



INSTITUTO POLITÉCNICO NACIONAL

UNIDAD PROFESIONAL INTERDISCIPLINARIA EN  
INGENIERÍA Y TECNOLOGÍAS AVANZADAS

*Trabajo Terminal II*

**“Optimización basada en métodos metaheurísticos  
para la sintonización de controladores PID en  
sistemas mecatrónicos”**

*Que para obtener el título de*  
**“Ingeniero en Mecatrónica”**

Presentan:

**Cortéz Conde Alexander**  
**Valdez Cruz Marco Antonio**

Asesores:

Dr. Cuervo Pinto Victor Darío  
M. en C. Fonseca Campos Jorge  
Dr. Rodríguez Molina Alejandro



Ciudad de México, a 27 de junio de 2022

# INSTITUTO POLITÉCNICO NACIONAL

UNIDAD PROFESIONAL INTERDISCIPLINARIA EN  
INGENIERÍA Y TECNOLOGÍAS AVANZADAS



*Trabajo Terminal II*

## “Optimización basada en métodos metaheurísticos para la sintonización de controladores PID en sistemas mecatrónicos”

*Que para obtener el título de*

“Ingeniero en Mecatrónica”

Presenta:

Cortéz Conde Alexander

Valdez Cruz Marco Antonio

Asesores:

Dr. Cuervo Pinto Victor

Darío

M. en C. Fonseca Campos

Jorge

Dr. Rodríguez Molina

Alejandro

Presidente del Jurado

Dr. Adolfo Rojas Pacheco

Profesor titular

M. en E. Rivas Bonilla

Elizabeth







## Autorización de uso de obra

**Instituto Politécnico Nacional**

**Presente**


Bajo protesta de decir verdad el que suscribe Alexander Cortéz Conde  
(se anexa copia simple de identificación oficial), manifiesto ser autor (a) y titular de los  
derechos morales y patrimoniales de la obra titulada Optimización basada en  
métodos metaheurísticos para la sintonización de controladores PID en  
sistemas mecatrónicos

\_\_\_\_\_, en adelante "La Tesis" y de la cual se adjunta copia, por lo que por medio  
del presente y con fundamento en el artículo 27 fracción II, inciso b) de la Ley Federal  
del Derecho de Autor, otorgo a el Instituto Politécnico Nacional, en adelante El IPN,  
autorización no exclusiva para comunicar y exhibir públicamente total o parcialmente  
en medios digitales, Plataforma de la Dirección de Bibliotecas del IPN y/o consulta  
directa en la Coordinación de Biblioteca de la UPIITA "La Tesis" por un periodo de 5  
años contado a partir de la fecha de la presente autorización, dicho periodo se renovará  
automáticamente en caso de no dar aviso expreso a "El IPN" de su terminación.

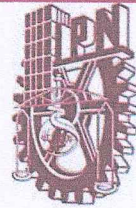
En virtud de lo anterior, "El IPN" deberá reconocer en todo momento mi calidad de autor de  
"La Tesis". Adicionalmente, y en mi calidad de autor y titular de los derechos morales y  
patrimoniales de "La Tesis", manifiesto que la misma es original y que la presente  
autorización no contraviene ninguna otorgada por el suscrito respecto de "La Tesis", por lo  
que deslindo de toda responsabilidad a El IPN en caso de que el contenido de "La Tesis" o  
la autorización concedida afecte o viole derechos autorales, industriales, secretos  
industriales, convenios o contratos de confidencialidad o en general cualquier derecho de  
propiedad intelectual de terceros y asumo las consecuencias legales y económicas de  
cualquier demanda o reclamación que puedan derivarse del caso.

Ciudad de México, a 27 de junio de 2022

**Atentamente**

  
\_\_\_\_\_  
Alexander Cortéz Conde





## Autorización de uso de obra

**Instituto Politécnico Nacional**

**Presente**

Bajo protesta de decir verdad el que suscribe Marco Antonio Valdez Cruz (se anexa copia simple de identificación oficial), manifiesto ser autor (a) y titular de los derechos morales y patrimoniales de la obra titulada Optimización basada en métodos metaheurísticos para la sintonización de Controladores PID en sistemas mecatrónicos

\_\_\_\_\_, en adelante "La Tesis" y de la cual se adjunta copia, por lo que por medio del presente y con fundamento en el artículo 27 fracción II, inciso b) de la Ley Federal del Derecho de Autor, otorgo a el Instituto Politécnico Nacional, en adelante El IPN, autorización no exclusiva para comunicar y exhibir públicamente total o parcialmente en medios digitales, Plataforma de la Dirección de Bibliotecas del IPN y/o consulta directa en la Coordinación de Biblioteca de la UPIITA "La Tesis" por un periodo de 5 años contado a partir de la fecha de la presente autorización, dicho periodo se renovará automáticamente en caso de no dar aviso expreso a "El IPN" de su terminación.

En virtud de lo anterior, "El IPN" deberá reconocer en todo momento mi calidad de autor de "La Tesis". Adicionalmente, y en mi calidad de autor y titular de los derechos morales y patrimoniales de "La Tesis", manifiesto que la misma es original y que la presente autorización no contraviene ninguna otorgada por el suscrito respecto de "La Tesis", por lo que deslindo de toda responsabilidad a El IPN en caso de que el contenido de "La Tesis" o la autorización concedida afecte o viole derechos autorales, industriales, secretos industriales, convenios o contratos de confidencialidad o en general cualquier derecho de propiedad intelectual de terceros y asumo las consecuencias legales y económicas de cualquier demanda o reclamación que puedan derivarse del caso.

Ciudad de México, a 27 de Junio de 2022

Atentamente

Marco Antonio Valdez Cruz



# AGRADECIMIENTOS

A mis asesores, el Dr. Alejandro Rodríguez, el M. en C. Victor Darío Cuervo y el M. en C. Jorge Fonseca, quienes con su conocimiento, apoyo, disponibilidad y tiempo, nos guiaron en cada una de las etapas de este proyecto. A mi compañero de Trabajo Terminal Marco Antonio Valdez, por su paciencia, dedicación, conocimiento pero sobre todo su amistad.

A mis padres, María Elena Conde y Vicente Cortéz, por su paciencia, amor y apoyo brindado a lo largo de toda mi vida. A mi hermana Stephanie Cortéz y mi abuela materna Catalina Hernández, que aunque pasé tiempo lejos de casa, siempre me hicieron recordar que tenía una. También a mi perro Odi, el ser que me extrañaba por semanas pero siempre me recibía moviendo la colita, y se acurrucaba conmigo cuando sentía que ya no podía más.

A los amigos que hice a lo largo de mi estancia en la UPIITA, Carlos Ramírez, Ciro Gil, Gabino Gallegos, Iván Estrada, Jonathan Ponce, Marco Solís, Miguel Gómez, Pedro Vargas, Ricardo Castillo, Jonathan Hernández y de manera especial a Brian Suvias, mi primer y mejor amigo durante la carrera, con quien no solo compartí clases, sino risas, tristezas y tiempo. De igual manera, a las amistades que me han acompañado en todo momento incluso antes de UPIITA; Andrea Rocha, Brandon Patoni, Camelia Montes de Oca, Daniela García, Diana Cid, Dulce del Ángel, Eduardo Montealegre, Enrique de León, Fernando Navarrete, Jair Soto, Marco Montes de Oca, Maylín Crespo, Michelle Capetillo y Patricia Orta. Y a todas las personas que no están en esta página pero sí en mi corazón.

*Alexander Cortéz Conde*



A mis padres, Isabel Cruz Cruz y Gaudencio Valdez De la Cruz, por su apoyo y consejos de toda la vida, por brindarme la oportunidad de cumplir una de mis metas, y todo cariño y cuidados, sin ustedes esto no habría sido posible, infinitas gracias por todo lo que han hecho por mí.

Agradezco a mis hermanas Ana Karen Valdez Cruz y Karime Isabel Valdez Cruz por ser mis compañera de vida y aguantar mi carácter.

A mis asesores M. en C. Jorge Fonseca Campos, Dr. Alejandro Rodríguez Molina, Dr. Víctor Darío Cuervo Pinto y a la profesora titular de Trabajo Terminal M. en E. Elizabeth Rivas Bonilla por su participación en este trabajo, sus recomendaciones y asesorías que fueron de gran ayuda para concluir de manera exitosa este trabajo, pero, sobre todo, la paciencia con la que nos trataron, muchas gracias.

A mis amigos que estuvieron conmigo desde el inicio de este trayecto, Sergio Valente Rodríguez Morquecho, Ricardo Castillo Botello, Brian Suvias Salazar, Rubén Licona Chávez, a mis amigos que se unieron en el camino y fueron de mucha ayuda durante el periodo de pandemia, Luis Oviedo Cedillo, Tsuioshi Alberto Luviano Murakawa, Daniel Terrazas Medina, Yonatan Hernández Ramírez y Emmanuel Alejandro Larralde Ortiz, a todos ustedes muchas gracias por brindarme su amistad y estar presentes en mi travesía por la UPIITA.

A mi compañero de tesis, Alexander Cortéz Conde, por tu apoyo, por nunca dejarme solo, por tu compañerismo y por tu amistad.

Por último, agradezco a Dios, por permitirme llegar hasta esta instancia, a la UPIITA y al IPN por brindarme los conocimientos teóricos y prácticos que fueron fundamentales para concluir este trabajo.

*Marco Antonio Valdez Cruz*



---

## Contenido

---

<b>Resumen/Abstract</b>	<b>VII</b>
<b>Abreviaturas</b>	<b>XIX</b>
<b>Simbología</b>	<b>XXI</b>
<b>Introducción</b>	<b>XXV</b>
0.1. Definición del problema . . . . .	XXV
0.2. Justificación . . . . .	XXVIII
0.3. Objetivo . . . . .	XXIX
Objetivos específicos . . . . .	XXX
0.4. Antecedentes . . . . .	XXXI
<b>1. Marco Referencial</b>	<b>1</b>
1.1. Marco Histórico . . . . .	1
Control Automático . . . . .	1
Optimización . . . . .	2



1.2. Marco teórico . . . . .	3
Controladores automáticos . . . . .	3
Métodos de integración . . . . .	6
Dinámica . . . . .	7
Cinemática . . . . .	8
1.3. Marco conceptual . . . . .	11
Métodos metaheurísticos . . . . .	14
Algoritmos Evolutivos . . . . .	15
Algoritmos genéticos (GA) . . . . .	16
Evolución diferencial (DE) . . . . .	20
Optimización por Enjambre de Partículas (PSO) . . . . .	22
Optimización multiobjetivo (MOP) . . . . .	25
Algoritmos Evolutivos multiobjetivo (MOEA) . . . . .	26
Índices de desempeño de un sistema controlado . . . . .	27
Indicadores de Calidad . . . . .	28
Pruebas no paramétricas . . . . .	30
Método Analítico Jerárquico (AHP) . . . . .	31
1.4. Marco procedimental . . . . .	33
Sintonización de controladores multiobjetivo . . . . .	33
Optimización multiobjetivo con métodos metaheurísticos para la sintonización de controladores . . . . .	35
Metodología . . . . .	36
<b>2. Diseño del sistema</b>	<b>39</b>
2.1. Diseño conceptual . . . . .	39
Necesidades y requerimientos . . . . .	39





Arquitectura funcional . . . . .	41
Arquitectura física . . . . .	46
Matriz morfológica . . . . .	47
Búsqueda morfológica . . . . .	48
Selección de diseño conceptual . . . . .	49
Propuesta solución . . . . .	51
Validación . . . . .	70
2.2. Diseño detallado . . . . .	73
Modelado del sistema. . . . .	73
Sintonización con métodos metaheurísticos para optimización multi- objetivo. . . . .	73
Interfaz Gráfica (GUI). . . . .	74
Adaptación del péndulo simple. . . . .	78
2.3. Validación e integración de los sistemas . . . . .	91
<b>3. Implementación</b>	<b>95</b>
Modelado de la planta . . . . .	96
Interfaz . . . . .	96
Adapación del péndulo simple . . . . .	106
Control del péndulo simple . . . . .	116
<b>4. Resultados</b>	<b>119</b>
Simulaciones . . . . .	119
Prueba de Wilcoxon . . . . .	128
Control de un péndulo simple . . . . .	134
Propuesta alternativa . . . . .	138



---

<b>5. Conclusiones</b>	<b>147</b>
<b>Referencias</b>	<b>151</b>
<b>Apéndices</b>	<b>161</b>
<b>Apéndice A: Modelos dinámicos</b>	<b>163</b>
.1. Péndulo simple . . . . .	163
.2. Péndulo invertido . . . . .	166
.3. Péndulo doble . . . . .	171
<b>Apéndice B: Códigos y planos</b>	<b>179</b>



---

## Resumen/Abstract

---

Este trabajo aborda la sintonización de controladores de tipo PID de tres sistemas mecatrónicos comúnmente usados en la industria: el péndulo simple, invertido y doble, mediante algoritmos evolutivos (Evolución Diferencial, Algoritmo Genético y Optimización por Enjambre de Partículas). El problema de sintonización se abordó como uno de optimización multiobjetivo, ya que se pretendía minimizar dos objetivos: el error mediante la Integral del Error Cuadrático (ISE por sus siglas en inglés) y las variaciones en la señal de control mediante la Integral del valor Absoluto de la señal de control (IADU, por sus siglas en inglés). Lo anterior, con la intención de obtener un compromiso adecuado entre la precisión en la regulación de la salida y el tiempo de vida del sistema mecatrónico. Se desarrolló una interfaz gráfica programada en Python que tiene como objetivo brindarle al usuario la posibilidad de realizar una sintonización *offline*. Finalmente, se implementó el valor de las ganancias  $K_p$ ,  $K_i$  y  $K_d$  encontradas, en un péndulo simple físico, con el fin de validar los resultados obtenidos en simulación.

**Palabras Clave:**

Optimización multiobjetivo, sintonización, Algoritmo Genético, Evolución Diferencial, Optimización por enjambre de partículas, control PID, aproximación al frente de Pareto

**Abstract**

This document presents the tuning of PID type controllers of three mechatronic systems commonly used in industry: single, inverted and double pendulum, by means of evolutionary algorithms (Differential Evolution, Genetic Algorithm and Particle Swarm Optimization). The tuning problem was approached as a multiobjective optimization problem, since it was intended to minimize two objectives: the error through of the Integral of the Quadratic Error (ISE) and the variations in the control signal through the Integral of the Absolute value of the control signal (IADU). The above, with the intention of obtaining an adequate compromise between the precision in the output regulation and the life time of the mechatronic system. A graphical interface programmed in Python was developed with the objective of providing the user with the possibility of performing an offline tuning. Finally, the value of the gains  $K_p$ ,  $K_i$  and  $K_d$  found were implemented in a simple physical pendulum in order to validate the results obtained in simulation.

**Key words:**

Multi-objective optimization, tuning, genetic algorithm, differential evolution, swarm particle, PID control, Pareto front.

