

## Diseño basado en modelos de la estrategia iMO-NMPC: Implementación en dispositivos industriales

Zabaljauregi, A.<sup>a</sup>, Larrea, M.<sup>b,\*</sup>, Irigoyen, E.<sup>a</sup>, Artaza, F.<sup>a</sup>

<sup>a</sup>UPV/EHU, Escuela de Ingeniería de Bilbao, Plaza Ingeniero Torres Quevedo, 1, 48013 Bilbao  
<sup>b</sup>UPV/EHU, Escuela de Ingeniería de Gipuzkoa, Plaza Europa, 1, 20018 Donostia / San Sebastián

**To cite this article:** Zabaljauregi, A., Larrea, M., Irigoyen, E., Artaza, F. 2023. Model-based design of the iMO-NMPC strategy: Implementation in industrial devices. XLIV Jornadas de Automática, 192-197. <https://doi.org/10.17979/spudc.9788497498609.192>

### Resumen

Este trabajo presenta la metodología que se está empleando, dentro del grupo de investigación de control inteligente (GICI) de la UPV/EHU, para el desarrollo de estrategias de control inteligente y su posterior implementación en plataformas de tiempo real. En trabajos previos se mostró la implementación de la metodología hasta su fase de simulación. En el presente trabajo se muestra uno de los últimos pasos que en el que tras validar la simulación del código se pasa este a una plataforma de hardware industrial. De este modo, se va consolidando la metodología y mostrando la capacidad real de la estrategia de control iMO-NMPC en entornos restringidos. Como plataforma hardware se ha empleado el modelo Baseline de la casa Speedgoat, ya que es de facto el único hardware soportado por el entorno Simulink real-time de Mathworks.

*Palabras clave:* Metodología, Control Predictivo, Tiempo-real, Sistemas complejos, Problemas de optimización

### Model-based design of the iMO-NMPC strategy: Implementation in industrial devices

#### Abstract

This paper presents the methodology being used within the Intelligent Control Research Group (GICI) of the UPV/EHU for the development of intelligent control strategies and their subsequent implementation in real-time platforms. Previous works have shown the implementation of the methodology up to the simulation phase. This work shows one of the last steps, in which, after validating the simulation of the code, it is transferred to an industrial hardware platform. In this way, the methodology is consolidated and the real capacity of the iMO-NMPC control strategy in restricted environments is demonstrated. The Baseline model from Speedgoat has been used as the hardware platform, as it is de facto the only hardware supported by the Mathworks Simulink real-time environment.

*Keywords:* Methodology, Predictive Control, Real-time, Complex systems, Optimization problems

## 1. Introducción

En el ámbito del desarrollo de software y del control la metodología de desarrollo basado en modelo o *Model Based Design* (MBD) permite la continua verificación y validación de los pasos que se están realizando. De este modo, se pretenden abarcar las fases de desarrollo de software desde los inicios con simulaciones hasta las fases finales que representan la ejecución de los desarrollos en plataformas reales. Esta metodología, con

sus posibles variantes, se emplea en el desarrollo de estrategias de control apoyadas en herramientas asistidas por computador Bemporad (2006); Nunna et al. (2016); Walica and Noskiewi (2022), siendo Matlab/Simulink una de las herramientas más empleadas para estos cometidos.

El Grupo de Investigación de Control Inteligente (GICI) de la Universidad del País Vasco / Euskal Herriko Unibertsitatea, emplea la metodología MBD para el desarrollo de la estrategia

\*Autor para correspondencia: [m.larrea@ehu.eus](mailto:m.larrea@ehu.eus)  
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

de control denominada "intelligent MultiObjective Nonlinear Model Predictive Control" (iMO-NMPC) presentada en Valera et al. (2012). Esta estrategia se está desarrollando en el marco del proyecto nacional PID2020-120087GB-C22 que tiene como uno de sus objetivos la implementación de controladores predictivos multivariables basados en inteligencia computacional en plataformas de hardware industrial.

