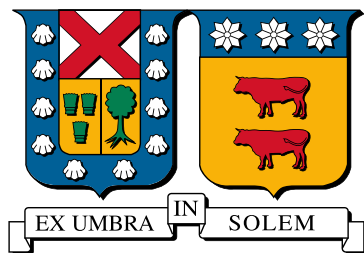


UNIVERSIDAD TÉCNICA FEDERICO SANTA
MARÍA

DEPARTAMENTO DE ELECTRÓNICA

VALPARAÍSO - CHILE



“IMPLEMENTACIÓN DE UN SISTEMA DE
CONTROL PREDICTIVO APLICADO A
ESTANQUES ACOPLADOS CON DSPACE”

DIEGO SEBASTIÁN ORREGO MATUS

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL
ELECTRÓNICO.

PROFESOR GUÍA: JUAN CARLOS AGÜERO

PROFESOR CORREFERENTE: ANGEL LEONEL CEDEÑO

MARZO 2024

Agradecimientos

En primer lugar, quisiera agradecer a mis profesores guías, Juan Carlos Agüero y Ángel Cedeño, por su gran compromiso y disposición para resolver las distintas dudas que se generaron en el camino. Me brindaron toda su experiencia y, con una gran paciencia (muchísima paciencia), me guiaron en el desarrollo de esta memoria.

En segundo lugar, a Ramón e Isabel, mis padres, quienes durante todos estos años han sido un pilar fundamental en mi vida. Estoy agradecido por su amor incondicional y su paciencia, a pesar de los muchos tropiezos que hubo. También quiero dedicar un espacio a mi hermano, mi tía, mi primo y mis mascotas, que de diversas formas han estado ahí para mí también en este largo proceso.

Por otro lado, a mis amigos que, si bien han ido evolucionando y migrando a distintas partes del país, estoy seguro de que sin su apoyo hubiera sido imposible llegar hasta aquí y poder, a pesar del tiempo y la distancia, mantener la amistad sincera y el cariño, incluso cuando el ritmo de la vida pareciera hacernos estar más ausentes.

Una mención a mi compañero Ariel Parra, que con su investigación previa [1], facilitó en gran medida el desarrollo teórico de esta investigación.

También a mi pareja, a quien coincidentemente conocí al iniciar mi vida universitaria y quien ha compartido conmigo mi periodo de mayor crecimiento, tanto profesional como personal. Estoy profundamente agradecido por el apoyo, el amor y el aprendizaje conjunto durante todos estos años.

Finalmente, agradezco el apoyo del Centro Basal "Advanced Center for Electrical and Electronic Engineering" (AC3E) - FB0008 y al proyecto ANID-FONDECYT 1211630

Implementación de un sistema de control predictivo aplicado a estanques acoplados con dSPACE

Diego Sebastián Orrego Matus

Memoria para optar al título de Ingeniero Civil Electrónico, mención Control e instrumentación, submención Electrónica industrial.

Universidad Técnica Federico Santa María

Profesor Guía: Juan Carlos Agüero

Profesor Correferente: Angel Leonel Cedeño

Marzo 2024

Resumen

Este proyecto introduce un avanzado método de control de procesos, denominado Control Predictivo por Modelo (MPC), aplicado específicamente a un sistema de estanques acoplados suministrado por Quanser, una empresa canadiense. El MPC, que se basa en la solución de un problema de optimización, utiliza el modelo del sistema para predecir las salidas futuras en un horizonte de tiempo definido y optimizar las acciones de control consecuentes. El objetivo principal es regular el nivel de agua en un estanque mediante la manipulación de una bomba.

Para la implementación del MPC, inicialmente se utiliza el método de Direcciones Alternadas de Multiplicadores de Lagrange (ADMM). Posteriormente, se aplica el método 'Active-set', un solver del entorno Quadprog en MATLAB especializado en problemas de programación cuadrática, para calcular las soluciones óptimas basadas en las predicciones y las restricciones establecidas.

Finalmente la implementación del controlador en la plataforma dSPACE, un sistema integrado que facilita la prueba de software y hardware. dSPACE proporciona capacidades de procesamiento avanzadas y una interfaz física

para monitorear y evaluar el desempeño del controlador, tanto en simulación como en la operación real del sistema.

Keywords: MPC; ADMM; Quadprog; Active-Set ;Estanques acoplados; MicroLabBox dSPACE

Implementation of a Predictive Control System Applied to Coupled Tanks Using dSPACE

Diego Sebastián Orrego Matus

Thesis for the fulfillment of the B.S Electronic Engineer, major in Control and
Instrumentation, minor in Industrial Electronics. (6 year program).

Universidad Técnica Federico Santa María

Advisor: Juan Carlos Aguero

Co-Advisor: Angel Leonel Cedeño

March 2024

Abstract

This project introduces an advanced process control method known as Model Predictive Control (MPC), specifically applied to a coupled tank system provided by Quanser, a Canadian company. MPC, which relies on solving an optimization problem, uses the system's model to predict future outputs over a defined time horizon and optimize subsequent control actions. The primary goal is to regulate the water level in a tank by manipulating a pump.

For the MPC implementation, the Alternating Direction Method of Multipliers (ADMM) is initially used. Subsequently, "Active-set", a Quadprog solver in MATLAB specialized in quadratic programming problems, is applied to compute optimal solutions based on predictions and established constraints.

Finally, the controller's implementation on the dSPACE platform, an integrated system that facilitates testing of software and hardware. dSPACE provides advanced processing capabilities and a physical interface to monitor and evaluate the controller's performance, both in simulation and in the

actual system operation.

Keywords: MPC; ADMM; Quadprog; Active-Set ;Coupled tanks; MicroLabBox dSPACE

Glosario

- **MPC:** Control predictivo por modelo
- **ADMM:** Método de direcciones alternadas de multiplicadores de Lagrange.
- **SISO:** Una entrada una salida.
- **MIMO:** Entradas múltiples y salidas múltiples.
- **SOLVER:** Algoritmo que provee soluciones al problema de optimización.
- **PID:** Control proporcional, derivativo e integrativo.
- **LQR:** Regulador cuadrático lineal.
- **QP:** Programación cuadrática
- **FPGA:** Matriz de compuertas programable en campo.
- **HMI:** Interfaz humano máquina.
- **A-D / D-A:** Conversores análogos digitales y digitales análogos.
- **RTI:** Interfaz de tiempo real.
- **KKT:** Condiciones de Karush-Kuhn-Tucker (Necesarias y suficientes para cumplir optimalidad)

Índice de contenidos

1. Introducción	1
1.1. Motivación y Contexto	1
1.2. Estado del Arte	2
1.2.1. Control MPC	2
1.2.2. Método de Optimización ADMM	3
1.2.3. Método de Optimización con Quadprog	3
1.2.4. MicroLabBox dSPACE	3
1.3. Objetivos del Trabajo	4
2. Control Predictivo por Modelo	5
2.1. Metodología del Control Predictivo	6
2.2. Programación Cuadrática (QP)	8
2.3. Formulación Problema de Control	9
2.4. Formulación Sparse y Restricciones	10
2.5. Seguimiento de Referencia	16
2.6. Observadores de Estados	18
2.7. Observador Extendido para perturbación	20
2.8. Seguimiento a Referencia con Perturbación Estimada	22
3. Método de optimización	24
3.1. Algoritmo de optimización ADMM	24
3.1.1. Descomposición Dual	24
3.1.2. Método de los Multiplicadores Aumentados de Lagrange	25
3.1.3. Método de los Multiplicadores de Lagrange con Direcciones Alternadas (ADMM)	26
3.1.4. Propiedades de convergencia en ADMM	26
3.1.5. Algoritmo ADMM para la Formulación Sparse	27
3.2. Algoritmos de optimización de Quadprog:	29
3.2.1. Metodología del Algoritmo Active-Set	29

4. Modelado del sistema	34
4.1. Estanques acoplados	34
4.2. Parámetros de modelo	35
4.3. Linealización del modelo	37
4.4. Sistema en variable de estados	40
4.5. Modelado en tiempo discreto	41
5. Implementación	46
5.1. MATLAB - Simulink	46
5.1.1. Diagrama de bloques MPC en MATLAB	47
5.2. Simulink - Quarc	49
5.2.1. Instrumentación para pruebas Quanser	50
5.2.2. Diagrama de bloques MPC Quanser	53
5.3. Implementación MicroLabBox dSPACE	54
5.3.1. Software para implementación en dSPACE	56
5.3.2. Implementación de estanques acoplados en MicroLabBox	57
5.3.3. Generación de un archivo ".sdf"	58
5.3.4. Creación de un proyecto en Control Desk	60
5.3.5. Toma de datos en Control Desk	62
5.3.6. Instrumentación para pruebas en MicroLabBox dSPACE	63
5.3.7. Consideraciones y problemas comunes con la implementación en Micro- LabBox	64
6. Resultados de Simulación - MATLAB	70
6.1. Simulación para estado cero	70
6.2. Simulación para seguimiento a referencia	72
6.3. Tabla comparativa de resultados	75
6.4. Simulación para seguimiento a referencia con rechazo a perturbaciones	77
6.5. Resultados implementación Quanser	79
7. Resultados implementación MPC - MicroLabBox:	81

8. Conclusiones	90
8.1. Trabajos futuros	92
9. Anexos	93
9.1. Código constantes MPC	93
9.2. Código de ADMM	96
Referencias	97

Índice de tablas

4.1. Tabla resumen de parámetros físicos del sistema.	36
6.2. Resultados de tiempos de ejecución solvers para distintos criterios de parada . .	76

Índice de figuras

2.1. Esquema de control MPC	6
2.2. RHC - Horizonte de control móvil	7
2.3. Cálculo de actuación u_{opt} en RHC	7
2.4. Esquema observador genérico	19
2.5. Diagrama Control predictivo con seguimiento a referencia y observadores	23
4.6. Planta estanques acoplados	34
4.7. Esquema de conexión	36
4.8. Esquema de control digital	41
5.9. Bloques de Simulink para MPC	46
5.10. Diagrama de bloques MPC	47
5.11. Bloque de seguimiento a referencia	47
5.12. Bloque MPC	48
5.13. Código de restricciones desigualdad g.	48
5.14. Código de restricciones de igualdad f.	49
5.15. Código para solver Quadprog - "Active-set" MATLAB	49
5.16. Esquema de conexión instrumentación Quanser	50

5.17. Amplificador 'VoltPAQ-x1' Quanser	51
5.18. Cables de medición - actuación estanques	51
5.19. Cables de alimentación sensores - bomba	52
5.20. Dispositivo de adquisición de datos Quanser	52
5.21. Diagrama de bloques esquema QUARC	53
5.22. Diagrama de bloques planta estanques acoplados QUARC	54
5.23. Diagrama de bloques MicroLabBox dSPACE [2]	55
5.24. Bloques RTI [3]	56
5.25. Control desk dSPACE [4]	57
5.26. Diagrama de bloques MPC en MicroLabBox	59
5.27. Archivos generados para el proyecto RTI	59
5.28. Descripción de las señales den Control Desk	61
5.29. Interfaz de visualización RTI Control Desk Estanques acoplados	61
5.30. Configuración "Recorder" para grabación de datos Control Desk	62
5.31. MicroLabBox 1202	63
5.32. Adaptador RCA-BNC para medición de Alturas	64
5.33. Cable para señal de control BNC-RCA	64
5.34. Configuración Solver de MATLAB en RTI1202	65
5.35. Configuración de cargado automático MicroLabBox	66
5.36. Ganancias DAC y ADC para MicroLabBox dSPACE	66
5.37. Menú superior RTI para cargar aplicaciones	68
5.38. Detener aplicación RTI	68
6.39. Planta para estado cero	70
6.40. MPC para estado cero	70
6.41. Altura del estanque $L_2(t)$ en estado cero Quadprog	71
6.42. Altura del estanque $L_2(t)$ en estado cero ADMM	71
6.43. Diagrama de bloques MPC seguimiento a referencia sin observadores	72
6.44. Seguimiento referencia única sin observadores ADMM	73
6.45. Seguimiento actuación ADMM sin observadores	73
6.46. Seguimiento referencia única sin observadores QP	74
6.47. Seguimiento actuación QP sin observadores	75

6.48. Seguimiento a referencia QP con observadores	77
6.49. Seguimiento de actuación $u(t)$ QP con observadores	78
6.50. Estados observados y referencia	78
6.51. Perturbación inducida d	79
6.52. Seguimiento de la referencia de altura $L_2(t)$	80
6.53. Seguimiento de la señal de actuación u_∞	80
6.54. Perturbación observada por efecto de transferencia estanque 1 a estanque 2	81
7.55. Seguimiento a referencia $L_2(t)$ QP simulado MicroLabBox	82
7.56. Seguimiento a actuación $u(t)$ QP simulado MicroLabBox	82
7.57. Seguimiento a referencia $L_2(t)$ QP planta física MicroLabBox	83
7.58. Seguimiento a actuación $u(t)$ QP planta física MicroLabBox	84
7.59. Observación de la perturbación estimada planta real MicroLabBox	85
7.60. Caso de falla seguimiento a referencia L_2	85
7.61. Caso de falla actuación $u(t)$	86
7.62. Seguimiento a referencia L_2 ADMM simulado MicroLabBox	87
7.63. Seguimiento a referencia $L_2(t)$ ADMM MicroLabBox	87
7.64. Seguimiento a actuación $u(t)$ ADMM MicroLabBox	88
7.65. Perturbación observada $d(t)$ MicroLabBox	89

1. Introducción

A lo largo de la historia y precisamente en la actualidad existe un alto interés en la optimización de los procesos industriales. La búsqueda de la mayor eficiencia, productividad, fiabilidad y por supuesto la rentabilidad tienen un factor clave a la toma de decisiones. En plena cuarta revolución tecnológica y a portas de llegar a la quinta, la tecnología ha tomado un rol fundamental al mejorar mayormente estos apartados y los sistemas de control no pueden quedar ajenos a los avances.

Una solución de los tiempos modernos para los procesos cada vez más complejos es la aplicación del control predictivo o MPC por sus siglas en inglés. Este basa su funcionamiento en la utilización de un modelo, con el cual es posible predecir el comportamiento futuro de la planta, que a su vez este utiliza un algoritmo de optimización que permite calcular y decidir eficazmente las actuaciones futuras.

Resulta particularmente interesante comparar los algoritmos de optimización disponibles, en primer lugar el algoritmo ADMM y por otro lado Quadprog con su propia suite de algoritmos, de tal manera de obtener el menor tiempo de ejecución, costo computacional, portabilidad y simplicidad en su aplicación.

1.1. Motivación y Contexto

La motivación para aplicar el avanzado método de control predictivo basado en modelos (MPC) radica en su capacidad para abordar eficazmente una amplia gama de sistemas actuales. Este enfoque integra el mundo de la optimización con el control automático, destacando por su habilidad para incorporar restricciones y ofrecer flexibilidad en el control. Además, aprovecha las modernas tecnologías de alto procesamiento, facilitando la adaptación entre sistemas y la integración con otras tecnologías emergentes como el aprendizaje automático, inteligencia artificial, analítica de datos, etc.

En específico la planta de estanques acoplados es de interés tanto industrial como académico, correspondiente a un desafío que permite operar una variedad de aplicaciones de diversa complejidad SISO o MIMO según sea su configuración.

Por otro lado, la implementación física con dSPACE permite dar un enfoque más industrial en

el testeo y monitoreo del sistema de control, sus características de procesamiento, visualización y conexión sencillas permite un entorno de prueba que destaca y permite romper ciertas barreras de compatibilidad entre sistemas.

1.2. Estado del Arte

1.2.1. Control MPC

El control MPC corresponde a un esquema de control avanzado altamente utilizado en la industria petroquímica, energética, etc. Este basa su funcionamiento a partir de un modelo de la planta a controlar, con el que será capaz de predecir salidas futuras en un instante de tiempo determinado conocido como horizonte de predicción. Ahora, con el modelo de predicción y la ayuda de un algoritmo de optimización es capaz de determinar las actuaciones futuras óptimas para la planta, haciendo así altamente efectiva su respuesta.

En relación a otros métodos de control regulatorio como el caso del PID que basa su respuesta a través de la realimentación de errores y el LQR que comparte la característica de minimización de una función de costo cuadrática, estos no tienen la capacidad de adelantarse al futuro donde radican las principales ventajas del MPC. Algunas otras ventajas se presentan a continuación:

- Altamente efectivo para sistemas no lineales.
- Adaptabilidad del sistema a cambios en el proceso.
- Facilidad de implementación de restricciones.
- Fácil adaptación para sistemas multivariables.

El control predictivo trabaja a partir de una función de costo cuadrática (QP) y con restricciones lineales de tal manera de minimizar la actuación necesaria considerando las predicciones previamente calculadas, para ello se utiliza un algoritmo de optimización cuadráticos. En específico en este proyecto se utilizan 2 correspondientes al ADMM y el método nativo de MATLAB Quadprog.

1.2.2. Método de Optimización ADMM

El Algoritmo del Método de los Multiplicadores de Lagrange con Direcciones Alternadas (ADMM) se trata de un procedimiento de optimización diseñado para abordar problemas convexos mediante su descomposición en partes más pequeñas. Este enfoque adopta una estructura de descomposición y coordinación, donde las soluciones a subproblemas locales se combinan de manera coordinada para encontrar una solución global a un problema mayor.

ADMM se destaca por ser especialmente apropiado en la optimización convexa distribuida y, en particular, para afrontar desafíos en problemas de gran escala que surgen en campos como estadística, aprendizaje automático y otras disciplinas afines.

1.2.3. Método de Optimización con Quadprog

Quadprog corresponde a una función de MATLAB, la cual está diseñada para resolver problemas cuadráticos de optimización a partir de los vectores de decisión y las matrices de coeficientes cuadráticos y lineales según sea el planteamiento del problema.

Quadprog, no es un método de optimización por si mismo, sino que se compone de algunos algoritmos propios para la optimización, como lo son:

- **Interior-Point-Convex.**
- **Trust-region-reflective.**
- **Active-set.**

1.2.4. MicroLabBox dSPACE

dSPACE es una empresa Alemana dedicada al desarrollo e innovación tecnológica, especializada en soluciones de desarrollo y prueba para sistemas de control en diversas áreas de la ingeniería donde se destacan la industria automotriz, energética, aeroespacial, robótica, etc.

Uno de los productos que provee la empresa es el dispositivo MicroLabBox, correspondiente a una unidad de prototipado y validación de sistemas de control embebido.

Está compuesto por principalmente por una unidad de procesamiento y una FPGA con más de 100 canales de entradas y salidas que proveen una conexión rápida del producto con la planta

real, sumado a un entorno para la visualización de información (HMI) y poder monitorear en tiempo real, la simulación con el mundo físico.

1.3. Objetivos del Trabajo

- Definir modelo y parámetros de planta no lineal.
- Definir modelo de planta linealizada para modelos de predicción, observadores, etc.
- Diseño de controlador MPC
 - Variables de decisión - Formulación problema de control.
 - Matrices de peso.
 - Función de costo.
 - Horizonte de predicción.
- Programación y prueba de optimización con ADMM.
- Programación y prueba de optimización con solvers de Quadprog.
- Comparación de algoritmos de optimización.
- Agregar acción integral con observadores.
- Implementación y prueba del sistema simulado.
- Implementación en Software y Hardware embebido con dSPACE.

2. Control Predictivo por Modelo

El control predictivo por modelo o MPC es un método avanzado de control, basado en la optimización del modelo de una planta, en el cual se predice el comportamiento futuro de la planta y a partir de las predicciones se calculan las acciones de control óptimas con tal de minimizar la función de costo cuadrático y poder seguir una referencia deseada en un determinado en el tiempo tal como se indica en [5]. En general, se pueden distinguir los 4 componentes principales del funcionamiento del control MPC y que se desarrollan con más profundidad en las secciones siguientes.

- Modelo matemático del sistema a controlar: Específicamente se requiere el sistema linealizado y discretizado para estimar las dinámicas del sistema.
- Horizonte de tiempo de N muestras, con el cual se decide la cantidad de acciones de control futuras a calcular para la aplicación del control.
- Conjunto de restricciones de estados y actuación del sistema: Limitaciones necesarias para que la optimización esté dentro de los parámetros físicos aceptados según la planta corresponda.
- Función de costo: Problema de optimización normalmente de carácter cuadrático que penaliza las desviaciones en los estados y actuaciones según se requiera para obtener los valores deseados de control.

El esquema de control predictivo más comúnmente utilizado se presenta en la figura 2.1.

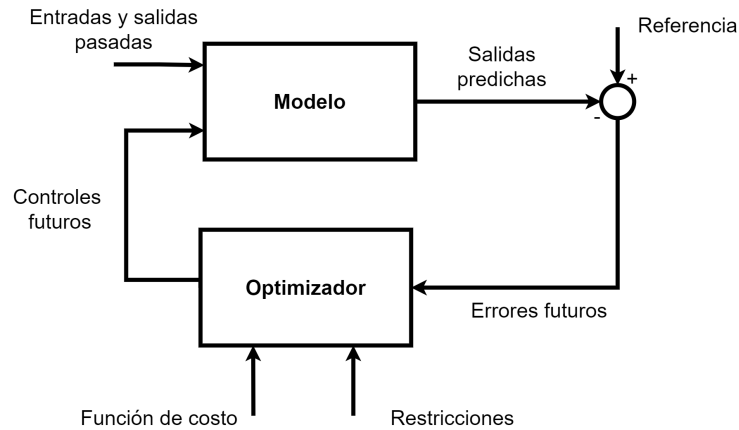


Figura 2.1: Esquema de control MPC

De la figura 2.1 se observa el bloque del modelo o planta, y el bloque optimizador, adicionalmente se suele incluir observadores para poder medir el estado actual de la planta, lo cual es entregado al bloque de optimización.

Algunas consideraciones, si bien existen diversas formas de implementar el control predictivo que son equivalentes, ya sea en términos de carácter no lineal o en tiempo continuo, en este trabajo nos centramos exclusivamente en el modelo linealizado y en su versión discreta.

2.1. Metodología del Control Predictivo

El control predictivo por modelo, como se ha explicado anteriormente está basado en una optimización de un modelo de la planta, esto se realiza de forma iterativa, dentro de un horizonte de predicción finito (N).

En un periodo de muestra discreto para un instante k se mide o se estima utilizando observadores el estado actual de la planta x_k , con dicha medición del estado y el modelo de matemático de la planta es posible predecir la secuencia de acciones de control futuras. Utilizando las predicciones de control se aplicará solo la primera entrada $u_{opt}(k)$ al sistema y se calculará nuevamente el valor óptimo siguiente para el instante $k + 1$, dentro del horizonte de control definido, esto se conoce como el método de horizonte móvil (RHC) [6]. A su vez estas acciones tienen que minimizar el problema de cuadrático de optimización y cumplir las restricciones impuestas, lo que se explicará a detalle en las siguientes secciones 2.2, 2.3, 2.4.

Para comprender mejor la metodología del control predictivo se presentan en las figuras 2.2, 2.3.

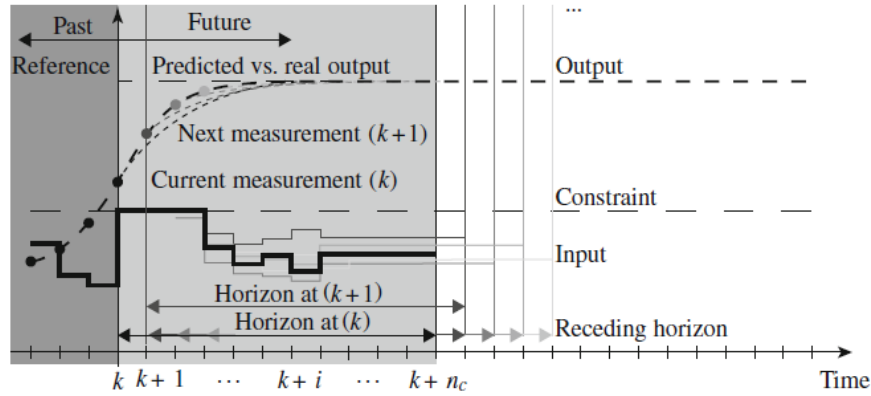


Figura 2.2: RHC - Horizonte de control móvil

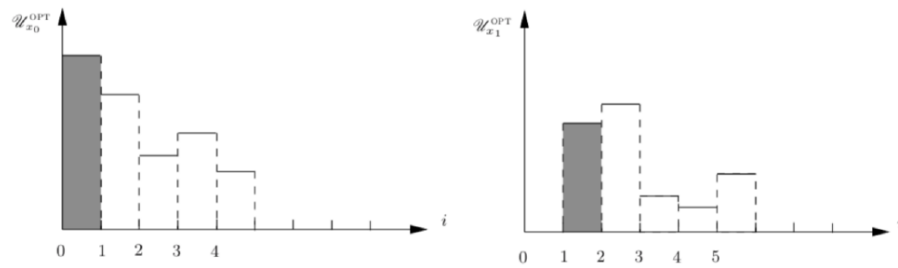


Figura 2.3: Cálculo de actuación u_{opt} en RHC

En la figura 2.2 del libro [6], se presenta la metodología del control predictivo con horizonte móvil de predicción o RHC. En la cual se puede observar en la sección de inputs equivalente a las actuaciones $u(k)$ de la planta, los distintos vectores de predicción óptimos, sin embargo, estos van cambiando en el tiempo y la secuencia aplicada corresponde a solo la primera entrada de dichas predicciones.

Se ejemplifica lo anterior, más claramente en la figura 2.3, utilizando un horizonte de predicción $N = 5$, donde inicialmente para el instante $t = 0$ se aplica la actuación $u_{opt}(0)$ correspondiente a la área sombreada entre los instantes $t = 0$ y $t = 1$, sin embargo, para el instante $t = 1$ se observa un cambio en la actuación $u_{opt}(1)$ calculada previamente y en siguientes los vectores de predicción. Por lo que nuevamente se aplicará solo la primera entrada y para los siguientes

instantes se repite el proceso en los respectivos horizontes establecidos.

Es importante destacar que el Control Predictivo por Modelo (MPC) en esencia se realiza como si fuera una respuesta en lazo abierto, lo que podría sugerir que el MPC no opera bajo principios de realimentación. Sin embargo, esta percepción es engañosa. Aunque el MPC se basa en la optimización de las acciones de control futuras sin realimentación directa durante ese proceso, la estrategia general del MPC sí incorpora un mecanismo de retroalimentación. Esto se debe a que en cada nuevo ciclo de muestreo k , el MPC actualiza sus predicciones y acciones de control basándose en el estado más reciente del sistema (x_0), lo que efectivamente le permite compensar desviaciones y seguir la trayectoria deseada.

2.2. Programación Cuadrática (QP)

El control predictivo tiene la capacidad de resolución de problemas lineales y no lineales, en general cuando se tiene un sistema lineal o se trabaja dentro de un rango de operación lineal es posible formular problemas cuadráticos *Quadratic Programming QP* utilizado en el desarrollo de esta memoria [7].

El problema cuadrático general, expresado de manera matricial, sujeto a restricciones de igualdad y desigualdad, se presenta en las ecuaciones 2.1,2.2.

$$\min J(x) = \frac{1}{2}x^T Hx + h^T x \quad (2.1)$$

Sujeto a las restricciones lineales:

$$\begin{aligned} Fx &= f \\ Gx &\leq g \end{aligned} \quad (2.2)$$

Tanto H , h que son las matrices del problema cuadrático dependen de las matrices de predicción del sistema. Donde H tiene que ser semidefinida positiva para que el problema sea convexo, es decir, que todos sus autovalores $\lambda \geq 0$. Cabe destacar que la importancia de que el problema sea convexo es para poder encontrar un mínimo global y por ende el valor óptimo real.

2.3. Formulación Problema de Control

Para plantear el problema de control MPC como un problema QP, se toma el sistema de la ecuación 2.3 de carácter lineal, en tiempo discreto e invariante en el tiempo.

$$\begin{aligned}x_{k+1} &= A_d x(k) + B_d u(k) \\ y_k &= C_d x(k)\end{aligned}\tag{2.3}$$

Con $x(k) \in \mathbb{R}^n$ con n número de estados, $u(k) \in \mathbb{R}^m$ m número de entradas, $y(k) \in \mathbb{R}^p$ p número de salidas. Tal que para el sistema discreto de la ecuación 2.3, se tiene que $A_d \in \mathbb{R}^{n \times n}$, $B_d \in \mathbb{R}^{n \times m}$, $C_d \in \mathbb{R}^{p \times n}$.

