

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



**DISEÑO Y PROPUESTA DE IMPLEMENTACIÓN DEL CONTROL DE NIVEL EN
UN PROCESO DE DOS TANQUES ACOPLADOS MEDIANTE CONTROL
PREDICTIVO**

Tesis para obtener el título profesional de Ingeniero Electrónico

AUTOR:

Diego Alejandro Oré Sánchez

ASESOR:

Juan Javier Sotomayor Moriano

Lima, octubre, 2022

Informe de Similitud

Yo, Juan Javier Sotomayor Moriano, docente de la Facultad de Ciencias e Ingeniería de la Pontificia

Universidad Católica del Perú, asesor de la tesis titulada

DISEÑO Y PROPUESTA DE IMPLEMENTACIÓN DEL CONTROL DE NIVEL EN UN PROCESO DE DOS TANQUES ACOPLADOS MEDIANTE CONTROL PREDICTIVO, del autor

Diego Alejandro Oré Sánchez,

dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 18%. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 03/02/2023.
- He revisado con detalle dicho reporte y la Tesis o Trabajo de Suficiencia Profesional, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lugar y fecha: Lima, 07/02/2023

Apellidos y nombres del asesor / de la asesora: <u>Paterno Materno, Nombre1 Nombre 2</u>	Sotomayor Moriano Juan Javier
DNI: 25558480	
ORCID: 0000-0003-0782-0530	

RESUMEN

La presencia de tanques hidráulicos en actividades industriales es recurrente debido a la gran cantidad de industrias que utilizan fluidos en sus procesos de producción. Para el correcto funcionamiento de estos procesos, es necesario contar con un adecuado control del nivel para que se logre aprovechar al máximo los recursos hidráulicos. En este trabajo se desarrolla un sistema de control de nivel multivariable aplicado a una planta de dos tanques acoplados, el cual es un proceso frecuente en la industria, asimismo, para este fin, se utilizó una estrategia del control predictivo basado en modelo, denominada control predictivo por matriz dinámica.

Esta investigación comienza revisando el estado del arte de las técnicas de control que se han diseñado para estos procesos basadas en teorías de control clásico, control moderno y control avanzado, entre estos últimos los controladores predictivos. Luego se presenta la base teórica del control predictivo por matriz dinámica. De igual manera, se describe el modelo matemático lineal de la planta piloto de dos tanques acoplados del Laboratorio de Control Avanzado de la PUCP el cual servirá como modelo de referencia para el diseño.

Posteriormente, se procede a diseñar el controlador requerido para esta aplicación. En el proceso de sintonización de los parámetros del mismo se aplicaron dos métodos recopilados de la literatura: el clásico de Shridar y Cooper y otro basado en el algoritmo de optimización PSO. Las respectivas simulaciones están orientadas a comparar el desempeño del controlador sintonizado con cada uno de estos métodos a fin de identificar las ventajas que ofrece cada uno.

Finalmente, se desarrolla una propuesta de implementación del controlador diseñado con el propósito de mostrar una forma posible de implementar el diseño obtenido en un entorno industrial.

Índice General

Introducción	1
Capítulo 1	
1. Estudio del Control de Nivel en procesos TITO.....	3
1.1. Problemática	3
1.2. Descripción de procesos TITO	3
1.2.1. Procesos Multivariantes – Procesos TITO.....	3
1.2.2. Proceso de tanques acoplados – Estructura TITO.....	4
1.3. Control de nivel en el proceso de tanques acoplados TITO.....	7
1.4. Justificación	9
1.5. Objetivos.....	10
1.5.1. Objetivo general.....	10
1.5.2. Objetivos específicos	10
Capítulo 2	
2. Control predictivo de un proceso de tanques acoplados TITO	11
2.1. Introducción	11
2.2. Control predictivo por matriz dinámica (DMC)	12
2.2.1. Introducción.....	12
2.2.2. Modelo de predicción.....	12
2.2.3. Función costo y algoritmo de control	15
2.2.4. DMC en sistemas multivariantes	16
2.3. Modelamiento de la planta piloto.....	17
2.4. Requerimientos para aplicar el DMC al proceso de tanques acoplados TITO	20
Capítulo 3	
3. Diseño del Controlador DMC para la planta TITO. Aplicación de Métodos de Sintonización	22
3.1. Introducción	22
3.2. Matriz Dinámica de la planta piloto.....	23
3.2.1. Cálculo de la Matriz Transferencia.....	23
3.2.2. Selección del Horizonte del Modelo y Tiempo de muestreo	24
3.2.3. Cálculo de los coeficientes de la Matriz Dinámica.....	25
3.3. Diseño de la lógica de control.....	26
3.3.1. Consideraciones respecto al algoritmo de control	26
3.3.2. Diagrama de flujo del controlador DMC	27
3.4. Sintonización del controlador DMC MIMO.....	30
3.4.1. Necesidad de Métodos de Sintonización.....	30
3.4.2. Método clásico de Shridar y Cooper.....	30

3.4.3. Método de sintonización mediante PSO	32
---	----

Capítulo 4

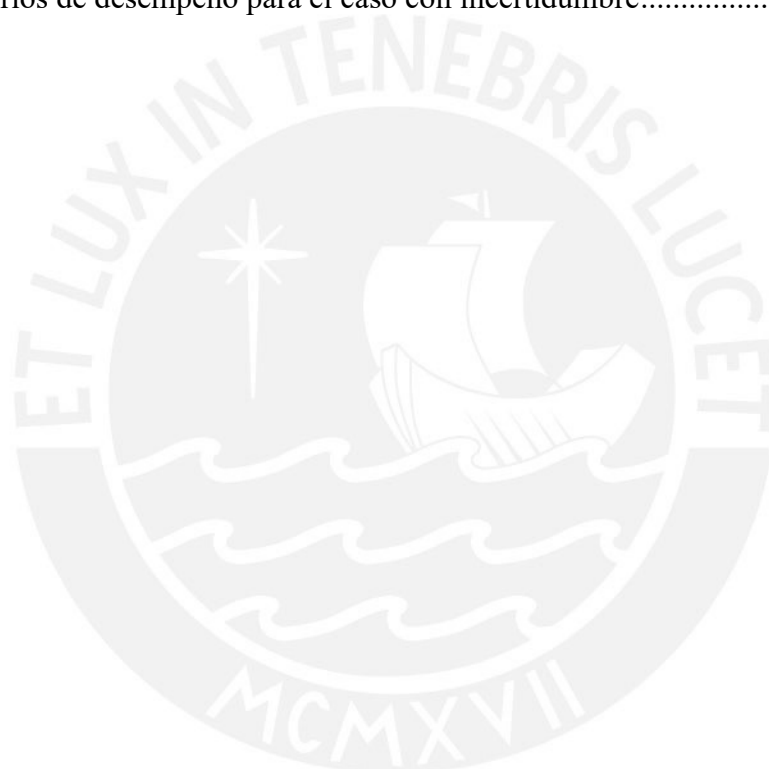
4. Pruebas de Simulación, Análisis de resultados y Propuesta de implementación del controlador DMC	37
4.1. Introducción	37
4.2. Pruebas de Simulación	37
4.2.1 Señales de Referencia	37
4.2.2. Pruebas con Modelo nominal.....	39
4.2.3. Pruebas con perturbaciones.....	42
4.2.4. Pruebas con Incertidumbre en el Modelo.....	44
4.3. Análisis de resultados	46
4.4. Propuesta de implementación	47
4.4.1 Implementación del controlador DMC para la Planta TITO.....	47
4.4.2 Algoritmo de control DMC.....	48
4.4.3 Propuesta de funcionamiento de la planta TITO no lineal.....	49
4.4.4 Interfaz gráfica HMI	50
CONCLUSIONES	52
RECOMENDACIONES	54
BIBLIOGRAFÍA	55
ANEXOS	57

Índice de Figuras

Figura 1.1. Diagrama esquemático del sistema de tanque cuádruple (QTS) [6].	4
Figura 1.2. Diagrama esquemático del sistema de tanques acoplados (CTS) [6].	5
Figura 1.3. Diagrama de bloques del CTS (adaptado) [3].	6
Figura 1.4. Planta piloto de cuatro tanques acoplados.	6
Figura 1.5. Esquema de control descentralizado con desacoplamiento.	7
Figura 2.1. Esquema de la planta piloto.	18
Figura 2.2. Diagrama de bloques del sistema de control.	21
Figura 3.1. Diagrama de flujo del controlador.	29
Figura 4.1. Señales de set point para los tanques.	38
Figura 4.2. Respuesta del sistema en el caso nominal y sintonización clásica.	39
Figura 4.3. Variables de control en el caso nominal y sintonización clásica.	39
Figura 4.4. Respuesta del sistema en el caso nominal y sintonización PSO.	40
Figura 4.5. Variables de control en el caso nominal y sintonización PSO.	40
Figura 4.6. Simulación de perturbaciones en el caso nominal y sintonización clásica.	42
Figura 4.7. Simulación de perturbaciones en el caso nominal y sintonización PSO.	43
Figura 4.8. Respuesta del sistema en el caso con incertidumbre y sintonización clásica.	45
Figura 4.9. Respuesta del sistema en el caso con incertidumbre y sintonización PSO.	45
Figura 4.10. Diagrama de flujo del algoritmo de control DMC.	49
Figura 4.11. Diagrama en Simulink de la planta TITO no lineal.	50
Figura 4.12. Interfaz HMI en Labview para monitoreo.	51

Índice de Tablas

Tabla 3.1. Parámetros de la planta piloto.....	23
Tabla 3.2. Parámetros del controlador DMC por sintonización clásica.....	32
Tabla 3.3. Parámetros auxiliares de la función EISTWE	34
Tabla 3.4. Parámetros del controlador DMC por sintonización PSO	36
Tabla 4.1. Criterios de desempeño en el caso nominal y sintonización clásica.....	40
Tabla 4.2. Criterios de desempeño en el caso nominal y sintonización PSO	41
Tabla 4.3. Índices ISE para el caso nominal con perturbación.....	43
Tabla 4.4. Parámetros del controlador para el caso con incertidumbre	44
Tabla 4.5. Criterios de desempeño para el caso con incertidumbre.....	46



Introducción

En la actualidad, existe una amplia variedad de plantas industriales en las cuales se utilizan sistemas de tanques hidráulicos interconectados que permiten el almacenamiento, mezcla y tratamiento de distintos tipos de líquidos. Ejemplos de estas industrias son la farmacéutica, petroquímica, minera, alimentaria, etc. En estos sistemas surge la necesidad de controlar el nivel líquido en cada tanque, así como el flujo que circula entre estos, por lo que estas tareas deben realizarse de manera óptima para aprovechar la mayor cantidad de recursos [13]. La planta de tanques acoplados propuesta por Johansson en 1998 permite estudiar en laboratorio este tipo de procesos hidráulicos con simplicidad, así como las características y los comportamientos de los procesos reales.

De igual manera, es importante mencionar que realizar el control de estos sistemas de tanques no es sencillo de lograr debido a las interacciones entre sus variables y las dinámicas no lineales propias de estos [3], [4]. Por ello, resulta relevante desarrollar controladores avanzados con un alto rendimiento y efectividad para estos sistemas.

Una técnica de control avanzado que ha adquirido gran aceptación tanto en el mundo académico como en la industria es la del control predictivo basado en modelo (MPC). El MPC presenta grandes ventajas para el control de sistemas multivariables y no lineales ya que realiza predicciones sobre las posibles futuras salidas y optimiza las acciones de control que ingresan a los procesos [10]. Esta técnica de control permite superar las desventajas que tienen los controladores descentralizados clásicos cuando se aplican a sistemas multivariables.

En el presente trabajo, se propone un sistema de control predictivo basado en control por matriz dinámica (DMC) para una planta de dos tanques acoplados instalada en el Laboratorio de Control Avanzado de la PUCP.

En el primer capítulo de este trabajo, se realiza una revisión del estado del arte de los controladores que han sido diseñados para esta planta utilizando diferentes técnicas y enfoques del control automático.

El marco teórico necesario para realizar el diseño del controlador DMC se expone en el segundo capítulo. De igual manera, se describe el modelo matemático de la planta y el procedimiento para linealizarlo alrededor de un punto de operación.

El tercer capítulo comprende tanto el diseño del controlador propuesto, así como la sintonización de sus parámetros a partir de aplicar dos métodos de sintonización existentes: uno clásico desarrollado por Shridar y Cooper [20], y otro basado en el algoritmo PSO [22].

El análisis de los resultados de las simulaciones realizadas para comparar el desempeño del controlador sintonizado con cada uno de los métodos mencionados se lleva a cabo en el cuarto capítulo. Finalmente, este capítulo culmina con la propuesta de implementación desarrollada para el sistema de control diseñado en un entorno industrial.

1. Estudio del Control de Nivel en procesos TITO

1.1. Problemática

En la actualidad, existen diferentes aplicaciones en la industria en las cuales el control de nivel es parte importante para el buen funcionamiento de sus procesos. Tal es el caso donde se utilizan tanques dentro de procesos de producción para el almacenamiento, tratamiento o mezcla de distintos líquidos. Ejemplo de ello son las industrias farmacéuticas, plantas petroquímicas, industrias de producción de bebidas y centrales nucleares [1]. Aquellos procesos de producción requieren con frecuencia el bombeo de líquido para llenar un tanque, el mantenimiento del nivel del mismo en determinada magnitud o la regulación del flujo en el traspase hacia otros tanques interconectados; por lo que, el control automático del nivel de líquidos en los tanques resulta ser una actividad primordial y permanente en la industria [1], [2].

Los sistemas de tanques interconectados pertenecen a la categoría de sistemas de múltiples entradas y salidas, también conocidos como MIMO por sus siglas en inglés. Esta clase de sistemas abundan en la industria y, en ocasiones, son a la vez de naturaleza no lineal [3]. Encontrar el control adecuado para ellos es un desafío ya que existe una alta interacción o acoplamiento entre sus variables de entrada y salida, presentan parámetros variantes en el tiempo, y poseen restricciones sobre las variables que se pueden manipular [3], [4].

1.2. Descripción de procesos TITO

1.2.1. Procesos Multivariables – Procesos TITO

Un proceso multivariable es aquel que está conformado por más de una variable de entrada y de salida. Se caracterizan porque la manipulación de una sola variable de entrada puede afectar las respuestas de distintas salidas a la vez. En particular, los procesos multivariables con solo

dos variables de entrada y dos de salida se conocen como procesos TITO, por sus siglas en inglés.

Seguidamente se describen las características y el funcionamiento del sistema hidráulico que es objeto de estudio en este trabajo. Luego, se presentarán algunas de las estrategias de control de nivel para este tipo de sistemas multivariable TITO.

1.2.2. Proceso de tanques acoplados – Estructura TITO

En el año 1998, Johansson presentó por primera vez un novedoso proceso de laboratorio denominado como el sistema de tanque cuádruple, abreviado como QTS debido a sus siglas en inglés. El proceso consiste de cuatro tanques de agua interconectados y dos bombas centrífugas, donde las entradas del sistema son los voltajes para ambas bombas mientras que las salidas son los niveles de líquido en los tanques, ver Fig. 1.1. En [5] se menciona que aunque el proceso es de sencilla implementación, aun así ilustra una gran variedad de fenómenos presentes en los sistemas multivariables. Esta característica ha hecho que el QTS se convierta en un punto de referencia, ampliamente aceptado y se utilice para probar técnicas de control multivariable.

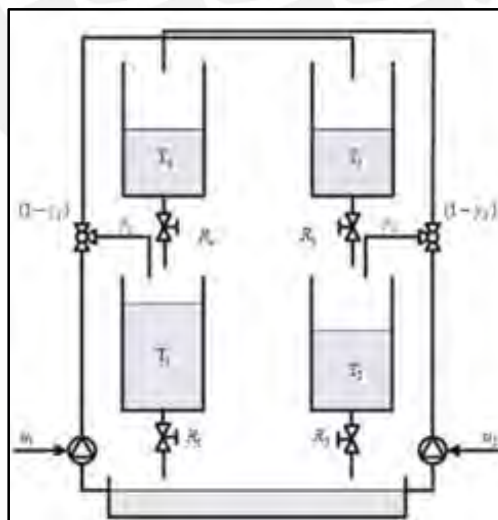


Figura 1.1. Diagrama esquemático del sistema de tanque cuádruple (QTS) [6].

Como consecuencia, varios investigadores han trabajado no solo con el proceso original sino también con distintas variaciones del mismo. Existen hasta 6 modificaciones útiles del QTS

para realizar pruebas de laboratorio [6]. En la presente tesis se trabajará con una de estas modificaciones, denominada sistema de tanques acoplados (CTS, en inglés), la cual consiste de solo dos tanques interconectados entre sí a través de una válvula, dos bombas para el llenado de cada tanque y un reservorio desde el cual se bombea el líquido. En la Fig. 1.2 se ilustra el diagrama esquemático de este sistema.

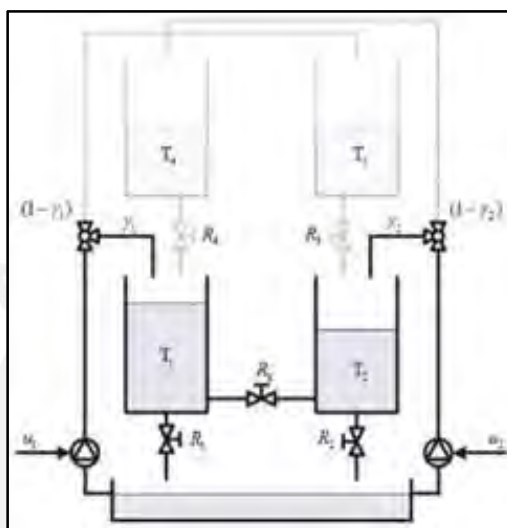


Figura 1.2. Diagrama esquemático del sistema de tanques acoplados (CTS) [6].

En este sistema CTS existe interrelación entre las variables de entrada y de salida de cada tanque. Ello implica que el nivel de líquido en cierto tanque no solo depende del flujo del líquido que ingresa al mismo desde el reservorio sino también del efecto que produce el otro tanque sobre este a través de la válvula de interconexión. Esta interdependencia se ilustra con el diagrama de bloques del CTS presentado en la Fig. 1.3.

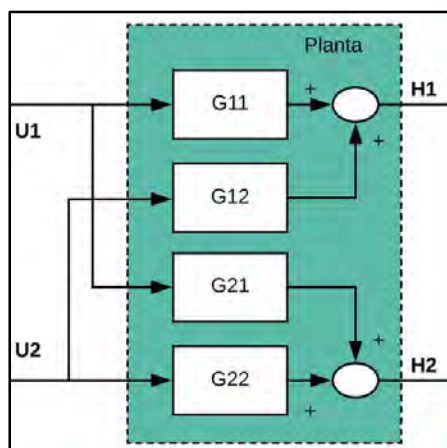


Figura 1.3. Diagrama de bloques del CTS (adaptado) [3].

La planta piloto sobre la cual se realizará el diseño del sistema de control de nivel que esta tesis propone se encuentra en el Laboratorio de Control Avanzado de la Pontificia Universidad Católica del Perú. Como se muestra en la Fig. 1.4, esta planta es un sistema de cuatro tanques acoplados que permite seis posibles configuraciones, dentro de las cuales se encuentra el proceso de tanques acoplados objeto de estudio. Los principales componentes de esta planta son los tanques superiores, el reservorio de alimentación, las bombas centrífugas, las válvulas de control, los sensores de caudal y de presión, los variadores de velocidad y las tuberías.



Figura 1.4. Planta piloto de cuatro tanques acoplados.

