

Proyecto Fin de Máster
Máster Universitario en Ingeniería Industrial

Control Offset-free basado en datos de la planta de
los cuatro tanques

Autor: Juan Luis García Sánchez

Tutor: Daniel Rodríguez Ramírez

Dpto. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2020



Proyecto Fin de Máster
Máster Universitario en Ingeniería Industrial

Control Offset-free basado en datos de la planta de los cuatro tanques

Autor:

Juan Luis García Sánchez

Tutor:

Daniel Rodríguez Ramírez

Profesor titular de Universidad

Dpto. de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2020

Proyecto Fin de Máster: Control Offset-free basado en datos de la planta de los cuatro tanques

Autor: Juan Luis García Sánchez

Tutor: Daniel Rodríguez Ramírez

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2020

El Secretario del Tribunal

Agradecimientos

Me gustaría agradecer todo este trabajo a mi familia y en especial a Reyes, por apoyarme en todo momento y animarme a seguir. También me gustaría agradecer a José Ramón su ayuda en todo este proyecto. Por último, agradecer a Dani el brindarme la posibilidad de trabajar en una planta real y tutorarme durante todo este proceso, además de fomentar mi interés por la investigación.

Juan Luis García Sánchez

Sevilla, 2020

Resumen

La modelización de sistemas siempre ha supuesto un problema a la hora de tratar de controlar un sistema. Aunque existen métodos de identificación de sistemas, a menudo ocurre que el sistema es demasiado grande, complejo y con tantas variables que resulta imposible de modelar. Una posible solución para este tipo de problemas es el hecho de prescindir de modelo, tratando de optimizar el problema sin importar el tamaño de este o el número de variables implicadas, además de todas las posibles perturbaciones que afectarían al sistema.

En este trabajo se plantea la implementación de un Control Predictivo Basado en Datos (CPBD), o en inglés Data-based Predictive Control (DbPC). La estrategia de este tipo de control se basa en la proposición de que el sistema ha sido operado el suficiente tiempo en bucle cerrado para acumular una base de datos a partir de la cual se pueda controlar el sistema. Se trata de técnicas de Machine Learning a partir de las cuales, un sistema puede ser controlado sin la necesidad de un conocimiento previo del mismo.

El CPBD será implementado en la planta de los cuatro tanques. Este sistema consta de dos tanques superiores que vierten agua sobre dos tanques inferiores. La dificultad del control reside en que las bombas utilizadas para controlar el nivel de los tanques inferiores no vierten solo en los inferiores, sino que también vierten de forma cruzada en los superiores. Esto hace que el sistema presente un comportamiento de fase no mínima, además de presentar una constante de tiempo muy elevada (en torno a un minuto). Se pretende controlar la planta con un CPBD en tracking, ampliando y mejorando los resultados obtenidos en el trabajo previo a éste donde se controló la planta con un CPBD en regulación.

El proyecto se estructura principalmente en dos bloques. En el primero, el cual abarca los dos primeros capítulos, se describirá la planta de los cuatro tanques. Además, se describirán las ecuaciones que rigen el sistema y se realizará un análisis detallado del comportamiento de este. El segundo bloque se centrará en el diseño de controladores PID clásicos que controlen el sistema y permitan obtener una base de datos. Para el control de sistema se describirán y compararán dos algoritmos: el algoritmo Data-Driven MPC y el algoritmo Offset-free, siendo el de Offset-free una evolución del de Data-Driven MPC. Finalmente, se realizarán unas conclusiones describiendo las ventajas e inconvenientes de los algoritmos, así como las dificultades encontradas a lo largo del desarrollo de este trabajo.

Abstract

Modelling systems has always been a problem when trying to control a system. Although there are systems identification methods, it often happens that the systems are too large, complex and with so many variables that are impossible to model. A possible solution for this type of problem is to dispense with the model, trying to optimize the problem regardless of its size or the number of variables involved, in addition to all possible disturbances that would affect the system.

This work proposes the implementation of a Data-based Predictive Control (DbPC). The strategy of this type of control is based on the proposition that the system has been operated long enough in a closed loop to accumulate a database from which the system can be controlled. These are Machine Learning techniques from which a system can be controlled without the need for previous knowledge on it.

The CPBD will be implemented at the plant of the four tanks. This system consists of two upper tanks that pour water on two lower tanks. The difficulty of control lies in the fact that the pumps used to control the level of the lower tanks do not pour only in the lower ones, but also pour in a crossed way in the upper ones. This makes the system present a non-minimal phase behaviour, in addition to presenting a very high time constant (around one minute). It is intended to control the plant with a DbPC in tracking, expanding and improving the results obtained in the previous work where the plant was controlled with a DbPC in regulation.

The project is mainly structured in two blocks. In the first one, which covers the first two chapters, the plant of the four tanks will be described. In addition, the equations that govern the system will be described and a detailed analysis of its behaviour will be performed. The second block will focus on the design of classic PID controllers that control the system and allow to obtain a database. For the system control, two algorithms will be described and compared: The Data-Driven MPC algorithm and the Offset-free Data-Driven MPC algorithm, the Offset-free algorithm being an evolution of the Data-Driven MPC algorithm. Finally, conclusions will be made describing the advantages and disadvantages of the algorithms, as well as the difficulties encountered throughout the development of this work.

Agradecimientos	vii
Resumen	ix
Abstract	xi
Índice	xiii
Índice de Tablas	xv
Índice de Figuras	xvii
1 Introducción	1
1.1 <i>Trabajo previo y objetivos</i>	2
2 Planta de los cuatro tanques	3
2.1 <i>Descripción de la planta</i>	3
2.2 <i>Modelo teórico</i>	4
2.3 <i>Parámetros del modelo de la planta real</i>	4
2.3.1 <i>Cálculo de los coeficientes de descarga</i>	4
2.3.2 <i>Parámetros del modelo de la planta real</i>	5
2.4 <i>Linealización y Análisis del modelo</i>	6
2.4.1 <i>Ecuaciones lineales</i>	6
2.4.2 <i>Funciones de transferencia</i>	7
2.5 <i>Análisis del sistema</i>	7
2.5.1 <i>Comparación entre sistema real, modelo no lineal y lineal</i>	8
2.5.2 <i>Significado físico de las funciones de transferencia</i>	9
2.5.3 <i>Análisis de ceros y polos del sistema</i>	9
2.5.4 <i>Estudio de interacciones y emparejamiento de entradas y salidas</i>	10
3 Generación de las base de datos	13
3.1 <i>Mapa de puntos factibles</i>	13
3.2 <i>Diseño de controladores</i>	14
3.3 <i>Obtención de trayectorias</i>	15
3.4 <i>Resultados</i>	16
3.4.1 <i>Resultados en simulación</i>	17
3.4.2 <i>Resultados en la planta real</i>	18
4 Control Predictivo Basado en Datos	19
4.1 <i>Creación de la base de datos</i>	20
4.2 <i>Formulación de los algoritmos de control</i>	20
4.2.1 <i>Algoritmo Data-Driven MPC</i>	20
4.2.2 <i>Algoritmo de Offset-free</i>	26
4.3 <i>Comparación del algoritmo Data-Driven MPC y Offset-free Data-Driven MPC</i>	30
4.4 <i>Discrepancias entre modelo Simulado y el sistema Real</i>	31
5 Conclusiones	33
Referencias	35
Anexo: Códigos usados	37
<i>Generación de las trayectorias</i>	37
<i>Algoritmo Data-Driven MPC</i>	38
<i>Algoritmo de Offset-free</i>	42

Índice de Tablas

Tabla 2-1. Parámetros del modelo de la planta real.	6
Tabla 2-2. Parámetros de las funciones de transferencia.	10
Tabla 3-1. Parámetros de la identificación de la planta.	15
Tabla 3-2. Parámetros iniciales de los PID.	15
Tabla 3-3. Rango de parámetros de los PID.	15
Tabla 4-1. Estructura de la base de datos.	20

Índice de Figuras

Figura 1-1. Esquema general de control.	1
Figura 1-2. Planta de los cuatro tanques real.	2
Figura 2-1. (a) Esquema de la planta de Johansson. (b) Esquema de la planta real.	3
Figura 2-2. Diagrama de bloques de la función de transferencia $G(s)$.	7
Figura 2-3. Comparación de comportamiento real-experimental respecto al simulado no lineal y lineal.	8
Figura 2-4. Lugar de las raíces de cada función de transferencia del sistema.	9
Figura 2-5. Mapa de ceros y polos del sistema.	10
Figura 3-1. Mapa de puntos factibles para el Tanque 3 y el Tanque 4.	13
Figura 3-2. Mapa de puntos factibles en el Tanque 1 y Tanque 2 y en el caudal a y caudal b .	14
Figura 3-3. Diagrama de control con PID para la planta de los 4 tanques.	14
Figura 3-4. Procedimiento para la obtención de las trayectorias.	16
Figura 3-5. Modelo en Simulink de la generación de trayectorias.	16
Figura 3-6. Base de datos de 50 trayectorias simuladas.	17
Figura 3-7. Base de datos de 100 trayectorias simuladas.	17
Figura 3-8. Base de datos de 50 trayectorias experimentales.	18
Figura 3-9. Base de datos de 100 trayectorias experimentales.	18
Figura 4-1. Estructura general del Control Predictivo Basado en Datos.	19
Figura 4-2. Diagrama de la red de distribución de aguas de Richmond.	19
Figura 4-3. Ejemplo de problema de factibilidad. A la izquierda no es factible, a la derecha sí lo es.	21
Figura 4-4. Diagrama de Simulink de la implementación del Control Predictivo Basado en Datos.	23
Figura 4-5. Altura de los tanques implementando el algoritmo Data-Driven MPC con una base de datos de 50 trayectorias simuladas.	23
Figura 4-6. Altura de los tanques implementando el algoritmo Data-Driven MPC con una base de datos de 100 trayectorias simuladas.	24
Figura 4-7. Altura de los tanques implementando el algoritmo Data-Driven MPC con una base de datos de 1000 trayectorias simuladas.	24
Figura 4-8. Altura de los tanques implementando el algoritmo Data-Driven MPC con una base de datos de 50 trayectorias reales.	25
Figura 4-9. Altura de los tanques implementando el algoritmo Data-Driven MPC con una base de datos de 100 trayectorias reales.	25
Figura 4-10. Porcentaje de betas negativas para la simulación descrita en la Figura 4-13.	27
Figura 4-11. Diagrama de bloques del subsistema CPBD para el algoritmo de Offset-free.	27
Figura 4-12. Altura de los tanques implementando el algoritmo de Offset-free con una base de datos de 50 trayectorias simuladas.	28
Figura 4-13. Altura de los tanques implementando el algoritmo de Offset-free con una base de datos de 100 trayectorias simuladas.	28
Figura 4-14. Altura de los tanques implementando el algoritmo de Offset-free con una base de datos de 50 trayectorias reales.	29

Figura 4-15. Altura de los tanques implementando el algoritmo de Offset-free con una base de datos de 100 trayectorias reales.	29
Figura 4-16. Altura de los tanques comparando el algoritmo Data-Driven MPC con el Offsetfree con una base de datos de 100 trayectorias simuladas.	30
Figura 4-17. Altura de los tanques comparando el algoritmo Data-Driven MPC con el Offsetfree con una base de datos de 100 trayectorias reales.	30
Figura 4-18. Altura de los tanques comparando la respuesta en simulación y real con una base de datos de 100 trayectorias simuladas y reales respectivamente utilizando el algoritmo de Offsetfree.	31

1 INTRODUCCIÓN

El ámbito de la ingeniería de control se encuentra entre los más relevantes para el desarrollo tecnológico. Éste permite el control de sistemas complejos sin la intervención humana directa. El campo de aplicación es total, abarcando desde industria y maquinaria simple, pasando por procesos tales como la gestión de la distribución de agua o eléctrica, hasta el manejo de autómatas, robots y drones.

Todo proceso requiere del control de éste mediante sensores, que permiten conocer la situación real del sistema, y actuadores, que permiten modificar el estado de este (Figura 1-1). Para que el control sea adecuado se deben utilizar técnicas de control para analizar el sistema a controlar, así como diseñar un controlador tal que el sistema trabaje siempre en un punto óptimo siendo lo más invulnerable posible a perturbaciones externas.

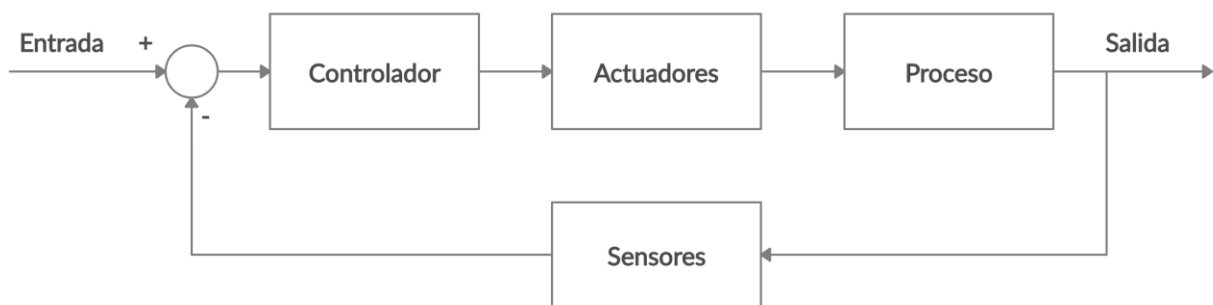


Figura 1-1. Esquema general de control.

Actualmente, el control por PID es el más extendido en las industrias, abarcando cerca del 99% del control de procesos, debiéndose esto a su simplicidad, robustez y versatilidad. Como se explicará en más adelante, el PID se usará para controlar el sistema seleccionado en el control de la planta en un principio, siendo necesarios 2 PID, uno por cada tanque que se desea controlar.

También existen controles predictivos como el MPC (Model Predictive Control) el cual se basa en un modelo dinámico para predecir el comportamiento futuro del sistema y, a partir de esta predicción, tomar la acción de control óptima. Además, este tipo de controladores ofrecen la adición de restricciones de forma natural porque estas se añaden a la función de minimización tal cual, así como facilidad para tratar con sistemas multivariables. A esto se le suma el hecho de que compensa intrínsecamente los retardos, evitando la utilización de bucles de control más complicados como el predictor de Smith.

No obstante, tanto los controladores PID como MPC requieren del conocimiento previo del modelo del sistema. Estos modelos pueden obtenerse de forma teórica, experimental o mediante ambas. El hecho de no poder deducir el sistema teóricamente, debido a su alta complejidad o enorme casuística, o experimentalmente, debido a un alto coste o imposibilidad, hace necesaria la búsqueda de soluciones.

De este problema nace el Control Predictivo Basado en Datos (CPBD) o en inglés Data-based Predictive Control (DbPC). Este control toma una base de datos en el que se recopila los estados de las trayectorias históricas del sistema y, a partir de ellas, se reconstruye los estados prediciendo los estados futuros, pudiendo tomar así una acción de control. La principal diferencia con el control MPC reside en que el CPBD no necesita de modelo. Aunque tienen similitudes en cuanto a la sencilla adición de restricciones, la compensación de retardos o la posibilidad de controlar sistemas multivariables. Sobre el CPBD se hablará en este trabajo, siendo el desarrollo de éste el objetivo del trabajo. Se explicará y demostrará tanto en simulación como experimentalmente sus resultados.

1.1 Trabajo previo y objetivos

Se parte de una planta real de 4 tanques de agua interconectados (Figura 1-2) entre sí, los cuales se alimentan a través de una bomba de agua y se controlan mediante válvulas proporcionales. Toda la a paramenta, cableado, conexionado y control de la bomba y válvulas de agua se encuentra implementado. Es posible el control de los caudales de alimentación a los tanques mediante la interfaz de LabVIEW, el cual se encarga de manera sencilla de gestionar las referencias de control de la bomba y válvulas.

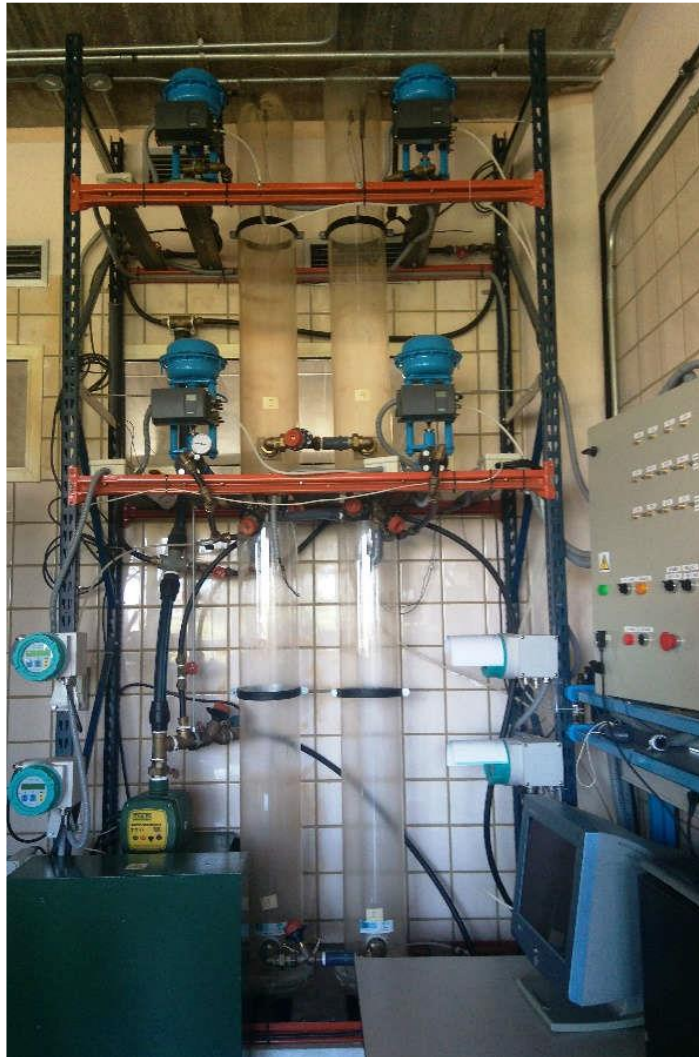


Figura 1-2. Planta de los cuatro tanques real.

Se parte de trabajo previo hecho para la planta real de los 4 tanques en cuanto a la recopilación de trayectorias en regulación para implementación de un CPBD [1]. Aunque los resultados obtenidos controlaban el sistema adecuadamente, en el régimen permanente no se conseguía alcanzar la referencia ya que presentaba un pequeño offset. Dichos resultados experimentales se usarán como punto de partida para este trabajo con el fin de mejorarlos.

