

UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO
FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA,
INFORMÁTICA Y MECÁNICA
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



TESIS

**DISEÑO E IMPLEMENTACIÓN DEL CONTROLADOR
PROPORCIONAL INTEGRAL (PI) Y REGULADOR
CUADRÁTICO LINEAL (LQR) BASADOS EN
APRENDIZAJE POR REFUERZO PARA EL CONTROL
DE VELOCIDAD DE UN MOTOR DE CORRIENTE
CONTINUA CON DEVANADO DE CAMPO**

PRESENTADO POR:

Br. DAVID CHIRME SISA

**PARA OPTAR AL TÍTULO PROFESIONAL
DE INGENIERO ELECTRÓNICO**

ASESOR:

Dr. Ing. ROGER JESÚS COAQUIRA CASTILLO

CUSCO - PERÚ

2026



Universidad Nacional de San Antonio Abad del Cusco

INFORME DE SIMILITUD

(Aprobado por Resolución Nro.CU-321-2025-UNSAAC)

El que suscribe, el Asesor ROGER JESÚS COAGUIRA CASTILLO
..... quien aplica el software de detección de similitud al
trabajo de investigación/tesis titulada: DISEÑO E IMPLEMENTACIÓN DEL CONTROLADOR
PROPORCIONAL INTEGRAL (PI) Y REGULADOR CUADRÁTICO LINEAL (LQR)
BASADOS EN APRENDIZAJE POR REFUERZO PARA EL CONTROL DE VELOCIDAD
DE UN MOTOR DE CORRIENTE CONTINUA CON DEVANADO DE CAMPO

Presentado por: DAVID CHIRME SISA DNI N° 72023116;
presentado por: DNI N°:
Para optar el título Profesional/Grado Académico de INGENIERO ELECTRÓNICO

Informo que el trabajo de investigación ha sido sometido a revisión por 02 veces, mediante el
Software de Similitud, conforme al Art. 6° del **Reglamento para Uso del Sistema Detección de**
Similitud en la UNSAAC y de la evaluación de originalidad se tiene un porcentaje de 10 %.

Evaluación y acciones del reporte de coincidencia para trabajos de investigación conducentes a grado académico o título profesional, tesis

Porcentaje	Evaluación y Acciones	Marque con una (X)
Del 1 al 10%	No sobrepasa el porcentaje aceptado de similitud.	☒
Del 11 al 30 %	Devolver al usuario para las subsanaciones.	☐
Mayor a 31%	El responsable de la revisión del documento emite un informe al inmediato jerárquico, conforme al reglamento, quien a su vez eleva el informe al Vicerrectorado de Investigación para que tome las acciones correspondientes; Sin perjuicio de las sanciones administrativas que correspondan de acuerdo a Ley.	☐

Por tanto, en mi condición de Asesor, firmo el presente informe en señal de conformidad y **adjunto**
las primeras páginas del reporte del Sistema de Detección de Similitud.

Cusco, 18 de Agosto de 2026

Firma

Post firma Dr. Ing. Roger Jesús Coaguira Castillo

Nro. de DNI 01333608

ORCID del Asesor 0000-0003-3791-110X

Se adjunta:

- Reporte generado por el Sistema Antiplagio.
- Enlace del Reporte Generado por el Sistema de Detección de Similitud: oid: 27259:614970959

David Chirme Sisa

chirme_tesis_v8.pdf



Universidad Nacional San Antonio Abad del Cusco

Detalles del documento

Identificador de la entrega

trn:oid:::27259:614970959

Fecha de entrega

17 ago 2026, 12:32 p.m. GMT-5

Fecha de descarga

18 ago 2026, 4:55 p.m. GMT-5

Nombre del archivo

chirme_tesis_v8.pdf

Tamaño del archivo

5.9 MB

204 páginas

48.511 palabras

230.081 caracteres

10% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- Bibliografía
- Texto citado
- Texto mencionado

Fuentes principales

- 7%  Fuentes de Internet
- 3%  Publicaciones
- 8%  Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.º de alertas de integridad para revisión

-  **Caracteres reemplazados**
332 caracteres sospechosos en N.º de páginas
Las letras son intercambiadas por caracteres similares de otro alfabeto.
-  **Texto oculto**
2143 caracteres sospechosos en N.º de páginas
El texto es alterado para mezclarse con el fondo blanco del documento.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

DEDICATORIA

A mi madre Agripina Sisa y a mi padre Flavio Chirme, que han sido una constante fuente de inspiración, apoyo y amor incondicional en cada momento de esta pequeña travesía.

A Gabriela, César, Maritza y Delfor, mis hermanos, a los que admiro y agradezco por las cosas vividas.

A Soly, mi pequeña bendición que ha sido, en su ignorancia, una motivación y apoyo permanente en mis momentos críticos y días sombríos.

A Joshua y Abigail, mis sobrinos.

A Dios, por permitirme conocer a estas personas maravillosas y hacer de mi vida un camino inexplorado.

—David Chirme Sisa

AGRADECIMIENTO

Extiendo mis agradecimientos a las personas que me han acompañado en el desarrollo de esta tesis. Agradezco en sobremanera a mis padres, hermanos y sobrinos por sus consejos y confianza en mis decisiones.

Agradezco a mis asesor, Dr. Ing. Roger Jesús Coaquira Castillo, por su apoyo, orientación e iniciativa de emprender este proyecto, y por brindarme las herramientas necesarias que me han servido para la culminación de este trabajo.

A mi amigo Roly Gutierrez, que entre tantas aficiones y momentos compartidos que han hecho llevadero mi estancia en esta ciudad, me ha motivado a seguir con mis proyectos. A mi amiga Mafer, por los momentos de diversión, tertulias y tardes de café. A Kevin, mi amigo y maestro.

A las personas que tuve la bendición de conocer en este corto periodo de mi vida; sepan que les guardo admiración y estima.

A mí por la obstinada y enfermiza persistencia.

Suele decirse que conviene juzgar a una persona no por las cosas que dice, sino por las que hace. En mi caso, soy un idiota incurable, sin remedio, lo mismo por las cosas que digo, que por las que hago. Puedo decir entonces que, cuando se trata de ser un idiota, he sido un hombre coherente, consistente.

—Jaime Bayly Letts

RESUMEN

En la industria, la robótica y los vehículos eléctricos, donde el control de velocidad de motores es crítico, los sistemas de control deben operar bajo distintos escenarios sin perder estabilidad, rapidez y eficiencia. Para abordar este desafío, en este trabajo se diseñan los controladores proporcional integral (PI) y regulador cuadrático lineal (LQR) orientados al control de velocidad de un motor DC con devanado de campo de Lucas Nülle y de potencia nominal 0.3 kW, utilizando algoritmos de aprendizaje por refuerzo. Los controladores diseñados se implementaron en el sistema de servoaccionamiento de Lucas Nülle del laboratorio de Electrónica de Potencia de la Escuela Profesional de Ingeniería Electrónica, para posteriormente ser sometidos a pruebas de cambios de referencia, rampas de velocidad y perturbaciones de carga. Se diseñaron tres variantes del controlador PI: el basado en el criterio del óptimo simétrico (PI-OS), el basado en Ziegler-Nichols (PI-ZN) y el basado en TD3 con término no lineal (PI-rlNL). Este último presenta mejoras notables respecto a PI-OS y PI-ZN, mostrando una respuesta más rápida y suave con una reducción promedio del consumo de potencia entre 3.6 % y 5.1 %. Por otro lado, se diseñó el LQR basado en el algoritmo de Hewer (LQR-RICC) y el LQR basado en el algoritmo MGPI (LQR-MRL). En particular, LQR-MRL logra, en promedio, mejores resultados en la mayoría de métricas de seguimiento, junto con una reducción marginal de la potencia del orden de ~ 0.02 %.

Palabras Clave: Motor DC con devanado de campo, Controlador PI, Controlador LQR, Aprendizaje por refuerzo.

CONTENIDO

Resumen	IV
Índice de figuras	XI
Índice de tablas	XVII
Introducción	XXI
1. Generalidades	1
1.1. Planteamiento del problema	1
1.1.1. Problema general	4
1.1.2. Problemas específicos	5
1.2. Objetivos	5
1.2.1. Objetivo general	5
1.2.2. Objetivos específicos	5
1.3. Variables	6
1.3.1. Variables independientes	6
1.3.2. Variables dependientes	6
1.4. Justificación	6
1.4.1. Justificación tecnológica	6
1.4.2. Justificación académica	7

1.4.3.	Justificación económica	7
1.4.4.	Justificación personal	7
1.5.	Delimitación del estudio	8
1.5.1.	Delimitación espacial	8
1.6.	Metodología	8
1.6.1.	Tipo de investigación	8
1.6.2.	Alcances	8
1.6.3.	Nivel de la investigación	9
1.6.4.	Diseño de la investigación	9
1.6.5.	Técnicas e instrumentos de recolección de datos	10
1.6.6.	Limitaciones	10
1.6.7.	Análisis de datos y métricas de evaluación	10
2.	Marco teórico	11
2.1.	Antecedentes	11
2.2.	Bases teóricas	16
2.2.1.	Sistemas de control	16
2.2.2.	El controlador PID	17
2.2.3.	Control óptimo	17
2.2.3.1.	El regulador cuadrático lineal - LQR	18
2.2.3.2.	Espacio de estados en tiempo discreto y LQR discreto	19
2.2.4.	Control adaptativo	21
2.2.4.1.	Controlador adaptativo directo	21
2.2.4.2.	Controlador adaptativo indirecto	22
2.2.5.	Aprendizaje por refuerzo	22
2.2.5.1.	Introducción al aprendizaje por refuerzo	24
2.2.5.2.	Algoritmo DQN	26

2.2.5.3.	Algoritmo DDPG	28
2.2.5.4.	Algoritmo TD3	33
2.2.5.5.	Algoritmo Generalized Policy Iteration	36
2.2.6.	Control óptimo adaptativo.	39
2.2.7.	Motor de corriente continua con devanado de campo	39
2.2.7.1.	Modelo del motor DC con devanado de campo	41
2.2.8.	Métricas de desempeño	42
3.	Modelamiento matemático y estimación de parámetros	44
3.1.	Modelo matemático del motor DC	44
3.1.1.	Modelo no lineal	45
3.1.2.	Modelo lineal	46
3.2.	Equipamiento y el sistema de servoaccionamiento	47
3.3.	Obtención de parámetros y funciones del modelo	49
3.3.1.	Parámetros eléctricos	49
3.3.1.1.	Resistencia e inductancia de armadura R_A y L_A	50
3.3.1.2.	Resistencia e inductancia de excitación R_E y L_E	53
3.3.2.	Función del flujo de excitación Ψ_E	55
3.3.3.	Constante de máquina C_m	59
3.3.4.	Momento de inercia del rotor Θ_M	60
3.3.5.	Función de pérdidas por fricción $M_b = f(\omega_m)$	63
3.4.	Parámetros físicos del motor DC	64
3.5.	Prueba y simulación de los modelos	65
3.5.1.	Métricas de evaluación de los modelos	67
3.5.2.	Simulación del modelo no lineal	68
3.5.3.	Simulación del modelo lineal	69
3.5.4.	Simulación del modelo lineal estimado	70

3.5.5. Comparación de los modelos	71
4. Diseño y sintonización de los controladores	72
4.1. Diseño del controlador proporcional integral - PI	72
4.1.1. Arquitectura del controlador PI	72
4.1.2. Sintonización del controlador PI del lazo de corriente mediante el criterio del módulo óptimo	74
4.1.3. Sintonización del controlador PI de velocidad mediante el criterio óptimo simétrico	76
4.1.4. Sintonización del controlador PI de velocidad mediante el método de Ziegler-Nichols	77
4.1.5. Sintonización del controlador PI basado en aprendizaje por refuerzo (algoritmo TD3)	80
4.1.5.1. Entorno, estado y acción	81
4.1.5.2. Recompensa	82
4.1.5.3. Agente	83
4.1.5.4. Entrenamiento (TD3-Algoritmo 2)	86
4.1.5.5. Término no lineal del controlador PI de velocidad basado en TD3	91
4.1.6. Verificación de los controladores PI de velocidad en simulación . . .	94
4.2. Diseño del regulador cuadrático lineal - LQR	96
4.2.1. Arquitectura del controlador LQR	96
4.2.2. Sintonización del controlador LQR basado en la solución a la ecua- ción de Riccati (algoritmo de Hewer)	100
4.2.3. Sintonización del controlador LQR basado en aprendizaje por re- fuerzo (algoritmo MGPI)	102
4.2.4. Verificación de los controladores LQR en simulación	105
5. Implementación, pruebas y resultados	108
5.1. Pruebas de seguimiento de trayectoria y perturbaciones de carga	111
5.2. Método de conteo de Borda para la evaluación del mejor controlador . . .	113

5.3.	Implementación, pruebas y resultados de los controladores PI	114
5.3.1.	Sobre el controlador PI basado en Ziegler-Nichols	115
5.3.2.	Pruebas y resultados de cambio de referencia	116
5.3.2.1.	Cambios de referencia: Prueba 1	116
5.3.2.2.	Cambios de referencia: Prueba 2	120
5.3.2.3.	Resumen de desempeño - Pruebas de cambio de referencia	123
5.3.3.	Pruebas y resultados de rampas de velocidad	124
5.3.3.1.	Rampas de velocidad: Prueba 1	124
5.3.3.2.	Rampas de velocidad: Prueba 2	127
5.3.3.3.	Resumen de desempeño - Pruebas de rampas de velocidad	130
5.3.4.	Pruebas y resultados de perturbaciones de carga	130
5.3.4.1.	Perturbaciones de carga: Prueba 1	130
5.3.4.2.	Perturbaciones de carga: Prueba 2	134
5.3.4.3.	Resumen de desempeño - Pruebas de perturbaciones de carga	137
5.3.5.	Resumen general del desempeño de los controladores PI	137
5.4.	Implementación, pruebas y resultados de los controladores LQR	139
5.4.1.	Pruebas y resultados de cambio de referencia	140
5.4.1.1.	Cambios de referencia: Prueba 1	140
5.4.1.2.	Cambios de referencia: Prueba 2	143
5.4.1.3.	Resumen de desempeño - Pruebas de cambio de referencia	145
5.4.2.	Pruebas y resultados de rampas de velocidad	147
5.4.2.1.	Rampas de velocidad: Prueba 1	147
5.4.2.2.	Rampas de velocidad: Prueba 2	150
5.4.2.3.	Resumen de desempeño - Pruebas de rampas de velocidad	152
5.4.3.	Pruebas y resultados de perturbaciones de carga	153
5.4.3.1.	Perturbaciones de carga: Prueba 1	153

5.4.3.2. Perturbaciones de carga: Prueba 2	156
5.4.3.3. Resumen de desempeño - Pruebas de perturbaciones de carga	159
5.4.4. Resumen general del desempeño de los controladores LQR	159
5.5. Discusión de resultados	161
Conclusiones	165
Recomendaciones	167
Referencias bibliográficas	169
Anexos	176

ÍNDICE DE FIGURAS

2.1. Arquitectura de control en lazo cerrado	16
2.2. Forma estándar del control adaptativo directo	22
2.3. Forma estándar del control adaptativo indirecto	22
2.4. Estructura y entorno del aprendizaje por refuerzo	23
2.5. Estructura actor-crítico del aprendizaje reforzado	24
2.6. Estructura actor-crítico del algoritmo DDPG	31
2.7. Estructura actor-crítico del algoritmo TD3	33
2.8. Circuito equivalente del motor DC	39
2.9. Características Torque-Velocidad normalizadas	40
3.1. El motor DC con devanado de campo	44
3.2. Sistema de servoaccionamiento Lucas Nülle (LNSD system)	47
3.3. Equipos destinados a la medición de los parámetros eléctricos	49
3.4. Circuito de medición para estimar R_A	50
3.5. Datos experimentales y curva de regresión lineal $[v_A] \propto [i_A]$	51
3.6. Circuito de medición para estimar L_A	51
3.7. Datos experimentales y curva de regresión lineal $[V_A(j\omega)] \propto [I_A(j\omega)]$. . .	52
3.8. Circuito de medición para estimar R_E	53
3.9. Datos experimentales y curva de regresión lineal $[v_E] \propto [i_E]$	54

3.10. Circuito de medición para estimar L_E	54
3.11. Datos experimentales y curva de regresión lineal $[V_E(j\omega)] \propto [I_E(j\omega)]$. .	55
3.12. Configuración experimental para estimar $\Psi_E(i_E)$	56
3.13. Montaje de los equipos para estimar $\Psi_E(i_E)$	57
3.14. Datos experimentales y curva de regresión cuadrática $\Psi_E(i_E) = f(i_E)$. . .	58
3.15. Datos experimentales y curva de regresión lineal $[v_A] \propto [\Psi_{Enom} \cdot \omega_m]$. . .	60
3.16. Montaje de los equipos para estimar Θ_M	61
3.17. Respuesta de velocidad frente al par constante programado	61
3.18. Rampas de aceleración y frenado del motor DC	62
3.19. Datos experimentales del par ejercido por el servomotor $M_s - \omega_m$	63
3.20. Datos experimentales y curva de ajuste $M_s(\omega_m)$	64
3.21. Diagrama de bloques de los equipos para las pruebas en lazo abierto	65
3.22. Montaje de los equipos para las pruebas en lazo abierto	66
3.23. Diagrama de bloques en Simulink para las pruebas en lazo abierto	66
3.24. Respuesta en lazo abierto del modelo no lineal contra la respuesta real . . .	68
3.25. Respuesta en lazo abierto del modelo lineal contra la respuesta experimental	69
3.26. Respuesta en lazo abierto del modelo lineal estimado contra la respuesta real	70
3.27. Gráfico de barras de la comparación de las métricas por variables de estado	71
4.1. Arquitectura del controlador PI de velocidad del motor DC	73
4.2. Dinámica del bucle de control de velocidad	73
4.3. Estructura para la sintonización del control de velocidad	76
4.4. Ganancia del sistema K_S para la sintonía del PI de velocidad	77
4.5. Curva de respuesta en forma de S	78
4.6. Respuesta de velocidad frente a una consigna de corriente $i_{A-ref} = 1 \text{ A}$. .	79
4.7. Estructura de sintonización del controlador PI de velocidad basado en TD3	81
4.8. Agente RL que implementa el lazo PI de velocidad	83

4.9. Arquitectura de la sintonización del controlador PI de velocidad usando TD3	83
4.10. Red crítica del agente TD3	84
4.11. Red actor del agente TD3	85
4.12. Arquitectura de la sintonización en MATLAB/Simulink usando TD3 . . .	86
4.13. Bloque de control de velocidad con agente RL en Simulink	87
4.14. Proceso de entrenamiento del agente TD3 en MATLAB/Simulink	88
4.15. Resultados de la exploración del agente TD3 en el espacio de ganancias del controlador PI (K_p, K_i)	89
4.16. Evolución por episodio de la respuesta del motor DC en lazo cerrado . . .	90
4.17. Dinámica de la ganancia integral no lineal $K_{i-nt}(e)$	92
4.18. Respuestas en simulación de los controladores PI de velocidad	94
4.19. Arquitectura de seguimiento del controlador LQR	96
4.20. Arquitectura del LQR para el control de velocidad del motor DC	99
4.21. Norma del vector de control K - $\ K\ $	102
4.22. Norma del vector de control K - $\ K\ $ - MGPI	103
4.23. Evolución de los estados para diferentes valores de λ	104
4.24. Esquema en Simulink para la simulación de los controladores LQR	105
4.25. Respuestas en simulación de los controladores LQR de velocidad	106
5.1. Diagrama de bloques de los equipos utilizados durante las pruebas de im- plementación	108
5.2. Configuración experimental de los equipos para las pruebas de implemen- tación	109
5.3. Integración del MCU C2000 con la rutina de control en Simulink	109
5.4. Programa ActiveServo para pruebas de carga de las máquinas eléctricas . .	110
5.5. Programa LNLiveScope del Toolbox CO3636-3D para la captura de datos .	111
5.6. Programa en Simulink del controlador PI	114
5.7. Primera respuesta del controlador PI basado en Ziegler-Nichols	116
5.8. Respuesta del controlador PI-OS (óptimo simétrico)	116

5.9. Respuesta del controlador PI-ZN (Ziegler Nichols)	117
5.10. Respuesta del controlador PI-rlNL (TD3 con término no lineal)	117
5.11. Comparación de los controladores PI - Cambios de referencia: prueba 1 . .	118
5.12. Gráfico de barras de las métricas de desempeño de los controladores PI frente a cambios de referencia - Prueba 1	119
5.13. Respuesta del controlador PI-OS (óptimo simétrico)	120
5.14. Respuesta del controlador PI-ZN (Ziegler Nichols)	120
5.15. Respuesta del controlador PI-rlNL (TD3 con término no lineal)	121
5.16. Comparación de los controladores PI - Cambios de referencia: prueba 2 . .	121
5.17. Gráfico de barras de las métricas de desempeño de los controladores PI frente a cambios de referencia - Prueba 2	123
5.18. Respuesta del controlador PI-OS (óptimo simétrico)	124
5.19. Respuesta del controlador PI-ZN (Ziegler Nichols)	124
5.20. Respuesta del controlador PI-rlNL (TD3 con término no lineal)	125
5.21. Comparación de los controladores PI - Rampas de velocidad: prueba 1 . . .	125
5.22. Gráfico de barras de las métricas de desempeño de los controladores PI frente a rampas de velocidad - Prueba 1	126
5.23. Respuesta del controlador PI-OS (óptimo simétrico)	127
5.24. Respuesta del controlador PI-ZN (Ziegler Nichols)	127
5.25. Respuesta del controlador PI-rlNL (TD3 con término no lineal)	128
5.26. Comparación de los controladores PI - Rampas de velocidad: prueba 2 . . .	128
5.27. Gráfico de barras de las métricas de desempeño de los controladores PI frente a rampas de velocidad - Prueba 2	129
5.28. Respuesta del controlador PI-OS (óptimo simétrico)	130
5.29. Respuesta del controlador PI-ZN (Ziegler Nichols)	131
5.30. Respuesta del controlador PI-rlNL (TD3 con término no lineal)	131
5.31. Comparación de los controladores PI - Perturbaciones de carga: prueba 1 .	132
5.32. Gráfico de barras de las métricas de desempeño de los controladores PI frente a perturbaciones de carga - Prueba 1	133

5.33. Respuesta del controlador PI-OS (óptimo simétrico)	134
5.34. Respuesta del controlador PI-ZN (Ziegler Nichols)	134
5.35. Respuesta del controlador PI-rlNL (TD3 con término no lineal)	135
5.36. Comparación de los controladores PI - Perturbaciones de carga: prueba 2 .	135
5.37. Gráfico de barras de las métricas de desempeño de los controladores PI frente a perturbaciones de carga - Prueba 2	136
5.38. Métricas de desempeño promedio de los controladores PI	138
5.39. Programa en Simulink del controlador LQR	139
5.40. Respuesta del controlador LQR-RICC (algoritmo de Hewer)	140
5.41. Respuesta del controlador LQR-MRL (algoritmo MGPI)	141
5.42. Comparación de los controladores LQR - Cambios de referencia: prueba 1 .	141
5.43. Gráfico de barras de las métricas de desempeño de los controladores LQR frente a cambios de referencia - Prueba 1	143
5.44. Respuesta del controlador LQR-RICC (algoritmo de Hewer)	143
5.45. Respuesta del controlador LQR-MRL (algoritmo MGPI)	144
5.46. Comparación de los controladores LQR - Cambios de referencia: prueba 2 .	144
5.47. Gráfico de barras de las métricas de desempeño de los controladores LQR frente a cambios de referencia - Prueba 2	146
5.48. Respuesta del controlador LQR-RICC (algoritmo de Hewer)	147
5.49. Respuesta del controlador LQR-MRL (algoritmo MGPI)	147
5.50. Comparación de los controladores LQR - Rampas de velocidad: prueba 1 .	148
5.51. Gráfico de barras de las métricas de desempeño de los controladores LQR frente a rampas de velocidad - Prueba 1	149
5.52. Respuesta del controlador LQR-RICC (algoritmo de Hewer)	150
5.53. Respuesta del controlador LQR-MRL (algoritmo MGPI)	150
5.54. Comparación de los controladores LQR - Rampas de velocidad: prueba 2 .	151
5.55. Gráfico de barras de las métricas de desempeño de los controladores LQR frente a rampas de velocidad - Prueba 2	152
5.56. Respuesta del controlador LQR-RICC (algoritmo de Hewer)	153

5.57. Respuesta del controlador LQR-MRL (algoritmo MGPI)	154
5.58. Comparación de los controladores LQR - Perturbaciones de carga: prueba 1	154
5.59. Gráfico de barras de las métricas de desempeño de los controladores LQR frente a perturbaciones de carga - Prueba 1	156
5.60. Respuesta del controlador LQR-RICC (algoritmo de Hewer)	156
5.61. Respuesta del controlador LQR-MRL (algoritmo MGPI)	157
5.62. Comparación de los controladores LQR - Perturbaciones de carga: prueba 2	157
5.63. Gráfico de barras de las métricas de desempeño de los controladores LQR frente a perturbaciones de carga - Prueba 2	158
5.64. Métricas de desempeño promedio de los controladores LQR	160

ÍNDICE DE TABLAS

3.1. Definición de símbolos del modelo matemático del motor DC	45
3.2. Descripción del equipamiento del sistema LNSD	48
3.3. Mediciones de v_A , i_A y estimación de R_A	50
3.4. Mediciones de v_A , i_A en RMS y estimación de L_A	52
3.5. Mediciones de v_E , i_E y estimación de R_E	53
3.6. Mediciones de v_E , i_E en RMS y estimación de L_E	55
3.7. Mediciones de i_E y v_A para estimar $\Psi_E(i_E)$; con $\omega_m = 2000RPM$	58
3.8. Mediciones de ω_m y v_A para estimar C_m ; con $\Psi_E = \Psi_{Enom} = 1$	59
3.9. Estimación de Θ_M , dado Θ_S por el fabricante	63
3.10. Parámetros físicos del motor DC	65
3.11. Métricas de evaluación del modelo no lineal	68
3.12. Métricas de evaluación del modelo lineal	69
3.13. Métricas de evaluación del modelo lineal estimado	70
3.14. Comparación de las métricas de evaluación promedio de los modelos	71
4.1. Lazo de control según el accionamiento	72
4.2. Primer método de sintonía de Ziegler-Nichols basada en la respuesta al escalón	78
4.3. Detalles de la arquitectura de la red crítica del agente TD3	84

4.4.	Detalles de la arquitectura de la red actor del agente TD3	85
4.5.	Variables de entrenamiento e hiperparámetros	86
4.6.	Especificaciones técnicas del ordenador y el software	87
4.7.	Indicadores de desempeño transitorio del motor DC frente a cambios de referencia bajo controladores PI en simulación	95
4.8.	Indicadores de desempeño del motor DC en seguimiento de trayectoria y consumo energético bajo controladores PI en simulación	95
4.9.	Evaluación de convergencia de los estados ω_m y ξ	105
4.10.	Indicadores de desempeño transitorio del motor DC frente a cambios de referencia bajo controladores LQR en simulación	107
4.11.	Indicadores de desempeño del motor DC en seguimiento de trayectoria y consumo energético bajo controladores LQR en simulación	107
5.1.	Pruebas de cambios de referencia	111
5.2.	Pruebas de rampas de velocidad	111
5.3.	Perfiles de perturbaciones de carga	112
5.4.	Indicadores de desempeño transitorio de los controladores PI frente a cambios de referencia en implementación - Prueba 1	118
5.5.	Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores PI frente a cambios de referencia en implementación - Prueba 1	118
5.6.	Indicadores de desempeño transitorio de los controladores PI frente a cambios de referencia en implementación - Prueba 2	122
5.7.	Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores PI frente a cambios de referencia en implementación - Prueba 2	122
5.8.	Resumen del puntaje de Borda de los controladores PI en pruebas de cambio de referencia	123
5.9.	Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores PI frente a rampas de velocidad en implementación - Prueba 1	126
5.10.	Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores PI frente a rampas de velocidad en implementación - Prueba 2	129

5.11. Resumen del puntaje de Borda de los controladores PI en pruebas de rampas de velocidad	130
5.12. Indicadores de desempeño transitorio de los controladores PI frente a perturbaciones de carga en implementación - Prueba 1	132
5.13. Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores PI frente a perturbaciones de carga en implementación - Prueba 1	132
5.14. Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores PI frente a perturbaciones de carga en implementación - Prueba 2	136
5.15. Resumen del puntaje de Borda de los controladores PI en pruebas de perturbaciones de carga	137
5.16. Indicadores de desempeño transitorio promedio de los controladores PI . . .	137
5.17. Indicadores de desempeño promedio en seguimiento de trayectoria y consumo energético de los controladores PI	138
5.18. Puntaje de Borda basado en las métricas de desempeño promedio de los controladores PI	138
5.19. Indicadores de desempeño transitorio de los controladores LQR frente a cambios de referencia en implementación - Prueba 1	142
5.20. Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores LQR frente a cambios de referencia en implementación - Prueba 1	142
5.21. Indicadores de desempeño transitorio de los controladores LQR frente a cambios de referencia en implementación - Prueba 2	145
5.22. Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores LQR frente a cambios de referencia en implementación - Prueba 2	145
5.23. Resumen del puntaje de Borda de los controladores LQR en pruebas de cambio de referencia	146
5.24. Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores LQR frente a rampas de velocidad en implementación - Prueba 1	148
5.25. Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores LQR frente a rampas de velocidad en implementación - Prueba 2	151

5.26. Resumen del puntaje de Borda de los controladores LQR en pruebas de rampas de velocidad	152
5.27. Indicadores de desempeño transitorio de los controladores LQR frente a perturbaciones de carga en implementación - Prueba 1	155
5.28. Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores LQR frente a perturbaciones de carga en implementación - Prueba 1	155
5.29. Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores LQR frente a perturbaciones de carga en implementación - Prueba 2	158
5.30. Resumen del puntaje de Borda de los controladores LQR en pruebas de perturbaciones de carga	159
5.31. Indicadores de desempeño transitorio promedio de los controladores LQR .	159
5.32. Indicadores de desempeño promedio en seguimiento de trayectoria y consumo energético de los controladores LQR	160
5.33. Puntaje de Borda basado en las métricas de desempeño promedio de los controladores LQR	160
5.34. Mejoras porcentuales del controlador PI-r1NL vs. PI-OS y PI-ZN	161
5.35. Mejoras porcentuales del controlador LQR-MRL vs. LQR-RICC	163

INTRODUCCIÓN

En esta tesis se diseñaron, sintonizaron e implementaron controladores PI y LQR mediante técnicas de aprendizaje por refuerzo para el control de velocidad de un motor DC con devanado de campo. Estos controladores se comparan con sus contrapartes obtenidas por métodos tradicionales bajo distintos escenarios: cambios de consigna, rampas de velocidad y perturbaciones de carga. El trabajo abarca cinco capítulos con diferentes aspectos progresivamente desarrollados para completar el proyecto.

El Capítulo 1 presenta las generalidades del trabajo de investigación, cuyo enfoque se limita al planteamiento del problema, que destaca y justifica la importancia de explorar nuevos métodos de sintonización de controladores. Además, se definen los objetivos, variables, delimitaciones, alcances y metodología del trabajo.

El Capítulo 2 aborda los antecedentes en los que se basa la tesis; además, proporciona el marco teórico que sienta las bases de los métodos, algoritmos y técnicas empleadas tanto para el diseño, la sintonización e implementación, como para el análisis de los resultados.

El Capítulo 3 presenta a detalle el proceso del modelamiento matemático del motor DC. Aquí se exponen los experimentos realizados en conjunto con las herramientas de software y hardware empleados.

El Capítulo 4 describe el proceso completo de diseño y sintonización de los controladores, desde la definición de los esquemas de control hasta la obtención sus parámetros,

culminando con la verificación en un entorno de simulación.

El Capítulo 5 es donde se expone, con detalle, el proceso de implementación de los controladores, así como la configuración de las rutinas de control y de protección en Simulink r2016a. Finalmente, se presenta el análisis de los resultados, los cuales son consecuencia directa del proceso de diseño.

CAPÍTULO 1:

GENERALIDADES

1.1. Planteamiento del problema

Las tecnologías de accionamiento de motores vienen desarrollándose activamente, e innovaciones y avances en el diseño, análisis y control de estos han emergido en años recientes (Chau, 2015). Los motores de corriente continua (motores DC¹) son ubicuos en la industria, robótica, automóviles, vehículos eléctricos (Electric Vehicle - EV) y otros equipos electrónicos de consumo (Beckett et al., 2020). No obstante, requieren mecanismos de control para alcanzar un comportamiento deseado (Tüfenkçi et al., 2023).

La clasificación de las máquinas eléctricas de corriente continua se basa principalmente en la fuente de excitación del campo. Esta puede ser fija, como en los motores de imanes permanentes, o generada mediante bobinados electromagnéticos, como en el *motor DC con devanado de campo* (Chau, 2015). Los motores con devanado de campo ofrecen alta flexibilidad operativa, permitiendo ajustar dinámicamente el flujo electromagnético, lo que mejora el control de frenado y posibilita un ajuste adaptativo de las características torque-velocidad. Además, el control de velocidad no se limita al voltaje de armadura. Por estas razones, en escenarios que demandan adaptabilidad a variaciones de carga, control de velocidad estable y bajo costo de producción, el motor DC con devanado de campo

¹Del inglés *Direct Current*, corriente continua en español.

suele preferirse sobre los motores de imán permanente (Anton, 2024; Chau, 2015).

En aplicaciones industriales, robótica y vehículos eléctricos, donde el control de velocidad es crítico, los sistemas de control deben operar los motores bajo distintos escenarios. Por ello, se utilizan pruebas estándar de perturbación de carga (*load disturbance - torque step*) para medir la reacción frente a cambios súbitos de torque, que pueden representar subida de pendientes en vehículos eléctricos, variaciones en las sujeciones o cambios de fricción en fajas transportadoras, máquinas CNC, etc. También se emplean perfiles de torque variables no lineales—como señales sinusoidales típicas de entornos dinámicos—, así como rampas de velocidad que simulan ciclos de aceleración/frenado (*spin-up/down*), lo que permite evaluar torque, aceleración y estabilidad a lo largo del rango operativo (Liceaga-Castro et al., 2022; Bermúdez et al., 2021; Çelik et al., 2024).

Al día de hoy existen múltiples métodos de control clásicos y modernos que han sido estudiados ampliamente en las últimas décadas. La introducción de la inteligencia artificial (IA) y los avances en computación han contribuido al desarrollo de estas tecnologías, conduciéndolas hacia una nueva generación. Así, las tecnologías actuales de control basadas en IA o asistidas por esta son alternativas recurrentes. No obstante, es importante mencionar que frente a la constante evolución de la inteligencia artificial en el área de control automático, los controladores clásicos y modernos continúan ocupando un lugar relevante en la industria (Hägglund & Guzmán, 2024; Sarhadi, 2025).

De acuerdo a Ogata (2010), el controlador clásico proporcional integral derivativo (PID) se ha extendido notablemente, ocupando más del 50 % de los sistemas automáticos en aplicaciones industriales. Siendo aún optimistas, Alejandro-Sanjines et al. (2023) señalan que alrededor del 95 % de sistemas en lazo cerrado adoptan este tipo de familias; refiriéndose al control PI. La adopción generalizada del control PID se debe a su eficiencia, buen desempeño e implementación práctica (Ogata, 2010; Hägglund & Guzmán, 2024).

El regulador cuadrático lineal (*Linear Quadratic Regulator* - LQR) es un método de control óptimo que se ha erigido como una piedra angular en la ingeniería de control

moderna (StudySmarter, [s.f.](#)), siendo adoptado en muchas aplicaciones complejas desde control de procesos, electrónica de potencia, robótica, vehículos autónomos, vehículos eléctricos y aéreos, por mencionar algunos (Jeong et al., 2020; Wang et al., 2023; Okyere et al., 2019; Chirme-Sisa et al., 2024). Una ventaja notable del LQR es que es idóneo en aplicaciones donde el consumo energético mínimo sea fundamental, sin perder desempeño en la regulación de los estados (Cho et al., 2023; Wang et al., 2023; Lewis et al., 2012).

El aprendizaje por refuerzo (*Reinforcement Learning* - RL) es una técnica de IA, que adopta el principio de aprendizaje mediante la interacción con el entorno, análogo al proceso humano de adquirir habilidades. En este proceso, existe una percepción consciente de las respuestas del entorno a las acciones ejecutadas y una intencionalidad de influenciar dicho entorno a partir del comportamiento. Contribuciones para el formalismo del aprendizaje reforzado provienen de disciplinas de la psicología afines a la ciencia del aprendizaje y la teoría de control relativas al control óptimo y su solución usando métodos de función de valor y programación dinámica. Estas disciplinas eran en esencia independientes, sin embargo, se interrelacionaron gracias a métodos de aprendizaje por diferencia temporal, introducido por Richard Sutton en 1988. Así, estas disciplinas convergieron para producir el campo moderno del aprendizaje reforzado a finales de 1980 (Barto & Sutton, 2018).

Las investigaciones computacionales y algorítmicas del aprendizaje reforzado se han extendido, con cientos de estudios activos en disciplinas diversas como la psicología, economía, teoría de control, inteligencia artificial y neurociencia (Barto & Sutton, 2018). En particular, contribuciones a la teoría de control revisten especial interés para la comunidad de ingeniería electrónica y de control. Como se expuso, a pesar de los avances gigantescos en la IA, los métodos de control como el PID prevalecen por su evidente eficiencia, fácil implementación y bajo costo computacional (Åström & Kumar, 2014). De igual forma, el LQR se prefiere sobre alternativas más complejas debido a su estabilidad garantizada, robustez y eficiencia computacional, una vez calculada la matriz de ganancia óptima (Lewis et al., 2012). Por este motivo, investigaciones para el diseño y sintonización de los controladores PID y LQR han llamado la atención de los profesionales y la comunidad

científica. La intención detrás de estos trabajos es usar los algoritmos de aprendizaje por refuerzo con el objeto de alcanzar parámetros óptimos del PID y LQR (Alejandro-Sanjines et al., 2023; Tufenkci et al., 2023; Filipović et al., 2024; Li et al., 2018, 2024).

Entonces, el PID y su variación PI tienen uso generalizado en la industria y son preferidos para el control y regulación de velocidad de motores industriales, incluyendo motores DC con devanado de campo (Husnaini et al., 2020; Chau, 2015; Rashid, 2014). También, el LQR ha sido empleado para el control de motores DC (Saisudha et al., 2016; Bharti et al., 2020). Los dos controladores logran un buen rendimiento si se sintonizan correctamente; sin embargo, alcanzar los parámetros óptimos puede requerir recursos significativos y prolongadas sesiones de prueba manual, resultando contraproducente (Alejandro-Sanjines et al., 2023). Por esto, han surgido alternativas automáticas basadas en algoritmos de aprendizaje por refuerzo para el diseño y sintonización de estos controladores (PI y LQR), capaces de explorar y encontrar los parámetros óptimos de control sin depender de un modelo matemático exacto cuando los algoritmos se ejecutan en un entorno simulado, o incluso prescindir de él si se implementan en tiempo real. No obstante, la implementación de estos algoritmos en tiempo real, aunque prometedora, presenta grandes retos (Brunton, 2021). Tomando en cuenta que el problema de la sintonización de los controladores PI y LQR llama el interés de la comunidad, y que cualquier mejora en el rendimiento del control de la velocidad de un motor DC conlleva beneficios, que entre tantas tiene implicaciones económicas, se hace necesario explorar y aplicar métodos que contribuyan a la sintonización de los controladores PI y LQR aplicado al control de velocidad del motor de corriente continua con devanado de campo.

1.1.1. Problema general

¿Cómo emplear algoritmos de aprendizaje por refuerzo para el diseño y posterior implementación de los controladores proporcional integral (PI) y regulador cuadrático lineal (LQR) aplicados al control de velocidad de un motor de corriente continua con devanado de campo?

1.1.2. Problemas específicos

1. ¿Qué algoritmos basados en aprendizaje por refuerzo son adecuados para la sintonización de los controladores PI y LQR?
2. ¿Cómo implementar los controladores PI y LQR garantizando su correcto funcionamiento?
3. ¿Cómo evaluar el desempeño de los controladores a partir de datos experimentales?

1.2. Objetivos

1.2.1. Objetivo general

Diseñar e implementar los controladores proporcional integral (PI) y regulador cuadrático lineal (LQR) aplicando algoritmos de aprendizaje por refuerzo para el control de velocidad de un motor de corriente continua con devanado de campo.

1.2.2. Objetivos específicos

1. Identificar los algoritmos de aprendizaje por refuerzo adecuados para la sintonización de los controladores PI y LQR.
2. Implementar los controladores PI y LQR en un ambiente controlado de laboratorio y verificar su correcto funcionamiento.
3. Evaluar y comparar el comportamiento de los controladores PI y LQR basados en algoritmos de aprendizaje por refuerzo con sus versiones obtenidas mediante métodos tradicionales a través de pruebas y métricas de desempeño.

1.3. Variables

1.3.1. Variables independientes

Las ganancias del controlador PI (K_p y K_i), y la matriz de control óptima del controlador LQR (K).

1.3.2. Variables dependientes

La velocidad del motor, voltaje de armadura y corriente de armadura.

1.4. Justificación

1.4.1. Justificación tecnológica

En la ingeniería de control, el diseño y sintonización de los controladores es fundamental para lograr que un sistema dinámico exhiba un comportamiento deseado. Sin embargo, estos procesos requieren de un experto y ajustes manuales según la aplicación específica con el fin de obtener un buen desempeño del sistema a controlar. Alternativamente, técnicas modernas de inteligencia artificial (IA) y aprendizaje automático (*machine learning* - ML), en especial de aprendizaje reforzado (RL), pueden sustituir los procesos manuales de sintonización. Estos métodos permiten explorar automáticamente un rango continuo de ganancias de los controladores, con el objetivo de lograr un rendimiento óptimo y robusto del sistema (Alejandro-Sanjines et al., 2023). Por esto, el RL se presenta como una estrategia idónea y versátil en aplicaciones de control automático, como es la sintonización de los controladores PI y LQR en el control de velocidad de un motor DC con devanado de campo, que constituye el tema central del presente trabajo.

1.4.2. Justificación académica

El trabajo intenta contribuir a la investigación en el área de control, automatización y robótica de la escuela profesional de Ingeniería Electrónica, ya que se aplican y condensan los conocimientos adquiridos en el campo la ingeniería de control, además de aportar con el uso de algoritmos de aprendizaje por refuerzo en la sintonización de los controladores PI y LQR para el control de velocidad de un motor DC con devanado de campo. Los aportes aquí expuestos servirán a los estudiantes o investigadores entusiastas en el área.

1.4.3. Justificación económica

Se conoce que controladores mal sintonizados generan consumos energéticos altos debido a oscilaciones excesivas que causan sobrecalentamientos, esfuerzos mecánicos innecesarios y fallas en motores. Esto se evidencia en el desgaste de la estructura y los materiales del motor, lo que degrada su desempeño eléctrico, y exige mantenimiento frecuente, reemplazo de partes, hasta pérdidas de producción, según la aplicación (Brokaw, [s.f.](#); Živković et al., 2025). Esto conlleva a impactos económicos negativos en industrias o en clientes que usan equipos o máquinas equipadas con motores DC. El desarrollo y sintonización eficiente de los controladores PI y LQR para el control de velocidad de un motor DC implica reducción en el consumo energético y en costos de mantenimiento, disminuyendo el esfuerzo excesivo de los motores, extendiendo de esta forma su vida útil, lo que se transforma en impactos económicos positivos en aplicaciones donde la regulación de velocidad de motores DC es necesaria, por ejemplo industrias de manufactura, vehículos eléctricos y robótica.

1.4.4. Justificación personal

Igualmente, el proceso de realización de la investigación brinda al autor adquirir conocimientos en el área de aprendizaje por refuerzo aplicado al control automático.

1.5. Delimitación del estudio

1.5.1. Delimitación espacial

El diseño, la sintonización, entrenamientos y simulaciones de los controladores PI y LQR se llevarán a cabo en las instalaciones del Laboratorio Institucional de Investigación, Emprendimiento e Innovación en Sistemas de Control Automático, Automatización y Robótica (LIECAR), ya que cuenta con equipos de cómputo adecuados. La implementación, despliegue, pruebas y mediciones se concretarán en el laboratorio de Electrónica de Potencia de la Escuela Profesional de Ingeniería Electrónica.

1.6. Metodología

1.6.1. Tipo de investigación

La tesis se clasifica como una investigación aplicada, pues emplea técnicas de aprendizaje por refuerzo en la sintonización de los controladores PI y LQR empleados en la regulación de velocidad del motor de corriente continua con devanado de campo. Asimismo, es cuantitativa ya que emplea métricas e índices de desempeño en la evaluación del comportamiento de los controladores diseñados e implementados.

1.6.2. Alcances

El trabajo de investigación abarca el diseño, sintonización e implementación de los controladores PI y LQR basados en algoritmos de aprendizaje por refuerzo para el control de velocidad de un motor DC con devanado de campo. La etapa de diseño y sintonización de los controladores se realizará en un entorno simulado, lo que exige contar con el modelo

matemático para ejecutar los algoritmos de sintonización. A continuación, se diseñarán configuraciones experimentales con el fin de obtener los parámetros del modelo, tras lo cual se procederá a diseñar los algoritmos de sintonía adecuados para cada controlador. La implementación se desplegará en los módulos del laboratorio de Electrónica de Potencia de la Escuela Profesional de Ingeniería Electrónica. Cada controlador será implementado de forma independiente. Los controladores serán sometidos a pruebas de cambio de referencia, y pruebas de torque: perturbación de torque (torque step), torque sinusoidal y rampas de velocidad generados mediante un banco de pruebas de frenado programable. De los experimentos serán extraídos datos, que serán procesados y evaluados mediante métricas de desempeño. Finalmente, los controladores sintonizados por algoritmos de aprendizaje por refuerzo se analizarán comparativamente con sus versiones sintonizadas con métodos tradicionales.

1.6.3. Nivel de la investigación

El trabajo es una investigación explicativa, en virtud del estudio causa efecto entre los controladores PI y LQR basados en aprendizaje por refuerzo y la velocidad del motor, voltaje de armadura, corriente de armadura y métricas de desempeño.

1.6.4. Diseño de la investigación

La investigación sigue un diseño experimental, cuyo enfoque estriba en la obtención de resultados que verifiquen la utilidad de los algoritmos de aprendizaje reforzado en la sintonización de los controladores PI y LQR por medio de datos que permitan ser evaluados con métricas de desempeño; haciendo factible un análisis comparativo con otros métodos de sintonización tradicionales.

1.6.5. Técnicas e instrumentos de recolección de datos

Los datos de medición de velocidad, corriente de armadura y voltaje de armadura son registrados utilizando los programas: osciloscopio en línea (Live-Scope) del LN MATLAB Toolbox y ActiveServo en conjunto con los módulos existentes en el laboratorio de Electrónica de Potencia, garantizando confiabilidad en la captura de datos. Paralelamente, estos datos se almacenan en archivos de extensión CSV, o compatibles, para su posterior procesamiento.

1.6.6. Limitaciones

1. El diseño y la sintonización de los controladores PI y LQR mediante los algoritmos de aprendizaje por refuerzo se desarrollan en un entorno de simulación.
2. Las configuraciones de montaje, pruebas y el despliegue se limitan a los equipos del laboratorio de Electrónica de Potencia de la escuela profesional de Ingeniería Electrónica, donde se cuenta con el sistema de servoaccionamiento de motores Lucas Nülle y un motor DC con devanado de campo.

1.6.7. Análisis de datos y métricas de evaluación

Los resultados y los datos obtenidos en el proceso de las pruebas y experimentos se evalúan usando las métricas de desempeño: Tiempo de establecimiento t_s , sobreimpulso M_p , desviación estándar del error σ_e , valor RMS de la señal de control u_{RMS} , error cuadrático integral (ISE), error absoluto integral (IAE), error absoluto integral de tiempo (ITAE), error cuadrático integral de tiempo (ITSE), potencia consumida (P).

CAPÍTULO 2:

MARCO TEÓRICO

2.1. Antecedentes

Alejandro-Sanjines et al. (2023). **Adaptive PI Controller Based on a Reinforcement Learning Algorithm for Speed Control of a DC Motor.**

En el artículo, los autores exponen que los controladores PID son eficientes para diferentes procesos, siempre y cuando el sistema no cambie sus condiciones iniciales. Siendo el caso, el controlador se tendrá que sintonizar nuevamente. Frente a esta situación se propone un control adaptativo PI para controlar la velocidad de un motor DC con imán permanente, usando un enfoque de inteligencia artificial basado en aprendizaje por refuerzo. Se usa un agente actor-crítico y el algoritmo *Twin Delayed Deep Deterministic Policy Gradient* (TD3) con el objetivo de optimizar los parámetros del controlador PI. La función de recompensa del algoritmo TD3 se basa en el criterio LQG (*Linear Quadratic Gaussian*). Finalmente, el desempeño del método es comparado con un controlador clásico PI diseñado con la herramienta sisotool de MATLAB usando la función de costo LQG sobre la respuesta pico y la respuesta transitoria de los controladores. Los resultados indican que el controlador adaptativo PI supera en desempeño al controlador PI diseñado por sisotool y al controlador NARMA-L2 en un entorno de simulación.

Tufenkci et al. (2023). **A theoretical demonstration for reinforcement learning of PI control dynamics for optimal speed control of DC motors by using Twin Delay Deep Deterministic Policy Gradient Algorithm**

El estudio llevado a cabo por los autores aborda la aplicación del aprendizaje reforzado en la sintonización de controladores PI para el control de velocidad de un motor DC de imán permanente. Los autores proponen la sintonización óptima del controlador PI empleando el algoritmo TD3. El agente de RL se encarga de aprender, tanto la dinámica del motor DC como los parámetros del PI, con el objetivo de minimizar una función de costo óptima. El desempeño del PI sintonizado mediante este método se compara con el de los controladores PI sintonizados por los algoritmos *Deep Deterministic Policy Gradient* (DDPG), *Genetic Algorithm* (GA), *Particle Swarm Optimization* (PSO), *Grey Wolf Optimization* (GWO) y *Differential Evolution* (DE). El error cuadrático medio (MSE) y el promedio del valor absoluto de la señal de control (AAC) se utilizan para evaluar los controladores, demostrando que el controlador propuesto presenta un rendimiento superior en un entorno de simulación.

Filipović et al. (2024). **Design of LQ regulators with prescribed degree of stability for dual-rate systems based on reinforcement learning**

El artículo se encarga del estudio de sistemas de control inferencial con muestreo de doble tasa, los cuales son usados regularmente en procesos químicos. La idea es convertir el sistema de doble tasa de muestreo en un modelo discreto con una tasa de muestreo rápida, y estimar las muestras faltantes por medio de un elemento retenedor de primer orden (FOH). Luego, los autores introducen un regulador cuadrático lineal (*Linear Quadratic Regulator* - LQR) con un grado de estabilidad prescrita (todos los valores propios en lazo cerrado están en un rango $0 < \lambda < 1$, en magnitud) para la regulación de este tipo de sistemas. La matriz de ganancia óptima se calcula usando el algoritmo *Generalized Policy Iteration* (GPI), catalogada dentro de los algoritmos de aprendizaje por refuerzo basados en modelos (*model-based RL*). Finalmente, el regulador diseñado por este método demostró efectividad en presencia de dinámicas multiplicativas no modeladas.

Li et al. (2024). **Robust Adaptive Optimal Control of a DC Motor System Based on Policy Iteration**

Los autores de esta investigación señalan la importancia del control de motores en aplicaciones industriales, y destacan que los estudios en estos temas permanecen activos. Asimismo, el objetivo del estudio es diseñar un controlador óptimo robusto para un motor DC de imán permanente que sea capaz de regular la velocidad cuando ocurren incertidumbres en los parámetros del sistema y en las perturbaciones externas. Entonces, Li et al. (2024) proponen un método de control óptimo robusto para lidiar con este tipo de incertidumbres. Luego, el trabajo empieza por definir el modelo en espacio de estados del motor DC, y tomando en cuenta el seguimiento de referencia, se aumenta la integral del error de velocidad e al modelo en espacio de estados. Una vez definido el modelo extendido en espacio de estados, el problema se convierte en el diseño de un controlador óptimo. El algoritmo *Policy Iteration* (RL-PI) basado en aprendizaje por refuerzo es propuesto para obtener la matriz de control óptima. Después, la convergencia del algoritmo introducido se prueba mediante simulación. El sistema de control se prueba frente a incertidumbres paramétricas, demostrando convergencia y una ventaja evidente al considerar que el algoritmo Policy Iteration implementado en simulación, no necesita del modelo completo o la matriz de estados A .

Mamani et al. (2024). **Deep Reinforcement Learning Via Nonlinear Model Predictive Control for Thermal Process with Variable Longtime Delay.**

Mamani et al. (2024) hacen mención que los problemas existentes en el control de procesos térmicos giran en torno a la adaptabilidad a incertidumbres y perturbaciones provenientes de fuentes externas o modelos simplificados. Por esto, el artículo propone una técnica de control robusto basado en aprendizaje por refuerzo usando el algoritmo DDPG integrando un Control Predictivo por Modelo No lineal (*Nonlinear Model Predictive Control* - NMPC), el cual se compara con los controladores PID y NMPC tradicionales. Los controles propuestos se someten a cuatro pruebas: cambios de referencia, robustez frente a perturbación externa de temperatura, robustez frente a cambios de tiempo muerto θ_s

y cambios en la constante de tiempo τ_s del modelo. Las métricas de evaluación adoptadas son: error cuadrático integral (ISE), error cuadrático integral de tiempo (ITSE), error absoluto integral (IAE) y error absoluto integral de tiempo (ITAE). Los resultados demuestran que el control propuesto (NMPC-based DDPG) exhibe un comportamiento robusto y estable en presencia de cambios de referencia, perturbaciones y cambios de parámetros del modelo.

Akhare et al. (2021). **Power Electronics and Drives: Simulation to Implementation in Minutes: A Novel Implementation Tool for Power Engineers**

Akhare et al. (2021) recalcan el gran impacto de la Electrónica de Potencia, y que el uso de los variadores y controladores en aplicaciones industriales, junto con sus sistemas de control, han ganado gran interés desde inicios de siglo. El proyecto llevado a cabo comprende el empleo del sistema de Servoaccionamiento Lucas Nülle (*Lucas Nülle Servo Drive - LNSD*) y la herramienta LN MATLAB Toolbox (Lucas Nülle MATLAB Toolbox) a fin de introducir dos casos de estudio en un Motor Síncrono de Imán Permanente (*Permanent-Magnet Synchronous Motor - PMSM*). El primer caso implica el control autónomo de velocidad del PMSM y el segundo es una propuesta novedosa que emplea el controlador NMPC tolerante a fallas de circuito abierto en una de las fases, para el control de velocidad del PMSM. Cada caso se diseña en MATLAB/Simulink y se implementa en el sistema LNSD exitosamente, evidenciando de este modo la efectividad del sistema LNSD, donde los modelos creados en Simulink pueden ser probados en aplicaciones reales, llegando del diseño a la implementación en unos cuantos minutos. Como conclusión, los investigadores destacan los beneficios que ofrece el sistema LNSD, en conjunto con MATLAB/Simulink, para profesionales de Electrónica de Potencia, estudiantes, investigadores y docentes.

Uscamaita Quispetupa (2019). **Diseño de un sistema de detección de fallas para el sensor de velocidad de un motor DC Lucas Nülle configurado en derivación** [Tesis de grado para optar el grado académico de Ingeniero Electrónico, Universidad Nacional de San Antonio Abad del Cusco]

La investigación de esta tesis se enfoca en el diseño de un sistema de detección de fallas aditivas en la medición de velocidad de un motor DC en derivación. Para abordar este problema la autora propone un esquema de diagnóstico basado en modelos y el diseño de tres observadores: uno basado en el filtro de Kalman y dos en el Observador Proporcional Integral de Entradas Desconocidas (*Unknown Inputs Proportional Integral Observer* - UIPIO). El modelo en espacio de estados del sistema necesario en el diseño de los observadores se obtiene mediante algoritmos de identificación usando la herramienta de MATLAB *system identification*. El sistema de detección de fallas comprende la estimación de velocidad del motor mediante los observadores y la comparación de la misma con la velocidad medida por el sensor. Posteriormente, la comparación procede a la evaluación de los residuales definidos como la Raíz de la Media Cuadrática o valor RMS del error entre la velocidad estimada (tomada como referencia) y la medida. Como parte final de este sistema, la detección de falla se activa cuando el valor RMS del residual supera un umbral, que confina el nivel de incertidumbre en la medición. La implementación de este sistema se realizó en el laboratorio de Electrónica de Potencia de la escuela de Ingeniería Electrónica usando el sistema de Servoaccionamiento Lucas Nülle (LNSD system) y la herramienta LN MATLAB Toolbox. El rendimiento de los sistemas de detección de fallas se prueban en presencia de torque, y se inyectan tres fallas a la medición del sensor: descalibración por offset, ruido acoplado senoidal y pérdida de conexión. De los resultados, la autora concluye que el sistema de diagnóstico o detección de fallas tiene un mejor desempeño con el Observador Proporcional Integral de Entradas Desconocidas, ya que no presenta problemas al detectar las tres fallas aditivas.

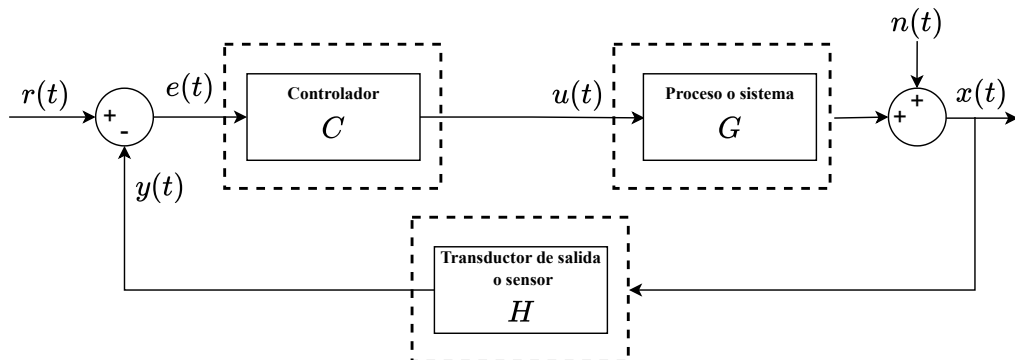
2.2. Bases teóricas

2.2.1. Sistemas de control

En teoría de control, un sistema de lazo cerrado se refiere a un componente, o conjunto de componentes, encargado de manipular o tomar decisiones sobre una señal de control que se alimenta a un proceso, planta o sistema, a partir de una señal de error generada por la comparación entre la medición actual del proceso y un valor de consigna. De esta forma, se asegura un comportamiento demandado (Nise, 2015).

Un esquema general de una arquitectura de control de lazo cerrado es mostrado en la Figura 2.1. El valor de consigna o referencia es $r(t)$, C representa al mecanismo de control cuya salida $u(t)$ es la señal de control. G es la planta o proceso a controlar con estado de salida $x(t)$. El bloque H es el transductor o sensor de salida que convierte el estado de salida del sistema $x(t)$ en una señal $y(t)$ que el controlador pueda manipular. La señal $e(t)$ es el error obtenido de la comparación de la referencia $r(t)$ y el estado actual del sistema $y(t)$. Los sistemas físicos también presentan señal de ruido; $n(t)$ es el ruido que se cuela a la lectura de los estados del sistema (Nise, 2015).

Figura 2.1: Arquitectura de control en lazo cerrado



2.2.2. El controlador PID

Gran parte de la teoría empleada en la actualidad para optimizar el desempeño de los sistemas de control es atribuido a Nicholas Minorsky (1885 - 1955). Sus contribuciones a los sistemas de control autónomo de navegación marítima condujeron a lo que hoy conocemos como controlador proporcional-integral-derivativo (PID) (Nise, 2015).

El control PID se define matemáticamente por la ecuación (2.1). Este consta de la suma de tres términos: el proporcional (P), el integral (I) y el derivativo (D). El término proporcional K_p representa la acción de control que actúa sobre el error actual; la acción integral se encarga de eliminar el error de estado estacionario mediante la acumulación del error pasado, y el término derivativo predice el error futuro, reduciendo las oscilaciones generadas por las acciones proporcional e integral (Åström & Hägglund, 2009).

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{d}{dt} e(t) \right) \quad (2.1)$$

Donde $u(t)$ es la señal de control, K_p la ganancia proporcional, T_i el tiempo integral y T_d el tiempo derivativo. El problema radica en encontrar las constantes K_p , T_i y T_d que hagan estable el sistema y que cumplan con las especificaciones dadas (Ogata, 2010).

2.2.3. Control óptimo

En la naturaleza, los recursos son limitados. Por tanto, en sistemas como células, organismos o especies animales, todas las acciones deben optimizar el consumo de recursos a fin de preservarlos. El control óptimo toma este principio para el diseño de controladores para sistemas de ingeniería humana (Lewis et al., 2012).

Dado un sistema dinámico lineal invariante en el tiempo descrito por

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (2.2)$$

con estados $x(t) \in \mathbb{R}^n$ y entrada de control $u(t) \in \mathbb{R}^m$. El control óptimo es un marco matemático donde uno selecciona una entrada de control $u(t) = h(x)$ para minimizar una función de costo $L(x, u, t)$. Westenbroek et al. (2022) aseveran que el proceso de diseño de la función de costo es generalmente heurístico. Lewis et al. (2012) tratan este tema sistemáticamente, formulando problemas de optimización de energía mínima (*minimum-energy problems*) con funciones de costo cuadráticas.

2.2.3.1. El regulador cuadrático lineal - LQR

El regulador cuadrático lineal (*Linear Quadratic Regulator* - LQR) es un método de control óptimo que provee ganancias de realimentación óptimas para asegurar estabilidad y buen rendimiento de un sistema (Chirme-Sisa et al., 2024).

El problema del LQR estriba en encontrar la matriz de control K (ecuación (2.3)) de tal manera que la función de costo cuadrático de horizonte infinito J , dada por

$$u = -Kx \tag{2.3}$$

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \tag{2.4}$$

sea mínima. Donde Q y R son matrices de penalización, definidas positivas o hermitianas semidefinidas positivas, que representan la importancia relativa de la regulación de los estados y el costo energético de la señal de control, respectivamente (Brunton, 2017). La principal ventaja del LQR es la optimización de la señal de control, tanto en el estado final como en los estados intermedios, dando lugar a un consumo mínimo de energía, sin comprometer la regulación de estados (Lewis et al., 2012). Además, este enfoque proporciona un método sistemático de cálculo de la matriz de control (Ogata, 2010).

La solución a este problema está dado por la ecuación (2.3), donde la matriz K es

$$K = R^{-1} B^T P, \tag{2.5}$$

y P es la matriz definida positiva solución a la ecuación de Riccati (ARE¹):

$$A^T P + PA + Q - PBR^{-1}B^T P = 0. \quad (2.6)$$

Bajo condiciones de controlabilidad de (A, B) , observabilidad de $(\sqrt{Q}, A)$ y siendo Q y R matrices definidas o semidefinidas positivas, la solución a la ecuación de Riccati será única y estable. Este método requiere pleno conocimiento de las matrices dinámicas del sistema (A, B) para dar con la solución a la ecuación de Riccati (Lewis et al., 2012).

2.2.3.2. Espacio de estados en tiempo discreto y LQR discreto

Los sistemas en espacio de estados en tiempo discreto describen la dinámica de sistemas muestreados mediante ecuaciones en diferencias. La representación general de un sistema lineal e invariante en el tiempo se expresa como:

$$x(k+1) = A_d x(k) + B_d u(k), \quad (2.7)$$

$$y(k+1) = C_d x(k) + D_d u(k), \quad (2.8)$$

(Lewis et al., 2012).

Dado un sistema en tiempo continuo: $\dot{x} = Ax + Bu$ (ecuación (2.2)), es posible obtener su representación discreta en espacio de estados a partir de las funciones presentadas a continuación.

$$\Phi(\tau) = e^{A\tau}, \quad (2.9)$$

$$\Gamma(\tau) = \int_0^\tau e^{A\eta} B d\eta. \quad (2.10)$$

Donde $\Phi(\tau)$ representa el efecto del estado inicial $x(k)$ sobre el estado siguiente $x(k+1)$, asumiendo una señal de entrada nula. Asimismo, $\Gamma(\tau)$ es el efecto de una señal de control

¹Del inglés *algebraic Riccati equation* (ecuación algebraica de Riccati). Esta ecuación se utiliza en la solución del problema LQR en control óptimo.

constante $u(k)$ sobre el estado $x(k+1)$.

De las ecuaciones (2.9) y (2.10) las matrices A_d y B_d se obtienen reemplazando el tiempo de muestreo T_s en las funciones $\Phi(\tau)$ y $\Gamma(\tau)$, respectivamente.

$$A_d = \Phi(T_s), \quad (2.11)$$

$$B_d = \Gamma(T_s). \quad (2.12)$$

Si A es invertible, B_d es calculado explícitamente usando:

$$B_d = A^{-1}(A_d - I)B. \quad (2.13)$$

Las matrices de salida C_d y de transferencia directa D_d se calculan usando sus versiones en tiempo continuo. Es decir, si $y = Cx + Du$, entonces:

$$C_d = C, \quad D_d = D. \quad (2.14)$$

(The MathWorks, Inc., 2025a).

Las integrales son calculadas usando las fórmulas de matrices exponenciales, las cuales están implementadas directamente en programas de computación numérica como MATLAB/Simulink. Además comandos como `c2d` implementan las ecuaciones presentadas anteriormente para convertir modelos en tiempo continuo a representaciones en tiempo discreto directamente (The MathWorks, Inc., 2025a).

Según Lewis et al. (2012) y The MathWorks, Inc. (2025a), el procedimiento para la solución del problema del LQR en tiempo discreto es el mismo; se definen las matrices de penalización Q y R , y la función de costo

$$J(u_k) = \sum_{k=1}^{\infty} (x_k^T Q x_k + u_k^T R u_k). \quad (2.15)$$

Luego, se busca la matriz solución P a la ecuación algebraica de Riccati en tiempo discreto

$$A^T P A - P + Q - A^T P B (B^T P B + R)^{-1} B^T P A = 0. \quad (2.16)$$

Y finalmente se obtiene la ley de control u_k y la matriz de control óptima discreta K_d , dada por:

$$u = -K_d x_k. \quad (2.17)$$

$$K_d = (B^T P B + R)^{-1} B^T P A, \quad (2.18)$$

2.2.4. Control adaptativo

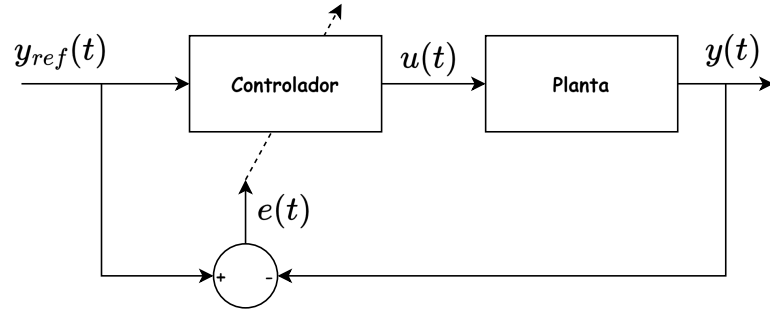
El control adaptativo se refiere a técnicas de realimentación en tiempo real que estabilizan un sistema y satisfacen criterios de diseño. Estos sistemas aprenden la dinámica del sistema en tiempo real a partir de mediciones a lo largo de las trayectorias² del sistema. Luego, valiéndose de una formulación adecuada del problema (un modelo adecuado), la dinámica o parámetros del sistema es aprendida en línea o en tiempo real para lograr los criterios de diseño (Lewis et al., 2012). Existen dos clases generales de control adaptativo: control adaptativo directo e indirecto.