1.3. Control de nivel en el proceso de tanques acoplados TITO

En la sección anterior se mencionó que este proceso de tanques acoplados se ha utilizado recurrentemente para probar diferentes técnicas de control de procesos multivariables. Entonces, resulta necesario investigar acerca de estas técnicas para conocer la manera en que otros autores han abordado el control de esta planta y las conclusiones a las cuales llegaron.

Desde el enfoque del control clásico, se han propuesto distintos esquemas de control basados en controladores de tipo proporcional integrativo y derivativo (PID) y sus variantes. En estos esquemas se diseña un controlador por cada lazo de control de la planta TITO, los cuales son dos, y se busca que cada uno de ellos regule el comportamiento de solo una variable de salida. De esta manera, el control se vuelve descentralizado, es decir, la planta TITO pasa a ser tratada como dos procesos SISO independientes el uno del otro. Para lograr lo último es necesario que se incluya una etapa de desacoplamiento entre los controladores y la planta, ver Fig. 1.5. Existen múltiples técnicas que trabajan con esta metodología entre las cuales se puede mencionar al control PID con sintonización secuencial [7], control PI de dos grados de libertad [3] o control por desacoplamiento invertido [8].

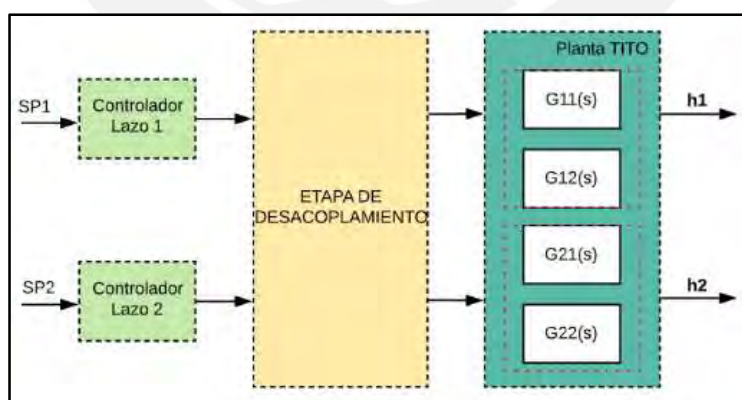


Figura 1.5. Esquema de control descentralizado con desacoplamiento.

Por otro lado, desde el enfoque de la teoría de control avanzado, también se han propuesto soluciones para el control de este proceso. Una de estas se basa en el control adaptativo por modelo de referencia (MRAC), en el cual se diseña un controlador PI en conjunto con un

bloque de ajuste que permite variar los parámetros del anterior gracias a la inclusión del modelo de la planta en el controlador [9]. Otra de ellas utiliza el control de modelo interno (IMC) para diseñar un controlador PI utilizando la inversa de la función de transferencia del modelo matemático de la planta [2]. Estos dos casos también aplican un control descentralizado como en los anteriores, pero se distinguen principalmente porque sus controladores ahora incluyen cierto modelo de la planta.

Otra técnica de control avanzado que comparte la última característica mencionada es el control predictivo basado en modelo de referencia (MPC). El MPC está orientado a realizar un control centralizado de la planta CTS, ello quiere decir que se cuenta con un solo controlador para todo el proceso y ya no dos como en los ejemplos anteriores. Esta técnica no define una estrategia en particular, sino que agrupa a diferentes algoritmos de control que comparten tres elementos básicos: un modelo de predicción de la planta, un bloque de optimización para las acciones de control y un horizonte de predicción. Estos controladores utilizan el modelo para predecir las respuestas de la planta ante determinados esfuerzos de control y así optimizarlos para lograr el mejor desempeño en la planta real [10].

Dentro de los trabajos que han aplicado el control predictivo en la planta de tanques acoplados se pueden mencionar tanto el de Usman y Bilal [11], el cual emplea la estrategia de control predictivo generalizado (GPC) y polinomios de Laguerre, como el de Awais et al. [12], en donde se emplea el MPC en tiempo discreto basado en la formulación de espacio de estados. En estos dos ejemplos se requiere un modelo lineal del proceso. Sin embargo, el MPC también puede basarse en un modelo no lineal tal como lo muestra la investigación de Owa et al. que emplea redes neuronales artificiales para el modelamiento del CTS y algoritmos genéticos para el bloque de optimización [13].

1.4. Justificación

Los controladores PID son ampliamente utilizados para el control de diferentes plantas industriales gracias a su implementación sencilla y su notable efectividad en una gran variedad de procesos [7]. Alrededor del 90% de lazos de control en la industria incluye controladores PID [7]. Este controlador ha demostrado ser altamente efectivo al trabajar con sistemas SISO ya que pertenece al enfoque del control clásico, que está principalmente orientado a este tipo de sistemas.

Sin embargo, los procesos industriales con tanques acoplados son identificados como MIMO, por lo que se han tenido que desarrollar estrategias de control que permitan aplicar los controladores PID a estos procesos. Como ya se mencionó, la idea fundamental en estas estrategias es lograr un control descentralizado de la planta a partir de descomponer el sistema MIMO en varios subcomponentes SISO. De esta forma, se reduce la interacción que existe entre variables de distintos lazos y así permitir la aplicación del PID en cada uno de ellos.

La principal desventaja con el control descentralizado es que las etapas de desacoplamiento que utiliza no logran suprimir por completo la interacción entre las variables de los tanques [2], lo cual le resta efectividad al sistema de control de nivel y esto implica un desaprovechamiento económico para las industrias [13]. En consecuencia, debido a la ventaja que presenta el MPC al no tener que desacoplar a los procesos MIMO, este ha encontrado muchas aplicaciones exitosas en la industria química, alimenticia y petroquímica [11]. Entonces, el MPC resulta ser una alternativa atractiva con la cual se puede realizar un control de nivel centralizado para procesos de tanques interconectados e inclusive permite considerar explícitamente restricciones en las variables del proceso. Aparte de ello, los controladores de tipo MPC pueden trabajar con dinámicas tanto lineales como no lineales para modelar el proceso [13].

Por lo tanto, en la presente tesis se plantea el diseño de un controlador de nivel utilizando estrategias del MPC para una planta de dos tanques hidráulicos interconectados de 2 entradas

y 2 salidas (TITO). El desarrollo de este controlador brindará una nueva solución para el control de este tipo de procesos industriales. Dado que la planta piloto con la que se trabajará refleja de manera satisfactoria el comportamiento no lineal, las restricciones y las perturbaciones de una planta real, el diseño a desarrollar será válido para su uso en la industria.

1.5. Objetivos

1.5.1. Objetivo general

Diseñar y proponer la implementación del sistema de control de nivel de un proceso hidráulico de 2 tanques acoplados usando control predictivo basado en modelo (MPC).

1.5.2. Objetivos específicos

- Elección de la estrategia de control predictivo para el diseño del controlador.
- Modelamiento de la planta de dos tanques interconectados acorde a la estrategia de control predictivo elegida.
- Diseño del controlador predictivo basado en modelo para la planta piloto y su simulación empleando dos métodos de sintonización.
- Propuesta de implementación del controlador MPC.

2. Control predictivo de un proceso de tanques acoplados TITO

2.1. Introducción

El control predictivo basado en modelo (MPC) es una técnica de control avanzado que se caracteriza por su capacidad de manipular procesos multivariables de manera sencilla, incluso ante la presencia de restricciones o comportamientos no lineales [14]. Debido a ello, su uso en la industria ha adquirido popularidad en las últimas décadas, encontrando aplicaciones en procesos industriales, por ejemplo, en el control de motores [15], en procesos de centrales térmicas [16], control de reactores [10], etc.

El término MPC no se refiere a una técnica de control en particular, sino que en realidad agrupa a diferentes estrategias que hacen uso de un modelo del proceso y de una función costo para calcular las leyes de control requeridas. Toda esta familia de estrategias comparte las siguientes características: emplean un modelo de la planta para pronosticar las salidas de la misma, obtienen las acciones de control a partir de minimizar una función costo de los errores futuros y siguen un principio de retroceso de horizonte.

Entre estas estrategias de control predictivo se encuentran el control predictivo generalizado (GPC) y el control por matriz dinámica (DMC). La principal diferencia entre estas radica en el tipo de modelo del proceso que emplean pues la primera utiliza un modelo CARIMA, mientras que la segunda, un modelo por respuesta al escalón. A comparación del GPC, un controlador DMC tiene una implementación más sencilla [17] y ello se debe a que no necesita resolver de manera recursiva una ecuación diofántica para calcular las predicciones, lo cual genera una mayor carga computacional para el controlador [18]. Debido a ello, se eligió la estrategia de DMC para realizar la propuesta de solución de este trabajo.

En el presente capítulo se desarrolla el marco teórico necesario para la aplicación de esta estrategia, así como, el modelamiento de la planta piloto de dos tanques acoplados.

2.2. Control predictivo por matriz dinámica (DMC)

2.2.1. Introducción

En esta sección se realiza un resumen de la fundamentación teórica del control DMC para procesos MIMO. Con este objetivo, se toma como base los estudios realizados en [14], [19]. En primer lugar, se describe el planteamiento del control DMC para procesos SISO ya que el entendimiento de sus fundamentos es más sencillo en estos procesos. Posteriormente, se extiende el método para plantas multivariadas.

2.2.2. Modelo de predicción

En esta estrategia de control se utiliza un modelo de predicción basado en la respuesta al escalón de la planta y las perturbaciones son consideradas constantes en todo el horizonte de predicción.

Como se emplea un modelo de respuesta al escalón, la salida de la planta queda descrita por la siguiente expresión:

$$y(t) = \sum_{i=1}^{\infty} g_i \Delta u(t - i) \quad (2.1)$$

Donde,

g_i : valores de salida muestreados;

$u(t)$: escalón unitario;

$$\Delta u(t) = u(t) - u(t - 1)$$

Luego, se puede pronosticar el valor de la salida para un instante futuro k a partir del instante actual de la siguiente forma:

$$\hat{y}(t+k|t) = \sum_{i=1}^{\infty} g_i \Delta u(t+k-i) + \hat{n}(t+k|t) \quad (2.2)$$

$$\hat{y}(t+k|t) = \sum_{i=1}^k g_i \Delta u(t+k-i) + \sum_{i=k+1}^{\infty} g_i \Delta u(t+k-i) + \hat{n}(t+k|t) \quad (2.3)$$

donde las perturbaciones en la predicción están representadas por el elemento $\hat{n}(t+k|t)$ y se consideran constantes a lo largo del horizonte de predicción, es decir,

$$\hat{n}(t+k|t) = \hat{n}(t|t) = y_m(t) - \hat{y}(t|t) \quad (2.4)$$

siendo $y_m(t)$ la salida medida de la planta real. Por consiguiente, considerando esta última expresión, (2.3) puede ser reescrita de la forma:

$$\hat{y}(t+k|t) = \sum_{i=1}^k g_i \Delta u(t+k-i) + \sum_{i=k+1}^{\infty} g_i \Delta u(t+k-i) + y_m(t) - \sum_{i=1}^{\infty} g_i \Delta u(t-i) \quad (2.5)$$

$$\hat{y}(t+k|t) = \sum_{i=1}^k g_i \Delta u(t+k-i) + f(t+k) \quad (2.6)$$

En (2.6) aparece un nuevo término $f(t+k)$ el cual representa la respuesta libre del sistema, respuesta que no depende de las futuras acciones de control y que se obtiene al realizar un cambio de variable en el segundo elemento de (2.5). Entonces, se cumple que

$$f(t+k|t) = y_m(t) + \sum_{i=1}^{\infty} (g_{k+i} - g_i) \Delta u(t-i) \quad (2.7)$$

y en el caso de que se trate de un sistema asintóticamente estable, los coeficientes g_i tienden a mantener un valor constante después de N periodos de muestreo. En ese caso, se considera

$$g_{k+i} - g_i \approx 0, \quad i > N \quad (2.8)$$

y, por ende, la respuesta libre del sistema quedará descrita por (2.9). Cabe mencionar que cuando el sistema no es asintóticamente estable, no es posible calcular $f(t+k)$.

$$f(t+k|t) = y_m(t) + \sum_{i=1}^N (g_{k+i} - g_i) \Delta u(t-i) \quad (2.9)$$

Con lo anterior, es posible calcular las predicciones de la salida sobre un horizonte de predicción p , considerando m acciones de control (horizonte de control). Cada predicción tendrá la forma:

$$\begin{aligned} \hat{y}(t+1|t) &= g_1 \Delta u(t) + f(t+1) \\ \hat{y}(t+2|t) &= g_2 \Delta u(t) + g_1 \Delta u(t+1) + f(t+2) \\ &\vdots \\ \hat{y}(t+p|t) &= \sum_{i=p-m+1}^p g_i \Delta u(t+p-i) + f(t+p) \end{aligned} \quad (2.10)$$

La expresión anterior puede representarse de manera más sencilla de forma matricial al definir una matriz G , de dimensiones $p \times m$, conocida como la matriz dinámica del sistema. Esta matriz contiene a todos los coeficientes provenientes del modelo de respuesta al escalón de la planta. Se tiene entonces las siguientes matrices:

$$G = \begin{bmatrix} g_1 & 0 & \cdots & 0 \\ g_2 & g_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_m & g_{m-1} & \cdots & g_1 \\ \vdots & \vdots & \ddots & \vdots \\ g_p & g_{p-1} & \cdots & g_{p-m+1} \end{bmatrix} \quad (2.11)$$

$$\hat{\mathbf{y}} = [y_1(t+1|t), \dots, y_1(t+p|t), \dots, y_{ny}(t+1|t), \dots, y_{ny}(t+p|t)]^T \quad (2.12)$$

$$\mathbf{u} = [\Delta u_1(t), \dots, \Delta u_1(t+m_1-1), \dots, \Delta u_{nu}(t), \dots, \Delta u_{nu}(t+m_{nu}-1)]^T \quad (2.13)$$

$$\mathbf{f} = [f_1(t+1|t), \dots, f_1(t+p|t), \dots, f_{ny}(t+1|t), \dots, f_{ny}(t+p|t)]^T \quad (2.14)$$

Por lo tanto, las predicciones de la salida del sistema se pueden representar de manera simplificada mediante (2.15). Esta será la expresión que se utilizará para calcular las acciones de control futuras necesarias para obtener la respuesta deseada del sistema.

$$\hat{\mathbf{y}} = \mathbf{G}\mathbf{u} + \mathbf{f} \quad (2.15)$$

2.2.3. Función costo y algoritmo de control

El objetivo principal del controlador DMC es lograr que la salida del sistema tenga una respuesta lo más cercana posible a la señal de referencia. Es por ello que las acciones de control se calculan de manera que se minimicen los errores futuros que se encuentran dentro de una función costo cuadrática J , que puede incluir los esfuerzos de control. Esta función queda definida de la siguiente forma:

$$J = \sum_{j=1}^p [\hat{y}(t+j|t) - w(t+j)]^2 + \sum_{j=1}^m \lambda [\Delta u(t+j-1)]^2 \quad (2.16)$$

donde $w(t+j)$ representa la referencia para un instante j posterior. En este caso, la función costo tiene como parámetros a el horizonte de predicción p , el horizonte de control m y al factor λ , el cual pondera la importancia de los esfuerzos de control cuando se minimiza J . Así, cuando $\lambda=0$ significa que el objetivo primordial es lograr minimizar los errores futuros sin prestar atención a cuánto puedan variar las acciones de control.

La función costo J puede ser representada en forma matricial considerando las matrices \mathbf{u} , $\hat{\mathbf{y}}$, y $\mathbf{w} = [w(t+1)w(t+2) \dots w(t+p)]^T$, con lo cual se obtienen las siguientes expresiones:

$$J = (\hat{\mathbf{y}} - \mathbf{w})^T (\hat{\mathbf{y}} - \mathbf{w}) + \lambda \mathbf{u}^T \mathbf{u} \quad (2.17)$$

$$J = (\mathbf{G}\mathbf{u} + \mathbf{f} - \mathbf{w})^T (\mathbf{G}\mathbf{u} + \mathbf{f} - \mathbf{w}) + \lambda \mathbf{u}^T \mathbf{u} \quad (2.18)$$

Luego, en el caso de que no se consideren restricciones, se puede calcular las acciones de control óptimas al hallar la derivada de la función costo J respecto de \mathbf{u} ($\frac{\partial J}{\partial \mathbf{u}}$) e igualarla a cero.

El resultado de ese procedimiento es el siguiente:

$$\mathbf{u} = (\mathbf{G}^T \mathbf{G} - \lambda \cdot \mathbf{I}_{m \times m})^{-1} \mathbf{G}^T (\mathbf{w} - \mathbf{f}) \quad (2.19)$$

Es importante recalcar que solo la primera fila del vector \mathbf{u} ($\Delta u(t)$) es aplicada a la planta real ya que las salidas reales siempre van a diferir de las predicciones debido a las perturbaciones.

2.2.4. DMC en sistemas multivariantes

Una de las razones por las cuales la estrategia DMC es exitosa en la industria se debe a su habilidad de lidiar con procesos multivariantes. El modelo de respuesta al escalón del sistema puede trasladarse de manera sencilla a plantas MIMO. Para ello, si se tiene una planta con n_u entradas, cada una de sus salidas y_j reflejará el efecto de todas las entradas de la siguiente manera:

$$y_j(t) = \sum_{k=1}^{n_u} \sum_{i=1}^{N_k} g_i^{kj} \Delta u^k(t-i) \quad (2.20)$$

en donde g_i^{kj} es la respuesta de la salida j debido a la aplicación de un escalón en la entrada k .

Se mantienen las mismas ecuaciones básicas de la formulación del DMC para sistemas SISO, con excepción de los vectores y matrices que ahora serán de mayor dimensión y con una definición diferente. Gracias a la linealidad del modelo, las predicciones de cada salida pueden superponerse y dan origen a un nuevo vector de predicciones:

$$\hat{\mathbf{y}} = [y_1(t+1|t), \dots, y_1(t+p_1|t), \dots, y_{n_y}(t+1|t), \dots, y_{n_y}(t+p_{n_y}|t)]^T \quad (2.21)$$

De igual forma, se redefinen los vectores de las futuras acciones de control de las n_u entradas y de la respuesta libre de la planta.

$$\mathbf{u} = [\Delta u_1(t), \dots, \Delta u_1(t+m_1-1), \dots, \Delta u_{n_u}(t), \dots, \Delta u_{n_u}(t+m_{n_u}-1)]^T \quad (2.22)$$

$$\mathbf{f} = [f_1(t+1|t), \dots, f_1(t+p_1|t), \dots, f_{n_y}(t+1|t), \dots, f_{n_y}(t+p_{n_y}|t)]^T \quad (2.23)$$

La respuesta libre de cada salida ($f_j(t+k)$) se calcula en base a los valores pasados de la salida correspondiente y a todas las acciones de control pasadas. Con los vectores ya definidos, la ecuación de predicción (2.15) se mantiene igual considerando a la matriz \mathbf{G} como:

$$\mathbf{G} = \begin{bmatrix} G_{11} & G_{12} & \cdots & G_{1n_u} \\ G_{21} & G_{22} & \cdots & G_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ G_{n_y} & G_{n_y2} & \cdots & G_{n_y n_u} \end{bmatrix} \quad (2.24)$$

en donde cada matriz G_{ji} contiene los coeficientes de la respuesta al escalón de la j -ésima salida debido a la i -ésima entrada.