Se pretende completar el trabajo ya realizado implementando un control en seguimiento o tracking, es decir, habilitando el control para un rango de referencias factibles. Sobre esto, el método de control continuará basándose en datos (CPBD), pero se añadirán restricciones en la función de optimización para hacer posible el tracking, además de disminuir o eliminar el offset ya existente.

2 PLANTA DE LOS CUATRO TANQUES

2.1 Descripción de la planta

La planta de los cuatro tanques que se va a utilizar en este trabajo está basada de una planta conocida como The Quadruple-Tank Process [2], propuesta por Karl H. Johansson a finales del siglo XX. Ha sido mundialmente utilizada para el estudio de todo tipo de controles PID, MPC, GPC, DMC, adaptativos, etc. Esto se debe a su versatilidad ya que ajustando las válvulas se pueden mover los ceros y polos del sistema obteniendo una gran variedad de tipologías. Tanto esta planta como la que se dispone son conceptualmente las mismas, pero se han modificado ciertos aspectos para hacerla constructivamente más sencilla y barata. Ambas tipologías se muestran en la Figura 2-1 donde se pueden observar sus diferencias.

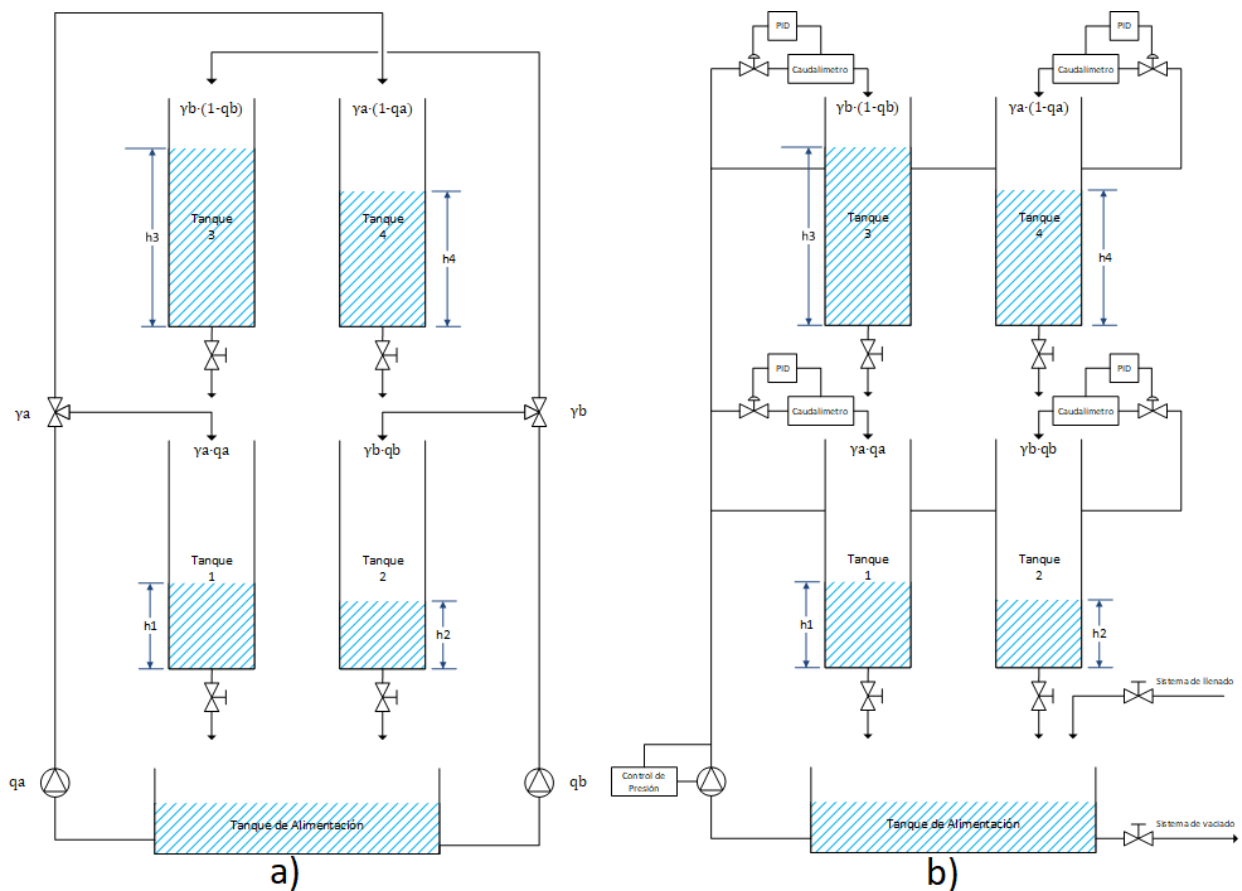


Figura 2-1. (a) Esquema de la planta de Johansson. (b) Esquema de la planta real.

En la planta de Johansson se tienen 2 bombas de agua. El caudal de cada una de ellas pasa por una válvula de tres vías, separando el flujo q en dos: $q\gamma$ y $q(1-\gamma)$. Los flujos del caudal a se vierten en los tanques 1 y 4, mientras que los del b se vierten en el 2 y 3. Además, el tanque 3 descarga sobre el 1, el 4 sobre el 2 y el 1 y 2 sobre el tanque de almacenamiento.

En la planta real tan solo se dispone de una bomba que inyecta presión a todas las válvulas neumáticas controladas. Aunque no se dispone de válvulas de 3 vías, el comportamiento de éstas se simula mediante un control PID en cada válvula neumática. Por tanto, se concluye en que el comportamiento de ambas plantas es el mismo.

Cabe destacar que dependiendo de la apertura de las válvulas de 3 vías (γ_a y γ_b) y debido al cruzamiento de los caudales, es decir, que la rama izquierda vierte en la derecha y viceversa, pueden darse situaciones donde el

control deja de ser trivial. Como se discutirá posteriormente, los parámetros de la planta real son tales que es más conveniente controlar con la bomba izquierda la rama derecha y viceversa. Además, se debe tener en cuenta los límites físicos del sistema, es decir, existen caudales máximos y mínimo que puede generar la bomba y alturas máximas y mínimas que no se pueden sobrepasar.

2.2 Modelo teórico

Como ya se ha explicado, el modelo que se utilizará para el análisis de la planta será el de la planta de Johansson. Las ecuaciones diferenciales que rigen el sistema son las siguientes,

$$\begin{aligned} A\dot{h}_1 &= -a_1\sqrt{2gh_1} + a_3\sqrt{2gh_3} + \gamma_a q_a \\ A\dot{h}_2 &= -a_2\sqrt{2gh_2} + a_4\sqrt{2gh_4} + \gamma_b q_b \\ A\dot{h}_3 &= -a_3\sqrt{2gh_3} + (1-\gamma_b)q_b \\ A\dot{h}_4 &= -a_4\sqrt{2gh_4} + (1-\gamma_a)q_a \end{aligned} \quad (2-1)$$

Donde A es la sección transversal de los tanques en metros, los cuales son todos iguales. \dot{h}_i es la variación de la altura del tanque i respecto del tiempo (dh_i/dt). h_i denota el nivel de líquido del tanque i en metros. a_i es la sección transversal del orificio de descarga del depósito i , en metros cuadrados. g es el valor de la gravedad, 9.8m/s^2 . γ_a y γ_b denotan la apertura de la válvula de 3 vías. q_a y q_b son los caudales aplicados en cada rama en metros cúbicos por segundo.

Este modelo es ideal, ya que no describe las dinámicas presentes en la planta real tales como:

- Evolución de las presiones en las tuberías.
- Caudales de alimentación a cada uno de los tanques.
- Vórtices en la descarga de los tanques.
- Ruidos en la instrumentación.
- Suciedad y obstrucción de las válvulas de descarga.

Al despreciar todas estas dinámicas se incurre en errores en cuanto a la evolución de los niveles de líquido de los tanques. A pesar de ello, es una buena aproximación y punto de inicio para el análisis del sistema. Además, se comprobará en dicho análisis que el modelo teórico se asimila bastante al real.

2.3 Parámetros del modelo de la planta real

2.3.1 Cálculo de los coeficientes de descarga

Los parámetros de las secciones de descarga de los tanques no se conocen a priori, aunque se pueden calcular experimentalmente. Cuando el sistema se encuentre régimen permanente, las derivadas de (2-1) se anulan y es posible despejar las a_i . Partiendo de los experimentos ya realizados en la planta, se tiene un histórico de 114 trayectorias en bucle abierto, donde el sistema ha alcanzado el régimen permanente. Con estos datos, se tiene un sistema incompatible sobredeterminado (más ecuaciones que incógnitas) del que se pretende obtener una solución en el sentido de los mínimos cuadrados. En primer lugar, se va a resolver por mínimos cuadrados a_3 y a_4 :

$$\begin{aligned} \begin{bmatrix} \sqrt{2gh_3^1} \\ \sqrt{2gh_3^2} \\ \vdots \\ \sqrt{2gh_3^{114}} \end{bmatrix} a_3 &= \begin{bmatrix} (1-\gamma_b)q_b^1 \\ (1-\gamma_b)q_b^2 \\ \vdots \\ (1-\gamma_b)q_b^{114} \end{bmatrix} \rightarrow M_3 a_3 = N_3 \rightarrow \text{Mínimos Cuadrados} \rightarrow a_3 = (M_3^T M_3)^{-1} M_3^T N_3 \\ \begin{bmatrix} \sqrt{2gh_4^1} \\ \sqrt{2gh_4^2} \\ \vdots \\ \sqrt{2gh_4^{114}} \end{bmatrix} a_4 &= \begin{bmatrix} (1-\gamma_a)q_a^1 \\ (1-\gamma_a)q_a^2 \\ \vdots \\ (1-\gamma_a)q_a^{114} \end{bmatrix} \rightarrow M_4 a_4 = N_4 \rightarrow \text{Mínimos Cuadrados} \rightarrow a_4 = (M_4^T M_4)^{-1} M_4^T N_4 \end{aligned} \quad (2-2)$$

Donde h_i^j es el nivel del tanque de agua i de la trayectoria j y $q_{a/b}^j$ es el caudal a o b de la trayectoria j . Con dichos parámetros calculados, se puede calcular a su vez, a_1 y a_2 de la siguiente forma:

$$\begin{aligned} \begin{bmatrix} \sqrt{2gh_1^1} \\ \sqrt{2gh_1^2} \\ \vdots \\ \sqrt{2gh_1^{114}} \end{bmatrix} a_1 &= \begin{bmatrix} a_3 \sqrt{2gh_3^1} + \gamma_a q_a^1 \\ a_3 \sqrt{2gh_3^2} + \gamma_a q_a^2 \\ \vdots \\ a_3 \sqrt{2gh_3^{114}} + \gamma_a q_a^{114} \end{bmatrix} \rightarrow M_1 a_1 = N_1 \rightarrow \text{Mínimos Cuadrados} \rightarrow a_1 = (M_1^T M_1)^{-1} M_1^T N_1 \\ \begin{bmatrix} \sqrt{2gh_2^1} \\ \sqrt{2gh_2^2} \\ \vdots \\ \sqrt{2gh_2^{114}} \end{bmatrix} a_2 &= \begin{bmatrix} a_4 \sqrt{2gh_4^1} + \gamma_b q_b^1 \\ a_4 \sqrt{2gh_4^2} + \gamma_b q_b^2 \\ \vdots \\ a_4 \sqrt{2gh_4^{114}} + \gamma_b q_b^{114} \end{bmatrix} \rightarrow M_2 a_2 = N_2 \rightarrow \text{Mínimos Cuadrados} \rightarrow a_2 = (M_2^T M_2)^{-1} M_2^T N_2 \end{aligned} \quad (2-3)$$

2.3.2 Parámetros del modelo de la planta real

El resto de los parámetros del modelo ya están definidos físicamente, por tanto, los parámetros son los siguientes:

Parámetro	Símbolo	Valor
Área Transversal	A	300 cm ²
Apertura de válvula a	γ_a	0.3
Apertura de válvula b	γ_b	0.4
Coefficiente de descarga del tanque 1	a_1	1.2873 cm ²
Coefficiente de descarga del tanque 2	a_2	1.5645 cm ²
Coefficiente de descarga del tanque 3	a_3	0.8646 cm ²
Coefficiente de descarga del tanque 4	a_4	0.8467 cm ²
Altura de los tanques	h_i	0 m - 1.4 m

Caudal de las bombas	q_j	$0 \text{ m}^3/\text{h} - 3 \text{ m}^3/\text{h}$
Tiempo de muestreo	T_m	30 s

Tabla 2-1. Parámetros del modelo de la planta real.

2.4 Linealización y Análisis del modelo

2.4.1 Ecuaciones lineales

Para analizar el comportamiento y estabilidad del sistema de manera sencilla es necesario linealizar las ecuaciones del modelo (2-1). Para ello se utiliza el desarrollo en polinomio de Taylor truncado en el término lineal y se suponen condiciones iniciales nulas. Con ello, las ecuaciones que se obtienen son las siguientes:

$$\begin{aligned}
 A\dot{h}_1 &= -a_1\sqrt{\frac{g}{2h_1^0}}\bar{h}_1 + a_3\sqrt{\frac{g}{2h_3^0}}\bar{h}_3 + \gamma_a\bar{q}_a \\
 A\dot{h}_2 &= -a_2\sqrt{\frac{g}{2h_2^0}}\bar{h}_2 + a_4\sqrt{\frac{g}{2h_4^0}}\bar{h}_4 + \gamma_b\bar{q}_b \\
 A\dot{h}_3 &= -a_3\sqrt{\frac{g}{2h_3^0}}\bar{h}_3 + (1-\gamma_b)\bar{q}_b \\
 A\dot{h}_4 &= -a_4\sqrt{\frac{g}{2h_4^0}}\bar{h}_4 + (1-\gamma_a)\bar{q}_a
 \end{aligned} \tag{2-4}$$

Donde h_i^0 es el nivel del tanque i en el punto de linealización, \bar{h}_i es la diferencia entre h_i y h_i^0 , y de forma análoga, $\bar{q}_{a/b}$ es la diferencia entre $q_{a/b}$ y $q_{a/b}^0$ (caudal en el punto de linealización).

Las ecuaciones lineales pueden representarse en espacio de estados [3] de la siguiente forma:

$$\begin{aligned}
 \begin{bmatrix} \dot{h}_1 \\ \dot{h}_2 \\ \dot{h}_3 \\ \dot{h}_4 \end{bmatrix} &= \underbrace{\begin{bmatrix} -\frac{a_1}{A}\sqrt{\frac{g}{2h_1^0}} & 0 & \frac{a_3}{A}\sqrt{\frac{g}{2h_3^0}} & 0 \\ 0 & \frac{a_2}{A}\sqrt{\frac{g}{2h_2^0}} & 0 & \frac{a_4}{A}\sqrt{\frac{g}{2h_4^0}} \\ 0 & 0 & -\frac{a_3}{A}\sqrt{\frac{g}{2h_3^0}} & 0 \\ 0 & 0 & 0 & -\frac{a_4}{A}\sqrt{\frac{g}{2h_4^0}} \end{bmatrix}}_A \begin{bmatrix} \bar{h}_1 \\ \bar{h}_2 \\ \bar{h}_3 \\ \bar{h}_4 \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{\gamma_a}{A} & 0 \\ 0 & \frac{\gamma_b}{A} \\ 0 & \frac{1-\gamma_b}{A} \\ \frac{1-\gamma_a}{A} & 0 \end{bmatrix}}_B \begin{bmatrix} \bar{q}_a \\ \bar{q}_b \end{bmatrix} \\
 \begin{bmatrix} \bar{h}_1 \\ \bar{h}_2 \end{bmatrix} &= \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}}_C \begin{bmatrix} \bar{h}_1 \\ \bar{h}_2 \\ \bar{h}_3 \\ \bar{h}_4 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}}_D \begin{bmatrix} \bar{q}_a \\ \bar{q}_b \end{bmatrix}
 \end{aligned} \tag{2-5}$$

La formulación en espacio de estados es útil a la hora de transformar las ecuaciones al dominio de Laplace, para poder analizar de forma adecuada su comportamiento.

2.4.2 Funciones de transferencia

El siguiente paso para realizar el análisis del sistema consiste en la transformación de las ecuaciones lineales al dominio de Laplace. Esta transformación se realiza siguiendo la relación [4]:

$$G(s) = [C(sI - A)]^{-1} B + D \quad (2-6)$$

Siendo A , B , C y D las matrices de la formulación en espacio de estados descrita en (2-5). Con ello se consigue una matriz de funciones de transferencia que representan las interacciones. A su vez, $G(s)$ puede representarse como:

$$G(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \quad (2-7)$$

Donde $G_{ij}(s)$ es la función de transferencia que relaciona la entrada i con la salida j . Estas interacciones se materializan en la Figura 2-2.

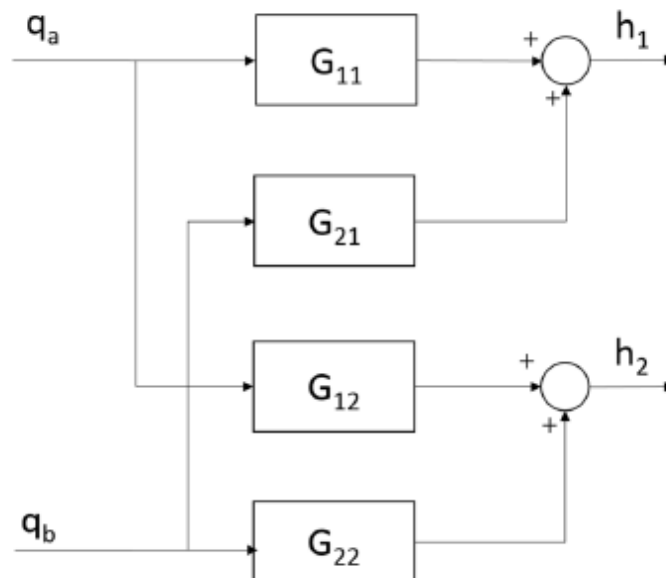


Figura 2-2. Diagrama de bloques de la función de transferencia $G(s)$.

