

MODELO BASADO EN OPTIMIZACIÓN Y BASES DE 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 aborda el problema de la identificación de sistemas por medio de la implementación de un modelo (MBD) basado en optimización y que usa una base de datos para predecir los estados futuros de un sistema. Este modelo predice un estado futuro a partir del comportamiento de los estados pasados recogidos en una base de datos resolviendo un problema de optimización. Esta base de datos recoge un histórico de las trayectorias llevadas a cabo por el sistema. Por último, se estudia el MBD sobre un sistema real y conocido analizándose luego los resultados en bucle abierto y bucle cerrado con las ecuaciones diferenciales del sistema.

Palabras Clave: Identificación de sistemas, base de datos, problema de optimización

1 INTRODUCCIÓN

Actualmente existen diferentes sistemas en la industria que presentan una dinámica difícil de modelar o simplemente no se dispone del modelo, pero que necesitan ser controlados. Así pues, surge un problema doble, ya que el modelo es necesario para probar la bondad de un controlador en el caso de que no podamos trabajar sobre el sistema real y además se vuelve imprescindible para resolución del problema de control, exceptuando controladores basados en datos como los PID o el DbPC [2].

De esta forma se nos presenta el objetivo de identificar un modelo de un sistema no lineal que presenta una dinámica difícil de modelar para posteriormente resolver un problema de control. La solución que se presenta es la obtención de un modelo basado en optimización y bases de datos (MBD). Este modelo predice un estado futuro a partir del comportamiento de los estados pasados recogidos en una base de datos.

Dicha base de datos almacena un conjunto de trayectorias del sistema a lo largo de un periodo de tiempo suficientemente grande que nos permita obtener los valores de los estados futuros sin necesidad de las ecuaciones que rigen el comportamiento del sistema.

Con el fin de estudiar la bondad de predicción de la solución propuesta se toma un sistema no lineal del que conocemos su modelo para poder comparar los comportamientos del sistema ante diferentes experimentos con el MBD. Dicho sistema se trata de una planta de cuatro tanques [3] de la que se genera una base de datos que nos permita obtener la respuesta del sistema. En esta base de datos se encuentran almacenados un conjunto de trayectorias que parten y convergen a estados aleatorios.

Asimismo se evalúan y analizan los diferentes parámetros que definen nuestro MBD, en especial la densidad de la base de datos y se realiza una batería de pruebas sobre el MBD para comparar dichos resultados con los obtenidos por las ecuaciones no lineales de la planta de los cuatro tanques. Por último, también se presenta el control de dicha planta con el MBD y el modelo del sistema.

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.

Así pues, se quiere obtener el estado futuro de un sistema a partir del estado y la acción de control actual y la información disponible en la base de datos de trayectorias pasadas. Esta base de datos nos ayudará

hacer esta predicción en lugar del modelo del sistema que se desconoce.

En esta base de datos se almacenan el conjunto de estados \tilde{x}_k^i , donde $\tilde{x}_k^i = [x_k^i, u_k^i]$, salidas y entradas 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.

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.

i	x_k	u_k	
Trayectoria $i = 1$	1	x_0^1	u_0^1
	1	x_1^1	u_1^1
	\vdots	\vdots	\vdots
	1	x_t^1	u_t^1
Trayectoria $i = m$	m	x_0^m	u_0^m
	m	x_1^m	u_1^m
	\vdots	\vdots	\vdots
	m	x_t^m	u_t^m

Tabla 1. Estructura de las base de datos

Una vez detallado la obtención de los datos necesarios para la solución en el siguiente apartado se pasa a explicar las herramientas que nos permite predecir los estados futuros.

3. SOLUCIÓN PROPUESTA

3.1 ESQUEMA DEL ALGORITMO

La solución que se propone se define como

$$x_{k+1} = MBD(x_k, u_k, BD) \quad (2)$$

Los datos de entrada son los mismos, sólo que depende además de los datos de la base de datos, a diferencia de un modelo que se rige por ecuaciones diferenciales donde sólo se tiene como entrada del sistema el estado actual. Asimismo, se usa un algoritmo que permite obtener los estados futuros resolviendo un problema de optimización, minimizando una función y aplicando las restricciones necesarias.