---

## Índice de figuras

---

1.1. Diagrama de bloques de un sistema de control industrial. Fuente: Adaptado de [26]. . . . .	3
1.2. Metodología de diseño por medio de optimización. . . . .	12
1.3. Diagrama de un EA. Fuente: Adaptado de . . . . .	16
1.4. Diagrama de un GA. . . . .	17
1.5. A) Full connected, B) Ring C) Star . . . . .	24
1.6. Hipervolumen (HV). . . . .	29
1.7. Pasos típicos para la sintonización de controladores multiobjetivo. . . . .	34
1.8. MOP optimizado con un método metaheurístico. . . . .	35
1.9. Modelo en V. . . . .	36
1.10. Macro ciclos y madurez del producto. . . . .	37
2.1. Diagrama IDEF0. . . . .	42
2.2. Diagrama IDEF0 Nodo A0. . . . .	44
2.3. Diagrama IDEF0 Nodo A2. . . . .	45





2.4. Matriz morfológica. . . . .	47
2.5. Ruta 1. . . . .	48
2.6. Ruta 2. . . . .	48
2.7. Ruta 3. . . . .	49
2.8. Péndulo simple. . . . .	52
2.9. Posición del péndulo simple en el plano XY. . . . .	53
2.10. Gráficas de salida con respecto al tiempo del péndulo simple a $\theta = 0$ . . . . .	54
2.11. Posición del péndulo en el plano XY. . . . .	54
2.12. Gráficas de salida con respecto al tiempo del péndulo simple a $\theta = \frac{\pi}{2}$ . . . . .	55
2.13. Péndulo invertido. . . . .	55
2.14. Posición del péndulo invertido en el plano XY. . . . .	57
2.15. Gráficas de salida respecto al tiempo péndulo invertido a $\theta = x = 0$ . . . . .	58
2.16. Posición en el plano XY del péndulo invertido. . . . .	58
2.17. Gráficas de salida con respecto al tiempo del péndulo invertido a $\theta = \frac{\pi}{2}, x = 0$ . . . . .	59
2.18. Posición en el plano XY (Péndulo doble). . . . .	62
2.19. Gráficas de salida con respecto al tiempo del Péndulo doble a condiciones iniciales $\theta_1 = \theta_2 = 0$ . . . . .	63
2.20. Posición en el plano XY (Péndulo doble). . . . .	63
2.21. Gráficas de salida con respecto al tiempo del péndulo doble a condiciones iniciales $\theta_1 = \frac{\pi}{2}, \theta_2 = 1$ . . . . .	64
2.22. ISE vs ISE. . . . .	71
2.23. IADU vs IADU. . . . .	71
2.24. Diagrama de flujo de la GUI. . . . .	75
2.25. Selección de materiales. . . . .	78
2.26. Sistema péndulo simple. . . . .	80



2.27. Motor. . . . .	81
2.28. Eje. . . . .	81
2.29. Chumacera. . . . .	82
2.30. Cople. . . . .	82
2.31. Elemento de sujeción. . . . .	82
2.32. Eje de aluminio. . . . .	83
2.33. Motor con encoder. . . . .	87
2.34. Teensy 4.1. . . . .	87
2.35. Configuración de pines. . . . .	88
2.36. Circuito de control. . . . .	89
2.37. Módulo L298N. . . . .	90
2.38. Aproximación al frente de Pareto. . . . .	91
2.39. Circuito de control del péndulo. . . . .	92
2.40. Ejecución de simulación. . . . .	92
2.41. Convertidor de nivel lógico de 5 V a 3.3 V. . . . .	93
2.42. Péndulo simple (Control). . . . .	94
2.43. Péndulo simple (Control). . . . .	94
3.1. Diagrama de flujo de la GUI. . . . .	97
3.2. Selección de mecanismo. . . . .	98
3.3. Sintonización péndulo simple. . . . .	99
3.4. Sintonización péndulo invertido. . . . .	99
3.5. Sintonización péndulo doble. . . . .	100
3.6. Parámetros para Evolución Diferencial. . . . .	101
3.7. Parámetros para Algoritmo Genético. . . . .	101
3.8. Parámetros para Optimización por Enjambre de Partículas. . . . .	101



3.9. Confirmación del algoritmo para el péndulo simple. . . . .	102
3.10. Confirmación del algoritmo para el péndulo invertido. . . . .	103
3.11. Confirmación del algoritmo para el péndulo doble. . . . .	103
3.12. Resultados de la optimización multiobjetivo y aproximación al Frente de Pareto. . . . .	104
3.13. Simulación péndulo simple. . . . .	105
3.14. Respuesta de control péndulo simple. . . . .	106
3.15. Diagrama para la estimación de parámetros ( $\omega$ ). . . . .	108
3.16. Mediciones de corriente y tensión para el cálculo de la resistencia. . .	108
3.17. Tensión vs velocidad. . . . .	111
3.18. Torque vs Velocidad. . . . .	112
3.19. Sistema péndulo simple. . . . .	113
3.20. Motor. . . . .	113
3.21. Posición 0. . . . .	114
3.22. Posición $2\pi$ . . . . .	114
3.23. Chumaceras KP08. . . . .	115
3.24. Elemento de sujeción. . . . .	115
3.25. Canales del encoder. . . . .	116
3.26. Implementación del control PID. . . . .	117
4.1. Sintonización PID del péndulo simple con DE. . . . .	121
4.2. Sintonización PID del péndulo simple con GA. . . . .	121
4.3. Sintonización PID del péndulo simple con PSO. . . . .	122
4.4. Sintonización PID del péndulo invertido con DE. . . . .	123
4.5. Sintonización PID del péndulo invertido con GA. . . . .	124
4.6. Sintonización PID del péndulo invertido con PSO. . . . .	124



4.7. Sintonización PD del péndulo doble con DE. . . . .	126
4.8. Sintonización PD del péndulo doble con GA. . . . .	126
4.9. Sintonización PD del péndulo doble con PSO. . . . .	127
4.10. Ejecución de los programas en laboratorios de la UPIITA. . . . .	129
4.11. Parámetros dinámicos del péndulo simple. . . . .	134
4.12. Parámetros DE. . . . .	135
4.13. Aproximación al Frente de Pareto con DE. . . . .	135
4.14. Péndulo simple . . . . .	137
4.15. Péndulo simple con control de posición. . . . .	137
4.16. Propuesta alternativa para el péndulo simple. . . . .	138
4.17. Motorreductor Pololu 37Dx73L . . . . .	138
4.18. Curvas características del motor Pololu. Consultado en [1]. . . . .	139
4.19. Datos nominales del Motor pololu 37D. Consultado en [1] . . . . .	140
4.20. Encoder rotativo A38S6-400-2-2-24. . . . .	141
4.21. Arduino UNO. . . . .	141
4.22. Nuez universal. . . . .	142
4.23. Péndulo simple alternativo. . . . .	143
4.24. Péndulo simple alternativo a referencia $\pi$ . . . . .	143
4.25. Péndulo simple alternativo a referencia $\frac{\pi}{2}$ . . . . .	144
4.26. Péndulo simple alternativo a referencia $\frac{\pi}{4}$ . . . . .	144
1. Péndulo simple. . . . .	163
2. Péndulo invertido. . . . .	166
3. Péndulo doble. . . . .	171





---

## Índice de Tablas

---

2.1. Necesidades del sistema. . . . .	40
2.2. Requerimientos. . . . .	40
2.3. Matriz de trazabilidad. . . . .	41
2.4. Familia IDEF. . . . .	41
2.5. Matriz de trazabilidad para la validación de funciones. . . . .	46
2.6. Variables y criterios. . . . .	50
2.7. Matriz de comparación de criterios. . . . .	50
2.8. Criterio: Número de elementos . . . . .	50
2.9. Criterio: Comunicación motor-sensor. . . . .	50
2.10. Criterio: Ensamble. . . . .	51
2.11. Selección de ruta. . . . .	51
2.12. Variables y criterios. . . . .	65
2.13. Matriz de comparación de criterios. . . . .	65
2.14. Criterio: Supresión de errores. . . . .	65
2.15. Criterio: Implementación digital. . . . .	65



2.16. Criterio: Evaluación analítica. . . . .	66
2.17. Selección de índice de desempeño. . . . .	66
2.18. Análisis de ventajas y desventajas en la selección de materiales. . . . .	79
2.19. Matriz de comparación de criterios. . . . .	85
2.20. Criterio: Resolución. . . . .	85
2.21. Criterio: Torque. . . . .	86
2.22. Criterio: Precio. . . . .	86
2.23. Criterio: Consumo energético. . . . .	86
2.24. Selección de motor con encoder. . . . .	86
3.1. Medición de la resistencia. . . . .	109
3.2. Medición de la velocidad. . . . .	110
4.1. Parámetros de optimización . . . . .	120
4.2. Sintonización PID del péndulo simple. . . . .	123
4.3. Sintonización PID del péndulo invertido. . . . .	125
4.4. Sintonización PD del péndulo doble. . . . .	128
4.5. Resultados de hipervolumen. . . . .	130
4.6. Valor promedio de HV para el péndulo simple. . . . .	131
4.7. Valor promedio de HV para el péndulo invertido. . . . .	131
4.8. Valor promedio de HV para el péndulo doble. . . . .	131
4.9. Prueba de Wilcoxon para el péndulo simple. . . . .	132
4.10. Prueba de Wilcoxon para el péndulo invertido. . . . .	133
4.11. Prueba de Wilcoxon para el péndulo doble. . . . .	133
4.12. Conteo de victorias para cada EA. . . . .	133
4.13. Conjunto de soluciones factibles. . . . .	136



---

1.	Notación péndulo simple. . . . .	164
2.	Notación péndulo invertido. . . . .	167
3.	Notación péndulo doble. . . . .	172





---

## Abreviaturas

---

<i>2D</i>	Bidimensional.
<i>AHP</i>	Analytic Hierarchy Process method (Método de Proceso Analítico Jeraquíco).
<i>DE</i>	Differential Evolution (Evolución Diferencial).
<i>FP</i>	Frente de Pareto.
<i>EA</i>	Evolutionary algorithm (Algoritmo evolutivo).
<i>GA</i>	Genetic Algorithm (Algoritmo Genético).
<i>HV</i>	Hipervolumen.
<i>IADU</i>	Integral of the Absolute value of the control action (Integral del valor absoluto de la variación de la señal de control).
<i>IDE</i>	Integrated Development Environment (Entorno de Desarrollo Integrado).
<i>IDEF</i>	Integration Definition for Function Modeling (Definición de integración de modelado de funciones).
<i>ISE</i>	Integral Square Error (Integral del error al cuadrado).
<i>MOEAs</i>	Multi-objective evolutionary algorithms (Algoritmos evolutivos multi-objetivo).



<i>MODE</i>	Multiobjective Differential Evolution (Evolución Diferencial Multiobjetivo).
<i>MOGA</i>	Multiobjective Genetic Algorithm (Algoritmo Genético Multiobjetivo).
<i>MOO</i>	Multi-objective optimisation (Optimización multi-objetivo).
<i>MOP</i>	Multi-objective Problem (Problema multi-objetivo).
<i>MOPSO</i>	Multiobjective Particle Swarm Optimization (Optimización por Enjambre de Partículas Multiobjetivo).
<i>NSGA-II</i>	Non-dominated Sorting Genetic Algorithm II (Algoritmo genético de clasificación no dominado II)
<i>PID</i>	Proporcional,Integral, Derivativo.
<i>PSO</i>	Particle Swarm Optimization (Optimización por enjambre de partículas).
<i>PWM</i>	Pulse-Width Modulation (Modulación por ancho de pulsos).
<i>S</i>	Conjunto solución óptimo.
<i>SOP</i>	Single-objective Problem (Problema mono-objetivo).
<i>VDI</i>	Verein Deutscher Ingenieure (Asociación de Ingenieros Alemanes).

---

## Simbología

---

$b$	Fricción viscosa del péndulo simple.
$b_1$	Fricción viscosa del carro (Péndulo invertido) \ Fricción viscosa del brazo 1 (Péndulo doble).
$b_2$	Fricción viscosa del brazo 2.
$C(\mathbf{q}), \dot{\mathbf{q}}$	Matriz de fuerzas centrípetas y de Coriolis.
$C_r$	Probabilidad de recombinación.
$\mathbf{D}(\dot{\mathbf{q}}, \mathbf{f}_e)$	Vector de fuerzas o pares de fricción.
$e(t)$	Señal de error.
$F$	Factor de escalado
$\mathbf{f}_e$	Fuerza de fricción estática.
$\mathbf{f}_f(\dot{\mathbf{q}}, \mathbf{f}_e)$	Vector de pares de fricción viscosa, Coulomb y estática.
$G$	Contador de generaciones.
$\mathbf{g}(\mathbf{q})$	Vector de fuerzas o pares gravitacionales.
$g$	Aceleración de la gravedad.
$g(x)$	Restricción de desigualdad.
$h(x)$	Restricción de igualdad.
$I$	Inercia de la barra.
$I_1$	Inercia del brazo 1.



$I_2$	Inercia del brazo 2.
$J(\mathbf{q})$	Jacobiano analítico.
$J(x) \in m$	Vector de objetivos de diseño (MOP).
$J_{IADU}$	Función objetivo IADU.
$J_{ISE}$	Función objetivo ISE
$K$	Energía cinética.
$k_d$	Ganancia derivativa.
$k_i$	Ganancia integral.
$k_p$	Ganancia proporcional.
$L$	Lagrangiano del sistema.
$l_1$	Longitud del brazo 1.
$l_2$	Longitud del brazo 2.
$l_c$	Longitud al centro de masa del barra.
$l_{c1}$	Longitud al centro de masa del brazo 1.
$l_{c2}$	Longitud al centro de masa del brazo 2.
$M$	Masa del carro.
$M(\mathbf{q})$	Matriz de inercia.
$m$	Masa de la barra.
$m_1$	Masa del brazo 1.
$m_2$	Masa del brazo 2.
$\mathbf{q}$	Vector de posiciones articuladas o coordenadas generalizadas.
$\dot{\mathbf{q}}$	Vector de velocidades articuladas.
$\tilde{\mathbf{q}}$	Error de posición.
$\dot{\tilde{\mathbf{q}}}$	Error de velocidad.
$\mathbf{q}_d$	Vector de posición deseada.
$P _0$	Población inicial.
$P _G$	Población actual .
$P _G^*$	Nueva población de GA.

$P _{G+1}$	Población siguiente.
$S1$	Sistema R3bótico.
$S2$	Sistema de Informaci3n.
$S3$	Sistema de administraci3n de energ3a.
$T_d$	Tiempo derivativo.
$T_i$	Tiempo integral.
$U$	Energ3a potencial.
$u_{i,j,G}0$	Vector hijo.
$u(t)$	Se3al de salida del controlador
$v$	Velocidad lineal.
$v_{i,G}$	Vector mutante.
$\mathbf{x}$	Vector de estado.
$\dot{\mathbf{x}}$	Ecuaci3n de estado.
$x \in R^n$	Vector de variables de desici3n (MOP).
$x^j _{best}$	Mejor posici3n conocida para la part3cula j (PSO).
$x^j _k$	Posici3n de una part3cula en la generaci3n k (PSO).
$x^{swarm} _{best}$	Mejor posici3n conocida por el enjambre (PSO).
$\dot{x}^j _k$	Velocidad de la part3cula en la generaci3n k (PSO).
$\underline{x}_i$	L3mite inferior del espacio de busqueda (MOP).
$\bar{x}_i$	L3mite superior del espacio de busqueda (MOP).
$\beta_1, \beta_2$	Factores de redondeo de crecimiento global.
$\theta$	Posici3n angular de la barra.
$\theta_1$	Posici3n angular del brazo 1.
$\theta_2$	Posici3n angular del brazo 2.
$\boldsymbol{\tau}$	Vector de pares aplicados.
$\varphi_1, \varphi_2$	Factores de redondeo de crecimiento local
$\omega$	Velocidad angular.



### 0.1. Definición del problema

La mecatrónica es una de las ingenierías que ha tomado un lugar muy importante en el mundo, al combinar distintas ramas como la electrónica, mecánica, computación y control, de modo que los procesos requeridos en la industria cada día requieren de una mayor participación de esta ingeniería. El control de sistemas involucra áreas de conocimiento de la mecatrónica, pues se requiere la obtención del modelo, ya sea mecánico, electrónico, o la combinación de ambos. Obtener el modelo es necesario para poder aplicar algún tipo de controlador. El controlador más usado en la industria es el PID, este incluye las ganancias: proporcional, integral y derivativa y el proceso de obtención de estas es conocido como sintonización.

Los métodos de sintonización se han clasificado según su naturaleza y uso en [2]:

- Métodos analíticos donde las ganancias de control se obtienen analizando la estabilidad del sistema en lazo cerrado.
- Métodos heurísticos donde la experiencia se considera con el ajuste manual del diseño del controlador para establecer los parámetros del controlador.
- Métodos de optimización donde se plantea un problema de programación ma-





temática y se optimiza utilizando técnicas de optimización para obtener las ganancias fijas del controlador.

- Métodos de ajuste adaptativo donde un proceso de identificación y combinación de los tres métodos anteriores se utilizan para ajustar las ganancias de control.

La Inteligencia Artificial y algunas ramas de ella, como lo son la lógica difusa, las redes neuronales, y el cómputo evolutivo, presentan una ventaja ante los métodos tradicionales de sintonización de controladores, pues estos buscan la optimización del sistema mediante algoritmos computacionales. Actualmente con la evolución de la informática y uso de los sistemas inteligentes, es mucho más conveniente usar algoritmos computacionales que nos proporcionen mejores soluciones a los problemas de control que se nos presenten. Debido al incremento de máquinas de precisión con un requerimiento de compensación, se ha abordado el problema del ajuste de control con el uso de métodos de optimización. En las últimas décadas, los algoritmos metaheurísticos y, en particular, los algoritmos evolutivos (EA, por sus siglas en inglés) [3], cuya inspiración se toma de la naturaleza de la teoría de la evolución y la supervivencia de los más aptos, han sido utilizados como una alternativa exitosa para el ajuste del controlador basado en métodos de optimización [4, 5, 6], ya que pueden manejar de manera eficiente las compensaciones altamente no lineales entre múltiples indicadores de rendimiento de lazo cerrado, e incorporan mecanismos (flexibilidad) para mejorar su convergencia y diversidad.