2.5 Análisis del sistema

Para comenzar a analizar el sistema se necesita conocer su función de transferencia, pero primero es necesario fijar un punto de linealización porque, tanto el espacio de estados como $G(s)$ dependen de $h^0 = [h_1^0 \ h_2^0 \ h_3^0 \ h_4^0]^T$ según (2-5). Por ello, se ha elegido un punto arbitrario contenido en la base de datos. También se supone que, para un análisis simple del sistema, la dinámica de este no varía sustancialmente con el punto de linealización. Con todo ello se obtienen el siguiente espacio de estados junto con sus funciones de transferencia correspondientes:

$$h^0 = [0.6215 \ 0.6151 \ 0.6213 \ 0.8265]^T \text{ [m]} \quad q^0 = [1.76 \ 1.78]^T \text{ [m}^3/\text{h}] \quad (2-8)$$

$$\begin{bmatrix} \dot{\bar{h}}_1 \\ \dot{\bar{h}}_2 \\ \dot{\bar{h}}_3 \\ \dot{\bar{h}}_4 \end{bmatrix} = \begin{bmatrix} -0.0121 & 0 & 0.0081 & 0 \\ 0 & -0.0147 & 0 & 0.0069 \\ 0 & 0 & -0.0081 & 0 \\ 0 & 0 & 0 & -0.0069 \end{bmatrix} \begin{bmatrix} \bar{h}_1 \\ \bar{h}_2 \\ \bar{h}_3 \\ \bar{h}_4 \end{bmatrix} + \begin{bmatrix} 0.0028 & 0 \\ 0 & 0.0037 \\ 0 & 0.0056 \\ 0.0065 & 0 \end{bmatrix} \begin{bmatrix} \bar{q}_a \\ \bar{q}_b \end{bmatrix} \quad (2-9)$$

$$\begin{bmatrix} \bar{h}_1 \\ \bar{h}_2 \\ \bar{h}_3 \\ \bar{h}_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{h}_1 \\ \bar{h}_2 \\ \bar{h}_3 \\ \bar{h}_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{q}_a \\ \bar{q}_b \end{bmatrix}$$

$$\begin{aligned} G_{11}(s) &= \frac{h_1}{q_a} = \frac{1}{360s + 4.34} & G_{12}(s) &= \frac{h_2}{q_a} = \frac{1}{2223s^2 + 448s + 2.17} \\ G_{21}(s) &= \frac{h_1}{q_b} = \frac{1}{2244s^2 + 485s + 2.27} & G_{22}(s) &= \frac{h_2}{q_b} = \frac{1}{270s + 3.98} \end{aligned} \quad (2-10)$$

2.5.1 Comparación entre sistema real, modelo no lineal y lineal

Con el fin de comprobar que tanto la modelización del sistema como su linealización se están haciendo de forma adecuada, se ha simulado su comportamiento y comparado con el comportamiento de la planta real (Figura 2-3).

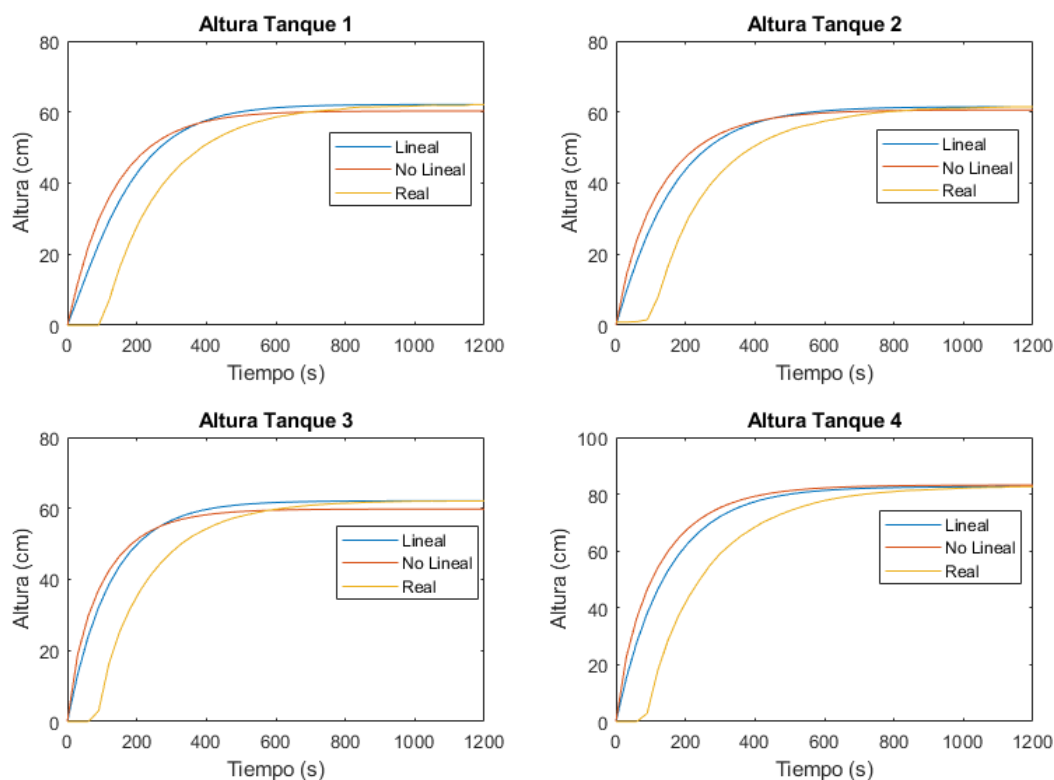


Figura 2-3. Comparación de comportamiento real-experimental respecto al simulado no lineal y lineal.

Como se puede comprobar, la planta real presenta un retardo notable en su comportamiento. Esto se debe a las no linealidades descritas en el apartado de Modelo teórico. A pesar de ello, el comportamiento es similar y válido para el futuro diseño de los PID que controlarán el sistema para la generación de la base de datos.

2.5.2 Significado físico de las funciones de transferencia

El significado físico de las funciones de transferencia para el sistema de los 4 tanques viene descrito por las interacciones que se producen al actuar sobre las variables de control (q_a y q_b). $G_{11}(s)$ describe la dinámica del tanque 1 cuando se modifica q_a . Análogamente, $G_{22}(s)$ describe la dinámica del tanque 2 cuando se modifica q_b . Por otra parte, $G_{12}(s)$ simula la dinámica del tanque 2 cuando se modifica q_a y $G_{21}(s)$ simula la dinámica del tanque 1 cuando se modifica q_b . Además, $G_{12}(s)$ y $G_{21}(s)$ representan las dinámicas de cómo los tanques superiores vierten agua en los inferiores.

Como se puede ver en dichas funciones de transferencia, la dinámica de $G_{11}(s)$ y $G_{22}(s)$ es más rápida que la de $G_{12}(s)$ y $G_{21}(s)$. Esto se debe a que, para llenar los tanques inferiores, es más rápido llenarlos directamente que a través de los tanques superiores. Por ello, $G_{12}(s)$ y $G_{21}(s)$ presentan dinámicas más lentas porque, para verter agua sobre los tanques inferiores, primero deben llenarse los superiores, introduciéndose así una dinámica de segundo orden. Las dinámicas de segundo orden podrían entenderse como un retraso ya que, al actuar sobre los caudales, la dinámica es tan lenta que no se presenta hasta varios tiempos de muestreo más tarde. Analizando las funciones de transferencia, el retraso rondaría entre 2 y 3 tiempos de muestreo.

2.5.3 Análisis de ceros y polos del sistema

En la Figura 2-4 se pueden ver los polos de cada una de las funciones de transferencia por separado. Como todos los polos son estables, el sistema es estable. Sin embargo, aunque no existan ceros en ninguna de las funciones de transferencia, éstos se presentan en la función de transferencia global del sistema (Figura 2-5).

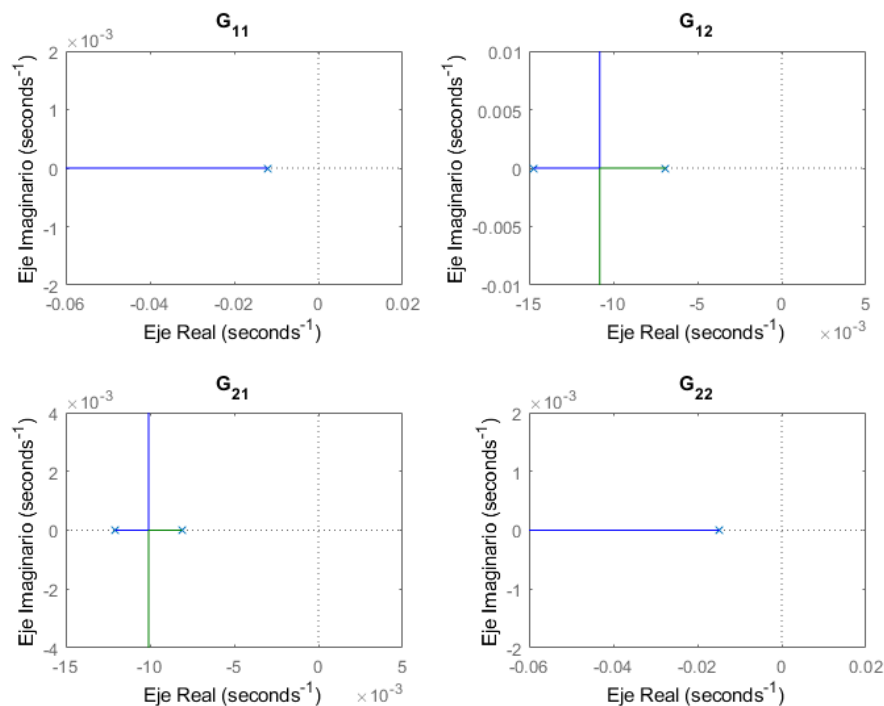


Figura 2-4. Lugar de las raíces de cada función de transferencia del sistema.

En el mapa de ceros y polos del sistema completo (Figura 2-5) hay un cero en el semiplano derecho y, por tanto, el sistema presenta un comportamiento de fase no mínima. Esto quiere decir que cuando cambia una variable controlada se ponen en marcha dos procesos [5]:

- Un proceso rápido, pero poco intenso.
- Otro proceso lento y más intenso que es el dominante.

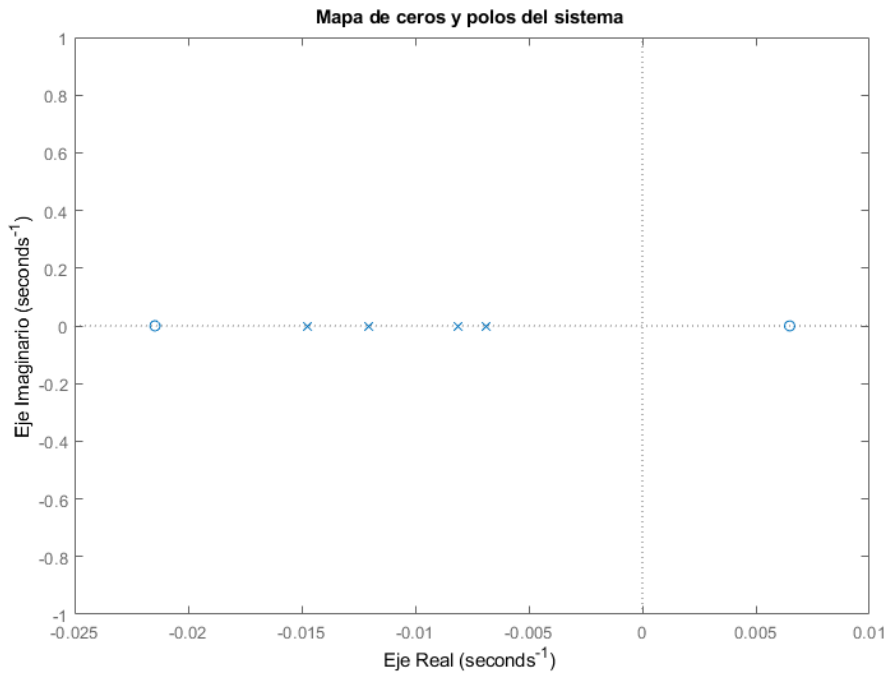


Figura 2-5. Mapa de ceros y polos del sistema.

Extrayendo las constantes de tiempo de cada función de transferencia y sus ganancias estáticas se obtiene la Tabla 2-2. En ella se puede ver cómo, G_{11} y G_{22} son más rápidas pero menos intensas que G_{12} y G_{21} , las cuales son más lentas y más intensas. Este comportamiento de fase no mínima no es muy intenso, pero influirá en el diseño de los controladores PID para la obtención de la base de datos. La forma de controlar un sistema de fase no mínima es mediante la desintonización de los controladores, haciéndolos más lentos, o mediante técnicas de control predictivo, como el Predictor de Smith, o incluso con controles MPC que ya incorporan la compensación de retrasos de forma natural.

	Símbolo	G_{11}	G_{22}	G_{12}	G_{21}
Tiempo de establecimiento	$t_s^{ij} = 3\tau_{ij}$	$3 \cdot 83 = 249$ s	$3 \cdot 68 = 204$ s	312 s	291 s
Ganancia estática	K_{ij}	0.23	0.25	0.46	0.44

Tabla 2-2. Parámetros de las funciones de transferencia.

2.5.4 Estudio de interacciones y emparejamiento de entradas y salidas

En los sistemas multivariables (MIMO) no es trivial el emparejamiento entre las variables de entrada y salida. Para el caso de la planta de los cuatro tanques, es necesario elegir correctamente qué bomba (q_a y q_b) controla cada tanque (h_1 y h_2). Para ello se va a utilizar el método de Bristol [4], el cual parte de la matriz de ganancias estáticas proveniente de las funciones de transferencia (2-10) (ya mostradas en la Tabla 2-2) y calcula la matriz de ganancias relativas (2-11). La matriz de ganancias relativas mide la influencia de cada entrada con cada salida, permitiendo emparejar la salida con la entrada que mayor influencia tiene sobre ella.

$$\Lambda = K * M = K * (K^{-1})^T = \begin{bmatrix} -0.40 & 1.40 \\ 1.40 & -0.40 \end{bmatrix} \quad (2-11)$$

Siendo $*$ el producto elemento a elemento, K la matriz de ganancias relativas y $M = (K^{-1})^T$. Como se pueden ver en (2-11), el único emparejamiento posible es:

- La entrada 2 con la salida 1 (q_b con h_1).
- La entrada 1 con la salida 2 (q_a con h_2).

Esto se debe a que la apertura de la válvula de 3 vías deja pasar bastante más caudal hacia los tanques superiores que a los inferiores, estando estos últimos cruzados.

3 GENERACIÓN DE LAS BASE DE DATOS

3.1 Mapa de puntos factibles

Para realizar los experimentos en la planta real es necesario tener en cuenta los límites físicos del sistema. Como ya se mostró en la Tabla 2-1, las alturas de los tanques no deben rebasar los 1.4 m, mientras que los caudales proporcionados por las bombas no pueden superar los $3 \text{ m}^3/\text{h}$. Debido a estas restricciones, se ha realizado un análisis de los puntos factibles a partir de los cuales el agua de los tanques no desborda o el caudal requerido no excede el físicamente posible.

Los tanques superiores son los más restrictivos ya que, la altura de los tanques superiores siempre es superior al de los inferiores, y nunca se alcanza el caudal máximo permitido. Por esta razón, se van a explorar dichos puntos barriendo el área de h_3 y h_4 entre 0.2 m y 1.2 m (Figura 3-1). Se elige un rango más reducido al que realmente se tiene para evitar que el sistema se acerque a los extremos, donde el comportamiento real se aleja del modelizado y, además, permitir sobreoscilaciones en el transitorio sin que desborde el agua.

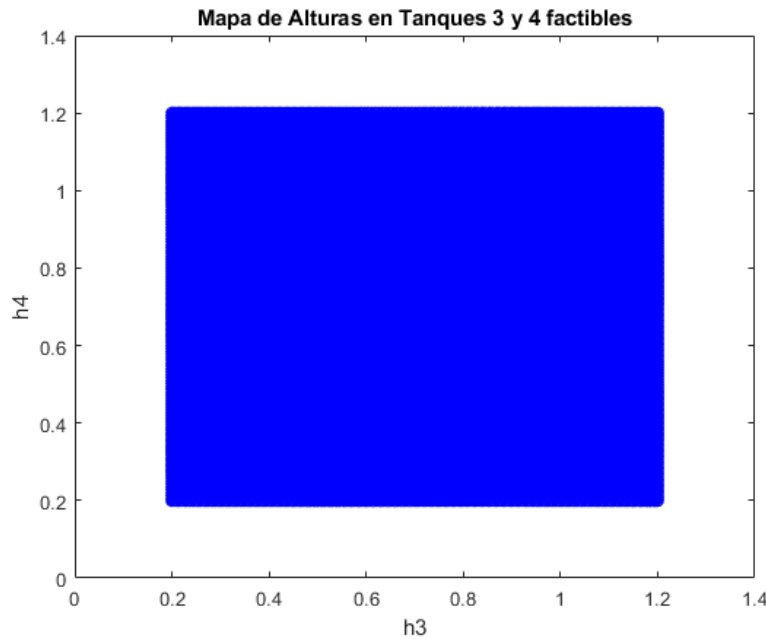


Figura 3-1. Mapa de puntos factibles para el Tanque 3 y el Tanque 4.

Para calcular estos puntos se ha utilizado el modelo del sistema (2-1) en régimen permanente (cuando las derivadas se anulan) y se han resuelto las ecuaciones descritas a continuación:

$$\left. \begin{aligned} q_a &= \frac{a_4 \sqrt{2gh_4}}{1-\gamma_a} \\ q_b &= \frac{a_3 \sqrt{2gh_3}}{1-\gamma_b} \end{aligned} \right\} \begin{aligned} h_1 &= \frac{\left(\frac{a_3 \sqrt{2gh_3} + \gamma_a q_a}{a_1} \right)^2}{2g} \\ h_2 &= \frac{\left(\frac{a_4 \sqrt{2gh_4} + \gamma_b q_b}{a_2} \right)^2}{2g} \end{aligned} \quad (3-1)$$

Con todo ello, los resultados son los mostrados en la Figura 3-2 en azul. Como se puede observar, se cumple lo anteriormente dicho de que los tanques inferiores nunca superan a los superiores y que los caudales nunca

exceden su máximo. Además, con este análisis se conoce todas las referencias factibles que hacen que el sistema no desborde ya que, para cada valor de h_1 elegido, se tiene un rango de valores de h_2 factibles.