2.2.4.1. Controlador adaptativo directo

El esquema básico de este controlador se muestra en la Figura 2.2. Este controlador busca minimizar el error de seguimiento de referencia $e(t)$. El controlador se define en términos de parámetros variables y reglas de sintonía adaptativa dependientes del error $e(t)$ que se implementan con el objetivo de actualizar los parámetros del controlador (Lewis et al., 2012).

²Aquí se refiere a la evolución de los estados del sistema $x(t)$ dado una entrada $u(t)$.

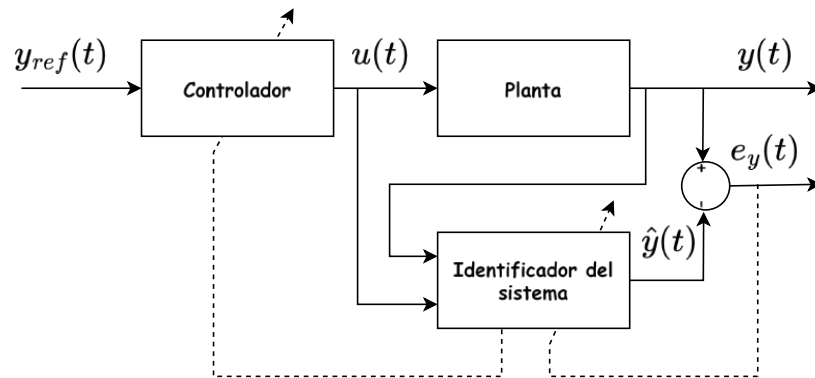
Figura 2.2: Forma estándar del control adaptativo directo



2.2.4.2. Controlador adaptativo indirecto

En la Figura 2.3 se tiene una arquitectura del control adaptativo indirecto. Para este caso, se usan dos redes de realimentación. Una de ellas aprende en tiempo real la dinámica del sistema o planta, $\dot{x} = f(x) + g(x)u$, de tal manera que el error de identificación $e_y(t)$ sea mínimo. Posteriormente, la segunda red actualiza los parámetros del controlador usando el modelo de la planta identificada (Lewis et al., 2012).

Figura 2.3: Forma estándar del control adaptativo indirecto

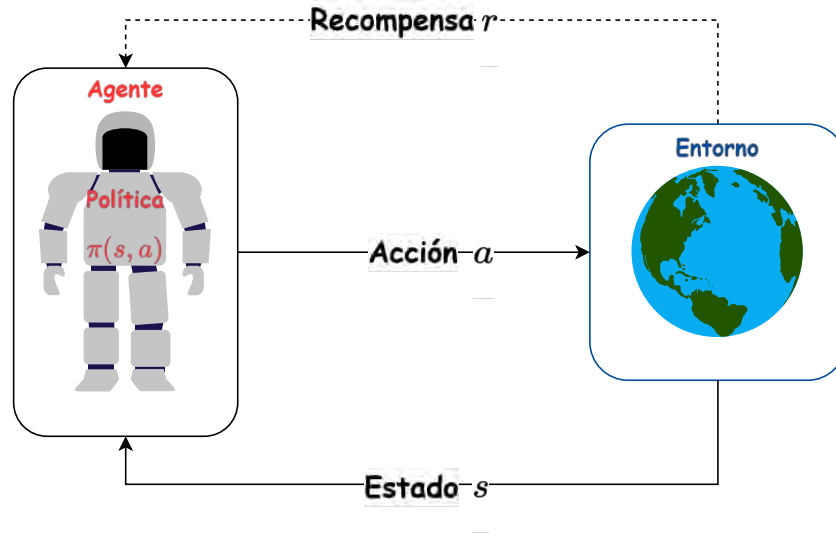


2.2.5. Aprendizaje por refuerzo

El aprendizaje por refuerzo (RL) es un tipo de aprendizaje automático, diferente al aprendizaje supervisado y no supervisado. El marco de referencia para el aprendizaje reforzado se muestra en la Figura 2.4. Esta ilustra cómo un *agente* aprende a partir de su experiencia y la interacción con su *entorno*. El concepto está inspirado en estudios

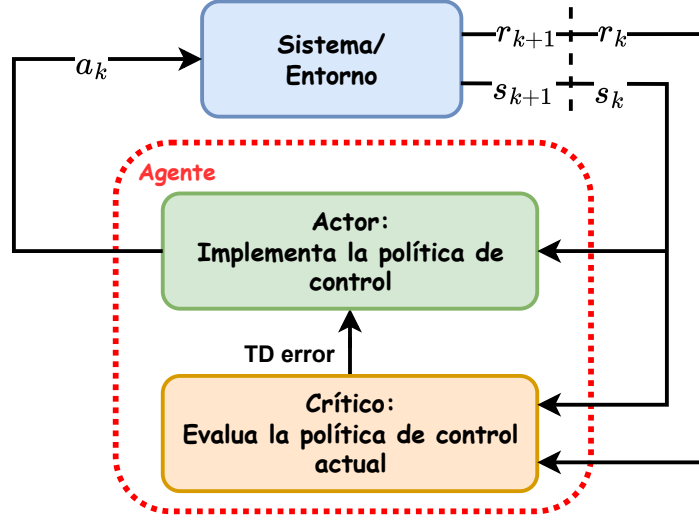
sobre el condicionamiento operante en comportamiento humano y animal: si una acción desencadena una recompensa, se tiende a repetirla; si en cambio, resulta en un castigo, tiende a evitarse (Encyclopedia Britannica, s.f.). El aprendiz (agente), sin recibir órdenes explícitas ni conocer su entorno, tiene la misión de descubrir el conjunto de *acciones* (a) que conducen a una *recompensa* óptima $r = r^*$. Entonces, aprende a mapear, por medio de *políticas* $\pi(s)$, situaciones o estados (s) a acciones (a), de tal manera se maximice una señal de recompensa numérica r en el largo plazo (Barto & Sutton, 2018).

Figura 2.4: Estructura y entorno del aprendizaje por refuerzo



En los esquemas de la Figuras 2.4 y 2.5 se distinguen dos elementos importantes: el agente y el entorno. El agente es responsable de la toma de decisiones y el entorno responde a las acciones (a_k) y presenta recompensas y nuevas situaciones (s_k, r_k) al agente (Barto & Sutton, 2018). La estructura de la Figura 2.5 es un tipo de arquitectura actor-crítico que implementa aprendizaje por refuerzo. El agente es conformado por dos redes neuronales: un actor, que selecciona acciones según la política actual, y un crítico, que evalúa las acciones y retroalimenta al actor para mejorar la política (Lewis et al., 2012). Vale la pena mencionar que en ingeniería de control el agente, entorno y acción son el controlador, sistema y señal de control, respectivamente.

Figura 2.5: Estructura actor-crítico del aprendizaje reforzado



2.2.5.1. Introducción al aprendizaje por refuerzo

Considerando los antecedentes presentados por Lakhani et al. (2022) y Lillicrap et al. (2015) a continuación se presenta una introducción formal al aprendizaje por refuerzo.

La política es comúnmente descrita por una función $\pi : \mathcal{S} \rightarrow \mathcal{A}$, los símbolos $\mathcal{S}$ y $\mathcal{A}$ denotan el conjunto de estados y acciones, respectivamente. El retorno acumulado (*cumulative return*) a partir de la muestra k se define como

$$R_k = \sum_{i=k}^T \gamma^{i-k} r_i(s_i, a_i), \quad (2.19)$$

donde $\gamma \in [0, 1]$ es el factor de descuento que pondera las recompensas inmediatas y futuras. Cuando las tareas se consideran continuas, el horizonte se extiende hasta $T = \infty$.

La función valor de estado (*state value function*) $V : \mathcal{S} \rightarrow \mathcal{R}$ mide el retorno esperado a partir de un estado determinado cuando el agente sigue la política π :

$$V^\pi(s) = \mathbb{E}_\pi [R_k | s = s_k]. \quad (2.20)$$

De forma similar, la función acción-valor $Q^\pi : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{R}$, mide el retorno esperado al comenzar en el estado s_k , tomar la acción a_k y, a partir de ese punto, seguir la política π . Esta función se expresa como

$$Q^\pi(s, a) = \mathbb{E}_\pi [R_k | s = s_k, a = a_k]. \quad (2.21)$$

A partir de la propiedad de Markov, la función acción-valor bajo la política π satisface la una relación recursiva conocida como la *ecuación de expectativa de Bellman*. Esta relación expresa el valor esperado para un par estado-acción como la suma de la recompensa inmediata y el valor futuro esperado bajo la misma política:

$$Q^\pi(s_k, a_k) = \mathbb{E}_{s' \sim P} [r(s_k, a_k) + \gamma \mathbb{E}_{a' \sim \pi(\cdot | s')} (Q^\pi(s', a'))]; \quad (2.22)$$

siendo $P = P(\cdot | s_k, a_k)$ la probabilidad de transición entre dos estados consecutivos s_k y $s' = s_{k+1}$, condicionada al par estado-acción actual (s_k, a_k) . En este contexto, $\pi(\cdot | s_k)$ es la política que define la distribución de posibles acciones siguientes $a' = a_{k+1}$ al llegar al estado s' , es decir, a' proviene de la distribución $\pi(\cdot | s')$. Aquí, las expectativas $\mathbb{E}_{s' \sim P}$ y $\mathbb{E}_{a' \sim \pi(\cdot | s')}$ representan el promedio sobre las posibles transiciones y acciones conforme estas distribuciones.

El problema del aprendizaje por refuerzo se puede formular, entonces, como la búsqueda de una política óptima π^* que maximice el retorno esperado a lo largo del espacio de estados y acciones, de modo que su función de acción-valor asociada Q^* supere a las funciones correspondientes de otras políticas.

Bajo la política óptima, la ecuación de *optimalidad de Bellman* para la el valor de acción óptima se expresa como

$$Q^*(s_k, a_k) = \mathbb{E}_{s' \sim P} \left[r(s_k, a_k) + \gamma \max_{a' \in \mathcal{A}} Q^*(s', a') \right]. \quad (2.23)$$

La ventaja de trabajar con las funciones de calidad Q es que, una vez que se obtiene Q^* , siguiendo el *principio de optimalidad de Bellman*, la política óptima π^* puede ser fácilmente recuperada al resolver

$$\pi^*(a|s_k) = \arg \max_{a \in \mathcal{A}} Q^*(s_k, a). \quad (2.24)$$

Existen diferentes métodos que dan solución a este problema; el objetivo principal de ellos es encontrar o aproximar las funciones $Q(s, a)$ y $\pi(s)$ que satisfagan las ecuaciones de optimalidad de Bellman.

2.2.5.2. Algoritmo DQN

Q-learning es un algoritmo ampliamente utilizado por ser *off-policy* y por su capacidad para aprender una política óptima sin requerir un modelo explícito del entorno (Lakhani et al., 2022). En este contexto, *off-policy* significa que Q-learning puede aprender la política óptima independientemente de la política usada durante la exploración, lo que permite exploraciones más flexibles mientras se estima el valor óptimo.

Q-learning resuelve de forma iterativa la ecuación de optimalidad de Bellman mediante el método de diferencia temporal (TD³), que combina recompensas inmediatas y estimaciones futuras. La diferencia temporal (TD_{error}) se define como

$$TD_{error} = r(s_k, a_k) + \gamma \max_{a' \in \mathcal{A}} Q(s', a') - Q(s_k, a_k), \quad (2.25)$$

y la actualización de la función de valor acción-estado como

$$Q(s_k, a_k) \leftarrow Q(s_k, a_k) + \alpha TD_{error}, \quad (2.26)$$

donde α es la tasa de aprendizaje. Estos pasos permiten al agente aproximar los valores óptimos $Q^*(s, a)$ para todos los pares de estado-acción mediante interacción con el entorno.

³Del inglés *Temporal Difference*.

Este método es *model-free* (sin modelo explícito del entorno), por lo que las actualizaciones por TD pueden aplicarse en tareas con horizonte continuo, incluso cuando no existe un estado terminal claramente definido. Sin embargo, las versiones tabulares clásicas de métodos basados en función valor—como Q-learning—asumen que los espacios de estados y de acciones son discretos y finitos, de modo que los valores $Q(s, a)$ pueden almacenarse y actualizarse en un tabla de búsqueda. En dominios con espacios continuos de estados y acciones, esta representación tabular no es viable y se requiere el uso de aproximadores de función para estimar los valores de Q en lugar de una tabla (Lillicrap et al., 2015).

Para generalizar Q-learning a espacios continuos de estados y acciones se utilizan funciones de aproximación paramétricas, como redes neuronales, para representar la función de calidad Q . En este enfoque, los parámetros de la red neuronal se ajustan para minimizar una medida del error entre el valor estimado por la red y el valor objetivo de Q .

La tarea principal es encontrar el conjunto de pesos w de la red neuronal que minimice la función de pérdida

$$J(w) = \mathbb{E}_{\pi} \left[(y_k - Q(s_k, a_k|w))^2 \right]. \quad (2.27)$$

La función de costo definida en la ecuación (2.27) es análoga a la de aprendizaje supervisado. En este caso, la red neuronal parametrizada por w está destinada a aproximar la función de calidad $Q(s_k, a_k)$, pero este valor objetivo no está disponible al inicio del entrenamiento. Por esta razón, se construye una aproximación objetivo, asumiendo por ejemplo: $y_k = Q(s_k, a_k) \approx r(s_k, a_k) + \gamma \max_{a' \in \mathcal{A}} Q(s', a'|w)$. Con esta aproximación, el entrenamiento de la red neuronal se realiza mediante descenso por gradiente estocástico (*stochastic gradient descent* - SGD). En cada iteración w se actualiza según

$$w' = w_{k+1} = w_k + \Delta w, \quad (2.28)$$

donde el incremento de los parámetros se calcula usando

$$\Delta w = \alpha [y_k - Q(s_k, a_k|w)] \nabla_w Q(s_k, a_k|w). \quad (2.29)$$

La combinación de Q-learning y funciones de aproximación basadas en redes neuronales, se conoce como *deep Q-Network* (DQN), un algoritmo de aprendizaje por refuerzo que emplea redes profundas para aproximar la función de valor acción-estado $Q(s, a)$ en espacios de alta dimensionalidad.

2.2.5.3. Algoritmo DDPG

Deep Deterministic Policy Gradient (DDPG) es un algoritmo *model-free* y *off-policy*⁴ diseñado para operar en entornos con espacios continuos de estados y acciones, donde las políticas deterministas son más eficientes que las basadas en distribuciones estocásticas (Lillicrap et al., 2015).

DDPG usa un enfoque actor-crítico (consulte la Figura 2.5) basado en el algoritmo *Deterministic Policy Gradient* (DPG). En DPG una política determinista parametrizada

$$\pi(s|\theta^\pi) \approx \pi(s) \quad (2.30)$$

se combina con un estimador de la función de valor acción-estado $Q(s, a|\theta^Q) = Q(s, a|w)$ ⁵ en un marco actor-crítico para aprender directamente un mapeo de estados a acciones continuas.

El esquema actor-crítico mostrado en la Figura 2.5 corresponde, en este contexto, a una política determinista $\pi(s|\theta^\pi)$ —el actor—y la función de calidad determinista $Q(s, a|\theta^Q)$ —el crítico—, donde el actor genera la acción dada una entrada de estado y el crítico evalúa dicha acción mediante la estimación del valor esperado a largo plazo. Esto se detalla a continuación.

Red Actor: Aproxima una política determinista $\pi(s|\theta^\pi) : \mathcal{S} \rightarrow \mathcal{A}$, encargada de mapear directamente un estado $s \in \mathcal{S}$ a una acción continua $a \in \mathcal{A}$. La política está parametrizada por θ^π , y sus parámetros se actualizan mediante el gradiente de política

⁴Se refiere a que el agente puede usar una política diferente a la que se está usando actualmente.

⁵Aquí se considera $w = \theta^Q$.

determinista (Deterministic Policy Gradient - DPG).

Los métodos basados en políticas buscan optimizar el retorno esperado inicial

$$J(\theta^\pi) = \mathbb{E}_{\pi(s|\theta^\pi), s} [R_1 | s_0], \quad (2.31)$$

donde s_0 es el estado inicial y la expectativa se toma sobre las trayectorias generadas por la política determinista. Para maximizar este objetivo, el algoritmo DPG usa SGD para actualizar θ^π tal que

$$\theta_{k+1}^\pi \leftarrow \theta_k^\pi + \beta \nabla_{\theta^\pi} J(\theta^\pi) |_{\theta^\pi = \theta_k^\pi}, \quad (2.32)$$

donde β es la tasa de aprendizaje de la red actor. El gradiente de $J(\theta^\pi)$, con respecto a los parámetros del actor, es

$$\nabla_{\theta^\pi} J(\theta^\pi) = \mathbb{E}_{s_k \sim P^\pi} [\nabla_a Q(s, a | \theta^Q) |_{s=s_k, a=\pi(s_k)} \nabla_{\theta^\pi} \pi(s | \theta^\pi) |_{s=s_k}], \quad (2.33)$$

donde P^π es la distribución de estados bajo la política determinista. Este gradiente combina la sensibilidad de la función Q respecto a las acciones con la sensibilidad de la política respecto a sus parámetros, guiando la actualización de θ^π hacia acciones que incrementen el retorno inicial (Lillicrap et al., 2015; Lakhani et al., 2022; Silver et al., 2014).

Red Crítica: Aproxima la función valor-acción $Q(s, a | \theta^Q)$, la cual estima el valor esperado del retorno a largo plazo al tomar una acción $a \in \mathcal{A}$ desde un estado $s \in \mathcal{S}$ bajo una política dada. Esta función es parametrizada por los pesos θ^Q de red la crítica.

Durante el entrenamiento, la red crítica θ^Q se ajusta minimizando la función de pérdida basada en el error cuadrático medio de Bellman (*mean-squared Bellman error*, MSBE), de forma similar a como se hace en el algoritmo DQN.

Además, el entrenamiento de los parámetros de la red crítica θ^Q se realiza sobre transiciones muestreadas aleatoriamente de un buffer de experiencias pasadas $\mathcal{D}$ (*replay buffer*). El cual almacena tuplas de experiencia (s_k, a_k, r_k, s') . Esta técnica reduce la correlación entre muestras consecutivas y mejora la estabilidad del aprendizaje mediante la

reutilización de experiencias anteriores (Lillicrap et al., 2015).

En suma al cambio sustancial expuesto líneas arriba, a fin de estabilizar más el sistema, se emplea una red objetivo (*target Network*) que aproxima el valor objetivo y_k de la siguiente manera:

$$y_k \approx r(s_k, a_k) + \gamma Q'(s', \pi'(s'|\theta^{\pi'})|\theta^{Q'}), \quad (2.34)$$

donde $Q'(s, a|\theta^{Q'})$ y $\pi'(s|\theta^{\pi'})$ son copias de la red crítica y actor, respectivamente. Estas redes objetivos se utilizan para calcular valores objetivos más estables durante el entrenamiento del crítico y el actor, evitando que el objetivo cambie abruptamente en cada actualización.

Los pesos de las redes objetivo se actualizan siguiendo un patrón de actualización suave (*soft target update*), que mezcla lentamente los parámetros aprendidos con los parámetros actuales de las redes objetivo según las siguientes ecuaciones:

$$\theta^{Q'} \leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'} \quad (2.35)$$

$$\theta^{\pi'} \leftarrow \tau \theta^{\pi} + (1 - \tau) \theta^{\pi'}, \quad (2.36)$$

donde τ es un factor pequeño que controla la velocidad de convergencia de la redes objetivo hacia las redes principales, lo que contribuye a la estabilidad del entrenamiento. Cabe mencionar, que las redes principales θ^Q y θ^{π} , se actualizan siguiendo las ecuaciones (2.28) y (2.33), respectivamente.

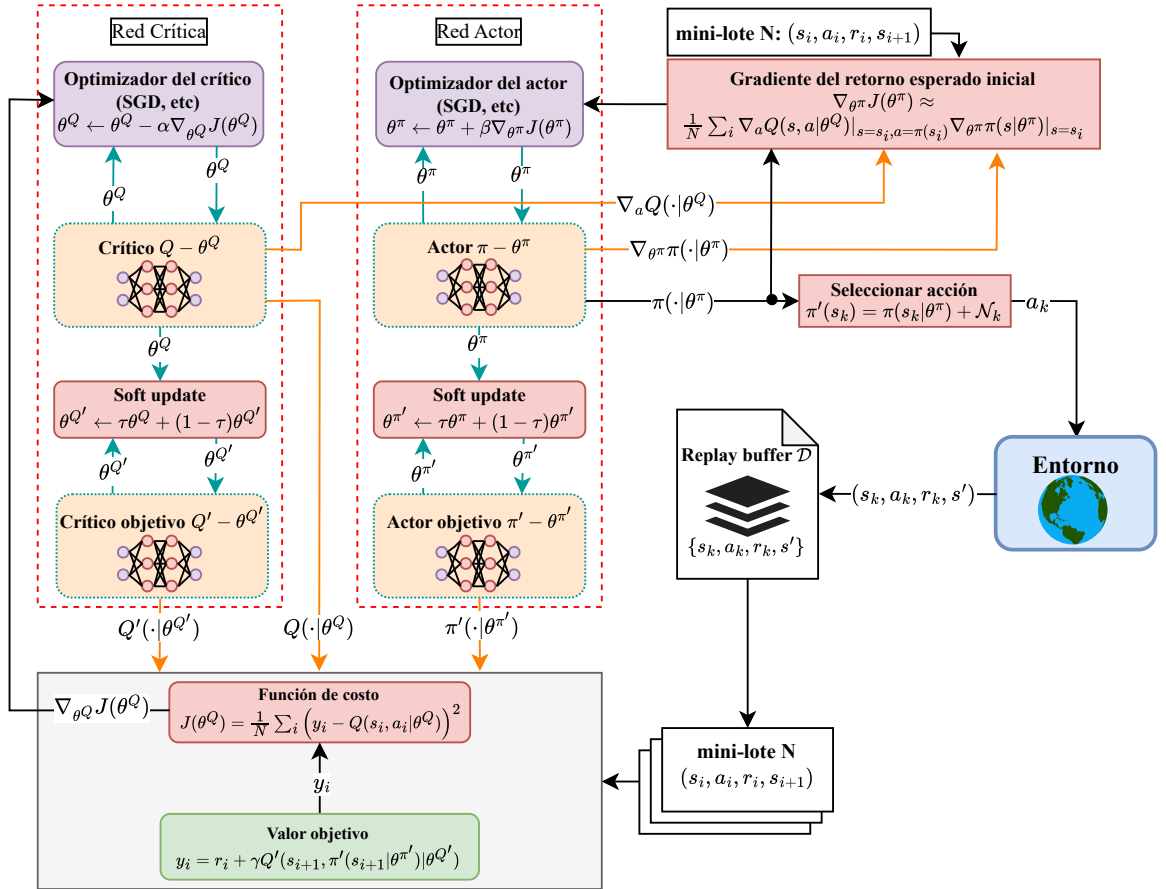
Para aprovechar las optimizaciones de hardware es esencial aprender en mini-lotes (*mini-batches*). Como en DQN, el buffer de transiciones—también llamado replay buffer o buffer de experiencias—es de longitud finita. En cada paso de tiempo, tanto el actor y el crítico se actualizan muestreando aleatoriamente un mini-lote de $\mathcal{D}$. Esto permite al algoritmo beneficiarse del aprendizaje de un conjunto de transiciones no correlacionadas (OpenAI, 2023a; Lillicrap et al., 2015).

Uno de los mayores desafíos de aprender en espacios continuos es la exploración. Una ventaja de los algoritmos off-policy como DDPG es que permiten tratar la exploración independientemente del algoritmo de aprendizaje. Por esta razón, Lillicrap et al. (2015) diseñan una política de exploración π' que se obtiene añadiendo ruido aleatorio $\mathcal{N}$ a la política del actor

$$\pi'(s_k) = \pi(s_k|\theta_k^\pi) + \mathcal{N}, \quad (2.37)$$

donde $\mathcal{N}$ es un proceso aleatorio utilizado para introducir variación en las acciones con el fin de promover la exploración del espacio de acciones. En el trabajo de Lillicrap et al. (2015), este ruido suele implementarse con un proceso de Ornstein-Uhlenbeck, que genera ruido correlacionado en el tiempo adecuado para entornos físicos con inercia, mejorando la eficiencia de la exploración en tareas de control continuo.

Figura 2.6: Estructura actor-crítico del algoritmo DDPG



Nota. Traducida y adaptada de Afzali et al. (2023) y Dong et al. (2023).

Algoritmo 1 Deep Deterministic Policy Gradient (DDPG)

- 1: Inicializar aleatoriamente la red crítica $Q(s, a|\theta^Q)$ y actor $\pi(s|\theta^\pi)$ con pesos θ^Q y θ^π .
- 2: Inicializar la red objetivo Q' y π' con pesos $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\pi'} \leftarrow \theta^\pi$
- 3: **Para** episodio = 1, M **hacer**
- 4: Inicializar un proceso aleatorio $\mathcal{N}$ para la exploración de acciones
- 5: Recibir el estado inicial s_1
- 6: **Para** t, k = 1, T **hacer**
- 7: Seleccionar la acción $a_k = \pi(s_k|\theta^\pi) + \mathcal{N}_k$ de acuerdo a la política actual y el ruido de exploración
- 8: Ejecutar la acción a_k y observar la recompensa r_k y el nuevo estado $s' = s_{k+1}$
- 9: Guardar la transición (s_k, a_k, r_k, s') en $\mathcal{D}$
- 10: Muestrear un mini-lote de N transiciones (s_i, a_i, r_i, s_{i+1}) del buffer $\mathcal{D}$
- 11: Asignar $y_i = r_i + \gamma Q'(s_{i+1}, \pi'(s_{i+1}|\theta^{\pi'})|\theta^{Q'})$
- 12: Actualizar el crítico minimizando la función costo:

$$L = \frac{1}{N} \sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$$

- 13: Actualizar la política del actor usando el gradiente de política muestreada:

$$\nabla_{\theta^\pi} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\pi(s_i)} \nabla_{\theta^\pi} \pi(s|\theta^\pi)|_{s=s_i}$$

- 14: Actualizar las redes objetivo:

$$\begin{aligned} \theta^{Q'} &\leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'} \\ \theta^{\pi'} &\leftarrow \tau \theta^\pi + (1 - \tau) \theta^{\pi'} \end{aligned}$$

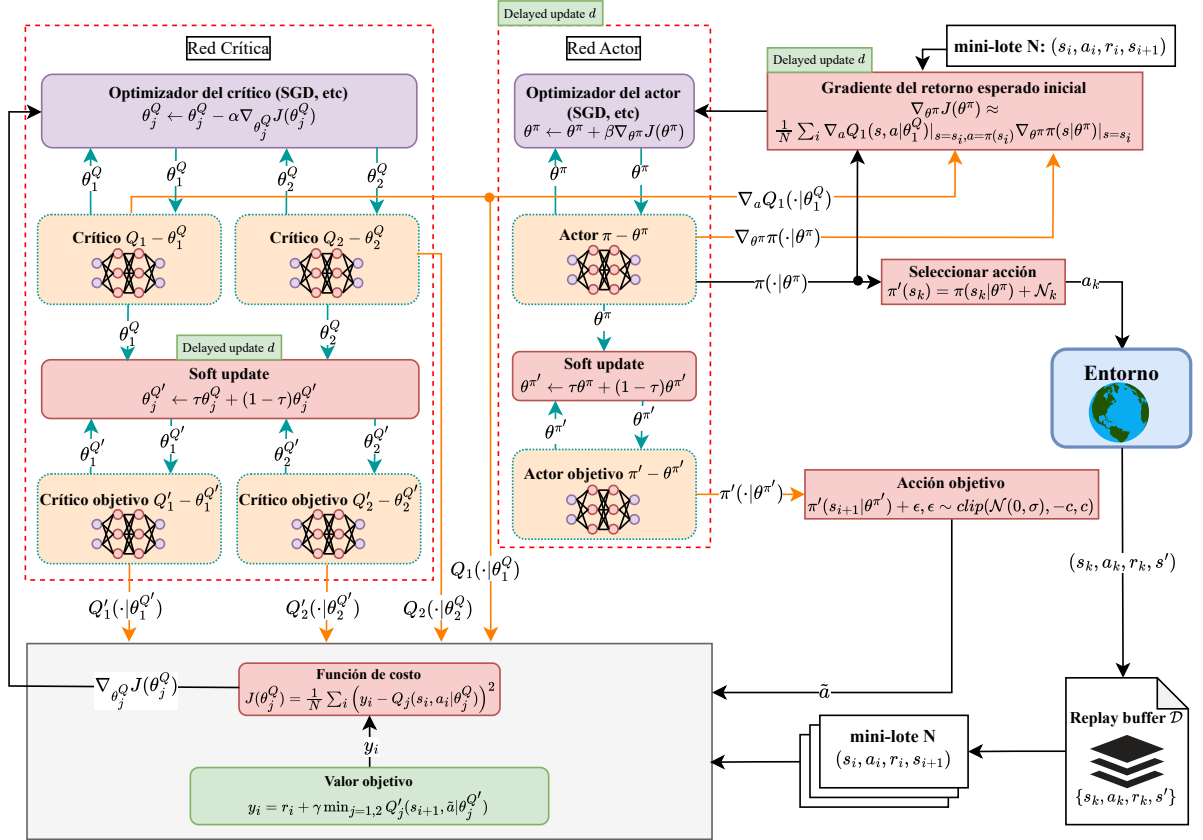
- 15: **fin Para**
 - 16: **fin Para**
-

El diagrama de bloques del algoritmo DDPG se muestra en la Figura 2.6. Este diagrama ilustra las interacciones entre los principales componentes del algoritmo—como las redes actor y crítico, las redes objetivo y el replay buffer—y cómo fluyen las señales entre ellos durante el entrenamiento y la interacción con el entorno. La estructura representada corresponde a la descripción secuencial del algoritmo que se detalla en el Algoritmo 1. En resumen, hay cuatro redes involucradas: (1) la red actor $\pi(s|\theta^\pi)$, que selecciona la acción basado en el estado actual; (2) la red crítica $Q(s, a|\theta^Q)$, que estima la función Q ; (3) la red objetivo actor $\pi'(s|\theta^{\pi'})$, que es una copia retardada de la red actor y se usa para generar acciones objetivo en el cálculo de los valores futuros; (4) la red objetivo crítico $Q'(s', a'|\theta^{Q'})|_{a'=\pi'(s'|\theta^{\pi'})}$, que es una copia retardada de la red crítica y evalúa el valor objetivo de Q para el siguiente estado y acción objetivo (Lakhani et al., 2022).

2.2.5.4. Algoritmo TD3

Twin Delayed Deep Deterministic Policy Gradient (TD3) es un algoritmo off-policy de RL basado en DDPG que reduce el sesgo de sobreestimación en las funciones de valor y mejora la estabilidad del entrenamiento. A diferencia de DDPG, TD3 usa dos redes críticas gemelas (*twin critics*) y emplea actualizaciones retardadas de la política y suavizado de la política objetivo para generar estimaciones de valor más conservadoras y confiables, lo que da lugar a políticas de control más robustas en espacios continuos de acción (Fujimoto et al., 2018; OpenAI, 2023b).

Figura 2.7: Estructura actor-crítico del algoritmo TD3



Nota. Traducida y adaptada de OpenAI (2023b) y Fujimoto et al. (2018).

Seguendo la Figura 2.7 y de acuerdo a Fujimoto et al. (2018) y Tufenkci et al. (2023), TD3 aborda las principales debilidades de DDPG mediante tres mecanismos clave:

Suavizado de políticas objetivo: Al calcular los valores objetivo de Q , se añade ruido recortado ϵ a la acción generada por la política objetivo $\pi'(s|\theta^{\pi'})$ para obtener la acción objetivo:

$$\tilde{a}_{i+1} = \pi'(s_{i+1}|\theta^{\pi'}) + \epsilon, \quad \epsilon \sim \text{clip}(\mathcal{N}(0, \sigma), -c, c). \quad (2.38)$$

Este ruido suaviza los valores Q alrededor de la acción objetivo, reduciendo la tendencia de la política a explotar picos erróneos en la estimación de valores.

Clipped double-Q learning: TD3 emplea dos críticos independientes, Q_1 y Q_2 , que estiman funciones Q separadas. El objetivo y_i usa el valor menor entre ambos críticos para limitar la sobreestimación:

$$y_i = r_i + \gamma \min_{j=1,2} Q'_j(s_{i+1}, \tilde{a}_{i+1}|\theta_j^{Q'}), \quad (2.39)$$

y las redes se actualizan minimizando las funciones de costo basadas en MSBE (ecuación (2.27)) y calculadas para cada crítico:

$$J(\theta_1^Q) = \mathbb{E}_\pi \left[\left(y_i - Q_1(s_i, a_i|\theta_1^Q) \right)^2 \right], \quad (2.40)$$

$$J(\theta_2^Q) = \mathbb{E}_\pi \left[\left(y_i - Q_2(s_i, a_i|\theta_2^Q) \right)^2 \right]. \quad (2.41)$$

Actualización de política retrasada: La política del actor se actualiza con menor frecuencia que las funciones Q . La actualización del gradiente para θ^π usa únicamente el crítico $Q_1(s, a|\theta_1^Q)$; manteniendo la formulación de gradiente determinista igual que en DDPG:

$$\nabla_{\theta^\pi} J(\theta^\pi) = \mathbb{E}_{s_k \sim P^\pi} \left[\nabla_a Q_1(s, a|\theta_1^Q)|_{s=s_i, a=\pi(s_i)} \nabla_{\theta^\pi} \pi(s|\theta^\pi)|_s = s_i \right]. \quad (2.42)$$

Una ventaja notable de TD3 es que las actualizaciones de la política se realizan

periódicamente tras varios pasos de actualización de los críticos, lo cual estabiliza la política y evita que se ajuste prematuramente a estimaciones de alto sesgo.

Esta combinación de técnicas se resume en el Algoritmo 2.

Algoritmo 2 Twin Delayed Deep Deterministic Policy Gradient (TD3)

- 1: Inicializar aleatoriamente las redes críticas $Q_1(s, a|\theta_1^Q)$, $Q_2(s, a|\theta_2^Q)$ con pesos θ_1^Q , θ_2^Q
- 2: Inicializar aleatoriamente la red actor $\pi(s, a|\theta^\pi)$ con pesos θ^π
- 3: Inicializar aleatoriamente las redes objetivo Q'_1 , Q'_2 y π' con pesos $\theta_1^{Q'} \leftarrow \theta_1^Q$, $\theta_2^{Q'} \leftarrow \theta_2^Q$ y $\theta^{\pi'} \leftarrow \theta^\pi$
- 4: Inicializar el replay buffer $\mathcal{D}$.
- 5: **Para** episodio = 1, M **hacer**
- 6: Inicializar un proceso aleatorio $\mathcal{N}$ para la exploración de acciones
- 7: Recibir el estado inicial s_1
- 8: **Para** t, k = 1, T **hacer**
- 9: Seleccionar la acción con ruido de exploración $a_k = \pi(s_k|\theta^\pi) + \mathcal{N}_k$
- 10: Ejecutar la acción a_k y observar la recompensa r_k y el nuevo estado $s' = s_{k+1}$
- 11: Guardar la transición (s_k, a_k, r_k, s') en $\mathcal{D}$
- 12: Muestrear un mini-lote de N transiciones (s_i, a_i, r_i, s_{i+1}) del buffer $\mathcal{D}$
- 13: Calcular las acciones objetivo

$$\tilde{a}_{i+1} = \pi'(s_{i+1}|\theta^{\pi'}) + \epsilon, \quad \epsilon \sim \text{clip}(\mathcal{N}(0, \sigma), -c, c)$$

- 14: Calcular los valores objetivos

$$y_i = r_i + \gamma \min_{j=1,2} Q'_j(s_{i+1}, \tilde{a}_{i+1}|\theta_j^{Q'})$$

- 15: Actualizar las funciones Q en un paso de descenso de por gradiente usando

$$L_j = \frac{1}{N} \sum_i \left(y_i - Q_j(s_i, a_i|\theta_j^Q) \right)^2, \quad \text{para } j = 1, 2$$

- 16: **Si** $t \bmod \text{policy_delay} = 0$ **entonces**
- 17: Actualizar la política del actor usando el gradiente ascendente:

$$\nabla_{\theta^\pi} J \approx \frac{1}{N} \sum_i \nabla_a Q_1(s, a|\theta_1^Q)|_{s=s_i, a=\pi(s_i)} \nabla_{\theta^\pi} \pi(s|\theta^\pi)|_{s=s_i}$$

- 18: Actualizar las redes objetivo:

$$\begin{aligned} \theta_j^{Q'} &\leftarrow \tau \theta_j^Q + (1 - \tau) \theta_j^{Q'} \quad \text{para } j = 1, 2 \\ \theta^{\pi'} &\leftarrow \tau \theta^\pi + (1 - \tau) \theta^{\pi'} \end{aligned}$$

- 19: **fin Si**
 - 20: **fin Para**
 - 21: **fin Para**
-

2.2.5.5. Algoritmo Generalized Policy Iteration

*Generalized Policy Iteration*⁶ (GPI) es un algoritmo fundamental de aprendizaje por refuerzo que busca la política óptima mediante iteraciones alternadas de evaluación y mejora de políticas. En el contexto de control óptimo, GPI se utiliza porque permite aproximar soluciones óptimas (asociadas a la ecuación de Hamilton-Jacobi-Bellman - HJB) de forma iterativa y con garantías de convergencia (Barto & Sutton, 2018; Lewis et al., 2012; Lee & Sutton, 2021).

La estructura de actor-crítico mostrada en la Figura 2.5 permite implementar algoritmos orientados a calcular decisiones óptimas. El mecanismo de aprendizaje soportado por esta estructura comprende dos etapas: *Policy Evaluation*⁷, ejecutado por el crítico, y *Policy Improvement*⁸, realizada por el actor.

La etapa de Policy Evaluation se lleva a cabo mediante la observación de los resultados obtenidos al aplicar las acciones de control actuales. Dichos resultados se evalúan a través de una función de costo que cuantifica cuán cercana se encuentra la acción actual al óptimo. Esta función puede definirse en términos de optimalidad, tales como energía mínima, riesgo mínimo o máxima recompensa. Y, con base en las evaluaciones de desempeño, el actor actualiza la política con el propósito de mejorar el valor del rendimiento previo (Lewis et al., 2012; Barto & Sutton, 2018).

En el caso del LQR, es conocido que la función de valor del estado puede expresarse como

$$V(x_k) = \frac{1}{2} x_k^T P x_k \equiv \bar{p}^T \phi(x_k), \quad (2.43)$$

para alguna matriz simétrica $P \in \mathbb{R}^{n \times n}$. El vector $\bar{p} = \text{vech}(P)$ corresponde a la semi-vectorización de P , obtenida al apilar los elementos independientes de la matriz simétrica en un único vector. Por su parte $\phi(x_k)$ el vector de términos cuadráticos que contiene todos los productos independientes de segundo orden formado a partir de los n componentes o

⁶Del inglés, que traducido es iteración generalizada de políticas.

⁷Del inglés, que traducido es evaluación de políticas.

⁸Del inglés, que traducido es mejora de políticas.

estados de x_k , los cuales constituyen una base cuadrática (Lewis et al., 2012).

En general para sistemas discretos no lineales de la forma

$$x_{k+1} = f(x_k) + g(x_k)u_k, \quad (2.44)$$

y de acuerdo al teorema de aproximación de orden superior de Weierstrass, existe una base densa $\{\phi_i(x)\}$ tal que

$$V(x) = \sum_{i=1}^{\infty} w_i \phi_i(x) = \sum_{i=1}^L w_i \phi_i(x) + \sum_{i=L+1}^{\infty} w_i \phi_i(x) = W^T \phi(x) + \epsilon_L(x), \quad (2.45)$$

donde el vector base $\phi(x) = [\phi_1(x), \phi_2(x), \dots, \phi_L(x)] : \mathbb{R}^n \rightarrow \mathbb{R}^L$, y $\epsilon_L(x)$ representa el error de aproximación, el cual converge uniformemente a cero cuando $L \rightarrow \infty$. Además, en la comunidad de redes neuronales (NN), L es el número de neuronas en la capa oculta, $\phi_i(x)$ es la función de activación, w_i denota los pesos de la red neuronal, W es la matriz de pesos, x es el vector de estados, y γ el factor de descuento (Lewis et al., 2012).

El Algoritmo 3 implementa el esquema de GPI. Este procedimiento puede llevarse a cabo mediante técnicas de identificación de sistemas y no requiere explícitamente el modelo dinámico en su formulación. Por lo tanto, abre la posibilidad de ser implementado en tiempo real almacenando las transiciones $(x_k, x_{k+1}, r(x_k, h_j(x_k)))$ y aplicando algún algoritmo de regresión para la estimación de parámetros W (Lewis et al., 2012).

Algoritmo 3 Control óptimo basado en Generalized Policy Iteration (GPI)

- 1: Seleccionar cualquier política admisible $h_0(x_k)$.
- 2: **Para** $j = 1$, hasta la convergencia **hacer**
- 3: **Policy Evaluation:**
- 4: Determinar la solución W_{j+1} por mínimos cuadrados dado

$$W_{j+1}^T (\phi(x_k) - \gamma \phi(x_{k+1})) = r(x_k, h_j(x_k)) = Q(x_k) + h_j^T(x_k) R h_j(x_k). \quad (2.46)$$

- 5: **Policy Improvement:**
- 6: Determinar una política mejorada usando

$$h_{j+1}(x_k) = -\frac{\gamma}{2} R^{-1} g^T(x_k) \nabla \phi^T(x_{k+1}) W_{j+1}. \quad (2.47)$$

- 7: **fin Para**
-

La relación (2.46) es una ecuación escalar, donde $W_{j+1} \in \mathbb{R}^L$ tiene L elementos. En el instante $k + 1$ se mide el estado previo x_k , la señal de control $u_k = h_j(x_k)$ y el estado siguiente x_{k+1} , y se calcula la recompensa resultante $r(x_k, h_j(x_k))$. Este procedimiento se repite hasta disponer de al menos L ecuaciones; en este punto, puede determinarse W_{j+1} mediante una solución por mínimos cuadrados, o alguna de sus variantes.

Existen también otras variantes del algoritmo GPI que dependen explícitamente del modelo del sistema en su formulación. Dado el sistema discreto

$$x_{k+1} = Ax_k + Bu_k, \quad (2.48)$$

el algoritmo GPI orientado a la solución de la ecuación de Riccati (ecuación 2.6), propuesto por Filipović et al. (2024), se presenta en el Algoritmo 4. En este esquema, $\lambda \in (0, 1]$ es un parámetro de restricción que confina la magnitud de los valores propios en lazo cerrado al rango establecido, y T_s es el periodo de muestreo.

Algoritmo 4 Generalized Policy Iteration basado en modelo (MGPI)

- 1: Inicializar las matrices $Q = Q^T \geq 0; R = R^T \geq 0; K_j = K_0$ (no necesariamente estabilizadora);

$$P_j = P_0 = I; \epsilon > 0; \lambda \in (0, 1); T_s, j = 0; i = 0, 1, 2, \dots, l.$$

- 2: **Para** $j = 0, \|K_{j+1} - K_j\| < \epsilon$ **hacer**
- 3: **Policy Evaluation:**
- 4: Determinar la matriz solución $P = P_{j+1}$
- 5: **Para** $i = 0, l$ **hacer**

$$P_j^{i+1} = (\lambda^{-T_s} A - \lambda^{-T_s} B K_j)^T P_j^i (\lambda^{-T_s} A - \lambda^{-T_s} B K_j) + Q + K_j^T R K_j.$$

- 6: **fin Para**
- 7: Actualizar la matriz solución

$$P_{j+1} = P_j^l$$

- 8: **Policy Improvement:**
- 9: Determinar una política mejorada usando

$$K_{j+1} = \lambda^{-2T_s} (R + \lambda^{-2T_s} B^T P_{j+1} B)^{-1} B^T P_{j+1} A.$$

- 10: **fin Para**
-

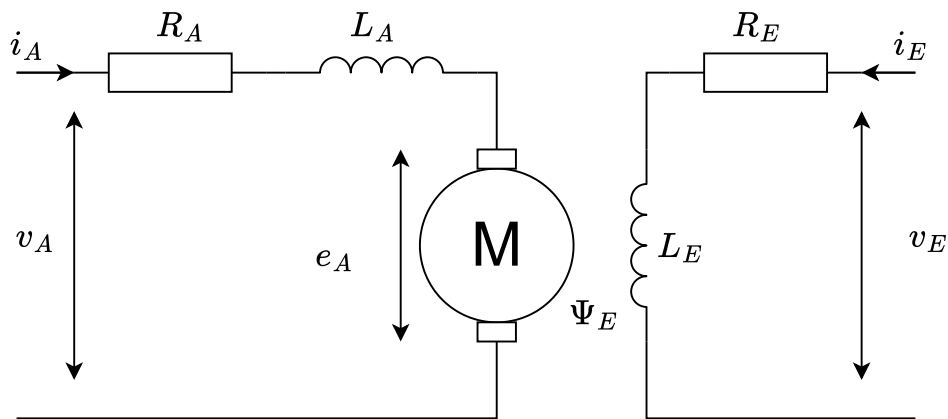
2.2.6. Control óptimo adaptativo.

A causa de su arraigo con el control óptimo, el aprendizaje reforzado se puede entender, en el contexto de la teoría de control, como una clase o categoría de métodos para el diseño de controladores adaptativos que aprenden en tiempo real las soluciones a problemas de optimización prescritos por el usuario. A esto se define como control óptimo adaptativo. Las arquitecturas de este tipo de controladores se basan en el esquema actor-crítico presentado en la Figura 2.5 (Lewis et al., 2012). Y algoritmos como el DDPG y el TD3 pueden ser usados para el diseño y sintonización de estos controladores (Tüfenkçi et al., 2023; Tufenkci et al., 2023).

2.2.7. Motor de corriente continua con devanado de campo

Un motor de corriente continua (DC) con devanado de campo es una máquina eléctrica que recibe energía eléctrica y la transforma en energía mecánica. En la Figura 2.8 se tiene el circuito equivalente del motor.

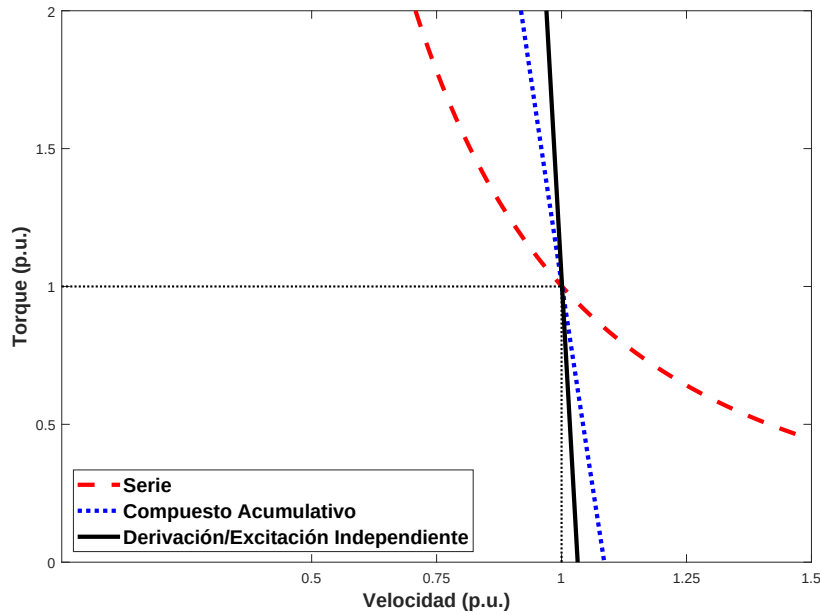
Figura 2.8: Circuito equivalente del motor DC



La estructura de un motor DC consiste principalmente de un estator, rotor y conmutador. Los motores DC con devanado de campo usan un devanado en el estator, en lugar de imanes permanentes, para generar el campo magnético. La clasificación de estas

máquinas es de acuerdo al tipo de excitación, a saber: excitación en serie, derivación, compuesta e independiente (Chapman, 2012). Estos arreglos han sido aplicados, por ejemplo, en el control de velocidad de vehículos eléctricos donde la demanda es mayor frente a aplicaciones industriales. La Figura 2.9 muestra las curvas características de las configuraciones de los motores DC. La configuración en derivación y excitación independiente son adecuados en aplicaciones demandantes de buena regulación y buen ajuste de velocidad. Esta característica, es en particular gracias la relación lineal entre la velocidad y el torque que no existe en los demás arreglos (Chau, 2015).

Figura 2.9: Características Torque-Velocidad normalizadas



Nota. Las curvas son referenciales. Traducida y adaptada de (Chau, 2015).

En el contexto de control de motores DC, la señal de control es el voltaje de armadura v_A , y la corriente de armadura i_A puede ser expresada en función de v_A y la fuerza contraelectromotriz e_A , es decir $i_A = f(v_A, e_A)$ (Traue et al., 2019). Esto implica que, al optimizar v_A en la formulación del problema de optimización, la función de costo—la cual penaliza grandes variaciones de la señal de control v_A —, también optimiza indirectamente i_A , por tanto, la potencia consumida

$$P = v_{A-RMS} \cdot i_{A-RMS}. \quad (2.49)$$

2.2.7.1. Modelo del motor DC con devanado de campo

En esencia, todos los tipos de motores DC pueden ser representados por un circuito equivalente de armadura, y un circuito de campo o un campo magnético constante si se trata de un motor de imán permanente (Chau, 2015). Tal representación se observa en la Figura 2.8. Las ecuaciones que gobiernan el funcionamiento de estas máquinas eléctricas se presentan a continuación:

$$v_A = R_A i_A + L_A \frac{d}{dt} i_A + e_A, \quad (2.50)$$

$$e_A = C_m \Psi_E \omega_m, \quad (2.51)$$

$$M_{Mi} = C_m \Psi_E i_A, \quad (2.52)$$

$$v_E = R_E i_E + L_E \frac{d}{dt} i_E, \quad (2.53)$$

$$\Psi_E = f(i_E) \ ; \ \Psi_E(0) = 0, \quad (2.54)$$

$$M_{Mi} - M_L - M_b = \Theta_M \frac{d}{dt} \omega_m. \quad (2.55)$$

Las ecuaciones (2.50), (2.51) y (2.52) son del circuito de armadura, donde R_A es la resistencia de armadura, L_A la inductancia de armadura, i_A la corriente de armadura, e_A la fuerza contraelectromotriz (FEM), C_m la constante de la máquina, Ψ_E el acoplamiento inductivo de campo o excitación, ω_m la velocidad del motor. El circuito de excitación o campo queda descrito por las ecuaciones (2.53) y (2.54), donde R_E es la resistencia de excitación, L_E la inductancia de excitación, i_E la corriente de excitación, aquí se considera Ψ_E como una función de la corriente i_E . Finalmente, la ecuación (2.55) describe las relaciones mecánicas del motor, donde M_{Mi} es el par del entrehierro, M_L el par de carga, M_b el par generado por fricción, Θ_M el momento de inercia del rotor.

2.2.8. Métricas de desempeño

El desempeño de los controladores se evalúa cuantitativamente por medio de métricas o índices de desempeño. Estos indicadores posibilitan el análisis del rendimiento de los controladores y establecen bases para análisis comparativos. Algunas de estas métricas usadas por Mamani et al. (2024) y Chirme-Sisa et al. (2024) se presentan a continuación.

Tiempo de establecimiento t_s : Es la duración que le toma a un sistema en alcanzar y mantener un valor de estado estacionario dentro de una banda de tolerancia típicamente del 5 % o 2 % respecto de su valor final o referencia.

Sobreimpulso M_p : Es la medida que cuantifica cuanto excede como máximo la salida del sistema a su valor de estado estable o referencia (setpoint).

$$M_p(\%) = \frac{y_{max} - y_{ss}}{y_{ss}} \times 100\%. \quad (2.56)$$

Desviación estándar del error σ_e : Mide la dispersión en el tiempo del valor absoluto del error de seguimiento $|e(t)| = |y_{ref} - y(t)|$ respecto de su media μ_e . Para mediciones discretas se tiene:

$$\sigma_e = \sqrt{\frac{1}{N} \sum_{k=1}^N [|e(k)| - \mu_e]^2}. \quad (2.57)$$

Valor RMS de la señal de control: Mide el consumo energético del controlador.

$$u_{RMS} = \sqrt{\frac{1}{N} \sum_{k=1}^N u(k)^2}. \quad (2.58)$$

Error cuadrático integral (ISE): Penalización cuadrática del error de seguimiento.

$$ISE = \sum_{k=1}^N e(k)^2. \quad (2.59)$$

Error absoluto integral (IAE): Penalización lineal del error de seguimiento

$$IAE = \sum_{k=1}^N |e(k)|. \quad (2.60)$$

Error absoluto integral de tiempo (ITAE): Penalización ponderada de errores persistentes en el tiempo.

$$ITAE = \sum_{k=1}^N k \cdot |e(k)|. \quad (2.61)$$

Error cuadrático integral de tiempo (ITSE): Penalización ponderada temporal del error cuadrático.

$$ITSE = \sum_{k=1}^N k \cdot e(k)^2. \quad (2.62)$$

Potencia consumida:

$$P = v_{A-RMS} \cdot i_{A-RMS}. \quad (2.63)$$

CAPÍTULO 3:

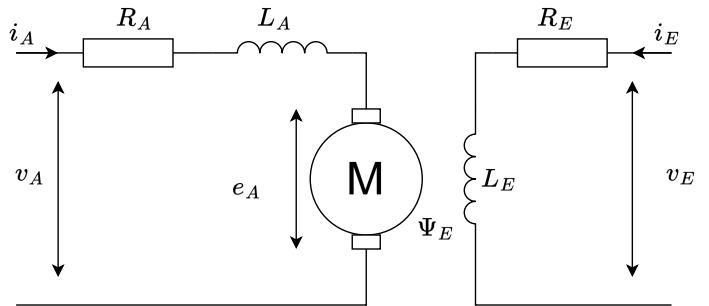
MODELAMIENTO MATEMÁTICO Y ESTIMACIÓN DE PARÁMETROS

3.1. Modelo matemático del motor DC

Figura 3.1: El motor DC con devanado de campo



(a) Motor DC (Lucas-Nuelle, 2023)



(b) Circuito equivalente del motor DC

Nota. La imagen de la izquierda se reproduce de la documentación técnica de Lucas Nülle.

Según Traue et al. (2019), el modelo matemático de todos los tipos de motores DC se derivan del motor de excitación separada o independiente. Este se compone de un circuito de excitación y armadura como se muestra en la Figura 3.1b. El sistema de ecuaciones diferenciales en (3.1) se deducen de las ecuaciones (2.50–2.55). La Tabla 3.1 recopila las variables y símbolos matemáticos usados en el modelo matemático.

$$\begin{pmatrix} \frac{d}{dt}i_A \\ \frac{d}{dt}i_E \\ \frac{d}{dt}\omega_m \end{pmatrix} = \begin{pmatrix} \frac{1}{L_A} (v_A - R_A i_A - C_m \Psi_E \omega_m) \\ \frac{1}{L_E} (v_E - R_E i_E) \\ \frac{1}{\Theta_M} (C_m \Psi_E i_A - M_L - M_b) \end{pmatrix}. \quad (3.1)$$

Tabla 3.1: Definición de símbolos del modelo matemático del motor DC

Símbolo	Descripción	Unidades
v_A	Voltaje de armadura	[V]
i_A	Corriente de armadura	[A]
R_A	Resistencia de armadura	[Ω]
L_A	Inductancia de armadura	[H]
e_A	Fuerza contraelectromotriz	[V]
C_m	Constante de máquina	[1/rad]
Ψ_E	Flujo de excitación	[Wb, V·s]
ω_m	Velocidad del motor	[rad/s]
v_E	Voltaje del circuito de campo	[V]
i_E	Corriente de campo	[A]
R_E	Resistencia de campo	[Ω]
L_E	Inductancia de campo	[H]
M_{Mi}	Par del entrehierro	[N·m]
M_b	Par de fricción	[N·m]
M_L	Par de carga	[N·m]
Θ_M	Momento de inercia del rotor	[kg·m ²]

3.1.1. Modelo no lineal

El modelo no lineal se obtiene de la ecuación (3.1) siguiendo la dinámica de la planta dada por

$$\dot{x} = f(x) + g(x)u. \quad (3.2)$$

La Figura 3.1b evidencia que la velocidad del motor se controla por el voltaje v_A , manteniendo un voltaje de excitación constante v_E frente a una carga M_L . De aquí, el vector de entradas es $u = (v_A \ v_E \ M_b + M_L)^T$ (siendo M_b el torque por fricción), dado los estados i_A , i_E y ω_m , dando lugar al modelo no lineal en la ecuación (3.3).

$$\begin{pmatrix} \frac{d}{dt} i_A \\ \frac{d}{dt} i_E \\ \frac{d}{dt} \omega_m \end{pmatrix} = \begin{pmatrix} -\frac{1}{L_A} (R_A i_A + C_m \Psi_E \omega_m) \\ -\frac{R_E}{L_E} i_E \\ \frac{C_m \Psi_E}{\Theta_M} i_A \end{pmatrix} + \begin{pmatrix} \frac{1}{L_A} & 0 & 0 \\ 0 & \frac{1}{L_E} & 0 \\ 0 & 0 & -\frac{1}{\Theta_M} \end{pmatrix} \begin{pmatrix} v_A \\ v_E \\ M_b + M_L \end{pmatrix}. \quad (3.3)$$

3.1.2. Modelo lineal

El circuito de excitación del motor DC se alimenta mediante una fuente de voltaje continua v_E , considerando esto, y según Hwang & Gu (2020) se asume que el circuito de excitación se encuentra en estado estable; por lo tanto, la corriente i_E es constante $i_E = i_{Ess} = cte$. Dadas estas condiciones, i_{Ess} se calcula a partir de la ecuación (3.3) como sigue.

$$\begin{aligned} i_E = i_{Ess} \Rightarrow \frac{d}{dt} i_E = 0 &= \frac{1}{L_E} v_E - \frac{1}{L_E} (R_E i_{Ess}), \\ \Rightarrow i_{Ess} &= \frac{v_E}{R_E}. \end{aligned} \quad (3.4)$$

Siendo el flujo de excitación $\Psi_E = f(i_E)$ una función de i_E (ecuación (2.54)), entonces, en estado estable

$$\Psi_E(i_{Ess}) = \Psi_{Ess}. \quad (3.5)$$

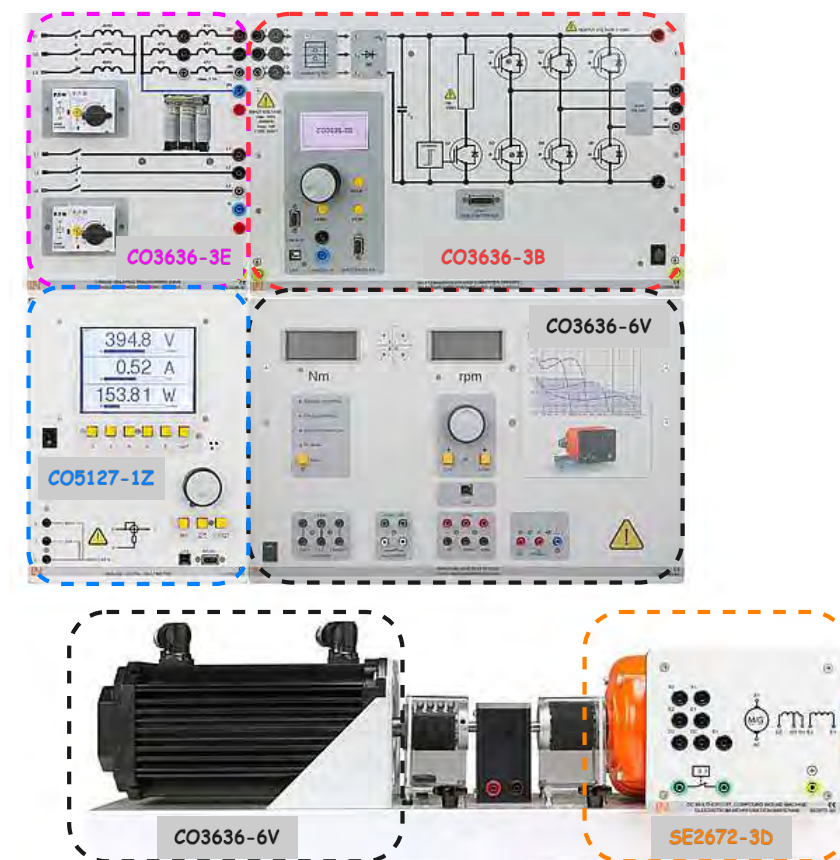
Las ecuaciones (3.4) y (3.5) en (3.3) sugieren que en estado estacionario, tanto i_E y $\Psi(i_E)$ son constantes para una fuente de alimentación de corriente directa v_E constante. Por tanto, la ecuación de estados no lineal en (3.3) se puede reducir a la ecuación (3.6); un modelo lineal de dos estados $x = (\omega_m \ i_A)^T$ con entradas $u = (M_b + M_L \ v_A)^T$.

$$\begin{pmatrix} \frac{d}{dt} \omega_m \\ \frac{d}{dt} i_A \end{pmatrix} = \begin{pmatrix} 0 & \frac{C_m \Psi_{Ess}}{\Theta_M} \\ -\frac{C_m \Psi_{Ess}}{L_A} & -\frac{R_A}{L_A} \end{pmatrix} \begin{pmatrix} \omega_m \\ i_A \end{pmatrix} + \begin{pmatrix} -\frac{1}{\Theta_M} & 0 \\ 0 & \frac{1}{L_A} \end{pmatrix} \begin{pmatrix} M_b + M_L \\ v_A \end{pmatrix}. \quad (3.6)$$

3.2. Equipamiento y el sistema de servoaccionamiento

A fin de llevar a cabo los experimentos y pruebas del funcionamiento del motor y la obtención de parámetros, se hace uso del sistema de servoaccionamiento Lucas Nülle (LNSD *system*). Los equipos del sistema de servoaccionamiento Lucas Nülle y su descripción se muestran en la Figura 3.2 y la Tabla 3.2, respectivamente.

Figura 3.2: Sistema de servoaccionamiento Lucas Nülle (LNSD system)



Nota. Tomada y adaptada de Linnertz (2016) ©2016 Lucas Nülle GmbH.

- **CO3636-3B:** Convertidor de potencia con IGBTs¹ de 1 kVA a 240 V DC. Además incorpora unidades de control, adquisición de datos, y comunicación basados en un microcontrolador (MCU) con capacidades DSP—el TI TMS320F28335 (serie C2000 Delfino)—que permite comunicación e integración con MATLAB/Simulink.

¹Del inglés *Insulated-Gate Bipolar Transistor*.

Tabla 3.2: Descripción del equipamiento del sistema LNSD

Equipo	Descripción del equipo
CO3636-3B	Convertidor autoconmutado/controlador de potencia con IGBTs
CO3636-3D	Toolbox de MATLAB (r2016a) para electrónica de potencia
CO3636-3E	Transformador de aislamiento trifásico 300 VA
SE2663-3T	Encoder incremental de 1024 pulsos - 6000 RPM max
CO5127-1Z	Multímetro analógico/digital y vatímetro
CO3636-6V	Banco de pruebas de servomotor para máquinas de 0.3 kW
CO3212-5Q	Transformador y rectificador monofásico
SE2672-3D	Motor de corriente continua de 0.3 kW

- **CO3636-3D:** Toolbox de MATLAB que permite el despliegue de proyectos de MATLAB/Simulink (r2016a) en el equipo CO3636-3B mediante un adaptador USB-JTAG. Comprende también la comunicación serial sobre USB con MATLAB, utilizada para el intercambio de datos de medición y configuración de parámetros.
- **CO3636-3E:** Transformador de aislamiento trifásico: 300 VA de 3×400 V, 50/60 Hz a 3×94 V con toma central de 47 V.
- **SE2663-3T:** Encoder incremental óptico de dos canales y un canal de referencia, con resolución máxima de 5000 pulsos por revolución.
- **CO5127-1Z:** Multímetro analógico/digital y vatímetro de potencia activa y reactiva y factor de potencia.
- **CO3636-6V:** El banco de pruebas de servomotor comprende un controlador digital, un freno (servomotor) y el software ActiveServo. Mide la velocidad del motor en prueba, y puede generar torque de carga. ActiveServo permite el registro de las características de los motores y los puntos de operación dinámicos y estáticos.
- **CO3212-5Q:** Transformador y rectificador monofásico con salida variable de 0-230 V, 2 A AC²/DC ; 0-30 V, 10 A AC/DC.
- **SE2672-3D:** Motor DC con devanado de campo: 220 V tensión de inducido, 220 V tensión de campo a 0.1 A, 0.3 kW, 1/1.2/1.2 A, 2000/1800/2000 RPM (configuración en derivación, compuesto y serie, respectivamente).

²Del inglés *Alternating Current*.

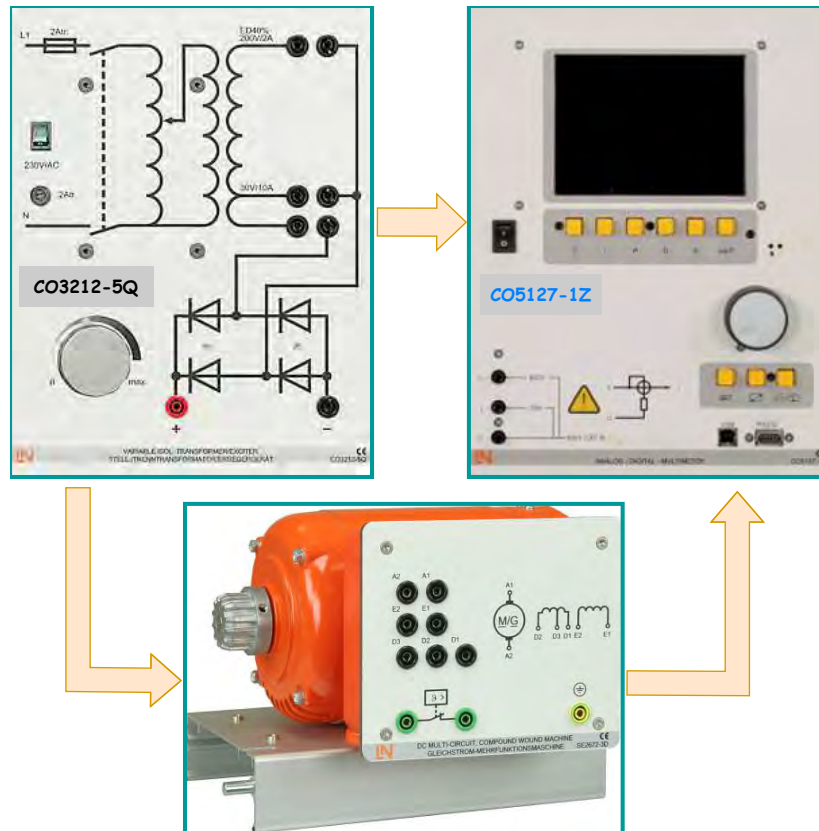
3.3. Obtención de parámetros y funciones del modelo

A partir de la Tabla 3.1 y la Figura 3.1b, los parámetros del motor DC se determinan a través de experimentos detallados en las secciones posteriores. Los experimentos y arreglos de los circuitos presentados se fundamentan en los trabajos de Linnertz (2016), Kuczmann (2024), Alejandro-Sanjines et al. (2023) y Meneses Morales (2013).

3.3.1. Parámetros eléctricos

La obtención de los parámetros eléctricos R_A , L_A , R_E y L_E se logra realizando mediciones independientes en los circuitos de campo y armadura. Los equipos usados se muestran en la Figura 3.3.

