

CONTROL PREDICTIVO BASADO EN DATOS

Vicente Simón Gómez, David Muñoz de la Peña, Teodoro Álamo

Dep. Ingeniería de Sistemas y Automática, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla
Camino de los Descubrimientos SN, 41092, Sevilla

Resumen

En esta comunicación se presenta la implementación de un controlador predictivo basado en datos (DbPC) que permite resolver el problema de regulación de cualquier estado objetivo sobre un sistema cuyo modelo es desconocido. A diferencia de otras estrategias de control, ésta no usa un modelo, sino una base de datos para abordar el problema de regulación. En esta base de datos se almacena un histórico de las trayectorias llevadas a cabo por el sistema estudiado. Por último, el DbPC se implementa sobre un sistema real y con modelo conocido para un análisis y validación de los resultados obtenidos que se comparan con las diferentes estrategias de control que se han utilizado para generar la base de datos de este ejemplo.

Palabras Clave: Control predictivo, base de datos, problema de optimización

1. INTRODUCCIÓN

Se presenta el objetivo de resolver un problema de control para un sistema del que no es posible obtener un modelo aproximado o preciso, ya que dicho sistema presenta una dinámica difícil de identificar. La solución que se presenta es la obtención de un controlador predictivo basado en datos (DbPC) que usa una base de datos de dicho sistema en lugar del propio modelo. En esta base de datos está recogido el comportamiento del sistema a lo largo de un periodo de tiempo suficientemente grande. Este histórico de trayectorias junto con la resolución de un problema de optimización nos ayuda a decidir los valores de los actuadores sin necesidad de un modelo.

Como paso previo a la implementación del control predictivo basado en datos sobre un sistema real en la industria cuyo modelo se desconozca se valida el algoritmo sobre un sistema académico del que ha sido previamente estudiado su modelo y, por tanto, es

conocido. Este sistema es una planta de cuatro tanques [1] de la que se genera una base de datos como punto de partida para estudiar la bondad de predicción del DbPC.

2. FORMULACIÓN DEL PROBLEMA

Consideramos el siguiente modelo no lineal discreto

$$x_{k+1} = f(x_k, u_k) \quad (1)$$

donde $x_k \in \mathbb{R}^n$ es la salida del sistema y $u_k \in \mathbb{R}^m$ es la entrada en el tiempo de muestreo k respectivamente y se supone desconocido f , el modelo de un sistema.

Se quiere diseñar una ley de control que nos permita resolver el problema de regulación para alcanzar un estado objetivo arbitrario x^{ref*} .

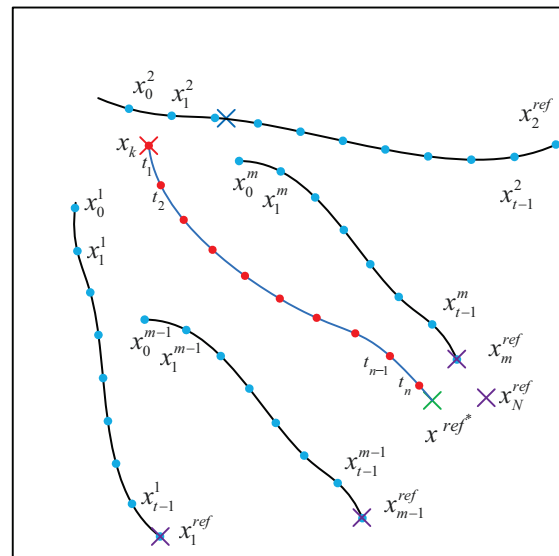


Fig 1. Problema de regulación en \mathbb{R}^2

Como desconocemos el modelo del sistema se usa una base de datos que nos ayudará a obtener los estados futuros.

En esta base de datos se almacenan el conjunto de estados x_k^i, u_k^i, x_i^{ref} , salidas, entradas y estado objetivo respectivamente, correspondiente a un conjunto de m diferentes trayectorias llevadas a cabo por el sistema durante un periodo de tiempo t . Así pues, se llama BD a todo el conjunto de estados que forma la base de datos. Este histórico de trayectorias en el ejemplo práctico es el histórico del comportamiento en bucle cerrado de varias estrategias de control que resuelven el problema de regulación desde diferentes estados iniciales a diferentes estados de equilibrio objetivo (fig. 1).

La bondad de la predicción de los estados futuros de la solución propuesta dependerá de los parámetros que hay que definir en la base de datos. Éstos son t, m, N siendo $N = t \cdot m$, t número de estados en cada trayectoria, m número de trayectorias, N número total de estados almacenados. Para entender mejor estos parámetros se representa la tabla 1 donde se recoge como están almacenados el conjunto de trayectorias y estados del sistema en la base de datos que se recoge en este estudio.