Ahora, para generar la función cuadrática QP de 2.1 con el sistema en variables de estado 2.3, con el objetivo de tener seguimiento a referencia, el vector de error $e(k)$:

$$e(k) = r(k) - y(k)\tag{2.4}$$

Ahora, la función de costo no puede seguir directamente una referencia, por lo que será necesario un arreglo matemático que se detalla en la sección 2.5. Al no es posible seguir directamente una referencia, por el momento se diseña de tal manera que $r_k = 0$ por lo que $e(k) = -y(k)$. El objetivo es llevar al sistema a su origen, buscando la penalización cuadrática en función de los estados del sistema 2.1, tal que reemplazando en el vector de error de 2.4 se obtiene equivalentemente:

$$\begin{aligned}J_k &= y_k^T y_k \\ J_k &= x_k^T \cdot C_k^T C_k \cdot x_k \\ \Omega &= C_k^T \cdot C_k\end{aligned}\tag{2.5}$$

De la ecuación 2.5 se renombra convenientemente el peso cuadrático de los estados Ω .

De igual manera, se busca penalizar las entradas del sistema como se presenta a continuación en la ecuación:

$$J_k = x_k^T \Omega x_k + u_k^T \Gamma u_k\tag{2.6}$$

Una vez formulado el problema cuadrático general, se añade la ponderación de los estados finales x_N , la cual se introduce para aproximar el comportamiento del lazo para un horizonte de tiempo infinito. Obteniendo así la función de costo cuadrático como se presenta en la ecuación 2.7.

$$J_N(x_k, u_k) = x_N^T \Omega_N x_N + \sum_{k=0}^{N-1} x_k^T \Omega x_k^T + u_k^T \Gamma u_k \quad (2.7)$$

Finalmente, el problema del control MPC se expresa como un problema de optimización cuadrático. Cuya función de costo 2.7, sujeto al sistema 2.3 con sus restricciones, queda expresado como 2.8.

$$u_k^{opt} = \arg \min_{u_k} J_N(x_k, u_k) \quad (2.8)$$

Sujeto a las restricciones:

$$\begin{aligned} x_0 &= x_k \\ x_{k+1} &= A_d x_k + B_d u_k \\ u^{min} &\leq u_k \leq u^{max} \\ x^{min} &\leq x_k \leq x^{max} \\ x_N^{min} &\leq x_N \leq x_N^{max} \end{aligned} \quad (2.9)$$

Cabe destacar se utiliza la notación x_0 para referirse al estado actual medido.

2.4. Formulación Sparse y Restricciones

Para el planteamiento matricial del problema de control anteriormente presentado en 2.8 y 2.9, la representación de la función de costo dependerá de las variables de decisión para la optimización.

En específico existen 2 formas para el planteamiento de las variables de decisión, la primera corresponde a la formulación densa, utilizando solo como variable de decisión la entrada u_k . La segunda opción corresponde a la formulación dispersa o 'Sparse' donde se utiliza tanto la entrada u_k y el estado x_k .

En este proyecto, se opta por la formulación Sparse debido a su capacidad de integrar direc-

tamente las restricciones en el problema de optimización.

La formulación Sparse [8], se presenta como el siguiente problema de optimización:

$$u_t^{opt} = \min_{\theta} J_N(x_k, \bar{u}) \quad (2.10)$$

Sujeto a las siguientes restricciones:

$$\begin{aligned} x_0 &= x(k) \\ x_{k+1} &= A_d x_k + B_d u_k \\ J x_k + E u_k &\leq d \\ J_N x_k^N &\leq d_N \end{aligned} \quad (2.11)$$

Las ecuaciones presentadas en 2.11 corresponden a las restricciones, tanto de igualdad como de desigualdad. Desglosando lo anterior, se definen las matrices de las restricciones como:

$$J = \begin{bmatrix} I_n \\ -I_n \\ C_{mn} \\ C_{mn} \end{bmatrix} \quad E = \begin{bmatrix} C_{nm} \\ C_{nm} \\ I_m \\ -I_m \end{bmatrix} \quad d = \begin{bmatrix} x_{\text{máx}} \\ -x_{\text{mín}} \\ u_{\text{máx}} \\ -u_{\text{mín}} \end{bmatrix} \quad (2.12)$$

$$J_N = \begin{bmatrix} I_n \\ -I_n \end{bmatrix} \quad d_N = \begin{bmatrix} x_N^{\text{máx}} \\ -x_N^{\text{mín}} \end{bmatrix} \quad (2.13)$$

Se desarrollan las restricciones del problema QP genérico presentado en 2.11, como se muestra a continuación:

$$x^{\text{mín}} \leq x_k \leq x^{\text{máx}} \quad (2.14)$$

$$x^{\text{mín}} \leq x_k \quad x_k \leq x^{\text{máx}} \quad (2.15)$$

Se acomodan las matrices para expresar en función de una misma inecuación de la manera:

$$-x_k \leq -x^{\text{mín}} \quad (2.16)$$

$$x_k \leq x^{\text{máx}} \quad (2.17)$$

Expresándolo de manera matricial:

$$\begin{bmatrix} I \\ -I \end{bmatrix} x_k \leq \begin{bmatrix} x^{\text{máx}} \\ -x^{\text{mín}} \end{bmatrix} \quad (2.18)$$

Equivalentemente para la entrada u_k , se tiene que:

$$u^{\text{mín}} \leq u_k \leq u^{\text{máx}} \quad (2.19)$$

$$x^{\text{mín}} \leq x_k \quad x_k \leq x^{\text{máx}} \quad (2.20)$$

Acomodando las inecuaciones con un mismo sentido, como:

$$-u_k \leq -u^{\text{mín}} \quad (2.21)$$

$$u_k \leq u^{\text{máx}} \quad (2.22)$$

Finalmente escribiéndolo se manera matricial:

$$\begin{bmatrix} I \\ -I \end{bmatrix} u_k \leq \begin{bmatrix} u^{\text{máx}} \\ -u^{\text{mín}} \end{bmatrix} \quad (2.23)$$

Agrupando las matrices calculadas en 2.18 y 2.23, se tiene la representación matricial del sistema.

$$\underbrace{\begin{bmatrix} I \\ -I \\ 0 \\ 0 \end{bmatrix}}_J x_k + \underbrace{\begin{bmatrix} 0 \\ 0 \\ I \\ -I \end{bmatrix}}_E u_k \leq \underbrace{\begin{bmatrix} x^{\text{máx}} \\ -x^{\text{mín}} \\ u^{\text{máx}} \\ u^{\text{mín}} \end{bmatrix}}_d \quad (2.24)$$

Donde $J \in \mathbb{R}^{(2n+2m) \times n}$ con n número de estados, m número de entradas, $E \in \mathbb{R}^{(2n+2m) \times n}$ para la matriz de entradas y finalmente la matriz $d \in \mathbb{R}^{2(n+m) \times 1}$ para las restricciones.

Se realiza el mismo trabajo anterior, para las restricciones de estados finales, como:

$$x_N^{\min} \leq x_N \leq x_N^{\max} \quad (2.25)$$

Separando nuevamente, como se presenta a continuación:

$$-x_N \leq -x_N^{\min} \quad (2.26)$$

$$x_N \leq x_N^{\max} \quad (2.27)$$

Llevando las relaciones directamente a su manera matricial, se obtiene que:

$$\underbrace{\begin{bmatrix} I \\ -I \end{bmatrix}}_{J_N} x_N \leq \underbrace{\begin{bmatrix} x_N^{\max} \\ -x_N^{\min} \end{bmatrix}}_{d_N} \quad (2.28)$$

Donde $J_N \in \mathbb{R}^{2n \times 1}$ equivalente para $d_N \in \mathbb{R}^{2n \times 1}$.

Para la formulación del problema QP Sparse, es necesario considerar que el vector de decisión, viene conformado tanto por las entradas y los estados del sistema de forma alternada, tal como:

$$\theta = \begin{bmatrix} x_0^T & u_0^T & x_1^T & u_1^T & \cdots & x_{N-1}^T & u_{N-1}^T & x_N^T \end{bmatrix}^T \quad (2.29)$$

Se reescribe la función de costo en función del vector de decisión 2.29, como:

$$J_N(\theta) = \begin{bmatrix} x_0 \\ u_0 \\ x_1 \\ u_1 \\ \vdots \\ x_{N-1} \\ u_{N-1} \\ x_N \end{bmatrix}^T \underbrace{\begin{bmatrix} \Omega & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \Gamma & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \Omega & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \Gamma & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \Omega & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & \Gamma & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \Omega_N \end{bmatrix}}_H \begin{bmatrix} x_0 \\ u_0 \\ x_1 \\ u_1 \\ \vdots \\ x_{N-1} \\ u_{N-1} \\ x_N \end{bmatrix} = \theta^T H \theta \quad (2.30)$$

En la formulación Sparse, a diferencia de la formulación densa el parámetro cuadrático respectivo a h de la ecuación 2.1 es nulo, debido a que las entradas se encuentran dentro del mismo vector de decisión, dependiendo así solo de la matriz H .

Ahora, se desarrollan las matrices de restricciones para expresarlas en función del vector de decisión.

- **Restricciones de Igualdad:**

En primer lugar, de la ecuación 2.11 las ecuaciones de igualdad están compuestas como:

$$x_0 = x_k \quad (2.31)$$

$$x_{k+1} = A_d x_k + B_d u_k \quad (2.32)$$

Las ecuaciones de igualdad de 2.31, se pueden reescribir matricialmente como:

$$\underbrace{\begin{bmatrix} -I_n & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ A & B & -I_n & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & A & B & -I_n & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & A & B & -I_n & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & A & B & -I_n \end{bmatrix}}_{\mathcal{F}} \begin{bmatrix} x_0 \\ u_0 \\ x_1 \\ u_1 \\ \vdots \\ x_{N-1} \\ u_{N-1} \\ x_N \end{bmatrix} = \underbrace{\begin{bmatrix} -x_0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{bmatrix}}_f \quad (2.33)$$

Lo anterior, lo podemos reescribir resumidamente como:

$$\mathcal{F}\theta = f \quad (2.34)$$

• **Restricciones de Desigualdad:**

Por otro lado, las restricciones de desigualdad, las podemos reescribir como:

$$\underbrace{\begin{bmatrix} J & E & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & J & E & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & J & E & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & J & E & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & J_N \end{bmatrix}}_{\mathcal{G}} \begin{bmatrix} x_0 \\ u_0 \\ x_1 \\ u_1 \\ \vdots \\ x_{N-1} \\ u_{N-1} \\ x_N \end{bmatrix} \leq \underbrace{\begin{bmatrix} d \\ d \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ d_N \end{bmatrix}}_g \quad (2.35)$$

Expresando resumidamente lo anterior, se obtiene que:

$$\mathcal{G} \leq g \quad (2.36)$$

Finalmente, el problema de optimización cuadrática Sparse, queda expresado como:

$$\theta^{opt} = \min_{\theta} J_N(\theta) \quad (2.37)$$

Sujeto a las restricciones:

$$\begin{aligned} \mathcal{F}\theta &= f \\ \mathcal{G}\theta &\leq g \end{aligned} \quad (2.38)$$

Ahora, en general las restricciones de igualdad y desigualdad se pueden resumir en una misma restricción de desigualdad de la forma:

$$M_x \theta \leq c_x \quad (2.39)$$

Bajo la misma lógica anterior, se reescriben en forma de una inecuación de igual dirección como:

$$M_x = \begin{bmatrix} \mathcal{G} \\ \mathcal{F} \\ -\mathcal{F} \end{bmatrix} \quad c_x = \begin{bmatrix} g \\ f \\ -f \end{bmatrix} \quad (2.40)$$

2.5. Seguimiento de Referencia

Como se menciona en secciones anteriores, el problema de control no tiene la capacidad directa de seguir referencias debido a que al momento de la optimización ocurre un problema matemático en el que se no es posible hacer tender el error $e(k)$ a cero sin hacer tender a cero también la entrada $u(k)$, lo que es no es deseado, inicialmente se define el problema tal que para una referencia $r(k) = 0$ se cumple que $e_k = -y_k$, sin embargo, es posible lograr un seguimiento a referencia cambiando el punto de operación del sistema.

Para que la salida siga una referencia $r \neq 0$ se necesita que el valor de salida estacionario denotado como y_{∞} tienda a la referencia tal que $y_{\infty} = r$.

Se reescribe el sistema en estado estacionario, con el objetivo de encontrar las relaciones ma-

triciales para que esto se cumpla:

$$\begin{aligned}x_{\infty} &= A_d x_{\infty} + B_d u_{\infty} \\y_{\infty} &= C_d x_{\infty}\end{aligned}\tag{2.41}$$

De las ecuaciones 2.41, se despeja x_{∞} y u_{∞} de tal manera de expresar el sistema en función de la referencia deseada, tal que:

$$\begin{aligned}x_{\infty} &= (I - A_d)^{-1} B_d u_{\infty} \\y_{\infty} &= C(I - A_d)^{-1} B_d u_{\infty}\end{aligned}\tag{2.42}$$

Despejando de 2.42 se despeja u_{∞} como:

$$u_{\infty} = [C(I - A_d)^{-1} B_d]^{-1} r\tag{2.43}$$

Una vez calculado el u_{∞} es posible calcular x_{∞} fácilmente.

El cambio de variable necesario para desplazar el punto de operación se expresa en función de los vectores estacionarios, como:

$$\begin{aligned}\bar{x} &= x - x_{\infty} \\ \bar{u} &= u - u_{\infty}\end{aligned}\tag{2.44}$$

Aplicando el cambio de variable al problema de optimización encontramos que se satisfacen las mismas condiciones dinámicas del sistema, sin embargo, no así de las restricciones por lo que será necesario de considerar los arreglos correspondientes como se muestra en 2.45.

$$\underbrace{\begin{bmatrix} \mathcal{O}_N \\ -\mathcal{O}_N \\ \mathcal{O} \\ -\mathcal{O} \\ I \\ -I \end{bmatrix}}_G \vec{u} \leq \underbrace{\begin{bmatrix} x_N^{max} - x_{\infty} - \mathcal{A}_N x_0 \\ -x_N^{min} + x_{\infty} - \mathcal{A}_N x_0 \\ \bar{x}^{max} - x_{\infty} - \mathcal{A} x_0 \\ -\bar{x}^{min} + x_{\infty} + \mathcal{A} x_0 \\ u^{max} - u_{\infty} \\ -u^{min} + u_{\infty} \end{bmatrix}}_g\tag{2.45}$$

Donde se mantienen los tamaños de las matrices como $G \in \mathbb{R}^{(2 \cdot n + 2 \cdot n \cdot N + 2 \cdot m \cdot N) \times (m \cdot N)}$ y $g \in \mathbb{R}^{(2 \cdot n + 2 \cdot n \cdot N + 2 \cdot m \cdot N) \times 1}$.

Ahora, es necesario considerar que el vector aplicado a la optimización corresponde a la desviación óptima de la entrada lo que permite seguir la referencia, sin embargo, a la planta es necesario de entregar la entrada óptima, tal que con el arreglo:

$$u^{opt} = \bar{u}_0^{opt} + u_\infty \quad (2.46)$$

Aplicando la entrada óptima de 2.46 basta para solucionar el problema de seguimiento, aunque esto no asegura tener un error estacionario cero, por lo que es necesario aplicar acción integral al controlador, y esto se hace a partir de los observadores de estados y perturbaciones en la siguiente sección.

2.6. Observadores de Estados

En las secciones anteriores se describe x_0 como la medición del estado actual, para la formulación del problema de control, sin embargo, en general para realizar la realimentación de estado no siempre es tan sencillo acceder a estas mediciones o simplemente no es económicamente viable.

Un observador de estados, es un sistema dinámico en el cual se tiene como objetivo que los estados observados converjan a los estados del sistema real [9].

Un esquema típico de un observador simple se presenta en la figura 2.4.

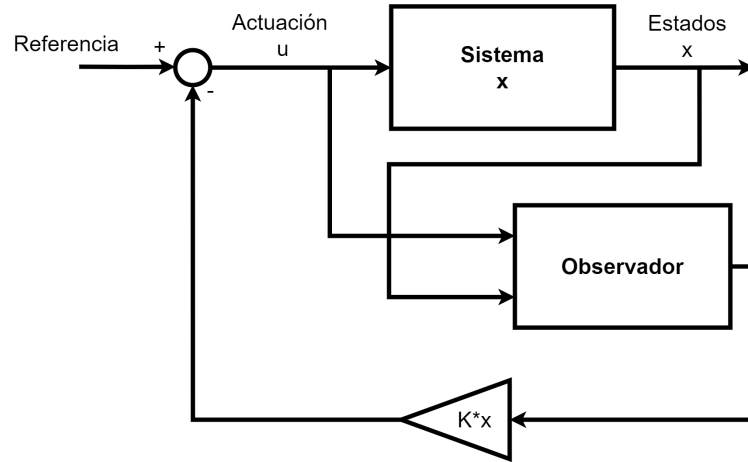


Figura 2.4: Esquema observador genérico

De la figura 2.4 se presenta el esquema de un observador simple, donde a partir de la entrada u_k y la salida del sistema y_k se estiman los estados $\hat{x}$ y donde K representa la matriz de ganancia de realimentación de estado.

Las ecuaciones que describen la dinámica de un observador se presentan a continuación:

$$\begin{aligned} x_{k+1} &= A_d x_k + B_d u_k \\ y_k &= C_d x_k \end{aligned} \quad (2.47)$$

Es necesario conocer el modelo del sistema para poder observarlo, por ende se asumen A_d, B_d, C_d conocidas y $D_d = 0$.

De la ecuación 2.47 es posible reconstruir el sistema como:

$$\hat{x}_{k+1} = A_d \hat{x}_k + B_d u_k \quad (2.48)$$

Tal que, utilizando la matriz de realimentación de estado mostrada en la figura 2.4, se obtiene que:

$$\hat{x}_{k+1} = A_d \hat{x}_k + B_d u_k + L_d (y_k - C \hat{x}_k) \quad (2.49)$$

Donde L_d equivalente a la matriz K de ganancia de retroalimentación de estados mencionada anteriormente.

Para la estimación de estado, se busca que el término $L_d(y_k - C_d\hat{x}_k)$ no tenga efecto, es decir, forzar que la salida y_k sea igual a la estimación, de tal manera que el vector de error queda expresado como:

$$\tilde{x}_k = x_k - \hat{x}_k \quad (2.50)$$

De la ecuación 2.50 se reescribe la ecuación 2.49, de donde se obtiene:

$$\tilde{x}_k = A_d\tilde{x}_k - L_d(y_k - C_d\hat{x}_k) \quad (2.51)$$

De lo anterior, se obtiene que:

$$A_{obs} = (A_d - L_dC) \quad (2.52)$$

Finalmente, se deben colocar convenientemente los polos de L_d , para que la matriz A_{obs} sea estable y por ende se puedan reconstruir los estados, disminuyendo el error a cero.

2.7. Observador Extendido para perturbación

En los sistemas de control es común tener perturbaciones que pueden o no ser medibles y las cuales pueden afectar el rendimiento del observador si estas no son modeladas correctamente. Para modelar las perturbaciones en la entrada del sistema, se define el modelo con perturbaciones como:

$$x_{k+1} = A_d x_k + B_d u_k + B_d d_k \quad (2.53)$$

$$y_k = C x_k + e_k \quad (2.54)$$

Donde d_k corresponde a la perturbación a estimar $e(k) \rightarrow 0$ correspondiente al error en la medición.

Para estimar la perturbación d_k , se aumenta el grado del sistema y se define un nuevo vector de estado z_k de la forma:

$$z_k = \begin{bmatrix} x_k \\ d_k \end{bmatrix} \quad (2.55)$$

Reescribiendo el sistema 2.53 de forma matricial, en base al vector de estado z_k extendido:

$$\underbrace{\begin{bmatrix} x_{k+1} \\ d_{k+1} \end{bmatrix}}_{\hat{z}_k} = \underbrace{\begin{bmatrix} A_d & B_d \\ 0 & I \end{bmatrix}}_{A_z} \underbrace{\begin{bmatrix} x_k \\ d_k \end{bmatrix}}_{z_k} + \underbrace{\begin{bmatrix} B_d \\ 0 \end{bmatrix}}_{B_z} u_k \quad (2.56)$$

$$y_k = \underbrace{\begin{bmatrix} C & 0 \end{bmatrix}}_{C_z} \begin{bmatrix} x_k \\ d_k \end{bmatrix} \quad (2.57)$$

Equivalente al observador simple diseñado anteriormente, se diseña la ley de realimentación de estado con una ganancia L_z :

$$\hat{z}_{k+1} = A_z \hat{z}_k + B_z u_k + L_z (y_k - C_z \hat{z}_k) \quad (2.58)$$

Reagrupando las matrices extendidas 2.56 y 2.57 con la ley de control propuesta en 2.58, se obtiene el sistema estimado observado:

$$\bar{x}_{k+1} = A_{obs} \bar{x}_k + B_{obs} \bar{u}_k \quad (2.59)$$

$$\bar{y}_k = C_{obs} \bar{x}_k \quad (2.60)$$

Donde las matrices observadas:

$$A_{obs} = A_z - L_z C_z; \quad B_{obs} = \begin{bmatrix} B_z \\ L_z \end{bmatrix}; \quad C_{obs} = I \quad (2.61)$$

Tal que los vectores de estado, tanto de entrada como salida, quedan de la forma:

$$\text{entrada : } \tilde{u}_k = \begin{bmatrix} u_k \\ y_k \end{bmatrix}; \quad \text{salida : } \tilde{y}_k = \begin{bmatrix} x_k \\ d_k \end{bmatrix} \quad (2.62)$$

2.8. Seguimiento a Referencia con Perturbación Estimada

Una vez realizadas las estimaciones de las perturbaciones, es necesario rediseñar el seguimiento a referencia del sistema. Esto no va a afectar directamente al planteamiento del problema de control, pero sí a las restricciones.

Se plantea al sistema extendido previamente presentado como:

$$x_{k+1} = A_d x_k + B_d u_k + B_d d_k \quad (2.63)$$

$$y_k = C x_k \quad (2.64)$$

Ahora, para el seguimiento a referencia, tal como se trabaja anteriormente el sistema se lleva a estado estacionario, donde $y_\infty \rightarrow r$ y además, se considerará la perturbación d_k estimada como una constante. Con esto, reemplazando la ecuación anterior:

$$x_\infty = A_d x_\infty + B_d u_\infty + B_d \hat{d}_\infty \quad (2.65)$$

$$y_\infty = C x_\infty \quad (2.66)$$

En primer lugar, se despeja para x_∞ los estados finales como:

$$\begin{aligned} x_\infty - A_d x_\infty &= B_d(u_\infty + \hat{d}) \\ (I - A_d)x_\infty &= B_d(u_\infty + \hat{d}) \\ x_\infty &= (I - A_d)^{-1} B_d(u_\infty + \hat{d}) \end{aligned} \quad (2.67)$$

Ahora, la salida del sistema queda de la forma:

$$y_\infty = C(I - A_d)^{-1} B_d(u_\infty + \hat{d}) \quad (2.68)$$

De la ecuación anterior, se despeja u_∞ :

$$\begin{aligned}
 [C(I - A_d)^{-1}B_d]^{-1}y_\infty &= u_\infty + \hat{d} \\
 u_\infty &= [C(I - A_d)^{-1}B_d]^{-1}y_\infty - \hat{d} \\
 u_\infty &= [C(I - A_d)^{-1}B_d]^{-1}r - \hat{d}
 \end{aligned}
 \tag{2.69}$$

Lo anterior, refleja que si bien la perturbación estimada no afecta directamente al problema de control, si cambia la forma de cálculo de los vectores u_∞ y x_∞ .

Finalmente, el esquema de control predictivo MPC con seguimiento a referencia, observación y estimación de estados, perturbaciones queda de la forma:

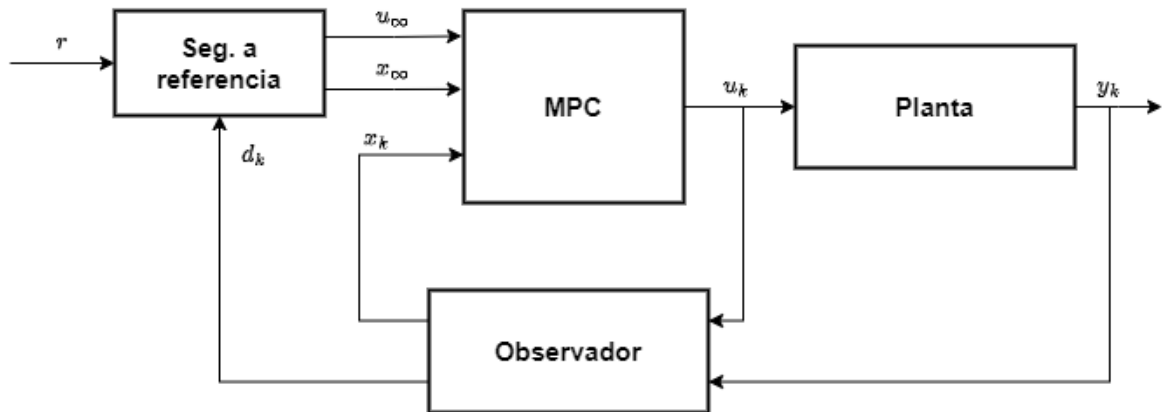


Figura 2.5: Diagrama Control predictivo con seguimiento a referencia y observadores

3. Método de optimización

Desde una perspectiva analítico matemática, entendemos la optimización como el proceso de obtener la respuesta factible que minimiza o maximiza una función objetivo dentro de un conjunto de alternativas.

En el control automático y especialmente en el control predictivo, los algoritmos de optimización tienen como objetivo calcular las acciones de control futuras más adecuadas dentro de un horizonte finito. Esto permite al controlador resolver de la manera más efectiva problemas altamente complejos.

Cabe destacar que el control predictivo aplica técnicas de optimización mayormente no lineales, como es el caso de la optimización cuadrática (QP), sin embargo, para la simplificación de los problemas y su computo generalmente se imponen restricciones lineales las cuales acotan los resultados a parámetros esperados.

Es en este contexto es donde se presentan 2 alternativas de algoritmos para solución de problemas cuadráticos, el primero de ellos el método de multiplicadores alternados de Lagrange (ADMM) y el segundo método llamado 'Active-set'.

3.1. Algoritmo de optimización ADMM

Para comprender la metodología del algoritmo de optimización con direcciones alternadas de multiplicadores de Lagrange, se presentan los métodos precursores que forman sus cimientos. Este análisis está principalmente basado en el libro [10].

3.1.1. Descomposición Dual

Entendemos la descomposición dual de un problema de optimización, como la división de un problema más complejo, de tal manera formular problemas más sencillos que iterativamente estos converjan a la solución óptima del problema inicial.

Se define el siguiente problema de minimización de una función $f(x)$:

$$\min_x f(x) \quad \text{sujeto a: } Ax = b \quad (3.70)$$

con $f(x)$ estrictamente convexa y restricciones lineales.

Se define el Lagrangiano de $f(x)$ como:

$$L(x, u) = f(x) + u^T(Ax - b) \quad (3.71)$$

El Lagrangiano también es separable en x , de tal forma que el gradiente de la descomposición dual se repite para $k = 1, 2, 3, \dots, N$ de la siguiente forma:

$$x^{(k)} = \arg \min_x L(x, u^{(k-1)}) \quad (3.72)$$

$$u^{(k)} = u^{(k-1)} + \tau^{(k)}(Ax^{(k)} - b) \quad (3.73)$$

3.1.2. Método de los Multiplicadores Aumentados de Lagrange

Aquí se aumenta el Lagrangiano de 3.71 para el parámetro ρ tal que $\rho > 0$:

$$\min_x \left(f(x) + \frac{\rho}{2} \|Ax - b\|_2^2 \right) \quad (3.74)$$

El Lagrangiano aumentado en función del parámetro ρ queda expresado como:

$$L_\rho(x, u) = f(x) + u^T(Ax - b) + \frac{\rho}{2} \|Ax - b\|_2^2 \quad (3.75)$$

Repitiendo iterativamente el proceso para $k = 1, 2, 3, \dots, N$:

$$x^{(k)} = \arg \min_x L_\rho(x, u^{(k-1)}) \quad (3.76)$$

$$u^{(k)} = u^{(k-1)} + \rho(Ax^{(k)} - b) \quad (3.77)$$

Con lo anterior, se logran mejores propiedades de convergencia, aunque a su vez se pierde descomponibilidad.