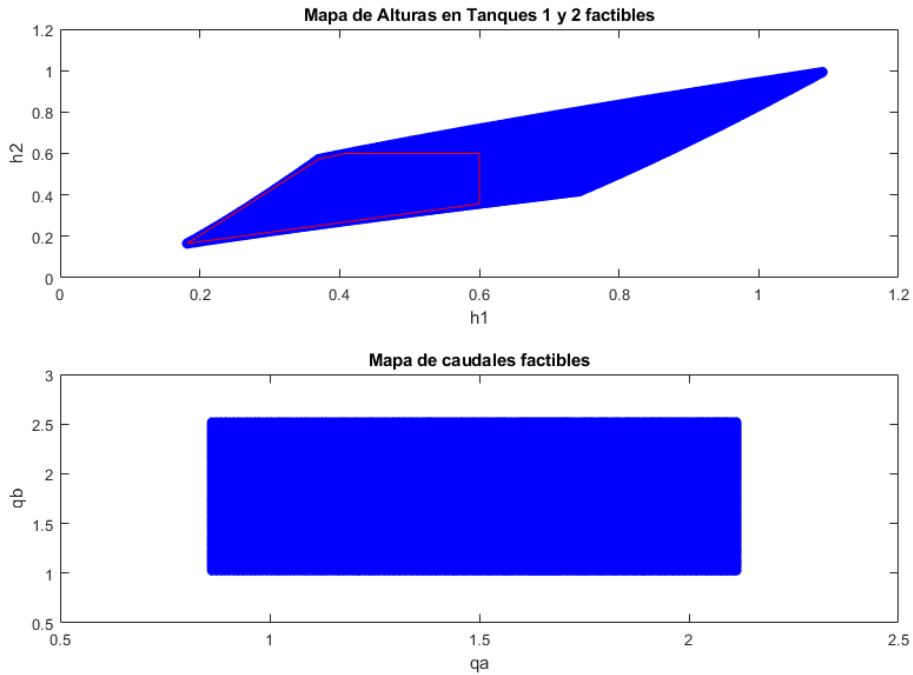


Figura 3-2. Mapa de puntos factibles en el Tanque 1 y Tanque 2 y en el caudal a y caudal b .

Aunque se dispone de todo el rango descrito en las referencias, se ha optado por reducir aún más el rango con el fin de acumular más trayectorias en el mismo entorno. Es por ello que todas las trayectorias obtenidas experimentalmente se sitúan entre 0.2 m y 0.6 m en los tanques inferiores, situándose dentro del área descrita en la Figura 3-2 en rojo.

3.2 Diseño de controladores

Al tratarse de un sistema multivariable donde se pretende tener 2 variables controladas (h_1 y h_2), es necesario disponer de 2 variables de control (q_a y q_b). Por tanto, es preciso de implementar 2 controladores, de tipo PID en este caso. Como ya se ha explicado, se va a controlar h_1 con q_b y h_2 con q_a siguiendo el esquema de la Figura 3-3.

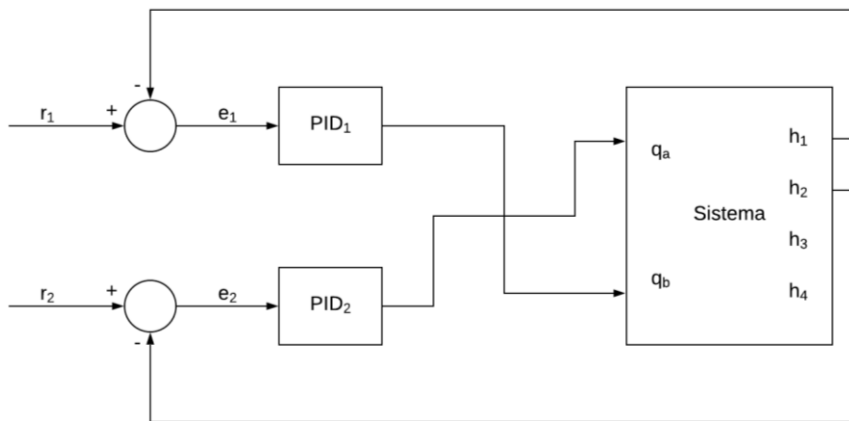


Figura 3-3. Diagrama de control con PID para la planta de los 4 tanques.

Se parte de un PID clásico como se describe a continuación:

$$PID(s) = \frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right) \quad (3-2)$$

La sintonización de los controladores se ha realizado utilizando Ziegler-Nichols en cada lazo de control para obtener unos valores de inicio. Se ha identificado el sistema en bucle abierto como un sistema de primer orden con retraso (3-3). Para ellos se ha variado q_a sin modificar q_b y se ha observado h_2 , haciéndolo de forma análoga con el tanque. Con ello se han obtenido los resultados de la Tabla 3-1.

$$G(s) = \frac{K}{1 + Ts} e^{-sL} \quad (3-3)$$

	K	T	L
Tanque 1	0.4729	180	60
Tanque 2	0.4562	180	60

Tabla 3-1. Parámetros de la identificación de la planta.

A partir de los sistemas identificados, y aplicando Z-N en bucle abierto, se han obtenido los siguientes controladores:

	K_p	T_i	T_d
PID_1	7.6	120	30
PID_2	7.9	120	30

Tabla 3-2. Parámetros iniciales de los PID.

Posteriormente se ha ajustado los valores obtenidos llegando a un compromiso entre sobreoscilación, tiempo de subida y tiempo de establecimiento. Además, para enriquecer la base de datos, se ha dado un rango para todos estos parámetros, los cuales rondan los valores obtenidos tras el ajuste fino.

	K_p	T_i	T_d
PID_1	1 - 3.5	300 - 500	10 - 30
PID_2	1 - 3.5	300 - 500	10 - 30

Tabla 3-3. Rango de parámetros de los PID.

3.3 Obtención de trayectorias

Las trayectorias se han obtenido siguiendo un mismo procedimiento. Por las razones ya explicadas, el rango de referencias obtenidas abarca entre 0.2 m y 0.6 m en los tanques inferiores. Además, se utilizan controladores aleatorios con un rango de parámetros descritos en la Tabla 3-3.

El procedimiento seguido se basa, para un experimento (Figura 3-4), en arrancar la planta en bucle abierto para unos caudales iniciales aleatorios contenidos entre 1.3 m³/h y 1.9 m³/h, controlar el sistema con unos PID aleatorios hasta que el error acumulado de las últimas 100 muestras sea menor de 0.35, guardar los datos

(alturas, caudales y referencias aplicados en cada tiempo de muestreo, junto con el tiempo en el que ocurre), y terminar el experimento. Sin embargo, para optimizar el tiempo y no tener que arrancar desde el principio la planta, se ha optado por cambiar la referencia en lugar de terminar el experimento. De esta forma es posible repetir el ciclo tantas veces como sea necesario ya que siempre se parte de un régimen permanente.

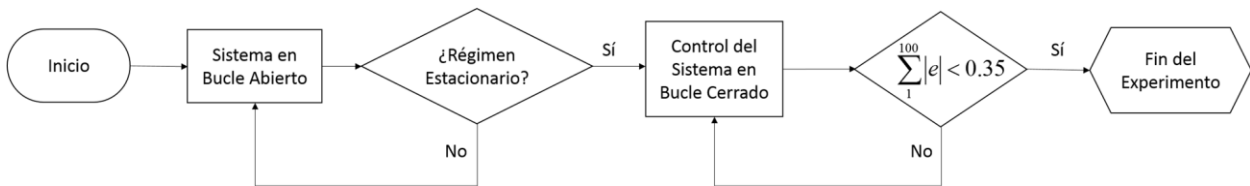


Figura 3-4. Procedimiento para la obtención de las trayectorias.

La ejecución de estas etapas se ha utilizado el código descrito en el Anexo: Códigos usados en el apartado de Generación de las trayectorias.

- El bucle abierto inicial hasta alcanzar el régimen permanente tarda alrededor de 20 min.
- Se cierra el bucle y se espera hasta que el valor absoluto de los errores en los 100 últimos tiempos de muestra sea inferior de 0.35. Ese momento, se guardan los datos y se cambia la trayectoria volviendo a esperar el tiempo necesario para alcanzar un error de 0.35. Cada obtención de trayectoria puede necesitar entre 1.5 h y 2 h, dependiendo del punto inicial, de los parámetros de los controladores y de las referencias asignadas.

El código se ha realizado en MATLAB para la generación de los parámetros de los controladores, referencias, caudales iniciales, etc. mientras que la simulación y gestión de la planta real se ha realizado mediante Simulink (Figura 3-5) y un cliente OPC para la comunicación con los autómatas.

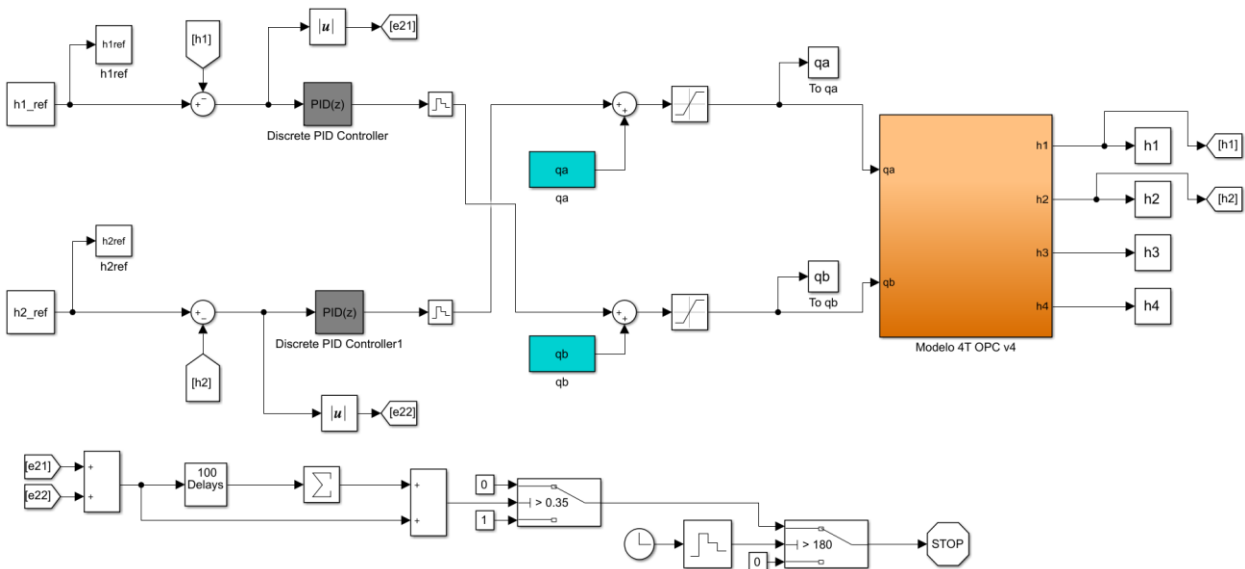


Figura 3-5. Modelo en Simulink de la generación de trayectorias.

3.4 Resultados

Aunque las simulaciones relevantes se han realizado en la planta real, se han simulado todas las trayectorias con el modelo descrito en los apartados anteriores con el fin de obtener una base de datos similar a la real. La razón de esto es que, mientras la obtención de las trayectorias en la planta real ha tomado meses, en simulación requiere de minutos. Por otra parte, la simulación de las trayectorias es mucho más sencilla ya que no es necesario conciliar con los problemas típicos de una planta real asociados a la supervisión, desgaste, mantenimiento, etc.

3.4.1 Resultados en simulación

Se han realizado dos conjuntos de trayectorias; una hasta 50 trayectorias (Figura 3-6) y la otra hasta las 100 trayectorias (Figura 3-7). Solo se van a mostrar estos dos conjuntos de trayectorias, aunque la ampliación de esta base de datos es sencilla.

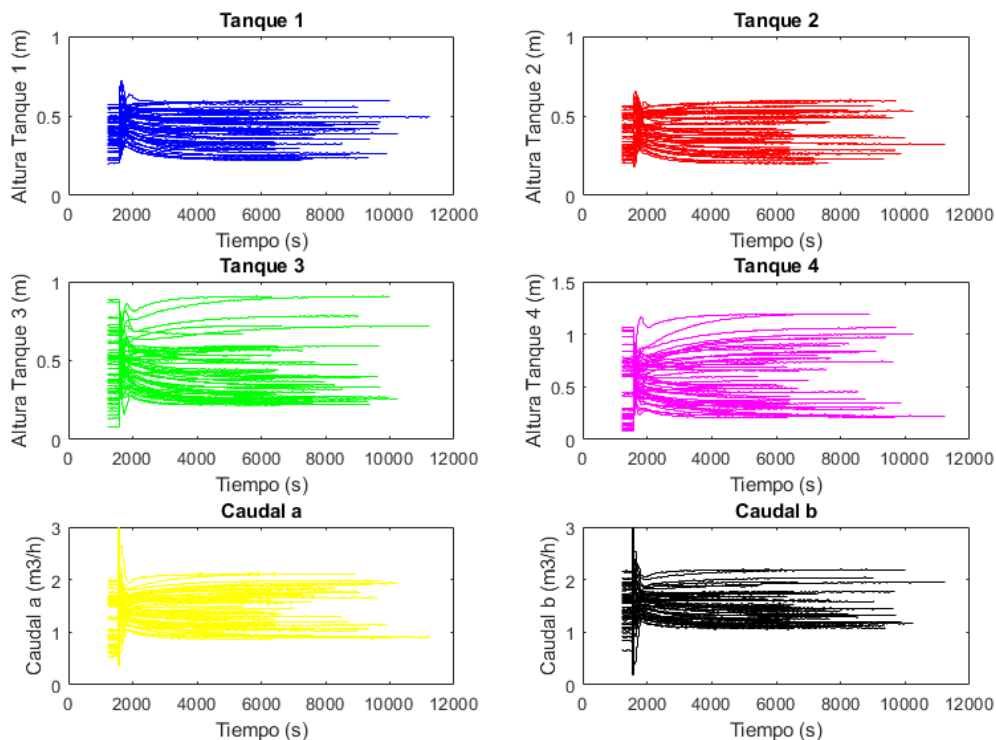


Figura 3-6. Base de datos de 50 trayectorias simuladas.

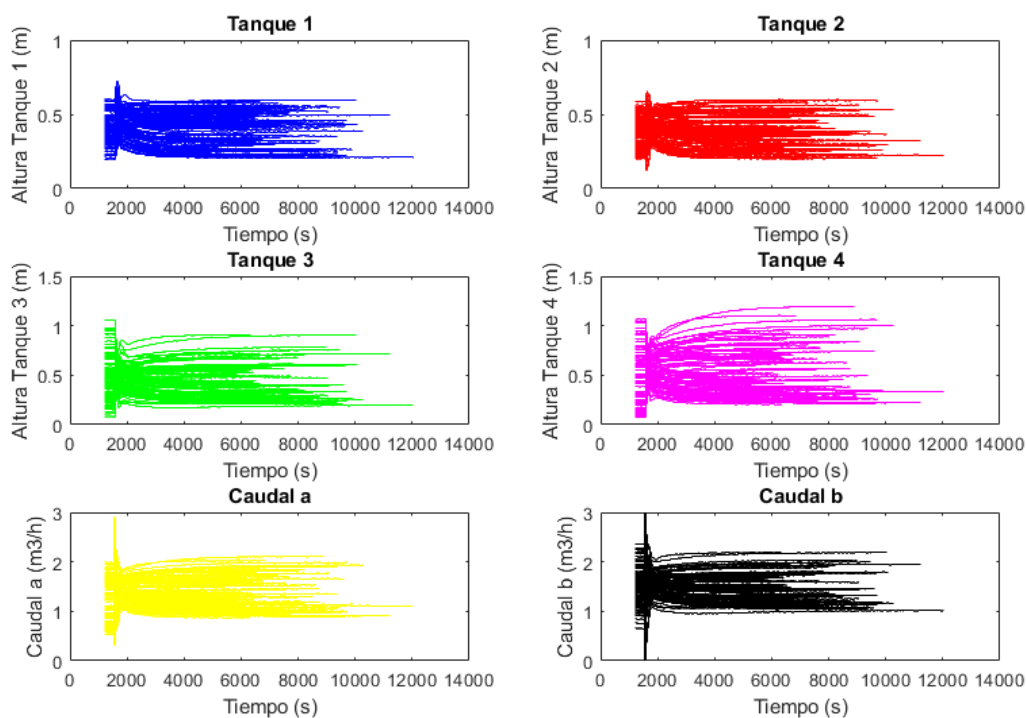


Figura 3-7. Base de datos de 100 trayectorias simuladas.

3.4.2 Resultados en la planta real

De la misma forma que en simulación, se han desarrollado dos conjuntos de trayectorias; una hasta 50 trayectorias (Figura 3-8) y la otra hasta las 100 trayectorias (Figura 3-9). Se han separado para observar la mejora cuando se apliquen los algoritmos de optimización correspondientes.

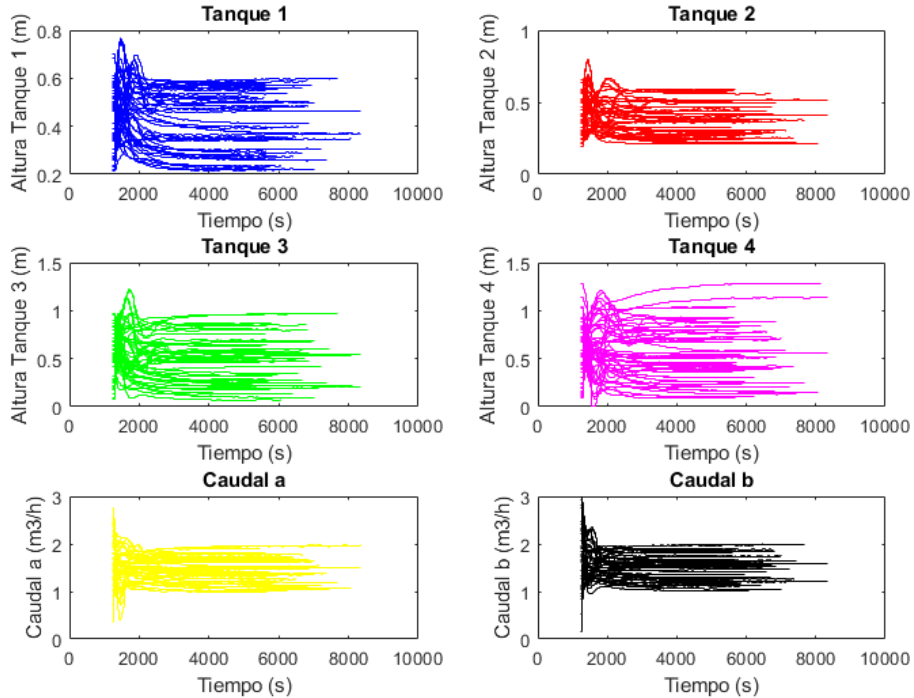


Figura 3-8. Base de datos de 50 trayectorias experimentales.