Una de las estrategias más utilizadas para evaluar el comportamiento del sistema de control, es el uso de indicadores de rendimiento, los cuales pueden medir características como la suavidad de la señal de control o la minimización del error mediante integraciones. Cabe recalcar que se puede considerar más de un indicador al mismo tiempo. Al plantearse más de un requerimiento o función objetivo en el diseño del controlador, obtenemos un problema de optimización multiobjetivo, pues se busca un conjunto de soluciones con distintos niveles de compromiso que cumpla con los requerimientos establecidos inicialmente.



Los problemas de optimización multiobjetivo se pueden resolver mediante la implementación de algoritmos metaheurísticos, como lo son la Evolución Diferencial, el Algoritmo Genético, y la técnica de Optimización por Enjambre de Partículas. Se proponen específicamente estos tres algoritmos debido a su uso extendido en la literatura y a su efectividad, además de realizar un estudio de la sintonización de controladores usando las técnicas metaheurísticas, para lograr una sintonización automática, tratando la sintonización como un problema de optimización multiobjetivo, y buscando que sea aplicable a sistemas mecatrónicos comunes, como son el péndulo, el péndulo doble, y el péndulo invertido. El propósito de usar los sistemas ya mencionados, es validar los resultados de manera sencilla, evaluando los objetivos mediante los indicadores de desempeño; ISE e IADU, por lo que en este proyecto se propone programar tres diferentes algoritmos evolutivos de uso recurrente en la literatura, que nos proporcionen las ganancias  $k_p, k_i, k_d$  adecuadas para satisfacer distintos niveles de compromiso que pueden presentarse en la sintonización de controladores PID. Como se ha mencionado con anterioridad, el péndulo simple, invertido y doble son sistemas usados con frecuencia en el sector industrial y por esta razón, han sido seleccionados para ejecutar un control de posición sobre ellos a través de una simulación computacional. Al mismo tiempo, se indaga en la implementación de un control PID para regular la posición de una barra que fungirá como brazo de un péndulo simple, a través del movimiento de un motor, y con ayuda de un sensor de posición, obtener los datos para su procesamiento por medio de una tarjeta de desarrollo. Los resultados obtenidos serán comparados con los de la simulación, buscando que el sistema sea estable en un tiempo reducido, así como minimizar el sobreimpulso y el error de las señales del sistema. Finalmente, se propone implementar una interfaz de usuario para los diseñadores de controladores, en la cual se podrán visualizar los resultados de la sintonización y una simulación 2D de la respuesta del sistema.



## 0.2. Justificación

En la mecatrónica, uno de los tópicos con mayor peso es la optimización, ya que la búsqueda de mejorar un determinado sistema es constante. Esto con el objetivo de aumentar la productividad y/o economizar algún proceso (generalmente industrial). Por lo tanto, el desarrollo de este estudio pretende aportar una relevancia significativa al área ya mencionada, de modo que se tratará a la sintonización de controladores como un problema de optimización. Dicho de otra forma, se trata de calcular o determinar el valor de las variables que intervienen en un sistema o proceso para minimizar o maximizar el valor de una función objetivo. Dado que el control y la automatización tienen un papel muy importante dentro de la mecatrónica, el implementar y corroborar nuevos métodos y técnicas para el diseño y sintonización de controladores puede incrementar la forma de controlar estos sistemas, pues al ser métodos comprobados brindan una mayor confianza al momento de implementarlos. En el planteamiento del problema, se mencionaron los sistemas que serán utilizados durante el estudio para comprobar el funcionamiento de la sintonización basada en optimización metaheurística multiobjetivo, por lo que es conveniente mencionar algunas aplicaciones que tienen estos sistemas o plantas, pues se encuentran en una amplia diversidad de contextos.

**Aplicaciones del péndulo simple:** En edificios para contrarrestar los fuertes vientos y posibles movimientos sísmicos, así como en puentes colgantes para contrarrestar las fuerzas del viento y movimientos telúricos [7].

**Aplicaciones del péndulo doble:** Sistemas con comportamiento caótico; atractores de Lorentz, fractales, circuitos RLC, entre otros [8]. Suele utilizarse también en la simulación de sismos e incluso en modelos utilizados en economía, biología, meteorología, etc [9].

**Aplicaciones del péndulo invertido:**

En [10] se ejemplifican diversas aplicaciones en las siguientes áreas:



En el sector aeroespacial para el control activo de un cohete durante su despegue. En biomecánica es frecuentemente utilizado para modelar bípedos caminantes y en transporte donde el control está basado en entradas sensoriales de giroscopios.

Por sí solos, estos sistemas son mecanismos, que, al agregar la parte de control, se convierten en sistemas más completos que tienen diversas aplicaciones en la industria, estos simples, pero muy representativos sistemas, al embeberse en sistemas más completos, conducen a excelentes resultados.

Para la codificación se seleccionó un lenguaje de alto nivel para desarrollar el proceso de sintonización. A continuación, se presentan algunas ventajas que presentan los lenguajes de alto nivel [11]:

Ventajas:

- Son más fáciles de leer, escribir y mantener por humanos.
- Se da forma a un código común que se podrá utilizar en diversos sistemas operativos y plataformas.
- Se pueden usar diferentes paradigmas de programación.
- Posibilidad de usar lenguajes interpretados.

### 0.3. Objetivo

Implementar mediante simulación tres métodos de sintonización para un controlador PID basados en optimización metaheurística multiobjetivo: Evolución Diferencial, Algoritmo Genético y Optimización por Enjambre de Partículas, (DE, GA Y PSO respectivamente, por sus siglas en inglés) en tres sistemas mecatrónicos comunes, mediante un estudio comparativo, con base en evidencia estadística, del comportamiento las técnicas metaheurísticas multiobjetivo ya mencionadas, y la aplicación física del algoritmo con mejor desempeño en un péndulo simple.



## Objetivos específicos

### Objetivos TT1

- Obtener los modelos en espacio de estados de los sistemas: péndulo simple, péndulo doble y péndulo invertido.
- Seleccionar del método numérico para simular los modelos en espacio de estados de los sistemas péndulo simple, péndulo doble y péndulo invertido.
- Seleccionar una estructura para el controlador PID, PD o PI.
- Determinar las variables de diseño del problema de optimización.
- Determinar las funciones objetivo del problema de optimización.
- Determinar las restricciones funcionales del problema de optimización.
- Determinar las restricciones de caja del problema de optimización.
- Plantear el problema de sintonización como un problema multiobjetivo.
- Determinar tres algoritmos metaheurísticos para sintonizar el controlador.
- Seleccionar técnicas estadísticas no paramétricas para evaluar el desempeño de los algoritmos metaheurísticos.
- Diseño del péndulo simple, circuito de control y acoplamiento.

### Objetivos TT2

- Implementar la simulación de los modelos en espacio de estados del péndulo simple, péndulo doble y péndulo invertido.
- Programar los algoritmos metaheurísticos para sintonizar el controlador.
- Resolver numéricamente los problemas de optimización para las variables de diseño.



- Obtener las aproximaciones de los frentes de Pareto correspondientes a cada una de las nueve sintonizaciones del controlador.
- Evaluar el desempeño de los algoritmos metaheurísticos sobre las aproximaciones de los frentes de Pareto.
- Implementación de una interfaz de usuario que permita la visualización del comportamiento de los diferentes algoritmos aplicados al controlador en un péndulo simple.
- Implementar el control en un péndulo simple.
- Implementar las ganancias de control que proporcionen el mejor compromiso entre los objetivos en un péndulo simple.

## 0.4. Antecedentes

Año	Nombre	Descripción	Objetivos	Controlador y variables de decisión	Restricciones	Referencia
2018	Tuning of Fractional Order PID Controllers using Evolutionary Optimization for PID Tuned Synchronous Generator Excitation System	Se utiliza un NSGA-II para resolver el problema de contradicción entre minimizar el error, escalamiento de la estabilidad robusta y la minimización del consumo de energía encuadrando controladores IO PID- FO PID como un problema multiobjetivo.	Integral Squared Error (ISE), Maximum Sensitivity (MS).	PID+FOPID: ganancias $k_p^j, k_i^j, k_d^j, j = 1, 2$ , <i>integral differential orders</i> ,	Condiciones de estabilidad de Routh.	[12]
2018	Control of refrigeration systems based on vapour compression using multi-objective optimization techniques.	Se hace uso de un algoritmo basado en DE para sintonizar las ganancias requeridas para un control de tipo PID usado en un sistema de refrigeración basado en compresión de vapor.	IAE (dos salidas), Total variation of the control signal (TV) (dos entradas), Log-Modulus (LM).	PID+PI: ganancias $k_p^j, k_i^j, j = 1, 2, k_d$ , <i>filtrof.</i>	-	[13]
2014	Design of a fractional order pid controller for hydraulic turbine regulating system using chaotic nondominated sorting genetic algorithm ii.	Se diseña un controlador PID fraccionario (FOPID, por sus siglas en inglés) apoyándose de un NSGA-II para un sistema regulador de una turbina hidráulica (HTRS). Las funciones objetivo son la integral del error cuadrático (ISE) y la integral del error cuadrático del tiempo multiplicado (ITSE), constituyendo de esta forma un problema de optimización multiobjetivo.	ISE e ITSE.	PID: ganancias $k_p, k_i, k_d$	-	[14]
2014	A novel design method of multiobjective robust pid controller for industrial process.	Se presenta un control PID robusto multiobjetivo, para un proceso no lineal en un reactor de tanque agitado continuo (CSTR, por sus siglas en inglés). Se busca que el control se robusto, limitado y un buen rendimiento de salida. Para esto, se emplea un algoritmo evolutivo; Optimización por Enjambre de Partículas Multiobjetivo (MOPSO, por sus siglas en inglés).	ISE, IAE, MS.	PID: parámetros $k_c, T_i, T_d, T_f$	Control máximo	[15]

Año	Nombre	Descripción	Objetivos	Controlador y variables de decisión	Restricciones	Referencia
2014	Design and performance analysis of pid controller for an avr system using multi-objective non-dominated shorting genetic algorithm-ii.	Se presenta el diseño y funcionamiento de un controlador PID para un sistema AVR usando NSGA-II , abordando el problema como uno de optimización multiobjetivo.	IAE, ITAE, ISE, ITSE.	PID: ganancias $k_p, k_i, k_d$	-	[16]
2011	Multi-objective pid controller tuning for a facts-based damping stabilizer using non-dominated sorting genetic algorithm-II.	Se estudia la aplicación de NSGA-II para la sintonización de un controlador PID para un Sistema Flexible de transmisión de CA (FACTS). El objetivo del diseño es mejorar la amortiguación del sistema de potencia cuando se somete a una perturbación con un esfuerzo de control mínimo.	SE index (para errores y señales de control).	PID: ganancias $k_p, k_i, k_d$	-	[17]
2015	Application of multiobjective controller to optimal tuning of pid gains for a hydraulic turbine regulating system using adaptive grid particle swam optimization.	Se emplea un algoritmo evolutivo multiobjetivo (AGPSO), para determinar las ganancias requeridas para sintonizar un HTRS. Se busca cuidar el tiempo de asentamiento y el nivel de sobreimpulso simultáneamente.	ISE, ITSE.	PID: ganancias $k_p, k_i, k_d$	-	[18]
2017	Multi-objective on-line optimization approach for the DC motor controller tuning using differential evolution.	Se propone la sintonización en línea de carácter multiobjetivo, usando algoritmos metaheurísticos, para poder ajustar los parámetros de control en la velocidad de un motor de imanes permanentes.	ISE, ISE.	Controlador por dinámica inversa: Parámetros dinámicos del motor CD.	Límites de la señal de control.	[19]
2008	A novel intelligent multiobjective simulated annealing algorithm for designing robust PID controllers.	Se hace uso de un algoritmo multiobjetivo para satisfacer las siguientes necesidades en un controlador PID: atenuación de perturbaciones, estabilidad robusta y seguimiento preciso del "setpoint".	MS, MCS, ISE.	PI: ganancias $k_p, k_i$	Límites superiores de MS y MCS.	[20]
2019	Adaptive controller tuning method based on online multiobjective optimization: A case study of the four-bar mechanism.	Se busca la sintonización de un controlador adaptativo basado en optimización metaheurística multiobjetivo para regular la velocidad en un mecanismo de cuatro barras.	ISE, suavidad de la señal de control.	Dinámica inversa del controlador: parámetros dinámicos de un mecanismo de 4 barras.	-	[21]





## 1.1. Marco Histórico

### Control Automático

El desarrollo de la humanidad puede reflejarse en sus avances tecnológicos y científicos. Entre los principales descubrimientos o inventos que han revolucionado la forma de vida del hombre, nos encontramos con la invención de la rueda, la imprenta, el desarrollo de la máquina de vapor, la electrónica y sus aplicaciones en la informática y la aparición de la red de redes, el internet.

El control automático representa una función vital en el avance de la ingeniería y la ciencia. Como primer trabajo importante en control automático, surge a finales del siglo XVIII el regulador de velocidad centrífugo de James Watt para el control de velocidad de una máquina de vapor [22].

Posterior a la Revolución Industrial, el diseño de sistemas de control realimentado estaba desarrollándose por medio de prueba y error junto con mucha intuición de la ingeniería [23]. En las matemáticas de mediados del siglo XIX, primero fue usado el análisis de la estabilidad de sistemas de control realimentado. Iba comenzando el siglo XX, cuando surgió la necesidad industrial de desarrollar instrumentos capaces



de medir y controlar presiones, temperaturas y otras variables. Con el paso del tiempo, se dieron grandes avances en conceptos y procesos de control, entre los cuales podemos destacar la demostración de Minorsky en 1922 de que la estabilidad de un sistema puede determinarse a partir de sus ecuaciones diferenciales, al mismo tiempo, se dio a conocer la importancia de un controlador PID. La función proporcional ya se conocía desde el comienzo del relé, sin embargo, la función integral no se conoció hasta 1920 y la derivativa hasta 1930. Durante esa década, Nyquist diseñó un procedimiento simple en 1932 para determinar la estabilidad en sistemas de lazo cerrado. 1940 fue un gran año para el mundo del control automático, ya que G Ziegler y N.B. Nichols ponen en el mercado el primer controlador PID: Fulscope modelo 100 [24]. Fue durante las décadas de 1960 y 1970, que se investigaron a profundidad tópicos sobre control óptimo, tanto de sistemas determinísticos como estocásticos, como de control adaptable, mediante el aprendizaje de sistemas complejos.

## Optimización

La optimización de problemas ha sido objeto de estudio desde el pasado y hasta nuestros días, ya que es un área activa de investigación. Debido a que la complejidad de los problemas de mundo real cada vez parece subir de nivel, ya sea en el campo de la ciencia o la ingeniería, es necesario contar con heurísticas capaces de resolver de forma rápida y eficiente estos problemas. En las últimas décadas, los algoritmos bio-inspirados han llamado la atención de los investigadores y han estado presentes en numerosas publicaciones (ver Antecedentes). Estos métodos son estocásticos y funcionan bien optimizando problemas en donde se requiere minimizar/maximizar una función, sin embargo, debido a que de forma global se busca una solución buena en un tiempo razonable, estos métodos no solucionan por sí solos el problema, pues únicamente se está optimizando un solo objetivo. Entonces, aquí es donde entra la optimización multiobjetivo basada en algoritmos evolutivos (MOEAs, por sus siglas en inglés) como una alternativa eficaz para optimizar problemas con más de un



objetivo, garantizando soluciones óptimas o semi-óptimas en un tiempo razonable [25].

## 1.2. Marco teórico

### Controladores automáticos

Los controladores automáticos comparan el valor real de la salida de una planta con la entrada de referencia (el valor deseado), determinan la desviación y producen una señal de control que reduce la desviación a cero o a un valor pequeño. La manera en la cual el controlador automático produce la señal de control se denomina acción de control. El controlador detecta la señal de error, que, por lo general, está en un nivel de potencia muy bajo, y la amplifica a un nivel lo suficientemente alto. La salida de un controlador automático alimenta a un actuador, como un motor, una válvula neumática, un motor hidráulico o un motor eléctrico. El sensor, o elemento de medición, es un dispositivo que convierte la variable de salida en otra variable manejable, como un desplazamiento, una presión o un voltaje, que pueda usarse para comparar la salida con la señal de entrada de referencia. En la Figura 1.1, se muestra el diagrama de bloques de un sistema de control industrial [26].