Por otro lado, la ley de control para estos casos se obtiene mediante la minimización de una nueva función costo que incluye matrices de ponderación R y Q , de dimensiones $ns.p \times ns.p$ y $nu.m \times nu.m$, para los errores futuros y los incrementos de control, respectivamente. La función costo queda representada mediante la siguiente expresión:

$$J = \sum_{j=1}^p \|\hat{y}(t+j|t) - w(t+j)\|_R^2 + \sum_{j=1}^m \|\Delta u(t+j-1)\|_Q^2 \quad (2.25)$$

Finalmente, los valores óptimos de los incrementos de control futuros se calculan con el mismo procedimiento que en el caso de una sola variable. Se obtiene la siguiente solución analítica cuando no se consideran restricciones:

$$\mathbf{u} = (G^T R G - Q)^{-1} G^T R (\mathbf{w} - \mathbf{f}) \quad (2.26)$$

Siguiendo el mismo principio que en el caso SISO, solamente los primeros incrementos en las acciones de control ($\Delta u_i(t)$) serán los que ingresen a la planta real.

2.3. Modelamiento de la planta piloto

Tal y como se describió en el capítulo anterior, la configuración de la planta piloto CTS para este trabajo consta de dos tanques hidráulicos, interconectados mediante una tubería y una válvula de conexión, que son alimentados por bombas centrífugas conectadas a un reservorio de agua. Cada tanque tiene un orificio de fuga conectado a una válvula para la salida de líquido hacia el reservorio, ver Fig. 2.1.

Las entradas de este proceso son los voltajes u_1 y u_2 de cada una de las bombas centrífugas y las salidas son los niveles de líquido en cada tanque, h_1 y h_2 . El objetivo es controlar estas

salidas mediante los caudales de entrada (Q_{i1} , Q_{i2}) tomando en consideración los flujos de salida de cada tanque (Q_{o1} , Q_{o2}) y el efecto producido por el flujo en la interconexión (Q_x).

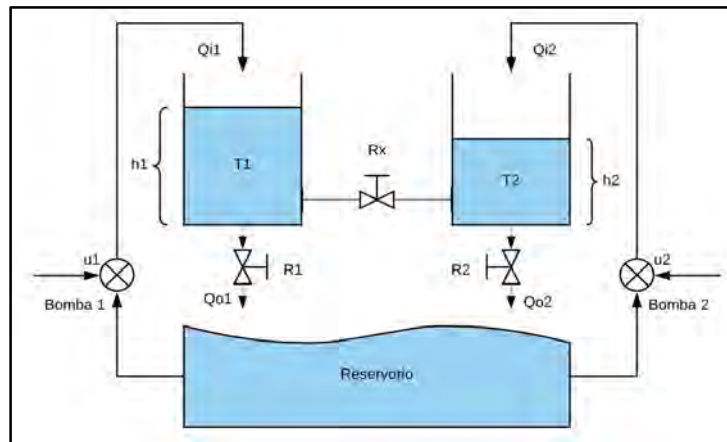


Figura 2.1. Esquema de la planta piloto.

Para el modelamiento de la planta mencionada se toma como referencia los trabajos [3] y [6]. En primer lugar, siguiendo el principio de Torricelli, los flujos de salida de cada uno de los tanques y el flujo por la tubería de interconexión se expresan como:

$$Q_{oi} = b_i a_i \sqrt{2gh_i} \quad (2.27)$$

$$Q_x = b_x a_x \operatorname{sgn}(h_1 - h_2) \sqrt{2g|h_1 - h_2|} \quad (2.28)$$

donde a_i , a_x representan la sección de orificios de salida; b_i , b_x , los parámetros de apertura de las válvulas y h_i , el nivel de líquido en el tanque correspondiente. El sentido del flujo en la tubería de interconexión dependerá de la cantidad de líquido en cada tanque ya que este flujo será de salida para el tanque con mayor nivel.

Luego, utilizando las dos ecuaciones anteriores, se puede obtener las ecuaciones no lineales de la planta:

$$\frac{dh_1(t)}{dt} = \frac{k_p}{A} u_1(t) - \frac{b_1 a_1}{A} \sqrt{2gh_1(t)} - \frac{b_x a_x}{A} \operatorname{sgn}(h_1(t) - h_2(t)) \sqrt{2g|h_1(t) - h_2(t)|} \quad (2.29)$$

$$\frac{dh_2(t)}{dt} = \frac{k_p}{A} u_2(t) - \frac{b_2 a_2}{A} \sqrt{2gh_2(t)} + \frac{b_x a_x}{A} \operatorname{sgn}(h_1(t) - h_2(t)) \sqrt{2g|h_1(t) - h_2(t)|} \quad (2.30)$$

donde:

A: sección transversal de los tanques (cm^2)

a_i : sección transversal del orificio de salida del tanque i (cm^2)

b_i : parámetro de apertura de las válvulas de salida del tanque i

b_x : parámetro de apertura de la válvula de conexión

h_i : altura del tanque i (m)

g : aceleración de la gravedad ($\frac{m}{s^2}$)

u_i : voltaje de entrada a de la bomba (V)

k_p : ganancia de las bombas ($\frac{cm^3}{V*s}$)

Seguidamente, para poder linealizar el sistema se debe de definir un punto de operación h_i^o para cada tanque, alrededor de los cuales la planta tendrá un comportamiento lineal. Se define la variable \bar{h}_i que representa la diferencia entre el nivel real y el punto de operación elegido. Entonces, con \bar{h}_1 , \bar{h}_2 se pueden definir las constantes de tiempo para cada tanque y para la interconexión, T_1, T_2 y T_x , con los cuales será posible obtener las ecuaciones lineales del sistema:

$$\frac{dh_1(t)}{dt} = \frac{k_p}{A} u_1(t) - \frac{h_1(t)}{T_1} + \frac{h_2(t) - h_1(t)}{T_x} \quad (2.31)$$

$$\frac{dh_2(t)}{dt} = \frac{k_p}{A} u_2(t) - \frac{h_2(t)}{T_2} - \frac{h_2(t) - h_1(t)}{T_x} \quad (2.32)$$

Consecutivamente, se aplica la transformada de Laplace en las ecuaciones (2.31) y (2.32) con la finalidad de obtener las funciones de transferencia de la planta. Por lo tanto, en (2.33) se puede observar el modelo de la planta en el dominio de Laplace de forma matricial.

$$\begin{bmatrix} h_1(s) \\ h_2(s) \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} \quad (2.33)$$

Los términos $G_{11}(s)$ y $G_{22}(s)$ representan las funciones de transferencia de cada una de las salidas respecto a su entrada correspondiente. La función $G_{12}(s)$ representa el efecto en el tanque 1 debido a la entrada u_2 y; de manera similar, la función $G_{21}(s)$ relaciona al tanque 2 con la entrada u_1 . Este conjunto de funciones transferencia conforma una matriz denominada como la Matriz Transferencia de la planta.

Por último, se muestran las expresiones para cada elemento de la matriz transferencia en función de las constantes de tiempo T_1 , T_2 y T_x :

$$G_{11}(s) = \frac{\frac{k}{A} \left(s + \frac{T_x + T_2}{T_2 T_x} \right)}{s^2 + \left(\frac{T_1 T_x + T_2 T_x + 2T_1 T_2}{T_1 T_2 T_x} \right) s + \left(\frac{1}{T_1 T_2} + \frac{1}{T_1 T_x} + \frac{1}{T_2 T_x} \right)}$$

$$G_{12}(s) = \frac{\frac{k}{A} \left(\frac{1}{T_x} \right)}{s^2 + \left(\frac{T_1 T_x + T_2 T_x + 2T_1 T_2}{T_1 T_2 T_x} \right) s + \left(\frac{1}{T_1 T_2} + \frac{1}{T_1 T_x} + \frac{1}{T_2 T_x} \right)}$$

$$G_{21}(s) = \frac{\frac{k}{A} \left(\frac{1}{T_x} \right)}{s^2 + \left(\frac{T_1 T_x + T_2 T_x + 2T_1 T_2}{T_1 T_2 T_x} \right) s + \left(\frac{1}{T_1 T_2} + \frac{1}{T_1 T_x} + \frac{1}{T_2 T_x} \right)}$$

$$G_{22}(s) = \frac{\frac{k}{A} \left(s + \frac{T_x + T_1}{T_1 T_x} \right)}{s^2 + \left(\frac{T_1 T_x + T_2 T_x + 2T_1 T_2}{T_1 T_2 T_x} \right) s + \left(\frac{1}{T_1 T_2} + \frac{1}{T_1 T_x} + \frac{1}{T_2 T_x} \right)}$$

2.4. Requerimientos para aplicar el DMC al proceso de tanques acoplados TITO

A modo de conclusión de este capítulo, se presenta la propuesta de solución que se aplicará en esta tesis, en donde se explica de manera general cómo aplicar la estrategia DMC en el proceso TITO estudiado.

En primer lugar, se calculará la matriz transferencia de la planta TITO en base a los parámetros físicos de la planta piloto del laboratorio y a determinados puntos de operación (h_1^o, h_2^o). Con los valores de esta matriz transferencia será posible obtener la matriz dinámica del sistema G .

En segundo lugar, se procederá con la sintonización de los parámetros del controlador, los cuales son el horizonte de predicción, el horizonte de control, las matrices de ponderación R y Q , y la señal de referencia para instantes futuros. La Fig. 2.2 presenta el diagrama de bloques del sistema de control y sus interacciones con la planta. Con todo ello, se podrá definir funciones en Matlab para el cálculo de la ley de control y de las predicciones. Posteriormente, se realizarán pruebas de simulación del controlador para analizar el desempeño del mismo ante variaciones en algunos de sus parámetros.

Finalmente, concluido el diseño del controlador, se desarrollará el diagrama de flujo del algoritmo de control desarrollado y se escribirá un programa del mismo para mostrar que el controlador propuesto puede ser implementado en un controlador industrial.

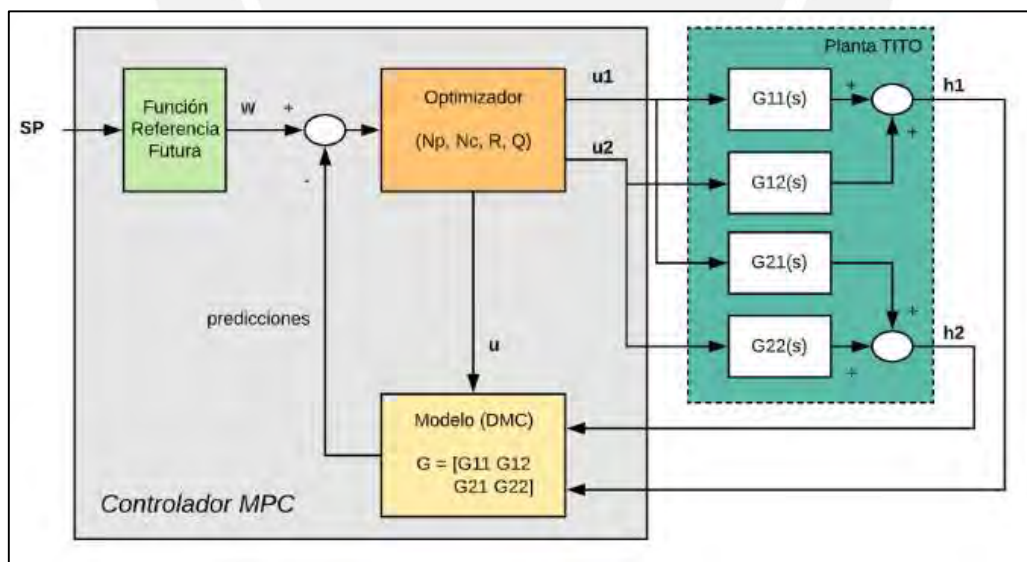


Figura 2.2. Diagrama de bloques del sistema de control.

3. Diseño del Controlador DMC para la planta TITO. Aplicación de Métodos de Sintonización

3.1. Introducción

En el presente capítulo, se describe la metodología de diseño empleada para llevar a cabo la propuesta de solución de esta tesis que consiste de un controlador predictivo DMC para el proceso multivariable de dos tanques acoplados. Los partes fundamentales de este diseño se centran en el desarrollo de un algoritmo de control que permita implementar el funcionamiento del controlador, que se detalló en el marco teórico, y en la sintonización de los parámetros del mismo de manera que se cumplan ciertos requerimientos.

Esos requerimientos de diseño se centran en el desempeño del controlador predictivo. Por lo tanto, se requiere que este control mejore la respuesta en el tiempo de la planta. De igual manera, ya que se trata de un control centralizado sin etapas de desacoplamiento, se debe lograr reducir el efecto de las interacciones entre los dos lazos de control presentes en la planta.

Por otro lado, también es necesario que el controlador pueda responder satisfactoriamente ante la presencia de perturbaciones en las salidas del proceso y que presente cierto nivel de robustez ante incertidumbre en los parámetros del modelo. Por último, existe un requerimiento adicional relacionado con lograr una reducción en el costo computacional del controlador a partir de la sintonización de sus parámetros.

En la sección 3.2, se aborda el procedimiento utilizado para calcular la matriz dinámica de la planta que servirá como modelo de predicción en el controlador. En la sección 3.3, se detallan los pasos del algoritmo de control desarrollado. Finalmente, en la sección 3.4, se realiza la sintonización del controlador siguiendo dos métodos: uno clásico (Shridhar&Cooper) y otro basado en el algoritmo de optimización por enjambre de partículas (PSO).

3.2. Matriz Dinámica de la planta piloto

3.2.1. Cálculo de la Matriz Transferencia

Como paso previo al diseño del controlador DMC es necesario contar con el modelo de respuesta al escalón de la planta TITO, el cual será el modelo de predicción utilizado por el controlador. Este modelo es denominado como la Matriz Dinámica (G) del sistema.

El primer paso para obtener G es calcular las funciones de transferencia de los cuatro subprocesos que conforman la planta en base a los valores numéricos de los parámetros de la planta piloto (ver Tabla 3.1) y a los puntos de operación, alrededor de los cuales se linealiza el modelo de la planta. La elección de estos puntos de operación ($h1_0, h2_0$) debe darse de tal manera que estos no requieran valores negativos en las entradas ($u1_0, u2_0$), los cuales se pueden calcular a partir de las ecuaciones no lineales de planta (2.29 y 2.30), ya que ello significaría un cambio en la dirección de los flujos de alimentación de los tanques.

En este caso, se eligieron como puntos de operación $h1_0 = 13 \text{ cm}$ y $h2_0 = 10 \text{ cm}$, a los cuales les corresponden entradas iguales a $u1_0 = 97,58 \text{ V}$ y $u2_0 = 16,29 \text{ V}$, respectivamente.

Tabla 3.1. Parámetros de la planta piloto

Parámetro	Valor
Área de los tanques (A)	754,76 [cm^2]
Sección de los orificios de salida (a_1, a_2)	1,266 [cm^2]
Sección del orificio de interconexión (a_x)	1,6 [cm^2]
Ganancia de la bomba centrífuga 1 (k_1)	3,33 [$\frac{\text{cm}^3}{\text{V} \cdot \text{s}}$]
Ganancia de la bomba centrífuga 2 (k_2)	3,35 [$\frac{\text{cm}^3}{\text{V} \cdot \text{s}}$]

Por consiguiente, se puede calcular los valores de las linealizaciones T_1, T_2 y T_x para luego caracterizar las funciones de transferencia de cada subproceso siguiendo las ecuaciones (2.34) hasta (2.37). De esta manera, a cada subproceso le corresponde una de las siguientes funciones:

$$G_{11}(s) = \frac{0,004412s + 0,0004086}{s^2 + 0,1825s + 0,003917}$$

$$G_{12}(s) = \frac{0,0002946}{s^2 + 0,1825s + 0,003917}$$

$$G_{21}(s) = \frac{0,0002928}{s^2 + 0,1825s + 0,003917}$$

$$G_{22}(s) = \frac{0,004438s + 0,0003988}{s^2 + 0,1825s + 0,003917}$$

3.2.2. Selección del Horizonte del Modelo y Tiempo de muestreo

Dado que el controlador DMC a diseñar es de naturaleza discreta, resulta necesario definir un tiempo de muestreo (T_s) adecuado para controlar esta planta. El T_s debe permitir capturar la dinámica del subproceso más rápido de la planta, el cual se caracteriza por tener la menor constante de tiempo.

Entonces, para poder identificar esta constante de tiempo de manera más sencilla se aproximan las funciones de transferencia anteriores a modelos de primer orden más tiempo muerto (FOPDT). Realizar esas aproximaciones es posible gracias a que las respuestas al escalón de cada subproceso son asintóticamente estables, lo cual es a su vez un requisito indispensable para poder utilizar la técnica DMC.

Luego de evaluar las respuestas al escalón, se obtuvieron las siguientes aproximaciones:

$$\begin{aligned} G'_{11}(s) &= \frac{0,237}{1+76,3s}; & G'_{12}(s) &= \frac{0,217}{1+106} \\ G'_{21}(s) &= \frac{0,216}{1+106}; & G'_{22}(s) &= \frac{0,229}{1+74,2s} \end{aligned} \quad (3.1)$$

Se identifica entonces que la menor constante de tiempo de la planta es igual a 106 segundos.

Como regla práctica, este valor se divide entre diez y así se logra definir el $T_s = 11$ s.

Por otro lado, el horizonte del modelo (N) define el máximo número de valores de la respuesta al escalón de cada subproceso que se utilizarán para conformar la matriz dinámica G . El valor de N debe ser lo suficientemente alto de manera que permita incluir valores en los cuales las respuestas se tornan asintóticamente estables. La manera más empleada en la literatura para calcular el valor de N sigue la siguiente fórmula:

$$N = \max\left(\frac{5*\tau_{ij}+1}{T_s}\right) \quad (3.2)$$

en donde τ_{ij} representa la constante de tiempo del modelo FOPDT de cada subproceso. De esta manera, se escoge el horizonte del modelo $N=50$.

3.2.3. Cálculo de los coeficientes de la Matriz Dinámica

Los coeficientes que conforman la matriz G vienen a ser valores específicos de las respuestas al escalón tomados en intervalos iguales al T_s y se les denota como coeficientes " g_j ", en donde j toma valores desde 1 hasta N . Antes de poder calcular estos coeficientes es necesario contar con las funciones de transferencia discretas de los cuatro subprocesos.

Ya conociendo los coeficientes b_i del numerador y a_i del denominador de estas funciones discretas, se puede calcular los coeficientes g_j para cada subproceso siguiendo la siguiente ecuación recursiva:

$$g_j = -\sum_{i=1}^j a_i g_{j-1} + \sum_{i=0}^{j-1} b_i \quad (3.3)$$

La expresión anterior se implementó en Matlab mediante la función "*plant_coef.m*". Esta función recibe como parámetros de entrada los vectores numerador y denominador de la FT discreta, los cuales contienen los valores de b_i y a_i , además del horizonte N . Como resultado

se obtiene un vector fila de N elementos con los coeficientes g_j que caracterizan a la respuesta escalón del subproceso correspondiente.

Posteriormente, se forman cuatro matrices G_{xy} , de la forma mostrada en (2.11), para cada subproceso utilizando los coeficientes g_j hallados. Las dimensiones de estas matrices dependerán del horizonte de predicción (P), para el número de filas, y del horizonte de control (M), para las columnas. Ambos horizontes son parámetros propios del controlador DMC y se definirán más adelante. Finalmente, estas cuatro matrices se reordenan para llegar a formar la matriz dinámica G del sistema completo, la cual tomaría la siguiente forma:

$$G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$$

3.3. Diseño de la lógica de control

3.3.1. Consideraciones respecto al algoritmo de control

En primer lugar, ya que la matriz dinámica corresponde al modelo lineal de la planta, las variaciones en las acciones de control (Δu_1 y Δu_2) que se calculan con el algoritmo están vinculadas directamente a las entradas linealizadas de la planta (\bar{u}_1 y \bar{u}_2) y no a las entradas reales de la misma, las cuales vienen a ser los voltajes de alimentación (u_1 y u_2) de las bombas centrífugas. En otras palabras, los valores de \bar{u}_1 y \bar{u}_2 describen en realidad la diferencia que existe entre las entradas reales y sus respectivos puntos de operación (u_{1_0} y u_{2_0}). Es por ello que en el algoritmo de control se suma las acciones de control calculadas con los puntos de operación para formar las entradas reales de la planta.

