



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

BRL-PID: DISEÑO Y EVALUACIÓN DE UN CONTROLADOR PID DE GANANCIAS
ADAPTABLES MEDIANTE APRENDIZAJE REFORZADO BAYESIANO, APLICADO
AL CONTROL EN TIEMPO REAL DE UN ESTANQUE CÓNICO

TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN ELÉCTRICA
MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

PABLO ANDRÉS MARTÍN GONZÁLEZ

PROFESOR GUÍA:
MARCOS ORCHARD CONCHA

MIEMBROS DE LA COMISIÓN:
JORGE SILVA SÁNCHEZ
JUAN CARLOS TRAVIESO TORRES

SANTIAGO DE CHILE
2023

RESUMEN DE LA TESIS PARA OPTAR
AL GRADO DE MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN ELÉCTRICA
Y MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: PABLO ANDRÉS MARTÍN GONZÁLEZ
FECHA: 2023
PROFESOR GUÍA: MARCOS ORCHARD CONCHA

BRL-PID: DISEÑO Y EVALUACIÓN DE UN CONTROLADOR PID DE GANANCIAS
ADAPTABLES MEDIANTE APRENDIZAJE REFORZADO BAYESIANO, APLICADO
AL CONTROL EN TIEMPO REAL DE UN ESTANQUE CÓNICO

Las estrategias de control adaptable modifican los parámetros del controlador en el tiempo con el fin de cumplir con un índice de desempeño. La forma en que se adaptan varía dependiendo de la información disponible, la cual puede ser, por ejemplo, información histórica de la planta y/o una expresión analítica del sistema. Se propone una solución de control para un caso en que no se dispone de ambos, comparando el resultado con una configuración clásica de control, la cual sí dispone de dicho modelo y la posibilidad de calibración previa a la operación del sistema.

En esta investigación propone el BRL-PID (*Bayesian Reinforcement Learning PID*) para controlar sin modelar la planta mediante un esquema de aprendizaje bayesiano para la actualización de parámetros, en tiempo real, discutiendo luego las ventajas y desventajas de la propuesta.

*A mis padres y familia por su incondicional apoyo,
y a mis mentores y seres queridos por hacer del camino algo muy especial.*

Agradecimientos

Con esta entrega cumulo un sueño que me tomó 20 años concretar y no puedo sino partir agradeciendo a mis padres Jorge Martín y Haydeé González por un listado infinito de cosas, este logro es sin duda más suyo que mío. A mis hermanos Catalina y Jorge, ambos referentes en sus distintas formas y estilos. A toda mi familia por todo su apoyo en los momentos difíciles. A mi profesor guía, Marcos Orchard, y a Juan Carlos Travieso por su admirable paciencia y conocimiento. Al profesor Pablo Estévez por haberme dado la oportunidad de ser parte del cuerpo docente de la FCFM. A Sandra Céspedes, Rodrigo Muñoz, Matías Macaya y Matías Taucare por haberme recibido en NICLabs y ser parte de un tremendo equipo de trabajo. A René Dintrans y Jessica Forner por tanto. A los amigos y amigas de siempre: Alejandra, Anita y Daniela de CPV. A Muriel, Alessandra, Oriana, Daniel, Anibal, Roberto, Luis y a todos en Consistorial. A Susana Carrasco por tanto. A Claudia Gálvez. A mis referentes en la música, fuente de perspectiva e inspiración para salir adelante de cualquier situación: Beatriz Salazar, Rodolfo Olea, Igor Ledermann, Alejandro Silva, Christian Gálvez, Mario Miño y Sergio Silva. Al maestro Ignacio Quinteros (†). A mis compañeros y amigos de la FCFM en especial a Rodrigo, Albert, Diego, Valesska, Néstor, Samuel, José, Kevin, Esteban, Ricardo, Gabriel y Eduardo. A Miguel por todas las clases de electrónica y apoyo desinteresado. A Sonia y Juan y Pepe. A Elena, Jovanna y Juana (†). Finalmente, a mis queridos cuadrúpedos compañeros de vida.

Sin duda esta es la parte más difícil de este texto. La lista es gigante y siempre faltará un héroe o heroína del cotidiano que ayudó mucho más de lo que cree. A todos ustedes un profundo agradecimiento por ser parte de esta etapa.

Tabla de Contenido

1. Introducción	1
1.1. Motivación	1
1.2. Hipótesis	1
1.3. Objetivo General	2
1.4. Objetivos Específicos	2
1.5. Estructura de la Tesis	2
1.6. Presentación del Problema	3
2. Marco Teórico	4
2.1. Señales y Sistemas: Definiciones Básicas	4
2.1.1. Representación de Sistemas	6
2.2. Control de Sistemas	7
2.2.1. Conceptos Fundamentales	7
2.2.2. Estabilidad y Evaluación de Controladores	8
2.2.3. Controlador PID	12
2.2.4. Controladores Adaptables	14
2.3. Procesos Estocásticos	19
2.3.1. Métodos Secuenciales de Monte Carlo	19
2.4. Aprendizaje de Máquinas	20
2.4.1. Aprendizaje Reforzado Bayesiano	23
2.5. Planta de Prueba: Estanque Cónico	24
3. Estado del Arte	26
3.1. <i>Deep Adaptive Control With Online Identification For Industrial Robots</i> [1]	26
3.2. <i>SnAKe: Bayesian Optimization Via Pathwise Exploration</i> [2]	28
3.3. <i>MRAC-RL: A Framework For On-Line Policy Adaptation Under Parametric Model Uncertainty</i> [3]	29
3.4. <i>Soft Computing-Based Fuzzy Integral Sliding Mode Control: A Real-Time Investigation on a Conical Tank Process</i> [4]	31
3.5. <i>Study of Different Control Strategies Applied to a Second-Order Nonlinear Tank Process</i> [5]	34
3.6. <i>Level Control Of A Conical Tank Using The Fractional Order Control</i> [6]	35
4. Metodología	37
4.1. Descripción del Sistema y Supuestos Simplificatorios	37
4.2. Propuesta de Control	39

4.2.1.	Vector de Valores Posibles	39
4.2.2.	Selección e Implantación de Parámetros $\vec{\theta}$ del BRL-PID	41
4.2.3.	Evaluación y Actualización de las Distribuciones de Probabilidad de los Parámetros	42
4.2.4.	Filtro de la Referencia	43
4.2.5.	Rutinas de Memoria y Emergencia	45
4.3.	Diagrama de Flujo Simplificado del Algoritmo	47
5.	Definición del Caso de Estudio y Resultado de Simulaciones	48
5.1.	Señales de Referencia	48
5.2.	Degradación de Parámetros del Estanque Cónico	50
5.3.	Selección de Parámetros de Simulación	51
5.4.	Comparación de Resultados	52
5.5.	Resultados de Simulaciones	54
5.5.1.	Simulaciones 1: Referencia Escalón, Sin Ruido, Sin Memoria	55
5.5.2.	Simulaciones 2: Referencia Escalón, Con Ruido, Sin Memoria	56
5.5.3.	Simulaciones 3: Referencia Escalón, Sin Ruido, Con Memoria	59
5.5.4.	Simulaciones 4: Referencia Escalón, Con Ruido, Con Memoria	61
5.5.5.	Simulaciones 5: Referencia de Escalones Sucesivos, Sin Ruido, Sin Memoria	62
5.5.6.	Simulaciones 6: Referencia de Escalones Sucesivos, Sin Ruido, Con Memoria	65
5.5.7.	Simulaciones 7: Referencia de Escalones Sucesivos, Con Ruido, Sin Memoria	67
5.5.8.	Simulaciones 8: Referencia de Escalones Sucesivos, Con Ruido, Con Memoria	69
5.5.9.	Simulaciones 9: Referencia Módulo de Sinusoidal, Sin Ruido, Sin Memoria	72
5.5.10.	Simulaciones 10: Referencia Módulo de Sinusoidal, Con Ruido, Sin Memoria	74
6.	Discusión	76
7.	Conclusiones y Trabajo Futuro	78
7.1.	Conclusiones	78
7.2.	Trabajo Futuro	80
8.	Bibliografía	82
	Anexo	87

Índice de Tablas

2.1. Ganancias de un controlador PID mediante el método Ziegler-Nichols (ZN) [7].	14
5.1. Parámetros que se mantuvieron constantes entre las simulaciones del control del estanque cónico controlado por el BRL-PID y el PID clásico. El número indica la referencia utilizada, si se añadieron los ruidos de sensado y proceso, y si la memoria contaba con valores almacenados al iniciar la operación. Las letras A y B indican la configuración de los parámetros γ , τ_{mem} y $\sigma(e_{obj})$ del BRL-PID.	52
5.2. Parámetros de simulación que fueron modificados entre simulaciones para el control del estanque cónico mediante el BRL-PID y el PID clásico. El número indica la referencia, presencia de ruido y de valores en la memoria, y las letras A y B indican la configuración de los parámetros γ , τ_{mem} y $\sigma(e_{obj})$ del BRL-PID.	53
5.3. Resultados obtenidos en todas las simulaciones seleccionadas, indicando nombre de simulación, características de la referencia, presencia o ausencia de ruido en la planta y presencia o ausencia de datos almacenados en la memoria al inicio de la simulación, ratio de las métricas globales de desempeño (BRL-PID sobre PID clásico).	54
5.4. Resultados obtenidos en las simulaciones 1A y 1B: Referencia escalón simple, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.	55
5.5. Resultados obtenidos en las simulaciones 2A y 2B: Referencia escalón simple, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.	56
5.6. Resultados obtenidos en las simulaciones 3A y 3B: Referencia escalón simple, sin ruidos en la planta y con valores en la memoria al inicio de la simulación.	59
5.7. Resultados obtenidos en las simulaciones 4A y 4B: Referencia escalón simple, con ruidos en la planta y con valores en la memoria al inicio de la simulación.	61
5.8. Resultados obtenidos en las simulaciones 5A y 5B: Referencia sucesión de escalones, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.	63
5.9. Resultados obtenidos en las simulaciones 6A y 6B: Referencia sucesión de escalones, sin ruidos en la planta y con valores en la memoria al inicio de la simulación.	65
5.10. Resultados obtenidos en las simulaciones 7A y 7B: Referencia sucesión de escalones, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.	67

5.11. Resultados obtenidos en las simulaciones 8A y 8B: Referencia sucesión de escalones, con ruidos en la planta y con valores en la memoria al inicio de la simulación.	69
5.12. Resultados obtenidos en las simulaciones 9A y 9B: Referencia módulo de sinusoidal, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.	72
5.13. Resultados obtenidos en las simulaciones 10A y 10B: Referencia módulo de sinusoidal, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.	74

Índice de Ilustraciones

2.1.	Descripción gráfica simplificada de los elementos no lineales clásicos: curvaturas (formas de señal no lineales, gráfico a), zonas muertas (secciones de la señal nulas, gráfico b), saturaciones (límites estrictos del valor de señal, gráfico c), histéresis (el recorrido de A a B es distinto que de B a A, gráfico d) y fenómenos de salto (la señal cambia abruptamente de un valor a otro entre dos muestras consecutivas, gráfico e).	5
2.2.	Diagrama de bloques básico de un sistema de lazo cerrado compuesto de las señales $R(s)$, $E(s)$ y $C(s)$ y el bloque cuya función es $G(s)$, más un restador en el cierre del lazo [8].	6
2.3.	Diagrama de bloques para un sistema de control de lazo cerrado con sus partes principales (controlador, actuador, planta y sensor), y las señales de referencia $ref(k)$, error de control $e_c(k)$, señal de control $u(k)$, señal del actuador $f(k)$, salida de la planta $\tilde{y}(k)$ y salida de la planta sensada $y(k)$ [9][10][11].	8
2.4.	Parámetros de desempeño de la respuesta transiente de un sistema [8].	11
2.5.	Diagrama de bloques para un sistema de control con un PID clásico (e.i. de ganancias fijas).	13
2.6.	Diagrama del sistema de control SISO LTI al cual se le calcula el lugar geométrico de las raíces [12].	14
2.7.	Diagrama de bloques de controlador MRAC directo [13].	15
2.8.	Esquema de control adaptable por red neuronal, la cual se utiliza para el ajuste de los parámetros de un controlador PID de ganancias variables [14].	16
2.9.	Diagrama de bloques de un sistema de control adaptable por red neuronal, utilizada ésta como controlador [14].	16
2.10.	Diagrama de bloques de una sistema de control adaptable RRNN-PID con arquitectura <i>Feed-Forward</i> [15].	17
2.11.	Representación gráfica de la técnica por modos deslizantes, con $f(x, u^+)$ y $f(x, u^-)$ la variación de estados según su posición con respecto a la superficie de conmutación $\sigma(x)$ [16].	18
2.12.	Estructura básica de un algoritmo de ML [17].	21
2.13.	Diagrama del estanque cónico ubicado en el Laboratorio de Automática del Departamento de Ingeniería Civil Eléctrica de la Universidad de Chile [18].	25
3.1.	Arquitectura del controlador con <i>Enhanced OSBL</i> [1].	27
3.2.	Esquema del algoritmo <i>Enhanced OSBL</i> [1].	28
3.3.	Ejemplo gráfico de una iteración del algoritmo SnAKe [2].	29
3.4.	Diagramas de bloques de RL tradicional (a) y la propuesta del <i>paper</i> (b) [3].	31

3.5.	Diagramas de <i>piping</i> e instrumentación del estanque cónico de [4].	32
3.6.	Resultados de los conjuntos difusos para describir la dinámica de la planta considerando como entradas S y \dot{S} , y utilizando algoritmos genéticos para optimizar las pertenencias [4].	33
3.7.	Comparación entre el seguimiento de una referencia de un PI de ganancia fija, el sistema GSPI, y su versión difusa (FGSPI) en el tiempo [5].	34
3.8.	Respuesta al escalón del FOIMC con dos grados de libertad, usando PSO y WOA, utilizando J_1 en (a) , J_2 en (b) y J_3 en (c) [6].	36
4.1.	Diagrama de bloques de la propuesta BRL-PID. Las ganancias adaptables del PID se modifican utilizando el algoritmo propuesto hasta mientras se cumpla el índice de desempeño.	39
4.2.	Diagrama simplificado del funcionamiento del BRL-PID en el tiempo.	41
4.3.	Distribución de probabilidad para un parámetro θ_i . La línea punteada roja marca el máximo valor de probabilidad del parámetro, la verde indica el $\xi\%$ de la máxima probabilidad, utilizada como umbral para la selección de parámetros y la negra un caso excepcional que puede asumir el umbral durante el proceso de actualización y que amerita una acción especial del mismo.	42
4.4.	Curva para amortiguación de escalones del filtro de la referencia del BRL-PID.	44
4.5.	Ejemplos de casos problemáticos para la detección de una referencia de tipo escalón: diagonales conectadas por salto (izquierda arriba), saltos cercanos con respecto a la dinámica de la planta (derecha arriba) y escalones con oscilaciones (abajo).	45
4.6.	Diagrama de flujo simplificado del algoritmo de la propuesta BRL-PID para una ejecución de n_b bloques de duración.	47
5.1.	Referencia escalón simple (e.i. solo un salto) utilizada para las simulaciones de control del estanque cónico.	49
5.2.	Referencia de escalones sucesivos para las simulaciones de control del estanque cónico.	49
5.3.	Referencia módulo de sinusoidal para las simulaciones de control del estanque cónico.	50
5.4.	Ejemplo de la degradación de los parámetros del estanque cónico en el tiempo de acuerdo a las funciones de degradación arbitrarias.	51
5.5.	Simulación 1A: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia escalón simple, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.	55
5.6.	Simulación 1B: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia escalón simple, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.	56
5.7.	Simulación 2A: Respuesta de la planta controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia escalón simple, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.	57
5.8.	Simulación 2B (con alteración del parámetro de evaluación, $\gamma = 0,5$): Respuesta controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia escalón simple, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.	57

5.9. Simulación 2B (γ sin alterar para el caso B): Respuesta de la planta controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia escalón simple, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.	58
5.10. Simulación 2A: Dinámica de los parámetros del BRL-PID. Referencia escalón simple, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.	58
5.11. Simulación 2B (con alteración del parámetro de evaluación, $\gamma = 0,5$): Dinámica de los parámetros del BRL-PID. Referencia escalón simple, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.	59
5.12. Simulación 3A: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia escalón simple, sin ruidos en la planta y con valores en la memoria al inicio de la simulación.	60
5.13. Simulación 3B: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia escalón simple, sin ruidos en la planta y con valores en la memoria al inicio de la simulación.	60
5.14. Simulación 4A: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia sucesión de escalones, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.	61
5.15. Simulación 4B: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia sucesión de escalones, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.	62
5.16. Simulación 5A: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia sucesión de escalones, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.	63
5.17. Simulación 5B: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia sucesión de escalones, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.	64
5.18. Simulación 5A: Acercamiento de la respuesta de la planta controlada con el BRL-PID entre los 2120[s] y 2220[s] (referencia sucesión de escalones, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación). Se aprecian oscilaciones en la salida de planta debido a la rutina de emergencia una vez se supera un umbral superior de emergencia de 55[cm].	64
5.19. Simulación 5A: Acercamiento de la respuesta de la planta controlada con el BRL-PID entre los 4150[s] y 4190[s] referencia sucesión de escalones, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación). Se aprecian oscilaciones en la salida de planta debido a la rutina de emergencia una vez se supera un umbral inferior de emergencia de 5[cm].	65
5.20. Simulación 6A: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia sucesión de escalones, sin ruidos en la planta y con valores en la memoria al inicio de la simulación.	66

5.21. Simulación 6B: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia sucesión de escalones, sin ruidos en la planta y con valores en la memoria al inicio de la simulación.	66
5.22. Simulación 7A: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia sucesión de escalones, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.	67
5.23. Simulación 7B: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia sucesión de escalones, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.	68
5.24. Simulación 7A: Dinámica de los parámetros del BRL-PID. Referencia sucesión de escalones, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.	68
5.25. Simulación 7B: Dinámica de los parámetros del BRL-PID. Referencia sucesión de escalones, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.	69
5.26. Simulación 8A: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia sucesión de escalones, con ruidos en la planta y con valores en la memoria al inicio de la simulación.	70
5.27. Simulación 8B: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia sucesión de escalones, con ruidos en la planta y con valores en la memoria al inicio de la simulación.	70
5.28. Simulación 8A: Dinámica de los parámetros del BRL-PID. Referencia sucesión de escalones, con ruidos en la planta y con valores en la memoria al inicio de la simulación.	71
5.29. Simulación 8B: Dinámica de los parámetros del BRL-PID. Referencia sucesión de escalones, con ruidos en la planta y con valores en la memoria al inicio de la simulación.	71
5.30. Simulación 9A: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia módulo de sinusoidal, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.	72
5.31. Simulación 9B: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia módulo de sinusoidal, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.	73
5.32. Simulación 9A: Dinámica de los parámetros del BRL-PID. Referencia módulo de sinusoidal, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.	73
5.33. Simulación 9B: Dinámica de los parámetros del BRL-PID. Referencia módulo de sinusoidal, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.	74

5.34. Simulación 10A: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia módulo de sinusoidal, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.	75
5.35. Simulación 10B: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia módulo de sinusoidal, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.	75
7.1. Diagrama de funciones de variables aleatorias [19].	90
7.2. Gráfico de una actualización bayesiana [20].	90

Glosario de Términos

- ADC: Conversor Análogo Digital.
- AE: *Auto Encoder*.
- BRL-PID: *Bayesian Reinforcement Learning PID*.
- CDF: *Cumulative Density Function*.
- DCA: Conversor Digital Análogo.
- DDPG: *Deep Deterministic Policy Gradient*.
- ECM: Error Cuadrático Medio.
- FF: *Feed Forward*.
- FGSPi: *Fuzzy Gain Scheduled PI*.
- FISMC: *Fuzzy Integral Sliding Mode Controller*.
- FOIMC: *Fractional Order Internal Model Control*.
- FOPDT: *First Order Plus Dead Time*.
- FOPID: *Fractional Order PID*.
- FST-PSO: *Fuzzy Self-Tuning with Particle Swarm Optimisation*.
- GA: *Genetic Algorithm*.
- GAFISMC: *Genetic Algorithm Fuzzy Integral Sliding Mode Control*.
- GSPI: *Gain Scheduled PI*.
- HMM: *Hidden Markov Model*.
- IA: Inteligencia Artificial.
- IAE: *Integral Absolute Error*.
- IMC: *Internal Model Control*.
- ISE: *Integrated Squared Error*.
- ISI: *Integrated Squared Input*.
- ISMC: *Integral Sliding Mode Control*.
- ITAE: *Integrated Time Multiplied Absolute Error*.
- ITSE: *Integral of Time Multiplied Square-Error*.
- LGR: Lugar Geométrico de las Raíces.
- LSE: *Least Squared Error*.
- LSTM: *Long/Short Term Memory Cell*.
- LTI: Lineal e Invariante en el Tiempo.
- LUT: *Look Up Table*.

- MAP: *Máximo a Posteriori*.
- MC: *Markov Chain*.
- MCMC: *Markov Chain Monte Carlo*.
- MIMO: *Multiple Input, Multiple Output*.
- MISO: *Multiple Input, Single Output*.
- MIT: *Massachusetts Institute of Technology*.
- ML: *Machine Learning*.
- MRAC: *Model Reference Adaptive Control*.
- MRAC-RL: *Model Reference Adaptive Control Reinforcement Learning*.
- MRAS: *Model Reference Adaptive System*.
- ARMAX: *Autoregressive Moving Average with Exogenous Inputs*.
- NB: *Negative Big*.
- NLL: *Negative-log Likelihood*.
- NM: *Negative Medium*.
- NN-MRAC: *Neural Network Model Reference Adaptive Control*.
- NS: *Negative Small*.
- OBSID: *Optimised Bayesian System Identification*.
- OSBL: *Online Sparse Bayesian Learning*.
- PB: *Positive Big*.
- PDF: *Probability Density Function*.
- PID: *Controlador Proporcional, Integral, Derivativo*.
- PM: *Positive Medium*.
- PMF: *Probability Mass Function*.
- POMPD: *Partially Observable Markov Decision Process*.
- PPO: *Proximal Policy Optimization*.
- PS: *Positive Small*.
- PSO: *Particle Swarm Optimization*.
- RBFN: *Radial Basis Function Network*.
- RNA : *Red Neuronal Artificial*.
- RL: *Reinforcement Learning*.
- RNN: *Recurrent Neural Network*.
- RRNN: *Redes Neuronales*.
- SAC: *Soft Actor-Critic*.
- SIMO: *Single Input, Multiple Output*.
- SIS: *Sequential Importance Sampling*.
- SISO: *Single Input, Single Output*.
- SMC: *Sequential Monte Carlo*.
- SnAke: *Sequential Bayesian Optimization Via Adaptive Connecting Samples*.
- SNDP: *Supervised Neural Dynamic Programming*.

- SVM: *Support Vector Machine*.
- WOA: *Whale Optimization Algorithm*.
- ZN: Ziegler-Nichols.

Capítulo 1

Introducción

1.1. Motivación

Las estrategias de control adaptable requieren de datos históricos o un modelo de planta que permita decidir una estrategia adecuada de adaptación, lo cual puede requerir una gran cantidad de trabajo previo y/o capacidad de cómputo [13][21][22]. Surge entonces la interrogante si acaso es posible lograr la adaptación de parámetros de un controlador PID de ganancias adaptables (elegido por lo masivo de su uso [8]) y el control de una planta con la menor cantidad de información y procesamiento previo, en tiempo real.

La dificultad del problema radica en su incertidumbre, por lo que se requiere de una revisión de técnicas avanzadas de control, siendo los esquemas de aprendizaje bayesiano candidatos naturales por cuanto representan de forma sencilla como distribuciones de probabilidad, las cuales se actualizan conforme se recibe nueva información [23]. Este trabajo explora los límites de acción de este tipo de soluciones y generar un aprendizaje sobre cuáles son los puntos de mayor complejidad ante problemas con alta incertidumbre. Se genera entonces una propuesta de control, la cual se comparó con una solución clásica que sí cuenta con información y procesamiento previo, en un caso de simulación, como una forma de establecer la utilidad relativa con respecto a un sistema ampliamente utilizado.

1.2. Hipótesis

Es posible diseñar un sistema de control consistente en un PID de ganancias ajustables que, mediante un sistema de actualización de parámetros del controlador que incorpore un esquema de aprendizaje reforzado bayesiano y sin conocer la dinámica de la planta (la cual presenta ruido de proceso y observación) ni modelarla, pueda lograr un rendimiento comparable al que se puede conseguir mediante un sistema de control clásico que sí dispone de un modelo de planta para la calibración de su controlador.

1.3. Objetivo General

Diseñar un sistema de control SISO para un PID de ganancias ajustables que, mediante un sistema de actualización de parámetros que incorpore un esquema de aprendizaje bayesiano, sea capaz de controlar en tiempo real una planta desconocida en presencia de ruido de observación y proceso, y cuyo rendimiento sea como máximo un 30 % peor que el obtenido por un control PID de ganancias constantes que sí cuenta con un modelo de planta y la opción de calibración previa para el mismo proceso, tomando como referencia las métricas IAE, ITAE, ISE e ITSE, en un caso de simulación del estanque cónico del laboratorio de automática del Departamento de Ingeniería Eléctrica de la Universidad de Chile.

1.4. Objetivos Específicos

1. Determinar las características de los sistemas no lineales que puede tener una planta en un sistema de control, en particular para el estanque cónico utilizado como planta de prueba.
2. Identificar las estrategias de control adaptable y de aprendizaje bayesiano existentes más adecuadas para el problema a resolver.
3. Diseñar un experimento para la evaluación del sistema propuesto y establecer sus métricas.
4. Desarrollar una simulación para describir el comportamiento tanto de la propuesta BRL-PID como el de un PID de ganancias constantes, calibrado mediante el método Ziegler-Nichols, para el control de una planta de prueba (el estanque cónico del Laboratorio de Automática del Departamento de Ingeniería Civil Eléctrica de la Universidad de Chile).
5. Establecer métricas de desempeño y comparar ambos controladores.
6. Identificar las ventajas y desventajas de la propuesta, al igual que sus mejoras posibles.

1.5. Estructura de la Tesis

En el Capítulo 1 se presenta la motivación del estudio, la hipótesis de investigación, sus objetivos, la estructura del documento y la presentación del problema para el cual se desarrollará el algoritmo propuesto. En el Capítulo 2, el Marco Teórico, se exponen los conceptos básicos de señales, control de sistemas, métodos secuenciales de Montecarlo e inferencia bayesiana, concluyendo el capítulo con la definición y caracterización de aprendizaje de máquinas con énfasis en aprendizaje reforzado bayesiano. En el Capítulo 3 se expone una selección de investigaciones recientes y afines al problema de investigación. En el Capítulo 4 se formaliza el problema y se describe el funcionamiento del algoritmo. En el Capítulo 5 se define el caso de estudio y se presentan los resultados de simulación. En el Capítulo 6 se presenta la discusión de los resultados, y finalmente en el Capítulo 7 se exponen las conclusiones y mejoras propuestas para la investigación.

1.6. Presentación del Problema

Considérese una planta SISO de entrada y salida reales, discretas y acotadas, cuya salida presenta ruido de proceso aditivo de amplitud máxima conocida y distribución desconocida, degradación de sus parámetros internos de dinámica desconocida y que puede presentar no linealidades de naturaleza y frecuencia desconocida. Los sensores para observar la salida de la planta presentan ruido de observación aditivo de amplitud máxima conocida y distribución desconocida. No se cuenta con datos históricos de la planta. El controlador es un PID adaptable de parámetros $\vec{\theta} = \{P, I, D\} \in \mathbb{R}^3, \vec{\theta} \geq 0$.

En este contexto se requiere diseñar un mecanismo de ajuste de parámetros del controlador mediante un esquema de aprendizaje reforzado bayesiano. Se comparará este resultado con una solución clásica (PID con ganancias constantes que sí cuenta con un modelo de planta y la opción de calibración previa), fijando como meta que el rendimiento de la propuesta sea a lo más un 30 % peor que el obtenido por la solución clásica, tomando como referencia las métricas IAE, ITAE, ISE e ITSE en un caso de simulación del estanque cónico del laboratorio de automática del Departamento de Ingeniería Civil Eléctrica de la Universidad de Chile.

Capítulo 2

Marco Teórico

Se presenta a continuación una selección de los contenidos estudiados para la elaboración de la propuesta, incluyendo tópicos de señales y sistemas, control de sistemas, aprendizaje de máquinas y esquemas de aprendizaje en el contexto de procesos estocásticos, con énfasis en el enfoque bayesiano.

2.1. Señales y Sistemas: Definiciones Básicas

Modelar se refiere al proceso de representar un ente de forma simplificada, con el fin de comunicar información relevante sobre un fenómeno determinado [10][24]. Un **sistema** se define como una colección de entes a los cuales se desea estudiar sus propiedades. Los sistemas se pueden clasificar de acuerdo a distintos criterios, destacando los siguientes [10][13][25][26][27][28]:

1. **Naturales o artificiales:** si la existencia del mismo depende o no de la acción del hombre.
2. **Deterministas o estocásticos:** si la relación entre las variables medidas y derivadas posee o no incertidumbre.
3. **Dinámico o estático:** si existe variación temporal del fenómeno observado o no.
4. **Causal:** Un sistema será causal si la respuesta del sistema solo depende de variables pasadas o presentes, pero no futuras. La mayoría de los sistemas de la naturaleza son causales.
5. **Con o sin memoria:** Si la salida en un tiempo determinado depende de valores obtenidos en instantes anteriores o no.
6. **Lineales o no lineales:** un sistema lineal satisface los principios de **homogeneidad** y **aditividad**, lo que permite tratarlas aplicando el **principio de superposición**. En un fenómeno no lineal no ocurre lo anterior.
7. **Mono o multivariable:** dependiendo si un sistema tiene una o más variables en su entrada o salida.
8. **Estables o inestables:** un sistema es estable si cumple con algún criterio de estabilidad establecido, como por ejemplo los criterios de estabilidad según Lyapunov.

9. **Distribuidos o concentrados:** en el caso de sistemas eléctricos, si las dimensiones físicas del sistema son relevantes en la evolución de las señales involucradas.
10. **De variables continuas, discretas o mixto:** según la naturaleza temporal de las variables involucradas.

En esta investigación se aborda el problema para un **sistema natural, estocástico, dinámico, causal, con memoria, no lineal, mono variable en la entrada y salida, el cual podrá ser estable o inestable.**

Por otra parte, una **señal** se define como una función que contiene información de un fenómeno físico, en este caso de una planta o proceso, aportando información sobre la naturaleza de cada dispositivo o ente del sistema a controlar [10][25][27]. En esta investigación existen **elementos no lineales** presentes en las señales como **curvaturas, zonas muertas, saturaciones, histéresis y fenómenos de salto**, los cuales se describen gráficamente en la Figura 2.1.

Descripción Gráfica Simplificada de
Elementos No Lineales Presentes en las Señales de Sistemas

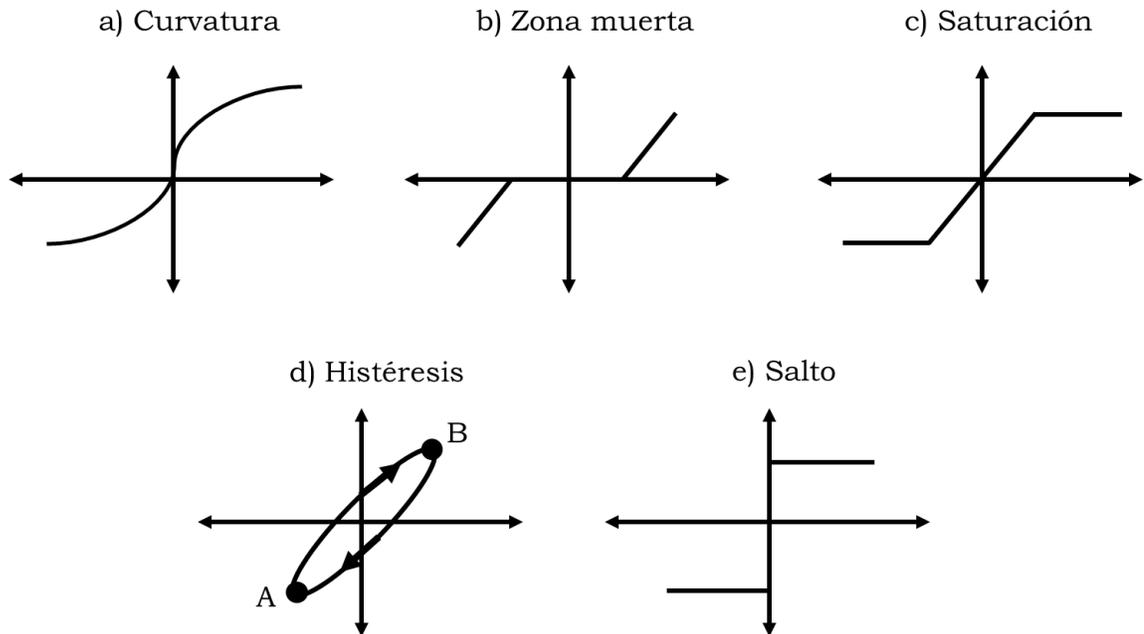


Figura 2.1: Descripción gráfica simplificada de los elementos no lineales clásicos: curvaturas (formas de señal no lineales, gráfico a), zonas muertas (secciones de la señal nulas, gráfico b), saturaciones (límites estrictos del valor de señal, gráfico c), histéresis (el recorrido de A a B es distinto que de B a A, gráfico d) y fenómenos de salto (la señal cambia abruptamente de un valor a otro entre dos muestras consecutivas, gráfico e).

2.1.1. Representación de Sistemas

Las formas típicas de representar un sistema es mediante señales en la forma de **funciones en el dominio del tiempo** o **funciones de transferencia (dominio de la frecuencia)** y **representación en variables de estado**. Los **diagramas de bloques** se utilizan para explicitar los componentes del sistema y sus conexiones mediante dichas señales [8].

Las **funciones en el dominio del tiempo** representan la dinámica del fenómeno. La ecuación 2.1 exhibe un ejemplo de una función de este tipo.

$$f(t) = P \cdot e(t) + \int_0^t e(t)dt, \quad t \in \mathbb{N}, P \in \mathbb{R}^+ \quad (2.1)$$

En un sistema LTI, una **función de transferencia** es la transformada de Laplace de la salida, dividido por la transformada de Laplace de la entrada, obteniendo así una expresión en el dominio de la frecuencia [8]:

$$H(s) = \frac{\mathcal{L}(y(t))}{\mathcal{L}(x(t))} = \frac{Y(s)}{X(s)} \quad (2.2)$$

Los **diagramas de bloques** son representaciones pictóricas en donde los bloques son los componentes del sistema y las flechas que los conectan representan los flujos de señal entre éstos, como se observa en el ejemplo de la Figura 2.2 [8].

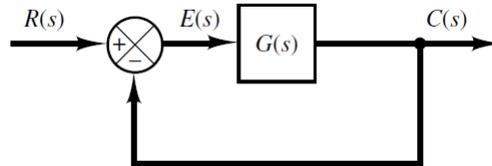


Figura 2.2: Diagrama de bloques básico de un sistema de lazo cerrado compuesto de las señales $R(s)$, $E(s)$ y $C(s)$ y el bloque cuya función es $G(s)$, más un restador en el cierre del lazo [8].

