XVII Simposio CEA de Control Inteligente
27-29 de junio de 2022, León

Implementación de un sistema de control predictivo inteligente para sistemas de dinámicas complejas

Asier Zabaljauregi, Aimar Alonso, Eloy Irigoyen, Mikel Larrea

Universidad de País Vasco / Euskal Herriko Unibertsitatea (UPV/EHU)

To cite this article: Zabaljauregi, A, Alonso, A., Irigoyen, E., Larrea, M., 2022. iMO-NMPC strategy development: First steps for implementation on industrial hardware. XVII Simposio CEA de Control Inteligente.

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. De este modo se pretende realizar una validación de dichas estrategias no solamente desde el punto de vista de simulación, sino acercando estos desarrollos a diferente hardware industrial. El caso de uso que se presenta y que está siendo implementado actualmente es el de la estrategia iMO-NMPC, el cual integra dentro de una estrategia de control predictivo, algoritmos evolutivos para la optimización y redes neuronales para el modelado de sistemas. La metodología que se está empleando hace uso de la plataforma de simulación MATLAB/Simulink®.

Palabras clave: NMPC, Hardware Industrial, *S-Function*, dinámicas complejas

Implementation of an intelligent predictive control strategy for systems with complex dynamics.

Abstract

This work presents the methodology used by the Intelligent Control Research Group (GICI) at UPV/EHU, for the development of intelligent control strategies and their further implementation in real time platforms. In this way, it is intended to provide validation of such strategies not only in simulation level but in several industrial devices. The use case that is being developed is the iMO-NMPC strategy which integrates predictive control strategies, evolutionary algorithms for optimization and neural networks for system modelling. The employed methodology involves the simulation platform MATLAB/Simulink ®.

Keywords: NMPC, Industrial Hardware, *S-Function*, complex dynamics

1. Introducción

En el mundo de la ingeniería existen muchos sistemas que presentan dinámicas complejas: procesos químicos, aplicaciones robóticas, vehículos aéreos, etc. Estos sistemas debido a su naturaleza no lineal, o a incertidumbres en su comportamiento, o al acoplamiento entre sus variables, u otras características que hacen compleja la tarea de diseñar un sistema de control específico, no son fácilmente controlables con técnicas de control clásicas. Adicionalmente, una linealización sobre un punto de operación tampoco basta para un control satisfactorio. Asimismo, frecuentemente la tarea de identificación y modelización del sistema a controlar, en base a un estudio matemático apoyado en las leyes que gobiernan el comportamiento de sus diferentes componentes, se ve dificultada al no tener acceso directo al mismo, por la gran cantidad de parámetros y limitaciones técnicas, o simplemente por razones económicas. Hoy en día, más que nunca, surge la necesidad de identificar y controlar estos sistemas con precisión debido a las especificaciones de fabricación y medidas de seguridad más estrictas. Una de las estrategias más empleadas en la industria es el Control Predictivo basado en Modelo (del inglés, MPC) (Camacho y Bordons, 2007b) y su extensión para sistemas no lineales (NMPC) Camacho y Bordons (2007a); algunos ejemplos aplicados recientemente (Vaneshani y Jazayeri-rad, 2011) (Veselov et al., 2019) (Schaaf, 2020) (Marchante et al., 2021).

El trabajo desarrollado y presentado en este documento se está realizando en el contexto de una novedosa estrategia en el ámbito del control predictivo no lineal denominada iMO-NMPC (*Intelligent Multi-objective Nonlinear Model Predictive Control*) (Valera García et al., 2012). Este trabajo da continuación a anteriores desarrollos realizados en esta línea (Eloy Iri-

Correos electrónicos: azabaljauregi001@ikasle.ehu.eus (Asier Zabaljauregi), aalonso198@ikasle.ehu.eus (Aimar Alonso), eloy.irigoyen@ehu.eus (Eloy Irigoyen), m.larrea@ehu.eus (Mikel Larrea)

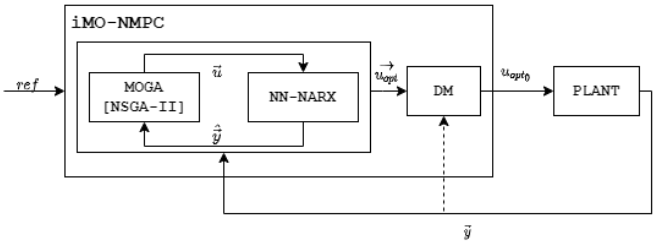


Figura 1: Esquema iMO-NMPC

goyen, 2014) (Viana et al., 2020). En este documento se presenta una evolución de los pasos dados en un esfuerzo de implementar esta técnica en una plataforma de simulación como MATLAB/ Simulink ®.