En segundo lugar, ya que el controlador DMC real requiere de los valores sensados de las variables de salida para su funcionamiento, se van a generar estos valores a partir de las ecuaciones no lineales de planta, ecuaciones (2.29) y (2.30). De esta manera, se busca lograr una mayor aproximación de las variables de salida reales.

En tercer lugar, para este trabajo no se incluye el efecto de ruido que podrían presentar las variables de salida de la planta real. Sin embargo, la lógica de control sí considera la adición y generación de perturbaciones en las salidas.

3.3.2. Diagrama de flujo del controlador DMC

A continuación, se describe la secuencia de pasos que sigue el programa desarrollado para implementar el algoritmo de control DMC.

- a) Definición de los valores para el horizonte de predicción (P), el horizonte de control (M), las matrices de ponderación R y Q , el tiempo de muestreo (T_s), tiempo de simulación ($tsim$) y los puntos de operación.
- b) Cálculo de la matriz de ganancias de la ley de control $\mathbf{K} = (G^T R G - Q)^{-1} G^T R$. Debido al principio de horizonte de retroceso, solo se utilizan las filas 1 y $M+1$ de la matriz K , las cuales serán los vectores ganancia \mathbf{k}_1 y \mathbf{k}_2 para $\Delta u_1(t)$ y $\Delta u_2(t)$, respectivamente.
- c) Inicialización de los vectores para variables de entradas y variables de salida de la planta.
- d) Definición de los vectores de *Set Point* para las variables de salida.
- e) Inicialización del bucle para el cálculo de las acciones de control y variables del proceso a lo largo del $tsim$: se inicializan vectores de referencias futuras (w), vectores de respuesta libre (f) y vectores de variaciones de control pasadas.
- f) Cálculo de las repuestas libres (f_1, f_2) a partir de las salidas anteriores, los vectores de variaciones de control pasadas y los coeficientes " g_j ".
- g) Formación de vectores de referencias futuras (w_1, w_2) a partir los vectores de *Set Point* antes definidos.
- h) Cálculo de las variaciones $\Delta u_1(t)$ y $\Delta u_2(t)$ conforme a la ecuación (2.26) y utilizando los vectores formados en los puntos f) y g).

- i) Actualización de las variables de control $\overline{u1}$ y $\overline{u2}$ en base a las variaciones halladas en el punto anterior y teniendo en cuenta la saturación de las mismas para no excedan límites físicos. Se suman los puntos de operación para generar y almacenar las variables de control $u1$ y $u2$.
- j) Cálculo de las variaciones en las variables de salida $h1$ y $h2$ utilizando las ecuaciones no lineales de la planta. Se actualizan y almacenan los valores estas variables.
- k) Deslizamiento de la ventana de predicción a partir de almacenar los valores de $\Delta u1(t)$ y $\Delta u2(t)$, calculados en esta iteración, en el vector de variaciones de control pasadas.
- l) Fin del programa cuando se haya completado el *tsim*.

De esta manera, se ha logrado implementar la ley de control asociada a un controlador DMC en un entorno de programación. Este será el programa utilizado para realizar las pruebas y simulaciones del siguiente capítulo.

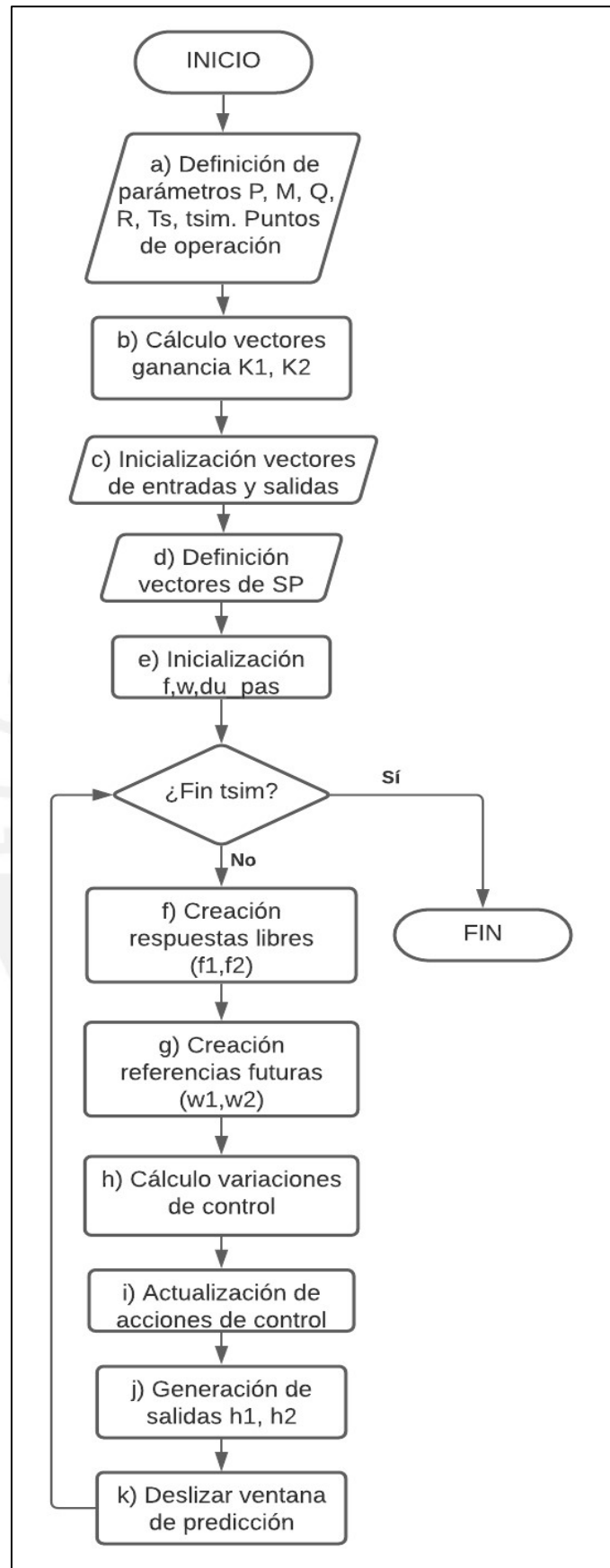


Figura 3.1. Diagrama de flujo del controlador.

3.4. Sintonización del controlador DMC MIMO

3.4.1. Necesidad de Métodos de Sintonización

En las secciones anteriores, se pudo notar la gran influencia que ejercen el horizonte de predicción (P) y el horizonte de control (M) en distintos bloques del controlador DMC. Estos parámetros son de suma importancia ya que sus valores van a definir el desempeño de la respuesta del sistema de control. Adicionalmente a ellos, la matriz de ponderación Q también juega un rol importante en el control ya que esta evita que el esfuerzo de control sea muy exigido o agresivo. Resulta entonces necesario realizar una adecuada selección de estos parámetros para lograr un control exitoso.

En la literatura existen distintos métodos de sintonización para los controladores DMC. Algunos de ellos establecen reglas de sintonización explícitas en base a parámetros de la planta a controlar. Otros en cambio hacen uso de algoritmos de optimización para buscar los mejores valores de P , M y Q que se ajusten al comportamiento específico de la planta. En esta tesis, se va a utilizar un método de cada tipo para luego comparar los resultados obtenidos. El primero de ellos es el método clásico de Shridar y Cooper [20] mientras que el segundo método emplea para la sintonización el algoritmo de Optimización por Enjambre de Partículas (PSO, por sus siglas en inglés).

3.4.2. Método clásico de Shridar y Cooper

Este método de sintonización se caracteriza por brindar reglas explícitas y prácticas para el cálculo de los parámetros P , M y Q . Para realizar la sintonización, el método se basa en los modelos aproximados FOPDT de cada uno de los subprocesos de la planta. Estos modelos ya se conocen y se pueden agrupar en la siguiente matriz:

$$G_{FOPDT}(s) = \begin{pmatrix} \frac{0,237}{1+76,3s} & \frac{0,217}{1+106s} \\ \frac{0,216}{1+106} & \frac{0,229}{1+74,2s} \end{pmatrix} \quad (3.4)$$

Cada uno de los elementos de esa matriz cuenta con una ganancia K_{rs} y una constante de tiempo τ_{rs} , en donde el valor de $r = (1 \text{ o } 2)$ hace referencia a una de las salidas del sistema y; de manera similar, $s = (1 \text{ o } 2)$ se asocia con una de las entradas. Ninguno de los modelos tiene tiempo muerto pues este es mínimo y se puede considerar despreciable en esta planta.

Una vez que se han identificado estos parámetros para cada modelo, el método sugiere evaluar funciones que dependen de los anteriores y del tiempo de muestreo para sintonizar el controlador DMC. Entonces, para el horizonte de predicción se tiene la siguiente fórmula:

$$P = N = \text{Max}\left(\frac{5*\tau_{rs}}{T_s} + 1\right) \quad (3.5)$$

Como se puede ver, el método recomienda escoger P igual al horizonte del modelo N y su valor está relacionado directamente con la mayor constante de tiempo de la planta. Resolviendo la ecuación se define el horizonte de predicción $P = 50$.

Por otro lado, para M se recomienda que este se igual al 63.2% del tiempo de establecimiento del subproceso más lento, lo cual se describe por la siguiente ecuación:

$$M = \text{Max}\left(\frac{\tau_{rs}}{T_s} + 1\right) \quad (3.6)$$

De esta manera, se define el horizonte de control $M = 11$.

Por último, la matriz Q se define a partir de los coeficientes que la conforman. Estos son denominados como coeficientes de supresión de movimiento y se representan como q_s . Cada una de las entradas tendrá un coeficiente q_s asociado y estos son calculados de la siguiente manera:

$$q_s = \frac{M}{500} \sum_{r=1}^2 \left[K_{rs}^2 \left\{ P + 1 - \frac{3\tau_{rs}}{2T_s} - \frac{M-1}{2} \right\} \right] \quad (3.7)$$

Por consiguiente, al resolver la ecuación anterior se obtienen los valores de $q_1 = 0,08$; $q_2 = 0,07$.

Luego, la matriz Q se define de la siguiente manera:

$$Q = \begin{bmatrix} q_1 I & 0 \\ 0 & q_2 I \end{bmatrix}_{2M \times 2M}$$

Si bien se han obtenido valores específicos para los coeficientes de Q , estos podrían tomarse como un valor referencial a partir del cual se puede ir ajustando la sintonización del controlador dependiendo del desempeño que se quiera alcanzar. En resumen, los parámetros calculados por este método se detallan en la Tabla 3.2.

Tabla 3.2. Parámetros del controlador DMC por sintonización clásica

P	M	$q1$	$q2$
50	11	0,08	0,07

3.4.3. Método de sintonización mediante PSO

Este método de sintonización difiere del anterior en que el actual no define fórmulas prácticas para el cálculo de los parámetros del controlador. En cambio, en este caso se utiliza el algoritmo de optimización PSO que se encargará de hallar los valores óptimos de los parámetros al minimizar cierto índice de desempeño del controlador. Para explicar cómo se aplica el PSO en el diseño de un controlador DMC es necesario antes entender de manera simplificada el funcionamiento del algoritmo.

El algoritmo fue desarrollado por Eberhart y Kennedy [21] y está basado en la metáfora del comportamiento social de un determinado grupo de partículas en la naturaleza. Estas partículas conforman una población de soluciones candidatas para la optimización de cierto problema. El algoritmo modifica la posición y la velocidad de las partículas de manera que estas puedan recorrer un espacio de búsqueda y así llegar a identificar la mejor solución para el problema.

La posición de las mismas está compuesta por las variables de las cuales depende el problema. Este desplazamiento se realiza a partir de las siguientes expresiones:

$$v_i = v_i + c_1 \cdot r_1 \cdot (pbest_i - x_i) + c_2 \cdot r_2 \cdot (gbest - x_i)$$

$$x_i = x_i + v_i \quad (3.8)$$

en donde, x_i y v_i representan la posición y la velocidad de determinada partícula, respectivamente; c_1 y c_2 son constantes de atracción al mejor resultado de la partícula y al mejor resultado global; y r_1 y r_2 son valores aleatorios entre 0 y 1. El término $pbest_i$ es la mejor posición que ha obtenido cada partícula a lo largo de su historia, mientras que $gbest$ es la mejor posición obtenida en todo el enjambre de partículas.

Ahora bien, para determinar si una posición es mejor que otra se necesita de una función objetivo f que dependa de las variables del problema y que cuyo resultado sea un número escalar. De este modo, las posiciones de las partículas pueden ser evaluadas en f y aquellas que obtengan el menor o mayor resultado, según se trate de una minimización o maximización, se convertirán en el nuevo $pbest_i$ o $gbest$ del enjambre.

Entonces, el algoritmo realizará el número de iteraciones que se le defina y cuando haya terminado se tendrá como resultado de su búsqueda el mejor valor alcanzado para la función objetivo y la posición de la partícula que ha logrado esta salida. Por ejemplo, en el caso de la sintonización del controlador DMC, el algoritmo brindará como resultado el mínimo valor para el índice de desempeño definido y los parámetros P , M , $q1$, $q2$ (posición de la partícula) que hacen posible este resultado.

Una vez explicado el funcionamiento del PSO, se procede a detallar el procedimiento mediante el cual se ha sintonizado el controlador DMC utilizando el algoritmo PSO. Para ello se ha tomado como referencia el trabajo realizado por Han, Zhao y Qian [22].

En primer lugar, se debe definir la función objetivo que se busca minimizar. Como se mencionó líneas arriba, esta función será un índice de desempeño del sistema de control, similar a índices populares como ISE o ITAE, con el cual se puede cuantificar el desempeño del controlador. Los autores de [22] proponen utilizar un índice denominado como EISTWE que se describe mediante la siguiente ecuación:

$$EISTWE = \left(1 + \frac{r(a+b)}{SP}\right) \cdot \int_0^{\infty} (t^n \cdot e(t)^2 + q \cdot \Delta u(t)^2) dt + \rho \cdot P \cdot M \quad (3.9)$$

en donde $e(t)$ representa el error entre la referencia y la salida del sistema, mientras que $\Delta u(t)$ simboliza las variaciones en las acciones de control. También se observa que este índice toma en consideración los valores de P y M de manera explícita. El primer término de la ecuación (3.9) está relacionado con la respuesta en lazo cerrado del sistema de control, en donde a y b se ilustran en la Fig. 3.2.

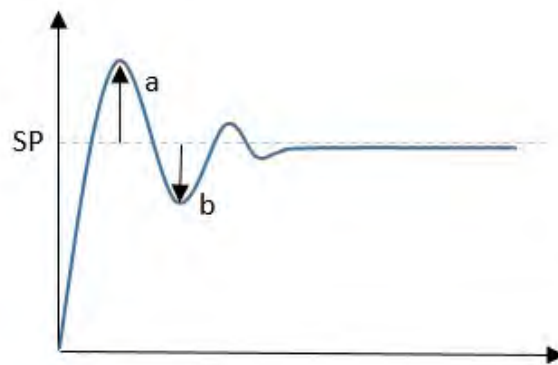


Figura 3.2. Parámetros a y b de la función EISTWE

En cuanto al resto de coeficientes que aparecen en el índice, se les ha asignado los siguientes valores:

Tabla 3.3. Parámetros auxiliares de la función EISTWE

r	n	q	ρ
1	1	0,1	1

Por otro lado, la forma en la cual el algoritmo PSO va a evaluar esta función será utilizando el algoritmo de control explicado en la sección 3.3.2. La posición de cada partícula le brindará los parámetros de entrada al controlador (P , M , $q1$, $q2$) y luego este realizará el cálculo de las acciones de control y variables del proceso para así devolver como salida al PSO el valor alcanzado por el índice EISTWE para la posición que se utilizó como entrada.

En segundo lugar, el algoritmo PSO requiere que se le especifique rangos de valores dentro de los cuales realizar la búsqueda de las posiciones de las partículas. En otras palabras, estos rangos van a delimitar el espacio de búsqueda del enjambre. Para definir los límites de estos rangos se utilizan como referencia los resultados obtenidos por la sintonización con el método clásico.

De esta manera, se define que el horizonte de control M podrá tomar valores en el rango [1; 11]. El horizonte de predicción P deberá tomar valores superiores a M y como máximo podrá ser igual al horizonte del modelo N , con lo cual su rango será [12; 50]. Por último, para los coeficientes $q1$ y $q2$ se tomará como límite superior el valor de 0,1 y como límite inferior el valor de 0,001, ello con el propósito de lograr mejores respuestas en las variables de salida.

Por lo tanto, tomando en consideración lo descrito anteriormente, ya es posible utilizar el algoritmo PSO para la sintonización del controlador. Para ello se ha utilizado la función de código libre “Particle_Swarm_Optimization.m” desarrollada por P. Biswas [23]. Para los coeficientes de la ecuación (3.9) se han utilizado los valores que recomienda este autor. En cuanto al número de partículas y el número de iteraciones a realizar por el algoritmo, estos se fueron modificando cada vez que se ejecutaba la función con la finalidad de obtener un banco de resultados para los parámetros del controlador.

Finalmente, luego de haber analizado los resultados obtenidos con esta estrategia de sintonización, se lista en la siguiente tabla los valores de los parámetros a utilizar para las simulaciones:

Tabla 3.4. Parámetros del controlador DMC por sintonización PSO

P	M	$q1$	$q2$
29	2	0,008	0,009



4. Pruebas de Simulación, Análisis de resultados y Propuesta de implementación del controlador DMC

4.1. Introducción

En el presente capítulo se desarrollan pruebas de simulación y posteriormente se realiza el análisis de los resultados a partir de las simulaciones con el controlador DMC diseñado. Mediante este análisis comparativo se busca determinar con cuál de las dos técnicas de sintonización aplicadas se logra un mejor desempeño del controlador de manera que se cumpla con los requerimientos detallados en el capítulo 3.

Las pruebas se han dividido en dos escenarios principales: en el primero se simula el funcionamiento del controlador utilizando el modelo nominal de la planta TITO; mientras que, en el segundo se realiza la simulación aplicando cierta incertidumbre en el modelo de la planta. De esta manera, se busca reconocer cuál de las técnicas de sintonización responde mejor ante incertidumbres en el modelado.

Por último, el capítulo concluye presentando la propuesta de implementación del sistema de control DMC diseñado. Para ello se desarrolla un controlador en Labview que interactúa con la planta experimental modelada en Simulink simulando el comportamiento real de la planta. Además, se incluye una interfaz gráfica para el panel del control del sistema.

4.2. Pruebas de Simulación

4.2.1 Señales de Referencia

Para verificar el funcionamiento del controlador y analizar su desempeño según el conjunto de parámetros que utilice se realizan pruebas de simulación. En la Fig. 4.1. se muestran las señales de referencia (*set points*) para cada una de las variables del proceso de la planta TITO. Estas

señales se encuentran en una vecindad del 15% alrededor de los puntos de operación para los cuales se ha linealizado la planta. Además, estas señales se han definido de tal que los escalones en cada una de ellas ocurren en distintos instantes de tiempo. Con ello será posible observar el efecto del acoplamiento entre los lazos de control de la planta multivariable.