En el trabajo previo Larrea et al. (2014) se presentó una primera aproximación del uso de la estrategia iMO-NMPC en un PC industrial ejecutándose sobre el sistema operativo en tiempo real RTAI. Pero la evolución de las plataformas de desarrollo, que priorizaron soluciones de código cerrado, la necesaria modificación y mejora del código junto con el uso de la metodología MBD, llevó al GICI a trabajar en la nueva línea de implementar el código en Matlab/Simulink para poder ejecutarlo en una plataforma speedgoat. De este modo, en Zabaljauregi et al. (2022) se presentaron los primeros pasos de la aplicación de la metodología MBD; se validó la estrategia iMO-NMPC en script, se codificó la estrategia en el lenguaje de programación C, se integró el código en un bloque s-function mediante plantillas obteniendo como resultado un bloque s-function de Simulink que integraba la estrategia iMO-NMPC.

En el presente trabajo se retoma el desarrollo en este último punto y se realizan pruebas en tiempo real empleando para ello la opción de Simulink denominada *Desktop Real-Time*. Tras verificar el correcto funcionamiento de la estrategia en un PC de desarrollo y en tiempo real, se realizan los experimentos en tiempo real de alto rendimiento en la plataforma Speedgoat con la herramienta Simulink Real-Time.

La distribución de este trabajo es la siguiente. En el apartado dos se presenta el flujo de trabajo que se lleva a cabo dentro de la metodología MBD, en el apartado 3 se resaltan las modificaciones que se realizan en el código respecto a anterior implementaciones. En el apartado 4 se describen los sistemas a controlar además de los experimentos que se han llevado a cabo además de las características de la plataforma hardware speedgoat y en el apartado 5 se muestran los resultados obtenidos. El trabajo finaliza con el apartado 6 de conclusiones.

## 2. Flujo de trabajo: desde la simulación hasta la ejecución en tiempo real

En este apartado se pretende mostrar el flujo de trabajo que se lleva a cabo en el GICI para el desarrollo y validación de las estrategias de control avanzado que se desarrollan. De este modo, además de proponer soluciones novedosas se pretende llevar estas soluciones a hardware industrial para que no se queden en la fase de simulación.

### 2.1. Fases

Las fases de las que se compone la metodología de trabajo que se sigue dentro del GICI son las siguientes.

- FASE 1 o Script: Desarrollo y validación de estrategias de control mediante scripts de Matlab. En esta fase se recogen los nuevos desarrollos o modificaciones que se plantean de forma teórica. Se consigue una validación de las estrategias de control diseñadas.

- FASE 2 o S-Function Simulación (Simulink): Codificación en s-functions de desarrollos validados en fase anterior. Para ello se emplean las plantillas de s-function que proporciona Mathworks que permiten encapsular los desarrollos en un bloque de *Simulink* para su posterior simulación en dicho entorno. Se valida la implementación de la s-function en el entorno Simulink.
- FASE 3 o S-Function Pseudo Tiempo-real (Simulink Desktop real-time: Simulink): Se comprueba el correcto funcionamiento de la s-function en el entorno Simulink Desktop Real-Time ejecutada en la plataforma de desarrollo. Se obtiene la validación de la estrategia en entornos en tiempo real no estricto.
- FASE 4 o S-Function Tiempo-real. (Simulink real-time Speedgoat): Se comprueba el correcto funcionamiento de la s-function en una plataforma industrial (paso I). Tanto el controlador como la planta son ejecutados en dicha plataforma.
- FASE 5 o S-Function con señales externalizadas (HiL): Se comprueba el correcto funcionamiento de la s-function en una plataforma industrial (II). El controlador se ejecuta en la plataforma industrial mientras que la planta se ejecuta en otra plataforma.

La aplicación de las primeras tres fases fueron presentadas en el trabajo Zabaljauregi et al. (2022) para sistemas no lineales SISO, mientras que en Zabaljauregi et al. (2023) se presentó la extensión a sistemas MIMO pero solamente en la primera fase (script). Este trabajo presenta la validación de la estrategia iMO-NMPC en la FASE 4 "S-Function Tiempo-real" para sistemas SISO, dando así continuidad a los trabajos mencionados.

## 3. Implementación de la estrategia iMO-NMPC

Desde los primeros trabajos realizados en Larrea et al. (2014) para la validación de la estrategia iMO-NMPC en entornos industriales, se ha llevado a cabo diversas mejoras en el código que merecen ser resaltadas.