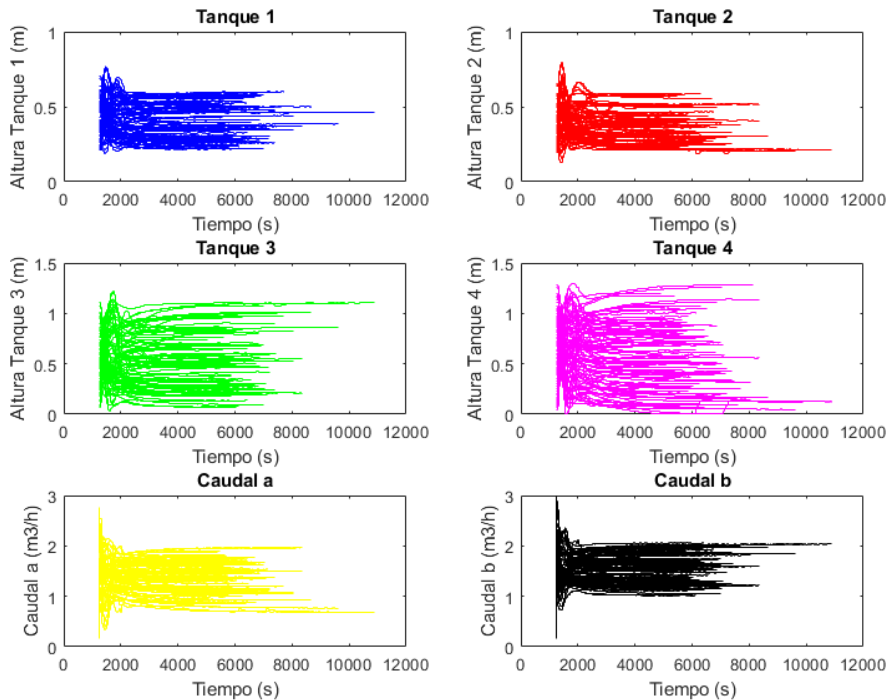


Figura 3-9. Base de datos de 100 trayectorias experimentales.

4 CONTROL PREDICTIVO BASADO EN DATOS

El Control Predictivo Basado en Datos (CPBD) [6] presenta un control que carece totalmente de modelo. La estrategia se basa en la proposición de que el sistema ha sido operado el suficiente tiempo, acumulando toda la información necesaria, para predecir el futuro de la evolución del control en bucle cerrado. Todo el conocimiento recogido a lo largo del tiempo es en bucle cerrado y, posiblemente, con diferentes estrategias de control. Con el CPBD es posible obtener una secuencia de controles óptimos en relación con una función de coste. La idea principal es que la base de datos sea lo más similar posible a las condiciones de operación deseadas, con el fin de computar la secuencia óptima de control de todas las trayectorias almacenadas. La estructura general del Control Predictivo Basado en Datos es la mostrada en la Figura 3-5, la cual es similar al MPC.

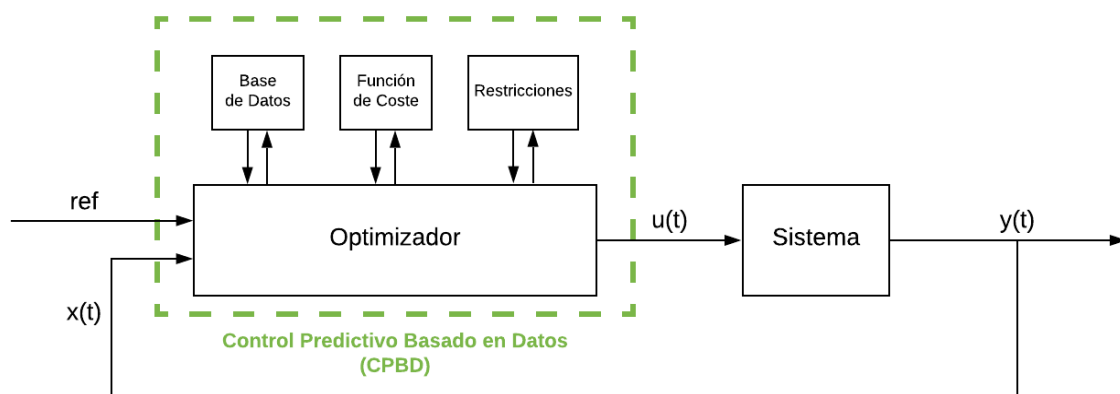


Figura 4-1. Estructura general del Control Predictivo Basado en Datos.

Esta estrategia de control puede ser aplicada a grandes procesos siempre y cuando se asegure una base de datos lo suficientemente grande. Un ejemplo de aplicación es la red de distribución de agua, el cual presenta problemas de control debido a su tamaño, naturaleza de los procesos y manipulación de las variables y la sensibilidad a perturbaciones en las condiciones de operación. Por estas razones, el CPBD se aplicó a la red de Richmond (Figura 4-2), del cual nace los algoritmos utilizados para la planta de los cuatro tanques.

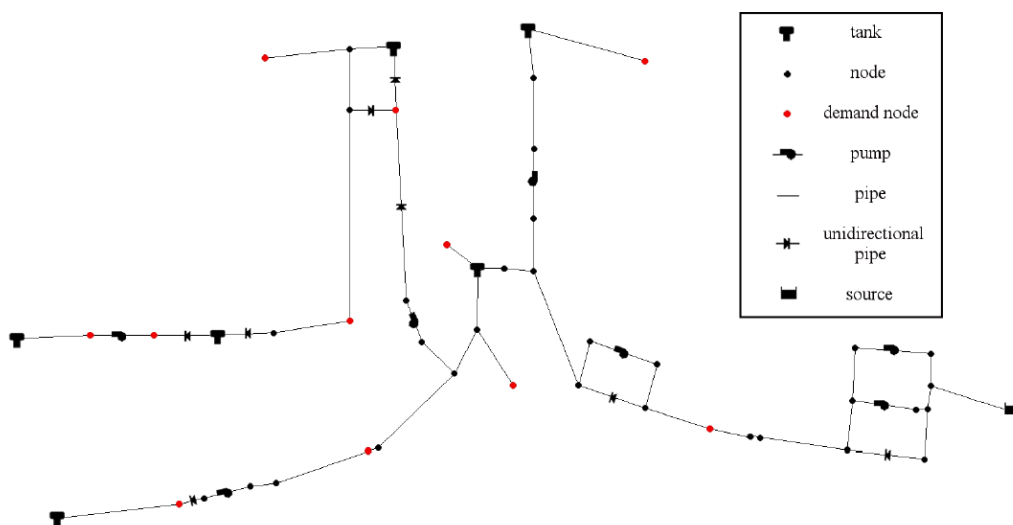


Figura 4-2. Diagrama de la red de distribución de aguas de Richmond.

4.1 Creación de la base de datos

Todas las trayectorias generadas se han introducido en una base de datos estática. Ésta recoge la información de las alturas de los 4 tanques, los caudales aplicados, la referencia aplicada y en instante en el que se muestreó. Además, se identifica cada trayectoria con un número identificador (ID). Como todas las trayectorias almacenadas no tienen el mismo número de muestras, ya que algunas alcanzan el régimen permanente antes y otras después, se han uniformizado el tamaño de las trayectorias para que todas sean iguales. Esto se ha conseguido manteniendo constante el último valor de la trayectoria tantas veces como sea necesaria para que alcance a la trayectoria más larga de la base de datos. De esta forma, la estructura de la base de datos es la reflejada en la Tabla 4-1.

ID	$t(s)$	$h_1(m)$	$h_2(m)$	$h_3(m)$	$h_4(m)$	$q_a(m^3/h)$	$q_b(m^3/h)$	$h_1^{ref}(m)$	$h_2^{ref}(m)$
1									
⋮									
2									
⋮									
⋮									
M									

Tabla 4-1. Estructura de la base de datos.

4.2 Formulación de los algoritmos de control

Se van a utilizar 2 algoritmos de CPBD en este trabajo: El algoritmo Data-Driven MPC [6] y el algoritmo de Offset-free [7], [8]. El algoritmo Data-Driven MPC fue el utilizado inicialmente, siendo, además, el algoritmo utilizado anteriormente en la planta de los 4 tanques, aunque tan solo en regulación. Las conclusiones de dicho trabajo [1], y de otros similares [9], dejaban ver la existencia de un pequeño offset. Por esta razón, se desarrolló el algoritmo de Offset-free, el cual pretende eliminar el offset de forma teórica para todo sistema lineal.

La principal diferencia entre los algoritmos descritos, además del offset comentado, es que el segundo (Offset-free) es más complejo, aumentando la dimensión del problema y complicándolo computacionalmente. Además, el algoritmo de Offset-free elimina las restricciones de desigualdad, lo cual permite acciones de control negativas.

4.2.1 Algoritmo Data-Driven MPC

En primer lugar, el algoritmo necesita seleccionar el subset de la base de datos que haga la mejor predicción de la evolución de los estados del sistema. Las trayectorias seleccionadas deben ser lo más cercanas posibles al estado actual del sistema. Para resolver este problema se utiliza la función de distancia definida a continuación:

$$dist_{BD}(x(t), n_r) = \left| x(t) - x_{n_r} \right|_{\lambda}^2 + \left| x_r - x_{n_r}^r \right|_{\lambda_r}^2 \quad (4-1)$$

Donde $x(t)$ es el vector de estado actual, n_r es la fila de la base de datos (Tabla 4-1), x_{n_r} es el vector de estado asociado a la fila n_r , λ es un vector de ponderación que relaciona los componentes de los estados, x_r es el valor de la referencia actual, $x_{n_r}^r$ es la referencia asociado a x_{n_r} y λ_r es un vector de ponderación que relaciona los componentes de los estados de referencia. Usando la función de distancia es posible encontrar las

Q filas de la base de datos más cercanas al estado actual $x(t)$ con al menos $N-1$ (con N siendo el horizonte de predicción) filas consecutivas restantes. Estas filas se consideran los estados iniciales de las trayectorias con una longitud de N tiempos de muestreo. Considerando que el primer estado de la trayectoria candidata q es almacenada en la fila n_r , se definen los estados de la q -ésima trayectoria candidata, $x_q(k)$, como la k -ésima fila que precede al estado después de la fila inicial n_r de la trayectoria candidata. De la misma forma, la notación se extiende a las variables de entrada de referencia x_q^r y de control $u_q(k)$.

Es remarcable que, cuando mayor sea Q , más probable es que existan candidatos que aseguren un espacio convexo a partir del cual asegurar la correcta optimización del problema (Figura 4-3). De la misma forma, Q podrá tener suficiente información cuanto mayor sea la base de datos almacenada.

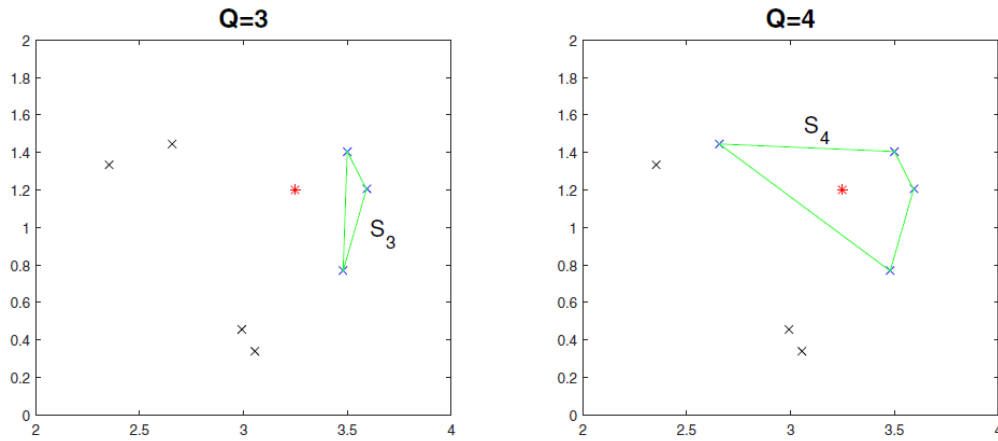


Figura 4-3. Ejemplo de problema de factibilidad. A la izquierda no es factible, a la derecha sí lo es.

Por otro lado, la función de coste (4-2), definida para resolver el problema de optimización, penaliza el error en el estado actual y el horizonte de predicción (N).

$$J_q = \sum_{k=0}^{N-1} (x_q(k) - x_r)^2 \quad (4-2)$$

Finalmente, el problema de optimización se plantea como la combinación convexa de las secuencias de control de cada trayectoria candidata de menor coste.

$$\begin{aligned} \beta_q^* &= \arg \min_{\beta_q} \sum_{q=1}^Q \beta_q J_q \\ \text{s.a.} \quad x(t) &= \sum_{q=1}^Q \beta_q x_q(0) \\ x_r &= \sum_{q=1}^Q \beta_q x_q^r \\ \sum_{q=1}^Q \beta_q &= 1 \\ \beta_q &\geq 0 \quad \forall q \in [1, \dots, Q] \end{aligned} \quad (4-3)$$

La acción óptima que se aplica tras resolver el problema anterior se calcula como:

$$u^*(t) = \sum_{q=1}^Q \beta_q u_q(0) \quad (4-4)$$

La dos primeras restricciones impuestas en (4-3) aseguran la combinación convexa de las trayectorias, siendo

la segunda necesaria en caso de implementar seguimiento de trayectorias. De esta forma, la combinación convexa de las trayectorias candidatas es una estimación de la trayectoria del estado si la combinación convexa de las acciones de control (4-4) se aplicase.

Debido a la reducida base de datos que se dispone, algunas restricciones duras comprometen la factibilidad del problema. Por esta razón, se ha variado el problema de optimización para asegurar la factibilidad de este. En primer lugar, se ha eliminado la restricción dura que asegura la combinación convexa de las referencias ($\sum_{q=1}^Q \beta_q x_q^r = x_r$), convirtiéndola en una restricción blanda, es decir, incluyéndola en la función de coste. De esta forma no se exige el cumplimiento de la restricción, pero al incluirla en la función de coste se obliga a que esa restricción sea lo más cercana posible a la restricción dura. Por tanto, si el problema (4-3) no resultara factible, se trataría de optimizar el siguiente:

$$\begin{aligned} \beta_q^* &= \arg \min_{\beta_q} \sum_{q=1}^Q \beta_q J_q + \left\| x_r - \sum_{q=1}^Q \beta_q x_q^r \right\|_2^2 \\ \text{s.a.} \quad x(t) &= \sum_{q=1}^Q \beta_q x_q(0) \\ \sum_{q=1}^Q \beta_q &= 1 \\ \beta_q &\geq 0 \quad \forall q \in [1, \dots, q] \end{aligned} \quad (4-5)$$

Si este problema tampoco fuera factible, se volvería a partir del problema inicial (4-3), solo que esta vez se trataría de relajar la última restricción ($\beta_q \geq 0$). La solución para relajar esta restricción consiste en implementar una cierta holgura a los valores de β_q , de tal forma que se permita que sea ligeramente negativos. El hecho de aceptar valores de β_q negativos no es realista ya que no es posible implementar acciones de control negativas, por esa misma razón se resuelve este problema en tercer lugar. Sin embargo, al realizar la combinación convexa de las acciones de control (4-4) resulta ser siempre positivo. Por otra parte, para que la combinación de los valores de β con la función de coste no se anulen (debido a los β negativos), se ha elevado al cuadrado esta combinación. Finalmente, se ha denotado como τ a los parámetros que permiten dichas holguras y como α al parámetro que pondera el peso en la función de coste. Introduciendo en la función de coste e implementado las restricciones se obtiene el siguiente problema:

$$\begin{aligned} \beta_q^* &= \arg \min_{\beta_q} \sum_{q=1}^Q \left(\beta_q^2 J_q^2 + \alpha (\tau_q^2 + \tau_q) \right) \\ \text{s.a.} \quad x(t) &= \sum_{q=1}^Q \beta_q x_q(0) \\ x_r &= \sum_{q=1}^Q \beta_q x_q^r \\ \sum_{q=1}^Q \beta_q &= 1 \\ \beta_q + \tau_q &\geq 0 \quad \beta_q - \tau_q \leq 1 \quad \tau_q \geq 0 \quad \forall q \in [1, \dots, q] \end{aligned} \quad (4-6)$$

Si todavía no fuera factible ninguno de los métodos utilizados, la solución pasaría por unir las dos estrategias anteriores. Por un lado, eliminar la restricción dura de tracking, volviéndola una restricción blanda en la función de coste, y por el otro lado, introduciendo holguras como las anteriormente descritas. Esta opción no ha llegado a necesitar en las simulaciones ni en los experimentos, pero se ha introducido para asegurar la factibilidad en cualquier caso y que no se dejase de controlar. La última opción después de ésta sería la de aplicar la acción de control de la trayectoria más cercana a la actual.

4.2.1.1 Implementación en Simulink

El diagrama utilizado en Simulink se describe en la Figura 4-4. El funcionamiento es el mismo que el descrito en la Figura 4-1; se introducen las referencias al optimizador junto con el estado actual del sistema, el optimizador resuelve los problemas planteados, calcula la acción de control y la aplica al sistema. Esto se repite en cada tiempo de muestreo consiguiendo el control de la planta. El código utilizado dentro del bloque "CPBD" de la Figura 4-4 está desarrollado en el Anexo: Códigos usados, en el apartado de Algoritmo Data-Driven MPC. El mismo diagrama es el utilizado en simulación y en la planta real. La única diferencia es que en simulación el bloque de la planta está simulado con las ecuaciones descritas en los apartados anteriores, mientras que en la planta real es un cliente OPC el que se encuentra dentro de dicho bloque y el cual gestiona las comunicaciones. Además, la base de datos que se introduce en las simulaciones es simulada, mientras que la base de datos de la planta real se corresponde la obtenida experimentalmente.