2. Desarrollo

La estrategia de control inteligente iMO-NMPC integra una base de MPC, paradigmas procedentes del ámbito de la Computación Inteligente como son los algoritmos evolutivos, redes neuronales artificiales (RNA) para la representación del modelo del sistema a controlar (Hornik et al., 1989) y un *Decision Maker* que da soporte a la etapa de optimización. Mediante la propuesta aportada en este trabajo se pretende abordar el control de procesos industriales no lineales sujetos a perturbaciones, con complejas dinámicas acopladas y presencia de varios objetivos contrapuestos. Un primer paso en el desarrollo de este trabajo es llegar a comprender la función de cada componente de dicha estrategia. La estrategia iMO-NMPC queda representada en el esquema del funcionamiento general de esta técnica en la Figura 1. De forma resumida, su funcionamiento general es el siguiente: el MOGA (*Multi-objective Genetic Algorithm*) (en el caso propuesto NSGA-II (Deb et al., 2002)), genera una población de individuos conformados por H_c cromosomas, es decir el número de acciones de control a aplicar en los instantes del horizonte de control, que se aplican al modelo neuronal de la planta, extendiendo la última accion de control hasta H_p, el horizonte de predicción. Estos individuos se evalúan según unos objetivos definidos (e.g el error ($ref - y$), la variación de la acción de control $\Delta u \ldots$) y según su coste (y *crowding index* en NSGA-II) se clasificarán y se emplearán para obtener la siguiente generación mediante operadores genéticos de mutación y *crossover*. Mediante este proceso genético, a lo largo de las iteraciones, se minimizará el coste de esos objetivos. Los individuos dominantes que forman el frente de Pareto se entregan después al *Decision Maker*, para que según el contexto en el que se encuentre la planta elija una u otra solución partiendo del conocimiento de un Agente Experto. De la secuencia de control seleccionada se aplica solamente la primera acción y se vuelve a realizar la optimización iterada una vez se retroalimenta la salida de la planta real.

2.1. Aprendizaje de las herramientas de Simulink

Se ha optado por MATLAB/Simulink para el desarrollo y simulación de el sistema de control propuesto, al ser el software estándar de facto de computación numérica y prototipado de algoritmos. Simulink, herramienta de modelización visual por bloques, cuenta con bloques de llamada a *S-Functions*, que permiten al usuario programar rutinas personalizadas para su ejecución en conjunción al resto de bloques propietarios. Se ha considerado emplear esta funcionalidad para llevar a cabo la tarea de implementar la estrategia, no solo porque otorga al usuario una mayor flexibilidad a la hora de programar, sino porque bajo condiciones específicas, permite generar el código de la simulación para su posterior ejecución en plataformas de tiempo real.

Simulink cuenta con varias formas de desarrollar estas *S-Functions* o funciones especiales, sin embargo el principal detalle a tener en cuenta es lenguaje de programación; se pueden escribir las *S-Functions* de las siguientes maneras:

1. *Level 1 MATLAB S-Functions*. Una manera muy simple y limitada de programar *S-Functions* mediante *Flags* utilizando lenguaje de MATLAB. Pronto será obsoleto y desechado.
2. *Level 2 MATLAB S-Functions*. Proporciona acceso a un conjunto más amplio de la API de las *S-Functions* y admite generación de código. En esta opción también se emplea el lenguaje de programación de MATLAB. Para habilitar la generación de código, el usuario debe proveer un archivo `.tlc` (*Target Language Compiler*), donde se especifican las pautas de generación.
3. *MEX S-Functions*. Permite implementar algoritmos en formato C MEX *S-Functions* o programar *wrappers* como llamada a código ya existente escrito en *C,C++* o *Fortran*. En este caso, proveer un archivo `.tlc` para la generación de código es opcional.

Se puede encontrar más información sobre el tema en la documentación proporcionada por MathWorks. Por razones de comodidad se ha optado por emplear la tercera manera. En el proceso de aprendizaje del API, se han implementado bloques de *S-Function* como PI, PIDs con filtro y sin filtro y funciones de transferencia entre otros. Un desarrollo que vale la pena destacar, es un control DMC (*Dynamic Matrix Control*) Figura 2, como primer paso a dar en la programación del control predictivo antes de afrontar la programación del iMO-NMPC.

