibida del PLC

Se observa en la figura 31, como la respuesta del sistema sigue la señal senoidal de referencia, sin embargo, tiene una amplitud levemente mayor y con un retardo de aproximadamente 300 ms. Por lo que se confirma que la solución adoptada es válida para el afrontar los objetivos del proyecto.

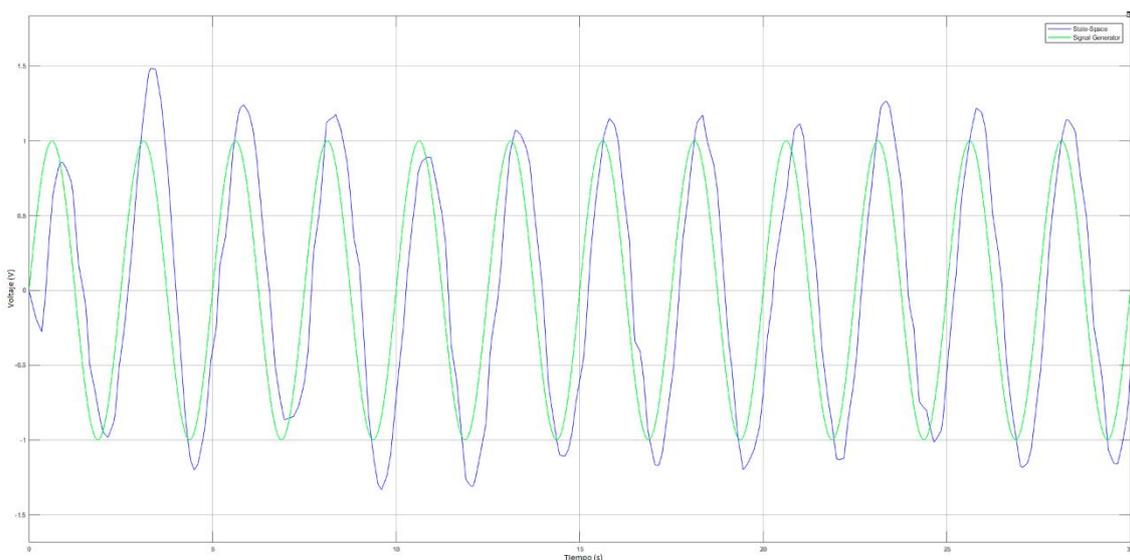


Fig. 31 Gráfica de la respuesta del sistema y la referencia

5.3. Implementación de la solución para TRMS

Para la implementación de la solución se realiza el proceso descrito en el diagrama de flujo de la figura 32, en el que se muestran todos los pasos que son necesarios para la correcta implementación del MPC en el PLC real. En este proceso se incluyen dos tipos de experimentos que determinarán la validez del controlador diseñado:

- El primer experimento es la realización de una simulación completa en Simulink, es decir, el regulador y el modelo lineal están implementados con bloques de Simulink, como se muestra en la figura 33.
- El segundo experimento se realiza cuando la simulación en Simulink solo consta del modelo lineal, el regulador se ejecuta en el PLC real. Por tanto, se debe realizar una comunicación entre ambos para enviar las entradas de control al modelo y las salidas de este al controlador, como se muestra en la figura 41.

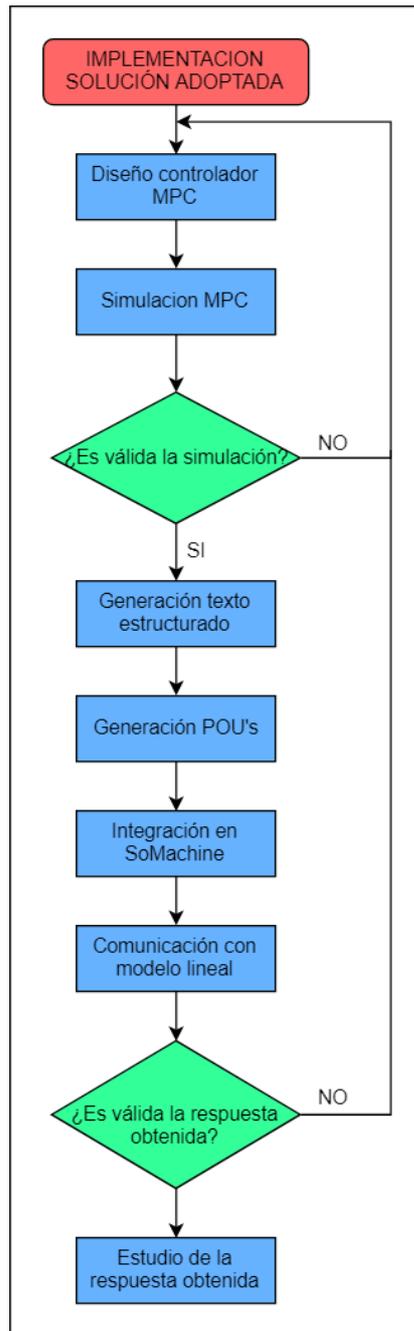


Fig. 32 Diagrama de flujo correspondiente a la implementación del MPC en el PLC real

5.3.1. Identificación de los parámetros de funcionamiento del TRMS

Para poder diseñar el controlador MPC de forma correcta hay que observar cuales son los límites de las entradas y de las salidas que posee el sistema. Para ello se conectan las entradas (Vmain y Vtail) a las salidas de la tarjeta de adquisición. Primero de todo se detecta que el terminal preparado para la entrada de Vmain es en realidad el terminal que actúa sobre el motor de cola y

viceversa, esta situación puede llevar a confusión a la hora de realizar las conexiones y por tanto las medidas tomadas no serán correctas.

Según el modelo de motor que se presenta en la tesis de Tim [1] el rango de funcionamiento de los dos motores es de -2.5 V a 2.5V, sin embargo se observa que proporcionando 0 V los motores están funcionando a alta velocidad y conforme se aumenta la tensión la velocidad disminuye, hasta llegar a 2.5 V que es a la tensión que los motores se detienen completamente, si se continúa aumentando la tensión se observa que el motor se vuelve a activar pero en sentido contrario. Por lo tanto el nuevo rango de funcionamiento con el que se controlara el sistema es de 0V a 5V. Debido a esta modificación, se puede usar todo el rango de trabajo de los motores, ya que el autómatas solo proporciona señales de 0V a 10V, lo que significa que sin esta modificación solo se podría utilizar el motor para desplazar la hélice en un sentido.

Las salidas se conectan a la tarjeta de adquisición para poder visualizar en el ordenador que valores proporciona el sistema para los ángulos de inclinación (Pitch) y el de rotación (Yaw). Se observa que la señal medida de inclinación tiene un rango de aproximadamente 16 V que puede oscilar entre -10V a 10V dependiendo de en qué posición se sitúe el origen, ya que no tiene el recorrido completo de 360°. La señal de rotación medida sí que abarca los 360° de recorrido por lo que el rango es de 20V, entre -10V y 10V.

5.3.2. Diseño controlador MPC

Antes de empezar con el diseño del regulador es necesario comprobar que el modelo del proceso es totalmente controlable, esto se realiza con el siguiente comando de Matlab “rank(ctrb(plant))”, siendo plant el modelo lineal del proceso descrito en el apartado 2.3, que proporciona el número de estados controlables de la planta siendo dicho valor 6 por lo que el sistema es completamente controlable. Además se estudia el número de estados que son controlables con cada entrada, mediante el comando “rank (ctrb (A, B(:,1)))”, siendo 1 para la primera entrada y 2 para la segunda, obteniéndose que con cada entrada es posible controlar 5 estados, por lo que 4 estados se pueden controlar con las dos entradas. Sin embargo, solo se tiene acceso a dos estados en el modelo, por lo que sería necesario el diseño de un observador del sistema para poder tener datos de todos los estados.

Partiendo del controlador que fue diseñado por Tim [1] se introducen los siguientes parámetros:

- Horizonte de predicción (p) = 28
- Horizonte de control (m) = 5
- Tiempo de muestreo (T_s) = 0.05 s

Además para adecuarse a las limitaciones físicas del modelo se introducen las siguientes restricciones en el controlador:

- Para las variables manipulables (MV), es decir, las acciones de control, se establecen unas restricciones duras ya que no se debe superar los límites introduciendo más tensión de la adecuada en el sistema. Por tanto, los límites de las acciones de control serán -2.5 y 2.5.
- Para las variables de salida (OV), se establecen unas restricciones blandas ya que podrían entrar en conflicto si se diseñan las todas las restricciones fuertes.

5.3.3. Simulación del modelo linealizado

Se implementa el siguiente esquema en Simulink (figura 34), donde el subsistema de control debe tener todos los terminales necesarios para las salidas realimentadas, las referencias y las acciones de control, para que se pueda generar el texto estructurado de forma correcta con todas las variables del sistema.