Figura 3.3: Equipos destinados a la medición de los parámetros eléctricos

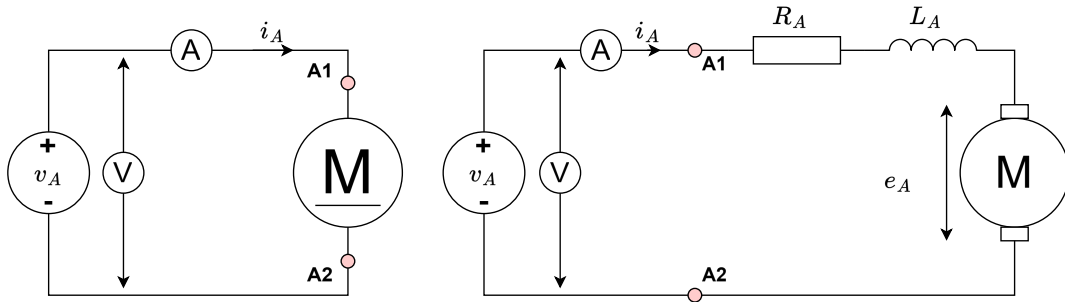


Nota. Tomada y adaptada de Linnertz (2016) ©2016 Lucas Nülle GmbH.

3.3.1.1. Resistencia e inductancia de armadura R_A y L_A

El circuito de campo se deja sin alimentar, y se inyecta voltaje continuo en la armadura a fin de medir la resistencia R_A siguiendo la configuración de los equipos de medición, voltímetro y amperímetro, como se muestra en la Figura 3.4.

Figura 3.4: Circuito de medición para estimar R_A

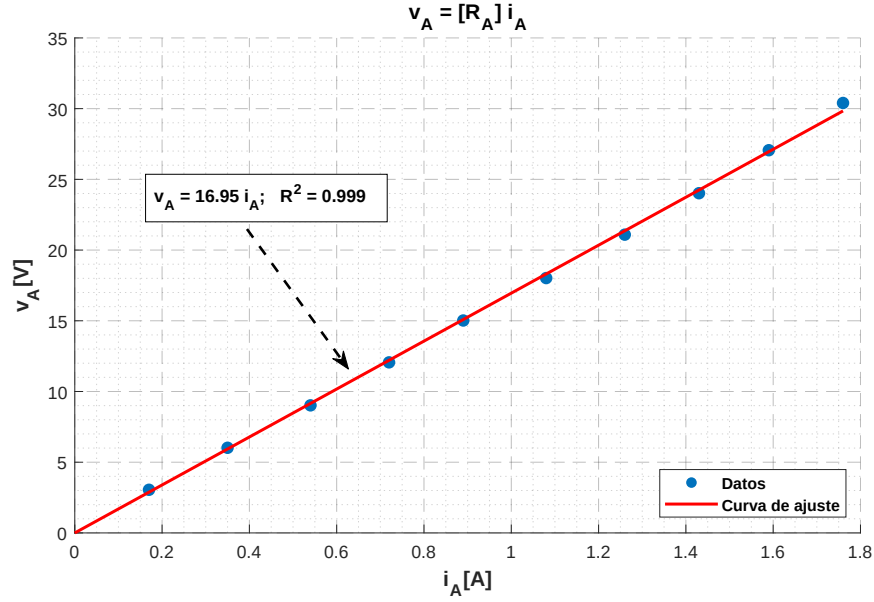


Si el circuito de campo no se alimenta se tiene $e_A = 0$ V (el rotor no presenta movimiento). En régimen permanente de corriente continua (CC), L_A actúa como un cortocircuito; entonces $v_A = R_A i_A$, dada esta condición, se adquieren las mediciones de voltaje y corriente, las cuales se recopilan en la Tabla 3.3 y sirven para el análisis de regresión lineal $v_A = m \cdot i_A$. La Figura 3.5 muestra la proporcionalidad esperada entre v_A e i_A , donde la pendiente de la curva de regresión corresponde con el valor de R_A .

Tabla 3.3: Mediciones de v_A , i_A y estimación de R_A

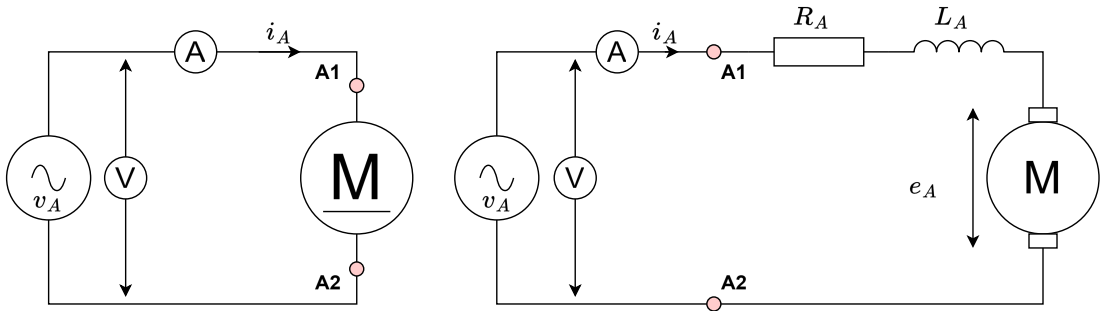
N°	v_A [V]	i_A [A]	R_A [Ω](v_A/i_A)
1	3.05	0.17	17.94
2	6.02	0.35	17.20
3	9.02	0.54	16.70
4	12.06	0.72	16.75
5	15.02	0.89	16.87
6	18.02	1.08	16.69
7	21.09	1.26	16.69
8	24.02	1.43	16.80
9	27.06	1.59	17.02
10	30.40	1.76	17.27
Regresión lineal $v_A = m \cdot i_A$ donde $m = R_A$			▷ 16.95

Figura 3.5: Datos experimentales y curva de regresión lineal $[v_A] \propto [i_A]$



L_A se estima indirectamente conociendo el valor de R_A , y siguiendo el circuito de la Figura 3.6. Se alimenta la armadura con voltaje de corriente alterna a $f = 60$ Hz ($\omega = 2\pi f$). El circuito se reduce a R_A y L_A en serie. El análisis en AC³ del circuito se define en la ecuación (3.7), y de ella se obtienen las ecuaciones (3.8) y (3.9) para el análisis de regresión y el cálculo de L_A , respectivamente.

Figura 3.6: Circuito de medición para estimar L_A



$$V_A(j\omega) = (R_A + j\omega L_A) I_A(j\omega), \quad (3.7)$$

$$|V_A(j\omega)| = |Z_A(j\omega)| \cdot |I_A(j\omega)| = \sqrt{[R_A^2 + (\omega L_A)^2]} |I_A(j\omega)|, \quad (3.8)$$

$$L_A = \sqrt{\frac{|Z_A(j\omega)|^2 - R_A^2}{\omega^2}} = \sqrt{\frac{|V_A(j\omega)|^2 - R_A^2 |I_A(j\omega)|^2}{\omega^2}}. \quad (3.9)$$

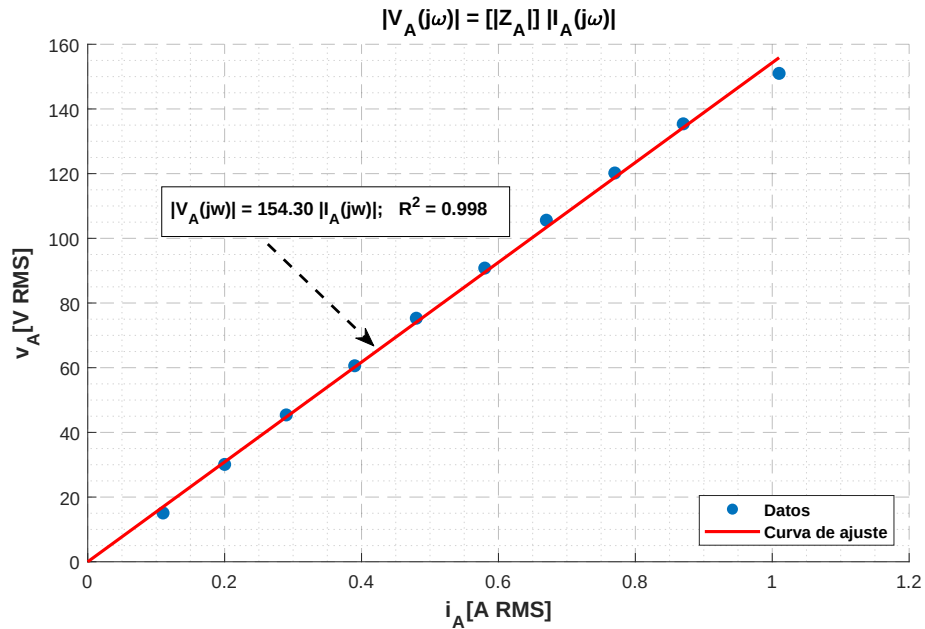
³Las funciones en letras mayúsculas denotan la Transformada de Fourier de sus contrapartes en el dominio temporal. Por ejemplo $\mathcal{F}[v_A(t)] = V_A(j\omega)$, $\mathcal{F}[i_A(t)] = I_A(j\omega)$

En (3.8) se expresa la relación proporcional entre $|V_A(j\omega)|$ y $|I_A(j\omega)|$. La expresión dentro de la raíz se considera constante debido a que la frecuencia de red es fija ($f = 60Hz$), y además R_A y L_A mantienen valores estables durante intervalos de tiempo significativos antes de sufrir variaciones a causa de averías, desgaste, etc. (Li et al., 2024).

Tabla 3.4: Mediciones de v_A , i_A en RMS y estimación de L_A

N°	v_A [V RMS]	i_A [A RMS]	L_A [mH]
1	15.08	0.11	360.85
2	30.10	0.20	396.67
3	45.40	0.29	412.83
4	60.60	0.39	409.71
5	75.30	0.48	413.69
6	90.80	0.58	412.83
7	105.60	0.67	415.65
8	120.20	0.77	411.63
9	135.40	0.87	410.37
10	151.00	1.01	394.02
Regresión lineal $ V_A(j\omega) = Z_A \cdot I_A(j\omega) $; $ Z_A = \sqrt{R_A^2 + \omega^2 L_A^2}$			$\triangleright 406.82$

Figura 3.7: Datos experimentales y curva de regresión lineal $[|V_A(j\omega)|] \propto [|I_A(j\omega)|]$



A partir de las mediciones en la Tabla 3.4 y la curva de regresión en la Figura 3.7 se estima L_A usando la ecuación (3.9), donde $|Z_A| = |V_A(j\omega)|/|I_A(j\omega)|$ es la pendiente de la curva de ajuste lineal.

3.3.1.2. Resistencia e inductancia de excitación R_E y L_E

Los criterios que definen el proceso y los métodos de estimación de R_A y L_A en la sección 3.3.1.1 son empleados también con el fin de estimar R_E y L_E .

Figura 3.8: Circuito de medición para estimar R_E

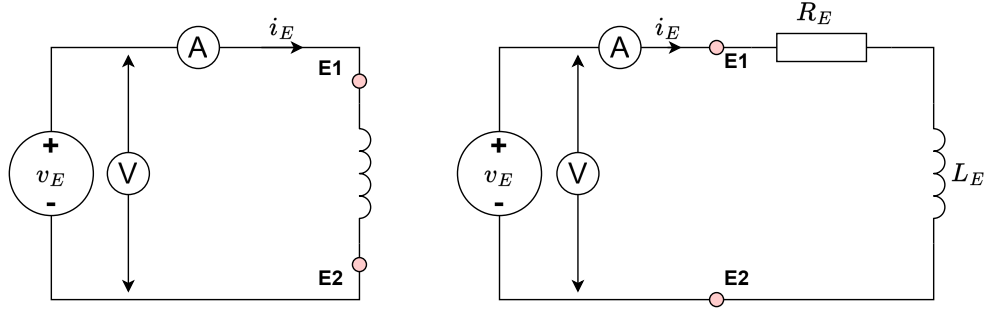


Tabla 3.5: Mediciones de v_E , i_E y estimación de R_E

N°	v_E [V]	i_E [mA]	R_E [Ω] (v_E/i_E)
1	22.30	11.27	1978.70
2	44.30	22.36	1981.22
3	66.10	33.34	1982.60
4	88.60	44.31	1999.55
5	110.60	55.10	2007.26
6	132.20	65.50	2018.32
7	154.50	75.90	2035.57
8	176.80	86.00	2055.81
9	198.80	95.80	2075.16
10	220.80	104.40	2114.94
Regresión lineal $v_E = m \cdot i_E$ dado $m = R_E$			$\triangleright$ 2059.39

Siguiendo el circuito de la Figura 3.8, de los datos registrados en la Tabla 3.5 y la curva de regresión en la Figura 3.9, la resistencia R_E se puede estimar directamente; este parámetro es la pendiente de la recta.

El circuito de campo es en esencia R_E y L_E en serie. Esto permite usar las ecuaciones (3.7–3.9) en la obtención de L_E ; basta con reemplazar $V_A(j\omega)$, $I_A(j\omega)$, R_A y L_A por $V_E(j\omega)$, $I_E(j\omega)$, R_E y L_E , respectivamente. La configuración de la Figura 3.10 proporciona una forma de adquirir mediciones de voltaje y corriente de campo a fin de estimar L_E .

Figura 3.9: Datos experimentales y curva de regresión lineal $[v_E] \propto [i_E]$

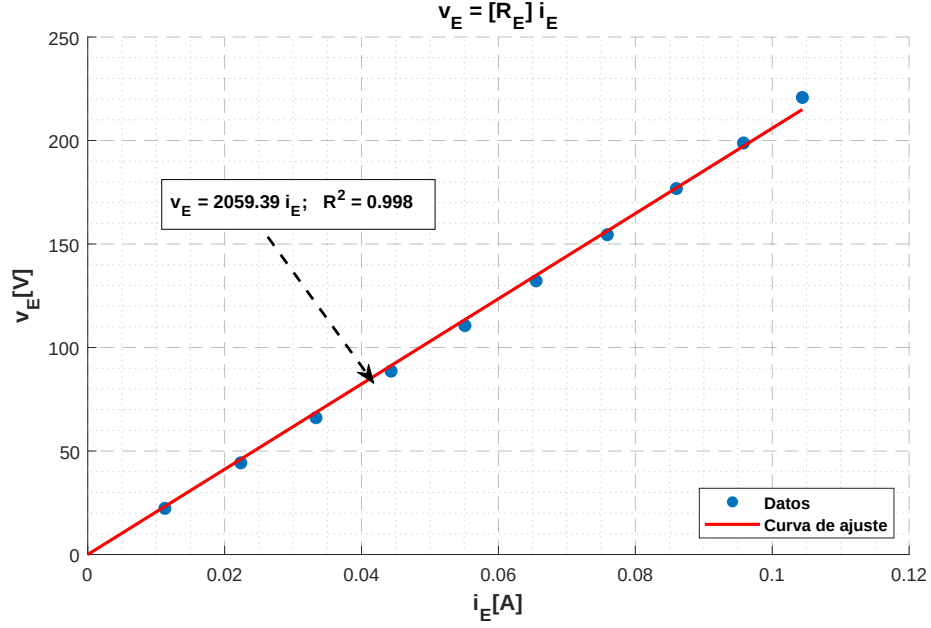
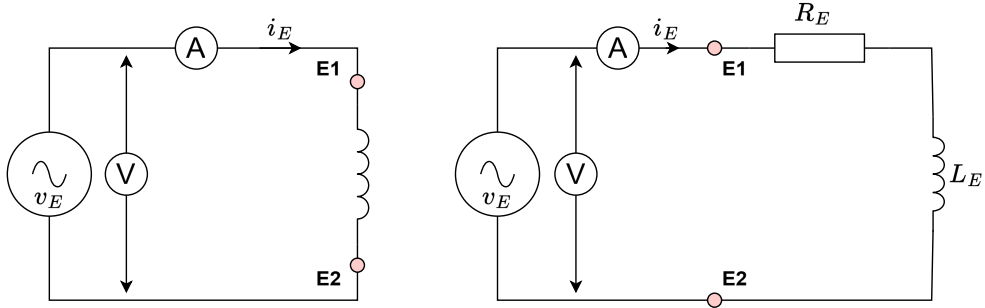


Figura 3.10: Circuito de medición para estimar L_E

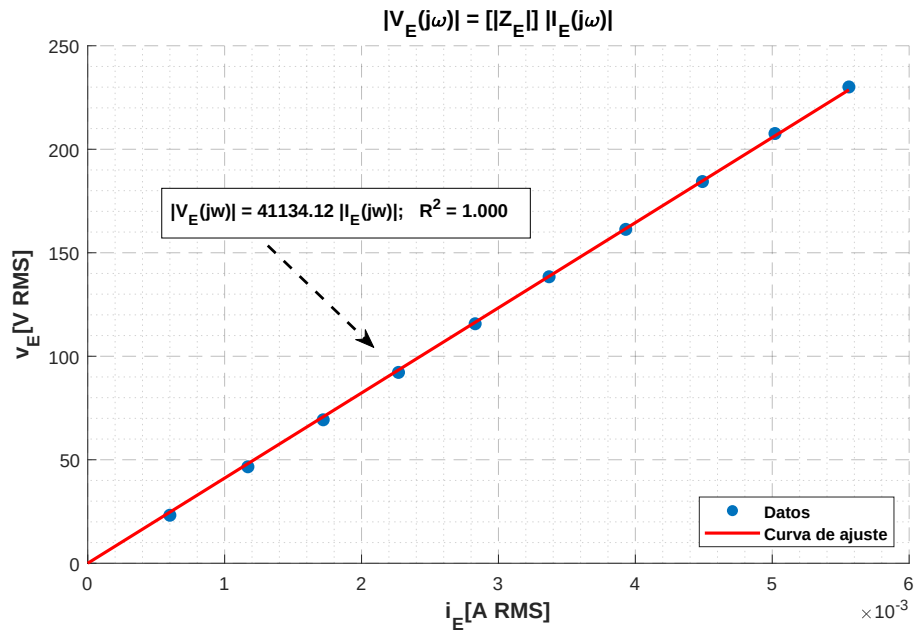


La Tabla 3.6 lleva el registro de las mediciones de v_E y i_E en RMS. Por las razones expuestas en la sección previa, $|Z_E| = \sqrt{R_E^2 + \omega^2 L_E^2}$ se considera constante. Por tanto, conociendo el valor de R_E es posible hallar el valor de L_E . De la Figura 3.11, la pendiente de la recta es el valor estimado de $|Z_E| = |V_E(j\omega)|/|I_E(j\omega)|$, y usando la ecuación (3.9), con los cambios necesarios, se logra calcular el valor de L_E .

Tabla 3.6: Mediciones de v_E , i_E en RMS y estimación de L_E

N°	v_E [V RMS]	i_E [mA RMS]	L_E [H]
1	23.20	0.60	102.42
2	46.60	1.17	105.51
3	69.30	1.72	106.73
4	92.20	2.27	107.60
5	115.70	2.83	108.31
6	138.40	3.37	108.80
7	161.30	3.93	108.73
8	184.40	4.49	108.80
9	207.60	5.02	109.56
10	230.10	5.56	109.64
Regresión lineal $ V_E(j\omega) = Z_E \cdot I_E(j\omega) $; $ Z_E = \sqrt{R_E^2 + \omega^2 L_E^2}$			$\triangleright 108.97$

Figura 3.11: Datos experimentales y curva de regresión lineal $[|V_E(j\omega)|] \propto [|I_E(j\omega)|]$



3.3.2. Función del flujo de excitación Ψ_E

Los circuitos de campo y de armadura están acoplados magnéticamente a través del flujo Ψ_E . Este, a su vez, es una función de la corriente de campo i_E y cumple

$$\Psi_E(i_E) = \begin{cases} f(i_E) & ; i_E > 0 \\ 0 & ; i_E = 0. \end{cases} \quad (3.10)$$

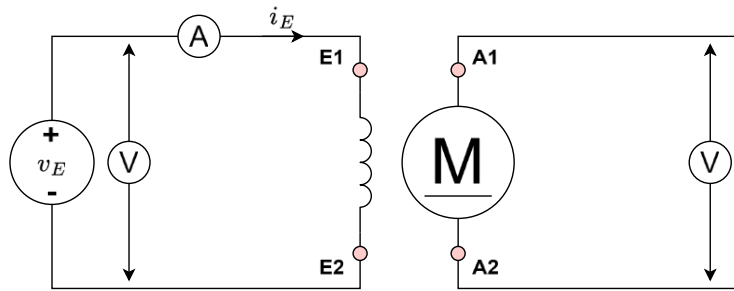
La relación entre i_E y Ψ_E es no lineal, entonces, tomando como referencia el trabajo de Linnertz (2016), $\Psi_E(i_E)$ se aproxima por medio de una función cuadrática en (3.11).

$$\Psi_E(i_E) = f(i_E) = a \cdot i_E^2 + b \cdot i_E + c. \quad (3.11)$$

En el modelo matemático estudiado, la constante de máquina C_m y el flujo Ψ_E están intrínsecamente acoplados, lo que implica que no se pueden desvincular directamente. El objetivo, por tanto, es estimar la función $f(i_E)$ normalizada a la unidad cuando i_E toma su valor nominal.

El circuito de medición y el montaje experimental se muestran en las Figuras 3.12 y 3.13, respectivamente. Siguiendo los arreglos mostrados, el equipo CO3636-6V se configura para activar el servomotor a una velocidad constante de 2000 RPM, esta velocidad se transmite al motor DC mediante un acople de los ejes, lo cual hace que el motor DC funcione como un generador (accionado externamente). La idea es adquirir mediciones del voltaje de armadura v_A en función de la variación de la corriente de campo i_E , ajustando la tensión de excitación v_E y manteniendo la velocidad del rotor en 2000 RPM.

Figura 3.12: Configuración experimental para estimar $\Psi_E(i_E)$

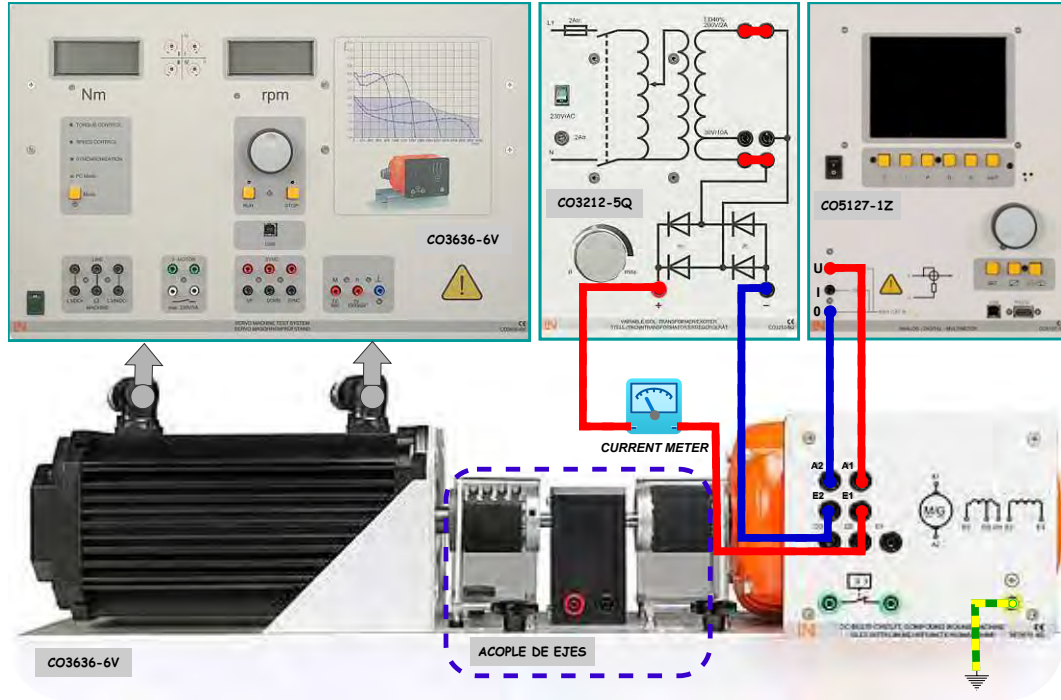


Del circuito de medición (Figura 3.12) y las relaciones (2.50) y (2.51). Si el circuito inducido se deja sin carga, no existirá flujo de corriente; por esto se cumple que

$$v_A = R_A i_A + L_A \frac{d}{dt} i_A + e_A \quad ; \text{ si } i_A = 0 \Rightarrow v_A = e_A$$

$$\therefore v_A = e_A = C_m \Psi_E \omega_m. \quad (3.12)$$

Figura 3.13: Montaje de los equipos para estimar $\Psi_E(i_E)$



Nota. Tomada y adaptada de Linnertz (2016) ©2016 Lucas Nülle GmbH.

La máquina gira a una velocidad constante asignada de 2000 RPM ($\omega_m = 209.440 \text{ rad/s}$) y dado que C_m es también constante, v_A y Ψ_E mantienen una relación proporcional de acuerdo a la ecuación (3.12). Aprovechando esto, se omiten C_m y ω_m para este experimento y se asume que

$$\Psi_E(i_E) = v_{Anorm}(i_E) = \frac{v_A(i_E)}{v_{Amax}}, \quad (3.13)$$

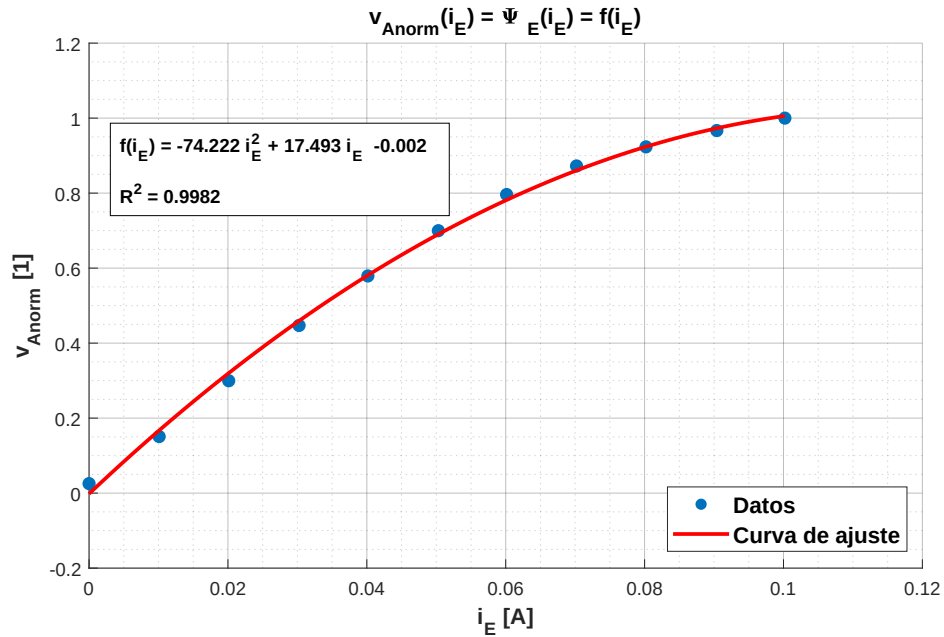
donde v_{Anorm} es el voltaje de armadura normalizado. Se debe considerar que las mediciones de v_A se normalizan a 1 cuando i_E se fija en su valor nominal de 0.1 A. Este método permite deducir la función de flujo de excitación $\Psi_E(i_E)$ a partir de las mediciones normalizadas del voltaje de armadura v_{Anorm} solo en función de i_E .

El montaje físico de los equipos se muestra en la Figura 3.13. El servomotor del módulo CO3636-6V se activa en modo de control de velocidad a 2000 RPM. Y las mediciones se registran en la Tabla 3.7.

Tabla 3.7: Mediciones de i_E y v_A para estimar $\Psi_E(i_E)$; con $\omega_m = 2000RPM$

N°	$i_E[\text{mA}]$	$v_A[\text{V}]$	$v_{Anorm}[1]$
1	0.00	5.13	0.025
2	10.09	30.50	0.151
3	20.11	60.60	0.300
4	30.24	90.40	0.447
5	40.15	117.10	0.579
6	50.30	141.50	0.700
7	60.13	161.00	0.796
8	70.20	176.40	0.872
9	80.20	186.70	0.923
10	90.40	195.50	0.967
11	100.20	202.20	1.000
$\triangleright \Psi_E(i_E) = v_{Anorm}(i_E) = -74.222i_E^2 + 17.493i_E - 0.002 [1]$			

Figura 3.14: Datos experimentales y curva de regresión cuadrática $\Psi_E(i_E) = f(i_E)$



Los datos de la Tabla 3.7 son empleados para el análisis de regresión polinómica de segundo grado. La Figura 3.14 presenta la superposición de los datos experimentales y la curva de regresión; aquí se evidencia la relación no lineal entre Ψ_E e i_E . Finalmente, de la Tabla 3.7, la función de flujo de excitación $\Psi_E(i_E) = f(i_E)$ es

$$\Psi_E(i_E) = -74.222i_E^2 + 17.493i_E - 0.0018 [1]. \quad (3.14)$$

3.3.3. Constante de máquina C_m

La constante de máquina C_m se estima bajo condiciones similares a las aplicadas en la determinación de la función $\Psi_E(i_E)$. A partir de la función de flujo $\Psi_E(i_E)$ y la relación en (3.12) es posible deducir C_m utilizando los esquemas de montaje experimental de las Figuras 3.12 y 3.13. El circuito de campo se alimenta con un voltaje nominal de CC, $v_E = 220 \text{ V}$, a una corriente nominal $i_E = 100.20 \text{ mA}$. Bajo esta configuración se cumple que el flujo de campo o excitación toma su valor nominal

$$\Psi_{Enom} = \Psi_E(i_{Enom}) = 1; \text{ si } i_{Enom} = 100.20 \text{ mA}, \quad (3.15)$$

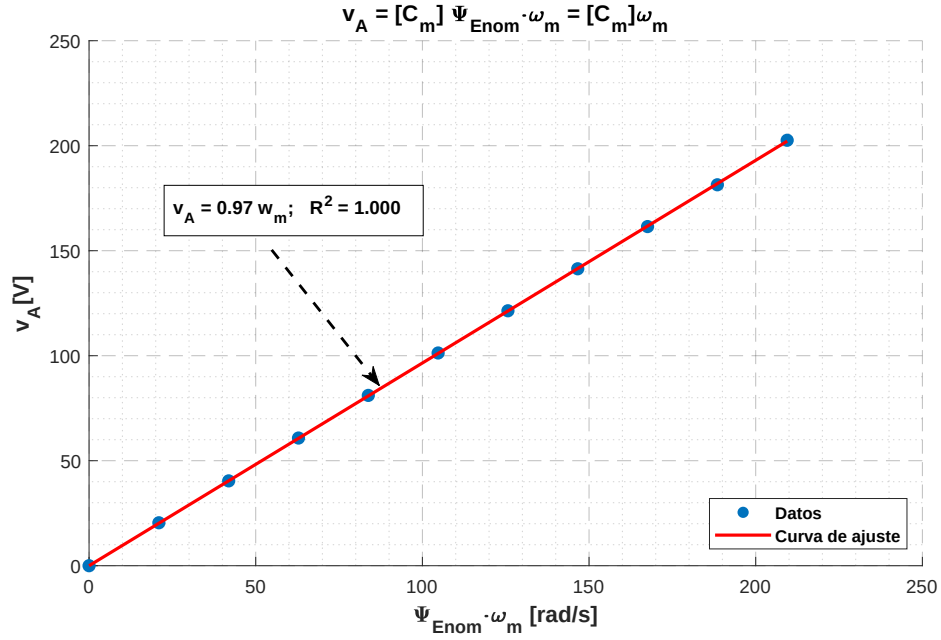
$$\Rightarrow v_A = e_A = C_m \Psi_{Enom} \omega_m = C_m \omega_m. \quad (3.16)$$

Si $\Psi_E = \Psi_{Enom} = 1$, la expresión (3.12) se reduce a (3.16), lo cual hace factible determinar C_m de forma directa. El proceso consiste en variar la velocidad del motor mediante el servomotor del módulo CO3636-6V, acoplado al eje del propio motor. Se capturan mediciones de v_A frente a ω_m con vistas a estimar C_m usando un análisis de regresión.

Tabla 3.8: Mediciones de ω_m y v_A para estimar C_m ; con $\Psi_E = \Psi_{Enom} = 1$

N°	ω_m [RPM]	ω_m [rad/s]	$\Psi_E \cdot \omega_m$ [rad/s]	$e_A = v_A$ [V]	C_m [V·s/rad]
1	0	0.000	0.000	0.00	
2	200	20.944	20.944	20.45	0.976
3	400	41.888	41.888	40.40	0.964
4	600	62.832	62.832	60.80	0.968
5	800	83.776	83.776	81.10	0.968
6	1000	104.720	104.720	101.30	0.967
7	1200	125.664	125.664	121.40	0.966
8	1400	146.608	146.608	141.40	0.964
9	1600	167.552	167.552	161.50	0.964
10	1800	188.496	188.496	181.40	0.962
11	2000	209.440	209.440	202.60	0.967
Regresión lineal $v_A = e_A = m \cdot \omega_m$ dado $m = C_m$					$\triangleright$ 0.970

Figura 3.15: Datos experimentales y curva de regresión lineal $[v_A] \propto [\Psi_{Enom} \cdot \omega_m]$



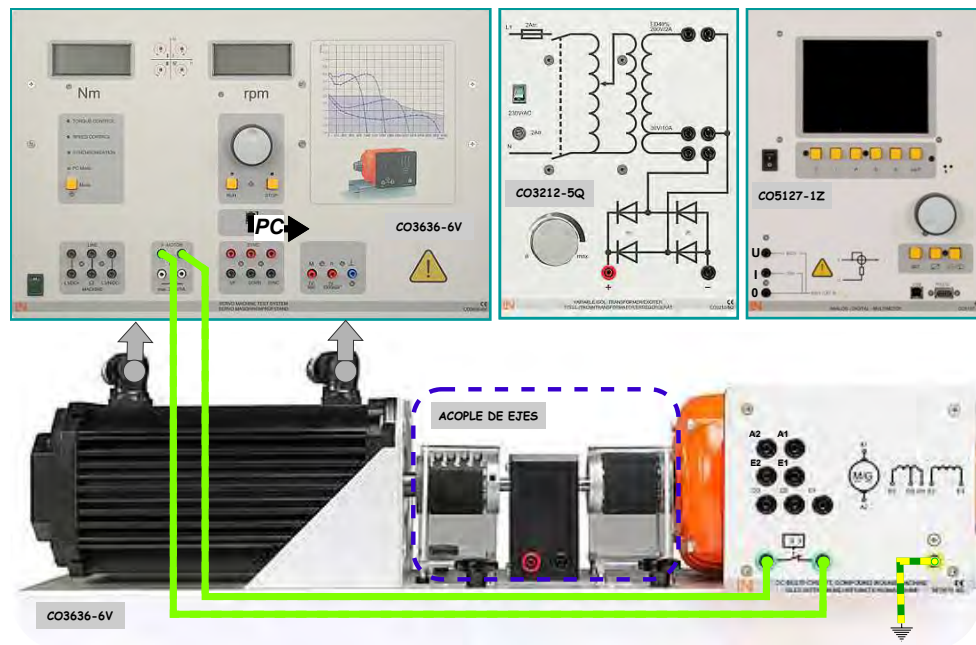
Los datos del experimento quedan registrados en la Tabla 3.8. La pendiente de la curva mostrada en la Figura 3.15 corresponde a la constante de máquina C_m . Es importante notar que, como Ψ_E y C_m son términos acoplados y Ψ_E es ahora adimensional, C_m tiene unidades $[V \cdot s/rad]$ con el fin de mantener la coherencia dimensional.

3.3.4. Momento de inercia del rotor Θ_M

La Figura 3.16 presenta el montaje experimental de los equipos empleados en la obtención del momento de inercia del rotor, Θ_M . En este experimento, el motor no se conecta a ninguna fuente de alimentación.

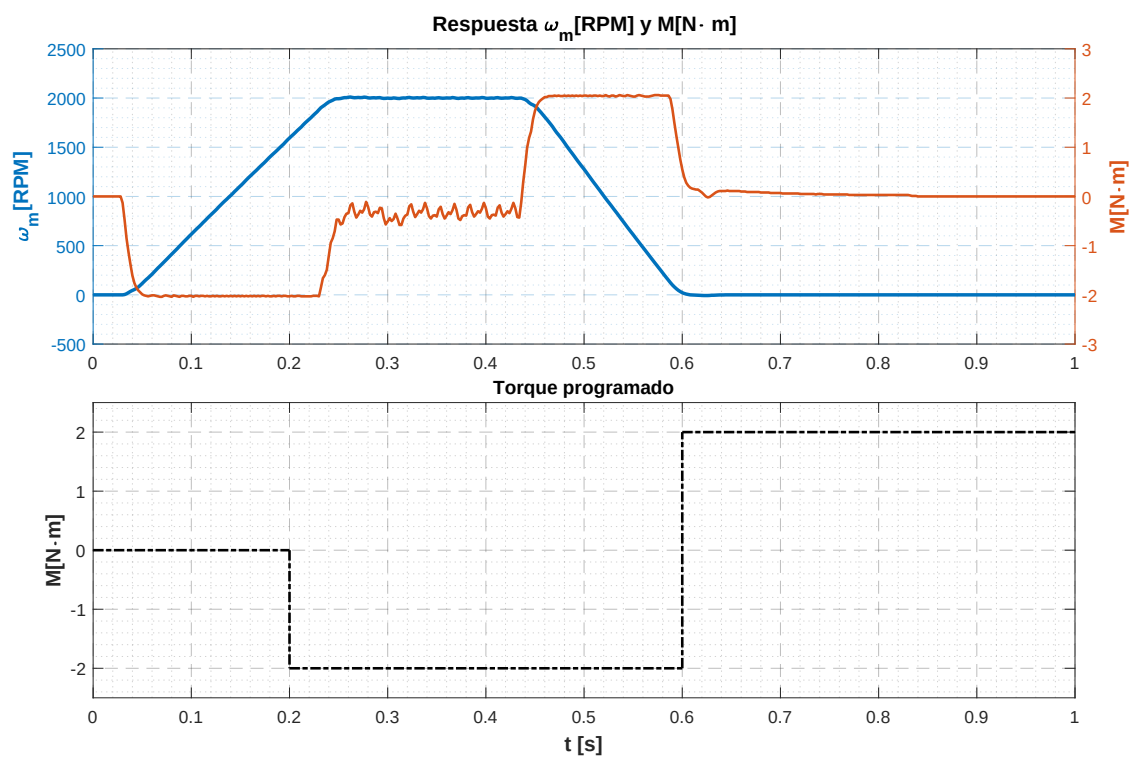
Tras ensamblar el circuito conforme con el diagrama de montaje, se procede a activar el banco de pruebas CO3636-6V y a ejecutar el programa ActiveServo. Se programa un torque de aceleración y frenado de $2 \text{ N} \cdot \text{m}$ en el ActiveServo, y se carga al módulo CO3636-6V. En la Figura 3.17 se tiene el torque programado y la respuesta del motor DC.

Figura 3.16: Montaje de los equipos para estimar Θ_M



Nota. Tomada y adaptada de Linnertz (2016) ©2016 Lucas Nülle GmbH.

Figura 3.17: Respuesta de velocidad frente al par constante programado



El par de inercia se puede determinar según la relación (2.55). Al aplicar un torque controlado por medio del servomotor, el torque neto que actúa sobre el eje del motor DC se define como (Linnertz, 2016; Meneses Morales, 2013)

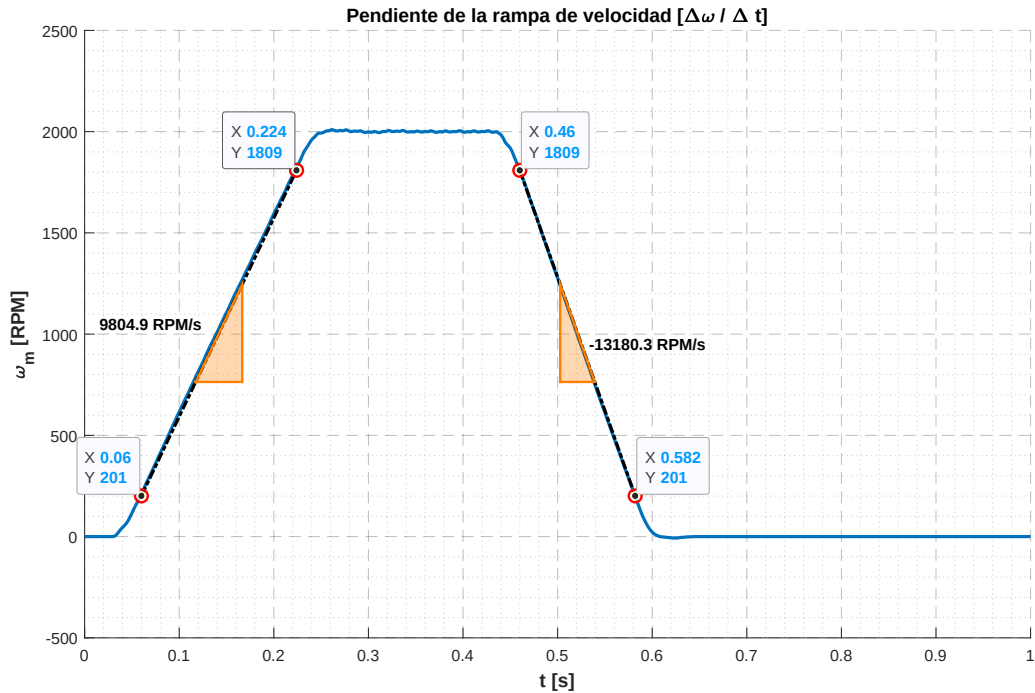
$$M = -\Theta_G \frac{d}{dt} \omega_m = -(\Theta_M + \Theta_S) \frac{d}{dt} \omega_m, \quad (3.17)$$

donde Θ_S es el par de inercia del servomotor del módulo CO3636-6V. De la Figura 3.17, las rampas de aceleración y frenado permiten la aproximación

$$M = -\Theta_G \frac{d}{dt} \omega_m = -\Theta_G \frac{\Delta}{\Delta t} \omega_m. \quad (3.18)$$

Usando esta aproximación y recurriendo al criterio del tiempo de subida —el intervalo que emplea la señal de respuesta para pasar del 10 % al 90 % de su valor final—⁴ se calcula la pendiente de las rampas de aceleración y frenado. En la Figura 3.18 y la Tabla 3.9 se registran los datos de medición. Además, el fabricante proporciona el valor del momento de inercia del servomotor, $\Theta_S = 8.3 \times 10^{-4} \text{ kg} \cdot \text{m}^2$. Por lo tanto, es posible obtener Θ_M .

Figura 3.18: Rampas de aceleración y frenado del motor DC



⁴Como lo especifica Nise (2015)

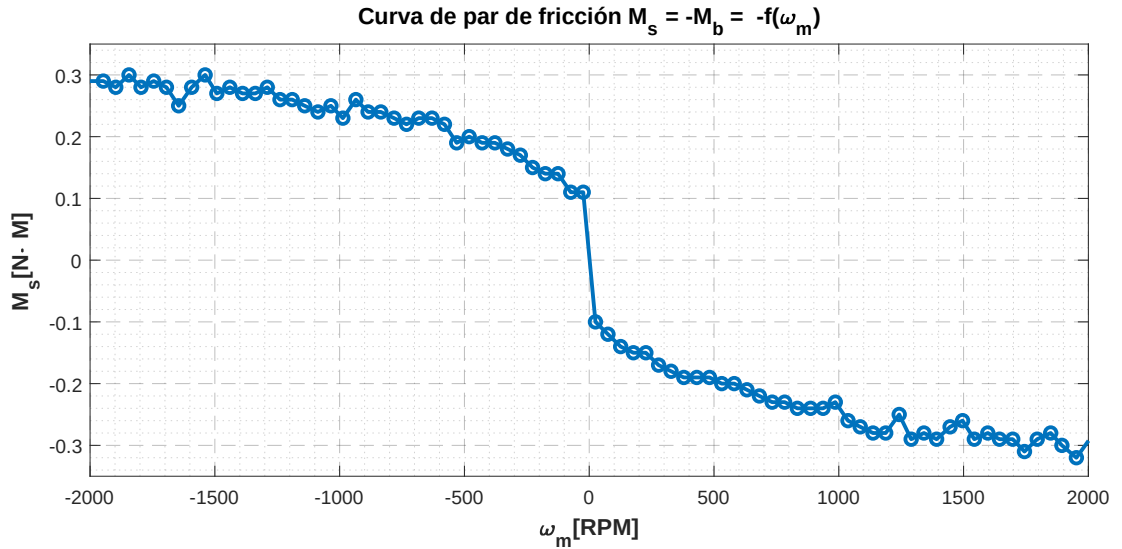
Tabla 3.9: Estimación de Θ_M , dado Θ_S por el fabricante

$M[\text{N}\cdot\text{m}]$	$\Delta\omega_m/\Delta t[\text{RPM/s}]$	$\Delta\omega_m/\Delta t[\text{rad/s}^2]$	$\Theta_G[\text{kg}\cdot\text{m}^2]$
-2	9804.90	1026.77	19.48×10^{-4}
2	-13180.30	1380.24	14.49×10^{-4}
Valor medio de $\Theta_G = \Theta_M + \Theta_S$			$\triangleright 16.99 \times 10^{-4}$
Valor estimado de Θ_M , con $\Theta_S = 8.3 \times 10^{-4} \text{ kg}\cdot\text{m}^2$			$\triangleright 8.69 \times 10^{-4}$

3.3.5. Función de pérdidas por fricción $M_b = f(\omega_m)$

El montaje experimental de la Figura 3.16 también se emplea para determinar las pérdidas por fricción. La configuración experimental implementa un método de medición del par necesario que permita mantener una velocidad constante en el eje del motor DC. La prueba consiste en activar el banco de pruebas CO3636-6V junto con el programa ActiveServo. A continuación, se programa un barrido de velocidades desde 2000 RPM a -2000 RPM y se registra el par ejercido por el servomotor del módulo CO3636-6V sobre el motor DC (véase Figura 3.19).

Figura 3.19: Datos experimentales del par ejercido por el servomotor $M_s - \omega_m$

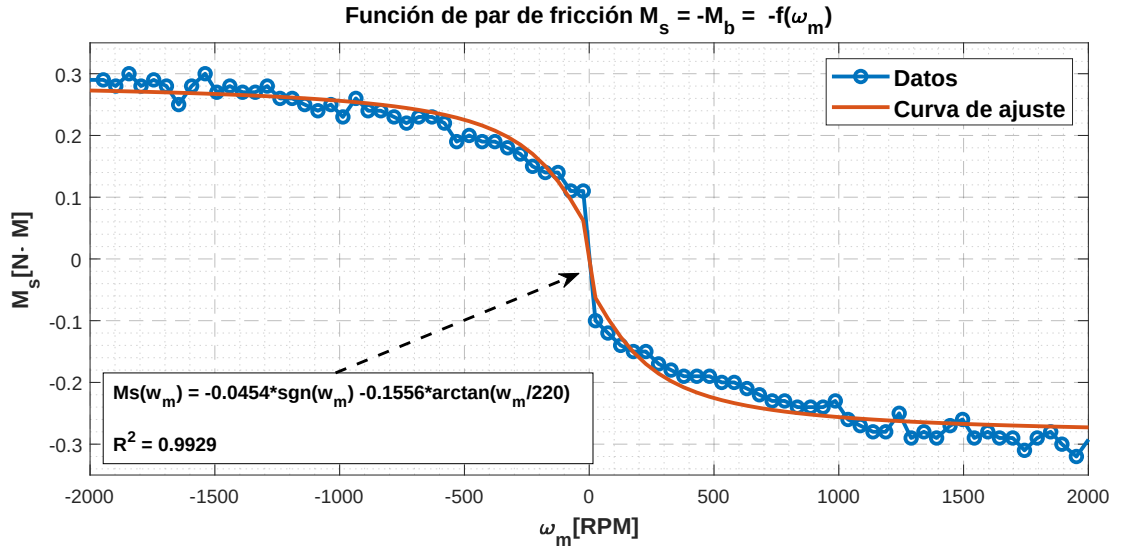


Según Linnertz (2016), las pérdidas por fricción se modelan mediante una función de la velocidad definida en la ecuación (3.19). Donde M_s es el par ejercido por el servomotor sobre el motor DC para mantener la velocidad programada.

$$M_b = -M_s = f(\omega_m) = a \cdot \text{sgn}(\omega_m) + b \cdot \tan^{-1}(\omega_m/220). \quad (3.19)$$

La herramienta de MATLAB se empleó para procesar y analizar los datos experimentales con el fin de ajustar la curva de compensación. Posteriormente, a partir de los datos registrados se ajustó la función de compensación definida en la ecuación (3.19).

Figura 3.20: Datos experimentales y curva de ajuste $M_s(\omega_m)$



La Figura 3.20 muestra los datos experimentales junto con la función de compensación. Sin embargo, dada la relación (3.19), es imperativo entender que los parámetros a y b corresponden con las constantes negativas de M_s , y que ω_m se mide en RPM para este caso; por tanto, se tiene

$$M_b(\omega_m) = 0.0454 \cdot \text{sgn}(\omega_m) + 0.1556 \cdot \tan^{-1}(\omega_m/220) \text{ [N} \cdot \text{m]}. \quad (3.20)$$

3.4. Parámetros físicos del motor DC

Los parámetros físico obtenidos en los experimentos descritos en las secciones anteriores se resumen en la Tabla 3.10.

Tabla 3.10: Parámetros físicos del motor DC

Símbolo	Descripción	Valor[Unidades]
R_A	Resistencia de armadura	16.95 [Ω]
L_A	Inductancia de armadura	406.82 [mH]
R_E	Resistencia de campo	2059.39 [Ω]
L_E	Inductancia de campo	108.97 [H]
C_m	Constante de máquina	0.97 [V·s/rad]
Θ_M	Momento de inercia del rotor	8.69×10^{-4} [kg·m ²]
Ψ_E	Flujo de excitación	$\Psi_E(i_E)$ (3.14) [1]
M_b	Par de fricción	$M_b(\omega_m)$ (3.20) [N·m]

3.5. Prueba y simulación de los modelos

Considerando los parámetros físicos en la Tabla 3.10 y los modelos no lineal y lineal definidos en las ecuaciones (3.3) y (3.6), se procedió a simular los modelos en lazo abierto y comparar los resultados obtenidos con los datos reales. El diagrama de bloques, el diagrama de montaje físico de los equipos y el modelo en Simulink para las pruebas en lazo abierto se muestran en las Figuras 3.21, 3.22 y 3.23, respectivamente. El Toolbox de MATLAB CO3636-3D y el *Code Composer Studio 5* de Texas Instruments permiten cargar el firmware generado en Simulink directamente al MCU C2000 del módulo CO3636-3B.

Figura 3.21: Diagrama de bloques de los equipos para las pruebas en lazo abierto

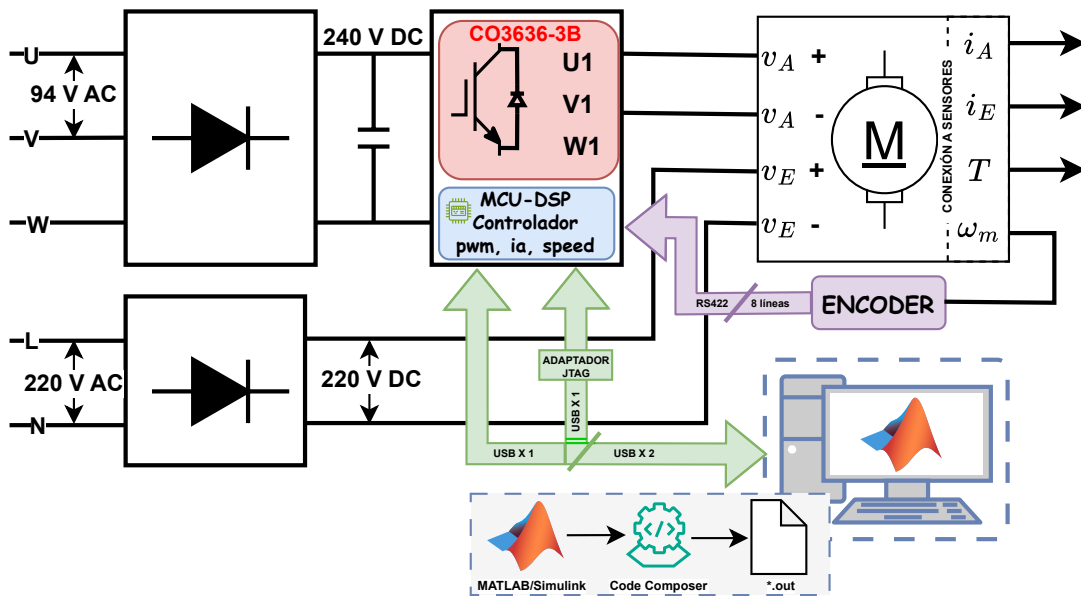
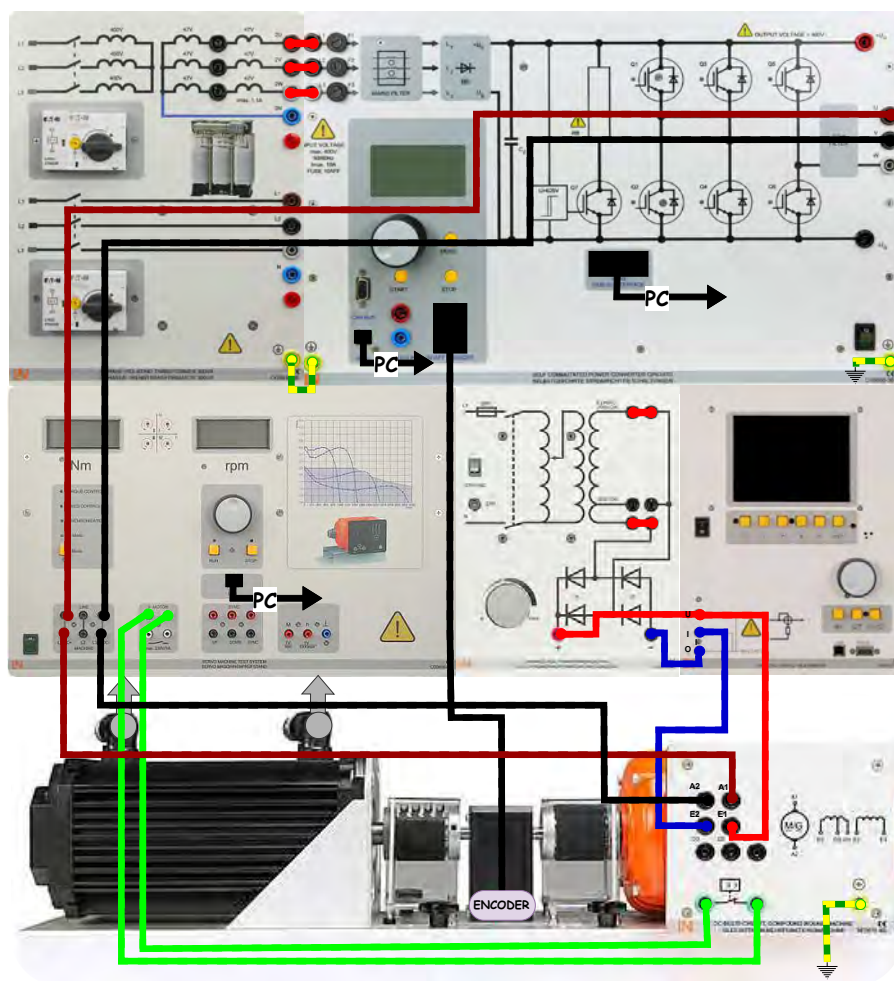
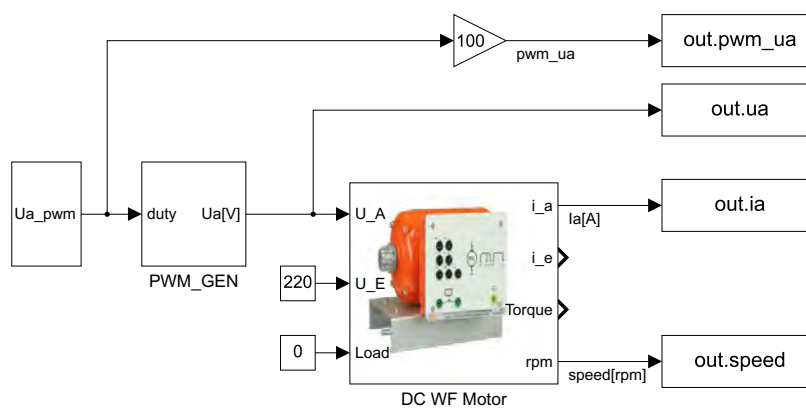


Figura 3.22: Montaje de los equipos para las pruebas en lazo abierto



Nota. Tomada y adaptada de Linnertz (2016) ©2016 Lucas Nülle GmbH.

Figura 3.23: Diagrama de bloques en Simulink para las pruebas en lazo abierto



Nota. Esquema de simulación de los modelos lineal y no lineal.

3.5.1. Métricas de evaluación de los modelos

Los modelos propuestos se evalúan mediante las métricas definidas a continuación. Donde y es el vector de las medidas reales, $\hat{y}$ es el vector de los valores predichos por el modelo, y_i es la i -ésima medida real, $\hat{y}_i$ es la i -ésima medida predicha por el modelo, $\bar{y}$ es la media de y , N el número de muestras y $\|\cdot\|$ es la norma euclidiana (L_2^5).

Error cuadrático medio (MSE):

$$MSE = \frac{1}{N} \sum_{k=1}^N (y_i - \hat{y}_i)^2. \quad (3.21)$$

Raíz del error cuadrático medio (RMSE):

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^N (y_i - \hat{y}_i)^2} = \sqrt{MSE}. \quad (3.22)$$

Error absoluto medio (MAE):

$$MAE = \frac{1}{N} \sum_{k=1}^N |y_i - \hat{y}_i|. \quad (3.23)$$

Porcentaje de ajuste basado en NRMSE (FitPercent):

$$NRMSE = \frac{\|y - \hat{y}\|}{\|y - \bar{y}\|}. \quad (3.24)$$

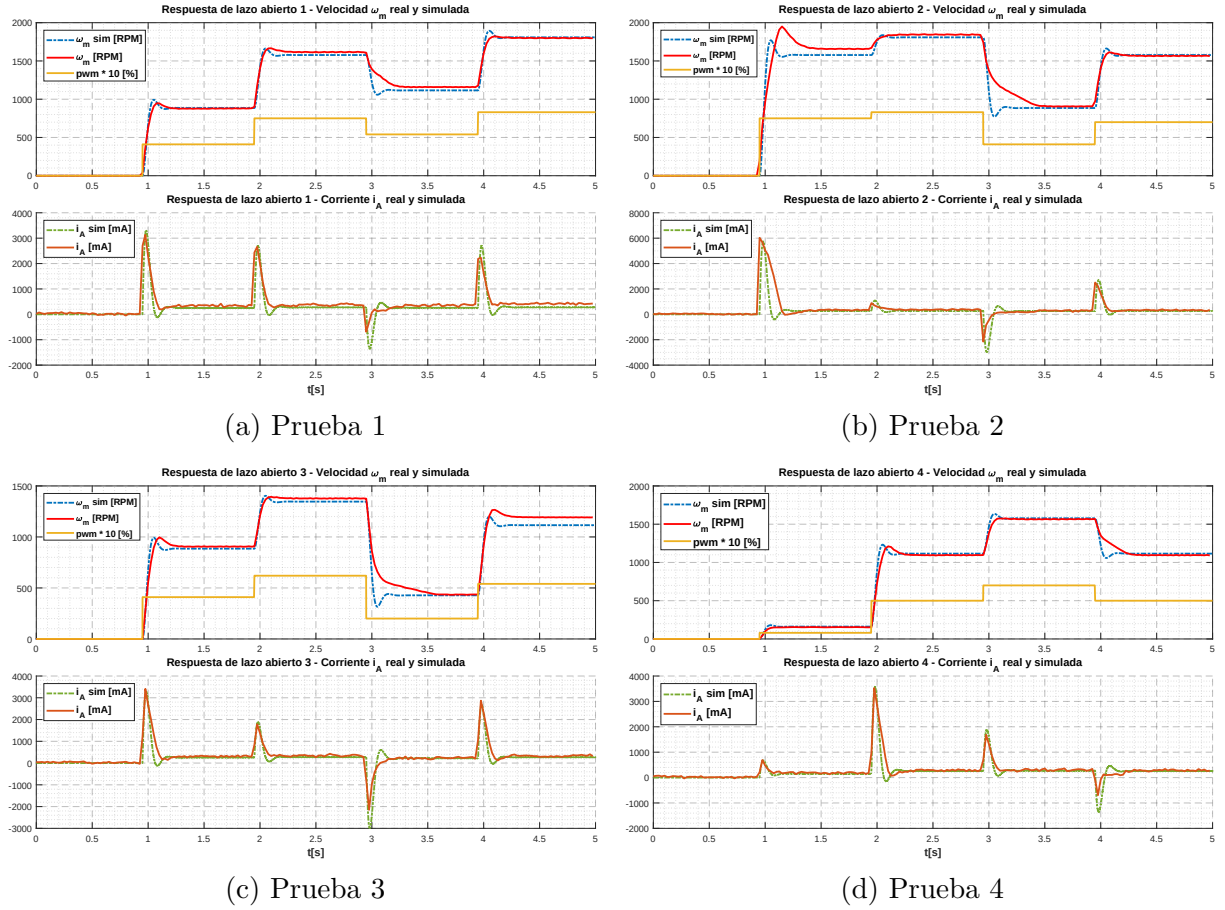
$$FitPercent = 100 \times (1 - NRMSE). \quad (3.25)$$

Se proponen tres modelos: el modelo no lineal y lineal obtenido en las secciones anteriores, y un tercer modelo estimado mediante el Toolbox *system identification* de MATLAB. Se llevan a cabo pruebas en lazo abierto a fin de evaluar y comparar la dinámica de los modelos con las mediciones reales. Las variables de interés de los modelos se limitan a la velocidad del motor ω_m y la corriente de armadura i_A , en ese orden de prioridad.

⁵También conocida como longitud euclidiana, se define como $\|(x_1, x_2, \dots, x_n)\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$.

3.5.2. Simulación del modelo no lineal

Figura 3.24: Respuesta en lazo abierto del modelo no lineal contra la respuesta real



Nota. La respuesta real corresponde a aquella etiqueta que no incluye el sufijo *sim*, reservado para las respuestas generadas mediante simulación. Además, la señal de PWM se encuentra escalda por un factor de 10.

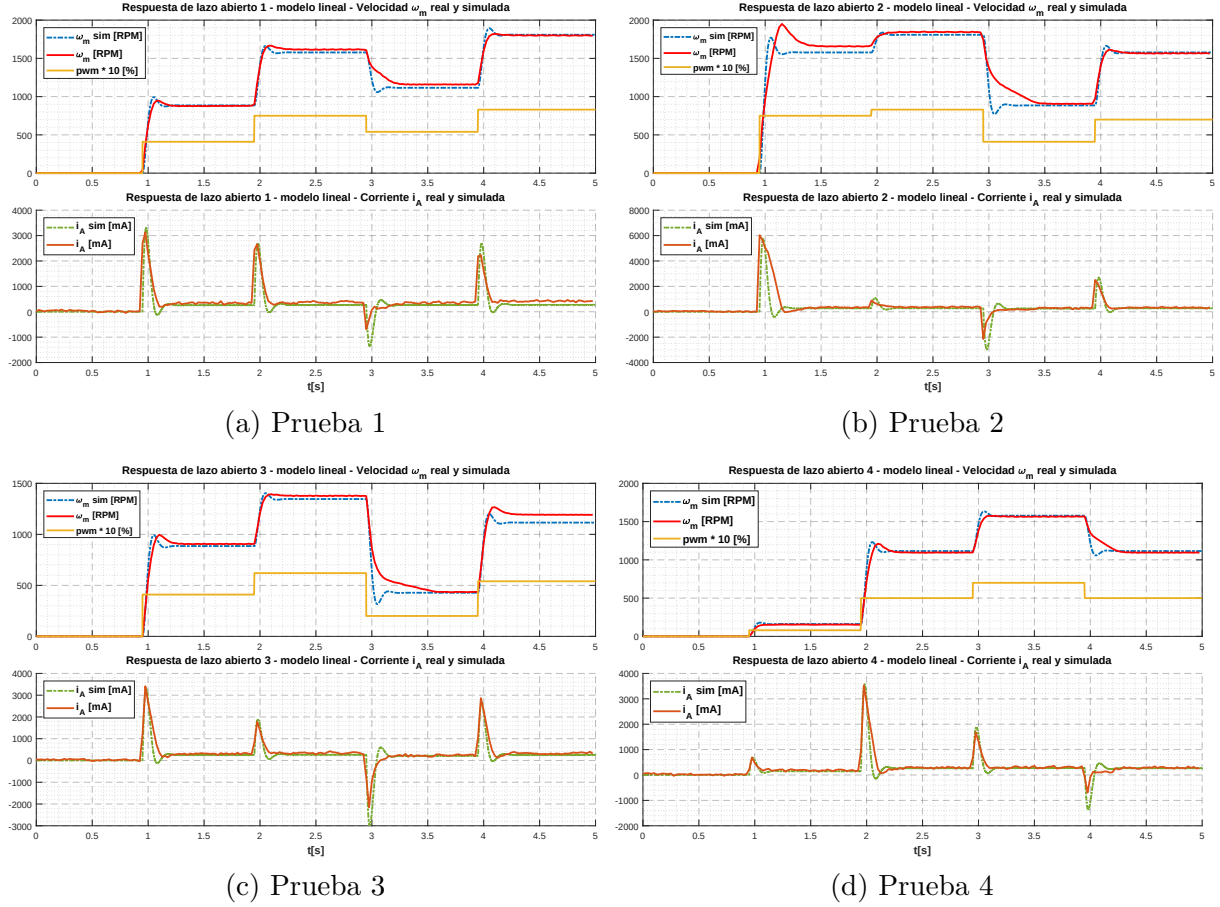
Tabla 3.11: Métricas de evaluación del modelo no lineal

Prueba	MSE	RMSE	MAE	FitPercent
1	[27.847, 0.121]	[5.277, 0.348]	[3.144, 0.168]	[92.035, 28.326]
2	[135.708, 0.484]	[11.649, 0.696]	[6.516, 0.232]	[83.359, 19.259]
3	[49.806, 0.061]	[7.057, 0.246]	[4.548, 0.124]	[86.381, 51.120]
4	[19.881, 0.034]	[4.459, 0.183]	[2.235, 0.085]	[92.966, 53.599]
Promedio	[58.311, 0.175]	[7.111, 0.368]	[4.111, 0.152]	[88.685, 38.076]

Nota. Cada métrica de evaluación tiene representación en un vector bidimensional, ya que se evalúan simultáneamente dos variables de estado: la velocidad del motor (ω_m [RPM]) y la corriente de armadura (i_A [A]); la forma del vector es $[Metric(\omega_m), Metric(i_A)]$.

3.5.3. Simulación del modelo lineal

Figura 3.25: Respuesta en lazo abierto del modelo lineal contra la respuesta experimental



Nota. La respuesta real corresponde a aquella etiqueta que no incluye el sufijo *sim*, reservado para las respuestas generadas mediante simulación. Además, la señal de PWM se encuentra escalda por un factor de 10.