3.1.3. Método de los Multiplicadores de Lagrange con Direcciones Alternadas (ADMM)

El método de los multiplicadores alternados de Lagrange, toma lo mejor de los métodos anteriormente descritos.

Para un problema de optimización, de forma:

$$\min_{x,z} f(x) + g(z) \quad \text{sujeto a: } Ax + Bz = c \quad (3.78)$$

Donde $f(x)$ y $g(z)$ funciones convexas, cerradas y por ende separables, se puede expresar el Lagrangiano aumentado para el parámetro $\rho > 0$, como:

$$L_\rho(x, z, u) = \underbrace{f(x) + g(z)}_{\text{Función De Costo}} + \underbrace{u^T(Ax + Bz - c)}_{\text{Multiplicador de Lagrange}} + \underbrace{\frac{\rho}{2}\|Ax + Bz - c\|_2^2}_{\text{Término Proximal Aumentado}} \quad (3.79)$$

Se repite para $(k = 1, 2, 3, \dots, N)$ quedando finalmente el método de ADMM en función de la minimización del Lagrangiano anterior, como:

$$x^{(k)} = \arg \min_x L_\rho(x, z^{(k-1)}, u^{(k-1)}) \quad (3.80)$$

$$z^{(k)} = \arg \min_z L_\rho(x^{(k-1)}, z, u^{(k-1)}) \quad (3.81)$$

$$u^{(k)} = u^{(k-1)} + \rho(Ax^{(k)} + Bz^{(k)} - c) \quad (3.82)$$

Igualmente al caso anterior, se ganan propiedades de convergencia, sin embargo, esto depende directamente del parámetro ρ . Por lo que es esencial ajustarlo adecuadamente, lo cual puede ser matemáticamente complejo.

3.1.4. Propiedades de convergencia en ADMM

Algunas propiedades de la convergencia, para f y g convexas:

- **Convergencia Residual:** $r^{(k)} = Ax^{(k)} + Bz^{(k)} - c \rightarrow 0$ como $k \rightarrow \infty$.

- **Convergencia Objetivo:** $f(x^{(k)}) + g(z^{(k)}) \rightarrow f^* + g^*$, donde $f^* + g^*$ son las soluciones óptimas primales.
- **Convergencia Dual:** $u^{(k)} \rightarrow u^*$, donde u^* es una solución dual.

El desarrollo anterior, se obtiene desde [11] donde se presentan resumidamente notas de las ecuaciones que definen el método ADMM y el cual se basa principalmente en el libro de optimización convexa [12].

3.1.5. Algoritmo ADMM para la Formulación Sparse

Para el planteamiento Sparse del método ADMM se basa principalmente en el artículo [13]. Se remonta al planteamiento general de un problema de optimización cuadrática, de la siguiente manera:

$$\min_x \frac{1}{2} x^T H x + h^T x \quad (3.83)$$

sujeto a las restricciones:

$$Gx \leq g \quad (3.84)$$

$$Fx = f \quad (3.85)$$

donde H tiene que ser definida semidefinida positiva, es decir, que sus autovalores tienen que ser todos mayores o iguales que cero.

De este planteamiento surge un problema, debido a que ADMM solo permite restricciones de igualdad. Para solucionar esto, se amplía a restricción G , con la variable de holgura z , tal que:

$$Gx + z = g \quad (3.86)$$

Podemos reescribir el problema de optimización QP en variables separables de x y z , como:

$$\begin{aligned} \min_{x,z} \quad & \frac{1}{2} \underbrace{x^T H x + h^T x}_{f(x)} + \underbrace{I_Z(z)}_{g(z)} \\ \text{sujeto a} \quad & Ax + Bz = c \\ & Z = \{z : z \geq 0\} \end{aligned} \quad (3.87)$$

Donde, las restricciones pueden expresarse como matrices de la siguiente forma:

$$A = \begin{bmatrix} G \\ F \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ I \end{bmatrix}, \quad c = \begin{bmatrix} g \\ f \end{bmatrix} \quad (3.88)$$

Una vez planteado el problema matricial, se retoman los subproblemas de ADMM descritos en las ecuaciones 3.80,3.81,3.82. Tal que, reemplazando el Lagrangiano aumentado de la ecuación 3.79, los subproblemas quedan expresados como:

$$x^{k+1} = \arg \min_x \left(f(x) + \frac{\rho}{2} \|Ax + Bz^k - c + u^k\|^2 \right) \quad (3.89)$$

$$z^{k+1} = \arg \min_z \left(g(z) + \frac{\rho}{2} \|Ax^{k+1} + Bz - c + u^k\|^2 \right) \quad (3.90)$$

$$u^{k+1} = u^k + Ax^{k+1} + Bz^{k+1} - c \quad (3.91)$$

Para encontrar las soluciones óptimas de estos subproblemas y también dejarlas expresadas en los términos cuadráticos, se debe derivar e igualar a cero las ecuaciones respectivas de 3.89 para x y 3.90 para z .

Realizando el trabajo matemático previamente descrito, se obtiene finalmente el método de optimización ADMM Sparse aplicado en este proyecto, el cual queda de la forma:

$$x^{(k+1)} = (H + \rho A^T A)^{-1} \left[-h - \rho A^T (Bz^{(k)} - c + u^{(k)}) \right] \quad (3.92)$$

$$z^{(k+1)} = \max \left\{ 0, g - Gx^{(k+1)} - u^{(k)} \right\} \quad (3.93)$$

$$u^{(k+1)} = u^{(k)} + Ax^{(k+1)} + Bz^{(k+1)} - c \quad (3.94)$$

3.2. Algoritmos de optimización de Quadprog:

En MATLAB, se disponen de diversas estrategias para la resolución de problemas, siendo Quadprog uno de los entornos más utilizados para abordar problemas cuadráticos. Quadprog incluye varios solvers para optimización cuadrática, entre los cuales se destaca el método "Active-set".

El método "Active-set", se trata de una estrategia de resolución de problemas cuadráticos, el cual basa su metodología, a partir de un conjunto de restricciones iniciales para encontrar un punto factible. Posteriormente se va iterando el conjunto, con el objetivo de encontrar otro punto factible que calce con las condiciones de optimalidad.

Ahora, la elección del algoritmo "Active-set" se debe a su compatibilidad con Simulink, ya que, dadas las características específicas del sistema, los otros métodos disponibles en Quadprog, como el de punto interior, no se logran compilar.

Siguiendo la metodología descrita en las referencias [14] y [15], se procede a explicar detalladamente el procedimiento empleado por el método "Active-set".

3.2.1. Metodología del Algoritmo Active-Set

Para un problema cuadrático de la forma:

$$\min_x \frac{1}{2} x^T Q x + c^T x \quad (3.95)$$

sujeto a

$$Ax = b \quad (\text{restricciones de igualdad}) \quad (3.96)$$

$$Dx \geq e \quad (\text{restricciones de desigualdad}) \quad (3.97)$$

Para una matriz "Q" Hessiana definida semipositiva, la función objetivo es cuadrática convexa y donde las restricciones describen ecuaciones e inecuaciones lineales. El método "Active-set", sigue la siguiente metodología:

- **Inicialización:**

En primer lugar, se inicializa el problema de optimización, buscando un punto factible de solución x_0 , que cumpla todas las condiciones de igualdad y que esté dentro de los

parámetros de las condiciones de desigualdad. Además, se debe escoger el conjunto A_0 , como las posibles soluciones de desigualdad las cuales cumplen con ser **Activas**, es decir, que la desigualdad se comporte como igualdad para el punto x_0 . Tal como, se presenta a continuación:

$$A_0 = \{i \mid D_i x_0 = e_i, i \text{ es un índice de todas las restricciones de desigualdad}\} \quad (3.98)$$

- **Resolución de subproblema cuadrático:**

Una vez con la posible solución, el conjunto de restricciones de igualdad y las restricciones de desigualdad activas (tratadas como igualdad), se formula el subproblema de optimización cuadrático de la forma:

$$\min_d \frac{1}{2} d^T Q d + c^T d, \quad \text{sujeto a } A_{eq} d = 0 \quad (3.99)$$

En 3.99, la variable "d" corresponde a la dirección de búsqueda.

- **Determinación de dirección y tamaño del paso:**

Para calcular la dirección d , presentada anteriormente, se utiliza el método de los multiplicadores de Lagrange, tal como se muestra en la siguiente ecuación:

$$Qd + A^T \lambda = -c \quad (3.100)$$

Así, λ son los multiplicadores de las restricciones activas. Por otro lado, el cálculo del tamaño de paso se calcula de la forma:

$$\alpha_k = \min \left(1, \min_{i \notin W_k, D_i d_k > 0} \frac{f_i - D_i x_k}{D_i d_k} \right) \quad (3.101)$$

De donde:

- α_k es el tamaño del paso en la iteración k .
- W_k es el conjunto activo en la iteración k , que contiene los índices de las restricciones que se sospecha están activas (tratadas como igualdades).
- D_i es la fila i -ésima de la matriz $\mathbf{D}$ que define las restricciones de desigualdad.

- d_k es la dirección de búsqueda en la iteración k .
- x_k es el punto actual en la iteración k .
- f_i es el elemento i -ésimo del vector $\mathbf{f}$ que define los términos constantes de las restricciones de desigualdad.

Ahora, en general podemos entender el cambio de dirección y el tamaño de paso, tiene que cumplir la siguiente ecuación:

$$x_{\text{nuevo}} = x_{\text{viejo}} + \alpha d \quad (3.102)$$

De tal manera que, la variable x_{nuevo} corresponderá al nuevo punto óptimo, función del punto anterior, la dirección a seguir y el tamaño del paso, dentro del conjunto de restricciones.

• **Verificación de condiciones de optimalidad KKT:**

Para verificar que el punto calculado sea óptimo, deben cumplirse las siguientes condiciones, que son tanto suficientes como necesarias para demostrar la optimalidad:

– **Factibilidad primal:**

En primer lugar, la factibilidad primal corresponde a que la solución cumpla con todas las restricciones, ya sean activas o no, de la forma:

$$Ax = b \quad (\text{restricciones de igualdad}) \quad (3.103)$$

$$Dx \geq f \quad (\text{restricciones de desigualdad}) \quad (3.104)$$

Una vez, se comprueba que la solución está dentro de las restricciones se avanza al siguiente criterio.

– **Estacionariedad:**

En segundo lugar, el criterio de estacionariedad, consta de que el gradiente de la función objetivo cuando se evalúa en el punto óptimo tiene que ser igual a cero, es decir, tiene que cumplir que sea un mínimo global, de la forma:

$$Qx^* + c + A^T \lambda_1 + D^T \lambda_2 = 0 \quad (3.105)$$

Donde x^* corresponden al punto óptimo y λ_1, λ_2 corresponde a los multiplicadores de Lagrange asociados a las restricciones de igualdad y desigualdad respectivamente.

– **Factibilidad dual:**

Al igual que el caso de la optimización ADMM, resulta útil plantear el comportamiento dual de la función, en específico, para las restricciones de desigualdad el multiplicador de Lagrange debe ser mayor o igual que cero, tal que:

$$\lambda_2 \geq 0 \quad (3.106)$$

– **Holgura complementaria:**

Por último, el criterio de holgura complementaria se expresa a través de la siguiente ecuación, que relaciona cada restricción de desigualdad con su respectivo multiplicador de Lagrange:

$$\lambda_i \cdot (D_i x - f_i) = 0 \quad \forall i \quad (3.107)$$

De la ecuación 3.107, se establece que, si $D_i x > f_i$ implica que la restricción no está activa y por ende el multiplicador de Lagrange λ_i está obligado a ser 0. Por otro lado, si $\lambda_i \leq 0$ entonces $D_i x = f_i$, se cumple que restricción es activa y por consecuente se puede tratar como igualdad.

• **Actualización del conjunto activo e iteración:**

Para ajustar el conjunto activo W_k si se encuentra que alguna restricción de desigualdad que está fuera de rango, es decir, que no se cumple, entonces se añade su índice al conjunto activo. Por otro lado, si al evaluar los multiplicadores de Lagrange λ_i asociados con las restricciones de desigualdad activas, se encuentra que alguno es negativo, se elimina la restricción correspondiente del conjunto activo, tal como se explicó en la sección anterior. Resumiendo lo anterior, en el siguiente procedimiento:

- *Añadir al Conjunto Activo:* Si $D_i x_{\text{nuevo}} < f_i$, entonces añade i a W_{k+1} .
- *Eliminar del Conjunto Activo:* Si $\lambda_i < 0$ para alguna restricción activa, entonces



elimina i de W_{k+1} .

- **Actualizar el Punto de Solución e iteraciones:**

Si se añade una restricción al conjunto activo, se debe calcular una nueva dirección de búsqueda y tamaño del paso antes de actualizar x y por ende se repiten los procesos anteriormente mencionados.

Esta metodología del solver "Active-set", garantiza encontrar un punto factible que sea también un punto estacionario de la función objetivo, cumpliendo con todas las restricciones y las condiciones de optimalidad.

4. Modelado del sistema

4.1. Estanques acoplados

La planta corresponde a un sistema hidráulico respectivo a 2 estanques acoplados verticalmente. Estos son un modelo a escala suministrados por la empresa canadiense Quanser para pruebas de laboratorio.

La planta de los estanques acoplados presenta un desafío de control muy variado y requerido tanto a nivel académico como industrial. Se presenta en la figura 4.6 la planta física y sus respectivos componentes.

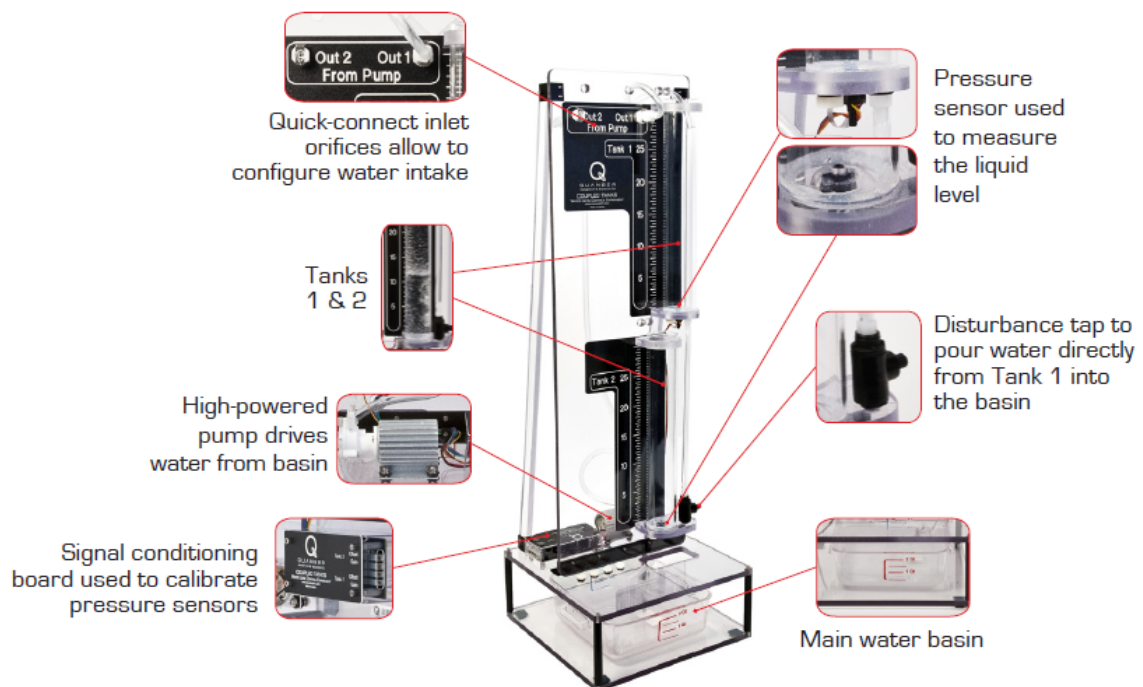


Figura 4.6: Planta estanques acoplados

En la Figura 4.6, se presentan los estanques y se detalla la ubicación de cada uno de sus componentes. Este montaje está equipado con conexiones de acople rápido que permiten una transición entre configuraciones de control para cada estanque, ya sea operando como un sistema de entrada y salida única (SISO, por sus siglas en inglés) o como un sistema multivariable

(MIMO, por sus siglas en inglés).

Además, se incorporan dos sensores de presión ubicados en la parte inferior de los estanques para realizar mediciones del nivel de líquido. Junto con esto se dispone de una placa de acondicionamiento de señal, cuya función es calibrar y mejorar las mediciones provenientes de los sensores.

Se dispone de solo un actuador principal, respectivo a una bomba de agua que se abastece desde una vasija principal.

La dinámica de conexión entre los estanques se fundamenta en la acción de la gravedad. Es relevante destacar que los orificios de salida de los estanques son ajustables en tamaño, lo que permite modificar la tasa de flujo de salida. Adicionalmente, se dispone de un mecanismo de perturbación, el cual se activa mediante un interruptor manual para permitir una descarga directa desde el estanque superior hacia la vasija principal.

Para los propósitos de esta memoria, se contempla una configuración SISO, en la que la alimentación de agua se dirige exclusivamente al estanque superior (Estanque 1). Esto establece un control sobre el nivel del estanque inferior (Estanque 2) mediante la regulación del flujo de líquido que se desplaza por gravedad desde el estanque superior.

4.2. Parámetros de modelo

El modelo del sistema y sus parámetros se obtienen desde el texto de trabajo del estudiante suministrado por Quanser [16].

Para obtener las ecuaciones del sistema, en primer lugar se simplifica la planta, con el fin de obtener un modelo donde se muestren las variables dinámicas de interés y su relación entre sí, tal como se muestra en la figura 4.7.

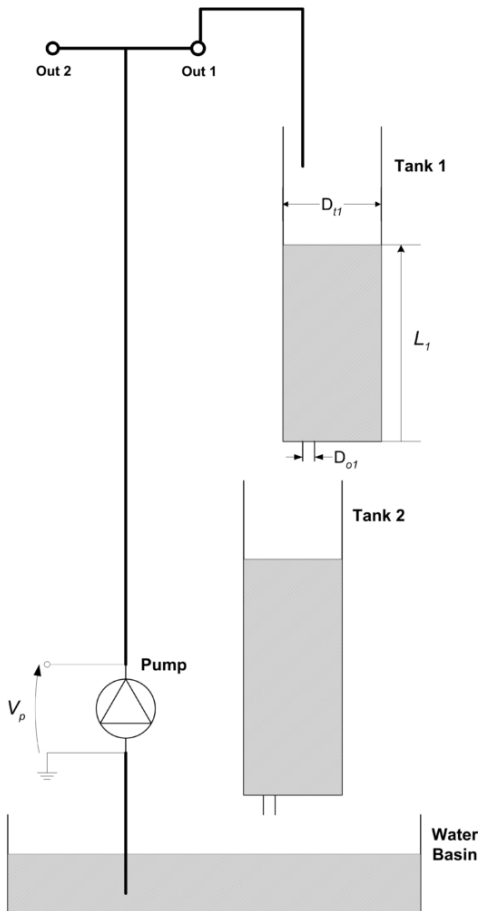


Figura 4.7: Esquema de conexión

Los parámetros del modelo se resumen en la tabla 4.1.

Parámetros físicos del sistema	
Altura máxima $L1_{max} = L2_{max}$	30 [cm]
Diámetro de salida $Dout1 = Dout2$	0.47625[cm]
Voltaje máximo continuo bomba	12[V]
Voltaje máximo peak	22[V]
Constante de bomba Kp	3.3 [$cm^3/s/V$]
Constante gravedad	981 [cm^2/s]

Tabla 4.1: Tabla resumen de parámetros físicos del sistema.

En la tabla 4.1 se muestran las alturas máximas de cada estanque, además de la configuración de los agujeros de salida de líquido para cada uno de los estanques. En específico para este modelo se ocupará un mismo diámetro de salida. Además, de algunas características de la bomba y la consideración de la constante gravitacional en las unidades de medida del sistema.

4.3. Linealización del modelo

Se describen las ecuaciones no lineales del modelo dinámico para el estanque 1 como:

$$A_{t1} \frac{dL_1(t)}{dt} = F_{i1} - F_{o1} \quad (4.108)$$

Donde A_{t1} área del estanque, también entenderemos que F_{i1} y F_{o1} corresponden a los flujos de entrada y salida respectivamente.

En detalle, las ecuaciones dinámicas del estanque 1 quedan expresadas de la forma:

$$\begin{aligned} F_{i1} &= K_p V_p(t) \\ F_{o1} &= A_{o1} V_{o1} \\ V_{o1} &= \sqrt{2gL_1(t)} \end{aligned} \quad (4.109)$$

Donde A_{o1} área del orificio de salida del estanque, V_{o1} velocidad de salida del estanque.

Para linealizar las ecuaciones dinámicas no lineales del estanque 1, se presenta el siguiente método de linealización:

$$\begin{aligned} \sum_{i=1}^n \frac{\partial g}{\partial x_i}(x_1, x_2, \dots, x_n) \Delta x_i &= 0 \\ \Delta x_i &= x_i - x_{iq} \end{aligned} \quad (4.110)$$

En el método de linealización descrito en 4.110, x_i representa una variable dinámica del sistema y Δx_i corresponde a la diferencia en su punto de equilibrio x_{iq} . Esto en general, se entiende como una simplificación matemática del método de aproximación por series de Taylor [17].

Procediendo con el método, de las ecuaciones del estanque 1 en 4.108 se reemplazan 4.109, tal

que las variables dinámicas del sistema quedan expresadas como:

$$\begin{aligned}x_1 &= L_1(t) \\x_2 &= \dot{x}_1(t) = \frac{dL_1(t)}{dt} \\x_3 &= V_p(t)\end{aligned}\tag{4.111}$$

Aplicando el método de linealización de 4.110, el sistema expandido con sus respectivos cambios de variable, queda expresado de la forma:

$$g(x_1, x_2, x_3) = A_{o1}\sqrt{2gx_1} + A_{t1}x_2 - K_px_3 = 0\tag{4.112}$$

Aplicando las derivadas parciales:

$$A_{o1}\frac{\partial g(x_1)}{\partial x_1}\Delta x_1 + A_{t1}\frac{\partial g(x_2)}{\partial x_2}\Delta x_2 - K_p\frac{\partial g(x_3)}{\partial x_3}\Delta x_3 = 0\tag{4.113}$$

Se calculan los valores de las derivadas parciales de 4.113, como:

$$\begin{aligned}\frac{\partial g(x_1)}{\partial x_1} &= \frac{g}{\sqrt{2gL_{1q}}} \\ \frac{\partial g(x_2)}{\partial x_2} &= 1 \\ \frac{\partial g(x_3)}{\partial x_3} &= 1\end{aligned}\tag{4.114}$$

Finalmente reemplazando los valores de las derivadas calculados en 4.114 en el modelo linealizado 4.113, se obtiene el modelo lineal del estanque 1.

$$A_{t1}\frac{d\Delta L_1(t)}{dt} + \frac{A_{o1}g}{\sqrt{2gL_{1q}}}\Delta L_1(t) - K_p\Delta V_p(t) = 0\tag{4.115}$$

Siguiendo la metodología empleada para la linealización del estanque 1, se plantea en primer lugar las ecuaciones dinámicas no lineales del estanque 2 como:

$$A_{t2}\frac{dL_2(t)}{dt} = F_{i2} - F_{o2}\tag{4.116}$$

De 4.116, se definen los flujos de entrada F_{i2} y salida F_{o2} como:

$$\begin{aligned} F_{i2} &= F_{o1} = A_{o1} \sqrt{2gL_1(t)} \\ F_{o2} &= A_{o2} \sqrt{2gL_2(t)} \end{aligned} \quad (4.117)$$

De la ecuación 4.117, el flujo de entrada al estanque 2 es equivalente al flujo de salida del estanque 1, tal como se ha descrito previamente. Esta relación permite comprender la interdependencia de las dinámicas entre los estanques. Establecer esta correlación es imprescindible para el control del estanque 2, el cual se regula en función de las variaciones observadas en el estanque 1.

Ahora, realizando el cambio de variable, se obtiene que:

$$\begin{aligned} x_1 &= L_1(t) \\ x_2 &= L_2(t) \\ x_3 &= \dot{x}_2 = \frac{dL_2(t)}{dt} \end{aligned} \quad (4.118)$$

Donde el sistema extendido, queda expresado como:

$$g(x_1, x_2, x_3) = A_{t2}x_3(t) - A_{o1}\sqrt{2gx_1(t)} + A_{o2}\sqrt{2gx_2(t)} = 0 \quad (4.119)$$

Se calculan las derivadas parciales linealizando la función de 4.119.

$$\frac{\partial g(x_1)}{\partial x_1} \Delta x_1(t) + \frac{\partial g(x_2)}{\partial x_2} \Delta x_2(t) + \frac{\partial g(x_3)}{\partial x_3} \Delta x_3(t) = 0 \quad (4.120)$$

Donde los valores de las derivadas son:

$$\begin{aligned} \frac{\partial x_1}{\partial x_1} &= -\frac{A_{o1}g}{\sqrt{2gL_{1q}}} \\ \frac{\partial x_2}{\partial x_2} &= \frac{A_{o2}g}{\sqrt{2gL_{2q}}} \\ \frac{\partial x_3}{\partial x_3} &= A_{t2} \end{aligned} \quad (4.121)$$

Finalmente reemplazando los valores de las constantes de 4.121, el modelo linealizado del

estanque 2 queda expresado como:

$$A_{t2} \frac{d\Delta L_2(t)}{dt} + \frac{A_{o2}g}{\sqrt{2gL_{2q}}} \Delta L_2(t) - \frac{A_{o1}g}{\sqrt{2gL_{1q}}} \Delta L_1(t) = 0 \quad (4.122)$$

4.4. Sistema en variable de estados

Es de utilidad para el diseño de sistemas de control expresar las ecuaciones en variables de estado. Esta representación matricial, permite ver las relaciones dinámicas de los estados, entradas y salidas. Además, permite calcular características primordiales como controlabilidad, observabilidad, etc. [18]. Para conformar el sistema en variable de estados, se utilizan las ecuaciones de 4.115 y 4.122, tal que despejando las derivadas respectivas de $\dot{L}_1(t)$ y $\dot{L}_2(t)$, se tiene que:

$$\begin{aligned} \frac{d\Delta L_1(t)}{dt} &= \frac{-A_{o1}g}{A_{t1}\sqrt{2gL_{1q}}} \Delta L_1(t) + \frac{K_p}{A_{t1}} \Delta V_p(t) \\ \frac{d\Delta L_2(t)}{dt} &= \frac{A_{o1}g}{A_{t2}\sqrt{2gL_{1q}}} \Delta L_1(t) - \frac{A_{o2}g}{A_{t2}\sqrt{2gL_{2q}}} \Delta L_2(t) \end{aligned} \quad (4.123)$$

De esta forma, los estados del sistema corresponden a las alturas $L_1(t)$ y $L_2(t)$ y la entrada corresponde al voltaje del único actuador $V_p(t)$. De esta forma se reescribe el sistema en variables de estado, como en la ecuación 4.124.

$$\underbrace{\begin{bmatrix} \frac{d\Delta L_1(t)}{dt} \\ \frac{d\Delta L_2(t)}{dt} \end{bmatrix}}_{\dot{x}(t)} = \underbrace{\begin{bmatrix} \frac{-A_{o1}g}{A_{t1}\sqrt{2gL_{1q}}} & 0 \\ \frac{A_{o1}g}{A_{t2}\sqrt{2gL_{1q}}} & \frac{-A_{o2}g}{A_{t2}\sqrt{2gL_{2q}}} \end{bmatrix}}_{A_c} \underbrace{\begin{bmatrix} \Delta L_1(t) \\ \Delta L_2(t) \end{bmatrix}}_{x(t)} + \underbrace{\begin{bmatrix} \frac{K_p}{A_{t1}} \\ 0 \end{bmatrix}}_{B_c} \underbrace{\Delta V_p(t)}_{u(t)} \quad (4.124)$$

Siendo la salida:

$$y(t) = \underbrace{\begin{bmatrix} 0 & 1 \end{bmatrix}}_{C_c} \begin{bmatrix} \Delta L_1(t) \\ \Delta L_2(t) \end{bmatrix} \quad (4.125)$$

Llevando el sistema a su punto de operación, es decir, haciendo tender las dinámicas $\dot{L}_1(t) = 0$

y $\dot{L}_2(t) = 0$ de la ecuación 4.124, se obtiene que el punto de operación queda descrito como:

$$L_{1q} = \left(\frac{A_{o2}}{A_{o1}} \right)^2 L_{2q} \quad (4.126)$$

De 4.126 se calcula el punto de operación del estanque L_{1q} en base a un punto definido del estanque 2 (L_{2q}) debido a que este es el estanque a controlar.