En una primera fase para la resolución del problema se tiene como objetivo encontrar un conjunto de estados de la base de datos que formen una envoltura convexa, *feasible solution set (FSS)*, y que contenga al estado actual $\tilde{x}_k \in S$, siendo $\tilde{x}_k = [x_k, u_k]$. A esta serie de estados que forman una envoltura convexa S y que hacen que la función a minimizar sea factible los llamaremos estados candidatos $x_i^c \quad \forall i = 1, \dots, nc$ siendo nc el número de estados candidatos. Así como a x_i^{c+} al estado candidato sucesor, esto es, el siguiente estado en la base de datos. Estos estados nos ayudarán a predecir el estado futuro x_{k+1} en una segunda fase del algoritmo.

Sin embargo, no siempre es posible encontrar una envoltura convexa que contenga al estado actual \tilde{x}_k , ya que existen situaciones, como la baja densidad de la base de datos o que \tilde{x}_k se encuentre cerca de la región de los límites del espacio de trabajo del sistema, que impiden cumplir dicha condición. De esta forma se usarán algoritmos distintos para el caso que $\tilde{x}_k \in S$ (caso A) o bien $\tilde{x}_k \notin S$ (caso B).

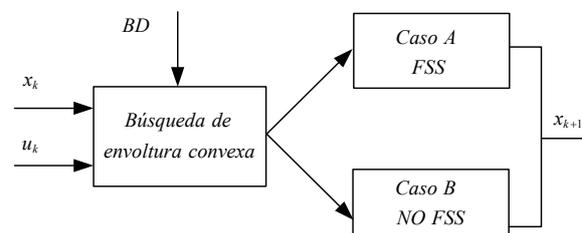


Fig 1. Esquema del MBD

En la segunda fase, caso A o B consiste en asignar unos pesos α_i a cada uno de los nc estados candidatos resolviendo el problema de optimización que obligue a que la suma de pesos por los estados candidatos futuros nos proporcione el estado futuro.

$$x_{k+1} = \sum_{i=1}^{nc} \alpha_i \cdot x_i^{c+} \quad (3)$$

3.2 BÚSQUEDA DE ENVOLUTURA CONVEXA

Como se ha señalado en el apartado anterior, el objetivo de esta primera fase es encontrar el conjunto de estados candidatos. Este conjunto \tilde{x}_i^c se encuentra a una distancia ϕ que distará como máximo ε .

$$\phi_i \leq \varepsilon \quad \forall i = 1, 2, \dots, nc \quad \phi_i = \|\tilde{x}_i^c - \tilde{x}_k\| \quad (4)$$

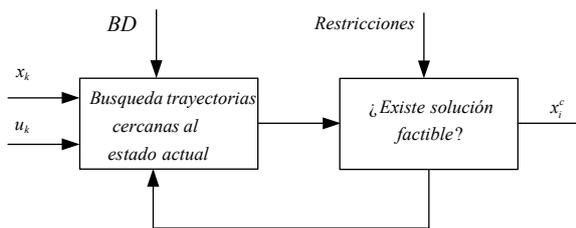


Fig 2. Esquema de la búsqueda de una envoltura convexa

Asimismo se intenta encontrar, en cuanto sea posible, que la envoltura convexa contenga a $\tilde{x}_k, \tilde{x}_k \in S$.

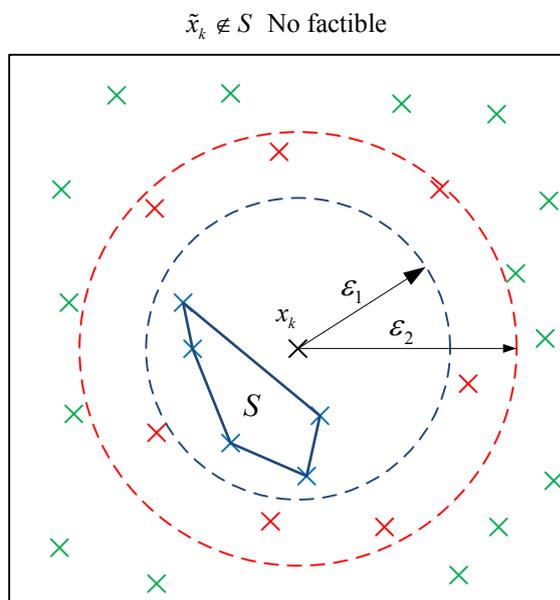


Fig 3. Representación en \mathbb{R}^2 de la búsqueda de una envoltura convexa donde $\tilde{x}_k \notin S$