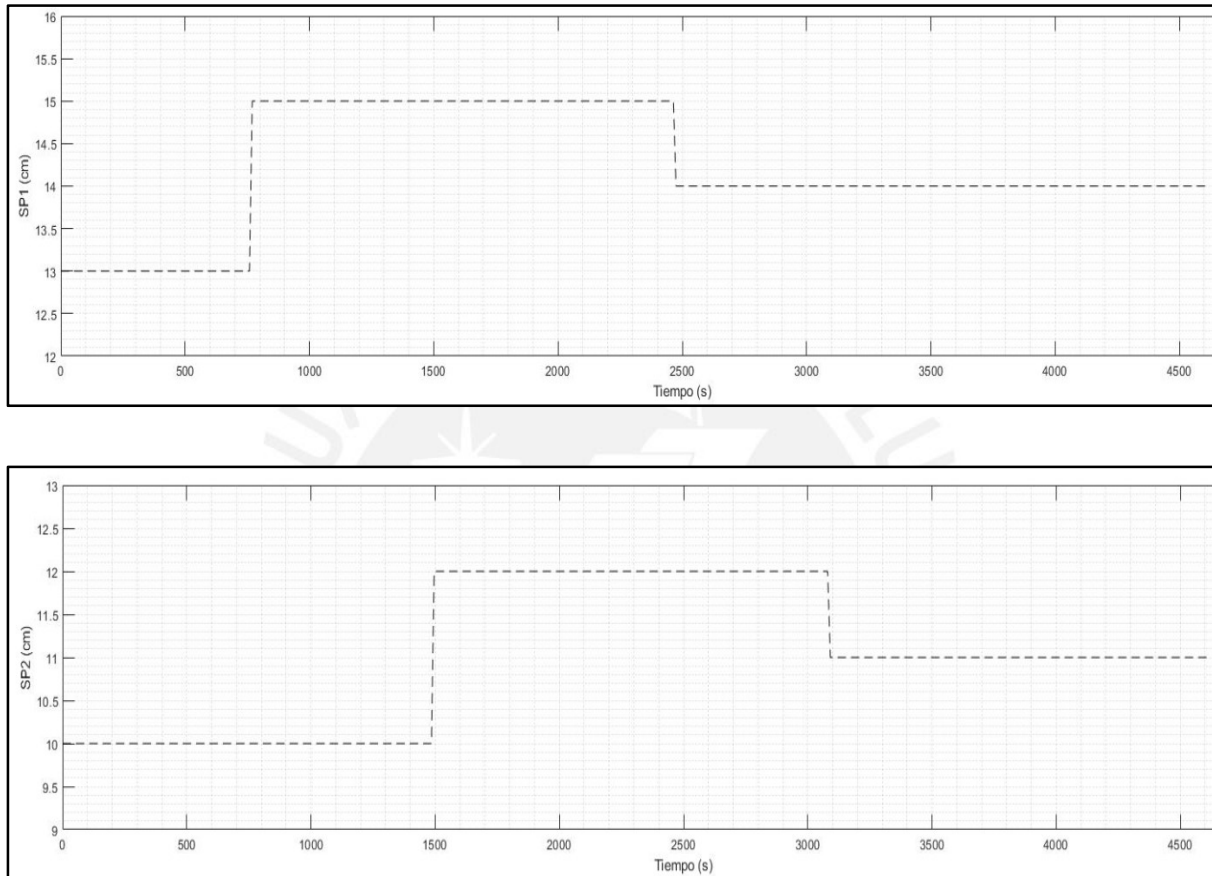


Figura 4.1. Señales de set point para los tanques.

Por otro lado, las simulaciones se han dividido en dos casos: el primero trabaja con el modelo nominal que se ha deducido para la planta objeto de estudio a partir de sus parámetros y los puntos de operación; el segundo, agrega variación del 30% en las ganancias de las FT del modelo nominal a fin de analizar la respuesta del controlador ante incertidumbre de modelado.

4.2.2. Pruebas con Modelo nominal

En primer lugar, se muestran los resultados para los parámetros obtenidos con la técnica clásica (ver Tabla 3.2): los niveles de los tanques en la Fig. 4.2. y las variables de control correspondientes en la Fig. 4.3.

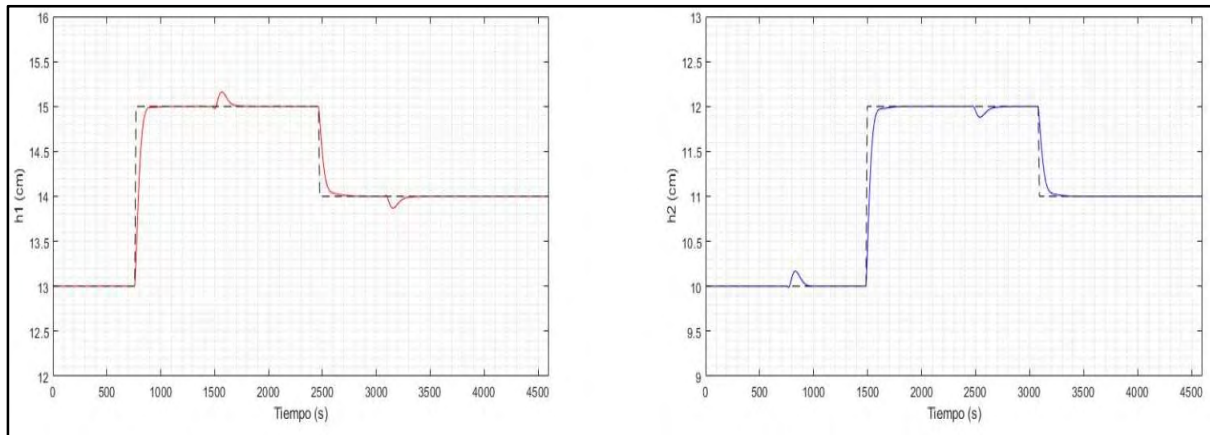


Figura 4.2. Respuesta del sistema en el caso nominal y sintonización clásica.

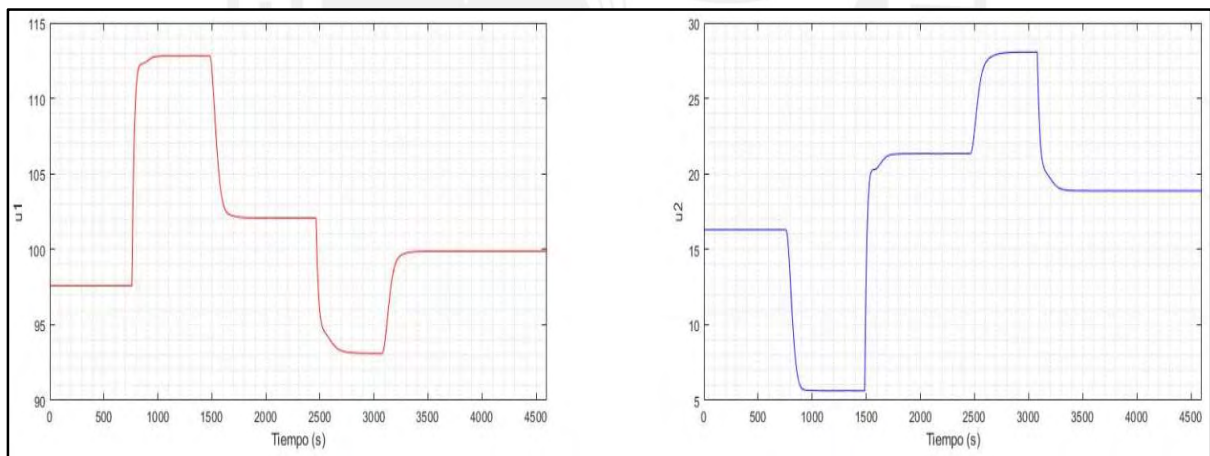


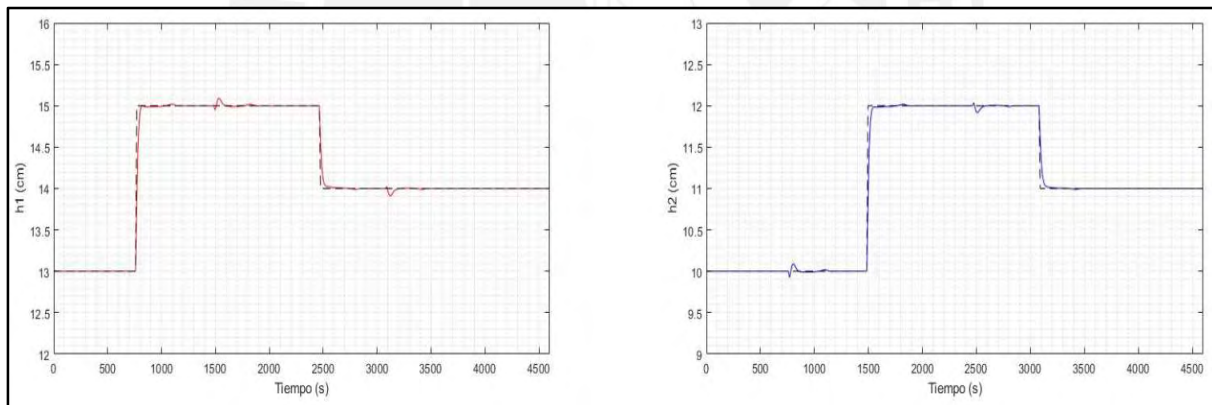
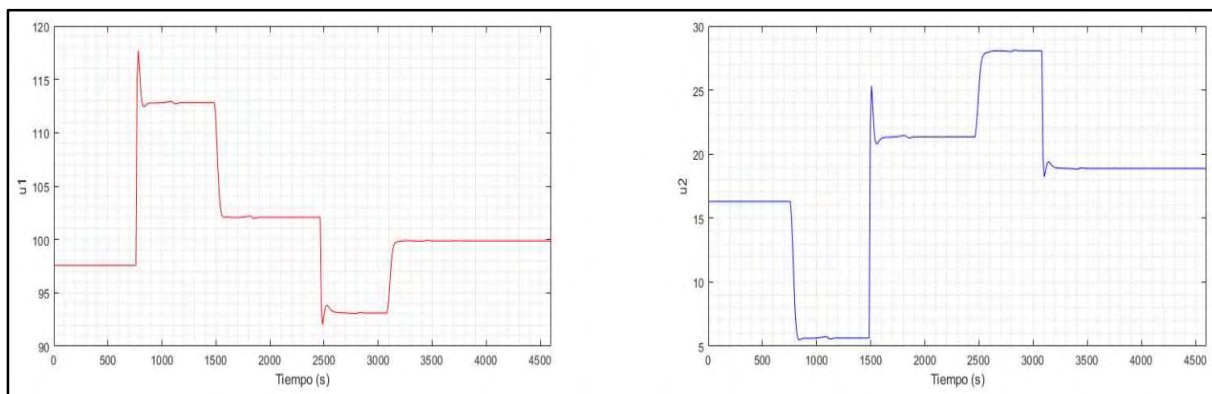
Figura 4.3. Variables de control en el caso nominal y sintonización clásica.

A partir de estos resultados, se puede afirmar que el controlador diseñado es efectivo pues logra llevar las salidas del sistema a los valores indicados por las señales de referencia. Luego, para medir el desempeño del controlador se han utilizado los criterios de desempeño detallados en la Tabla 4.1.

Tabla 4.1. Criterios de desempeño en el caso nominal y sintonización clásica

<i>Criterio</i>	H1	H2
<i>Máximo sobreimpulso</i>	0 %	0 %
<i>Tiempo de establecimiento (2%)</i>	121 s	165 s
<i>Tiempo de subida (90%)</i>	77 s	77 s
<i>ISE</i>	8,802	8,644
<i>Tiempo de ejecución promedio</i>	0,23 ms	

En segundo lugar, de manera similar, se presentan los resultados obtenidos al utilizar los parámetros de sintonización obtenidos con la técnica PSO (ver Tabla 3.4): los niveles de los tanques en la Fig. 4.4 y las variables de control correspondientes en la Fig. 4.5.

**Figura 4.4.** Respuesta del sistema en el caso nominal y sintonización PSO.**Figura 4.5.** Variables de control en el caso nominal y sintonización PSO.

De igual manera, la siguiente tabla resume los criterios de desempeño obtenidos bajo esta técnica de sintonización.

Tabla 4.2. Criterios de desempeño en el caso nominal y sintonización PSO

<i>Criterio</i>	H1	H2
<i>Máximo sobreimpulso</i>	0 %	0 %
<i>Tiempo de establecimiento (2%)</i>	77 s	77 s
<i>Tiempo de subida (90%)</i>	33 s	33 s
<i>ISE</i>	2,147	2,285
<i>Tiempo de ejecución promedio</i>	0,12 ms	

Haciendo una comparación entre los resultados obtenidos para ambas técnicas, se puede concluir que con la sintonización PSO se han obtenido respuestas más rápidas y un mejor seguimiento de las señales de referencia lo cual se evidencia con menores valores de los índices ISE. También se observa que disminuye las variaciones originadas por el acoplamiento existente entre los lazos de control de la planta propios de un proceso multivariable.

Por otro lado, ya que con esta técnica se obtienen menores valores para los horizontes de predicción y control, el tiempo de ejecución promedio es 47% más rápido que con la otra técnica. Ello se debe principalmente a que se reducen las dimensiones de las matrices involucradas en el cálculo de las acciones de control pues estas dependen directamente de los horizontes P y M.

En cuanto a los puntos en contra, se observan ligeras variaciones en las respuestas al llegar al valor estacionario, pero estas no superan el criterio del 2%, y las acciones de control varían de forma más brusca con lo cual se tendría una mayor exigencia en los actuadores. Además, es importante notar que la sintonización mediante PSO no es tan directa como lo es la clásica ya

que se deben ajustar parámetros propios del algoritmo PSO y realizar varias iteraciones antes de obtener el conjunto final de parámetros para el controlador.

4.2.3. Pruebas con perturbaciones

Por otro lado, se realizaron simulaciones incluyendo el efecto de una perturbación en las variables del proceso a fin de conocer el rendimiento del controlador ante esta situación. La perturbación se aplica alrededor del instante 3600 y podría representar una fuga momentánea de 0,5 cm en ambos tanques. Para evaluar los resultados de este caso, solo se aplica el índice ISE pues las respuestas solo varían respecto a las pruebas anteriores a partir del instante en el que se aplica la perturbación. En la Fig. 4.6 se muestran los resultados para la técnica clásica y en la Fig. 4.7, los correspondientes para la técnica basada en PSO.

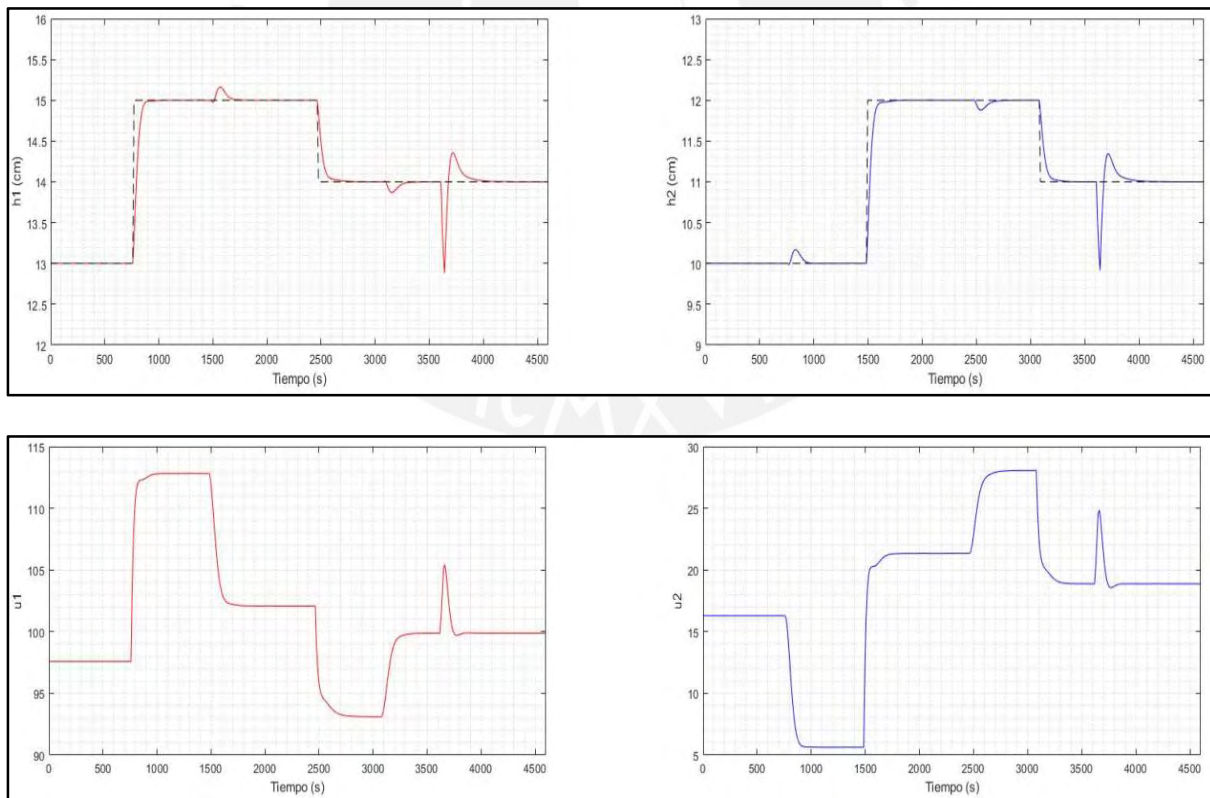


Figura 4.6. Simulación de perturbaciones en el caso nominal y sintonización clásica.

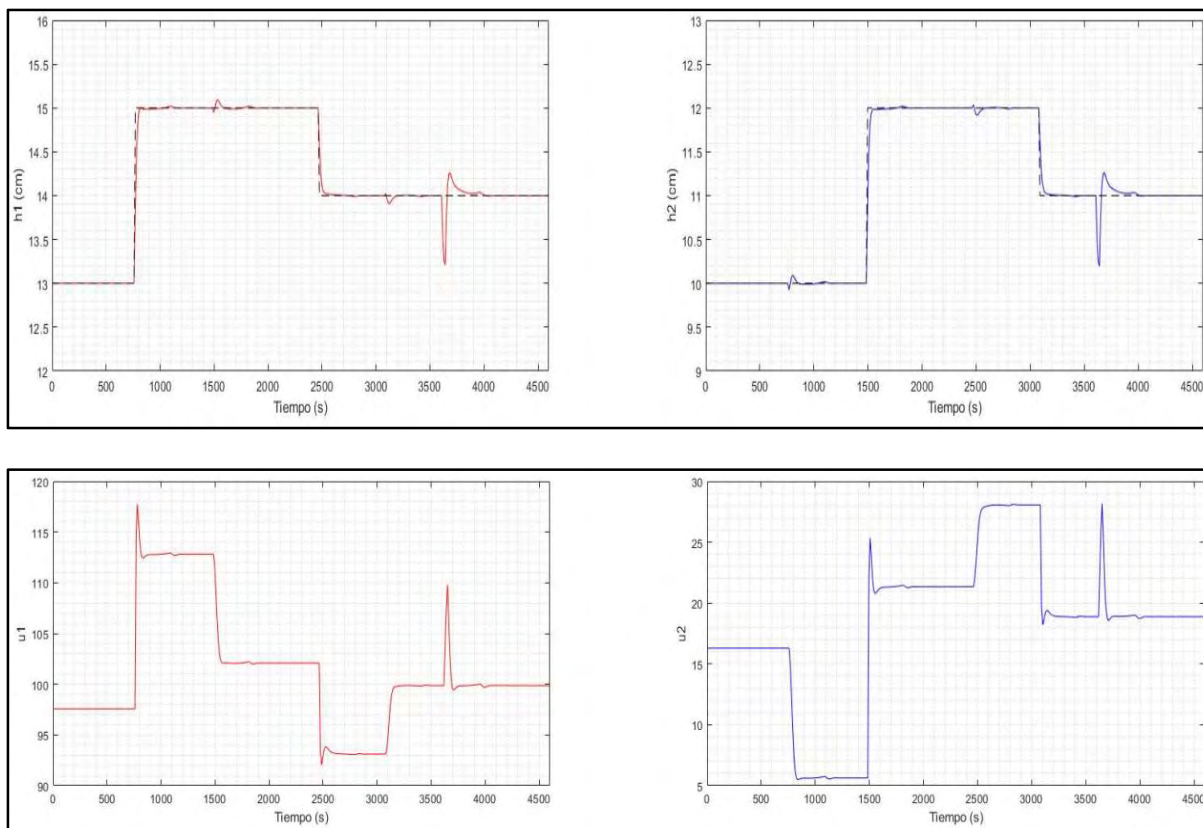


Figura 4.7. Simulación de perturbaciones en el caso nominal y sintonización PSO.