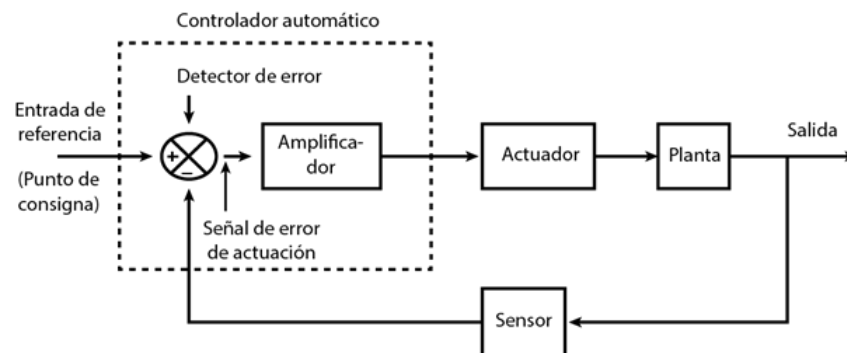


Figura 1.1: Diagrama de bloques de un sistema de control industrial. Fuente: Adaptado de [26].



*Clasificación de los controladores industriales.*

Los controladores usados en la industria pueden ser clasificados con base en sus acciones de control, como:

1. De dos posiciones o controladores on-off.
2. Controladores proporcionales.
3. Controladores integrales.
4. Controladores proporcionales, integrales, derivativos y sus variantes [26].

Este último tipo de controladores industriales se utilizan ampliamente en robots manipuladores industriales. Como se menciona en [27], las ventajas del controlador PID sobre los otros es que estos son simples y tienen significados físicos muy claros. Sin embargo, las ganancias del controlador PID deben ser sintonizadas para garantizar un buen desempeño. A pesar de no ser algo nuevo, este controlador es ampliamente usado en la industria, debido a su facilidad de implementación.

Algunos métodos de sintonización son los que se pueden consultar en [27]:

1. Modelo basado en sintonización analítica.
2. Métodos heurísticos.
3. Métodos en dominio de la frecuencia.
4. Métodos de optimización.
5. Métodos Adaptables.

En [27], se describen a los modelos heurísticos, como modelos inteligentes, basados en la experiencia, en este trabajo se propone tratar el problema de sintonización con metaheurística, lo que permite plantear el problema de sintonización como un problema de formal de optimización.



### **Acción de control proporcional**

El controlador proporcional, es, en esencia, un amplificador con una ganancia ajustable. Para un controlador con acción proporcional, la relación entre la salida del controlador  $u(t)$  y la señal de error  $e(t)$  es [26]:

$$u(t) = K_p e(t)$$

### **Acción de control integral**

Es un controlador con acción de control integral, el valor de salida del controlador  $u(t)$  se cambia a una razón proporcional a la señal de error  $e(t)$  [26]. Es decir:

$$\frac{du(t)}{dt} = K_i e(t).$$

o bien:

$$u(t) = K_i \int_0^t e(t) dt$$

### **Acción de control proporcional-integral**

La acción de control de un controlador proporcional-integral (PI) se define mediante [26]:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt$$

### **Acción de control proporcional-derivativa**

La acción de control de un controlador proporcional-derivativa (PD) se define mediante [26]:

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt}$$

### **Acción de control proporcional-integral-derivativa**

Esta acción combinada tiene las ventajas de cada una de las tres acciones individuales.



$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt}$$

O bien:

$$u(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right]$$

donde  $K_p$  es la ganancia proporcional,  $T_i$  es el tiempo integral y  $T_d$  es el tiempo derivativo [26].

## Métodos de integración

Los métodos de integración numérica generalmente son descritos como una combinación de evaluaciones del integrando para obtener una aproximación a la integral [28]. Durante el análisis de cualquier método de integración numérica, estudiar el comportamiento del error de aproximación como una función del número de evaluaciones del integrando es fundamental. Existen diversos métodos de integración, sin embargo el método de Euler destaca por ser uno de los más simples para resolver ecuaciones diferenciales ordinarias a partir de un valor inicial dado. Fue propuesto por Leonhard Euler en 1768. Se trata básicamente en aproximar la derivada por el cociente incremental en la ecuación mencionada:

Consiste en dividir el intervalo  $t_0, t_0 + \alpha$  en  $N$  partes iguales:

$$t_1 = t_0 + h, t_2 = t_0 + 2h, \dots, t_N = t_0 + Nh = t_0 + \alpha, h = \frac{\alpha}{N}$$

Aplicando la definición de la derivada:

$$y'(t_k) = \lim_{h \rightarrow \infty} \frac{y(t_k + h) - y(t_k)}{h},$$

Se deduce que para  $h$  "suficientemente pequeño":

$$y'(t_k) = f(t_k, y(t_k)) \simeq \frac{y(t_k + h) - y(t_k)}{h}$$



Por tanto:

$$y(t_{k+1}) \simeq y(t_k) + hf(t_k, y(t_k)), k = 0, 1, \dots, N - 1,$$

partiendo de  $y(0) = y_0$ . La ecuación anterior se conoce como método de Euler [29]. Este método resulta adecuado para desarrollar este trabajo.

## Dinámica

### Ecuaciones de Euler-Lagrange

El lagrangiano  $L(\mathbf{q}, \dot{\mathbf{q}})$  de  $n$  grados de libertad se define como la diferencia de la energía cinética  $K(\mathbf{q}, \dot{\mathbf{q}})$  y la energía potencial  $U(\mathbf{q})$ .

$$L(\mathbf{q}, \dot{\mathbf{q}}) = K(\mathbf{q}, \dot{\mathbf{q}}) - U(\mathbf{q})$$

Las ecuaciones de movimiento de Euler-Lagrange de  $n$  grados de libertad están dadas por:

$$\frac{d}{dt} \left[ \frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} \right] - \left[ \frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q}} \right] = \boldsymbol{\tau} - \mathbf{D}(\dot{\mathbf{q}}, \mathbf{f}_e)$$

donde  $\mathbf{q} = [q_1, q_2, \dots, q_n]^T \in \mathbb{R}^n$  representa el vector de posiciones articuladas o coordenadas generalizadas,  $\dot{\mathbf{q}} = [\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n]^T \in \mathbb{R}^n$  es el vector de velocidades articuladas,  $\boldsymbol{\tau} = [\tau_1, \tau_2, \dots, \tau_n]^T \in \mathbb{R}^n$  es el vector de pares aplicados, donde el  $i$ -ésimo par  $\tau_i$  se encuentra asociado con la  $i$ -ésima coordenada generalizada  $q_i$ , y  $\mathbf{D}(\dot{\mathbf{q}}, \mathbf{f}_e)$  es el vector de fuerzas o pares de fricción que depende de la velocidad articular  $\dot{\mathbf{q}}$  y de la fricción estática  $\mathbf{f}_e$  que se encuentra presente en las articulaciones del mismo;  $t \in \mathbb{R}_+$  representa el tiempo,  $n \in N$  el número de grados de libertad [30].

### Modelo dinámico

El modelo dinámico de un robot manipulador de  $n$  grados de libertad más ampliamente utilizado en el área de robótica esta dado por la siguiente ecuación [30]:

$$\boldsymbol{\tau} = M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{f}_f(\dot{\mathbf{q}}, \mathbf{f}_e)$$





O alternativamente:

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{q}} \\ M^{-1}(\mathbf{q}) [\tau - C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} - g(\mathbf{q}) - \mathbf{f}_f(\dot{\mathbf{q}}, \mathbf{f}_e)] \end{bmatrix}$$

Aquí se tiene que:

$\mathbf{q} \in \mathbb{R}^n$  es el vector de coordenadas generalizadas o posiciones articulares,

$\dot{\mathbf{q}} \in \mathbb{R}^n$  es el vector de velocidades articuladas,

$\ddot{\mathbf{q}} \in \mathbb{R}^n$  es el vector de aceleraciones articuladas,

$M(\mathbf{q}) \in \mathbb{R}^{n \times n}$ , es la matriz de inercia, la cual es simétrica, y definida positiva,

$C(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$  es la matriz de fuerzas centrípetas y de Coriolis,

$$C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} = \dot{M}(\mathbf{q}) \dot{\mathbf{q}} - \frac{\partial}{\partial \mathbf{q}} \left[ \frac{1}{2} \dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}} \right]$$

$\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$  es el vector fuerzas o pares gravitacionales, es decir:

$$\mathbf{g}(\mathbf{q}) = \frac{\partial \mathbf{U}(\mathbf{q})}{\partial \mathbf{q}}$$

debida a la acción de la gravedad,  $\mathbf{f}_f(\dot{\mathbf{q}}, \mathbf{f}_e) \in \mathbb{R}^n$  es el vector de pares de fricción viscosa, Coulomb, y estática ( $\mathbf{f}_e$ ) de cada articulación del robot [30].

## Cinemática

La cinemática es la parte de la física que aborda el problema de la descripción numérica del movimiento de sistemas mecánicos sin tomar en cuenta las fuerzas que lo producen.

**Cinemática directa:** Es una función vectorial que relaciona las coordenadas articulares  $\mathbf{q} \in \mathbb{R}^n$  con las coordenadas cartesianas  $[x, y, z]^T \in \mathbb{R}^3$  del robot  $\mathbf{f}_R : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , así como la orientación  $[\theta, \phi, \psi]^T \in \mathbb{R}^3$  de la herramienta colocada en el extremo final, tomando en cuenta las propiedades geométricas del sistema mecánico del robot.



**Cinemática inversa:** Es un problema no lineal que relaciona las coordenadas articulares en función de las coordenadas cartesianas. Dada la posición cartesiana y la orientación de la herramienta colocada en el extremo final del robot, obtener los ángulos de las articulaciones [30].

$$\mathbf{q} = \mathbf{f}_{\mathbf{R}}^{-1}(x, y, z, \tau, \phi, \psi)$$

### Cinemática diferencial

La cinemática diferencial directa es la derivada con respecto al tiempo de la cinemática directa:

$$\frac{d}{dt}[x \ y \ z \ \tau \ \phi \ \psi]^T = \begin{bmatrix} v \\ \omega \end{bmatrix} = \frac{d}{dt}\mathbf{f}_{\mathbf{R}}(\mathbf{q}) = \frac{\partial \mathbf{f}_{\mathbf{R}}(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} = J(\mathbf{q})\dot{\mathbf{q}}$$

El mapeo descrito por una matriz  $J(\mathbf{q} = \frac{\partial \mathbf{f}_{\mathbf{R}}(\mathbf{q})}{\partial \mathbf{q}} \in \mathbb{R}^{6 \times n})$  denominada jacobiano analítico:

$$J(\mathbf{q}) = \begin{bmatrix} J_v(\mathbf{q}) \\ J_\omega(\mathbf{q}) \end{bmatrix}$$

La cinemática diferencial inversa representa la relación entre la velocidad articular  $\dot{\mathbf{q}}$  con la velocidad lineal de movimiento  $v$  y la velocidad angular  $\omega$ , expresada en términos de la matriz inversa del jacobiano:

$$\dot{\mathbf{q}} = J^{-1}(\mathbf{q}) \begin{bmatrix} v \\ \omega \end{bmatrix}$$

donde  $J^{-1}(\mathbf{q}) \in \mathbf{R}^{6 \times n}$  es la matriz inversa del jacobiano, la cual existe si es una matriz cuadrada y su determinante es diferente de cero [30]..

### Regulación de posición

La tarea de regular posición consiste en hacer que el sistema se mueva desde una condición inicial de posición hasta una posición deseada final [30].



Formalmente, el objetivo del problema de control de posición está determinado por encontrar una ley de control  $\tau$  que proporcione los pares aplicados a los actuadores del sistema, de tal forma que la posición actual  $\mathbf{q}(t)$  y la velocidad articular de movimiento  $\dot{\mathbf{q}}(t)$  tiendan asintóticamente hacia la posición deseada  $\mathbf{q}_d$  y velocidad cero, respectivamente, sin importar las condiciones iniciales [30].

### Control de trayectoria

El control de trayectoria radica en determinar una función vectorial  $\tau$  de tal manera que las posiciones y las velocidades asociadas a las articulaciones del robot sigan con exactitud a las posiciones y velocidades deseadas, respectivamente.

El objetivo de control consiste en encontrar  $\tau$  tal que:

$$\lim_{t \rightarrow \infty} \begin{bmatrix} \tilde{\mathbf{q}} \\ \dot{\tilde{\mathbf{q}}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \in \mathbb{R}^{2n}$$

donde  $\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}} \in \mathbb{R}^n$  representa el error de posición definido entre la posición o trayectoria deseada  $\mathbf{q}_d(t)$  y la posición actual del robot  $\mathbf{q}(t)$ , es decir  $\tilde{\mathbf{q}} = \mathbf{q}_d(t) - \mathbf{q}(t)$  y error de velocidad definido como  $\dot{\tilde{\mathbf{q}}} = \dot{\mathbf{q}}_d(t) - \dot{\mathbf{q}}(t)$ , es decir entre la velocidad deseada de movimiento  $\dot{\mathbf{q}}_d(t)$  y la velocidad articular  $\dot{\mathbf{q}}(t)$  [30].



### 1.3. Marco conceptual

#### Diseño planteado como problema de optimización

Sin pérdida de generalidad, un problema de diseño puede plantearse como el siguiente problema de optimización.

$$\mathbf{J}(\mathbf{x}) = [J_1(x), \dots, J_m(x)]$$

Sujeto a:

$$g(x) \leq 0$$

$$h(x) = 0$$

$$\underline{x}_i \leq x_i \leq \bar{x}_i, i = [1, \dots, n]$$

Donde  $x \in R^n$  es el vector de variables de decisión,  $J(x) \in m$ , es el vector de objetivos de diseño,  $g(x), h(x)$  el vector de restricciones de desigualdad e igualdad respectivamente, y  $\underline{x}_i \leq x_i \leq \bar{x}_i$  cota de búsqueda del espacio de decisión para la variable  $x_i$  [31].

En la Figura 1.2 se muestra una metodología general para resolver un problema de diseño como un problema de optimización. A continuación se aborda cada una de las etapas mostradas.

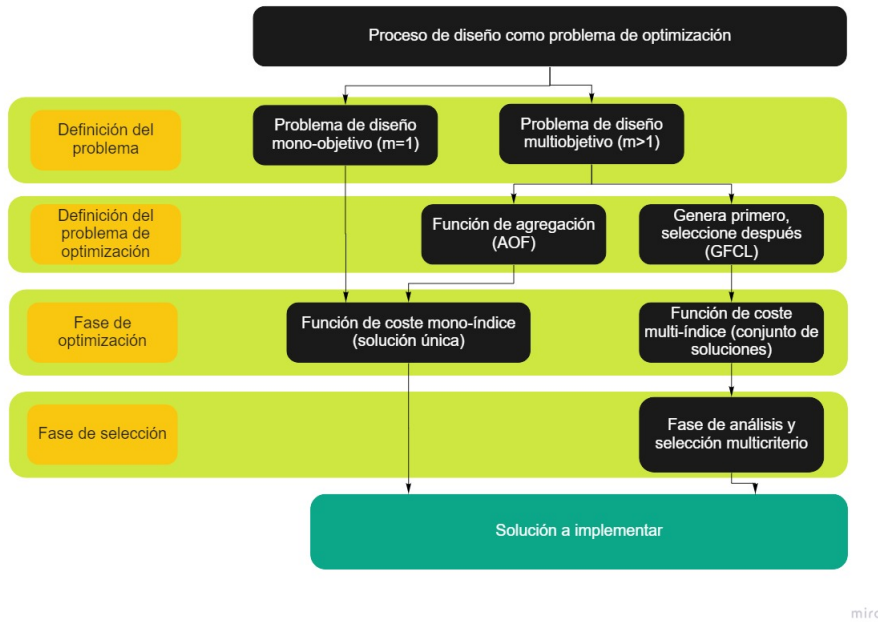


Figura 1.2: Metodología de diseño por medio de optimización.

Fuente: Adaptado de [31].

### Definición del problema

Implica identificar las variables de decisión, los objetivos de diseño y las restricciones (si las hay) del problema. Si  $m = 1$  se habla de un problema mono-objetivo (SOP por sus siglas en inglés) mientras que si  $m > 1$  se habla de un problema multiobjetivo (MOP por sus siglas en inglés).

### Definición del problema de optimización

Se refiere a la forma en que se traduce el problema anterior en una expresión que el optimizador pueda interpretar. El concepto de optimización de sistemas de control abarca dos etapas, una de selección de índices de rendimiento y otra de diseño con base en la minimización o maximización de dichos índices. En el caso de un SOP, el índice a optimizar será el mismo (y a la vez el único) objetivo de diseño. En el



caso de un MOP, su resolución se puede abordar desde dos perspectivas [32]: empleando funciones de agregación (Aggregate Objective Function; AOF) o calculando un conjunto de soluciones para seleccionar después la más indicada atendiendo a las preferencias del diseñador (Generate First, Choose Later; GFCL). En el caso de técnicas MOP/AOF, se busca especificar todos los requerimientos de diseño y las preferencias del diseñador en un solo índice. Con el planteamiento anterior el optimizador calcula una solución única para implementar. Por otra parte, la MOP/GFCL es una estrategia que busca determinar un conjunto de soluciones potenciales, para que el diseñador, a partir de ellas, elija la más indicada de acuerdo a sus preferencias de diseño. Lo anterior se justifica con el hecho de que, cuando se consideran simultáneamente todos los objetivos, generalmente no hay una solución mejor que las otras, pues no existe una solución única que mejore a todas las demás en todos los objetivos.