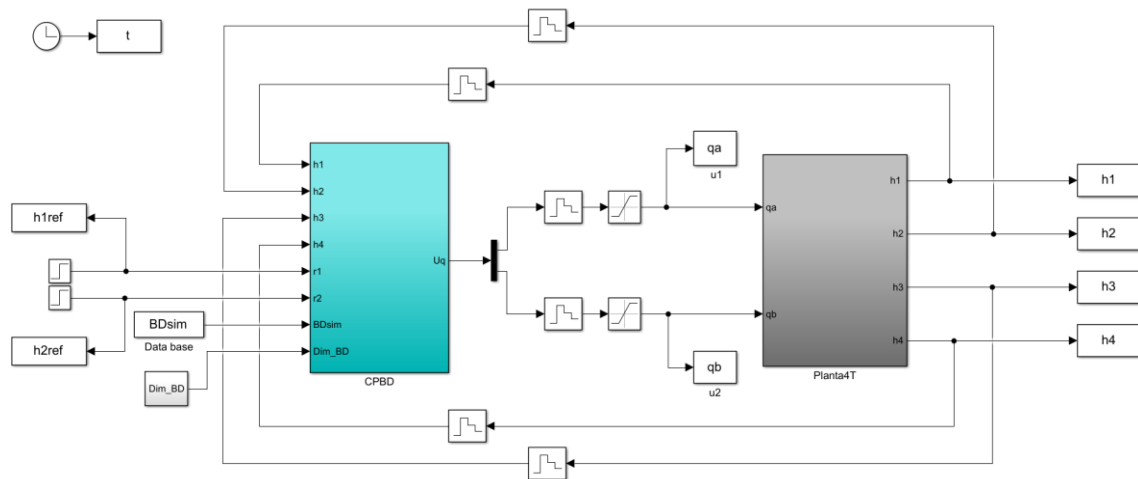


Figura 4-4. Diagrama de Simulink de la implementación del Control Predictivo Basado en Datos.

4.2.1.2 Resultados en simulación

Los resultados de simulación para 50 y 100 trayectorias se describen en la Figura 4-5 y Figura 4-6, respectivamente.

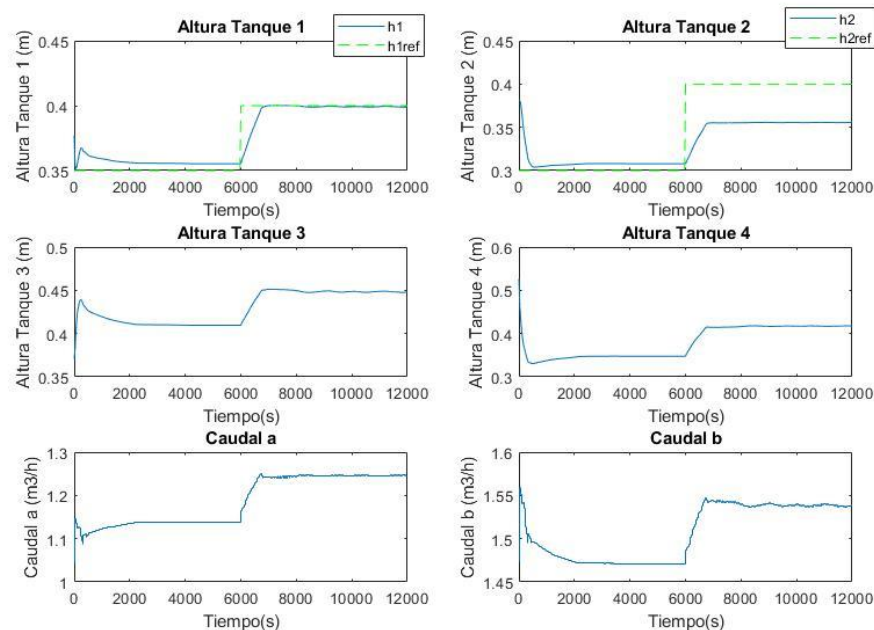


Figura 4-5. Altura de los tanques implementando el algoritmo Data-Driven MPC con una base de datos de 50 trayectorias simuladas.

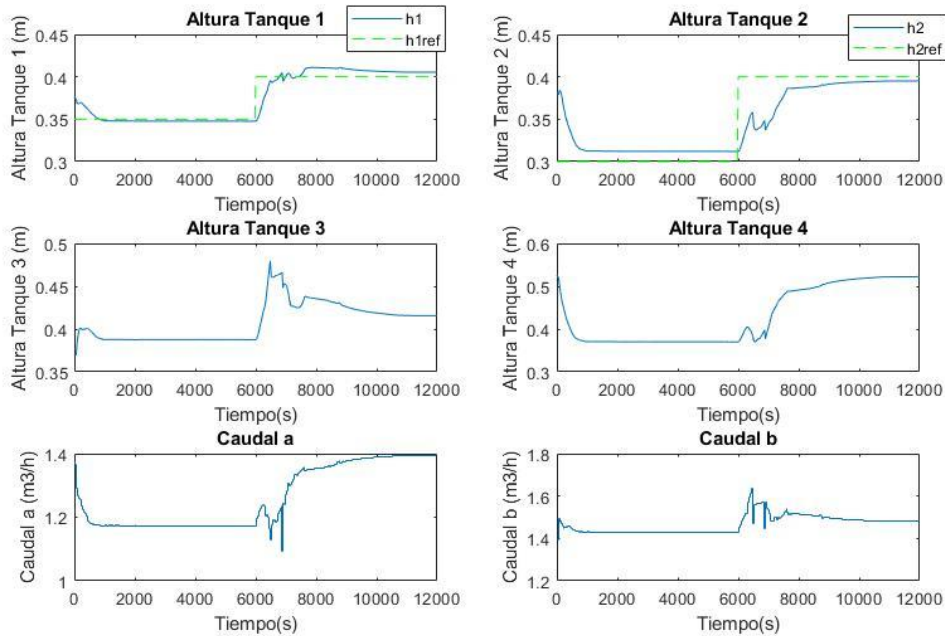


Figura 4-6. Altura de los tanques implementando el algoritmo Data-Driven MPC con una base de datos de 100 trayectorias simuladas.

Al carecer de un gran número de trayectorias, el offset que presenta este algoritmo es bastante elevado. Si el número de trayectorias creciera en orden de magnitud, el error en régimen permanente se reduciría de forma clara, como se ha simulado en la Figura 4-7. Además, se puede observar que el incremento en el número de trayectorias es positivo tanto en la respuesta transitoria como en el error en régimen permanente. Aun así, la respuesta presenta una fuerte dependencia de las trayectorias cercanas a la referencia dada. De esta forma, una base de datos de 50 trayectorias podría presentar mejores resultados que una de 100 siempre y cuando la referencia dada tenga suficientes trayectorias cercanas en la base de datos.

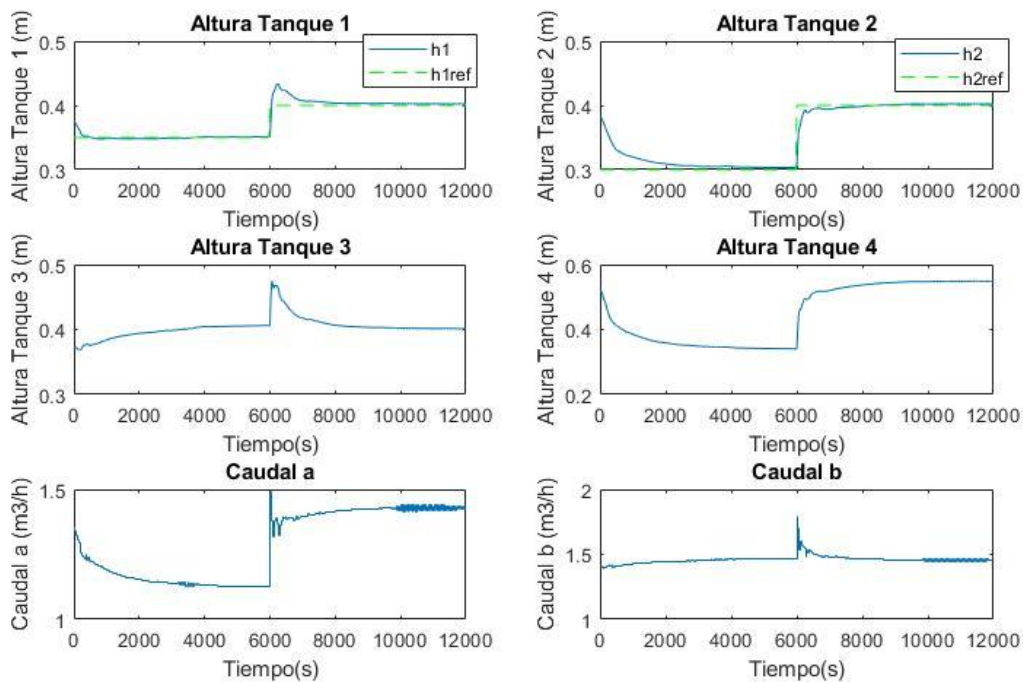


Figura 4-7. Altura de los tanques implementando el algoritmo Data-Driven MPC con una base de datos de 1000 trayectorias simuladas.

4.2.1.3 Resultados en la planta real

Los resultados de simulación para 50 y 100 trayectorias se describen en la Figura 4-8 y, Figura 4-9 respectivamente.

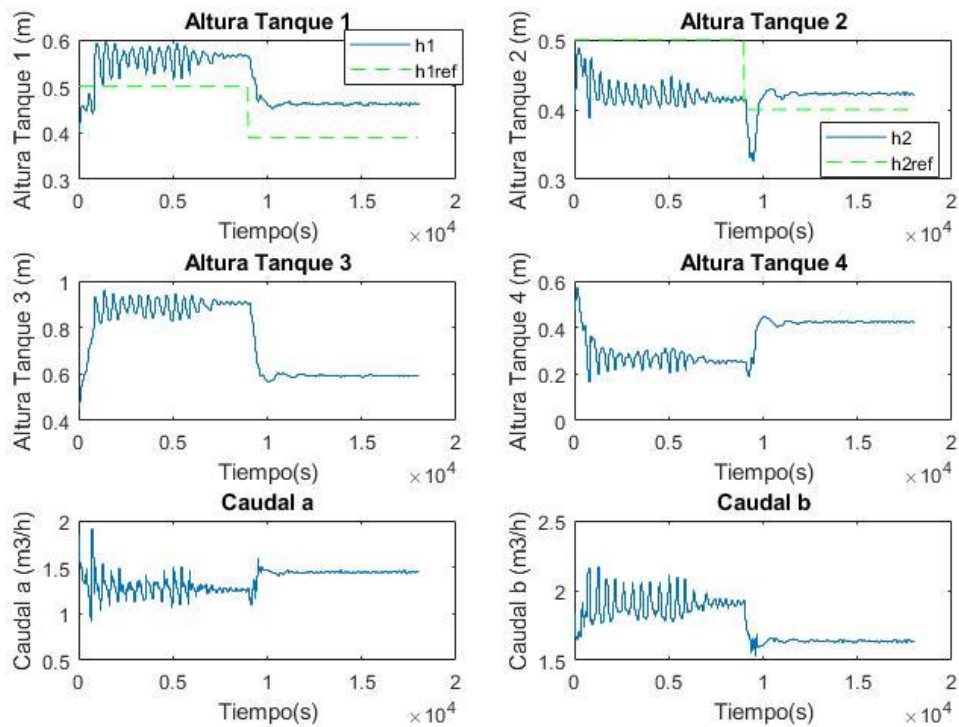


Figura 4-8. Altura de los tanques implementando el algoritmo Data-Driven MPC con una base de datos de 50 trayectorias reales.

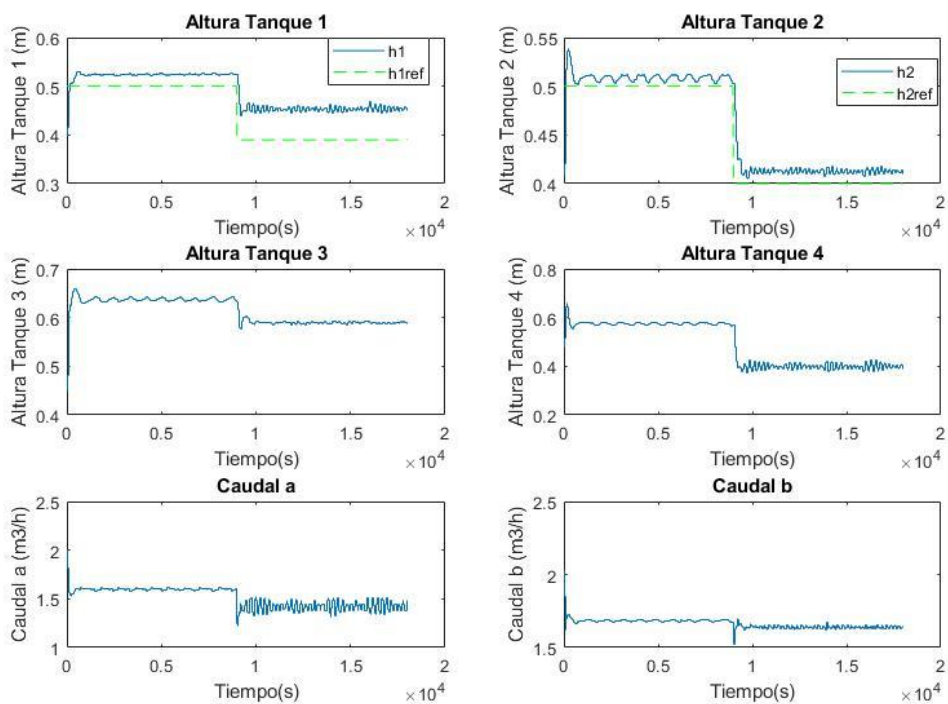


Figura 4-9. Altura de los tanques implementando el algoritmo Data-Driven MPC con una base de datos de 100 trayectorias reales.

4.2.2 Algoritmo de Offset-free

El algoritmo de Offset-free se centra en eliminar el offset producido en el anterior algoritmo. Aunque solo es válido en sistemas lineales [7], [8], el algoritmo puede mejorar los resultados anteriormente obtenidos minimizando el error.

Inicialmente, se define una función distancia para elegir a los Q candidatos. Se va a incluir el estado actual junto con las referencias, pero, además, se va a tener en cuenta los estados anteriores más los estados controlados (siendo $y(t)$ los estados controlados actuales y y_q los estados en la base de datos) junto con las acciones de control anteriores. Definiendo $z(t)$ y z_q como (4-7), se define la función distancia según (4-8).

$$z(t) = [x(t) \quad y(t) \quad x_r(t) \quad x(t-1) \quad u(t-1)]^T \quad (4-7)$$

$$z_q = [x_q(0) \quad y_q(0) \quad x_q^r \quad x_q(-1) \quad u_q(-1)]^T$$

$$dist_{BD}(z(t), z_q) = \|z(t) - z_q\|_2^2 \quad (4-8)$$

Una vez reordenado el vector de distancia y elegidos las Q trayectorias candidatas se resuelve el problema de optimización siguiente:

$$\begin{aligned} \beta_q^* &= \arg \min_{\beta_q} \sum_{q=1}^Q \beta_q^2 \\ \text{s.a.} \quad x(t) &= \sum_{q=1}^Q \beta_q x_q(0) \\ x(t-1) &= \sum_{q=1}^Q \beta_q x_q(-1) \\ u(t-1) &= \sum_{q=1}^Q \beta_q u_q(-1) \\ y(t) &= \sum_{q=1}^Q \beta_q y_q(0) \\ x_r &= \sum_{q=1}^Q \beta_q x_q^r \\ \sum_{q=1}^Q \beta_q &= 1 \end{aligned} \quad (4-9)$$

Realmente, la función de coste trata de minimizar la esperanza del error cuadrático, pero se asume que éste nunca es superior que la desviación típica y, por tanto, puede suponerse constante, eliminándose de la función de optimización. Por otro lado, continúan las restricciones que imponen una reconstrucción del estado actual junto con la restricción de que el sumatorio de las β_q sea la unidad. Además de las ya explicadas, se incluyen 3 más, cuya función es reconstruir los estados anteriores. Esto proporciona al algoritmo un conocimiento de los errores pasados, de tal forma que se consiga integrar el error de forma similar al controlador PI.

Un inconveniente de este algoritmo está asociado a que no existe ninguna restricción que imponga que los valores de β_q sean positivos, lo cual implica que podría calcularse una acción de control negativa en un instante determinado. Para la simulación realizada en la Figura 4-13 es posible ver que en torno al 50% de las β_q son negativas, sin embargo, la acción de control total en este experimento siempre era positivo (Figura 4-10).

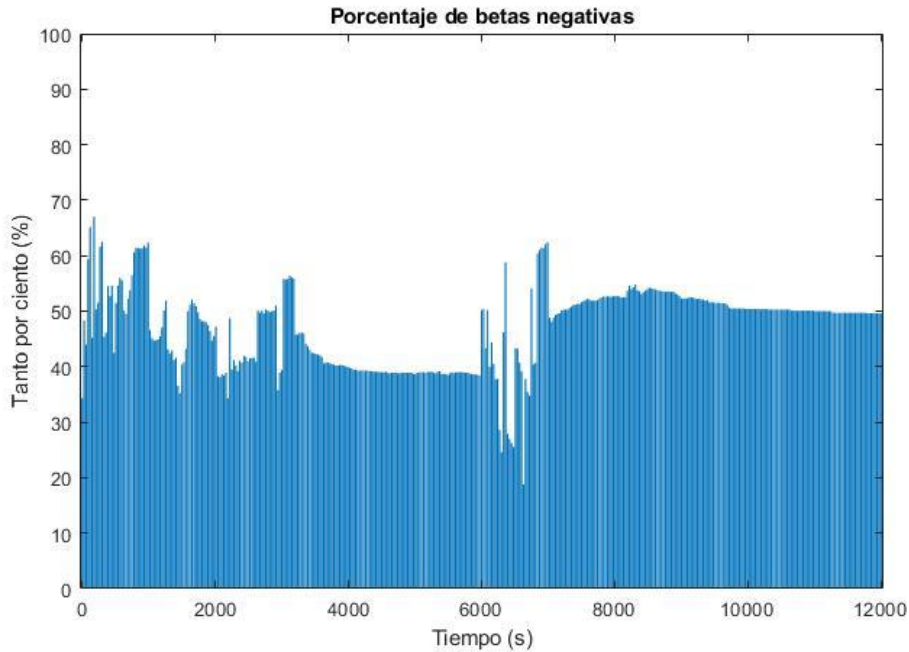


Figura 4-10. Porcentaje de betas negativas para la simulación descrita en la Figura 4-13.

4.2.2.1 Implementación en Simulink

Para la implementación del algoritmo de Offset-free se ha utilizado el código descrito en el Anexo: Códigos usados en el apartado de Algoritmo de Offset-free. Además, el diagrama de Simulink utilizado es el mismo que en la Figura 4-4, con la salvedad de que el bloque de CPBD se ha modificado para introducir un retraso de un tiempo de muestreo en las variables necesarias para la aplicación del código (Figura 4-11).