En la siguiente tabla se indican los valores del índice ISE para cada uno de los resultados y en ellos se observa que para la sintonización PSO, los valores son cercanos a los obtenidos previamente en el caso sin perturbaciones y tres veces menores a los valores obtenidos con la sintonización clásica. Por ende, es posible concluir que al utilizar la técnica del PSO también se logra un mayor rechazo ante las perturbaciones.

Tabla 4.3. Índices ISE para el caso nominal con perturbación

	<i>Sintonización clásica</i>	<i>Sintonización PSO</i>
<i>ISE h1</i>	12,506	3,937
<i>ISE h2</i>	11,668	4,128

4.2.4. Pruebas con Incertidumbre en el Modelo

Luego de haber analizado los resultados para el caso en el que se trabaja con el modelo nominal de la planta, ahora se presentan los resultados para el caso en el cual se adiciona a este modelo una variación del 30% en las ganancias de las funciones de transferencia G_{12} y G_{21} para representar la incertidumbre existente en todo modelo. Lo que se busca con ello es evaluar si las técnicas de sintonización permiten variar los parámetros del controlador ante esta situación y seguidamente analizar su desempeño ante este tipo de incertidumbre. Si bien es cierto que la incertidumbre en el modelo no se puede determinar específicamente, lo que sí se puede hacer es brindar un rango estimado para su valor y con ello realizar la nueva sintonización.

Entonces, considerando esta variación en el modelo de la planta, se han vuelto a sintonizar los parámetros con las dos técnicas y los resultados obtenidos se muestran en la siguiente tabla.

Tabla 4.4. Parámetros del controlador para el caso con incertidumbre

	P	M	q1	q2
Sintonización Clásica	50	11	0,09	0,09
Sintonización PSO	28	5	0,003	0,007

Se observa que la técnica de Shridar y Cooper solo modifica los valores de los coeficientes $q1$ y $q2$, no hay mayor variación en los horizontes de predicción o de control. Por otro lado, la segunda técnica sí modifica los cuatro parámetros y entre ellos el que más ha variado es el horizonte de control (M). Ello da indicios de que la sintonización basada en PSO permite un mayor reajuste de los parámetros ante estas situaciones, con lo cual también se espera una mejor respuesta del controlador. Ya que se conocen los nuevos parámetros, se procede a aplicar las mismas pruebas y criterios de desempeño que en el caso con el modelo nominal de la planta.

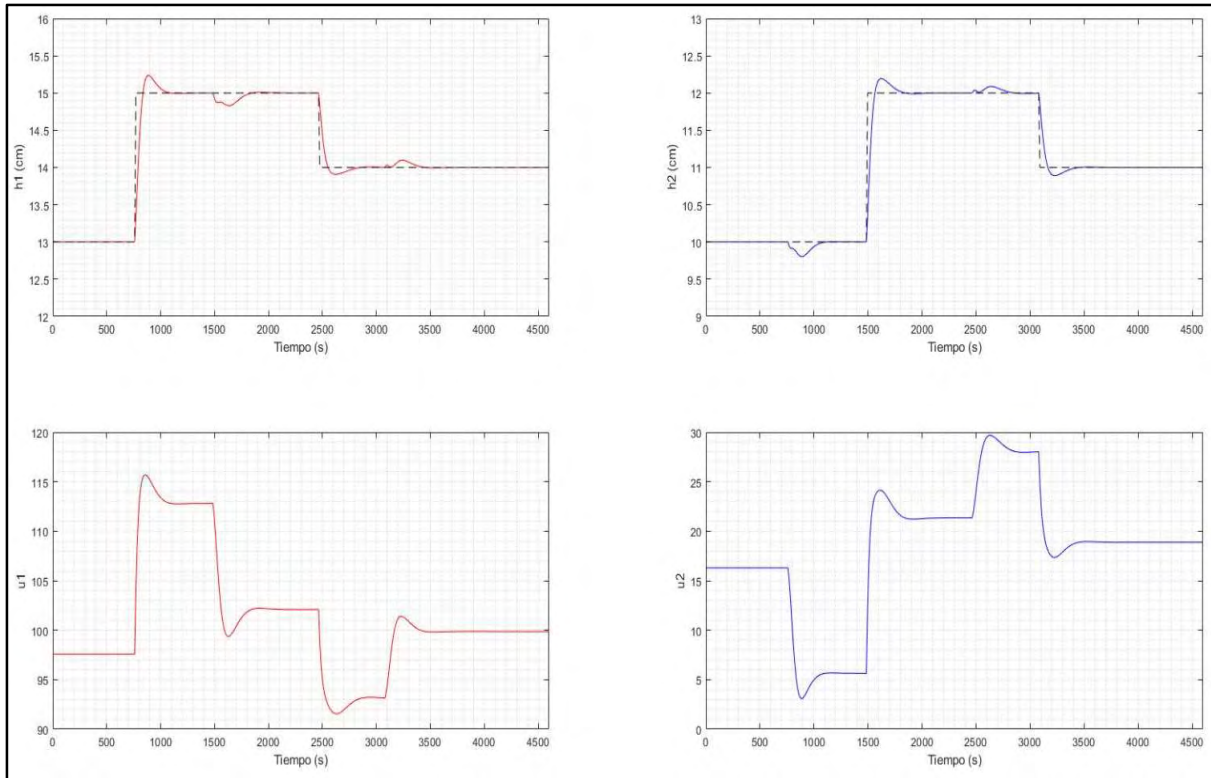


Figura 4.8. Respuesta del sistema en el caso con incertidumbre y sintonización clásica.

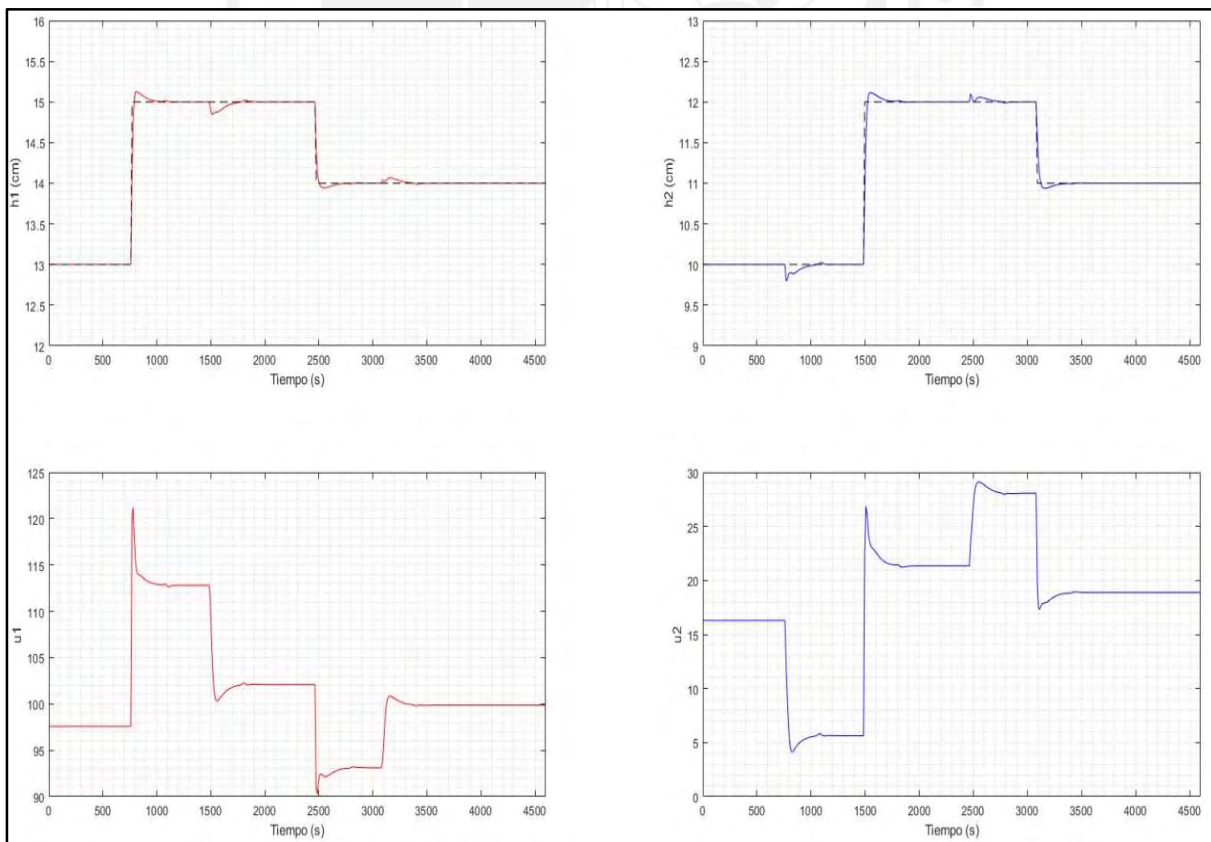


Figura 4.9. Respuesta del sistema en el caso con incertidumbre y sintonización PSO.

Tabla 4.5. Criterios de desempeño para el caso con incertidumbre

<i>Criterio</i>	S. Clásica	S. PSO
<i>Máximo sobreimpulso</i>	1,45 %	0,84 %
<i>Tiempo de establecimiento (2 %)</i>	291,5 s	192,5 s
<i>Tiempo de subida (90 %)</i>	73 s	28 s
<i>ISE h1</i>	8,243	1,445
<i>ISE h2</i>	8,016	2,398

Según los valores de la Tabla 4.5, se observa que en este caso sí hay presencia de sobreimpulsos debido a los tiempos de subida más cortos. De igual manera, las respuestas tardan más en establecerse prácticamente duplicando los valores del caso nominal. A pesar de ello, haciendo la comparación entre las técnicas, otra vez se puede ver que con los parámetros derivados de la sintonización PSO se obtienen mejores valores para todos los criterios de desempeño. Adicionalmente, el efecto del acoplamiento entre los lazos de control también es menor con esta técnica. Todo lo anterior se explica a partir de que la técnica basada en PSO sí permite modificar los valores de los horizontes de predicción y control ante la inclusión de incertidumbre en el modelo pues no depende únicamente de fórmulas generalizadas basadas en los parámetros del modelo como ocurre con la técnica clásica.

4.3. Análisis de resultados

A partir de los resultados obtenidos en las simulaciones, es posible comparar el desempeño del controlador diseñado bajo ambos métodos de sintonización: clásica y PSO.

La respuesta del sistema de control es más rápida con la sintonización PSO lo cual se evidencia a través de menores tiempos de subida y de establecimiento sin presentar sobreimpulsos al igual que en con la otra sintonización. Asimismo, la técnica PSO permitió alcanzar valores de

los índices ISE hasta cuatro veces menores mostrando así un mejor seguimiento de las señales de referencia.

En relación con el tiempo de ejecución promedio del algoritmo de control, existe una diferencia del 47% entre ambas técnicas, siendo la del PSO aquella con mejores resultados. Ello se debe a que esta técnica utiliza menores valores de Horizontes de Predicción y Control, reduciendo así el costo computacional del algoritmo de control.

Por otro lado, al considerar incertidumbre en las simulaciones, la sintonización PSO realiza un mayor ajuste en los parámetros del controlador mejorando la robustez de las variables del proceso bajo este escenario.

Si bien la técnica clásica presenta una ventaja en cuanto a su mayor simplicidad para realizar la sintonización del controlador, los resultados obtenidos en el presente estudio confirman que la técnica de PSO mejora el desempeño del sistema de control en la mayoría de indicadores justificando así su mayor complejidad.

4.4. Propuesta de implementación

4.4.1 Implementación del controlador DMC para la Planta TITO

En esta sección se presenta la propuesta de implementación del controlador DMC diseñado para lo cual se emplearán los siguientes recursos: softwares LabView, Matlab y Simulink.

Se realiza la programación de la rutina del controlador en LabView. Esta rutina está dividida básicamente en tres actividades: la lectura y envío de datos para la interacción con la planta; la ejecución periódica del algoritmo de control DMC y la presentación de las variables en la interfaz.

Al tratarse de una propuesta y no de la implementación real, la planta objeto de estudio se simula con un modelo en Simulink el cual considera la dinámica no lineal de la planta TITO y

los periféricos para la comunicación con el controlador. Por último, también se desarrolla una interfaz gráfica para el monitoreo del sistema de control en funcionamiento.

El objetivo de esta propuesta es mostrar una forma en la cual se puede trasladar el controlador diseñado en un entorno de simulación como Matlab a una plataforma de implementación real como lo es LabView.

4.4.2 Algoritmo de control DMC

El algoritmo de control se desarrolló en Visual C de manera tal que el código generado pueda incluirse después en el programa del controlador en Labview. Este algoritmo tiene como entradas los valores de set point y las variables del proceso medidas por los sensores. Ya que ambas técnicas de sintonización se desarrollan en modo *offline*, no es necesario incluirlas en el algoritmo de control pues este solo necesita los valores finales de los parámetros de control. Por ello, es suficiente con definir como constantes los parámetros del controlador y las ganancias de la ley de control. Estas últimas también son conocidas pues se calcularon previamente durante la etapa de diseño.

En la Fig. 4.10 se presenta el diagrama de flujo que sigue el algoritmo de control. Este será invocado por el controlador cada vez que se necesite recalcular el valor de las acciones de control para seguir la trayectoria deseada. Por ello el algoritmo de control se desarrolla secuencialmente y no incluye ningún lazo de iteración.

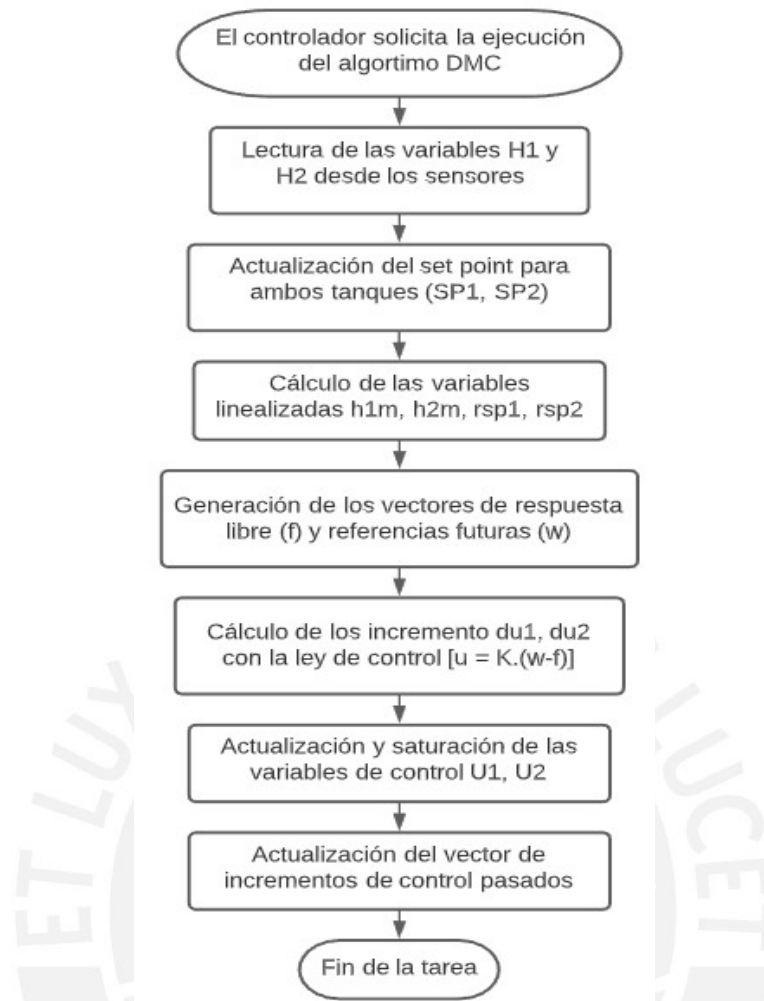


Figura 4.10. Diagrama de flujo del algoritmo de control DMC.

4.4.3 Propuesta de funcionamiento de la planta TITO no lineal

Ya que se trata de una propuesta de implementación, se simula la dinámica no lineal de la planta real a través de un modelo en Simulink. Este modelo incluye un bloque *s-function* en el cual se resuelven las ecuaciones diferenciales que describen el comportamiento de la planta (Ecuaciones 2.29 y 2.30). De esta manera, se suprime la necesidad de trabajar en físico con la planta real y se puede validar el correcto funcionamiento del controlador a través de pruebas con software antes de aplicarlo directamente al proceso real. El modelo completo se muestra en la Fig. 4.11.

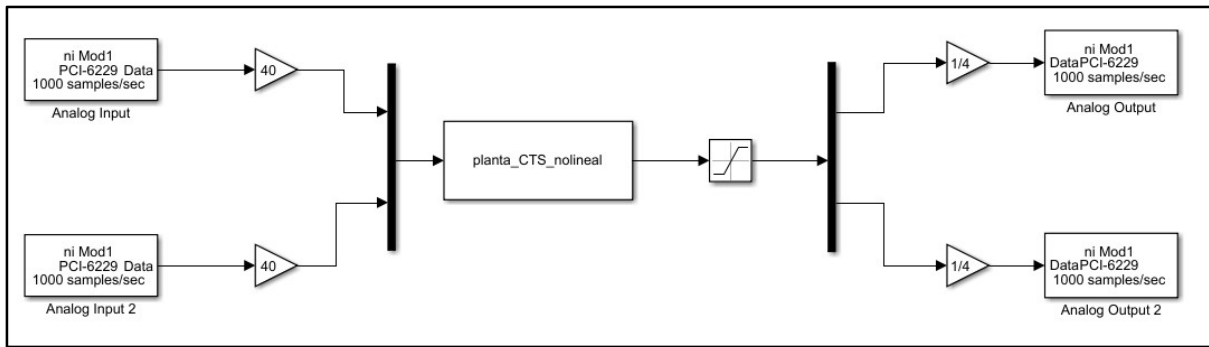


Figura 4.11. Diagrama en Simulink de la planta TITO no lineal.

El diagrama mostrado incluye también bloques de ganancia a la entrada y a la salida los cuales sirven para realizar el escalamiento de las señales correspondientes ya que se propone utilizar la tarjeta de adquisición de datos NI PCI-6229 del fabricante *National Instruments*, la misma que podría utilizarse para adaptar las señales de los actuadores y sensores cuando se implemente el controlador con la planta real. Para la propuesta de implementación, basta con simular la conexión entre las salidas y entradas de estos módulos analógicos de la tarjeta de adquisición de datos.