Una vez detallado la obtención de los datos necesarios para la solución (tabla 1), en el siguiente apartado se pasa a explicar las herramientas que nos permiten decidir las acciones de control futuras.

	i	x_i^{ref}	x_k	u_k
Trayectoria $i = 1$	1	x_1^{ref}	x_0^1	u_0^1
	1	x_1^{ref}	x_1^1	u_1^1
	\vdots	\vdots	\vdots	\vdots
	1	x_1^{ref}	x_t^1	u_t^1
	m	x_m^{ref}	x_0^m	u_0^m
Trayectoria $i = m$	m	x_m^{ref}	x_1^m	u_1^m
	\vdots	\vdots	\vdots	\vdots
	m	x_m^{ref}	x_t^m	u_t^m

Tabla 1. Estructura de la base de datos

3. SOLUCIÓN PROPUESTA

3.1 ESQUEMA DEL ALGORITMO

La solución que se propone se define como

$$u_k = DbPC(x_k, x^{ref*}, BD, ID_i) \quad (2)$$

A diferencia de un sistema de control donde sólo se tiene como entrada del sistema la salida actual x_k y las referencias x^{ref*} , en la solución propuesta además es necesario usar una base de datos BD y un indicador de la bondad de las diferentes trayectorias de nuestra base de datos respecto una serie de parámetro que definen el problema a optimizar que se define como ID_i .

Asimismo, se usa un algoritmo que permite obtener el valor de los actuadores u_k resolviendo un problema de optimización, minimizando una función de costes y aplicando las restricciones necesarias.

3.2 PRINCIPIOS DEL DBPC

El control predictivo mediante base de datos es una estrategia de control que nos permite resolver el problema de regulación a partir de los datos proporcionados por trayectorias almacenadas en una base de datos.

En primer lugar, se parte de un estado inicial $x_0 \in \mathbb{R}^n$ siendo el objetivo alcanzar el estado objetivo que hemos definido como $x^{ref*} \in \mathbb{R}^n$. Luego, el estado actual, que parte del estado inicial, se actualiza cada instante de muestreo con el objetivo de obtener los valores óptimos de los actuadores y alcanzar el valor de las referencias deseadas en nuestro problema de regulación.

El algoritmo sobre el que se ha implementado el controlador para cada instante de muestreo es el siguiente (fig. 2):

En una primera fase para la resolución del problema se tiene como objetivo encontrar dos envolturas convexas para llegar a $x_k \in S_1 \cup x^{ref*} \in S_2$.

En primer lugar, encontrar un conjunto de estados de la base de datos que formen una envoltura convexa, *feasible solution set (FSS)*, llamados estados candidatos x_i^c que contenga al estado actual $x_k \in S_1$. En segundo lugar, los estados de referencia x_i^{refc} de aquellas trayectorias que contienen los estados

candidatos formen otra envoltura convexa que contenga a x^{ref^*} .

Se define un estado candidato x_i^c aquel estado que cumpla las siguientes condiciones.

- a. Forma parte de la trayectoria candidata x_i^c
- b. La distancia entre x_i^c y x_k ha de ser menor que ε
- c. x_i^c es el estado más cercano a x_k entre los estados de su trayectoria