4.5. Modelado en tiempo discreto

El control digital implica la implementación de sistemas de control utilizando dispositivos electrónicos para procesar señales digitales, como en el caso de los computadores. Este proceso requiere muestrear las señales, es decir, recoger muestras en intervalos regulares para aproximar la señal continua original. En base a este principio, el control por computador se apoya en dos componentes principales: el conversor analógico-digital (A-D) y el conversor digital-analógico (D-A).

El conversor A-D muestrea las señales continuas en tiempos regulares, permitiendo la discretización de señales. Por otro lado, el conversor D-A se encarga de reconstruir la señal analógica a partir de los datos digitales muestreados. Este proceso suele emplear un retentor de orden cero (ZOH), que mantiene cada muestra constante durante un intervalo de tiempo, para la reconstrucción de las señales continuas.

El esquema más común de control digital se presenta en la siguiente figura:

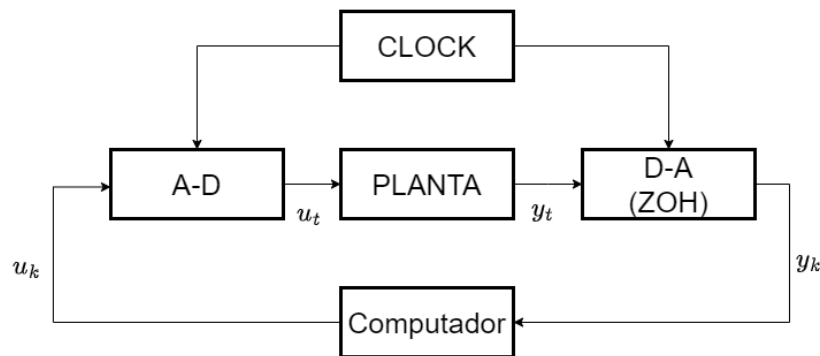


Figura 4.8: Esquema de control digital

Para obtener la discretización de la planta respectiva los estanques acoplados, definimos el sistema discreto genérico como:

$$\begin{aligned}x_{k+1} &= A_d x_k + B_d u_k \\ y_k &= C_d x_k\end{aligned}\tag{4.127}$$

Las ecuaciones de estado 4.124 y 4.125, corresponden a la representación continua del modelo de los estanques.

En el libro [19], plantea que para un conversor A-D y un conversor D-A perfectamente sincronizados, es posible obtener las entradas y salidas en un mismo instante t_k , con esto el sistema sampleado queda de la forma:

$$x(t_k + 1) = \Phi(t_{k+1}, t_k)x(t_k) + \Gamma(t_{k+1}, t_k)u(t_k)\tag{4.128}$$

$$y(t_k) = Cx(t_k) + Du(t_k)\tag{4.129}$$

De esta manera, podemos decir que $\Phi = A_d$ y que $\Gamma = B_d$, donde C no se verá afectada ya que es igual para las salidas y D para sistemas causales es nula.

Para obtener las matrices discretizadas, se plantean las siguientes relaciones:

$$A_d = \mathcal{L}^{-1}\{(sI - A_c)^{-1}\}\Big|_{t=T_s} = e^{A_c T_s}\tag{4.130}$$

$$B_d = \int_0^{T_s} A_d(\tau) d\tau \cdot B_c\tag{4.131}$$

Donde se utilizan la transformada inversa de Laplace y se evalúa en un instante T_s el cual corresponde al tiempo de muestreo.

Para el modelo se consideran las siguientes consideraciones:

$$\begin{aligned}A_{o1} &= A_{o2} \\ A_{t1} &= A_{t2} \\ L_{1q} &= \left(\frac{A_{o1}}{A_{o2}}\right)^2 L_{2q} \Rightarrow L_{1q} = L_{2q}\end{aligned}\tag{4.132}$$

Considerando las aberturas de salida de igual tamaño y que las áreas de los estanques son

iguales, se obtienen que los puntos de equilibrio de los estanques también serán iguales.

Para calcular A_d de la ecuación 4.130 y considerando 4.132, en el modelo lineal presentado en 4.124, se obtiene una equivalencia del sistema, de tal forma de que se simplifican las constantes como:

$$A_c = \begin{bmatrix} -\gamma & 0 \\ \gamma & -\gamma \end{bmatrix} \quad (4.133)$$

De donde:

$$\gamma = \frac{A_{o1}g}{A_{t1}\sqrt{2gL_{1q}}} \quad (4.134)$$

Con lo anterior, se desarrolla 4.130, llevándolo en una primera instancia al dominio de Laplace como:

$$(sI - A_c) = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} -\gamma & 0 \\ \gamma & -\gamma \end{bmatrix} = \begin{bmatrix} s + \gamma & 0 \\ -\gamma & s + \gamma \end{bmatrix} \quad (4.135)$$

Para obtener la inversa de una matriz 2x2 definida como:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad (4.136)$$

Cuya matriz inversa queda definida como:

$$A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \quad (4.137)$$

Reemplazando los valores en el dominio de Laplace, se tiene que:

$$(sI - A_c)^{-1} = \frac{1}{(s + \gamma)^2} \begin{bmatrix} s + \gamma & 0 \\ \gamma & s + \gamma \end{bmatrix} = \begin{bmatrix} \frac{1}{(s + \gamma)} & 0 \\ \frac{\gamma}{(s + \gamma)^2} & \frac{1}{(s + \gamma)} \end{bmatrix} \quad (4.138)$$

Para la matriz 4.138, se debe aplicar la transformada de Laplace inversa para llevar al dominio temporal. De las transformaciones inversas más comúnmente utilizadas se presentan de la forma:

$$\mathcal{L}^{-1} \left\{ \frac{1}{s + a} \right\} = e^{-at} \quad (4.139)$$

$$\mathcal{L}^{-1} \left\{ \frac{1}{(s+a)^2} \right\} = t \cdot e^{-at} \quad (4.140)$$

Teniendo las transformaciones inversas ya calculadas, se reemplaza en 4.130 y se evalúa en el instante T_s , tal que la matriz A_d discretizada queda de la forma:

$$A_d = \begin{bmatrix} e^{-\gamma T_s} & 0 \\ \gamma \cdot T_s \cdot e^{-\gamma T_s} & e^{-\gamma T_s} \end{bmatrix} \quad (4.141)$$

Por otro lado, para calcular la matriz B_d con la ecuación presentada en 4.131. A partir de A_d previamente calculada en 4.141 y B_c respectiva a la matriz de entradas del sistema.

$$B_d = \int_0^T \underbrace{\begin{bmatrix} e^{-\gamma \tau} & 0 \\ \gamma \cdot \tau \cdot e^{-\gamma \tau} & e^{-\gamma \tau} \end{bmatrix}}_{A_d} d\tau \cdot \underbrace{\begin{bmatrix} \frac{K_p}{A_t} \\ 0 \end{bmatrix}}_{B_c} \quad (4.142)$$

De la ecuación 4.142, se observa que solo se necesitan calcular 2 integrales las cuales renombraremos como B_{d1} y B_{d2} . Desarrollando las integrales y evaluando en el tiempo de muestreo:

$$B_{d1} = \int_0^{T_s} e^{-\gamma \tau} d\tau = \frac{-1}{\gamma} (e^{-\gamma T_s} - e^0) = \frac{1}{\gamma} (1 - e^{-\gamma T_s}) \quad (4.143)$$

$$B_{d2} = \int_0^{T_s} \gamma \cdot \tau \cdot e^{-\gamma \tau} d\tau = \gamma \cdot \left(\frac{1 - \gamma e^{-\gamma T_s} - e^{-\gamma T_s}}{\gamma^2} \right) = \frac{1}{\gamma} (1 - \gamma e^{-\gamma T_s} - e^{-\gamma T_s}) \quad (4.144)$$

Para simplificar, en primer lugar se reemplaza en las matrices:

$$B_d = \begin{bmatrix} B_{d1} & 0 \\ B_{d2} & B_{d1} \end{bmatrix} \cdot \begin{bmatrix} \frac{K_p}{A_t} \\ 0 \end{bmatrix} \quad (4.145)$$

Donde multiplicando, finalmente se obtiene B_d como una matriz 2x1, tal como:

$$B_d = \begin{bmatrix} \frac{K_p}{A_t} \cdot \frac{1}{\gamma} (1 - e^{-\gamma T_s}) \\ \frac{K_p}{A_t} \cdot \frac{1}{\gamma} (1 - \gamma e^{-\gamma T_s} - e^{-\gamma T_s}) \end{bmatrix} \quad (4.146)$$

Finalmente, reemplazando numéricamente con los valores de la planta presentados anterior-

mente en 4.2, los valores de $\gamma = 0.0658$, $A_t = 15.5179$ y $K_p = 3.3$, en las ecuaciones 4.141, 4.146, para un tiempo de muestreo de $T_s = 1$, se obtienen los siguientes valores:

$$A_d = \begin{bmatrix} 0.9365 & 0 \\ 0.0615 & 0.9365 \end{bmatrix} \quad (4.147)$$

$$B_d = \begin{bmatrix} 0.2058 \\ 0.0067 \end{bmatrix} \quad (4.148)$$

$$C = \begin{bmatrix} 0 & 1 \end{bmatrix} \quad (4.149)$$

Cabe destacar que según sea el tiempo de muestro T_s , estos valores van a ir cambiando.

5. Implementación

5.1. MATLAB - Simulink

MATLAB es un software de programación de alto nivel frecuentemente utilizado para cálculo numérico y análisis de datos. En específico su extensión de Simulink permite mediante la programación de bloques, el modelado y simulación de sistemas dinámicos.

Esto resulta bastante útil para implementación de sistemas de control, debido a que se puede configurar rápidamente distintos enfoques, ya sea con nuevos programas, nuevos bloques, optimizar diversos parámetros, etc.

En específico para la implementación del MPC algunos de los bloques a utilizar son los siguientes:

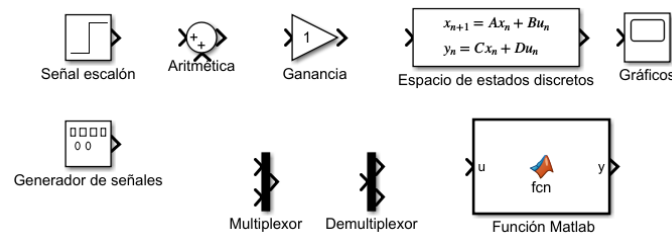


Figura 5.9: Bloques de Simulink para MPC

Los bloques presentados en la figura 5.9 se desglosan a continuación:

- **Discrete States-Space:** Permite simular un sistema en tiempo discreto, en forma de variables de estado, lo que resulta útil por como se diseña el sistema.
- **MATLAB Function:** El bloque de funciones de MATLAB, permite programar funciones más complejas con el mismo lenguaje de MATLAB. Por ejemplo, en las secciones posteriores se utiliza para implementar los algoritmos de optimización y sus restricciones.
- **Aritmética:** Algunos bloques de aritmética simple como sumas y multiplicaciones son altamente efectivos para el modelado de sistemas dinámicos.
- **MUX - DEMUX:** Componentes para multiplexar y demultiplexar son útiles para trabajar con distintas señales.

- **Scope - Gráficas:** Para visualización de señales.

5.1.1. Diagrama de bloques MPC en MATLAB

La implementación del sistema de control predictivo en MATLAB - Simulink para los estanques acoplados, se presenta en la siguiente figura:

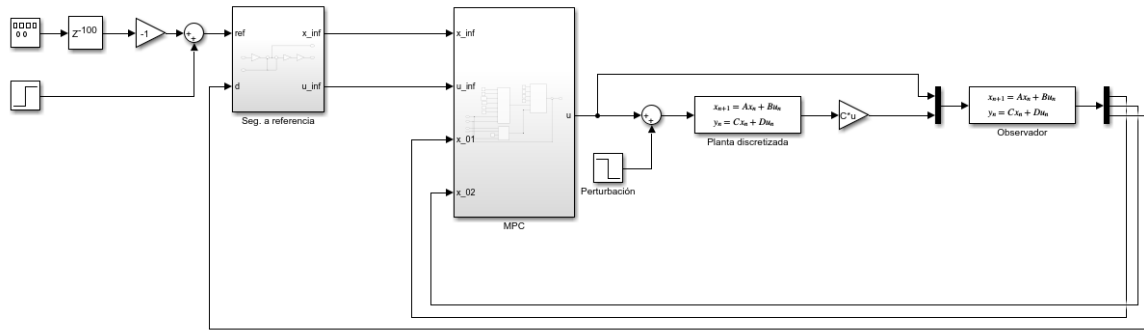


Figura 5.10: Diagrama de bloques MPC

De la figura 5.10 se muestra el esquema MPC genérico aplicado en esta memoria, compuesto por los bloques respectivos al seguimiento a referencia, el controlador MPC, la planta y el observador de estados y perturbaciones.

El primero de ellos, el bloque de seguimiento a referencia se detalla a continuación:

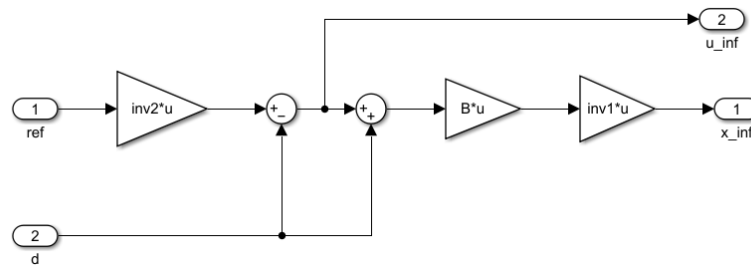


Figura 5.11: Bloque de seguimiento a referencia

De la figura 5.11, se tienen como entradas la referencia y la perturbación estimada denotada como d . Mientras que sus salidas corresponden a los estados y actuaciones finales x_{∞} y u_{∞} .

El segundo bloque corresponde al controlador MPC, el cual se presenta a continuación:

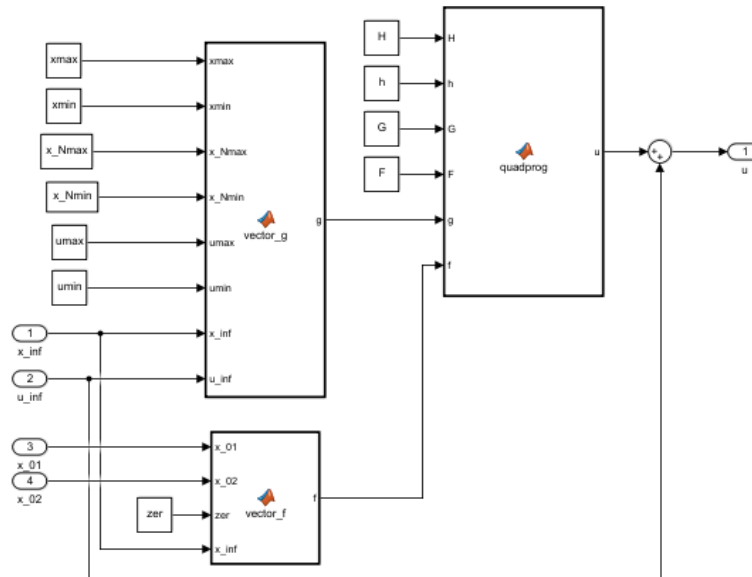


Figura 5.12: Bloque MPC

El bloque MPC, presentado en la Figura 5.12, integra funciones de restricción y el bloque de optimización. Las entradas de este bloque corresponden a los estados y actuaciones finales x_{inf} , u_{inf} como los estados estimados x_{obs1} y x_{obs2} . El bloque de restricciones se divide en dos funciones principales: "vector_g" para las restricciones de desigualdad, y "vector_f" para las restricciones de igualdad. Los códigos para estas funciones se describen en las figuras 5.13 y 5.14, respectivamente. Estas funciones, junto con las matrices utilizadas para la formulación cuadrática del problema de control, las cuales son constantes y se muestran en el código 9.1, constituyen las entradas del bloque de optimización.

A continuación, se presentan los códigos de las funciones para las restricciones mencionadas anteriormente, como:

```
1  %%%%%%%%%%%%% Restricciones de Desigualdad %%%%%%%%%%%%%
function g = vector_g(xmax,xmin,x_Nmax,x_Nmin,umax,umin,x_inf,u_inf)

    dx=[xmax-x_inf;-xmin+x_inf];
    du=[umax-u_inf;-umin+u_inf];
6  dN=[x_Nmax-x_inf;-x_Nmin+x_inf];
    g=[kron(ones(10,1),[dx;du]);dN];
    end
```

Figura 5.13: Código de restricciones desigualdad g.

```

2 %%%%%%%%%%%%% Restricciones de Igualdad %%%%%%%%%%%%%
function f = vector_f(x_01,x_02,zer,x_inf)
x0 = [x_01;x_02]-x_inf;
f=[-x0;zer];
end

```

Figura 5.14: Código de restricciones de igualdad f.

El proceso de optimización se puede llevar a cabo mediante dos enfoques: el uso de multiplicadores con direcciones alternadas de Lagrange (ADMM) o el comando 'Quadprog', empleando específicamente el algoritmo 'Active-set'. Mientras que el código para el ADMM se adjunta en los anexos (9.2), el código para 'Quadprog' se muestra a continuación en la Figura 5.15.

```

5 %%%%%%%%%%%%% Optimizacion Quadprog %%%%%%%%%%%%%
function u = quadprog(H,h,G,F,g,f)
options = optimoptions('quadprog','Algorithm','active-set');
x = quadprog(H,h,G,g,F,f,[],[],zeros(32,1),options);
u = x(3);
end

```

Figura 5.15: Código para solver Quadprog - "Active-set" MATLAB

Cabe destacar que, anteriormente, se explica que para tener seguimiento a referencia es necesario de sumarle el parámetro u_{∞} a la actuación calculada en el bloque de optimización, lo cual para mantener la unicidad de los códigos se realizan fuera del bloque como una operación aritmética.

5.2. Simulink - Quarc

Quanser es una empresa canadiense dedicada al suministro de software y Hardware, diseñados principalmente para el ámbito académico e investigativo, en áreas de control automático, robótica, mecatrónica, etc.

Uno de los productos que proporciona es el software Quarc, el cual está diseñado para combinar el modelado matemático de Simulink con aplicaciones prácticas. De esta manera, siendo posible testear diversos problemas de control y obtener datos en tiempo real, aprovechando al máximo el entorno de gráfico y la alta capacidad de procesamiento que provee MATLAB-Simulink.

En esta memoria, se da un breve contexto para las pruebas con este software, debido a que está principalmente enfocada a la implementación con MicroLabBox de dSPACE, sin embargo,

en una primera instancia permite corroborar el funcionamiento de la aplicación y dar una comparativa general entre ambos equipos.

5.2.1. Instrumentación para pruebas Quanser

Los instrumentos son especificados en el manual de usuario de los estanques [20] proporcionado por Quanser.

Para entender la conexión general de la instrumentación se presenta el siguiente esquema:

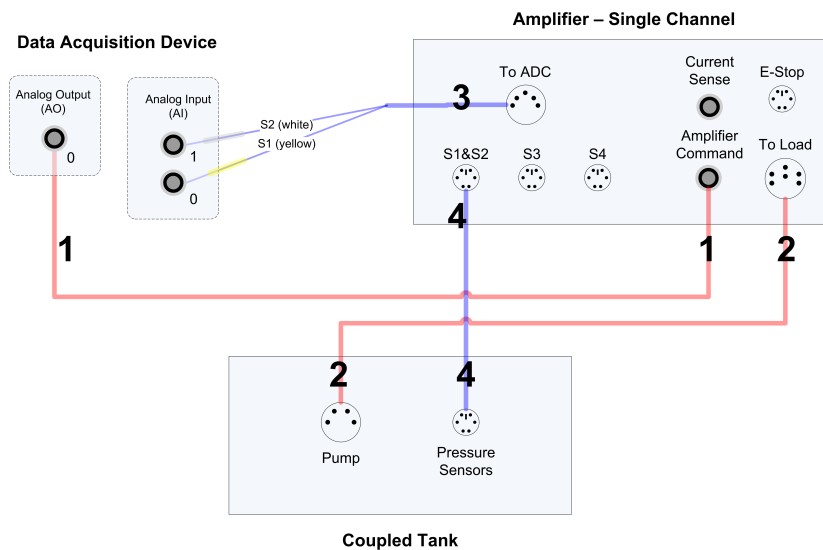


Figura 5.16: Esquema de conexión instrumentación Quanser

Del esquema de la figura 5.16 se destacan 3 componentes principales, los cuales corresponde al amplificador, el dispositivo de adquisición de datos y la planta.

El primer componente es el amplificador 'VoltPAQ-x1' de Quanser, cuya función es recibir y amplificar la señal de control que viene desde el computador. Esta señal amplificada es necesaria para activar la bomba. Adicionalmente, el amplificador suministra energía a los sensores de presión encargados de medir los niveles de líquido, y posteriormente tomar estos datos y enviarlos al ADC mediante el borne etiquetado como 'to ADC'.



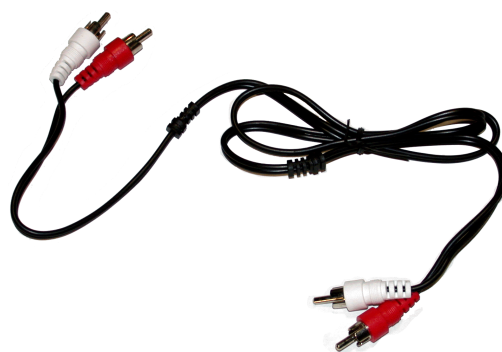
Figura 5.17: Amplificador 'VoltPAQ-x1' Quanser

Cabe destacar que en la figura 5.17, se muestra un switch respectivo a la ganancia de la señal de amplificación, esto es de particular importancia porque decide cuál será finalmente el voltaje entregado a la bomba.

Por otra parte, los cables de mediciones correspondientes a la altura de los estanques en los bornes 'To ADC', además, la señal de comando o actuación en el borne de 'Amplifier Command', se presentan dichas conexiones a continuación:



(a) Cable a adquisición de datos 5pin - RCA



(b) Cable de señal de comando (Actuación)

Figura 5.18: Cables de medición - actuación estanques

El cable de la adquisición del ADC corresponde a un cable de 5 pines a RCA, mientras que la señal de actuación se conecta directamente al tipo RCA.

Por otro lado, los cables de alimentación, tanto de los sensores como de la carga se describen a continuación:

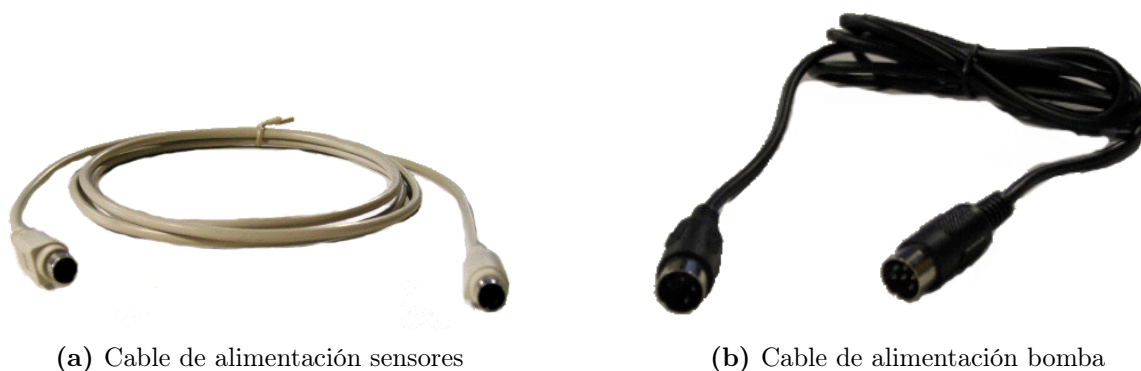


Figura 5.19: Cables de alimentación sensores - bomba

En una primera instancia, se utiliza el dispositivo de adquisición de datos 'Q2-USB' de Quanser, este se compone de un dispositivo DAC - ADC y un Encoder para el envío y medición de señales, el cual se presenta en la siguiente figura:



Figura 5.20: Dispositivo de adquisición de datos Quanser

Este dispositivo va conectado vía USB al computador, de tal manera de enviar y recibir tanto las señales de control a escribir a la planta como las mediciones de los estanques para procesar en el computador.

5.2.2. Diagrama de bloques MPC Quanser

El diagrama utilizado para testear la aplicación MPC anteriormente diseñada con el software Quarc, se presenta a continuación:

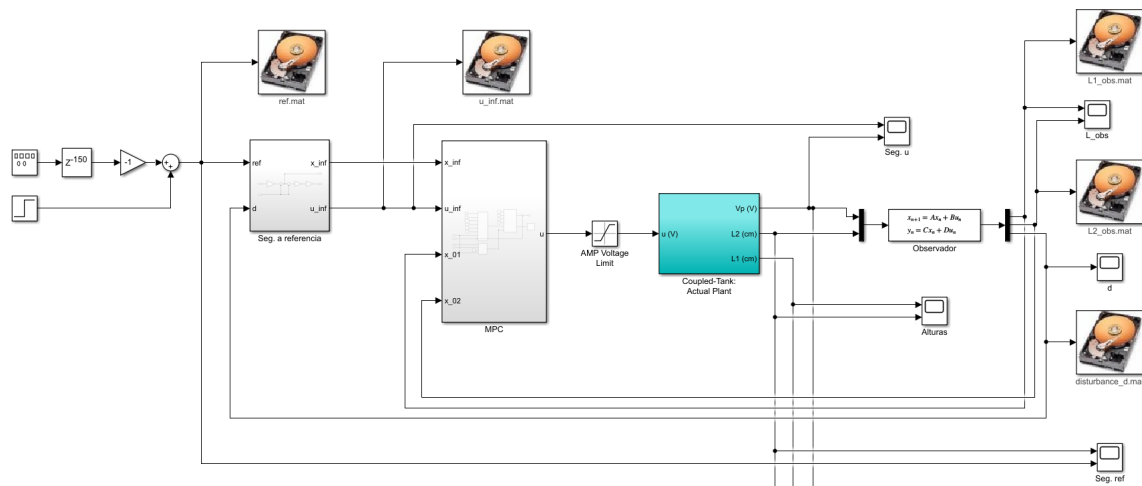


Figura 5.21: Diagrama de bloques esquema QUARC

El bloque correspondiente a la planta, es reemplazado por las funciones de Quarc, para escribir y leer datos del sistema físico, como se presenta a continuación:

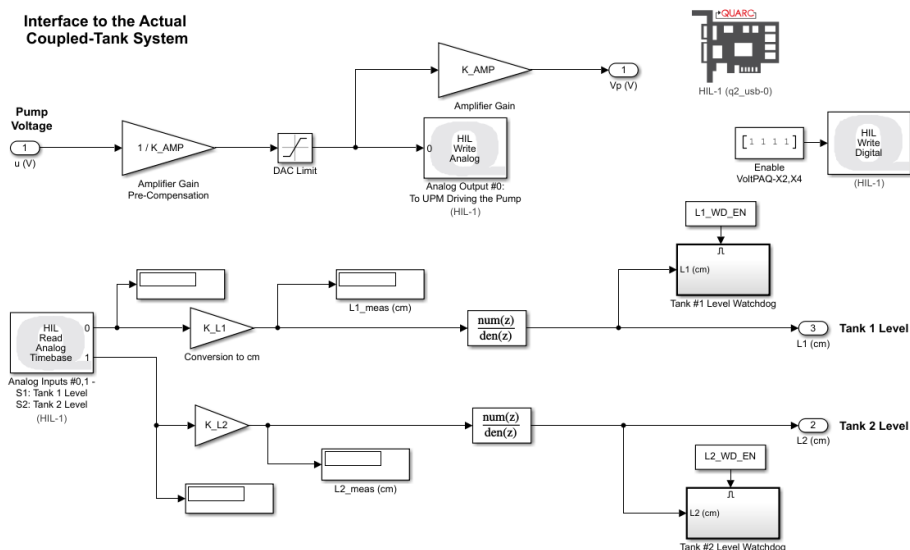


Figura 5.22: Diagrama de bloques planta estanques acoplados QUARC

Cabe destacar que, los bloques anteriormente utilizados son obtenidos de los programas de prueba de Quanser, donde el único requisito adicional fue discretizar la función de filtro de la lectura respectiva a las alturas de los estanques.