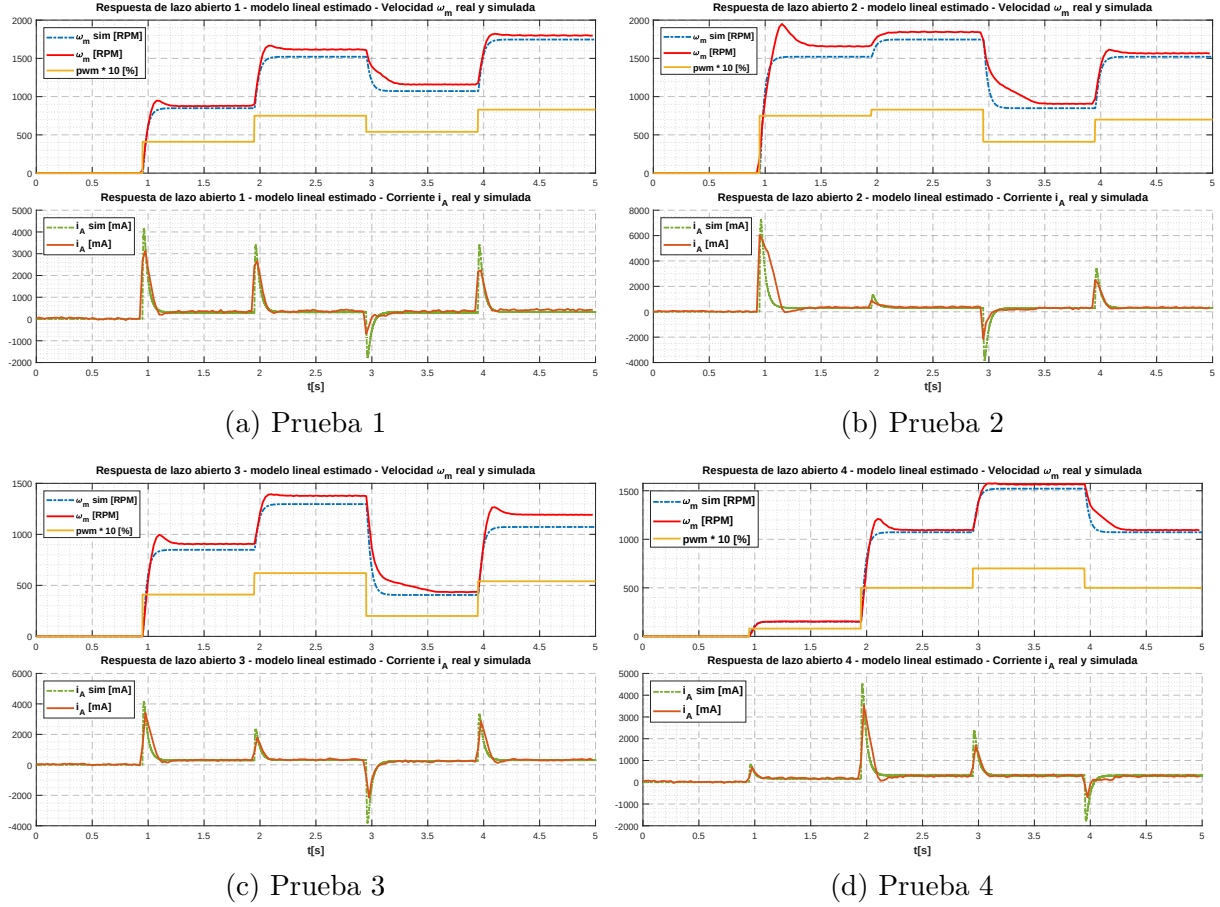
Tabla 3.12: Métricas de evaluación del modelo lineal

Prueba	MSE	RMSE	MAE	FitPercent
1	[27.847, 0.121]	[5.277, 0.348]	[3.144, 0.168]	[92.035, 28.326]
2	[135.708, 0.484]	[11.649, 0.696]	[6.516, 0.232]	[83.359, 19.259]
3	[49.806, 0.061]	[7.057, 0.246]	[4.548, 0.124]	[86.381, 51.120]
4	[19.881, 0.034]	[4.459, 0.183]	[2.235, 0.085]	[92.966, 53.599]
Promedio	[58.311, 0.175]	[7.111, 0.368]	[4.111, 0.152]	[88.685, 38.076]

Nota. Cada métrica de evaluación tiene representación en un vector bidimensional, ya que se evalúan simultáneamente dos variables de estado: la velocidad del motor (ω_m [RPM]) y la corriente de armadura (i_A [A]); la forma del vector es $[Metric(\omega_m), Metric(i_A)]$.

3.5.4. Simulación del modelo lineal estimado

Figura 3.26: Respuesta en lazo abierto del modelo lineal estimado contra la respuesta real



Nota. La respuesta real corresponde a aquella etiqueta que no incluye el sufijo *sim*, reservado para las respuestas generadas por simulación. En este experimento el modelo lineal en espacio de estados se estimó usando el Toolbox *system identification* de MATLAB.

Tabla 3.13: Métricas de evaluación del modelo lineal estimado

Prueba	MSE	RMSE	MAE	FitPercent
1	[66.183, 0.106]	[8.135, 0.325]	[6.524, 0.138]	[87.721, 32.954]
2	[172.170, 0.431]	[13.121, 0.657]	[9.689, 0.203]	[81.257, 23.778]
3	[86.183, 0.046]	[9.284, 0.215]	[7.516, 0.101]	[82.085, 57.295]
4	[22.919, 0.026]	[4.787, 0.160]	[2.989, 0.071]	[92.447, 59.387]
Promedio	[86.864, 0.152]	[8.832, 0.339]	[6.680, 0.128]	[85.878, 43.354]

Nota. Cada métrica de evaluación tiene representación en un vector bidimensional, ya que se evalúan simultáneamente dos variables de estado: la velocidad del motor (ω_m [RPM]) y la corriente de armadura (i_A [A]); la forma del vector es $[Metric(\omega_m), Metric(i_A)]$.

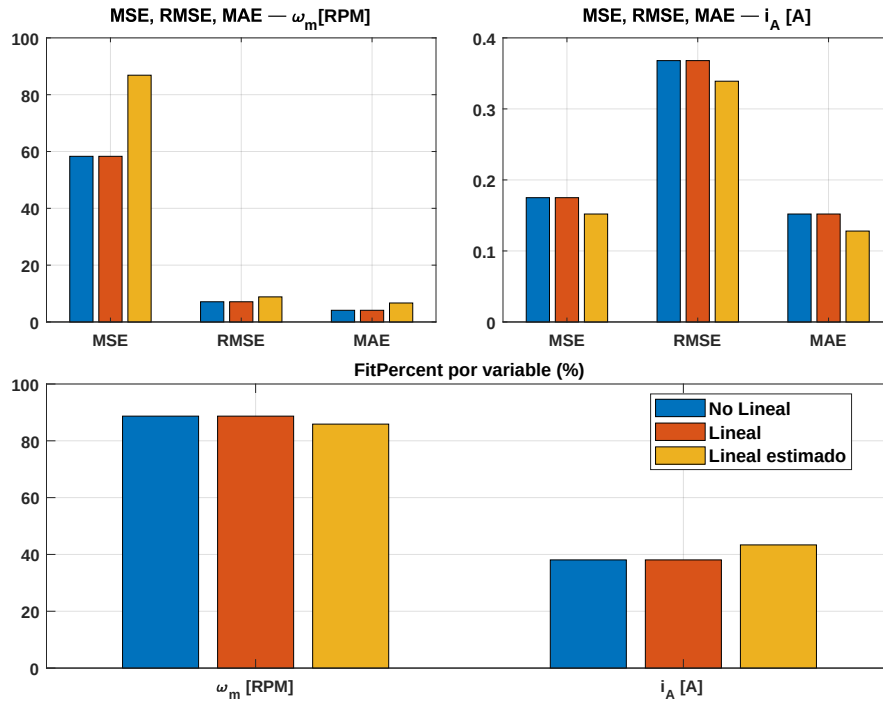
3.5.5. Comparación de los modelos

Tabla 3.14: Comparación de las métricas de evaluación promedio de los modelos

Modelo	MSE promedio	RMSE promedio	MAE promedio	FitPercent promedio
1	[58.311, 0.175]	[7.111, 0.368]	[4.111, 0.152]	[88.685, 38.076]
2	[58.311, 0.175]	[7.111, 0.368]	[4.111, 0.152]	[88.685, 38.076]
3	[86.864, 0.152]	[8.832, 0.339]	[6.680, 0.128]	[85.878, 43.354]

Nota. 1, 2 y 3 corresponden a los modelos no lineal, lineal y lineal estimado, respectivamente.

Figura 3.27: Gráfico de barras de la comparación de las métricas por variables de estado



La Tabla 3.14 y la Figura 3.27 muestran que los modelos teóricos—no lineal y lineal, definidos en las ecuaciones (3.3) y (3.6)—coinciden en estado estacionario. Estos, se aproximan en promedio 5.278 % mejor en la velocidad frente al modelo estimado por MATLAB. Además, este último mejora en promedio un 2.807 % la estimación de la corriente i_A . No obstante, este modelo no reproduce correctamente la dinámica de velocidad, como se aprecia en la Figura 3.26. En consecuencia, para la sintonización de los controladores, se emplean los modelos teóricos no lineal y lineal.

CAPÍTULO 4:

DISEÑO Y SINTONIZACIÓN DE LOS CONTROLADORES

4.1. Diseño del controlador proporcional integral - PI

Esta sección describe el diseño y la sintonización de un controlador PI para regular la velocidad del motor DC, aplicando métodos clásicos—el método de Ziegler-Nichols, el método del óptimo simétrico—y un enfoque moderno de aprendizaje por refuerzo.

4.1.1. Arquitectura del controlador PI

Los sistemas de control de motores están constituidos por múltiples lazos de control dispuestos en cascada, según la aplicación y la variable que se desea regular. La Tabla 4.1 resume los lazos de control necesarios para cada accionamiento (Linnertz, 2016).

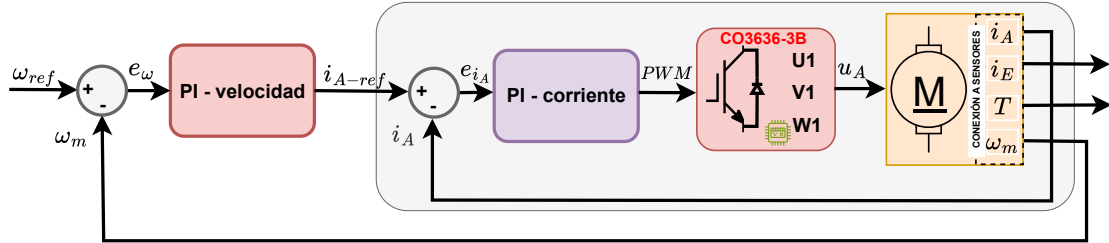
Tabla 4.1: Lazo de control según el accionamiento

Accionamiento (control)	Bucle de control de corriente	Bucle de control de velocidad	Bucle de control de posición
Par de giro	X		
Velocidad	X	X	
Posición	X	X	X

Nota. Tomada, traducida y adaptada de (Linnertz, 2016) ©2016 Lucas Nülle GmbH.

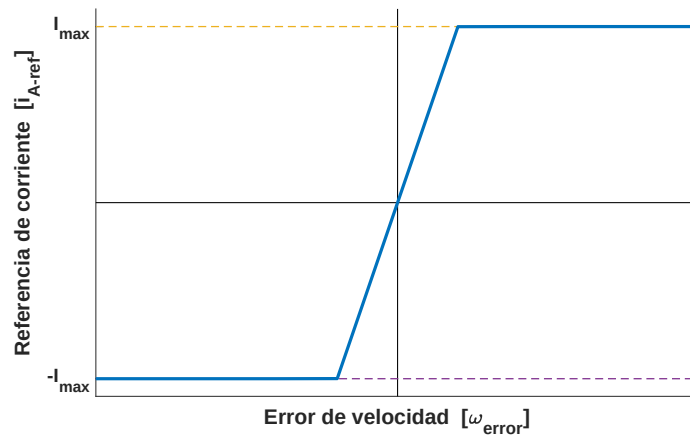
En esta investigación, la variable de interés es la velocidad del motor, en consecuencia, el esquema de control se compone de una lazo de control de corriente, y uno de velocidad, tal como se muestra en la Figura 4.1.

Figura 4.1: Arquitectura del controlador PI de velocidad del motor DC



El lazo interno regula la corriente—y por ende el par del motor—, mientras que el lazo externo controla la velocidad del mismo. Según Hughes & Drury (2019), el lazo de control de velocidad proporciona los mecanismos para implementar rutinas de protección y limitar la corriente a fin de proteger el sistema contra flujos excesivos a través del motor y sus controladores. La Figura 4.2 ilustra la dinámica del bucle de control de velocidad: la referencia de corriente crece en proporción al error de velocidad; cuando el error supera un límite, la salida del bloque de control de velocidad se satura para garantizar que la corriente de referencia i_{A-ref} nunca exceda el valor máximo admisible por el motor. Este mismo mecanismo de protección se aplica a la señal de control v_A , regulada por el lazo interno de corriente.

Figura 4.2: Dinámica del bucle de control de velocidad



4.1.2. Sintonización del controlador PI del lazo de corriente mediante el criterio del módulo óptimo

Dado que el accionamiento de velocidad requiere de dos lazos (véase la Figura 4.1). Es imprescindible implementar primero el lazo de corriente. Con base en los trabajos de Linnertz (2016) y Bankovic et al. (2021), se calcularon las ganancias del controlador PI de corriente aplicando el criterio del módulo óptimo.

El criterio del módulo óptimo pretende determinar las ganancias del controlador de manera que la respuesta en lazo cerrado tenga una magnitud lo más plana posible (≈ 1) en una banda de frecuencias bajas-media. Esto permite al sistema rechazar perturbaciones, lograr un buen seguimiento y evitar modos de oscilación indeseados. Formalmente, se diseña el bucle de control para que el módulo de la función de transferencia de lazo abierto cumpla

$$|L(j\omega)| = |G_{\text{controlador}}(j\omega)G_{\text{planta}}(j\omega)| \gg 1$$

(Papadopoulos et al., 2017).

Las ecuaciones para calcular los parámetros según el criterio del módulo óptimo son:

$$T_i = T_1, \tag{4.1}$$

$$K_p = \frac{T_1}{2K_s T_\sigma}, \tag{4.2}$$

bajo la condición

$$T_\sigma \ll T_1. \tag{4.3}$$

siendo T_i el tiempo de acción integral, K_p la ganancia proporcional, T_1 la constante dominante o mayor de tiempo, T_σ la constante menor de tiempo (retardo puro equivalente y tiempos muertos) y K_s la ganancia de la planta (Linnertz, 2016; Bankovic et al., 2021).

Partiendo de la ecuación (2.50) y el circuito equivalente del motor (véase la Figura 3.1b), si el motor no presenta movimiento—es decir, $e_A = 0$ V—, el circuito de armadura

se reduce a un sistema de primer orden:

$$\frac{I_A(s)}{V_A(s)} = \frac{1/R_A}{1 + (L_A/R_A)s}. \quad (4.4)$$

El convertidor de transistores IGBT (módulo CO3636-3B) requiere de una señal de PWM en el rango de $\pm 100\%$, y se alimenta con una fuente de 240 V DC. Dado que el controlador gestiona directamente el ciclo de trabajo PWM y no el voltaje de armadura v_A , el sistema modelado en la ecuación (4.4) puede expresarse en función a la señal de entrada D_{PWM} . En primer lugar, el voltaje promedio de armadura se formula como

$$V_A(s) = D_{PWM}(s) * \frac{v_{Amax}}{100}, \quad (4.5)$$

donde D_{PWM} es el ciclo de trabajo PWM, y v_{Amax} es el voltaje máximo de armadura suministrado por el convertidor (240 V). Por tanto, la función de transferencia es

$$\frac{I_A(s)}{D_{PWM}(s)} = \frac{v_{Amax}/(100R_A)}{1 + (L_A/R_A)s} = \frac{K_s}{1 + T_1s}. \quad (4.6)$$

De la expresión (4.6) se obtienen

$$K_s = \frac{v_{Amax}}{100R_A}, \quad (4.7)$$

$$T_1 = \frac{L_A}{R_A}. \quad (4.8)$$

donde K_s es la ganancia de la planta y T_1 es la constante o polo dominante del lazo de corriente. Utilizando los valores de la Tabla 3.10, y sustituyendo las ecuaciones (4.7) y (4.8) en (4.1) y (4.2), asumiendo $T_\sigma = 500\mu s$ según Linnertz (2016), se obtienen las constantes del controlador PI:

$$T_i = 24.001 [\text{ms}], \quad (4.9)$$

$$K_p = 169.5083. \quad (4.10)$$

4.1.3. Sintonización del controlador PI de velocidad mediante el criterio óptimo simétrico

El criterio óptimo simétrico está diseñado para maximizar el margen de fase en la función de lazo abierto del sistema. Este da más margen de fase y por tanto permite tolerar retardos y variaciones (Stumper, 2012). En lazo abierto se busca una respuesta simétrica en torno a la frecuencia de cruce. Es decir, se busca un diseño donde la magnitud y fase estén bien balanceadas (EEEGUIDE, s.f.)

Las relaciones para el cálculo de las ganancias del controlador PI de velocidad son:

$$T_i = 4T_\sigma, \quad (4.11)$$

$$K_p = \frac{1}{2K_sT_\sigma}, \quad (4.12)$$

$$K_s = \frac{\Delta\omega}{\Delta t}. \quad (4.13)$$

donde T_i es la constante de tiempo de acción integral, K_p es la ganancia proporcional y T_σ la constante menor de tiempo y K_s ganancia del sistema controlado (Linnertz, 2016).

La estructura para la sintonización del controlador PI de velocidad se muestra en la Figura 4.3. Se aplicó una consigna de 1 A transcurridos 0.1 s; la curva de respuesta de velocidad se muestra en la Figura 4.4.

Figura 4.3: Estructura para la sintonización del control de velocidad

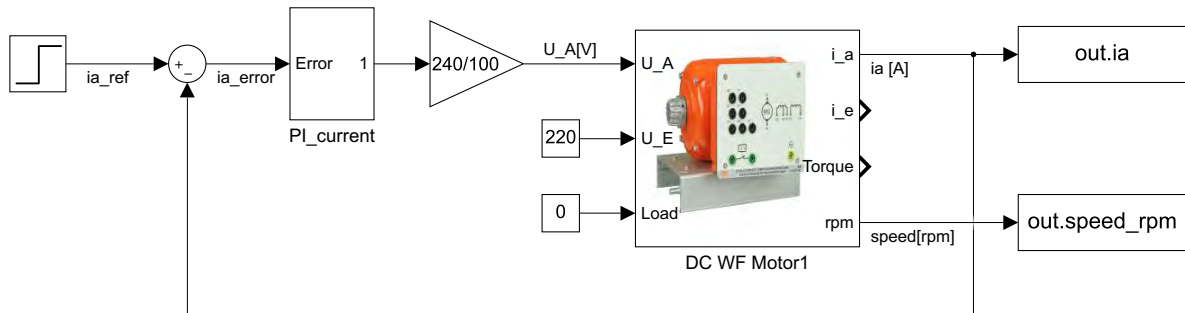
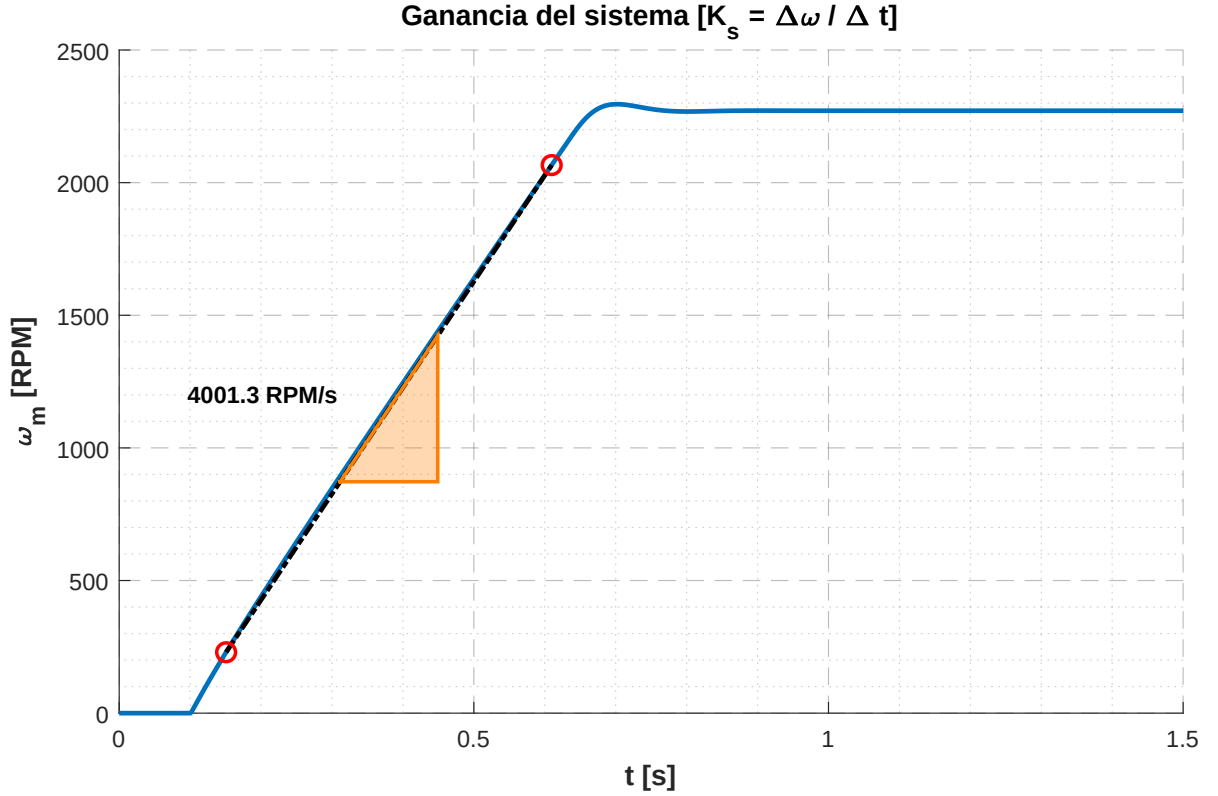


Figura 4.4: Ganancia del sistema K_S para la sintonía del PI de velocidad



La ganancia planta K_s corresponde a la pendiente de aceleración de la respuesta de velocidad. Según Linnertz (2016), se adopta un valor de $T_\sigma = 5 \text{ ms}$ como constante de menor de tiempo. Con estos criterios, se calculan los parámetros del controlador PI:

$$T_{i-os} = 20 \text{ [ms]}, \quad (4.14)$$

$$K_{p-os} = 0.025. \quad (4.15)$$

4.1.4. Sintonización del controlador PI de velocidad mediante el método de Ziegler-Nichols

Uno de los métodos propuestos por Ziegler-Nichols para sintonizar un controlador PID consiste en obtener experimentalmente la respuesta al escalón. Este método es válido cuando la respuesta se aproxima a una curva con forma de **S**, es decir, exhibe un tiempo

de retardo inicial L seguido de una dinámica aproximable por un sistema de primer orden con constante de tiempo T y valor de estado estacionario K (véase la Figura 4.5). A partir de estas constantes, se calculan los parámetros del PID según la Tabla 4.2 (Ogata, 2010; Åström & Hägglund, 2009).

Figura 4.5: Curva de respuesta en forma de S

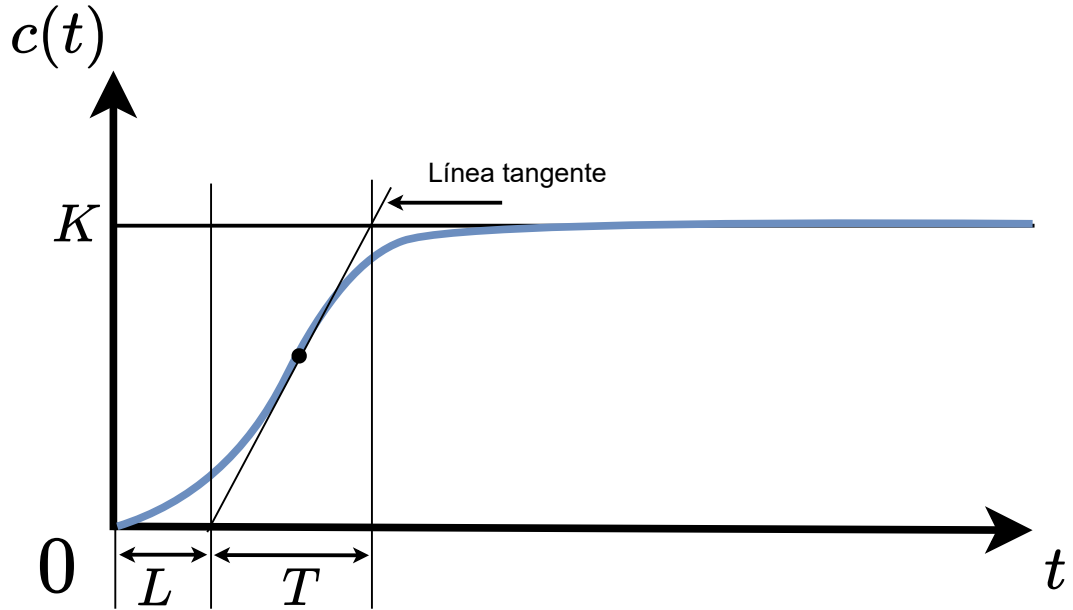


Tabla 4.2: Primer método de sintonía de Ziegler-Nichols basada en la respuesta al escalón

Tipo de controlador	K_p	T_i	T_d
P	$T/(KL)$	∞	0
PI	$0.9T/(KL)$	$L/0.3$	0
PID	$1.2T/(KL)$	$2L$	$0.5L$

Nota. Tomada y adaptada de Ogata (2010).

La estructura mostrada en la Figura 4.3 se utiliza para obtener la respuesta de velocidad representada en la Figura 4.6. Se aplicó una consigna de $i_{A-ref} = 1$ A en $t = 0.1$ s, y se determinaron la constante de retardo L , la constante de tiempo T , y la ganancia del sistema K . A partir de la curva de velocidad y el valor de la constante de tiempo menor

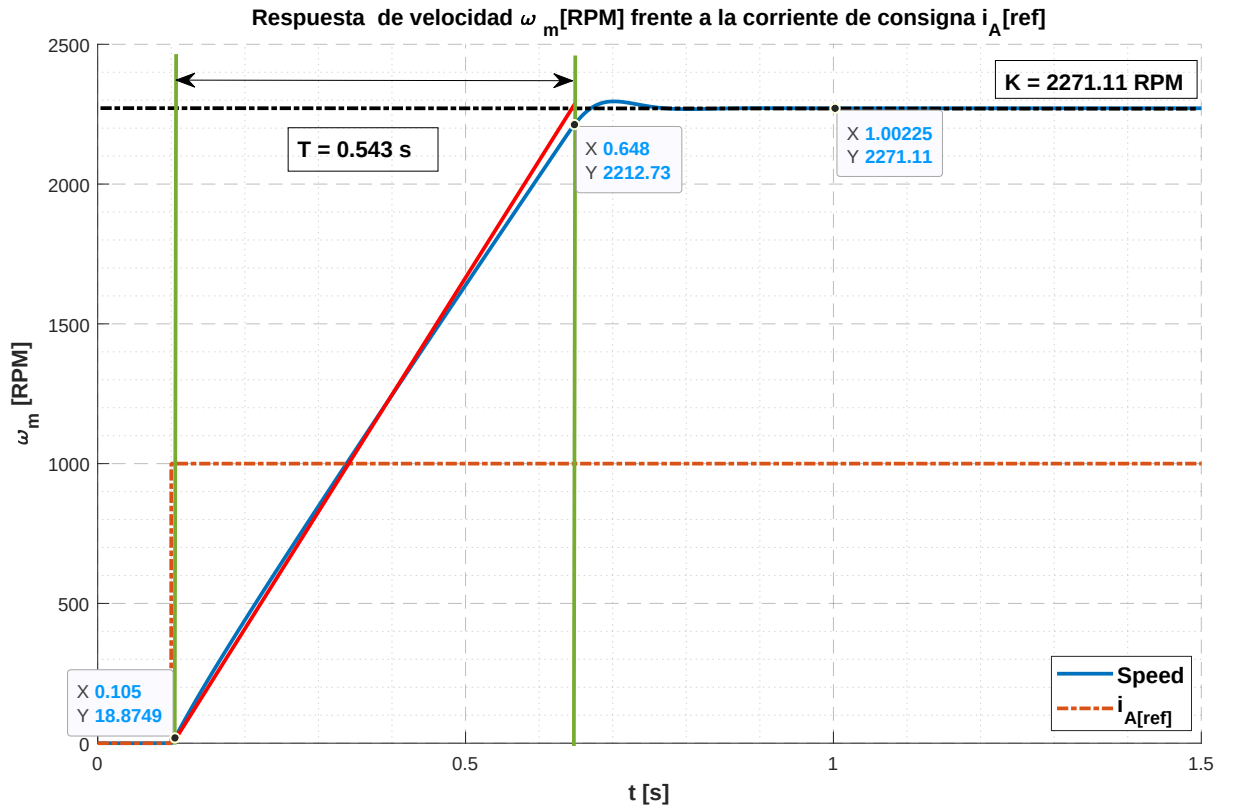
adoptado por Linnertz (2016) se obtuvo

$$T = 543 \text{ ms}, \quad (4.16)$$

$$L = 5 \text{ ms}, \quad (4.17)$$

$$K = 2271.1 \text{ RPM}. \quad (4.18)$$

Figura 4.6: Respuesta de velocidad frente a una consigna de corriente $i_{A-ref} = 1 \text{ A}$



Finalmente, usando la Tabla 4.2, se obtienen los parámetros del controlador PI:

$$T_{i-zn} = 16.67 \text{ [ms]}, \quad (4.19)$$

$$K_{p-zn} = 0.043. \quad (4.20)$$

4.1.5. Sintonización del controlador PI basado en aprendizaje por refuerzo (algoritmo TD3)

En el trabajo *Continuous control with deep reinforcement learning* de Lillicrap et al. (2015) se introduce el algoritmo DDPG, un método *model-free* actor-crítico basado en *deterministic policy gradient*¹ (DPG), capaz de operar en espacios de acción continuos. Los autores lo verifican en más de veinte sistemas físicos simulados—entre ellos de robótica, locomoción y control de vehículos—demostrando que DDPG puede aprender políticas de control incluso a partir de estados de entrada sin procesar.

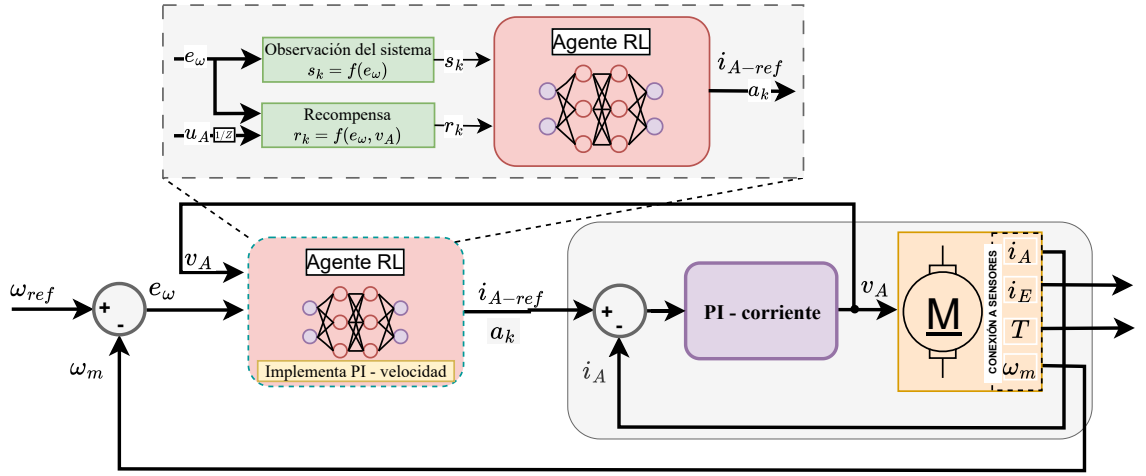
No obstante, DDPG sufre de problemas conocidos de sobreestimación de valores y cierta inestabilidad. Para mitigar estas deficiencias, Fujimoto et al. (2018) proponen TD3, una versión mejorada que incorpora mecanismos para reducir el sesgo, y lograr mayor estabilidad y rendimiento en tareas de control continuo.

Además de DDPG y TD3, existen múltiples algoritmos en la literatura (Barto & Sutton, 2018). En trabajos recientes se observa un uso frecuente de DDPG o su versión mejorada—TD3—debido a su desempeño robusto, su estabilidad relativa y su aplicabilidad práctica para sistemas de control y actuadores (Liu, 2024; Lakhani et al., 2022; Tufenkci et al., 2023; Alejandro-Sanjines et al., 2023; CusiHuallpa-Huamantupa et al., 2026).

Teniendo en cuenta los antecedentes revisados, se opta por emplear el algoritmo TD3 para la sintonización del lazo de control de velocidad. La arquitectura del sistema propuesto para esta sintonización se muestra en la Figura 4.7. Dicha arquitectura conserva el lazo de control interno de corriente y añade un agente de aprendizaje por refuerzo que ajusta el lazo externo de control de velocidad.

¹Del inglés, que traducido es gradiente de política determinística.

Figura 4.7: Estructura de sintonización del controlador PI de velocidad basado en TD3



4.1.5.1. Entorno, estado y acción

En el esquema propuesto en la Figura 4.7, el entorno consta del motor DC con devanado de campo en conjunto con el lazo interno de regulación de corriente. Siguiendo este diagrama y el de la Figura 4.8, el vector de estados observado por el agente está definido en función del error de velocidad como

$$s_k = f(e_\omega) = [e_\omega(t_k), \int_0^{t_k} e_\omega(\tau) d\tau]^T, \quad (4.21)$$

donde se usa el error de velocidad e_ω [rad/s] e integral del error porque el agente implementa un controlador PI. Esta representación se propaga a la red actor, como se muestra en la Figura 4.9 (Alejandro-Sanjines et al., 2023; Tüfenkçi et al., 2023).

La acción que ejerce el agente sobre el entorno es la señal de referencia de la corriente de armadura limitada al intervalo $[-3.5, 3.5]$ A:

$$a_k = i_{A-ref}(t_k). \quad (4.22)$$

Esta acción permite la regulación de velocidad siguiendo el esquema original del con-

trolador PI en cascada—configuración indispensable para el control de actuadores de potencia—como se muestra en la Figura 4.1.

4.1.5.2. Recompensa

La recompensa r_k , es una función del error de velocidad $e_\omega(t_k)$, y el voltaje de armadura $v_A(t_k)$. Y esta se define como una combinación de la función de costo basado en la teoría de control óptimo—que son sistemáticamente funciones cuadráticas— y una función exponencial cuadrática, tal como se usan en los trabajos de Nguyen et al. (2024), Alejandro-Sanjines et al. (2023) y Tufenkci et al. (2023):

$$r_k = f(e_\omega, v_A) = \exp \{-q_2 \cdot e_\omega^2(t_k)\} - q_1 \cdot e_\omega^2(t_k) - r \cdot v_A^2(t_k), \quad (4.23)$$

donde q_1 y q_2 son los coeficientes de penalización del seguimiento de trayectoria, y r el coeficiente de penalización de la señal de control. Westenbroek et al. (2022) aseveran que el proceso de diseño de las funciones de costo de un problema de control óptimo, como es el aprendizaje por refuerzo, es heurístico. Además Filipović et al. (2024) y Lewis et al. (2012) sostienen que los coeficientes de penalización se escogen mediante procesos de prueba y error, siguiendo la lógica: si el objetivo es tener errores cuadráticos pequeños, los coeficientes $q_{i=1,2}$ deben ser grandes, y en aplicaciones prácticas, escoger valores pequeños de r acelera la respuesta en lazo cerrado, mientras que si esta es grande, lo ralentiza.

Siguiendo la literatura y tomando como referencia los trabajos de Alejandro-Sanjines et al. (2023), Tufenkci et al. (2023) y Nguyen et al. (2024), junto con pruebas realizadas, los coeficientes empleados en este trabajo se definen en las ecuaciones (4.24–4.27).

$$q_1 = 1, \quad (4.24)$$

$$q_2 = 1 \times 10^4, \quad (4.25)$$

$$r = 0.02, \quad (4.26)$$

$$r_k = \exp \{-10000 \cdot e_\omega^2(t_k)\} - e_\omega^2(t_k) - 0.02 \cdot v_A^2(t_k). \quad (4.27)$$

4.1.5.3. Agente

Como se indicó en la introducción de esta sección, el algoritmo empleado para la sintonización del controlador PI es TD3. El esquema general del agente TD3 se presenta en la Figura 4.8, mientras que la Figura 4.9 muestra con mayor detalle su integración en el lazo externo de control de velocidad.

Figura 4.8: Agente RL que implementa el lazo PI de velocidad

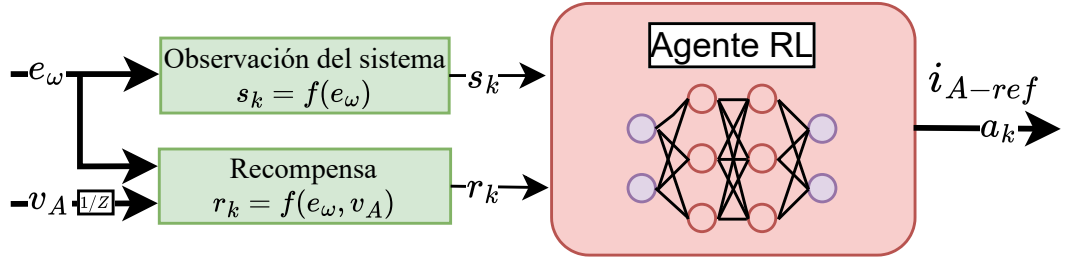
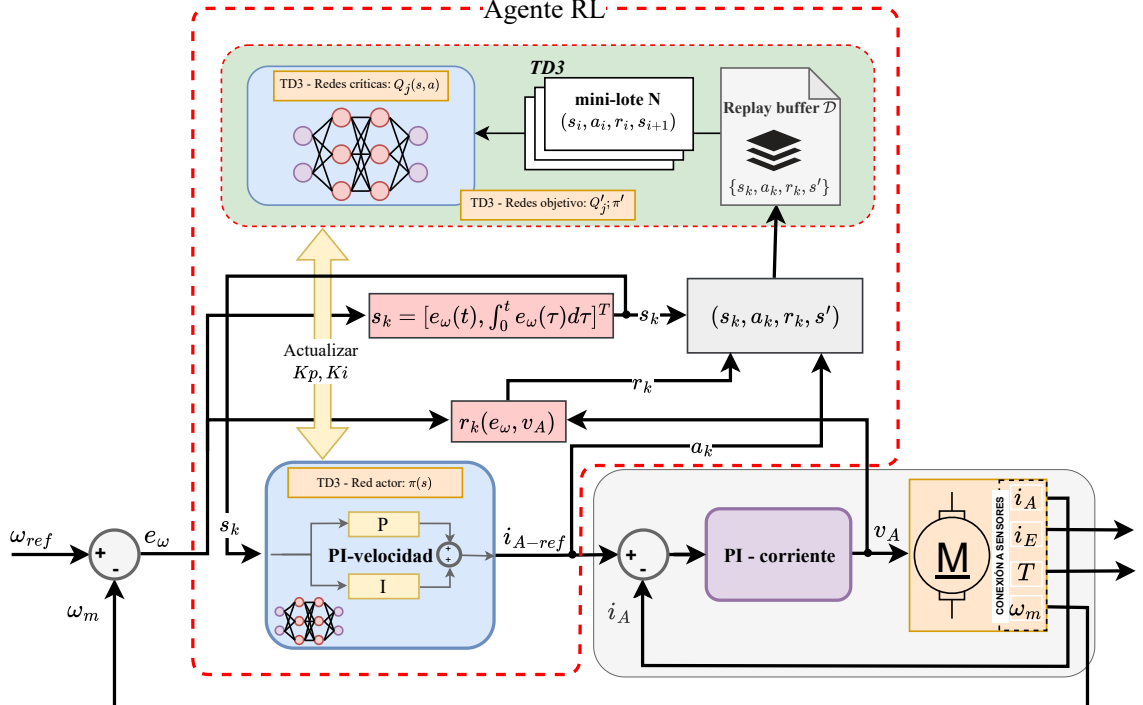


Figura 4.9: Arquitectura de la sintonización del controlador PI de velocidad usando TD3



Como se expuso en la sección teórica del algoritmo TD3 (Sección 2.2.5.4), el agente utiliza dos redes críticas y una red actor. En la Figura 4.10 se presenta la arquitectura de

la red crítica, y en la Tabla 4.3 las especificaciones de la misma. Estos valores se adoptan de trabajos previos y posterior pruebas heurísticas (Tufenkci et al., 2023).

Figura 4.10: Red crítica del agente TD3

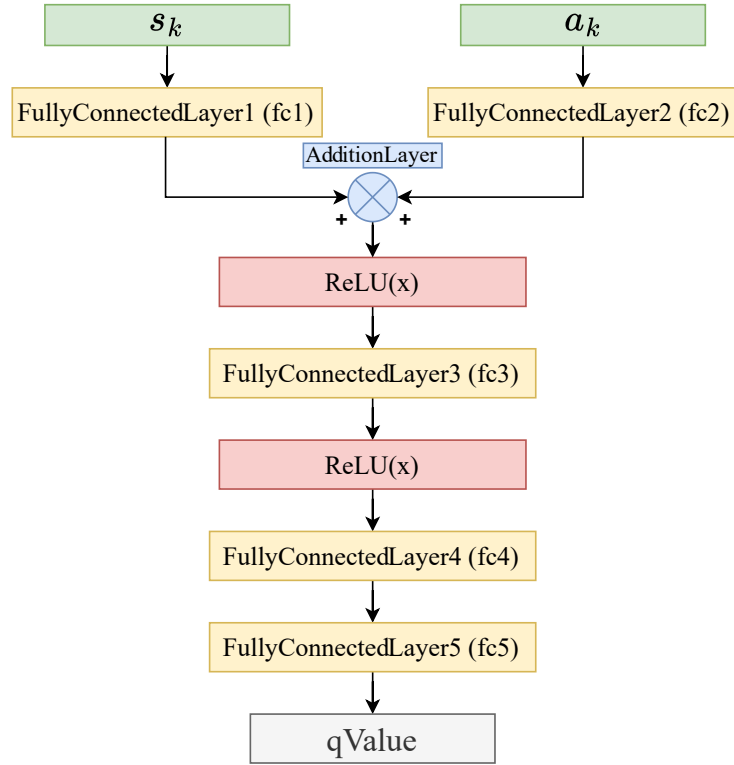


Tabla 4.3: Detalles de la arquitectura de la red crítica del agente TD3

Subred	Capa	Neuronas	Activación/Tipo
Red de estado	s_k	2	input
	fc1	128	-
Red de acción	a_k	1	input
	fc2	128	-
Red común	AdditionLayer	128	ReLU
	fc3	64	ReLU
	fc4	16	-
	fc5	1	-
	qValue	1	output

La red crítica recibe como entradas el estado observado s_k y la acción a_k del agente, definidos en las ecuaciones (4.21) y (4.22), respectivamente. Como se discutió en la sección 2.2.5, la red crítica aproxima la función de valor-acción $Q(s, a)$, la cual representa el retorno esperado a partir del estado s_k , ejecutar la acción a_k y seguir la política π . La

política π que maximice el retorno esperado a largo plazo se considera óptima y es la que se usa en la implementación.

La arquitectura de la red actor y las especificaciones de la misma se presentan en la Figura 4.11 y la Tabla 4.4, respectivamente.

Figura 4.11: Red actor del agente TD3

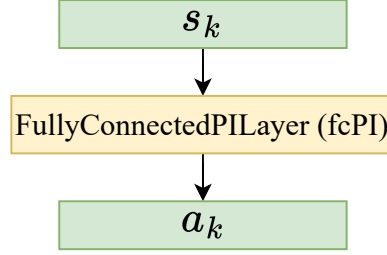


Tabla 4.4: Detalles de la arquitectura de la red actor del agente TD3

Capa	Neuronas	Activación/Tipo
s_k	2	input
fcPI	1	-
a_k	1	output

La red actor define un controlador PI; por ello, se emplea una capa *fullyConnectedPI* que recibe el estado s_k y genera como salida la acción $a_k = i_{A-ref}$. La salida de la red actor se define como

$$a_k = \theta^\pi \cdot s_k = [w_p, w_i] \begin{bmatrix} e_\omega(t_k) \\ \int_0^{t_k} e_\omega(\tau) d\tau \end{bmatrix} = i_{A-ref}(t_k), \quad (4.28)$$

donde los parámetros de la red neuronal actor $\theta^Q = [w_p, w_i]$ corresponden a las ganancias proporcional e integral del controlador PI de velocidad.

Las redes críticas $Q_1(s, a|\theta_1^Q)$ y $Q_2(s, a|\theta_2^Q)$, junto con sus respectivas redes objetivo $Q'_1(s, a|\theta_1^{Q'})$ y $Q'_2(s, a|\theta_2^{Q'})$, comparten la misma arquitectura y se entrenan conforme al Algoritmo 2. No obstante, sus parámetros no son necesariamente idénticos. El diseño de la red crítica y la red actor se basa en los trabajos de Alejandro-Sanjines et al. (2023), Tufenkci et al. (2023), Lakhani et al. (2022) y Cusihualpa-Huamantupa et al. (2026).

4.1.5.4. Entrenamiento (TD3-Algoritmo 2)

El proceso de entrenamiento se lleva a cabo en MATLAB/Simulink r2020b. En la Figura 4.12 se presenta el esquema de entrenamiento para la sintonización del controlador PI de velocidad. Como se muestra en la Figura 4.13, este trabajo emplea el bloque del agente de aprendizaje por refuerzo—agente RL o agente TD3—disponible en Simulink. El programa en MATLAB para la creación del agente TD3 se incluye en el Anexo A, mientras que los hiperparámetros del entrenamiento se detallan en la Tabla 4.5. Dichos hiperparámetros se definen a partir de pruebas heurísticas e iterativas de prueba y error, siguiendo las recomendaciones de CusiHuallpa-Huamantupa et al. (2026) y tomando como referencia inicial los trabajos de Alejandro-Sanjines et al. (2023), Tufenkci et al. (2023) y Lakhani et al. (2022).

Tabla 4.5: Variables de entrenamiento e hiperparámetros

Descripción	Parámetro	Valor
Tasa de aprendizaje del actor	β	5×10^{-3}
Tasa de aprendizaje del crítico	α	1.5×10^{-4}
Factor de descuento	γ	0.9995
Tamaño del Buffer de transiciones	$\mathcal{D}$	1×10^6
Tamaño del mini-lote	$\mathbf{N}$	1024
Tasa de la actualización suave	τ	0.005
Tiempo de muestreo	$T_{s-agent}$	0.004 s
Número de episodios	$\mathbf{M}$	400
Número de de pasos por episodio	$T_{f-agent}$	300
Frecuencia de actualización de las redes objetivo	policy_delay	$20 \times T_{s-agent}$
Varianza del ruido de exploración	$\mathcal{N}_k$	0.06
Varianza del ruido de la política objetivo	ϵ	0.05

Figura 4.12: Arquitectura de la sintonización en MATLAB/Simulink usando TD3

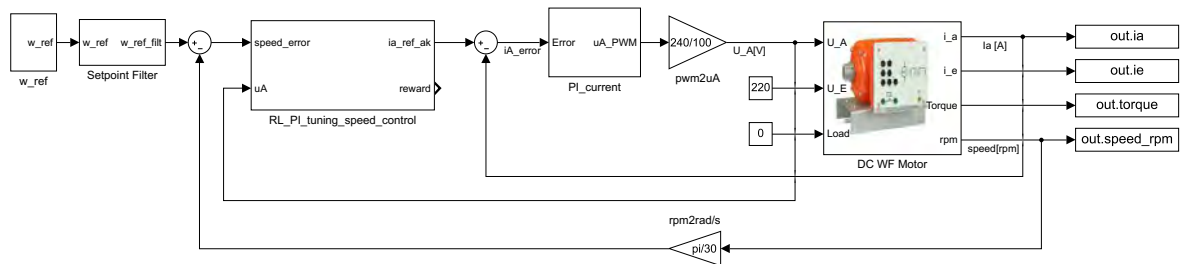
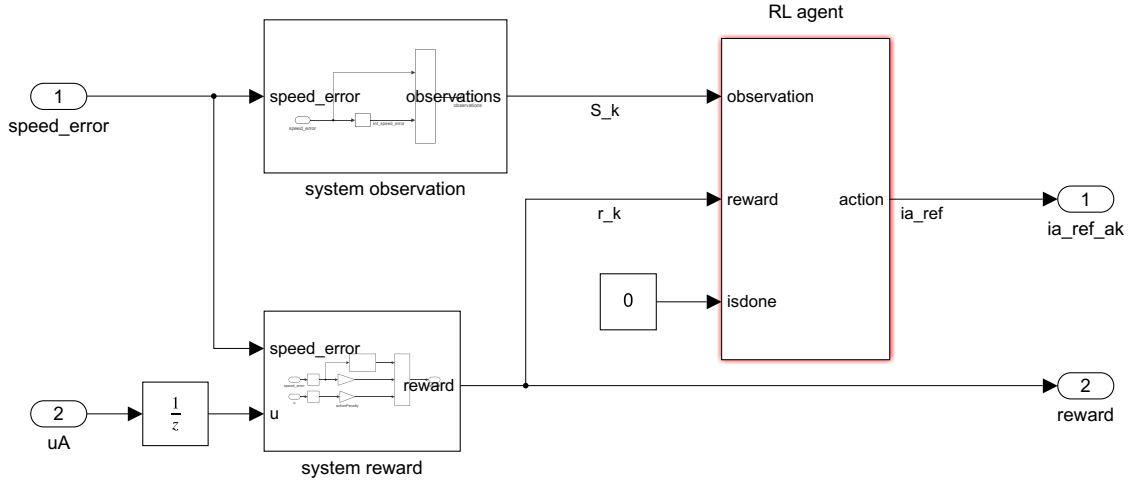


Figura 4.13: Bloque de control de velocidad con agente RL en Simulink



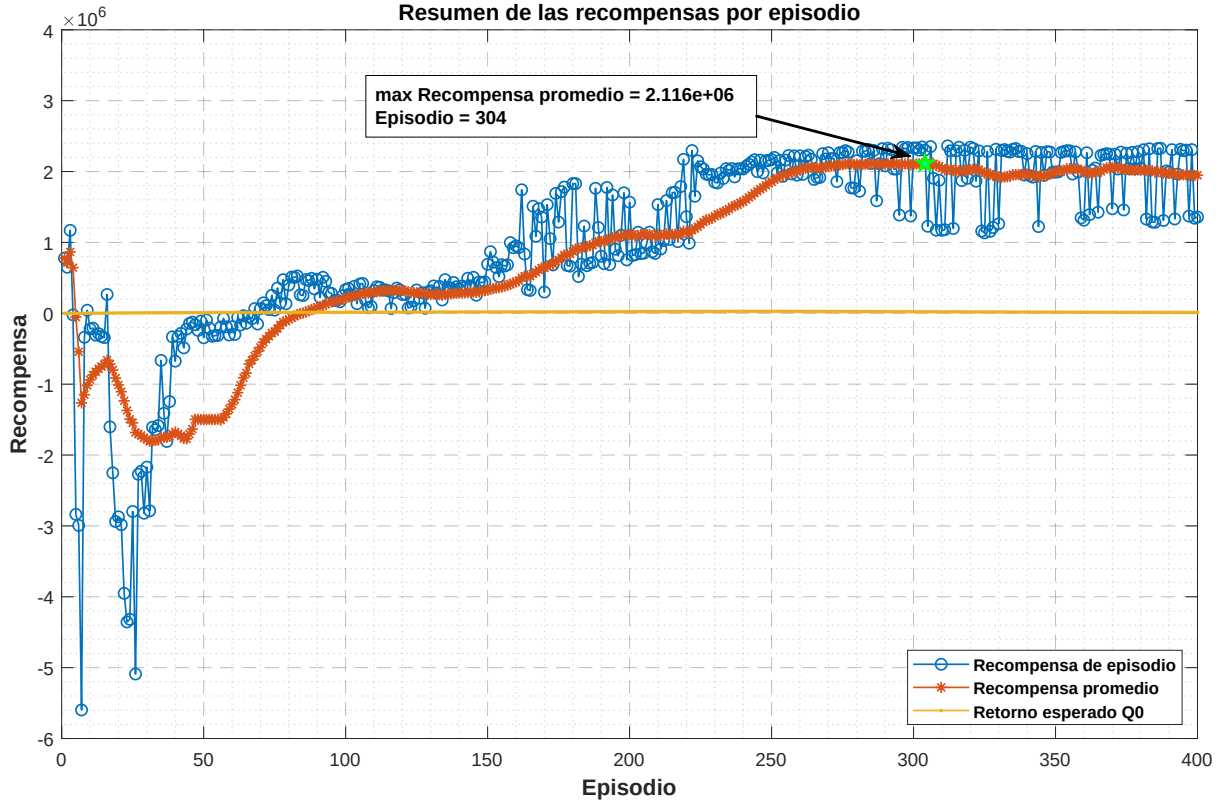
El entrenamiento se realizó en un entorno de simulación usando el modelo no lineal del motor DC, tal como se muestra en la Figura 4.12, en un computador cuyas características se detallan en la Tabla 4.6. El proceso de *training* se lleva a cabo durante 1.2 s por episodio, con una duración de 300 pasos de tiempo de muestreo $T_{s-agent}$. Durante este proceso, el replay buffer almacena un conjunto de $\mathcal{D} = 1 \times 10^6$ transiciones y, en cada paso de tiempo $T_{s-agent}$, se seleccionan $\mathbf{N} = 1024$ muestras aleatorias para realizar la regresión de los parámetros de las redes críticas y del actor mediante SGD. Adicionalmente, la actualización de las redes objetivo se retrasa cada `policy_delay` = 0.08 s, lo que contribuye a estabilizar el aprendizaje durante el entrenamiento (Fujimoto et al., 2018).

Tabla 4.6: Especificaciones técnicas del ordenador y el software

	Características
Sistema operativo	Windows 10 Pro - 64 bits
Procesador	Intel(R) Xeon(R) W-2123 CPU
Frecuencia del procesador	3.60 GHz
Núcleos del procesador	4
Tarjeta gráfica (GPU)	NVIDIA Quadro P2000 de 5 GB VRAM
Almacenamiento del ordenador	256 GB SSD + 1 TB HDD
Memoria RAM del ordenador	32 GB
Software utilizado	MATLAB/Simulink r2020b

Los resultados del entrenamiento en el entorno de MATLAB/Simulink se muestran en la Figura 4.14. Las recompensas por episodio exhiben alta variabilidad debido a la exploración que el agente realiza durante el entrenamiento (Cusihuallpa-Huamanttupa et al., 2026). Las líneas en naranja representan la media móvil calculada cada 40 episodios.

Figura 4.14: Proceso de entrenamiento del agente TD3 en MATLAB/Simulink



Nota. El entrenamiento tanto de las redes críticas como de la red actor se realizó en la GPU especificada en la Tabla 4.6. El tiempo de entrenamiento fue de 12 549 s (03:29:09).

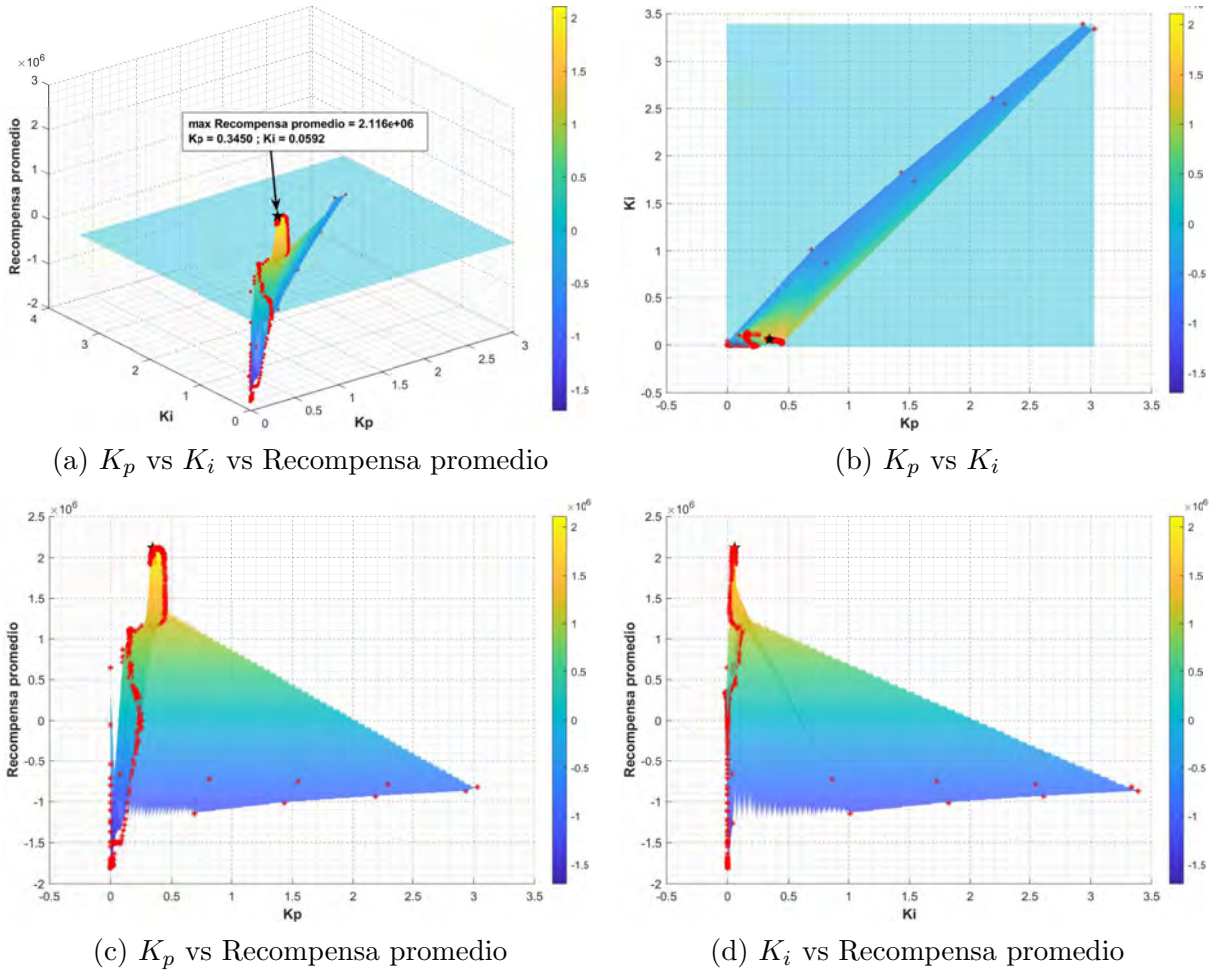
Dados los resultados del entrenamiento, los criterios para obtener las ganancias óptimas del controlador PI pueden basarse en superar un umbral específico de la recompensa promedio o en utilizar la última política explotada por el agente al finalizar el número de episodios (Tufenkci et al., 2023; Alejandro-Sanjines et al., 2023; Cusihuallpa-Huamanttupa et al., 2026). En esta investigación se permite la exploración durante todos los episodios y, al finalizar el entrenamiento, se seleccionan los parámetros de la red actor

$\theta^\pi = [w_p, w_i] = [K_p, K_i]$ que generan la mayor recompensa promedio, es decir,

$$[K_{p-opt}, K_{i-opt}] = \arg \max_{(K_p, K_i)} \{avgReward(K_p, K_i)\}, \quad (4.29)$$

donde $[K_{p-opt}, K_{i-opt}]$ representa el vector de parámetros óptimos del controlador PI de velocidad y $avgReward$ es la recompensa promedio asociada a las ganancias K_p y K_i .

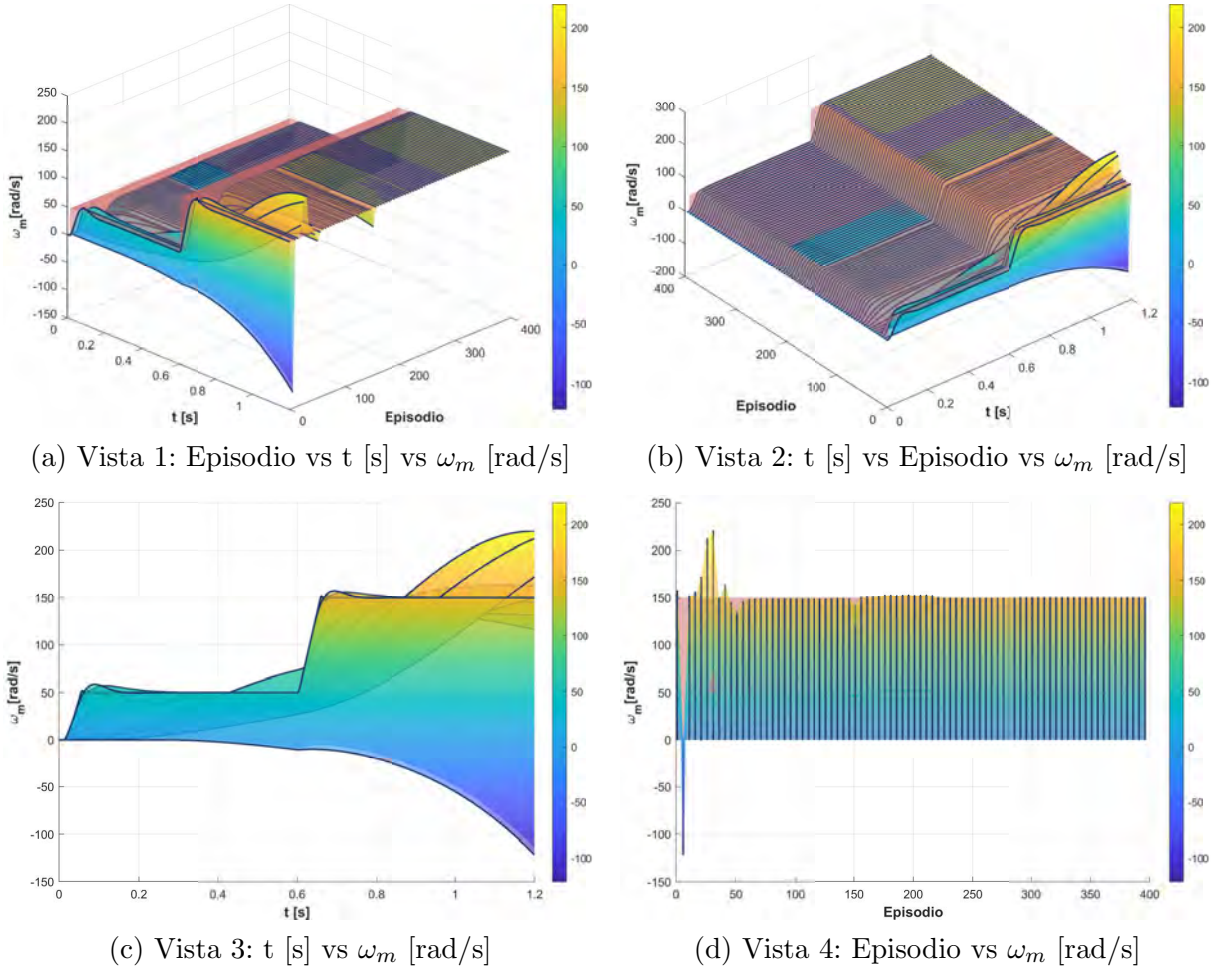
Figura 4.15: Resultados de la exploración del agente TD3 en el espacio de ganancias del controlador PI (K_p, K_i)



Nota. Las figuras representan la exploración del agente TD3 en el espacio de parámetros definido por todos los posibles valores $\theta^\pi = [K_p, K_i] = [w_p, w_i]$.

En la Figura 4.15 se exhibe la exploración del agente RL sobre el espacio de ganancias (K_p, K_i), mientras que en la Figura 4.16 se presenta la evolución de la respuesta del sistema ante los diferentes controladores PI implementados por el agente en cada episodio.

Figura 4.16: Evolución por episodio de la respuesta del motor DC en lazo cerrado



Nota. Las figuras presentan la evolución de las respuestas en lazo cerrado del sistema de control de velocidad del motor DC a medida que el agente RL explora e implementa las ganancias $[K_p, K_i]$ en cada episodio. Nótese que, en este caso, la velocidad se expresa en [rad/s].

Finalmente, de la ecuación (4.27) se infiere que el agente se entrena para el seguimiento de trayectoria y minimizar el esfuerzo de la señal de control (Nguyen et al., 2024; Tufenkci et al., 2023; Alejandro-Sanjines et al., 2023; Lewis et al., 2012). Al analizar los resultados presentados en las Figuras 4.14, 4.15 y 4.16, junto con la ecuación (4.29), se determinan que las ganancias óptimas del controlador PI de velocidad—escaladas en rad/s—corresponden al episodio 304 y son las siguientes:

$$K_{p-rl} = K_{p-opt} = 0.3450, \quad (4.30)$$

$$K_{i-rl} = K_{i-opt} = 0.0592. \quad (4.31)$$

4.1.5.5. Término no lineal del controlador PI de velocidad basado en TD3

En el diseño de controladores PID/PI, está bien establecido que la sintonización debe equilibrar objetivos como el rendimiento en el seguimiento de trayectoria, la capacidad de rechazo de perturbaciones y la robustez. Los marcos de ajuste multiobjetivo que ponderan estos criterios demuestran que ninguna sintonización puede optimizar simultáneamente todos los objetivos, especialmente en sistemas con comportamiento dinámico no lineal. En consecuencia, las decisiones de sintonización están intrínsecamente adaptados a los requerimientos del sistema (Zhang et al., 2021; Çelik et al., 2025; Sevilla-Hidalgo et al., 2025; The MathWorks, Inc., 2025b; Garpinger et al., 2014; Texas Instruments, 2021).

Estudios recientes destacan que los controladores PID/PI tradicionales presentan conflictos inherentes de desempeño: ganancias que mejoran el seguimiento de trayectoria generalmente degradan el rechazo de perturbaciones y viceversa, mientras que acciones integrales sintonizadas para reducir el sobreimpulso pueden comprometer la robustez y la precisión en estado estacionario de sistemas sujetos a perturbaciones, tales como motores DC. (Çelik et al., 2025; Çelik et al., 2024; Zhang et al., 2021; Campos-Salazar et al., 2024).

Dados los conflictos inherentes de la sintonización multiobjetivo de los controladores PID/PI y considerando los avances orientados a satisfacer la mayor cantidad posible de objetivos sin incrementar significativamente el costo computacional, este trabajo propone añadir un término no lineal al controlador PI de velocidad sintonizado mediante TD3. Dicho controlador ha sido ajustado para optimizar el seguimiento de trayectoria y minimizar el esfuerzo de la señal de control; no obstante, presenta limitaciones inherentes en el rechazo de perturbaciones y frente a no linealidades (Çelik et al., 2025).

Esta decisión se fundamenta en la literatura que utiliza enfoques híbridos, tales como la sintonización de los controladores PI y PI no lineales mediante algoritmos metaheurísticos inteligentes como PSO, GA y GWO (Çelik et al., 2024; Khellaf et al., 2025; Çelik et al., 2025; Tufenkci et al., 2023; Sevilla-Hidalgo et al., 2025). En particular, el presente trabajo combina un controlador PI ajustado mediante TD3 con una función no lineal adicional.