### Fase de optimización

La fase de optimización se refiere al uso/ejecución del algoritmo de optimización; los algoritmos evolutivos (EA's por sus siglas en inglés) se han utilizado como una alternativa válida para resolver estos planteamientos en diversos campos de la ingeniería [33]. En el caso de MOP/GFCL, se busca (por lo general) un conjunto de soluciones que aproxime lo que se conoce como conjunto de Pareto  $\mathcal{P}^*$ , y su imagen en el espacio de objetivos, frente de Pareto  $\mathcal{PF}^*$  [5].

### Fase de selección

La etapa de selección de la solución es exclusiva de la rama MOP/GFCL. La búsqueda de una solución con las prestaciones deseadas se efectúa (en general) *a posteriori*, analizando el conjunto de soluciones y el intercambio de sus prestaciones [31].



## Métodos metaheurísticos

Las metaheurísticas son una familia de algoritmos aproximados de propósito general. Suelen ser procedimientos iterativos que guían una heurística subordinada de búsqueda, combinando de forma inteligente distintos conceptos para explorar y explotar adecuadamente el espacio de búsqueda [34].

### Clasificación de los métodos metaheurísticos

En la actualidad se conocen varias metaheurísticas que se han desarrollado exitosamente en la resolución de determinados problemas, tales como: los algoritmos voraces, la ascensión de colinas, el recocido simulado, colonias de hormigas, algoritmos de enjambre, la búsqueda tabú, los algoritmos genéticos, los algoritmos meméticos, entre otros. No obstante, aunque suele ser una tarea poco sencilla, algunos autores han recopilado algunas formas en las que se pudieran clasificar las metaheurísticas. A continuación se presenta una posible forma de clasificación de las metaheurísticas [34]:

1. *Basadas en métodos constructivos*: Parten de una solución inicial vacía y se van agregando componentes hasta construir una solución. En este grupo se pueden mencionar: GRASP, optimización basada en colonias de hormigas.
2. *Basadas en trayectorias*: Utilizan como heurística subordinada cualquier algoritmo de búsqueda local, que sigue una trayectoria en el espacio de búsqueda, mediante iteraciones que tratan de remplazar una solución inicial por otra de mejor calidad. Allí se encuentran: búsqueda local, templado simulado, búsqueda tabú, búsqueda local iterativa, entre otras.
3. *Basadas en poblaciones*: El proceso contempla múltiples puntos de búsqueda en el espacio, que evolucionan en paralelo. Dentro de ellas están: algoritmos genéticos, algoritmos meméticos, algoritmos basados en nubes de partículas, búsqueda dispersa, entre otros.



## Algoritmos Evolutivos

Dentro de los métodos metaheurísticos basados en poblaciones, se encuentran los algoritmos evolutivos, también conocidos como optimizadores metaheurísticos. Estos son métodos estocásticos que están basados o bien en las leyes de la selección natural o inspirados por el comportamiento de ciertas especies.

Los optimizadores metaheurísticos se han usado ampliamente en problemas de optimización, debido a operación relativamente simple, su capacidad para manejar problemas complejos a un coste computacional razonable, así como su flexibilidad en variedad de contextos [35]. Además, son preferentemente usados en optimización multiobjetivo debido a su capacidad para obtener múltiples soluciones en el frente de Pareto en una sola ejecución completa del programa, así como aprovechar similitudes de soluciones mediante recombinación. El conjunto de soluciones no dominadas es llamado aproximación al conjunto de Pareto  $\mathcal{P}^A$ , estas soluciones en el espacio de objetivos forman la aproximación al frente de Pareto  $\mathcal{PF}^A$ . Las soluciones óptimas, que en la vida real son desconocidas, forman el conjunto de Pareto  $\mathcal{P}^*$  y en el espacio de objetivos forman el verdadero frente de Pareto  $\mathcal{PF}^*$  [31]. Es importante recalcar que los optimizadores metaheurísticos no garantizan encontrar el verdadero frente de Pareto, sino la aproximación al frente de Pareto a cambio de un coste computacional razonable [35]. En algoritmos evolutivos, la selección natural se simula mediante un proceso de selección estocástica.

Las estrategias más populares (para la ingeniería de control) en los que se basan los diferentes EA's parecen ser los Algoritmos Genéticos (GA), la Optimización por Enjambre de Partículas (PSO) y la Evolución Diferencial (DE). En este punto, es importante señalar que no existe una técnica evolutiva mejor que las otras, dado que cada una de ellas tiene sus ventajas y desventajas.

En la Figura 1.3 se puede observar de manera gráfica el proceso general de un algoritmo evolutivo.



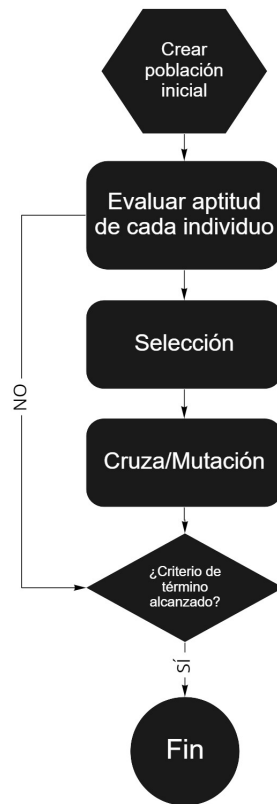


Figura 1.3: Diagrama de un EA. Fuente: Adaptado de Fuente: Adaptado de [36].

### Algoritmos genéticos (GA)

De acuerdo con la definición dada por Goldberg, se pueden describir formalmente de la siguiente manera: “Son algoritmos de búsqueda basados en la mecánica de selección natural y de la genética natural. Combinan la supervivencia del más apto entre estructuras de secuencias con un intercambio de información estructurado, aunque aleatorizado, para constituir así un algoritmo de búsqueda que tenga algo de las genialidades de las búsquedas humanas” [Goldberg, 1989]. Su desarrollo se debe en gran medida a John Holland, investigador de la Universidad de Michigan. A



finales de la década de los 60 desarrolló una técnica que imitaba en su funcionamiento a la selección natural.

Las soluciones factibles de un problema específico corresponden a los miembros de una especie particular, donde la aptitud de cada miembro ahora se mide por el valor de la función objetivo. El proceso general del algoritmo genético se puede ver en la figura 1.4

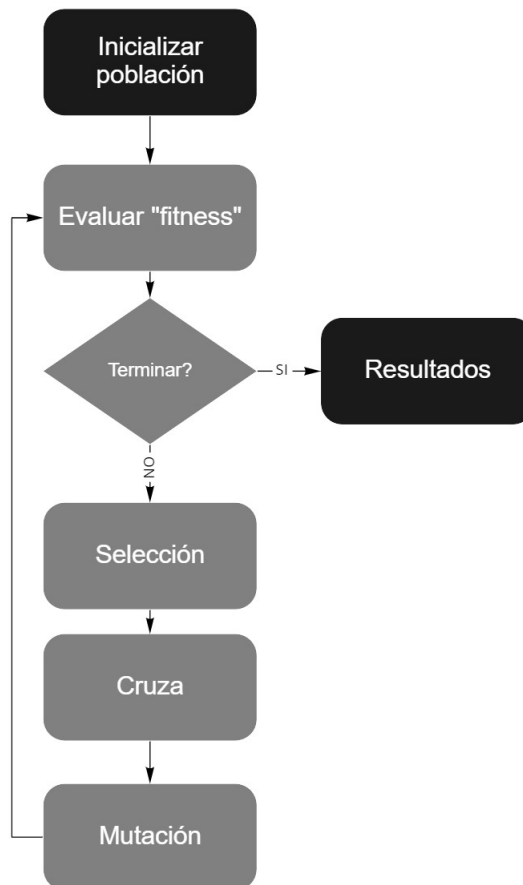


Figura 1.4: Diagrama de un GA.

Fuente: Adaptado de [36].



## Operadores genéticos

Para el paso de una generación a la siguiente se aplican una serie de operadores genéticos. Los más empleados son los operadores de selección, cruce y mutación. Estos se detallan a continuación.

**Selección:** El objetivo de los operadores de selección es conservar preferentemente las mejores soluciones, es decir, los individuos con mayor fitness. Los métodos más utilizados son los siguientes:

- Selección proporcional: Cada individuo tiene una probabilidad de ser seleccionado como padre que es proporcional al valor de su función objetivo
- Selección por torneo: La idea principal de este método consiste en realizar la selección en base a comparaciones directas entre individuos
- Selección elitista: Se garantiza la selección de los miembros más aptos de cada generación.

**Cruza:** Una vez seleccionados los individuos, estos son recombinados para producir la descendencia que se insertará en la siguiente generación. Los más empleados son los que se detallan a continuación:

- Cruce de un punto: Una vez seleccionados dos individuos, se cortan sus cromosomas seleccionado un punto aleatoriamente con el propósito de generar dos segmentos diferenciados en cada uno de ellos: la cabeza y la cola. Se intercambian las colas entre los dos individuos para generar los nuevos descendientes. De esta manera ambos descendientes heredan información genética de los padres.
- Cruce de dos puntos: Se trata de una generalización del cruce de 1 punto. Se realizan dos cortes en los los cromosomas de los padres.
- Cruce uniforme: La técnica implica la generación de una máscara de cruce con valores binarios. Si en una de las posiciones de la máscara hay un 1, el gen



situado en esa posición en uno de los descendientes se copia del primer padre. Si por el contrario hay un 0, el gen se copia del segundo padre. Para producir el segundo descendiente se intercambian los papeles de los padres, o bien se intercambia la interpretación de los unos y los ceros de la máscara de cruce.

- **Cruce Binario Simulado (SBX):** Se clasifica como un operador centrado en los padres y se basa en una función de distribución que usa el parámetro  $\eta$  (llamado índice de distribución y que debe ser fijado por el usuario) para determinar la apertura de la distribución, y por tanto, su capacidad de exploración. Si llamamos  $p_1$  y  $p_2$  a los padres, cuya media es  $\bar{p}$ , este operador consiste en generar de forma un parámetro  $\beta$  y después, crear los hijos  $c_1$  y  $c_2$  como:

$$\begin{aligned}c_1 &= \bar{p} - \frac{1}{2}\beta(p_2 - p_1) \\c_2 &= \bar{p} + \frac{1}{2}\beta(p_2 - p_1)\end{aligned}$$

**Mutación:** Proporciona un pequeño elemento de aleatoridad en la vecindad (entorno) de los individuos de la población.

- **Mutación polinomial:** dada una solución que actúa como padre, este operador genera soluciones cercanas con una probabilidad más alta que soluciones lejanas a él, independientemente de la iteración. Utiliza una distribución de probabilidad polinomial de la siguiente manera:

$$x_i^{(t+1)} = x_i^{(t)} + (x_i^{(U)} - x_i^{(L)})\delta_i$$

donde siendo  $u_i < 0.5$ , la  $\delta_i$  es igual a:

$$(2 \cdot u_i)^{\frac{1}{n+1}} - 1$$

y si  $u_i \geq 0.5$ , entonces  $\delta_i$  resulta:

$$1 - [2 \cdot (1 - u_i)]^{\frac{1}{n-1}}$$

$u_i \sim U[0, 1]$  y  $n$  es un parámetro que controla la variabilidad de la perturbación.



- Mutación uniforme: Este operador de mutación reemplaza el genoma, ya sea inferior o superior con destino al azar. Esto puede ser usado para los genes enteros y de coma flotante.
- Mutación no uniforme: Este operador hace, en las generaciones iniciales, una búsqueda uniforme en el espacio (exploración), pero esta búsqueda es más local en las generaciones finales (explotación).

En Algorithm 1, se puede ver el pseudo-código de un algoritmo genético básico para una optimización mono-objetivo.

---

**Algorithm 1** GA Básico
 

---

- 1: Generar población inicial  $P|_0$  con  $N_p$  cromosomas
  - 2: Evaluar  $P|_0$
  - 3: **while** Criterio de terminación no alcanzado
  - 4:   Leer contador de generaciones  $G$
  - 5:   Crear nueva población  $P|_G^*$  a partir de  $P|_G$
  - 6:   Seleccionar cromosomas padres
  - 7:   Efectuar el operador de cruce para crear un hijo a partir de los padres
  - 8:   Efectuar el operador mutación en el hijo
  - 9:   Seleccionar los cromosomas más aptos de  $P|_G^* \cup P|_G$  para generar a población  $P|_{G+1}$
  - 10:    $G = G + 1$
  - 11: **end while**
- 

## Evolución diferencial (DE)

El algoritmo de Evolución Diferencial [37] [38] [39] es utilizado principalmente por su simplicidad y compactibilidad [39]. Existen muchas versiones del algoritmo; la versión original [37] emplea tres operadores: mutación, recombinación y selección.

Es una técnica metaheurística basada en poblaciones de vectores numéricos.



En Algorithm 2, se muestra el pseudo-código para el algoritmo de DE básico para optimización mono-objetivo:

---

**Algorithm 2** DE Básico
 

---

- 1: Generar población inicial  $P|_0$  con  $N_p$  individuos
  - 2: Evaluar  $P|_0$
  - 3: **while** Criterio de terminación no alcanzado
  - 4:   Leer contador de generaciones  $G$
  - 5:   **do** Para cada  $\in P|_G$
  - 6:     Generar un vector mutante (1.1)
  - 7:     Generar un vector hijo (1.2)
  - 8:     Efectuar una comparación por pares Padre-Hijo (1.3)
  - 9:   **end do**
  - 10:    $G = G + 1$
  - 11: **end while**
- 

$$\mathbf{v}_{i,G} = \mathbf{x}_{r_1,G} + F(\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}) \quad (1.1)$$

$$u_{i,j,G} = \begin{cases} v_{i,j,G} & \text{si } \text{rand}(0, 1) \leq C_r \\ x_{i,j,G} & \text{de otra forma} \end{cases} \quad (1.2)$$

$$\mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G} & \text{si } f(\mathbf{u}_{i,G}) \leq f(\mathbf{x}_{i,G}) \\ \mathbf{x}_{i,G} & \text{de otra forma} \end{cases} \quad (1.3)$$

Donde  $x_{r_1,G}$  es el vector base,  $F$  el factor de escalamiento (entre el rango de  $0 \leq F \leq 1.2$ . Los autores aconsejan un valor en el rango de 0.4 a 1.0.) y  $x_{r_2,G}$   $x_{r_3,G}$  como los vectores diferenciales.  $C_r$  define la similaridad entre el vector trial y el vector mutante.

La variante de DE descrita con anterioridad es conocida como DE/rand/1/bin,



donde “rand” significa el criterio utilizado para la selección del vector base, “1” indica el número de vectores diferenciales empleados y “bin” es el tipo de cruce (binomial) y es empleada en este trabajo debido a que es la de mayor éxito y también la más utilizada.

Para la optimización multiobjetivo, la comparación se realiza mediante el concepto de dominancia. Este concepto será presentado en la sección de optimización multiobjetivo.

### Optimización por Enjambre de Partículas (PSO)

La optimización por enjambre de partículas (Particle Swarm Optimization) es un método de optimización heurística orientado a encontrar mínimos o máximos globales. PSO simula el comportamiento colaborativo de las especies en la búsqueda de recursos para su supervivencia. Cada partícula está determinada por su posición  $x$  y velocidad  $\dot{x}$  en la generación  $k$  [40] [41].

$$\dot{x}^j|_k = \dot{x}^j|_{k-1} + \varphi_1\beta_1(x^j|_{best}) + \varphi_2\beta_2(x^{swarm}|_{best})$$

$$x^j|_k = x^j|_{k-1} + \dot{x}^j|_k$$

$$x^j|_k = x^j|_{k-1} + \dot{x}^j|_k$$

donde  $\varphi_1, \varphi_2$  son factores empleados para ponderar el conocimiento local y global;  $\beta_1, \beta_2$  son números aleatorios; donde  $x^{swarm}|_{best}$  es la mejor posición (solución) conocida por el enjambre (conocimiento colectivo) mientras que  $x^j|_{best}$  es la mejor posición conocida para la partícula  $j$  (conocimiento individual). En Algorithm 3 se muestra el pseudo-código para un PSO básico.