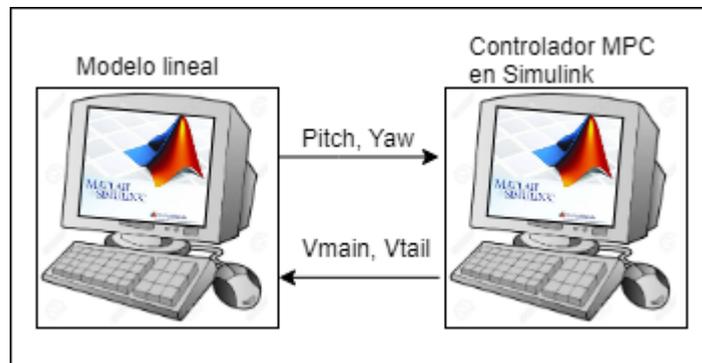


Fig. 33 Experimento de simulación 1

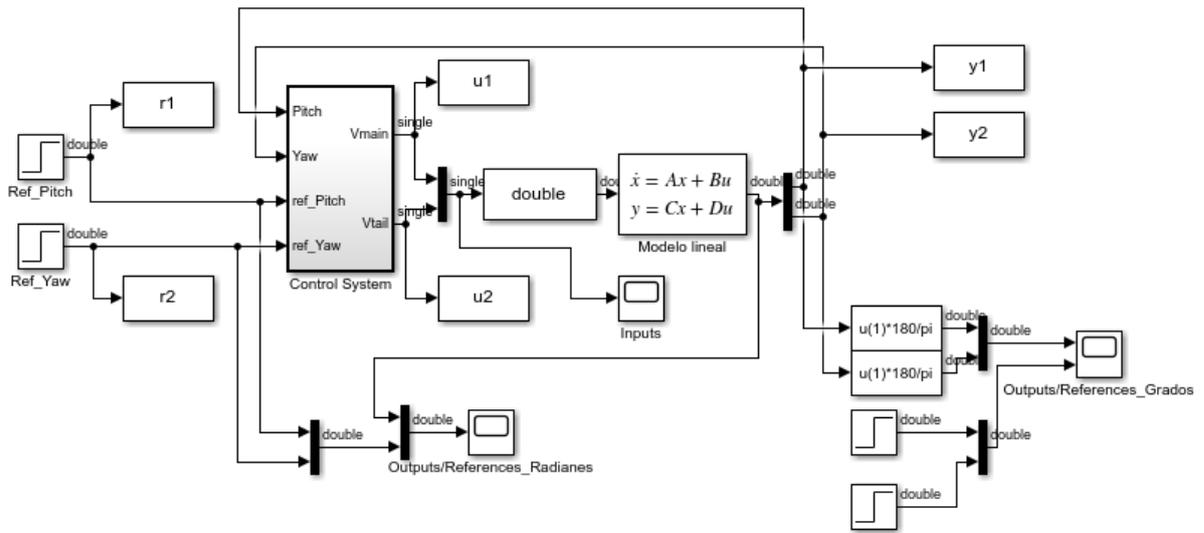


Fig. 34 Esquema Simulink con el controlador MPC

Las salidas que proporciona el modelo lineal están en radianes, pero para poder entender y observar la respuesta obtenida del sistema con mayor claridad, se transforman las salidas de radianes a grados, y se representan frente a dos señales de tipo escalón de valor final el de la referencia a alcanzar en grados.

Para poder observar con más claridad la influencia de una salida sobre la otra, se repiten las simulaciones cambiando las referencias. Primero se mantiene la inclinación a 25° pero se cambia la rotación a 0° para que se mantenga en el posición de origen. En la figura 35 se observa cómo se desvía la rotación en aproximadamente 2° y al cabo de 5s vuelve a la posición inicial.

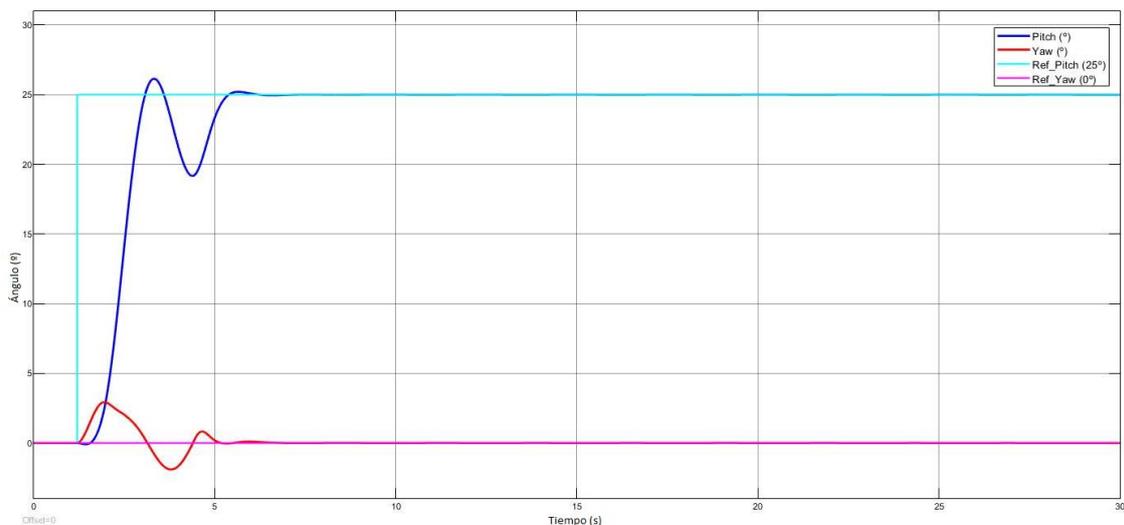


Fig. 35 Gráfica de las salidas para observar el efecto de la inclinación sobre la rotación

En la figura 36 se observa como en el regimen permanente, para mantener la rotación en el punto inicial es necesario que el motor de cola esté activado a una determinada velocidad, para así compensar la fuerza ejercida por el motor principal.

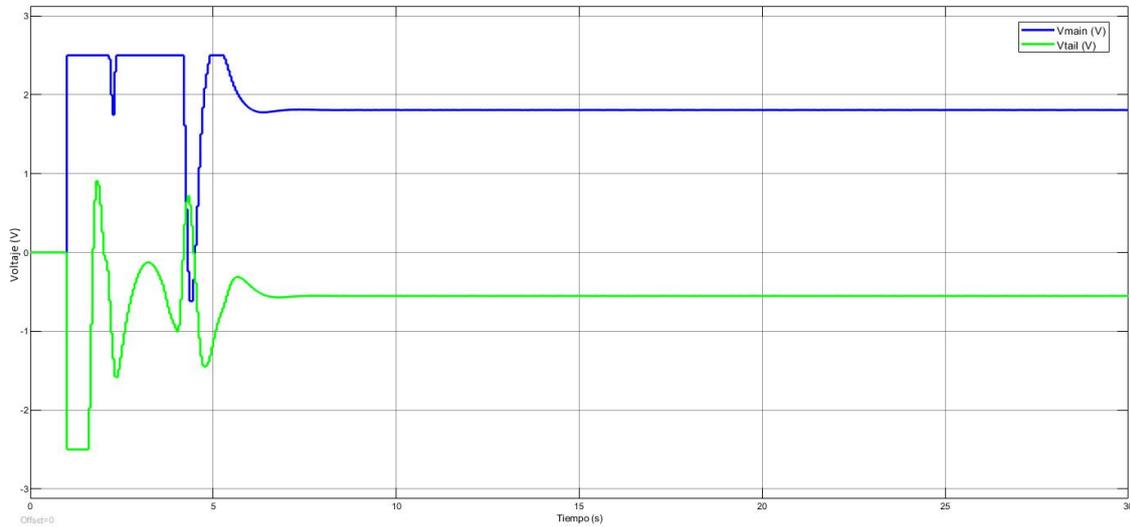


Fig. 36 Gráfica de las acciones de control influencia inclinación

En esta segunda simulación se cambia la referencia de la inclinación a 0° y la de la rotación a 25° , en la figura 37 se muestra que la desviación en este caso es de 6.5° en la inclinación, sin embargo el tiempo de establecimiento es mucho menor, por lo que aún siendo mayor la desviación permanece durante menos tiempo.

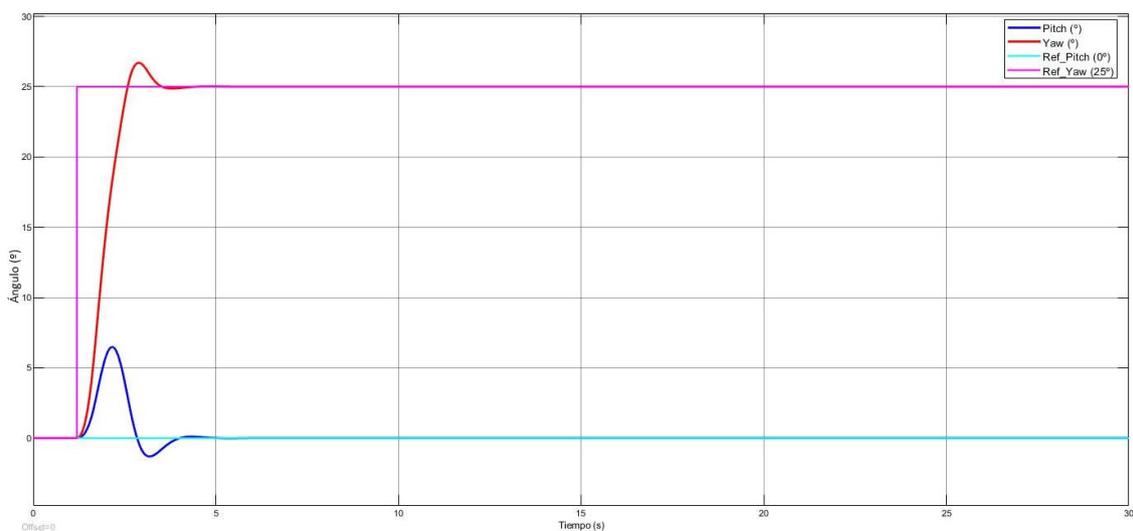


Fig. 37 Gráfica de las salidas para observar el efecto de la rotación sobre la inclinación

En la figura 38 se muestra como en el régimen permanente, para mantener la rotación en 25° apenas se proporciona voltaje al motor de cola, y además el motor principal está parado.