El **estado** de un sistema se refiere a un grupo de variables del sistema, llamadas **variables de estado**, tales que, teniendo conocimiento de sus valores y la entrada del sistema en $t = t_0$, permiten conocer la dinámica de éste. Nótese que las variables de estado **no necesariamente son de naturaleza física**, permitiendo considerar otros aspectos del problema para optimizar el control, como por ejemplo el costo económico de la solución. Puede ocurrir además que no todas sean medibles u observables. En su forma genérica, para las entradas $\{u_1, \dots, u_r\}$, las salidas $\{y_1, \dots, y_m\}$ y variables de estado $\{x_1, \dots, x_n\}$, las ecuaciones de estado de un sistema son [8]:

VARIABLES DE ESTADO:

$$\dot{x}_1 = f_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \quad (2.3)$$

$$\dot{x}_2 = f_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \quad (2.4)$$

⋮

$$\dot{x}_n = f_n(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \quad (2.5)$$

SALIDAS DEL SISTEMA:

$$y_1 = g_1(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \quad (2.6)$$

$$y_2 = g_2(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \quad (2.7)$$

⋮

$$y_m = g_m(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \quad (2.8)$$

En el tratamiento de sistemas de control clásicos es frecuente la utilización de modelos lineales en el dominio del tiempo y la frecuencia. **El BRL-PID trabaja con las representaciones en el tiempo y diagramas de bloques.**

2.2. Control de Sistemas

2.2.1. Conceptos Fundamentales

Los sistemas de control son aquellos que, mediante la manipulación de sus variables, intentan asegurar un comportamiento determinado en una planta o proceso [9][10][25]. Se pueden representar usando diagramas de bloques que interactúan entre sí mediante señales procesadas por éstos [8][9][10] y, a grandes rasgos, se componen de: una **planta o proceso** a controlar, un **controlador**, una **referencia** a seguir por la planta, **sensores** que capturan datos del entorno y **actuadores** que ejecutan las acciones de control, las cuales derivan de la **ley de control**, función que describe el proceso interno del controlador. También puede haber **perturbaciones** tanto en el proceso como en la observación [9][10]. Podrá haber también **convertidores análogo-digitaes y digitaes-análogos** (llamados comúnmente ADC y DAC, respectivamente) [9]. Existen sistemas de control de lazo abierto o cerrado, siendo la diferencia entre ambos que en el segundo caso se devuelve al sistema de control información de la salida de la planta, con lo cual se puede saber si se está logrando un buen resultado o no con respecto a la referencia, mediante el cálculo del **error de control** [9]. Dependiendo de la cantidad de señales de entradas y salidas se definen los términos **SISO** (*Single Input, Single Output*), **MIMO** (*Multiple Input, Multiple Output*), **SIMO** (*Single Input, Multiple Output*) y **MISO** (*Multiple Input, Single Output*) [11]. La Figura 2.3 expone una representación de un sistema de control de lazo cerrado mediante diagrama de bloques.

En la presente investigación se considera un sistema SISO de señales discretas. Las **referencias** utilizadas han sido seleccionadas con el objetivo de testear el desempeño del BRL-PID, la **planta** es el estanque cónico, el **bloque de control** será un PID de ganancias fijas para el caso de calibración por ZN, y de ganancias variables para la propuesta BRL-PID, el **actuador** es variador de frecuencia que modifica el flujo de líquido, siendo el porcentaje de variación su salida. La elección de un PID como controlador responde a que es masivamente utilizado [8]. Asimismo, la selección de la planta responde a que se buscó un sistema real, no lineal y con experimentación previa existente [4][5].

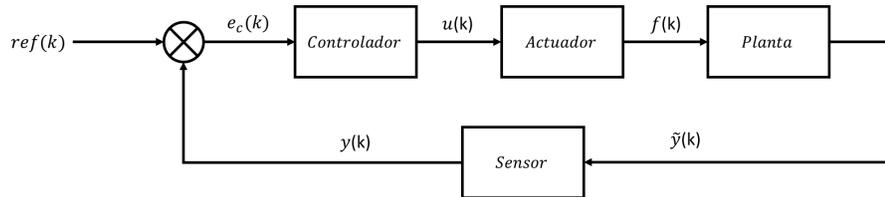


Figura 2.3: Diagrama de bloques para un sistema de control de lazo cerrado con sus partes principales (controlador, actuador, planta y sensor), y las señales de referencia $ref(k)$, error de control $e_c(k)$, señal de control $u(k)$, señal del actuador $f(k)$, salida de la planta $\tilde{y}(k)$ y salida de la planta sensada $y(k)$ [9][10][11].

Se dice que una planta es **controlable** si se puede llevar de un estado A a otro B en un tiempo determinado y finito. Un sistema se considera **observable** si es posible conocer su estado inicial a partir de la observación de la entrada y salida durante un intervalo de tiempo finito posterior al inicial [8]. A finales de la década de 1950 se estableció en la teoría de control el uso de un **índice de desempeño**, función que permite determinar si la planta controlada se comporta de forma deseada, es decir, representa en su formulación los requisitos de control [13].

Existen diferentes técnicas de control dependiendo del objetivo, la información y herramientas disponibles. Para establecer los objetivos se plantean **conceptos y criterios de estabilidad**, para luego decidir la estrategia de control adecuada.

2.2.2. Estabilidad y Evaluación de Controladores

El concepto de **equilibrio** se refiere a un estado estacionario en el que el sistema mantiene su comportamiento, lo cual en el caso de la representación de variables de estado se traduce en que no hay variación de éstas [29]. La **estabilidad** se refiere a la capacidad del control para mantener el sistema en torno a un estado de equilibrio, a pesar de las perturbaciones o cambios externos [29]. Se define la **estabilidad asintótica** como aquella que ocurre si el vector de estado converge al punto de equilibrio cuando el tiempo tiende a infinito; un sistema será **localmente estable** (asintóticamente o no) si la estabilidad se obtiene cuando el sistema se mantiene en una vecindad del punto de equilibrio, y será **globalmente estable** si se puede lograr la estabilidad para cualquier situación inicial [29]. **Los criterios de estabilidad dependen de una función que represente qué se desea optimizar**, lo cual a su vez dependerá de los objetivos de diseño. Por ejemplo, los criterios basados en almacenamiento de energía suelen demostrar que estas funciones se mantienen acotadas o decrecen a lo largo de las trayectorias del sistema [28]. Criterios clásicos como Routh-Hurwitz y Jury, que no se

pueden aplicar directamente a sistemas no autónomos, se pueden adaptar para analizar la **estabilidad en el dominio de la frecuencia** [29].

Criterios de Lyapunov

Se dice que el equilibrio del sistema (se asume $x = 0$) es **estable en el sentido de Lyapunov** si para cada $\varepsilon > 0$ existe un valor $\delta > 0$ (posiblemente dependiente de ε) tal que:

$$\|x(0)\| < \delta \Rightarrow \|x(t)\| < \varepsilon, \forall t \geq 0 \quad (2.9)$$

Es decir, si el estado inicial está suficientemente cerca del equilibrio, la trayectoria no se aleja críticamente del mismo [29].

Los métodos de Lyapunov se basan en encontrar funciones que se vinculan a las del sistema a controlar y cuyas características permitan concluir la estabilidad de éste [28][29]. El primer método (indirecto), establece que la estabilidad en un punto de equilibrio de un sistema no lineal descrito por las Ecuaciones 2.10 y 2.11, puede determinarse analizando la linealización por serie de Taylor de $f(x)$ en torno al punto de equilibrio, despreciando los términos de segundo orden y superiores, de manera que la aproximación es de la forma descrita por la Ecuación 2.12.

$$\dot{\vec{x}}(t) = \vec{f}(\vec{x}) \quad (2.10)$$

$$\vec{x}(t_0) = \vec{x}_0 \quad (2.11)$$

$$\tilde{x}(t) = J(\tilde{x}(t)) \quad (2.12)$$

Con $J(\tilde{x}(t))$ el jacobiano de $x(t)$ evaluado en el punto de equilibrio. Entonces, el teorema de Lyapunov establece que:

- Si el sistema aproximado es asintóticamente estable, entonces el original es asintóticamente estable.
- Si el sistema aproximado es inestable, entonces el sistema es inestable.
- Si el sistema aproximado es estable, pero no asintóticamente, no hay información suficiente para determinar la estabilidad y se deberá analizar con los términos de orden superior de Taylor.

Las limitaciones de este método son que sólo se puede asegurar estabilidad local, y que para utilizarlo es necesario que exista la aproximación lineal de primer orden. En general no serán aplicables si existen discontinuidades de tipo umbral o relé.

El segundo método (directo), indica que si existe una función escalar $V(x, t)$ continua y con primeras derivadas parciales continuas con respecto a x y t , tal que en una vecindad del punto de equilibrio se cumple que $V(x, t)$ es definida positiva (e.i. $V(x, t) > 0$), entonces:

- Si $\dot{V}(x, t) = \nabla V \cdot \dot{x} \leq 0$, el sistema es estable en el punto de equilibrio.
- Si $\dot{V}(x, t) = \nabla V \cdot \dot{x} < 0$, el sistema es asintóticamente estable en el equilibrio.
- Si $\dot{V}(x, t) = \nabla V \cdot \dot{x} < 0$ y $V(x) \rightarrow \infty$ cuando $x \rightarrow \infty$, entonces es globalmente asintóticamente estable.

Este método permite estudiar la estabilidad de sistemas a partir de las ecuaciones del mismo, pero sin tener que resolverlas, estableciendo condiciones suficientes y necesarias para la estabilidad, es decir, si se puede encontrar una región del espacio donde se verifiquen las condiciones descritas, entonces $f(\vec{x})$ es estable en ésta, de lo contrario no se puede asegurar la estabilidad. **Las funciones de Lyapunov no son únicas y existen distintos métodos para obtenerlas, como por ejemplo tanteo o variación de gradiente [29].**

Respuesta Transiente

Los parámetros de la respuesta transiente entregan información sobre la conducta del sistema al cambiar de un estado otro (generalmente ante un cambio de escalón en la referencia), siendo otra forma de evaluación del desempeño de controladores [8]. Se detallan a continuación las definiciones de cada uno de éstos y en la Figura 2.5 se indican gráficamente [8].

1. **Delay time (t_d):** tiempo requerido para que la salida del sistema llegue al 50 % de su valor referencial.
2. **Rise time (t_r):** tiempo requerido para llegar del 10 % al 90 %, del 5 % al 95 % o del 0 % al 100 % del valor final, según el criterio escogido.
3. **Peak time (t_p):** tiempo que demora la señal en llegar al valor más alto de sobrepaso con respecto a la referencia.
4. **Maximum percent overshoot (M_p):** *peak time* expresado como un porcentaje de la respuesta en estado estacionario para una respuesta al escalón:

$$M_p = \frac{y(t_p) - y(\infty)}{y(\infty)} \cdot 100\% \quad (2.13)$$

5. **Settling time (t_s):** tiempo requerido para que la salida de la planta se mantenga dentro de un rango de error con respecto a la referencia. Se usa normalmente un rango de tolerancia del 2 % o 5 %.

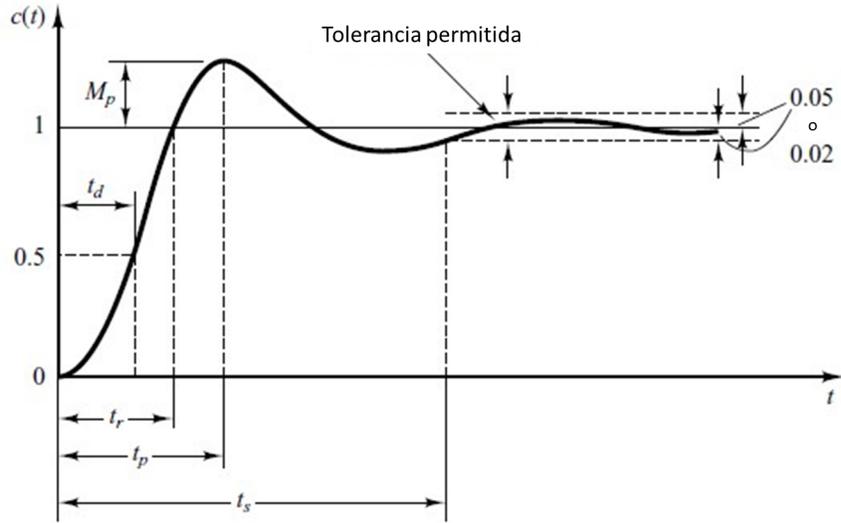


Figura 2.4: Parámetros de desempeño de la respuesta transiente de un sistema [8].

Métricas de Desempeño Global

Las métricas de desempeño global se basan en la dinámica del error y el esfuerzo de control. Se definen entonces los índices IAE (*Integrated Absolute Error*), ISE (*Integrated Squared Error*), ITAE (*Integrated Time Multiplied Absolute Error*) e ITSE (*Integral of Time Multiplied Square-Error*) [30]:

$$IAE = \int_0^T |e(t)| dt \quad (2.14)$$

$$ITAE = \int_0^T t |e(t)| dt \quad (2.15)$$

$$ISE = \int_0^T e(t)^2 dt \quad (2.16)$$

$$ITSE = \int_0^T t e(t)^2 dt \quad (2.17)$$

$$ISI = \int_0^T u(t)^2 dt \quad (2.18)$$

Los índices IAE e ISE aumentan si el módulo del error no converge a cero en el tiempo, y los índices ITAE e ITSE agregan una penalización sobre el tiempo de estabilización, y el índice ISI representa el gasto energético del controlador. Utilizando aproximación rectangular se generan las versiones discretas de los índices, siendo T_s el período de muestreo del sistema:

$$IAE = \sum_{i=0}^T |e(i)|T_s \quad (2.19)$$

$$ITAE = \sum_{i=0}^T i|e(i)|T_s \quad (2.20)$$

$$ISE = \sum_{i=0}^T e(i)^2T_s \quad (2.21)$$

$$ITSE = \sum_{i=0}^T ie(i)^2T_s \quad (2.22)$$

$$ISI = \sum_{i=0}^T u(i)^2T_s \quad (2.23)$$

Dado que en la propuesta BRL-PID se asume que la planta es una caja negra y no se dispone de datos históricos que permitan inferir su dinámica más allá del rango de valores en los que se desenvuelve, sumado a la variedad de referencias que se utilizarán, **se decidió trabajar con las métricas globales de desempeño IAE, ITAE, ISE, ITSE e ISI**, permitiendo así tener una métrica que permita comparar todos los casos revisados.

2.2.3. Controlador PID

En 1922, Nicholas Minorsky formuló la ley de control que generó el controlador PID, inspirado en cómo los timoneles de barcos manejaban sus naves [21], siendo en la actualidad un controlador lineal ampliamente utilizado [8]. Una de las formas en que se expresa la ley de control es la siguiente, para $t \in \mathbb{R}, t \geq 0$:

$$u(t) = P \cdot e(t) + I \cdot \int_0^t e(x)dx + D \cdot \frac{\partial e(t)}{\partial t} \quad (2.24)$$

Con P, I y D los parámetros del controlador, valores reales no negativos que representan la importancia del error instantáneo, el acumulado y su variación temporal en la respuesta del sistema, respectivamente. En el caso discreto, utilizando discretización rectangular y siendo T_s el período de muestreo, el controlador toma la forma descrita en la Ecuación 2.25. Luego, para $k \in \mathbb{N}$:

$$u(k) = P \cdot e(k) + I \cdot \sum_{i=1}^k e(i) \cdot T_s + D \cdot \frac{e(k) - e(k-1)}{T_s} \quad (2.25)$$

Un PID clásico tiene sus ganancias fijas, mientras que en el caso de un PID adaptable son generalmente constantes y sus valores varían dependiendo del mecanismo de ajuste [13][31].

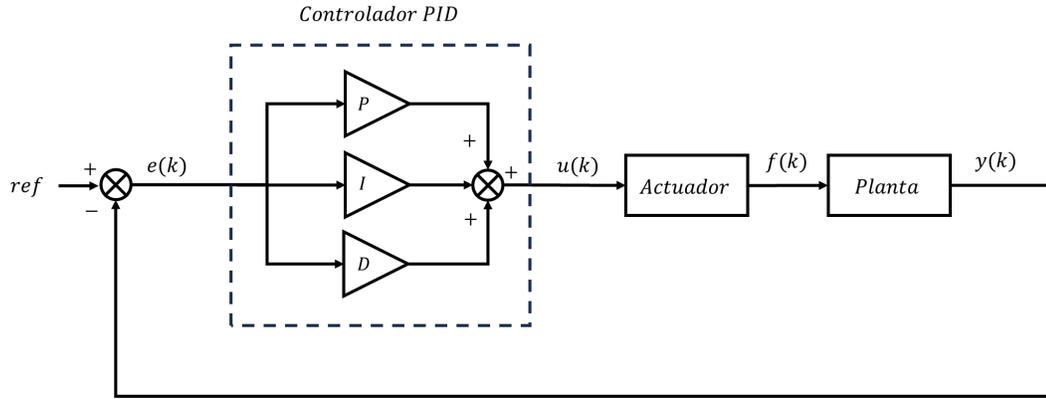


Figura 2.5: Diagrama de bloques para un sistema de control con un PID clásico (e.i. de ganancias fijas).

Calibración de un Controlador PID Clásico: Ensayo y Error, LGR y Ziegler-Nichols

Para calibrar un PID clásico existen distintos métodos de calibración, de los cuales se exponen a continuación: ensayo y error, LGR y Ziegler-Nichols. La literatura consultada también menciona la existencia otros métodos como el de Cohen-Coon, Chien-Hrones-Reswick y el método Kappa-Tau [32].

Ensayo y Error

Consiste en probar valores empíricamente y modificarlos dependiendo del resultado obtenido. Depende principalmente de la experiencia del operador a cargo.

Lugar Geométrico de las Raíces (LGR)

Sea un sistema SISO LTI tal que $G(s)$ es la función de transferencia correspondiente a la planta y el controlador, y $H(s)$ la de un dispositivo de observación. En estas condiciones, la función de transferencia de este sistema viene dado por la Ecuación 2.26 ([11]).

$$\frac{Y(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)} \quad (2.26)$$

La función $G(s)H(s)$ es igual al siguiente polinomio, con K la ganancia del sistema, $n \geq m$ (función propia), $p_j, j = \{1, \dots, l\}$ los polos y $z_i, i = \{1, \dots, m\}$ los ceros de lazo abierto [11].

$$G(s)H(s) = \frac{K \prod_{i=1}^m (s + z_i)}{s^n \prod_{j=1}^l (s + p_j)} \quad (2.27)$$

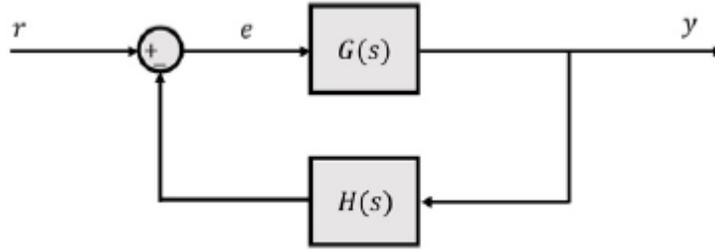


Figura 2.6: Diagrama del sistema de control SISO LTI al cual se le calcula el lugar geométrico de las raíces [12].

Reemplazando 2.27 en 2.26 y despejando:

$$\prod_{j=1}^l (s + p_j) + K \prod_{i=1}^m (s + z_i) = 0 \quad (2.28)$$

Con z_i los ceros de lazo abierto y p_j los polos de lazo abierto. La solución de la ecuación 2.28 entrega los polos de lazo cerrado. Cuando $K \rightarrow 0$ la posición de los polos de lazo cerrado es la misma que la de los polos de lazo abierto, y si $K \rightarrow \infty$, los polos de lazo cerrado están en la misma posición que los ceros de lazo abierto [11].

Ziegler-Nichols

Es un método heurístico el cual se inicia llevando a cero las ganancias integral y derivativa, elevando la proporcional hasta que el sistema se vuelva inestable, definiendo entonces P_{MAX} como dicho valor, en el cual el sistema oscilará con frecuencia constante f . Con esta información se determinan los siguientes valores para cada ganancia según sea el caso [7].

Controlador / Ganancia	P	I	D
P	$0.5 \cdot P_{MAX}$	0	0
PI	$0.45 \cdot P_{MAX}$	$0.54 \cdot P_{MAX} \cdot f$	0
PD	$0.8 \cdot P_{MAX}$	0	$0.1 \cdot P_{MAX} \cdot f$
PID	$0.6 \cdot P_{MAX}$	$1.2 \cdot P_{MAX} \cdot f$	$0.075 \cdot P_{MAX} \cdot f$

Tabla 2.1: Ganancias de un controlador PID mediante el método Ziegler-Nichols (ZN) [7].

2.2.4. Controladores Adaptables

Un control adaptable o adaptativo es aquél capaz de modificar su estructura o parámetros con el fin de cumplir un índice de desempeño [13]. El cómo se realicen los cambios dependerá del índice establecido, las restricciones del problema, la fenomenología de la planta y las decisiones de diseño, por lo que hay infinitas formas de diseñar este tipo de controladores, si bien existen en la literatura configuraciones conocidas, de las cuales se presenta una selección.

Los Controladores Adaptables por Modelo de Referencia (MRAC por su nombre en inglés, *Model Reference Adaptive Control*) son un tipo de control adaptable cuyo objetivo es que la planta y_p se comporte como un modelo de referencia establecido por diseño y_m , los cuales se pueden definir, por ejemplo, de acuerdo a la siguiente estructura:

$$\dot{y}_p(t) = -a_p y_p(t) + k_p u(t) \quad (2.29)$$

$$\dot{y}_m(t) = -a_m y_m(t) + b_m \text{ref}(t), \text{ con } a_m > 0 \quad (2.30)$$

En este caso, además del error de control $e_c(t) = \text{ref}(t) - y_p(t)$, se define el **error de seguimiento o identificación** $e_i = y_m(t) - y_p(t)$ que indica cuánto se parece la planta al modelo de referencia [13]. Se establece entonces un **modelo de error y una ley de control** $u(t)$, y desarrollando las expresiones disponibles se genera una **ley de adaptación de los parámetros del controlador**. Un controlador MRAC podrá ser **indirecto**, si para modificar sus parámetros estima los de la planta, o **directo** si no lo hace [13]. Dentro de los controladores indirectos existen además las subcategorías **indirecto algebraico e indirecto dinámico** [33]. La figura 2.7 expone un diagrama de bloques de un MRAC directo [13].

Para los casos en que se utilice un controlador PID de ganancias ajustables en una estructura MRAC se debe establecer un criterio para modificar las ganancias del controlador, como por ejemplo el uso de la regla MIT, la cual considera una función de costo a optimizar $J(\theta) = \frac{1}{2}e^2(t)$, función que valdrá cero en un caso óptimo y cuya derivada se desea que sea negativa hasta llegar a cero en el óptimo, por lo que se deduce que los parámetros deben cambiar de manera que $\frac{\partial \theta}{\partial t} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta}$, con $\gamma > 0$ la ganancia de adaptación del controlador [34].

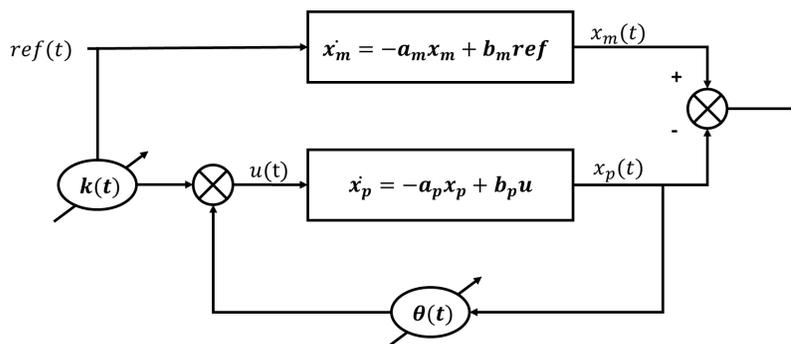


Figura 2.7: Diagrama de bloques de controlador MRAC directo [13].

Otra forma de resolver la adaptación es utilizando lógica difusa (Fuzzy-PID). Un sistema de este tipo propone describir la dinámica de los procesos a partir de nociones conceptuales que denotan un grado de incertidumbre, y no un valor determinista como la lógica convencional [22]. Los modelos difusos clásicos son los de Mamdani y de Takagi-Sugeno, los cuales se componen de una **interfaz de fuzzificación** que convierte una señal determinista en una difusa, una **base de conocimiento o conjunto de reglas difusas**, basadas en conocimiento experto o modelos del proceso, un **motor de inferencia** que permite calcular las variables de salida y finalmente una **interfaz de defuzzificación** que convierte el resultado de la inferencia difusa en una salida determinista [22].

Los **controladores adaptables por redes neuronales artificiales**, utilizan dicha tecnología para controlar la planta o identificarla [14]. Como se detalla en la sección 2.4, existen distintas arquitecturas de redes, y cada será mejor que otra según el problema a resolver, sumado a que, como se observa en las figuras 2.8 y 2.9 se puede usar una red para ajustar los parámetros de un controlador, o como controlador propiamente tal, respectivamente [35][36]. A modo de ejemplo, se expone en la Figura 2.10 una configuración con una red *Feed-Forward* [15].

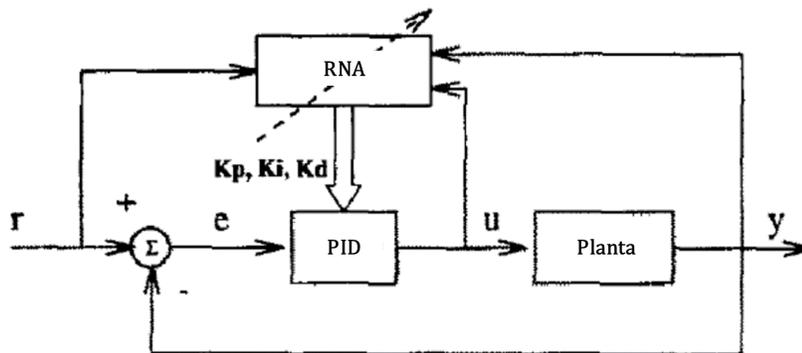


Figura 2.8: Esquema de control adaptable por red neuronal, la cual se utiliza para el ajuste de los parámetros de un controlador PID de ganancias variables [14].

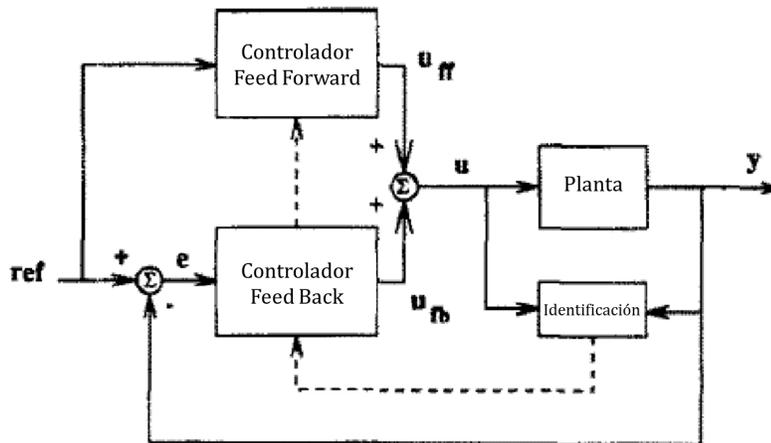


Figura 2.9: Diagrama de bloques de un sistema de control adaptable por red neuronal, utilizada ésta como controlador [14].

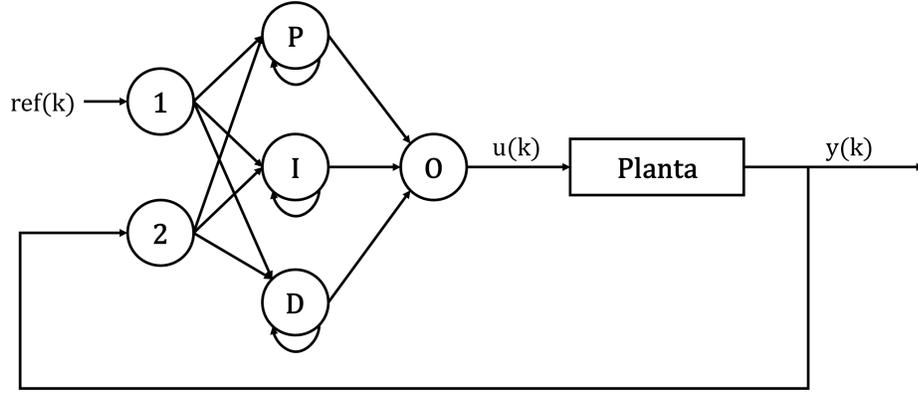


Figura 2.10: Diagrama de bloques de una sistema de control adaptable RRNN-PID con arquitectura *Feed-Forward* [15].

Los **APBC (Adaptive Passivity-Based Controller)** utilizan pasividad para lograr la adaptación. Este concepto se define a partir de un sistema dinámico representado por el modelo de estados $\dot{x} = f(x, u) \in \mathbb{R}$, $y = h(x, u) \in \mathbb{R}$, con f una función localmente Lipschitz, h continua, $f(0, 0) = 0$ y $h(0, 0) = 0$, tal que tiene la misma cantidad de entradas y salidas. Este sistema será entonces pasivo si existe una función continua, diferenciable, semidefinida positiva $V(x)$ (llamada **función de almacenamiento**) tal que [26]:

$$u^T y \geq \dot{V} = \frac{\partial V}{\partial x} f(x, u), \forall (x, u) \quad (2.31)$$

Otra forma de hacer un control adaptable es mediante **modos deslizantes**, un tipo de control de estructura variable que parte de la base que existe una descripción en variables de estado como el de las Ecuaciones 2.32 y 2.33, con $x \in \mathbb{X}^n$ el vector de estados, $y \in \mathbb{R}$ la salida del sistema, $f : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$ un campo vectorial, $h : \mathbb{R}^n \rightarrow \mathbb{R}$ un campo escalar, $u \in \mathbb{R}^u$ la salida del controlador, y S una función llamada **superficie de conmutación** (Ecuación 2.34). Nótese que $\sigma : \mathbb{R}^n \rightarrow \mathbb{R}$ es tal que permite acotar el espacio de estados a las combinaciones adecuadas para cumplir el índice de desempeño [16].

$$\dot{\vec{x}} = f(\vec{x}, y) \quad (2.32)$$

$$y = h(\vec{x}) \quad (2.33)$$

$$S = \{\vec{x} \in \mathbb{X} \subset \mathbb{R}^n : \sigma(\vec{x}) = 0\} \quad (2.34)$$

El controlador entonces enviará una señal de control dependiendo del valor de $\sigma(\vec{x})$, con el fin que el sistema se mantenga en una combinación de estados tal que $\sigma(\vec{x}) = 0$

$$\dot{\vec{x}} = \begin{cases} f(\vec{x}, u^+) & \text{si } \sigma(x) > 0 \\ f(\vec{x}, u^-) & \text{si } \sigma(\vec{x}) < 0 \end{cases} \quad (2.35)$$

Para asegurar esto se genera la condición de existencia de modo deslizante, que modifica la respuesta del controlador si es que el vector de estado se aleja de $\sigma(\vec{x}) = 0$:

$$\dot{\sigma}(\vec{x})\sigma(\vec{x}) < 0 \quad (2.36)$$

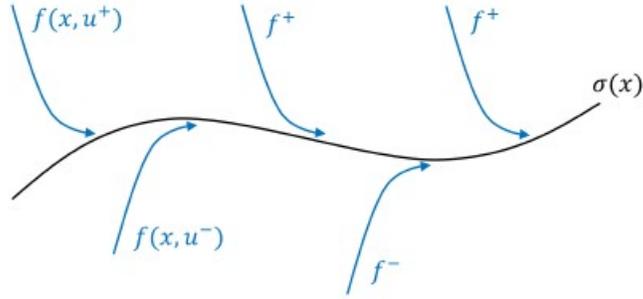


Figura 2.11: Representación gráfica de la técnica por modos deslizantes, con $f(x, u^+)$ y $f(x, u^-)$ la variación de estados según su posición con respecto a la superficie de conmutación $\sigma(x)$ [16].

Dado que la función es discontinua y no satisface las condiciones de existencia y unicidad, los métodos de Filipov y del control equivalente se utilizan para regularizarlo.

Los controladores *backstepping* se basan en la idea de **descomponer el sistema en etapas y diseñar recursivamente controladores desde la última hasta la primera**, de manera que cada vez que se resuelva una, se genere información para las anteriores. El proceso continúa hasta alcanzar la etapa inicial del sistema (de ahí el nombre "*backstepping*"). Este tipo de controladores puede resolver el control de sistemas no lineales complejos y proporciona estabilidad global. Se aplica a sistemas con la siguiente estructura (para este caso, cada ecuación de estado se analiza como un subsistema donde el término acoplado al estado siguiente es tratado como una señal de control virtual) [37][38].

$$\dot{x}_1 = f_1(x_1) + g_1(x_1)x_2 \quad (2.37)$$

$$\dot{x}_2 = f_2(x_1, x_2) + g_2(x_1, x_2)x_3 \quad (2.38)$$

⋮

$$\dot{x}_{n-1} = f_{n-1}(x_1, \dots, x_{n-1}) + g_{n-1}(x_1, \dots, x_{n-1})x_n \quad (2.39)$$

$$\dot{x}_n = f_n(x_1, \dots, x_n) + g_n(x_1, \dots, x_n)u \quad (2.40)$$

En este ámbito también se mencionan los controladores **PID adaptables que utilizan cálculo fraccionario (FOPID)**, pero el mecanismo de adaptación corresponde generalmente a alguno de los casos antes mencionados [39][40][41].

El **BRL-PID** modifica en tiempo real los parámetros del controlador utilizando **aprendizaje reforzado bayesiano, seleccionando parámetros de acuerdo a su probabilidad de éxito**, evaluado éste con respecto al índice de desempeño propuesto. El sistema con el que se compara la propuesta BRL-PID está calibrada por Ziegler-Nichols.

2.3. Procesos Estocásticos

En los anexos se detallan las definiciones y conceptos fundamentales de Probabilidad que sustentan la información de esta sección, en la cual se definen y caracterizan los procesos estocásticos, métodos secuenciales de Monte Carlo, aprendizaje de máquinas y aprendizaje reforzado bayesiano.

Un **proceso aleatorio o estocástico** es una colección de variables aleatorias $\{X(\omega, t), \omega \in \Omega, t \in T\}$, con $T \subset \mathbb{N}$ un rango de tiempo discreto (si $T \subset \mathbb{R}^+$, entonces será continuo), de manera que permiten representar matemáticamente procesos en los que existe incertidumbre [19][42]. Se pueden clasificar de acuerdo con los siguientes criterios [37]:

- **Según las características temporales de las variables aleatorias:** discretos o continuos.
- **Según su regularidad:** un proceso será estacionario si las distribuciones de probabilidad no cambian al hacer una traslación temporal arbitrario.
- **Según la memoria:** un proceso puramente estocástico o sin memoria es aquel cuyo valor en un tiempo determinado es independiente de los anteriores.
- **Gaussianos estocásticos:** son un tipo de procesos que utilizan puntualmente con una ley de distribución gaussianas, lo cual le concede una serie de características convenientes para el análisis.

En este ámbito, los **métodos de Monte Carlo** se basan en la repetición del muestreo aleatorio para estimar una cantidad deseada [43][44].

2.3.1. Métodos Secuenciales de Monte Carlo

Si se desea obtener el valor esperado de una función aleatoria $f(X) : X \rightarrow \mathbb{R}$, donde X representa un mapa, pero no es posible calcular las métricas de éste, si se pueden muestrear versiones X_1, \dots, X_n independientes e idénticamente distribuidas entre sí (*iid*) y con la misma distribución que la del conjunto de todos los mapas, entonces se puede aproximar la esperanza de X , $\hat{\mu}$ con la Ecuación 2.41 [43].

$$\hat{\mu} = E(f(X)) = \frac{1}{n} \cdot \sum_{i=1}^n f(X_i) \quad (2.41)$$

Es decir, el promedio de los valores obtenidos de los mapas. La ley de los grandes números fuerte permite aseverar que con un n suficientemente grande se tiene una representación adecuada de la función [19]. La expresión anterior es un tipo de función aleatoria llamada **estimador**, las cuales reciben datos de un experimento aleatorio y permiten aproximar el valor de una función [45]. Un estimador será **insesgado** si su valor esperado coincide con el del valor a estimar, y será **consistente** si converge en probabilidad (es decir, que cuando la cantidad de datos muestreados tiende a infinito la esperanza es igual al valor real a estimar) [45].

Dentro de los algoritmos de Monte Carlo son de interés los de tipo secuencial (*Sequential Monte Carlo, SMC*), es decir, aquéllos que lidian con el tratamiento de información en el tiempo y utilizan el muestreo para obtener información necesaria. Se definen a continuación de acuerdo a [46].