La primera de ellas es el uso de una nueva versión del código en C del algoritmo genético NSGA-II Deb et al. (2002). Aunque en esos primeros trabajos también se utiliza el algoritmo genético NSGA-II, se estaba empleando una revisión de código por K.Deb obsoleta. Por un lado, en un esfuerzo de modularización del código y reformulación de la lógica interna para resolver algunos errores de código y por otro lado, debido a cambios en el API y la conveniencia de emplear una revisión actualizada del NSGA-II se decide reprogramar la estrategia de control. La última versión del código del GA de K.Deb es la 1.1.6 (Yusoff et al. (2011); Deb et al. (2011)).

La segunda mejora va encaminada a la modularización de la estrategia iMO-NMPC que busca facilitar el intercambio ágil de las técnicas que se emplean dentro de ella. Las técnicas básicas son; optimizador multiobjetivo para la obtención de soluciones óptimas, el modelo de predicciones a emplear dentro de la estrategia NMPC y el gestor de decisiones para la selección de la solución más adecuada del frente de Pareto. En la figura 1 se muestran estos elementos y las variaciones aplicadas en el paso del código en script a s-function.

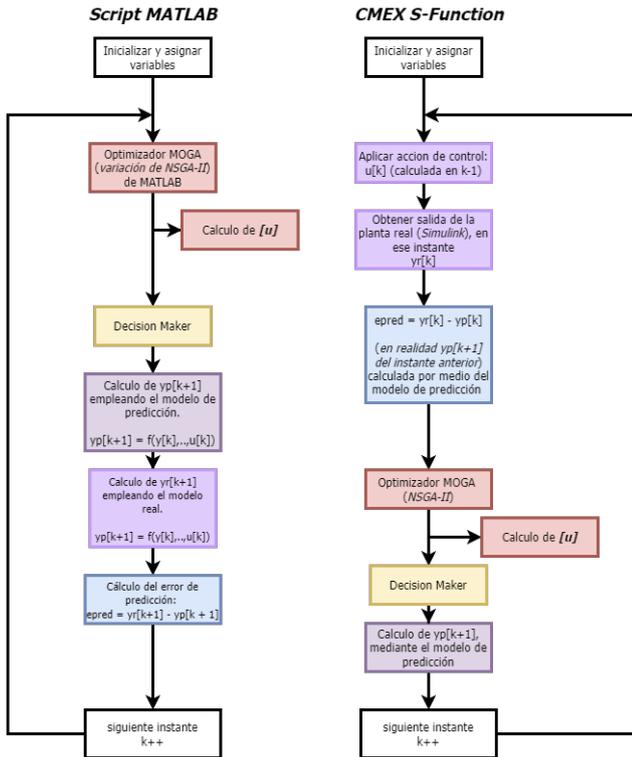


Figura 1: Diagrama de flujo de la estrategia iMO-NMPC

#### 4. Experimentos a realizar

La estrategia de control se valida con los sistemas de laboratorio que se definen a continuación.

**SNL1:** En 1 se muestra la ecuación del primer sistema no lineal. Se presentó por primera vez en Narendra and Parthasarathy (1990) en el marco de un trabajo sobre la identificación de sistemas dinámicos con redes neuronales.

$$y_1(k+1) = f_1(y_1(k), u_1(k)) = u_1(k)^3 + \frac{y_1(k)}{1 + y_1(k)^2} \quad (1)$$

**SNL5:** En 2 se muestra la ecuación del segundo sistema no lineal. Se presentó por primera vez en Harris (1994) en un libro sobre avances en Control Inteligente.

$$\begin{aligned} y_2(k+1) &= f_2(y_2(k), y_2(k-1), u_2(k)) = \\ &= \frac{1,5 \cdot y_2(k) \cdot y_2(k-1)}{1 + y_2(k)^2 + y_2(k-1)^2} + 0,7 \sin [0,5 (y_2(k) + y_2(k-1))] \cdot \\ &\quad \cdot \cos [0,5 (y_2(k) + y_2(k-1))] + 1,2u_2(k) \end{aligned} \quad (2)$$