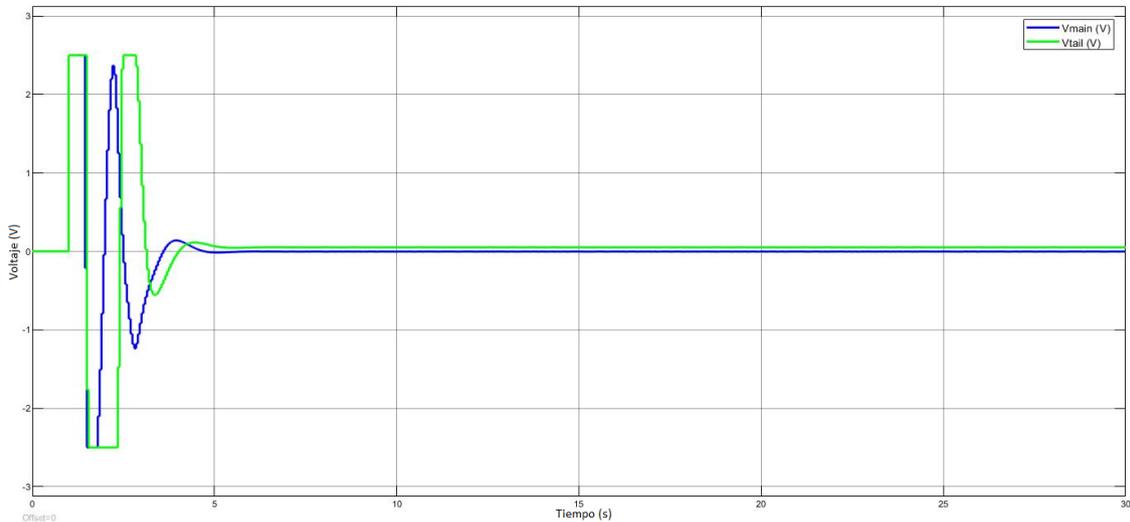


Fig. 38 Gráfica de las acciones de control para observar el efecto de la rotación sobre la inclinación

Vista estos resultados, se deduce que el motor principal tiene una mayor influencia en las dos posiciones del sistema que el motor de cola, por lo que si se elige una referencia de inclinación demasiado grande, las acciones de control no serán capaces de hacer que el sistema alcance alguna de las referencias.

Para la realización del experimento 1 se introducen al controlador como trayectoria de referencia, dos señales de tipo escalón, con un valor final para la inclinación de 0.43633 rad equivalente a 25°, y un valor final de 0.2618 rad para la rotación equivalente a 15°.

En la respuesta temporal en régimen transitorio (figura 39) se observa que es un sistema estable subamortiguado con un tiempo de establecimiento de 4.96 s \approx 5 s, con unas sobreoscilaciones de:

$$\bullet \delta_{Pitch} = \frac{M_p - V_f}{V_f} = \frac{25,37 - 25,05}{25,05} = 1,277\% \quad (11)$$

$$\bullet \delta_{Yaw} = \frac{M_p - V_f}{V_f} = \frac{18,53 - 15,06}{15,06} = 23,04\% \quad (12)$$

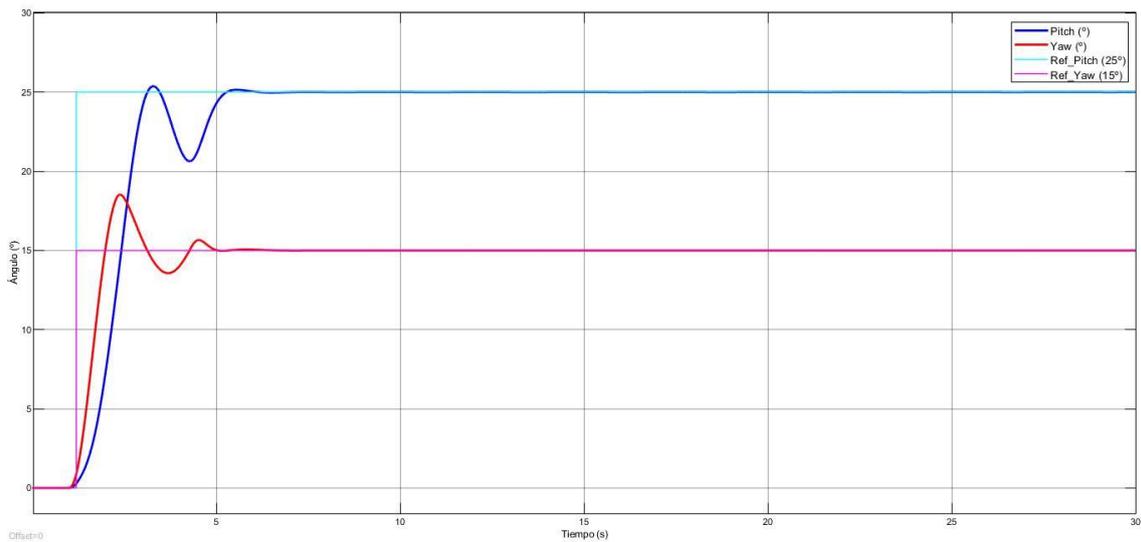


Fig. 39 Gráfica de las salidas del sistema

En el régimen permanente se observa como el error de posición es nulo, es decir, la diferencia entre la referencia aplicada al servosistema y la realimentación de la salida es cero. Dado que el sistema alcanza la referencia deseada, se intenta aumentar los parámetros de diseño (horizonte de predicción y de control) para mejorar la respuesta en régimen transitorio, haciendo un controlador más rápido y robusto. Sin embargo, se observa que estos cambios mejoran mínimamente la respuesta, por lo que solamente se habría conseguido aumentar la complejidad computacional.

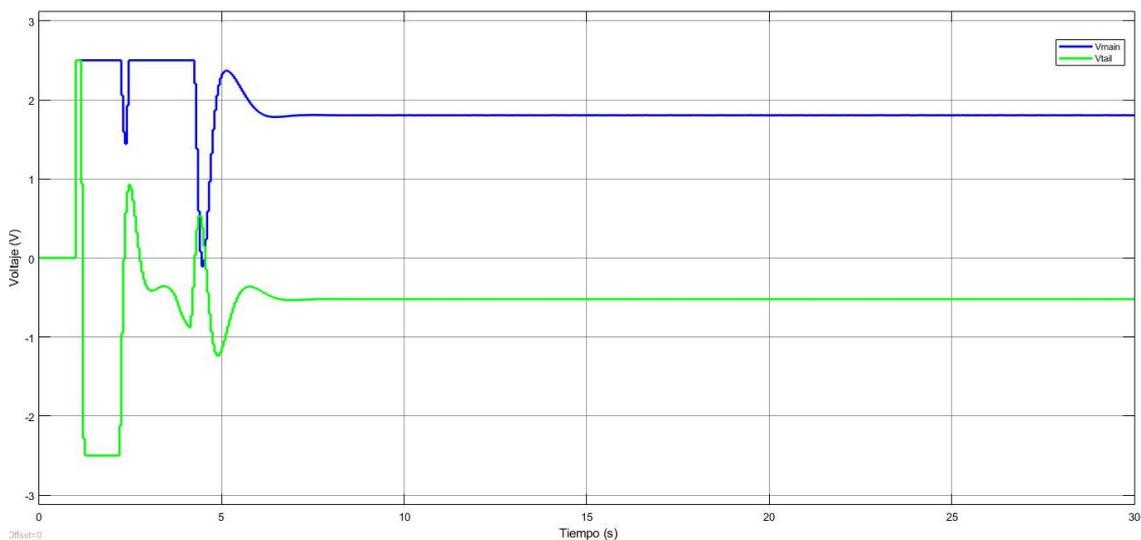


Fig. 40 Gráfica de las acciones de control del MPC en Simulink

En la figura 40 se observa que cuando el sistema está estabilizado, las señales de control no se encuentran saturadas por lo que se puede alcanzar dicha referencia, ya que si se elige una referencia mucho más grande la acción de control se saturará y no podrá alcanzarla.

5.3.4. Integración del código en SoMachine

Una vez se ha generado el texto estructurado del controlador diseñado y las posteriores POU's en CodeSys. Se importan dichas POU's en SoMachine. Se debe de crear el programa principal (de nombre MPC), el cual se debe seleccionar en la tarea MASK para que se ejecute cíclicamente. En este diagrama de contactos se deben procesar las señales para adecuarlas al MPC diseñado. Como el MPC se ha diseñado teniendo en cuenta el modelo lineal del proceso en el que se proporciona las salidas en radianes, las señales que se deben introducir como salidas realimentadas deben estar también en radianes. Al recibirse estas señales en el rango de 0 a 10000, es necesario realizar un escalado a $-\pi$ a π . Además la salida de las acciones de control se debe transformar para que se envíen en el rango de 0 a 5000. Para la realización de estas transformaciones se utilizan los bloques descritos en el apartado 5.2.3.