Considérese un secuencia de distribución de probabilidad $\{\pi_t(x_{0:t})\}_{t \geq 1}$, donde se asume que $\pi_t(x_{0:t})$ puede ser evaluado punto a punto **hasta una constante de normalización**. Los métodos SMC son una clase de algoritmo diseñados para aproximar mediante muestreo $\{\pi_t\}$, es decir, generar una colección de $N \geq 1, N \in \mathbb{N}$ muestras ponderadas de variables aleatorias generando un conjunto de pares $\{\omega_t^i, x_{0:t}^i\}_{i=1, \dots, N}, \omega_t^i \geq 0, \forall t \geq 1$, que satisfacen 2.42, con φ_t **cualquier función π_t integrable**.

$$\sum_{i=1}^N \omega_t^i \varphi_t(x_{0:t}^i) \xrightarrow{N \rightarrow \infty} \int \varphi_t(x_{0:t}) \pi_t(x_{0:t}) dx_{0:t} \quad (2.42)$$

Los métodos SMC se pueden ocupar para aproximar distribuciones de probabilidad, como la distribución *a posteriori* de un parámetro. Algunos ejemplos de métodos SMC que utilizan inferencia bayesiana son [47][48]:

- **Markov Chain Monte Carlo (MCMC)**: una cadena de Markov es un proceso aleatorio con la característica que una salida en un tiempo determinado sólo depende de la muestra inmediatamente anterior. Los métodos MCMC son un tipo de SMC que utiliza las cadenas de Markov para obtener la distribución *a posteriori*. Ejemplos de MCMC son Metropolis-Hastings, Gibbs Sampling, y Hamiltonian Monte Carlo (HMC) [49][50].
- **Filtros de partículas**: utilizan el concepto de partícula, un ente que retiene información de la dinámica del sistema, las cuales se obtienen mediante muestreo, que corresponde normalmente a observaciones ruidosas. Dependiendo del filtro se procesan las partículas y se obtiene la distribución deseada [51].
- **Sequential Importance Sampling (SIS)**: estos algoritmos se implementan en dos etapas, la primera es tomar una muestra de un conjunto de partículas a partir de una distribución propuesta, y luego las partículas son ajustadas mediante valores denominados **pesos**, basándose en la verosimilitud del estado actual, dadas las observaciones. Estos pasos se repiten hasta llegar a la distribución objetivo [52].

2.4. Aprendizaje de Máquinas

La Inteligencia Artificial (IA) es la ciencia y la ingeniería de hacer máquinas inteligentes, entendiendo la inteligencia en este contexto como la capacidad de razonar como un ser humano o de forma puramente racional, dependiendo del enfoque [17][53]. Una rama de la IA es el *Machine Learning* (ML), campo de la informática que tiene como objetivo programar computadoras para que puedan aprender y actuar sin ser programadas explícitamente [17]. La Figura 2.12 muestra la estructura básica de un sistema de aprendizaje de este tipo, la cual se compone de las siguientes partes:

1. **Metas de aprendizaje:** definidas en base a la tarea a realizar.
2. **Modelo:** función matemática que mapea las percepciones y acciones.
3. **Reglas de aprendizaje:** ayudan al algoritmo a optimizar su respuesta y actualizar los parámetros del modelo.
4. **Experiencia:** conjunto de percepciones y acciones correspondientes.

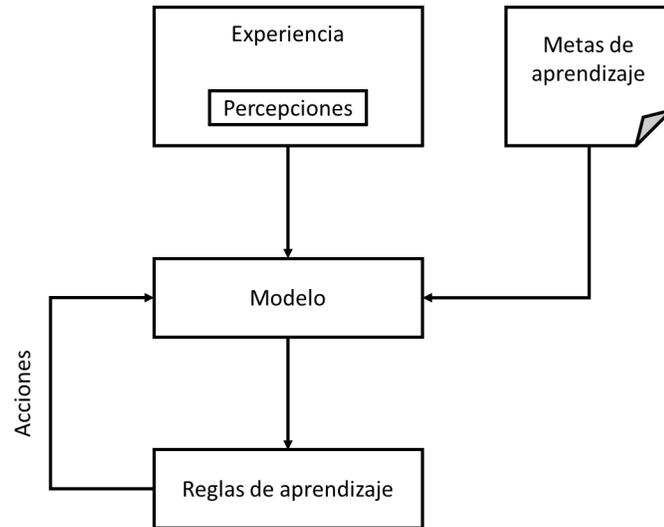


Figura 2.12: Estructura básica de un algoritmo de ML [17].

Para clasificar los algoritmos de ML se puede optar por diversos criterios [17][54]:

1. Clasificación en base a las metas, tareas y función objetivo:

- (a) Predicción de la salida en base a pares entrada/salida previos.
- (b) Regresión: se estima una función a partir de una colección de datos.
- (c) Clasificación (categorización): se definen categorías o clases, y según las características de los datos se asignan a cada una de éstas.
- (d) Clustering: el sistema agrupa conjuntos de datos en grupos homogéneos.
- (e) Planificación: se genera una secuencia óptima de acciones para resolver un problema.

2. Clasificación basada en los modelos:

- (a) Árboles de decisión: consiste en un grafo cuyos nodos son clases y las ramas son elementos de decisión, que pueden representar el cumplimiento de ciertas características del dato entrante [55].
- (b) Separadores lineales (uso del perceptrón): mediante un perceptrón se define un umbral de clasificación lineal de conjuntos [54].
- (c) Redes neuronales (RRNN): son sistemas de aprendizajes cuya programación se inspira en el funcionamiento de las neuronas humanas. Existen múltiples configuraciones (llamadas *arquitecturas*) de red, destacando las siguientes [54][56]:
 - Perceptrón: es el modelo más antiguo de RRNN. Toma entradas, genera una suma ponderada de ésta y luego una función de activación entrega una salida.

- Feed Forward (FF): en esta arquitectura todos los nodos están conectados, hay una capa oculta y no existe retroalimentación entre la entrada y la salida. En la mayoría de estos casos se usa el algoritmo de retropropagación o *backpropagation* en inglés.
 - Radial Basis Function Network (RBFN): es una Feed Forward pero cuya función de activación es la función de base radial en vez de la logística.
 - Recurrent Neural Network (RNN): utiliza en sus capas ocultas neuronas recurrentes, es decir, que reciben su propia salida con retraso.
 - Long/Short Term Memory Cell (LSTM): se incorporan elementos de memoria.
 - Autoencoder (AE): su construcción fuerza que generalice la información y detecte patrones similares, siendo especialmente útil para clasificación, *clustering* y extracción de características.
 - Markov Chain (MC): sirven para hacer clasificación basada en probabilidades, como filtros bayesianos, *clustering* y máquinas de estado finitas.
- (d) Algoritmos genéticos: se basan en los modelos evolutivos de las especies, en los que existe una inicialización, evaluación de los individuos, y luego una selección, recombinación, mutación o reemplazo, modificando así la población hasta llegar a una solución.
- (e) Modelos gráficos.
- (f) Support Vector Machines (SVM): es un método de aprendizaje supervisado que busca separar mediante un hiperplano conjuntos de datos, procurando que la distancia de éstos con respecto a dicho hiperplano sea la mayor posible.
- (g) Modelos ocultos de Markov (HMM): modelos estadísticos en los que se asume que el sistema a modelar es un proceso de Markov de parámetros desconocidos (ocultos), siendo el objetivo determinarlos a partir de los parámetros observables.

3. Clasificación basada en las reglas de aprendizaje:

- (a) Gradiente descendente: algoritmo que estima numéricamente dónde una función genera sus valores más bajos según la dinámica de su gradiente [57].
- (b) Menor error cuadrático medio (LSE): utiliza el error cuadrático como función a optimizar [58].
- (c) Maximización de expectativa: busca la máxima verosimilitud (local) o estimaciones máximas *a posteriori* (MAP) de parámetros en modelos estadísticos, en casos en que el modelo depende de variables latentes no observadas [59].
- (d) Maximización de márgenes: el margen es la misma función que se optimiza en SVM, y es la distancia entre un hiperplano y los datos más cercanos a éste [60].

4. Clasificación basada en la experiencia :

- (a) Aprendizaje supervisado: la máquina recibe una colección de pares entrada/salida para establecer un nuevo par una vez recibida una nueva entrada [61] [62].
- (b) Aprendizaje no supervisado: el objetivo de la máquina es generar el modelo que genera los pares [61] [62].
- (c) Aprendizaje activo: la máquina puede elegir los datos de los cuales aprende, y consultar a un oráculo (por ejemplo, una persona) sobre el etiquetado de datos [63].

- (d) Aprendizaje reforzado: las acciones de la máquina son premiadas o castigadas, siendo el objetivo maximizar las recompensas en el largo plazo [64].

De la literatura consultada se deduce que los problemas habituales para implementar estas soluciones están relacionados con la cantidad y calidad de los datos y su etiquetado, la dimensionalidad del problema, la rapidez del proceso (capacidad de cómputo), si es un aprendizaje en tiempo real o no y el nivel de precisión requerido, entre otros.

2.4.1. Aprendizaje Reforzado Bayesiano

Un modelo de aprendizaje reforzado (RL por su nombre en inglés *Reinforcement Learning*), se compone de un **agente** que interactúa con el **entorno** y realiza acciones dependiendo de la recompensa o castigo de las anteriores, lo cual implica modificaciones en los estados del sistema. Para esto se requiere de una **política de aprendizaje** (la forma en que el agente se comportará), una **función de recompensa** (cómo se premia o castiga al agente), una **función de valor** (estima qué tan buena puede ser la política elegida en el largo plazo, asignando valor a los pares estado-acción del ambiente) y opcionalmente, un **modelo del entorno**. A la secuencia ordenada de estados y acciones de control en el tiempo $\tau = \{(x_0, u_0), (x_1, u_1), \dots\}$ se le denomina **trayectoria** [64][65].

Los algoritmos de aprendizaje pueden ser basados en modelos, o directamente planificar una acción. Dependiendo del caso las soluciones podrán considerar la trayectoria hasta un tiempo t y evaluarla, estimar un estado terminal, determinar cuál de los elementos de decisión son más importantes para la toma de decisiones, entre otras opciones [65][64].

En el caso del RL bayesiano se generan distribuciones de probabilidad de los valores de interés, lo cual permite:

- Representar el grado de conocimiento del entorno en las distribuciones *a priori*, acelerando así el aprendizaje.
- Optimizar naturalmente el compromiso entre exploración y explotación.
- Considerar representaciones del riesgo en la elaboración de la política de aprendizaje.

Ejemplos de algoritmos de aprendizaje reforzado bayesiano son:

- **Bayesian Q-learning:** este algoritmo agrega métodos para considerar la incertidumbre en un algoritmo Q-learning tradicional, permitiendo al agente considerarla en sus decisiones. La función Q del algoritmo devuelve la recompensa y representa qué tan buena fue la acción tomada en un estado determinado. En la ecuación 2.43 se observa que el par acción recompensa en $t + 1$ está definida por el valor previo, un parámetro α que indica cuán importante es la modificación en el estado futuro, la recompensa r_t , una tasa de descuento γ que representa la probabilidad de éxito futuro, y la estimación de la mejor posible salida futura según la acción tomada. En un caso con incertidumbre estos valores pueden considerarse variables aleatorias dependiendo de las características del problema [66].

$$Q_{t+1}(s_t, a_t) = (1 - \alpha)Q_t + \alpha(r_t + \gamma \cdot \max_a(Q(s_{t+1}, a))) \quad (2.43)$$

- **Particle Filter-Based Reinforcement Learning:** análogo a la explicación anterior, se incorpora la incertidumbre a los algoritmos de filtros de partículas mediante variables aleatorias, permitiendo así que ésta sea incorporada al mecanismo de toma de decisiones [67].
- **POMDP-Based RL:** combina RL con Procesos de Decisión Markovianos Parcialmente Observables (POMDP en inglés), los cuales son utilizados para modelar casos donde el agente no es totalmente observable. Se utilizan métodos bayesianos para estimar la distribución de probabilidad de los estados ocultos versus las observaciones [68].

La propuesta BRL-PID considera un filtro de partículas para la selección de parámetros del controlador en base a sus distribuciones de probabilidad y el desempeño de éstos. Las distribuciones se actualizan mediante un esquema bayesiano de aprendizaje.

2.5. Planta de Prueba: Estanque Cónico

Los estanques cónicos son utilizados en gran parte de los sectores productivos: procesamiento de alimentos, la industria química, producción de cerveza, industria farmacéutica y tratamiento de desechos, entre otros [5]. La masividad de su utilización así como sus características no lineales lo hacen un sujeto de prueba interesante en el ámbito de control de sistemas [4][5]. Es por esto que para evaluar el controlador BRL-PID se escogió al estanque cónico ubicado en el Laboratorio de Control Automático del Departamento de Ingeniería Civil Eléctrica de la Universidad de Chile como planta de prueba. Basándose en el modelo de [18] y consultas realizadas a los encargados de laboratorio se recopiló la información necesaria para simular una situación de control.

El sistema toma el agua almacenada en el estanque de recirculación y la lleva mediante cañerías y una bomba hasta el estanque cónico. La Figura 2.13 expone un diagrama de lo anterior, y la Ecuación 2.44 representa la dinámica del estanque cónico en su versión discreta, con T_s el período de muestreo, f la salida del actuador y ω_{proc} , ω_{sens} los ruidos de proceso y observación, aditivos y gaussianos de media cero, y p_i , $i = \{1, \dots, 6\}$ los parámetros de la planta.

$$h(k+1) = h(k) + T_s \cdot \frac{p_1 \cdot f - p_2 - p_3 \cdot \sqrt{h(k)}}{p_4 \cdot h^2(k) + p_5 \cdot h(k) + p_6} + \omega_{proc} + \omega_{sens} \quad (2.44)$$

De acuerdo con [18], los valores de los parámetros de la planta son $p_1 = 5,43$, $p_2 = 78,23$, $p_3 = 20,21$, $p_4 = 0,63$, $p_5 = 11,4$ y $p_6 = 17,1$. En las simulaciones se aplica un modelo de degradación arbitrario tomando como punto de partida estos valores.

El nivel de agua del estanque varía típicamente entre $0[cm]$ y $60[cm]$, si bien en la práctica el límite inferior es del orden de $3[cm]$ de acuerdo a lo indicado por los encargados de laboratorio. El ADC del sistema que recibe la información de sensado funciona con un período de muestreo de $0,05[s]$. El actuador es un saturador que entrega valores entre $0[\%]$ y $100[\%]$, los que controlan el flujo de líquido mediante un variador de frecuencia.

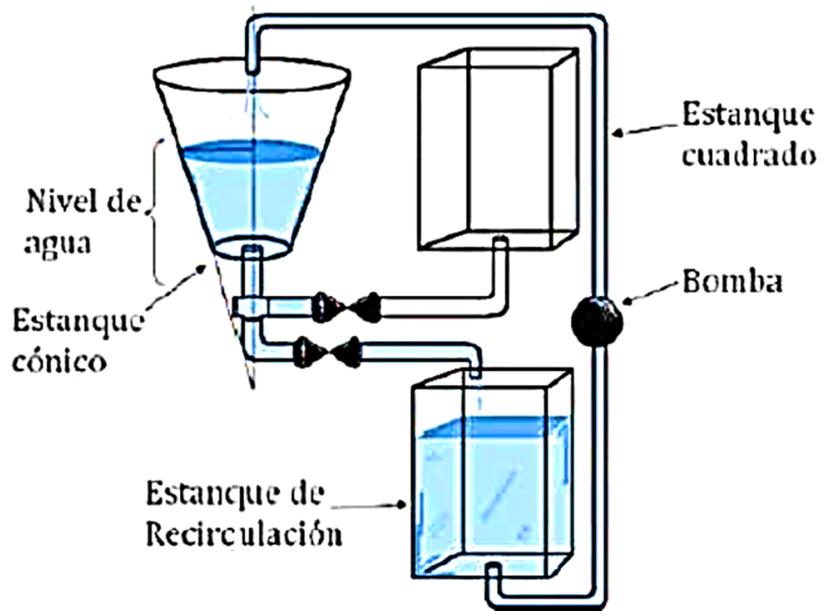


Figura 2.13: Diagrama del estanque cónico ubicado en el Laboratorio de Automática del Departamento de Ingeniería Civil Eléctrica de la Universidad de Chile [18].

Capítulo 3

Estado del Arte

La revisión bibliográfica contempló la búsqueda de soluciones de tiempo real mediante controladores adaptables, esquemas de aprendizaje bayesianos, filtros de partículas, soluciones específicas para estanques cónicos y soluciones *model-free* y *black box* entre 2020 y 2023. Se presenta a continuación una selección de los resultados.

3.1. *Deep Adaptive Control With Online Identification For Industrial Robots* [1]

Esta investigación aborda el problema de la automatización de robots industriales en la actualidad, los cuales requieren mayor precisión y rapidez de operación. Según [1], para ciertas tareas específicas se requiere entrenar y preparar intensamente una máquina para obtener un buen resultado, y que utilizar controladores PID en cada unión mecánica de los robots suele ser la solución más utilizada, sin embargo esto obliga a mantener una rapidez de trabajo baja. En una situación práctica, robots manipuladores (por ejemplo de ensamblaje, soldadura o corte) requieren de control sobre la posición, velocidad y fuerza o torque, lo cual necesita mayor rapidez en el procesamiento y el establecimiento de una trayectoria óptima considerando la física del problema.

Para identificar los parámetros del robot en tiempo real se propone una solución recursiva basada en la optimización de mínimos cuadrados que combina aproximaciones basadas en datos y en modelos llamada algoritmo OSBL (*Online Sparse Bayesian Learning*), la cual se combina con una red neuronal profunda para la identificación en tiempo real de sistemas no lineales, lo que se denomina *Enhanced OSBL*. Los aportes del *paper* en este contexto son los siguientes:

1. La red se entrena con el error residual de modelación del OSBL, aportando robustez al ruido y dinámicas no modeladas.
2. Se propone una configuración de control adaptable llamado *Deep Adaptive Control*, el cual garantiza estabilidad según Lyapunov y es robusto ante dinámicas no modeladas.

3. Se propone una estrategia de control e identificación para robots industriales con *Enhanced OSBL*.
4. La propuesta se validó en un robot industrial con 6 grados de libertad.

El algoritmo OSBL está diseñado para identificar modelos no lineales de la forma:

$$y = \Phi_{Dict} \cdot \vec{\omega} + \vec{\xi} \quad (3.1)$$

Con Φ_{Dict} una matriz que contiene funciones candidatas, $\vec{\omega}$ el vector lineal de pesos asociado a las candidatas y $\vec{\xi}$ un vector de ruido gaussiano $\mathcal{N}(0, \lambda)$, con lambda el *forgetting factor* de la actualización, la cual utiliza la información de la trayectoria de los últimos N pasos del diccionario, de manera que, por las características del problema, se puede representar la dinámica del robot como una combinación lineal de las candidatas. La incorporación de la red neuronal parte de la base que, además del ruido, existe incertidumbre asociada a las dinámicas no modeladas, por lo que ésta se usa para aprender las características del error y perfeccionar el modelo.

$$y = \Phi_{Dict} \cdot \vec{\omega} + \vec{\xi} + \eta(\vec{X}) \quad (3.2)$$

Para su entrenamiento la red utiliza un modelo de referencia similar al control MRAC, utilizando modelos lineales para resolver. La Figura 3.1 expone el esquema de control de la propuesta.

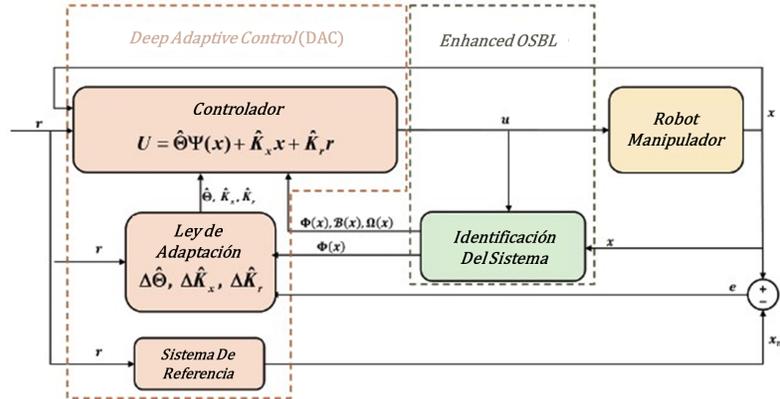


Figura 3.1: Arquitectura del controlador con *Enhanced OSBL* [1].

Esta investigación se consideró relevante para este trabajo por cuanto se trabaja con un PID adaptable en tiempo real y en presencia de incertidumbre, sumado al hecho que se anexa una red neuronal para perfeccionar el modelo, lo cual enriquece el control MRAC en tiempo real. **Este algoritmo inspiró el uso momentáneo de parámetros en la planta (equivalente a las funciones candidatas en este caso), reemplazando en el BRL-PID la red neuronal por la planta misma, todo esto con el objetivo de reducir el cómputo asociado a la red neuronal.**

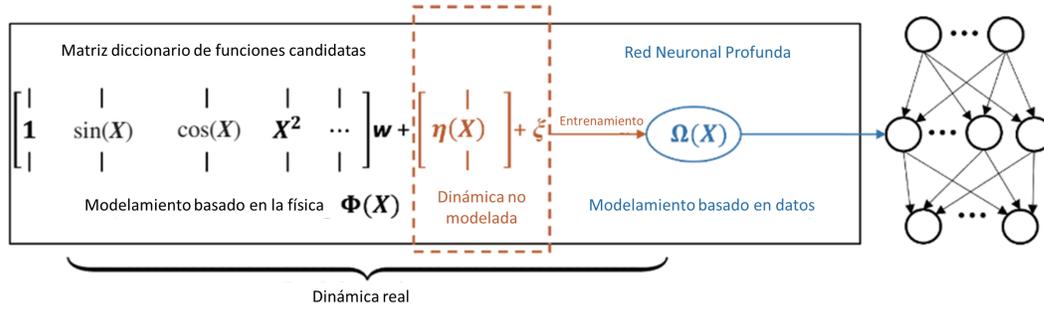


Figura 3.2: Esquema del algoritmo *Enhanced OSBL* [1].

3.2. *SnAKe: Bayesian Optimization Via Pathwise Exploration* [2]

Uno de los problemas del control de una caja negra es que es costosa de evaluar con respecto al tiempo y recursos, ya que, entre otras herramientas, no se tiene acceso a la optimización por gradiente, y que se logra la optimización por consultas sucesivas al sistema. Este *paper* propone un esquema de aprendizaje bayesiano que considera la distancia entre los valores a utilizar de manera de reducir el esfuerzo de control. La idea se inspira en el juego *Snake*, en el que una serpiente se hace más grande a medida que llega a puntos específicos en la pantalla, y no puede chocar consigo misma, de manera que la ruta óptima se consigue haciendo una ruta más larga conforme va creciendo, para reducir la posibilidad de choques, salvo que en este caso la ruta se relaciona a la cantidad de consultas a realizar y en qué orden. Puntualmente, en el caso de los reactores mencionados, estos requieren además cambios pequeños en la entrada en cada iteración y existe un retraso en la obtención de la información, por cuanto las gotas deben abandonar el reactor para conseguir la información.

La solución considera encontrar el máximo $x^* = \arg \max_{x \in \mathcal{X}} f(x)$, con f una caja negra, con \mathcal{X} un conjunto compacto en \mathbb{R}^d , f continuamente diferenciable y costosa de evaluar, siendo dicho costo $\mathcal{C}(x_{x_t, x_{t+1}})$. Se obtiene entonces una secuencia de T muestras de la planta las cuales contienen ruido aditivo gaussiano. Se asume que además hay un tiempo de retraso en la obtención de cada valor t_{delay} entre elegir una consulta y recibir la respuesta de la planta (sin este retardo el problema se convierte en una optimización bayesiana típica). Se usa la función $SR_t = f(x^*) - \max_{i=1, \dots, t} f(x_i)$ como la métrica de rendimiento, y la función a optimizar es la suma de los costos de cada consulta. Se impone un proceso gaussiano como *prior* de la caja negra, por lo que, dado que el ruido aditivo de la observación también es de este tipo, la distribución *a posteriori* también es un proceso gaussiano tal que su varianza μ_t y covarianza κ_t pueden ser calculadas analíticamente.

La propuesta busca crear una cantidad de consultas por lotes que se desean evaluar, y luego se planifica a futuro una ruta de optimización, lo que permite hacer la estimación óptima de costos, lidiar con los retrasos de la planta y mantenerse computacionalmente factible. La forma en que se producen los lotes es mediante diferente técnicas de muestreo, y se establece un presupuesto como límite de la cantidad de consultas realizadas en cada iteración. En el *paper* se utilizan muestreos generados por la técnica

de Thompson, y la ruta se optimiza con el *Salesman Problem*. La optimización se realiza utilizando una técnica de optimización ingenua. Esta solución debe resolver el problema de **estancamiento y áreas sin puntos estacionarios**. La figura 3.3 expone gráficamente una iteración del algoritmo.

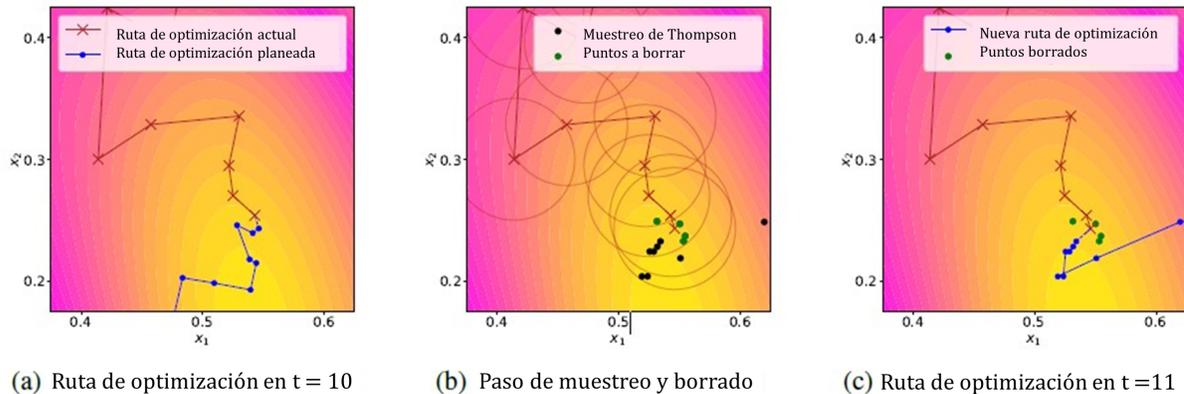


Figura 3.3: Ejemplo gráfico de una iteración del algoritmo SnAKe [2].

De esta investigación se rescata la idea de elegir adecuadamente los parámetros de manera de evitar un gasto elevado, ya que en el contexto de la propuesta BRL-PID, la exploración aumenta el valor de la métrica ISI, lo que en un caso real puede ser más costoso o incluso infactible. Esto motivó a que el algoritmo de selección de parámetros tuviera una lógica de selección basada en umbrales móviles según el desempeño obtenido y la forma de elegir los parámetros, dada una distribución.

3.3. *MRAC-RL: A Framework For On-Line Policy Adaptation Under Parametric Model Uncertainty* [3]

El aprendizaje reforzado (RL por su nombre en inglés, *Reinforcement Learning*), ha sido utilizado con éxito en el desarrollo de políticas de control para sistemas dinámicos, siendo frecuente que el entrenamiento de las políticas sea en entornos de simulación, habiendo **problemas al pasar de la simulación a la realidad**. En este *paper* se propone un algoritmo que combina el **control adaptable MRAC con RL para resolver este problema**.

El control adaptable tiene en general la ventaja de conseguir objetivos de control tomando en cuenta incertidumbre de parámetros en tiempo real, y que **su desventaja radica en la incapacidad de integrar objetivos complejos de optimización** por cuanto los métodos normalmente se basan en la minimización de los errores de seguimiento y regulación. Por otra parte, las políticas de RL abordan una amplia gama de tareas y objetivos, **pero fallan en generalizar apropiadamente en presencia de errores de modelado**, por lo que la idea de combinar ambas soluciones se propone para que se apoyen mutuamente en sus falencias. A diferencia del MRAC clásico, en este *paper* **se utiliza el modelo de referencia para entrenar la política de RL, por lo que ésta nunca interactúa con el modelo real, a diferencia de un RL tradicional**.

Sea entonces $f : X \times U \rightarrow X$, función continua, y un sistema dinámico dado por:

$$\dot{x} = f(x(t), u(t), \phi), \quad x(0) = x_0, \quad u(t) \in U \quad (3.3)$$

Donde ϕ corresponde a los parámetros del sistema que pueden estar sujetos a incertidumbre. Sea también un funcional de costo $c : X \times U \times \mathbb{N} \rightarrow \mathbb{R}$ de manera que el problema de optimización para el control viene dado por:

$$\min_{u(t) \in U, \forall t \in [0, T]} \int_0^T c(x(t), u(t), t) dt \quad (3.4)$$

$$\text{sujeto a } \dot{x} = f(x(t), u(t), \phi), \quad \forall t \in [0, T] \quad (3.5)$$

$$x(0) = x_0 \quad (3.6)$$

Suponiendo que mediante técnicas de RL se genera una política π tal que $u(t) = \pi(x(t))$ produce aproximadamente soluciones óptimas para el problema, si el sistema real a ser controlado es físico, se preferirá entrenar la política mayormente en simulación, **para lo cual se utiliza un modelo de referencia** $\dot{x}_r = f_r(x_r(t), u_r(t), \phi_r)$, donde ϕ_r denota los valores nominales de los parámetros de sistema. El problema a resolver con esta configuración es que, si el modelo de referencia es erróneo, el sistema compromete su rendimiento. La Figura 3.4 se expone la diferencia entre una configuración de RL tradicional y el de la propuesta adaptable MRAC. Nótese que la política nunca interactúa directamente con el modelo real de la planta y los algoritmos utilizados mayormente en esta investigación son del tipo **Deep Reinforcement Learning (DRL)**, los cuales usan redes neuronales profundas para aproximar funciones.

Los sistemas no lineales en esta investigación son de la forma $\dot{x} = A\xi(x) + \lambda B_r u$, con $\xi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ un mapa no lineal conocido de la forma $\xi(x) = [\phi(x_1), x_2, x_3, \dots, x_n]^T$ y $\phi : \mathbb{R} \rightarrow \mathbb{R}$ una función no lineal conocida, y λ un escalar desconocido mayor que cero. A es desconocido y B_r son matrices conocidas en el modelo de referencia deseado: $\dot{x} = A_r \xi + B_r u_r$. El objetivo es determinar u tal que el módulo del error, definido como $e(t) = x(t) - x_r(t)$, tienda a cero. La convergencia se determina teóricamente utilizando teoremas basados en Lyapunov.

Se testó el sistema en la resolución del control del SRIP (*Set-point Randomized Inverted Pendulum*) lineal y no lineal, usando tres algoritmos populares de RL: **PPO (Proximal Policy Optimization)**, **SAC (Soft Actor-Critic)** y **DDPG (Deep Deterministic Policy Gradient)**. Se utilizó una función de costo cuadrático para el entrenamiento. Las políticas fueron entrenadas con una frecuencia de interacción agente-entorno de 10[Hz]. Los ambientes de prueba se generaron con modelos de perturbación de parámetros acotados. Además de esta frecuencia se indica la del MRAC, cuyos valores eran de 10[Hz] o 100[Hz] según el caso. Se concluyó que la solución MRAC-RL mejoró el rendimiento, sin embargo, la condición de excitación permanente para la convergencia de parámetros no puede ser asegurada, quedando propuesta la investigación de políticas que promuevan el cumplimiento de esta condición.

De esta investigación se destaca el hecho que mezcla explícitamente control adaptable con RL, y el hecho de utilizar un intermediario (la política π) entre

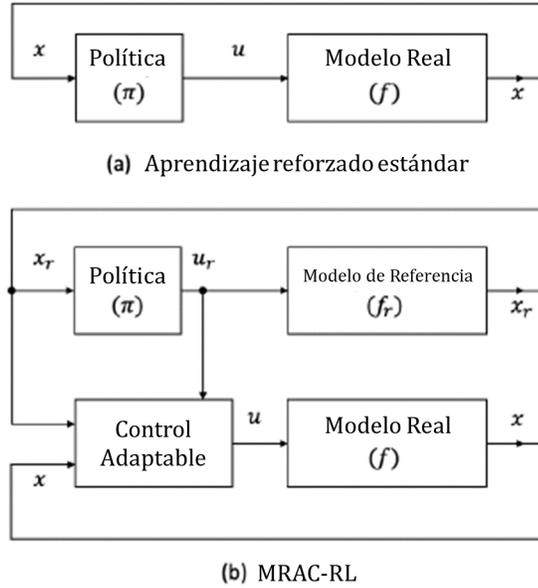


Figura 3.4: Diagramas de bloques de RL tradicional (a) y la propuesta del *paper* (b) [3].

el modelo de referencia y el controlador adaptable para mejorar su rendimiento en un caso real, lo cual motivó la creación del filtro descrito en la metodología, simplificando su funcionamiento en la forma del filtro de la referencia que se ocupó en el diseño final.

3.4. *Soft Computing-Based Fuzzy Integral Sliding Mode Control: A Real-Time Investigation on a Conical Tank Process* [4]

En el entorno industrial hay actividades en las que los estanques cónicos son preferidos por sobre los verticales, sin embargo **el control de nivel para este tipo de estanques en tiempo real no es trivial**, por lo que en este trabajo se propone una variante del control adaptable por modo deslizante llamado *Fuzzy Integral Sliding Mode Controller* (FISMC).

Según se explica en este *paper*, el problema del control por modo deslizante tradicional es que la robustez con respecto a las variaciones de parámetros y perturbaciones externas puede lograrse sólo en la fase deslizante y no durante la llegada a la misma, sumado a la existencia de *chattering* (oscilaciones no deseadas del sistema). El *Integral Sliding Mode Controller* (ISMC) ayuda a mitigar estos problemas, y el sistema *fuzzy* que se añade es para aproximar la variable de deslizamiento. La ley de control propuesta busca también reducir el *chattering* de la señal de control. Se utilizan algoritmos genéticos (GA) para afinar las funciones de pertenencia y los parámetros de la ley de control. En implementación experimental en tiempo real se validó que la propuesta GAFISMC (*Genetic Algorithm Fuzzy Integral Sliding Mode Control*) funciona mejor que el ISMC en términos de variaciones de nivel, de parámetros y reyección de las perturbaciones.

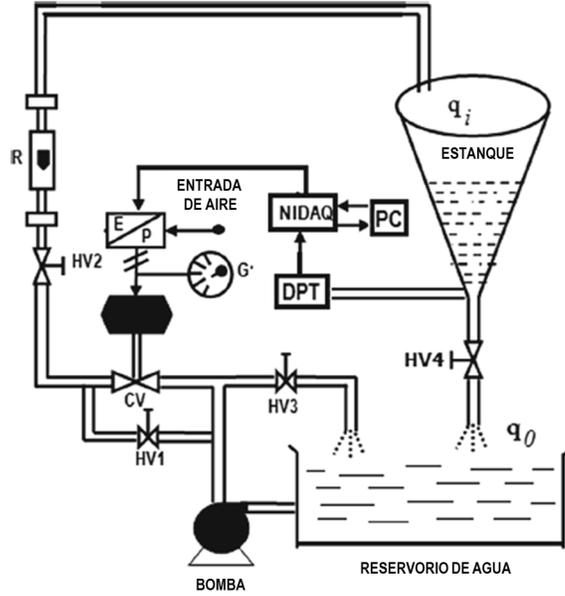


Figura 3.5: Diagramas de *piping* e instrumentación del estanque cónico de [4].

El estanque cónico presenta más no linealidades en la parte inferior versus la superior debido a la menor área de contacto del líquido y alta presión cabeza en el fondo del estanque. A medida que el nivel del tanque aumenta, el punto de equilibrio se mueve a la parte superior, lo que hace que el proceso se estabilice más lentamente y que disminuya la sensibilidad a la señal de control. Por lo tanto, el proceso del estanque cónico se convierte en proceso no lineal adecuado y considerado como ilustrativo para validar el algoritmo de control propuesto.

En este caso se hizo un modelo de primer orden de la planta a partir de una medición de escalón cuyo nivel final es el objetivo de control, y el controlador es un PI calibrado por ensayo y error. La función de deslizamiento es la siguiente, con h_{obj} la altura objetivo y λ un parámetro modificable que permite modificar la respuesta transiente del sistema:

$$S = \lambda \int h - h_{obj} dt + h - h_{obj} = \lambda \int e(t)dt + e(t) \quad (3.7)$$

La ley de control se define como (con u_{eq} voltaje de *bias* inicial equivalente a la tasa de flujo inicial aplicado a la planta):

$$u = u_{eq} + K_u sgn(S) \cdot |S|^\alpha, \quad K_u > 0, \quad \alpha > 0, \quad (3.8)$$

$$sgn(S) = \begin{cases} 1 & S > 0 \\ 0 & S = 0 \\ -1 & S < 0 \end{cases} \quad (3.9)$$

En cuanto a la parte difusa, ésta se utiliza para describir la dinámica del sistema a ser controlado. Se utilizó una matriz asociativa difusa tomando como entradas S y \dot{S} , y como salida u . Se consideraron 49 reglas para asegurar su estabilidad, y las entradas son divididas

en 7 variables lingüísticas: *Positive Big* (PB), *Positive Medium* (PM), *Positive Small* (PS), *Zero* (Z), *Negative Small* (NS), *Negative Medium* (NM) y *Negative Big* (NB).