Estos dos sistemas no lineales han sido empleados con anterioridad dentro del GICI Larrea et al. (2014); Zabaljauregi

et al. (2022), en la validación de la estrategia iMO-NMPC para el control de sistemas SISO. En Zabaljauregi et al. (2023), se emplearon estos sistemas para realizar pruebas de la estrategia para el control de sistemas MIMO desacoplados. Tal y como se ha mencionado anteriormente, la estrategia iMO-NMPC ha sido validada hasta la FASE 2 de la metodología para su versión SISO y hasta la FASE 1 en la variante MIMO. Este trabajo, muestra los resultados de las FASES 3 y 4 para sistemas SISO.

En la figura 2 se muestra la implementación de la estrategia de control en Simulink. En este caso, la referencia que el sistema de control debe hacer seguir está incrustada dentro del bloque sfun\_imo\_nmpc en forma de archivo de texto para que el controlador pueda conocerla en todo momento.

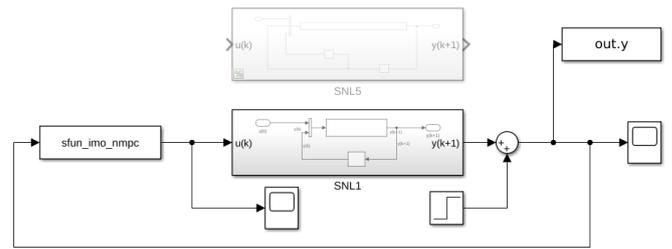


Figura 2: Esquema de bloques para iMO-NMPC S-Function en Simulink

Se debe remarcar que como modelo de predicción se ha empleado el correspondiente modelo matemático de cada sistema a controlar.

Por último se define el número de objetivos, siendo dos; el error de seguimiento y la variación de la acción de control.

##### 4.1. Experimentos FASE 3: Simulink Desktop Real-Time

Se realizarán dos tipos de experimentos en esta fase; una primera con variaciones en la referencia y la segunda con respuesta ante un escalón unitario con perturbación en la salida en el instante  $t = 2 \text{ seg}$ . La referencia que se aplicará en el primer experimento consistirá en escalones dentro del rango  $[-2, 2]$

En los experimentos se aplicarán diferentes horizontes de predicción para mostrar su efecto en la respuesta del lazo de control. Los horizontes seleccionados han sido  $h_c = 2, 4, 6$ .

Todos los experimentos se van a llevar a cabo en el PC de desarrollo (procesador Intel Core i9 y 16GB de memoria RAM), de modo que no se espera tener que limitar los individuos o generaciones del algoritmo genético. Estos parámetros quedan definidos para los experimentos como:

- Población = 200
- Generaciones = 200

##### 4.2. Experimentos FASE 4: Simulink Real-Time (Speedgoat)

Para concluir se procede con la simulación del algoritmo de control sobre una plataforma Speedgoat. La plataforma Speedgoat Baseline-S empleada es un ordenador industrial con las siguientes características:

- CPU: Intel Celeron 2 GHz, 4 cores.
- Memoria RAM: 4GB
- Unidad principal: 256 GB SSD.



Figura 3: Plataforma HW: Speedgoat Baseline

Para poder interactuar con la plataforma se debe contar con el Add-on de Simulink Real-Time y las librerías proporcionadas por el fabricante. La comunicación con el baseline se realiza mediante ethernet y se debe configurar la conexión IP en la plataforma de desarrollo para realizar la primera conexión. El siguiente paso en la configuración de la conexión, es el ajuste en Simulink Real-Time Explorer, en el ordenador de desarrollo. Mediante la herramienta slrtExplorer de MATLAB, se puede realizar la búsqueda de dispositivos target (baseline) en red. Finalmente, se puede dar al botón de conectar para establecer la conexión con la plataforma industrial.

Debido a la reducida capacidad de cómputo del ordenador industrial, frente al ordenador de desarrollo que se ha empleado, no se podrán emplear tantos individuos y tantas generaciones como en las anteriores fases. Si se cometiese el error de emplear un número demasiado elevado de individuos o generaciones en el Speedgoat, saltaría un error de sobrecarga; es decir, el TET (Target Execution Time/ Tiempo de ejecución del objetivo) sería superior al tiempo de muestreo establecido.