**Algorithm 3** PSO Básico

- 
- 1: Generar el enjambre inicial  $P|_0$  con  $N_p$  partículas
  - 2: Evaluar  $P|_0$
  - 3: Inicializar para cada partícula su mejor posición conocida  $\mathbf{x}^j|_{best}$
  - 4: Inicializar la mejor posición conocida del enjambre  $\mathbf{x}^{swarm}|_{best}$
  - 5: Inicializar la velocidad de cada partícula  $\dot{\mathbf{x}}^j|_0$
  - 6: **while** Criterio de terminación no alcanzado
  - 7:     Leer contador de generaciones  $G$
  - 8:     **do** para cada partícula  $\mathbf{x}^j \in P|_G$
  - 9:         Actualizar su velocidad  $\dot{\mathbf{x}}^j|_G$
  - 10:         Actualizar su posición  $\mathbf{x}^j|_G$
  - 11:         Actualizar la mejor posición conocida  $\mathbf{x}^j|_{best}$
  - 12:         Actualizar el conocimiento colectivo del enjambre  $\dot{\mathbf{x}}^{swarm}|_{best}$
  - 13:     **end do**
  - 14:      $G = G + 1$
  - 15: **end while**
- 

**Topologías**

La topología es la que define el tipo de interacción en el enjambre. El algoritmo de PSO, en su versión canónica permite que todas las partículas se comuniquen entre sí, por lo que todas siguen a la que ha encontrado la mejor posición en algún momento de la búsqueda, a esta topología se le conoce como todos conectados (*full connected*). Existen otras topologías, como la de anillo (*ring*) en la que cada partícula se conecta con las partículas vecinas y es afectada solo por sus vecinos inmediatos. Por otra parte, en la topología de estrella (*star*), la información pasa a través de una sola partícula, ya que un nodo central se conecta con todas las demás.



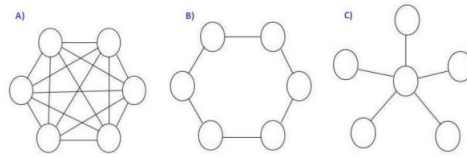


Figura 1.5: A) Full connected, B) Ring C) Star

Fuente: Adaptado de [42].

Adicionalmente, algunas de las características deseables de los EA's son [31]:

- Convergencia
- Diversidad de soluciones
- Manejo de preferencias
- Tratamiento de restricciones
- Escalabilidad
- Optimización en entornos dinámicos
- La optimización robusta
- Optimización de funciones con alto índice computacional

De la anterior lista, es importante señalar el tratamiento de restricciones (operación, implementación, físicas, entre otras). Una de las técnicas más populares para afrontar las restricciones son las *reglas de factibilidad*. Un ejemplo clásico es el presentado por Deb [43], el cual consiste en 3 reglas sencillas.

1. Al comparar dos soluciones factibles, aquella que tenga la mejor función objetivo es seleccionada.
2. Al comparar una solución factible con una no-factible, la primera es seleccionada sobre la segunda.



3. Al comparar dos soluciones no-factibles, aquella con el menor grado de violación de las restricciones es seleccionada.

## Optimización multiobjetivo (MOP)

Un MOP se expresa como:

$$\mathbf{min} \quad \vec{F}(\vec{p}) = [f_1(\vec{p}), \dots, f_m(\vec{p})]^T$$

sujeto a:

$$g_i(\vec{p}) \leq 0, i = 1, \dots, n_g$$

$$h_j(\vec{p}) = 0, j = 1, \dots, n_h$$

$$p_k^{min} \leq p_k \leq p_k^{max}, k = 1, \dots, d$$

donde  $\vec{p} = [x_1, \dots, x_d]^T$  es el vector de variables de diseño,  $\vec{F}$  de  $m > 1$  es el vector de funciones objetivo, sujeto a restricciones funcionales de la forma de  $g_i(\vec{p})$  y  $h_j(\vec{p})$ , restricciones de desigualdad e igualdad respectivamente. Las variables de diseño están limitadas por restricciones del mundo real, i.e.,  $p_k \in [p_k^{min}, p_k^{max}]$ . El espacio de soluciones factibles (donde se satisfacen todas las restricciones) se denota por  $\Omega$  [35].

### Dominancia de Pareto

Un vector  $\vec{F}(\vec{p}) = [f_1(\vec{p}), \dots, f_m(\vec{p})]^T$  se dice que domina a  $\vec{F}(\vec{q}) = [f_1(\vec{q}), \dots, f_m(\vec{q})]^T$  (denotado por  $\vec{F}(\vec{p}) \preceq \vec{F}(\vec{q})$ ) si y solo si  $\vec{F}(\vec{p})$  es tan bueno como  $\vec{F}(\vec{q})$  para todo los objetivos, i.e.,  $f_i(\vec{p}) \leq f_i(\vec{q}), \forall i \in \{1, \dots, m\}$ , y para al menos un objetivo  $f_i(\vec{p}) < f_i(\vec{q})$  [35].

### Óptimo de Pareto

Un vector de decisión  $\vec{p} \in \Omega$  es un óptimo de Pareto si no existe una función  $f_i(\vec{p})$  que pueda mejorar sin empeorar el resto, i.e.,  $\nexists \vec{q} \in \Omega$  tal que  $\vec{F}(\vec{p}) \preceq \vec{F}(\vec{q})$  [35].



### Conjunto óptimo de Pareto

El conjunto óptimo de Pareto  $\mathcal{P}^*$  contiene lo más posible el vector de decisión óptimo  $\vec{p} \in \Omega$ , i.e.,  $\mathcal{P}^* = \{\vec{p} \in \Omega \mid \nexists \vec{q} \in \Omega, \vec{F}(\vec{p}) \preceq \vec{F}(\vec{q})\}$ . [35]

### Frente de Pareto

El frente de Pareto contiene los valores de los vectores objetivo evaluados en  $\mathcal{P}^*$ , i.e.,  $\mathcal{PF}^* = \{\vec{F}(\vec{p}) \mid \vec{p} \in \mathcal{P}^*\}$ . [35]

### Algoritmos Evolutivos multiobjetivo (MOEA)

Los EA resultan muy atractivos para el análisis de problemas multiobjetivo, pues ofrecen grandes ventajas en comparación con los métodos clásicos, ya que, en la toma de decisiones multicriterio, existe un compromiso a valorar entre los diferentes objetivos.

En el algoritmo 4 se puede observar la estructura general del un EA multiobjetivo (MOEA, por sus siglas en inglés) básico [44].

De manera general, se pueden usar los algoritmos evolutivos mencionados anteriormente haciendo unas variaciones, de las cuales destacan el proceso de selección, donde ahora el proceso de selección se realiza usando el concepto de dominancia y las reglas de Deb. Adicionalmente, se usa un archivo donde solo se almacenan las soluciones factibles inmejorables, las cuales se analizan usando la distancia de apiñamiento (crowding distance), donde se cuida la diversidad de las soluciones, actualizando el archivo solo con las soluciones que arroje este análisis, pues el tamaño del archivo es menor al de la población.




---

**Algorithm 4** MOEA básico

---

- 1: Generar población inicial  $P|_0$  con  $N_p$  individuos
  - 2: Evaluar  $P|_0$
  - 3: Generar aproximación inicial de Pareto  $X_P^*|_0$
  - 4: Inicializar contador de generación en  $G = 0$  **while** *Criterio de convergencia alcanzado* **do**
  - 5:
    - end**
    - G=G+1
  - 6: Generar población  $P^*|_G$  usando  $P|_G$  y un EA
  - 7: Evaluar la nueva población  $P^*|_G$
  - 8: Generar aproximación al PF  $X_P^*|_G$  con  $X_P^*|_{G-1} \cup P^*|_G$
  - 9: Actualizar la población  $P_{G+1}$  con  $P^*|_G \cup P|_G$
  - 10:
  - 11: RETURN Aproximación al PF  $X_P^*|_G$
- 

## Índices de desempeño de un sistema controlado

Los índices de desempeño son una medida cuantitativa del desempeño de un sistema y se eligen de manera que se de énfasis a las especificaciones importantes del mismo [45]. Son utilizados en sistemas controlados para evaluar la respuesta del sistema frente a perturbaciones o cambios de parámetros. A continuación se mencionan algunos de los más usados:

### **Integral Absolute Error Criterion (IAE)**

Un sistema óptimo basado en este criterio es un sistema que posea un amortiguamiento razonable y una respuesta transitoria satisfactoria [46].

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



### **Integral of the Absolute value of the control action (IADU)**

Se caracteriza por indicar la suavidad de la señal de control[47].

$$IADU = \int_0^T |u(t) - u(t-1)|dt$$

### **Integral Time Absolute Error (ITAE)**

Se caracteriza por ser un criterio muy selectivo pero difícil de evaluar analíticamente [45].

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

### **Integral square-error criterion(ISE)**

Un sistema se considera óptimo si consigue minimizar este índice de desempeño. En particular el ISE se usa con frecuencia cuando se implementan entradas determinísticas (escalón) y entradas estadísticas dado a su facilidad de implementación tanto digital como analíticamente [45]. Este criterio está definido por:

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

## **Indicadores de Calidad**

Durante el proceso de diseño MOO, existen algunas consideraciones que deben de presentar los indicadores de rendimiento Podemos categorizar a estos indicadores MOO en cuatro grupos principales [48]. Siendo  $S$  el conjunto solución óptimo:

- **Indicadores de capacidad:** Las métricas de este grupo cuentan el conjunto de soluciones no dominadas en  $S$  que satisface a los requisitos predefinidos.
- **Indicadores de convergencia:** Estas son métricas para medir la proximidad del conjunto de solución óptima en el FP.



- Indicadores de diversidad: Estas métricas indican dos tipos de información: 1) La distribución y 2) la propagación
- Indicadores de convergencia-diversidad: Indican tanto la convergencia como diversidad de  $S$  en una sola escala.

Para este trabajo, estamos buscando medir convergencia y diversidad en un frente dado, por lo que haremos uso de alguna métrica del cuarto grupo mencionado. Por esta razón, usaremos la métrica llamada Hipervolumen (HV), que está definida como:

$$HV(S, R) = \text{volume} \left( \bigcup_{i=1}^{|S|} v_i \right)$$

Donde  $R = \{W\}^2$

Matemáticamente, para cada  $i \in S$  se construye un hipercubo  $v_i$  con un punto de referencia  $W$  y la solución  $i$  que definen la diagonal del mismo.

Los algoritmos que alcanzan mayores valores para HV son mejores. En la Figura 1.6 se puede observar esta métrica.

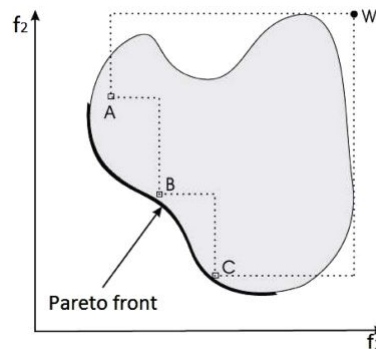


Figura 1.6: Hipervolumen (HV).

Fuente: Adaptado de [49].



## Pruebas no paramétricas

Son aquellas que no presuponen una distribución de probabilidad de datos, se conocen también como de distribución libre [50].

### Pruebas del signo

Usa los signos + y - para establecer una diferencia en función de la dirección.

Esta prueba es aplicable para el caso donde se tengan dos muestras relacionadas y se requiere mostrar que ambas condiciones son diferentes. El único requisito a la prueba es la continuidad a la variable considerada [51].

### Procedimiento de la prueba

- Se determina el signo de diferencia entre los dos datos de cada pareja (ordenados previamente).
- Se determina el valor de “n”, es decir, el número de parejas cuyas diferencias exhiben un signo, aquellas parejas con igual valor tienen una diferencia de cero, por lo que no deben ser consideradas en el conteo de signos.
- Se formula el planteamiento de la hipótesis.
- Se aplica la aproximación a una distribución normal con el resto de los pasos de una prueba de hipótesis [51].

### Prueba de rango con signo de Wilcoxon

Se utiliza comúnmente como alternativa a la prueba paramétrica t de Student. La prueba de Wilcoxon con rangos de signo muestra que tan grande o pequeña es una muestra frente a una mediana hipotética [51].



### Procedimiento de la prueba

- Mediante la diferencia, se le asigna un signo a cada pareja, donde el orden ya se ha establecido, es decir  $z_i = y_i - x_i$  donde  $i = 1, 2, 3, \dots, n$ .
- Obtener un grupo de  $n$  diferencias absolutas, sin tomar en cuenta los signos.
- Se omiten los pares con diferencia absoluta 0, por lo que se forma un conjunto de  $\acute{n}$ , donde  $\acute{n} \leq n$ .
- Ordenar de manera incremental las diferencias absolutas entre pares de valores, esto con el fin de asignarle un rango  $R_i$  de 1 hasta  $\acute{n}$ , es posible que se repitan valores, entonces se les asigna el rango promedio de sus posiciones.
- Una vez ordenados, se le asigna en su rango el signo que le corresponde a cada valor de los  $\acute{n}$  valores, para posteriormente sumar los rangos positivos y negativos por separado.

Se suman los rangos positivos ( $W^+ = \sum R_i(+)$ ) y negativos ( $W^- = \sum R_i(-)$ ).

La prueba se fundamenta en la hipótesis nula  $H_0 : \mu_A = \mu_B$ , eso significa que los totales de las sumas de los rangos positivos y negativos son aproximadamente iguales, en caso contrario la hipótesis alternativa esta denotada por  $H_1 : \mu_A \neq \mu_B$ , lo que significa es, si ambos totales ( $W^+, W^-$ ) son pequeños en una prueba de dos extremos [51]. En otras palabras:

$H_1 : \mu_A < \mu_B$ , sucede si  $W^+$  es pequeña y  $W^-$  es grande.

$H_1 : \mu_A > \mu_B$ , sucede si  $W^+$  es grande y  $W^-$  es pequeña.

### Método Analítico Jerárquico (AHP)

Desarrollado por Thomas L. Saaty en 1980, se trata de un procedimiento de comparación por pares de los criterios que parte de una matriz cuadrada en la cual el número de filas y columnas está definido por el número de criterios a ponderar.





De esta manera, se establece una matriz de de comparación entre pares de criterios, comparando la importancia de cada uno de ellos con los demás. Después, se establece el eigenvector principal, el cual establece los pesos que paralelamente proporcionan una medida cuantitativa de la consistencia de los juicios de valor entre pares de factores [52]. Este método requiere que quien toma las decisiones proporcione evaluaciones subjetivas respecto a la importancia relativa de cada uno de los criterios y que después especifique su preferencia con respecto a cada una de las alternativas de desición y para cada criterio. El resultado es una jerarquización con prioridades que muestran la preferencia global para cada una de las alternativas de decisión [53]. Algunas de las ventajas de AHP frente a otros métodos de decisión multicriterio son:

- Permite analizar y descomponer un problema por partes.
- Presenta un sustento matemático.
- Permite medir criterios cuantitativos y cualitativos mediante una escala común.

Este método utiliza comparaciones pareadas como base fundamental, utiliza una escala subyacente con valores de 1 a 9 para calificar las preferencias relativas de los dos elementos. Saaty [52] establece una escala numérica del 1 al 9, donde dos elementos que se están comparando pueden tener diferentes niveles de importancia:

- 1 : Igualmente importante.
- 3 : Ligeramente más importante.
- 5 : Notablemente más importante.
- 7 : Demostrablemente más importante.
- 9 : Absolutamente más importante.

Los valores 2,4,6 y 8 son valores intermedios.



## 1.4. Marco procedimental

El propósito principal de los controladores es garantizar la estabilidad de la respuesta dinámica del sistema. El problema general de sintonización del controlador se refiere a la búsqueda de un conjunto de parámetros del controlador que establezca la respuesta dinámica del sistema bajo un criterio de rendimiento establecido que se ajuste a las necesidades anteriores [35].

### Sintonización de controladores multiobjetivo

Las aplicaciones de ingeniería se han vuelto cada vez más exigentes a medida que pasa el tiempo, ya que requieren sistemas dinámicos que satisfagan necesidades diferentes y a menudo conflictivas. Los pasos habituales para la sintonización del controlador multiobjetivo son descritos en [35] y se muestran en la figura 1.7.

A continuación se describe brevemente cada uno de los pasos mostrados en la figura [35]:

1. **Estudiar las características de la planta:** Analizar el número de entradas y salidas, comportamiento lineal o no lineal y la naturaleza de la actividad a realizar.
2. **Seleccionar una estructura para el controlador:** Escoger una estructura apropiada para el controlador, de tal manera que el comportamiento de la planta alcance la estabilidad.
3. **Determinar los parámetros de sintonización para el controlador:** Toda estructura de controlador conlleva parámetros directamente relacionados con el comportamiento de la planta. Es de suma importancia que el diseñador determine dichos parámetros.