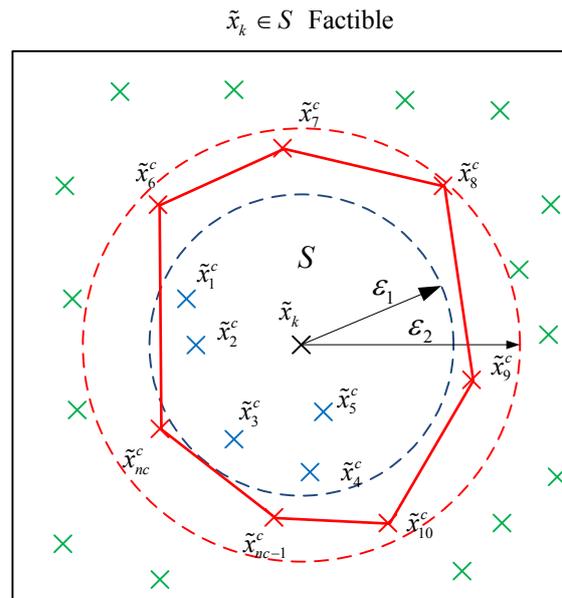


Fig 4. Representación en \mathbb{R}^2 de la búsqueda de una envoltura convexa donde $\tilde{x}_k \in S$

3.2.1 Búsqueda de estados candidatos

Se define un estado candidato \tilde{x}_i^c aquel que

- Forma parte de la trayectoria \tilde{x}_i^c
- La distancia entre \tilde{x}_i^c y \tilde{x}_k ha de ser menor que ε
- \tilde{x}_i^c es el estado más cercano a \tilde{x}_k entre los estados de su trayectoria
- Cada candidato tiene un sucesor \tilde{x}_i^{c+} , esto es, siguiente estado de la base de datos.

$$\phi_{\tilde{x}_i^c} = \min(\|\tilde{x}_i^c - \tilde{x}_k\|) \quad (5)$$

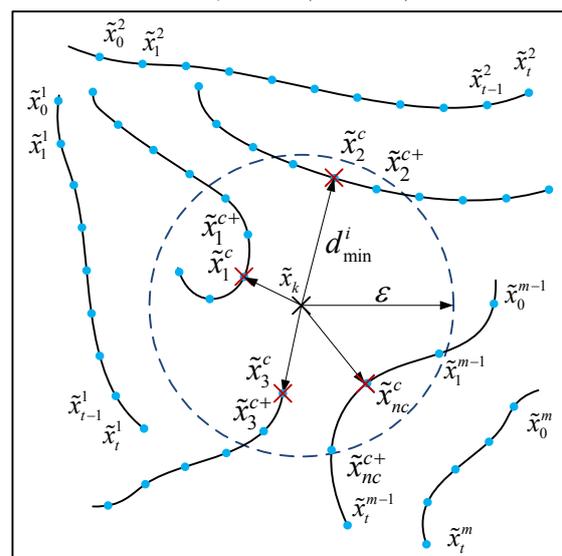


Fig 5. Representación en \mathbb{R}^2 de la búsqueda de los estados candidatos

3.2.2 Búsqueda de la factibilidad del problema de optimización. 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. 7 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}^{aprox} y de la bisección N_{iter}^{bisecc} no sea demasiado grande y no tenga un coste computacional.