Los parámetros del algoritmo genético para los experimentos en el hardware industrial quedan definidos como:

- Población = 80
- Generaciones = 100

## 5. Resultados

En esta sección se presentan los resultados de los experimentos expuestos en apartado 4.

### 5.1. Sistema no lineal SNL1 / SNL5 Simulink Desktop Real-Time

Las gráficas 4, 5 muestran la evolución de los dos sistemas ante una referencia basada en escalones. Los experimentos se realizan para diferentes horizontes de control ( $h_c$ ), con valores de 2, 4 y 6. En todos los experimentos, con independencia del horizonte de control, la salida llega sin problemas a la referencia. El seguimiento se deteriora levemente a medida que se va aumentando el horizonte. Esto se debe al incremento del coste computacional causado por el incremento de cromosomas por individuo. Es por eso que debido a que el número máximo de iteraciones para todos los rangos es el mismo, el resultado se va deteriorando para mayores horizontes ya que no se llega tan cerca del óptimo. Adicionalmente, para los casos de  $h_c = 4$  y

$h_c = 6$ , se puede apreciar la antelación de la salida antes del cambio de referencia.

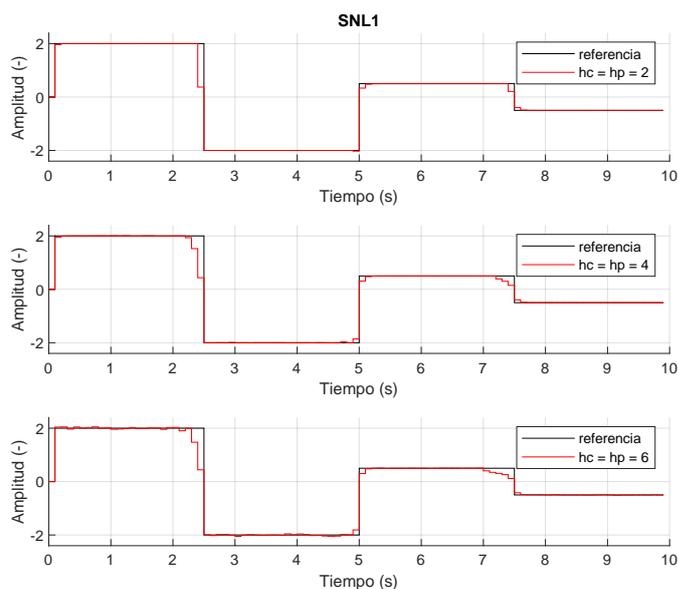


Figura 4: SNL1 en TR,  $h_c=hp=2:4:6$

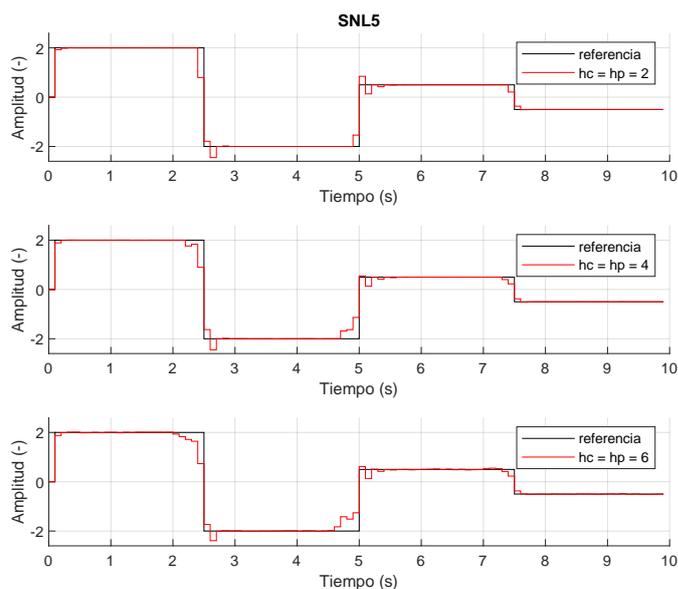


Figura 5: SNL5 en TR,  $h_c=hp=2:4:6$

Las tabla 1 representa el error cuadrático medio y máximo absoluto para simulaciones SISO en Simulink Desktop Real-Time.