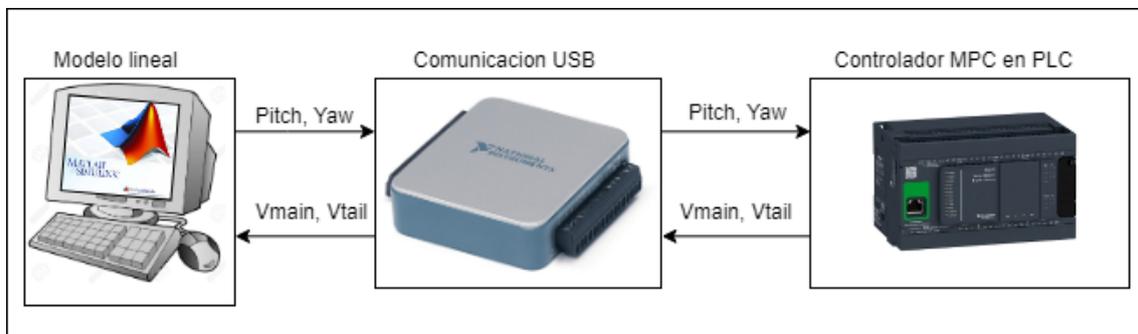


Fig. 41 Experimento de simulación 2

Las conexiones que se deben realizar entre el PLC y la tarjeta de adquisición son las siguientes:

- El terminal de masa del PLC con cualquier terminal de masa de la tarjeta.
- La salida 1 del PLC con la entrada E0 de la tarjeta (Vmain).
- La salida 2 del PLC con la entrada E1 de la tarjeta (Vtail).
- La entrada 1 del PLC con la salida S0 de la tarjeta (Pitch).
- La entrada 2 del PLC con la salida S1 de la tarjeta (Yaw).

Como el M241 solo trabaja de 0 a 10 V, se deben escalar las señales para adecuarlas a este rango de trabajo, tanto las que se van a recibir como las que se van a enviar, para ello se utilizan los bloques de función donde se introducen las siguientes ecuaciones lineales:

- (Entradas) $\frac{x}{5} = \frac{y+2,5}{2,5+2,5} \rightarrow x = y + 2,5 \rightarrow y = x - 2,5$ (13)

- (Salidas) $\frac{x+\pi}{\pi+\pi} = \frac{y}{10} \rightarrow 2\pi \cdot y = 10x + 10\pi \rightarrow y = \frac{5}{\pi}x + 5$ (14)

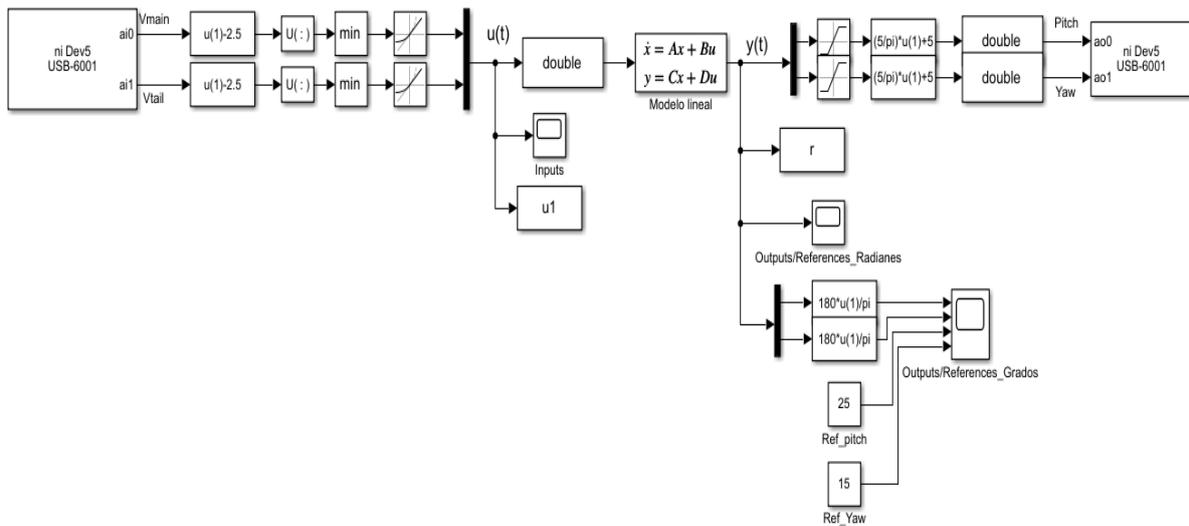


Fig. 42 Esquema Simulink PLC-Modelo Lineal

6. Resultados obtenidos

Al simular el programa en el PLC, en cuanto se cambia el estado del bloque de control del modo inicialización al modo activo, se produce un “error de excepción” que detiene el programa, no permitiendo la simulación hasta que se resetea el PLC.

Este error es debido a que el bloque de control tiene una complejidad de cálculos muy grande, ya que está procesando los cálculos necesarios para obtener las acciones de control futuras y optimizándolas, para que sitúe la salida lo más próxima a la referencia, en cada instante de tiempo. Siendo el autómata incapaz de procesar tanta información en un tiempo de ciclo de 50ms, se decide aumentar el tiempo de ciclo al que se ejecutara la tarea hasta que no salte el error de excepción. Esto se consigue para un tiempo de ciclo de 0.7s (Figura 43). Al realizar este cambio hay que volver a diseñar el MPC cambiando el T_s a 0.7s ya que debe estar sincronizado con el tiempo de ejecución de la tarea con el del MPC.

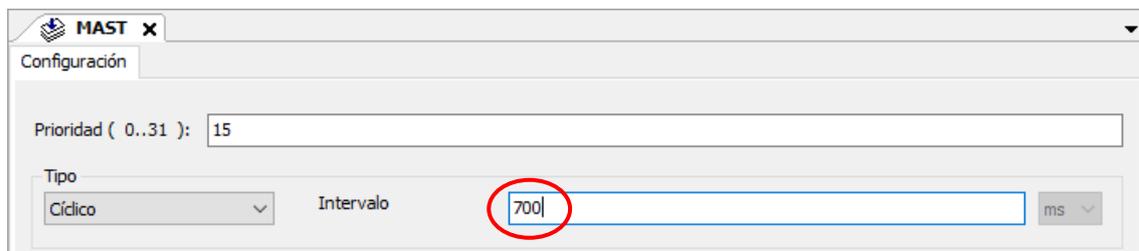


Fig. 43 Configuración tarea MAST

Debido a este cambio, teóricamente la salida del sistema deberá empeorar porque el MPC es más lento, lo que puede provocar que no tenga la capacidad de controlar completamente la dinámica del sistema además de que la influencia que pueda tener una perturbación en la respuesta del sistema sea mayor, debido a la lenta reacción que tendría el regulador.

Primero de todo, se realiza el proceso del experimento 1 (figura 33) para comprobar cómo es la respuesta con el nuevo controlador simulado, que como se observa en la figura 44 ha empeorado en comparación con la respuesta que se obtiene con el controlador inicial (Figura 39)

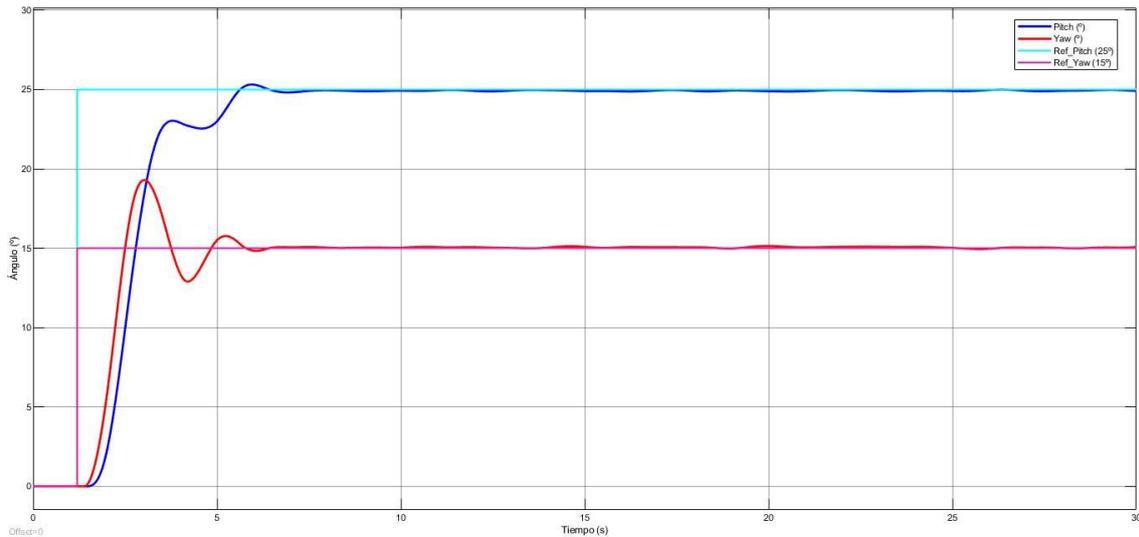


Fig. 44 Gráfica de la respuesta para el controlador simulado

En la respuesta temporal en régimen transitorio se observa que es un sistema estable subamortiguado en con un tiempo de establecimiento de 6.5s, y con unas sobreoscilaciones de:

- $\delta_{Pitch} = \frac{M_p - V_f}{V_f} = \frac{25,31 - 25}{25} = 1,24\% \quad (15)$

- $\delta_{Yaw} = \frac{M_p - V_f}{V_f} = \frac{19,3 - 15}{15} = 28,6\% \quad (16)$

En el régimen permanente se observa como existe cierto error de posición ya que la salida va oscilando sobre la referencia, con una amplitud de 0.02 rad.