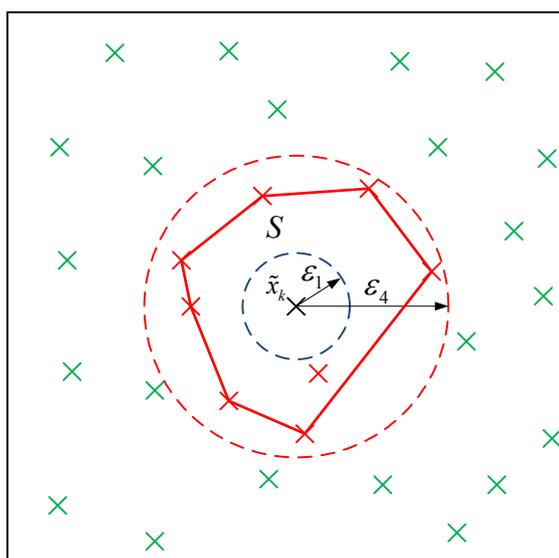


Fig 6. Optimización del parámetro ε

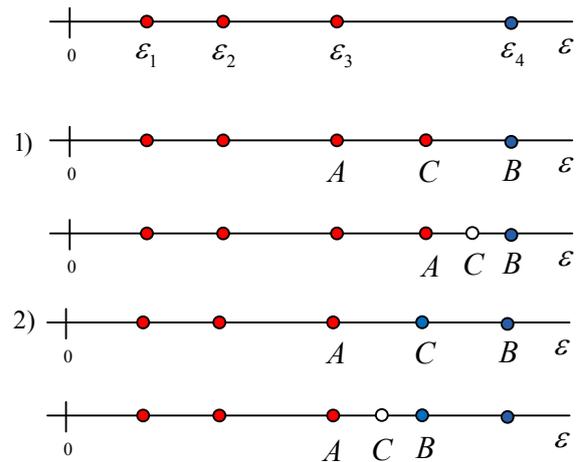


Fig 7. Método de la bisección

3.3 CÁLCULO DE LOS PESOS DE LOS ESTADOS CANDIDATOS

En este apartado se explica como en el MBD llega a calcular los diferentes pesos de los estados candidatos. Para una vez calculados los diferentes pesos poder calcular el estado futuro resolviendo un problema de optimización. Cabe resaltar que tras la búsqueda de candidatos obtenemos $\tilde{x}_i^c \forall i = 1, 2, \dots, nc$, pero para calcular el estado futuro \tilde{x}_{k+1} usamos los estados candidatos $\tilde{x}_i^{c+} \forall i = 1, 2, \dots, nc$.

$$\tilde{x}_{k+1} = \sum_{i=1}^{nc} \alpha_i \cdot \tilde{x}_i^{c+} \tag{6}$$

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

3.3.1 Caso A

En el caso de que $\tilde{x}_0 \in S$, fig. 4, se resuelve el siguiente problema de optimización

$$\min \sum_{i=1}^{nc} \alpha_i^2 \tag{8}$$

$$\text{s.a } \sum_{i=1}^{nc} \alpha_i = 1 \tag{9}$$

$$\sum_{i=1}^{nc} \alpha_i \tilde{x}_i^c = \tilde{x}_k \quad \forall i = 1, 2, \dots, nc \tag{10}$$

$$\tilde{x}_i^c, \tilde{x}_k \in \mathbb{R}^n$$

$$\alpha_i \geq 0 \quad \forall i = 1, 2, \dots, nc \tag{11}$$

$$\phi_i \leq \varepsilon \quad \forall i = 1, 2, \dots, nc \quad (12)$$

$$\phi_i = \|\tilde{x}_i^c - \tilde{x}_k\| \quad (13)$$

Donde se minimiza la suma de los pesos al cuadrado de cada estado candidato y se impone que la suma de peso asociado a cada estado candidato por el propio estado candidato sea igual al estado actual.

3.3.2 Caso B

Para los casos en los que la envoltura convexa no contenga al estado actual \tilde{x}_k (fig. 9) debido a la baja densidad de la base de datos o que \tilde{x}_0 se encuentre cerca de la región de los límites del espacio de trabajo del sistema, se modificará la restricción de que $\alpha_i \geq 0$ por $\alpha_i \in \mathbb{R}$.

Esta modificación se demuestra gráficamente en la fig. 8 para un espacio \mathbb{R} donde se observa que con la condición $\alpha_i \geq 0$ el estado actual no se puede encontrar fuera de la envoltura convexa (13), (14). No obstante, existen regiones en el espacio de trabajo donde $\tilde{x}_k \notin S$ por lo que liberamos dicha restricción $\alpha_i \in \mathbb{R}$. Así pues se puede decir que se extrapola la combinación convexa para la región exterior \bar{S} .