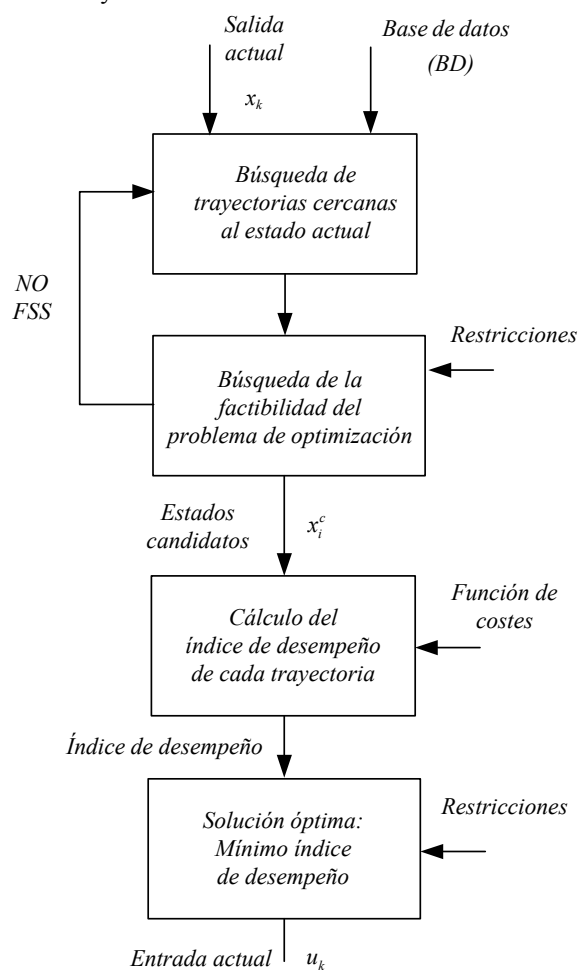


Fig 2. Esquema del algoritmo del DbPC

Asimismo se define una trayectoria candidata \vec{x}_i^c como el vector de estados formado por x_r^c y los T estados siguientes de la base de datos donde $x_r^c = x_i^c$ y T es el horizonte de predicción.

$$\vec{x}_i^c = \begin{bmatrix} x_r^c \\ x_{r+1}^c \\ \vdots \\ x_{r+T-1}^c \\ x_{r+T}^c \end{bmatrix} \quad \forall i = 1, \dots, nc \quad (3)$$

Una vez obtenidas las trayectorias candidatas se calcula el índice de desempeño asociado a cada una. El índice de desempeño se calcula como:

$$ID(t) = \int_0^T (J_i - J^*) d\tau \quad (4)$$

Siendo T el horizonte de predicción y la función J_i el coste de etapa para el sistema en estudio y J^* el coste correspondiente al estado objetivo. Señalar que este índice de desempeño se calcula en línea y no se encuentra almacenado en la base de datos.

A continuación se resuelve el problema de optimización y se calculan los diferentes pesos α_i que ponderan a cada estado candidato. Luego, se obtiene el valor de los actuadores u_k , la primera componente de los controladores futuros.

$$u_k = \sum_{i=1}^{nc} \alpha_i \cdot u_k^i \quad (5)$$

Y por último, se aplica la acción de los actuadores sobre el sistema y se actualiza tanto el valor del estado actual como el de los actuadores en cada periodo de muestreo t .

Sin embargo, no siempre es posible cumplir $x_k \in S_1 \cup x^{ref^*} \in S_2$ ya que existen situaciones, como la baja densidad de la base de datos o que x_k o x^{ref^*} se encuentre cerca de la región de los límites del espacio de trabajo del sistema, que impiden cumplir dicha condición. Para estos casos en este estudio se opta por tomar el valor de los actuadores de la trayectoria más cercana.

3.3 SOLUCIÓN ÓPTIMA: MÍNIMO ÍNDICE DE DESEMPEÑO

Supuesto conocidas las trayectorias candidatas se resuelve el problema de optimización para poder obtener los diferentes pesos α_i que ponderan a cada estado candidato. Para ello se minimiza la siguiente función objetivo con las siguientes restricciones.

$$\min \sum_{i=1}^{nc} \alpha_i ID_i \quad (6)$$

$$\text{s.a} \quad \sum_{i=1}^{nc} \alpha_i = 1 \quad (7)$$

$$\sum_{i=1}^{nc} \alpha_i x_i^c = x_k \quad (8)$$

$$\sum_{i=1}^{nc} \alpha_i x_i^{ref^c} = x^{ref^*} \quad (9)$$

$$\alpha_i \geq 0 \tag{10}$$

$$\phi_i \leq \varepsilon \tag{11}$$

$$\phi_i = \|x_i^c - x_k\| \tag{12}$$

$$\forall i = 1, 2, \dots, nc$$

$$x_i^c, x_k, x_i^{refc}, x^{ref*} \in \mathbb{R}^n$$

3.4 BÚSQUEDA DE ENVOLTURA CONVEXA

3.4.1 Búsqueda de la factibilidad del problema de optimización. Obtención de estados candidatos

Como paso previo a la resolución del problema de optimización se debe encontrar el conjunto de estados candidatos x_k^c . Este conjunto x_k^c se encuentra a una distancia ϕ_i que distará como máximo ε .

$$\phi_i \leq \varepsilon \quad \forall i = 1, 2, \dots, nc \quad \phi_i = \|x_i^c - x_k\| \tag{13}$$

Asimismo se intenta encontrar que la envoltura convexa contenga a x_k y x^{ref*} , $x_k \in S_1 \cup x^{ref*} \in S_2$.