2.2. Estudio del algoritmo mediante scripts de MATLAB

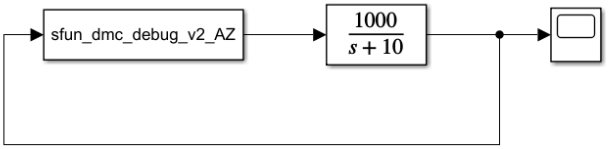
Antes de pasar a la implementación en C, se ha estudiado el comportamiento de la estrategia iMO-NMPC mediante *scripts* de MATLAB, con diferentes modelos no-lineales como *benchmark*. Los modelos en cuestión han sido empleados anteriormente en (Larrea et al., 2015) y (Larrea et al., 2010). Las ecuaciones de los modelos no lineales empleados son las siguientes:

$$y_{k+1} = \frac{1{,}5 \cdot y_k \cdot y_{k-1}}{1 + y_k^2 + y_{k-1}^2} + 0{,}7 \sin\left[0{,}5\,(y_k + y_{k-1})\right] \cdot \cos\left[0{,}5\,(y_k + y_{k-1})\right] + 1{,}2u_k \tag{1}$$

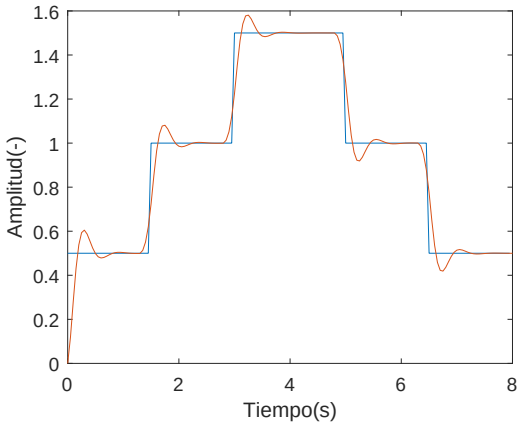
$$y_{k+1} = u_k^3 + \frac{y_k}{1 + y_k^2} \tag{2}$$

Los pasos que se reproducen dentro de la estrategia iMO-NMPC se presentan a continuación:

- Paso1: Lectura del estado actual de la simulación.



(a) Esquema de bloques para DMC en Simulink



(b) Salida del sistema y referencia

Figura 2: Modelo de prueba DMC en Simulink

- Paso2: Generación de cromosomas (acciones de control).
- Paso3: Proceso de optimización con predicciones de la RNA.
- Paso4: Selección de acción de control a aplicar.
- Paso5: Aplicar la acción de control y volver al Paso1.

Para el apartado de la optimización se ha empleado el algoritmo genético multi-objetivo `@gamultiobj` propietario de MATLAB. El `@gamultiobj` proporciona un frente de pareto con múltiples soluciones entre las que se debe seleccionar la más adecuada. El criterio empleado en este trabajo consiste en la seleccionar aquella solución cuya distancia al origen del frente de pareto sea mínima.

Como modelos de predicción se han empleado tanto modelos matemáticos inicialmente, como redes neuronales entrenadas para representar las ecuaciones (1) y (2). Una profundización sobre el empleo de redes neuronales para la identificación de dinámicas no lineales se presenta en (Jagannathan y Lewis, 1996). También se han realizado pruebas con una red neuronal MIMO para representar ambas ecuaciones.

Por otro lado, en un esfuerzo de acelerar los tiempos de ejecución y gracias a que la función `@gamultiobj` permite una entrada vectorizada, se ha modificado el script existente para evaluar vectorialmente las acciones de control (cromosomas) de todos los individuos de la población en cada iteración y para cada instante de muestreo. Se muestra en la Figura 3, la evolución de las salidas de ambos sistemas bajo control, estando los sistemas representados por una misma red neuronal empleando