A continuación, se realiza el proceso para poder realizar el experimento 2 (Figura 41). Al ejecutar el programa con el PLC y procesar las señales en Simulink se obtiene la siguiente respuesta (Figura 45), en la que se observa como la inclinación del sistema es oscilatoria, con un amplitud inicial de 0.2 rad (11.5°), esta oscilación va minimizándose con el tiempo pero sin llegar a estabilizarse en un valor constante. Y en la rotación del sistema se produce una desviación inicial bastante significativa, pero al contrario que la inclinación esta sí que se estabiliza en un valor de 25,83 segundos, pero no en un valor fijo si no que oscila mínimamente alrededor de la referencia.

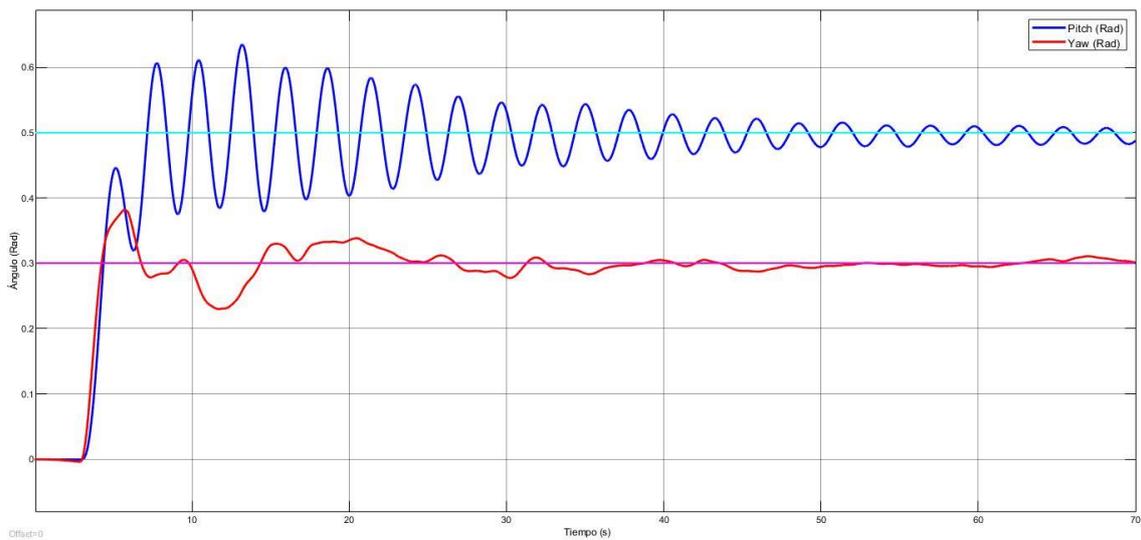


Fig. 45 Gráfica de la respuesta para el controlador real

En la figura 46, se observan las acciones de control generadas por el PLC, estas van oscilando para intentar minimizar las oscilaciones existentes en las salidas del sistema. La señal del motor principal, Vmain, está bastante cerca del valor de saturación, por lo que el regulador está al límite de ser capaz de alcanzar la referencia.

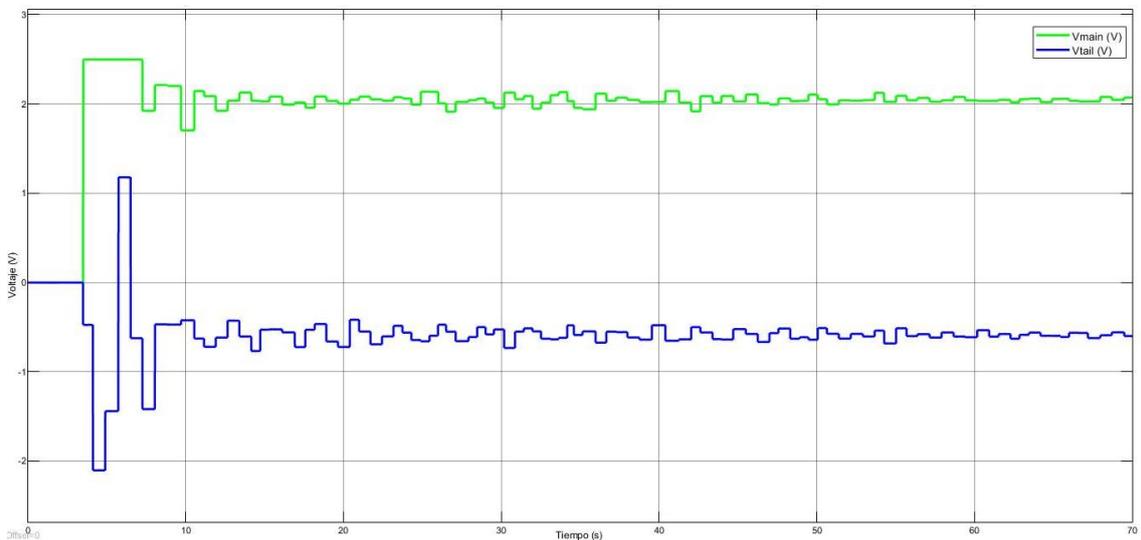


Fig. 46 Gráfica de las acciones de control para el controlador real

Las oscilaciones que aparecen en la respuesta del sistema son causadas en gran medida porque debido a la lentitud del controlador, este no es capaz de reducir las oscilaciones en un periodo de tiempo razonable. Para intentar minimizar dichas oscilaciones se vuelve a diseñar el MPC para un tiempo de ciclo mínimo con el que no se produzca el error de excepción,

para ello se debe de reducir la complejidad de los cálculos a realizar por el controlador en cada ciclo, esto se realiza disminuyendo los parámetros del MPC por la mitad del valor actual:

- Horizonte de predicción (p) = 14
- Horizonte de control (m) = 3
- Tiempo de muestreo (T_s) = 0.3 s

Se realiza de nuevo el experimento 1 con el MPC más sencillo que ha sido diseñado, obteniéndose la siguiente respuesta (Figura 47).

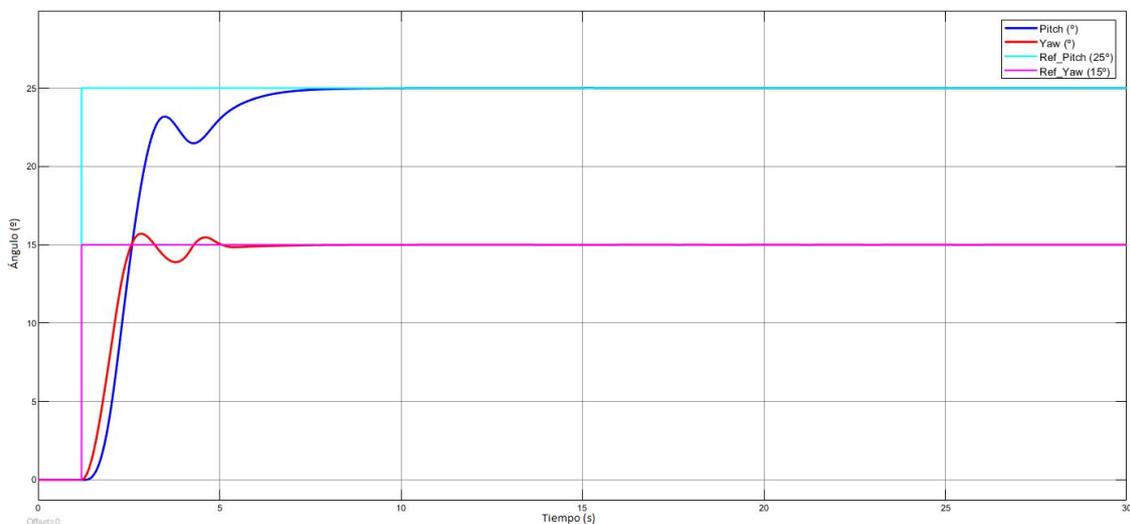


Fig. 47 Gráfica de la respuesta para el controlador simulado

En este caso, se observa como la respuesta temporal en el régimen transitorio es mejor que el anterior controlador, estabilizándose la inclinación en un valor de 6,75 segundos y la rotación en 4,9 segundos. En este caso solo hay sobreoscilación en el ángulo de rotación, de un valor de 4.73%.

En el régimen permanente se observa como el error de posición es nulo, mejorando la respuesta obtenida con el MPC anterior donde había una pequeña oscilación.