En primer lugar se buscarán las trayectorias que cumplan que tanto sus estados de referencia x_i^{refc} formen una envoltura convexa que contenga a x^{ref*} , como que los estados candidatos x_k^c a su vez formen otra envoltura convexa que contenga a x_k .

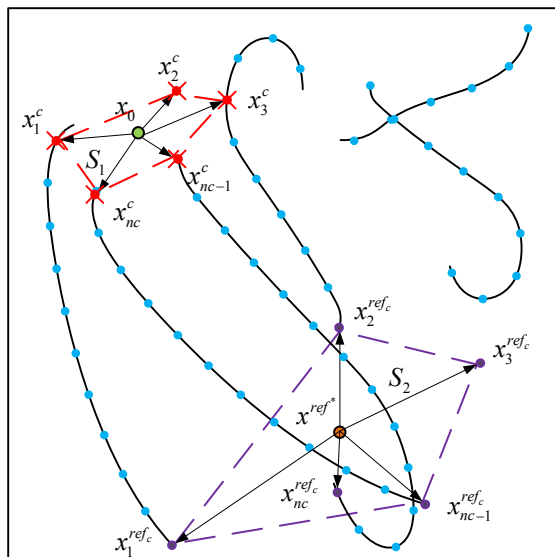


Fig 3. Representación en \mathbb{R}^2 de la búsqueda de trayectorias candidatas

De esta forma, la principal dificultad en encontrar las envolturas convexas radica en calcular el parámetro ε en cada instante de t .

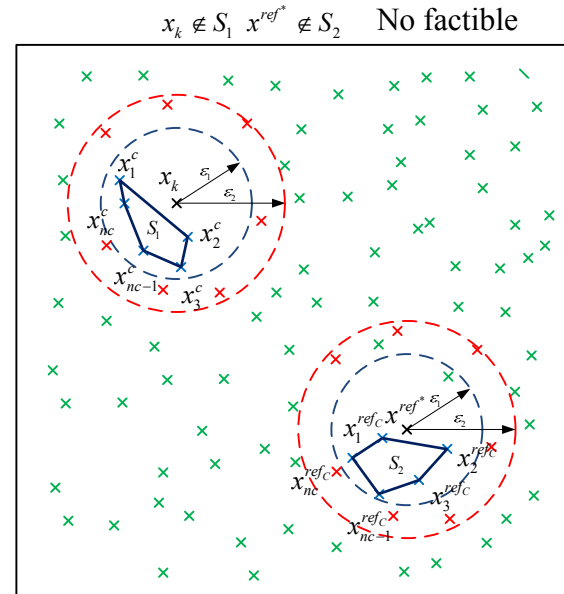


Fig 4. Representación en \mathbb{R}^2 de la búsqueda de una envoltura convexa donde $x_k \notin S_1$ $x^{ref*} \notin S_2$

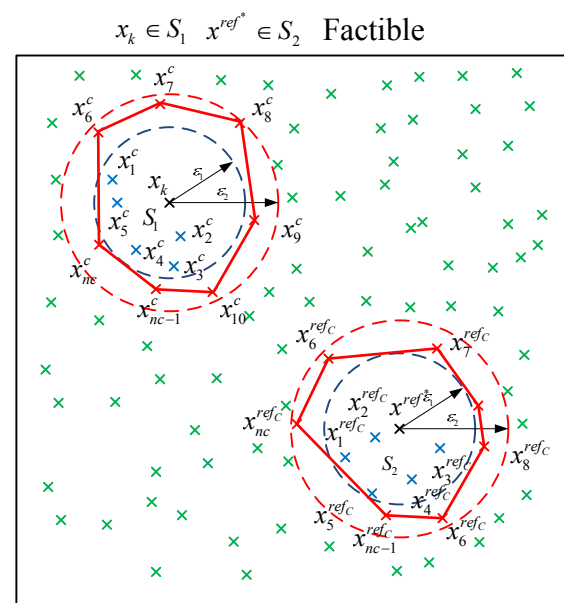


Fig 5. Representación en \mathbb{R}^2 de la búsqueda de una envoltura convexa donde $x_k \in S_1$ $x^{ref*} \in S_2$

Sin embargo, no siempre es posible encontrar una envoltura convexa que contenga al estado actual x_k y a x^{ref*} , ya que existen situaciones, como la baja densidad de la base de datos o que tanto x_k como

x^{ref^*} se encuentren cerca de la región de los límites del espacio de trabajo del sistema, que impiden cumplir dicha condición. En dichos casos en la solución propuesta se opta por tomar u_k de la trayectoria más próxima al estado actual, así como que x^{ref^c} de dicha trayectoria esté cerca de x^{ref^*} .