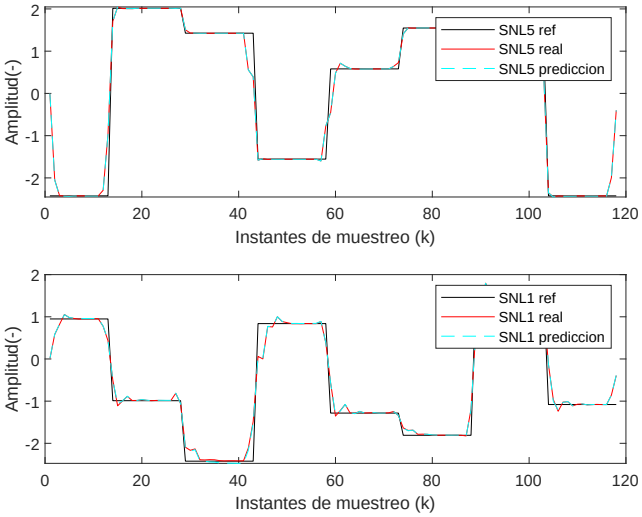


Figura 3: Evolución del control de los sistemas (1)(SNL5) y (2)(SNL1)

un horizonte de control y predicción $H_c = H_p = 4$, 200 individuos de población y un tiempo de muestreo de 0.5 s. Debido al tiempo de muestreo tan reducido, en el equipo donde se ha realizado la simulación, en cada instante de muestreo solo se llega a 30 iteraciones antes de que se corte por límite de tiempo. No obstante, se puede observar que se obtiene un resultado satisfactorio de control.

Se debe añadir en este apartado también, una de las pruebas heurísticas realizadas en la gestión de la inicialización de la población del siguiente instante de muestreo. Para inicializar los cromosomas de los individuos de la siguiente población se emplean los calculados para la población actual desplazados en un instante de muestreo tal y como se aprecia en la Figura 4. Para asignar un valor al último cromosoma de los individuos, simplemente se llama a una función que genera un número aleatorio entre los límites inferior y superior de la señal de control. Pese a que no se han realizado pruebas exhaustivas del efecto de esta inicialización, la técnica demuestra consistentemente reducir el tiempo de cómputo del control predictivo, aún siendo una reducción ligera, cuando se han realizado ejecuciones sin límite de tiempo en cada ciclo de iteración.

2.3. *Implementación en CMEX S-Function*

Finalmente, tras haber dado los pasos anteriores se expone en este apartado el desarrollo hasta el momento de la implementación en *S-Function* de iMO-NMPC. Partiendo del código de NSGA-II de Kalyanmoy Deb y ulteriores avances partiendo de (Larrea et al., 2015), se han creado archivos adicionales, para encapsular ahí la lógica necesaria para el modelo de predicción y los criterios de selección, a parte del archivo *S-Function* principal, el cual es llamado por la rutina principal de simulación en Simulink mostrada en la Figura 5. La lógica principal de la estrategia se divide en los pasos de la rutina recuadrados en azul. En el paso `mdlStart` se realiza la adjudicación de memoria, definición y asignación de variables; en el paso `mdlOutputs` se actualiza la salida con la acción de control seleccionada; finalmente en `mdlUpdate` se realiza la optimización para encontrar las soluciones del siguiente instante.