En las investigaciones desarrolladas por Khellaf et al. (2025), Çelik et al. (2024) y Çelik et al. (2025), las funciones no lineales empleadas en el diseño de controladores PI no lineales varían desde funciones potenciales y exponenciales hasta funciones trigonométricas. Tomando como referencia estos trabajos, se propone agregar una ganancia no lineal al término integral, definida como

$$k(e(t)) = 1 + \mu e^{-\rho e^2(t)}, \quad (4.32)$$

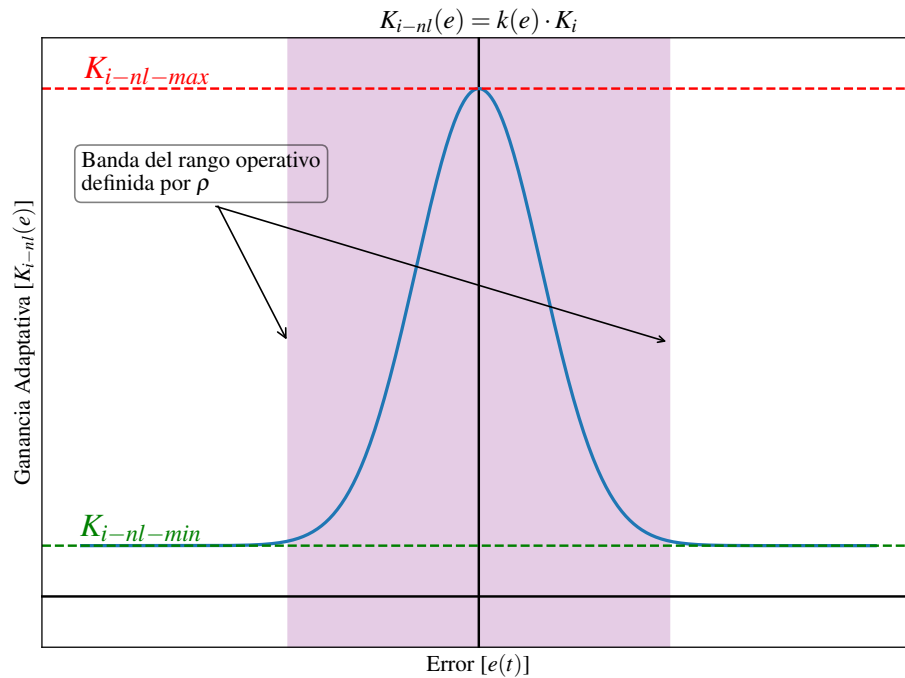
donde $e(t)$ es la señal de error, μ es el factor de escalamiento y ρ determina el ancho del rango operativo de la función no lineal.

Luego, utilizando la ganancia adaptativa $k(e(t))$ (ecuación (4.32)), la ganancia integral no lineal propuesta se define como

$$K_{i-nl}(e) = k(e) \cdot K_i, \quad (4.33)$$

y su comportamiento dinámico se muestra en la Figura 4.17.

Figura 4.17: Dinámica de la ganancia integral no lineal $K_{i-nl}(e)$



Debido a la forma de campana inducida por la función exponencial cuadrática negativa, la dinámica de la ganancia K_{i-nl} es más agresiva cuando el error $e(t)$ se confina dentro del rango operativo determinado por ρ , alcanzando su valor máximo $K_{i-nl-max}$ cuando $e(t) = 0$, lo que contribuye a reducir el error en estado estacionario. En contraste, cuando $e(t)$ se aleja de dicho rango, la ganancia disminuye progresivamente hasta su valor mínimo $K_{i-nl-min}$, favoreciendo una respuesta más suave y evitando sobreimpulsos excesivos.

El parámetro μ se selecciona considerando la ganancia integral obtenida por el método del óptimo simétrico. De las ecuaciones (4.14) y (4.15), la ganancia integral asociada al óptimo simétrico se calcula mediante la relación

$$K_{i-os} = \frac{K_{p-os}}{T_{i-os}} = 1.25. \quad (4.34)$$

Dado que las ganancias del controlador PI basado en TD3 se encuentran escaladas en rad/s, mientras que las obtenidas por el método del óptimo simétrico y Ziegler-Nichols están expresadas en RPM, la ganancia integral basada en TD3 convertida a la escala de RPM se calcula como

$$K_{i-rl}^{RPM} = \frac{30}{\pi} K_{i-rl} = 0.0062. \quad (4.35)$$

De las ecuaciones (4.34) y (4.35), se obtiene que $K_{i-os} \approx 201 K_{i-rl}^{RPM}$. Luego, con base en los criterios expuestos por Zhang et al. (2021), se adopta un factor máximo equivalente a la mitad de dicho valor, es decir, 100. Asimismo, a partir de la revisión de Çelik et al. (2025) y Khellaf et al. (2025), se selecciona $\rho = 0.01$, lo que delimita la banda operativa del error al intervalo $(-25, 25)$ rad/s, equivalente a $(-240, 240)$ RPM. En consecuencia, los parámetros del controlador PI—expresados en rad/s—basado en RL con término no lineal (véase la ecuación (4.33), (4.30) y (4.31)) quedan definidos por

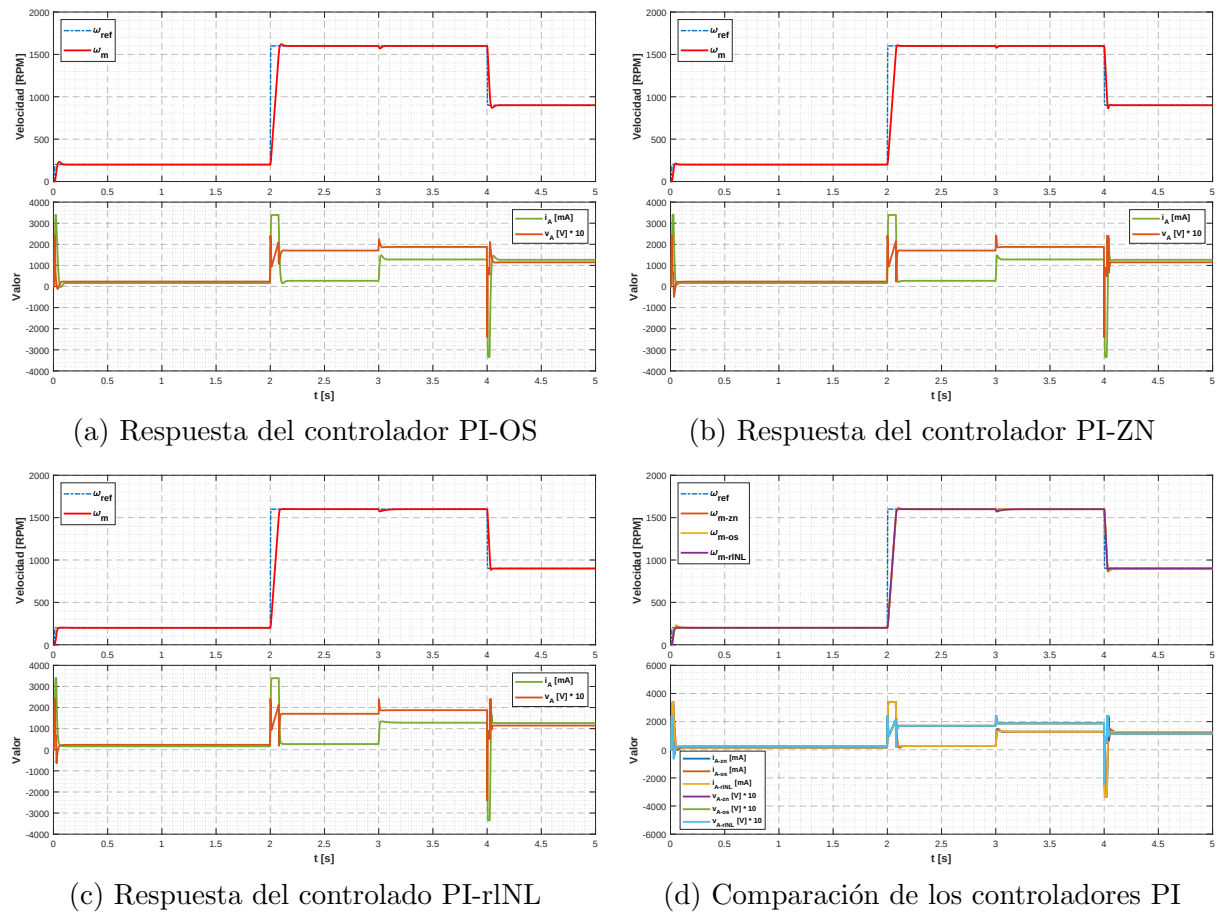
$$K_{p-rlNL} = K_{p-rl}, \quad (4.36)$$

$$K_{i-rlNL} = K_{i-rl}(1 + 99 \cdot \exp\{-0.01 \cdot e_{\omega}^2(t)\}). \quad (4.37)$$

4.1.6. Verificación de los controladores PI de velocidad en simulación

Los resultados de la verificación en simulación de los controladores PI de velocidad, diseñados en el entorno de MATLAB/Simulink, se presentan en la Figura 4.18. Por su parte, las Tablas 4.7 y 4.8 resumen los valores de los índices de desempeño transitorios y de seguimiento de trayectoria, respectivamente, los cuales son utilizados con el fin de evaluar cada controlador.

Figura 4.18: Respuestas en simulación de los controladores PI de velocidad



Nota. La prueba de simulación consiste en aplicar tres cambios de consigna en los instantes 0.01s, 2s y 4s, junto con una perturbación de par de $1\text{ N}\cdot\text{m}$ en el instante 3s. Dado que se evalúan tres cambios de referencia, se obtienen tres respuestas transitorias. Para el análisis del tiempo de establecimiento t_s se adopta el criterio del 1% para referencias mayores a 1000 RPM y el criterio del 2% en caso contrario. Los sufijos *OS*, *ZN* y *rlNL*, introducidos en la figura, corresponden a los métodos de sintonización óptimo simétrico, Ziegler-Nichols y TD3 con término no lineal, respectivamente.

Tabla 4.7: Indicadores de desempeño transitorio del motor DC frente a cambios de referencia bajo controladores PI en simulación

Métrica	PI-OS	PI-ZN	PI-rlNL	Mejor
$t_{s1}[s]$	0.0890	0.0675	0.0454	PI-rlNL
$M_{p1}(\%)$	15.700	6.285	1.222	PI-rlNL
$t_{s2}[s]$	0.1096	0.0838	0.0857	PI-ZN
$M_{p2}(\%)$	1.226	0.499	0.076	PI-rlNL
$t_{s3}[s]$	0.0728	0.0446	0.0411	PI-rlNL
$M_{p3}(\%)$	-3.417	-4.181	-1.894	PI-rlNL

Tabla 4.8: Indicadores de desempeño del motor DC en seguimiento de trayectoria y consumo energético bajo controladores PI en simulación

Métrica	PI-OS	PI-ZN	PI-rlNL	Mejor
σ_e	109.4237	109.4155	109.4006	PI-rlNL
ISE	4.8826×10^8	4.8763×10^8	4.8780×10^8	PI-ZN
IAE	6.1018×10^5	5.9159×10^5	6.0156×10^5	PI-ZN
ITAE	1.0945×10^{10}	1.0623×10^{10}	1.0833×10^{10}	PI-ZN
ITSE	8.4724×10^{12}	8.4612×10^{12}	8.4641×10^{12}	PI-ZN
$v_{A-RMS} [V]$	125.5913	125.8336	125.7446	PI-OS
$i_{A-RMS} [A]$	0.9654	0.9706	0.9654	PI-OS
$P [W]$	121.2429	122.1344	121.3982	PI-OS

Una conclusión directa a partir de los índices de desempeño presentados es que el controlador PI-rlNL no presenta sobreimpulsos superiores al 2% respecto del valor de consigna y exhibe, en promedio, un tiempo de establecimiento menor que los controladores PI-OS y PI-ZN.

No obstante, los resultados de simulación indican que, aunque el controlador PI-rlNL supera al PI-OS en métricas de seguimiento, este último resulta más eficiente en términos de consumo energético. Asimismo, el controlador PI-ZN supera a los controladores PI-rlNL y PI-OS en la mayoría de métricas de seguimiento, seguido por el PI-rlNL.

Sin embargo, estos resultados corresponden a un entorno ideal de simulación, donde las limitaciones propias de la implementación real no intervienen. Por lo tanto, no pueden considerarse concluyentes si se pretende emitir un juicio confiable sustentado en evidencia experimental.

4.2. Diseño del regulador cuadrático lineal - LQR

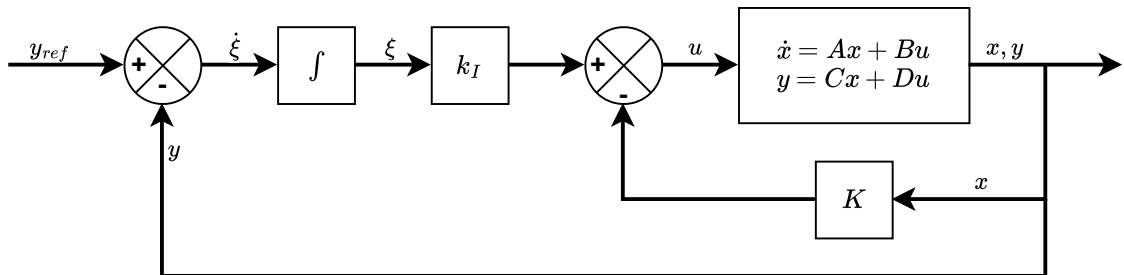
Aunque el LQR proporciona un desempeño óptimo bajo suposiciones ideales de modelamiento, los sistemas de control de motores generalmente se implementan con estructuras en cascada basadas en controladores PI debido a su robustez, simplicidad y facilidad de implementación en la industria (Alejandro-Sanjines et al., 2023; Çelik et al., 2024; Texas Instruments, 2021; Bose, 2002; Ogata, 2010). No obstante, en este trabajo se diseña el controlador LQR a fin de demostrar su factibilidad y analizar su desempeño en el control de velocidad de un motor DC con devanado de campo.

En esta sección se describe el proceso de diseño del LQR mediante la solución de la ecuación de Riccati y técnicas de aprendizaje por refuerzo, específicamente el algoritmo Generalized Policy Gradient.

4.2.1. Arquitectura del controlador LQR

Según Ogata (2010), en el diseño de servosistemas la arquitectura del LQR que se emplea para el seguimiento de trayectoria se muestra en la Figura 4.19. Este esquema incorpora un término integral del error con el fin de lograr un error nulo en estado estacionario; es decir, la arquitectura amplía el controlador LQR mediante una acción integral en su formulación, similar a la de un controlador PI, lo que asegura el seguimiento de referencia sin error en régimen permanente. (Satrianata, 2024; Eviningsih et al., 2025).

Figura 4.19: Arquitectura de seguimiento del controlador LQR



Del esquema en la Figura 4.19, considerando la señal de salida escalar y y la señal de control escalar u , se definen

$$\dot{x} = Ax + Bu, \quad (4.38)$$

$$y = Cx, \quad (4.39)$$

$$u = -Kx + k_I \xi, \quad (4.40)$$

$$\dot{\xi} = y_{ref} - y = y_{ref} - Cx, \quad (4.41)$$

donde se aumenta el estado ξ , que representa la integral del error de referencia $\dot{\xi} = y_{ref} - y$.

De las ecuaciones (4.38–4.38) se plantea el espacio de estados aumentado

$$\begin{pmatrix} \dot{x} \\ \dot{\xi} \end{pmatrix} = \begin{pmatrix} A & \mathbf{0} \\ -C & 0 \end{pmatrix} \begin{pmatrix} x \\ \xi \end{pmatrix} + \begin{pmatrix} B \\ 0 \end{pmatrix} u + \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} y_{ref}. \quad (4.42)$$

Para el diseño del controlador LQR la referencia se centra en cero, $y_{ref} = 0$ (Chirme-Sisa et al., 2024). Por tanto, el modelo aumentado en espacio de estados se reduce a

$$\begin{pmatrix} \dot{x} \\ \dot{\xi} \end{pmatrix} = \begin{pmatrix} A & \mathbf{0} \\ -C & 0 \end{pmatrix} \begin{pmatrix} x \\ \xi \end{pmatrix} + \begin{pmatrix} B \\ 0 \end{pmatrix} u. \quad (4.43)$$

Asumiendo que el motor DC no experimenta carga ni resistencia debido a la fricción, es decir $M_L = 0$ y $M_b = 0$, el modelo lineal en espacio de estados en (3.6) se reduce a

$$\begin{pmatrix} \frac{d}{dt}\omega_m \\ \frac{d}{dt}i_A \end{pmatrix} = \begin{pmatrix} 0 & \frac{C_m \Psi_{E_{ss}}}{\Theta_M} \\ -\frac{C_m \Psi_{E_{ss}}}{L_A} & -\frac{R_A}{L_A} \end{pmatrix} \begin{pmatrix} \omega_m \\ i_A \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{L_A} \end{pmatrix} v_A, \quad (4.44)$$

y el estado ξ queda definido por

$$\dot{\xi} = e_\omega = \omega_{ref} - \omega_m, \quad (4.45)$$

$$\xi = \int_0^t e_\omega dt. \quad (4.46)$$

De la ecuación (4.43), la representación en espacio de estados aumentado del sistema queda definida por

$$\begin{pmatrix} \frac{d}{dt}\omega_m \\ \frac{d}{dt}i_A \\ \frac{d}{dt}\xi \end{pmatrix} = \begin{pmatrix} 0 & \frac{C_m\Psi_{Ess}}{\Theta_M} & 0 \\ -\frac{C_m\Psi_{Ess}}{L_A} & -\frac{R_A}{L_A} & 0 \\ -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \omega_m \\ i_A \\ \xi \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{L_A} \\ 0 \end{pmatrix} v_A + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega_{ref} \quad (4.47)$$

Además, la ganancia de control óptimo y la señal de control son (véase las ecuaciones (2.3) y (4.40)):

$$K = [k_\omega, k_{i_A}, -k_I], \quad (4.48)$$

$$u = -[k_\omega, k_{i_A}, -k_I] \begin{bmatrix} \omega_m \\ i_A \\ \xi \end{bmatrix} = -[k_\omega, k_{i_A}] \begin{bmatrix} \omega_m \\ i_A \end{bmatrix} + k_I \xi. \quad (4.49)$$

De las relaciones (4.45), (4.46) y (4.49), si se centra la referencia en cero, $\omega_{ref} = 0$, la señal de control se puede expresar de la siguiente manera:

$$u = -k_\omega \omega_m - k_{i_A} i_A - k_I \int_0^t \omega_m dt. \quad (4.50)$$

Siguiendo el trabajo de Chirme-Sisa et al. (2024) para definir la arquitectura de seguimiento basada en LQR, se modifica la ecuación (4.50) como sigue:

$$u = \left(k_\omega(0 - \omega_m) + k_I \int_0^t (0 - \omega_m) dt \right) - k_{i_A} i_A.$$

Dado que el objetivo es el seguimiento de la velocidad de referencia, Chirme-Sisa et al. (2024) proponen reemplazar el término constante cero por la señal de referencia ω_{ref} , obteniéndose

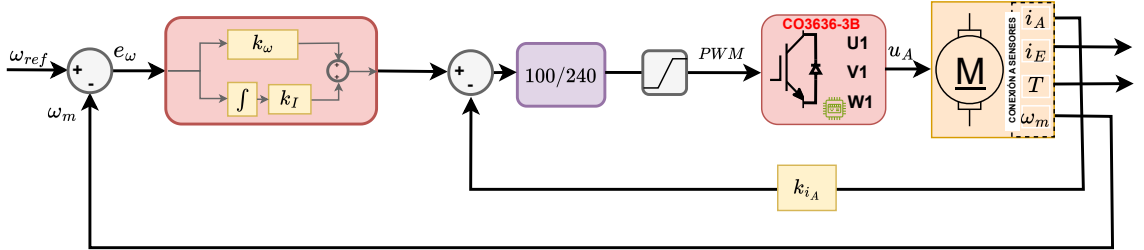
$$u = \left(k_\omega(\omega_{ref} - \omega_m) + k_I \int_0^t (\omega_{ref} - \omega_m) dt \right) - k_{i_A} i_A.$$

Por lo tanto la señal de control—que para el motor DC es el voltaje de armadura v_A —queda expresada como

$$v_A = u = \left(k_\omega e_\omega + k_I \int_0^t e_\omega dt \right) - k_{i_A} i_A. \quad (4.51)$$

La expresión en paréntesis de la ecuación (4.51) recuerda a un controlador PI. Tomando en cuenta esta observación, la arquitectura final que se emplea en este trabajo se muestra en la Figura 4.20. Esta estructura, además de embeber un controlador PI, permite lidiar con el fenómeno de la sobre-integración (*windup*²) de la componente integral del controlador, el cual puede ocurrir cuando la acción integradora acumula error mientras la señal de control se encuentra limitada por las saturaciones físicas del actuador, provocando sobreimpulsos no deseados, prolongados tiempos de establecimiento y a veces inestabilidad (Al-Haddad, 2018; Chen et al., 2025; Yapp et al., 2024).

Figura 4.20: Arquitectura del LQR para el control de velocidad del motor DC



El diseño empleado hace factible que la señal de control se limite a un rango de operación aceptable, ya que métodos de protección—denominados *anti-windup*—, como los expuestos en la sección 4.1.1, han sido ampliamente estudiados y aplicados en controladores PI (Chen et al., 2025; Yapp et al., 2024). En esta tesis, la señal de control v_A se limita a un rango de $[0 \text{ V}, 240 \text{ V}]$.

²Del inglés, que en este contexto hace referencia a la saturación de la acción integral o a la sobre-integración, el cual se da cuando $|e(t)| \neq 0$ y $|u(t)| = |u_{max}|$.

4.2.2. Sintonización del controlador LQR basado en la solución a la ecuación de Riccati (algoritmo de Hewer)

Para el seguimiento de trayectoria se emplea el modelo en espacio de estados aumentado definida en (4.47). Si $\omega_{ref} = 0$, este modelo se reduce a

$$\begin{pmatrix} \frac{d}{dt}\omega_m \\ \frac{d}{dt}i_A \\ \frac{d}{dt}\xi \end{pmatrix} = \begin{pmatrix} 0 & \frac{C_m\Psi_{Ess}}{\Theta_M} & 0 \\ -\frac{C_m\Psi_{Ess}}{L_A} & -\frac{R_A}{L_A} & 0 \\ -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \omega_m \\ i_A \\ \xi \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{L_A} \\ 0 \end{pmatrix} v_A, \quad (4.52)$$

el cual se discretiza usando la herramienta `c2d` de MATLAB (consúltase el Anexo B) (Kuczmann, 2024).

La función de costo cuadrática discreta de horizonte infinito está dada por

$$J(u) = \sum_{k=1}^{\infty} (x_k^T Q x_k + u_k^T R u_k) = \sum_{k=1}^N (x_k^T Q x_k + u_k^T R u_k) + \sum_{k=N+1}^{\infty} (x_k^T Q x_k + u_k^T R u_k). \quad (4.53)$$

No obstante, si se escoge un horizonte finito N suficientemente largo de tal manera que los estados terminales tiendan cero, la aproximación

$$J(u) = \sum_{k=1}^N (x_k^T Q x_k + u_k^T R u_k) + x_N^T Q_f x_N \approx \sum_{k=1}^N (x_k^T Q x_k + u_k^T R u_k) \quad (4.54)$$

es válida (Lewis et al., 2012). Donde Q_f es la matriz de costo terminal.

Las matrices de penalización

$$Q = \begin{pmatrix} q_{11} & 0 & 0 \\ 0 & q_{22} & 0 \\ 0 & 0 & q_{33} \end{pmatrix}, \quad (4.55)$$

$$R = r, \quad (4.56)$$

se eligen de acuerdo al conocimiento del sistema y los objetivos de control, generalmente basándose en estudios previos, criterios de diseño y ajuste heurístico como sugieren Filipović et al. (2024), Brunton (2017) y Lewis et al. (2012). Los pesos de q_{11} , q_{22} , q_{33} y r penalizan el error de velocidad e_ω , la corriente i_A , la integral de error ξ y la señal de control v_A , respectivamente, de forma que un mayor valor en cada elemento de Q o de R implica una mayor penalización sobre la magnitud de la variable asociada en la función de costo (Chirme-Sisa et al., 2024; Brunton, 2017; Lewis et al., 2012).

Luego de pruebas en el entorno de simulación de MATLAB/Simulink y posterior verificación del funcionamiento del sistema de control de velocidad tanto en el modelo lineal como no lineal, las matrices Q y R adoptadas en esta tesis son:

$$Q = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 200 \end{pmatrix}, \quad (4.57)$$

$$R = 1.8. \quad (4.58)$$

La solución de la ecuación de Riccati discreta permite obtener la ley de control óptima K . Para tal fin se utiliza el Algoritmo 5, que corresponde al método iterativo de Hwer sugerido por The MathWorks, Inc. (2026) y Lewis et al. (2012) para sistemas en tiempo discreto. La discretización del sistema con el periodo de muestreo base $T_s = 0.000125$ s, así como la implementación del algoritmo en MATLAB, se describen en el Anexo B.

Algoritmo 5 Método de Hwer para la solución de la ecuación de Riccati

- 1: Inicializar las matrices $Q = Q^T \geq 0$; $R = R^T \geq 0$; $Qf = Q$; $P = Qf$;
- 2: **Para** $j = N$, 0 **hacer**

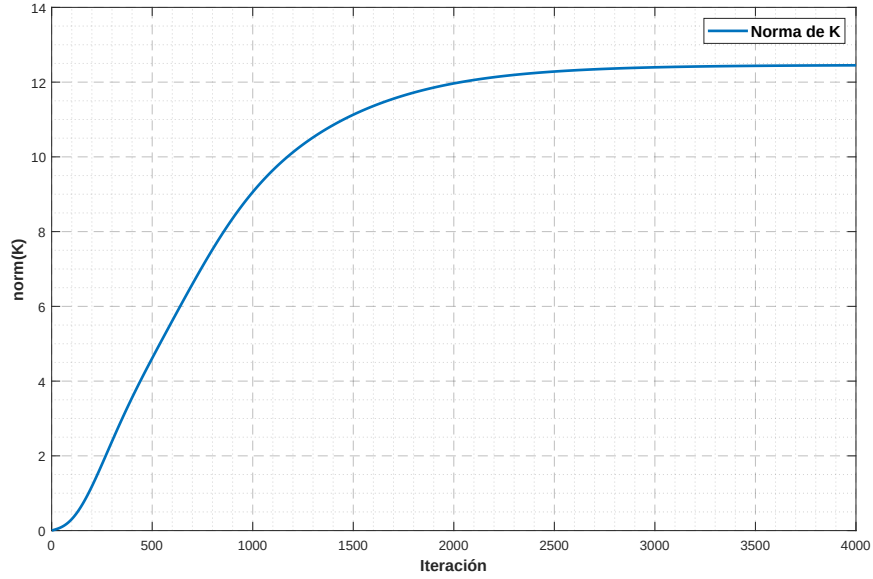
$$\begin{aligned} S &= R + B^T P B \\ K_j &= S^{-1} B^T P A \\ P &= Q + A^T P A - A^T P B K_j \end{aligned}$$

- 3: **fin Para**

$$K = (R + B^T P B)^{-1} B^T P A$$

Se establece un valor de $N = 4000$ iteraciones. Este valor número se elige de manera que la norma euclidiana del vector de control óptimo K converja a un valor estacionario, lo que indica estabilidad numérica y consistencia en la solución de la ecuación de Riccati, tal como se muestra en la Figura 4.21.

Figura 4.21: Norma del vector de control K - $\|K\|$



Finalmente el algoritmo sugiere la siguiente ganancia de control óptima:

$$K = [k_\omega, k_{i_A}, -k_I] = [0.5550, 6.6301, -10.5228]. \quad (4.59)$$

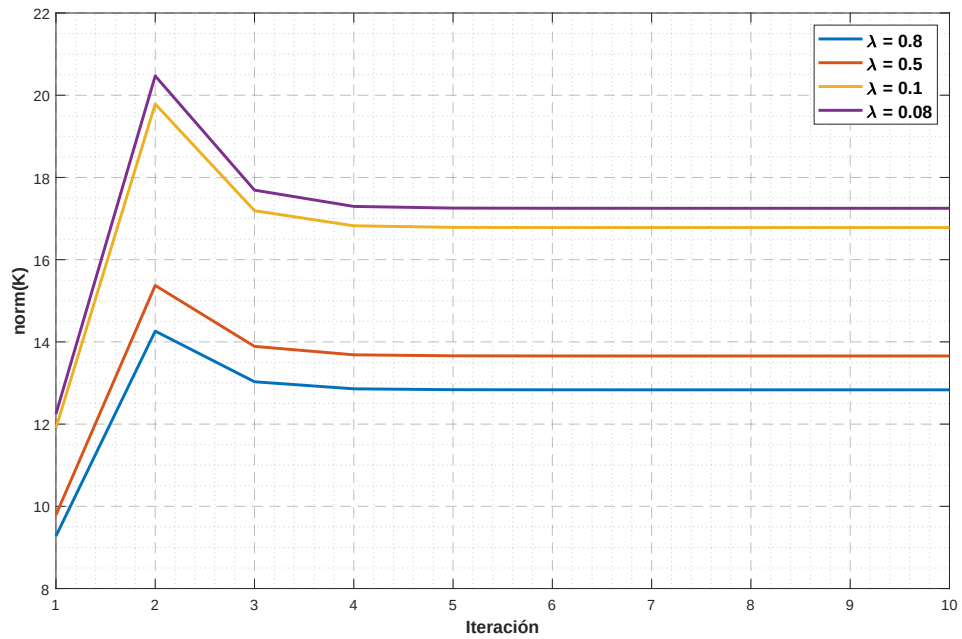
4.2.3. Sintonización del controlador LQR basado en aprendizaje por refuerzo (algoritmo MGPI)

El algoritmo GPI se emplea en problemas de control óptimo, dado que el aprendizaje por refuerzo y el control óptimo comparten fundamentos teóricos comunes basados en programación dinámica y en la optimización de políticas de control (Barto & Sutton, 2018). En el contexto del controlador LQR, la solución óptima se vincula formalmente con la ecuación de HJB, cuya formulación lineal-cuadrática conduce a la ecuación algebraica de Riccati (Lewis et al., 2012; Lee & Sutton, 2021; Li et al., 2024; Filipović et al., 2024).

Tomando como referencia los trabajos de Filipović et al. (2024) y Li et al. (2024), el algoritmo seleccionado para esta investigación es el Algoritmo 4, el cual proporciona la solución de la ecuación algebraica discreta de Riccati (DARE³). Se opta por la versión del algoritmo GPI basada en el modelo dinámico del sistema (MGPI), dado que incorpora un parámetro de estabilidad λ que puede ajustarse para modificar el comportamiento del proceso de convergencia y estabilidad del controlador implementado. Según Filipović et al. (2024), valores más pequeños de λ tienden a acelerar la convergencia de los estados de sistema hacia el equilibrio deseado.

Las matrices de penalización Q y R , junto con el modelo definido en (4.57), (4.58) y (4.52), respectivamente, adoptadas en la sintonización del controlador LQR en la sección anterior, se mantienen. A partir de dichas matrices, y siguiendo el Procedimiento heurístico sugerido por Filipović et al. (2024), se evalúan distintos valores de λ , cuyos efectos de sobre la convergencia del algoritmo se muestra en la Figura 4.22. Durante este proceso se verifica que el sistema en lazo cerrado permanezca estable y que la señal de control resultante se mantenga dentro de los límites de operación establecidos.

Figura 4.22: Norma del vector de control K - $\|K\|$ - MGPI



³Del inglés *Discrete-time Algebraic Riccati Equation*

Los valores de inicialización que producen las respuestas mostradas en la Figura 4.22 se especifican a continuación:

$$P_j = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (4.60)$$

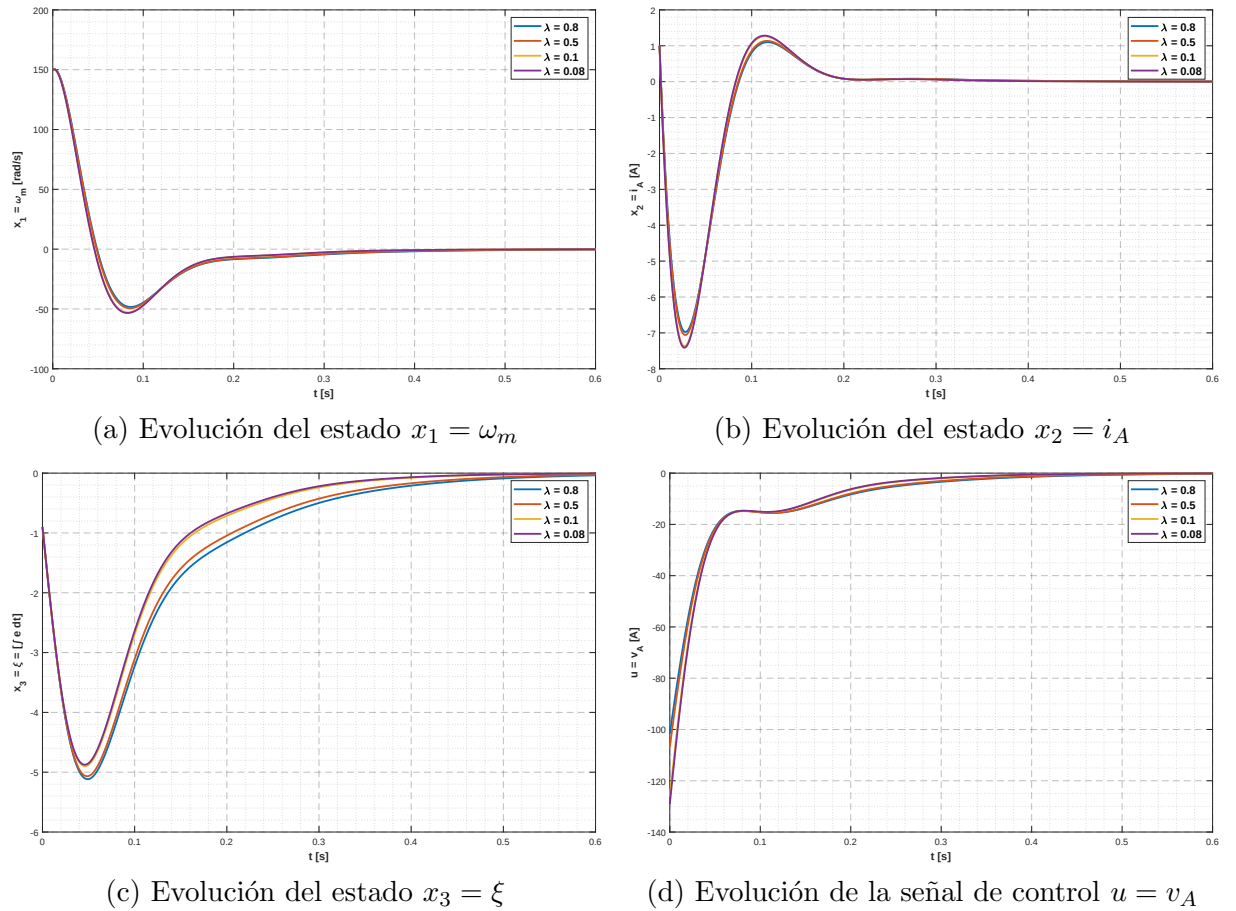
$$T_s = 0.000\,125\text{ s}, \quad (4.61)$$

$$l = 1000, \quad (4.62)$$

$$\lambda = 0.8, 0.5, 0.1, 0.08, \quad (4.63)$$

donde T_s corresponde al periodo de muestreo base del sistema de servoaccionamiento de Lucas Nülle. La implementación del algoritmo en MATLAB se presenta en el Anexo C.

Figura 4.23: Evolución de los estados para diferentes valores de λ



La Figura 4.23 muestra la evolución de estados para distintos valores de λ . A partir de la de los estados $x_1 = \omega_m$ y $x_3 = \xi$ se evidencia, como sugiere Filipović et al. (2024), que la respuesta del sistema es más rápida cuando λ toma un valor pequeño ($\lambda = 0.08$).

Dado que se busca una convergencia rápida del sistema respecto de la variable velocidad, se evalúa la regulación de los estados $x_1 = \omega_m$ y $x_3 = \xi$ para los distintos valores de λ . La evaluación se lleva a cabo aplicando la norma L1:

$$norm(x_s, 1) = ||x_s||_1 = \sum_{i=1}^N |x_{s,i}|, \quad (4.64)$$

donde x_s representa el vector de evolución de estados.

Tabla 4.9: Evaluación de convergencia de los estados ω_m y ξ

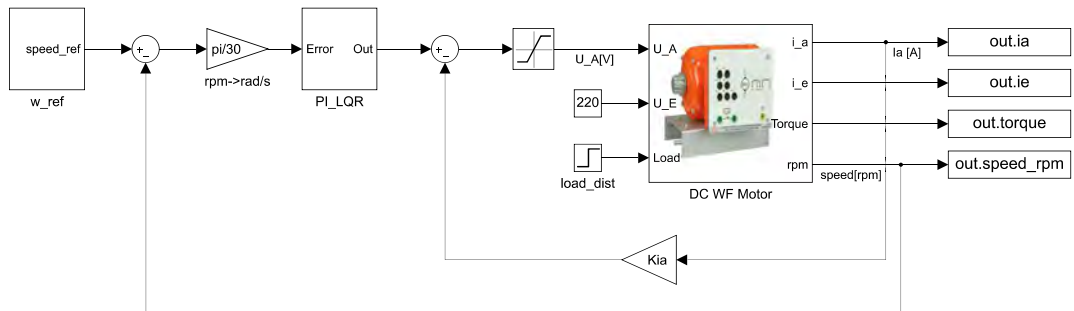
Métrica	$\lambda = 0.8$	$\lambda = 0.5$	$\lambda = 0.1$	$\lambda = 0.08$	Mejor
$ \omega_m $	7512.7	7441.9	7184.9	7149.3	$\lambda = 0.08$
$ \xi $	579.2	549.6	463.9	453.7	$\lambda = 0.08$

La Tabla 4.9 demuestra que el valor de $\lambda = 0.08$ garantiza una convergencia rápida hacia la velocidad deseada; por tanto la ganancia de control óptima obtenida mediante MGPI es

$$K_{mgpi} = [k_\omega, k_{i_A}, -k_I] = [0.7144, 8.5279, -14.9776]. \quad (4.65)$$

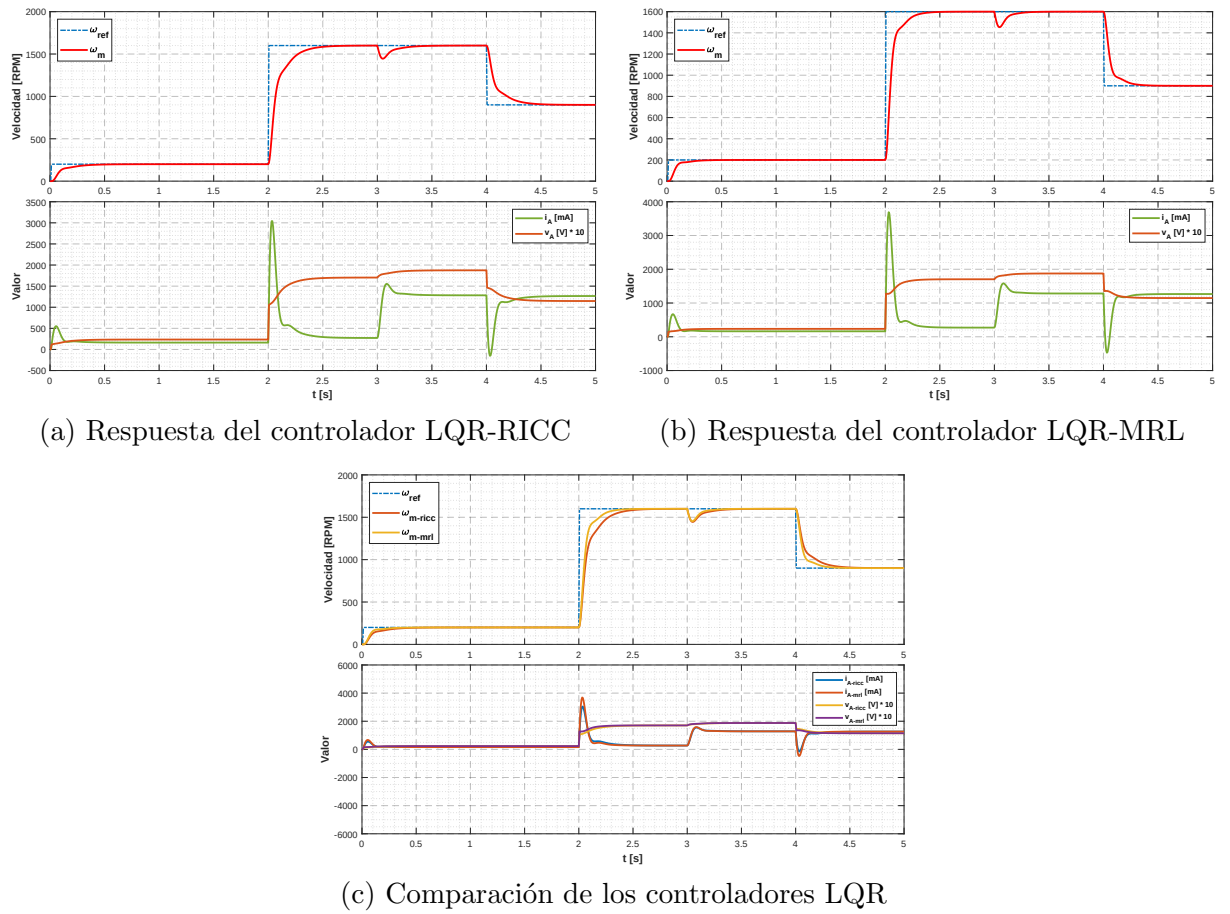
4.2.4. Verificación de los controladores LQR en simulación

Figura 4.24: Esquema en Simulink para la simulación de los controladores LQR



La Figura 4.24 muestra el esquema utilizado para la simulación de los controladores LQR. Es imperativo mencionar que las ganancias están escaladas a unidades del Sistema Internacional. Por ello, se añade una ganancia de conversión del error de velocidad de RPM a rad/s, de modo que el controlador no experimente problemas al actuar sobre el sistema. Dado que se trata de un entorno de simulación, el actuador no se considera dentro del esquema; en consecuencia la señal de control se alimenta directamente al motor DC a través de un bloque de saturación.

Figura 4.25: Respuestas en simulación de los controladores LQR de velocidad



Nota. La prueba de simulación consiste en aplicar tres cambios de consigna en los instantes 0.01 s, 2 s y 4 s, junto con una perturbación de par de $1 \text{ N} \cdot \text{m}$ en el instante 3 s. Dado que se evalúan tres cambios de referencia, se obtienen tres respuestas transitorias. Para el análisis del tiempo de establecimiento t_s se adopta el criterio del 1 % para referencias mayores a 1000 RPM y el criterio del 2 % en caso contrario. Los sufijos *RICC* y *MRL*, introducidos en la figura, corresponden a los métodos de sintonización basados en el algoritmo de Hewer para la solución la DARE y en el algoritmo MGPI respectivamente.

Tabla 4.10: Indicadores de desempeño transitorio del motor DC frente a cambios de referencia bajo controladores LQR en simulación

Métrica	LQR-RICC	LQR-MRL	Mejor
$t_{s1}[s]$	0.4813	0.3438	LQR-MRL
$M_{p1}(\%)$	0.0000	0.0000	LQR-MRL
$t_{s2}[s]$	0.4875	0.3438	LQR-MRL
$M_{p2}(\%)$	0.0000	0.0000	LQR-MRL
$t_{s3}[s]$	0.4750	0.3312	LQR-MRL
$M_{p3}(\%)$	0.0000	0.0000	LQR-MRL

Tabla 4.11: Indicadores de desempeño del motor DC en seguimiento de trayectoria y consumo energético bajo controladores LQR en simulación

Métrica	LQR-RICC	LQR-MRL	Mejor
σ_e	143.1944	127.3382	LQR-MRL
ISE	1.8261×10^7	1.3898×10^7	LQR-MRL
IAE	3.8354×10^4	2.6997×10^4	LQR-MRL
ITAE	1.5719×10^7	1.0901×10^7	LQR-MRL
ITSE	7.0401×10^9	5.3216×10^9	LQR-MRL
$v_{A-RMS} [V]$	123.2051	123.9287	LQR-RICC
$i_{A-RMS} [A]$	0.8645	0.8868	LQR-RICC
$P [W]$	106.5088	109.8974	LQR-RICC

Los índices de desempeño presentados en las Tablas 4.10 y 4.11 muestran que los controlares LQR no presenta sobreimpulsos. Asimismo, el controlador LQR basado en el algoritmo MGPI (LQR-MRL) mejora las respuestas tanto transitorias como de seguimiento en comparación con el LQR-RICC. No obstante, el controlador LQR-RICC optimiza el consumo energético, por lo que resulta superior bajo este criterio.

El fenómeno del balance multiobjetivo expuesto en la sección 4.1.5.5 también se manifiesta en el caso de controladores LQR que buscan un balance entre rendimiento y consumo energético (Brunton, 2017). En términos generales, puede afirmarse que el controlador LQR-MRL supera al LQR-RICC en desempeño transitorio y en seguimiento de trayectoria, presentando únicamente una diferencia del 1.0318 % en el consumo de potencia respecto al controlador más eficiente. No obstante, estos resultados corresponden a un entorno ideal de simulación y, por tanto, deben contrastarse con los resultados experimentales.

CAPÍTULO 5:

IMPLEMENTACIÓN, PRUEBAS Y RESULTADOS

Los equipos del sistema de servoaccionamiento Lucas Nüelle presentados en la sección 3.2, se emplean para la implementación de los controladores diseñados en el capítulo anterior.

Las Figuras 3.21 y 3.22, que muestran la integración de los equipos del sistema, también se usan en las pruebas de implementación de los controladores. Por este motivo, estas configuraciones se presentan nuevamente en las Figuras 5.1 y 5.2.

Figura 5.1: Diagrama de bloques de los equipos utilizados durante las pruebas de implementación

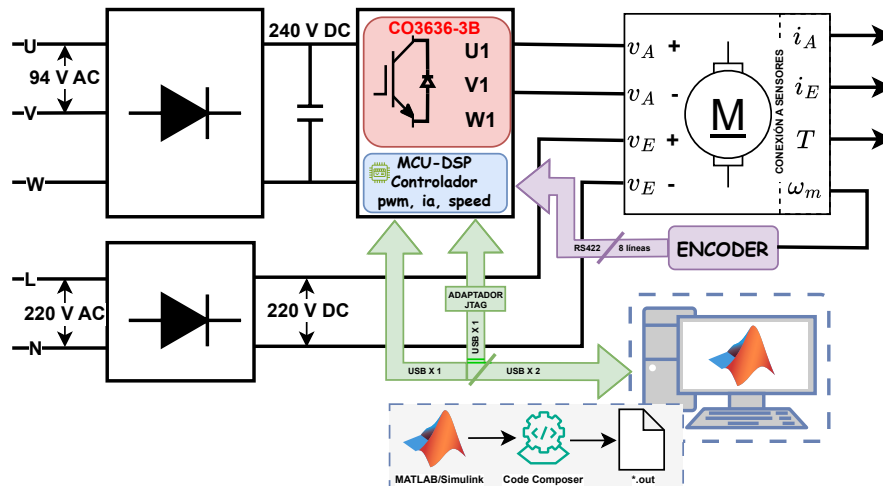
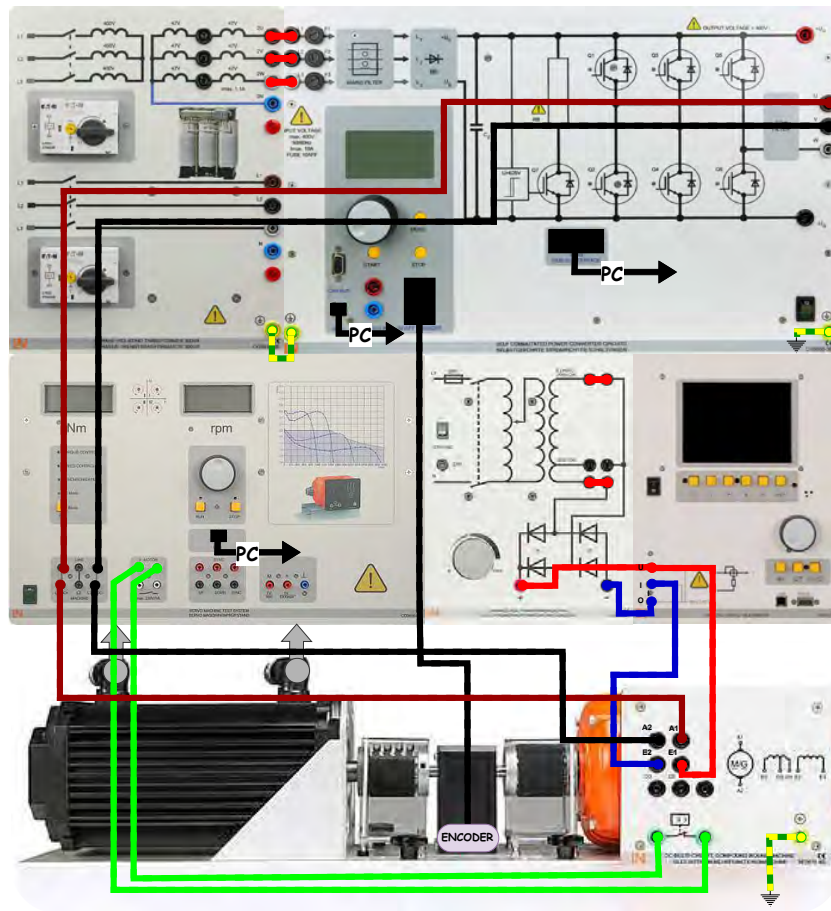
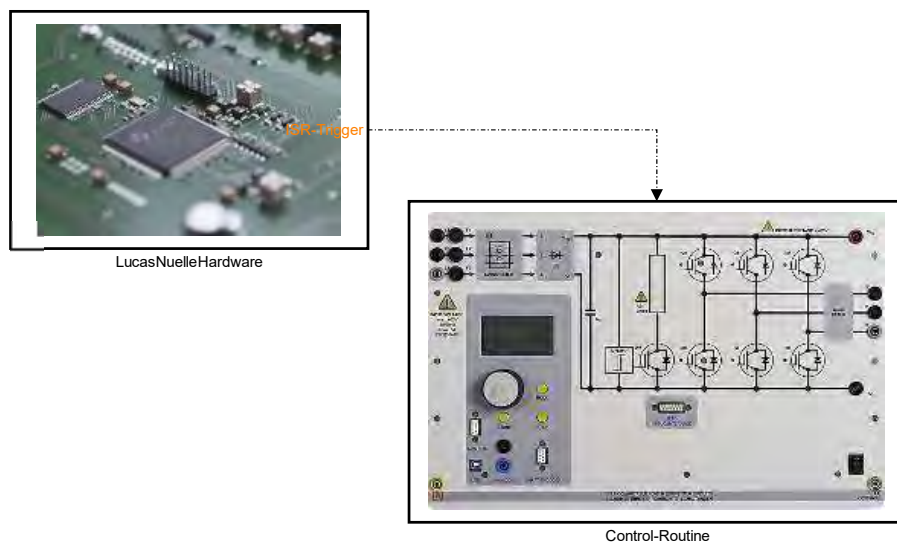


Figura 5.2: Configuración experimental de los equipos para las pruebas de implementación



Nota. Tomada y adaptada de Linnertz (2016) ©2016 Lucas Nülle GmbH.

Figura 5.3: Integración del MCU C2000 con la rutina de control en Simulink



Como se explica en la sección 3.5, el Toolbox de MATLAB CO3636-3D, junto con el Code Composer Studio 5, permite compilar y cargar el firmware generado por los bloques de Simulink directamente en el MCU C2000 del módulo CO3636-3B. Los componentes de software que permiten este flujo de trabajo se muestran en la Figura 5.3. Por su parte, el diagrama de bloques que describe las conexiones entre estas herramientas y los componentes de hardware se muestra en la Figura 5.1.

De la Figura 5.3, el bloque *LucasNuelleHardware* contiene la configuración necesaria para compilar y cargar el programa generado por el bloque *Control-Routine*, el cual se ejecuta con un periodo de muestreo de $T_s = 0.000125$ s. Por tanto, la frecuencia de muestreo de la rutina de control es $f_s = 1/T_s = 8000$ Hz.

La Figura 5.4 muestra el programa ActiveServo, el cual se usa en conjunto con el banco de pruebas CO3636-6V para generar perfiles de carga que permiten evaluar el desempeño de los controladores diseñados para el motor DC. Por su parte, la Figura 5.5 muestra la herramienta *LNLiveScope*, utilizada para capturar las variables de interés y exportarlas posteriormente para su análisis.

Figura 5.4: Programa ActiveServo para pruebas de carga de las máquinas eléctricas

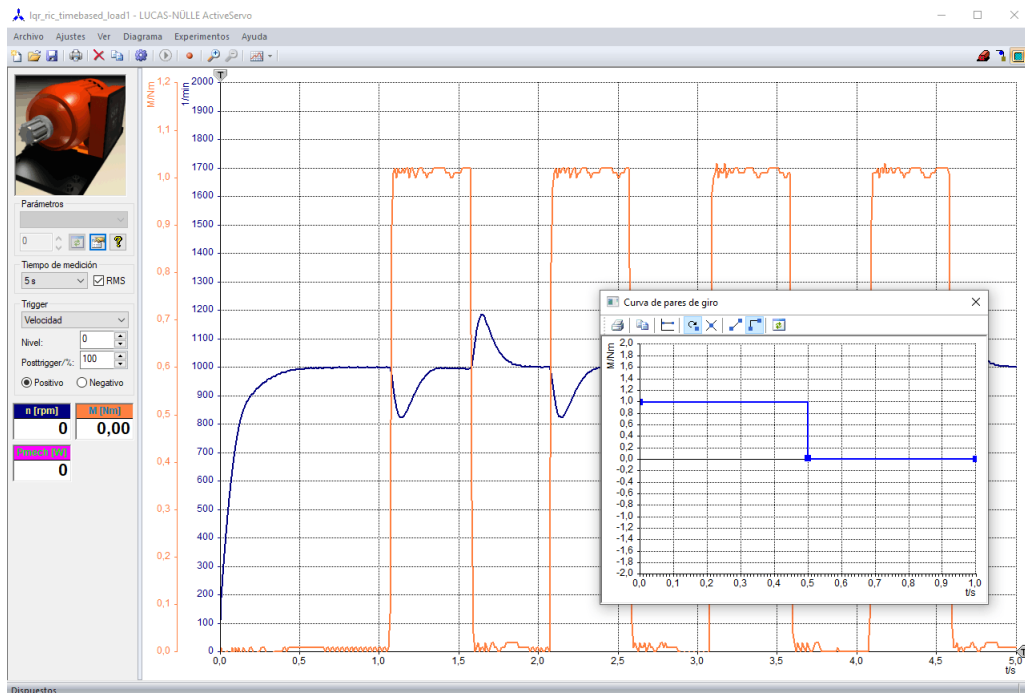
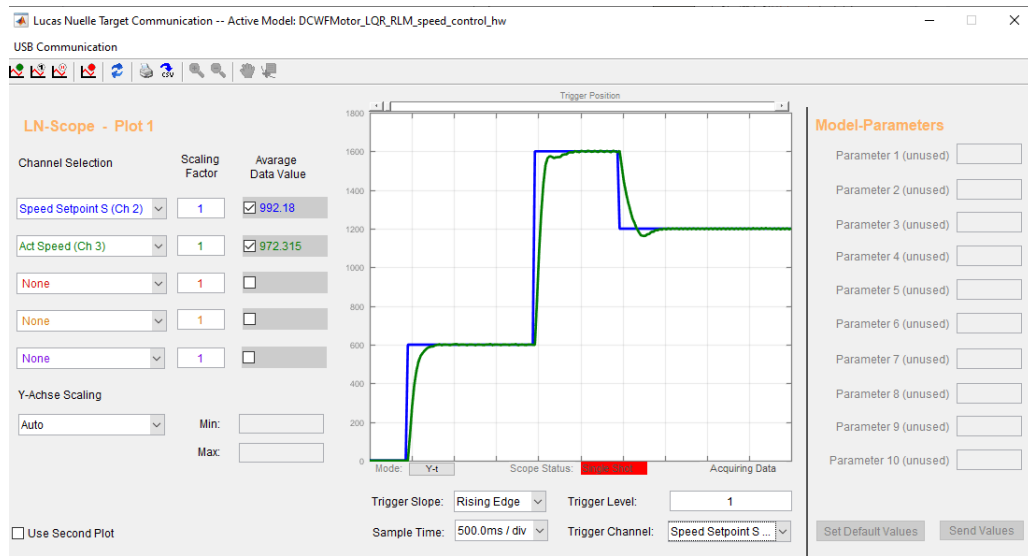


Figura 5.5: Programa LNLiveScope del Toolbox CO3636-3D para la captura de datos



5.1. Pruebas de seguimiento de trayectoria y perturbaciones de carga

Las pruebas que se efectúan en este trabajo se ciñen a los alcances definidos para la presente investigación. Estas pruebas se agrupan en tres categorías: cambios de valor de consigna, rampas de velocidad y perturbaciones de carga. Cada una de estas categorías se presenta en las Tablas 5.1, 5.2 y 5.3, respectivamente.

Tabla 5.1: Pruebas de cambios de referencia

Prueba	Setpoint 1 [$t_1 = 0.5$ s]	Setpoint 2 [$t_2 = 2.0$ s]	Setpoint 3 [$t_3 = 3.0$ s]
1	100 RPM	1100 RPM	300 RPM
2	600 RPM	1600 RPM	1200 RPM

Tabla 5.2: Pruebas de rampas de velocidad

Prueba	Subida	Setpoint	Bajada
Rampa 1	1600 RPM/s [1-1.5]s	800 RPM [1.5-3]s	1600 RPM/s [3-3.5]s
Rampa 2	3000 RPM/s [1-1.5]s	1500 RPM [1.5-3]s	3000 RPM/s [3-3.5]s

Tabla 5.3: Perfiles de perturbaciones de carga

Prueba	Perfil de torque M_L [N·m]
1	$1 : 0$ [N·m], $T = 1$ s
2	$0.5 \cdot \sin(2\pi t/T)$ [N·m], $T = 1$ s

La Tabla 5.2 indica las rampas de subida, las rampas de bajada y el valor máximo de la consigna—junto con su duración—a los cuales los controladores serán expuestos. En la Tabla 5.3 se presentan los perfiles de carga utilizados para evaluar el rechazo a perturbaciones de los controladores.

Es importante señalar que la primera prueba de torque es una señal cuadrada con periodo T que oscila entre los valores indicados ($1 : 0$). Estas condiciones simulan exigencias repentinas de torque a la que pueden estar sujetos los motores DC en distintos contextos y aplicaciones.

Observación: Las métricas de evaluación utilizadas se presentan en la sección 2.2.8. Para el análisis del tiempo de establecimiento t_s , se adopta el criterio del 1 % para referencias mayores a 1000 RPM y el criterio del 2 % en caso contrario.

Dado que métricas de desempeño transitorio, como el sobreimpulso máximo M_p y el tiempo de establecimiento t_s , se utilizan para evaluar cambios de referencia escalonados (Ogata, 2010), en las pruebas con rampas de velocidad o perfiles de carga no lineales el análisis se restringe a los índices de desempeño de seguimiento de trayectoria (σ_e , ISE, IAE, ITAE, ITSE) y a los índices asociados al consumo energético.

Los experimentos aquí diseñados se basan en los trabajos de Liceaga-Castro et al. (2022), Bermúdez et al. (2021), Bankovic et al. (2021), Linnertz (2016) y Stumper (2012).

5.2. Método de conteo de Borda para la evaluación del mejor controlador

Cuando varios índices de desempeño son utilizados para evaluar controladores, la selección del mejor controlador puede formularse como un problema de toma de decisiones multicriterio. En estos casos, el método de conteo de Borda permite combinar las clasificaciones obtenidas a partir de múltiples criterios para producir un ranking final de las alternativas (Matejová & Paralič, 2025).

El método consiste en asignar una puntuación a cada controlador de acuerdo con su posición relativa dentro de cada criterio. Posteriormente se suman para obtener una puntuación total que determina el ranking final (Matejová & Paralič, 2025).

Sea $C = \{C_1, C_2, \dots, C_n\}$ el conjunto de controladores y $J = \{J_1, J_2, \dots, J_m\}$ el conjunto de índices de desempeño. Para cada criterio J_j , los controladores se ordenan según su desempeño, obteniendo una posición $r_{i,j}$ para el controlador C_i . A cada posición se le asigna una puntuación $P_{i,j}$, y el puntaje total de Borda del controlador C_i se calcula como

$$B_i = \sum_{j=1}^m P_{i,j}, \quad (5.1)$$

donde m es el número total de criterios evaluados. El controlador con el mayor valor de B_i es considerado el mejor controlador global (Gharbi et al., 2025).

En la práctica, cuando el número de alternativas es reducido, la asignación de puntos puede simplificarse utilizando una escala ordinal basada en la posición relativa. Por ejemplo, si se comparan tres controladores, se asignan puntuaciones decrecientes a cada posición del ranking de la siguiente forma: **mejor controlador** 2 puntos, **controlador intermedio** 1 punto y **peor controlador** 0 puntos.

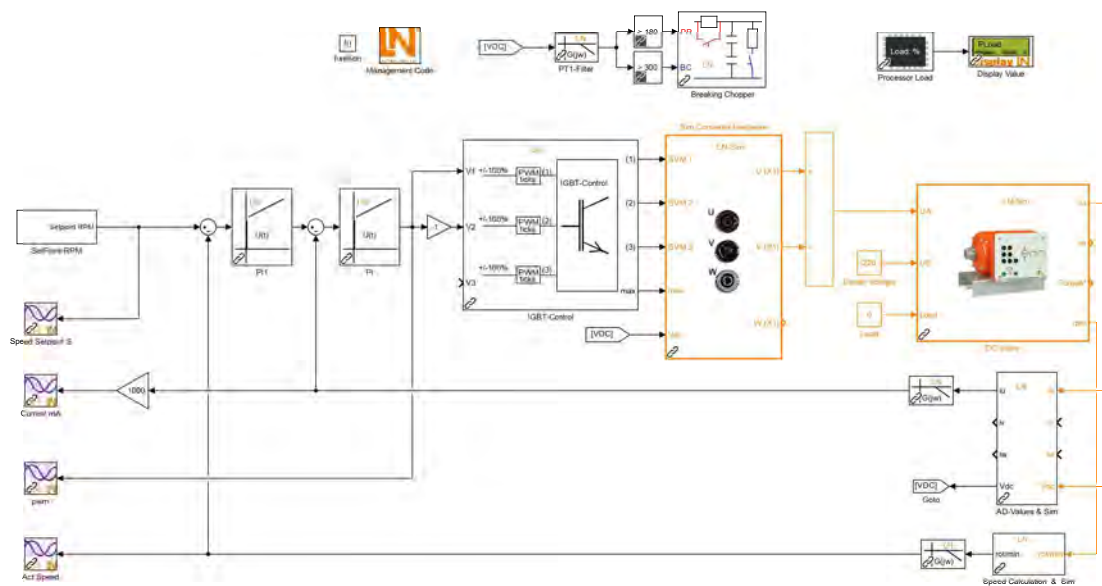
De esta manera, cada índice de desempeño genera una clasificación de los controladores y los puntos asignados se acumulan para obtener una evaluación global. Este enfoque

permite identificar el controlador que presenta el mejor desempeño global considerando simultáneamente todos los índices evaluados, favoreciendo aquellos que muestran un comportamiento consistente entre múltiples criterios (Matejová & Paralič, 2025).

5.3. Implementación, pruebas y resultados de los controladores PI

A partir de la arquitectura del controlador PI presentada en la Figura 4.1, se implementa la rutina de control en Simulink r2016a, tal como se muestra en la Figura 5.6. Esta versión r2016a es compatible con el Toolbox de MATLAB CO3636-3D Lucas Nülle y se encuentra instalada en los equipos del laboratorio de Electrónica de Potencia.

Figura 5.6: Programa en Simulink del controlador PI



Las ganancias del controlador PI de velocidad, diseñadas en el capítulo anterior, se actualizan dentro del bloque de control del lazo externo. Este bloque corresponde a un subsistema de Simulink que implementa el controlador PI junto con las rutinas de discretización y protección necesarias.

La salida del lazo externo de control, correspondiente a la señal de corriente de referencia i_{A-ref} , se limita al intervalo $[-3, 3]$ A, que son los límites máximos de corriente de arranque y de frenado. Por su parte, la salida del lazo interno, que es el ciclo de trabajo D_{PWM} aplicado a los convertidores de potencia, se limita al intervalo $[-100, 100]$ % (Linnertz, 2016).

Observación: Es necesario recalcar que las ganancias diseñadas mediante los métodos del óptimo simétrico y Ziegler-Nichols están escaladas para trabajar con errores de velocidad expresados en RPM. Sin embargo, las ganancias basadas en TD3 con término no lineal están escaladas en unidades de velocidad angular de rad/s. Por lo tanto, es necesaria una conversión de escala al momento de implementar los controladores.

5.3.1. Sobre el controlador PI basado en Ziegler-Nichols

Como afirma Ogata (2010), las ganancias del controlador PI obtenidas mediante este método constituyen la primera aproximación; no obstante, no se garantiza su adecuado funcionamiento en la planta real. Al desplegar las ganancias del controlador PI definidas en (4.19) y (4.20), se obtiene la respuesta mostrada en la Figura 5.7, la cual evidencia un comportamiento inestable. En consecuencia, el procedimiento establece la necesidad de ajustar las ganancias.

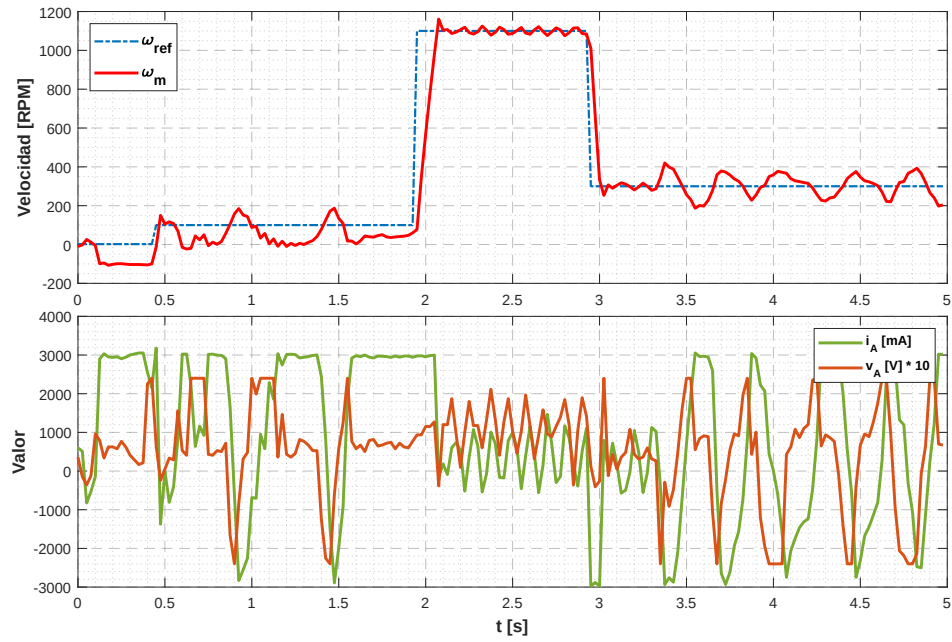
Tras efectuar ajustes empíricos en el sistema real, se obtiene un comportamiento estable cuando los parámetros del controlador PI se actualizan a:

$$T_{i-znN} = T_{i-zn} = 16.67 \text{ ms}, \quad (5.2)$$

$$K_{p-znN} = K_{p-zn}/2 = 0.0215. \quad (5.3)$$

Estas ganancias se emplean posteriormente en el análisis comparativo presentado en las secciones siguientes.