3.2.2 Elección de ε

Para determinar ε se utiliza el siguiente algoritmo. En primer lugar, resolvemos un problema de aproximación haciendo ε el doble cuando el problema de optimización no sea factible. Esto hace que se converja a una solución de forma rápida, no obstante usando sólo este método no se proporciona una solución óptima, ya que se quiere usar una cantidad lo más pequeña posible de estados candidatos, pero que solucione nuestro problema de optimización sin que tenga una carga computacional elevada.

Por ello tratamos de obtener una solución óptima aplicando el método de la bisección para hallar aquella solución factible con menor número de estados candidatos. El método de la bisección divide aquella ε mayor no factible y aquella menor factible, en la fig. 10 A y B respectivamente, de manera iterativa hasta que $B - C \leq Tol$.

También cabe la posibilidad en el método de las aproximaciones $\varepsilon \geq \varepsilon_{max}$, es decir, buscaría estados candidatos para la convexidad fuera del espacio de trabajo. ε_{max} es un parámetro que se fijará dependiendo de los límites físicos del sistema.

Para el ejemplo práctico, desarrollado en el apartado 4, ε_0 y Tol se han calculado de manera experimental para que el número de iteraciones en el caso de la aproximación N_{iter}^{approx} y de la bisección $N_{iter}^{bi sec}$ no sea demasiado grande y no tenga un coste computacional.

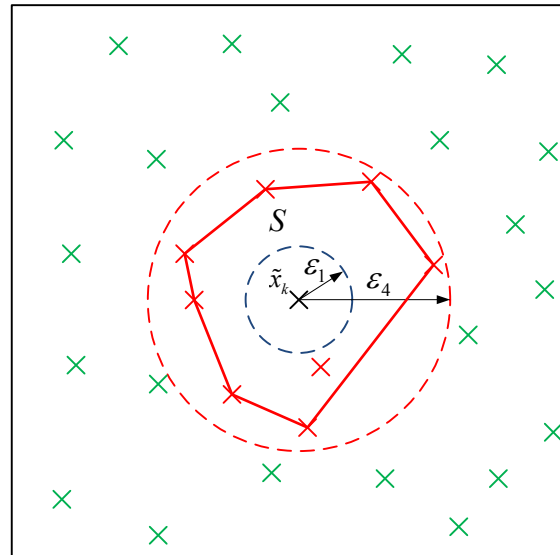


Fig 6. Optimización del parámetro ε

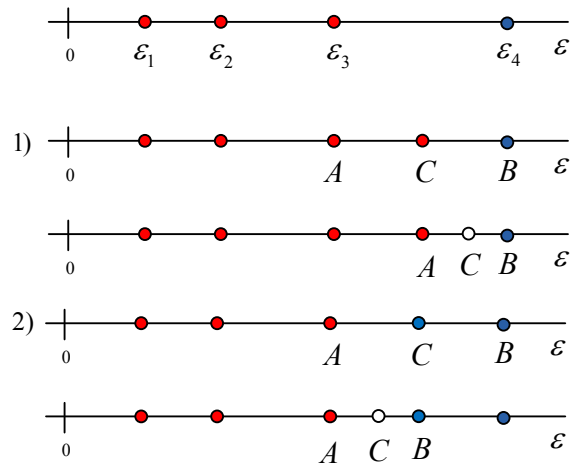


Fig 7. Método de la bisección

3.5 DIFERENCIAS BÁSICAS ENTRE MPC Y DBPC

Tanto el MPC como el DbPC comparten una misma estrategia de control (fig 8), esto es, ambos controladores predictivos toman como entrada las referencias y las medidas de los sensores del sistema a controlar y genera un salida que son los valores de los actuadores del sistema.

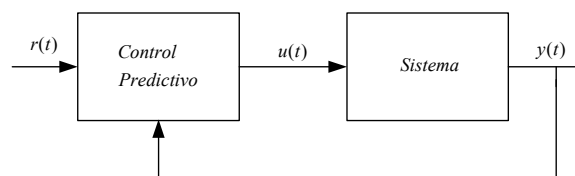


Fig 8. Esquema de control de controlador predictivo

Asimismo existen ideas que se presentan que la familia de controladores predictivos que comparten tanto el MPC como el DbPC. Éstas son el cálculo de las señales de control minimizando una cierta función objetivo y el uso de una estrategia deslizante, de forma que en cada instante el horizonte se va desplazando hacia el futuro, repitiendo el cálculo en cada instante de muestreo.