5.3. Implementación MicroLabBox dSPACE

dSPACE es una empresa fundada en Alemania, encargada de suministrar Software y Hardware de la más alta tecnología para simulación y validación de sistemas de control, en diversos ámbitos industriales tales como, la industria automotriz, aeroespacial, generación energética, maquinaria industrial, ferroviaria, etc.

Dentro de este contexto se desarrolla MicroLabBox, la cual corresponde a un sistema de prototipado en laboratorio, teniendo como principales ventajas un tamaño reducido, de bajo costo y capaz de entregar una alta performance y versatilidad de aplicaciones.

Esta unidad está compuesta por un bloque de procesamiento de alta potencia combinando con una FPGA que permite una comunicación en tiempo real de los sistemas simulados y físicos con una baja latencia y flexibilidad para probar sistemas extremadamente rápidos como, por ejemplo, el control de motores, cancelación de ruido, cancelación de vibración, etc. En la

siguiente figura se presenta el diagrama de bloques operacional de MicroLabBox.

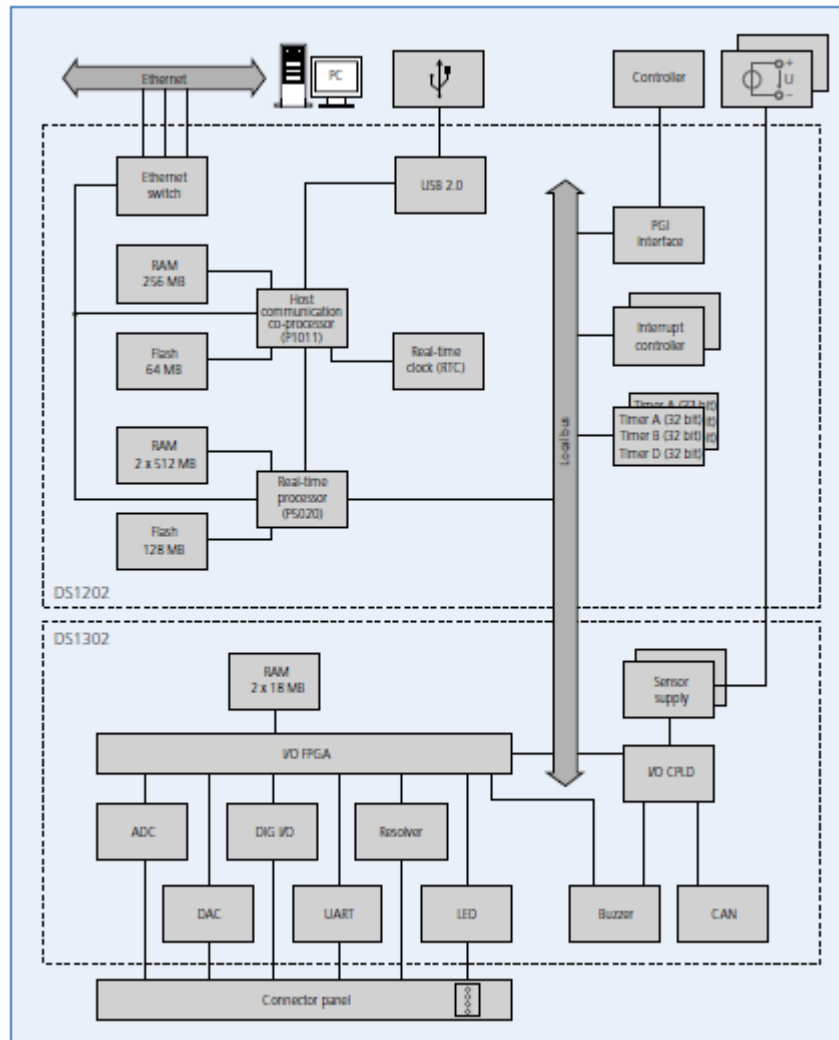


Figura 5.23: Diagrama de bloques MicroLabBox dSPACE [2]

La Figura 5.23 muestra la configuración de de MicroLabBox desde un computador que, mediante una conexión de cable Ethernet, carga los programas previamente diseñados. Para el trabajo presentado en esta memoria se implementa desde el software de MATLAB.

5.3.1. Software para implementación en dSPACE

A continuación, se describe el software utilizado para la implementación de proyectos en MicroLabBox:

- Programa Model Vitis Composer:** Para cargar el archivo desde MATLAB-Simulink, se utiliza el software "Model Vitis Composer" [21], el cual corresponde a una herramienta de **AMD-Xilinx** encargada de facilitar la exploración de los programas de MathWorks, tal como lo son MATLAB y Simulink, de tal forma de facilitar la implementación y optimización del código a generar para el RTI.
- RTI:** La interfaz de tiempo real "RTI", corresponde a un software de extensión para Simulink desarrollado por dSPACE, al cual solo se puede acceder a través del programa Model Vitis Composer y que tiene como finalidad habilitar todos los bloques que permiten la interacción con MicroLabBox como lo son el ADC, DAC, etc. Se presentan algunos de estos en la siguiente figura:

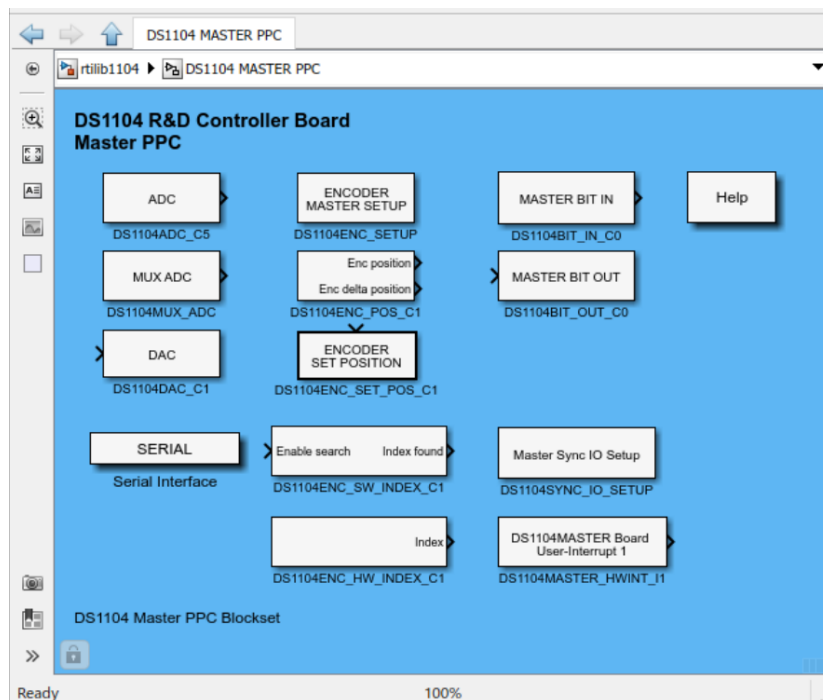


Figura 5.24: Bloques RTI [3]

- Control Desk:** El centro de control "Control Desk" corresponde al software desarrollado

por dSPACE, para interactuar con los programas en tiempo real cargados al dispositivo MicroLabBox. En el cual, es posible configurar aspectos de la simulación, monitorear y modificar los sistemas, dando una forma interactiva de disposición e interacción directa con una suite variada de instrumentos, siendo entonces un símil a una interfaz HMI y que se puede visualizar a continuación:



Figura 5.25: Control desk dSPACE [4]

5.3.2. Implementación de estanques acoplados en MicroLabBox

Se aconseja iniciar con una etapa de familiarización con el software y hardware de MicroLabBox, testeando inicialmente con el envío y recepción de señales simples. Para facilitar este proceso, se recomienda consultar el vídeo tutorial realizado por Michel Azua [22], el cual sirve de inspiración y resume los pasos para la implementación del proyecto de estanques acoplados, detallados en esta sección.

Una vez entendiendo los lineamientos generales de la aplicación en el dispositivo, se presenta una guía específica con los pasos a seguir, consideraciones y algunos problemas comunes encontrados en la realización del proyecto. Entre las primeras consideraciones es necesario que MicroLabBox esté conectado al computador por un cable Ethernet y también encendido antes de continuar con los siguientes pasos.

5.3.3. Generación de un archivo ".sdf"

Para compilar un proyecto de MATLAB para su uso en MicroLabBox, es esencial comprender primero el proceso general de implementación. En grandes rasgos, este proceso consiste en convertir la programación realizada con bloques en Simulink a un código que el dispositivo (MicroLabBox) pueda interpretar. El archivo resultante de esta compilación se denomina con la extensión ".sdf". A continuación, se describe detalladamente la creación de este archivo.

- **Abrir MATLAB desde Model Vitis Composer:**

Al utilizar Model Vitis Composer, se inicia MATLAB automáticamente. En este momento, se deben activar las librerías correspondientes de dSPACE. Normalmente, se presenta una ventana emergente con diferentes modelos de dSPACE, siendo para este proyecto el modelo DS1202.

- **Crear el archivo Simulink para RTI1202:**

Una vez cargadas las librerías de dSPACE, se debe abrir el archivo de Simulink que contiene el sistema previamente diseñado. Es aconsejable mantener una carpeta exclusiva para todo el proyecto. Se inicia un nuevo archivo en Simulink con la extensión RTI1202 y se transfiere el contenido del proyecto existente a este nuevo entorno, asegurando que se adapte al contexto del RTI 1202, en específico aparecerá un bloque llamado "RTI Data".

- **Compilar el archivo ".sdf":**

El nuevo archivo de Simulink, ahora con la extensión RTI, debe configurarse para utilizar los periféricos específicos de MicroLabBox, como el DAC y el ADC. A continuación, se muestra el esquema implementado para el proyecto de los estanques acoplados en la Figura 5.26:

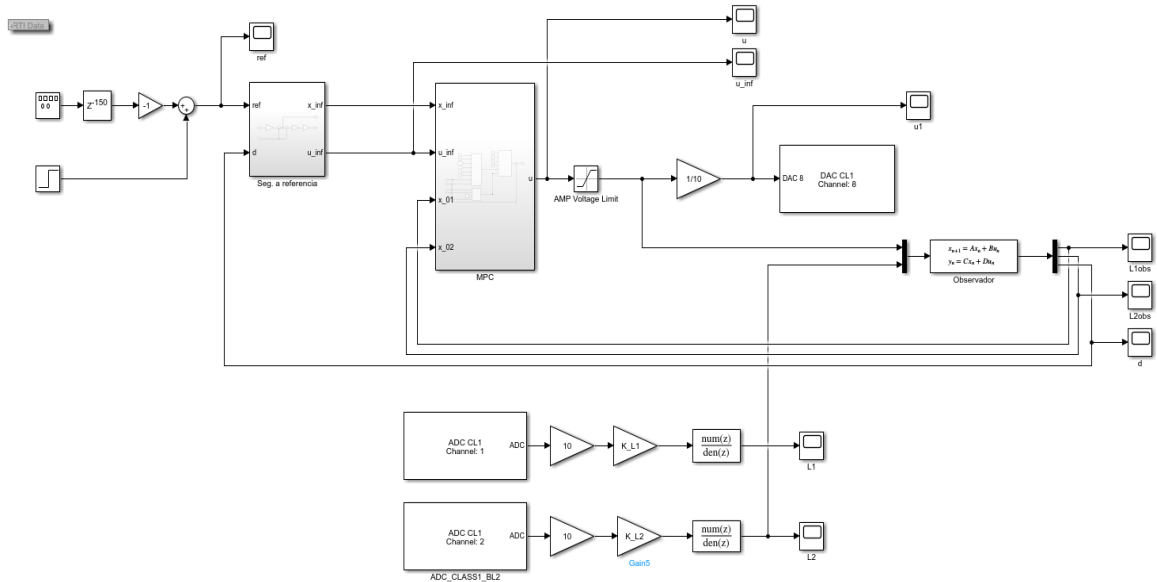


Figura 5.26: Diagrama de bloques MPC en MicroLabBox

En la figura 5.26, particularmente en la esquina superior izquierda, se destaca el bloque "RTI Data", el cual proporciona la interfaz de trabajo para MicroLabBox. Después de asegurar que las variables necesarias estén disponibles en el Workspace y que el directorio actual sea el de la carpeta del proyecto, la compilación se inicia con la combinación de teclas "CTRL + B", lo que resulta en la generación del archivo ".sdf" y otros archivos asociados, como se muestra en la Figura 5.27:

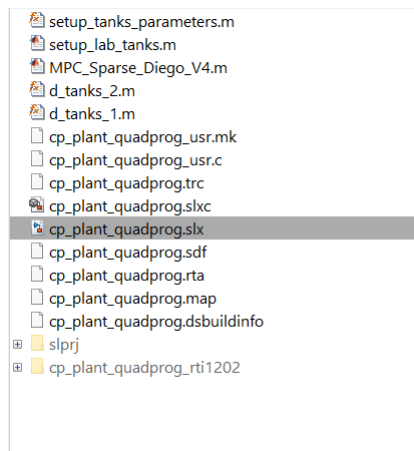


Figura 5.27: Archivos generados para el proyecto RTI

Se recomienda encarecidamente revisar la sección de consideraciones y problemas comu-

nes 5.3.7, donde se detallan algunos procedimientos importantes de verificar.

5.3.4. Creación de un proyecto en Control Desk

Una vez creado el archivo de extensión ".sdf", se necesita crear un proyecto en el software Control Desk, este programa tal como se menciona anteriormente, es el encargado de establecer la comunicación computador - MicroLabBox cargando los programas y también habilita la suite de interacción y visualización del dispositivo.

Para iniciar un proyecto en Control Desk, se debe acceder a la pestaña de proyectos ubicada en la esquina superior izquierda de la interfaz. Aunque cabe destacar que en este caso se continuó con un proyecto ya existente, es importante comprender que la creación de un nuevo proyecto implica la definición de parámetros específicos para el funcionamiento óptimo de la tarjeta, incluyendo aspectos como la latencia y la frecuencia del reloj. Por esta razón, se decidió proseguir con el proyecto previamente creado. No obstante, al iniciar un nuevo experimento, se ofrece la posibilidad de generar archivos desde cero sin necesidad de reutilizar información previa, para más información del marco de trabajo (Framework) se debe visitar el siguiente instructivo [23].

Una vez abierta la pestaña, se procede a crear un nuevo experimento. Es recomendable cambiar el nombre para facilitar su identificación. A continuación, se debe seleccionar la tarjeta que se utilizará, en este caso, la placa MicroLabBox 1202. Es importante asegurarse de que la placa esté encendida antes de proceder con la creación del proyecto. Finalmente, una vez se tiene creado el proyecto es necesario importar el archivo ".sdf" desde la carpeta del proyecto.

Una vez se tiene el proyecto creado, en la parte izquierda del software se encontrará el siguiente listado de información:

Group	Description
All Variable Descriptions	
Calculated Variables	
cp_plant_quadprog.sdf	
Task Info	
Tunable Parameters	
State Machine Data	
Dynamic Variables	
XIL API	
Model Root	
Observador	
Gain1	
Gain2	
Gain3	
Gain4	
Gain5	
Gain6	
AMP Voltage Limit	
SignalInGenerator	

Figura 5.28: Descripción de las señales den Control Desk

De la figura 5.28, se presentan como obtener las señales del archivo ".sdf", en específico, en la pestaña de "Model Root", se muestran todas las señales del archivo con el mismo nombre que tenían en Simulink. Dichas señales, se pueden arrastrar a la ventana principal o unir con algún instrumento, de tal manera de crear un HMI personalizado, según sean las señales que se requieren mostrar. A continuación, se presenta el modelo de HMI creado para testear esta memoria.

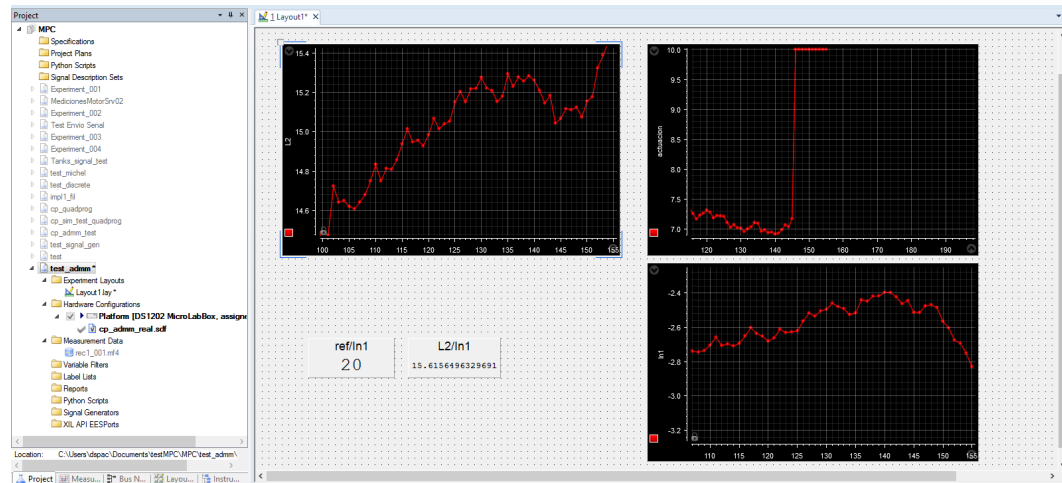


Figura 5.29: Interfaz de visualización RTI Control Desk Estanques acoplados

En la Figura 5.29, se observa la interfaz de visualización, donde se monitorea la altura del estanque 2, $L_2(t)$, la señal de actuación, $u(t)$, y la perturbación estimada, $d(t)$. Todas las señales

mencionadas se presentan en un gráfico de tiempo denominado "Time-Plotter". Además, hay un display que muestra la referencia y la altura medida en el instante actual del estanque, L_2 . Cabe destacar que el análisis se realiza justo en el instante de cambio en la referencia, momento en el cual los datos están cambiando su dinámica.

5.3.5. Toma de datos en Control Desk

Para tomar datos en control Desk, es necesario habilitar la opción de "Recorder". En la ventana inferior izquierda de la figura 5.29, se observa la segunda pestaña llamada "Measurements", en la cual se habilitan las opciones para grabar datos.

Para grabar los datos, se habilita la opción de "Recorders", tal que, con el click derecho se activarán las opciones y aparecerá el siguiente display de opciones:

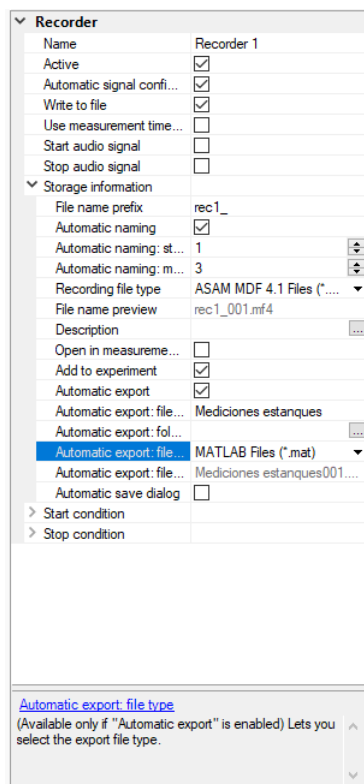


Figura 5.30: Configuración "Recorder" para grabación de datos Control Desk

En la figura 5.30, se presentan las configuraciones para el grabado de datos, donde es necesario destacar lo siguiente:

- Activar la opción "Automatic export", que habilita la configuración para la exportación automática de archivos.
- Justo debajo de la opción anterior, introducir un nombre para el archivo en el campo designado.
- Utilizar la opción "Automatic export folder" para seleccionar la carpeta donde se desea guardar los datos exportados.
- Por último, en la opción "Automatic export file extension", configurar la extensión del archivo de exportación. Se sugiere utilizar la extensión de MATLAB (.mat) para facilitar el manejo posterior de los datos.

5.3.6. Instrumentación para pruebas en MicroLabBox dSPACE

La instrumentación utilizada en esta memoria se presenta a continuación:



Figura 5.31: MicroLabBox 1202

En la figura 5.31, se presenta el dispositivo MicroLabBox de la empresa dSPACE, en específico se utiliza la tarjeta 1202 cuyas conexiones son mediante cables del tipo BNC.

Como se presenta en las secciones anteriores, los cables que utilizan los sensores de Quanser son del tipo RCA, por lo que para conectarse con dSPACE se necesita el siguiente adaptador RCA-BNC.



Figura 5.32: Adaptador RCA-BNC para medición de Alturas

Principalmente para lo que corresponde a los sensores de medición de altura de los estanques se utilizan estos adaptadores 5.32. Finalmente, el cable encargado de la actuación de control, este al ser una salida desde MicroLabBox, corresponde a un cable BNC-RCA, el cual se presenta en la figura 5.33.



Figura 5.33: Cable para señal de control BNC-RCA

5.3.7. Consideraciones y problemas comunes con la implementación en MicroLabBox

Algunas de las consideraciones importantes para la implementación en MicroLabBox desde Simulink se presentan a continuación:

- Es necesario que los archivos ".mat" del proyecto en Simulink estén guardados en una misma carpeta y previamente cargados al Workspace, es por esto que se recomienda tener una carpeta individual para el proyecto en cuestión.
- Es fundamental, especificar correctamente el Solver a utilizar en Simulink al momento de crear el proyecto RTI1202, ya que este proyecto viene con las configuraciones por defecto. A continuación, se presenta el solver utilizado para este proyecto:

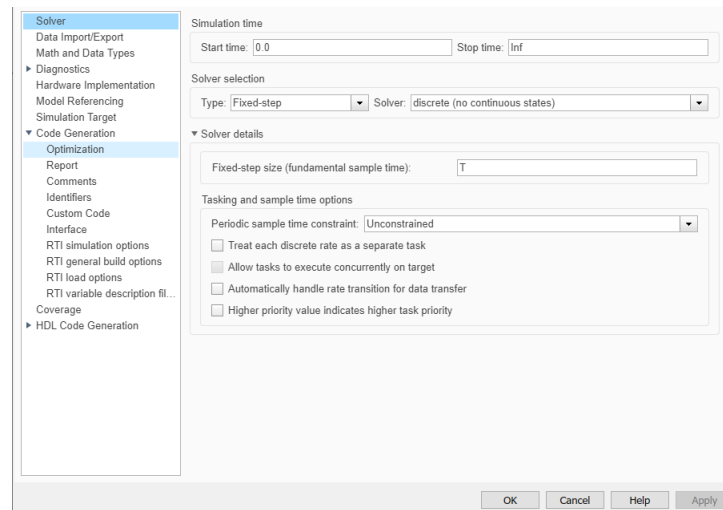


Figura 5.34: Configuración Solver de MATLAB en RTI1202

- Por otro lado, Revisar que el tiempo de simulación esté en infinito .debido a que tiempos cortos de simulación pueden tener problemas para ejecutarse en MicroLabBox.
- Al momento de crear el archivo ".sdf", es necesario tener la siguiente configuración "Load application after build" desactivada, tal como se presenta a continuación:

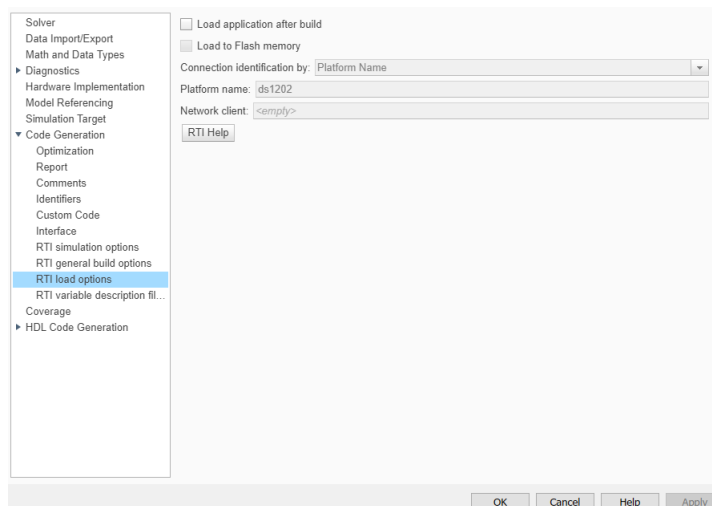


Figura 5.35: Configuración de cargado automático MicroLabBox

Esta configuración carga directamente los archivos a MicroLabBox, sin embargo, esto induce errores, por lo que es necesario cargar el archivo manualmente a través de Control Desk.

Algunas consideraciones para la implementación de proyectos en Simulink para control Desk se presentan a continuación:

- **Ganancias internas de MicroLabBox:**

Es indispensable considerar en Simulink, que para enviar señales a MicroLabBox estas deben ser previamente adaptadas. El factor de escalamiento, tanto para enviar como para recibir señales se presenta a continuación:

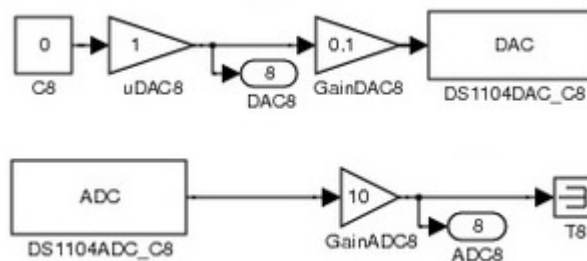


Figura 5.36: Ganancias DAC y ADC para MicroLabBox dSPACE

En la Figura 5.36, se ilustra que para enviar señales a través de un DAC en MicroLabBox, es necesario atenuar la señal aplicando una ganancia de 0.1. Por otro lado, para leer señales externas con un ADC, se requiere amplificar las señales utilizando una ganancia de 10.

Esto se debe, en principio, a que las salidas de MicroLabBox están limitadas a un rango entre 10 a -10 V. Sin embargo, es importante considerar que estas ganancias están predeterminadas por el hardware. Por lo tanto, al enviar una señal de 10 V mediante un DAC, esta se atenúa en el software por la ganancia de 0.1. No obstante, en el hardware la señal transmitida efectivamente es de 10 V. Comprender este detalle es crucial para mantener seguro el funcionamiento del equipo.

- **Compilar software nativo de MATLAB:**

Se encontraron problemas al compilar ciertas funciones nativas de MATLAB, incluido el comando "tic-toc", que se utiliza para medir el tiempo de ejecución de un programa. De la misma forma, se presentaron dificultades con algunos comandos destinados a enviar información al Workspace, como "To Workspace" y algunos archivos de Quanser relacionados con las alertas de seguridad de la planta, entre otros.

- **Cargar nuevamente los archivos desde la carpeta correcta (Reload):**

Después de crear experimentos en el software RTI, si se modifica alguna configuración en el archivo de Simulink, es imprescindible generar de nuevo el archivo ".sdf". No obstante, no es necesario crear otro experimento en RTI. Para gestionar este proceso correctamente, siempre que el nuevo archivo ".sdf" se guarde en la misma ubicación y con el mismo nombre que el archivo anterior, es posible recargar este archivo actualizado en el experimento ya existente. Sin embargo, es crucial eliminar los archivos anteriores para prevenir problemas relacionados con el "caché". Además, es importante tener en cuenta que cambiar el nombre del archivo podría resultar en la pérdida de algunas señales y configuraciones ya diseñadas en el RTI.

- **Detener la simulación en control Desk (RTI):**

Un error común en la aplicación del RTI, es no detener correctamente la aplicación del

programa. A continuación, se muestra el menú utilizado para cargar un programa en MicroLabBox:

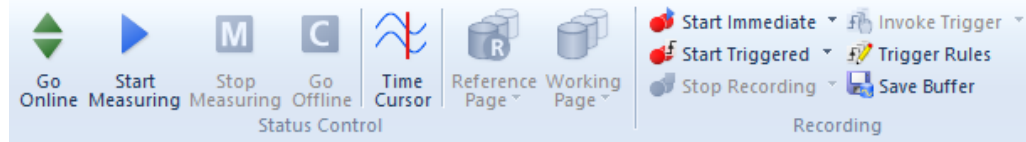


Figura 5.37: Menú superior RTI para cargar aplicaciones

La Figura 5.37 muestra las opciones disponibles para cargar la aplicación en el dispositivo MicroLabBox. Específicamente, el botón "Go Online" permite subir y ejecutar inmediatamente el programa, mientras que "Start Measuring" inicia únicamente las mediciones. Esto puede generar problemas de temporización, dado que puede no ser posible activar esta opción inmediatamente después de cargar el programa. Para evitar estos inconvenientes, se puede utilizar la opción "Start Immediate", que carga el programa, inicia las mediciones y también graba los datos según sea la configuración previa.