Figura 5.7: Primera respuesta del controlador PI basado en Ziegler-Nichols



5.3.2. Pruebas y resultados de cambio de referencia

5.3.2.1. Cambios de referencia: Prueba 1

Figura 5.8: Respuesta del controlador PI-OS (óptimo simétrico)

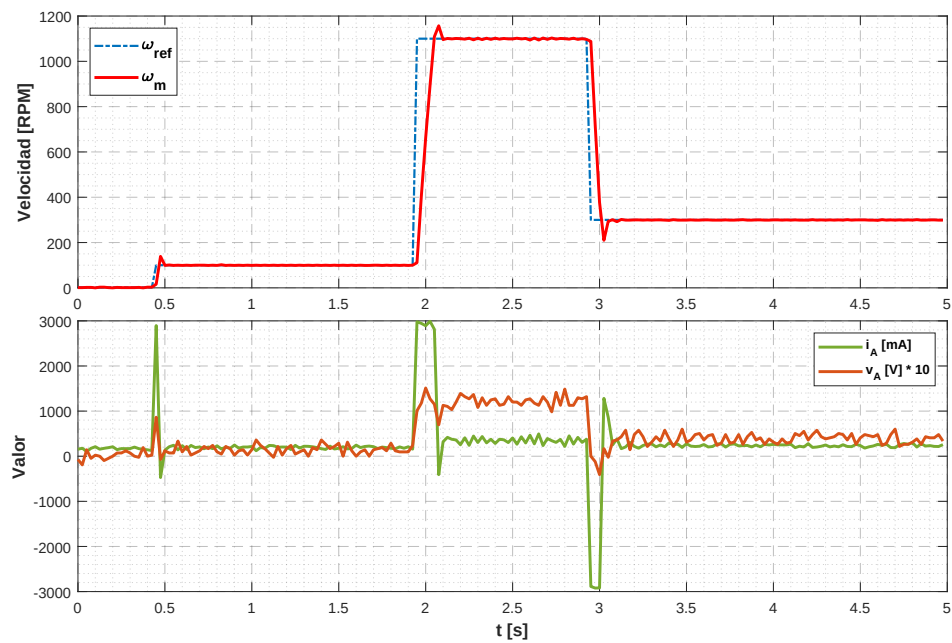


Figura 5.9: Respuesta del controlador PI-ZN (Ziegler Nichols)

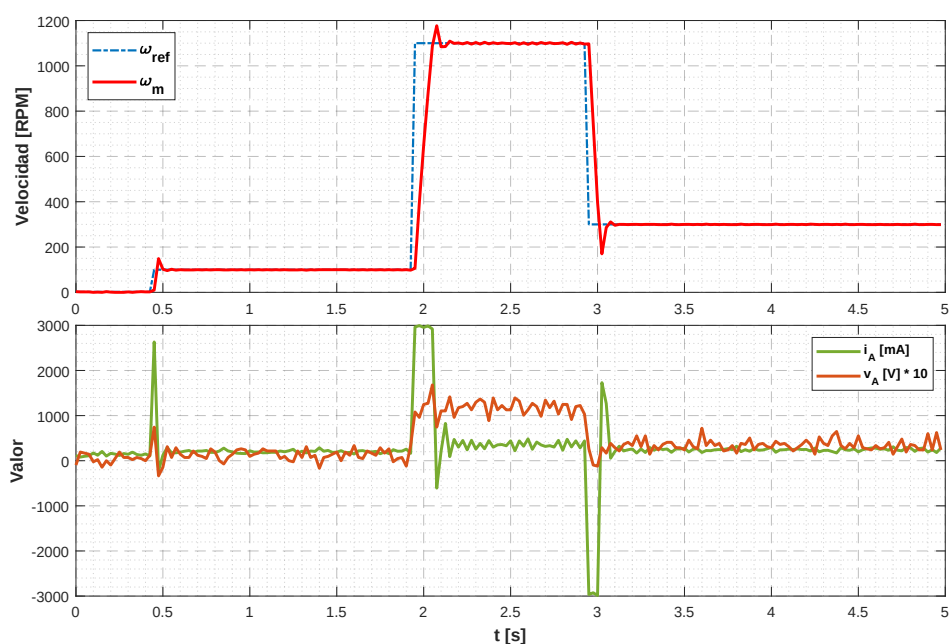
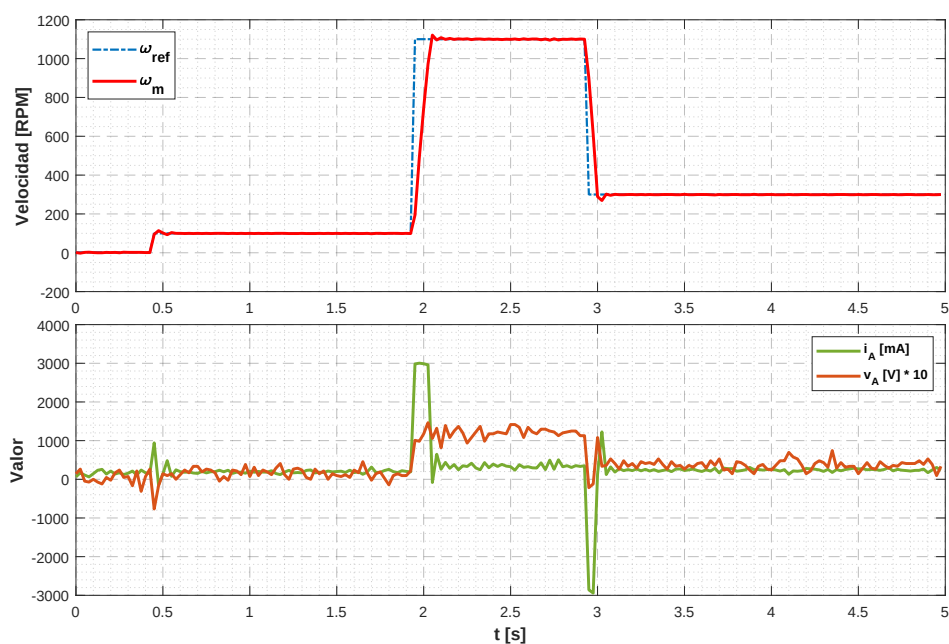


Figura 5.10: Respuesta del controlador PI-rlNL (TD3 con término no lineal)



La prueba 1 consiste en tres cambios de referencia entre velocidades bajas 100 RPM, 300 RPM y medias 1100 RPM (Tabla 5.1). Dado que se evalúan tres cambios de referencia, se obtienen tres respuestas transitorias. La evaluación correspondiente a las pruebas mostradas en las figuras anteriores se presenta en las Tablas 5.4, 5.5 y en la Figura 5.12.

Figura 5.11: Comparación de los controladores PI - Cambios de referencia: prueba 1

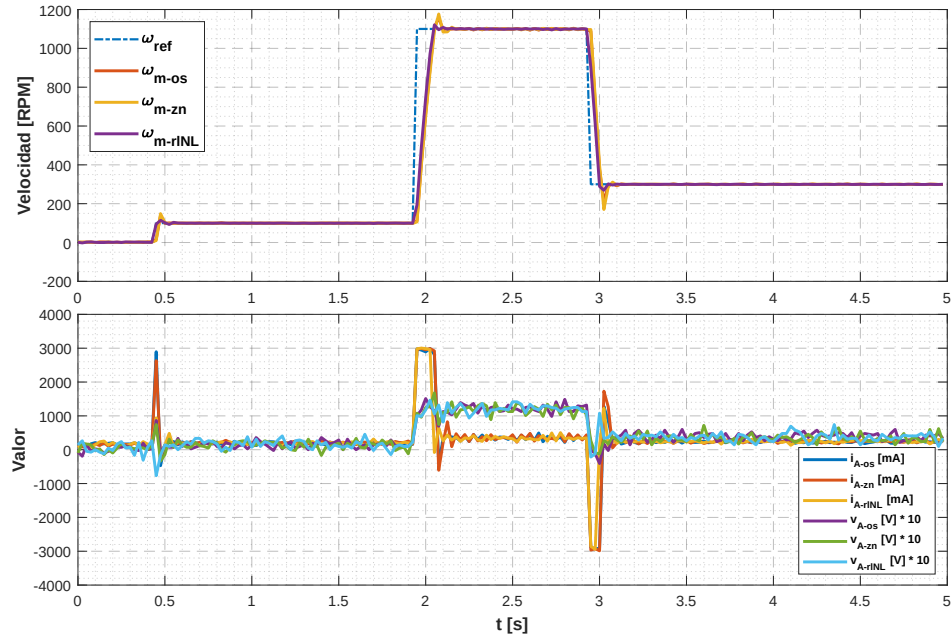


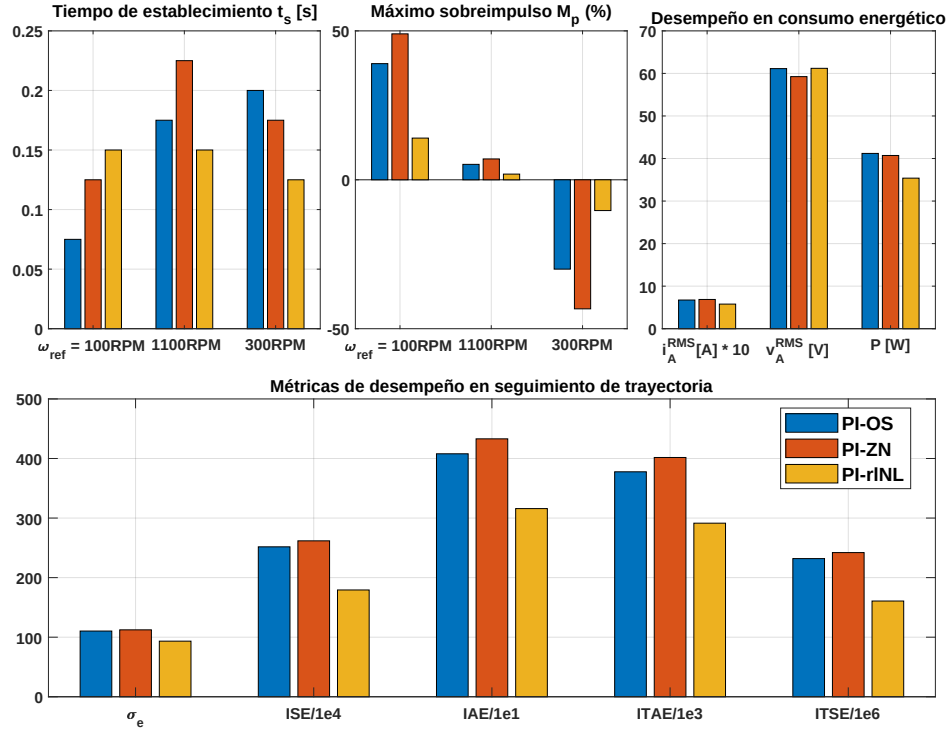
Tabla 5.4: Indicadores de desempeño transitorio de los controladores PI frente a cambios de referencia en implementación - Prueba 1

Métrica	PI-OS	PI-ZN	PI-rINL	Mejor(2)	Medio(1)	Peor(0)
$t_{s1}[s]$	0.0750	0.1250	0.1500	PI-OS	PI-ZN	PI-rINL
$M_{p1}(\%)$	39.000	49.000	14.000	PI-rINL	PI-OS	PI-ZN
$t_{s2}[s]$	0.1750	0.2250	0.1500	PI-rINL	PI-OS	PI-ZN
$M_{p2}(\%)$	5.182	7.000	1.909	PI-rINL	PI-OS	PI-ZN
$t_{s3}[s]$	0.2000	0.1750	0.1250	PI-rINL	PI-ZN	PI-OS
$M_{p3}(\%)$	-30.000	-43.333	-10.333	PI-rINL	PI-OS	PI-ZN

Tabla 5.5: Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores PI frente a cambios de referencia en implementación - Prueba 1

Métrica	PI-OS	PI-ZN	PI-rINL	Mejor(2)	Medio(1)	Peor(0)
σ_e	110.3139	112.3359	93.3531	PI-rINL	PI-OS	PI-ZN
ISE	2 517 023	2 617 616	1 792 855	PI-rINL	PI-OS	PI-ZN
IAE	4079	4330	3159	PI-rINL	PI-OS	PI-ZN
ITAE	377 662	401 702	291 421	PI-rINL	PI-OS	PI-ZN
ITSE	231 952 686	242 038 108	160 730 279	PI-rINL	PI-OS	PI-ZN
$v_A^{\text{RMS}} [\text{V}]$	61.1315	59.2470	61.1981	PI-ZN	PI-OS	PI-rINL
$i_A^{\text{RMS}} [\text{A}]$	0.6739	0.6872	0.5782	PI-rINL	PI-OS	PI-ZN
$P [\text{W}]$	41.1980	40.7142	35.3862	PI-rINL	PI-ZN	PI-OS

Figura 5.12: Gráfico de barras de las métricas de desempeño de los controladores PI frente a cambios de referencia - Prueba 1



Las Figuras 5.8, 5.9, 5.10 y 5.11 muestran que el controlador PI-rINL presenta una respuesta más rápida y con menor sobreimpulso frente a los cambios de consigna de velocidad en comparación con los demás controladores.

De los resultados numéricos en las Tablas 5.4, 5.5 y en la Figura 5.12, se observa que se consideran 14 casos de evaluación. Usando el criterio de conteo de Borda (ecuación (5.1)), se calcula el puntaje de Borda de cada controlador:

$$B_{PI-OS}^{ST1} = 1 \times (2) + 11 \times (1) + 2 \times (0) = 13, \quad (5.4)$$

$$B_{PI-ZN}^{ST1} = 1 \times (2) + 3 \times (1) + 10 \times (0) = 5, \quad (5.5)$$

$$B_{PI-rINL}^{ST1} = 12 \times (2) + 0 \times (1) + 2 \times (0) = 24. \quad (5.6)$$

Estos resultados sugieren que el controlador PI-rINL obtiene el mayor puntaje de Borda en la prueba 1, con 24 puntos, seguido del PI-OS con un puntaje de 13.

5.3.2.2. Cambios de referencia: Prueba 2

Figura 5.13: Respuesta del controlador PI-OS (óptimo simétrico)

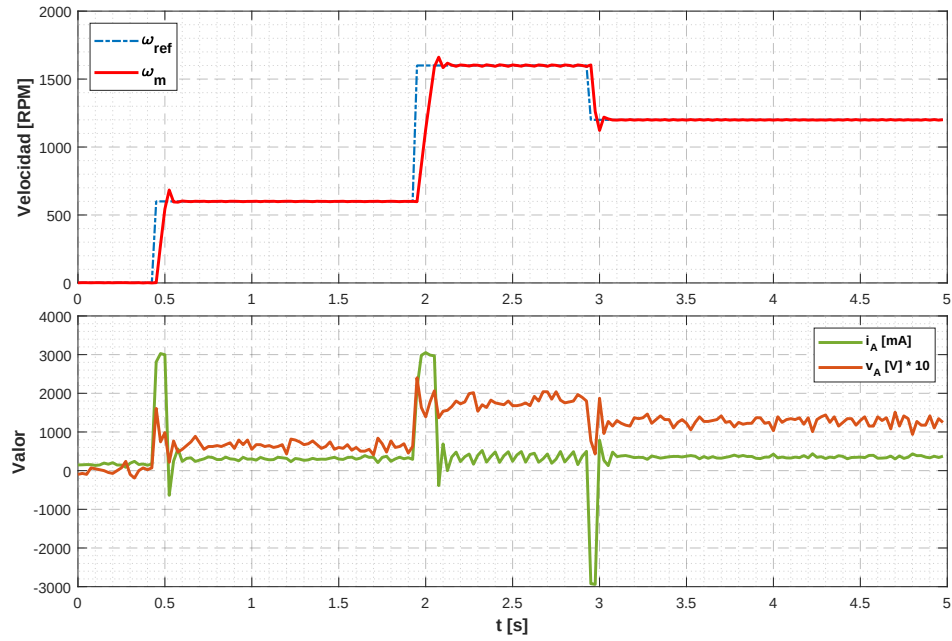


Figura 5.14: Respuesta del controlador PI-ZN (Ziegler Nichols)

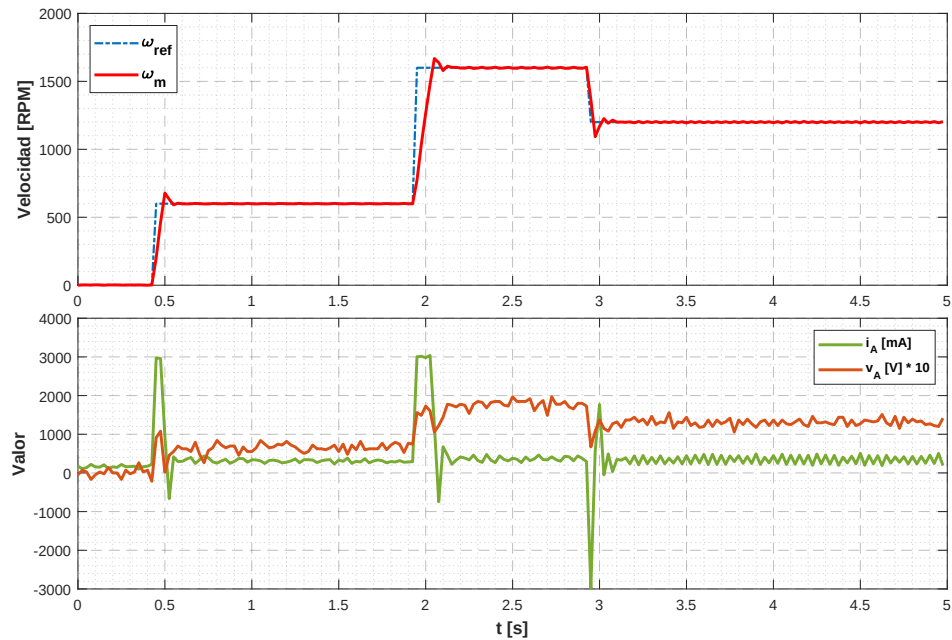


Figura 5.15: Respuesta del controlador PI-rlNL (TD3 con término no lineal)

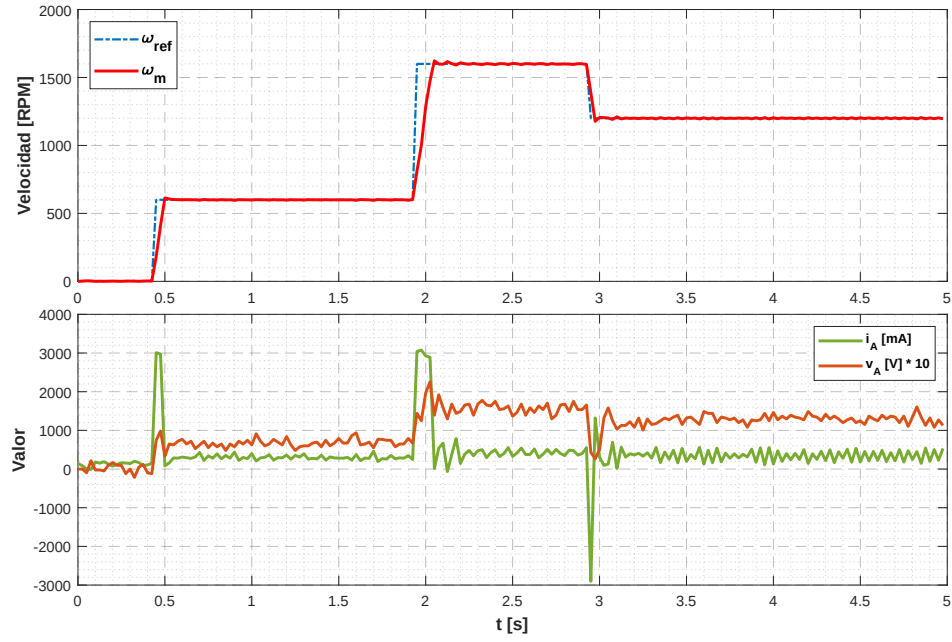
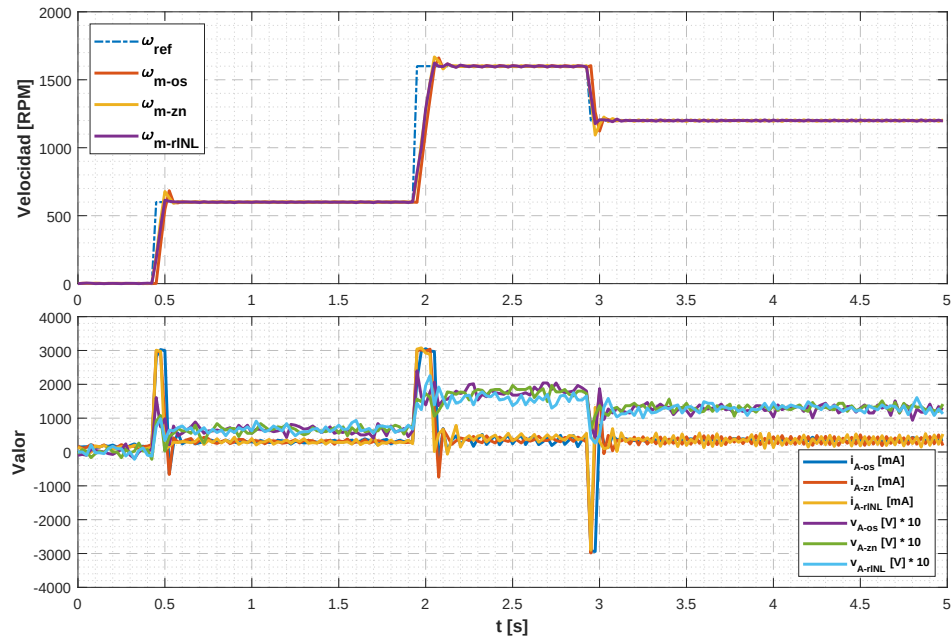


Figura 5.16: Comparación de los controladores PI - Cambios de referencia: prueba 2



La prueba 2 consiste en tres cambios de referencia entre velocidades medias 600 RPM y altas 1600 RPM, 1200 RPM (Tabla 5.1). Dado que se evalúan tres cambios de referencia, se obtienen tres respuestas transitorias. La evaluación correspondiente a las pruebas mostradas en las figuras anteriores se presenta en las Tablas 5.6, 5.7 y en la Figura 5.17.

Tabla 5.6: Indicadores de desempeño transitorio de los controladores PI frente a cambios de referencia en implementación - Prueba 2

Métrica	PI-OS	PI-ZN	PI-rINL	Mejor(2)	Medio(1)	Peor(0)
$t_{s1}[s]$	0.1250	0.1250	0.1000	PI-rINL	PI-ZN	PI-OS
$M_{p1}(\%)$	14.000	13.000	2.333	PI-rINL	PI-ZN	PI-OS
$t_{s2}[s]$	0.2250	0.2000	0.2250	PI-ZN	PI-OS	PI-rINL
$M_{p2}(\%)$	3.813	4.250	1.438	PI-rINL	PI-OS	PI-ZN
$t_{s3}[s]$	0.1250	0.1750	0.0750	PI-rINL	PI-OS	PI-ZN
$M_{p3}(\%)$	-6.500	-8.917	-1.917	PI-rINL	PI-OS	PI-ZN

Nota. En caso de empate, se prioriza el controlador que haya obtenido previamente un menor puntaje de Borda, con el propósito de refinar el criterio de comparación entre controladores.

Tabla 5.7: Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores PI frente a cambios de referencia en implementación - Prueba 2

Métrica	PI-OS	PI-ZN	PI-rINL	Mejor(2)	Medio(1)	Peor(0)
σ_e	108.8498	80.6752	80.4269	PI-rINL	PI-ZN	PI-OS
ISE	2 470 233	1 361 659	1 342 994	PI-rINL	PI-ZN	PI-OS
IAE	4485	3463	3140	PI-rINL	PI-ZN	PI-OS
ITAE	328 431	274 746	241 375	PI-rINL	PI-ZN	PI-OS
ITSE	175 267 125	98 890 112	95 634 795	PI-rINL	PI-ZN	PI-OS
$v_A^{RMS} [V]$	120.2516	119.4288	114.5248	PI-rINL	PI-ZN	PI-OS
$i_A^{RMS} [A]$	0.7273	0.6731	0.6612	PI-rINL	PI-ZN	PI-OS
$P [W]$	87.4548	80.3916	75.7262	PI-rINL	PI-ZN	PI-OS

A partir de los resultados numéricos en las Tablas 5.6 y 5.7, así como en la Figura 5.17, se calcula el puntaje de Borda de cada controlador:

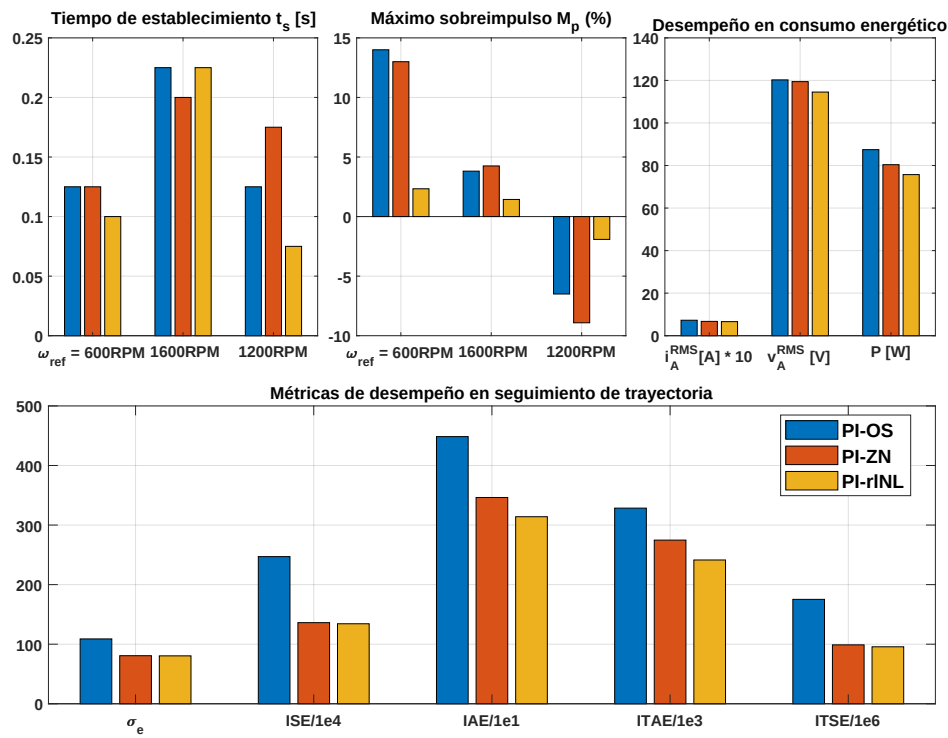
$$B_{PI-OS}^{ST2} = 0 \times (2) + 4 \times (1) + 10 \times (0) = 4, \quad (5.7)$$

$$B_{PI-ZN}^{ST2} = 1 \times (2) + 10 \times (1) + 3 \times (0) = 12, \quad (5.8)$$

$$B_{PI-rINL}^{ST2} = 13 \times (2) + 0 \times (1) + 1 \times (0) = 26. \quad (5.9)$$

Estos resultados indican que el controlador PI-rINL obtiene el mayor puntaje de Borda en la prueba 2, con 26 puntos, seguido del PI-ZN con 12 puntos, mientras que PI-OS obtiene el menor puntaje en esta prueba.

Figura 5.17: Gráfico de barras de las métricas de desempeño de los controladores PI frente a cambios de referencia - Prueba 2



5.3.2.3. Resumen de desempeño - Pruebas de cambio de referencia

Tabla 5.8: Resumen del puntaje de Borda de los controladores PI en pruebas de cambio de referencia

Prueba	B_{PI-OS}	B_{PI-ZN}	$B_{PI-rINL}$	Mejor	Medio	Peor
1	13	5	24	PI-rINL	PI-OS	PI-ZN
2	4	12	26	PI-rINL	PI-ZN	PI-OS
Promedio	8.5	8.5	25.0	PI-rINL	PI-ZN y PI-OS	-

La Tabla 5.8 presenta un resumen del puntaje de Borda de los controladores PI (ecuación (5.4–5.9)) en pruebas de cambio de referencia. En promedio, el controlador PI-rINL muestra el mejor desempeño en comparación con los controladores PI-OS y PI-ZN. Por su parte, estos dos últimos presentan un comportamiento similar: PI-OS supera a PI-ZN en la primera prueba, mientras que el PI-ZN obtiene un mejor resultado en la segunda, de modo que ambos alcanzan el mismo puntaje de Borda promedio.

5.3.3. Pruebas y resultados de rampas de velocidad

5.3.3.1. Rampas de velocidad: Prueba 1

Figura 5.18: Respuesta del controlador PI-OS (óptimo simétrico)

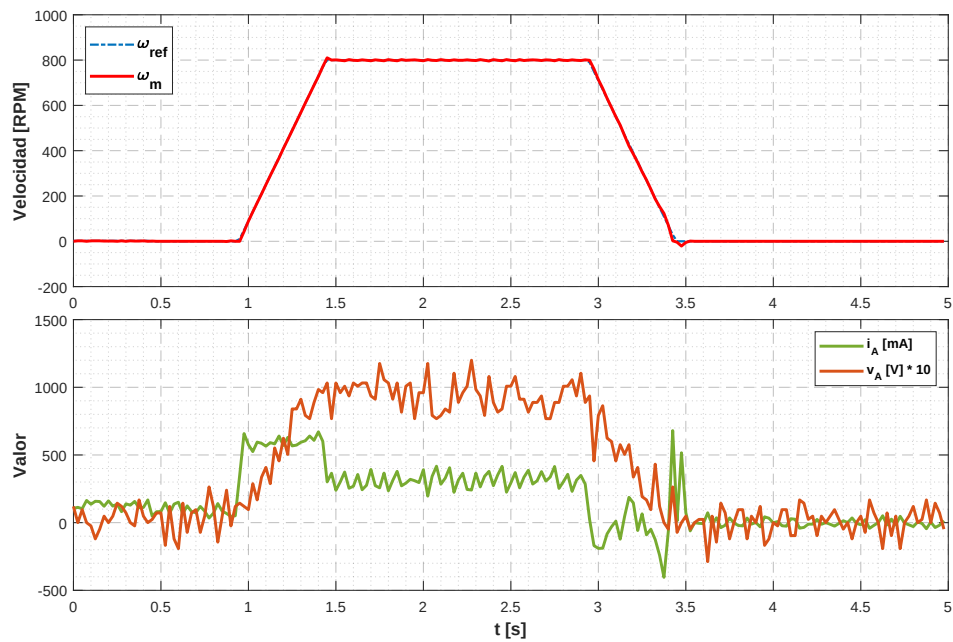


Figura 5.19: Respuesta del controlador PI-ZN (Ziegler Nichols)

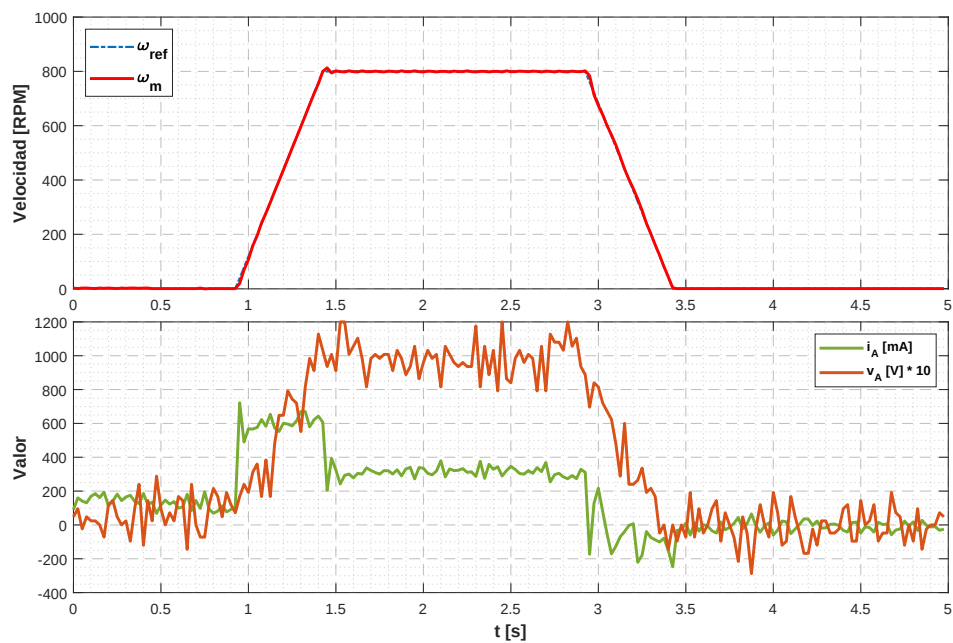


Figura 5.20: Respuesta del controlador PI-rlNL (TD3 con término no lineal)

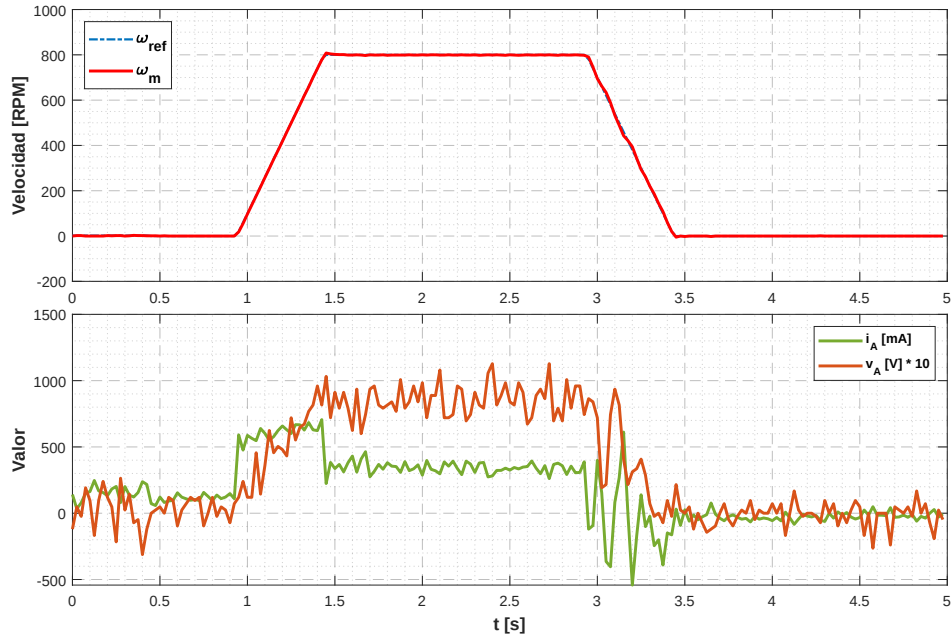
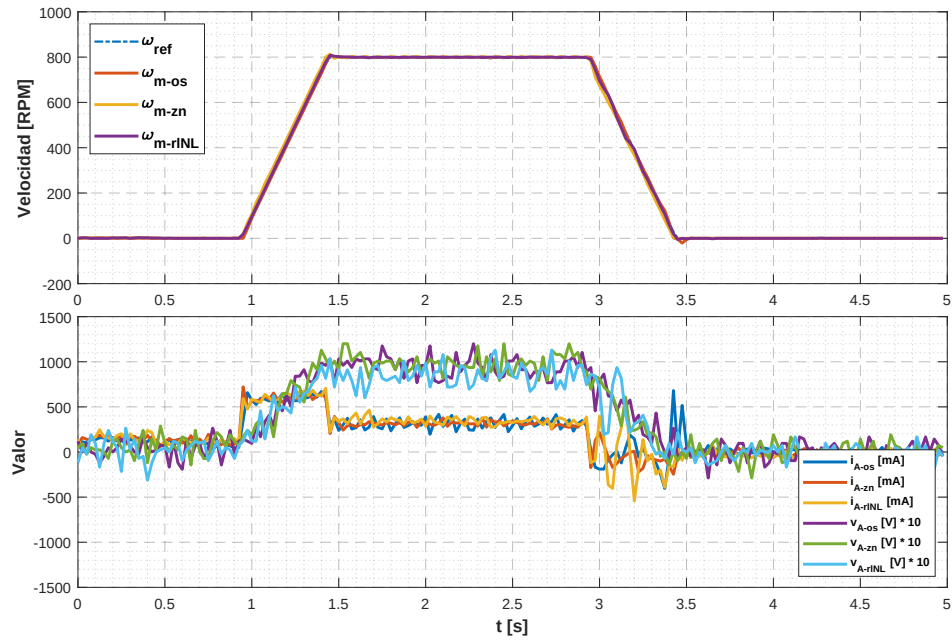


Figura 5.21: Comparación de los controladores PI - Rampas de velocidad: prueba 1

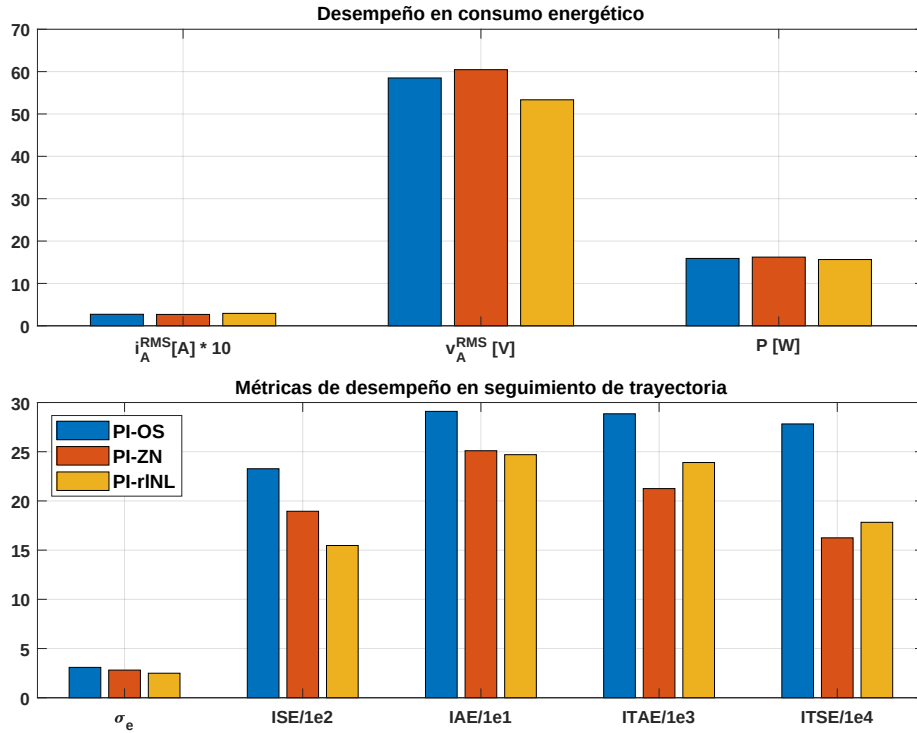


La prueba 1 consiste en rampas de aceleración y frenado de velocidad con una pendiente de 1600 RPM/s y una duración de 0.5 s (Tabla 5.2). Las métricas de evaluación del comportamiento mostrado en las figuras anteriores se presentan en la Tabla 5.9 y en la Figura 5.22.

Tabla 5.9: Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores PI frente a rampas de velocidad en implementación - Prueba 1

Métrica	PI-OS	PI-ZN	PI-rINL	Mejor(2)	Medio(1)	Peor(0)
σ_e	3.0851	2.8107	2.4919	PI-rINL	PI-ZN	PI-OS
ISE	2327	1895	1547	PI-rINL	PI-ZN	PI-OS
IAE	291	251	247	PI-rINL	PI-ZN	PI-OS
ITAE	28 853	21 253	23 903	PI-ZN	PI-rINL	PI-OS
ITSE	278 235	162 495	178 273	PI-ZN	PI-rINL	PI-OS
v_A^{RMS} [V]	58.5012	60.4607	53.3459	PI-rINL	PI-OS	PI-ZN
i_A^{RMS} [A]	0.2718	0.2681	0.2932	PI-ZN	PI-OS	PI-rINL
P [W]	15.8995	16.2116	15.6395	PI-rINL	PI-OS	PI-ZN

Figura 5.22: Gráfico de barras de las métricas de desempeño de los controladores PI frente a rampas de velocidad - Prueba 1



A partir de los resultados numéricos en las Tablas 5.9 así como en la Figura 5.22, se calcula el puntaje de Borda de cada controlador (resultando mejor PI-rINL):

$$B_{PI-OS}^{RT1} = 0 \times (2) + 3 \times (1) + 5 \times (0) = 3, \quad (5.10)$$

$$B_{PI-ZN}^{RT1} = 3 \times (2) + 3 \times (1) + 2 \times (0) = 9, \quad (5.11)$$

$$B_{PI-rINL}^{RT1} = 5 \times (2) + 2 \times (1) + 1 \times (0) = 12. \quad (5.12)$$

5.3.3.2. Rampas de velocidad: Prueba 2

Figura 5.23: Respuesta del controlador PI-OS (óptimo simétrico)

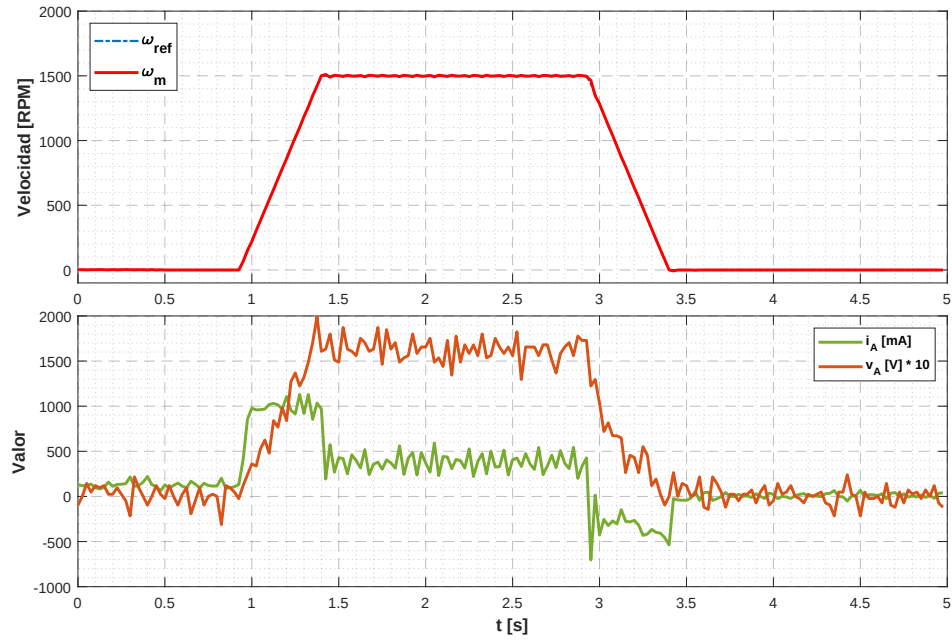


Figura 5.24: Respuesta del controlador PI-ZN (Ziegler Nichols)

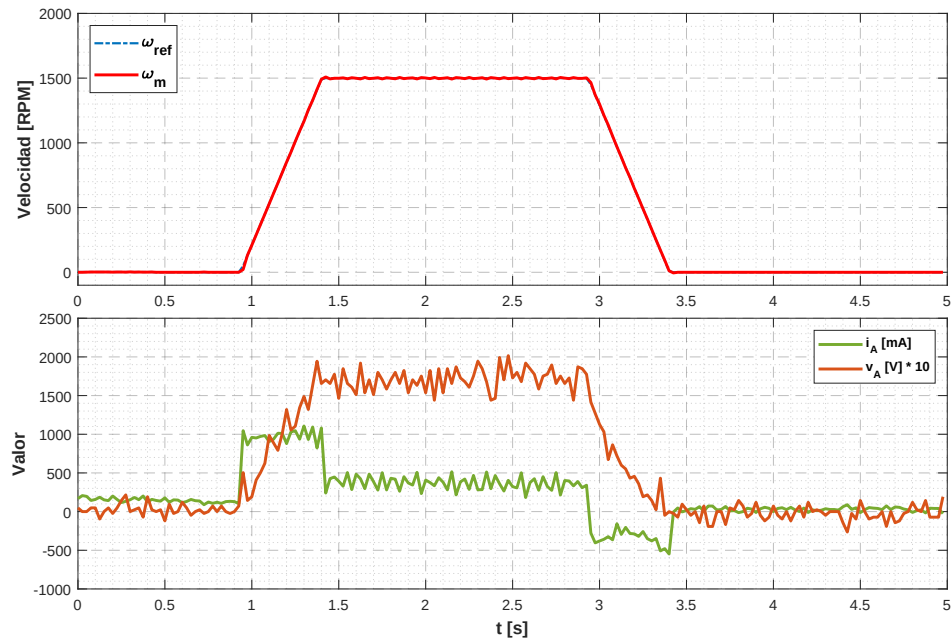


Figura 5.25: Respuesta del controlador PI-rlNL (TD3 con término no lineal)

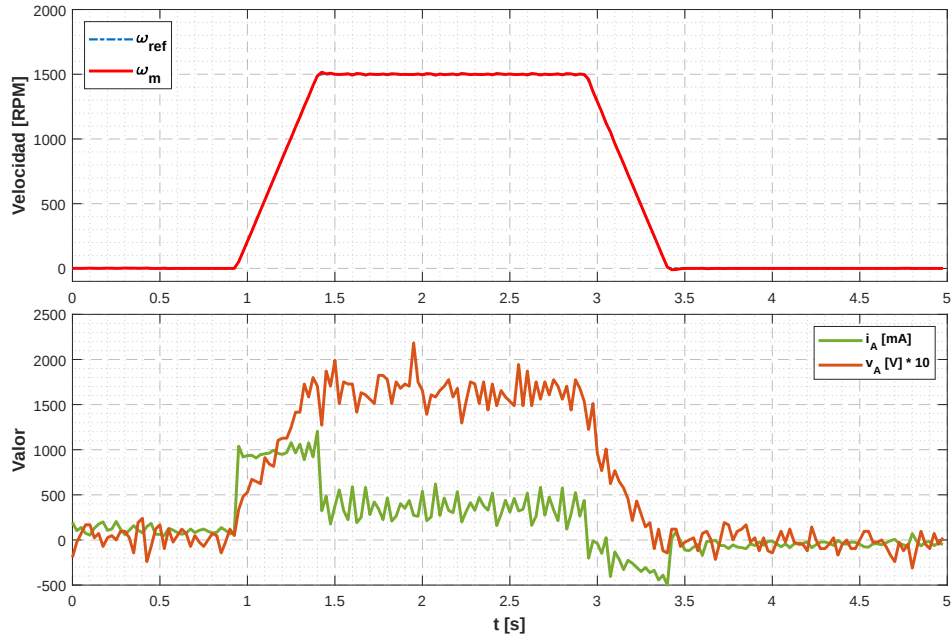
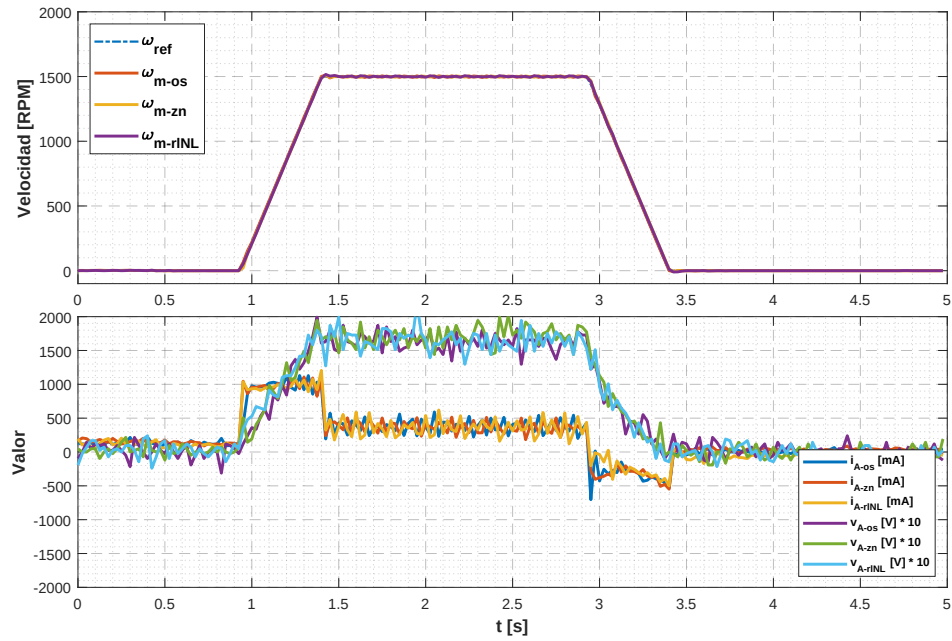


Figura 5.26: Comparación de los controladores PI - Rampas de velocidad: prueba 2

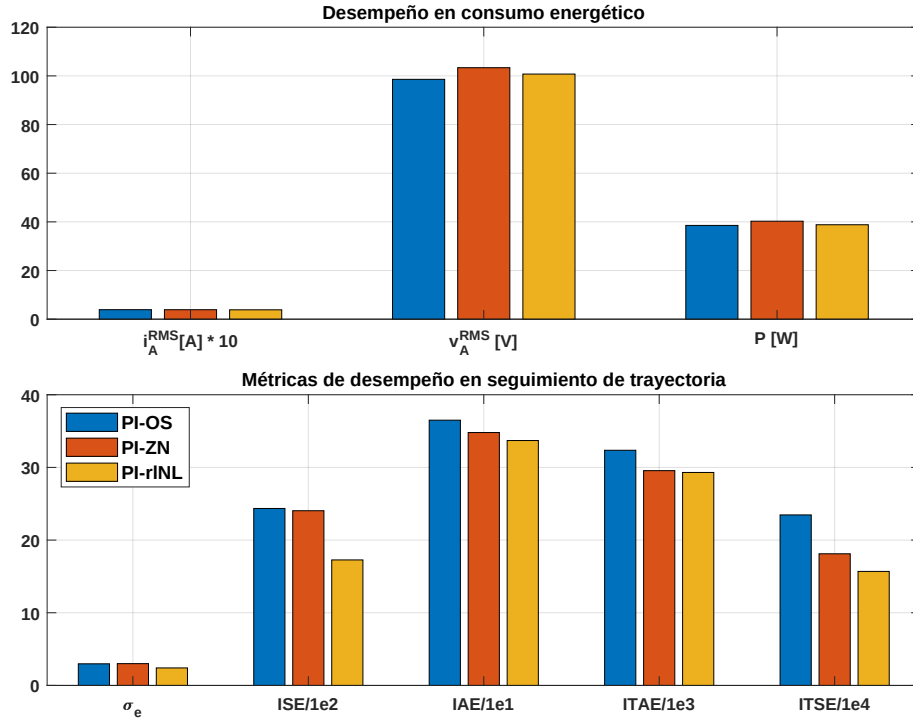


La prueba 2 consiste en rampas de aceleración y frenado de velocidad con una pendiente de 3000 RPM/s y una duración de 0.5 s (Tabla 5.2). Las métricas de evaluación del comportamiento mostrado en las figuras anteriores se presentan en la Tabla 5.10 y en la Figura 5.27.

Tabla 5.10: Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores PI frente a rampas de velocidad en implementación - Prueba 2

Métrica	PI-OS	PI-ZN	PI-rINL	Mejor(2)	Medio(1)	Peor(0)
σ_e	2.9739	2.9987	2.4074	PI-rINL	PI-OS	PI-ZN
ISE	2435	2404	1727	PI-rINL	PI-ZN	PI-OS
IAE	365	348	337	PI-rINL	PI-ZN	PI-OS
ITAE	32 360	29 554	29 306	PI-rINL	PI-ZN	PI-OS
ITSE	234 626	181 144	156 900	PI-rINL	PI-ZN	PI-OS
v_A^{RMS} [V]	98.5826	103.3670	100.7508	PI-OS	PI-rINL	PI-ZN
i_A^{RMS} [A]	0.3910	0.3896	0.3853	PI-rINL	PI-ZN	PI-OS
P [W]	38.5424	40.2730	38.8145	PI-OS	PI-rINL	PI-ZN

Figura 5.27: Gráfico de barras de las métricas de desempeño de los controladores PI frente a rampas de velocidad - Prueba 2



A partir de los resultados numéricos en las Tablas 5.10 así como en la Figura 5.27, se calcula el puntaje de Borda de cada controlador (resultando mejor PI-rINL):

$$B_{PI-OS}^{RT2} = 2 \times (2) + 1 \times (1) + 5 \times (0) = 5, \quad (5.13)$$

$$B_{PI-ZN}^{RT2} = 0 \times (2) + 5 \times (1) + 3 \times (0) = 5, \quad (5.14)$$

$$B_{PI-rINL}^{RT2} = 6 \times (2) + 2 \times (1) + 0 \times (0) = 14. \quad (5.15)$$

5.3.3.3. Resumen de desempeño - Pruebas de rampas de velocidad

Tabla 5.11: Resumen del puntaje de Borda de los controladores PI en pruebas de rampas de velocidad

Prueba	B_{PI-OS}	B_{PI-ZN}	$B_{PI-rlNL}$	Mejor	Medio	Peor
1	3	9	12	PI-rlNL	PI-OS	PI-ZN
2	5	5	14	PI-rlNL	PI-ZN	PI-OS
Promedio	4.0	7.0	13.0	PI-rlNL	PI-ZN	PI-OS

Nota. Se favorece a PI-ZN, ya que es mejor en tres casos, frente a los dos casos de PI-OS.

La Tabla 5.11 presenta un resumen del puntaje de Borda de los controladores PI (ecuación (5.10–5.15)) en pruebas de rampas de velocidad. En promedio, el controlador PI-rlNL muestra el mejor desempeño en comparación con los controladores PI-OS y PI-ZN. Por su parte, PI-ZN supera a PI-OS en estas dos pruebas.

5.3.4. Pruebas y resultados de perturbaciones de carga

5.3.4.1. Perturbaciones de carga: Prueba 1

Figura 5.28: Respuesta del controlador PI-OS (óptimo simétrico)

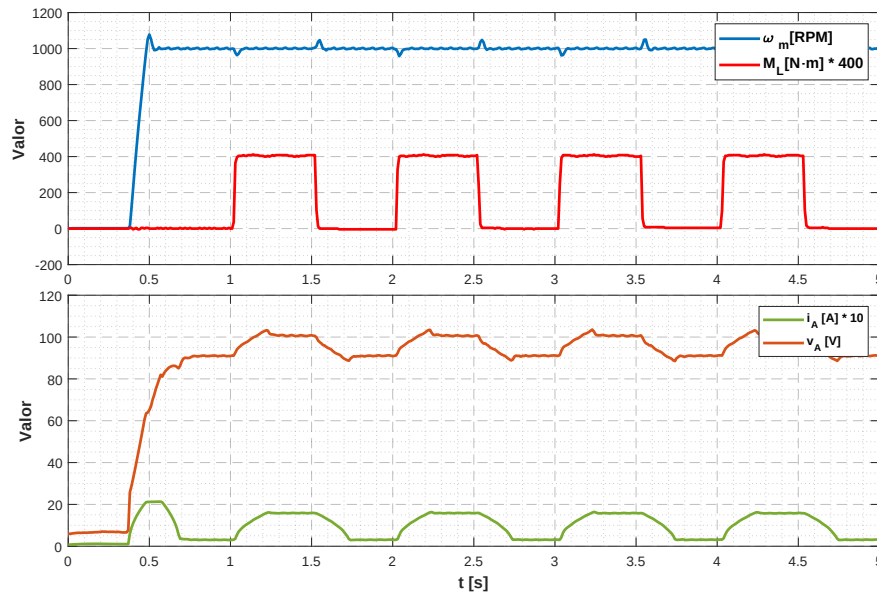


Figura 5.29: Respuesta del controlador PI-ZN (Ziegler Nichols)

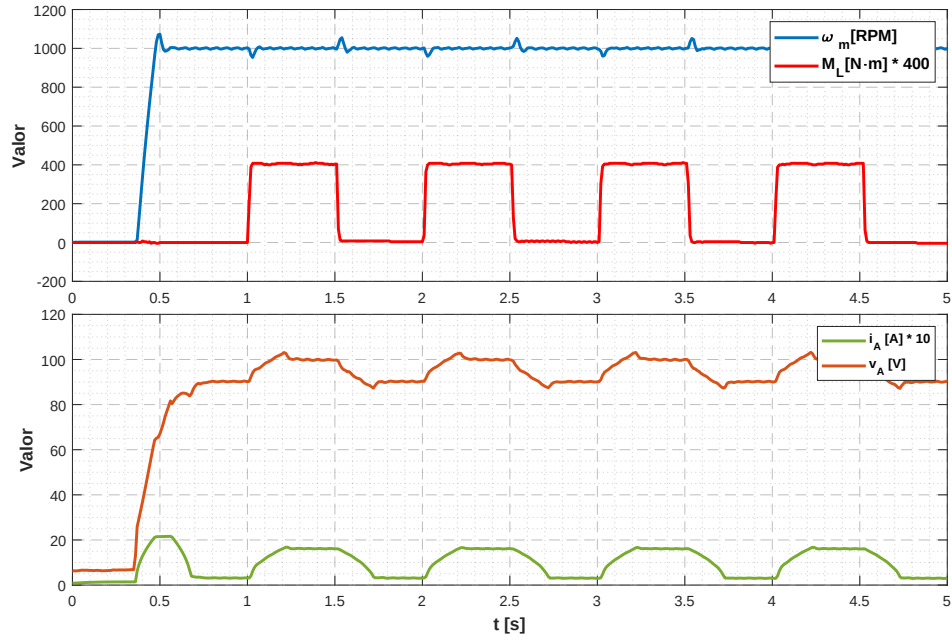
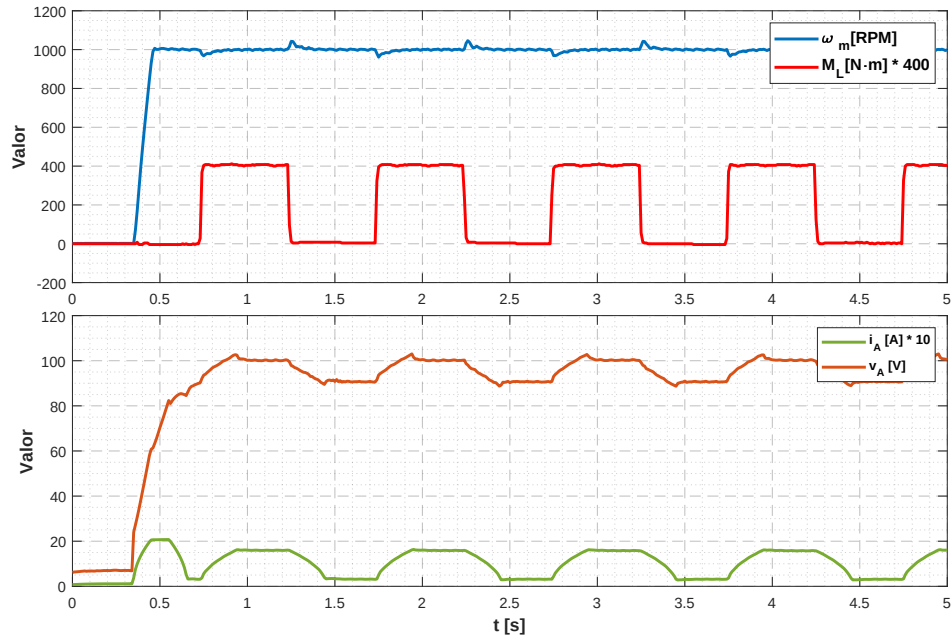


Figura 5.30: Respuesta del controlador PI-rlNL (TD3 con término no lineal)



La prueba 1 consiste en evaluar los controladores frente a un perfil de carga M_L que oscila entre $0 \text{ N} \cdot \text{m}$ y $1 \text{ N} \cdot \text{m}$, con un periodo de $T = 1 \text{ s}$ (Tabla 5.3). Las métricas de evaluación del comportamiento mostrado en las figuras anteriores se presentan en las Tablas 5.12, 5.13, así como en la Figura 5.32.

Figura 5.31: Comparación de los controladores PI - Perturbaciones de carga: prueba 1

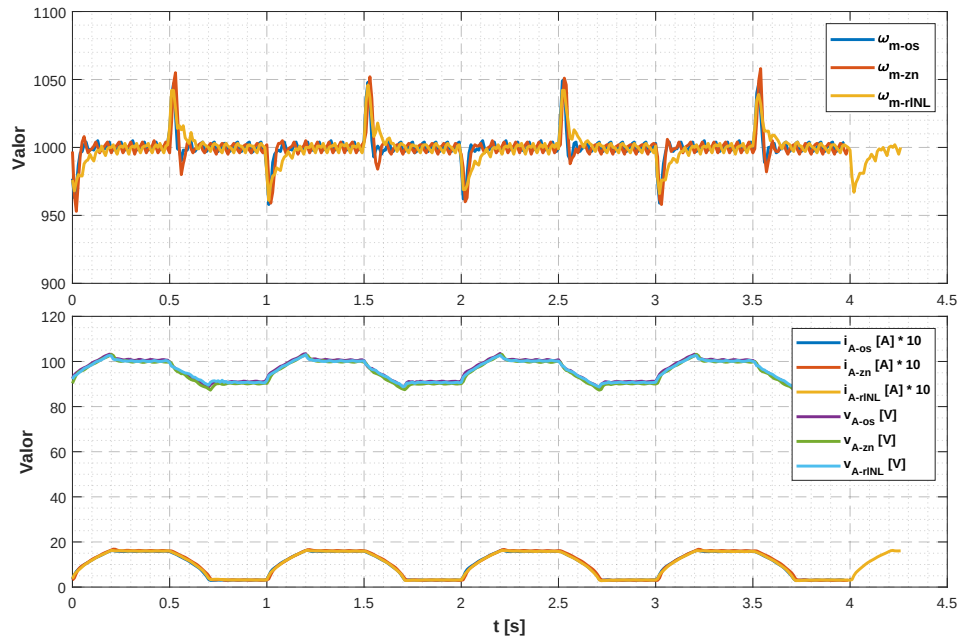


Tabla 5.12: Indicadores de desempeño transitorio de los controladores PI frente a perturbaciones de carga en implementación - Prueba 1

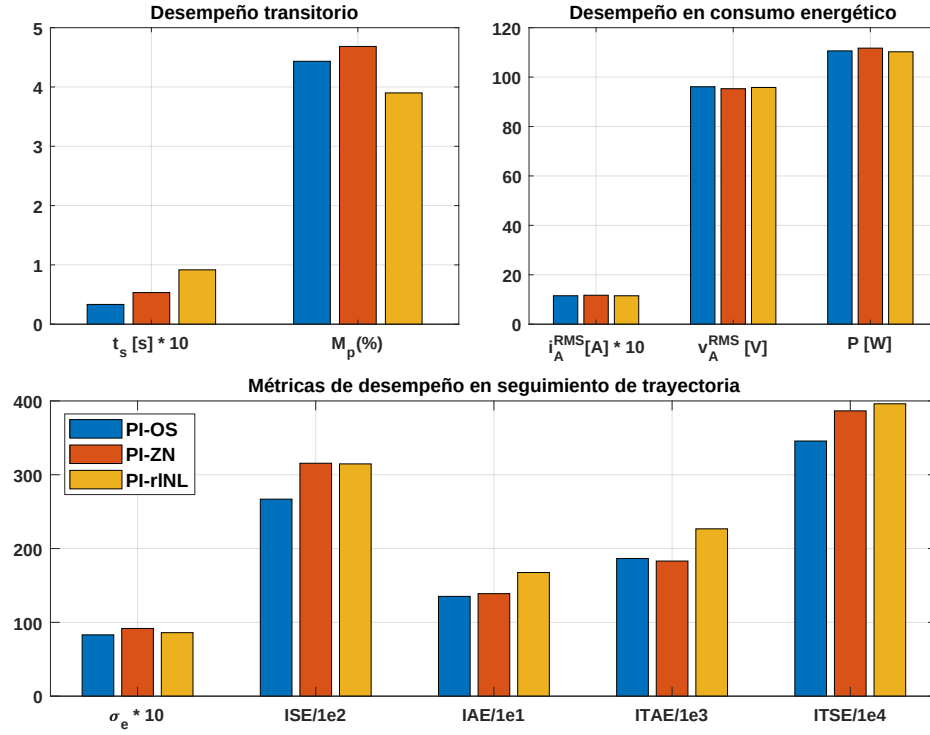
Métrica	PI-OS	PI-ZN	PI-rlNL	Mejor(2)	Medio(1)	Peor(0)
$t_s[s]$	0.0333	0.0533	0.0917	PI-OS	PI-ZN	PI-rlNL
$M_p(\%)$	4.433	4.683	3.900	PI-rlNL	PI-OS	PI-ZN

Nota. Estos valores representan el promedio de tres fluctuaciones de carga entre $0 \text{ N} \cdot \text{m}$ y $1 \text{ N} \cdot \text{m}$, con el propósito de evaluar de forma más representativa el comportamiento general de los controladores.

Tabla 5.13: Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores PI frente a perturbaciones de carga en implementación - Prueba 1

Métrica	PI-OS	PI-ZN	PI-rlNL	Mejor(2)	Medio(1)	Peor(0)
σ_e	8.3058	9.1744	8.6027	PI-OS	PI-rlNL	PI-ZN
ISE	26 692	31 557	31 469	PI-OS	PI-rlNL	PI-ZN
IAE	1352	1389	1675	PI-OS	PI-ZN	PI-rlNL
ITAE	186 531	183 047	226 701	PI-ZN	PI-OS	PI-rlNL
ITSE	3 456 821	3 865 377	3 961 213	PI-OS	PI-ZN	PI-rlNL
v_A^{RMS} [V]	96.0976	95.2991	95.8073	PI-ZN	PI-rlNL	PI-OS
i_A^{RMS} [A]	1.1510	1.1723	1.1506	PI-rlNL	PI-OS	PI-ZN
P [W]	110.6055	111.7176	110.2375	PI-rlNL	PI-OS	PI-ZN

Figura 5.32: Gráfico de barras de las métricas de desempeño de los controladores PI frente a perturbaciones de carga - Prueba 1



Las Figuras 5.28, 5.29, 5.30 y 5.31 muestran que el controlador PI-rINL presenta una respuesta suave y con menor sobreimpulso frente a los cambios repentinos de carga en comparación con los demás controladores.

De los resultados numéricos en las Tablas 5.12, 5.13 y en la Figura 5.32, se calcula el puntaje de Borda de cada controlador:

$$B_{PI-OS}^{LT1} = 5 \times (2) + 4 \times (1) + 1 \times (0) = 14, \quad (5.16)$$

$$B_{PI-ZN}^{LT1} = 2 \times (2) + 3 \times (1) + 5 \times (0) = 7, \quad (5.17)$$

$$B_{PI-rINL}^{LT1} = 3 \times (2) + 3 \times (1) + 4 \times (0) = 9. \quad (5.18)$$

Estos resultados demuestran que el controlador PI-OS obtiene el mayor puntaje de Borda en la prueba 1, con 14 puntos, seguido por PI-rINL con un puntaje de 9 y PI-ZN con 7 puntos.