4.4.4 Interfaz gráfica HMI

Por último, se propone una interfaz gráfica humano-máquina (HMI) que permita a los operadores monitorear el nivel en cada uno de los tanques, así como la trayectoria de las variables del proceso a lo largo de cierto rango de tiempo y los valores de las acciones de control instantáneas correspondientes. También permite modificar los valores de *set point* de acuerdo a las necesidades del usuario, manteniendo el sistema dentro de la vecindad de los puntos de operación.

Se incluye además botones de inicio y parada para seleccionar los momentos en que se requiere la operación del sistema de control. Cuando se presiona el botón de inicio, el controlador inicializa los vectores auxiliares necesarios y ejecuta el algoritmo de control repetidamente según el periodo asignado en el programa. Finalmente, en la Fig.4.12. se muestra un ejemplo del funcionamiento de esta interfaz.

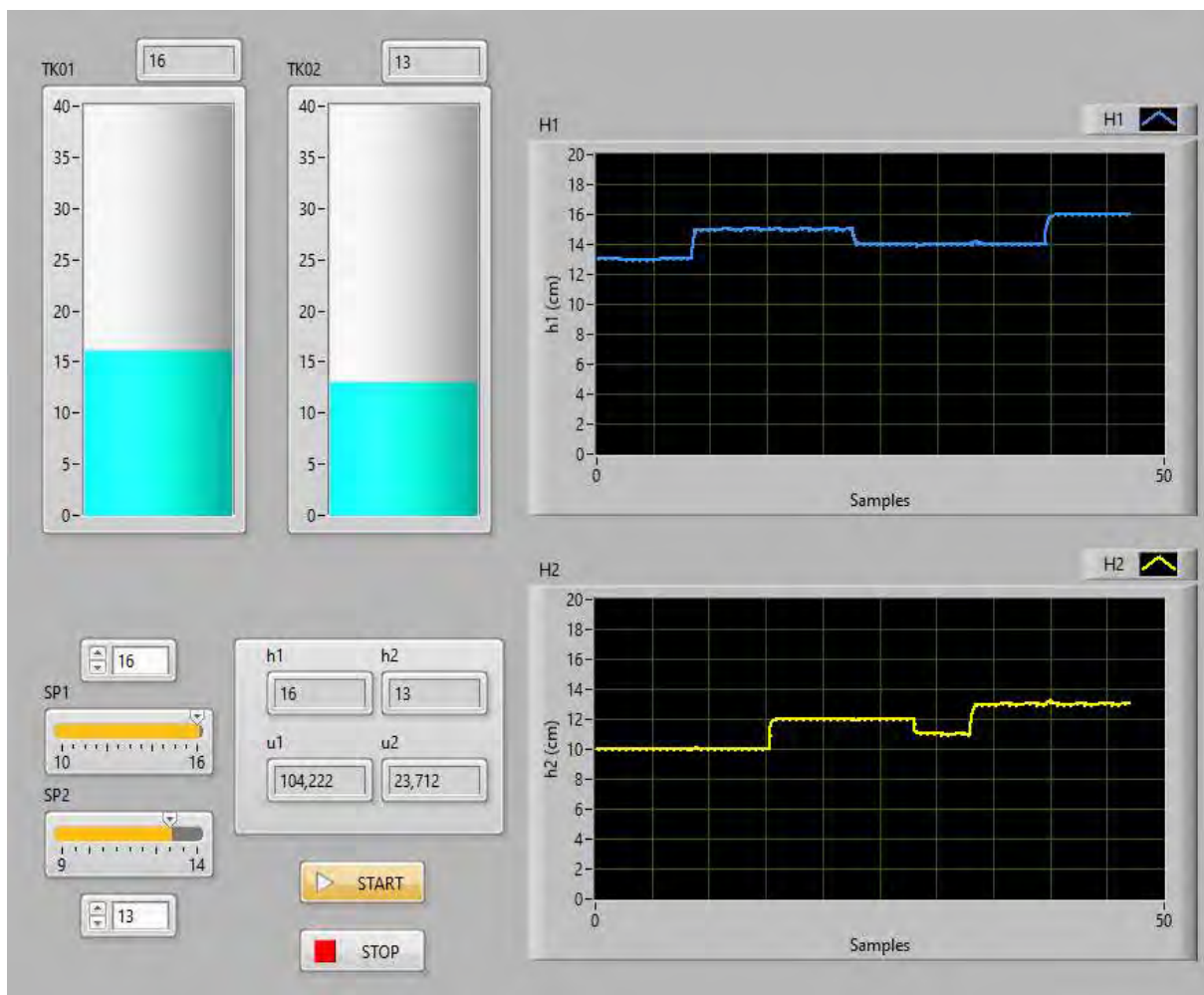


Figura 4.12. Interfaz HMI en Labview para monitoreo.

Como se puede ver, esta interfaz es lo suficientemente simple y amigable con el usuario de tal manera que no se requiere que este tenga un alto grado de conocimiento técnico para operar el sistema de control. También se valida que las gráficas de las variables del proceso H1 y H2 son similares a las que se obtuvieron en las simulaciones al inicio de este capítulo.

CONCLUSIONES

Se presentó la problemática de los procesos hidráulicos con tanques interconectados y se concluye que una planta piloto de dos tanques acoplados es una planta de laboratorio válida para el estudio del diseño de controladores, pues refleja de buena manera el comportamiento y fenómenos que se presentan en procesos industriales reales.

Se estudió el funcionamiento de la planta piloto del Lab. de Control Avanzado de la PUCP y su modelo matemático no lineal. Luego, se realizó la linealización de este modelo alrededor de un punto de operación, a fin de obtener funciones de transferencia para el cálculo del modelo de referencia para el diseño del controlador DMC.

Se diseñó el controlador DMC para la planta piloto de dos tanques acoplados calculando la matriz dinámica del sistema en base al modelo lineal de la planta. De igual manera, se desarrolló el algoritmo de control que permite implementar en software el funcionamiento del controlador.

La sintonización del controlador se realizó aplicando dos métodos distintos: el clásico, desarrollado por Shridar y Cooper; y el segundo basado en el algoritmo de optimización PSO, explicando su procedimiento de sintonización y las consideraciones necesarias para aplicarlos a la planta objeto de estudio.

A partir de los resultados de las simulaciones del controlador sintonizado, se comprueba que el desempeño del mismo mejora la sintonización basada en PSO. Así, con este método se logran tiempos de establecimiento y de subida más rápidos, una reducción en un factor de cuatro en los índices ISE y tiempos de ejecución promedio 47% menores para el algoritmo de control, demostrando el menor costo computacional de esta propuesta.

Finalmente, se presentó la propuesta de implementación del controlador DMC diseñado mediante programación en Labview del sistema de control y utilizando a la vez un modelo no lineal de la planta en Simulink. Mediante las simulaciones se demuestra que es posible realizar la implementación real del controlador diseñado.



RECOMENDACIONES

Se propone utilizar el presente trabajo como punto de partida para realizar el diseño y comparación de controladores predictivos DMC sintonizados utilizando otras técnicas basadas en algoritmos de optimización o redes neuronales artificiales.

Se recomienda aplicar los métodos de sintonización estudiados en un controlador DMC para la configuración de cuatro tanques hidráulicos interconectados para analizar su funcionamiento en una dinámica más compleja y en casos de fase no mínima.

La implementación de este controlador también puede ser realizada utilizando un equipo industrial CompactRIO, para lo cual se puede programar su módulo FPGA en Labview y realizar las pruebas con la planta piloto real utilizando módulos de entradas y salidas analógicas compatibles con el equipo.

BIBLIOGRAFÍA

- [1] J. Anna John, N. E. Jaffar y R. Mary Francis, “Modelling and Control of Coupled Tank Liquid Level System using Backstepping Method.”
- [2] S. Datta, U. M. Nath y C. Dey, “Design and implementation of decentralized IMC-PI controllers for real time coupled tank process,” en *IET Conference Publications*, 2015, vol. 2015, no. CP683, pp. 93–98.
- [3] A. Numsomran, T. Suksri y M. Thumma, “Design of 2-DOF PI controller with decoupling for coupled-tank process,” en *2007 International Conference on Control, Automation and Systems*, 2007, pp. 339–344.
- [4] R. J. Raj y S. N. Deepa, “Modeling and implementation of various controllers used for Quadruple-Tank,” en *Proceedings of IEEE International Conference on Circuit, Power and Computing Technologies, ICCPCT 2016*, 2016.
- [5] 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, mayo 2000.
- [6] A. Numsomran, V. Tipsuwanporn y K. Tirasesth, “Modeling of the Modified Quadruple-Tank Process,” en *Proceedings of the SICE Annual Conference*, 2008, pp. 818–823.
- [7] H. G. Santana, S. D. Coelho y O. D. Almeida, “Application of multivariable PID controllers in a coupled tank system,” en *13th IEEE International Conference on Industry Applications, INDUSCON 2018 - Proceedings*, 2019.
- [8] S. R. Mahapatro, B. Subudhi, S. Ghosh y P. Dworak, “A comparative study of two decoupling control strategies for a coupled tank system,” en *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, pp. 3447–3451, 2017.
- [9] S. Kangwanrat, V. Tipsuwannaporn y A. Numsomran, “Design of PI controller using MRAC techniques for coupled-tanks process,” en *ICCAS 2010 - Int. Conf. Control. Autom. Syst.*, vol. 1, pp. 485–490, 2010.
- [10] E. F. Camacho y C. Bordons, “Introduction to model predictive control,” en *Model Predictive Control*, 2ª edición, Springer International Publishing, 2007, pp. 1–11.
- [11] M. U. Khalid y M. B. Kadri, “Liquid level control of nonlinear Coupled Tanks System using linear model predictive control,” en *Proceedings - 2012 International Conference on Emerging Technologies, ICET 2012*, 2012, pp. 31–35.
- [12] M. Awais, H. Alam y A. Muhammad, “Model Predictive Control Design for coupled Tank System,” en *The International conference Quality Time 2017*, 2018.
- [13] K. O. Owa, S. K. Sharma y R. Sutton, “Optimised multivariable nonlinear predictive control for coupled tank applications,” en *IET Conference on Control and Automation 2013: Uniting Problems and Solutions*, 2013.
- [14] E. F. Camacho y C. Bordons, “Model predictive controllers,” in *Model Predictive Control*, 2ª edición, Springer International Publishing, 2007, pp. 13–30.

- [15] A. X. Pan, G. Y. Wang y D. Huang, "Control of permanent magnet synchronous motor based on GPC," en *2008 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2008*, 2008.
- [16] P. Guo, "Research of multivariable GPC and its application in thermal power plant," en *First International Conference on Innovative Computing, Information and Control 2006, ICICIC'06*, 2006, pp. 35–38.
- [17] Y. Cheng, "A novel DMC-like implementation of GPC," in *Proceedings - 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer, MEC 2013*, 2013, pp. 362–366.
- [18] K. Kramer y H. Unbehauen, "Aspects in selecting predictive adaptive control algorithms," en *Proceedings of the American Control Conference*, 1992, vol. 3, pp. 2396–2401.
- [19] E. F. Camacho y C. Bordons, "Multivariable Model Predictive Control," en *Model Predictive Control*, 2ª edición, Springer International Publishing, 2007, pp. 127–176.
- [20] R. Shridhar y D. J. Cooper. "A Tuning Strategy for Unconstrained Multivariable Model Predictive Control," en *Ind. Eng. Chem. Res.*, 1998, vol. 37, no. 10, pp. 4003-4016.
- [21] R. C. Eberhart y J. Kennedy, "A New Optimizer Using Particle Swarm Theory," *Proc. the Sixth Int. Symposium on Micro Machine and Human Science*, Nagoya, Japan, 1995.
- [22] Kai Han, Jun Zhao y Jixin Qian, "A Novel Robust Tuning Strategy for Model Predictive Control," en *2006 6th World Congress on Intelligent Control and Automation*, Dalian, 2006, pp. 6406-6410.
- [23] Prमित Biswas, "Particle Swarm Optimization (PSO)", en *MATLAB Central File Exchange*, 2020.
[https://www.mathworks.com/matlabcentral/fileexchange/43541-particle-swarm - optimization-pso](https://www.mathworks.com/matlabcentral/fileexchange/43541-particle-swarm-optimization-pso)

ANEXOS

A1. Programa Principal - MATLAB

```

1  close all; clc; clear;
2
3  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Matriz Dinámica Sistema TITO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5
6  %%Parámetros de la planta TITO
7  H=40; %[cm]
8  A=754.76; a1=1.266; ax=1.6; %[cm2]
9  g=9.8; %[m/s2] Aceleración gravedad
10 k1=3.33; k2=3.35; %[cm3/Vs] Ganancia Bombas
11
12 %%Puntos de operación - linealización
13 ho1 = 0.13; %[m]
14 ho2 = 0.1; %[m]
15
16 %%Constantes de Tiempo: 1/T1, 1/T2, 1/Tx
17 T1 = (A/a1)*(sqrt((2*ho1)/g));
18 T2 = (A/a1)*(sqrt((2*ho2)/g));
19 Tx = (A/ax)*(sqrt((2*(ho1-ho2)/g)));
20
21 %%Ecuación Característica Planta TITO - Denominador
22 den = [1 (T1*Tx+T2*Tx+2*T1*T2)/(Tx*T2*T1) (T1+T2+Tx)/(Tx*T2*T1)];
23
24 %%Numeradores
25 num11 = (k1/A)*[1 1/Tx+1/T2];
26 num12 = 1*k2/(A*Tx); %puede agregar incertidumbre
27 num21 = 1*k1/(A*Tx); %puede agregar incertidumbre
28 num22 = (k2/A)*[1 1/Tx+1/T1];
29
30 %%Funciones de Transferencia
31 G11s = tf(num11,den);
32 G12s = tf(num12,den);
33 G21s = tf(num21,den);
34 G22s = tf(num22,den);
35
36 %%Discretización de la Matriz Transferencia
37
38 Ts = 11; %Tiempo de Muestreo
39 N_m = 50; %Horizonte de Modelo
40
41 N = 29; %Horizonte de Predicción
42 M = 2; %Horizonte de Control
43
44 q1 = 0.008; %factores ponderación q para las acciones de control
45 q2 = 0.009;
46
47 %%Numeradores y Denominador discretos
48 [num11d,den11d] = c2dm(num11,den,Ts,'zoh');
49 [num12d,den12d] = c2dm(num12,den,Ts,'zoh');

```



```

50 [num21d,den21d] = c2dm(num21,den,Ts,'zoh');
51 [num22d,den22d] = c2dm(num22,den,Ts,'zoh');

52
53 %Coeficientes del modelo de respuesta al escalón
54 g11 = plant_coef(num11d,den11d,N_m);
55 g12 = plant_coef(num12d,den11d,N_m);
56 g21 = plant_coef(num21d,den11d,N_m);
57 g22 = plant_coef(num22d,den11d,N_m);
58
59
60 %%Formación de la matriz dinámica G dimensiones: 2N * 2M %%
61 %Submatrices G - sistemas SISO
62
63 G11 = zeros(N,M);
64 G12 = zeros(N,M);
65 G21 = zeros(N,M);
66 G22 = zeros(N,M);
67
68 for i=1:N
69     k=i;
70
71     if i>M
72         lim=M;
73     else
74         lim=i;
75     end
76
77     for j=1:lim
78         G11(i,j) = g11(k);
79         G12(i,j) = g12(k);
80         G21(i,j) = g21(k);
81         G22(i,j) = g22(k);
82         k=k-1;
83     end
84 end
85 %%%
86
87 %%% Matriz Dinámica "G" del Sistema TITO %%%
88 G = [G11 G12; G21 G22];
89 %%%
90
91
92
93 %%%
94 %%% Controlador DMC %%%
95 %%%
96
97 tsim = 4620; %tiempo de simulación
98 numite = round(tsim/Ts); %número de iteraciones
99
100 %Puntos de operación del sistema
101 h10 = 13;      h20 = 10;      %[cm]
102 u10 = 97.58;   u20 = 16.29;   %[V]
103
104 %Definición de vectores iniciales
105 dfpas_u1 = zeros(1,N);      % variaciones pasadas de u1
106 dfpas_u2 = zeros(1,N);      % variaciones pasadas de u2

```

```

107
108 du1_index = zeros(1,numite);% vectores para cálculo indice EISTWE
109 du2_index = zeros(1,numite);

110
111 %Vectores para graficar la respuesta
112 U1 = zeros(1,numite); U2 = zeros(1,numite);
113 H1 = zeros(1,numite); H2 = zeros(1,numite);
114 SP1 = zeros(1,numite); SP2 = zeros(1,numite);
115
116 %Definición Vectores Set Point para las salidas linealizadas (h1m,h2m)
117 r1_sp = 1*ones(1,numite);
118 r1_sp(1:70)=0; r1_sp(71:225)=2;r1_sp(226:309)=1;r1_sp(310:end)=1;
119
120 r2_sp = 0.5*ones(1,numite);
121 r2_sp(1:136)=0; r2_sp(137:281)=2; r2_sp(282:end)=1;
122
123 %Definición de Vector Peturbación para simulación
124 pert = zeros(1,numite);
125 pert(330:332) = 0; %intervalo en donde se presenta la perturbacion
126
127 %acciones control linealiz. iniciales [um = uxreal-uX0]
128 u1m = 0; u2m = 0;
129
130 %variables salida linealiz. iniciales [hm = hxreal-hX0]
131 h1m = 0; h2m = 0;
132
133
134 %%%% Vector de Ganancias K --> u = K(w-f)%%%
135 R = [1*eye(N) zeros(N); zeros(N) 1*eye(N)];
136 Q = [q1*eye(M) zeros(M); zeros(M) q2*eye(M)];
137
138 K = inv((G'*R*G)+Q)*G'*R;
139
140 %Ganancias relevantes para el control
141 K1 = K(1,:); % primera fila de K corresponde a du1(t)
142 K2 = K(M+1,:); % fila M+1 de K corresponde a du2(t)
143 %%%%
144
145 %Condiciones Iniciales Salidas igual Pto. Operación
146 h1= h10 + h1m;
147 h2= h20 + h2m;
148
149
150 %Inicio de simulación y control%%
151 for j=1:numite
152
153     tic
154
155     %%%% Cálculo de la respuesta libre
156     f1 = zeros(1,N);
157     f2 = zeros(1,N);
158
159     for i=1:N
160         for rep=1:N
161             if (rep+i)>N
162                 rep_i=N;
163             else

```

```

164         rep_i=rep+i;
165         end

166         f_g11(rep) = g11(rep_i) - g11(rep);
167         f_g12(rep) = g12(rep_i) - g12(rep);
168         f_g21(rep) = g21(rep_i) - g21(rep);
169         f_g22(rep) = g22(rep_i) - g22(rep);
170     end
171
172     f1(i) = h1m + f_g11*dfpas_u1' + f_g12*dfpas_u2';
173     f2(i) = h2m + f_g21*dfpas_u1' + f_g22*dfpas_u2';
174 end
175
176 f = [f1 f2]; % Vector respuesta libre f (1x2N)
177 %%%
178
179 %%% Cálculo de referencias futuras
180 w1 = r1_sp(j)*ones(1,N);
181 w2 = r2_sp(j)*ones(1,N);
182
183 w = [w1 w2]; %vector w (1x2N)
184 %%%
185
186 %%% Cálculo de nuevas variaciones du1 y du2
187 du1 = K1*(w-f)';
188 du2 = K2*(w-f)';
189 %%%
190
191 %Para el índice EISTWE
192 du1_index(j) = du1;
193 du2_index(j) = du2;
194
195 %%% Actualización de variables de control linealizadas
196 u1m = u1m + du1;
197 u2m = u2m + du2;
198
199 %Saturador para evitar valores de u1,u2 < 0 (Voltajes positivos)
200 if u1m<-u10
201     u1m=-u10;
202 end
203
204 if u2m<-u20
205     u2m=-u20;
206 end
207 %%%
208
209 %%% Acciones de control
210 u1 = u10 + u1m;
211 u2 = u20 + u2m;
212 %%%
213
214 dfpas_u1 = [du1 dfpas_u1(1:end-1)];% actualizar el vector de
215 dfpas_u2 = [du2 dfpas_u2(1:end-1)];% du pasados (du -> du(-1))
216
217
218