En la fig. 9,10 se muestran los dos esquemas de control de ambos controladores. La diferencia más notable es que el controlador MPC hace uso del modelo del sistema, mientras que el DbPC de una base de datos. Del mismo modo el MPC utiliza el modelo para predecir la salida del proceso a lo largo de un intervalo de tiempo de muestreo futuro y toma normalmente sólo la primera componente de los controladores futuros como valor en el actuador del sistema. Mientras tanto el DbPC almacena en su base de datos las trayectorias de las cuales toma una solución óptima dependiendo del estado actual y de la referencia que queramos alcanzar. El problema principal del DbPC es qué conjunto de trayectorias se ha de tomar bajo estudio para obtener un valor de actuadores óptimos.

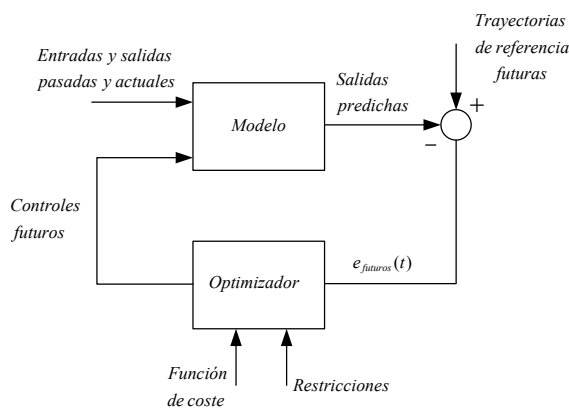


Fig 9. Esquema de MPC

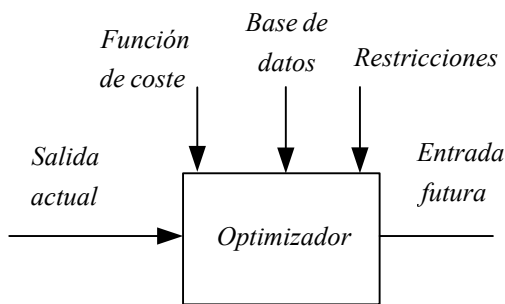


Fig 10. Esquema de DbPC

4. EJEMPLO PRÁCTICO

En el siguiente apartado se aplicará el estudio realizado del DbPC sobre una planta de cuatro tanques que se describe a continuación y se realizará un análisis comparando los resultados obtenidos por el DbPC y diferentes esquemas de control.

4.1 DESCRIPCIÓN DE LA PLANTA [1]

La planta de los cuatro tanques es una implementación de la planta propuesta por Karl H. Johansson en el artículo titulado “The quadruple-tank process” [3], publicado en la revista *IEEE Transactions on Control Systems Technology*, vol 8 (2000).

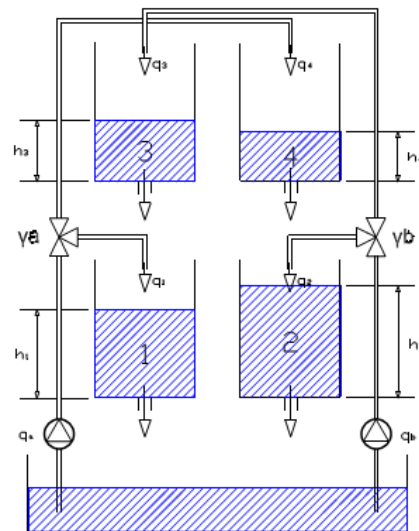


Fig 11. Esquema de la planta de Johansson

4.2 BASE DE DATOS UTILIZADA

En la base de datos se recoge un histórico de trayectorias que parten desde estados aleatorios y convergen a estados de equilibrio aleatorios, estados objetivos. Los estados iniciales aleatorios son $h_i \in \mathbb{R}^4$ siendo $h_i = \{h_1, h_2, h_3, h_4\}$ que alcanzan estados de equilibrios aleatorios $h_i^{ref} \in \mathbb{R}^4$ siendo $h_i^{ref} = \{h_1^{ref}, h_2^{ref}, h_3^{ref}, h_4^{ref}\}$.

Una vez se tiene tanto los estados iniciales y objetivos se pasa a almacenar los históricos del comportamiento en bucle cerrado de varios controladores. Esto se traduce en generar diferentes trayectorias sobre el modelo aplicado resolviendo el problema de regulación con dos controladores diferentes: PI (proporcional integrativo) y LQR

(linear-quadratic regulator) de esta forma el conjunto de trayectorias generadas en la base de datos contengan la mitad de las trayectorias contraladas por un PI y la otra mitad por un LQR.