SISO Sim (pop = 200, gen = 200)			
	$h_c = hp$	MSE	MaxAbsE
SNL1	2	0,0535	1,5999
	4	0,0501	1,543
	6	0,0526	1,5411
SNL5	2	0,0524	1,3958
	4	0,0542	1,4285
	6	0,0558	1,3

Tabla 1: Tabla de error cuadrático medio y máximo absoluto para simulaciones SISO

Las gráficas 6, 7 muestran la evolución de los dos sistemas ante una perturbación de 0.5 en la salida (segundo experimento de la FASE 3). Tal y como se ha mostrado en el anterior experimento, las gráficas muestran la respuesta para diferentes horizontes de control (hc), con valores de 2, 4 y 6. Se observa que el sistema SNL1 rechaza rápidamente las perturbaciones pero el sistema SNL5 muestra bastantes oscilaciones, posiblemente a raíz de la naturaleza oscilante de su ecuación.

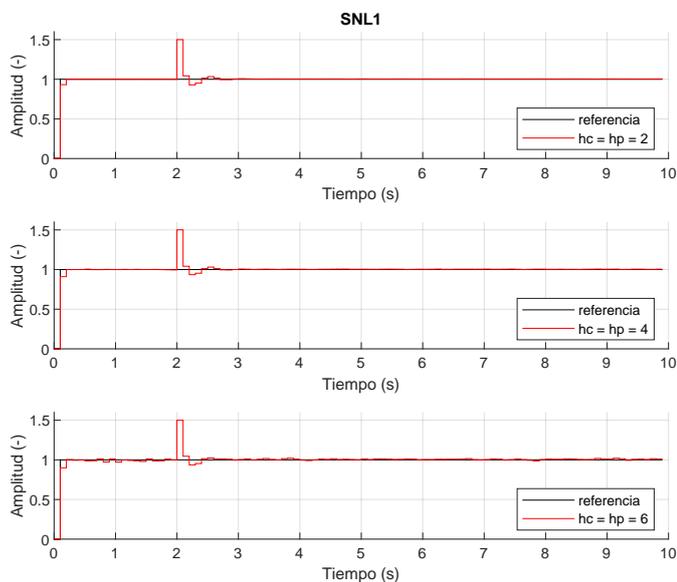


Figura 6: SNL1 en TR con perturbación, hc=hp=2:4:6

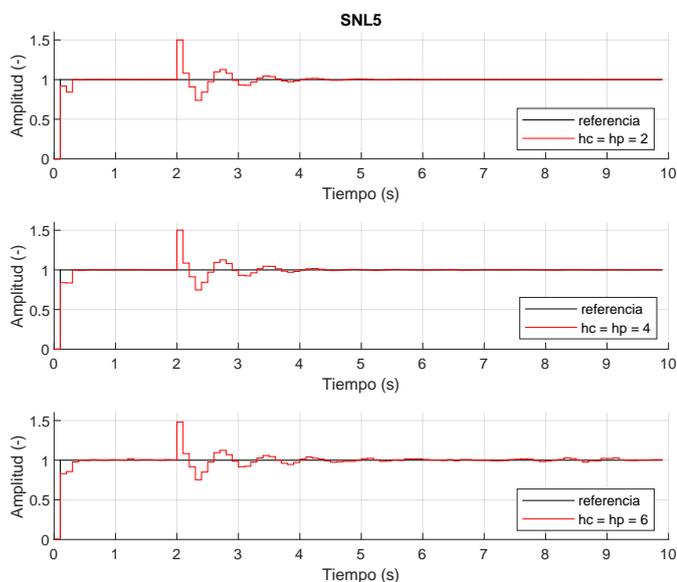


Figura 7: SNL5 en TR con perturbación, hc=hp=2:4:6

SISO Sim + pert (pop = 200, gen = 200)			
	hc = hp	MSE	MaxAbsE
SNL1	2	0,0066	0,508
	4	0,0071	0,6041
	6	0,0069	0,5818
SNL5	2	0,0073	0,5
	4	0,0069	0,4983
	6	0,0072	0,5094

Tabla 2: Tabla de error cuadrático medio y máximo absoluto para simulaciones SISO con perturbación

### 5.2. Sistema no lineal SNL1 / SNL5 Simulink Real-Time (Speedgoat)

Las gráficas 8, 9 muestran la evolución de los dos sistemas ante una referencia basada en escalones. Aunque con una población e iteraciones inferior a la habitual para simulaciones anteriores en *Simulink*, el resultado que se puede apreciar es satisfactorio.