En FISMC, una elección *a priori* de los valores de pertenencia para los valores de las variables de entrada y salida es una tarea complicada, para lo que en esta propuesta se utilizan algoritmos genéticos (GA por su nombre en inglés *Genetic Algorithm*), tomando como función de *fitness* la integral del error absoluto (IAE) $J = \int_{-\infty}^{\infty} |e(t)|dt$. La Figura 3.6 muestra los resultados de la optimización.

El control propuesto resultó ser exitoso en experimentación de laboratorio para el caso de estudio, detectándose un compromiso entre rendimiento y *chattering*.

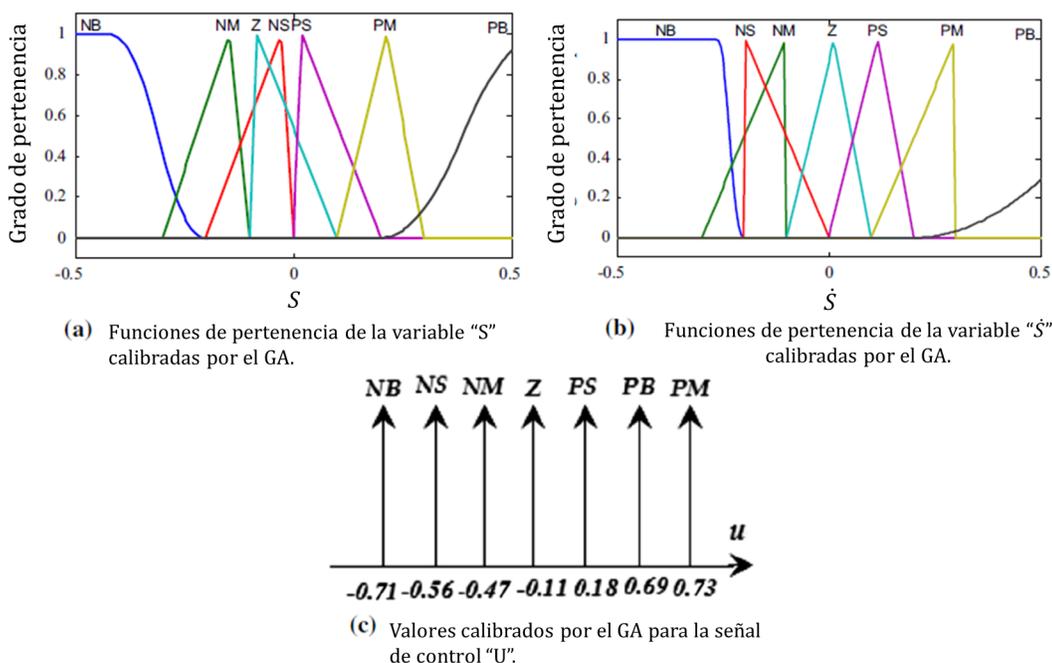


Figura 3.6: Resultados de los conjuntos difusos para describir la dinámica de la planta considerando como entradas S y \dot{S} , y utilizando algoritmos genéticos para optimizar las pertenencias [4].

Este *paper* se consideró por cuanto aborda el problema del estanque cónico y aporta información relevante sobre su conducta no lineal. Además, la solución por tramos ajustado por inteligencia aportó ideas al sistema de memoria, ya que el diseño final se relaciona directamente con esto y puede ser una potencial mejora añadir un sistema difuso en vez de un sistema de memorias si se relaja la condición de cómputo o se define una rutina de trabajo paralela para generar un sistema difuso a medida que surja más información de la planta.

3.5. Study of Different Control Strategies Applied to a Second-Order Nonlinear Tank Process [5]

En esta publicación se compara un PI de ganancias fijas con un GSPI (*Gain Scheduled PI*) y su versión difusa, FGSPI (*Fuzzy Gain Scheduled PI*), diseñado en base a Takagi Sugeno en simulaciones de salto de escalón para un estanque cónico. Para la obtención de las reglas utilizó conocimiento experto y se menciona que para el FGSPI un sistema de inferencia difuso Sugeno es eficiente para el *gain scheduling*, por cuanto opera como un interpolador de múltiples controladores lineales locales combinados para representar todo el rango dinámico del sistema no lineal. La comparación se realizó utilizando las métricas de *settling time*, porcentaje de máximo *overshoot* y las métricas globales ISE e IAE.

La generación de un sistema de lógica difusa incluye asignar funciones de pertenencia y establecer reglas basadas en el conocimiento previo del proceso. Para desarrollar un controlador FGSPI se deben generar cinco modelos lineales de flujo a lo largo del rango de valores del estanque. Las ganancias del controlador fueron calibradas mediante una librería de Matlab. En el diseño Takagi-Sugeno, la variable de entrada es el nivel, y las de salida son las ganancias del PI. Las variables lingüísticas asignadas para los niveles son muy bajo, bajo, medio, alto y muy alto.

Un punto interesante de esta investigación es que se plantea la necesidad de transiciones suaves de una regla a otra de manera que el PID funcione adecuadamente, lo que se tradujo en el BRL-PID en la utilización del filtro de frecuencia, en particular la combinación de funciones coseno y exponenciales para generar una versión de la referencia amortiguada ante saltos de escalón, **haciendo menos agresivo el proceso de exploración de la nueva referencia.**

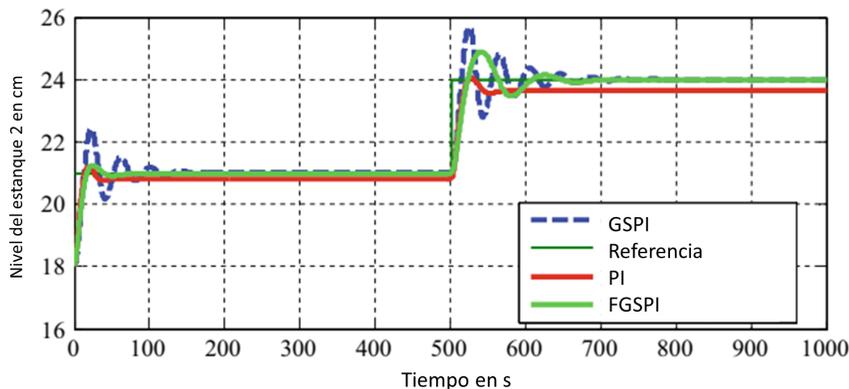


Figura 3.7: Comparación entre el seguimiento de una referencia de un PI de ganancia fija, el sistema GSPI, y su versión difusa (FGSPI) en el tiempo [5].

3.6. *Level Control Of A Conical Tank Using The Fractional Order Control [6]*

Este trabajo propone un controlador de modelo interno de orden fraccionario (FOIMC por su nombre en inglés *Fractional Order Internal Model Control*) basado en técnicas de inteligencia de enjambre para controlar la altura de un estanque cónico. Se compara una solución FOIMC de dos grados de libertad versus una solución con PSO y otra con WOA (*Particle Swarm Optimization* y *Whale Optimization Algorithm*, respectivamente). La motivación para la realización de esta propuesta es la utilización de la inteligencia de enjambre para reducir las imprecisiones de la calibración por ensayo y error.

La propuesta utiliza un filtro fraccionario en cascada con un PID del mismo tipo. **El FOIMC combina las ventajas del IMC como la cantidad reducida de parámetros a afinar y un control estable con las ventajas del control fraccionario como robustez, flexibilidad en la afinación de parámetros y un margen de estabilidad amplio.** La linealización del sistema no lineal utiliza el resto de Lagrange para compensar las derivada de orden superior. Se utiliza PSO (*Particle Swarm Optimization*) y WOA (*Whale Optimization Algorithm*, basado el comportamiento de caza de las ballenas jorobadas) para optimizar los parámetros del FOIMC.

Para ejecutar esta propuesta, se linealiza el modelo de planta de manera de obtener un modelo de primer orden con tiempo muerto, o FOPDT por su nombre en inglés *First-Order Plus Dead Time*. El controlador FOIMC es sintonizado con un algoritmo PSO y WOA para el control de altura de un tanque cónico, utilizando las métricas ISE, IAE e ITAE. Finalmente se diseña un FOIMC basado en filtros fraccionarios. **Las propuestas PSO y WOA utilizan tres funciones de costo**, las que consideran el *peak overshoot* contrapesado y cada una de las métricas indicadas anteriormente:

$$J_1 = 2 \cdot M_p + \int e(t)^2 dt \quad (3.10)$$

$$J_2 = 2 \cdot M_p + \int t \cdot |e(t)| dt \quad (3.11)$$

$$J_3 = 2 \cdot M_p + \int |e(t)| dt \quad (3.12)$$

Se hicieron experimentos de rendimiento considerando una referencia de escalón para que la planta alcance los 30[cm] de altura, de un máximo de 70[cm] con una referencia de 20[cm] que luego pasa a 50[cm] para luego volver a 30[cm]. Se consideraron las métricas ISE, ITAE e IAE en ambos casos. La Figura 3.8 muestra los resultados obtenidos para una referencia de escalón para las tres propuestas, considerando las distintas funciones de costo. La investigación concluye que el FOIMC combinado con PSO resultó se la mejor opción para este caso de estudio.

El uso de diferentes funciones de costo aportó la necesidad de poder elegir en el algoritmo si es preferible favorecer el promedio o el ECM al evaluar los parámetros. Además, el uso de PSO para generar los parámetros del modelo interno como

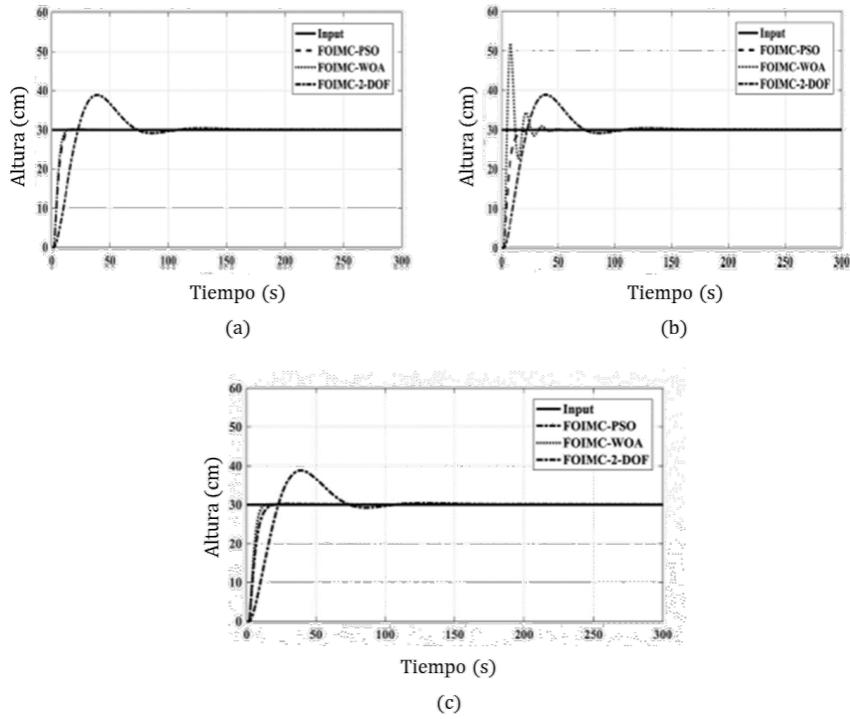


Figura 3.8: Respuesta al escalón del FOIMC con dos grados de libertad, usando PSO y WOA, utilizando J_1 en (a) , J_2 en (b) y J_3 en (c) [6].

una forma de facilitar el proceso, sabiendo que la sintonización de sistemas de orden fraccionario es compleja, **confirmó las decisiones de diseño del algoritmo de decisión de parámetros del BRL-PID**, en el que se implementa un sistema similar a un PSO para la selección de los parámetros al no poder obtener una expresión para optimizar la selección. En la evaluación del BRL-PID no se consideró el *overshoot* ya que no en todas las referencias escogidas tiene sentido medirlo y se buscó una medición universal.

Capítulo 4

Metodología

Habiendo presentado las condiciones del problema en 1.6 y la información investigada, se formaliza el problema y **se describe la propuesta BRL-PID**, detallando los supuestos necesarios para su realización, para luego profundizar en sus elementos y procesos primordiales.

4.1. Descripción del Sistema y Supuestos Simplificatorios

Sea un sistema de bloques discretos con un controlador PID de ganancias adaptables $\vec{\theta}$ acotadas y con límites conocidos, representado en el diagrama de bloques de la Figura 4.1.

$$\vec{\theta} = \{\theta_1, \theta_2, \theta_3\} = \{P(k), I(k), D(k)\}^T \in \mathbb{R}^3 \quad (4.1)$$

$$\theta_i \in [\theta_{i_{min}}, \theta_{i_{max}}], \quad i = \{1, 2, 3\} \quad (4.2)$$

La planta es una caja negra luego se desconoce la dinámica de su salida $\tilde{y}(k)$, sin embargo y dado que se asume que se trabaja con sistemas reales, se asume la existencia de límites relativos a las capacidades físicas de la misma que permiten asumir que sus valores están acotados. Se sabe además que $\tilde{y}(k)$ incluye ruido de proceso $\omega_{proc}(k)$, aditivo y de valor máximo conocido y distribución desconocida.

$$\tilde{y}(f(k)) : F \subset \mathbb{R} \rightarrow \tilde{Y}_{out} \subset \mathbb{R} \quad (4.3)$$

$$\omega_{proc}(k) : k \in \mathbb{N} \rightarrow \Omega_{proc} \subset \mathbb{R} \quad (4.4)$$

La planta es observada mediante un sensor ubicado en el lazo cerrado que aporta ruido de observación $\omega_{sens}(k)$, aditivo, de amplitud máxima conocida y distribución desconocida.

$$\omega_{sens}(k) : k \in \mathbb{N} \rightarrow \Omega_{sens} \subset \mathbb{R} \quad (4.5)$$

Se define $y(k)$ como la señal de la planta contaminada con el ruido del sensor, e $y_{ref}(k)$ la salida del filtro (Figura 4.1).

$$y(k) = \tilde{y}(f(u(k))) + \omega_{proc}(k) + \omega_{sens}(K) \quad (4.6)$$

$$y(k) \in Y_{out} \subset \mathbb{R} \quad (4.7)$$

$$y_{ref}(k) : R_{out} = Y_{ref} \rightarrow Y_{ref} = Y_{out} \quad (4.8)$$

La referencia es una señal real, observable y acotada:

$$ref(k) : \mathbb{N} \rightarrow R_{out} = Y_{ref} \subset \mathbb{R} \quad (4.9)$$

Los errores de control y de identificación se definen respectivamente como:

$$e_c = ref - y(k) \quad (4.10)$$

$$e_i = y_{ref} - y(k) \quad (4.11)$$

En este caso, los conjuntos Y_{out} , Y_{ref} y R_{out} son el intervalo $[0, 60][cm]$ de valores reales, los que puede tomar el nivel de líquido del estanque cónico. Los errores están acotados entonces en el intervalo real $[-60, 60][cm]$.

La salida del PID adaptable, dado un período de muestreo real no negativo T_s y una señal de error en el tiempo $e(k)$ está dada por la aproximación rectangular de la versión continua de éste:

$$u(k) = P(k) \cdot e(k) + I(k) \cdot \sum_{i=1}^k e(k) \cdot T_s + D(k) \cdot \frac{e(k) - e(k-1)}{T_s} \quad (4.12)$$

La señal del actuador dependerá de la señal de control (salida del PID adaptable):

$$f(u(k)) : U \subset \mathbb{R} \rightarrow F \subset \mathbb{R} = [0, 100][\%] \quad (4.13)$$

Se establecen los siguientes supuestos simplificadorios, los cuales se determinaron procurando no condicionar drásticamente la dinámica de la planta:

1. **Señales cuantizadas y cubriendo al menos rango nominal de la planta:** se asume que los dispositivos entregan valores cuantizados en amplitud y tiempo, luego existen un período de muestreo $T_s [seg] \in \mathbb{R} \setminus \{0\}$ (o una frecuencia de muestreo $f_s = T_s^{-1} [Hz]$) y una tasa de bits $b [bits] \in \mathbb{N} \setminus \{0\}$ y que el rango de éstos abarcan al menos el rango de operación nominal del dispositivo a sensar.
2. **Nociones sobre la dinámica de la planta:** se asume que existe información sobre la dinámica de la planta, en particular sobre el orden de magnitud del tiempo reacción de ésta, por ejemplo, si la planta se estabiliza en el orden de minutos, segundos, horas, etc.
3. **La actualización de distribuciones ocurre entre dos muestras consecutivas:** es decir, el sistema es lo suficientemente rápido para actualizar distribuciones sin modificar la frecuencia de muestreo.

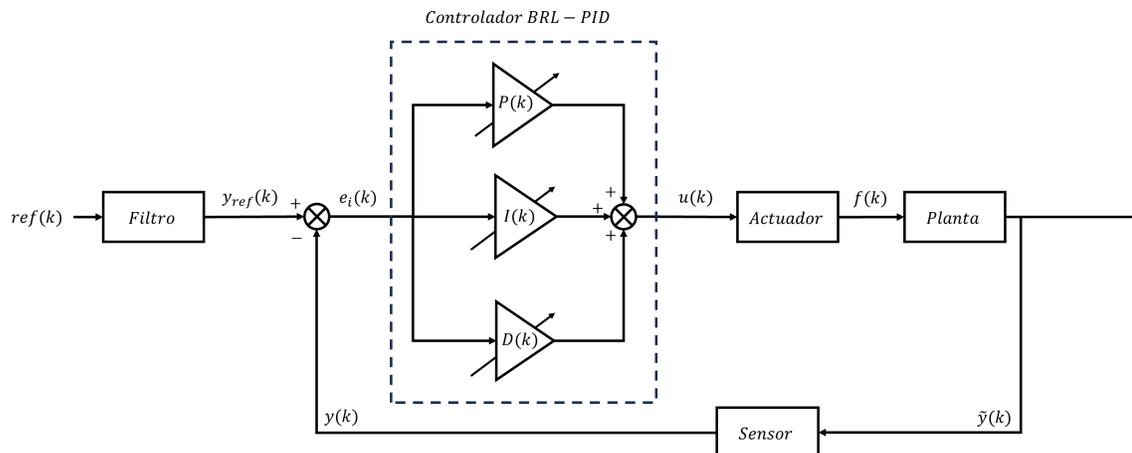


Figura 4.1: Diagrama de bloques de la propuesta BRL-PID. Las ganancias adaptables del PID se modifican utilizando el algoritmo propuesto hasta mientras se cumpla el índice de desempeño.

4. **El error objetivo es gaussiano de media cero y varianza determinada por diseño** $\sigma(e_{obj})$: como se verá más adelante, el error del sistema en la propuesta es tratado como una variable aleatoria, por lo que debe tener una distribución de probabilidad asociada, la cual se asume gaussiana de media cero y varianza determinada por diseño dependiendo de las necesidades del problema.

Habiendo establecido lo anterior, se detallan los elementos necesarios para la solución de control, su funcionamiento y características principales.

4.2. Propuesta de Control

Se presentan los vectores de valores posibles, utilizados para acotar los rangos de las señales del sistema. Posteriormente, se detalla cómo son elegidos e implantados los parámetros del controlador en la operación dependiendo de su probabilidad, luego la evaluación de su desempeño y la actualización de sus distribuciones de probabilidad, el funcionamiento del filtro de la referencia para finalmente detallar las rutinas de memoria y emergencia, destinadas a mejorar el rendimiento y hacer el sistema más robusto ante posibles fallas.

4.2.1. Vector de Valores Posibles

Los sistemas digitales cuantizan la información en frecuencia y amplitud, luego se tendrá una colección de valores posibles que se pueden recibir de un sensor discretizado. Los límites físicos por otra parte pueden generar señales saturadas. Con la intención de reducir la capacidad de cómputo y tener la capacidad de gestionar la cantidad de datos que el sistema analizará, se define una **resolución o paso** de señal, dependiente de la tasa de bits de la cuantización α , como la distancia que hay entre dos valores consecutivos posibles de recibir de una señal, el cual se asume idéntico para cualquier par consecutivo. Se define entonces el vector de **valores posibles de una señal** $\vec{VP}(\cdot)$, tal que para una señal cualquiera θ_i , con resolución α_{θ_i} y valor mínimo $\theta_{i_{min}}$, se genera un vector de valores posibles $\vec{VP}(\theta_i)$ de $(r_{\theta_i} + 1)$

elementos como se representa en la Ecuación 4.14.

$$V\vec{P}(\theta_i) = \{(\theta_{i_{min}}), (\theta_{i_{min}} + \alpha_{\theta_i}), (\theta_{i_{min}} + 2\alpha_{\theta_i}) + \dots + (\theta_{i_{min}} + r_{\theta_i}\alpha_{\theta_i})\}, i \in \mathbb{N} \quad (4.14)$$

Para esta propuesta se asume que $V\vec{P}(y(k)) = V\vec{P}(y_{ref}(k)) = V\vec{P}(ref(k))$, con lo cual los errores de control e identificación pueden escribirse en base a los índices de posición de los valores posibles de las plantas o la referencia y la resolución α_{θ} . Las ecuaciones 4.15 y 4.16 exponen lo anterior:

$$e_c(k) = ref(k) - y_c(k) = (ref_{min} + i \cdot \alpha_y) - (y_{c_{min}} + j \cdot \alpha_y) = (i - j)\alpha_y \quad (4.15)$$

$$e_i(k) = y_{ref}(k) - y_c(f, k) = (y_{ref_{min}} + i \cdot \alpha_y) - (y_{c_{min}} + j \cdot \alpha_y) = (i - j)\alpha_y \quad (4.16)$$

Con $i, j \leq r_y$ con lo cual se puede construir el vector de valores posibles de los errores de control e identificación de acuerdo a la Ecuación 4.17.

$$V\vec{P}(e_c) = V\vec{P}(e_i) = \{-r_y\alpha_y, (-r_y + 1)\alpha_y, \dots, 0, \dots, (r_y - 1)\alpha_y, r_y\alpha_y\} \quad (4.17)$$

Dado que la salida sensada de la planta es una secuencia aleatoria de la cual depende la elección de parámetros futura, los parámetros son también variables aleatorias, cada uno con una distribución de probabilidad asociada, discretas y relativas a los vectores de valores posibles, es decir, si se desea generar la distribución de una señal del sistema, se genera un vector de igual tamaño que el de valores posibles, de manera que los índices sean correlativos, es decir, la probabilidad del valor en la i -ésima posición del vector de valores es el valor en la misma posición del vector de probabilidades. Las ecuaciones 4.18 y 4.19 resumen lo anteriormente expuesto.

$$\begin{bmatrix} \vec{\mathbb{P}}(\theta_{i,j}) \\ V\vec{P}(\theta_i) \end{bmatrix} = \begin{bmatrix} \mathbb{P}(\theta_{i_{min}}) & \mathbb{P}(\theta_{i_{min}} + \alpha_{\theta_i}) & \mathbb{P}(\theta_{i_{min}} + 2\alpha_{\theta_i}) & \dots & \mathbb{P}(\theta_{i_{min}} + r_{\theta_i}\alpha_{\theta_i}) \\ (\theta_{i_{min}}) & (\theta_{i_{min}} + \alpha_{\theta_i}) & (\theta_{i_{min}} + 2\alpha_{\theta_i}) & \dots & (\theta_{i_{min}} + r_{\theta_i}\alpha_{\theta_i}) \end{bmatrix} \quad (4.18)$$

$$\sum_{i=0}^{r_{\theta_i}} \mathbb{P}(\theta_{min} + i \cdot \alpha_{\theta_i}) = 1 \quad (4.19)$$

4.2.2. Selección e Implantación de Parámetros $\vec{\theta}$ del BRL-PID

Si bien es esperable que exista una expresión analítica y/o una lógica de funcionamiento de la planta, ante el desconocimiento de esta información la salida sensada es observada como una secuencia aleatoria por el sistema adaptable, lo que implica que los parámetros del controlador también sean de este tipo de variables. Para la elección e implantación de los parámetros del controlador BRL-PID se inicia el algoritmo con una distribución *a priori*, de la cual se extraen $n_v \in \mathbb{N} \setminus \{0\}$ tríadas de parámetros que serán implantados secuencialmente en el controlador durante un tiempo determinado. Se define entonces un **índice de selección** $\beta \in \mathbb{R} \cap [0, 100][\%]$ tal que $\beta \cdot n_v$ valores de cada parámetro serán elegidos aleatoriamente sobre un **umbral de selección**, definido según la Ecuación 4.20, con ξ el **porcentaje de exploración**. El resto de las tríadas se generan con valores bajo el umbral, salvo que éstos últimos serán elegidos procurando dar más posibilidad de selección a los cercanos al umbral.

$$\Xi = \xi \cdot MAX(\mathbb{P}), \quad \xi \in \mathbb{R} \cap [0, 100][\%] \quad (4.20)$$

Cada tríada será implantada durante un tiempo establecido por diseño en la planta, o lo que es equivalente, un número de **muestras** $n_s \in \mathbb{N} \setminus \{0\}$, lo que se denomina **ventana**. Cuando se han colocado las n_v tríadas de parámetros durante n_s muestras cada uno se conforma un **bloque**, el cual se procesa y se actualiza la distribución de los parámetros dependiendo de su rendimiento en la planta, repitiendo el proceso hasta que se logre un índice de desempeño, como se detalla en la Sección 4.2.3. La Figura 4.2 resume este procedimiento.

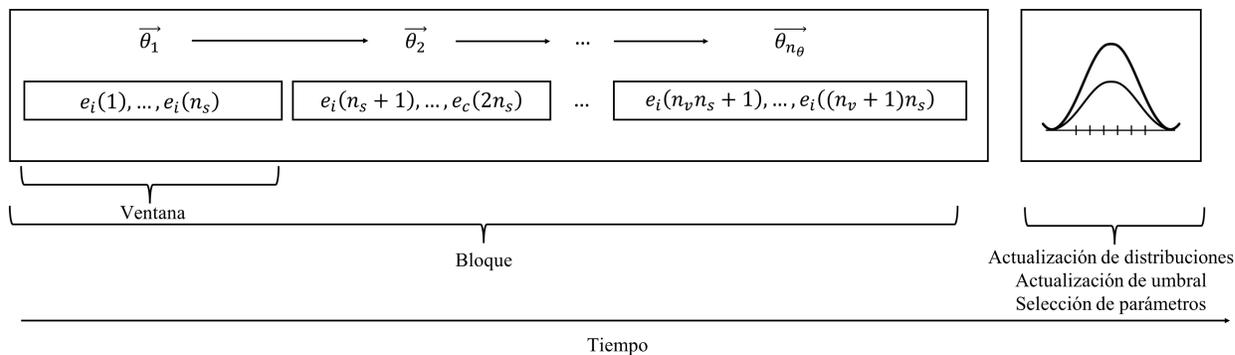


Figura 4.2: Diagrama simplificado del funcionamiento del BRL-PID en el tiempo.

Es importante que el valor del umbral nunca quede bajo o sobre la curva de la distribución, de lo contrario el algoritmo colapsa (la Figura 4.3 expone lo anterior, indicado el caso con la notación *Excepción*), lo que puede ocurrir cuando la distribución se asemeje a una uniforme, en cuyo caso se deben establecer acciones especiales.

Para optimizar la respuesta del controlador, el valor de ξ varía según el desempeño obtenido, para lo cual se define una curva que tenderá a cero si el desempeño es muy bueno o muy malo, bajo la premisa que en ambos casos se prefiere un perfil de exploración más acotado ya que se está cercano a una situación de falla u óptimo, y aumenta conforme se está en un rango de desempeño intermedio.

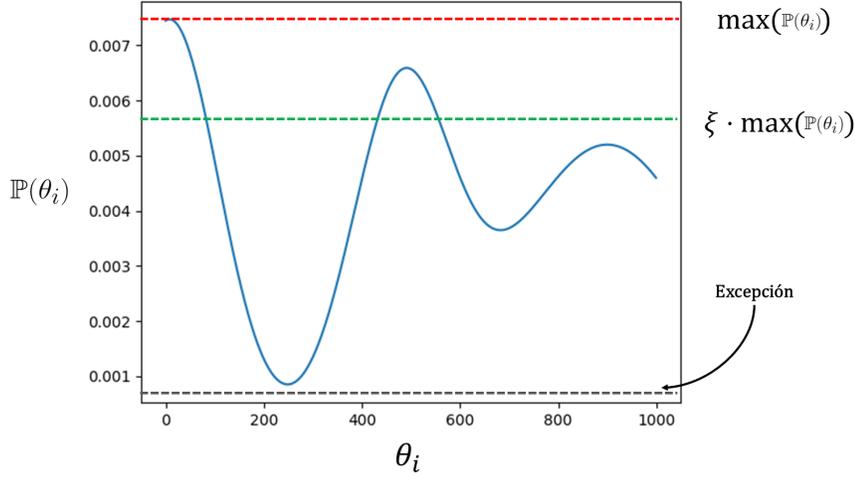


Figura 4.3: Distribución de probabilidad para un parámetro θ_i . La línea punteada roja marca el máximo valor de probabilidad del parámetro, la verde indica el ξ % de la máxima probabilidad, utilizada como umbral para la selección de parámetros y la negra un caso excepcional que puede asumir el umbral durante el proceso de actualización y que amerita una acción especial del mismo.

4.2.3. Evaluación y Actualización de las Distribuciones de Probabilidad de los Parámetros

Los procesos de evaluación de parámetros y actualización de sus distribuciones de probabilidad ocurren entre dos bloques consecutivos, para lo cual se define una **función de evaluación que premia o castiga la probabilidad de los valores implantados en el controlador dependiendo de la dinámica del error** (Ecuación 4.21).

$$\tau(\vec{e}_1, n_{se}) = \frac{1}{\left(\gamma \cdot \frac{|\sum_{i=1}^{n_s} e_{ci}|}{n_s} + (1 - \gamma) \cdot \frac{\sqrt{\sum_{i=1}^{n_s} e_{ci}^2}}{n_s} + 1 \right)} \cdot \left(1 - \frac{n_{se}}{n_{se} + 0,5n_s} \right) \in \mathbb{R} \cap (0, 1] \quad (4.21)$$

Con $\gamma \in \mathbb{R} \cap [0, 1]$, factor que representa la importancia de los primeros dos criterios de penalización (el promedio y el error cuadrático medio del error de identificación con respecto al error cero), $n_{se} \in \mathbb{N}^* \cap [0, n_s]$ el número de muestras de una ventana en la que se debió activar el sistema de emergencia (tercer criterio de penalización, descrito en 4.2.5).

El procedimiento de actualización consiste entonces en multiplicar la probabilidad *a priori* de cada valor posible de cada parámetro por $(1 + \tau)$, generando así la distribución *a posteriori* de cada uno. Por ejemplo, si se tiene el vector de parámetros aplicado en un tiempo k , $\vec{\theta}(k) = \{\theta_1(k), \theta_2(k), \theta_3(k)\} = \{10, 22, 35\}$, y el puntaje asociado a esta combinación es τ , la actualización de las distribuciones es de la forma:

$$\mathbb{P}(\theta_1 = 10) \rightarrow \mathbb{P}(\theta_1 = 10) \cdot (1 + \tau) \quad (4.22)$$

$$\mathbb{P}(\theta_2 = 22) \rightarrow \mathbb{P}(\theta_2 = 22) \cdot (1 + \tau) \quad (4.23)$$

$$\mathbb{P}(\theta_3 = 35) \rightarrow \mathbb{P}(\theta_3 = 35) \cdot (1 + \tau) \quad (4.24)$$

Dado que luego de este proceso la suma de las probabilidades de cada parámetro puede sumar valores distintos a uno (condición 4.19), se debe normalizar la distribución.

Se determinan 4 estados posibles para la operación del sistema dependiendo de la dinámica del error de control, el tiempo y los requisitos de diseño [69]:

1. **Óptimo:** si en promedio durante un bloque o una cantidad determinada de ventanas se mantiene la evaluación sobre un valor estimado por diseño (por ejemplo 70 % o más del óptimo teórico que es 1).
2. **Anomalía:** se asume anomalía si ocurre alguna de las siguientes situaciones:
 - (a) El módulo del error de identificación supera un valor determinado por diseño.
 - (b) La planta llegó a un valor considerado riesgoso (por ejemplo, 95 % del valor máximo del rango de la planta).
3. **Falla:** el sistema se mantiene en valores extremos de su rango de operación durante un tiempo establecido por diseño.
4. **Operación normal:** si ninguna de las situaciones anteriores está sucediendo.

Una vez que la planta esté en estado óptimo se detiene la exploración y se asigna probabilidad uno al máximo valor de cada distribución y cero al resto, de manera que se mantendrá fijo en la planta hasta que se salga de dicho estado, en cuyo caso se genera artificialmente una distribución gaussiana cuya esperanza es el valor máximo y la varianza se impone por diseño según cuánto se desea explorar en esta situación. Nótese que si la salida del óptimo responde a un cambio en la referencia de tipo escalón, se dará prioridad al uso de la memoria en caso de haber información almacenada (se describe esta rutina a continuación). En el caso del estanque cónico la reacción ante anomalía o cercanía al óptimo está considerada en la curva de evolución del índice ξ y el sistema de memoria, mientras que la reacción ante una falla la ejecuta la rutina de emergencia.

4.2.4. Filtro de la Referencia

Ante una referencia de tipo escalón la respuesta transiente del sistema genera un error en la evaluación del algoritmo sobre los parámetros, comprometiendo las métricas de desempeño (se evalúa muy negativamente los parámetros por lo que se genera una exploración innecesaria), ante lo cual se optó por la utilización del sistema de memoria y la aplicación de una función que activa temporalmente un rutina y_{ref}^{salto} cuando se produce un salto en la referencia que supera un valor Δ_{salto} y no hay memoria disponible, y que en el resto de la operación se utilice otra función y_{ref}^{normal} . La solución utilizada se expone con el pseudo código 1 y la Figura 4.4. Nótese que se utilizó $y_{ref}^{normal}(k) = ref(k)$, pero puede ser cualquier otra establecida por diseño. También considerar que la duración de la rutina de salto depende del supuesto del conocimiento de las nociones de la dinámica de la planta y los requisitos de control. La condición $ref(k) - y(k) = 0 \Rightarrow y_{ref}(k) = ref(k)$ se estableció pensando en el caso que se presente una referencia que presente un salto pero no sea exactamente un escalón.