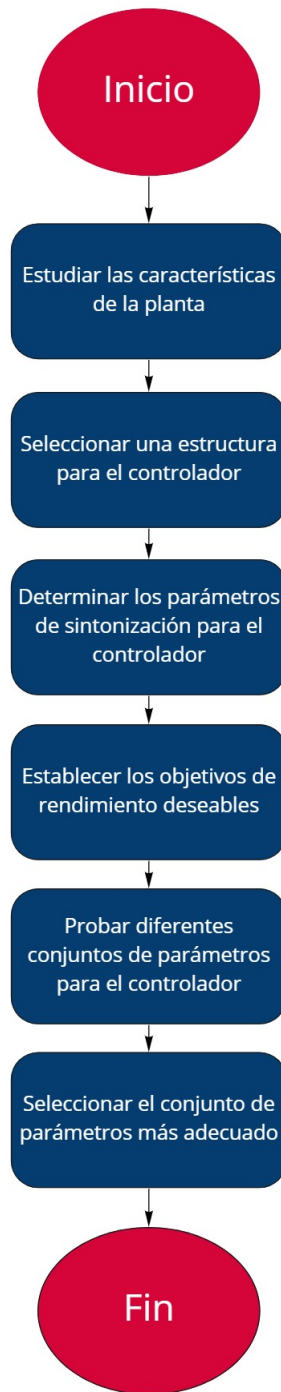


Figura 1.7: Pasos típicos para la sintonización de controladores multiobjetivo.

Fuente: Adaptado de [35]



4. **Establecer los objetivos de rendimiento deseables:** Estos criterios son los que definen el problema multiobjetivo y están directamente relacionados con la minimización o maximización de indicadores cuantitativos.
5. **Probar diferentes conjuntos de parámetros para el controlador:** Los métodos metaheurísticos buscan un conjunto de soluciones candidatas que satisfagan distintos niveles de compromiso entre los objetivos propuestos.
6. **Seleccionar el conjunto de parámetros mas adecuado:** Se selecciona una de solución de acuerdo con el criterio del diseñador, para ser implementada.

### Optimización multiobjetivo con métodos metaheurísticos para la sintonización de controladores

En la Figura 1.8, se presenta un esquema de un MOP de sintonización de controladores basado en diferentes criterios de desempeño, donde se utiliza un optimizador metaheurístico para encontrar los parámetros del controlador con los mejores compromisos. Una vez que se encuentran los mejores compromisos, el diseñador es responsable de elegir un solo parámetro de configuración a implantar en el controlador real [35].

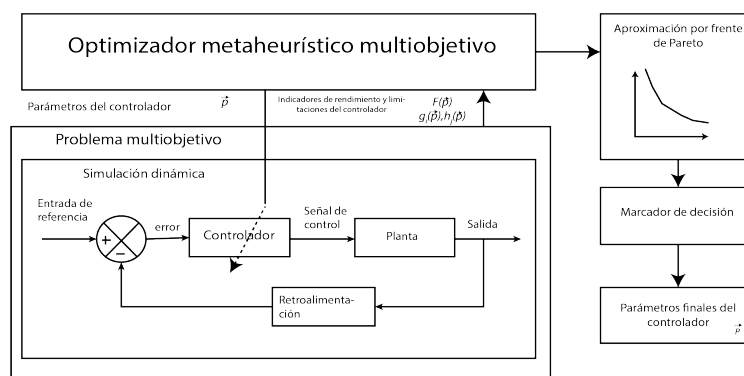


Figura 1.8: MOP optimizado con un método metaheurístico.

Fuente: Adaptado de [35].



## Metodología

En este proyecto se propone usar la norma VDI 2206, la cual está destinada al desarrollo de productos mecatrónicos. Esta norma propone un modelo en V, y se basa en la concurrencia para el diseño de sistemas multifuncionales [54]. La Figura 1.9 representa la metodología propuesta en forma general.

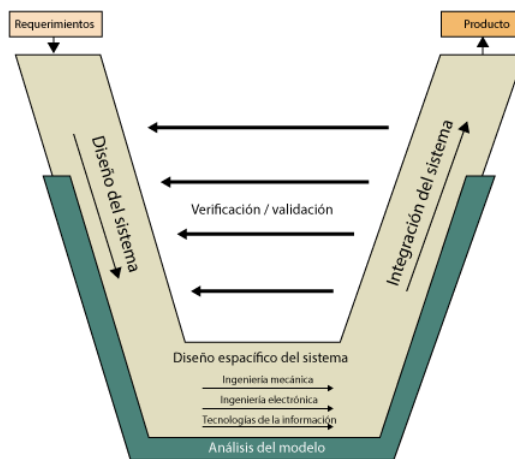


Figura 1.9: Modelo en V.

Fuente: Adaptado de [54].

El proceso consta de tres fases.

- Diseño del sistema
- Diseño específico
- Integración del sistema

Estas fases se llevan a cabo de manera paralela, a partir de los requerimientos propuestos. Las principales tareas de la norma VDI 2206 son:

1. Definición de requisitos: En esta etapa se investigan, analizan y enlistan los requerimientos más representativos del proyecto.



2. Diseño preliminar del sistema: Durante esta tarea se realiza la división de los subsistemas del proyecto.
3. Diseño específico del sistema: Al desarrollar cada subsistema se obtiene su completo dominio y entendimiento para permitir cualquier cambio o resolución del problema si es que fuera necesario.
4. Integración del sistema: El diseño del proyecto se basa en trabajar por partes pensando en su unión final para evitar problemas con la integración.
5. Verificación / Validación: Se realiza una validación constante para generar retroalimentación.
6. Modelado y análisis del modelo: Se realizará un modelado del sistema a desarrollarse y analizará su funcionalidad, desempeño y mejoras.

Generalmente un producto mecatrónico complejo requiere de varios ciclos, como se puede observar en la Figura 1.10.

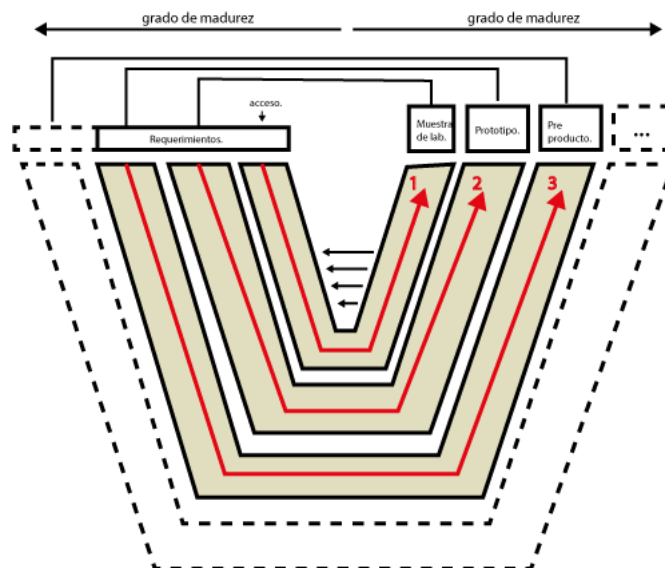


Figura 1.10: Macro ciclos y madurez del producto.

Fuente: Adaptado de [54].



## **2.1. Diseño conceptual**

En esta sección, se presentarán el proceso de diseño de concepto del sistema. Es decir, se realizará la identificación de necesidades y requerimientos con el fin de buscar y presentar una propuesta de concepto solución que satisfaga estas necesidades, posteriormente se realizará la validación del concepto y se presentará el concepto solución final.

### **Necesidades y requerimientos**

Es necesario identificar las necesidades que debe resolver el sistema. Las necesidades que se lograron plantear para este trabajo son las que contiene la Tabla 2.1.





Tabla 2.1: Necesidades del sistema.

Identificador	Necesidad
N1	Sintonización de controladores PID con métodos metaheurísticos.
N2	Agilizar el proceso de la sintonización.
N3	Mostrar en un frente de Pareto el conjunto de soluciones entre los distintos niveles de compromiso de las funciones objetivo.
N4	Mostrar en una tabla el conjunto de soluciones entre los distintos niveles de compromiso de las funciones objetivo.
N5	Una interfaz intuitiva.
N6	Cumplir con tarea de regulación de posición.
N7	Una señal de control suave.
N8	Mostrar una simulación del resultado esperado.

Los requerimientos del sistema funcionan como indicadores de las especificaciones del sistema, los requerimientos que se identificaron se presentan en la Tabla 2.2.

Tabla 2.2: Requerimientos.

Identificador	Requerimientos	Valor
R1	Error	Bajo
R2	Suavidad	Alto
R3	Tiempo de ejecución	Adecuado
R4	Interfaz de usuario	Intuitiva
R5	Métodos metaheurísticos	3
R6	Consumo energético	Bajo
R7	Precisión	Alta
R8	Coincidencia en respuesta	Alta

### Validación de requerimientos

En la Tabla 2.3 se muestra la verificación de necesidades con los requerimientos, pues de esta manera se valida que cada requerimiento se asocie al menos con una necesidad.



Tabla 2.3: Matriz de trazabilidad.

	N1	N2	N3	N4	N5	N6	N7	N8
R1	X					X	X	
R2	X	X					X	
R3		X						X
R4			X	X	X			X
R5	X			X				
R6						X		
R7						X	X	X
R8	X		X	X		X	X	X

## Arquitectura funcional

En esta sección se realizará la división por funciones, en funciones secundarias, esto con el propósito de hacer el proceso de diseño del concepto más claro. Para llevar a cabo esto, se utilizará la metodología gráfica IDEF, la cual consiste en el modelado de procesos y es utilizada en la implementación de sistemas y software en ingeniería. La Tabla 2.4 enumera los 16 métodos IDEF, siendo IDEF0 a IDEF4 los métodos más utilizados [55] .

Tabla 2.4: Familia IDEF.

IDEF0	Modelado De Funciones
IDEF1	Modelado De Información
IDEF1X	Modelado De Datos
IDEF2	Diseño De Simulación De Diseño
IDEF3	Captura De Descripción De Procesos
IDEF4	Diseño Orientado A Objetos
IDEF5	Captura De Descripción Ontológica
IDEF6	Diseño De Captura Racional
IDEF7	Revisión De Cuentas De Información Del Sistema
IDEF8	Modelado De Interfaz De Usuario
IDEF9	Diseño De Escenarios
IDEF10	Modelado De Artefactos De Implementación
IDEF11	Modelado De Artefactos De Información
IDEF12	Modelado De Organización
IDEF13	Diseño De Mapeo De Tres Esquemas
IDEF14	Diseño De Redes



Dado que nos interesa el modelado de funciones, IDEF0 resulta ser la metodología mas adecuada. En la Figura 2.1 se describe la función general en un diagrama IDEF0.

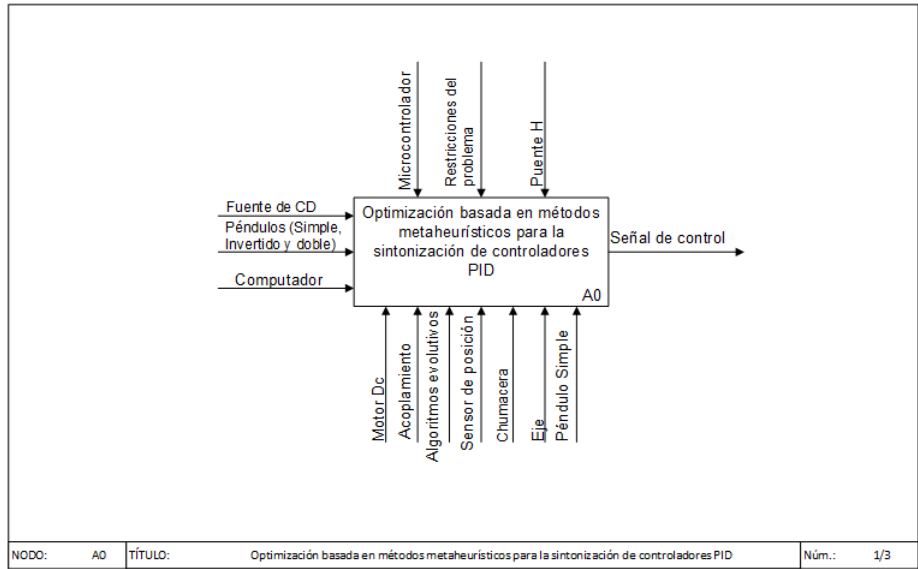


Figura 2.1: Diagrama IDEF0.

Para la sintonización de controladores PID es necesario el modelado de las plantas, convertir las características de la planta en un problema multiobjetivo, y para una verificación de los resultados se implementará el control en el péndulo físico de manera física. Las funciones se presentan a continuación:

A1. Modelado del sistema

El modelado del sistema permite tomar y verificar que la ecuación obtenida de la dinámica de cada péndulo sea correcta. Para esto se implementará una simulación 2D del sistema en Python, así como el monitoreo de las variables de control ya establecidas en el modelo dinámico del sistema.

A2. Sintonización con métodos metaheurísticos para optimización multiobjetivo.

La sintonización de los sistemas de control PID se realizará con algoritmos evolu-



tivos, para eso es necesario plantear el problema de optimización, posteriormente resolver estos problemas con 3 algoritmos evolutivos. Los resultados se presentarán en frentes de Pareto, los cuales proporcionan un conjunto de soluciones que cumplen con diferentes niveles de compromisos con las funciones objetivo.

### A3. Interfaz visual

Esta función cumple con la acción de presentar los resultados en una interfaz de usuario donde estarán presentes los frentes de Pareto, así como los valores de las variables de diseño. Esto permitirá la selección de las ganancias de control que se adapten de mejor manera a las necesidades en cada péndulo.

### A4. Adaptación del péndulo simple

El péndulo simple debe de ser adaptado para que se pueda mover con un actuador.

### A5. Control del péndulo simple

Se implementará el control de regulación de posición del péndulo, pues ya se tienen los valores de las ganancias del controlador se debe verificar que los valores calculados cumplan con los objetivos deseados.

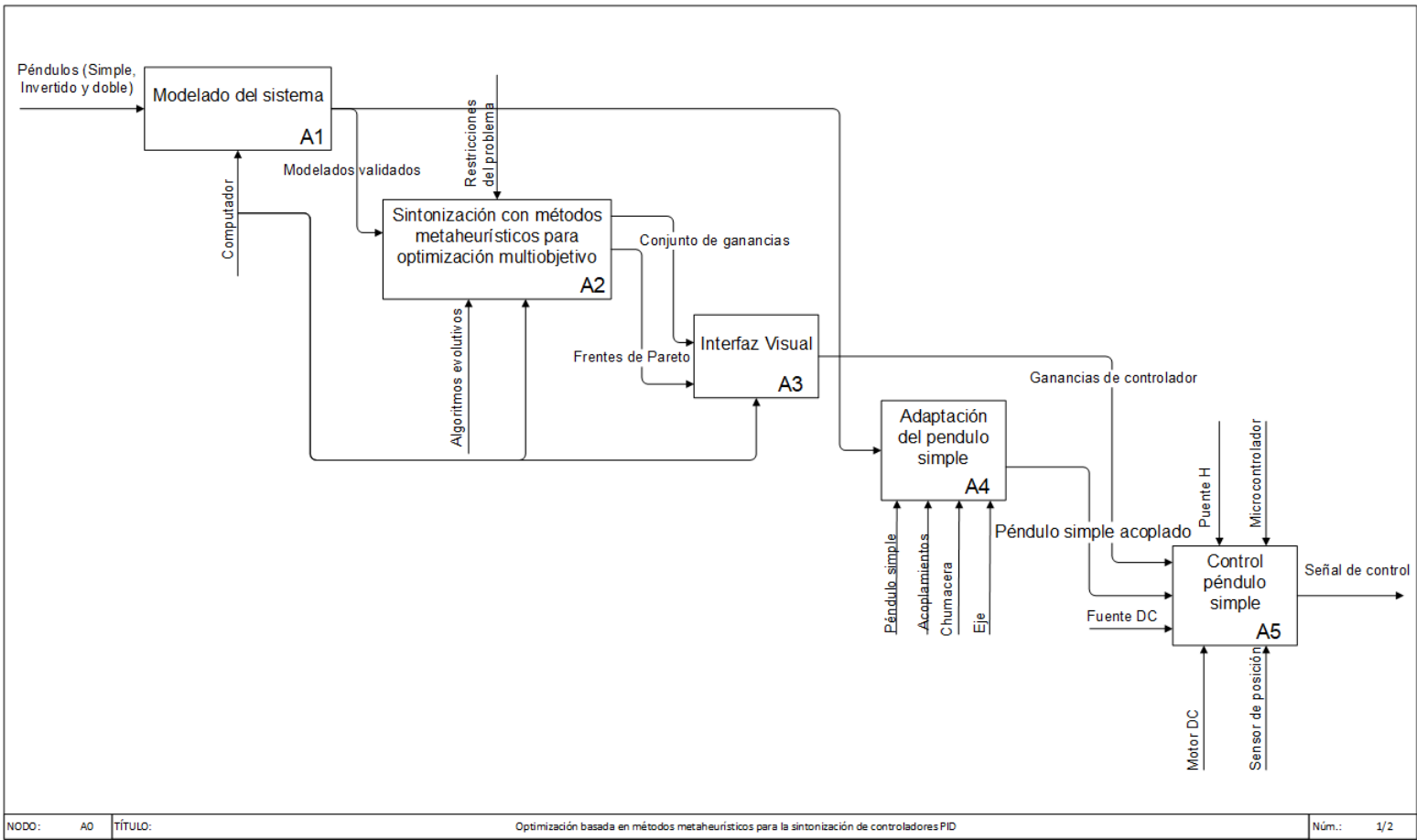


Figura 2.2: Diagrama IDEF0 Nodo A0.