Sin embargo, al estimar más allá de la región S aparece un error que crece con la distancia a S . Con el fin de minimizar este problema, se añade a la expresión que se minimiza el término de la distancia para que pondere en mayor medida aquellos estados candidatos cercanos a la envoltura convexa y penalice aquellos que están en los límites del espacio de trabajo o estén muy alejados.

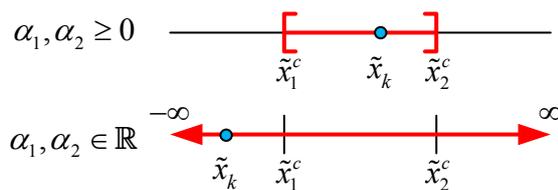


Fig 8. Principio del algoritmo caso B.

$$\tilde{x}_1^c \alpha_1 + \tilde{x}_2^c \alpha_2 = \tilde{x}_k \quad (13)$$

$$\alpha_1 + \alpha_2 = 1 \quad (14)$$

$$\alpha_1, \alpha_2 \geq 0 \quad (15)$$

$$\alpha_1, \alpha_2 \in \mathbb{R}$$

El problema de optimización para el caso B se define como:

$$\min \sum_{i=1}^{nc} \alpha_i^2 (a \cdot \phi_i + b) \quad (16)$$

$$\text{s.a.} \quad \sum_{i=1}^{nc} \alpha_i = 1 \quad (17)$$

$$\sum_{i=1}^{nc} \alpha_i \tilde{x}_i^c = \tilde{x}_k \quad (18)$$

$$\forall i = 1, 2, \dots, nc \quad \tilde{x}_i^c, \tilde{x}_k \in \mathbb{R}^n$$

$$\alpha_i \in \mathbb{R} \quad \forall i = 1, 2, \dots, nc \quad (19)$$

$$\phi_i \leq \varepsilon \quad \forall i = 1, 2, \dots, nc \quad (20)$$

$$\phi_i = \|\tilde{x}_i^c - \tilde{x}_0\| \quad (21)$$

Siendo a un parámetro que pondera la importancia de la distancia de los puntos de la envoltura convexa y b constante que permite la resolución matemática en caso de indeterminación.

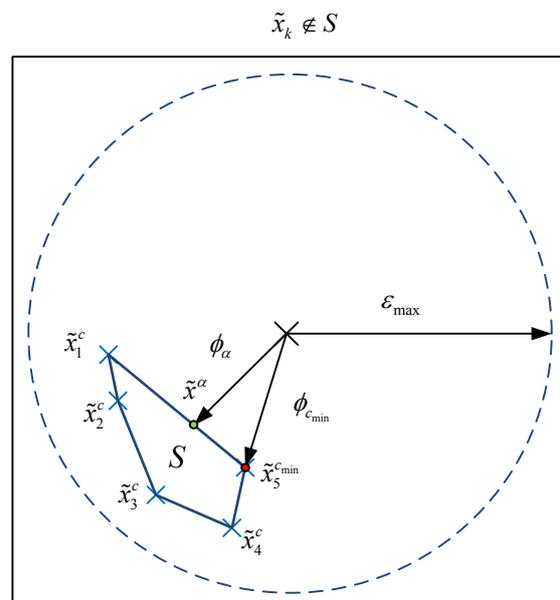


Fig 9. Representación en \mathbb{R}^2 de los diferentes métodos de solución para $\tilde{x}_k \notin S$

En la figura 9 se muestran otros métodos para calcular \tilde{x}_{k+1} cuando $\tilde{x}_k \notin S$, siendo los puntos verde y rojo respectivamente \tilde{x}^α y \tilde{x}_5^{\min} .

En primer lugar, se define \tilde{x}_k^α como el estado, que da como resultado de la combinación de los estados candidatos, más cercano al estado actual y en segundo lugar, $\tilde{x}_{k_{\min}}^c$ el estado candidato más cercano al estado actual. No obstante, \tilde{x}_k^{MBD} coincide con \tilde{x}_k se puede decir que:

$$\phi_\alpha > \phi_{c_{min}} > \phi_{x_k}^{MBD} \quad (22)$$

Por lo que optamos por elegir el modelo en base de datos sobre los demás estudiados demostrándose en el apartado siguiente.