Algoritmo 1 Filtro de la referencia

```
if  $|ref(k-1) - ref(k)| > \Delta_{salto}$  then  
     $i = i_{max}$  ▷ Muestras para la amortiguación  
     $ref_{prev}$  ▷ Valor previo al salto  
     $ref_{post}$  ▷ Valor posterior al salto  
    if  $i \neq 0$  then  
        if  $ref(k) - y(k) = 0$  then ▷ Si la referencia iguala a la planta  
             $y_{ref}(k) = ref(k)$   
             $i = i - 1$   
        else ▷ Aplicación de la función de amortiguación  
             $y_{ref}(k) = ref_{prev} + (ref_{prev} - ref_{post}) \cdot e^{\frac{-i}{i_{max}}} \cdot \cos\left(\pi \cdot 0,5 \cdot \frac{i}{i_{max}}\right)$   
             $i = i - 1$   
        end if  
    else ▷ Para otras situaciones  
         $y_{ref}(k) = ref(k)$   
         $i = i - 1$   
    end if  
else  
     $y_{ref}(k) = ref(k)$   
end if
```

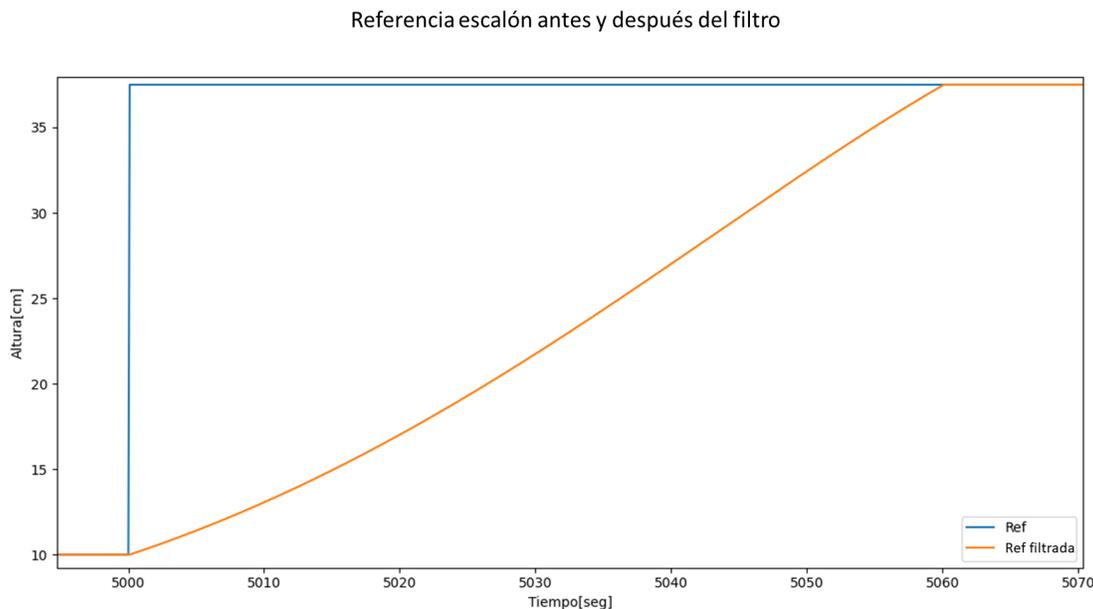


Figura 4.4: Curva para amortiguación de escalones del filtro de la referencia del BRL-PID.

4.2.5. Rutinas de Memoria y Emergencia

Dado que se requiere tiempo para recolectar datos suficientes para la evaluación, sumado a la incertidumbre que presenta el problema, es esperable que haya casos en que convenga realizar acciones especiales en el control, por ejemplo ante la ocurrencia de un salto de escalón en la referencia o situaciones consideradas críticas. Para esto, además del filtro de la referencia se generó un sistema de memoria cuyo objetivo es que, si la referencia genera un escalón en el que se logró el objetivo de control previamente, en vez de actualizar la distribución tomando como aquella *a priori* la del bloque anterior, se haga desde aquella que logró el óptimo. Por otra parte, el procedimiento de emergencia parte del supuesto que la planta puede entrar en una situación no deseada, ante lo cual se ejecuta temporalmente una acción de control predefinida, penalizando además el parámetro aplicado a la planta durante el tiempo que tuvo que modificarse la respuesta del controlador (para lo que se usa el segundo factor de la ecuación 4.21).

Puntualmente, el problema de **detectar un escalón en tiempo real** y el almacenaje en memoria no es trivial, habiendo problemas importantes que resolver, a saber:

1. En un caso real, una señal podría presentar no linealidades de salto y **no ser un escalón puro**, luego se debe considerar una métrica de tolerancia al respecto. La Figura 4.5 ilustra algunos casos posibles y conflictivos.
2. Dado que los escalones pueden tomar tantos valores como los posibles para la planta, se elige un subconjunto de valores de interés, y se establece una métrica de tolerancia para valores cercanos a éstos. A la selección de valores escogidos se denomina **centros de memoria**, cuyos valores serán elegidos por diseño.

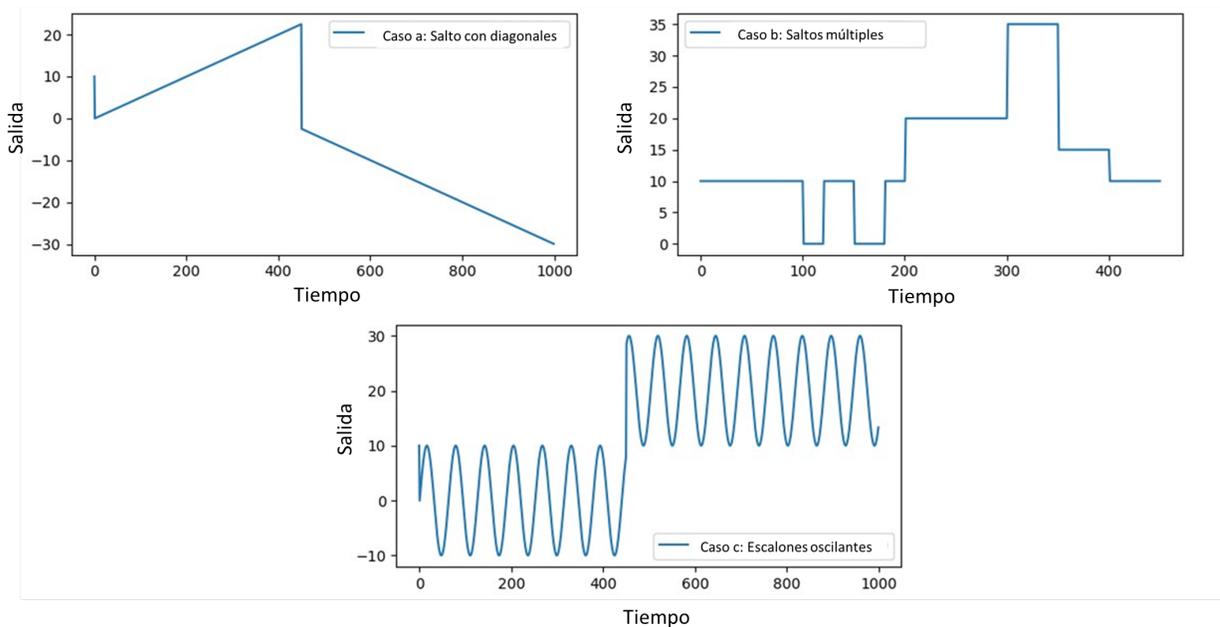


Figura 4.5: Ejemplos de casos problemáticos para la detección de una referencia de tipo escalón: diagonales conectadas por salto (izquierda arriba), saltos cercanos con respecto a la dinámica de la planta (derecha arriba) y escalones con oscilaciones (abajo).

Se establecen entonces los siguientes procedimientos para consulta y almacenamiento de datos en la memoria, así como una respuesta de emergencia.

Guardado de Datos en la Memoria

Se calcula el promedio de las puntuaciones de parámetros de una las últimas muestras (por ejemplo, las 100 últimas) y se verifica que sea mayor a un valor establecido por diseño, $\tau_{mem} \in \mathbb{R} \cap [0, 1]$. Si esta condición se cumple, se calcula el promedio de las últimas muestras de la referencia y el error cuadrático medio con respecto a dicho promedio, y si éste es menor que un valor establecido también por diseño ECM_{mem} (prueba de suficiencia de similitud con escalón, se guarda la distribución existente en ese momento en el centro de memoria más cercano al promedio).

Consulta de Datos en la Memoria

Se verifica que $|ref(k) - ref(k - 1)| > \xi_{jump}$ y se busca en los centros de memoria el valor más cercano a $ref(k)$ y, si dicho valor tiene asociada una distribución, se carga en el algoritmo y se rehace la selección de parámetros. Esta verificación tiene el problema que, dado que no se sabe cómo evolucionará la referencia, por lo que podría presentarse alguna situación como la de la Figura 4.5, lo cual podría comprometer el desempeño, si bien el filtro busca compensar esta limitación.

Respuesta de Emergencia

Dado que pueden existir casos en que el algoritmo no logre dar una correcta solución, se generaron los siguientes criterios para activar una respuesta de emergencia. En ambos casos se sanciona el desempeño de los parámetros utilizando el segundo factor de la ecuación 4.21:

1. **Si la planta está cercana a sus valores límite**, se activa una respuesta del actuador establecida por diseño, por ejemplo, desactivar un actuador o configurarlo en un valor máximo.
2. **Si el módulo del error de control supera un valor determinado**, se asume que el algoritmo no está logrando el control, activando también una respuesta de emergencia.

4.3. Diagrama de Flujo Simplificado del Algoritmo

Para ilustrar cómo se relacionan los procedimientos del algoritmo, en la Figura 4.6 se expone un diagrama de bloques simplificado de las simulaciones efectuadas.

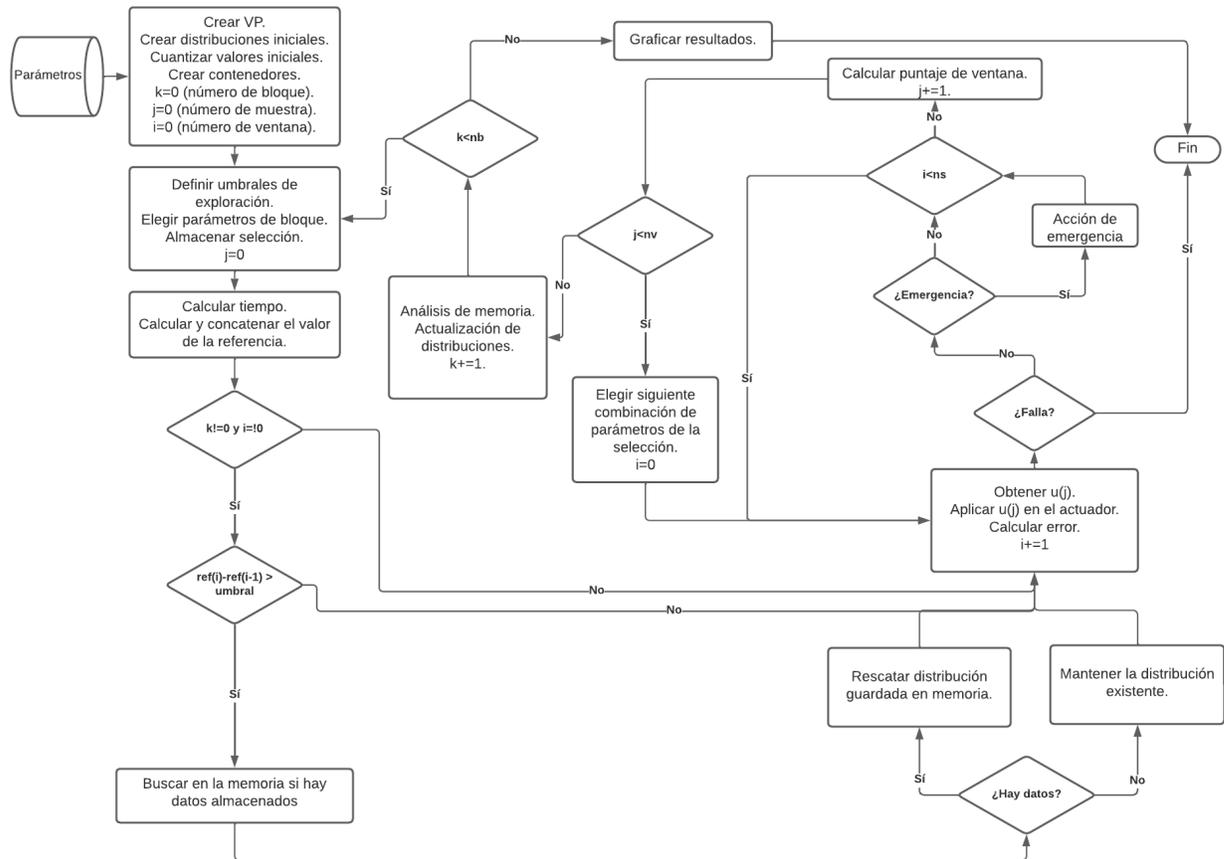


Figura 4.6: Diagrama de flujo simplificado del algoritmo de la propuesta BRL-PID para una ejecución de n_b bloques de duración.

Capítulo 5

Definición del Caso de Estudio y Resultado de Simulaciones

Se simuló en Python la propuesta BRL-PID y el PID clásico calibrado mediante ZN, variando los parámetros hasta lograr un entendimiento empírico del funcionamiento del algoritmo, puntualmente cuáles son los parámetros más influyentes para el caso de estudio. Posteriormente se hizo una selección de los casos más informativos, los cuales se detallan a continuación.

5.1. Señales de Referencia

Para estudiar la conducta del sistema se testearon distintas funciones como señal de referencia, de las cuales se escogieron las siguientes:

1. **Escalón simple:** su valor está situado en la zona media del rango de la planta. Se eligió este valor porque se observó que en esa zona hay más problemas con la estabilización de la planta (en general, la descarga de agua es más difícil de controlar) (Figura 5.1).
2. **Sucesión de escalones:** se replicaron los valores de la referencia utilizada en [18], pero los escalones duran más tiempo y se añadió una repetición de la misma secuencia para observar variaciones y el uso de la memoria (Figura 5.2).
3. **Módulo de sinusoidal:** asume valores entre $5[cm]$ y $45[cm]$, abarcando así la mayor parte del rango de operación del estanque (Figura 5.3).

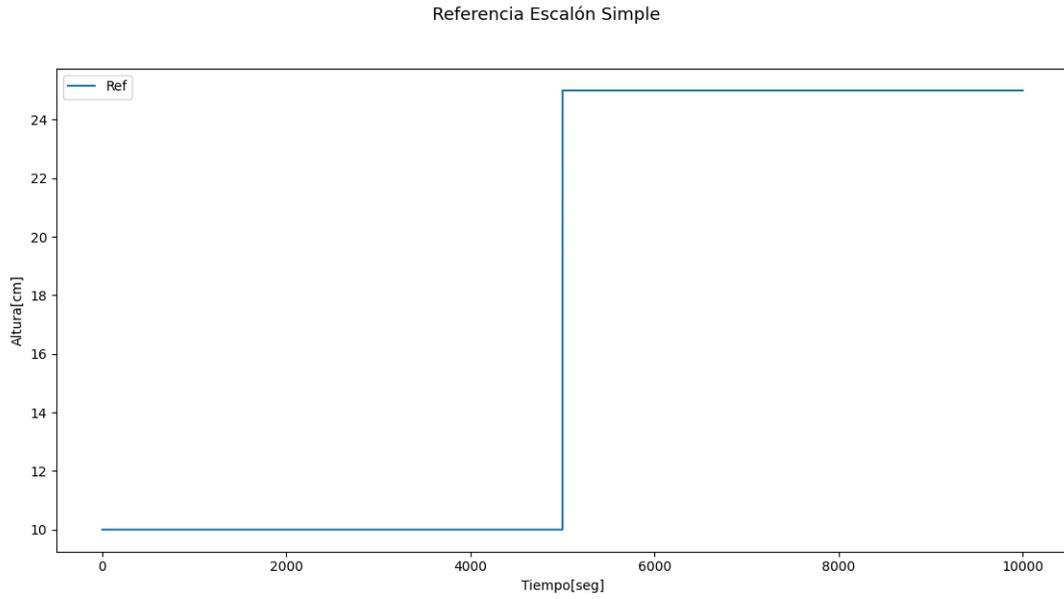


Figura 5.1: Referencia escalón simple (e.i. solo un salto) utilizada para las simulaciones de control del estanque cónico.

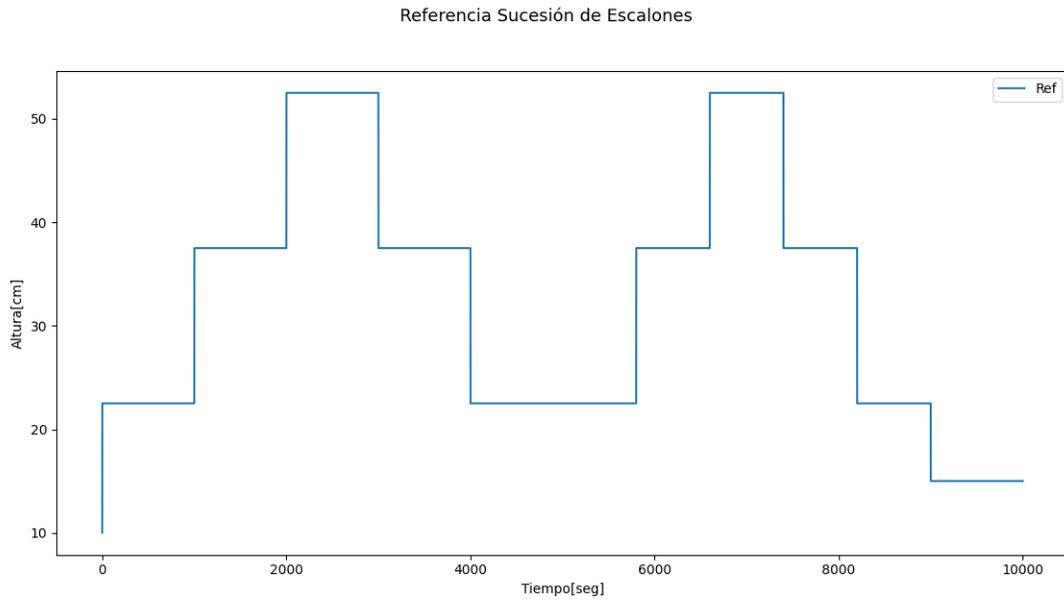


Figura 5.2: Referencia de escalones sucesivos para las simulaciones de control del estanque cónico.

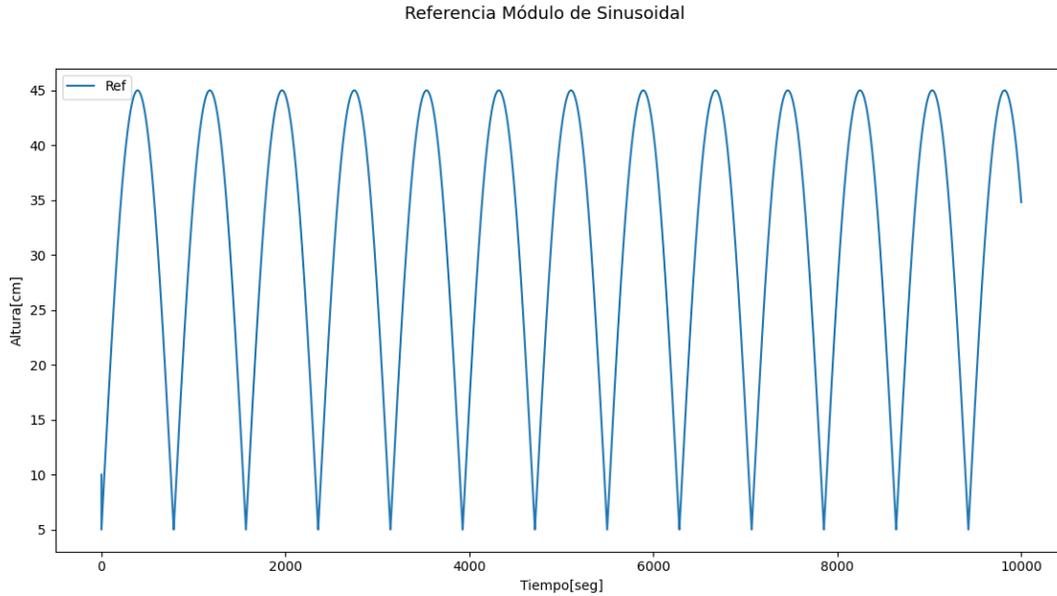


Figura 5.3: Referencia módulo de sinusoidal para las simulaciones de control del estanque cónico.

5.2. Degradación de Parámetros del Estanque Cónico

Dado que no hay información suficiente para desarrollar un perfil de degradación acorde a la dinámica real de la planta, se optó por crear funciones arbitrarias de degradación con la intención de generar un escenario de aplicación (en la planta se asignaron al azar). La Figura 5.4 expone estos resultados gráficamente:

1. **Aleatoria binaria:** el valor del parámetro para un tiempo k , con λ tal que asume aleatoriamente los valores -1 o 1, es la siguiente:

$$\delta(k + 1) = \delta(k) \cdot (1 + 9 \cdot 10^{-6} \cdot \lambda) \quad (5.1)$$

2. **Variación sinusoidal:** viene dado por la ecuación:

$$\delta(k + 1) = \delta(k) \cdot (1 - 9 \cdot 10^{-6} \cdot \sin(k \cdot 0,01)) \quad (5.2)$$

Degradación de los Parámetros del Estanque Cónico en el Tiempo

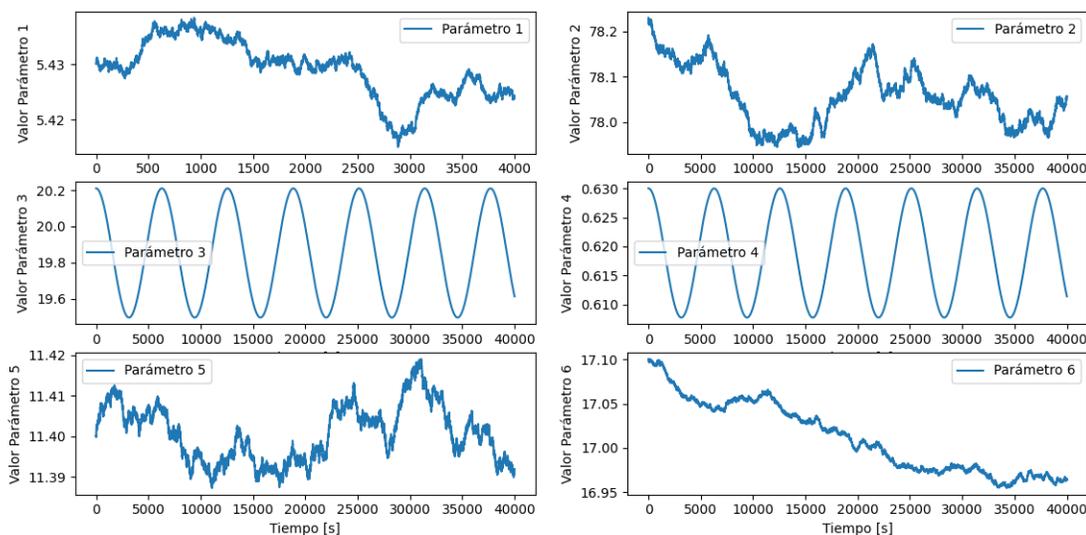


Figura 5.4: Ejemplo de la degradación de los parámetros del estanque cónico en el tiempo de acuerdo a las funciones de degradación arbitrarias.

5.3. Selección de Parámetros de Simulación

La Tabla 5.1 detalla los parámetros que se mantuvieron fijos en todas las simulaciones realizadas y sus valores. La cantidad de bloques simulados fue de $n_b = 400$ para la referencia de escalón simple y módulo de sinusoidal, y para la sucesión de escalones se consideraron $n_b = 1000$ bloques. La Tabla 5.2 expone los parámetros que sí cambiaron entre simulaciones y los valores escogidos en cada caso, etiquetadas mediante un número y una letra: el número indica la referencia utilizada, si se añadieron los ruidos de sensado y proceso, y si la memoria contaba con valores almacenados al iniciar la operación. Las letras A y B indican la configuración de los parámetros γ , τ_{mem} y $\sigma(e_{obj})$ del BRL-PID: para los casos A se utilizaron los valores $\gamma = 0, 5$, $\tau_{mem} = 0, 7$ y $\sigma(e_{obj}) = 5$, y en los casos B $\gamma = 0, 1$, $\tau_{mem} = 0, 1$ y $\sigma(e_{obj}) = 11$.

Parámetro	Valores o rangos
T_s [s]	0, 1
n_s	10
n_v	10
α_y [cm]	0, 5
α_θ [cm]	0, 001
$VP(y)$ [cm]	[0, 60]
$VP(\theta)$ [cm ⁻¹]	[0, 50]
h_{low}^* [cm]	5
h_{high}^* [cm]	55
Δ_{salto} [cm]	5
β [%]	0,8
Centros de memoria [cm]	[5, 15, 22,5, 37,5, 52,5]
$f(0)$ [%]	5
$h(0)$ [cm]	10

Tabla 5.1: Parámetros que se mantuvieron constantes entre las simulaciones del control del estanque cónico controlado por el BRL-PID y el PID clásico. El número indica la referencia utilizada, si se añadieron los ruidos de sensado y proceso, y si la memoria contaba con valores almacenados al iniciar la operación. Las letras A y B indican la configuración de los parámetros γ , τ_{mem} y $\sigma(e_{obj})$ del BRL-PID.

5.4. Comparación de Resultados

Las métricas IAE, ISE, ITAE, ITSE e ISI se calculan con una secuencia de error y el tiempo en que se obtuvo cada uno de sus valores. En el caso del BRL-PID existen los errores de control ($e_c = ref(k) - y(k)$) y de identificación ($e_i = ref(k) - y_{ref}(k)$ según la figura 4.1). Se utilizaron las secuencias del error de control por cuanto permiten la comparación directa del BRL-PID con el PID clásico calibrado por el método Ziegler-Nichols (indicado en las definiciones de los ratios como PID_{ZN}), y porque el error de identificación entrega valores más favorable para el BRL-PID, decidiendo entonces utilizar el valor más pesimista para el análisis. Se definen entonces los ratios de comparación de las métricas, las cuales para cumplir el objetivo deberán ser iguales o menores a 1,3 (Ecuaciones 5.3, 5.4, 5.5, 5.6 y 5.7).

$$\varphi(IAE) = \frac{IAE(PID_{ZN})}{IAE(PID_{BRL-PID})} \quad (5.3)$$

$$\varphi(ISE) = \frac{ISE(PID_{ZN})}{ISE(PID_{BRL-PID})} \quad (5.4)$$

$$\varphi(ITAE) = \frac{ITAE(PID_{ZN})}{ITAE(PID_{BRL-PID})} \quad (5.5)$$

$$\varphi(ITSE) = \frac{ITSE(PID_{ZN})}{ITSE(PID_{BRL-PID})} \quad (5.6)$$

$$\varphi(ISI) = \frac{ISI(PID_{ZN})}{ISI(PID_{BRL-PID})} \quad (5.7)$$

Simulación	Referencia	Ruido	Memoria	γ	τ_{mem}	$\sigma(e_{obj})$
1A	Escalón simple	No	No	0,5	0,7	5
1B	Escalón simple	No	No	0,1	0,1	11
2A	Escalón simple	Sí	No	0,5	0,7	5
2B	Escalón simple	Sí	No	0,1	0,1	11
3A	Escalón simple	No	Sí	0,5	0,7	5
3B	Escalón simple	No	Sí	0,1	0,1	11
4A	Escalón simple	Sí	Sí	0,5	0,7	5
4B	Escalón simple	Sí	Sí	0,1	0,1	11
5A	Sucesión de escalones	No	No	0,5	0,7	5
5B	Sucesión de escalones	No	No	0,1	0,1	11
6A	Sucesión de escalones	No	Sí	0,5	0,7	5
6B	Sucesión de escalones	No	Sí	0,1	0,1	11
7A	Sucesión de escalones	Sí	No	0,5	0,7	5
7B	Sucesión de escalones	Sí	No	0,1	0,1	11
8A	Sucesión de escalones	Sí	Sí	0,5	0,7	5
8B	Sucesión de escalones	Sí	Sí	0,1	0,1	11
9A	Módulo de sinusoidal	No	No	0,5	0,7	5
9B	Módulo de sinusoidal	No	No	0,1	0,1	11
10A	Módulo de sinusoidal	Sí	No	0,5	0,7	5
10B	Módulo de sinusoidal	Sí	No	0,1	0,1	11

Tabla 5.2: Parámetros de simulación que fueron modificados entre simulaciones para el control del estanque cónico mediante el BRL-PID y el PID clásico. El número indica la referencia, presencia de ruido y de valores en la memoria, y las letras A y B indican la configuración de los parámetros γ , τ_{mem} y $\sigma(e_{obj})$ del BRL-PID.

5.5. Resultados de Simulaciones

Se seleccionaron las simulaciones que resultaron más informativas del funcionamiento del sistema. La Tabla 5.3 expone los ratios de las métricas obtenidas para cada una, utilizando el error de control $e_c = ref(k) - y(k)$ y las ecuaciones 5.3, 5.4, 5.5, 5.6 y 5.7. Los valores inferiores a 1,3 se indican con color verde, los resultados en el rango $(1, 3, 1, 5]$ se indican de color ocre, y los que están sobre 1,5 ese valor de color rojo.

Simulación	Descripción	$\varphi(IAE)$	$\varphi(ISE)$	$\varphi(ITAE)$	$\varphi(ITSE)$	$\varphi(ISI)$
1A	Escalón Simple, Sin Ruido, Sin Memoria	0,5529	0,8908	0,5261	0,8580	39,7992
1B	Escalón Simple, Sin Ruido, Sin Memoria	2,1201	1,9838	2,7832	2,4114	57,8008
2A	Escalón Simple, Con Ruido, Sin Memoria	1,9774	4,0593	2,2156	5,02297	58159,2587
2B	Escalón Simple, Con Ruido, Sin Memoria	1,6389	2,8863	1,7586	3,3274	776,1035
3A	Escalón Simple, Sin Ruido, Con Memoria	2,2239	2,5962	2,7296	3,0879	799,8173
3B	Escalón Simple, Sin Ruido, Con Memoria	0,5935	1,1393	0,5518	1,0995	1,0680
4A	Escalón Simple, Con Ruido, Con Memoria	1,0582	1,4880	1,1848	1,7591	2049,7172
4B	Escalón Simple, Con Ruido, Con Memoria	0,5946	0,4476	0,5909	0,4138	0,8334
5A	Sucesión de Escalones, Sin Ruido, Sin Memoria	1,6774	1,7605	1,6265	1,7111	10142,9414
5B	Sucesión de Escalones, Sin Ruido, Sin Memoria	1,2712	1,2124	1,2230	1,1809	459,0202
6A	Sucesión de Escalones, Sin Ruido, Con Memoria	1,6672	2,1321	1,7003	1,9479	6738,1315
6B	Sucesión de Escalones, Sin Ruido, Con Memoria	1,2935	1,7201	1,1718	1,4488	1703,8488
7A	Sucesión de Escalones, Con Ruido, Sin Memoria	1,3896	1,5602	1,4172	1,6439	11543,0289
7B	Sucesión de Escalones, Con Ruido, Sin Memoria	1,2440	1,7914	1,2350	1,7756	12672,3331
8A	Sucesión de Escalones, Con Ruido, Con Memoria	1,2262	1,6523	1,1742	1,5704	5241,9243
8B	Sucesión de Escalones, Con Ruido, Con Memoria	1,1404	1,6822	1,1429	1,6303	15150,9771
9A	Módulo de Sinusoidal, Sin Ruido, Sin Memoria	0,0452	0,0323	0,0333	0,0031	1,0335
9B	Módulo de Sinusoidal, Sin Ruido, Sin Memoria	0,0452	0,0323	0,0333	0,0031	1,0335
10A	Módulo de Sinusoidal, Con Ruido, Sin Memoria	1,6785	2,3978	1,6800	2,5831	28910,6006
10B	Módulo de Sinusoidal, Con Ruido, Sin Memoria	0,8933	0,6733	0,7901	0,5045	653,3582

Tabla 5.3: Resultados obtenidos en todas las simulaciones seleccionadas, indicando nombre de simulación, características de la referencia, presencia o ausencia de ruido en la planta y presencia o ausencia de datos almacenados en la memoria al inicio de la simulación, ratio de las métricas globales de desempeño (BRL-PID sobre PID clásico).

5.5.1. Simulaciones 1: Referencia Escalón, Sin Ruido, Sin Memoria

La Tabla 5.4 presenta los resultados del primer par de simulaciones, en los que se observa un mejor desempeño del BRL-PID sobre el PID clásico, excepto en el esfuerzo de control, en el caso A. Las figuras 5.5 y 5.6 exponen la respuesta en el tiempo de la planta para los casos A y B, cada uno con el BRL-PID (gráfico superior) y el PID clásico (gráfico inferior). En el caso B el menor desempeño se atribuye al hecho que la relajación en las condiciones de optimalidad (τ_{mem} y $\sigma(e_{obj})$) fomentan la existencia de oscilaciones, las cuales son además menos penalizadas por el cambio en el parámetro γ . El mayor esfuerzo de control en ambos casos se relaciona al proceso de exploración.

Simulación	Descripción	$\varphi(IAE)$	$\varphi(ISE)$	$\varphi(ITAE)$	$\varphi(ITSE)$	$\varphi(ISI)$
1A	Escalón Simple, Sin Ruido, Sin Memoria	0,5529	0,8908	0,5261	0,8580	39,7992
1B	Escalón Simple, Sin Ruido, Sin Memoria	2,1201	1,9838	2,7832	2,4114	57,8008

Tabla 5.4: Resultados obtenidos en las simulaciones 1A y 1B: Referencia escalón simple, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.

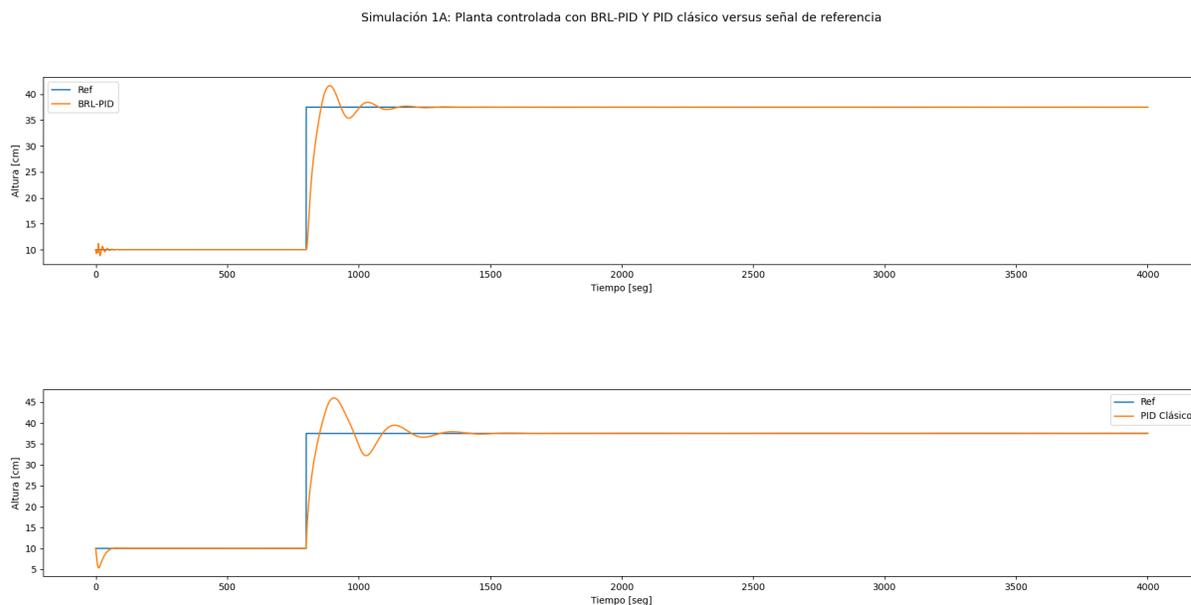


Figura 5.5: Simulación 1A: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia escalón simple, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.

Simulación 1B: Planta controlada con BRL-PID Y PID clásico versus señal de referencia

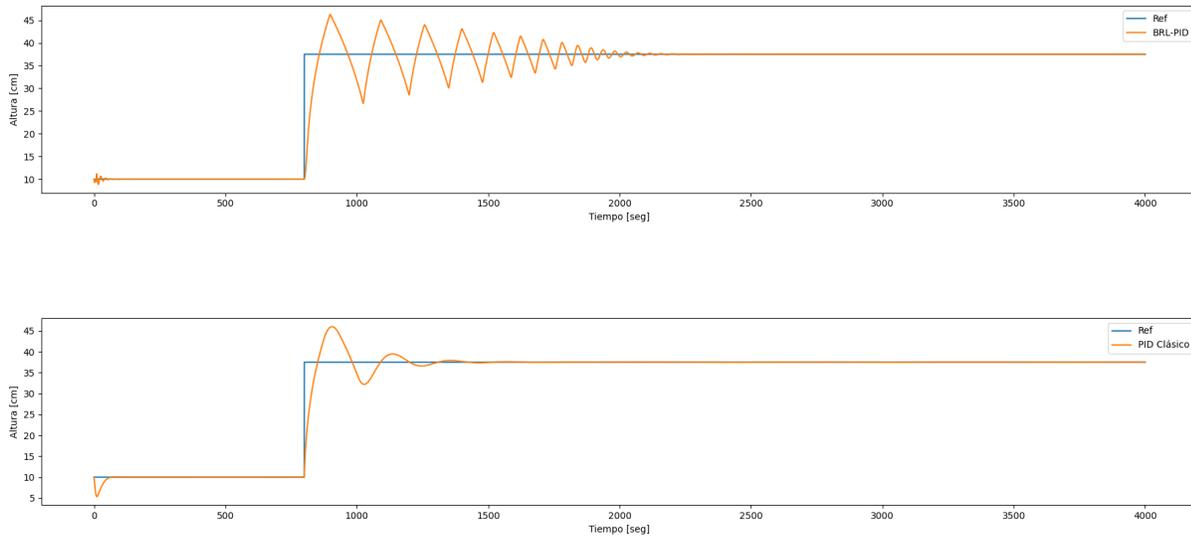


Figura 5.6: Simulación 1B: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia escalón simple, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.