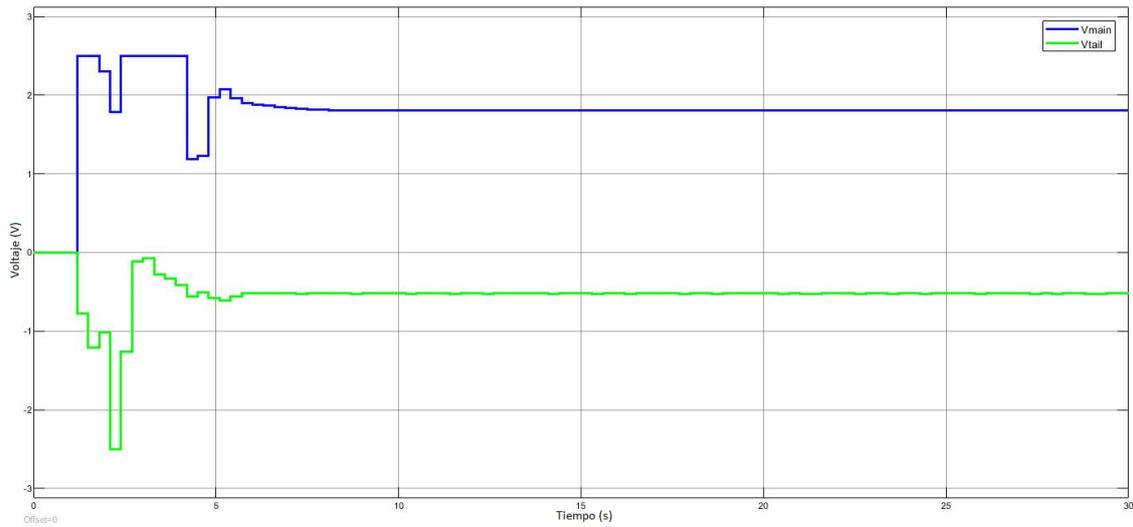


Fig. 48 Gráfica de las acciones de control para el controlador simulado

Realizando de nuevo el experimento 2, donde se debe de cambiar el tiempo de ciclo de la ejecución de la tarea MASK a 0.3s para concordar al valor de muestreo del controlador MPC. Se obtiene la siguiente respuesta representada en la figura 49.

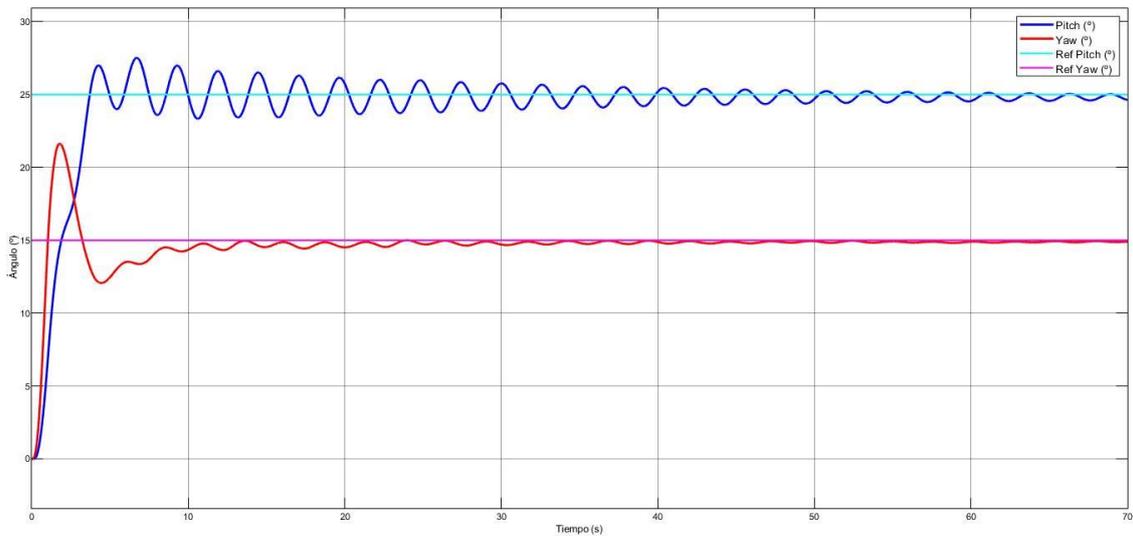


Fig. 49 Gráfica de la respuesta para el controlador real

A simple vista se observa como esta respuesta es mucho mejor que la que había sido obtenida con el controlador anterior. Minimizando las oscilaciones iniciales en la inclinación a 4° y estabilizándose en 32,7 segundos (el tiempo de estabilización se calcula cuando la respuesta del sistema varía en un 2-5% del valor final). En la rotación, se disminuye la oscilación a 1° aproximadamente, estabilizándose en 10,27 segundos. También aparece un gran sobreoscilación en la rotación de un valor de 44,26%, esto es debido a la gran influencia que tiene el motor principal en esta salida, sin embargo, en apenas 5 segundos ha sido minimizada.

- $\delta_{Pitch} = \frac{M_p - V_f}{V_f} = \frac{27 - 25}{25} = 8\% \quad (18)$
- $\delta_{Yaw} = \frac{M_p - V_f}{V_f} = \frac{21.64 - 15}{15} = 44.26\% \quad (19)$

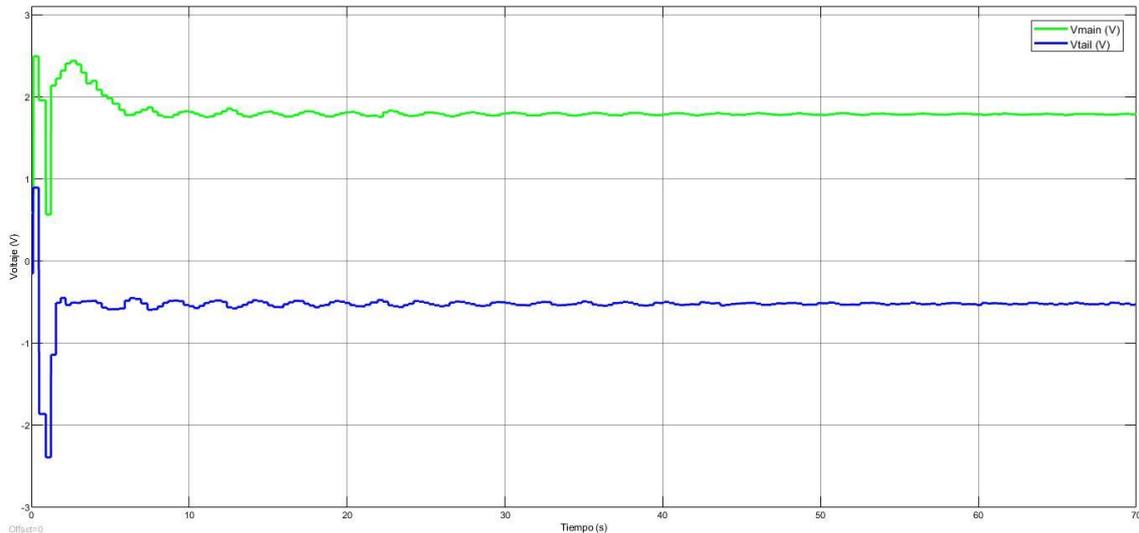


Fig. 50 Gráfica de las acciones de control para el controlador real

En las tablas 1 y 2 se muestra la comparativa de las características de las diferentes respuestas de todos los reguladores que han sido diseñados para el experimento 1 y el experimento 2.

RESULTADOS EXPERIMENTO 1							
Ts	p	m	Te Pitch(s)	Te Yaw(s)	δ_{Pitch}	δ_{Yaw}	Ep
0,05	28	5	4,96	4,96	1,277	23,04	0
0,7	25	5	6,5	5,6	1,24	28,6	1.14
0,3	14	3	6,75	4,9	0	4,73	0

Tabla 1 Comparativa resultados experimento 2

La tabla 1 revela que el último controlador diseñado, con un tiempo de muestreo de 0.3s, es más lento ya que el tiempo de establecimiento es mayor, pero sin embargo la sobreoscilación es mucho menor en comparación a las otras respuestas.

RESULTADOS EXPERIMENTO 2						
Ts	p	m	Te Pitch(s)	Te Yaw(s)	δ_{Pitch}	δ_{Yaw}
0,05	28	5	-	-	-	-
0,7	25	5	-	25,83	21,36	27,3
0,3	14	3	32,7	10,27	8	44,26

Tabla 2 Comparativa resultados experimento 2

En los controladores implementados en el PLC real, se observa que estos son mucho más lentos que los que han sido simulados en Simulink, llegando incluso en el regulador con el Ts de 0,7s a no estabilizarse, además de tener unas sobreoscilaciones demasiado grandes. La principal razón de que las respuestas tengan grandes oscilaciones y se obtengan las características mostradas en la tabla 2, es porque la comunicación entre el autómatas y el Simulink no se está realizando en tiempo real absoluto, por lo que las acciones que genera el regulador son para las salidas en instantes anteriores, de ahí que se generen oscilaciones.

Otra razón por la que se producen estas oscilaciones a la salida es porque el controlador no tiene acceso a todas las variables de estado que describen la dinámica del sistema, produciéndose una pérdida de parte de la información. Mediante un observador del sistema se obtendría dicha información, que mejoraría la respuesta, sin embargo supondría un aumento considerable de la complejidad computacional, ya que la cantidad de cálculos que se deberán realizar en el optimizador del MPC aumenta significativamente, pudiendo provocar que el autómatas no pueda realizar la ejecución del programa en el ciclo de tiempo para el que se ha diseñado el regulador. Lo que supondría tener que rediseñar los parámetros del MPC.