5.3.4.2. Perturbaciones de carga: Prueba 2

Figura 5.33: Respuesta del controlador PI-OS (óptimo simétrico)

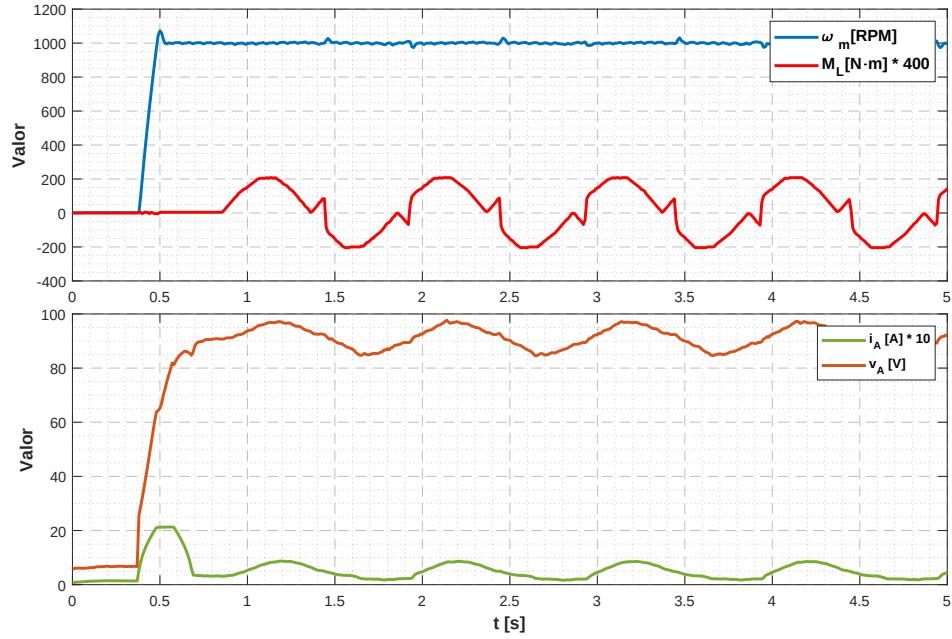


Figura 5.34: Respuesta del controlador PI-ZN (Ziegler Nichols)

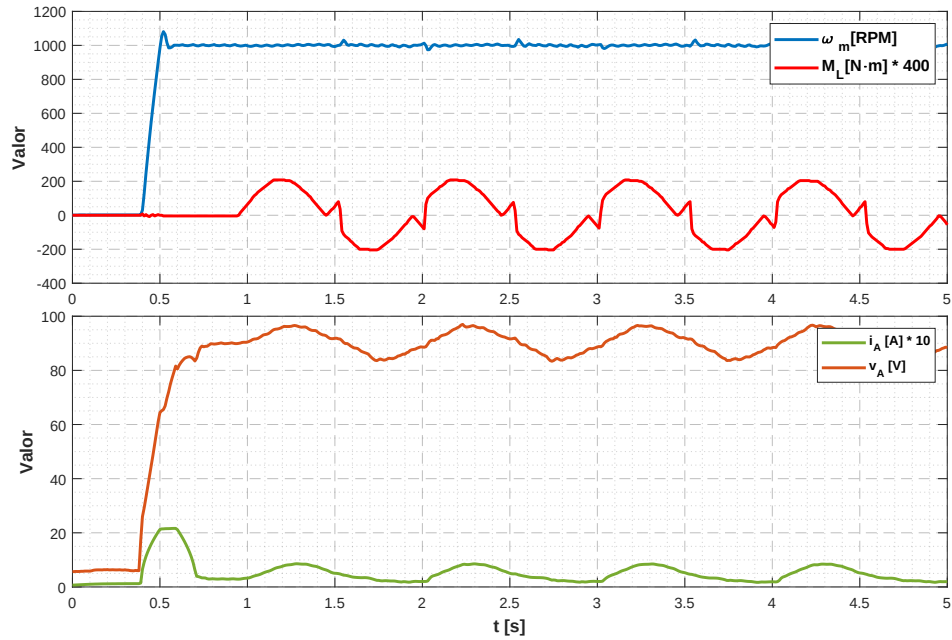


Figura 5.35: Respuesta del controlador PI-rlNL (TD3 con término no lineal)

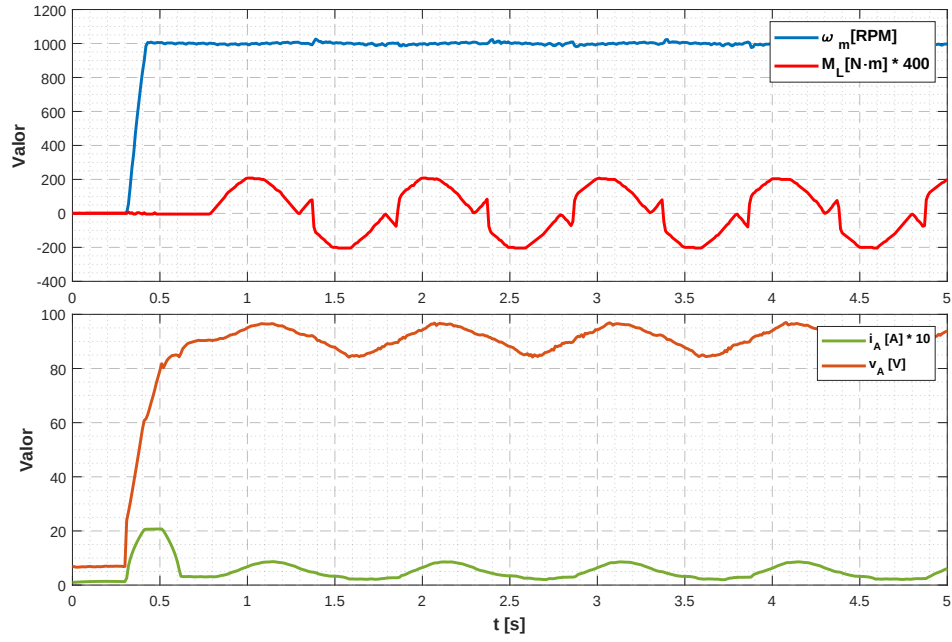
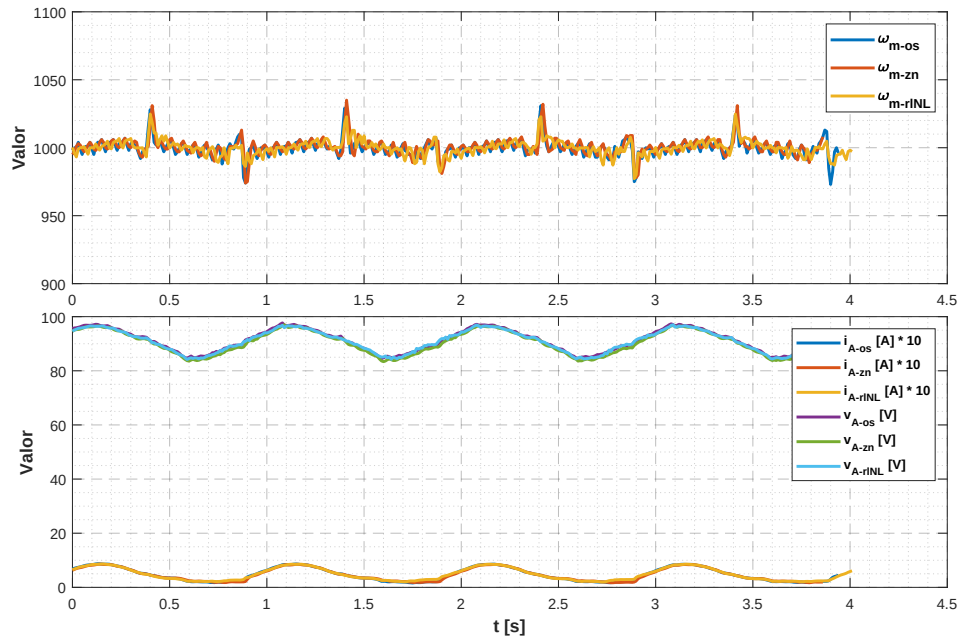


Figura 5.36: Comparación de los controladores PI - Perturbaciones de carga: prueba 2

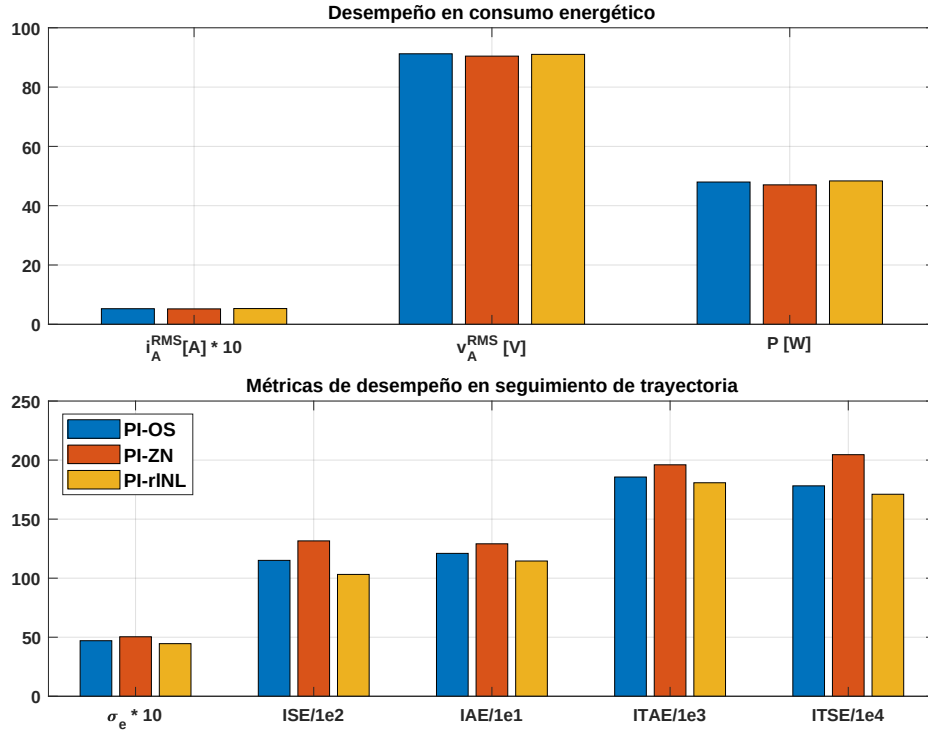


La prueba 2 consiste en evaluar los controladores frente a un perfil de carga M_L sinusoidal con amplitud de $0.5 \text{ N} \cdot \text{m}$ y periodo $T = 1 \text{ s}$ (Tabla 5.3). Las métricas de evaluación del comportamiento mostrado en las figuras anteriores se presentan en la Tabla 5.14, así como en la Figura 5.37.

Tabla 5.14: Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores PI frente a perturbaciones de carga en implementación - Prueba 2

Métrica	PI-OS	PI-ZN	PI-rINL	Mejor(2)	Medio(1)	Peor(0)
σ_e	4.7043	5.0381	4.4555	PI-rINL	PI-OS	PI-ZN
ISE	11 508	13 157	10 318	PI-rINL	PI-OS	PI-ZN
IAE	1210	1291	1145	PI-rINL	PI-OS	PI-ZN
ITAE	185 664	196 003	18 080	PI-rINL	PI-OS	PI-ZN
ITSE	1 781 814	2 045 787	1 710 708	PI-rINL	PI-OS	PI-ZN
v_A^{RMS} [V]	91.2287	90.4440	91.0319	PI-ZN	PI-rINL	PI-OS
i_A^{RMS} [A]	0.5260	0.5200	0.5312	PI-ZN	PI-OS	PI-rINL
P [W]	47.9835	47.0279	48.3535	PI-ZN	PI-OS	PI-rINL

Figura 5.37: Gráfico de barras de las métricas de desempeño de los controladores PI frente a perturbaciones de carga - Prueba 2



$$B_{PI-OS}^{LT2} = 0 \times (2) + 7 \times (1) + 1 \times (0) = 7, \quad (5.19)$$

$$B_{PI-ZN}^{LT2} = 3 \times (2) + 0 \times (1) + 5 \times (0) = 6, \quad (5.20)$$

$$B_{PI-rINL}^{LT2} = 5 \times (2) + 1 \times (1) + 2 \times (0) = 11. \quad (5.21)$$

De los resultados numéricos en la Tabla 5.14 y en la Figura 5.37, se calcula el puntaje de Borda de cada controlador, cuyos valores se muestran en las ecuaciones (5.19), (5.20) y (5.21). En esta prueba, PI-rINL obtiene el mejor resultado, seguido de PI-OS y PI-ZN.

5.3.4.3. Resumen de desempeño - Pruebas de perturbaciones de carga

Tabla 5.15: Resumen del puntaje de Borda de los controladores PI en pruebas de perturbaciones de carga

Prueba	B_{PI-OS}	B_{PI-ZN}	$B_{PI-rINL}$	Mejor	Medio	Peor
1	14	7	9	PI-rINL	PI-OS	PI-ZN
2	7	6	11	PI-rINL	PI-ZN	PI-OS
Promedio	10.5	6.5	10.0	PI-rINL	PI-ZN	PI-OS

La Tabla 5.15 presenta un resumen del puntaje de Borda de los controladores PI (ecuación (5.16–5.21)) en pruebas de perturbaciones de carga. En promedio, el controlador PI-OS muestra el mejor desempeño en comparación con los controladores PI-rINL y PI-ZN. Por su parte, PI-rINL supera a PI-ZN en estas dos pruebas.

5.3.5. Resumen general del desempeño de los controladores PI

Tomando en cuenta el rendimiento de los controladores PI-OS, PI-ZN y PI-rINL en las Tablas 5.4, 5.5, 5.6, 5.7, 5.9, 5.10, 5.12, 5.13 y 5.14 se presenta un resumen general promedio de los indicadores de desempeño transitorio y de seguimiento de trayectoria en las Tablas 5.16 y 5.17.

Tabla 5.16: Indicadores de desempeño transitorio promedio de los controladores PI

Métrica	PI-OS	PI-ZN	PI-rINL	Mejor(2)	Medio(1)	Peor(0)
$t_s[s]$	0.1369	0.1540	0.1310	PI-rINL	PI-OS	PI-ZN
$M_p(\%)$	14.7040	18.5980	5.1186	PI-rINL	PI-OS	PI-ZN

Basado en el criterio de conteo de Borda se concluye a partir de las Tablas 5.16, 5.17 y 5.18, que, en promedio, el controlador PI-rINL demuestra un mejor desempeño frente a las pruebas a las que fueron sometidos estos controladores (PI-OS, PI-ZN y PI-rINL).

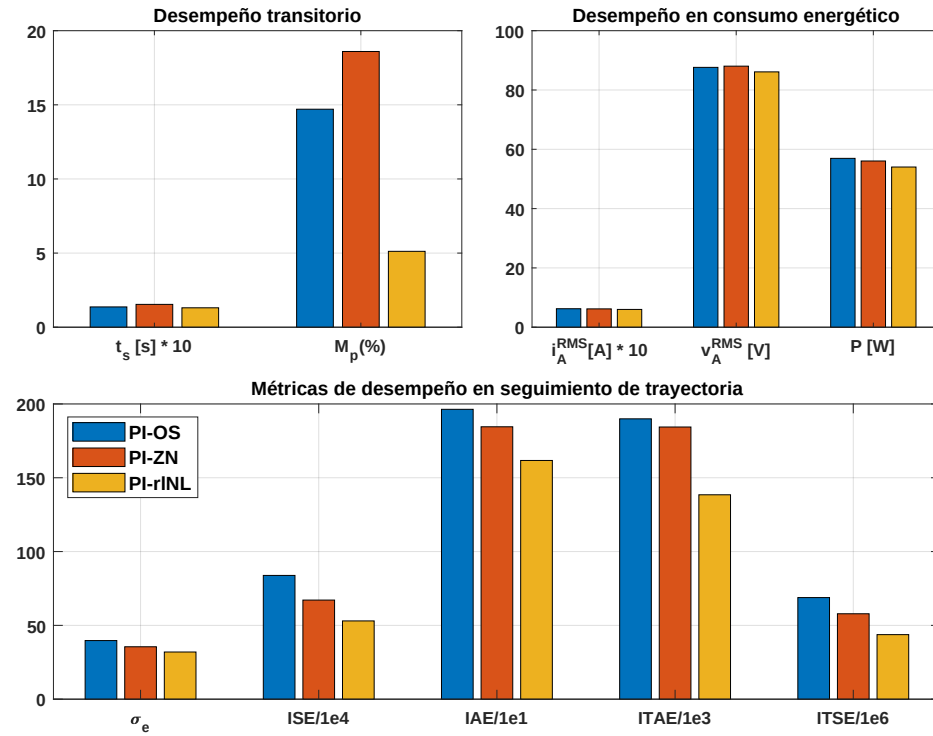
Tabla 5.17: Indicadores de desempeño promedio en seguimiento de trayectoria y consumo energético de los controladores PI

Métrica	PI-OS	PI-ZN	PI-rINL	Mejor(2)	Medio(1)	Peor(0)
σ_e	39.7055	35.5055	31.9563	PI-rINL	PI-ZN	PI-OS
ISE	838 369.7	671 381.3	530 151.7	PI-rINL	PI-ZN	PI-OS
IAE	1963.7	1845.3	1617.2	PI-rINL	PI-ZN	PI-OS
ITAE	189 916.8	184 384.2	138 464.3	PI-rINL	PI-ZN	PI-OS
ITSE	68 828 551.2	57 863 837.2	43 728 694.7	PI-rINL	PI-ZN	PI-OS
v_A^{RMS} [V]	87.6322	88.0411	86.1098	PI-rINL	PI-OS	PI-ZN
i_A^{RMS} [A]	0.6235	0.6184	0.59995	PI-rINL	PI-ZN	PI-OS
P [W]	56.9473	56.0560	54.0262	PI-rINL	PI-ZN	PI-OS

Tabla 5.18: Puntaje de Borda basado en las métricas de desempeño promedio de los controladores PI

	B_{PI-OS}	B_{PI-ZN}	$B_{PI-rINL}$	Mejor	Medio	Peor
Puntaje	3	7	20	PI-rINL	PI-ZN	PI-OS

Figura 5.38: Métricas de desempeño promedio de los controladores PI



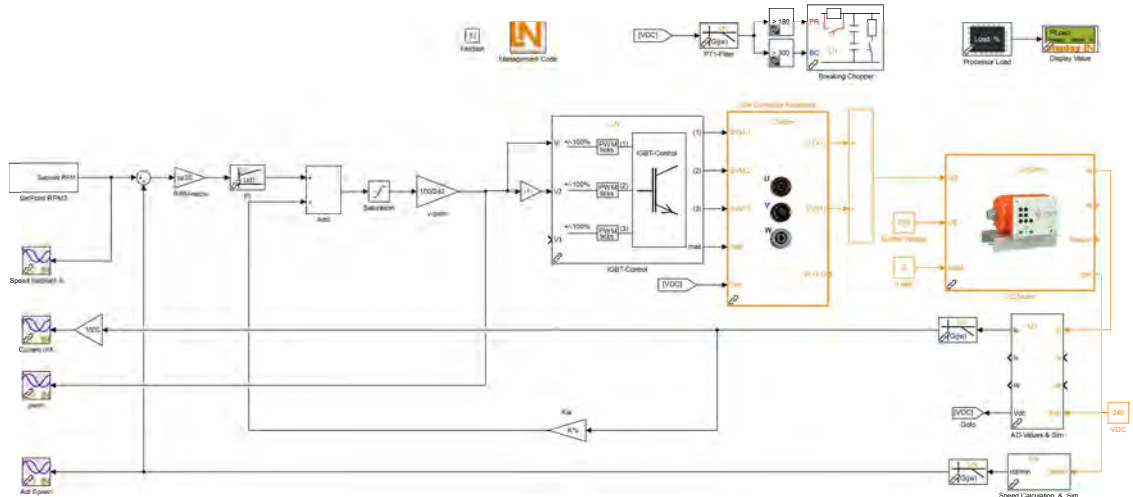
Nota. Este gráfico de barras resume el desempeño general frente a las seis pruebas a las cuales han estado sujetas los controladores.

En particular, PI-rlNL presenta una respuesta suave y estable en la variable velocidad ω_m en comparación con PI-OS y PI-ZN. Además, este controlador muestra una respuesta rápida y un bajo consumo energético promedio, lo que evidencia su efectividad y eficiencia en el control de velocidad de un motor DC con devanado de campo. Estos resultados sugieren que el enfoque de diseño basado en aprendizaje por refuerzo y control no lineal es adecuado para este tipo de aplicaciones.

5.4. Implementación, pruebas y resultados de los controladores LQR

A partir de la arquitectura del controlador LQR presentada en la Figura 4.20, se implementa la rutina de control en Simulink r2016a, tal como se muestra en la Figura 5.39.

Figura 5.39: Programa en Simulink del controlador LQR



Las ganancias del controlador LQR de velocidad, diseñadas en el capítulo anterior, se integran en el bloque de control PI y la ganancia K_{ia} , aprovechando sus rutinas de discretización y protección.

La salida del controlador, correspondiente a la señal de voltaje de armadura v_A , se satura en el intervalo $[0, 240]$ V para no exceder el límite de alimentación de los contro-

ladores de potencia. Asimismo, se emplea un bloque de escalamiento para transformar dicha señal en el ciclo de trabajo D_{PWM} aplicado a los convertidores (Linnertz, 2016).

Observación: Las ganancias de los controladores LQR se encuentran escaladas en unidades de velocidad angular rad/s. En consecuencia, es necesario realizar una conversión de escala a unidades de velocidad en RPM para su despliegue.

5.4.1. Pruebas y resultados de cambio de referencia

5.4.1.1. Cambios de referencia: Prueba 1

Figura 5.40: Respuesta del controlador LQR-RICC (algoritmo de Hewer)

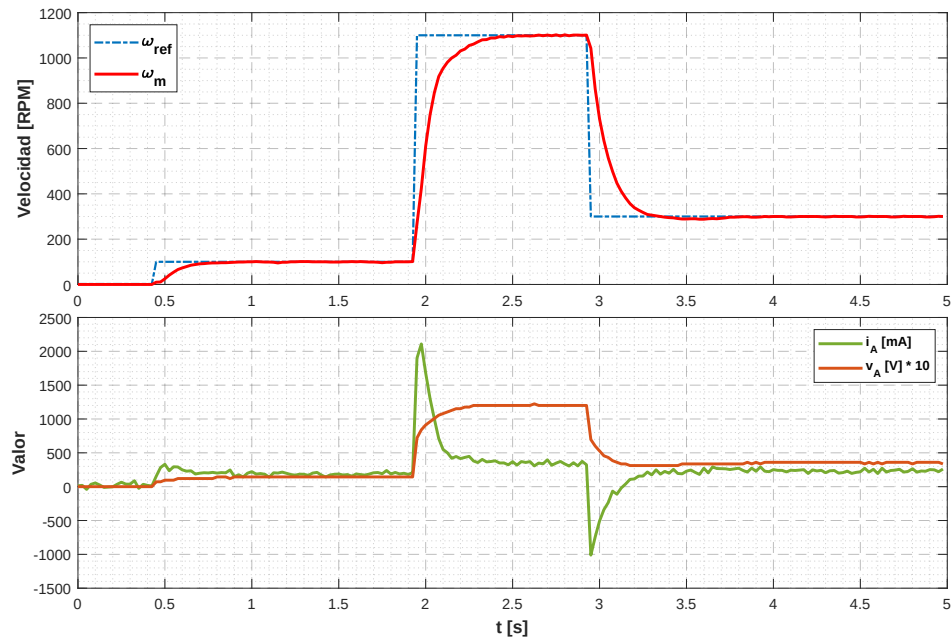


Figura 5.41: Respuesta del controlador LQR-MRL (algoritmo MGPI)

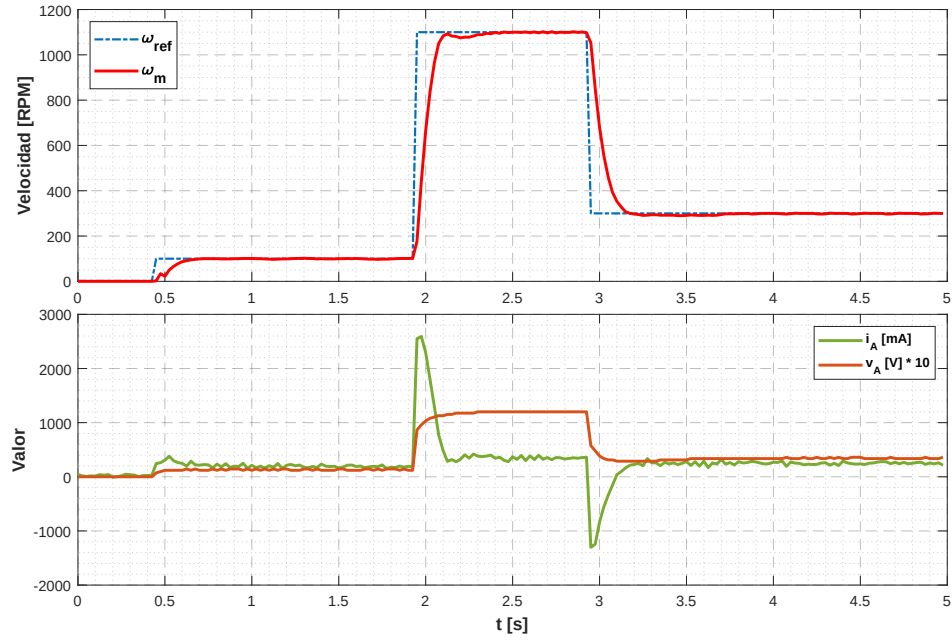
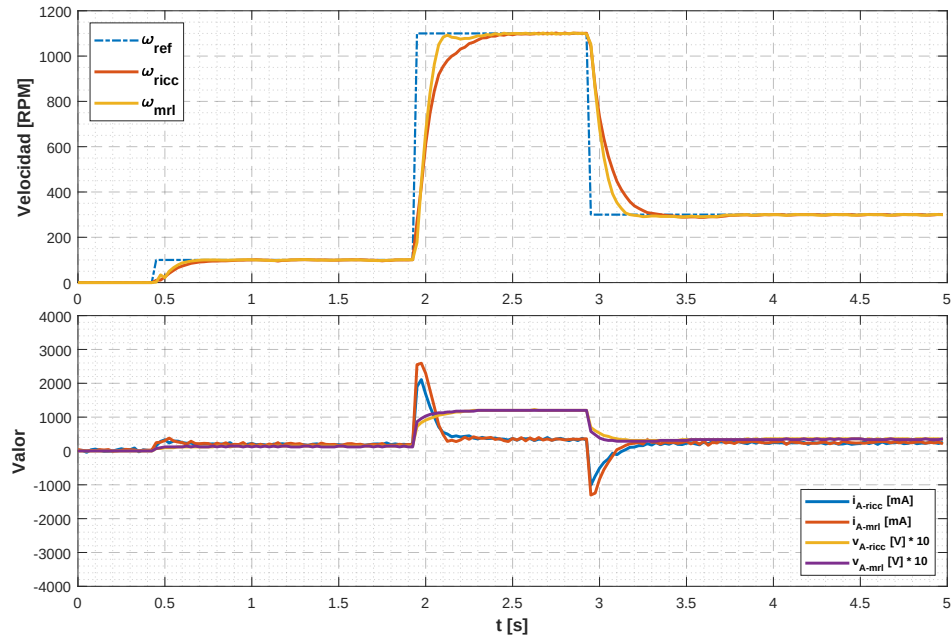


Figura 5.42: Comparación de los controladores LQR - Cambios de referencia: prueba 1



La prueba 1 consiste en tres cambios de referencia entre velocidades bajas 100 RPM, 300 RPM y medias 1100 RPM (Tabla 5.1). Dado que se evalúan tres cambios de referencia, se obtienen tres respuestas transitorias. La evaluación correspondiente a las pruebas mostradas en las figuras anteriores se presenta en las Tablas 5.19, 5.20 y en la Figura 5.43.

Tabla 5.19: Indicadores de desempeño transitorio de los controladores LQR frente a cambios de referencia en implementación - Prueba 1

Métrica	LQR-RICC	LQR-MRL	Mejor(1)	Peor(0)
$t_{s1}[s]$	1.3500	1.3250	LQR-MRL	LQR-RICC
$M_{p1}(\%)$	0.0000	0.0000	LQR-RICC	LQR-MRL
$t_{s2}[s]$	0.5000	0.3750	LQR-MRL	LQR-RICC
$M_{p2}(\%)$	0.182	0.091	LQR-MRL	LQR-RICC
$t_{s3}[s]$	0.8000	0.7750	LQR-MRL	LQR-RICC
$M_{p3}(\%)$	-4.000	-3.333	LQR-MRL	LQR-RICC

Nota. En caso de empate, se prioriza el controlador LQR-RICC con el fin de evitar un sesgo en favor del controlador LQR-MRL.

Tabla 5.20: Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores LQR frente a cambios de referencia en implementación - Prueba 1

Métrica	LQR-RICC	LQR-MRL	Mejor(1)	Peor(0)
σ_e	117.3790	113.0139	LQR-MRL	LQR-RICC
ISE	3 028 772	2 712 855	LQR-MRL	LQR-RICC
IAE	7392	5629	LQR-MRL	LQR-RICC
ITAE	707 343	534 709	LQR-MRL	LQR-RICC
ITSE	294 505 865	259 791 951	LQR-MRL	LQR-RICC
$v_A^{RMS} [V]$	56.9941	57.0319	LQR-RICC	LQR-MRL
$i_A^{RMS} [A]$	0.3697	0.4468	LQR-RICC	LQR-MRL
$P [W]$	21.0709	25.4844	LQR-RICC	LQR-MRL

Las Figuras 5.40, 5.41, 5.42 muestran que el controlador LQR-MRL presenta una respuesta más rápida en comparación con el controlador LQR-RICC.

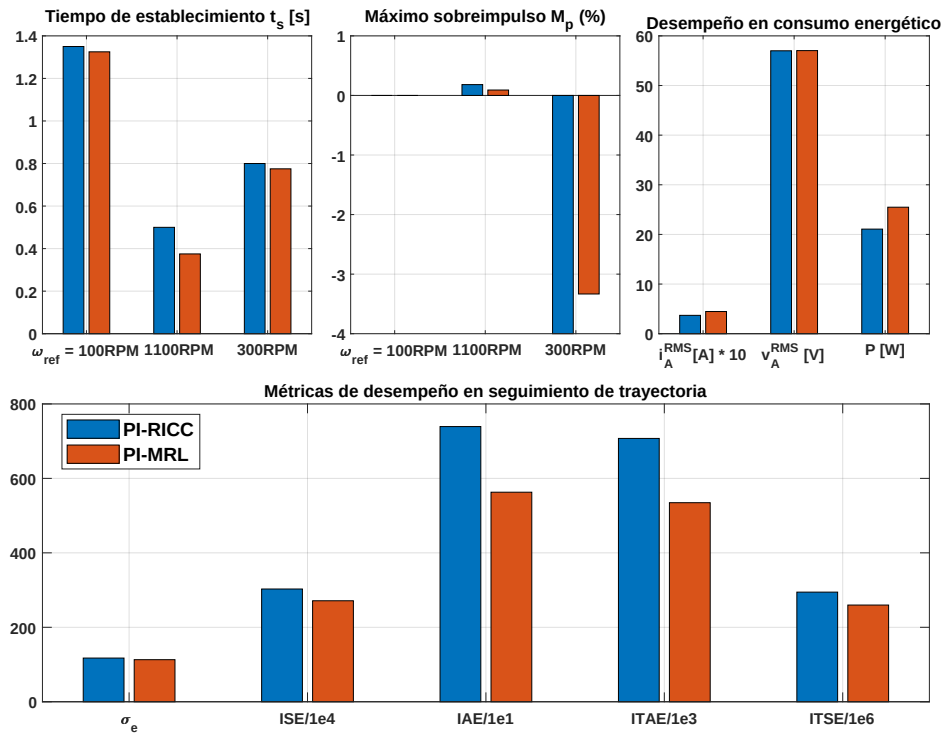
De los resultados numéricos en las Tablas 5.19, 5.20 y en la Figura 5.43, se observa que se consideran 14 casos de evaluación, y se calcula el puntaje de Borda de cada controlador:

$$B_{LQR-RICC}^{ST1} = 4 \times (1) + 10 \times (0) = 4, \quad (5.22)$$

$$B_{LQR-MRL}^{ST1} = 10 \times (1) + 4 \times (0) = 10. \quad (5.23)$$

Estos resultados sugieren que el controlador LQR-MRL obtiene el mayor puntaje de Borda en la prueba 1, con 10 puntos, y por tanto tiene un mejor rendimiento.

Figura 5.43: Gráfico de barras de las métricas de desempeño de los controladores LQR frente a cambios de referencia - Prueba 1



5.4.1.2. Cambios de referencia: Prueba 2

Figura 5.44: Respuesta del controlador LQR-RICC (algoritmo de Hewer)

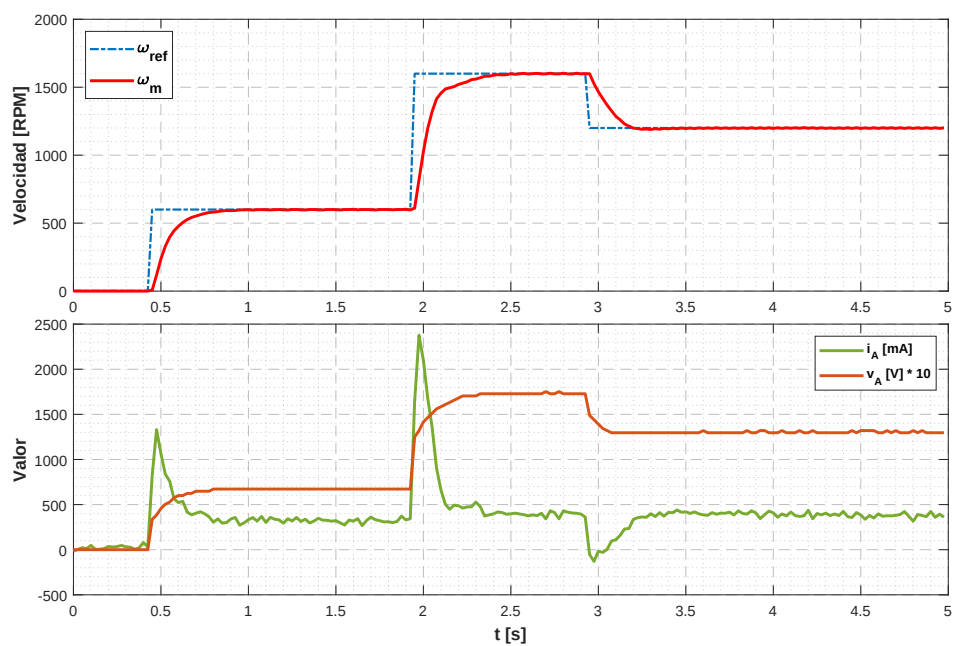


Figura 5.45: Respuesta del controlador LQR-MRL (algoritmo MGPI)

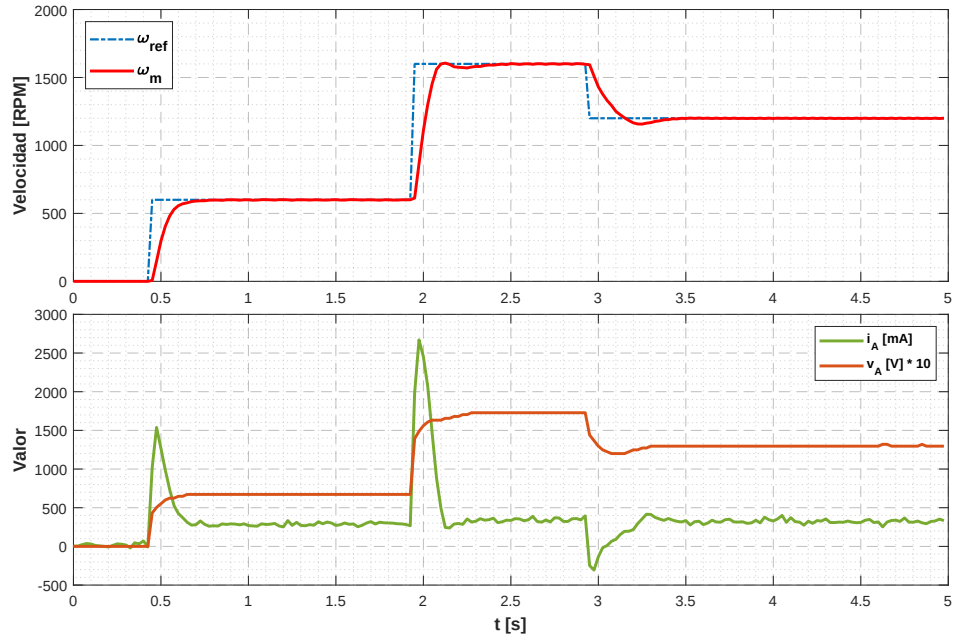
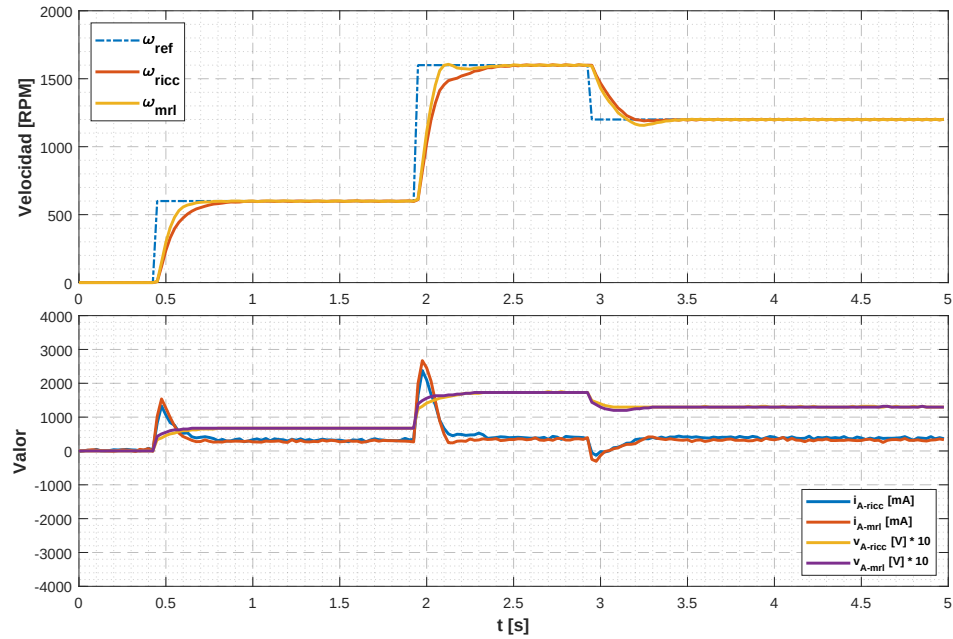


Figura 5.46: Comparación de los controladores LQR - Cambios de referencia: prueba 2



La prueba 2 consiste en tres cambios de referencia entre velocidades medias 600 RPM y altas 1600 RPM, 1200 RPM (Tabla 5.1). Dado que se evalúan tres cambios de referencia, se obtienen tres respuestas transitorias. La evaluación correspondiente a las pruebas mostradas en las figuras anteriores se presenta en las Tablas 5.21, 5.22 y en la Figura 5.47.

Tabla 5.21: Indicadores de desempeño transitorio de los controladores LQR frente a cambios de referencia en implementación - Prueba 2

Métrica	LQR-RICC	LQR-MRL	Mejor(1)	Peor(0)
$t_{s1}[s]$	0.4250	0.2750	LQR-MRL	LQR-RICC
$M_{p1}(\%)$	0.000	0.167	LQR-RICC	LQR-MRL
$t_{s2}[s]$	0.4750	0.4250	LQR-MRL	LQR-RICC
$M_{p2}(\%)$	0.125	0.375	LQR-RICC	LQR-MRL
$t_{s3}[s]$	0.2750	0.4500	LQR-RICC	LQR-MRL
$M_{p3}(\%)$	-0.917	-3.583	LQR-RICC	LQR-MRL

Tabla 5.22: Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores LQR frente a cambios de referencia en implementación - Prueba 2

Métrica	LQR-RICC	LQR-MRL	Mejor(1)	Peor(0)
σ_e	126.9446	116.8793	LQR-MRL	LQR-RICC
ISE	3 602 134	2 954 667	LQR-MRL	LQR-RICC
IAE	8708	6671	LQR-MRL	LQR-RICC
ITAE	638 700	503 733	LQR-MRL	LQR-RICC
ITSE	253 483 536	208 506 419	LQR-MRL	LQR-RICC
$v_A^{\text{RMS}} [\text{V}]$	117.7530	117.8171	LQR-RICC	LQR-MRL
$i_A^{\text{RMS}} [\text{A}]$	0.4855	0.4882	LQR-RICC	LQR-MRL
$P [\text{W}]$	57.1663	57.5204	LQR-RICC	LQR-MRL

A partir de los resultados numéricos en las Tablas 5.21 y 5.22, así como en la Figura 5.47, se calcula el puntaje de Borda de cada controlador:

$$B_{\text{LQR-RICC}}^{\text{ST2}} = 7 \times (1) + 7 \times (0) = 7, \quad (5.24)$$

$$B_{\text{LQR-MRL}}^{\text{ST2}} = 7 \times (1) + 7 \times (0) = 7, \quad (5.25)$$

Estos resultados indican que los controladores obtienen la misma puntuación de Borda, sugiriendo que los dos tienen un comportamiento similar en esta prueba.

5.4.1.3. Resumen de desempeño - Pruebas de cambio de referencia

La Tabla 5.23 resume el puntaje de Borda de los controladores LQR (ecuación (5.22–5.25)) en pruebas de cambio de referencia. En promedio, LQR-MRL presenta un mejor desempeño global que LQR-RICC. Este último exhibe una respuesta más lenta y suave,

Figura 5.47: Gráfico de barras de las métricas de desempeño de los controladores LQR frente a cambios de referencia - Prueba 2

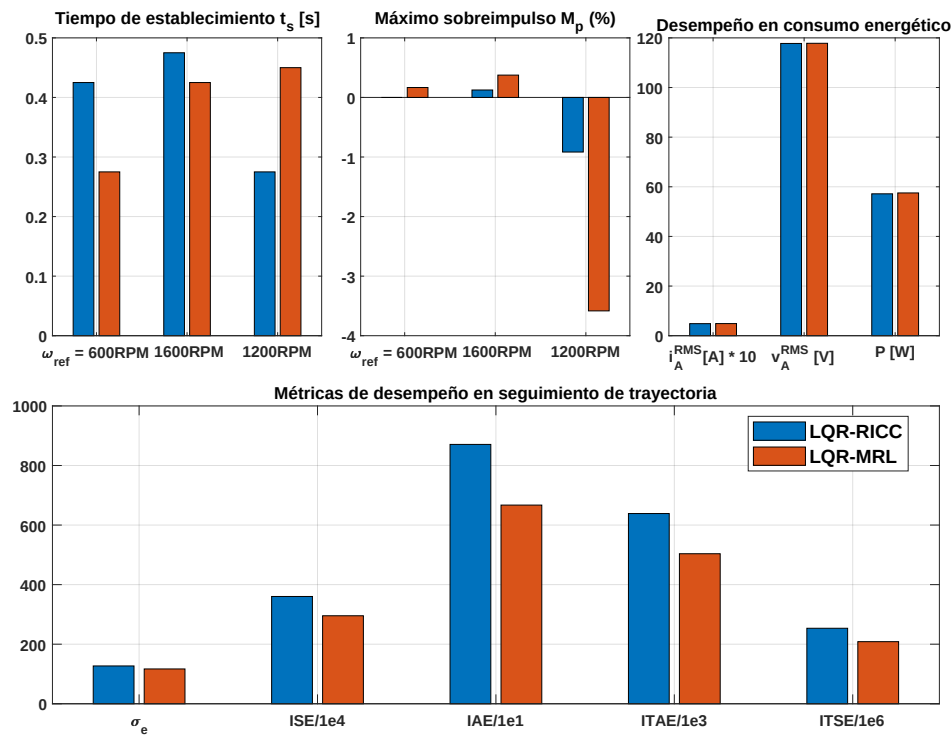


Tabla 5.23: Resumen del puntaje de Borda de los controladores LQR en pruebas de cambio de referencia

Prueba	$B_{LQR-RICC}$	$B_{LQR-MRL}$	Mejor	Peor
1	4	10	LQR-MRL	LQR-RICC
2	7	7	LQR-RICC	LQR-MRL
Promedio	5.5	8.5	LQR-MRL	LQR-RICC

Nota. En caso de empate, se prioriza el controlador LQR-RICC con el fin de evitar un sesgo en favor del controlador LQR-MRL.

favoreciendo el consumo energético, mientras que LQR-MRL presenta una respuesta más rápida y agresiva, con mejor desempeño en seguimiento de la referencia, alcanzando un consumo energético máximo de 4.8 W respecto de LQR-RICC.

5.4.2. Pruebas y resultados de rampas de velocidad

5.4.2.1. Rampas de velocidad: Prueba 1

Figura 5.48: Respuesta del controlador LQR-RICC (algoritmo de Hewer)

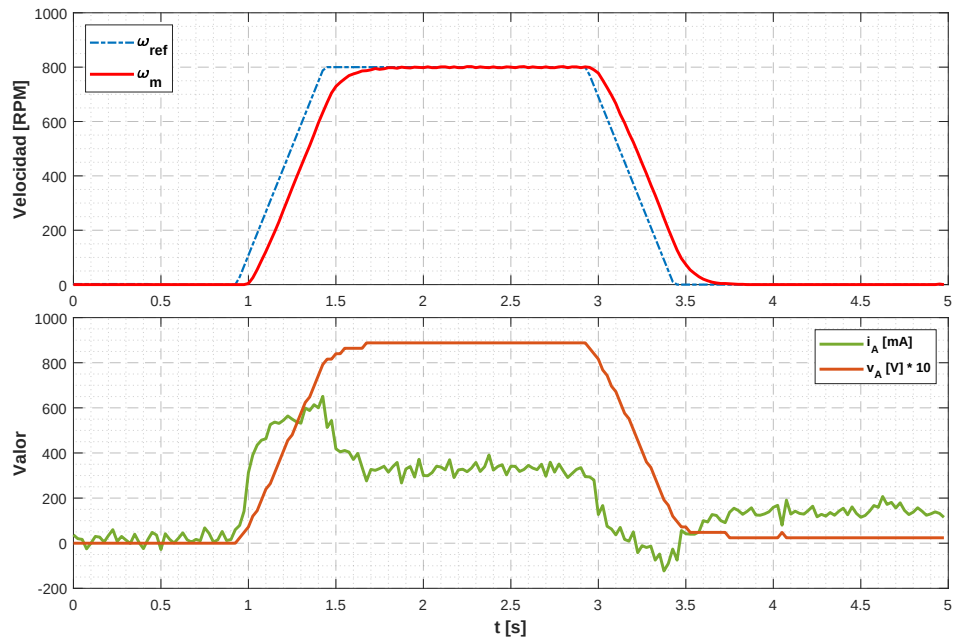


Figura 5.49: Respuesta del controlador LQR-MRL (algoritmo MGPI)

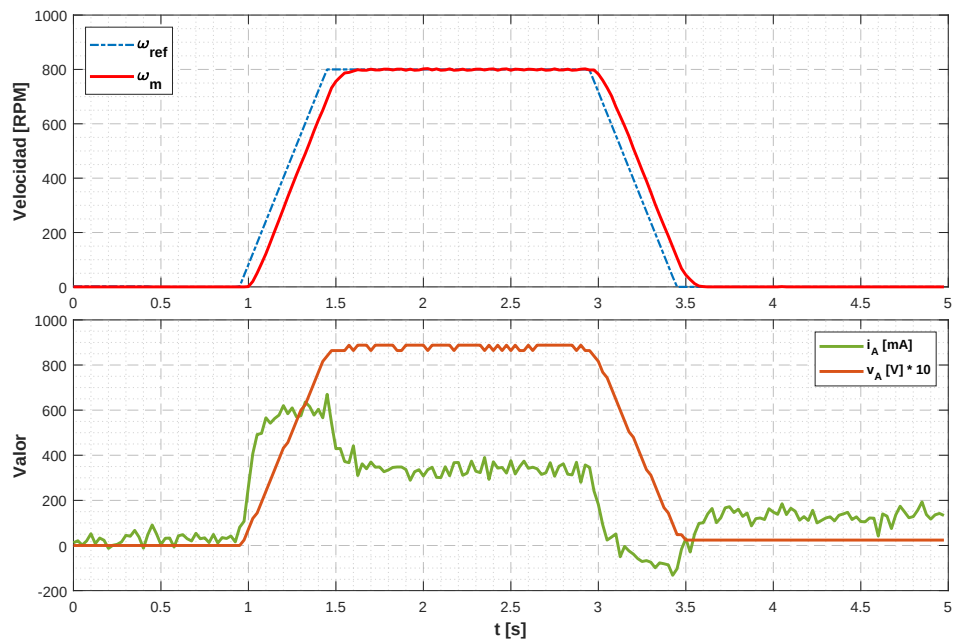
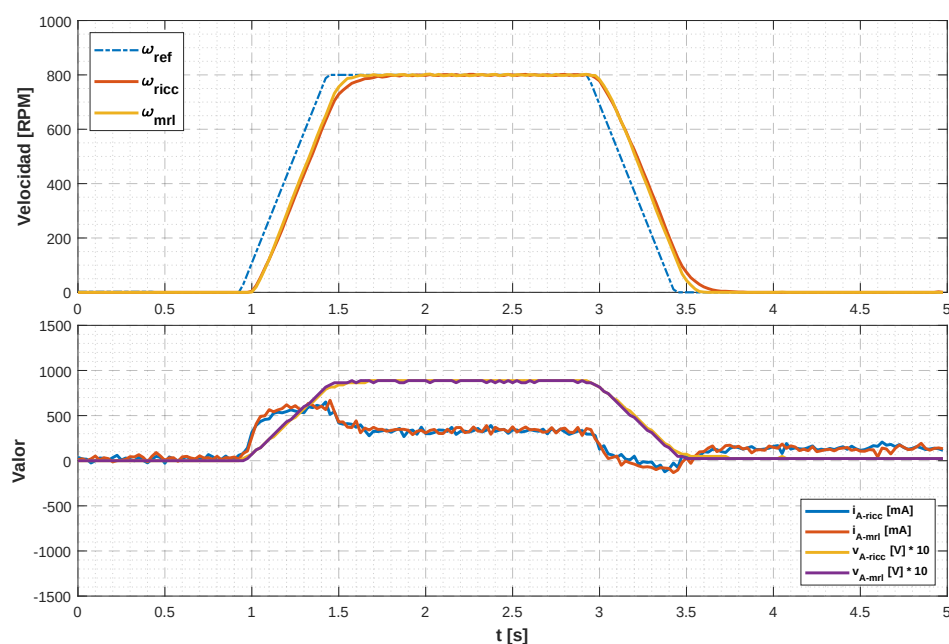


Figura 5.50: Comparación de los controladores LQR - Rampas de velocidad: prueba 1

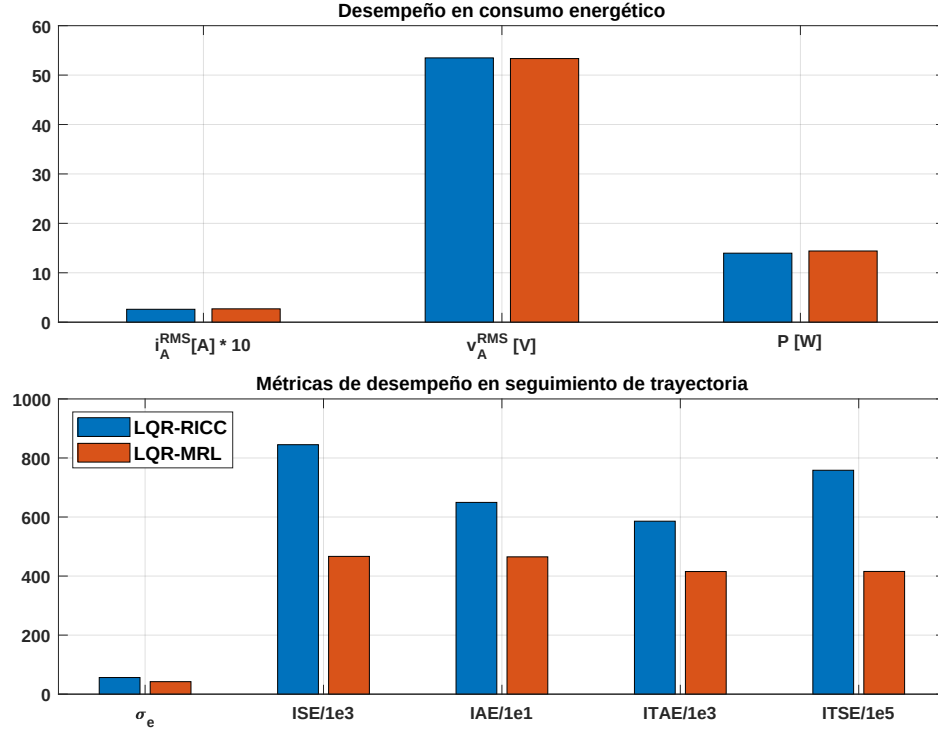


La prueba 1 consiste en rampas de aceleración y frenado de velocidad con una pendiente de 1600 RPM/s y una duración de 0.5 s (Tabla 5.2). Las métricas de evaluación del comportamiento mostrado en las figuras anteriores se presentan en la Tabla 5.24 y en la Figura 5.51.

Tabla 5.24: Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores LQR frente a rampas de velocidad en implementación - Prueba 1

Métrica	LQR-RICC	LQR-MRL	Mejor(1)	Peor(0)
σ_e	56.3051	42.3370	LQR-MRL	LQR-RICC
ISE	844 979	466 643	LQR-MRL	LQR-RICC
IAE	6495	4651	LQR-MRL	LQR-RICC
ITAE	585 845	415 351	LQR-MRL	LQR-RICC
ITSE	75 840 407	41 565 589	LQR-MRL	LQR-RICC
v_A^{RMS} [V]	53.4791	53.3435	LQR-MRL	LQR-RICC
i_A^{RMS} [A]	0.2612	0.2701	LQR-RICC	LQR-MRL
P [W]	13.9669	14.4081	LQR-RICC	LQR-MRL

Figura 5.51: Gráfico de barras de las métricas de desempeño de los controladores LQR frente a rampas de velocidad - Prueba 1



A partir de los resultados numéricos en las Tablas 5.24 así como en la Figura 5.51, se calcula el puntaje de Borda de cada controlador:

$$B_{LQR-RICC}^{RT1} = 2 \times (1) + 6 \times (0) = 2, \quad (5.26)$$

$$B_{LQR-MRL}^{RT1} = 6 \times (1) + 2 \times (0) = 6. \quad (5.27)$$

Estos resultados indican que LQR-MRL logra un mejor rendimiento, al responder más rápidamente a las rampas de velocidad, con un incremento de solo 0.5 W en el consumo energético en comparación con LQR-RICC.

5.4.2.2. Rampas de velocidad: Prueba 2

Figura 5.52: Respuesta del controlador LQR-RICC (algoritmo de Hewer)

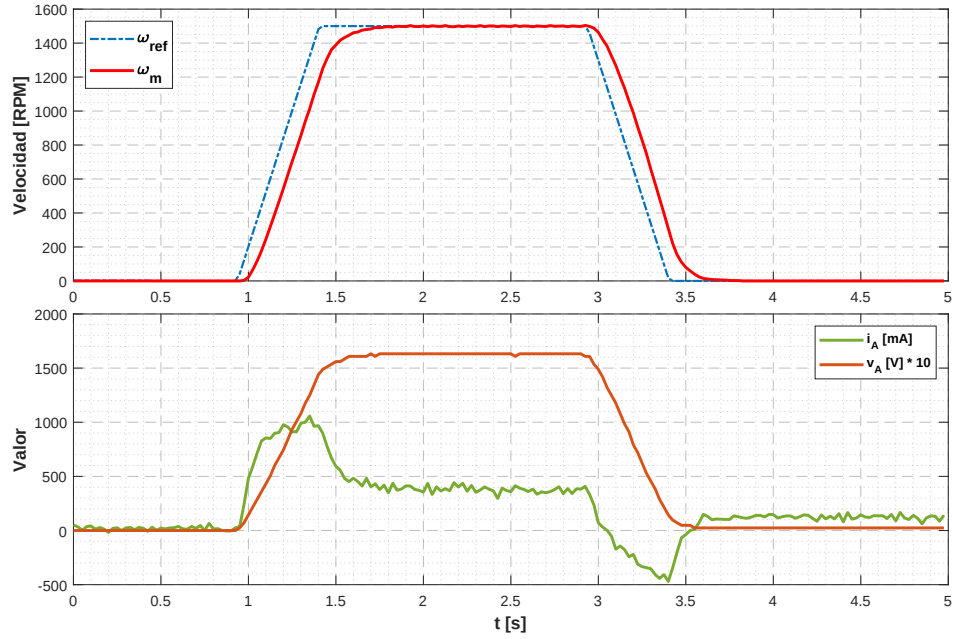


Figura 5.53: Respuesta del controlador LQR-MRL (algoritmo MGPI)

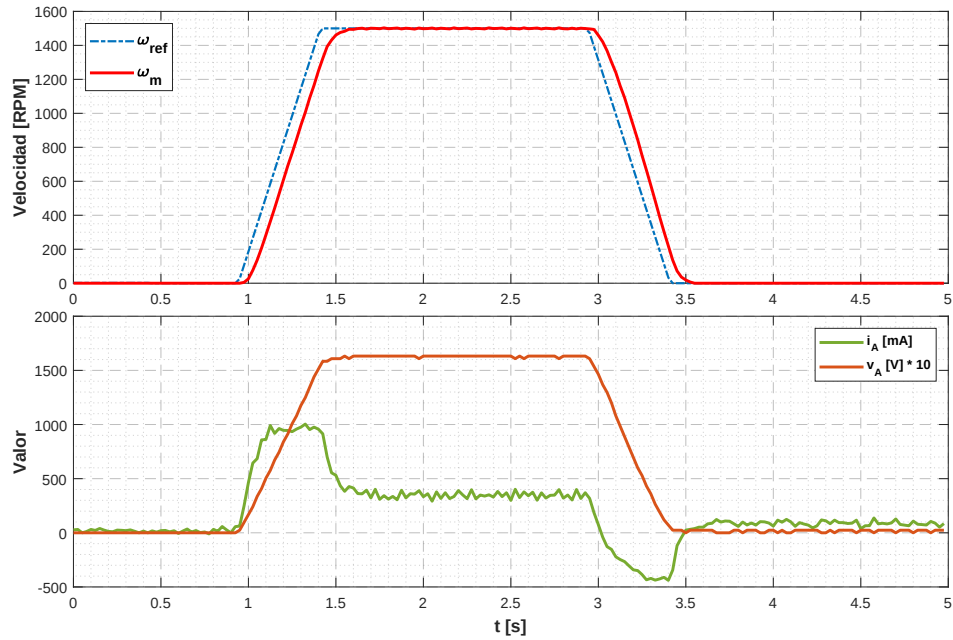
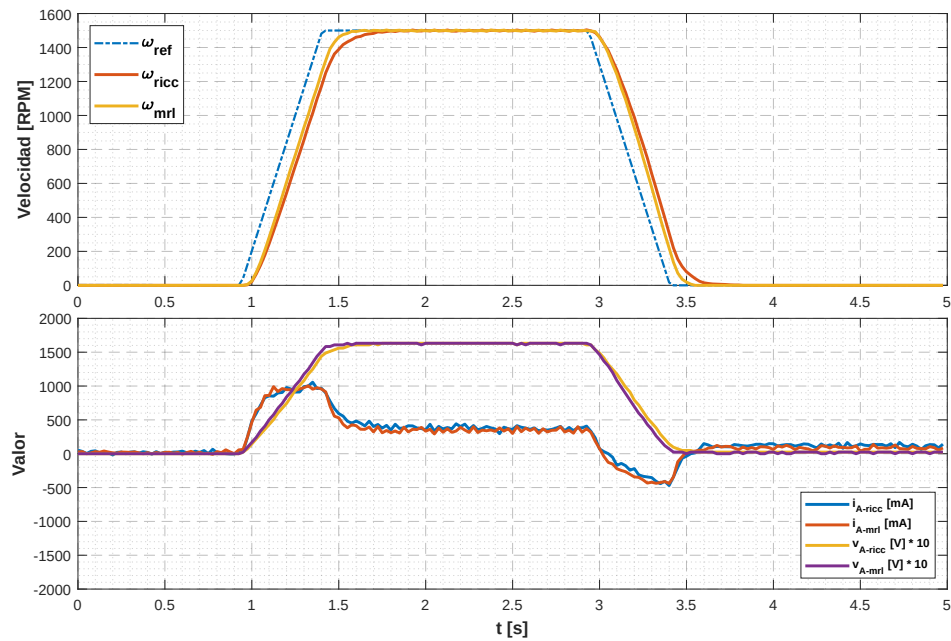


Figura 5.54: Comparación de los controladores LQR - Rampas de velocidad: prueba 2

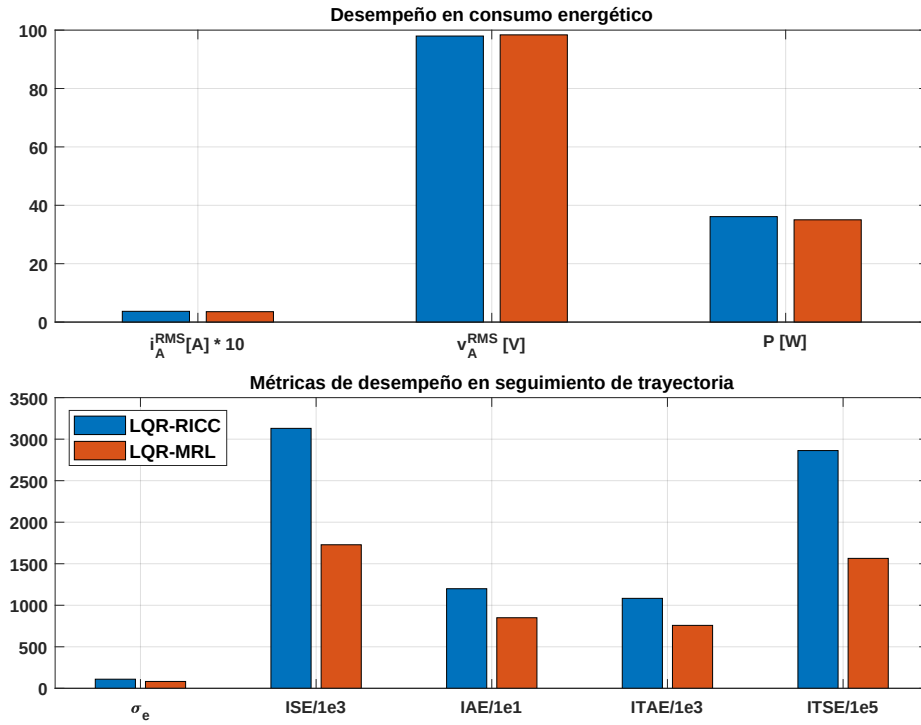


La prueba 2 consiste en rampas de aceleración y frenado de velocidad con una pendiente de 3000 RPM/s y una duración de 0.5 s (Tabla 5.2). Las métricas de evaluación del comportamiento mostrado en las figuras anteriores se presentan en la Tabla 5.25 y en la Figura 5.55.

Tabla 5.25: Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores LQR frente a rampas de velocidad en implementación - Prueba 2

Métrica	LQR-RICC	LQR-MRL	Mejor(1)	Peor(0)
σ_e	109.7944	82.6856	LQR-MRL	LQR-RICC
ISE	3 130 004	1 728 462	LQR-MRL	LQR-RICC
IAE	11 992	8498	LQR-MRL	LQR-RICC
ITAE	1 083 240	757 875	LQR-MRL	LQR-RICC
ITSE	286 272 308	156 462 527	LQR-MRL	LQR-RICC
v_A^{RMS} [V]	97.9709	98.3745	LQR-RICC	LQR-MRL
i_A^{RMS} [A]	0.3687	0.3561	LQR-MRL	LQR-RICC
P [W]	36.1181	35.0276	LQR-MRL	LQR-RICC

Figura 5.55: Gráfico de barras de las métricas de desempeño de los controladores LQR frente a rampas de velocidad - Prueba 2



A partir de los resultados numéricos en las Tablas 5.25 así como en la Figura 5.55, se calcula el puntaje de Borda de cada controlador (resultando mejor LQR-MRL):

$$B_{LQR-RICC}^{RT2} = 1 \times (1) + 7 \times (0) = 1, \quad (5.28)$$

$$B_{LQR-MRL}^{RT2} = 7 \times (1) + 1 \times (0) = 7. \quad (5.29)$$

5.4.2.3. Resumen de desempeño - Pruebas de rampas de velocidad

Tabla 5.26: Resumen del puntaje de Borda de los controladores LQR en pruebas de rampas de velocidad

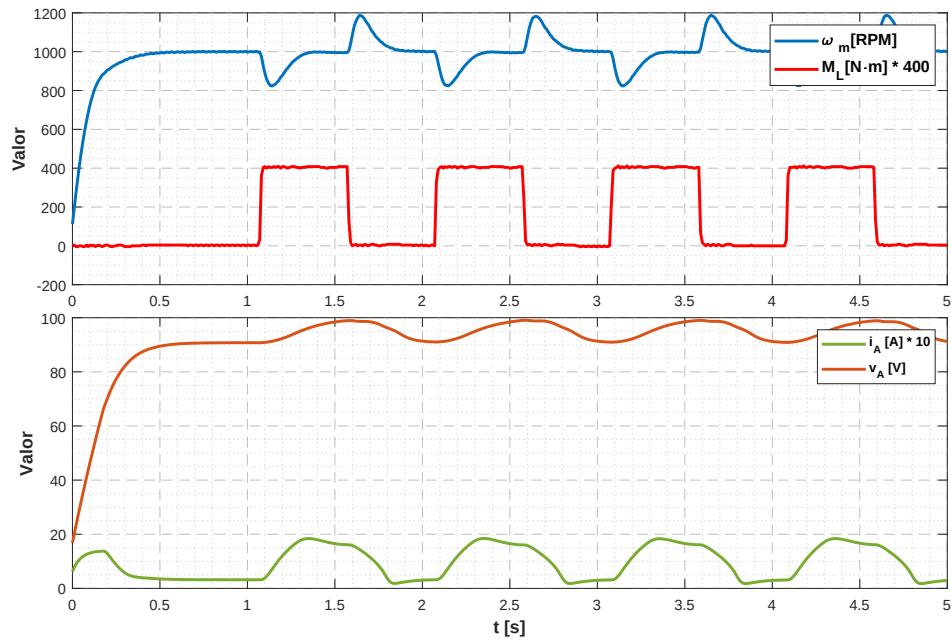
Prueba	$B_{LQR-RICC}$	$B_{LQR-MRL}$	Mejor	Peor
1	2	6	LQR-MRL	LQR-RICC
2	1	7	LQR-MRL	LQR-RICC
Promedio	1.5	6.5	LQR-MRL	LQR-RICC

La Tabla 5.26 presenta un resumen del puntaje de Borda de los controladores LQR (ecuación (5.26–5.29)) en pruebas de rampas de velocidad. En promedio, el controlador LQR-MRL muestra el mejor desempeño en comparación con LQR-RICC.

5.4.3. Pruebas y resultados de perturbaciones de carga

5.4.3.1. Perturbaciones de carga: Prueba 1

Figura 5.56: Respuesta del controlador LQR-RICC (algoritmo de Hewer)



La prueba 1 consiste en evaluar los controladores frente a un perfil de carga M_L que oscila entre $0 \text{ N} \cdot \text{m}$ y $1 \text{ N} \cdot \text{m}$, con un periodo de $T = 1 \text{ s}$ (Tabla 5.3). Las métricas de evaluación del comportamiento mostrado en las figuras se presentan en las Tablas 5.27, 5.28, así como en la Figura 5.59.