5.5.2. Simulaciones 2: Referencia Escalón, Con Ruido, Sin Memoria

En esta simulación no hay ruido y se añadieron los valores de memoria al BRL-PID, obteniendo las métricas globales que se indican en la Tabla 5.5). **En el caso 2B se hizo una modificación en el valor γ : en vez de 0,1 se utilizó 0,5**, ya que de lo contrario empeoraban las métricas, lo que se atribuye a que, al penalizar las oscilaciones alrededor del valor objetivo, en este caso se lograba mejorar la respuesta. Las Figuras 5.7, 5.8 y 5.9 exponen los resultados de las plantas en los 3 casos. En cuanto al esfuerzo de control, la exploración es mucho menor en el caso B, como se observa en las Figuras 5.10 y 5.11 en las que se aprecia la dinámica de los parámetros durante la simulación.

Simulación	Descripción	$\varphi(IAE)$	$\varphi(ISE)$	$\varphi(ITAE)$	$\varphi(ITSE)$	$\varphi(ISI)$
2A	Escalón Simple, Con Ruido, Sin Memoria	1,9774	4,0593	2,2156	5,02297	58.159,2587
2B	Escalón Simple, Con Ruido, Sin Memoria	1,6389	2,8863	1,7586	3,3274	776,1035

Tabla 5.5: Resultados obtenidos en las simulaciones 2A y 2B: Referencia escalón simple, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.

Simulación 2A: Planta controlada con BRL-PID Y PID clásico versus señal de referencia

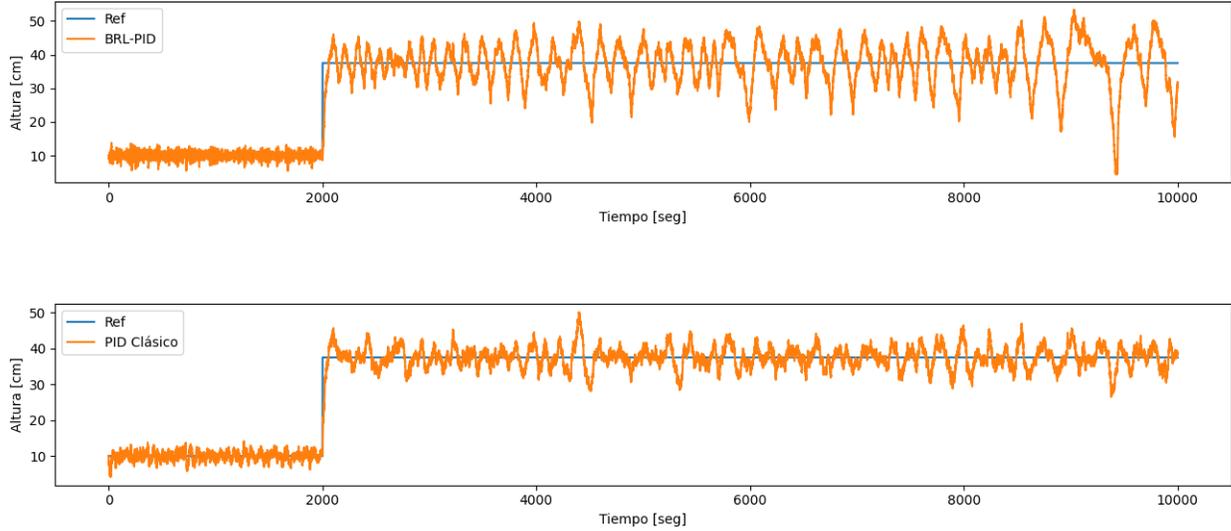


Figura 5.7: Simulación 2A: Respuesta de la planta controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia escalón simple, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.

Simulación 2B: Planta controlada con BRL-PID Y PID clásico versus señal de referencia

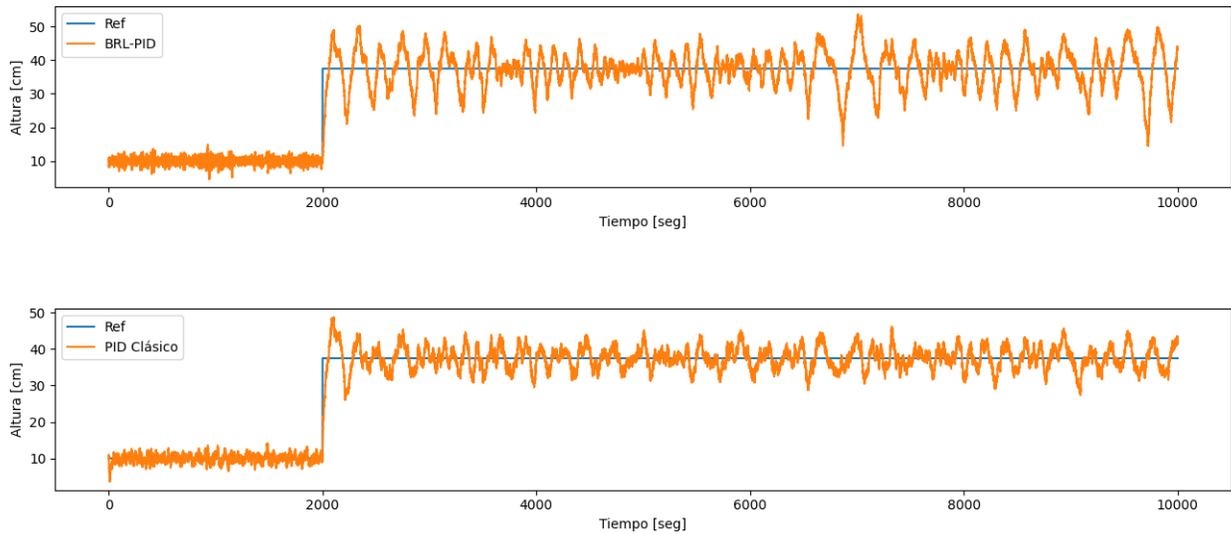


Figura 5.8: Simulación 2B (con alteración del parámetro de evaluación, $\gamma = 0,5$): Respuesta controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia escalón simple, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.

Simulación 2B: Planta controlada con BRL-PID Y PID clásico versus señal de referencia

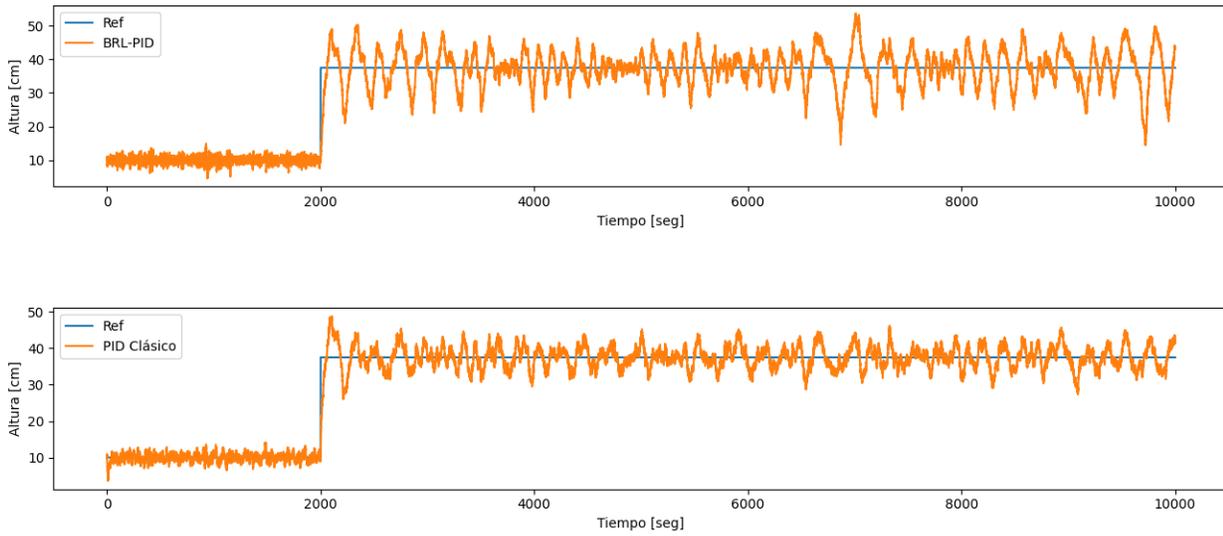


Figura 5.9: Simulación 2B (γ sin alterar para el caso B): Respuesta de la planta controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia escalón simple, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.

Simulación 2A: Desarrollo de los parámetros del BRL-PID en el tiempo

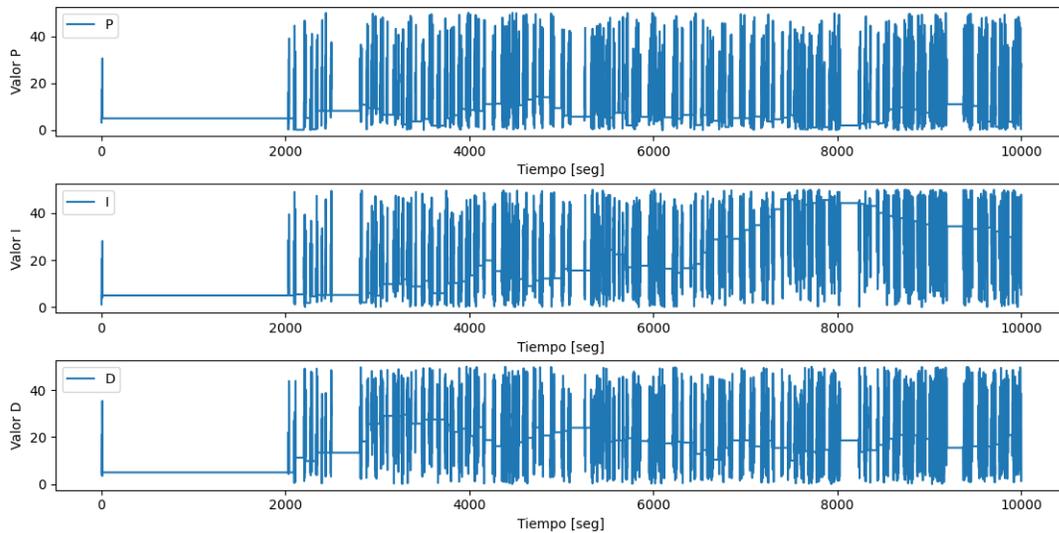


Figura 5.10: Simulación 2A: Dinámica de los parámetros del BRL-PID. Referencia escalón simple, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.

Simulación 2B: Desarrollo de los parámetros del BRL-PID en el tiempo

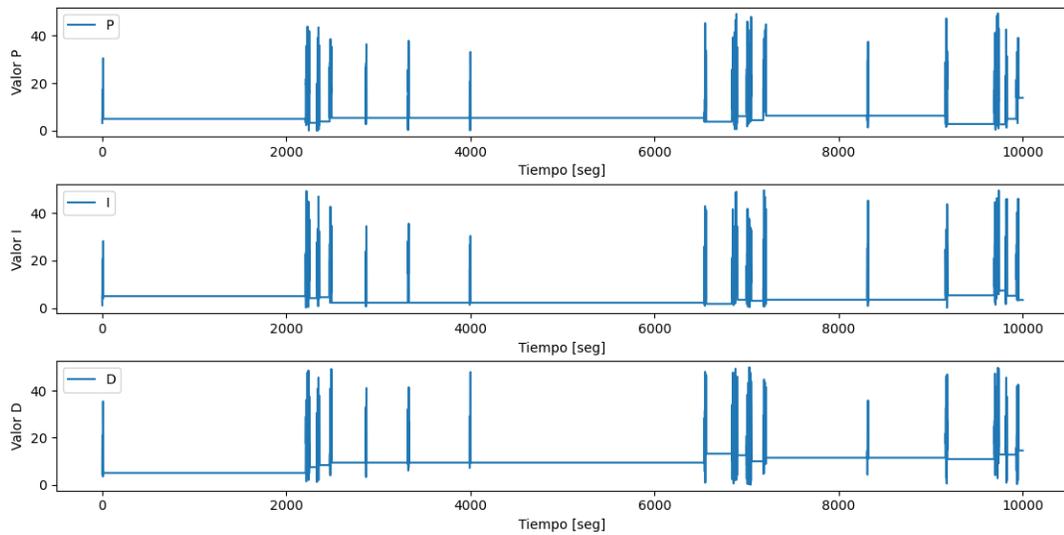


Figura 5.11: Simulación 2B (con alteración del parámetro de evaluación, $\gamma = 0,5$): Dinámica de los parámetros del BRL-PID. Referencia escalón simple, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.

5.5.3. Simulaciones 3: Referencia Escalón, Sin Ruido, Con Memoria

La Tabla 5.6 expone los resultados de las métricas y las Figuras 5.12 y 5.13 las dinámicas observadas. En este caso se simuló sin ruido de planta y con memoria almacenada al inicio, obteniendo mejores resultados en la simulación B en todas las métricas, lo que se atribuye a que ésta permitió dar directamente con una combinación que cumplía una condición de optimalidad relajada en el caso B, no así en el A en el que el sistema exploró más de lo necesario.

Simulación	Descripción	$\varphi(IAE)$	$\varphi(ISE)$	$\varphi(ITAE)$	$\varphi(ITSE)$	$\varphi(ISI)$
3A	Escalón Simple, Sin Ruido, Con Memoria	2,2239	2,5962	2,7296	3,0879	799,8173
3B	Escalón Simple, Sin Ruido, Con Memoria	0,5935	1,1393	0,5518	1,0995	1,0680

Tabla 5.6: Resultados obtenidos en las simulaciones 3A y 3B: Referencia escalón simple, sin ruidos en la planta y con valores en la memoria al inicio de la simulación.

Simulación 3A: Planta controlada con BRL-PID Y PID clásico versus señal de referencia

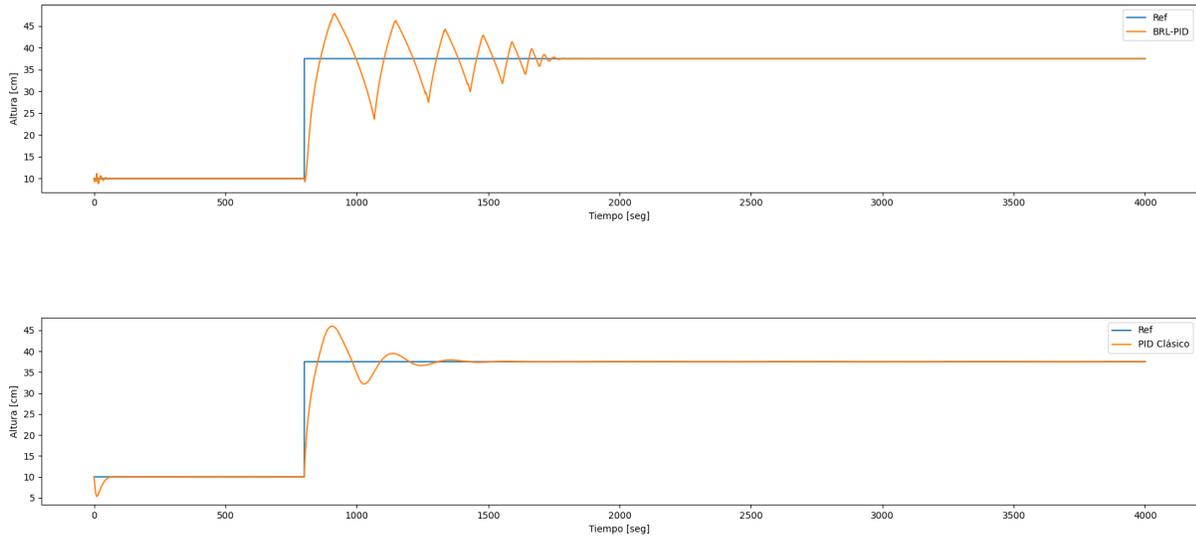


Figura 5.12: Simulación 3A: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia escalón simple, sin ruidos en la planta y con valores en la memoria al inicio de la simulación.

Simulación 3B: Planta controlada con BRL-PID Y PID clásico versus señal de referencia

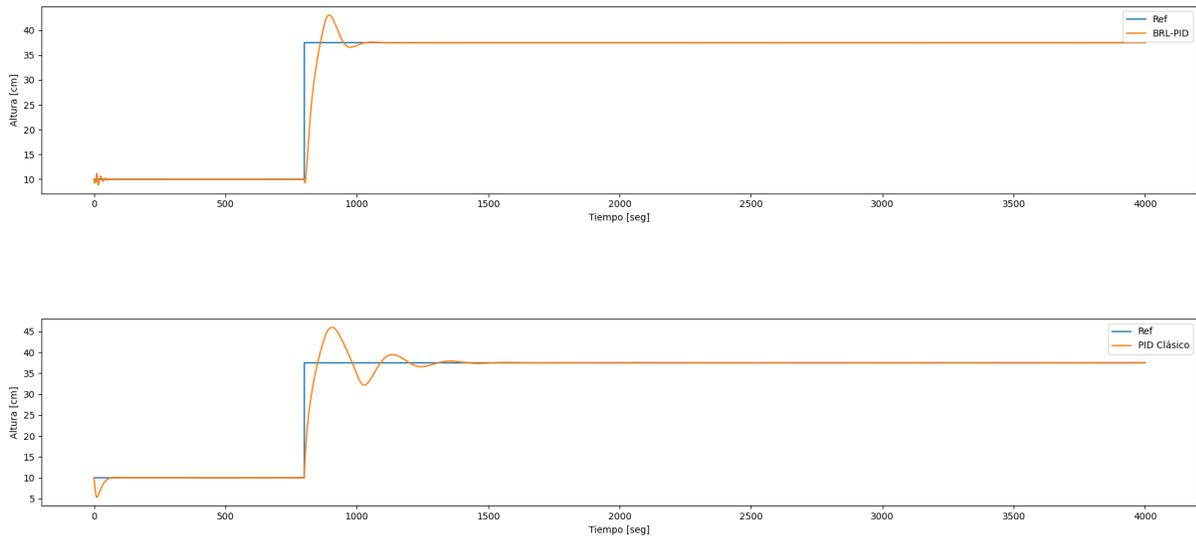


Figura 5.13: Simulación 3B: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia escalón simple, sin ruidos en la planta y con valores en la memoria al inicio de la simulación.

5.5.4. Simulaciones 4: Referencia Escalón, Con Ruido, Con Memoria

En las simulaciones 4A y 4B se añadió ruido y valores en la memoria, observando un resultado muy similar al de las simulaciones 3. Se deduce que la justificación de lo observado responde también al efecto de la memoria sumado a la relajación de las condiciones, lo que detiene la exploración antes en el caso B.

Simulación	Descripción	$\varphi(IAE)$	$\varphi(ISE)$	$\varphi(ITAe)$	$\varphi(ITSE)$	$\varphi(ISI)$
4A	Escalón Simple, Con Ruido, Con Memoria	1,0582	1,4880	1,1848	1,7591	2.049,7172
4B	Escalón Simple, Con Ruido, Con Memoria	0,5946	0,4476	0,5909	0,4138	0,8334

Tabla 5.7: Resultados obtenidos en las simulaciones 4A y 4B: Referencia escalón simple, con ruidos en la planta y con valores en la memoria al inicio de la simulación.



Figura 5.14: Simulación 4A: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia sucesión de escalones, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.

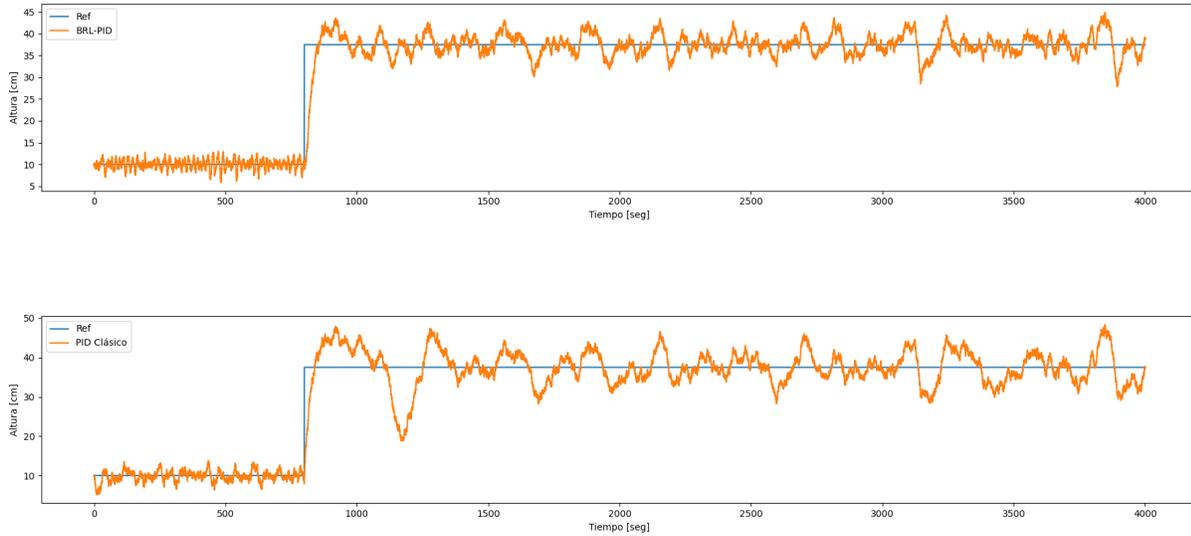


Figura 5.15: Simulación 4B: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia sucesión de escalones, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.

5.5.5. Simulaciones 5: Referencia de Escalones Sucesivos, Sin Ruido, Sin Memoria

Las simulaciones 5A y 5B exponen el caso de una referencia de sucesión de escalones, sin ruido ni valores previos en la memoria, obteniendo mejores resultados en el caso B, lo cual se atribuye a las mismas razones que en las simulaciones 3 y 4, añadiendo que en este caso se activaron las rutinas de emergencia. La tabla 5.8 expone las métricas y las Figuras 5.16, 5.17 la dinámica de la planta para los casos A y B, respectivamente. La figura 5.18 expone un detalle de la respuesta en el caso A donde se observa la respuesta de emergencia (cuando se llega a un nivel igual o superior a 55 [cm]), que hace que la salida del controlador sea $u(k) = 0$ (establecida por diseño), y penaliza los parámetros que estaban en el controlador durante la ejecución de ésta. El mismo efecto se produce en el rango inferior de la planta, salvo que en este caso la respuesta predeterminada del control es $u(k) = 100$, como se detalla en la figura 5.19. Observando la dinámica de la planta, se deduce que esto hace que las métricas globales sean mejor de lo esperado al observar la respuesta del resto de la simulación, si bien se detecta visualmente una respuesta con más oscilaciones que la que logra el PID clásico. Las diferencias en el esfuerzo de control entre los casos A y B se deduce que es por la modificación en los criterios de optimalidad.

Simulación	Descripción	$\varphi(IAE)$	$\varphi(ISE)$	$\varphi(ITAE)$	$\varphi(ITSE)$	$\varphi(ISI)$
5A	Sucesión de Escalones, Sin Ruido, Sin Memoria	1,6774	1,7605	1,6265	1,7111	10.142,9414
5B	Sucesión de Escalones, Sin Ruido, Sin Memoria	1,2712	1,2124	1,2230	1,1809	459,0202

Tabla 5.8: Resultados obtenidos en las simulaciones 5A y 5B: Referencia sucesión de escalones, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.

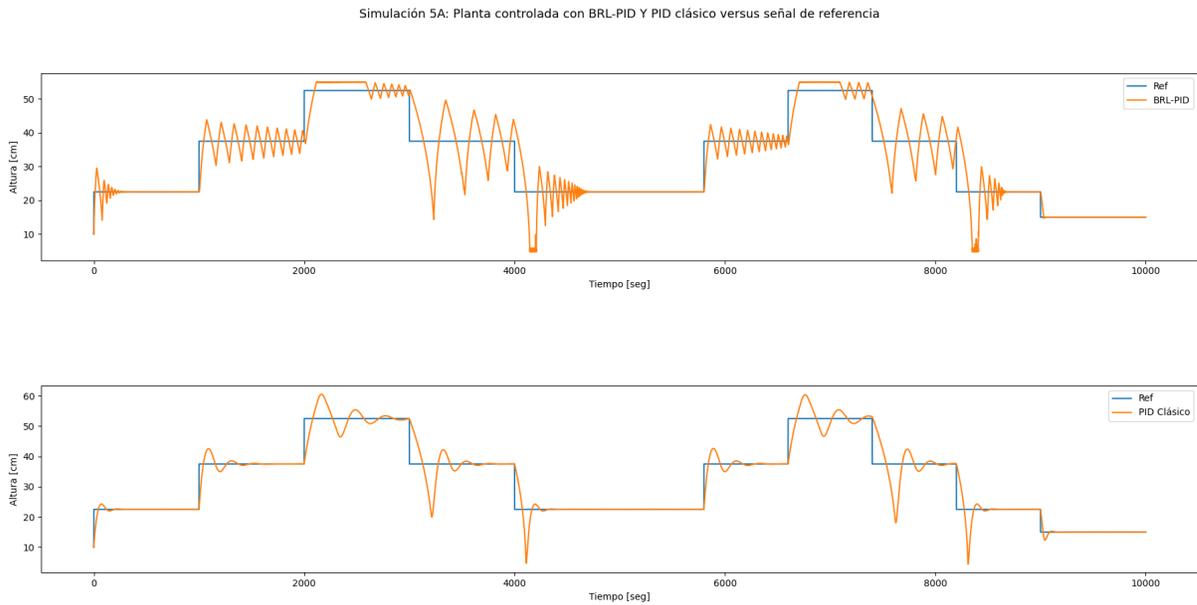


Figura 5.16: Simulación 5A: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia sucesión de escalones, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.

Simulación 5B: Planta controlada con BRL-PID Y PID clásico versus señal de referencia

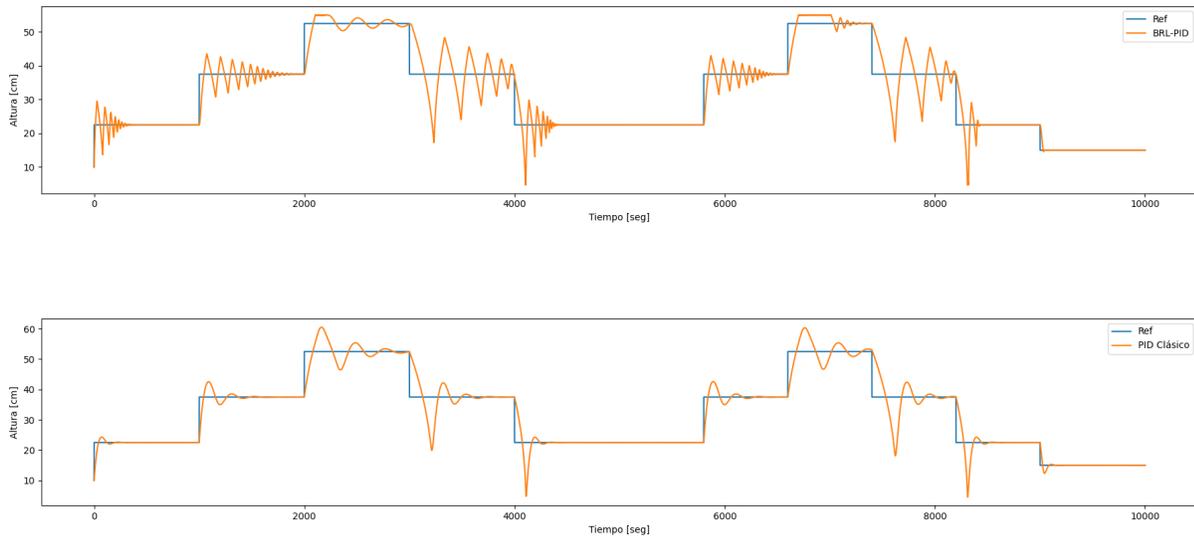


Figura 5.17: Simulación 5B: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia sucesión de escalones, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.

Simulación 5A: Planta controlada con BRL-PID Y PID clásico versus señal de referencia

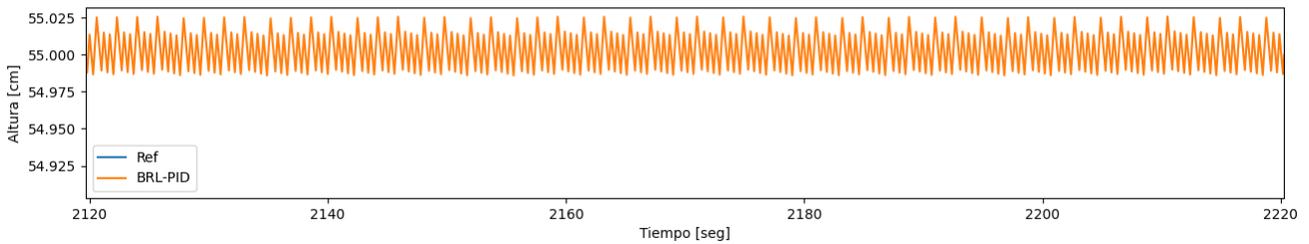


Figura 5.18: Simulación 5A: Acercamiento de la respuesta de la planta controlada con el BRL-PID entre los 2120[s] y 2220[s] (referencia sucesión de escalones, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación). Se aprecian oscilaciones en la salida de planta debido a la rutina de emergencia una vez se supera un umbral superior de emergencia de 55[cm].

Simulación 5A: Planta controlada con BRL-PID Y PID clásico versus señal de referencia

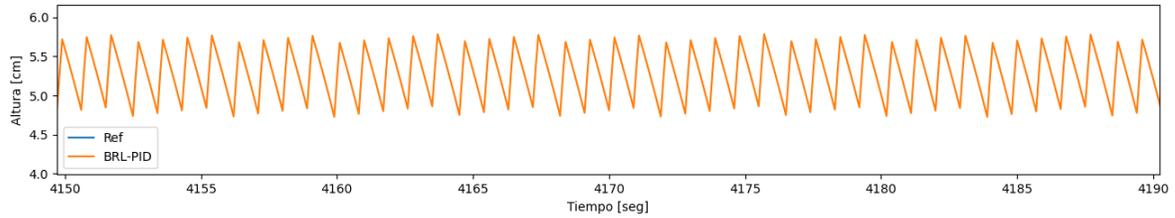


Figura 5.19: Simulación 5A: Acercamiento de la respuesta de la planta controlada con el BRL-PID entre los 4150[s] y 4190[s] referencia sucesión de escalones, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación). Se aprecian oscilaciones en la salida de planta debido a la rutina de emergencia una vez se supera un umbral inferior de emergencia de 5[cm].

5.5.6. Simulaciones 6: Referencia de Escalones Sucesivos, Sin Ruido, Con Memoria

En las simulaciones 6A y 6B ocurre lo mismo que en las simulaciones 5A Y 5B con respecto a las métricas, el esfuerzo de control y el efecto de las variaciones de los parámetros. En particular se observa cómo la reacción de emergencia del controlador BRL-PID mejora las métricas en las situaciones extremas, si bien en el resto de las simulaciones (por ejemplo en la primera bajada de los escalones) se observa una conducta menos estable.

Simulación	Descripción	$\varphi(IAE)$	$\varphi(ISE)$	$\varphi(ITAE)$	$\varphi(ITSE)$	$\varphi(ISI)$
6A	Sucesión de Escalones, Sin Ruido, Con Memoria	1,6672	2,1321	1,7003	1,9479	6.738,1315
6B	Sucesión de Escalones, Sin Ruido, Con Memoria	1,2935	1,7201	1,1718	1,4488	1.703,8488

Tabla 5.9: Resultados obtenidos en las simulaciones 6A y 6B: Referencia sucesión de escalones, sin ruidos en la planta y con valores en la memoria al inicio de la simulación.

Simulación 6A: Planta controlada con BRL-PID Y PID clásico versus señal de referencia

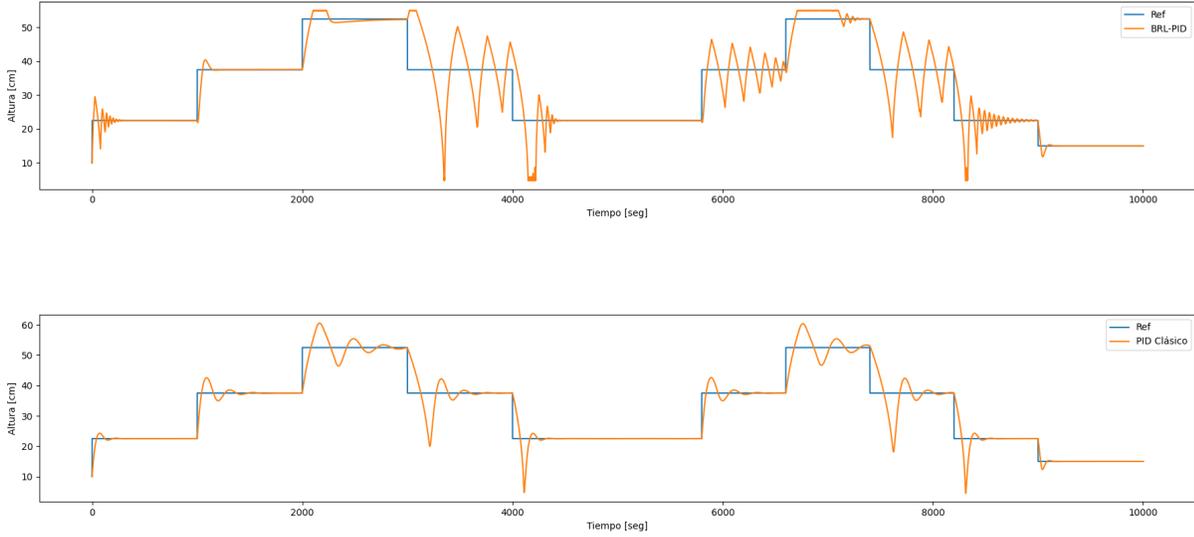


Figura 5.20: Simulación 6A: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia sucesión de escalones, sin ruidos en la planta y con valores en la memoria al inicio de la simulación.

Simulación 6B: Planta controlada con BRL-PID Y PID clásico versus señal de referencia

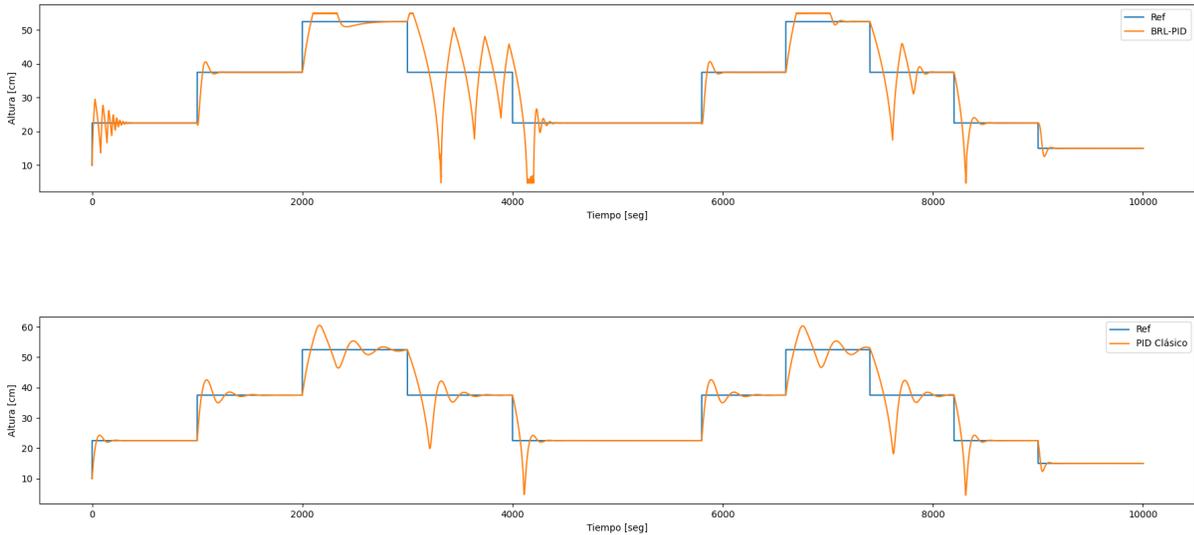


Figura 5.21: Simulación 6B: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia sucesión de escalones, sin ruidos en la planta y con valores en la memoria al inicio de la simulación.