Para minimizar el tamaño de la base de datos se opta por almacenar sólo el régimen transitorio de las trayectorias. Por lo que sólo se almacena la parte de la trayectoria que diste entre h_i^{ref} y h_i más del tres por ciento.

4.3 EVALUCIÓN DE RESULTADOS

Para conocer la bondad del DbPC es necesario el cálculo del índice de desempeño ID (4).

Los criterios de evaluación [1] en los que se basa este ejemplo son:

$$J_i(h, q, c, p) = \underbrace{q_a^2 + c \cdot q_b^2}_{J_{ce}} + p \underbrace{\frac{V_{min}}{A(h_1 + h_2)}}_{J_{va}} \quad (14)$$

Donde J_{ce} es el coste de operación de la minimización del consumo energético de la planta y J_{va} es el coste de operación del volumen acumulado en los depósitos inferiores.

4.4 ANÁLISIS/VALIDACIÓN DE RESULTADOS

En este análisis se compara el comportamiento que presenta el DbPC frente a varios controladores (PI, LQR). Hay que señalar que aunque los controladores PI y LQR son los mismos que se han usado para la creación de la base de datos, en las simulaciones no se hace uso de la base de datos para obtener la trayectoria de éstos. Así pues, se utiliza el esquema de control que se utilizó para la creación de la base de datos, pero para la obtención de las trayectorias de los controladores PI y LQR.

4.4.1 Optimización de la base de datos

Con el objetivo de estudiar la dimensión óptima de la base de datos que podría servir para alcanzar cualquier estado objetivo en un problema de regulación en la planta de cuatro tanques, se realiza la siguiente serie de experimentos.

En primer lugar, se crean bases de datos de densidad diferentes, esto es, dimensiones de bases de datos comprendidas entre diez y siete mil quinientas trayectorias, almacenando las trayectorias como en la tabla 1. En segundo lugar, se toman cien estados

iniciales aleatorios que convergen a otros cien estados de equilibrio aleatorios. A continuación se resuelve el problema de regulación de esos cien puntos con el controlador DbPC haciendo uso de las diferentes bases de datos. Y por último, se calculan los valores medios de índice de desempeño, coste computacional y porcentaje de factibilidad y se comparan para extraer conclusiones.

Ante los resultados obtenidos en la fig. 12 se concluye que a mayor densidad nos encontramos con un menor índice de desempeño, mayor porcentaje de soluciones factibles, pero mayor coste computacional. También se observa que experimentos donde no se llega a encontrar puntos de trayectorias candidatas convexas de modo que no podemos alcanzar una solución factible. En nuestro caso ocurre para bases de datos que almacenan menos de alrededor de cien trayectorias aleatorias.

En un segundo análisis comparamos estos resultados con la media de los índices de desempeño usando un controlador LQR o PI aplicando sobre la misma batería de ensayos. Estos parten de cien puntos aleatorios y convergen a otros cien puntos de equilibrio aleatorios.