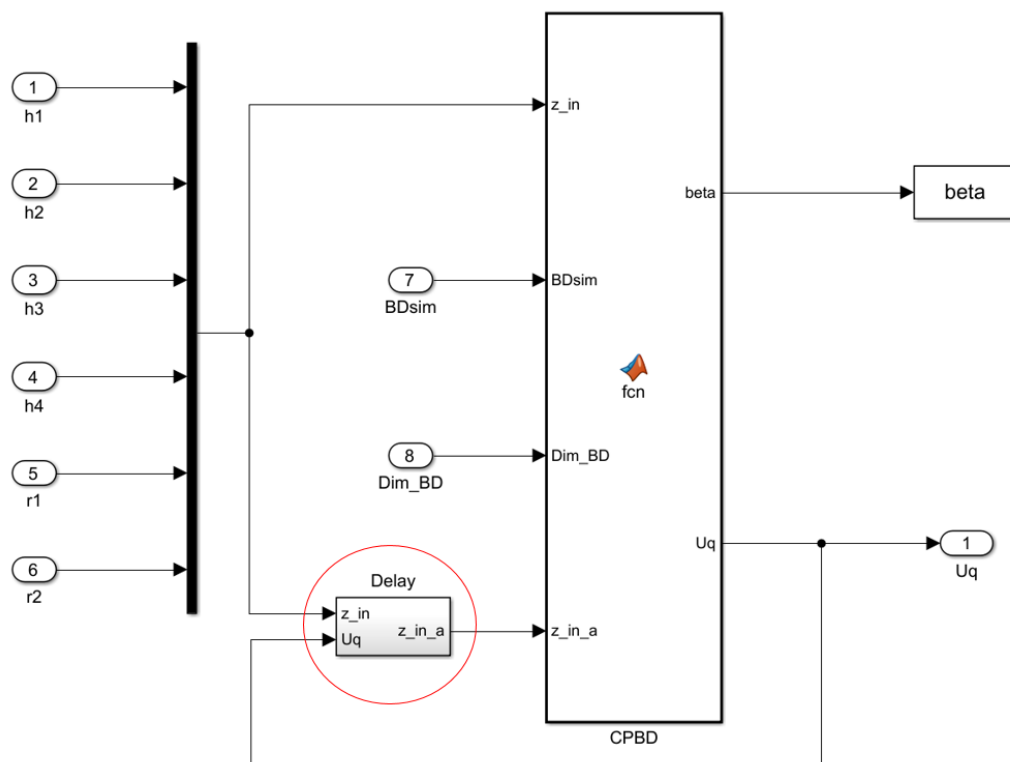


Figura 4-11. Diagrama de bloques del subsistema CPBD para el algoritmo de Offset-free.

4.2.2.2 Resultados en simulación

Las simulaciones realizadas para 50 y 100 trayectorias están descritas en las Figura 4-12 y Figura 4-13, respectivamente.

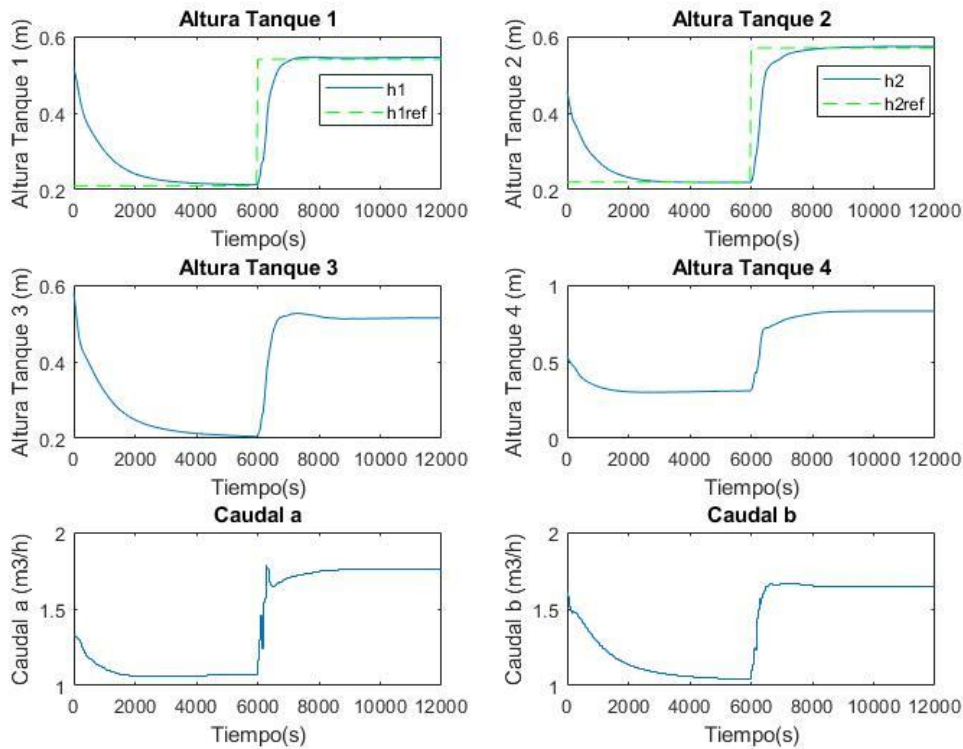


Figura 4-12. Altura de los tanques implementando el algoritmo de Offset-free con una base de datos de 50 trayectorias simuladas.

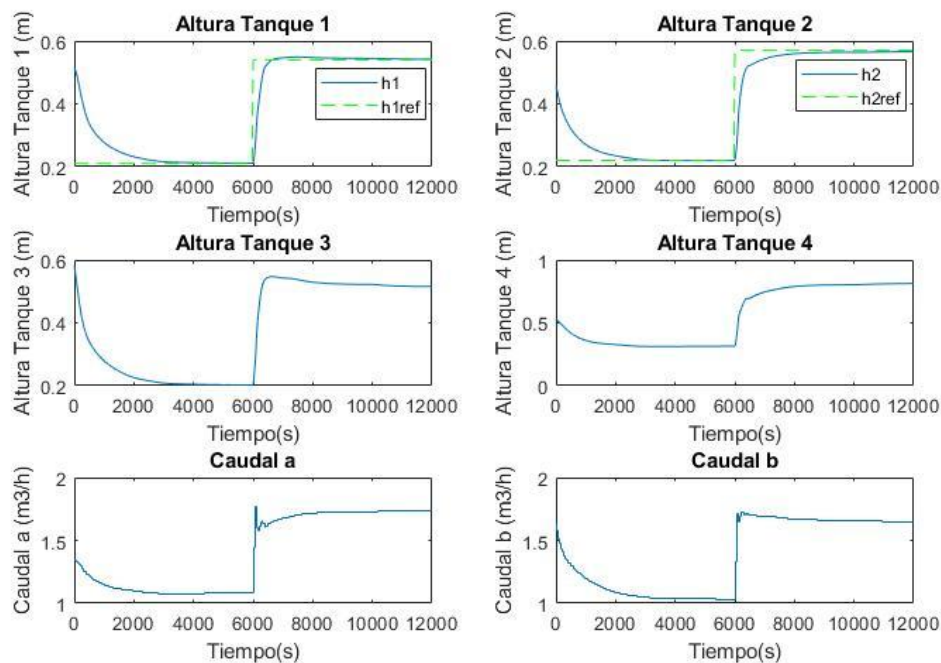


Figura 4-13. Altura de los tanques implementando el algoritmo de Offset-free con una base de datos de 100 trayectorias simuladas.

4.2.2.3 Resultados en la planta real

Las simulaciones realizadas para 50 y 100 trayectorias están descritas en la Figura 4-14 y Figura 4-15, respectivamente.

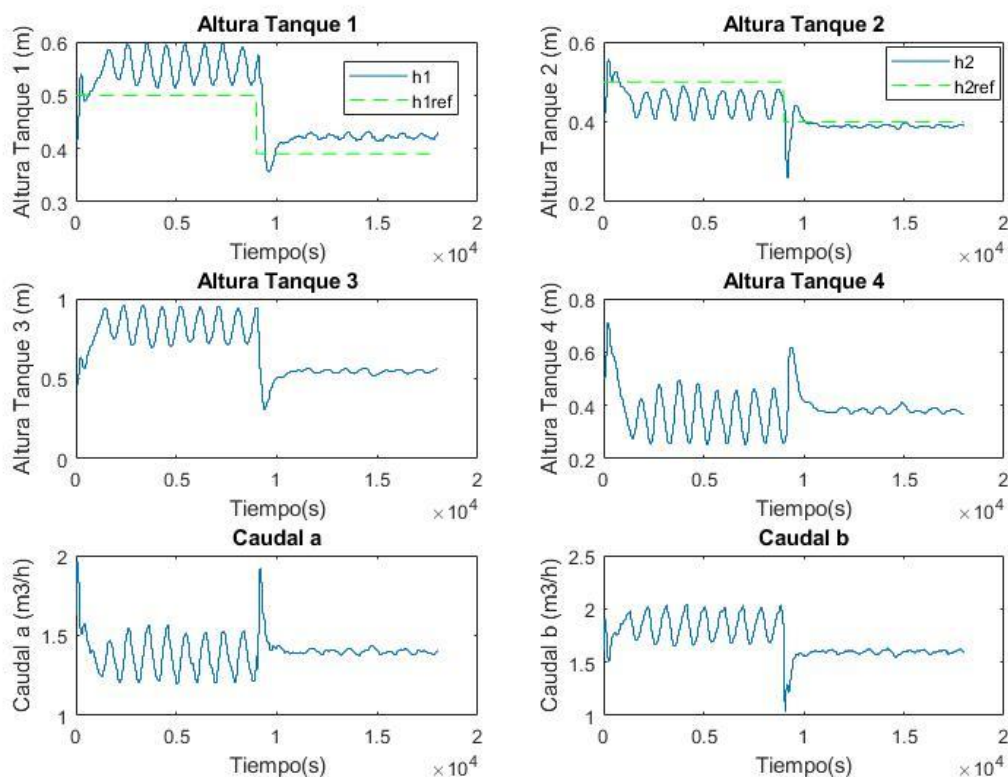


Figura 4-14. Altura de los tanques implementando el algoritmo de Offset-free con una base de datos de 50 trayectorias reales.

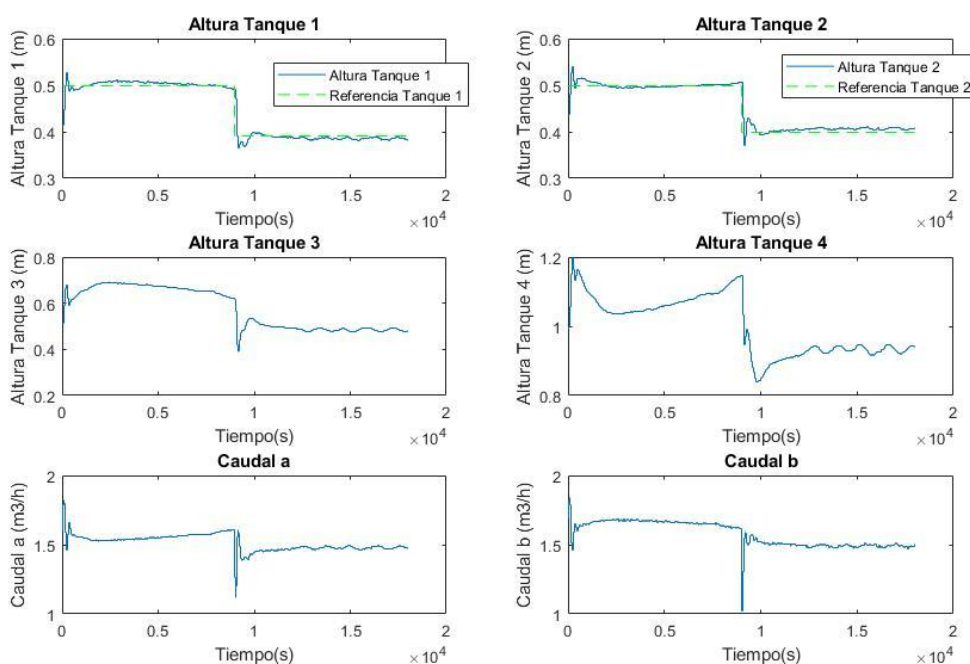


Figura 4-15. Altura de los tanques implementando el algoritmo de Offset-free con una base de datos de 100 trayectorias reales.

4.3 Comparación del algoritmo Data-Driven MPC y Offset-free Data-Driven MPC

Una vez vistos los resultados de ambos algoritmos, es interesante ver las diferencias para un mismo punto de referencia (Figura 4-16 y Figura 4-17). La principal diferencia es la reducción del offset de un algoritmo a otro para la misma base de datos. Con ello se demuestra el correcto funcionamiento. Sin embargo, ambos algoritmos son perfectamente válidos una vez la base de datos obtenida sea lo suficientemente grande, aunque no haya sido el caso para este trabajo.

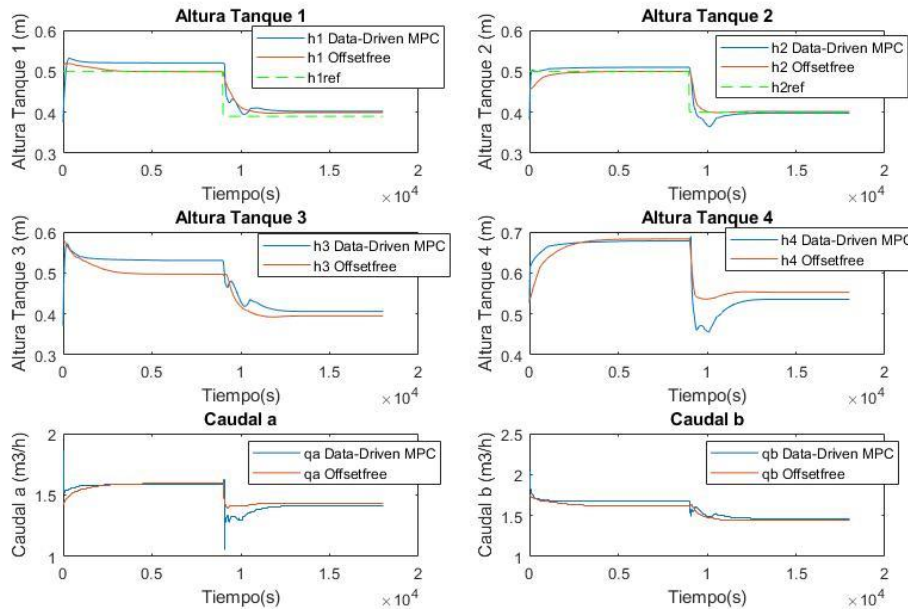


Figura 4-16. Altura de los tanques comparando el algoritmo Data-Driven MPC con el Offsetfree con una base de datos de 100 trayectorias simuladas.

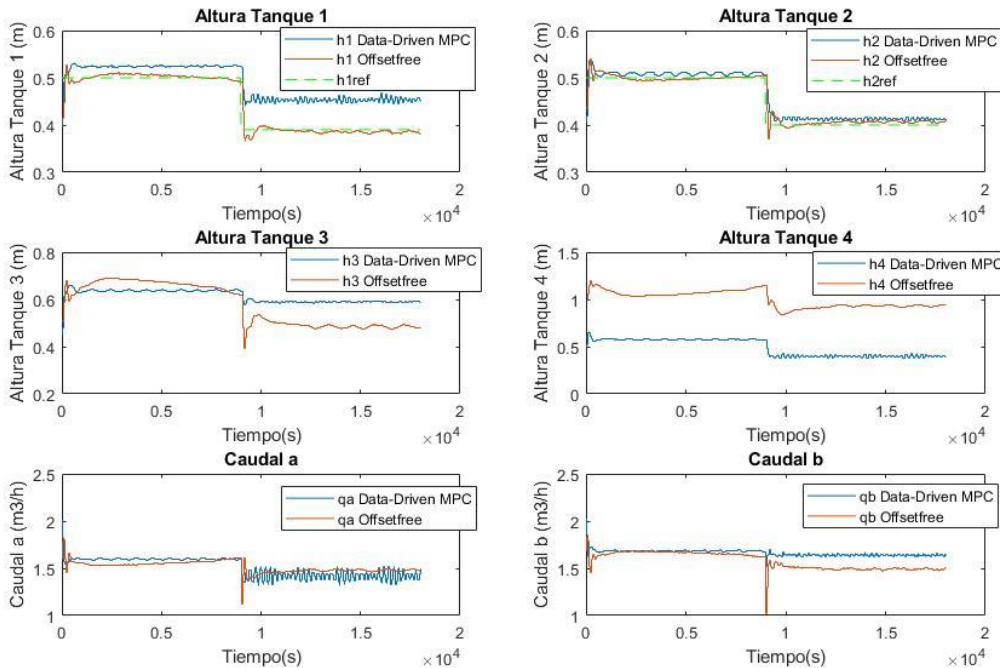


Figura 4-17. Altura de los tanques comparando el algoritmo Data-Driven MPC con el Offsetfree con una base de datos de 100 trayectorias reales.

Cabe destacar que el algoritmo Data-Driven MPC necesita resolver distintos problemas de optimización dependiendo de la factibilidad, hecho que no ocurre con el algoritmo de Offset-free. Esto ayuda a que el comportamiento sea más suave utilizando Offset-free frente al comportamiento del Data-Driven MPC donde se aprecian distintos cambios en las acciones de control debido a la resolución de problemas diferentes de optimización.

4.4 Discrepancias entre modelo Simulado y el sistema Real

Las principales diferencias entre el modelo simulado y el sistema real aparecen en los tanques superiores (3 y 4), como se puede ver en la Figura 4-18. Esto puede deberse a discrepancias entre los coeficientes de descarga de los tanques y a la no linealidad existente en la planta real, como ya se ha explicado. A pesar de ello, también se debe tener en cuenta que los niveles de los tanques superiores no son controlados, además de que las bases de datos simulada y experimental son distintas y no presentan el mismo comportamiento.