7. Conclusiones

Mediante el trabajo realizado se concluye que se ha alcanzado el objetivo principal que se había marcado al inicio del mismo, que era la integración de un algoritmo de control avanzado en un PLC real, a pesar de haber tenido problemas de compatibilidad entre los programas utilizados, ya que al necesitarse tantos programas, se debe seguir el proceso a conciencia porque saltarse algún paso supone que el siguiente programa no te acepte el archivo.

También se ha alcanzado el objetivo de diseño del controlador MPC para el modelo lineal del TRMS, ya que en simulación se obtienen unas respuestas verdaderamente buenas. Sin embargo, estas respuestas empeoran significativamente al implementar el MPC en el PLC real, aunque la respuesta siempre va a ser peor en el PLC real que en la simulación a causa de las pérdidas que existen. Siendo este empeoramiento debido a que la comunicación entre este y la simulación en Simulink no se realiza en tiempo real. Este es un punto por mejorar en el futuro para conseguir una respuesta óptima.

Otra mejora futura, será la implementación del MPC para controlar el modelo real del TRMS, con unos parámetros de diseño que minimicen al máximo la oscilación del sistema, además de tener en cuenta el punto inicial de funcionamiento y el rango de trabajo del sistema junto con el PLC.

Gracias al proyecto he adquirido conocimientos que no se han impartido en la carrera como es el control predictivo, siendo de una amplia aplicación, además de ampliar la gama de controladores lógicos con sus interfaces de uso ya que antes de la realización del proyecto solo había trabajado con PLC de Omron.

8. Bibliografía

Tim Zieger. "Control of a twin rotor MIMO system using model predictive control and a Kalman-filter based on a parametric model identified by utilizing the maximum likelihood method". 2018: [1]

Eduardo F. Camacho y Carlos Bordons. "Control Predictivo: Pasado, Presente y Futuro". Octubre 2004: [2]

Es.mathworks.com. (2019). *Simulation and Structured Text Generation Using PLC Coder-MATLAB & Simulink- MathWorks España*. [online] Available at: <https://es.mathworks.com/help/MPC/ug/simulation-and-structured-text-generation-using-PLC-coder.html?prodcode=MP&language=en>: [3]

Se.com. (2019). *EcoStruxure™ SoMachine | Schneider Electric*. [online] Available at: <https://www.se.com/es/es/product-range-presentation/2226-ecostruxure%e2%84%a2-somachine/>: [4]

Ni.com. (2019). *USB-6001 - National Instruments*. [online] Available at: <http://www.ni.com/es-es/support/model.usb-6001.html>: [5]

Se.com. (2019). *EcoStruxure™ Controlador lógico Modicon M241 | Schneider Electric*. [online] Available at: <https://www.se.com/es/es/product-range-presentation/62129-ecostruxure%E2%84%A2-controlador-l%C3%B3gico-modicon-m241/#tabs-top>: [6]

Es.mathworks.com. (2019). *Understanding Model Predictive Control*. [online] Available at: <https://es.mathworks.com/videos/series/understanding-model-predictive-control.html> [7].



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

TRABAJO FINAL DE GRADO

IMPLEMENTACIÓN DE ALGORITMOS DE CONTROL AVANZADOS
MPC EN PLC'S INDUSTRIALES

PLIEGO DE CONDICIONES

Curso 2018-2019

Autor:

Diego Ferri Beltrán (*dieferbe@etsid.upv.es*)

Tutor:

Sergio García-Nieto Rodríguez

Grado en Ingeniería Electrónica Industrial y Automática
Escuela Técnica Superior de Ingeniería del Diseño - ETSID
Universidad Politécnica de Valencia - UPV

Valencia, Julio de 2019

INDICE DEL PLIEGO DE CONDICIONES:

1.	Definición y alcance del pliego	67
2.	Normativa y condiciones de carácter general.....	67
3.	Condiciones particulares	68
3.1.	Condiciones técnicas.....	68
3.1.1.	Condiciones de los materiales.....	68
3.1.2.	Condiciones de la ejecución	68
3.2.	Condiciones facultativas	69
3.2.1.	Responsabilidades del contratista.....	69
3.2.2.	Responsabilidades del contratante	69
3.3.	Condiciones económicas.....	70
3.3.1.	Base General.....	70
3.3.2.	Fianza	70
3.3.3.	Penalizaciones	70
3.4.	Condiciones legales.....	71
3.4.1.	El contratista.....	71
3.4.2.	El contrato	71
3.4.3.	Rescisión del contrato	71

1. Definición y alcance del pliego

El objeto de este documento es fijar las condiciones técnicas mínimas que debe cumplir el diseño y la implementación de un MPC en un PLC real, especificando los requisitos de durabilidad, fiabilidad y seguridad.

El ámbito de aplicación de este documento se extiende a todos los sistemas mecánicos, eléctricos y electrónicos que forman parte de la instalación. En determinados supuestos se podrán adoptar, por la propia naturaleza del mismo o del desarrollo tecnológico, soluciones diferentes a las exigidas en este documento, siempre que quede suficientemente justificada su necesidad y que no impliquen una disminución de las exigencias mínimas de calidad especificadas en el mismo.

2. Normativa y condiciones de carácter general

El proyecto deberá cumplir con las siguientes normas de carácter general:

- RD 177/2008, de 8 febrero, por el que se establece el título de técnico de instalaciones Eléctricas y Automáticas.
- Norma IEC-1131 sobre la estandarización de los lenguajes de programación y sobre los diferentes tipo de autómatas programables.
- UNE 20 514 1M, recoge las normas de seguridad para aparatos electrónicos.
- ITC-BT-23, protección contra sobretensiones
- ITC-BT-24, protección contra contactos directos e indirectos
- IEC 61508, se establecen los requisitos para Sistemas Electrónicos Programables empleados en Seguridad.
- IEC/TS 62443-1-1, de 2009: Especificación técnica que define la terminología, conceptos y modelos para los sistemas de control y automatización industrial.

3. Condiciones particulares

3.1. Condiciones técnicas

3.1.1. Condiciones de los materiales

- Ordenador Personal:
El ordenador de tener las siguientes características mínimas: procesador COREi5, memoria RAM de 2GB y espacio libre en el disco duro de 50Gb y un sistema operativo Windows 8 o posterior. Además debe tener instalado el siguiente software: Matlab 2017b o posteriores de Mathworks (con los siguientes plugins: Simulink, PLC Coder, Data Acquisition Toolbox, Control Predictive Model), SoMachine V4.3 de Schneider, CodeSys (con el plug-in Simulink PLC Coder Import), NI Device Monitor.
- Autómata programable:
Se utilizará el PLC M241 de Schneider Electric compatible con SoMachine, la alimentación de este puede ser de 24 V de continua o de 100-240V de alterna.
Debe contener un módulo de entradas analógicas y un módulo de salidas analógicas, con un mínimo de dos puertos accesibles en cada módulo.
- Tarjeta de adquisición de datos:
Se utilizará el USB 6001 de National Instruments, debe de tener 2 entradas analógicas y 2 salidas analógicas libres y accesibles para realizar las conexiones pertinentes.

3.1.2. Condiciones de la ejecución

- Diseño del controlador MPC
En el diseño del MPC se debe de tener en cuenta las limitaciones físicas que contiene el sistema a controlar. Incluyendo estas limitaciones como restricciones del regulador, duras para las acciones de control y suaves para las salidas del sistema.
- Programación PLC
La programación del PLC se realizará de acuerdo con los estándares de los lenguajes de programación según la Norma IEC-

1131, en la que se incluyen todos los lenguajes válidos para la programación de controladores lógicos.

- Comunicación PC-PLC

Se deben realizar las siguientes conexiones que entre el PLC y la tarjeta de adquisición, para que se realice de forma correcta la comunicación entre el regulador en el PLC y el modelo lineal en Simulink. Ya que en caso de no realizarse de esta manera las señales mandadas no serán las adecuadas por lo que la respuesta que se obtendrá del sistema será incorrecta.

- El terminal de masa del PLC con cualquier terminal de masa de la tarjeta.
- La salida 1 del PLC con la entrada E0 de la tarjeta.
- La entrada 1 del PLC con la salida S0 de la tarjeta (Salida del modelo).
- La entrada 2 del PLC con la salida S1 de la tarjeta (Señal senoidal de referencia).

3.2. Condiciones facultativas

3.2.1. Responsabilidades del contratista

- Debe conocer toda la normativa vigente que rigen su actividad profesional y actuar conforme a ella.
- Asegurarse que todos los materiales utilizados cumplen con las especificaciones técnicas mínimas recogidas en este documento, rechazando aquellos que no la cumplan.
- Debe realizar los controles necesarios para asegurar la calidad de acuerdo con el proyecto y la normativa.
- Realizadas las comprobaciones pertinentes tiene la responsabilidad de dar por válido el resultado obtenido o no.

3.2.2. Responsabilidades del contratante

- Debe proporcionar toda la información que disponga relacionada con el proyecto al contratista, además de las especificaciones de diseño del proyecto que quiere llevar a cabo.
- Debe obtener todos los permisos necesarios para la realización del proyecto.

- Cuando se produzcan retrasos en la finalización del proyecto por motivos ajenos a la contrata no podrá reclamar, sin embargo deben de estar justificados correctamente, en caso contrario se aplicarán las penalizaciones recogidas en el apartado 3.3.3

3.3. Condiciones económicas

3.3.1. Base General

- El contratista tiene derecho a cobrar por la realización del proyecto, siempre que se haya realizado de acuerdo con las condiciones especuladas.
- El precio de la ejecución del proyecto viene recogido en el Presupuesto. El cálculo de este precio es el resultado de la suma de los costes directos, indirectos, gastos generales e impuestos.