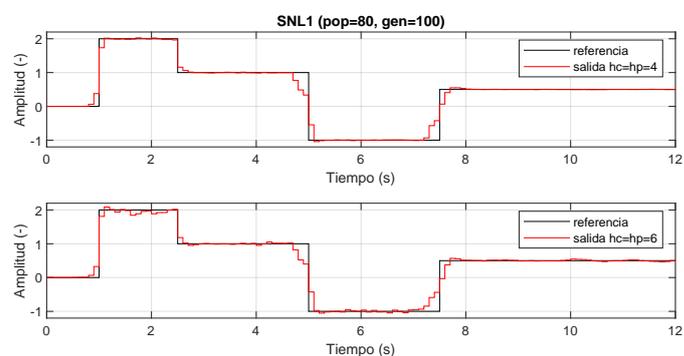


Figura 8: SNL1 en TR - Speedgoat, hc=hp=4:6

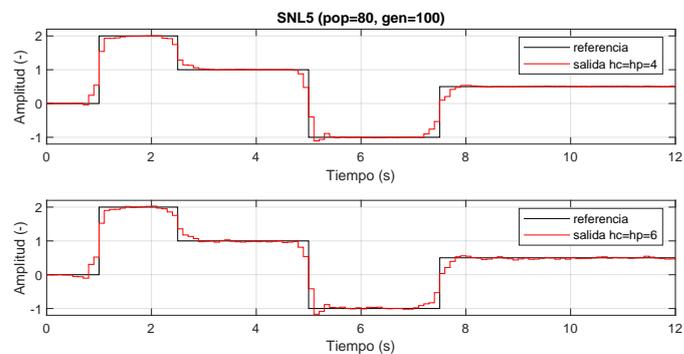


Figura 9: SNL5 en TR - Speedgoat, hc=hp=4:6

En la tabla 3 se exponen los errores cuadráticos medios y errores absolutos cometidos por ambas salidas en su control para ajustarse a las referencias. Se puede observar que el MSE se mantiene por debajo de 0.02 para todos los sistemas.

SISO RT (pop = 80, gen = 100)			
	hc = hp	MSE	MaxAbsE
SNL1	4	0,01608	0,6652
	6	0,01711	0,6003
SNL5	4	0,01656	0,6022
	6	0,01802	0,5839

Tabla 3: Tablas de error cuadrático medio y absoluto para simulaciones en tiempo real SISO

La tabla 2 representa el error cuadrático medio y máximo absoluto para simulaciones SISO con perturbaciones en *Simulink Desktop Real-Time*.

## 6. Conclusiones

En este trabajo se ha presentado la implementación en un dispositivo industrial de la estrategia iMO-NMPC. Una estrategia de control avanzada que emplea técnicas inteligentes como los algoritmos genéticos o las redes neuronales. A la hora de desarrollar la implementación de la estrategia, en el GICI se ha optado por una metodología MBD, y se han desgranado las distintas fases para situar al lector en el flujo de trabajo y en qué fase se centra el trabajo. De este modo, tomando como punto de partida el trabajo presentado en las anteriores Jornadas de Automática Zabaljauregi et al. (2022), se han presentado el desarrollo realizado para la validación de la estrategia en tiempo real. Inicialmente en el marco de Simulink Desktop Real-Time, el cual se ejecuta en un kernel sobre el sistema operativo de la plataforma de desarrollo. En el segundo paso se ejecuta en el marco de Simulink Real-Time, el cual se ejecuta en la plataforma Speedgoat con un sistema operativo en tiempo real como es QNX Neutrino.