Figura 5.57: Respuesta del controlador LQR-MRL (algoritmo MGPI)

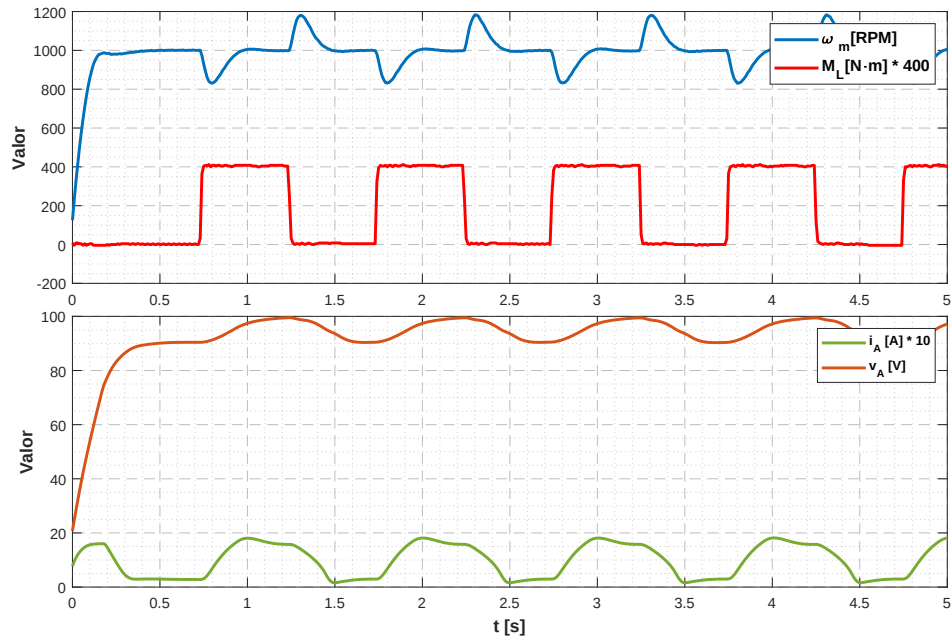
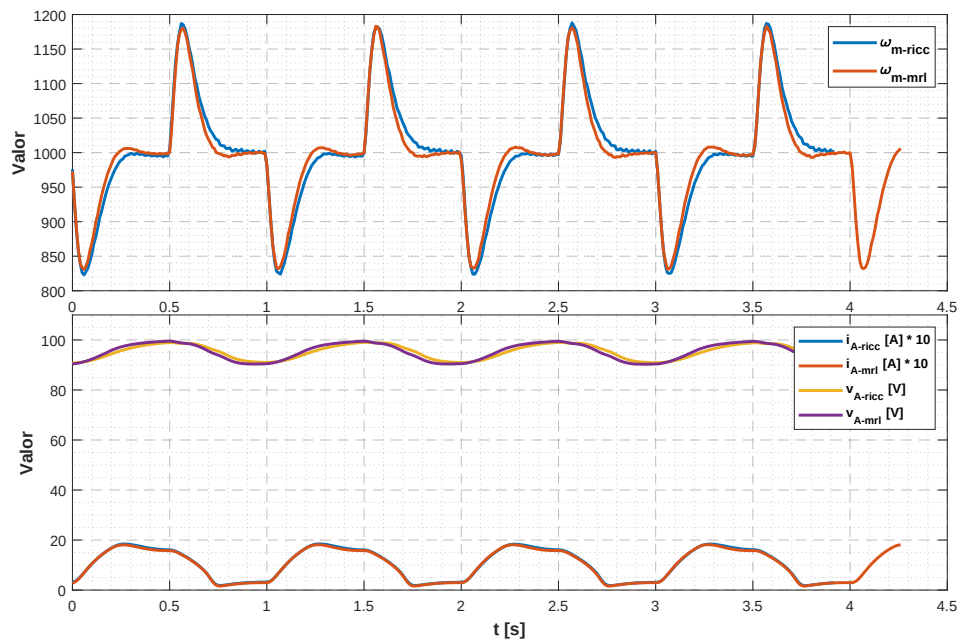


Figura 5.58: Comparación de los controladores LQR - Perturbaciones de carga: prueba 1



Las Figuras 5.56, 5.57 y 5.57 muestran que el controlador LQR-MRL presenta una respuesta rápida y con menor sobreimpulso frente a los cambios repentinos de carga en comparación con LQR-RICC.

Tabla 5.27: Indicadores de desempeño transitorio de los controladores LQR frente a perturbaciones de carga en implementación - Prueba 1

Métrica	LQR-RICC	LQR-MRL	Mejor(1)	Peor(0)
$t_s[s]$	0.2600	0.2100	LQR-MRL	LQR-RICC
$M_p(\%)$	18.067	17.467	LQR-MRL	LQR-RICC

Nota. Estos valores representan el promedio de tres fluctuaciones de carga entre $0 \text{ N} \cdot \text{m}$ y $1 \text{ N} \cdot \text{m}$, con el propósito de evaluar de forma más representativa el comportamiento general de los controladores.

Tabla 5.28: Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores LQR frente a perturbaciones de carga en implementación - Prueba 1

Métrica	LQR-RICC	LQR-MRL	Mejor(1)	Peor(0)
σ_e	61.1791	58.8728	LQR-MRL	LQR-RICC
ISE	1 872 729	1 589 475	LQR-MRL	LQR-RICC
IAE	15 023	12 879	LQR-MRL	LQR-RICC
ITAE	2 057 928	1 749 566	LQR-MRL	LQR-RICC
ITSE	252 404 504	211 487 612	LQR-MRL	LQR-RICC
$v_A^{\text{RMS}} [\text{V}]$	95.2049	95.1666	LQR-MRL	LQR-RICC
$i_A^{\text{RMS}} [\text{A}]$	1.2074	1.1864	LQR-MRL	LQR-RICC
$P [\text{W}]$	114.9457	112.9047	LQR-MRL	LQR-RICC

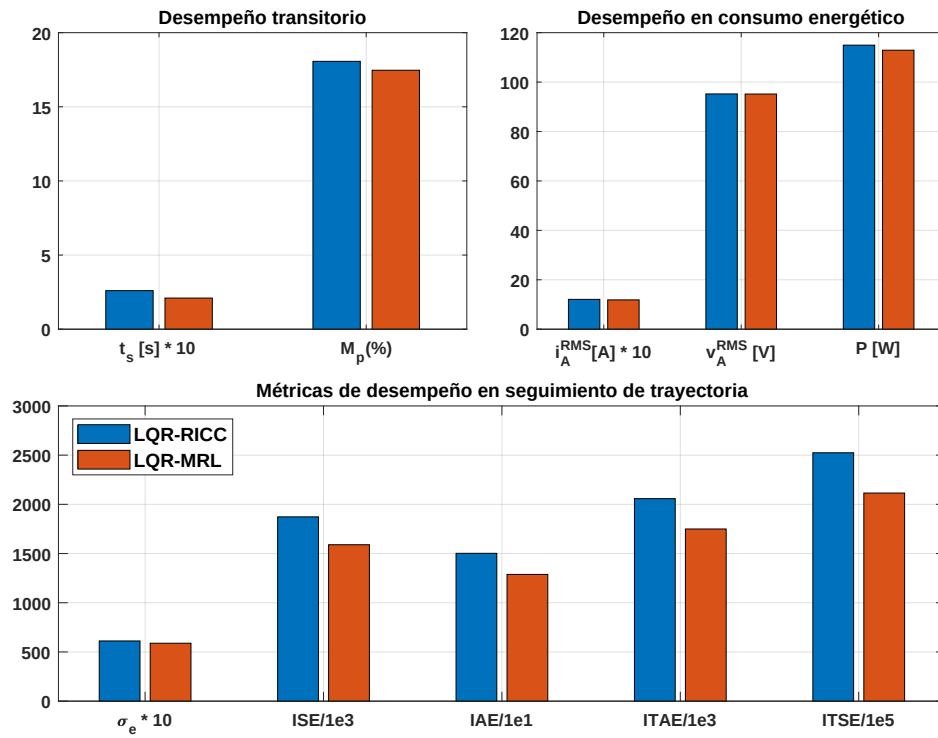
De los resultados numéricos en las Tablas 5.27, 5.28 y en la Figura 5.59, se calcula el puntaje de Borda de cada controlador:

$$B_{\text{LQR-RICC}}^{\text{LT1}} = 0 \times (1) + 10 \times (0) = 0, \quad (5.30)$$

$$B_{\text{LQR-MRL}}^{\text{LT1}} = 10 \times (1) + 0 \times (0) = 10. \quad (5.31)$$

Estos resultados demuestran que el controlador LQR-MRL obtiene el mayor puntaje de Borda en la prueba 1, con 10 puntos, demostrando un mejor rechazo a perturbaciones de carga en comparación con LQR-RICC.

Figura 5.59: Gráfico de barras de las métricas de desempeño de los controladores LQR frente a perturbaciones de carga - Prueba 1



5.4.3.2. Perturbaciones de carga: Prueba 2

Figura 5.60: Respuesta del controlador LQR-RICC (algoritmo de Hewer)

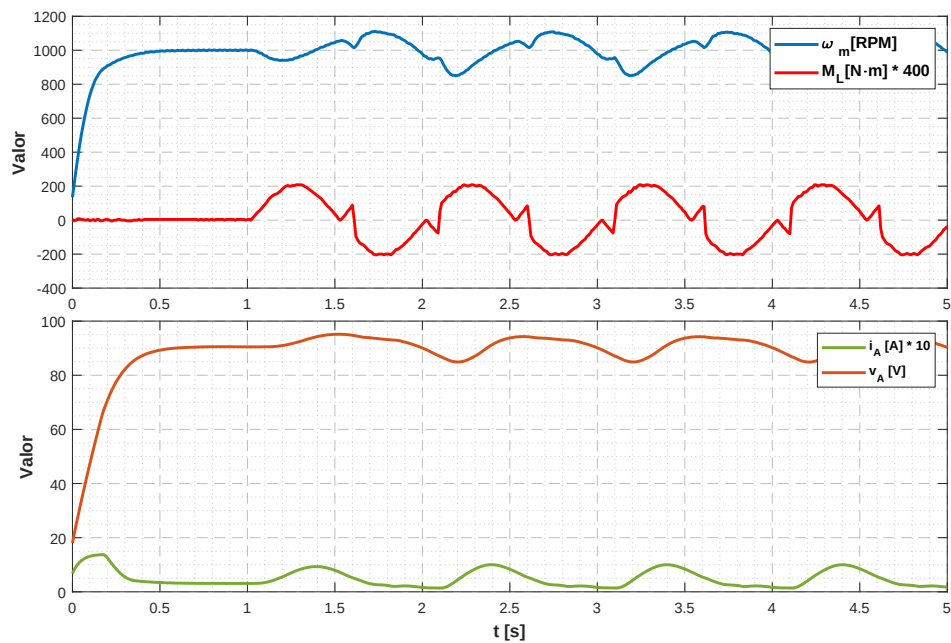


Figura 5.61: Respuesta del controlador LQR-MRL (algoritmo MGPI)

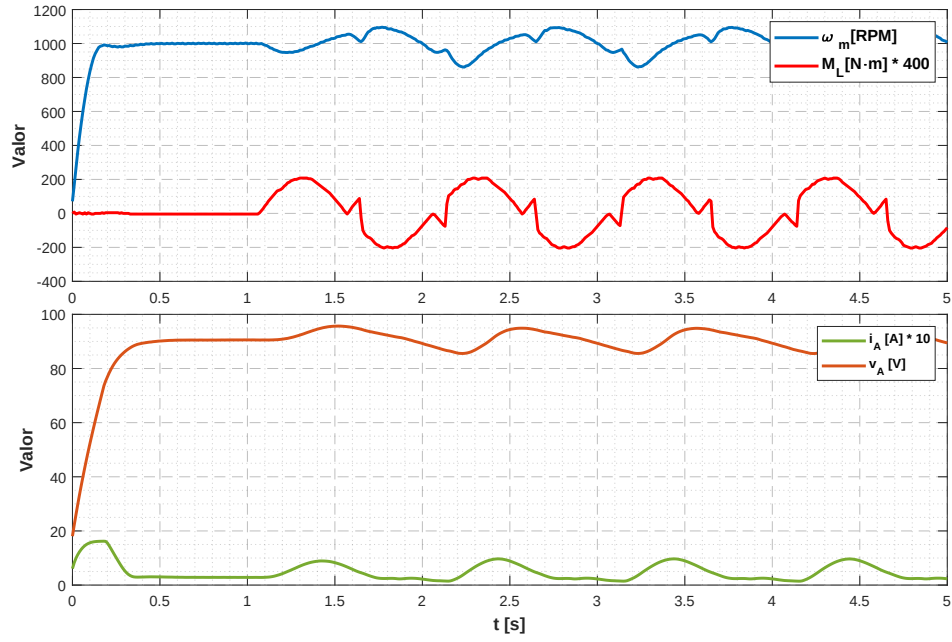
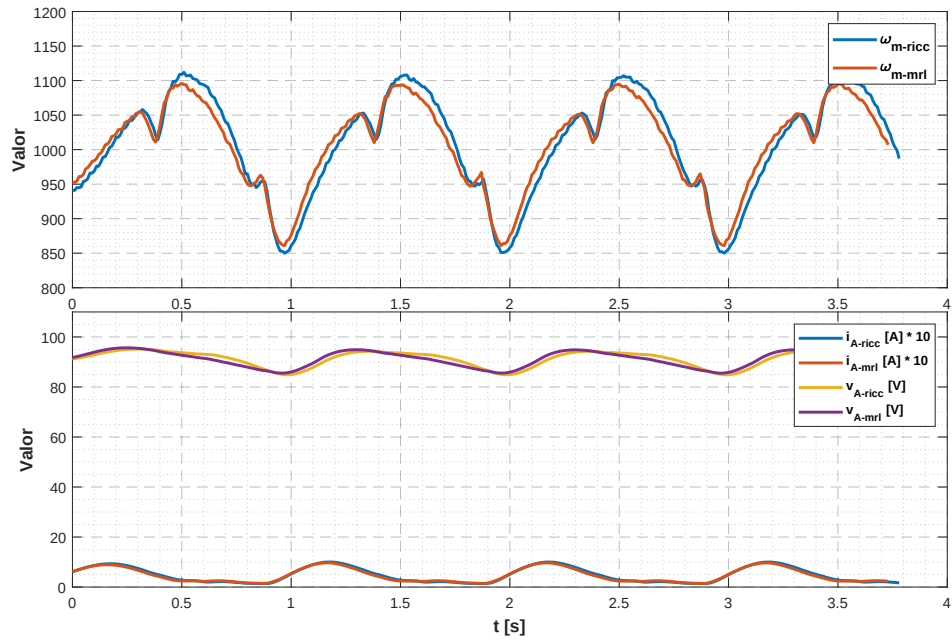


Figura 5.62: Comparación de los controladores LQR - Perturbaciones de carga: prueba 2

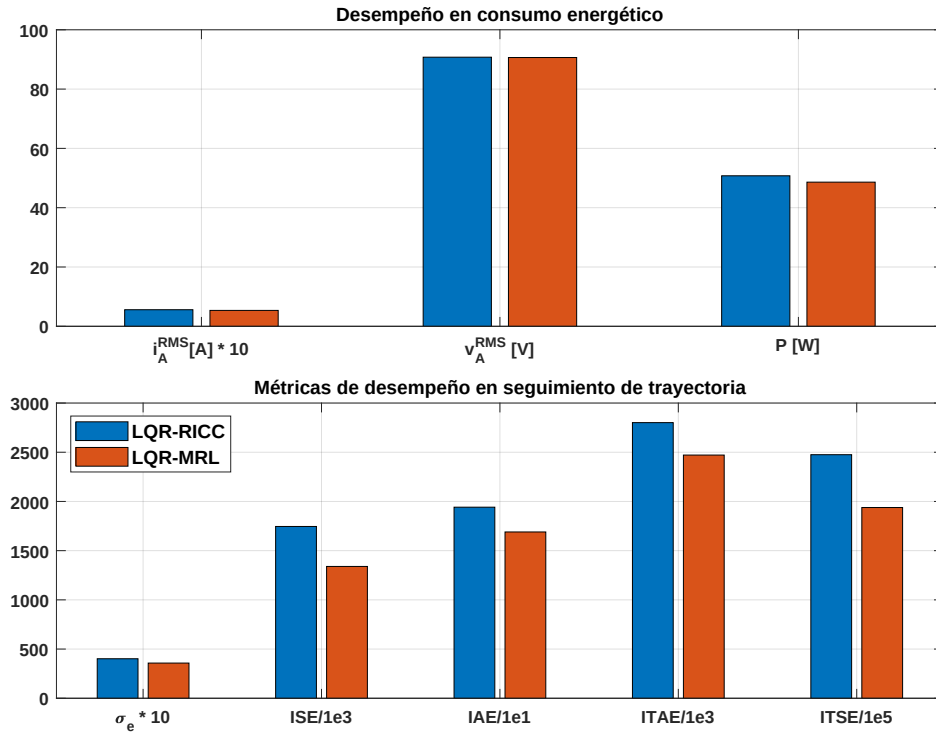


La prueba 2 consiste en evaluar los controladores frente a un perfil de carga M_L sinusoidal con amplitud de $0.5 \text{ N} \cdot \text{m}$ y periodo $T = 1 \text{ s}$ (Tabla 5.3). Las métricas de evaluación del comportamiento mostrado en las figuras anteriores se presentan en la Tabla 5.29, así como en la Figura 5.63.

Tabla 5.29: Indicadores de desempeño en seguimiento de trayectoria y consumo energético de los controladores LQR frente a perturbaciones de carga en implementación - Prueba 2

Métrica	LQR-RICC	LQR-MRL	Mejor(1)	Peor(0)
σ_e	40.1602	35.7686	LQR-MRL	LQR-RICC
ISE	1 745 795	1 339 569	LQR-MRL	LQR-RICC
IAE	19 417	16 899	LQR-MRL	LQR-RICC
ITAE	2 800 968	2 471 407	LQR-MRL	LQR-RICC
ITSE	247 505 876	193 835 145	LQR-MRL	LQR-RICC
v_A^{RMS} [V]	90.7569	90.6462	LQR-MRL	LQR-RICC
i_A^{RMS} [A]	0.5593	0.5364	LQR-MRL	LQR-RICC
P [W]	50.7601	48.6186	LQR-MRL	LQR-RICC

Figura 5.63: Gráfico de barras de las métricas de desempeño de los controladores LQR frente a perturbaciones de carga - Prueba 2



De los resultados numéricos en las Tablas 5.29, 5.29 y en la Figura 5.63, se calcula el puntaje de Borda de cada controlador:

$$B_{LQR-RICC}^{LT1} = 0 \times (1) + 8 \times (0) = 0, \quad (5.32)$$

$$B_{LQR-MRL}^{LT1} = 8 \times (1) + 0 \times (0) = 8. \quad (5.33)$$

Estos resultados demuestran que el controlador LQR-MRL obtiene el mayor puntaje de Borda en la prueba 2, con 10 puntos, demostrando un mejor rechazo a perturbaciones de carga en comparación con LQR-RICC.

5.4.3.3. Resumen de desempeño - Pruebas de perturbaciones de carga

Tabla 5.30: Resumen del puntaje de Borda de los controladores LQR en pruebas de perturbaciones de carga

Prueba	$B_{\text{LQR-RICC}}$	$B_{\text{LQR-MRL}}$	Mejor	Peor
1	0	10	LQR-MRL	LQR-RICC
2	0	8	LQR-MRL	LQR-RICC
Promedio	0.0	9.0	LQR-MRL	LQR-RICC

La Tabla 5.30 presenta un resumen del puntaje de Borda de los controladores LQR (ecuación (5.30–5.33)) en pruebas de perturbaciones de carga. En promedio, el controlador LQR-MRL muestra el mejor desempeño en comparación con el controlador LQR-RICC.

5.4.4. Resumen general del desempeño de los controladores LQR

Tomando en cuenta el rendimiento de los controladores LQR-RICC y LQR-MRL en las Tablas 5.19, 5.20, 5.21, 5.22, 5.24, 5.25, 5.27, 5.28 y 5.29 se presenta un resumen general promedio de los indicadores de desempeño transitorio y de seguimiento de trayectoria en las Tablas 5.31 y 5.32.

Tabla 5.31: Indicadores de desempeño transitorio promedio de los controladores LQR

Métrica	LQR-RICC	LQR-MRL	Mejor(1)	Peor(0)
$t_s[s]$	0.5836	0.5479	LQR-MRL	LQR-RICC
$M_p(\%)$	3.3273	3.5737	LQR-RICC	LQR-MRL

Basado en el criterio de conteo de Borda se concluye a partir de las Tablas 5.31, 5.32 y 5.33, que, en promedio, el controlador LQR-MRL demuestra un mejor desempeño frente a las pruebas a las que fueron sometidos estos controladores (LQR-RICC y LQR-MRL).

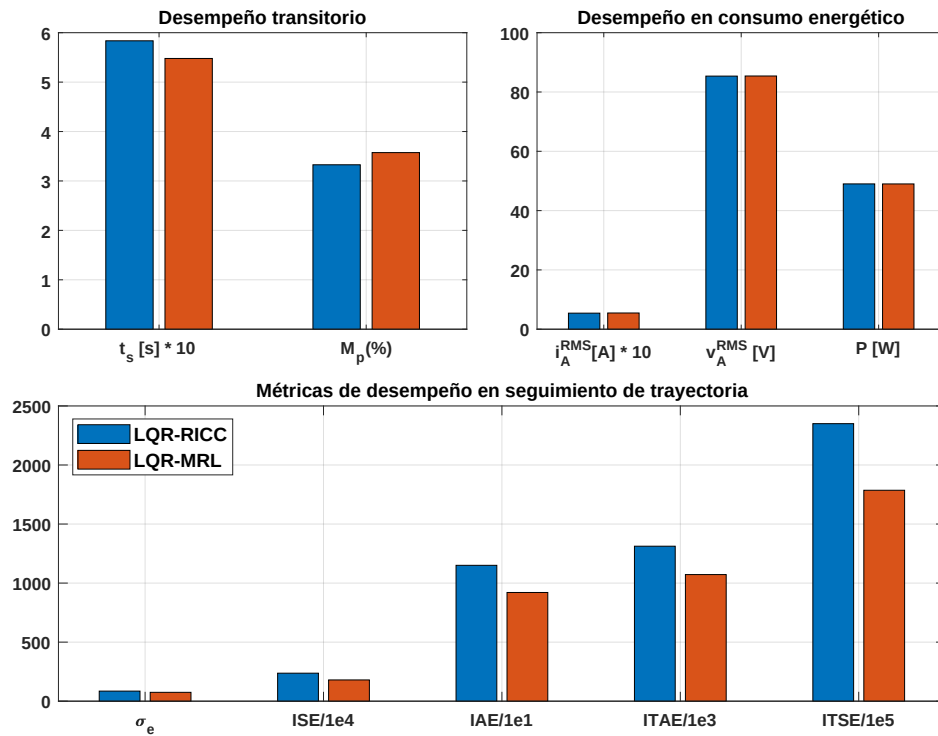
Tabla 5.32: Indicadores de desempeño promedio en seguimiento de trayectoria y consumo energético de los controladores LQR

Métrica	LQR-RICC	LQR-MRL	Mejor(1)	Peor(0)
σ_e	85.29373333	74.9262	LQR-MRL	LQR-RICC
ISE	2 370 735.5	1 798 611.8	LQR-MRL	LQR-RICC
IAE	11 504.5	9204.5	LQR-MRL	LQR-RICC
ITAE	1 312 337.3	1 072 106.833	LQR-MRL	LQR-RICC
ITSE	235 002 082.7	178 608 207.2	LQR-MRL	LQR-RICC
v_A^{RMS} [V]	85.3598	85.3966	LQR-RICC	LQR-MRL
i_A^{RMS} [A]	0.5420	0.5473	LQR-RICC	LQR-MRL
P [W]	49.0047	48.9940	LQR-MRL	LQR-RICC

Tabla 5.33: Puntaje de Borda basado en las métricas de desempeño promedio de los controladores LQR

	$B_{LQR-RICC}$	$B_{LQR-MRL}$	Mejor	Peor
Puntaje	3	7	LQR-MRL	LQR-RICC

Figura 5.64: Métricas de desempeño promedio de los controladores LQR



Nota. Este gráfico de barras resume el desempeño general frente a las seis pruebas a las cuales han estado sujetas los controladores.

En particular, LQR-MRL presenta una respuesta rápida y estable en la variable velocidad ω_m en comparación con LQR-RICC. Asimismo, logra un mejor desempeño promedio en seguimiento de trayectoria, con un consumo energético prácticamente equivalente al de LQR-RICC, lo que evidencia su efectividad y eficiencia en el control de velocidad de un motor DC con devanado de campo. Estos resultados respaldan el uso de enfoques de aprendizaje por refuerzo basados en modelo para este tipo de aplicaciones.

5.5. Discusión de resultados

De las Tablas 5.16 y 5.17, se realiza una comparación porcentual del desempeño promedio del controlador PI-rlNL respecto a los controladores PI-OS y PI-ZN para cada métrica, cuyos resultados se presentan en la Tabla 5.34.

Tabla 5.34: Mejoras porcentuales del controlador PI-rlNL vs. PI-OS y PI-ZN

Métrica	PI-rlNL vs. PI-OS	PI-rlNL vs. PI-ZN
$t_s[s]$	4.3 % (más rápido)	15.0 % (más rápido)
$M_p(\%)$	65.2 % (menor)	72.5 % (menor)
σ_e	19.5 % (menor)	10.0 % (menor)
ISE	36.8 % (menor)	21.0 % (menor)
IAE	17.6 % (menor)	12.4 % (menor)
ITAE	27.1 % (menor)	24.9 % (menor)
ITSE	36.5 % (menor)	24.4 % (menor)
$v_A^{\text{RMS}} [V]$	1.7 % (menor)	2.2 % (menor)
$i_A^{\text{RMS}} [A]$	3.8 % (menor)	3.0 % (menor)
$P [W]$	5.1 % (más eficiente)	3.6 % (más eficiente)

Al comparar el controlador PI-rlNL con los controladores PI-OS y PI-ZN, se observan mejoras sustanciales en la mayoría de las métricas. El sobreimpulso $M_p(\%)$ se reduce en un 65 % y 72 %, respectivamente, mientras que el tiempo de establecimiento t_s disminuye en un 4.3 % y 15 %. En términos de seguimiento, el PI-rlNL presenta menores valores de ISE e ITAE en comparación con ambos controladores. En particular, el ISE se reduce en aproximadamente un 36.8 % respecto a PI-OS y un 21.0 % respecto a PI-ZN, mientras que el ITAE disminuye en un 27.1 % y 24.9 %, respectivamente. Resultados similares se

observan en otras métricas de seguimiento (véase la Tabla 5.34).

Las métricas asociadas al consumo energético también favorecen al controlador propuesto: v_{A-RMS} e i_{A-RMS} presentan valores porcentuales ligeramente menores en comparación con los otros controladores, lo que se traduce en una reducción del consumo de potencia P del orden de 3.6 % a 5.1 %. Estas disminuciones indican que PI-rlNL logra una respuesta rápida y suave con un menor esfuerzo eléctrico.

La reducción del sobreimpulso es crítica en el control de motores, ya que permite evitar torques excesivos y estrés mecánico. El término no lineal incorporado en la acción integral de PI-rlNL actúa como una ganancia adaptativa, limitando la acumulación excesiva de la acción integral cuando el error es grande. Aunque el tiempo de establecimiento t_s mejora solo en unas pocas unidades porcentuales ($\sim 4\%$), esta mejora no compromete el sobreimpulso, lo que contribuye a suavizar la respuesta y reducir el sobreimpulso.

Estos resultados son consistentes con estudios reportados en la literatura en los cuales se basa este trabajo. Por ejemplo, Tufenkci et al. (2023) muestra que un controlador PI sintonizado mediante TD3 mejora en el orden de $\sim 0.03\%$ en métricas asociadas al esfuerzo de la señal de control, en comparación con controladores PI ajustados mediante métodos metaheurísticos como DDPG, GA, PSO, GWO y DE. En dicho estudio, esta métrica se evalúa a través del promedio del valor absoluto de la señal de control (AAC¹). De manera similar, Alejandro-Sanjines et al. (2023) reporta que este enfoque de sintonización logra una reducción de aproximadamente $\sim 41\%$ en el tiempo de respuesta y una mejora cercana al $\sim 42\%$ en el seguimiento de trayectoria, empleando una función de costo LQG, en comparación con métodos de sintonización manual asistidos por herramientas como Sisotool de MATLAB. No obstante, cabe destacar que dichos estudios han sido desarrollados en entornos de simulación y no consideran las limitaciones inherentes a implementaciones reales, a las cuales sí están sujetos los controladores diseñados en este trabajo.

¹Del inglés *Absolute Control Signal*.

A partir de las Tablas 5.31 y 5.32, se realiza una comparación porcentual del desempeño promedio del controlador LQR-MRL respecto al controlador LQR-RICC para cada métrica, cuyos resultados se presentan en la Tabla 5.35.

Tabla 5.35: Mejoras porcentuales del controlador LQR-MRL vs. LQR-RICC

Métrica	LQR-MRL vs. LQR-RICC
$t_s[s]$	6.1 % (más rápido)
$M_p(\%)$	7.4 % (mayor)
σ_e	12.2 % (menor)
ISE	24.1 % (menor)
IAE	20.0 % (menor)
ITAE	18.3 % (menor)
ITSE	24.0 % (menor)
$v_A^{\text{RMS}} [\text{V}]$	0.04 % (mayor)
$i_A^{\text{RMS}} [\text{A}]$	0.99 % (mayor)
$P [\text{W}]$	0.02 % (más eficiente)

Nota. Los valores que juegan en contra al desempeño del controlador se resaltan en negrita.

Al comparar el controlador LQR-MLR con el controlador LQR-RICC, se observan mejoras sustanciales en la mayoría de las métricas. El tiempo de establecimiento t_s disminuye en un 6.1 % respecto a LQR-RICC. En términos de seguimiento, LQR-MRL presenta menores valores de ISE e ITAE. En particular, el ISE se reduce en aproximadamente un 24.1 %, mientras que el ITAE disminuye en un 18.3 %. Resultados similares se observan en otras métricas de seguimiento (véase la Tabla 5.35).

Las métricas asociadas al consumo energético muestran un desempeño comparable entre los controladores. Tanto v_{A-RMS} como i_{A-RMS} del LQR-MRL son en promedio ligeramente mayores (del orden de $\sim 0.04\%$ y $\sim 0.99\%$, respectivamente) respecto del LQR-RICC. La potencia P promedio exhibe una reducción del orden de $\sim 0.02\%$. Estas variaciones, de magnitud reducida, indican que LQR-MRL logra una respuesta rápida con un consumo energético prácticamente equivalente al del controlador LQR-RICC. Esto se considera así, ya que la exactitud de los equipos de medición de Lucas Nülle son en general del 2 % (Lucas-Nuelle, 2026); encontrándose los resultados obtenidos dentro de este rango de tolerancia.

Aunque el sobreimpulso es mayor usando LQR-MLR, este se mantiene en un valor promedio de 3.5737 %, frente a 3.3273 % del LQR-RICC (véase la Tabla 5.32). Asimismo, aunque el tiempo de establecimiento t_s mejora solo en unas pocas unidades porcentuales ($\sim 6\%$), dicha mejora es suficiente para lograr un seguimiento de trayectoria eficiente.

Estos resultados son consistentes con estudios reportados en la literatura en los cuales se basa este trabajo. Por ejemplo, Li et al. (2024) muestra que un controlador LQR sintonizado mediante un algoritmo basado en Policy Iteration (RL-PI) exhibe un comportamiento estable y robusto. Por otro lado, Kuczmam (2024) corrobora que, en pruebas de seguimiento de rampas de velocidad, los controladores basados en espacio de estados, como el LQR, puede presentar mayores errores de seguimiento. Asimismo, las mejoras en el índice IAE del 20.0 % obtenidas en este trabajo, son comparables con las reportadas por Maghfiroh et al. (2022), quien informa una reducción del IAE del 11.28 %, con un incremento en consumo energético de solo 1 % de un controlador LQR basado en métodos de optimización metaheurísticos, como PSO, o algoritmos avanzados como el GPI, respecto de un controlador basado en la solución de la ecuación algebraica de Riccati (ARE).

CONCLUSIONES

1. Se diseñaron e implementaron controladores proporcional integral (PI) y regulador cuadrático lineal (LQR) mediante técnicas de aprendizaje por refuerzo, aplicados al control de velocidad de un motor de corriente continua con devanado de campo. En el caso de los controladores PI, se diseñaron tres variantes: el PI basado en el criterio del óptimo simétrico (PI-OS), el PI basado en Ziegler-Nichols (PI-ZN) y el PI basado en TD3 con término no lineal (PI-rlNL). Por otro lado, se diseñaron dos controladores LQR: el LQR basado en el algoritmo de Hwer para la solución de la ecuación de Riccati (LQR-RICC) y el LQR basado en el algoritmo MGPI (LQR-MRL). Los resultados experimentales frente a pruebas de cambios de referencia, rampas de velocidad (aceleración y frenado) y perturbaciones de carga, evidencian que los controladores basados en aprendizaje por refuerzo (PI-rlNL y LQR-MRL) superan, en promedio, a sus contrapartes clásicas (PI-OS, PI-ZN y LQR-RICC).
2. Se identificaron los algoritmos de aprendizaje por refuerzo en función del tipo de controlador considerado. En la literatura, se observa que la sintonización de controladores PI suele abordarse mediante algoritmos basados en TD3. Por otro lado, el ajuste de controladores basados en espacio de estados se fundamenta en métodos iterativos de aprendizaje por refuerzo, tales como GPI y MGPI. Todos los controladores fueron diseñados y sintonizados utilizando herramientas de software como MATLAB y Simulink, con base en la literatura técnico-científica. La sintonización de los controladores PI se realizan sobre el modelo no lineal del motor DC. En con-

traste, los controladores LQR se diseñan y sintonizan empleando el modelo lineal del motor DC con devanado de campo.

3. Los controladores se desplegaron en los equipos del laboratorio de Electrónica de Potencia de la Escuela Profesional de Ingeniería Electrónica, demostrando un correcto funcionamiento. Las rutinas de control se implementan en Simulink r2016a, el cual cuenta con el Toolbox CO3636-3D de Lucas Nülle, y permite cargar el firmware generado directamente en el módulo convertidor de potencia CO3636-3B. Asimismo, en dichas rutinas se configuraron los límites de operación del convertidor; un límite de 240 V DC para el voltaje y de 3 A DC para la corriente. Garantizando el funcionamiento seguro y a prueba de posibles fallas del motor u otros componentes.
4. Se evaluaron y compararon los controladores PI y LQR basados en algoritmos de aprendizaje por refuerzo con sus versiones sintonizadas mediante métodos clásicos, empleando métricas de desempeño transitorio (M_p y t_s), de seguimiento (σ_e , ISE, IAE, ITAE y ITSE) y de consumo energético (i_{A-RMS} , v_{A-RMS} y P). El controlador PI-rlNL presenta mejoras sustanciales respecto a PI-OS y PI-ZN, mostrando una respuesta más rápida y suave, con reducciones promedio del 65 % y 72 % en el sobreimpulso M_p (%), y una disminución del 4.3 % y 15 % en el tiempo de establecimiento t_s . Asimismo, exhibe un mejor desempeño promedio en la mayoría de métricas de seguimiento (σ_e , ISE, IAE, ITAE, ITSE), junto con una reducción promedio del consumo de potencia entre 3.6 % y 5.1 %. Por su parte, LQR-MRL es ligeramente inferior en términos de sobreimpulso M_p (%), exhibiendo un incremento promedio de 7.4 % respecto a LQR-RICC. No obstante, LQR-MRL exhibe mejoras en la mayoría de métricas de desempeño. En particular, logra una reducción en el tiempo de establecimiento t_s del orden de ~ 6 %. Además, este controlador presenta, en promedio, mejores resultados en la mayoría de métricas de seguimiento, junto con una reducción marginal de la potencia del orden de ~ 0.02 %. Estos resultados indican que LQR-MRL ofrece una respuesta rápida con un consumo energético prácticamente igual o equivalente al de LQR-RICC.

RECOMENDACIONES

- Durante el proceso de sintonización de controladores PI mediante el algoritmo TD3, se recomienda disponer de herramientas de cómputo con capacidad suficiente para soportar el entrenamiento de redes neuronales profundas. En especial, resulta conveniente contar con un procesador moderno, una memoria RAM de al menos 16 GB y una tarjeta gráfica compatible con herramientas como MATLAB o Python.
- El proceso de entrenamiento mediante el algoritmo TD3 demanda elevados recursos computacionales y tiempo de ejecución. En este trabajo el entrenamiento de aproximadamente 400 iteraciones tomó alrededor de 03:30 horas. Por tanto, se recomienda investigar métodos que permitan optimizar el tiempo de entrenamiento de los agentes de aprendizaje por refuerzo.
- La sintonización de los controladores LQR se basa en algoritmos iterativos de política. No obstante, es recomendable explorar y adaptar algoritmos como TD3 a este tipo de problemas, que en lo particular comparten fundamentos teóricos claves, de tal manera que se aproveche de mejor forma los beneficios tanto de los controladores basados en espacio de estados y arquitecturas sofisticadas de búsqueda de parámetros de control óptimos.
- Las investigaciones de este tipo de técnicas muchas veces se reducen a entornos de simulación. Teóricamente, estos algoritmos tienen implicaciones prometedoras; no obstante, su implementación en tiempo real se ve limitada tanto por los sistemas

físicos mismos, capaces de sufrir daños, como por las limitaciones de la capacidad de cómputo de sistemas embebidos pequeños (Brunton, 2021). Por lo tanto, se recomienda investigar la implementación de estos algoritmos en tiempo real, teniendo en cuenta que los parámetros encontrados mediante simulación no siempre resultan estables en el sistema físico, debido a que dependen fuertemente del modelo matemático.

REFERENCIAS BIBLIOGRÁFICAS

- Afzali, S., Shoaran, M., & Karimian, G. (2023). A modified convergence ddpg algorithm for robotic manipulation. *Neural Processing Letters*, 55:1–16.
- Akhare, Y. P., Ali, W. H., Fuller, J. H., & Attia, J. O. (2021). Power electronics and drives: Simulation to implementation in minutes: A novel implementation tool for power engineers. *IEEE Power Electronics Magazine*, 8(2):44–52.
- Al-Haddad, S. A. R. (2018). Decoupled integral lqr controller with anti-windup compensator for mimo two rotor aerodynamical system (tras).
- Alejandro-Sanjines, U., Maisincho-Jivaja, A., Asanza, V., Lorente-Leyva, L. L., & Peluffo-Ordóñez, D. H. (2023). Adaptive pi controller based on a reinforcement learning algorithm for speed control of a dc motor. *Biomimetics*, 8(5).
- Anton, M.-E. (2024). A comparative fem study between permanent magnet and wound field synchronous machines. *Bulletin of the Polytechnic Institute of Iasi. Electrical Engineering, Power Engineering, Electronics Section*, 70(1):75–85.
- Åström, K. J. & Hägglund, T. (2009). *Control PID avanzado*. Pearson, Madrid.
- Åström, K. J. & Kumar, P. (2014). Control: A perspective. *Automatica*, 50(1):3–43.
- Bankovic, B., Petronijevic, M., Mitrovic, N., Kostic, V., & Filipovic, F. (2021). Analytical magnitude optimum criterion using the rotor field oriented (rfo) control algorithm. *Journal of E - Technology*, 12(3).
- Barto, A. & Sutton, R. S. (2018). *Reinforcement Learning: An Introduction*. The MIT Press.
- Beckett, J., Hysell, C., Jones, M., Hatipoglu, K., & Davari, A. (2020). Dynamic dc motor excitation controller. In *2020 SoutheastCon*, volume 2, pages 1–5.
- Bermúdez, M., Barrero, F., Martín, C., & Perales, M. (2021). Performance analysis of direct torque controllers in five-phase electrical drives. *Applied Sciences*, 11(24).
- Bharti, J., Phadke, G., & Patil, D. (2020). Optimization of dc motor speed control using lqr technique. In *ICDSMLA 2019: Proceedings of the 1st International Conference on Data Science, Machine Learning and Applications*, pages 1492–1499. Springer.

-
- Bose, B. K. (2002). *Modern power electronics and ac drives*. Pretince Hall.
- Brokaw, J. (s.f.). *Using PID Motion Control in Robotics*. Recuperado el 05 de Marzo, 2025, de <https://www.valin.com/resources/articles/using-pid-motion-control-robotics>.
- Brunton, S. (2017). *Linear Quadratic Regulator (LQR) Control for the Inverted Pendulum on a Cart [Control Bootcamp]*. [Video]. YouTube. https://youtu.be/1_UobILf3cc.
- Brunton, S. L. (2021). *Deep Reinforcement Learning: Neural Networks for Learning Control Laws*. [Video]. YouTube. https://youtu.be/IUiKAD6cuTA?si=dFr5zJG1d281AbG_.
- Campos-Salazar, J. M., Lecaros, P., & Sandoval-García, R. (2024). Dynamic analysis and comparison of the performance of linear and nonlinear controllers applied to a nonlinear non-interactive and interactive process. *AIMS Electronics and Electrical Engineering*, 8(4):441–465.
- Çelik, D., Khosravi, N., Khan, M. A., Waseem, M., & Ahmed, H. (2025). Advancements in nonlinear pid controllers: A comprehensive review. *Computers and Electrical Engineering*, 129:110775.
- Çelik, E., Bal, G., Öztürk, N., Bekiroglu, E., Houssein, E. H., Ocak, C., & Sharma, G. (2024). Improving speed control characteristics of pmc motor drives using nonlinear pi control. *Neural Computing and Applications*, 36(16):9113–9124.
- Chapman, S. (2012). *Máquinas eléctricas*. Mc Graw Hill.
- Chau, K. (2015). *Electric Vehicle Machines and Drives: Design, Analysis and Application*. Wiley.
- Chen, C.-W., Wu, H.-M., & Nian, C.-Y. (2025). A class of anti-windup controllers for precise positioning of an x-y platform with input saturations. *Electronics*, 14(3).
- Chirme-Sisa, D., Utrilla Mego, L. W., Herrera-Levano, J. C., & Coaquira-Castillo, R. J. (2024). Modeling, simulation, design and comparative analysis of the pid and lqr controllers for an inverted pendulum. In *2024 IEEE XXXI International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, pages 1–8.
- Cho, M., Abdallah, A., & Rasouli, M. (2023). Low-rank lqr optimal control design over wireless communication networks. *arXiv preprint*. <https://arxiv.org/abs/2301.13729>.
- Cusihualpa-Huamantupa, K., Sacoto-Cabrera, E. J., Coaquira-Castillo, R. J., Utrilla Mego, L. W., Herrera-Levano, J. C., Concha-Ramos, Y., & Moreno-Cardenas, E. (2026). Deep reinforcement learning-based intelligent water level control: From simulation to embedded implementation. *Sensors*, 26(1).
- Dong, R., Du, J., Liu, Y., Heidari, A. A., & Chen, H. (2023). An enhanced deep deterministic policy gradient algorithm for intelligent control of robotic arms. *Frontiers in Neuroinformatics*, 17:1096053.
- EEEGUIDE (s.f.). *Symmetrical Optimum*. Recuperado el 17 de Noviembre, 2025, de <https://www.eeeguide.com/symmetrical-optimum/>.

-
- Encyclopedia Britannica (s.f.). *Operant Conditioning*. Recuperado el 13 de Julio, 2025, de <https://www.britannica.com/science/operant-conditioning>.
- Eviningsih, R. P., Nugraha, A. T., & Sobhita, R. A. (2025). Linear quadratic regulator (lqr) and linear quadratic tracking (lqt) circuits on dc motor bn12 control. *Sustainable Energy Control and Optimization*, 1(1):10–19.
- Filipović, V., Matijević, M., & Cuković, S. (2024). Design of lq regulators with prescribed degree of stability for dual-rate systems based on reinforcement learning. *Measurement and Control*.
- Fujimoto, S., van Hoof, H., & Meger, D. (2018). Addressing function approximation error in actor-critic methods. In Dy, J. & Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1587–1596. PMLR.
- Garpinger, O., Hägglund, T., & Åström, K. J. (2014). Performance and robustness trade-offs in pid control. *Journal of Process Control*, 24(5):568–577.
- Gharbi, A., Ayari, M., Albalawi, N., El Touati, Y., & Klai, Z. (2025). A comparative analysis of fairness and satisfaction in multi-agent resource allocation: Integrating borda count and k-means approaches with distributive justice principles. *Mathematics*, 13(15).
- Hägglund, T. & Guzmán, J. L. (2024). Give us pid controllers and we can control the world**this work has been financed by the following projects: Pid2020–112709rb-c21 project financed by the spanish ministry of science. *IFAC-PapersOnLine*, 58(7):103–108. 4th IFAC Conference on Advances in Proportional-Integral-Derivate Control PID 2024.
- Hughes, A. & Drury, B. (2019). *Electric motors and drives: fundamentals, types and applications*. Newnes.
- Husnaini, I., Krismadinata, K., et al. (2020). Pi and pid controller design and analysis for dc shunt motor speed control. *PI and PID Controller Design and Analysis for DC Shunt Motor Speed Control*.
- Hwang, D. & Gu, B.-G. (2020). Field current control strategy for wound-rotor synchronous motors considering coupled stator flux linkage. *IEEE Access*, 8:111811–111821.
- Jeong, M., Fuchs, S., & Biela, J. (2020). High performance lqr control of modular multilevel converters with simple control structure and implementation. In *2020 22nd European Conference on Power Electronics and Applications (EPE'20 ECCE Europe)*, pages P.1 – P.10, Piscataway, NJ. IEEE. 22nd European Conference on Power Electronics and Applications (EPE'20 ECCE Europe) (virtual); Conference Location: Lyon, France; Conference Date: September 7–11, 2020; Due to the Coronavirus (COVID-19) the conference was conducted virtually.
- Khellaf, L., Djellal, A., & Mayache, H. (2025). Metaheuristic-based optimization of non-linear pi controllers for maximum power extraction in pv systems. *Journal Européen des Systèmes Automatisés*, 58(8):1561.
- Kuczmam, M. (2024). Review of dc motor modeling and linear control: Theory with laboratory tests. *Electronics*, 13(11).

-
- Lakhani, A. I., Chowdhury, M. A., & Lu, Q. (2022). Stability-preserving automatic tuning of pid control with reinforcement learning. *Complex Engineering Systems*, 2(1).
- Lee, J. & Sutton, R. S. (2021). Policy iterations for reinforcement learning problems in continuous time and space — fundamental theory and methods. *Automatica*, 126:109421.
- Lewis, F. L., Vrabie, D., & Syrmos, V. L. (2012). *Optimal control*. John Wiley & Sons.
- Li, X., Xue, L., & Sun, C. (2018). Linear quadratic tracking control of unknown discrete-time systems using value iteration algorithm. *Neurocomputing*, 314:86–93.
- Li, Y., Xu, D., Huang, J., & Ma, Y. (2024). Robust adaptive optimal control of a dc motor system based on policy iteration. In *2024 IEEE 13th Data Driven Control and Learning Systems Conference (DDCLS)*, pages 577–583.
- Liceaga-Castro, J. U., Siller-Alcalá, I. I., González-San Román, J. D., & Alcántara-Ramírez, R. A. (2022). Pi speed control with reverse motion of a series dc motor based on the noise reduction disturbance observer. *Actuators*, 11(5).
- Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv e-prints*, page arXiv:1509.02971.
- Linnertz, R. (2016). *DC drives using Matlab Simulink (Course No. SO2800-2U, Version 2.0.0.1)*. Lucas Nülle GmbH. [LabSoft course manual, Version 7.43]. <https://www.lucas-nuelle.us/>.
- Liu, S. (2024). An evaluation of ddpq, td3, sac, and ppo: Deep reinforcement learning algorithms for controlling continuous system. In *Proceedings of the 2023 International Conference on Data Science, Advanced Algorithm and Intelligent Computing (DAI 2023)*, pages 15–24. Atlantis Press.
- Lucas-Nuelle (2023). *DC Multi-Circuit, Compound-Wound Machine, 0.3 kW*. [Equipment technical documentation]. Recuperado el 8, de Abril, 2025 de <https://www.lucas-nuelle.us/2776/pid/22625/apg/11289/DC-multi-circuit,-compound-wound-machine,-03-kW.htm>.
- Lucas-Nuelle (2026). *Power-Multimeter*. [Equipment technical documentation]. Recuperado el 06, de Mayo, 2026 de <https://www.lucas-nuelle.us/2769/pid/37277/apg/19923/Power-Multimeter.htm>.
- Maghfiroh, H., Nizam, M., Anwar, M., & Ma'Arif, A. (2022). Improved lqr control using pso optimization and kalman filter estimator. *IEEE Access*, 10:18330–18337.
- Mamani, K. M. S., Camacho, O., & Prado, A. (2024). Deep reinforcement learning via nonlinear model predictive control for thermal process with variable longtime delay. In *2024 IEEE ANDESCON*, pages 1–6.
- Matejová, M. & Paralič, J. (2025). A multi-criteria decision-making approach for the selection of explainable ai methods. *Machine Learning and Knowledge Extraction*, 7(4).
- Meneses Morales, P. (2013). Diseño e implementación de un módulo educativo para el control de sistema bola y varilla. [Tesis de grado]. Pontificia Universidad Católica del Perú.

-
- Nguyen, K. H. N., Rivas, A., Delipei, G. K., & Hou, J. (2024). Reinforcement learning-based control sequence optimization for advanced reactors. *Journal of Nuclear Engineering*, 5(3):209–225.
- Nise, N. S. (2015). *Control Systems Engineering*. Wiley, Hoboken, NJ, 7 edition.
- Ogata, K. (2010). *Ingeniería de control Moderna*. Pearson, Madrid, España, 5 edition.
- Okyere, E., Bousbaine, A., Poyi, G. T., Joseph, A. K., & Andrade, J. M. (2019). Lqr controller design for quad-rotor helicopters. *The Journal of Engineering*, 2019(17):4003–4007.
- OpenAI (2023a). *Deep Deterministic Policy Gradient (DDPG)*. [Spinning Up documentation]. Recuperado el 7 de Abril, 2025, de <https://spinningup.openai.com/en/latest/algorithms/ddpg.html>.
- OpenAI (2023b). *Twin Delayed DDPG (TD3)*. [Spinning Up documentation]. Recuperado el 8 de Abril, 2025, de <https://spinningup.openai.com/en/latest/algorithms/td3.html>.
- Papadopoulos, K. G., Yadav, P. K., & Margaris, N. I. (2017). Explicit analytical tuning rules for digital pid controllers via the magnitude optimum criterion. *ISA Transactions*, 70:357–377.
- Rashid, M. H. (2014). *Power electronics: devices, circuits and application*. Pearson, 4th edition.
- Saisudha, V., Seeja, G., Pillay, R. V., Manikutty, G., & Bhavani, R. (2016). Analysis of speed control of dc motor using lqr method. *International Journal of Control Theory and Applications*, 9(15):7377–7385.
- Sarhadi, P. (2025). On the standard performance criteria for applied control design: Pid, mpc or machine learning controller? *arXiv preprint*. <https://arxiv.org/abs/2503.14379>.
- Satrianata, L. J. (2024). Design and simulation of dc motor control based on lqr and lqt for optimal control system. In *Conference of Electrical, Marine and Its Application*, volume 3, pages 161–168.
- Sevilla-Hidalgo, A., Uscamaita-Quispetupa, R., Herrera-Levano, J., Mego, L. U., & Coaquira-Castillo, R. (2025). Low-speed scalar control of induction motor by fuzzy logic. *IAES International Journal of Artificial Intelligence (IJ-AI)*, 14(6):4623–4635.
- Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014). Deterministic policy gradient algorithms. In Xing, E. P. & Jebara, T., editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 387–395, Beijing, China. PMLR.
- StudySmarter (s.f.). *LQR Control: Applications & Advantages*. Recuperado el 05 de Marzo, 2025, de <https://www.studysmarter.co.uk/explanations/engineering/aerospace-engineering/lqr-control/>.

-
- Stumper, J.-F. (2012). *Flatness-based predictive and optimal control for electrical drives*. [Tesis doctoral]. Technische Universität München.
- Texas Instruments (2021). *InstaSPIN-FOC and InstaSPIN-MOTION User's Guide, Literature Number SPRUHJ1I*. [Published January 2013; revised October 2021]. Recuperado el 14 de Febrero, 2026, de <https://www.ti.com/lit/ug/spruhj1i/spruhj1i.pdf>.
- The MathWorks, Inc. (2025a). *Design discrete linear-quadratic (LQ) regulator for continuous plant*. [MATLAB Documentation]. Recuperado el 22 de Junio, 2026, de <https://it.mathworks.com/help/control/ref/lqrd.html>.
- The MathWorks, Inc. (2025b). *Tune PID Controller to Favor Reference Tracking or Disturbance Rejection (PID Tuner)*. [MATLAB Documentation]. Recuperado el 14 de Febrero, 2026, de <https://www.mathworks.com/help/control/getstart/tune-pid-controller-to-balance-tracking-and-disturbance-rejection.html>.
- The MathWorks, Inc. (2026). *dlqr – Linear-quadratic (LQ) state-feedback regulator for discrete-time state-space system*. [MATLAB Documentation]. Recuperado el 2 de Marzo, 2026, de <https://www.mathworks.com/help/control/getstart/tune-pid-controller-to-balance-tracking-and-disturbance-rejection.html>.
- Traue, A., Book, G., Kirchgässner, W., & Wallscheid, O. (2019). Towards a reinforcement learning environment toolbox for intelligent electric motor control. *arXiv preprint*. <https://arxiv.org/abs/1910.09434>.
- Tüfenkçi, S., Kavuran, G., & Yeroğlu, C. (2023). An Approach for DC Motor Speed Control with Off-Policy Reinforcement Learning Method. *Balkan Journal of Electrical and Computer Engineering*, 11(2):184–189.
- Tufenkci, S., Baykant Alagoz, B., Kavuran, G., Yeroğlu, C., Herencsar, N., & Mahata, S. (2023). A theoretical demonstration for reinforcement learning of pi control dynamics for optimal speed control of dc motors by using twin delay deep deterministic policy gradient algorithm. *Expert Systems with Applications*, 213:119192.
- Uscamaita Quispetupa, R. (2019). *Diseño de un sistema de detección de fallas para el sensor de velocidad de un motor DC Lucas Nülle configurado en derivación*. [Tesis de grado]. Universidad Nacional de San Antonio Abad del Cusco.
- Živković, L., Benšić, T., Barukčić, M., & Kurtović, G. (2025). A survey on mission-profile-based stress testing of electric motor drives. *Applied Sciences*, 15(9).
- Wang, M., Wang, K., Zhao, Q., Zheng, X., Gao, H., & Yu, J. (2023). Lqr control and optimization for trajectory tracking of biomimetic robotic fish based on unreal engine. *Biomimetics*, 8(2).
- Westenbroek, T., Siththaranjan, A., Sarwari, M., Tomlin, C. J., & Sastry, S. S. (2022). On the computational consequences of cost function design in nonlinear optimal control. *arXiv preprint*. <https://arxiv.org/abs/2204.01986>.
- Yapp, K. K. C., Hoo, C. L., & Lai, C. H. (2024). New anti-windup proportional-integral-derivative for motor speed control. *Asian Journal of Control*, 26(6):2854–2866.

Zhang, W., Dong, H., Xu, Y., Cao, D., & Li, X. (2021). Multiobjective tuning and performance assessment of pid using teaching–learning-based optimization. *ACS Omega*, 6(47):31765–31774.

ANEXOS

Anexo A: Creación y configuración del agente TD3 para la sintonización del controlador PI de velocidad en el entorno MATLAB/Simulink

```
clear; clc;
mdl = 'DCWFMotorSpeedPID_RL_LQG';
open_system(mdl);
Q = diag([1, 0]);
R = 0.02; % this is functional (this leads to it 304 which is a good result from ↔
previous simulation)
% R = 0.009

% action definition
actionInfo = rlNumericSpec([1 1]);
actionInfo.Name = 'ia_ref';
actionInfo.LowerLimit = -3.5;
actionInfo.UpperLimit = 3.5;
numActions = actionInfo.Dimension(1);

% observation definition
observationInfo = rlNumericSpec([2 1]);
observationInfo.Name = 'observations';
observationInfo.Description = 'Information on error and reference signal';
numObservations = observationInfo.Dimension(1);

% create environment object
agentblk = 'DCWFMotorSpeedPID_RL_LQG/speed_control/RL Agent';
env = rlSimulinkEnv(mdl, agentblk, observationInfo, actionInfo);

% reset function
env.ResetFcn = @(in)localResetFcn(in);

% simulation time and agent sample time
Ts = 32/8000; % agent sample time
Tf = 1.2; % simulation time (number of iterations)
rng(0);

% critic network
statePath = [featureInputLayer(numObservations, 'Normalization', 'none', 'Name', 'State')
    fullyConnectedLayer(64*2, 'Name', 'fc1')];
actionPath = [featureInputLayer(numActions, 'Normalization', 'none', 'Name', 'Action')
    fullyConnectedLayer(64*2, 'Name', 'fc2')];
commonPath = [additionLayer(2, 'Name', 'add')]
```

```

reluLayer('Name','relu2')
fullyConnectedLayer(64, 'Name','fc3')
reluLayer('Name','relu3')
fullyConnectedLayer(16, 'Name','fc4')
fullyConnectedLayer(1, 'Name','qValue')];

criticNetwork = layerGraph();
criticNetwork = addLayers(criticNetwork,statePath);
criticNetwork = addLayers(criticNetwork,actionPath);
criticNetwork = addLayers(criticNetwork,commonPath);
criticNetwork = connectLayers(criticNetwork,'fc1','add/in1');
criticNetwork = connectLayers(criticNetwork,'fc2','add/in2');
% analyzeNetwork(criticNetwork)

figure
plot(criticNetwork)

% critic NN options
criticOptions = rlRepresentationOptions('LearnRate',1.5e-4,'GradientThreshold',1,'↔
    UseDevice','gpu');
critic1 = rlQValueRepresentation(criticNetwork,observationInfo,actionInfo,...
    'Observation',{'State'},'Action',{'Action'},criticOptions);

% this is for T3DDPG
critic2 = rlQValueRepresentation(criticNetwork,observationInfo,actionInfo,...
    'Observation',{'State'},'Action',{'Action'},criticOptions);
critic = [critic1, critic2];

% Actor NN
% scale = 3.5;
initial = [0.1, 0.1];

% actorNN = [featureInputLayer(numObservations,'Normalization','none','Name','State')
%     fullyConnectedLayer(64, 'Name','actorFC1')
%     tanhLayer('Name','tanh1')
%     fullyConnectedLayer(32, 'Name','actorFC2')
%     tanhLayer('Name','tanh2')
%     fullyConnectedLayer(numActions,'Name','Action')
%     tanhLayer('Name','tanh3')
%     scalingLayer('Name','ActorScaled','Scale',scale)];

% RL pid tuning 1
% actorNN = [featureInputLayer(numObservations,'Normalization','none','Name','State')
%     fullyConnectedPILayer(initial, 'action')
%     tanhLayer('Name','tanh1')
%     scalingLayer('Name','ActorScaled','Scale',scale)];

% RL pid tuning 2
actorNN = [featureInputLayer(numObservations,'Normalization','none','Name','State')
    fullyConnectedPILayer(initial, 'action')
];

actorNetwork = layerGraph();
actorNetwork = addLayers(actorNetwork, actorNN);

figure
plot(actorNetwork)

%actorOptions = rlRepresentationOptions('LearnRate',1e-1,'GradientThreshold',1);
actorOptions = rlRepresentationOptions('LearnRate',5e-3,'GradientThreshold',1,'↔
    UseDevice','gpu');
actor = rlDeterministicActorRepresentation(actorNetwork,observationInfo,actionInfo,...
    'Observation',{'State'},'Action',{'action'},actorOptions);

% TD3agent
agentOpts = rlTD3AgentOptions(...
    'SampleTime',Ts, ...
    'DiscountFactor', 0.9995,...
    'ExperienceBufferLength',1e6, ...
    'MiniBatchSize',1024, ...
    'TargetUpdateFrequency', 20);

```

```

agentOpts.ExplorationModel.Variance = 0.06; % before 0.05
agentOpts.TargetPolicySmoothModel.Variance = 0.05;

agent = rlTD3Agent(actor, critic, agentOpts);

%agent training
maxepisodes = 2000; %1000
maxsteps = ceil(Tf/Ts);
trainOpts = rlTrainingOptions(...
    'MaxEpisodes',maxepisodes,...
    'MaxStepsPerEpisode',maxsteps,...
    'ScoreAveragingWindowLength',40,...
    'Verbose',false,...
    'Plots','training-progress',...
    'StopTrainingCriteria','AverageReward',...
    'StopTrainingValue',1e10, ... % save agent each episode -> next lines
    'SaveAgentCriteria', 'EpisodeCount', ... % Save based on episode count
    'SaveAgentValue', 1, ... % Save every 1 episode
    'SaveAgentDirectory', 'saved_agents'); % Directory to save);

doTraining = true;
if doTraining
    testReferences = 0;
    trainingStats = train(agent,env,trainOpts);

    save('DCWF_PID_TD3DDPG.mat', 'agent', 'mdl', 'maxsteps', 'env', 'Ts',...
        'trainingStats');
    testReferences = 1;
else
    testReferences = 1;
    load('DCWF_PID_TD3DDPG.mat');
end

% reste function implementation
function in = localResetFcn(in)

    % Randomize reference signal
    blk = sprintf('DCWFMotorSpeedPID_RL_LQG/speed_ref');
    blk2 = sprintf('DCWFMotorSpeedPID_RL_LQG/speed_ref2');

    % randomize reference signal
    %refSpeedValues = [20, 30, 50, 70, 80];
    refSpeedValues = [70, 90, 120, 150];
    %refSpeedValues = [20, 40, 60, 80];
    idx = randperm(length(refSpeedValues),1);
    refspeed = refSpeedValues(idx);
    %idx2 = randperm(length(refSpeedValues2),1);
    %refspeed2 = refSpeedValues2(idx2);
    in = setBlockParameter(in,blk,'After',num2str(refspeed));
    %in = setBlockParameter(in,blk2,'After',num2str(refspeed2));

end

```

Anexo B: Algoritmo de Hwer para obtener la ganancia óptima de control del LQR en MATLAB

```

%% LQR for speed control of a WF DC Motor (Hwer algorithm)
% DC motor model
clear; clc
Le = 108.97; % Le = 108.97 H
Re = 2059.39; % Re = 2059.39 Ohm
La = 0.40682; % La = 406.82 mH
Ra = 16.95; % Ra = 16.95 Ohm

```

```

cm = 0.97; % machine constant
% theta = 0.000869; % Rotor inertial
theta = 0.001699; % Rotor inertia with the servo brake coupled
%% Motor State Space Model
ve = 220;
ie = ve/Re; % for constant ve = 220V in steady state
Psi_e = -74.222*(ie^2) + 17.493*ie - 0.0018;

Ac = [0, (cm*Psi_e)/theta, -(cm*Psi_e)/La, -Ra/La];
Bc = [0; 1/La];
Cc = [1, 0]; % output speed
Dc = [0];

% state space model
wfdc_motor_ss = ss(Ac, Bc, Cc, Dc);
%% Discrete System
Ts = 1/8000; % sampling period
sysd = c2d(wfdc_motor_ss, Ts);
Ad = sysd.A; Bd = sysd.B;
Cd = sysd.C; Dd = sysd.D;
%% Integral state feedback - servo controller
Ts = 1/8000;
A_fs = [Ac, zeros(2, 1); -Cc, 0];
B_fs = [Bc; 0];
C_fs = eye(3);
D_fs = [0; 0; 0];

% method 1
wfdc_motor_fs_ss = ss(A_fs, B_fs, C_fs, D_fs);
sysd_fs = c2d(wfdc_motor_fs_ss, Ts);
Ad_fs = sysd_fs.A;
Bd_fs = sysd_fs.B;
%%
Q = diag([1, 10, 200]);
R = 1.8;

A = Ad_fs; B = Bd_fs;
Qf = Q; % terminal cost (often same as Q)
N = 4000; % horizon length, choose large if infinite horizon

% Initialization
P = Qf;

% [K_his, K_his_norm, K] = DLQR_FS(A, B, Q, R, N)
[K_his, K_his_norm, K] = DLQR_FS(A, B, Q, R, N);

disp('Optimal LQR gain K =');
disp(K);

% DLQR function
function [K_his, K_his_norm, K] = DLQR_FS(A, B, Q, R, N)
%DLQR_FS computes the optimal K gain of a discrete system using the
%Hewer algorithm
% -
P = Q;
% K history
K_his = zeros([3, N]);
K_his_norm = zeros([1, N]); % Kk norm
for k = N-1 : -1 : 0
    S = R + B' * P * B;
    Kk = S \ (B' * P * A); % gain at time k
    % save Kk to K_his
    K_his(:, N - k) = Kk';
    % store the Kk norma
    K_his_norm(N - k) = norm(Kk);
    P = Q + A' * P * A - A' * P * B * Kk;
end
% Final optimal gain at k=0
K = (R + B' * P * B) \ (B' * P * A);
end

```

Anexo C: Algoritmo MGPI par obtener la ganancia óptima de control del LQR en MATLAB

```
% LQR for speed control of a WF DC Motor (MGPI algorithm)
% WF DC motor model
clear; clc
Le = 108.97; % Le = 108.97 H
Re = 2059.39; % Re = 2059.39 Ohm
La = 0.40682; % La = 406.82 mH
Ra = 16.95; % Ra = 16.95 Ohm
cm = 0.97; % machine constant
% theta = 0.000869; % Rotor inertial
theta = 0.001699; % Rotor inertia with the servo brake coupled
%% Motor State Space Model
ve = 220;
ie = ve/Re; % for constant ve = 220V in steady state
Psi_e = -74.222*(ie^2) + 17.493*ie - 0.0018;

Ac = [0, (cm*Psi_e)/theta, -(cm*Psi_e)/La, -Ra/La];
Bc = [0; 1/La];
Cc = [1, 0]; % output speed
Dc = [0];

% state space model
wfdc_motor_ss = ss(Ac, Bc, Cc, Dc);
%% Discrete System
Ts = 1/8000; % sampling period
sysd = c2d(wfdc_motor_ss, Ts);
Ad = sysd.A; Bd = sysd.B;
Cd = sysd.C; Dd = sysd.D;
%% Integral state feedback - servo controller
Ts = 1/8000;
A_fs = [Ac, zeros(2, 1); -Cc, 0];
B_fs = [Bc; 0];
C_fs = eye(3);
D_fs = [0; 0; 0];

% method 1
wfdc_motor_fs_ss = ss(A_fs, B_fs, C_fs, D_fs);
sysd_fs = c2d(wfdc_motor_fs_ss, Ts);
Ad_fs = sysd_fs.A;
Bd_fs = sysd_fs.B;

% method 2
% Ad_fs = [Ad, zeros(2, 1); -Cd, 0];
% Bd_fs = [Bd; 0];
%% LQR design
Q = diag([1, 10, 200]);
R = 1.8;
% initial K0
K0 = [1, 1, 1];

% initial ricatti solution
P0 = eye(3);
%PO_j = P0;
Pi_j = P0;
Pj = P0;
Kj = K0;
% policy iteration algorithm

%Ts = h;
epsilon = 0.0005;
% iteration vals
L = 1000;
N = 10;

% K history
```

```

% K_his = zeros([3, N]);
% K_his_norm = zeros([1, N]); % Kk norm
%% calculate the gain and the
% [K_his, K_his_norm, K] = MGPI_FS(A, B, Q, R, N, L, P0, K0, Ts, lambda)
lambda1 = 0.8; % 0.5
[K_his1, K_his_norm1, K1] = MGPI_FS(Ad_fs, Bd_fs, Q, R, N, L, P0, K0, Ts, lambda1);

lambda2 = 0.5;
[K_his2, K_his_norm2, K2] = MGPI_FS(Ad_fs, Bd_fs, Q, R, N, L, P0, K0, Ts, lambda2);

lambda3 = 0.1;
[K_his3, K_his_norm3, K3] = MGPI_FS(Ad_fs, Bd_fs, Q, R, N, L, P0, K0, Ts, lambda3);

lambda4 = 0.08;
[K_his4, K_his_norm4, K4] = MGPI_FS(Ad_fs, Bd_fs, Q, R, N, L, P0, K0, Ts, lambda4);

% MGPI_FS function
function [K_his, K_his_norm, K] = MGPI_FS(A, B, Q, R, N, L, P0, K0, Ts, lambda)
%MGPI_FS computes the optimal K gain of a discrete system using the
%Generalized Policy Gradient Hwewer algorithm
% -
% policy iteration for K
% K history
K_his = zeros([3, N]);
K_his_norm = zeros([1, N]); % Kk norm
Pj = P0; % initial value of Pj
Kj = K0; % initial value of Kj
for j = 1:N
    Pi_j = Pj;
    % Policy evaluation
    for i = 1:L
        Pi_j = (lambda^(-Ts) * A - lambda^(-Ts)* B * Kj)' * Pi_j...
            *(lambda^(-Ts) * A - lambda^(-Ts)* B * Kj) + Q + Kj'*R*Kj;
    end
    Pj = Pi_j;
    % Policy improvement
    Kj = lambda^(-2*Ts)*inv(R + lambda^(-2*Ts)*B'*Pj*B)*B'*Pj*A;
    % save Kj to K_his
    K_his(:, j) = Kj';
    % store the Kk norma
    K_his_norm(j) = norm(Kj);
end
K = Kj;
end

```