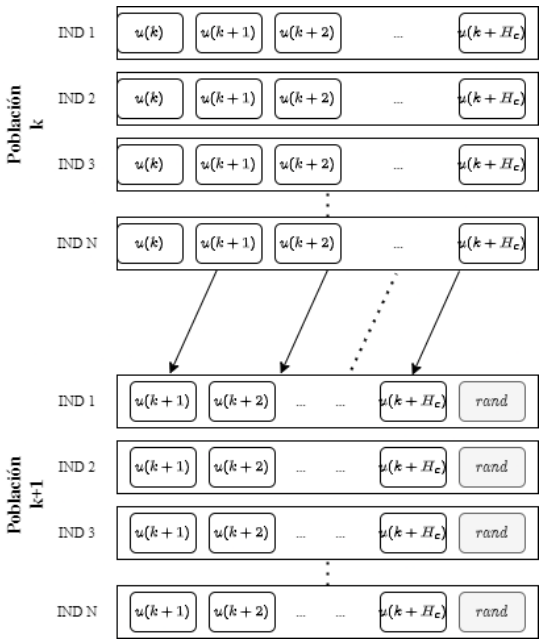


Figura 4: Esquema inicialización de individuos de la siguiente población

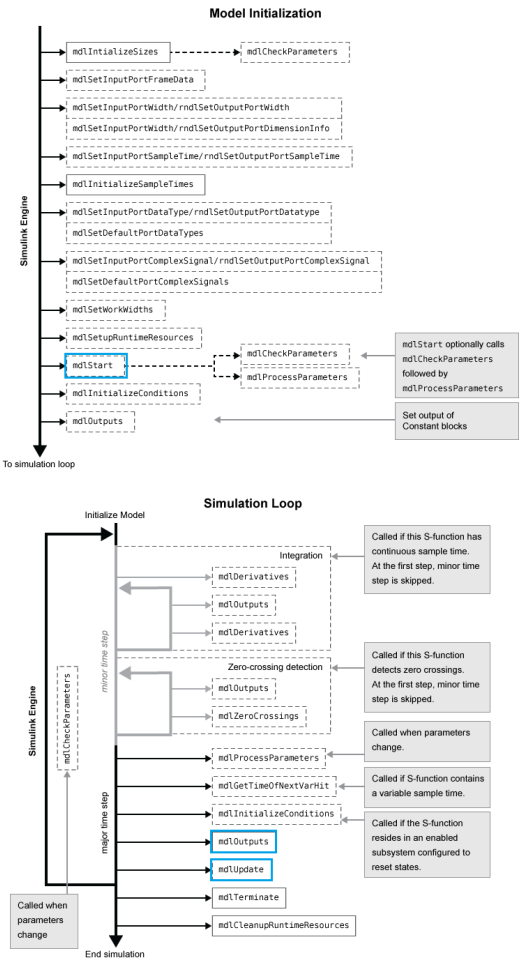
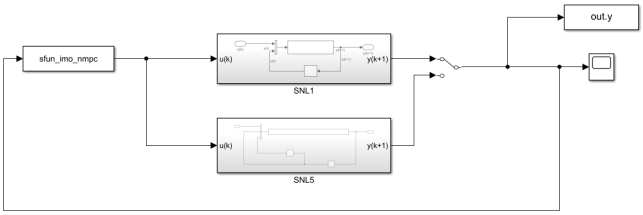
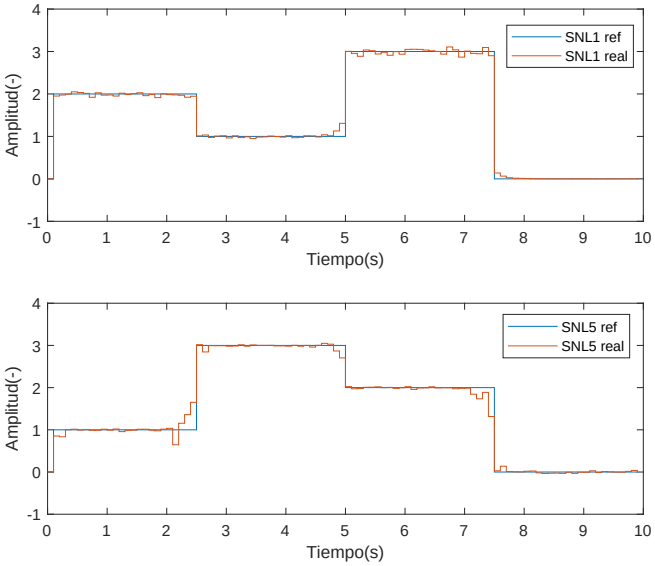


Figura 5: Interacción del Motor de Simulink con C S-Functions

En la siguiente Figura 6 se muestran las respuestas de los sistemas (2) y (1) para $H_c = 4$, $H_p = 6$, una población de 200, número de generaciones 200 y con un tiempo de muestreo de 0.1 s. En este caso el tiempo de muestreo no limita el número de iteraciones realizadas, solamente sirve para dar una escala de tiempo a la simulación. Para ambos sistemas, se ha empleando su correspondiente modelo matemático para realizar las predicciones.



(a) Esquema de bloques para iMO-NMPC *S-Function* en Simulink



(b) Salida del sistema y referencia

Figura 6: Modelo de prueba *S-Function* de iMO-NMPC

3. Próximos desarrollos y Líneas Futuras

Un desarrollo clave en el futuro es el traslado de la ejecución a una plataforma de tiempo real para una validación y experimentación adicional, similar a lo realizado en (Larrea et al., 2014). Para esta labor Simulink cuenta con aplicaciones como *Simulink Real Time* o *Simulink PLC Coder*.

En un esfuerzo de asegurar la concurrencia del software, facilitar la depuración del código y aseguramiento de la periodicidad del cómputo en cada instante de muestreo, se plantea introducir una ejecución similar a una máquina de estados con algún mecanismo de *timeout* en la optimización del algoritmo genético.