Tras detallar la metodología de desarrollo que se emplea, se presentan los experimentos que a realizar para realizar la validación de la estrategia en tiempo real. Los resultados de los experimentos son satisfactorios, tanto en la plataforma de desarrollo como en el speedgoat. En la plataforma de desarrollo se han llevado a cabo también experimentos en los que el lazo de control sufre de perturbaciones en la salida del lazo. El resultado de rechazo de estas perturbaciones ha sido también adecuado.

Como líneas futuras faltaría implementar la FASE 5, que consistiría en externalizar las señales entre controlador y planta, dando así el último paso en la metodología MBD. Sin embargo se debe remarcar que el código está en una fase de continua mejora y que se pretende mejorar el desempeño de la estrategia en plataformas con recursos de computación limitados.

Por último, falta por demostrar la validez de la estrategia en las FASES 2 5 con sistemas MIMO, ya que los resultados de la fase de script han sido buenos.

## Agradecimientos

Este trabajo se ha desarrollado en el marco del proyecto PID2020-120087GB-C22 financiado por el Ministerio de Cien-

cia e Innovación del Gobierno de España. (AEI / <http://dx.doi.org/10.13039/501100011033>)

## Referencias

- Bemporad, A., 2006. Model predictive control design: New trends and tools. In: Proceedings of the 45th IEEE Conference on Decision and Control. pp. 6678–6683.  
DOI: 10.1109/CDC.2006.377490
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation* 6 (2), 182–197.  
DOI: 10.1109/4235.996017
- Deb, K., Steuer, R. E., Tewari, R., Tewari, R., 2011. Bi-objective portfolio optimization using a customized hybrid nsga-ii procedure. In: Takahashi, R. H. C., Deb, K., Wanner, E. F., Greco, S. (Eds.), *Evolutionary Multi-Criterion Optimization*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 358–373.
- Harris, C. (Ed.), 1994. *Advances in Intelligent Control*. CRC Press.
- Larrea, M., Larzabal, E., Irigoyen, E., Valera, J., Dendaluze, M., 2014. Implementation and testing of a soft computing based model predictive control on an industrial controller. *Journal of Applied Logic*.  
DOI: 10.1016/j.jal.2014.11.005
- Narendra, K. S., Parthasarathy, K., 1990. Identification and control of dynamical systems using neural networks. *IEEE Trans Neural Networks* 1, 4–27.
- Nunna, K., Gautier, N., Malack, S., Kim, M., 2016. Model-based design for rapid controller prototyping of furuta pendulum: A case study using low-cost hardware. In: 2016 UKACC 11th International Conference on Control (CONTROL). pp. 1–5.  
DOI: 10.1109/CONTROL.2016.7737641
- Valera, J., Gómez, V., Irigoyen, E., Artaza, F., Larrea, M., 2012. Intelligent multi-objective nonlinear model predictive control (imo-nmpc): Towards the ‘on-line’ optimization of highly complex control problems. *Expert Systems with Applications* 39 (7), 6527 – 6540.  
DOI: 10.1016/j.eswa.2011.12.052
- Walica, D., Noskiewi, P., 2022. Application of the mil and hil simulation techniques in stewart platform control development. *Applied Sciences* 12 (5).  
URL: <https://www.mdpi.com/2076-3417/12/5/2323>  
DOI: 10.3390/app12052323
- Yusoff, Y., Ngadiman, M. S., Zain, A. M., 2011. Overview of nsga-ii for optimizing machining process parameters. *Procedia Engineering* 15, 3978–3983, cEIS 2011.  
URL: <https://www.sciencedirect.com/science/article/pii/S1877705811022466>  
DOI: <https://doi.org/10.1016/j.proeng.2011.08.745>
- Zabaljauregi, A., Alonso, A., Larrea, M., Irigoyen, E., 2022. Desarrollo de la estrategia imo-nmpc: primeros pasos para su implementación en dispositivos industriales. In: XLIII Jornadas de Automática.  
URL: <http://hdl.handle.net/2183/31387>
- Zabaljauregi, A., Alonso, A., Larrea, M., Irigoyen, E., Sanchis, J., 2023. Control of mimo systems with imo-nmpc strategy. In: 17th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2022). Springer Nature Switzerland, pp. 474–483.