4. EJEMPLO PRÁCTICO

En el siguiente apartado se aplicará el estudio realizado del MBD sobre una planta de cuatro tanques que se describe a continuación y se realizará un análisis comparando los resultados obtenidos por el MBD y las ecuaciones diferenciales de la planta.

4.1 DESCRIPCIÓN DE LA PLANTA

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).

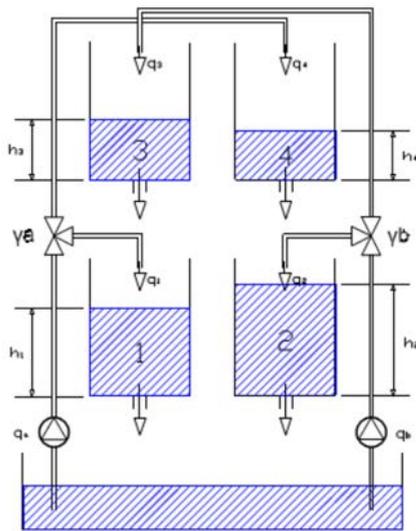


Fig 10. Esquema de la planta de Johansson

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 pasamos 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 forma que 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 ANÁLISIS Y RESULTADOS

4.3.1 Estudio en BA

4.3.1.1 Estimación de m óptimo

En este análisis se compara el comportamiento que presenta el MBD frente al modelo no lineal (MNL) que se rige por las ecuaciones diferenciales del sistema para diferentes bases de datos. De esta forma se estudia la densidad de base de datos, esto es, m número de trayectorias que ha de almacenar una base de datos para que el MBD pueda predecir con suficiente precisión los estados futuros.



Fig 11. Error de predicción para m trayectorias.

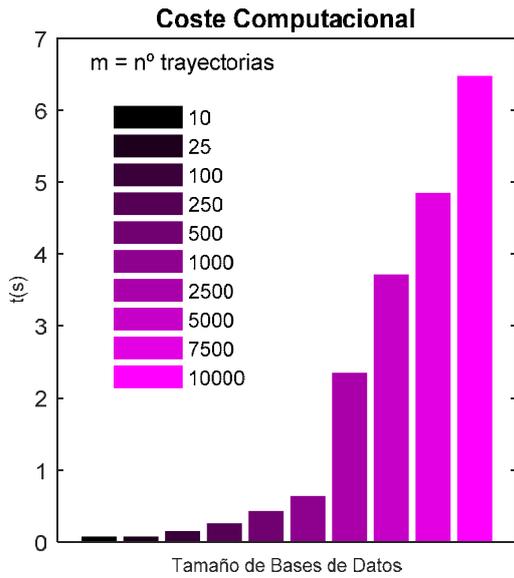


Fig 12. Coste computacional para m trayectorias

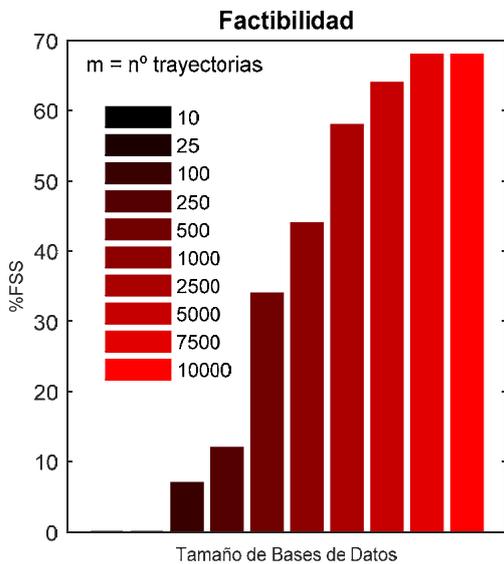


Fig 13. Porcentaje de factibilidad para m trayectorias

Se realiza una batería de experimentos en BA donde se calcula x_{k+1} partiendo de cien estados aleatorios \tilde{x}_k y se obtiene la media del error de predicción, el coste computacional y el porcentaje de veces que $\tilde{x}_k \in S$ (% FSS).

$$e^{Pred} = \sum_{k=1}^n \|x_{k+1}^{MBD} - x_{k+1}^{NL}\| \quad (23)$$

Cabe señalar que almacenar una trayectoria en este experimento pesa alrededor de diez kbs. Por tanto para diez mil trayectorias ocupa unos cien mbs.