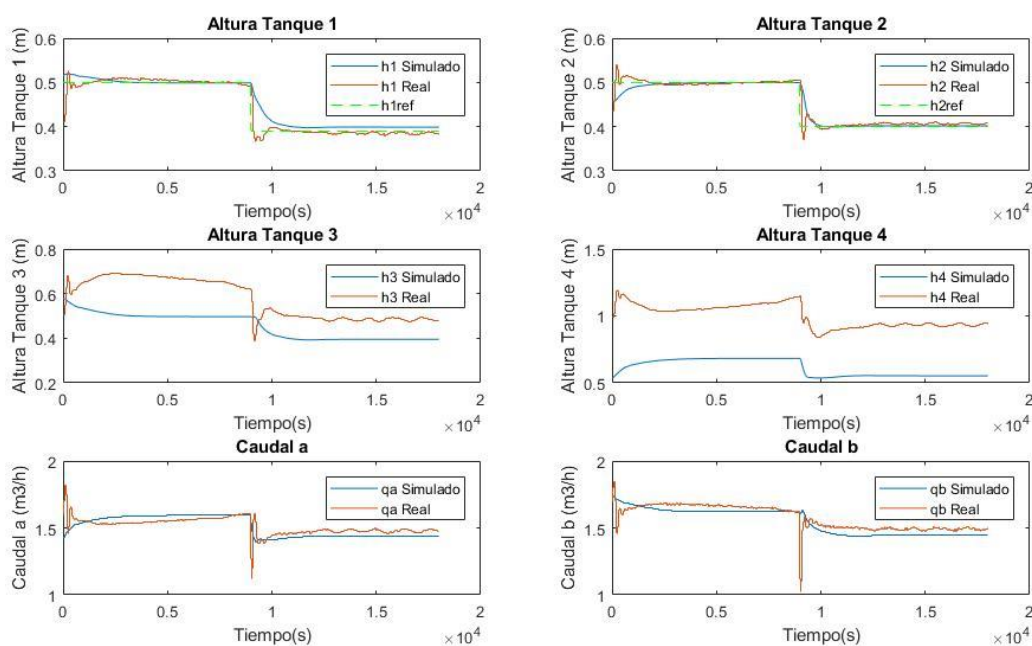


Figura 4-18. Altura de los tanques comparando la respuesta en simulación y real con una base de datos de 100 trayectorias simuladas y reales respectivamente utilizando el algoritmo de Offsetfree.

5 CONCLUSIONES

Se ha podido comprobar a lo largo de este trabajo como el Control Predictivo Basado en Datos es una buena solución para sistemas de los cuales se desconoce el modelo y se tiene a disposición un histórico o base de datos de la operación del sistema. Además, se ha comparado el funcionamiento de los distintos algoritmos, concluyendo que el algoritmo de Offset-free mejora el comportamiento y la factibilidad de la planta de los cuatro tanques.

Sin embargo, para la obtención de los resultados mostrados se ha necesitado una base de datos de unas 100 trayectorias, las cuales son relativamente pocas. Esto se debe, como se ha explicado, a que la constante de tiempo del sistema es muy elevada y un experimento dura de media 2 horas. Añadiendo problemas logísticos dado que la planta no puede estar operativa el 100% del día, se ha tardado más de lo deseado para obtener resultados. Esto deja ver que, un posible problema o defecto en la obtención de las trayectorias haría perder muchísimas horas de trabajo.

También es destacable la complejidad del sistema y de las ecuaciones que se tratan de optimizar en los algoritmos de control. Para el algoritmo Data-Driven MPC se optimiza en cada tiempo de muestreo 6 dimensiones (correspondiente a los 4 tanques de agua y sus referencias) con una longitud de $Q = 1000$ trayectorias candidatas. Habría que tener en cuenta el hecho de resolver en cada tiempo de muestreo de 1 a 4 problemas dependiendo de la factibilidad. Además, en el caso de añadir las holguras (algo necesario debido al tamaño de la base de datos) habría que sumar su contribución a la complejidad del problema. Por otra parte, el algoritmo de Offset-free aumenta las dimensiones del problema a 12 (correspondiente a los 4 tanques de agua, sus estados anteriores, las referencias y las acciones de control pasadas), lo cual complica el estudio.

En cuanto al trabajo previo realizado [1] se han obtenido buenos resultados. Mientras que el trabajo anterior se centraba en regulación y tan solo con el algoritmo Data-Driven MPC, en este trabajo se ha ampliado incluyendo el seguimiento y añadiendo el algoritmo de Offset-free, además de modificar el algoritmo Data-Driven MPC. Los resultados previos en regulación eran bastante buenos, con un offset más que aceptable, sin embargo, la base de datos utilizada contenía 100 trayectorias con una misma referencia. Aunque a priori pareciera igual, en regulación solo se tiene 1 referencia con 100 trayectorias, mientras que en tracking se tiene un rango de referencias (0.2 m a 0.6 m) para 100 trayectorias, diluyendo así la importancia de cada trayectoria en la base de datos.

Finalmente, con este trabajo se deja ver que la aplicación de un CPBD para una planta real con constante de tiempo apreciables no es útil. Además, la presencia de no linealidades que modifiquen el sistema a lo largo del tiempo perjudica la fiabilidad de la base de datos, teniendo que modificarla regularmente. Sin embargo, la aplicación de este tipo de algoritmos en ámbitos donde el flujo de información sea constante, fácil de obtener y poco cambiante en el tiempo es idónea.

Referencias

- [1] E. Masero and D. R. Ramirez, “Actualización y Control Basado en Datos de un Actualización y Control Basado en Datos de un Proceso de Cuatro Tanques Interconectados,” 2018.
- [2] K. H. Johansson, “The quadruple-tank process: A multivariable laboratory process with an adjustable zero,” *IEEE Trans. Control Syst. Technol.*, vol. 8, no. 3, pp. 456–465, 2000.
- [3] D. R. Ramirez and T. Alamo, “Ingeniería de Control Tema 9 . Espacio de Estados : Representación y propiedades importantes,” p. 34, 2005.
- [4] D. R. Ramírez, “Tema 5 . Introducción al control de sistemas multivariables,” pp. 1–19, 2018.
- [5] D. R. Ramirez and T. Alamo, “El Predictor de Smith y estructuras afines,” p. 34.
- [6] J. R. Salvador, D. M. D. La Pena, D. R. Ramirez, and T. Alamo, “Historian Data Based Predictive Control of a Water Distribution Network,” *2018 Eur. Control Conf. ECC 2018*, no. 1, pp. 1716–1721, 2018.
- [7] J. R. Salvador, D. R. Ramirez, T. Alamo, D. M. De La Pena, and G. Garcia-Marin, “Data driven control: An offset free approach,” *2019 18th Eur. Control Conf. ECC 2019*, pp. 23–28, 2019.
- [8] J. R. Salvador, D. R. Ramirez, T. Alamo, and D. M. De La Peña, “Offset free data driven control: Application to a process control trainer,” *IET Control Theory Appl.*, vol. 13, no. 18, pp. 3096–3106, 2019.
- [9] G. García and D. R. Ramirez, “Control Basado en Datos de un proceso de laboratorio,” 2018.

ANEXO: CÓDIGOS USADOS

Generación de las trayectorias

```
clear;clc;close all;

%Inicio de parámetros de la planta
inicio4tanques2;

%q varía de 1.3 a 1.9
qa = 1.3 + 0.6*rand;
qb = 1.3 + 0.6*rand;
sim('sim_BucleAbierto'); % 20 min

for i=1:100
    %Kp varía de 1 a 3.5
    kp1 = 1 + 2.5*rand;
    kp2 = 1 + 2.5*rand;

    %Ti varía de 300 a 500
    Ti1 = 300 + 200*rand; ki1=kp1/Ti1;
    Ti2 = 300 + 200*rand; ki2=kp2/Ti2;

    %Td varía de 10 a 30
    Td1 = 20 + (20*rand-10); kd1=kp1*Td1;
    Td2 = 20 + (20*rand-10); kd2=kp2*Td2;

    load(BucleA)

    %h1_ref aleatorio entre 0.2 a 0.6 m
    h1_ref = 0.2 + 0.4*rand;
    %Obtención de h2_ref factible a partir de h1_ref
    h2_ref = P_Factible(h1_ref);

    qa=qa(end);
    qb=qb(end);

    Ts = plant.Tm;
    sim('sim_Regulacion')
    BucleC=strcat('Simulaciones\BC\BC_',num2str(i),'.mat');
    save(BucleC);
end

sim('sim_apagado') % 2.5 minutos apagado
```

Algoritmo Data-Driven MPC

```

function [beta, tau, flag,Uq]= fcn(z_in,BDsim,Dim_BD)
N = 5; %horizonte prediccion
Q = 1000; %no. trayectorias candidatas

ro = z_in(5:6,1)'; %referencia actual
xo = z_in(1:4,1)'; %estado actual

f = Dim_BD(1); c=Dim_BD(2); %filas columnas BD 500

BD = reshape(BDsim,[f,c]);

%===== Calculo de distancia =====%
vec = repmat( [zeros(Dim_BD(4)-N,1);ones(5,1)] , [Dim_BD(3) 1] );

X = BD(:,3:6); %[h1,h2,h3,h4]
R = BD(:,9:10); %[r_h1, r_h2]

Xo = repmat(xo,f,1); %Repeticion en f filas del estado actual
Xo_norm = Xo/1.2;
X_norm = X/1.2;
dist1 = sum( (Xo_norm-X_norm).^2 ,2); %Norma del vector

Ro = repmat(ro,f,1); %Repeticion en f filas de la referencia actual
Ro_norm = Ro/1.2;
R_norm = R/1.2;
dist2 = sum( (Ro_norm-R_norm).^2 ,2); %Norma del vector

dist = dist1 + dist2; %Distancia total

%Se da peso a las 5 ultimas filas para no elegirlas
peso = max(dist) * vec;
%Se añade el peso para que nunca se seleccionen al ordenar el vector
dist = dist + peso;

%I índices de posición
[~,I] = sort(dist); %Se ordena de menor a mayor y se guardan los índices

%numero de la fila en la BD de las Q tray candidatas
index = I(1:Q,1);

%===== Calculo del coste =====%
%El estado de las Q trayectorias más cercanas al estado actual y
%las N predicciones futuras
Zq0 = BD(index,(3:4));
Zq1 = BD(index+1,(3:4));
Zq2 = BD(index+2,(3:4));
Zq3 = BD(index+3,(3:4));
Zq4 = BD(index+4,(3:4));

Zq = [Zq0, Zq1, Zq2, Zq3, Zq4]; %Hay N*(numero de estados) columnas

Ro = repmat(ro,Q,N); %matriz de misma dimension que Zq
J_matriz = (Zq-Ro).^2; %matriz con costes
J = sum(J_matriz,2); %columna con costes de etapas
%===== Restricciones =====%
%sum( beta*xq(0) )= x(t)

```



```

xqo = BD(index,3:6); %h1,h2,h3,h4

Aeq1 = xqo';
beq1 = xo';

%sum( beta ) = 1
Aeq2 = ones(1,Q);
beq2 = 1;

%sum( beta*rq(0) )= r(t)
rqi = BD(index,9:10); %r_h1 y r_h2

Aeq3 = rqi';
beq3 = ro';

%Restricciones en Estado actual y sum(beta)=1
Aeq = [Aeq1; Aeq2; Aeq3];
beq = [beq1; beq2; beq3];

%===== Cálculo de la Acción de Control =====%
%Inicialización de valores
beta = zeros(Q,1);
tau = zeros(Q,1);

exitflag = 1;
flag2 = 1;
flag3 = 1;
flag4 = 1;

coder.extrinsic('quadprog');
coder.extrinsic('linprog');

%Hiperparámetros
delta = 100;
alfa = 1000;

%:::::::::: Problema con todas las restricciones ::::::::::::::%
[Opt,~,exitflag] = linprog(J,[],[],Aeq,beq,zeros(Q,1),[]);

if exitflag ==1
    beta = Opt;
    flag1 = 1; %Sí factible
else
    flag1 = 0; %No factible
end

if flag1 == 0 %Si no es factible
    %:::::::::: Restriccion 3 Soft (Relajación de restricciones) ::::::::::::::%
    beta = zeros(Q,1);

    Aeq = [Aeq1; Aeq2];
    beq = [beq1; beq2];

    H = delta*(rqi*rqi');
    f = J'/2 - delta*(ro*rqi');

    [Opt,~,exitflag] = quadprog(H,f,[],[],Aeq,beq,zeros(Q,1),[]);

```

```

    if exitflag == 1
        beta = Opt;
        flag2 = 1; %Si factible
    else
        flag2 = 0; %No factible
    end
end

if flag2 == 0 %Si no es factible
    %:::::::::::::::::::: Implementación de holguras ::::::::::::::%
    beta = zeros(Q,1);

    Aeq = [Aeq1; Aeq2; Aeq3];
    beq = [beq1; beq2; beq3];

    nf = size(Aeq,1);
    Aeq = [Aeq zeros(nf,Q)];

    H = [J*J', zeros(Q) ; zeros(Q), alfa*eye(Q)];
    f = [zeros(1,Q) , alfa*ones(1,Q)];

    A = [-1*ones(1,Q), -1*ones(1,Q); 1*ones(1,Q), -1*ones(1,Q)];
    b = [0;1];

    lb = [-inf*ones(Q,1);zeros(Q,1)];

    [Opt,~,exitflag] = quadprog(H,f,A,b,Aeq,beq,lb,[]);

    if exitflag == 1
        beta = Opt(1:Q);
        tau = Opt((Q+1):2*Q);
        flag3 = 1; %Si factible
    else
        flag3 = 0; %No factible
    end
end

if flag3 == 0 %Si no es factible
    %::::::::::::: Restriccion 3 Soft e implementación de holguras ::::::::::::::%
    beta = zeros(Q,1);

    Aeq = [Aeq1; Aeq2; Aeq3];
    beq = [beq1; beq2; beq3];

    nf = size(Aeq,1);
    Aeq = [Aeq zeros(nf,Q)];

    H = [J*J' + delta*(rqrqrq'), zeros(Q) ; zeros(Q), alfa*eye(Q)];
    f = [J'/2 - delta*(rqrqrq') , alfa*ones(1,Q)];

    A = [-1*ones(1,Q), -1*ones(1,Q); 1*ones(1,Q), -1*ones(1,Q)];
    b = [0;1];

    lb = [-inf*ones(Q,1);zeros(Q,1)];

    [Opt,~,exitflag] = quadprog(H,f,A,b,Aeq,beq,lb,[]);

    if exitflag == 1
        beta = Opt(1:Q);

```

```
tau    = Opt((Q+1):2*Q);
flag3  = 1; %Sí factible
else
flag4  = 0; %No factible
%Se toma la acción de control más cercana a la actual
beta   = [1; zeros(Q-1,1)];
end
end

flag = [flag1;flag2;flag3;flag4];

Uq = beta'*BD(index,7:8); %beta*u_opt
Endo
```

Algoritmo de Offset-free

```

function [beta,Uq]= fcn(z_in,BDsim,Dim_BD, z_in_a)
N = 5; %horizonte prediccion
Q = 1000; %no. trayectorias candidatas

ro = z_in(5:6,1)'; %referencia actual
xo = z_in(1:4,1)'; %estado actual
xo_a = z_in_a(1:4,1)'; %estado anterior
uo_a = z_in_a(7:8,1)'; %acciones de control anteriores (qa y qb)

f = Dim_BD(1); c=Dim_BD(2); %filas columnas BD 500

BD = reshape(BDsim,[f,c]);
%===== Calculo de distancia =====
vec = repmat( [1;zeros(Dim_BD(4)-(N+1),1);ones(5,1)] , [Dim_BD(3) 1] );

X = BD(:,3:6); %[h1,h2,h3,h4]
X_a = [zeros(1,4); BD(1:end-1,3:6)]; %[h1,h2,h3,h4] retrasado en 1 posicion
%Se añade la fila de ceros porque se desconoce el valor inicial
U_a = [zeros(1,2); BD(1:end-1,7:8)]; %[qa,qb] retrasado en 1 posicion
%Se añade la fila de ceros porque se desconoce el valor inicial
R = BD(:,9:10); %[r_h1, r_h2]

Xo = repmat(xo,f,1); %Repeticion en f filas del estado actual
Xo_norm = Xo/1.2;
X_norm = X/1.2;
dist1 = (Xo_norm-X_norm).^2*[2;2;1;1]; %Norma del vector

Xo_a = repmat(xo_a,f,1); %Repeticion en f filas del estado anterior
Xo_a_norm = Xo_a/1.2;
X_a_norm = X_a/1.2;
dist2 = sum( (Xo_a_norm-X_a_norm).^2 ,2) ; %Norma del vector

Uo_a = repmat(uo_a,f,1); %Repeticion en f filas de la actuacion anterior
Uo_a_norm = Uo_a/3;
U_a_norm = U_a/3;
dist3 = sum( (Uo_a_norm-U_a_norm).^2 ,2) ; %Norma del vector

Ro = repmat(ro,f,1);%Repeticion en f filas de la referencia actual
Ro_norm = Ro/1.2;
R_norm = R/1.2;
dist4 = sum( (Ro_norm-R_norm).^2 ,2) ; %Norma del vector

dist = dist1 + dist2 + dist3 + dist4; %Distancia total

%Se da peso a la primera y las 5 últimas filas para no elegirilas
peso = max(dist) * vec;
%Se añade el peso para que nunca se seleccionen al ordenar el vector
dist = dist + peso;

%I índices de posición
[~,I] = sort(dist); %Se ordena de menor a mayor y se guardan los índices

%numero de la fila en la BD de las Q tray candidatas
index = I(1:Q,1);

%===== Restricciones =====%
%sum( beta*xq(0) )= x(t)

```

```

xqo = BD(index,3:6); %h1,h2,h3,h4

Aeq1 = xqo';
beq1 = xo';

%sum( beta ) = 1
Aeq2 = ones(1,Q);
beq2 = 1;

%sum( beta*xq(-1) )= x(t-1)
xqo_a = BD(index-1,3:6); %h1,h2,h3,h4 retrasados en 1 posición

Aeq3 = xqo_a';
beq3 = xo_a';

%sum( beta*uq(-1) )= u(t-1)
uqo_a = BD(index-1,7:8); %qa y qb retrasados en 1 posición

Aeq4 = uqo_a';
beq4 = uo_a';

%sum( beta*yq(0) )= y(t)
yqo = BD(index,3:4); %h1 y h2

Aeq5 = yqo';
beq5 = xo(1:2)';

%sum( beta*rq(0) )= r(t)
rqo = BD(index,9:10); %r_h1 y r_h2

Aeq6 = rqo';
beq6 = ro';

%Todas las restricciones
Aeq = [Aeq1; Aeq2; Aeq3; Aeq4; Aeq5; Aeq6];
beq = [beq1; beq2; beq3; beq4; beq5; beq6];

%===== Cálculo de la Acción de Control =====%
%Inicialización de valores
beta=zeros(Q,1);

coder.extrinsic('quadprog');
Opt=quadprog(eye(Q), [], [], [], Aeq, beq, [], []);
beta = Opt;

Uq=beta'*BD(index,7:8); %beta*u_opt
end

```