5.5.7. Simulaciones 7: Referencia de Escalones Sucesivos, Con Ruido, Sin Memoria

Para las simulaciones 7A y 7B se utilizó la referencia de sucesión de escalones, la planta con ruido y con memoria inicial. En este caso la configuración A mejora las métricas IAE e ITAE, no así las ISE e ITSE, de lo que se deduce que hubo mayores oscilaciones en el BRL-PID que en el PID clásico, o que las oscilaciones que se observan alrededor de los 1000[s], 3000[s], 4000[s], 7300[s] y 8300[s] generaron este resultado. El mayor esfuerzo de control para el caso B, si bien las Figuras 5.24 Y 5.25 muestran que en el caso B el sistema exploró menos, se explica porque la exploración en el caso A tendió a ser en valores más bajos que los de estabilización de B, causando el resultado obtenido.

Simulación	Descripción	$\varphi(IAE)$	$\varphi(ISE)$	$\varphi(ITAE)$	$\varphi(ITSE)$	$\varphi(ISI)$
7A	Sucesión de Escalones, Con Ruido, Sin Memoria	1,3896	1,5602	1,4172	1,6439	11.543,0289
7B	Sucesión de Escalones, Con Ruido, Sin Memoria	1,2440	1,7914	1,2350	1,7756	12.672,3331

Tabla 5.10: Resultados obtenidos en las simulaciones 7A y 7B: Referencia sucesión de escalones, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.

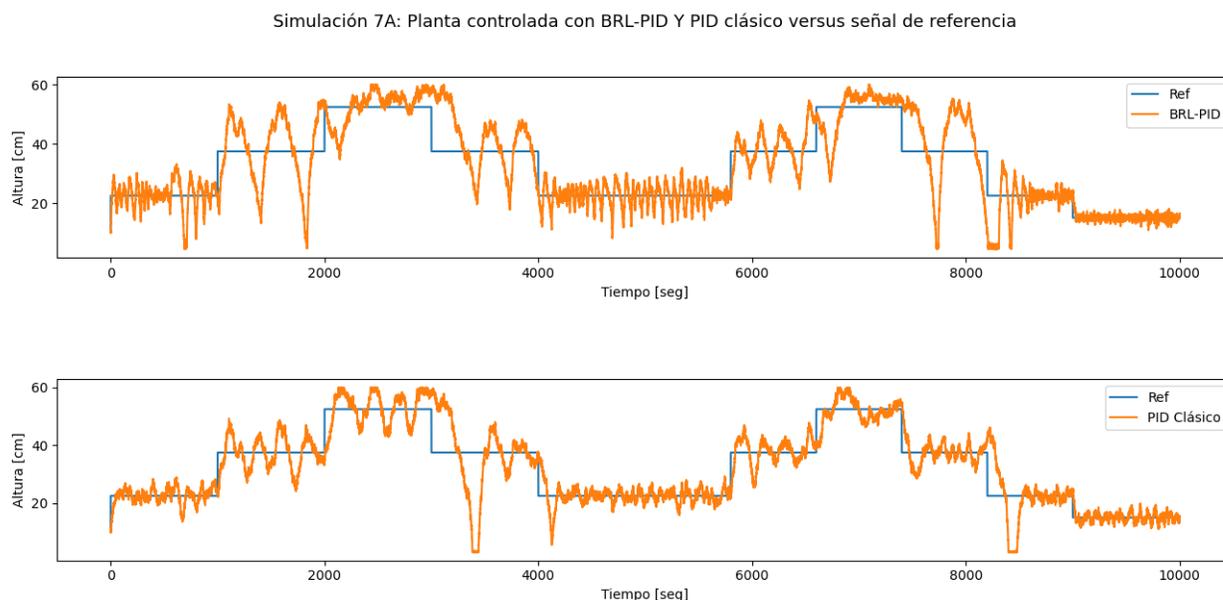


Figura 5.22: Simulación 7A: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia sucesión de escalones, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.

Simulación 7B: Planta controlada con BRL-PID Y PID clásico versus señal de referencia

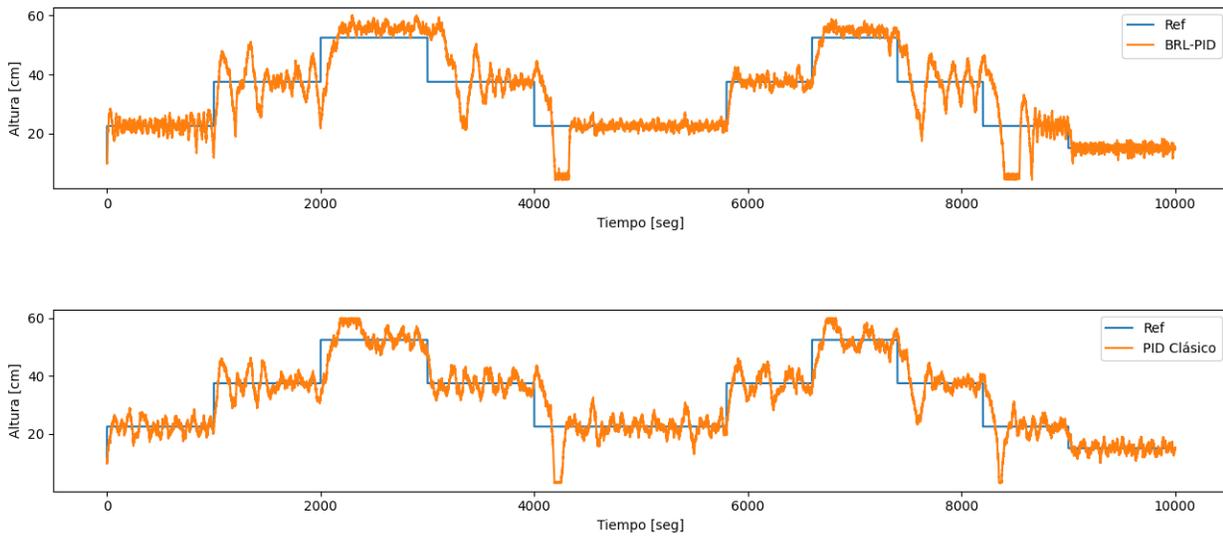


Figura 5.23: Simulación 7B: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia sucesión de escalones, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.

Simulación 7A: Desarrollo de los parámetros del BRL-PID en el tiempo

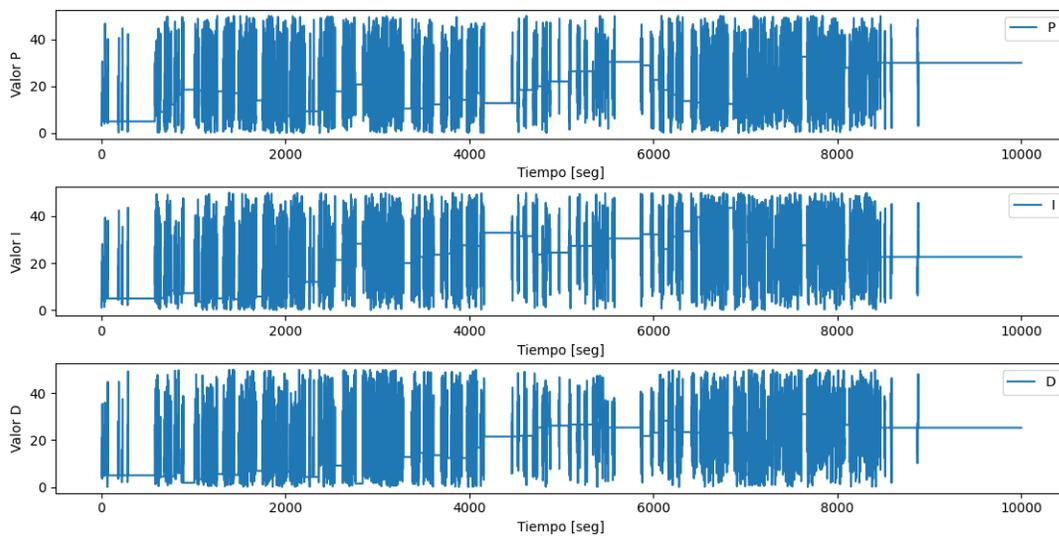


Figura 5.24: Simulación 7A: Dinámica de los parámetros del BRL-PID. Referencia sucesión de escalones, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.

Simulación 7B: Desarrollo de los parámetros del BRL-PID en el tiempo

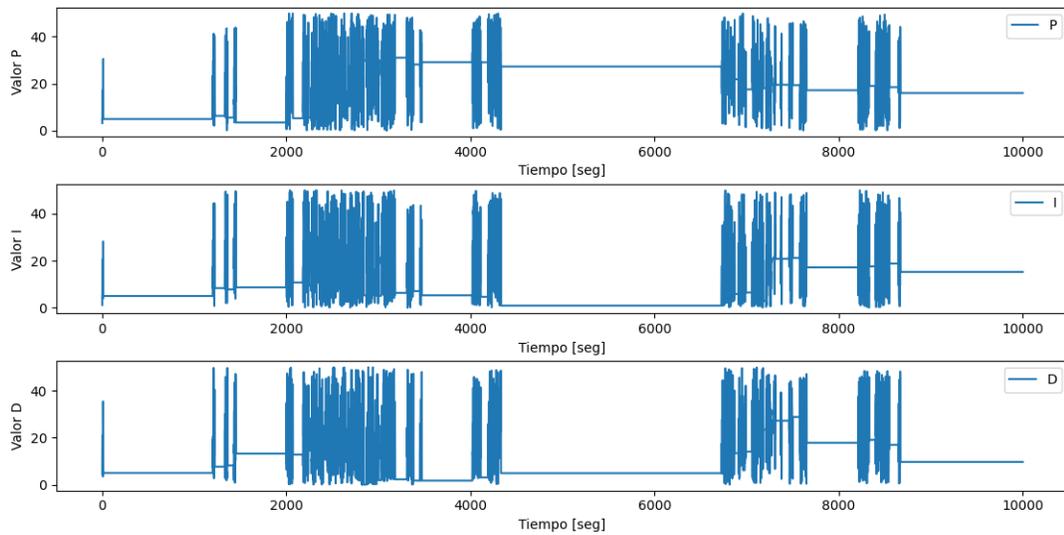


Figura 5.25: Simulación 7B: Dinámica de los parámetros del BRL-PID. Referencia sucesión de escalones, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.

5.5.8. Simulaciones 8: Referencia de Escalones Sucesivos, Con Ruido, Con Memoria

En las simulaciones 8A y 8B ocurrió algo similar a las 7A y 7B con respecto al esfuerzo de control y las métricas con ponderación temporal, con la diferencia que la memoria ayudó a mejorar las métricas IAE e ITAE. Las Figuras 5.28 y 5.29 refuerzan lo anterior, donde se observa que a pesar de haber mayor exploración, ésta tiende a ser en valores más altos en el caso B, aumentando así el esfuerzo de control con respecto al caso A.

Simulación	Descripción	$\varphi(IAE)$	$\varphi(ISE)$	$\varphi(ITAE)$	$\varphi(ITSE)$	$\varphi(ISI)$
8A	Sucesión de Escalones, Con Ruido, Con Memoria	1,2262	1,6523	1,1742	1,5704	5.241,9243
8B	Sucesión de Escalones, Con Ruido, Con Memoria	1,1404	1,6822	1,1429	1,6303	15.150,9771

Tabla 5.11: Resultados obtenidos en las simulaciones 8A y 8B: Referencia sucesión de escalones, con ruidos en la planta y con valores en la memoria al inicio de la simulación.

Simulación 8A: Planta controlada con BRL-PID Y PID clásico versus señal de referencia

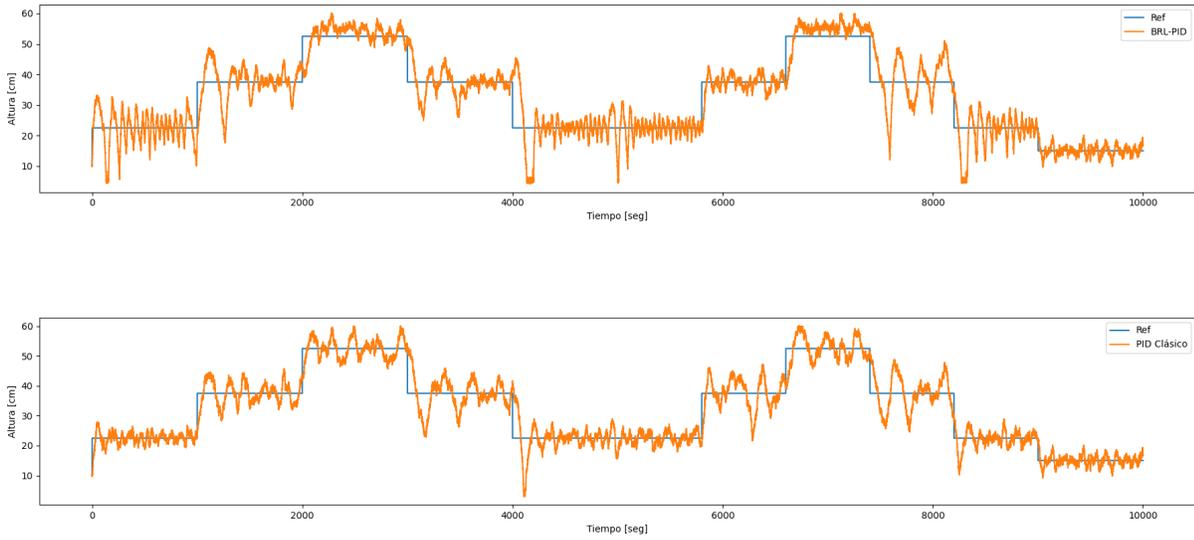


Figura 5.26: Simulación 8A: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia sucesión de escalones, con ruidos en la planta y con valores en la memoria al inicio de la simulación.

Simulación 8B: Planta controlada con BRL-PID Y PID clásico versus señal de referencia

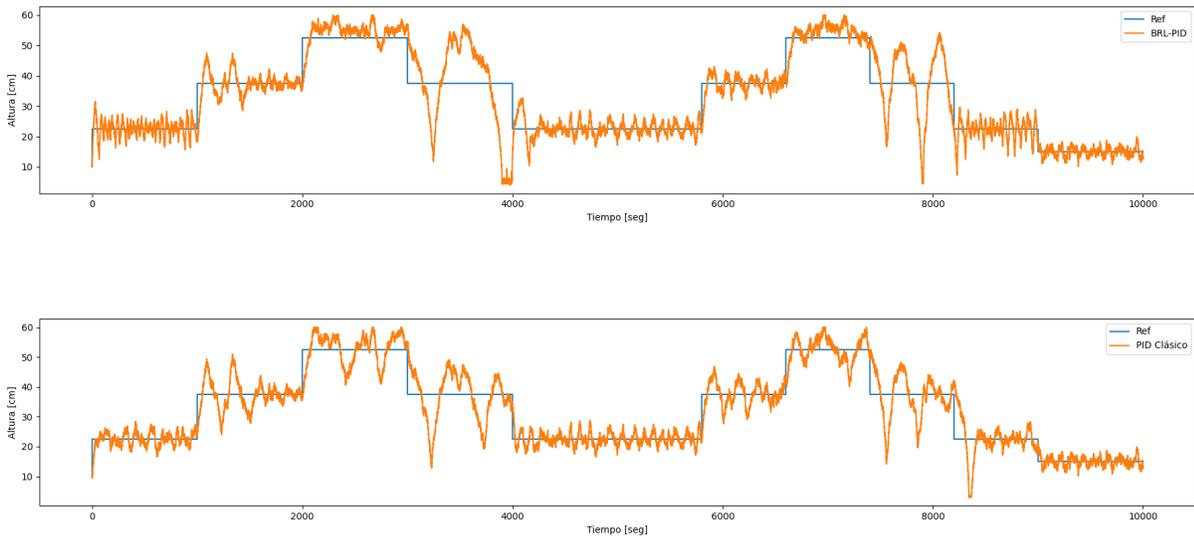


Figura 5.27: Simulación 8B: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia sucesión de escalones, con ruidos en la planta y con valores en la memoria al inicio de la simulación.

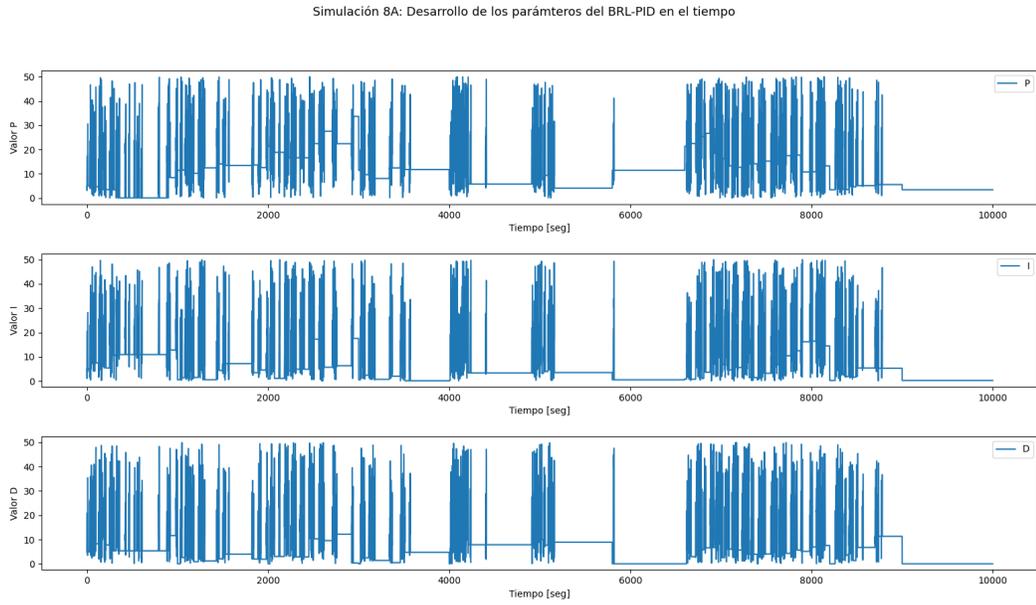


Figura 5.28: Simulación 8A: Dinámica de los parámetros del BRL-PID. Referencia sucesión de escalones, con ruidos en la planta y con valores en la memoria al inicio de la simulación.

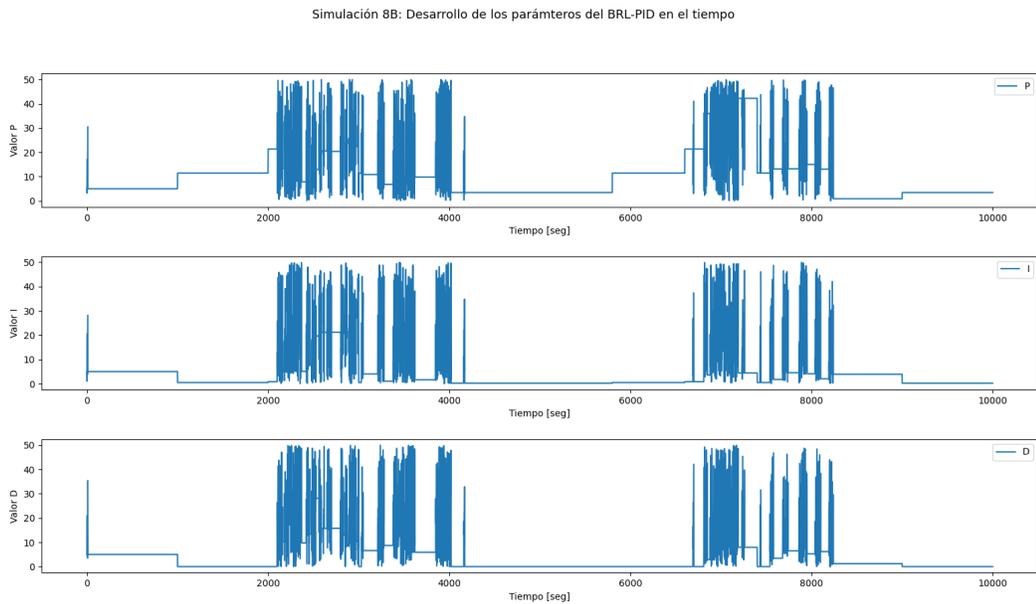


Figura 5.29: Simulación 8B: Dinámica de los parámetros del BRL-PID. Referencia sucesión de escalones, con ruidos en la planta y con valores en la memoria al inicio de la simulación.

5.5.9. Simulaciones 9: Referencia Módulo de Sinusoidal, Sin Ruido, Sin Memoria

Las simulaciones 9A y 9B contemplan una referencia módulo de sinusoidal, planta sin ruido y controlador BRL-PID (para las referencias de este tipo no se hicieron pruebas con memoria ya que la referencia no se comporta como escalón, y se verificó que no se cumplen las condiciones de guardado). Se observa que en ambos casos los resultados son mejores que los del PID clásico, y el esfuerzo de control es casi el mismo, lo cual es coherente con la dinámica de los parámetros en el tiempo (Figuras 5.32 y 5.33). En este caso además se observa que las métricas obtenidas son las mismas para los casos A y B, mejor dicho con diferencias mucho menores que el resto de los casos, lo cual se atribuye al hecho que en ambos casos la exploración consigue un buen resultado desde el inicio, y no requiere de una exploración en el resto de la simulación.

Simulación	Descripción	$\varphi(IAE)$	$\varphi(ISE)$	$\varphi(ITAE)$	$\varphi(ITSE)$	$\varphi(ISI)$
9A	Módulo de Sinusoidal, Sin Ruido, Sin Memoria	0,0452	0,0323	0,0333	0,0031	1,0335
9B	Módulo de Sinusoidal, Sin Ruido, Sin Memoria	0,0452	0,0323	0,0333	0,0031	1,0335

Tabla 5.12: Resultados obtenidos en las simulaciones 9A y 9B: Referencia módulo de sinusoidal, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.

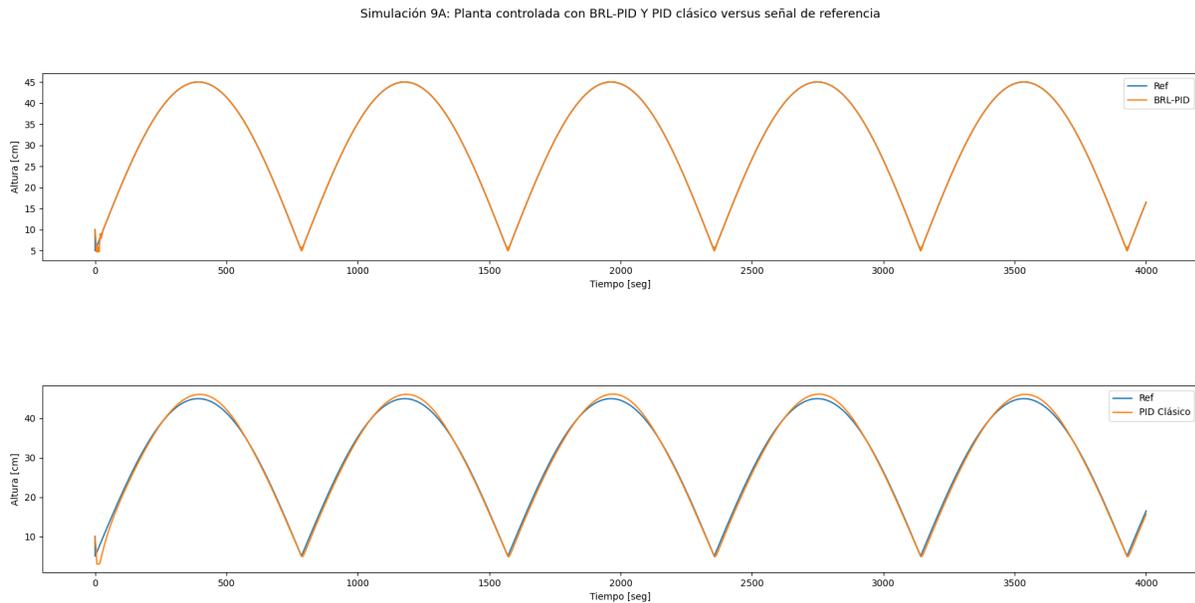


Figura 5.30: Simulación 9A: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia módulo de sinusoidal, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.

Simulación 9B: Planta controlada con BRL-PID Y PID clásico versus señal de referencia

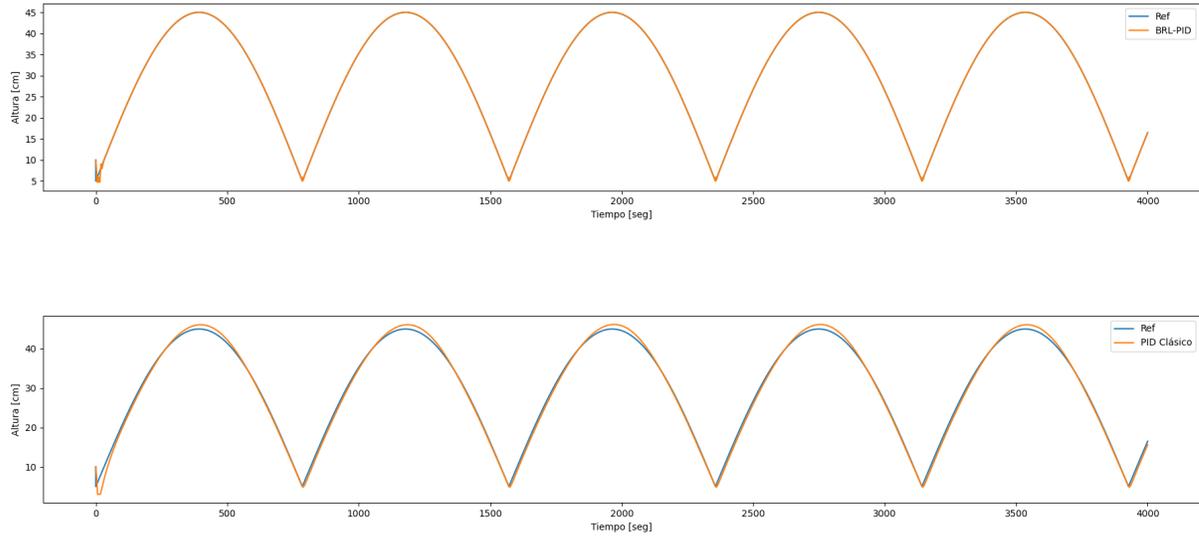


Figura 5.31: Simulación 9B: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia módulo de sinusoidal, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.

Simulación 9A: Desarrollo de los parámetros del BRL-PID en el tiempo

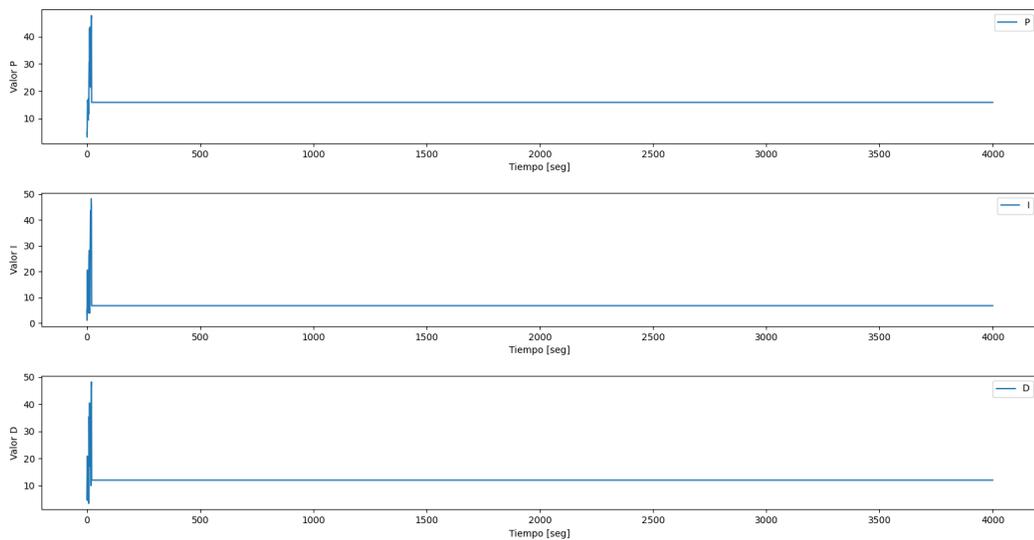


Figura 5.32: Simulación 9A: Dinámica de los parámetros del BRL-PID. Referencia módulo de sinusoidal, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.

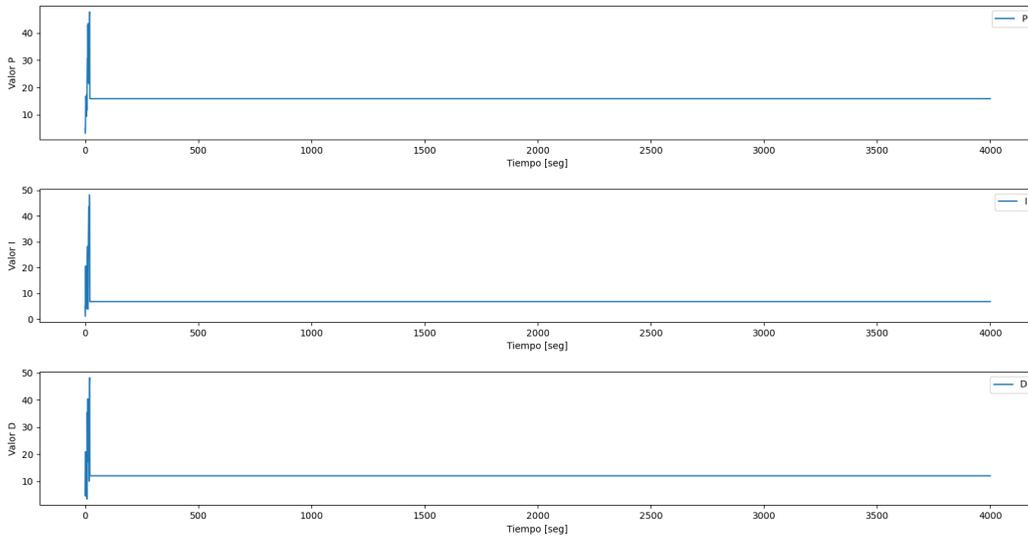


Figura 5.33: Simulación 9B: Dinámica de los parámetros del BRL-PID. Referencia módulo de sinusoidal, sin ruidos en la planta y sin valores en la memoria al inicio de la simulación.

5.5.10. Simulaciones 10: Referencia Módulo de Sinusoidal, Con Ruido, Sin Memoria

Para las simulaciones 10A y 10B se añadió ruido a la planta, observando un mejor desempeño del BRL-PID al relajar las condiciones de optimalidad, lo cual se atribuye, al igual que en las simulaciones 9A y 9B, a que se logra un óptimo desde el principio que resulta mejor que el del PID clásico, salvo que el ruido produjo más momentos de exploración. En parte también se atribuye el mejor resultado a que la planta controlada con el PID clásico sufre una mayor inestabilidad alrededor de los 3000[s], lo que se atribuye a la presencia del ruido, ante lo cual el BRL-PID inició procesos de exploración que fueron beneficiosos para mitigar sus efectos.

Simulación	Descripción	$\varphi(IAE)$	$\varphi(ISE)$	$\varphi(ITAE)$	$\varphi(ITSE)$	$\varphi(ISI)$
10A	Módulo de Sinusoidal, Con Ruido, Sin Memoria	1,6785	2,3978	1,6800	2,5831	28.910,6006
10B	Módulo de Sinusoidal, Con Ruido, Sin Memoria	0,8933	0,6733	0,7901	0,5045	653,3582

Tabla 5.13: Resultados obtenidos en las simulaciones 10A y 10B: Referencia módulo de sinusoidal, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.

Simulación 10A: Planta controlada con BRL-PID Y PID clásico versus señal de referencia

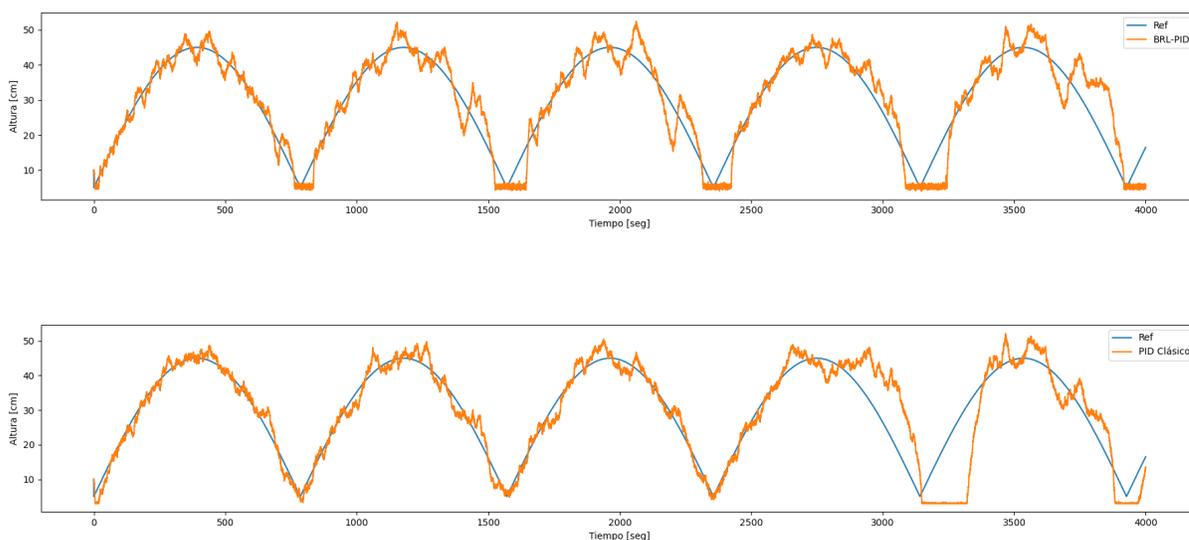


Figura 5.34: Simulación 10A: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia módulo de sinusoidal, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.

Simulación 10B: Planta controlada con BRL-PID Y PID clásico versus señal de referencia

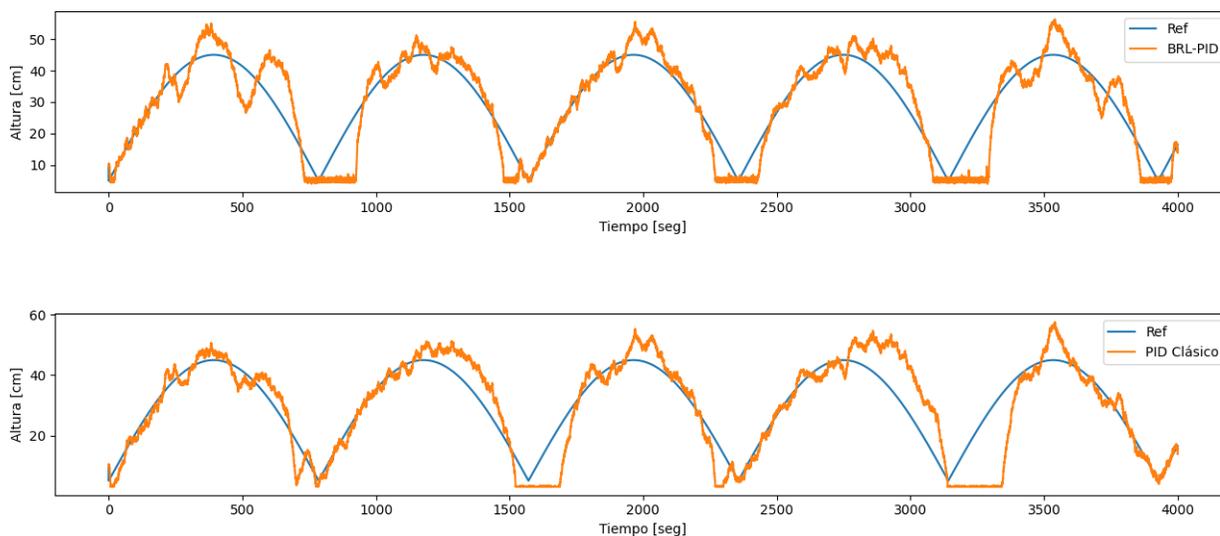


Figura 5.35: Simulación 10B: Respuesta de la planta ante una referencia de escalón controlada por el BRL-PID (arriba) y el PID clásico (abajo). Referencia módulo de sinusoidal, con ruidos en la planta y sin valores en la memoria al inicio de la simulación.