```

```

219     %%%% Cálculo de variables de salida reales
220
221     %Modelo no lineal del sistema TITO
222     dh1 = Ts*(k1*(u1) - a1*sqrt(2*981*h1) - ax*sign(h1-h2)*sqrt(2*981*abs(h1-
223     h2)))/A;
224     dh2 = Ts*(k2*(u2) - a1*sqrt(2*981*h2) + ax*sign(h1-h2)*sqrt(2*981*abs(h1-
225     h2)))/A;
226
227     h1m = dh1 + h1m + pert(j);
228     h2m = dh2 + h2m + pert(j);
229
230     h1 = h10 + h1m;
231     h2 = h20 + h2m;
232     %%%%
233     %%%% Almacenamiento de valores para gráficas
234     u1(j) = u1;
235     u2(j) = u2;
236
237     SP1(j) = h10 + r1_sp(j);% + pert(j);
238     SP2(j) = h20 + r2_sp(j);% + pert(j);
239
240     H1(j) = h1;
241     H2(j) = h2;
242     %%%%
243     duration(j) = toc;
244 end
245
246 % PLOTEO DE LA RESPUESTA
247 t=0:Ts:tsim-Ts;
248
249 figure(1);
250 subplot(2,2,1);
251 plot(t,H1,'r-',t,SP1,'k--');
252 xlabel('Tiempo (s)');ylabel('h1 (cm)');
253 xlim([0 4600]);
254 ylim([12 16]);
255 grid minor;
256
257 subplot(2,2,2);
258 plot(t,H2,'b-',t,SP2,'k--');
259 xlabel('Tiempo (s)');ylabel('h2 (cm)');
260 xlim([0 4600]);
261 ylim([9 13]);
262 grid minor;
263
264 subplot(2,2,3);
265 plot(t,U1,'r-');
266 xlabel('Tiempo (s)');ylabel('u1');
267 xlim([0 4600]);
268 grid minor;
269
270 subplot(2,2,4);
271 plot(t,U2,'b-');

```

```

272 xlabel('Tiempo (s)');ylabel('u2');
273 xlim([0 4600]);
274 grid minor;

275
276 %%% Cálculo del índice EISTWE final %%%
277 %valores desde inicio de escalones de SP
278 qindex = 0.5; %q
279 delt = 0.1;
280
281 ct1_mov1 = qindex*(du1_index(71:end)*du1_index(71:end)');
282 ct1_mov2 = qindex*(du2_index(71:end)*du2_index(71:end)');
283
284 error1x = 0;
285 error2x = 0;
286
287 zx=0;
288 for jj=71:numite
289     zx = (SP1(jj)-H1(jj))^2;
290     zx = jj*zx;
291     error1x = error1x + zx;
292 end
293
294 zx=0;
295 for jj=71:numite
296     zx = (SP2(jj)-H2(jj))^2;
297     zx = jj*zx;
298     error2x = error2x + zx;
299 end
300
301
302 %Para el multiplicador del error%
303 a1_index = max(H1(71:end));
304 a2_index = max(H2(71:end));
305
306 b1_index = min(H1(71:end));
307 b2_index = min(H2(71:end));
308
309 mult1 = (1+a1_index+b1_index)/SP1(72);
310 mult2 = (1+a2_index+b2_index)/SP2(72);
311
312 %Indicador de desempeño
313 EISTWE = delt*N*M + mult1*(error1x + ct1_mov1)+ mult2*(error2x + ct1_mov2);
314 %%%
315
316 %%% Cálculo de índice ISE
317 ER1 = SP1-H1;
318 ER2 = SP2-H2;
319
320 ISE1 = ER1*ER1';
321 ISE2 = ER2*ER2';
322 %%%

```

A2. Función *Particle_Swarm_Optimization.m* - Primit Biswas (2022).

```

% Función en MATLAB para evaluación del algoritmo PSO

% Particle_Swarm_Optimization
% Author: Primit Biswas (2022). Particle Swarm Optimization (PSO)
% https://www.mathworks.com/matlabcentral/fileexchange/43541-particle-swarm-optimization-psy,
% MATLAB Central File Exchange. Retrieved June 11, 2022.

% INPUT VARIABLES
% Bird_in_swarm=Number of particle=agents=candidate
% Number_of_quality_in_Bird=Number of Variable
%
% MinMaxRange: jx2 matrix; jth row contains minimum and maximum values of the jth variable
% say you have a variable N1
% which can have maximum value M1 and minimum value m1
% then your matrix will be [m1 M1]
% for more:
% [m1 M1; m2 M2; mj Mj]
%
% Food_availability=Objective function with one input variable (for more than one variable
you may use array)
% example for two variable
% function f = funfunc(array)
%     a=array(1);
%     b=array(2);
%     f = a+b ;
% end
% Food_availability is a string, for above example : 'funfunc'
%
% availability_type is string 'min' or 'max' to check depending upon need to minimize or
maximize the Food_availability
% velocity_clamping_factor (normally 2)
% cognitive_constant=c1=individual learning rate (normally 2)
% social_constant=c2=social parameter (normally 2)
% normally c1+c2>=4
%
% Inertia_weight=At the beginning of the search procedure, diversification is heavily
weighted, while intensification is heavily weighted at the end of the search procedure.
% Min_Inertia_weight=min of inertia weight (normally 0.4)
% Max_Inertia_weight=max of inertia weight (normally 0.9)
% max_iteration=how many times readjust the position of the flock/swarm of birds its quest
for food
%
% OUTPUT VARIABLE
% optimised_parameters : Optimal parameters

% EJEMPLO DE APLICACIÓN DE LA FUNCIÓN Particle_Swarm_Optimization
% Particle_Swarm_Optimization(30,4,[11 53; 1 10;0.001 0.01; 0.001
0.01],'DMC','min',2,2,2,0.4,0.9,100)

```

```

function [ optimised_parameters ] = Particle_Swarm_Optimization (Bird_in_swarm,
Number_of_quality_in_Bird, MinMaxRange, Food_availability, availability_type,
velocity_clamping_factor, cognitive_constant, social_constant, Min_Inertia_weight,
Max_Inertia_weight, max_iteration)
%{
    Checking all functions are present
%}
if (exist ('MinMaxCheck.m')==0)
    clc;
    fprintf ('Please download the following submission from: <a
href="http://www.mathworks.com/matlabcentral/fileexchange/43251-bound-values-of-an-
array">MATLAB File Exchange (Click here to open link)</a> \ndownload code by clicking
"Download Submission" button \nthen extract and put MinMaxCheck.m in current directory and
try again\n');
    return;
end
%{
    Checking all parameteres are entered
%}
if nargin < 11
    error('Missing input parameter(s)!!')
end

%{
    universalize availability type
%}
availability_type=lower(availability_type(1:3));

%{
    Checking for proper boundary values and entered Matrix
%}
[row,col]=size(MinMaxRange);
if row~=Number_of_quality_in_Bird || col~=2
    error('Not a proper MinMaxRange Matrix')
end
for i=1:Number_of_quality_in_Bird
    if MinMaxRange(i,1)>=MinMaxRange(i,2)
        error('Minimum value greater than Maximum value!!!')
    end
end

%{
    counter to display % of completion
%}
N=Bird_in_swarm*max_iteration;
q=0;

%{
    distinguishing min and max range
%}
bird_min_range=MinMaxRange(:,1);
bird_max_range=MinMaxRange(:,2);

%{
%}

```

```

format long;
for i=1:Number_of_quality_in_Bird
    bird(:,i)=bird_min_range(i)+(bird_max_range(i)-bird_min_range(i))*rand(Bird_in_swarm,1);
end

Vmax=bird_max_range*velocity_clamping_factor;
Vmin=-Vmax;

for i=1:Number_of_quality_in_Bird
    velocity(:,i)=Vmin(i)+(Vmax(i)-Vmin(i))*rand(Bird_in_swarm,1);
end

for itr=1:max_iteration
    fprintf('Completed %d %% ...', uint8(q*100/N ))

    for p=1:Bird_in_swarm
        parameter=bird(p,:,itr);
        availability(p,itr)=feval(Food_availability,parameter);

        switch availability_type
            case 'min'
                format long;
                [pBest_availability,index]=min(availability(p,:));
                pBest=bird(p,:,index);

                if(p==1 && itr==1)
                    gBest=pBest;
                    gBest_availability=pBest_availability;
                elseif availability(p,itr)<gBest_availability
                    gBest_availability=availability(p,itr);
                    gBest=bird(p,:,itr);
                end

            case 'max'
                format long;
                [pBest_availability,index]=max(availability(p,:));
                pBest=bird(p,:,index);

                if(p==1 && itr==1)
                    gBest=pBest;
                    gBest_availability=pBest_availability;
                elseif availability(p,itr)>gBest_availability
                    gBest_availability=availability(p,itr);
                    gBest=bird(p,:,itr);
                end

            otherwise
                error('availability_type mismatch')
        end

        w(itr)=((max_iteration - itr)*(Max_Inertia_weight-Min_Inertia_weight))
            /(max_iteration-1) + Min_Inertia_weight;
    end
end

```



```

velocity(p,:,itr+1)=w(itr)*velocity(p,:,itr) +
social_constant*rand(1,Number_of_quality_in_Bird).*(gBest-bird(p,:,itr)) +
cognitive_constant*rand(1,Number_of_quality_in_Bird).*(pBest-bird(p,:,itr));

velocity(p,:,itr+1)=MinMaxCheck(vmin, vmax, velocity(p,:,itr+1));

bird(p,:,itr+1)= bird(p,:,itr) + velocity(p,:,itr+1);
bird(p,:,itr+1)=MinMaxCheck(bird_min_range, bird_max_range, bird(p,:,itr+1));
q=q+1;
end

clc;
end
optimised_parameters=gBest;

```

A3. Función *DMC.m*

```

1
2 %% Función DMC %%
3 % Función utilizada por el algoritmo PSO para la sintonización del
4 % controlador DMC
5
6 % INPUT: Vector sint = [N, M, q1, q2]
7 % OUTPUT: Valor del indicador EISTWE resultante
8
9 function f=DMC(sint)
10
11 %Parámetros de la planta
12 H=40; %[cm]
13 A=754.76; a1=1.266; ax=1.6; %[cm2]
14 g=9.8; %[m/s2] Aceleración gravedad
15 k1=3.33; k2=3.35; %[cm3/Vs] Ganancia Bombas
16
17 %Puntos de operación
18 ho1 = 0.13; %[m]
19 ho2 = 0.1; %[m]
20
21 %Constantes de Tiempo 1/T1, 1/T2, 1/Tx
22 T1 = (A/a1)*(sqrt((2*ho1)/g));
23 T2 = (A/a1)*(sqrt((2*ho2)/g));
24 Tx = (A/ax)*(sqrt((2*(ho1-ho2)/g)));
25
26 %Ecuación Característica - Denominador
27 den = [1 (T1*Tx+T2*Tx+2*T1*T2)/(Tx*T2*T1) (T1+T2+Tx)/(Tx*T2*T1)];
28
29 %Numeradores
30 num11 = (k1/A)*[1 1/Tx+1/T2];
31 num12 = 0.7*k2/(A*Tx);
32 num21 = 1.3*k1/(A*Tx);
33 num22 = (k2/A)*[1 1/Tx+1/T1];
34

```

```

35  %Funciones de Transferencia
36  G11s = tf(num11,den);
37  G12s = tf(num12,den);
38  G21s = tf(num21,den);
39  G22s = tf(num22,den);
40
41  %Discretizando la Matriz Transferencia
42
43  Ts = 11; %Tiempo Muestreo
44  N_m = 50; %Horizonte Modelo
45
46  N = round(sint(1)); %Horizonte de Predicción
47  M = round(sint(2)); %Horizonte de Control
48
49  [num11d,den11d] = c2dm(num11,den,Ts,'zoh');
50  [num12d,den12d] = c2dm(num12,den,Ts,'zoh');
51  [num21d,den21d] = c2dm(num21,den,Ts,'zoh');
52  [num22d,den22d] = c2dm(num22,den,Ts,'zoh');
53
54  %Coeficientes del modelo de respuesta al escalón
55  g11 = plant_coef(num11d,den11d,N_m);
56  g12 = plant_coef(num12d,den12d,N_m);
57  g21 = plant_coef(num21d,den21d,N_m);
58  g22 = plant_coef(num22d,den22d,N_m);
59
60
61  %Formación de la matriz dinámica G (2N*2M)
62  %Submatrices
63  G11 = zeros(N,M);
64  G12 = zeros(N,M);
65  G21 = zeros(N,M);
66  G22 = zeros(N,M);
67
68  for i=1:N
69      k=i;
70
71      if i>M
72          lim=M;
73      else
74          lim=i;
75      end
76
77      for j=1:lim
78          G11(i,j) = g11(k);
79          G12(i,j) = g12(k);
80          G21(i,j) = g21(k);
81          G22(i,j) = g22(k);
82          k=k-1;
83      end
84  end
85
86  %%Matriz Dinámica del Sistema TITO%%
87  G = [G11 G12; G21 G22];
88

```

```

89  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Controlador DMC %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
90
91  tsim = 1496; %tiempo de simulación
92  numite = round(tsim/Ts); %número de iteraciones
93
94  q1 = sint(3); %factores ponderación para las acciones de control
95  q2 = sint(4);
96
97  %Vectores iniciales
98  dfpas_u1 = zeros(1,N); % variaciones u1 pasadas
99  dfpas_u2 = zeros(1,N); % variaciones u2 pasadas
100 du1_index = zeros(1,numite);
101 du2_index = zeros(1,numite);
102
103 u1m = 0; u2m = 0; % acciones de control iniciales
104 h1m = 0; h2m = 0; %variables de salida medidas iniciales
105
106 %Puntos de operación
107 h10 = 13; h20 = 10; %[cm]
108 u10 = 97.58; u20 = 16.29; %
109 %
110
111 %Vectores para la gráfica de la respuesta
112 U1 = zeros(1,numite); U2 = zeros(1,numite);
113 H1 = zeros(1,numite); H2 = zeros(1,numite);
114 SP1 = zeros(1,numite); SP2 = zeros(1,numite);
115
116 %Vector de Ganancias K --> u = K(w-f)
117 R = [1*eye(N) zeros(N); zeros(N) 1*eye(N)];
118 Q = [q1*eye(M) zeros(M); zeros(M) q2*eye(M)];
119
120 K = inv((G'*R*G)+Q)*G'*R; %K = ((G'RG+Q)^-1)*G'*R
121
122 %Vectores de ganancia relevantes
123 K1 = K(1,:); %primera fila de K corresponde a du1(t)
124 K2 = K(M+1,:); %fila M+1 de K corresponde a du2(t)
125
126 %Vectores de set point para las salidas
127 r1_sp = 1*ones(1,numite);
128 r1_sp(1:50)=0; r1_sp(51:end)=2;
129 r2_sp = 0.5*ones(1,numite);
130 r2_sp(1:80)=0; r2_sp(81:end)=2;
131
132 h1 = h10+h1m;
133 h2 = h20+h2m;
134
135 for j=1:numite
136
137     %Calcular respuesta libre
138     f1 = zeros(1,N);
139     f2 = zeros(1,N);
140
141     for i=1:N
142         for rep=1:N
143             if (rep+i)>N
144                 rep_i=N;
145             else
146                 rep_i=rep+i;

```

```

147         end
148         f_g11(rep) = g11(rep_i) - g11(rep);
149         f_g12(rep) = g12(rep_i) - g12(rep);
150         f_g21(rep) = g21(rep_i) - g21(rep);
151         f_g22(rep) = g22(rep_i) - g22(rep);
152     end
153
154     f1(i) = h1m + f_g11*dfpas_u1' + f_g12*dfpas_u2';
155     f2(i) = h2m + f_g21*dfpas_u1' + f_g22*dfpas_u2';
156 end
157
158 f = [f1 f2]; %vector respuesta libre (1x2N)
159
160 % Calcular de referencias futuras
161 w1 = r1_sp(j)*ones(1,N);
162 w2 = r2_sp(j)*ones(1,N);
163
164 w = [w1 w2]; %vector w (1x2N)
165
166 % Calcular nuevas variaciones du1 y du2
167 du1 = k1*(w-f)';
168 du2 = k2*(w-f)';
169
170 du1_index(j) = du1;
171 du2_index(j) = du2;
172
173 u1m = u1m + du1; %actualizar variables de control
174 u2m = u2m + du2;
175
176 if u1m < -u10
177     u1m = -u10;
178 end
179
180 if u2m < -u20
181     u2m = -u20;
182 end
183
184 u1 = u10 + u1m; %calcular las acciones de control reales
185 u2 = u20 + u2m;
186
187 dfpas_u1 = [du1 dfpas_u1(1:end-1)]; %actualizar el vector de valores
188 dfpas_u2 = [du2 dfpas_u2(1:end-1)]; %du pasados (du -> du(-1))
189
190 % Calcular h1m, h2m
191
192 dh1 = Ts*(k1*(u1) - a1*sqrt(2*981*h1) - ax*sign(h1-h2)*sqrt(2*981*abs(h1-h2)))/A;
193 dh2 = Ts*(k2*(u2) - a1*sqrt(2*981*h2) + ax*sign(h1-h2)*sqrt(2*981*abs(h1-h2)))/A;
194
195 h1m = dh1 + h1m;
196 h2m = dh2 + h2m;
197 % calcular variables de salida reales
198 h1 = h10 + h1m;
199 h2 = h20 + h2m;
200

```

```

201     H1(j) = h1;
202     H2(j) = h2;
203
204 end
205
206 %%% Indicador EISTWE %%%
207 % tomar valores desde 71 - cambio en el valor referencia SP
208 qindex = 0.5;%q
209 delt = 0.1;
210
211 ct1_mov1 = qindex*(du1_index(71:end)*du1_index(71:end)');
212 ct1_mov2 = qindex*(du2_index(71:end)*du2_index(71:end)');
213
214 error1x = 0;
215 error2x = 0;
216
217 zx=0;
218 for jj=71:numite
219     zx = (SP1(jj)-H1(jj))^2;
220     zx = jj*zx;
221     error1x = error1x + zx;
222 end
223
224 zx=0;
225 for jj=71:numite
226     zx = (SP2(jj)-H2(jj))^2;
227     zx = jj*zx;
228     error2x = error2x + zx;
229 end
230
231
232 %Para el multiplicador del error%
233 a1_index = max(H1(71:90));
234 a2_index = max(H2(71:90));
235
236 b1_index = min(H1(71:90));
237 b2_index = min(H2(71:90));
238
239 mult1 = (1+a1_index+b1_index)/SP1(72);
240 mult2 = (1+a2_index+b2_index)/SP2(72);
241
242 %Indicador de desempeño EISTWE
243 EISTWE = delt*N*M + mult1*(error1x + ct1_mov1)+ mult2*(error2x + ct1_mov2);
244
245 f = EISTWE;

```