4.3.1.2 Estudio de los diferentes modelos para el Caso B

En este apartado se analizan las diferentes soluciones que se estudia para obtener el estado futuro de nuestro modelo en el caso B, esto es, situación donde la envoltura convexa no contiene al estado actual $\tilde{x}_k \notin S$. Si en la fig. 9 se mostraban la diferencias entre los diferentes modelos donde $\phi_\alpha > \phi_{c_{min}} > \phi_{\tilde{x}_k}^{MBD}$ para los estados \tilde{x}_k , en la fig. 13 se obtiene el error de predicción de \tilde{x}_{k+1} sobre la planta de los cuatro tanques.

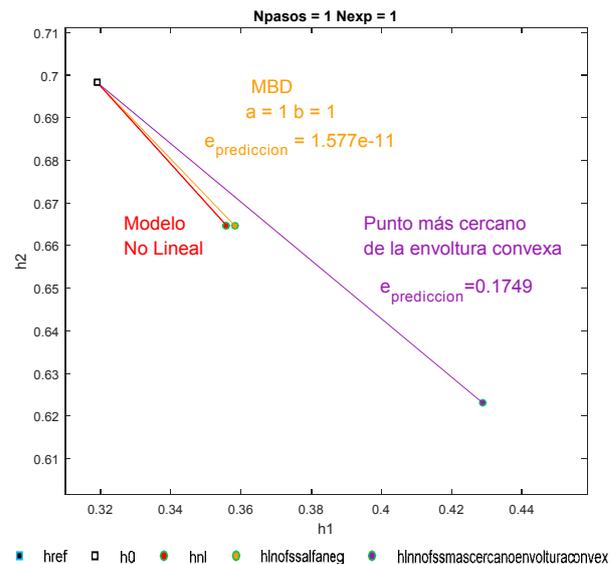


Fig 14. Comparativa de los diferentes métodos de resolución analizados para $\tilde{x}_k \notin S$

Se demuestra por tanto que el error de predicción del MBD es mucho menor que la solución del estado más cercano de la envoltura convexa y con un error muy pequeño.

4.3.2 Estudio en BC. Ejemplo de una trayectoria

Por último, se estudia el comportamiento en bucle cerrado del MBD y el modelo no lineal controlados por un controlador LQR. Con ese fin, se evalúa sobre diferentes consignas que establecen el concurso de la CEA del año 2014.

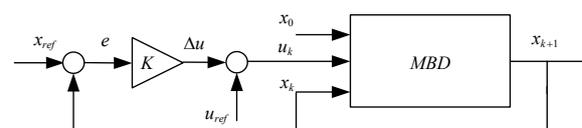


Fig 15. Esquema de control LQR con MBD

Asimismo hay que decidir qué tamaño asignamos a la base de datos. Ante los experimentos realizados en BA tomamos un valor de m (número de trayectorias) como solución de compromiso entre un bajo error de predicción, un bajo coste computacional y un elevado porcentaje de factibilidad. En este caso particular se toman mil trayectorias que ocupan alrededor de cien mbs.

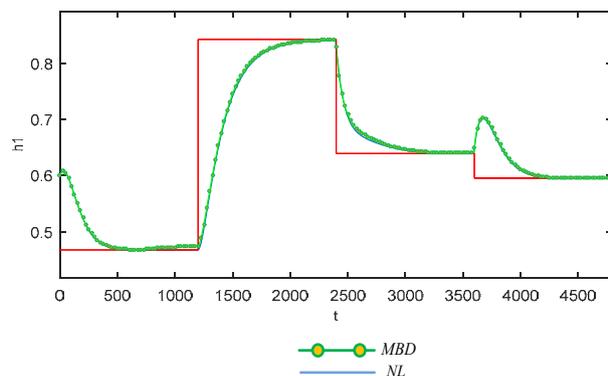


Fig 16. Representación de h_1 respecto el t para el concurso de la CEA año 2014 [1]

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)