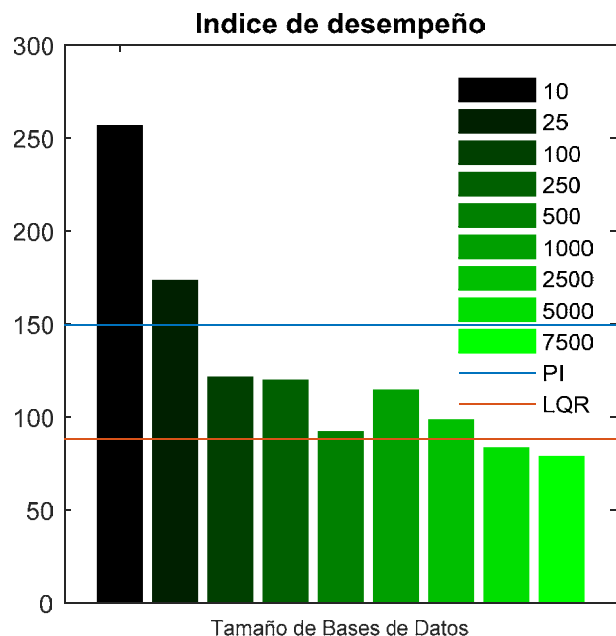


Fig 12. Análisis de dimensión de la base de datos óptima: índice de desempeño

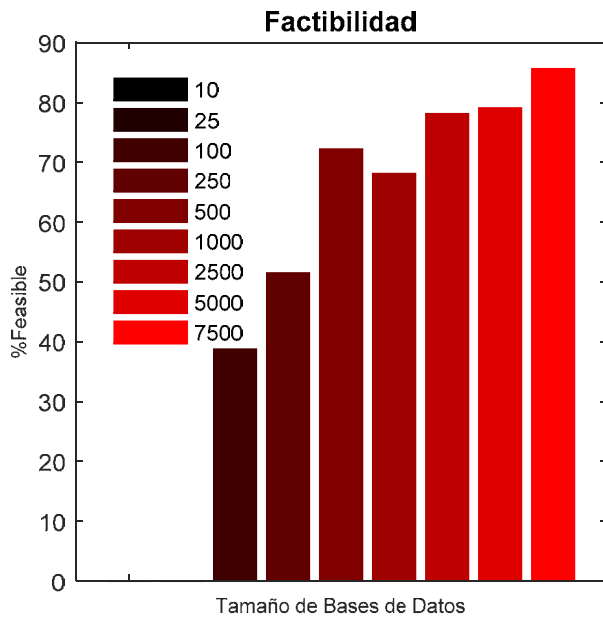


Fig 13. Análisis de dimensión de la base de datos óptima: porcentaje de factibilidad

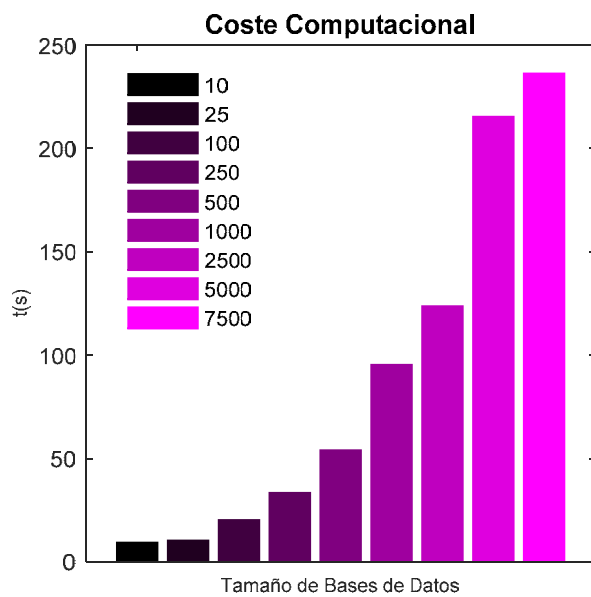


Fig 14. Análisis de dimensión de la base de datos óptima: coste computacional

4.4.2 Ejemplo práctico. Concurso de la CEA 2014

Para mostrar un ejemplo se aplica en el concurso de la CEA 2014 el controlador DbPC. Se utiliza una base de cinco mil trayectorias generadas de modo aleatorio para calcular las diferentes consignas que indica la prueba. Ya que este histórico de trayectorias se han generado en base a un control en bucle cerrado de un controlador PI y LQR, se compara el DbPC con controladores PI y LQR diseñados para cada estado referencia del concurso.

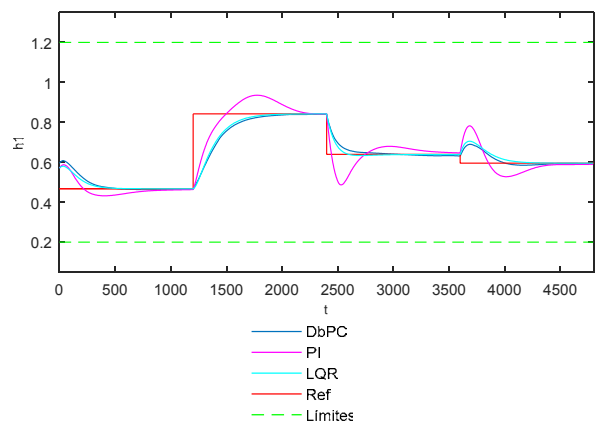


Fig 15. Comparativa de h_1 del concurso de la CEA usando un controlador DbPC, PI y LQR

Agradecimientos

Trabajo financiado por el Ministerio de Economía y Competitividad a través del proyecto Estimación y Optimización Dinámica de la Eficiencia en Infraestructuras Críticas (DPI2013-48243-C2-2-R).

Referencias

- [1] Concurso en Ingeniería de Control 2014. “Operación óptima de la planta de los 4 tanques” (CEA). Organizado por el Grupo Temático en Ingeniería de Control
- [2] V. Simón “Control Predictivo Basado en Datos. Aplicación a un Sistema de Cuatro Tanques” (TFG)
- [3] K.H. Johansson “The quadruple-tank process” *IEEE Transactions on Control Systems Technology*, vol 8 (2000)