Capítulo 6

Discusión

De la información presentada se deduce que el BRL-PID es capaz de controlar la planta de forma similar al PID clásico que cuenta con más información, si bien el esfuerzo de control es considerablemente mayor en casi todos los casos excepto los 8A y 8B, y que el rendimiento está mayormente influenciado por la adecuada configuración de los parámetros γ , τ_{mem} , $\sigma(e_{obj})$ y la presencia de ruidos en la planta, por cuanto estos últimos generan oscilaciones y respuestas no deseadas que llevan al BRL-PID a situaciones de exploración de parámetros del controlador innecesarias, causando problemas de rendimiento. Además, observando la Tabla 5.3 se observa que las métricas IAE e ITAE se cumplen en más casos que las ISE e ITSE, lo que indica que el sistema tiene más problemas con las oscilaciones alrededor del valor objetivo que con estar cercano a éste, lo que se refuerza al revisar los casos A y B más lo mencionado anteriormente sobre la influencia del ruido en las evaluaciones.

Con respecto a la acción de emergencia, su uso ayuda a obtener mejores resultados con la respuesta escogida en el diseño, lo cual sugiere que idealmente se debe tener cierto grado de conocimiento de la fenomenología de la planta para escoger acciones más eficientes para los casos críticos. En cuanto al sistema de memoria, también queda demostrada su utilidad, si bien para que sea práctica se debe tener conocimiento experto previo.

Sobre la modificación del parámetro γ que representa la importancia si el error es en promedio cero o si las oscilaciones de éste con respecto al cero son altas o bajas, una corrección adecuada permite que el sistema reaccione mejor ante la presencia de ruido ya que éste genera oscilaciones que no dependen del parámetro implantado en el BRL-PID, por lo que si se da mayor importancia en la evaluación a este fenómeno se castigan dichos parámetros de manera incorrecta. En cuanto a los valores τ_{mem} y $\sigma(e_{obj})$, relacionados a las condiciones de optimalidad, se observa que al bajar el primero y subir el segundo, es decir, reducir la exigencia del estado óptimo, permite en general mejorar los resultados sobre todo en presencia de ruido. Esto se asocia a que el ruido genera una baja en los valores de evaluación promedio, luego si la condición de optimalidad es muy exigente, puede ocurrir que nunca se consiga o que se logre en pocas ocasiones, forzando procesos de exploración que no son convenientes para el control. En resumen, la correcta calibración de estos tres parámetros es determinante en el rendimiento del BRL-PID.

En cuanto a las referencias y el filtro, **se observa una mejor respuesta en curvas sinusoidales sin ruido por cuanto los cambios graduales en la referencia son más fáciles de abordar para el BRL-PID.** Esto se deduce a partir de las simulaciones presentadas y otras realizadas durante la investigación en las que **se observó que durante la respuesta transiente, en el caso en que no se tenga el filtro de la referencia, se generan evaluaciones muy negativas de los parámetros del controlador siendo que ese mal resultado no depende directamente de éstos,** confirmando entonces la utilidad de la curva de amortiguación que genera dicho filtro mientras sea lo suficientemente lenta para que la planta pueda seguirla, ya que también se observó en iteraciones previas que rutinas de amortiguación demasiado rápidas no eran posibles de seguir por la planta, y que curvas demasiado lentas reducen el rendimiento con respecto a las métricas propuestas ya que si bien el error de identificación es bajo y la planta sigue la referencia filtrada adecuadamente, la métrica considera el error de control para la evaluación.

Empíricamente se detectó que pocas muestras por distribución, del orden de 10 a 50, genera mejores resultados que obtener muchas muestras y luego actualizar, produciéndose un **compromiso entre la cantidad de muestras para exploración y capacidad de adaptación del algoritmo,** lo cual es relevante si se considera además la dinámica de la planta. Evidentemente esto también depende de la fenomenología de la misma.

La degradación de parámetros, al menos como fue programado, tiene un menor impacto que la presencia de ruido en el rendimiento, el que genera un efecto similar a las no linealidades de salto, quedando pendiente profundizar en los perfiles de degradación propios del fenómeno.

Sobre la dinámica del estanque cónico, su respuesta varía dependiendo si está llenando o vaciando, lo cual se compensa parcialmente con el sistema de memoria, sobre todo en casos como la sucesión de escalones en los que se retorna a valores anteriores, si bien ocurre que en algunos casos, dada la no linealidad de la planta, al volver al nivel se generan oscilaciones relevantes que hacen entrar al BRL-PID en un estado de exploración que no siempre es necesario o más conveniente.

Capítulo 7

Conclusiones y Trabajo Futuro

7.1. Conclusiones

La motivación de esta investigación fue explorar soluciones de control adaptable para casos donde hay casi nula información de la planta y se requiera una baja capacidad de cómputo, para lo cual se propuso plantear el problema de control a partir de una situación lo más genérica posible y con mínimas condiciones sobre la planta, procurando agregar detalladamente las variables del problema con el fin de tener disponible un marco de trabajo que facilitara la modificación de éstos y generar aprendizaje empírico sobre el problema. Además, esta aproximación permite reemplazar dichas variables por valores deterministas a medida que se consigue mayor conocimiento del problema, o se acota a un caso específico.

En cuanto al problema del cómputo, se propuso una solución basada en acotar la cantidad de valores posibles que pueden tomar los parámetros asumiendo la cuantización conocida de los valores, basándose en que en un caso real la frecuencia de muestreo y la tasa de bits siempre impondrán esta condición al trabajar con sistemas digitales. Los parámetros α , pensados originalmente como múltiplos de la tasa de bits, permiten modificar la cantidad de valores posibles en caso de requerir mayor rapidez de cómputo, si bien esto puede implicar un eventual deterioro del desempeño si la resolución es muy baja. Los valores posibles a su vez fomentan en el algoritmo el trabajo entre vectores en operaciones de suma y producto término a término, lo cual es más sencillo de programar y almacenar. Además, dado que las dimensiones de dichos vectores quedan fijos una vez configurado el sistema, en un caso real es más fácil predecir el consumo de espacio y cómputo. Valores más complejos podrían estar almacenados en una LUT.

Se deduce del trabajo empírico que es posible generar una calibración estándar para el sistema y que en casos con ruido hay una relación directa entre el rendimiento y el valor del parámetro γ , e inversa con respecto a $\sigma(e_{obj})$ e inversa con τ_{mem} , lo cual es material de futuras investigaciones. La necesidad de tiempo para conseguir las muestras produce un bajo desempeño ante no linealidades de salto en la señal de referencia, como es el caso del escalón, lo cual pudo ser mitigado con el sistema de memoria y la señal de amortiguación del filtro de la referencia. Las rutinas de emergencia resultaron ser un apoyo en los rangos extremos de la planta ante casos de inestabilidad por efecto de las no linealidades de la planta.

Del desarrollo expuesto se concluye que el filtro cumple su función de apoyar ante la presencia de no linealidades de salto, suavizando la referencia y evitando así la mala evaluación que obtendrán los parámetros durante la respuesta transiente, motivando así exploraciones necesarias. Con respecto a los escalones, el sistema de detección presenta un problema que no es posible evadir en tiempo real en esta propuesta, y es que no se puede distinguir al momento de detectar un salto si se trata de un escalón o no, lo cual es un problema si es un impulso o una señal similar como una perturbación puntual. Aún así el criterio de similitud con el escalón es una herramienta útil para optimizar el funcionamiento del BRL-PID.

Otro punto importante de abordar es el consumo energético: si bien esta propuesta requiere de una baja cantidad de procesos versus otras opciones presentadas en el estado del arte, se observa un mayor esfuerzo de control por cuanto se debe explorar para encontrar la combinación de parámetros que consigue el desempeño deseado, por lo que este punto deberá ser mejorado si se requiere menor gasto, mencionando además no se discute en esta investigación las consecuencias de esto en el desgaste de los componentes físicos. La modificación de los parámetros γ , τ_{mem} y $\sigma(e_{obj})$ se pueden utilizar también como herramienta para reducir la exploración, observando que el rendimiento no se vea comprometido.

En cuanto a la estabilidad del sistema, en comparación a las últimas investigaciones este problema es particularmente complejo al proponer la no identificación de modelo de planta, carencia de información y las referencias utilizadas, lo cual no permite asegurar estabilidad por medios convencionales como los criterios de Lyapunov. Considerando lo anterior, una posible mejora es añadir sistemas de identificación que a medida que se disponga de datos de la planta generen modelos y realicen modificaciones en los parámetros del BRL-PID o genera un cambio de controlador calibrado con la información obtenida.

Lo anteriormente mencionado permite concluir que este sistema implica un alto riesgo de uso en un caso real, si bien es capaz de resolver el control en la mayoría de los casos, ya que no se puede asegurar su estabilidad sin más información de la planta o acotando las condiciones de uso. Además, se debe considerar que la degradación de parámetros en un caso real puede repercutir en la utilidad del sistema de memoria: en un caso extremo puede ocurrir que haya transcurrido un tiempo tal desde la última vez que se estuvo en un estado almacenado, que por efecto de la degradación de la planta sea más conveniente dejar al algoritmo buscar un nuevo óptimo, lo cual es un tema a indagar.

Es esperable que esta propuesta no sea eficiente para casos MIMO o con más parámetros en el controlador, tanto por el tiempo de cómputo como por el hecho que podría cobrar mayor relevancia las dependencias de las variables aleatorias en el desarrollo de la solución, y que, en el caso de los parámetros del controlador, la selección por bloque sea cada vez menos representativa del total de combinaciones posibles, lo cual podría requerir ventanas de tiempo más largas, comprometiendo así los tiempos de respuesta.

Para finalizar, el BRL-PID es capaz de responder adecuadamente a una serie de casos de alta incertidumbre, es moldeable y permite su adecuación en presencia de nuevo conocimiento de la planta, si bien en algunos casos no obtiene métricas cercanas al PID clásico, logra un grado de control que, considerando la incertidumbre que enfrenta, se considera un resultado favorable, quedando pendiente cómo profundizar para casos específicos o con soluciones de mayor requerimientos de cómputo.

7.2. Trabajo Futuro

Sobre el aprendizaje de la dinámica de la planta se considera necesario añadir un sistema de identificación paralelo al BRL-PID que permita modelar la planta y mejorar su aproximación en el tiempo, aportando nueva información al algoritmo de decisión y evaluación de las combinaciones de parámetros del BRL-PID. Teniendo en cuenta estos sistemas suelen implicar alta capacidad de cómputo, si se operan de forma paralela no se pierde la capacidad de controlar en tiempo real ya que el BRL-PID seguirá funcionando mientras la identificación ocurre o se actualiza. En este ámbito también se puede considerar la generación de perfiles de ruido y las relaciones de dependencia entre los parámetros y el rendimiento, para lo cual se propone la utilización de la entropía de Shannon para evaluar la calidad de la información obtenida y según el resultado alimentar un sistema de redes neuronales u otro de aprendizaje para obtener resultados.

Otra línea de investigación posible es profundizar teóricamente en la controlabilidad y estabilidad para casos conocidos, es decir, abordar casos clásicos o familias de plantas y evaluar teóricamente si la propuesta es controlable o no y bajo qué requisitos, lo cual complementado con los puntos anteriores puede ayudar a determinar el mínimo de elementos a añadir o modificar para lograr un control exitoso.

La expansión de la propuesta a sistemas MIMO y controladores con mayor cantidad de parámetros también es una línea de investigación posible. Como se mencionó en las conclusiones, sistemas con mayor cantidad de entradas, salidas o parámetros del controlador aumentarían el requisito de cómputo y por ende podrían comprometer los tiempos de respuesta. Además, es posible que la dependencia de las variables aleatorias sea más relevante en esos casos, sumado a que la cantidad de combinaciones posibles aumenta (haciendo menos representativa una muestra, lo cual acentúa el conflicto del tamaño de bloques y ventanas versus los tiempos de respuesta del sistema). Ante esto la solución propuesta es la mencionada en el párrafo anterior (sistemas de identificación y caracterización de estas propiedades) e incorporar al BRL-PID la evaluación de la dependencia de las variables dependiendo del resultado de control.

El esfuerzo de control es un punto débil en esta propuesta. El sistema requiere explorar para encontrar los puntos que cumplen con el desempeño, y en un caso real muy probablemente esto implica un mayor gasto y eventual deterioro acelerado de los componentes físicos del sistema, en particular los actuadores y la planta misma. Una posible mejora es condicionar las acciones de control de manera que el desgaste sea un factor de decisión, si bien esto puede entrar en conflicto con el tiempo para llegar a valores óptimos de los parámetros del controlador, o asumir mayor conocimiento de la dinámica de la planta y mejorar el algoritmo de selección de parámetros.

Sobre la memoria y el tiempo de degradación de la planta, como se mencionó en las conclusiones, existe el problema que si ha pasado un tiempo tal que el valor almacenado deja de ser óptimo por el desgaste de la planta, podría convenir más que el BRL-PID aprenda un nuevo óptimo, por lo que una solución sencilla podría ser que se borren los valores luego de cierta cantidad de tiempo.

De las simulaciones se infiere que existe una correlación entre las variables γ , τ_{mem} y $\sigma(e_{obj})$ y el rendimiento, y que son muy importantes en la obtención de buenos resultados, ante lo cual se propone validar este resultado empírico y generar un método automático de adaptación considerando la naturaleza del ruido y la dinámica de la planta con respecto a los resultado de las evaluaciones τ y el promedio y ECM del error de control.

Bibliografía

- [1] T. Shen, X. Qiao, Y. Dong, Y. Wang, W. Zhang, and Y. Yuan, “Deep adaptive control with online identification for industrial robots,” *Science China Technological Sciences*, vol. 65, no. 11, pp. 2593–2604, 2022.
- [2] J. P. Folch, S. Zhang, R. M. Lee, B. Shafei, D. Walz, C. Tsay, M. van der Wilk, and R. Misener, “Snake: Bayesian optimization with pathwise exploration,” *arXiv preprint arXiv:2202.00060*, 2022.
- [3] A. Guha and A. Annaswamy, “Mrac-rl: A framework for on-line policy adaptation under parametric model uncertainty,” 2020.
- [4] S. Nagammai, S. Latha, and M. Varatharajan, “Soft computing-based fuzzy integral sliding mode control: a real-time investigation on a conical tank process,” *Soft Computing*, vol. 24, 09 2020.
- [5] S. Nagammai, S. Latha, D. Pradeepkannan, A. Umarani, and S. Balamurugan, “Study of different control strategies applied to a second-order nonlinear tank process,” in *Proceedings of Fourth International Conference on Communication, Computing and Electronics Systems* (V. Bindhu, J. M. R. S. Tavares, and C. Vuppapapati, eds.), (Singapore), pp. 1–13, Springer Nature Singapore, 2023.
- [6] S. K. Vavilala, V. Thirumavalavan, and C. K., “Level control of a conical tank using the fractional order controller,” *Computers Electrical Engineering*, vol. 87, p. 106690, 2020.
- [7] J. G. Ziegler, N. B. Nichols, *et al.*, “Optimum settings for automatic controllers,” *Trans. ASME*, vol. 64, no. 11, 1942.
- [8] K. Ogata *et al.*, *Modern control engineering*, vol. 5. Prentice hall Upper Saddle River, NJ, 2010.
- [9] K. Ogata, *Sistemas de control en tiempo discreto*. Prentice Hall, second ed., 1996.
- [10] S. Haykin and B. Van Veen, *Señales y sistemas*. Limusa, Wiley, first ed., 2001.
- [11] R. Cárdenas, “Apuntes de fundamentos de control EL-4113,” Agosto 2022.
- [12] C. Jauregui, M. Duarte Mermoud, G. Lefranc, R. Orostica, J. C. Travieso Torres, and O. Beytia, “Conical tank level control with fractional PID,” *IEEE Latin Ameri-*

- ca Transactions*, vol. 14, no. 6, pp. 2598–2604, 2016.
- [13] K. Narendra and A. Annaswamy, *Stable adaptive systems*. Dover Publications, first ed., 2005.
- [14] E. Camacho and M. Arahall, “Neural network based adaptive control,” *Annual Review in Automatic Programming*, vol. 19, pp. 13–24, 1994.
- [15] F. Shahraki, M. Fanaei, and A. Arjomandzadeh, “Adaptive system control with pid neural networks,” *Chemical Engineering transaction*, vol. 17, pp. 1395–1401, 2009.
- [16] F. Osorio Arteaga, “Descripción de la técnica de control por planos deslizantes y su diseño en sistemas multivariables,” 2018.
- [17] S. J. Russell, *Artificial intelligence, a modern approach*. Pearson Education, Inc., 2010.
- [18] C. A. Jáuregui Salinas, “Evaluación de estrategias de sintonización de controladores fraccionarios para planta no lineal: sistema de estanques,” 2016.
- [19] B. Hajek, *Random processes for engineers*. Cambridge University Press, 2015.
- [20] J. V. Candy, *Bayesian signal processing: Classical, modern, and particle filtering methods*. John Wiley & Sons, 2016.
- [21] S. Bennett, “A brief history of automatic control,” *IEEE Control Systems Magazine*, vol. 16, no. 3, pp. 17–25, 1996.
- [22] D. Saez, “Modelación difusa (apuntes EL7012 - Control Inteligente de Sistemas),” Agosto 2016.
- [23] M. Reynolds, “When to use bayesian,” Agosto 2022.
- [24] P. A. Fishwick, *Handbook of dynamic system modeling*. CRC Press, 2007.
- [25] M. D. Mermoud, “Introducción a la modelación y el análisis de sistemas,” Junio 2014.
- [26] H. Khalil, *Nonlinear systems*. Prentice Hall, second ed., 2002.
- [27] L. Ljung and T. Glad, *Modeling of dynamic systems*. Prentice-Hall, Inc., 1994.
- [28] J.-J. E. Slotine, W. Li, *et al.*, *Applied nonlinear control*, vol. 199. Prentice hall Englewood Cliffs, NJ, 1991.
- [29] M. Duarte, “Estabilidad de sistemas dinámicos,” 1980-1990.
- [30] W. C. Schultz and V. C. Rideout, “Control system performance measures: Past, present, and future,” *IRE transactions on automatic control*, vol. AC-6, no. 1, pp. 22–35, 1961.
- [31] K. Astrom, “Adaptive feedback control,” *Proceedings of the IEEE*, vol. 75, no. 2, pp. 185–217, 1987.

- [32] A. Abdulameer, M. Sulaiman, M. S. M. Aras, and D. Saleem, "Tuning methods of pid controller for dc motor speed control," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 3, no. 2, pp. 343–349, 2016.
- [33] J. C. Travieso, "Apuntes de control adaptativo," Junio 2020.
- [34] K. Pirabakaran and V. Becerra, "Automatic tuning of pid controllers using model reference adaptive control techniques," in *IECON'01. 27th Annual Conference of the IEEE Industrial Electronics Society (Cat. No.37243)*, vol. 1, pp. 736–740 vol.1, 2001.
- [35] I. Rodríguez Villar, "Evaluación de distintas arquitecturas de redes neuronales aplicadas a la clasificación de datos tabulares," 2022.
- [36] L. Moreno-Díaz-Alejo, "Análisis comparativo de arquitecturas de redes neuronales para la clasificación de imágenes," Master's thesis, 2020.
- [37] "Chapter 3 stochastic processes and their classifications," in *Random Differential Equations in Science and Engineering* (T. Soong, ed.), vol. 103 of *Mathematics in Science and Engineering*, pp. 33–71, Elsevier, 1973.
- [38] P. Kokotovic, "The joy of feedback: nonlinear and adaptive," *IEEE Control Systems Magazine*, vol. 12, no. 3, pp. 7–17, 1992.
- [39] M. E. Ortiz-Quisbert, M. A. Duarte-Mermoud, F. Milla, R. Castro-Linares, and G. Lefranc, "Optimal fractional order adaptive controllers for avr applications," *Electrical Engineering*, vol. 100, pp. 267–283, 2018.
- [40] Z. Liu, H. Jahanshahi, J. Gómez-Aguilar, G. Fernandez-Anaya, J. Torres-Jiménez, A. A. Aly, and A. M. Aljuaid, "Fuzzy adaptive control technique for a new fractional-order supply chain system," *Physica Scripta*, vol. 96, no. 12, p. 124017, 2021.
- [41] S. Kumar, A. E. Matouk, H. Chaudhary, and S. Kant, "Control and synchronization of fractional-order chaotic satellite systems using feedback and adaptive control techniques," *International Journal of Adaptive Control and Signal Processing*, vol. 35, no. 4, pp. 484–497, 2021.
- [42] J. L. Doob, *Stochastic processes*, vol. 7. Wiley New York, 1953.
- [43] W. K. T. Cho and Y. Y. Liu, "Sampling from complicated and unknown distributions: Monte carlo and markov chain monte carlo methods for redistricting," *Physica A: Statistical Mechanics and its applications*, vol. 506, pp. 170–178, 2018.
- [44] I. M. Sobol, *A primer for the Monte Carlo method*. CRC press, 2018.
- [45] E. Lehmann and G. Casella, *Theory of point estimation*. Berlin: Springer, first ed., 1998.
- [46] M. E. Orchard and G. J. Vachtsevanos, "A particle-filtering approach for on-line fault diagnosis and failure prognosis," *Transactions of the Institute of Measurement and Control*, vol. 31, p. 221–246, 2009.

- [47] A. Doucet, N. De Freitas, and N. Gordon, “An introduction to sequential monte carlo methods,” in *Sequential Monte Carlo methods in practice*, pp. 3–14, Springer, 2001.
- [48] D. J. C. Mackay, “Introduction to monte carlo methods,” in *Learning in graphical models*, pp. 175–204, Springer, 1998.
- [49] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, “An introduction to mcmc for machine learning,” *Machine learning*, vol. 50, no. 1, pp. 5–43, 2003.
- [50] C. P. Robert and W. Changye, “Markov chain monte carlo methods, a survey with some frequent misunderstandings,” *arXiv preprint arXiv:2001.06249*, 2020.
- [51] S. Thrun, “Particle filters in robotics,” in *UAI*, vol. 2, pp. 511–518, Citeseer, 2002.
- [52] S. T. Tokdar and R. E. Kass, “Importance sampling: a review,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 1, pp. 54–60, 2010.
- [53] J. McCarthy, “What is artificial intelligence?,” 2007.
- [54] C. Grosan and A. Abraham, “Intelligent systems,” *Intelligent systems reference library*, 2011.
- [55] S. Suthaharan, *Decision Tree Learning*, pp. 237–269. Boston, MA: Springer US, 2016.
- [56] S. Haykin, *Neural networks and learning machines*. Pearson Education, 2009.
- [57] B. Polyak, *Introduction to optimization*. 07 2020.
- [58] M. Viswanathan, “Linear models - least squares estimator (lse),” Nov 2020.
- [59] R. Bassett and J. Deride, “Maximum a posteriori estimators as a limit of bayes estimators,” *Mathematical programming*, vol. 174, no. 1-2, p. 129–144, 2018.
- [60] S. Maldonado, A. Flores, and R. Weber, “Selección de atributos y support vector machines adaptado al problema de fuga de clientes,” *Revista Ingeniería de Sistemas*, vol. 24, pp. 85–107, 09 2015.
- [61] C. M. Bishop, “Pattern recognition and machine learning (information science and statistics),” 2007.
- [62] Y. Baştanlar and M. Özuysal, *Introduction to Machine Learning*, pp. 105–128. Totowa, NJ: Humana Press, 2014.
- [63] B. Settles, “Active learning literature survey,” Computer sciences technical report 1648, University of Wisconsin-Madison, 2009.
- [64] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [65] M. P. Deisenroth, G. Neumann, and J. Peters, “A survey on policy search for robotics,”

Foundations and Trends in Robotics, vol. 2, no. 1–2, p. 1–142, 2011.

- [66] R. Dearden, N. Friedman, and S. Russell, “Bayesian Q-Learning,” *Aaai/iaai*, vol. 1998, pp. 761–768, 1998.
- [67] J. L. Carrera Villacrés, Z. Zhao, T. Braun, and Z. Li, “A particle filter-based reinforcement learning approach for reliable wireless indoor positioning,” *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2457–2473, 2019.
- [68] L. Meng, R. Gorbet, and D. Kulić, “Memory-based deep reinforcement learning for pomdps,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5619–5626, IEEE, 2021.
- [69] M. Orchard, “Introduction to phm,” 2020.
- [70] O. Kallenberg, “Foundations of modern probability,” 1997.
- [71] R. B. Ash, *Basic probability theory*. Courier Corporation, 2008.
- [72] J. K. Blitzstein and J. Hwang, *Introduction to probability*. Crc Press Boca Raton, FL, 2015.
- [73] H. Pishro-Nik, “Introduction to probability, statistics, and random processes,” 2016.
- [74] A. Jøsang, “Generalising Bayes’ theorem in subjective logic,” in *MFI*, pp. 462–469, 2016.

Anexo

Probabilidades: Definiciones Fundamentales

La Probabilidad es una rama de la teoría de la medida, la cual estudia fenómenos aleatorios o con algún grado de incertidumbre [70] [71]. Para el trabajo con procesos aleatorios, se presenta un marco teórico de probabilidades basado en los axiomas de probabilidad de Kolmogorov [19].

Se define entonces el **espacio muestral** Ω como el conjunto de todos los posibles resultados de un experimento aleatorio; cualquier subconjunto de éste se denomina **realización** o resultado (ω). El conjunto \mathcal{F} es un subconjunto de subconjuntos de Ω ; cada elemento de \mathcal{F} se denomina **evento**. Se asume que es una σ -álgebra, lo que implica que se satisfacen los siguientes axiomas [19]:

1. $\Omega \in \mathcal{F}$.
2. Si $A \in \mathcal{F} \Rightarrow A^C \in \mathcal{F}$.
3. Si $A, B \in \mathcal{F}, \Rightarrow A \cup B \in \mathcal{F}$. Además, si $\{A_1, A_2, \dots\}$ es una secuencia de elementos de \mathcal{F} , $\bigcup_{i=1}^{\infty} A_i \in \mathcal{F}$

La intersección de secuencias también satisface la pertenencia al conjunto \mathcal{F} ; si la intersección entre cualquier par de componentes de la secuencia $\{A_1, A_2, \dots\}$ es vacío, entonces son eventos mutuamente excluyentes [19].

Un **espacio de probabilidad** \mathbb{P} en \mathcal{F} es una función que cumple [19]:

1. $\mathbb{P} \geq 0, \forall A \in \mathcal{F}$.
2. Si $A, B \in \mathcal{F}$ y son mutuamente excluyentes, $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B)$. Si se tiene una secuencia infinita de eventos mutuamente excluyentes $\{A_1, A_2, \dots\}, \Rightarrow \mathbb{P}(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \mathbb{P}(A_i)$.
3. $\mathbb{P}(\Omega) = 1$

La tripleta $(\Omega, \mathcal{A}, \mathbb{P})$ conforma un **espacio de probabilidad**, y la relación entre sus partes genera una estructura matemática de la que derivan propiedades y lemas que permiten elaboraciones superiores, como modelos de procesos aleatorios [19].

Dos o más eventos $\{A_1, A_2, \dots, A_n\}$ son independientes si se cumple que la probabilidad de cualquier selección de éstos es el producto de las probabilidades individuales. Por ejemplo, si

$n = 3$ debe ocurrir que [19]:

$$\mathbb{P}(A_1 \cap A_2) = \mathbb{P}(A_1)\mathbb{P}(A_2) \quad (7.1)$$

$$\mathbb{P}(A_1 \cap A_3) = \mathbb{P}(A_1)\mathbb{P}(A_3) \quad (7.2)$$

$$\mathbb{P}(A_2 \cap A_3) = \mathbb{P}(A_2)\mathbb{P}(A_3) \quad (7.3)$$

$$\mathbb{P}(A_1 \cap A_2 \cap A_3) = \mathbb{P}(A_1)\mathbb{P}(A_2)\mathbb{P}(A_3) \quad (7.4)$$

La probabilidad condicional para dos eventos A Y B con probabilidad no nula representa la posibilidad que ocurra A dado B y viene dada por [19]:

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)} \quad (7.5)$$

Se define una **partición** de Ω como una colección de eventos $E_i, i \in \mathbb{N}, i \leq k$ mutuamente excluyentes tales que $\Omega = \bigcup_{i=1}^k E_i$. Se deduce entonces que $\sum_{i=1}^k \mathbb{P}(E_i) = 1$, y la ley de probabilidades totales se mantiene porque A es la unión de los eventos mutuamente excluyentes $A \cap E_1, \dots, A \cap E_k$ y si cumple [19]:

$$\mathbb{P}(A) = \mathbb{P}(A \cap E_1) + \dots + \mathbb{P}(A \cap E_k) \quad (7.6)$$

Si además $\mathbb{P}(E_i) \neq 0, \forall i \in \mathbb{N}, i \leq k$, 7.6 se puede reescribir como [19]:

$$\mathbb{P}(A) = \sum_{i=1}^k \mathbb{P}(A|E_i)\mathbb{P}(E_i) \quad (7.7)$$

La manipulación de los conceptos de probabilidad condicional y la ley de probabilidades totales permite generar la **fórmula de Bayes** para $\mathbb{P}(E_i|A), \mathbb{P}(A) \neq 0$ en sus formas simple y expandida, respectivamente [19]:

$$\mathbb{P}(E_i|A) = \frac{\mathbb{P}(A|E_i)}{\mathbb{P}(A)} \quad (7.8)$$

$$\mathbb{P}(E_i|A) = \frac{\mathbb{P}(A|E_i)}{\sum_{i=1}^k \mathbb{P}(A|E_i)\mathbb{P}(E_i)} \quad (7.9)$$

Finalmente, se define un experimento aleatorio como una serie de pruebas que permiten conocer el comportamiento de un fenómeno determinado, y una realización es una iteración del experimento del que se obtiene un resultado determinado [72].

Para representar un proceso estocástico es fundamental presentar las variables y funciones aleatorias.

Las variables aleatorias son funciones medibles, reales y definidas en un espacio de probabilidad $(\Omega, \mathcal{F}, \mathbb{P})$, $X : \Omega \rightarrow \mathcal{R}$, medible en \mathcal{F} , lo que significa que para cualquier real $c \in \mathbb{R}$, $\{\omega : X(\omega) \leq c\} \in \mathcal{F}$. Si Ω es finito o contable finito, entonces \mathcal{F} puede ser el conjunto de todos los subconjuntos de Ω , en cuyo caso cualquier función real de Ω será una variable aleatoria [19] [42].

La **distribución de probabilidad** de una variable aleatoria es una función que asigna a cada evento definido en la variable un valor de probabilidad. Las hay paramétricas, es decir, aquellas que se quedan caracterizadas mediante la definición de parámetros determinados, (gaussianas, Bernoulli y geométricas, por ejemplo), y no paramétricas. La distribución de una variable aleatoria discreta se denomina ***probability mass function (PMF)***, y de una continua ***probability density function (PDF)***. La distribución de una variable aleatoria puede depender del valor que asume otra en un contexto dado, en cuyo caso la distribución de la primera será **dependiente** de la segunda [19] [20] [42] [73].

Se define la función real de **distribución acumulada** (CDF por su nombre en inglés), como [19]:

$$F_X(c) = \mathbb{P}(\omega : X(\omega) \leq c) \quad (7.10)$$

Se puede encontrar escrita de forma corta en la literatura como $\mathbb{P}(X \leq c)$ [19]. La distribución acumulada cumple que [19]:

1. Es no decreciente.
2. $\lim_{x \rightarrow \infty} = 1$ y $\lim_{x \rightarrow -\infty} = 0$.
3. F es continua por la derecha.

Una variable aleatoria puede ser discreta, continua o mixta, y puede ser acotada (su recorrido en este contexto se denomina **rango**). Una o más variables aleatorias se denominan **dependientes** si el resultado de una depende de las otras [19] [73].

Una función aleatoria se caracteriza por contener variables aleatorias, lo cual además la convierte en una variable aleatoria [19] [73][45].

Dos funciones aleatorias relevantes son la **esperanza** μ (o valor esperado, *mean* en inglés) y la **varianza** σ , ya que entregan información sobre la variable aleatoria a la que se refieren: La primera es una función lineal que indica el valor esperado de la misma, y la segunda es el grado de dispersión con respecto al valor esperado, en términos de probabilidad, ambas definidas para los casos discreto y aleatorio de acuerdo a las ecuaciones siguientes [19] [42] [73].

$$\mu = E[X] := \sum_{i=1}^m x_i \mathbb{P}(X = x_i) \quad (7.11)$$

$$\mu = E[X] := \int_{\Omega} X(\omega) d\omega \quad (7.12)$$

$$\sigma = Var[X] := E[(X - EX)^2] = E[X^2 - (EX)^2] \quad (7.13)$$

En este ámbito, el procesamiento bayesiano de señales es aquél que se basa en la regla de Bayes para obtener información en situaciones donde existe incertidumbre, la cual puede ocurrir por información ruidosa o carente, representada mediante variables aleatorias. Siendo X e Y variables aleatorias, la regla de Bayes en este caso se expresa de la siguiente forma [19] [20] [74]:

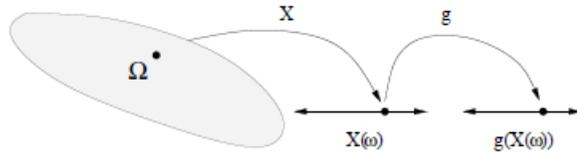


Figura 7.1: Diagrama de funciones de variables aleatorias [19].

$$\mathbb{P}(X|Y) = \frac{\mathbb{P}(Y|X) \cdot \mathbb{P}(X)}{\mathbb{P}(Y)} \quad (7.14)$$

Siendo $\mathbb{P}(X)$ llamada la **distribución a priori o prior** (información antes de la medición), $\mathbb{P}(Y)$ la **evidencia** y $\mathbb{P}(Y|X)$ la **verosimilitud o likelihood**. El procesamiento bayesiano busca actualizar una distribución a priori considerando la nueva información recibida [20] [74].

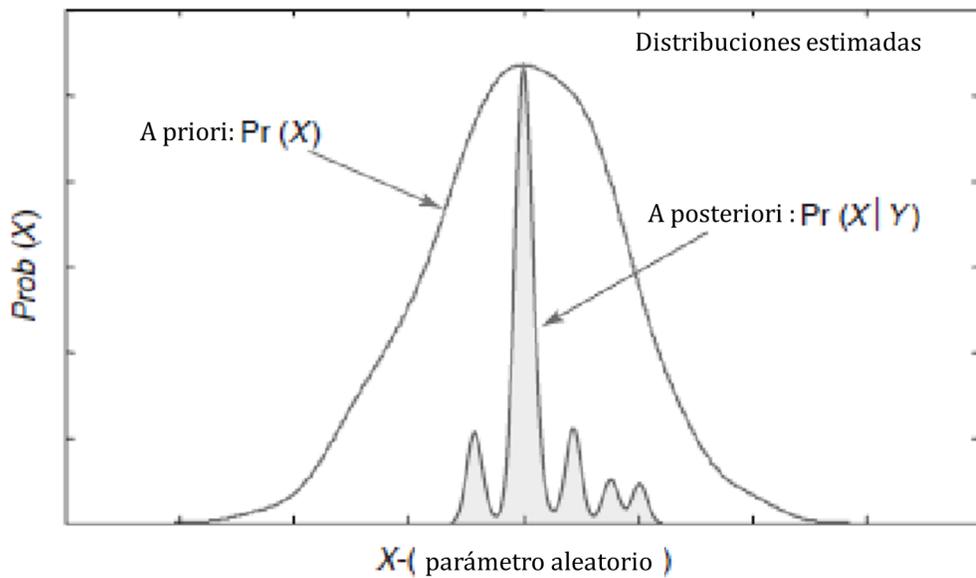


Figura 7.2: Gráfico de una actualización bayesiana [20].