Por otro lado, el botón "Stop Measuring" solo detiene las mediciones y el botón "Go Offline" desactiva el programa cargado en el dispositivo.

Ahora, para detener un programa correctamente, además de seleccionar "Go Offline", es necesario detener la ejecución del dispositivo presionando sobre el nombre del programa con el click derecho la opción de "Stop" en el menú "Platforms/Devices", que usualmente se encuentra en la parte inferior de la pantalla.

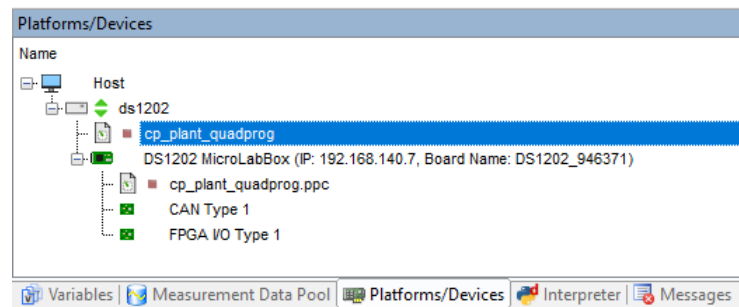


Figura 5.38: Detener aplicación RTI

No detener correctamente la aplicación puede provocar daños en la planta debido a la falta de monitoreo del sistema, además de errores de sincronización al ejecutar nuevamente un programa y la captura de datos.

- **Consideraciones de periféricos:**

A continuación se presenta la configuración de los periféricos utilizados en 5.26.

Es importante revisar la configuración correcta de los canales de entrada, los cuales están dispuestos en la parte superior de MicroLabBox, como las salidas que están en la parte inferior del dispositivo.

6. Resultados de Simulación - MATLAB

En la siguiente sección se muestran los resultados obtenidos para la implementación del control MPC en la planta de los estanques acoplados. Inicialmente, se presentan los resultados en MATLAB, para los solver de ADMM y Active-set de Quadprog.

6.1. Simulación para estado cero

Para la simulación del estado cero, se han efectuado ciertas modificaciones en el esquema MPC presentado anteriormente (ver 5.10). Se omiten los bloques de seguimiento a referencia y observadores, dejando, por lo tanto, sin efecto las entradas correspondientes a los estados observados L_{1obs} y L_{2obs} y a las variables de tendencia, x_{inf} y u_{inf} . De esta manera, el bloque MPC para el estado cero se configura como se muestra en la figura siguiente:

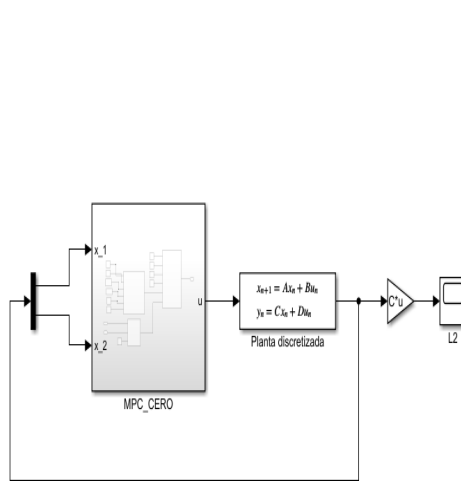


Figura 6.39: Planta para estado cero

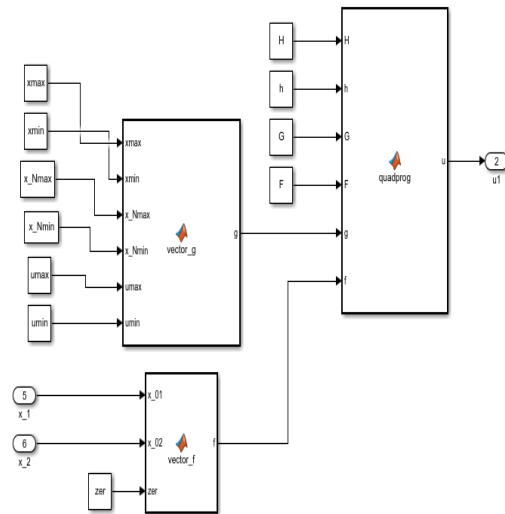


Figura 6.40: MPC para estado cero

En la figura 6.39 se observa la configuración de la planta para el estado cero, mientras que en la figura 6.46b el diagrama de bloques para el MPC estado cero.

Cabe destacar que este esquema resulta igual para el solver Quadprog, como ADMM.

Ahora, simulando los esquemas presentados anteriormente, se obtiene la simulación en estado cero como:

En la figura 6.41 se presenta el caso del estado cero utilizando el método "Active-set". Se

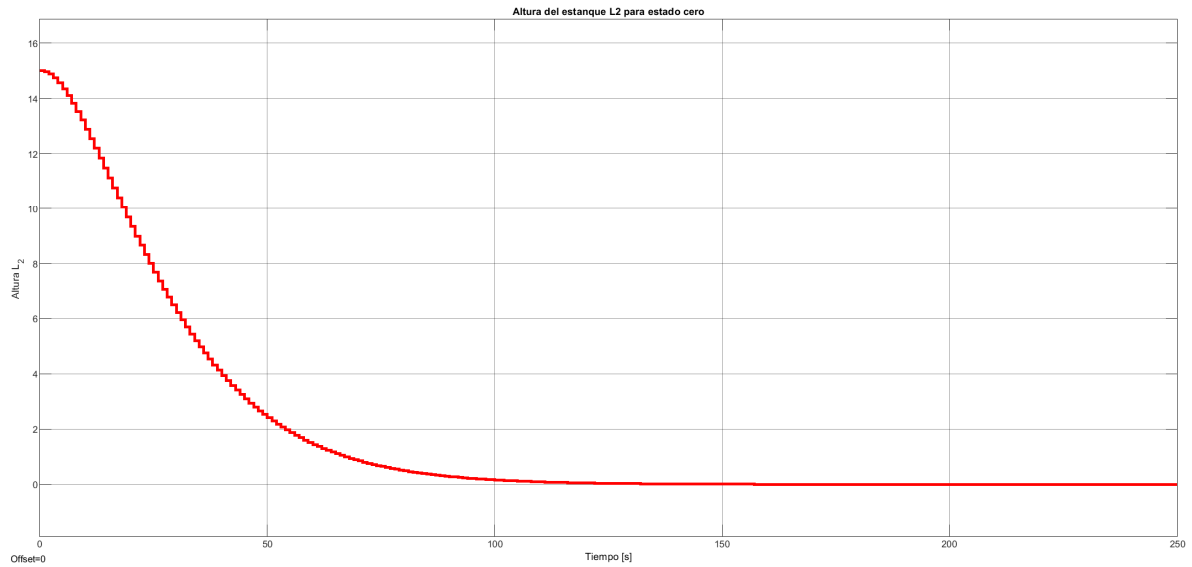


Figura 6.41: Altura del estanque $L_2(t)$ en estado cero Quadprog

observa, que teniendo una condición inicial para ambos estanques en su punto de equipo (15[cm]) estos empiezan a descargarse, finalmente llegando al estado cero esperado. Además, al no haber referencia y por ende no necesitar esfuerzo de control la actuación medida es $u = 0$. Se realiza el mismo ejercicio para el método ADMM, obteniéndose lo siguiente:

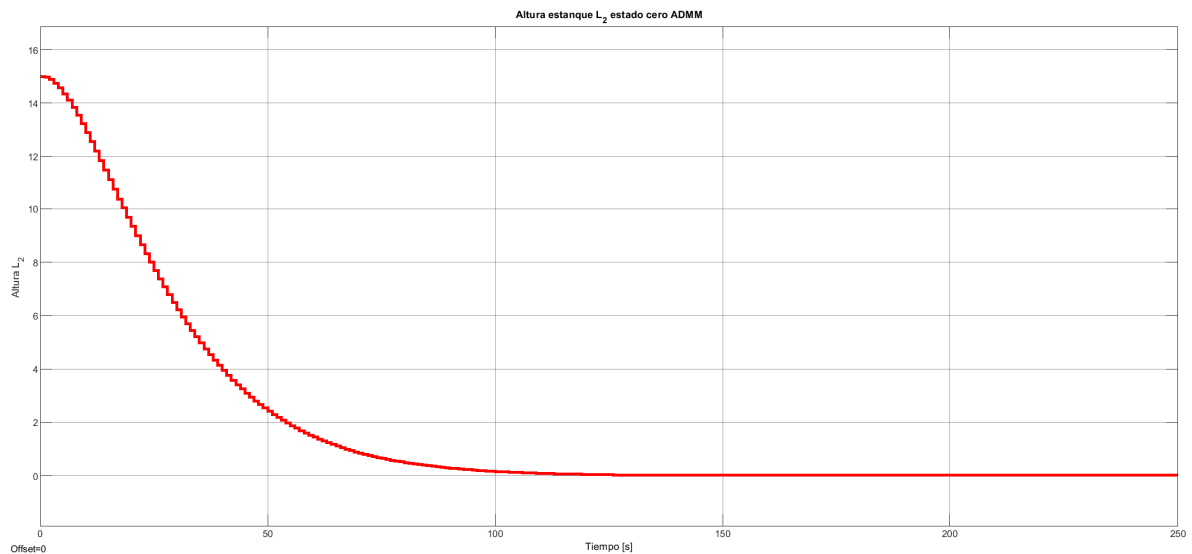


Figura 6.42: Altura del estanque $L_2(t)$ en estado cero ADMM

En general, se observa el mismo comportamiento, teniendo un tiempo de asentamiento $t \approx 150[s]$ y logrando estacionar la altura en el estado cero. Sin embargo, en base a distintas simu-

laciones es importante destacar que el método ADMM depende directamente de la cantidad de iteraciones límite que se utilicen. Por ejemplo, se utilizaron 100 iteraciones como condición final y con dicho parámetro no se alcanzó a llegar al estado cero completamente, estacionándose en un valor $L_2 = 0.016$ lo cual si bien es un valor bastante pequeño, es un orden de magnitud considerablemente mayor al observado en Quadprog de aproximadamente $L_2 = 2.833 \cdot 10^{-5}$ al mismo instante de tiempo.

6.2. Simulación para seguimiento a referencia

Para la implementación del seguimiento a referencia, no se utilizará el bloque de observadores, por lo que el diagrama de control a utilizar se presenta a continuación:

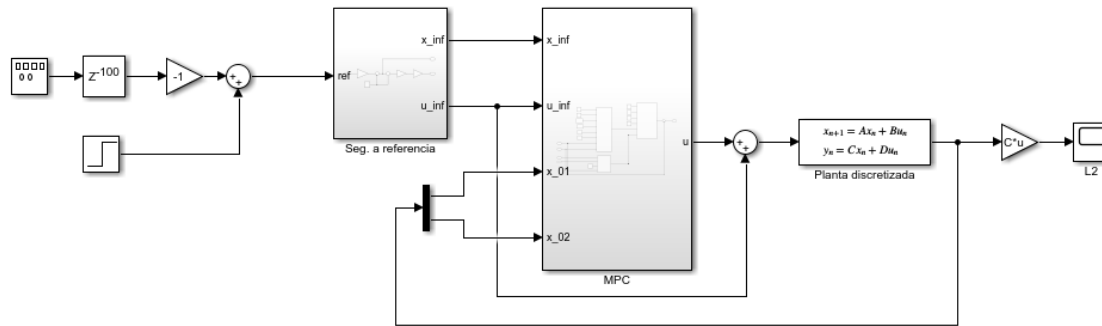


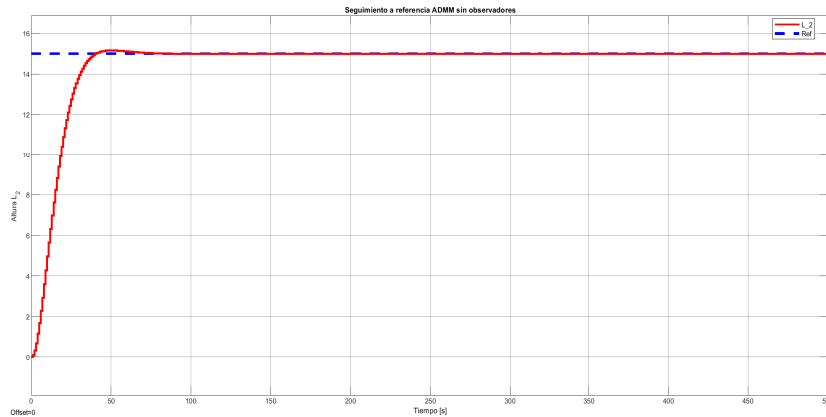
Figura 6.43: Diagrama de bloques MPC seguimiento a referencia sin observadores

De la figura 6.43 se observa que la realimentación de estados ahora viene directamente desde la planta. Además, ahora si se considerará los estados finales u_{inf} y x_{inf} .

Válido para ambos métodos de optimización, se realizan las simulaciones, primero para el solver ADMM, tal como se presenta en la figura 6.44.

De la figura 6.44, se observa el seguimiento a referencia para una única excitación de entrada. Además, se puede observar en el lado derecho el tiempo de asentamiento que toma en alcanzar el valor esperado.

Considerando lo anterior, podemos ver que se logra estacionarse sin mayores complicaciones, sin embargo, el problema del seguimiento a referencia sin observadores es que esto se complica con variaciones en el modelo para lo cual, luego será necesario añadir la sección de estimación



(a) Seguimiento a referencia sin observadores

	Time	Value
1	0.000e+00	0.000e+00
2	86.182	1.500e+01
ΔT	86.182 s	ΔY 1.500e+01
	$1 / \Delta T$	11.603 mHz
	$\Delta Y / \Delta T$	174.084 (/ks)

(b) Tiempo de asentamiento

Figura 6.44: Seguimiento referencia única sin observadores ADMM

de perturbaciones.

Adicionalmente, se presenta el actuación del sistema en la figura 6.45.

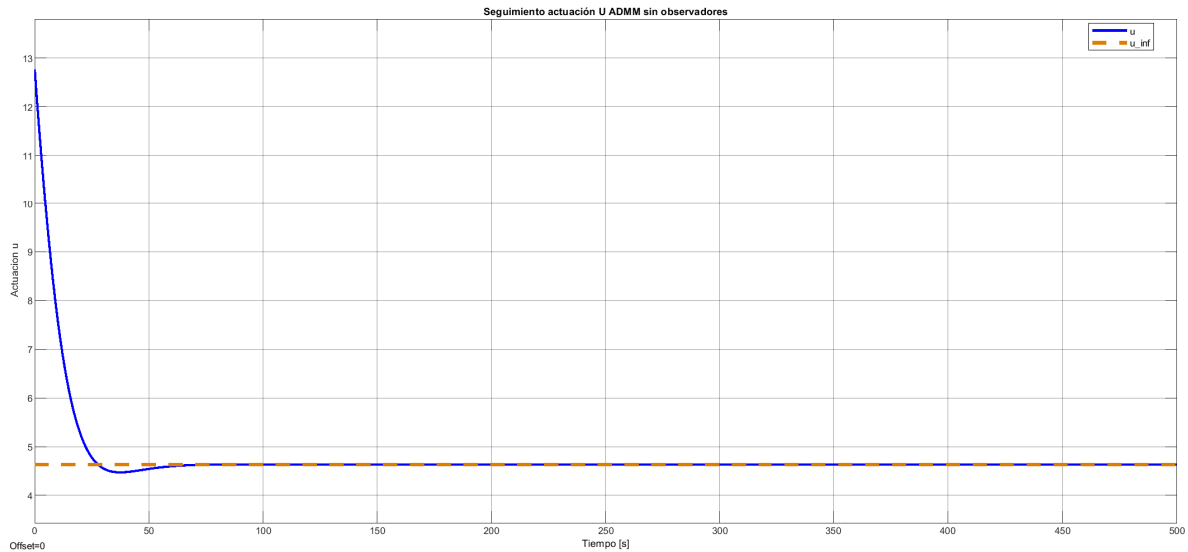


Figura 6.45: Seguimiento actuación ADMM sin observadores

Además, la actuación del sistema se ilustra en la Figura 6.45, donde se presenta tanto la señal de actuación $u(t)$ como la señal $u_{inf}(t)$, indicativa de los valores que el sistema debe mantener en estado estacionario. Estos resultados demuestran, en principio, un comportamiento coherente, con un tiempo de asentamiento comparable al ya mencionado en la Figura 6.44.

Por otro lado, se realiza la misma simulación ahora utilizando el solver QP - "Active set",

resultados los cuales se presentan en la figura 6.46.

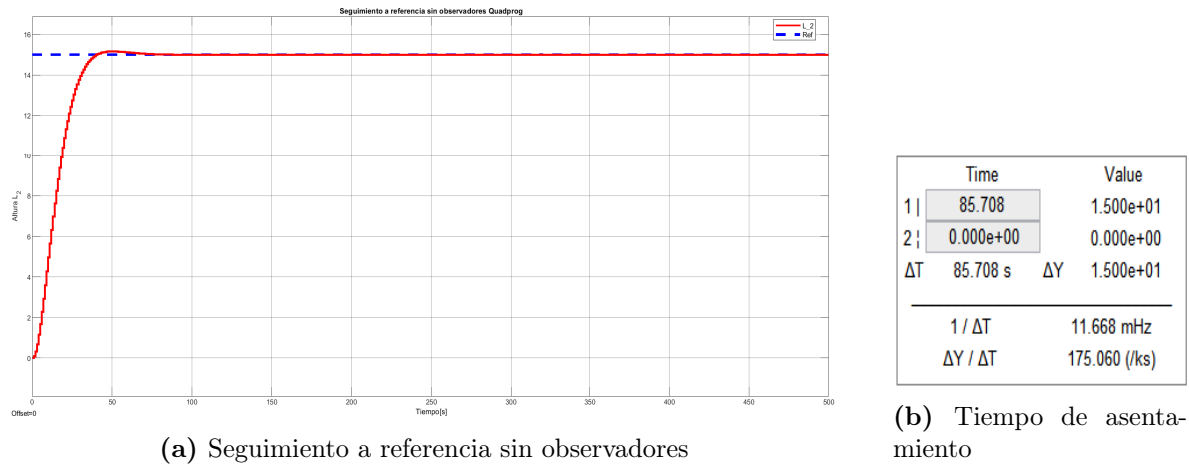


Figura 6.46: Seguimiento referencia única sin observadores QP

En la figura 6.46, se observa un comportamiento similar al mostrado anteriormente con el solver ADMM, con una pequeña variación en el tiempo de asentamiento $\delta T = 85.708$ versus los $\Delta 86.161$ del caso anterior.

En general, esta planta es bastante lenta, por lo que es normal no observar variaciones en los tiempos de asentamiento ya que se están utilizando los mismos parámetros de restricciones y horizonte de predicción. Por lo cual, en la siguiente sección se hará una comparativa del tiempo de resolución de cada uno de los Solver para poder comparar más efectivamente su funcionamiento.

Mismo caso para la actuación utilizando solver Active set que se muestra en la figura 6.47.

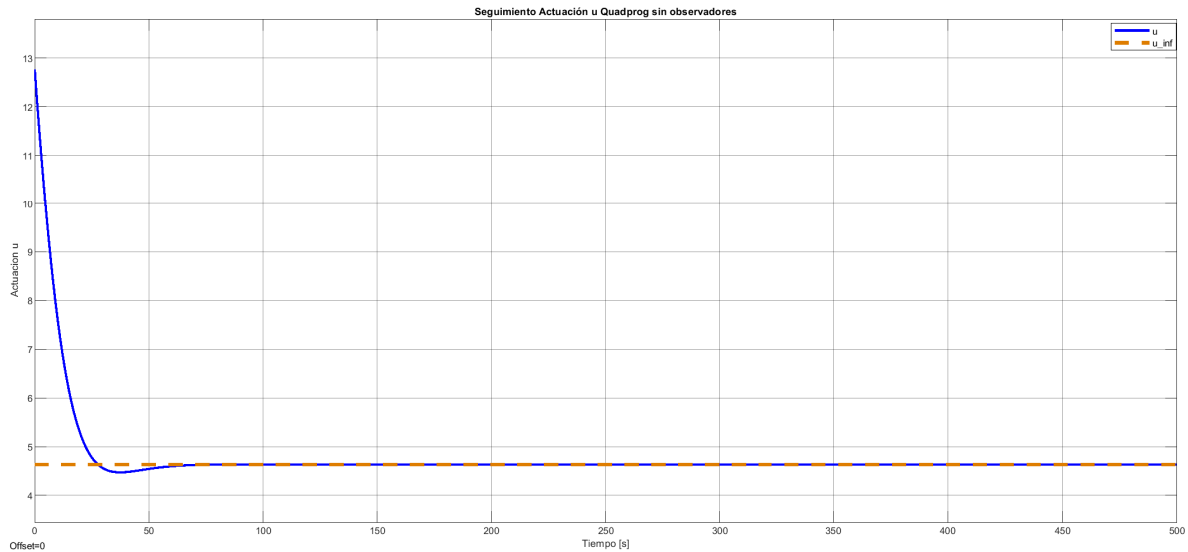


Figura 6.47: Seguimiento actuación QP sin observadores

Como se observa en la figura 6.47, se logra que la actuación $u(t)$ logre seguir a $u_{inf}(t)$ en estado estacionario, no viendo mayores diferencias con el caso anterior. Además, en ambos casos el peak inicial que presenta la actuación del sistema está dentro de los parámetros de las restricciones impuestas.

6.3. Tabla comparativa de resultados

Para determinar la eficiencia de los métodos implementados, se emplea el comando "tic-toc" de MATLAB, el cual permite la medición del tiempo de ejecución de los programas. Utilizando un parámetro de referencia única, sin observadores ni perturbaciones, se realizan las simulaciones del métodos "ADMM", para distintos criterios de parada y el método "Active-set". Los resultados obtenidos se presentan en la Tabla 6.2.

	Active-set	ADMM 1000	ADMM 500	ADMM 250	ADMM 100
Tiempo					
máximo	0.0010682	0.1095315	0.0942053	0.097176	0.0986225
Tiempo					
mínimo	0.0003101	0.0331322	0.0162829	0.0081346	0.006061
Tiempo					
promedio	0.000368939	0.047157404	0.021149503	0.0101862	0.01027834
Unidades de ingeniería	3.70E-04	4.18E-02	2.11E-02	1.11E-02	1.10E-02

Tabla 6.2: Resultados de tiempos de ejecución solvers para distintos criterios de parada

La Tabla 6.2 muestra los tiempos de ejecución comparados entre el solver ADMM y el método Active-set, destacándose este último por una notable diferencia en los tiempo de ejecución, siendo mucho más rápido en comparación con el método ADMM. Esta diferencia se debe principalmente a la optimización específica que MATLAB aplica al método Active-set, dada su integración nativa dentro del software, a diferencia del ADMM, que es implementado como un algoritmo de propósito general. Adicionalmente, el método Active-set se caracteriza por tener un código más corto y simple para entender e implementar.

Es importante mencionar que el método ADMM se evaluó hasta un criterio de parada específico de 60, punto después del cual no fue posible lograr un seguimiento adecuado de la referencia. Por otra parte, se observó que, incluso al reducir el número de iteraciones, el tiempo de resolución del ADMM tiende a estacionarse, sin presentar mejoras significativas.

6.4. Simulación para seguimiento a referencia con rechazo a perturbaciones

Como se presenta en la sección anterior, el método más óptimo para resolución del problema es utilizando el solver de Quadprog - "Active set". Ahora, se realizan las simulaciones del sistema completo con observadores y estimación de perturbaciones, tal como, se presenta en la figura 5.10. En primer lugar, se presenta el seguimiento a referencia del sistema en la figura 6.48.

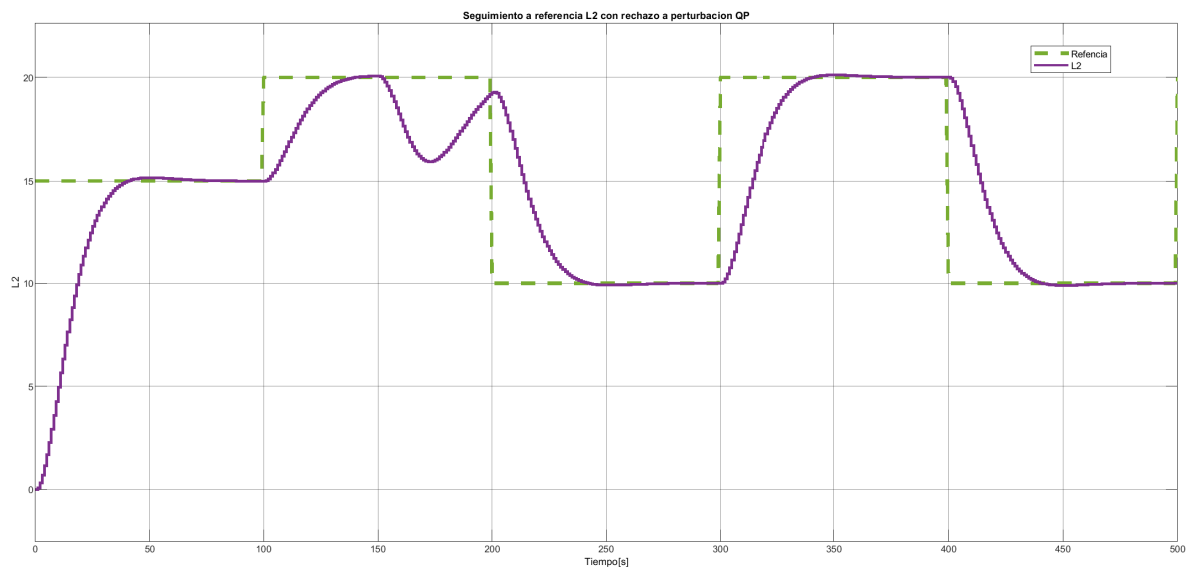


Figura 6.48: Seguimiento a referencia QP con observadores

La Figura 6.48 muestra que ahora se usa una referencia que cambia con el tiempo. Al principio, el estanque está vacío y cuando llega a un equilibrio en 15[cm], se mueve el punto de equilibrio a 20[cm]. Donde el sistema se estabiliza sin complicaciones, hasta que, cerca de los 150[s] se induce una perturbación de carácter constante, la que simula la extracción de líquido del estanque. Después de esto, el sistema comienza ajustarse rápidamente para seguir la referencia. Sin embargo, nuevamente se cambia el punto de operación, esta vez a 10[cm], donde el sistema ya tiene el tiempo necesario para estabilizarse correctamente y así sucesivamente en las siguientes variaciones.

De la misma manera, podemos ver el efecto de la perturbación del sistema en la actuación, la cual se presenta en la figura 6.49.

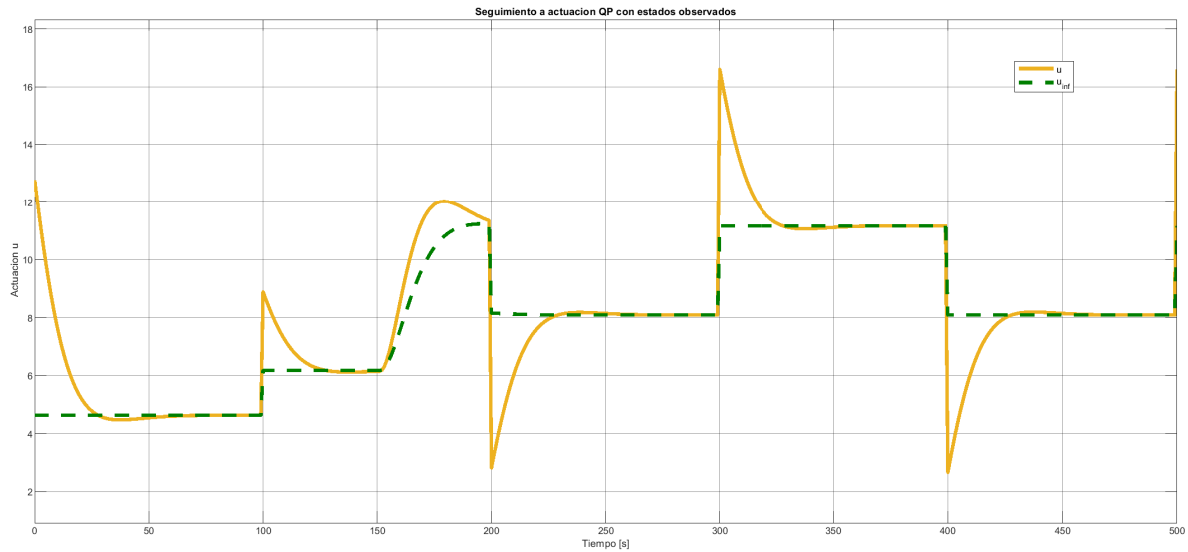


Figura 6.49: Seguimiento de actuación $u(t)$ QP con observadores

Ahora, en la Figura 6.49, observamos que el sistema también sigue la referencia, en este caso, la señal $u_{inf}(t)$. Cerca del instante de 150[s], el sistema comienza a actuar más bruscamente para compensar la perturbación. En general, también podemos ver que para el cambio en las referencias el sistema tiene un estado transiente elevado y termina estacionándose, lo cual es normal siempre que se mantenga dentro del límite de las restricciones impuestas.