3.3.2. Fianza

La fianza es el porcentaje del precio total del proyecto que debe ser depositado por el contratista como garantía a la firma del contrato, tendrá el siguiente valor:

- Depósito en metálico o transferencia bancaria por el importe de 10% del precio total del proyecto.

Dicha fianza será devuelta al contratista en un plazo inferior a 30 días hábiles después de la finalización del proyecto.

3.3.3. Penalizaciones

Se distinguen dos tipos de penalizaciones:

- Por retraso en el plazo de terminación:

Si se produce un retraso no justificado a la hora de terminar el proyecto se cargarán recargos por valor de 2% por día hábil de retraso, descontando y reteniendo dicho valor a la fianza.

- Por demora en los pagos:

Si el propietario no abona la cantidad estipulada en el contrato en un plazo hábil de quince días, se aumentará el precio un 1% por día hábil de demora en el pago.

3.4. Condiciones legales

3.4.1. El contratista

El contratista es el responsable de la ejecución del proyecto en las condiciones que se establecen en el contrato y en los documento del proyecto. Además, se obliga a lo establecido en la Ley de contratos de trabajo y dispuesto en la de accidentes de trabajo, subsidio familiar y seguro sociales.

Corre a cargo del contratista el pago de los impuestos municipales cuyo abono deberá efectuarse durante el periodo que dure la ejecución del proyecto.

3.4.2. El contrato

El contrato se realizará por escrito deberá firmarse por todas las partes que estén involucradas en el proyecto, contendrá el precio final del proyecto recogido en el presupuesto, junto con las penalizaciones vistas en el apartado

3.3.3

También recogerá las demás cláusulas que se hayan negociado entre las partes.

3.4.3. Rescisión del contrato

Para que sea posible una rescisión del contrato ya firmado por todas las partes se deberá de dar alguna de las siguientes situaciones:

- Acuerdo común para la rescisión
- Incumplimiento de los pagos acordados
- Modificación de las especificaciones principales del proyecto



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

TRABAJO FINAL DE GRADO

IMPLEMENTACIÓN DE ALGORITMOS DE CONTROL AVANZADOS
MPC EN PLC'S INDUSTRIALES

PRESUPUESTO

Curso 2018-2019

Autor:

Diego Ferri Beltrán (*dieferbe@etsid.upv.es*)

Tutor:

Sergio García-Nieto Rodríguez

Grado en Ingeniería Electrónica Industrial y Automática
Escuela Técnica Superior de Ingeniería del Diseño - ETSID
Universidad Politécnica de Valencia - UPV

Valencia, Julio de 2019

INDICE DEL PRESUPUESTO:

Introducción:	76
Cuadro de Mano de Obra	77
Cuadro de Maquinaria.....	78
Cuadro de Materiales	79
Cuadro de Precios Descompuestos	80
Resumen del Presupuesto	81

Introducción:

El presente presupuesto proporciona toda la información detallada del coste que supone la ejecución del proyecto: “Implementación de Algoritmos de Control Avanzados MPC en PLCs Industriales”, se expone el coste del diseño, simulación e implementación en PLC real del controlador MPC.

Se ha dividido en 4 apartados, de acuerdo con cómo se ha realizado el proyecto:

1. Identificación de los parámetros de funcionamiento del TRMS
2. Diseño controlador MPC
3. Simulación del MPC con el modelo linealizado
4. Integración del código en SoMachine

Para cada apartado se calcularán todos sus costes, de materiales, maquinaria y de mano de obra. Además se incluye un resumen con los costes directos y los impuestos.

Cuadro de Mano de Obra

Nº	Código	Ud	Descripción		Total
1	MO 01	75 h	Ingeniero técnico	18,940	1420,50

Precio total por1420,50

Son mil cuatrocientos veinte Euros con cincuenta céntimos

Cuadro de Maquinaria

Nº	Código	Ud	Descripción		Total
1	MAQ 01	1 ud	El sistema TRMS muestra los principios de un sistema no lineal MIMO (entrada múltiple, salida múltiple), con acoplamiento cruzado significativo. Su comportamiento se asemeja a un helicóptero, pero el ángulo de ataque de los rotores es fijo y las fuerzas aerodinámicas se controlan variando las velocidades de los motores. Se observa un acoplamiento cruzado significativo entre las acciones de los rotores, y cada rotor influye en ambas posiciones de ángulo.	15000,000	384,00
2	MAQ 02	1 ud	PC Sobremesa DELL Optiplex 3060 con procesador Intel i5-8500, 4GB de RAM y 500GB de memoria HDD.	609,990	609,99
3	MAQ 03	1 ud	Controlador lógico de la familia modicon 241 de Schneider Electric.	357,040	357,04
4	MAQ 04	1 ud	Dispositivo USB de E/S Multifunción, 8 AI (14 Bits, 20 kS/s), 2 AO (5 kS/s/canal), 13 DIO.	216,000	216,00
5	MAQ 05	1 ud	Monitor de pantalla Dell de 21".	110,400	110,40

Precio total por16.293,43

Son dieciséis mil doscientos noventa y tres Euros con cuarenta tres céntimos

Cuadro de Materiales

Nº	Código	Ud	Descripción		Total
1	MAT 01	0,480 ud	Licencia Anual otorga el derecho de usar el software, y el Servicio de mantenimiento del software MathWorks está incluido en la tarifa de la licencia.	800,000	384,00
2	MAT 02	10,000 ud	Cable de cobre para conexiones de tipo banana-banana, de 0.5 m de longitud.	3,440	34,40

Precio total por418,40

Son cuatrocientos dieciocho Euros con cuarenta céntimos

Cuadro de Precios Descompuestos

Nº	Código	Ud	Descripción		Total
1 Implementación de Algoritmos de Control Avanzados M...					
1.1	D01		Estudio de los rangos de funcionamiento de las entradas y de las salidas del TMRS, mediante el uso de Matlab y la tarjeta de adquisición USB-6001		
	MO01	10,000 h	Ingeniero técnico	18,940	189,40
	MAQ01	1,000 ud	Twin Rotor MIMO System	15.000,000	15.000,00
	MAQ02	0,250 ud	PC Sobremesa DELL Optiplex 3060	609,990	152,50
	MAQ03	0,250 ud	Monitor de pantalla Dell 21"	110,400	27,60
	MAQ05	0,500 ud	USB-60001	216,000	108,00
	MAT01	0,120 ud	Licencia Anual de Matlab	800,000	96,00
	MAT02	5,000 ud	Cable de conexión tipo banana-banana	3,440	17,20
Precio total por					15.590,70
Son quince mil quinientos noventa Euros con setenta céntimos					
1.2	D02		Diseño del controlador MPC mediante Model Predictive Control Toolbox de Matlab		
	MO01	40,000 h	Ingeniero técnico	18,940	757,60
	MAQ02	0,250 ud	PC Sobremesa DELL Optiplex 3060	609,990	152,50
	MAQ03	0,250 ud	Monitor de pantalla Dell 21"	110,400	27,60
	MAT01	0,120 ud	Licencia Anual de Matlab	800,000	96,00
Precio total por					1.033,70
Son mil treinta y tres Euros con setenta céntimos					
1.3	D03		Simulación del MPC en Simulink con el modelo linealizado		
	MO01	5,000 h	Ingeniero técnico	18,940	94,70
	MAQ02	0,250 ud	PC Sobremesa DELL Optiplex 3060	609,990	152,50
	MAQ03	0,250 ud	Monitor de pantalla Dell 21"	110,400	27,60
	MAT01	0,120 ud	Licencia Anual de Matlab	800,000	96,00
Precio total por					370,80
Son trescientos setenta Euros con ochenta céntimos					
1.4	D04		Implementación MPC en el PLC M241 para el modelo lineal en Simulink		
	MO01	20,000 h	Ingeniero técnico	18,940	378,80
	MAQ02	0,250 ud	PC Sobremesa DELL Optiplex 3060	609,990	152,50
	MAQ03	0,250 ud	Monitor de pantalla Dell 21"	110,400	27,60
	MAQ04	1,000 ud	MODICON 241 TM241CE40R	357,040	357,04
	MAQ05	0,500 ud	USB-60001	216,000	108,00
	MAT01	0,120 ud	Licencia Anual de Matlab	800,000	96,00
	MAT02	5,000 ud	Cable de conexión tipo banana-banana	3,440	17,20
Precio total por					1.137,14
Son mil ciento treinta y siete Euros con catorce céntimos					

Resumen del Presupuesto

Importe

Capítulo 1 Implementación de Algoritmos de Control Avanzados MPC en PLCs Industriales	18.132,34
Presupuesto de ejecución material	18.132,34
0% de gastos generales	0,00
0% de beneficio industrial	0,00
Suma	18.132,34
21% IVA	3.807,79
Presupuesto de ejecución por contrata	21.940,13

Asciende el presupuesto de ejecución por contrata a la expresada cantidad de VEINTIUN MIL NOVECIENTOS CUARENTA EUROS CON TRECE CÉNTIMOS.

Valencia, julio de 2019
Grado en Ingeniería Electrónica Industrial y Automática

Diego Ferri Beltrán