Por otro lado queda por extender la utilidad del programa a sistemas MIMO multi-objetivo. Adicionalmente queda como línea futura, la adición de diferentes lógicas de *Decision Maker*, a parte del criterio de distancia mínima implementado en el momento, que estén basadas en técnicas de *Soft Computing* tales como la lógica difusa. El método de inicialización de la pobla-

ción, mencionada anteriormente, queda por ser implementada también en la versión programada en C.

Finalmente, queda como posible desarrollo futuro el análisis y la experimentación con técnicas de optimización mas recientes, como puede el NSGA-III (Deb y Jain, 2014), una versión mejorada sobre el NSGA-II, o MOEA/D (Zhang y Li, 2007) (Zhou et al., 2012) un algoritmo evolutivo multi-objetivo basado en descomposición y operadores diferenciales.

Agradecimientos

Este trabajo se ha desarrollado en el marco del proyecto PID2020-120087GB-C22 financiado por el Ministerio de Ciencia e Innovación del Gobierno de España.
(AEI / http://dx.doi.org/10.13039/501100011033)

Referencias

Camacho, E., Bordons, C., 2007a. Nonlinear Model Predictive Control: An Introductory Review.
DOI: 10.1007/978-3-540-72699-9_1

Camacho, E. F., Bordons, C., Apr. 2007b. Model Predictive Control, 2nd Edición. Springer.

Deb, K., Jain, H., 2014. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation 18 (4), 577–601.
DOI: 10.1109/TEVC.2013.2281535

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

Eloy Irigoyen, Ekaitz Larzabal, J. J. V. M. L. A. G., 2014. Primeros resultados de un control genético predictivo sobre maqueta de helicóptero (twinrotor). Jornadas de Automática.

Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. Neural Networks 2, 359–366.

Jagannathan, S., Lewis, F., 1996. Identification of nonlinear dynamical systems using multilayered neural networks. Automatica 32 (12), 1707 – 1712.
DOI: 10.1016/S0005-1098(96)80007-0

Larrea, M., Irigoyen, E., Gómez, V., 2010. Adding nonlinear system dynamics to levenberg-marquardt algorithm for neural network control. En: ICANN 2010. Vol. 6354 of Lecture Notes in Computer Science. pp. 352–357.
DOI: 10.1007/978-3-642-15825-4_47

Larrea, M., Larzabal, E., Irigoyen, E., García, J. J. V., Dendaluce, M., 2015. Implementation and testing of a soft computing based model predictive control on an industrial controller. J. Appl. Log. 13, 114–125.

Larrea, M., Larzabal, E., Irigoyen, E., Valera, J., Dendaluce, 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

Marchante, G., Acosta, A., González, A., Zamarreño, J., Álvarez, V., abr. 2021. Comfort constraints evaluation in predictive controller for energy efficiency. RIAI Journal 18 (2), 146–159.
DOI: 10.4995/riai.2020.13937

Schaaf, M., jul. 2020. Hybrid model predictive control of a gravity separator with intermittent product extraction. RIAI Journal 17 (3), 318–328.
DOI: 10.4995/riai.2020.11957

Valera García, J. J., Gómez Garay, V., Irigoyen Gordo, E., Artaza Fano, F., Larrea Sukia, 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: https://doi.org/10.1016/j.eswa.2011.12.052

Vaneshani, S., Jazayeri-rad, H., 2011. Nonlinear control of a chemical plant employing a combination of fuzzy logic and particle swarm optimization techniques.

Veselov, G., Sklyarov, A., Chávez, J. V., 2019. Non-linear control of a tracked robot. En: 2019 IEEE 17th International Conference on Industrial Informatics (INDIN). Vol. 1. pp. 641–646.
DOI: 10.1109/INDIN41052.2019.8972253

Viana, K., Larrea, M., Irigoyen, E., Diez, M., Zubizarreta, A., 2020. MIMO neural models for a twin-rotor platform: Comparison between mathematical simulations and real experiments. En: Álvaro Herrero, et al. (Eds.), 15th Int. Conf. SOCO'20, Burgos, Spain, 2020. Vol. 1268 of Advances in Intelligent Systems and Computing. Springer, pp. 407–417.
DOI: 10.1007/978-3-030-57802-2_39

Zhang, Q., Li, H., 2007. Moea/d: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation 11 (6), 712–731.
DOI: 10.1109/TEVC.2007.892759

Zhou, A., Zhang, Q., Zhang, G., 06 2012. A multiobjective evolutionary algorithm based on decomposition and probability model. pp. 1–8.
DOI: 10.1109/CEC.2012.6252954