Por otro lado, los estados observados y la perturbación se muestran en las figura 6.50, 6.51.

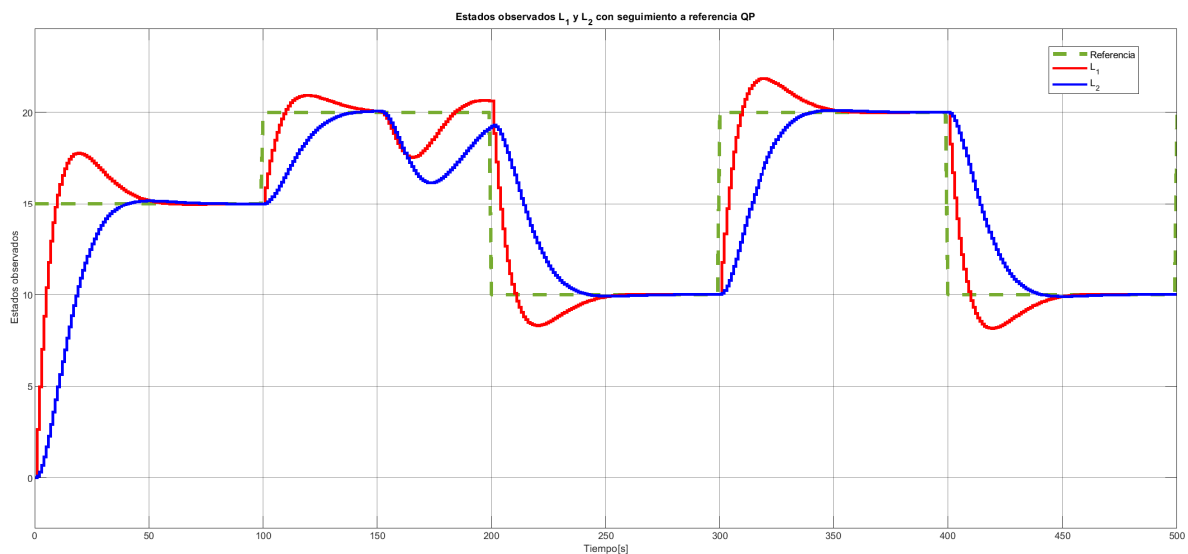


Figura 6.50: Estados observados y referencia

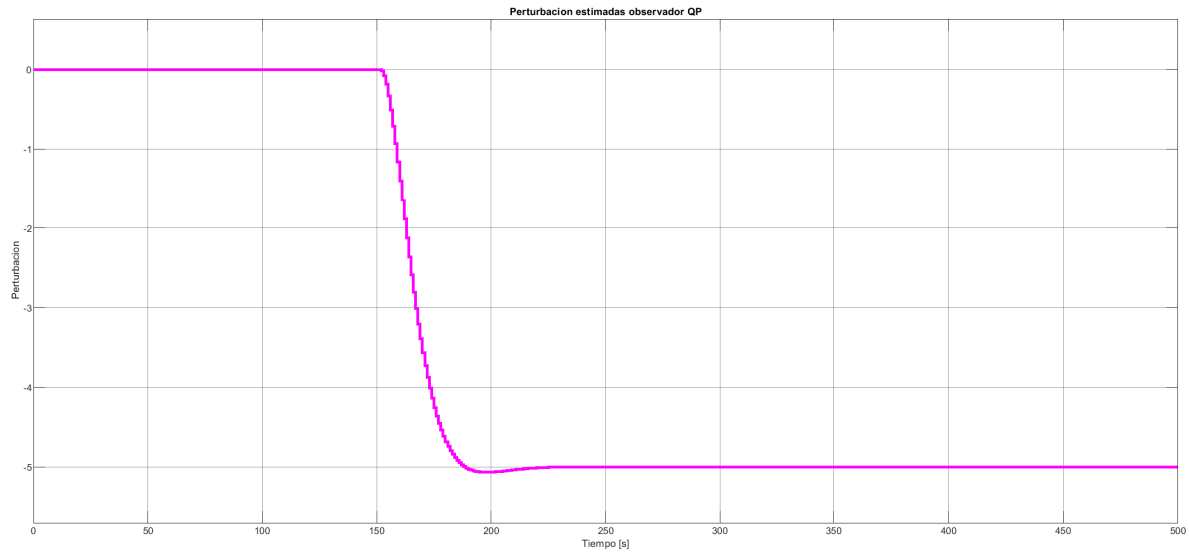


Figura 6.51: Perturbación inducida d

En la figura 6.50 se observan los estados estimados de ambos estanques, donde podemos ver que la perturbación afecta directamente a ambos sistemas al estar interconectados. Además, al ver la altura de los 2 estanques comprobamos que el estanque 1 se carga más que el estanque 2 en el transiente para poder llenar el estanque inferior, también que tienden al mismo punto de operación - referencia, esto es debido a que la configuración de la planta al tener las mismas aberturas de entrada y salida los puntos de operación son iguales, lo que no implica tener control sobre el estanque 1.

Por otro lado, la perturbación de la figura 6.51 alrededor del instante $t = 150[s]$ con un valor de $-5[cm]$ es de carácter constante, por lo que se concluye que el sistema es bastante robusto. Cabe destacar, que si bien solo se presenta el método de Active set, también se prueba con el solver ADMM, pudiendo conseguir resultados factibles para implementar con dicho método.

6.5. Resultados implementación Quanser

Considerando resultados discutidos anteriormente se tiene una preferencia por el uso del método de optimización "Active-set" de Quadprog. Sin embargo, su implementación en el software de Quanser presenta limitaciones, ya que Quanser no admite la compilación de métodos directamente desde MATLAB. Por esta razón, se opta por emplear el método ADMM como alternativa inicial para evaluar la solución de control.

En las Figuras 6.52 y 6.53 se presenta la implementación del sistema de control a través de Simulink-Quarc, para verificar la implementación física del control en la planta.

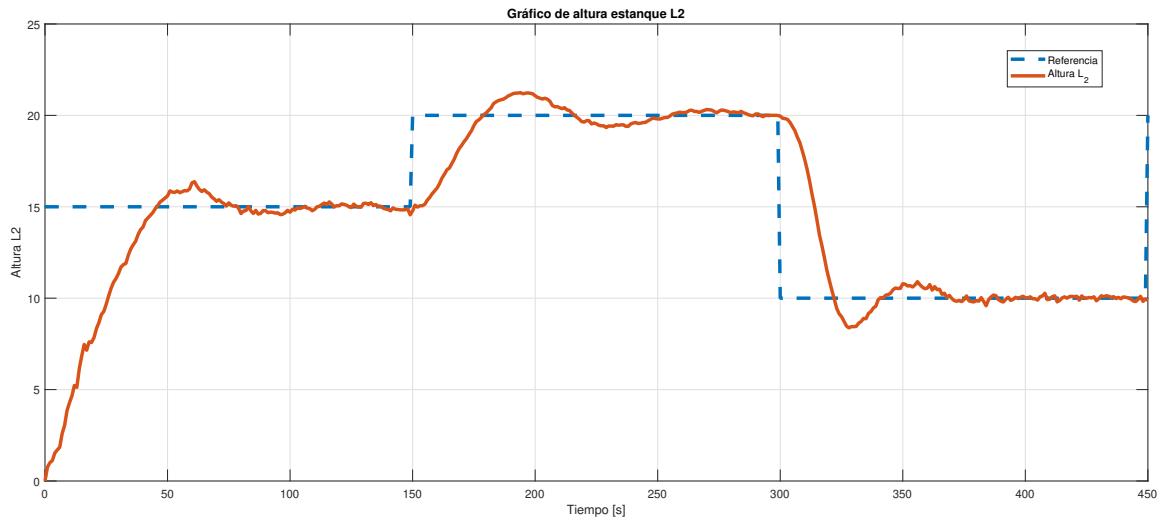


Figura 6.52: Seguimiento de la referencia de altura $L_2(t)$

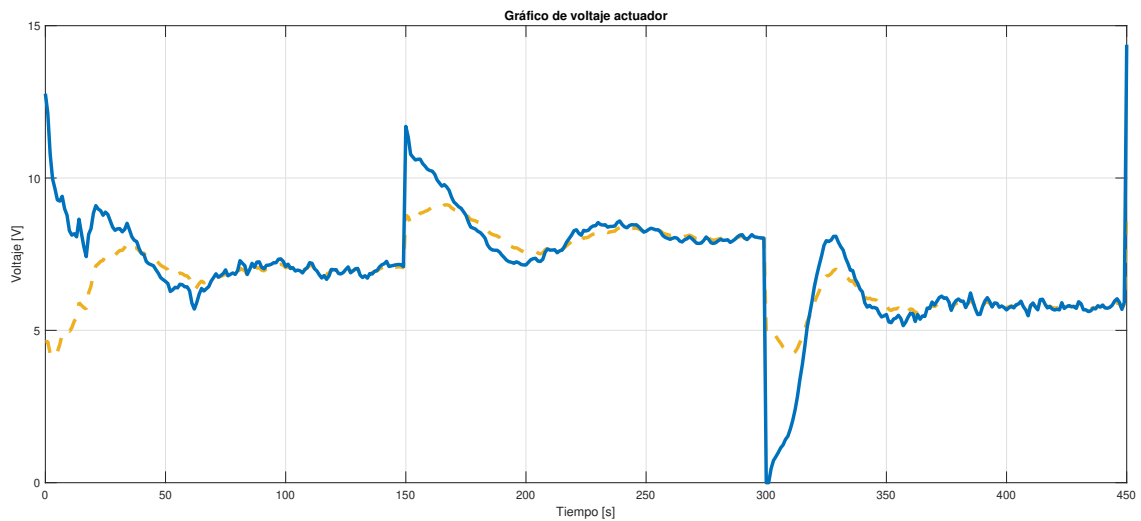


Figura 6.53: Seguimiento de la señal de actuación u_∞

La Figura 6.52 muestra el seguimiento de la referencia de altura $L_2(t)$ para diferentes valores de referencia, lo cual se logra sin mayores inconvenientes. Por otro lado, la Figura 6.53 presenta el seguimiento de la señal de actuación, donde el voltaje aplicado a la bomba logra el seguimiento

al voltaje objetivo u_{∞} . Estos resultados demuestran la viabilidad de la implementación del control en MicroLabBox de dSPACE.

Cabe destacar que la calidad del control puede variar en relación a la sección anterior de simulaciones, en específico las mediciones tienen retrasos, variaciones, dependen de una buena calibración de los sensores. Además, del error propio del modelo y algunas consideraciones como que el líquido que entra al estanque 2 salpica por lo que genera una perturbación constante que se observa en la figura 6.54 y esto también afecta el tiempo de asentamiento del control.

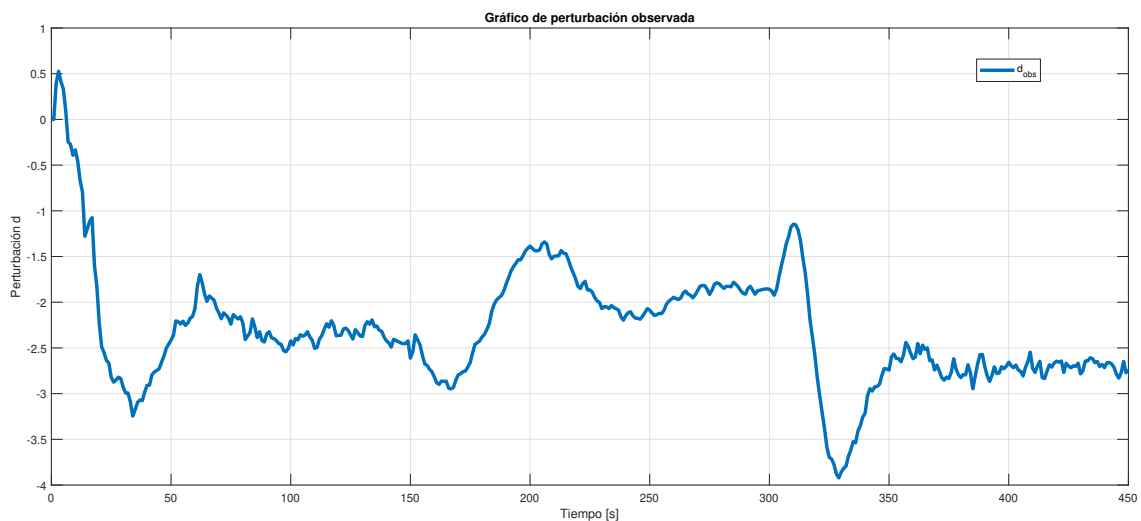


Figura 6.54: Perturbación observada por efecto de transferencia estanque 1 a estanque 2

7. Resultados implementación MPC - MicroLabBox:

Se presentan los resultados obtenidos de las simulaciones con MicroLabBox, así como el funcionamiento de la planta física.

En un principio se presenta el caso para el solver Quadprog - "Active-set", solo simulado en las figuras 7.55, 7.56.

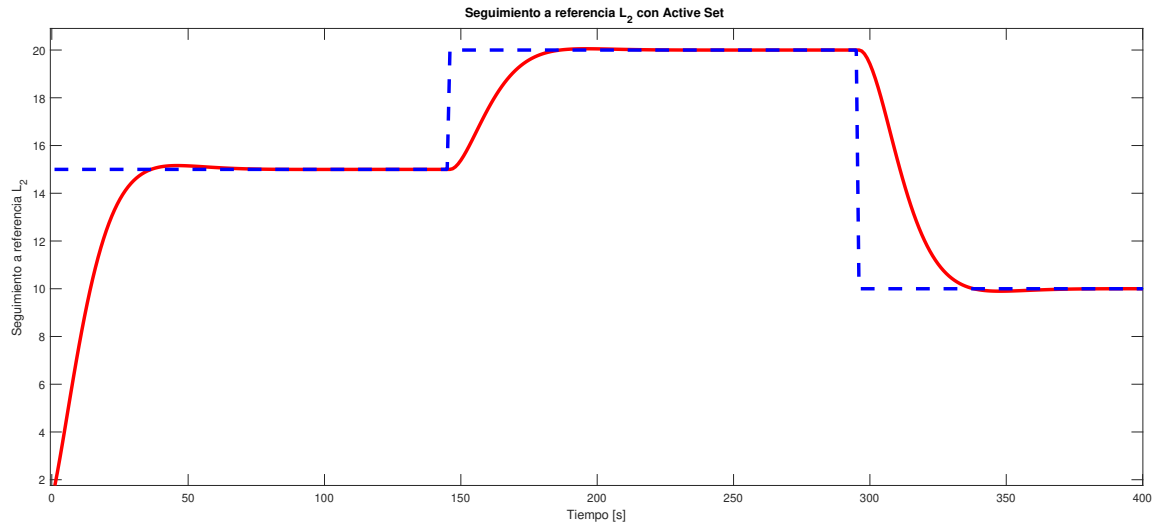


Figura 7.55: Seguimiento a referencia $L_2(t)$ QP simulado MicroLabBox

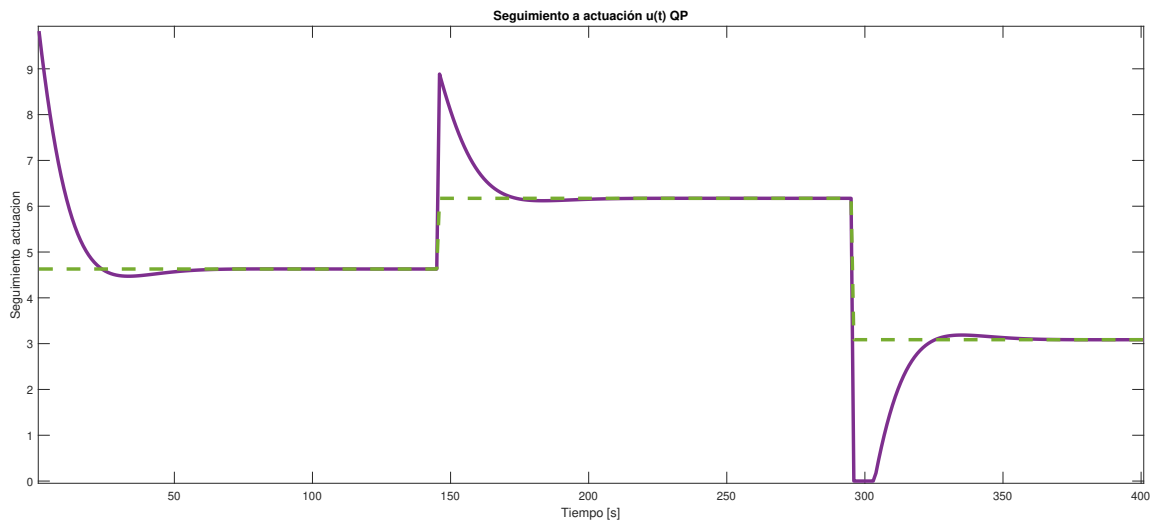


Figura 7.56: Seguimiento a actuación $u(t)$ QP simulado MicroLabBox

En la figura 7.55, se observa el seguimiento a referencia para la altura $L_2(t)$ simulada en MicroLabBox, donde podemos apreciar que logran seguir las referencias especificadas, con un tiempo de asentamiento cercano a los 80[s] y con un "overshoot" bastante pequeño de líquido, lo que permite concluir a priori que el control cumple con su objetivo, aunque cabe destacar que al estar pensado inicialmente para la planta física, no se incluye una perturbación en el

sistema, debido a que las perturbaciones de la planta real se hacen de manera manual, tal como se explica en la sección de *Modelado del sistema*.

Por otra parte en la figura 7.56, observamos el seguimiento a de la actuación $u(t)$, en este caso a la señal $u_{\infty}(t)$, con un comportamiento bastante similar al observado anteriormente en MATLAB, con un overshoot al momento del cambio de referencia que eventualmente se estacionan en el valor esperado. Sin embargo, una leve diferencia, es que aquí la última referencia de altura es de $10[cm]$, lo cual se da en un instante cercano a los $300[s]$ y donde podemos observa que la planta detiene su acción un momento limitándose a $0[V]$, lo que concuerda con las restricciones impuestas al controlador MPC.

Ahora, llevando el mismo esquema al sistema real se presentan los resultados obtenidos en la figuras 7.57, 7.58, 7.59.

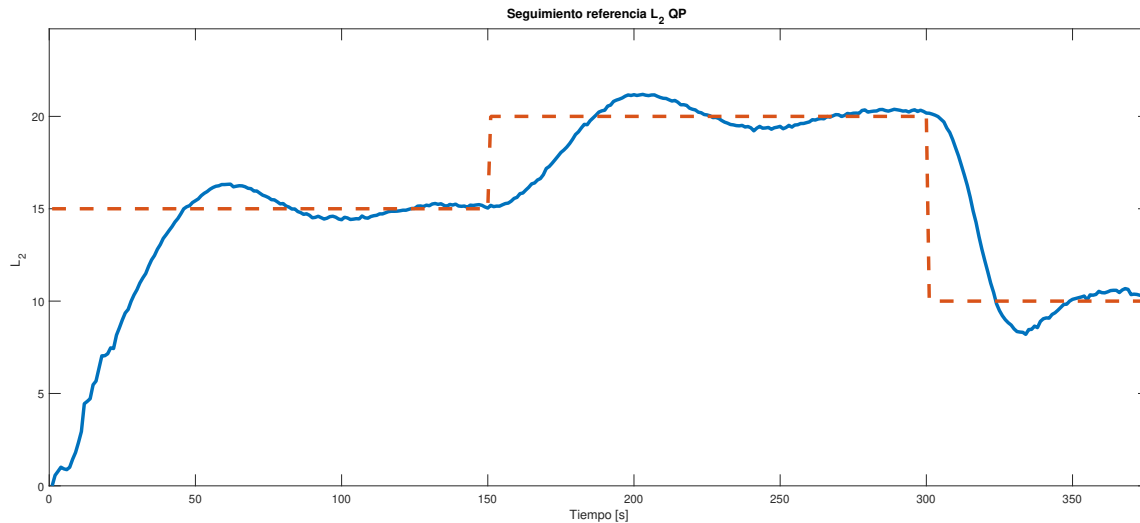


Figura 7.57: Seguimiento a referencia $L_2(t)$ QP planta física MicroLabBox

Para implementar la planta física en MicroLabBox, fue necesario cambiar algunas restricciones del código 9.1, específicamente $u_{max} = 10[V]$, esto es debido a como se explica con anterioridad, MicroLabBox no admite voltajes mayores que $10[V]$, lo cual si bien es suficiente para resolver el problema de control expuesto limita en parte las capacidades del seguimiento a referencia. En la figura 7.57, se presentan las señales de medidas en la altura del estanque $L_2(t)$ real y la señal de referencia digital. Donde podemos observar que si bien ahora existen más variaciones en la altura esta logra eventualmente estacionarse y seguir la referencia, tal como se espera

comprobando la simulación, incluso con las restricciones anteriormente mencionadas.

Ahora, la actuación se presenta como:

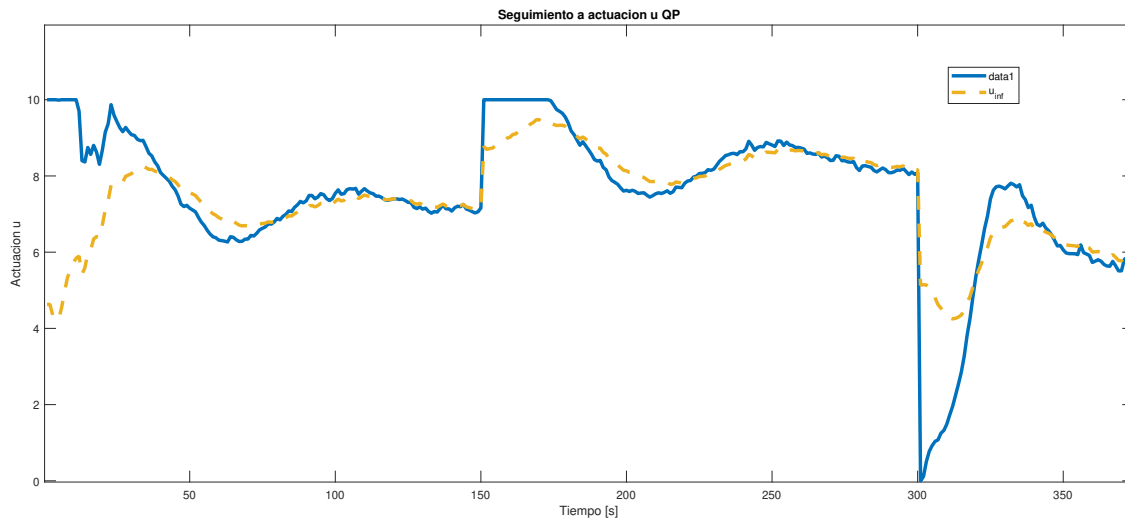


Figura 7.58: Seguimiento a actuación $u(t)$ QP planta física MicroLabBox

En la figura 7.58, se presenta el esfuerzo de control $u(t)$ de la bomba para seguir la referencia, donde también, se incluye la señal u_{∞} corresponde a la tendencia del sistema. Podemos ver que en ciertos instantes la actuación se ve saturada en $10[V]$ como al momento de correr el sistema y también en el cambio de referencia en $t = 150[s]$, sin embargo, el controlador rápidamente comienza a adaptarse para poder seguir la señal u_{∞} y por consecuencia alcanza la altura requerida. De la misma manera vemos que al cambiar la referencia a un valor menos en el instancia $t = 300[s]$ el sistema se satura inferiormente en $0[v]$ demostrando así que cumple con ambas restricciones impuestas.

Ahora, cabe mencionar tal como se muestra en la figura 7.59, se induce una perturbación en el sistema, por la acción de la caída del líquido entre el estanque 1 y el estanque 2 que no es regular.

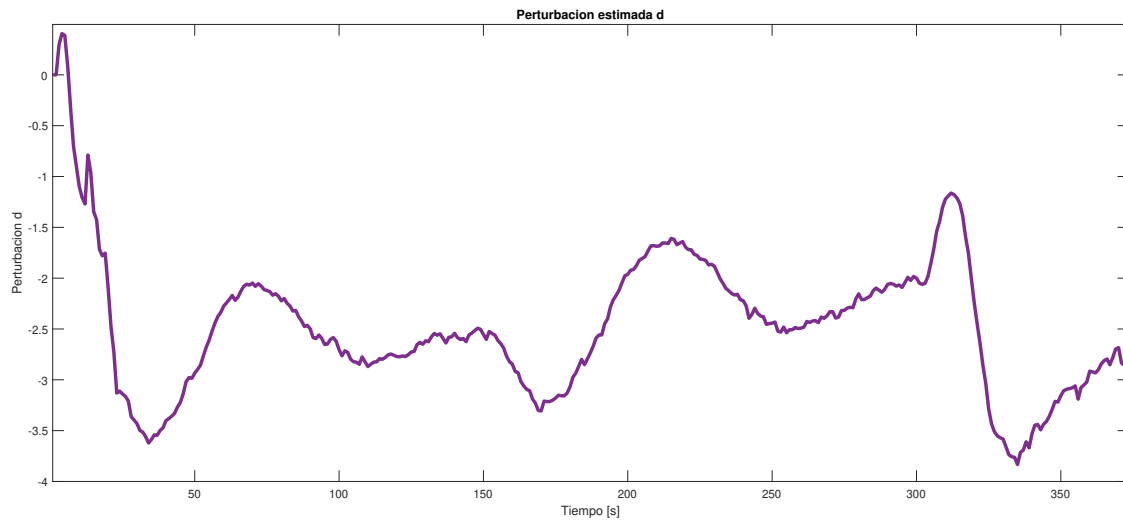


Figura 7.59: Observación de la perturbación estimada planta real MicroLabBox

Esto genera una leve diferencia, en relación al caso simulado. Sin embargo, se logra con el objetivo de control de igual manera.

En general, se comprueba que la nueva restricción impuesta en la actuación $u_{max} = 10[V]$, es suficiente para el sistema de control presentado, sin embargo, para algunas perturbaciones constantes de mayor magnitud este valor no es suficiente, por ejemplo, se simula el caso de falla en la figura 7.60.

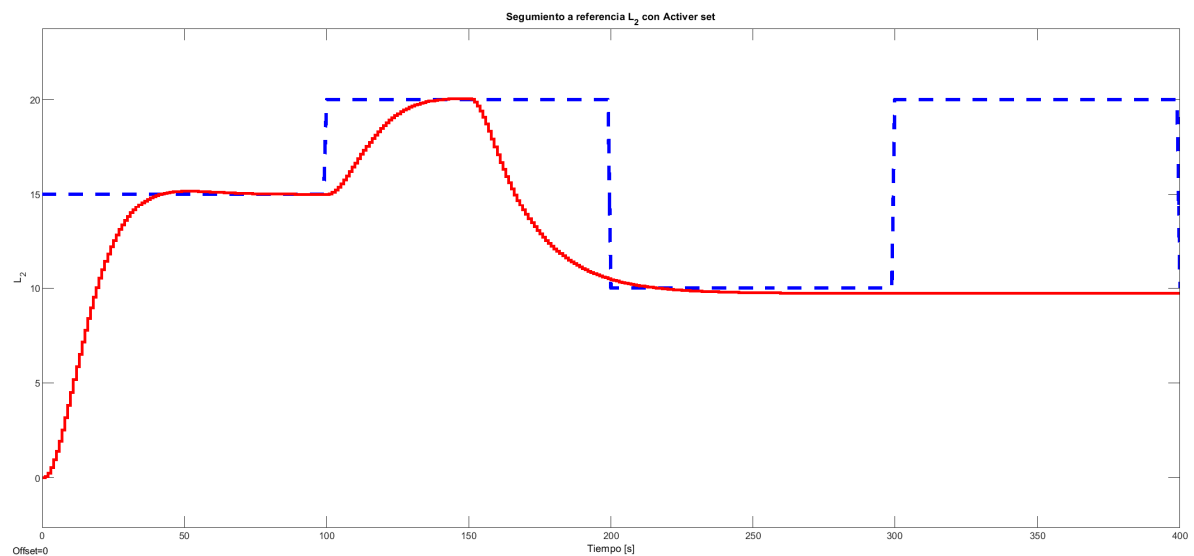


Figura 7.60: Caso de falla seguimiento a referencia L_2

En la figura 7.60, se presenta el caso para una perturbación $d = -7[cm]$, con un voltaje $u_{max} = 10[V]$, con el cual el sistema ya no es capaz de seguir la referencia al momento de inducir la perturbación en $t = 150[s]$. Del mismo modo, en la figura 7.61, se presenta el comportamiento de la actuación del sistema.

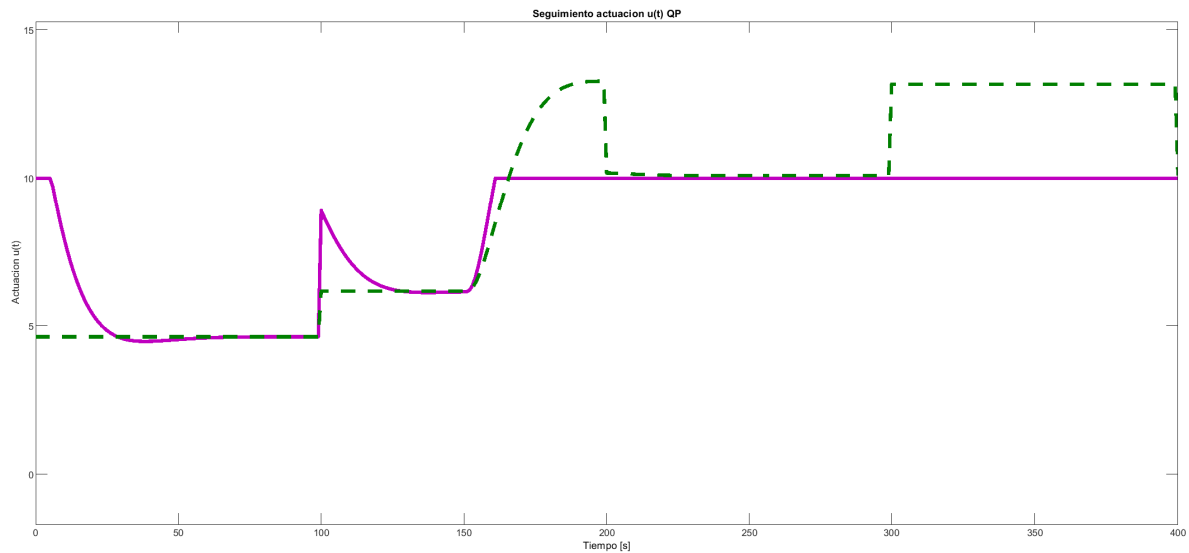


Figura 7.61: Caso de falla actuación $u(t)$

En la figura 7.61, se observa que al momento de inducir la perturbación la señal u_{∞} incrementa el valor de la actuación para poder seguir la referencia, sin embargo, al estar limitado el controlador la señal $u(t)$ solo se satura en $10[V]$ cumpliendo así con las restricciones estipuladas. Lo anterior, se debe tener en consideración al momento de testear la planta física, ya que mantener prolongadamente la perturbación va a saturar el controlador y no será capaz de cumplir con el objetivo de control.

De la misma manera, se realizan las simulaciones para el caso del método con ADMM, tal como se presenta en la figura 7.62.

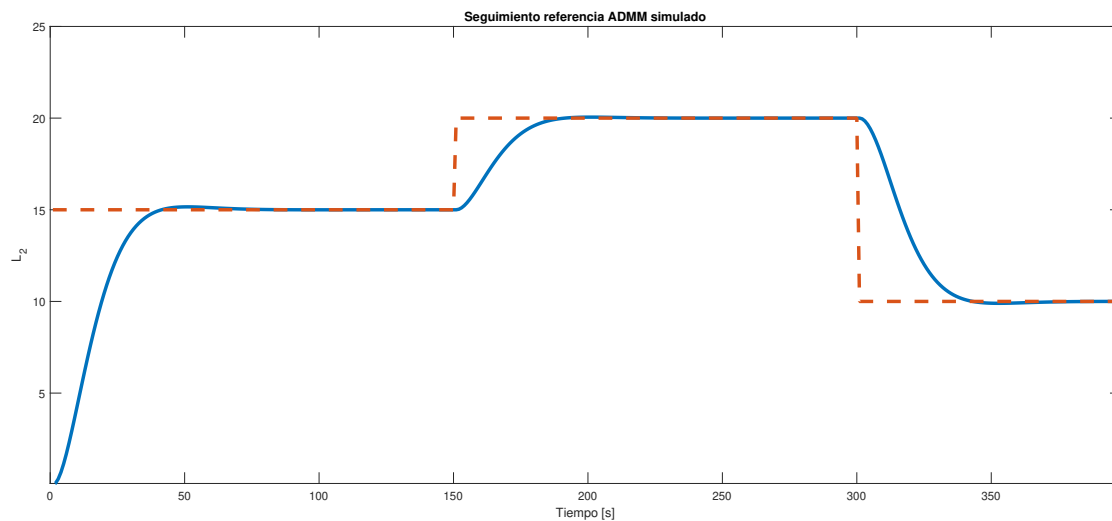


Figura 7.62: Seguimiento a referencia L_2 ADMM simulado MicroLabBox

En la figura 7.62, se observa el seguimiento a referencia con el solver ADMM simulado, donde se prueba que de igual manera que el caso con Active-set, este es capaz de seguir la referencia y estacionarse con un tiempo cercano a los 80[s].

Equivalentemente para el caso de la planta real, se presentan los resultados en las figuras 7.63, 7.64.

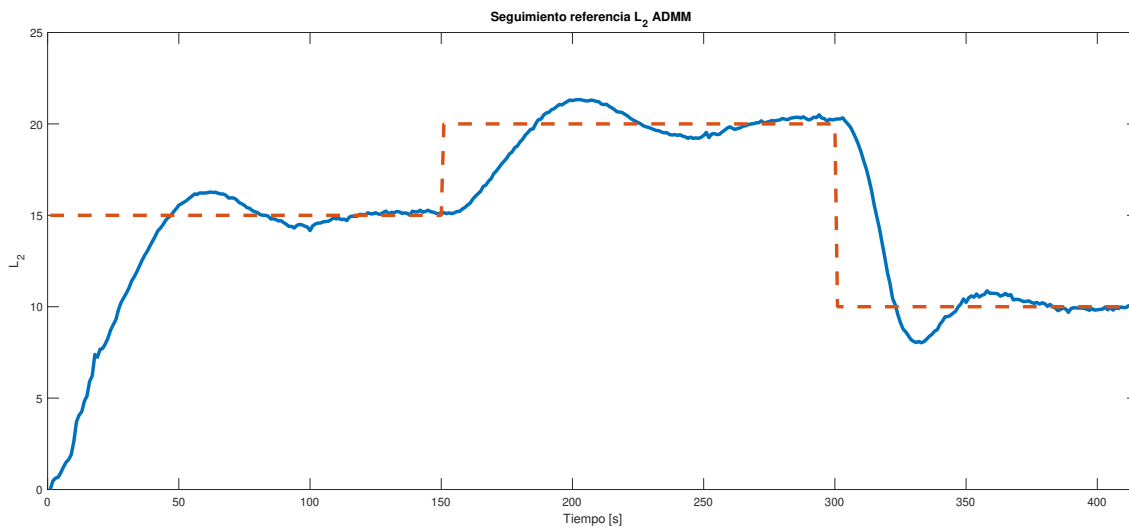


Figura 7.63: Seguimiento a referencia $L_2(t)$ ADMM MicroLabBox

En la figura 7.63, al igual que lo observado en el caso anterior con el solver Active set, el tiempo de estacionamiento de la planta bastante grande debido a las características de esta misma, por lo que se comprueba que sigue bien la referencia en los tiempos esperados. A su vez, si observamos la actuación del sistema en la siguiente figura:

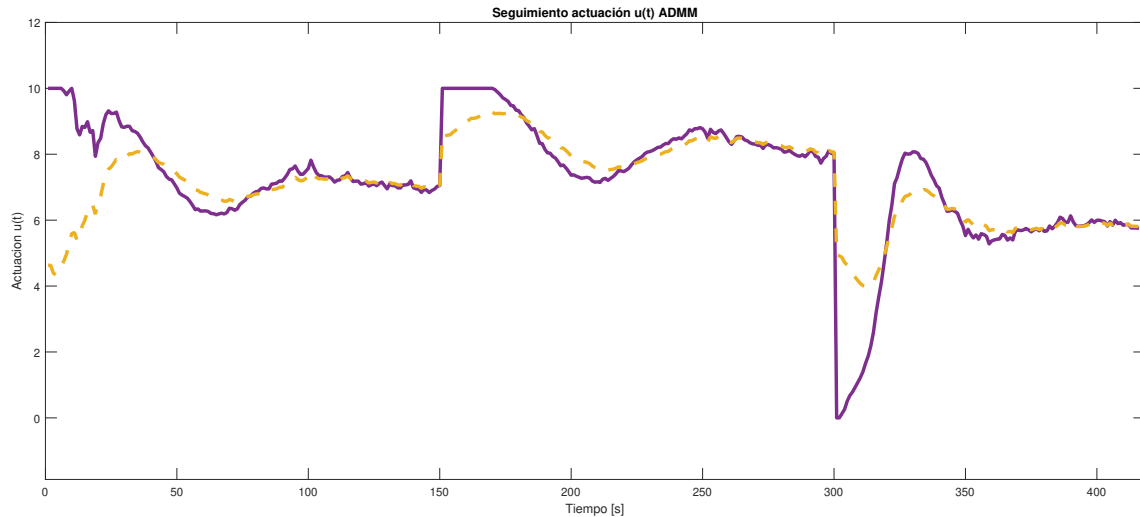


Figura 7.64: Seguimiento a actuación $u(t)$ ADMM MicroLabBox

En la figura 7.64 se notan unas sutiles diferencias en relación al caso presentado para el solver Active set en 7.58. Donde en un principio el controlador pareciera estar menos tiempo saturado y si bien tiende a ser un poco más errático el comportamiento, se estaciona de igual manera siendo bastante certero y pudiendo seguir la actuación u_{∞} esperada.

Por otro lado, la perturbación observada para el caso del ADMM, se presenta en la siguiente figura:

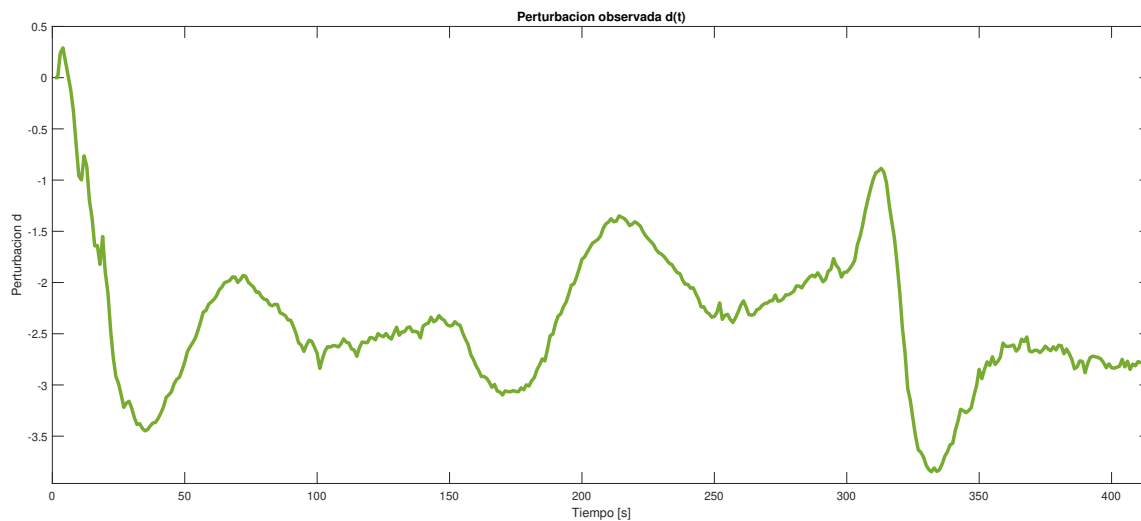


Figura 7.65: Perturbación observada $d(t)$ MicroLabBox

De la figura 7.65, es bastante similar a la observada anteriormente en 7.59, sin embargo, existen pequeñas variaciones las cuales dependen netamente de como el sensor de altura y el modelo del observador sean capaces de reconstruir esta señal.

Cabe destacar que es importante tener en cuenta las variaciones del control considerando los estados previos de la planta, es decir, si es que el sensor tenía agua previamente o si estaba en seco y la calibración entre cada prueba irán cambiando levemente los resultados de control.

8. Conclusiones

En esta memoria, se ha desarrollado tanto un marco teórico como práctico para el diseño e implementación de un sistema de control predictivo aplicado a una planta hidráulica respectiva a los estanques acoplados, a través de dos enfoques de optimización distintos: el método ADMM y el método Active-set de Quadprog. Se ha buscado demostrar la factibilidad de su implementación en el dispositivo MicroLabBox de dSPACE y comprobar con esto su funcionamiento. Inicialmente, dentro del marco teórico del diseño de aplicaciones MPC, se comprueba la versatilidad de su implementación. Debido a que los sistemas dinámicos (lineales y/o linealizados), expresados de forma matricial, hacen bastante sencillo cambiar el modelo y los parámetros de estos mismos para distintas aplicaciones industriales. En general, esto se puede apreciar más directamente en el entendimiento del código 9.1. En una segunda instancia, pasando directamente a la implementación, MATLAB - Simulink son herramientas realmente poderosas para las aplicaciones de sistemas dinámicos, donde su entorno de programación y gráfico permite una experiencia de usuario bastante amigable para una gran variedad de aplicaciones en los ámbitos de la ingeniería. En específico, para el MPC los bloques de sistemas de estado discretizado, funciones de MATLAB, etc.

Continuando con el tema, entrando al detallar la implementación del algoritmo MPC, el que para ambas metodologías de optimización ADMM y active Set, se utilizan los parámetros de horizonte de tiempo $N = 10$ y paso de discretización $T = 1[s]$, lo cual resulta suficiente considerando que las características físicas de la planta hacen que el tiempo de estabilización, independientemente del Solver utilizado sea alrededor de los 80[s].

Ahora, entrando a la comparación de los Solver, se realizan las simulaciones de estado cero, seguimiento a referencia sin observadores y finalmente el sistema completo considerando observadores y rechazo a perturbaciones. Principalmente enfocados en la simulación, se realiza una comparativa de los solver ADMM y Active Set, en donde se pudo concluir que el método Active Set es altamente más efectivo por velocidad de resolución, compatibilidad con el software y simplicidad en su código. Sin embargo, tanto el horizonte de tiempo N , como el paso de discretización T se mantienen constantes en ambos casos probados, por lo que, en general no se pueden apreciar mejoras significativas en el control comparando los solvers. Esto se debe a que, con el método Active set, hay una holgura mayor para tomar decisiones de control, pero

al ser una planta tan lenta, la carga computacional y la cantidad de datos generados, no se contraponen con alguna mejora significativa en la rapidez del controlador.

Desarrollando aún más la idea anterior, dentro de las pruebas físicas del sistema, inicialmente se utiliza en Solver ADMM con la instrumentación de Quanser y posteriormente con el Software y Hardware MicroLabBox de dSPACE. La principal diferencia de las pruebas en los equipos, es que la central de procesamiento del control con Quanser es el computador personal, mientras que MicroLabBox tiene Hardware dedicado que permite explotar mucho más la capacidad de procesamiento. Por esto, MicroLabBox resulta un sistema atractivo para la implementación y prueba de sistemas de control. En específico, para esta aplicación al ser una planta en particular lenta, no se utilizan todas las capacidades de procesamiento del dispositivo. Además, se ve limitado por la alimentación físicamente del sistema, debido a que en Quanser se utiliza un voltaje máximo de actuación de $u_{max} = 22[V]$, mientras que MicroLabBox está limitado a $u_{max} = 10[V]$, esto es debido a que el dispositivo no puede entregar más voltaje y si bien la aplicación no quiere de más potencia para funcionar correctamente, tal como se presentó anteriormente, existen casos donde esto podría no ser suficiente para cubrir todos los requisitos de control.

Finalmente, se comprueba el funcionamiento del controlador predictivo, para distintas condiciones de operación y su factibilidad de aplicación en el dispositivo MicroLabBox de dSPACE, tanto en el entorno de simulación, comprobando que para ambos solvers se tienen resultados satisfactorios de control, tiempo de asentamiento, overshoot y cumplimiento de las restricciones impuestas. Se destaca especialmente, el poder de la aplicación de MicroLabBox, como un dispositivo altamente versátil, en los ámbitos de la simulación física, la interfaz HMI, obtener datos casi en tiempo real, en conjunto con una conexión sencilla se pueden estudiar conceptos que son escalables al ámbito industrial actual como el control avanzado de procesos MPC, HIL (Hardware in the loop), gemelos digitales, etc. Por otro lado, mencionar que una de las pocas desventajas del sistema es que no se logró medir el tiempo de ejecución de los solver nativamente en MicroLabBox y que se encontraron pequeñas variaciones de las mediciones reales, versus lo que presentaba el HMI.

8.1. Trabajos futuros

Se plantean algunos desafíos futuros como:

- Cambiar los sensores de presión por sensores de cuantizados de bajo coste.
- Mejorar el diseño e interacción con la interfaz HMI de MicroLabBox, por ejemplo, poder dar una referencia directamente desde el programa, efectos de visualización en gráficos de barras, animaciones, etc.
- Utilizando el Solver Quadprog, poder cambiar la configuración de la planta por ejemplo a un sistema MIMO, con un mayor horizonte de predicción y un menor tiempo de paso. Principalmente para mejorar la velocidad de asentamiento de la planta y testear las capacidades de MicroLabBox.
- Testear las capacidades máximas del dispositivo MicroLabBox, disminuyendo el paso del muestro al menor tiempo posible.
- Cambiar la implementación del MPC por aplicaciones más avanzadas, en específico redes neuronales y estrategias de Machine learning. Esto podría unificar las formas de solucionar problemas de control.
- Cambiar la implementación en código del MPC por descripción de Hardware, es decir, utilizar la FPGA para implementar MPC y comparar eficiencia.

9. Anexos

9.1. Código constantes MPC

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MPC con formulacion Sparse %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PARAMETROS PLANTA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4  %-----%

%Parametros generales

g1 = 981;                                %Gravedad [cm^2/s]
Kp = 3.3;                                %Constante de bomba [cm^3/s/V]
9

% Areas de los estanques

At1 = 15.5179; %[cm^2]
Ao1 = 0.17810; %[cm^2]                  %Ao - variable - 0.0792/ 0.1785 /0.2425
14 Ao2= 0.17810;  %[cm^2]

%Puntos de operacion

L2q = 15;                                %Punto de operacion estanque 2
L1q = ((Ao2/Ao1)^2)*L2q;                  %Punto de operacion estanque 1
19

%Variables de Estado en Tiempo Continuo

Ac = [(-Ao1*g1)/(At1*sqrt(2*g1*L1q))    0; (Ao1*g1)/(At1*sqrt(2*g1*L1q))  (-Ao2*g1)/(At1*
      sqrt(2*g1*L2q))];
Bc = [Kp/At1; 0];
C = [0 1];
24 D = 0;

sys_continuo = ss(Ac,Bc,C,D);              % Planta en variables de estados

%DEFINICION SISTEMA MUESTREADO
29

T = 1;                                    % Tiempo de muestreo
sys_discreto = c2d(sys_continuo,T);        %Def. sist discreto
[Ad,Bd,Cd,Dd] = ssdata(sys_discreto);     %Matrices discretas
A = Ad;
34 B = Bd;
C = Cd;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SEG. REFERENCIA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

39 inv1 = inv(eye(size(Ad,1))-Ad);

```

```

inv2 = inv(Cd*(inv1)*Bd);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MPC %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

44  N = 10;                                %Horizonte de prediccion
    Omega = 1*(C')*C;                      %Matriz de peso estado
    Gamma = 1.5;                          %Matriz de peso actuacion

    [Lopt,OmegaN,~] = dlqr(Ad,Bd,Omega,Gamma); %Regulador cuadratico
49
    % Observadores de estado
    %Matrices extendidas observador
    Az = [Ad Bd; 0 0 1];
    Bz = [Bd; 0];
54  Cz = [Cd 0];

    [L,~,~] = dlqr(Az',Cz',eye(3),eye(1));
    Lz = L';

59  %Matrices observadas (Sistema)

    A_obs = Az - Lz*Cz;
    B_obs = [Bz Lz];
    C_obs = eye(3);                      %Salidas x1(k),x2(k),d(k)
64  D_obs = zeros(3,2);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MATRICES H %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

69  H = blkdiag(kron(eye(N),blkdiag(Omega,Gamma)),OmegaN); %Hessiana
    H = (H+H')/2;
    h = zeros(size(H,1),1);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RESTRICCIONES MPC %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

74
    x0 = [0;0];                          %Punto inicial
    d0 = 0;                              %Perturbacion inicial
    xmax = [30;30];                      %Restriccion estado maximo
    xmin = [0;0];                        %Restriccion estado minimo
79  umax = 12;                          %Restriccion actuacion maxima
    umin = 0;                          %Restriccion actuacion minima
    x_Nmin = xmin;                      %Restriccion final minima
    x_Nmax = xmax;                      %Restriccion final maxima

```

```

84  %Dimensiones del sistema

n = size(Ad,1);          % Tama o del vector de estados (n=2)
m = size(Bd,2);          %Tama o del vector Bd entrada  (m=1)

%Matrices de restricciones
89  J=[eye(n);-eye(n);zeros(m,n);zeros(m,n)];          % I [2x2] // Cmn [1x2]
    JN=[eye(n);-eye(n)];          %I[2x2]
    E=[zeros(n,m);zeros(n,m);eye(m);-eye(m)];          %Cnm [2x1] // I [1x1]

%-----RESTRICCIONES DE IGUALDAD -----%
94  In = eye(n);
    f = [Ad,Bd,-In];
    F = [];
    col_F = (N+1)*n+N*m;
        for i=1:1:N
99            if i == 1
                F = [-In,zeros(n,col_F-n)];
            end
            M = [zeros(n,(i-1)*(n+m)),f,zeros(n,col_F-((i-1)*(n+m)+(2*n+m)))];
            F = [F;M];
104        end
        ncero=size(F,1)-n;          %Cantidad de ceros de f
        zer=zeros(ncero,1);          %Ceros para f[-x0,zer]

%-----RESTRICCIONES DE DESIGUALDAD -----%
109  G = blkdiag(kron(eye(N),[J,E]),JN);          %Matriz de desigualdad

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CONFIGURACION QUARC %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

114  %Esto es para que discretice correctamente la planta fisica
        tau_fil1 = tau_t1+T;
        tau_fil2 = tau_t2+T;

        % Correr setup_lab_tanks
119  % Correr MPC Sparse
        % Recordar incluir d_tanks1, d_tank2, setup_tank_parameters, en un mismo archivo

```

9.2. Código de ADMM

```
function u = admm(H,h,G,F,g,f)

%=====
% Quadratic programing (QP) problem solution:
%           min 0.5 x'*H*x + h'*x
5 %           s.t   F*x = f
%           G*x <= g
% x           : Optimal solution of qp problem
% val         : Value of the cost function in tk optimal
%=====

10
    n=size(H,1);
    q=size(G,1);
    p=size(F,1);

15
    x=1*ones(n,1);
    z=0.5*ones(q,1);
    tau=0.5*ones(q+p,1);

    A=[G; F];
20    B=[eye(q);zeros(p,q)];
    c=[g;f];
    ep=1e-9; ed=1e-9;

    rho=1.5; %%Se tiene que escoger este valor
25    rk=A*x+B*z-c;
    sk=rho*A'*B*(z); k=0;
    while norm(rk)>=ep || norm(sk)>=ed
        k=k+1;
        x=-(H+rho*A'*A)\(h+rho*A'*(B*z+tau-c));
30    z=max(0,-G*x-tau(1:q)+g);
        tau=tau+A*x+B*z-c;
        if k==1000; break; end %%Se tiene que escoger este valor
    end
    u = x(3);

35 end
```

Referencias

- [1] A. Parra Antillanca, “Implementación de la técnica de control predictivo utilizando admm para un sistema de estanques acoplados,” pp. 7–29, 2023.
- [2] dspace. (2020) Microlabbox brochure. [Online]. Available: https://www.dspace.com/en/ltd/home/medien/product_info/microlabbox_product-brochure.cfm
- [3] M. Ázua, “Implementación de un sistema de control predictivo usando el toolbox quarc,” pp. 25–36, 2022.
- [4] dSPACE. (2024) Control desk. [Online]. Available: <https://www.dspace.com/en/inc/home/products/sw/experimentandvisualization/controldesk.cfm>
- [5] G. Takács and B. Rohal-Ilkiv, *Model Predictive Vibration Control*. Springer, 2012, efficient constrained mpc vibration control for lightly damped mechanical structures Cap. 6, pp. 207–249.
- [6] J. A. D. D. Graham C. Goodwin, María M. Seron, *Constrained Control and Estimation*. Springer, 2005.
- [7] M. Hovd, “A brief introduction to model predictive controls.” Engineering Cybernetics Department NTNU, 2004.
- [8] A. L. Cedeño, “Control predictivo por modelo.” Angel L. Cedeño, 2020.
- [9] R. J. Mantz, “Observadores de estados.” Universidad nacional de la Plata, Argentina, Facultad de Ingeniería, departamento de Electrotecnica, 2003.
- [10] E. C. B. P. J. E. Stephen Boyd, Neal Parikh, *Distributed Optimization and Statistical Learning via The Alternating Direction Method of Multipliers*. Foundations and Trends in Machine Learning, 2010, pp. 7–20.
- [11] R. Tibshirani. (2009) Alternating directions method of multipliers. [Online]. Available: <https://www.stat.cmu.edu/~ryantibs/convexopt/lectures/admm.pdf>
- [12] L. V. Stephen Boyd, *Convex Optimization*. Cambridge University Press, 2004, pp. 10 – 725.
- [13] T. V. Dang, K. Ling, and J. Maciejowski, “Embedded admm-based qp solver for mpc with polytopic constraints,” in *2015 European Control Conference (ECC)*, 2015, pp. 3446–3451.
- [14] E. Wong, “Active-set method for quadratic programming.” University of California, San Diego, 2011.
- [15] A. B. F. Borrelli and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge: Cambridge University Press, 2017.
- [16] Quanser. (2012) Student workbook. [Online]. Available: <https://www.made-for-science.com/de/quanser/?df=made-for-science-quanser-coupled-tanks-coursewarestud-labview.pdf>

- [17] I. I. T. Dr. Antonio Ferremosca. (2018) Teoría de sistemas y control automático. [Online]. Available: https://frrq.cvg.utn.edu.ar/pluginfile.php/8717/mod_resource/content/0/T3_linealizar.pdf
- [18] K. Ogata, in *Modern Control Engineering*, vol. 5, 2009, pp. 29–43.
- [19] B. W. Karl J. Astrom, in *Computer-Controlled Systems Theory And Design*, vol. 3, 1996, pp. 30–66.
- [20] Quanser. (2012) User manual. [Online]. Available: <https://www.made-for-science.com/de/quanser/?df=made-for-science-quanser-coupled-tanks-usermanual.pdf>
- [21] AMD-Xinilinx. (2024) Vitis model composer. [Online]. Available: <https://www.xilinx.com/products/design-tools/vitis/vitis-model-composer.html>
- [22] M. Azua. (2023) Tutorial dspace - primeros pasos - usm elo. [Online]. Available: <https://www.youtube.com/watch?v=hO7xzqsI904>
- [23] F. A. Osorio Villalobos, “Simulación en tiempo real de un convertidor dc-dc flying capacitor de tres niveles,” 2022.