En la Figura 2.2 se muestra en mayor detalle el nodo A0, las funciones que componen el diseño, así como la relación que tienen entre cada una de ellas.

Es necesario descomponer el bloque A2 del diagrama IDEF0 para una mejor comprensión de esta función, es por ello que, la Figura 2.3 muestra la descomposición de este bloque.

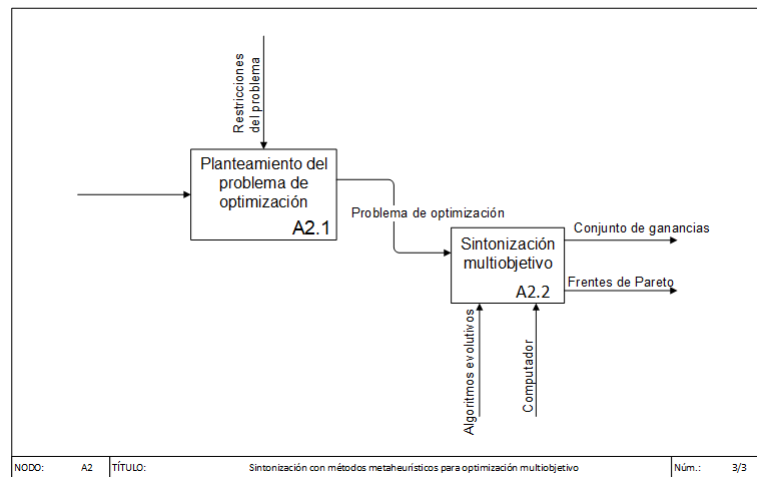


Figura 2.3: Diagrama IDEF0 Nodo A2.

### Validación de funciones

Una vez que se han establecido las funciones que el sistema debe de llevar a cabo, es importante realizar una matriz de trazabilidad entre las funciones y los requerimientos del sistema, esto se puede observar en la Tabla 2.5. La interfaz de usuario (R4) debe de mostrar la información más relevante del proceso de control, es por ello que este requerimiento, junto con R5 Y R8 están ligados con las funciones A1 y A3, puesto que ambas están enfocadas a la visualización de datos arrojados una vez que se ha ejecutado uno de los algoritmos evolutivos en busca de la sintonización multiobjetivo. Por otra parte, se busca que el sistema sobre el cual se probará la sintonización (A2) fuera de línea (A4), es decir, el sistema de péndulo simple, no requiera de una gran cantidad de energía para funcionar(R6). Uno de los aspectos



más importantes a evaluar, es que el resultado obtenido a través de las simulaciones realizadas en Python, coincida (R8) con los datos recolectados mediante un sensor implementado en el sistema físico, observando datos relevantes acerca del error (R1) y de la suavidad de la señal de control (R2)(A5). Por último, se espera que el costo computacional (R3) concuerde con la complejidad del problema de optimización.

Tabla 2.5: Matriz de trazabilidad para la validación de funciones.

	R1	R2	R3	R4	R5	R6	R7	R8
A1	X	X	X	X			X	X
A2	X	X	X		X			X
A3				X			X	X
A4						X		X
A5	X	X					X	

## Arquitectura física

### S1 Robótico

- Manipulador: Elemento que fungirá como brazo robótico de 1 grado de libertad (1GDL).
- Movimiento: Elementos que en conjunto permitan al sistema llevar a cabo una función específica.

### S2 Información

- Percepción: Sensores cuyo propósito es obtener información del sistema, para posteriormente ser procesada y tomar decisiones con base en ella.
- Comunicación: Dispositivos que permitan realizar la transmisión y recepción de datos.
- Procesamiento: Componentes electrónicos que permitan analizar la información recopilada a través de los sensores implementados.



S3 Administración energética

- Fuente de alimentación que permita a los circuitos funcionar y realizar la tarea especificada para cada uno.

**Matriz morfológica**

Una matriz morfológica es una herramienta usada para generar conceptos de diseño, combinando opciones solución con funciones inherentes al sistema [56]. En la Figura 2.4 se muestra la matriz morfológica que será de utilidad para establecer rutas de diseño para el mecanismo.

FUNCIÓN	OPCIÓN 1 (O1)	OPCIÓN 2 (O2)	OPCIÓN 3 (O3)
PERCEPCIÓN DE DATOS (F1)			
MOVIMIENTO DEL EJE (F2)			
ADMINISTRACIÓN DE ENERGÍA (F3)			
CIRCUITO DE CONTROL (F4)	VISTA SUPERIOR 	VISTA SUPERIOR 	VISTA SUPERIOR 

Figura 2.4: Matriz morfológica.





## Búsqueda morfológica

- Ruta 1: O1F1-O1F2-O3F3-O3F4

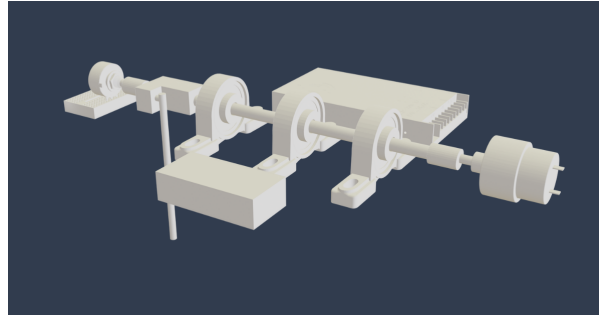


Figura 2.5: Ruta 1.

Para la Ruta 1 que se observa en la Figura 2.5, se propone usar un motor en uno de los extremos, mientras que en el extremo opuesto se encuentra un sensor. Por otra parte, para mantener el eje fijo en traslación, se utilizarían tres chumaceras situadas a lo largo del mismo. Con esta distribución, la fuente de alimentación para el motor y circuito de control estaría situada a uno de los costados. Por último, la tarjeta de desarrollo estaría en otro de los costados, esperando que el computador se encuentre cerca de la ya mencionada.

- Ruta 2: O2F1-O3F2-O1F3-O2F4

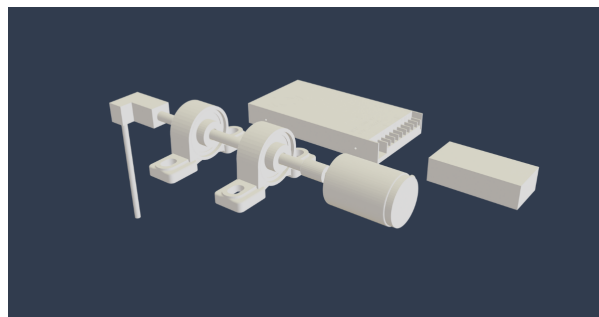


Figura 2.6: Ruta 2.



En la Ruta 2 que se observa en la Figura 2.6, se propone utilizar un motor que tenga integrado un sensor de posición, de esta forma, tanto la fuente de alimentación como el circuito de control, estarían ubicados cerca del motor con sensor integrado, reduciendo el espacio ocupado por el mecanismo completo. Bastaría con 2 chumaceras para el soporte. Finalmente, en el extremo opuesto al motor, se encontraría el bloque que sujeta a la barra péndulo.

- Ruta 3: O3F1-O2F2-O2F3-O2F4

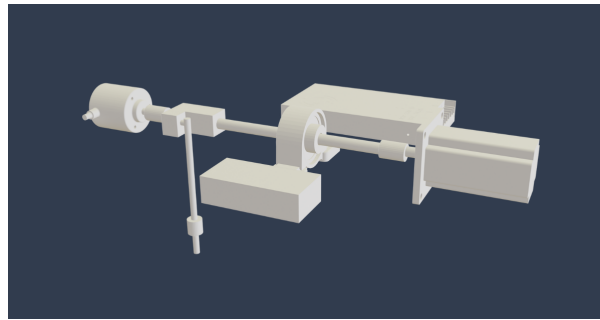


Figura 2.7: Ruta 3.

Por último, en la Ruta 3 que se observa en la Figura 2.7, se sigue una distribución similar a la vista anteriormente en la Figura 2.5, teniendo un motor en uno de los extremos y un sensor en el opuesto, sin embargo, en esta propuesta se tendría solo una chumacera para darle soporte al eje y tanto la fuente de alimentación como el circuito de control se encontrarían opuestos uno del otro sobre los costados del eje, teniendo en cuenta que la tarjeta de desarrollo encargada de recopilar los datos y mandar las señales al motor, se encuentre a una distancia cercana del computador.

## Selección de diseño conceptual

Una vez descritas las rutas propuestas de diseño, se procede a utilizar el método AHP, descrito en la sección 1.3. La Tabla 2.6 contiene las variables a considerar durante el proceso, así como las rutas descritas en las Figuras 2.5, 2.6, y 2.7



Tabla 2.6: Variables y criterios.

	Numero de elementos	Comunicación motor-sensor	Ensamble
Ruta 1	10	Media	Fácil
Ruta 2	8	Alta	Fácil
Ruta 3	8	Media	Fácil

En la Tabla 2.7 se comparan entre sí los criterios a evaluar, con el objetivo de obtener un vector de ponderación que será de utilidad más adelante.

Tabla 2.7: Matriz de comparación de criterios.

Criterios	Número de elementos	Comunicación motor-sensor	Ensamble	Matriz normalizada			Ponderación
Número de elementos	1.00	0.13	5.00	0.11	0.10	0.38	0.20
Comunicación motor-sensor	8.00	1.00	7.00	0.87	0.79	0.54	0.73
Ensamble	0.20	0.14	1.00	0.02	0.11	0.08	0.07
Total	9.20	1.27	13.00				

En la tabla 2.8 se tiene como primer criterio el número de elementos que principalmente conformarían el sistema, ya que se busca la reducción de elementos para facilitar el ensamble del ya mencionado.

Tabla 2.8: Criterio: Número de elementos

Alternativas	Ruta 1	Ruta 2	Ruta 3	Matriz normalizada			Vector promedio
Ruta 1	1	0.2	0.2	0.0909	0.1304	0.0476	0.0897
Ruta 2	5	1	3	0.4545	0.6522	0.7143	0.6070
Ruta 3	5	0.33333333	1	0.4545	0.2174	0.2381	0.3033
Total	11	1.53333333	4.2				

Es importante que la información proveniente del motor, sea recibida adecuadamente por el sensor, por lo que este punto es de gran importancia, la Tabla que tiene este criterio como eje central es la 2.9.

Tabla 2.9: Criterio: Comunicación motor-sensor.

Alternativas	Ruta 1	Ruta 2	Ruta 3	Matriz normalizada			Vector promedio
Ruta 1	1	0.142857143	0.2	0.0769	0.1064	0.0323	0.0719
Ruta 2	7	1	5	0.5385	0.7447	0.8065	0.6965
Ruta 3	5	0.2	1	0.3846	0.1489	0.1613	0.2316
Total	13	1.342857143	6.2				



Por último, se busca que el ensamble del sistema sea sencillo, por lo que este punto está relacionado con el primer criterio. La comparación se muestra en la Tabla 2.10.

Tabla 2.10: Criterio: Ensamble.

Alternativas	Ruta 1	Ruta 2	Ruta 3	Matriz normalizada			Vector promedio
Ruta 1	1	0.333333333	0.333333333	0.1429	0.2174	0.0526	0.1376
Ruta 2	3	1	5	0.4286	0.6522	0.7895	0.6234
Ruta 3	3	0.2	1	0.4286	0.1304	0.1579	0.2390
Total	7	1.533333333	6.333333333				

Finalmente, en la Tabla 2.11 se muestra el resultado obtenido con el método AHP, dando como resultado la **Ruta 2** mostrada en la Figura 2.6 como la ruta de diseño a desarrollar.

Tabla 2.11: Selección de ruta.

Criterio/Alternativa	Número de elementos	Comunicación motor-sensor	Ensamble	Priorización
Ruta 1	0.089654307	0.071854707	0.13762668	0.07999996
Ruta 2	0.607001694	0.696531334	0.62340634	0.67371569
Ruta 3	0.303343999	0.231613959	0.23896698	0.24628435
Ponderación	0.20	0.73	0.07	

## Propuesta solución

### Modelado del sistema

En este apartado se presentaran los modelos dinámicos de cada sistema obtenidos mediante la ecuación de Euler-Lagrange, posteriormente, se expresan estas ecuaciones en sus equivalentes en el espacio de estados, presentando el modelo final usando la expresión del modelo dinámico no lineal de un robot.



## Péndulo Simple

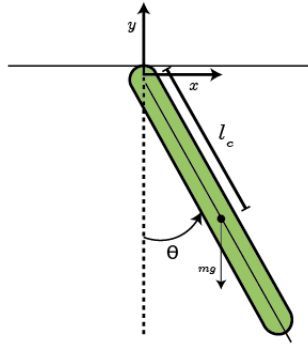


Figura 2.8: Péndulo simple.

La posición de la barra se describe mediante la siguiente ecuación:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_c \sin \theta \\ -l_c \cos \theta \end{bmatrix}$$

Derivando con respecto del tiempo, la velocidad esta dada por la siguiente ecuación:

$$v = \frac{d}{dt} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_c \cos \theta \\ l_c \sin \theta \end{bmatrix} \dot{\theta}$$

La energía cinética total del sistema:

$$K = \frac{1}{2} m l_c^2 \dot{\theta}^2 + \frac{1}{2} I \dot{\theta}^2$$

La energía potencial del sistema:

$$U = mgh = -mgl_c \cos \theta$$

El lagrangiano que describe al sistema:

$$L = \frac{1}{2} m l_c^2 \dot{\theta}^2 + \frac{1}{2} I \dot{\theta}^2 - (-mgl_c \cos \theta) = \frac{1}{2} m l_c^2 \dot{\theta}^2 + \frac{1}{2} I \dot{\theta}^2 + (mgl_c \cos \theta)$$



$$D = \frac{1}{2}b\dot{\theta}^2$$

La ecuación de movimiento de Euler-Lagrange:

$$\ddot{\theta} = \frac{\tau - ((mgl_c \sin \theta) + b\dot{\theta})}{ml_c^2 + I}$$

Representación del modelo en ecuaciones de estado:

Las variables de estado son:  $\theta, \dot{\theta}$

Ecuación de estados:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\theta} \\ \frac{u - ((mgl_c \sin \theta) + b\dot{\theta})}{ml_c^2 + I} \end{bmatrix}$$

La descripción paso a paso del desarrollo matemático se presenta en la sección de Apéndice A.

La validación del comportamiento del péndulo simple se realizó probando sus funcionamiento a diferentes condiciones iniciales, siendo el péndulo más sencillo y con menos grados de libertad es el más fácil de entender.

La programación de la simulación 2D de este péndulo se realizó en Python, el código fuente se puede consultar en la sección de Apéndice B.

A continuación, se presentan los resultados:

$$\theta = 0$$

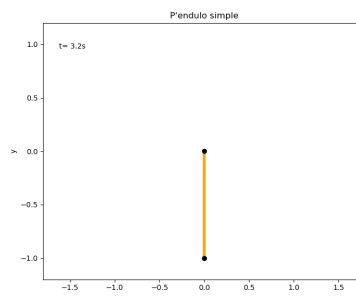


Figura 2.9: Posición del péndulo simple en el plano XY.



El péndulo no es forzado, por lo que, como se puede ver en la Figura 2.9 en este punto el péndulo se encuentra en equilibrio, por lo que se mantiene estático en todo el periodo de tiempo de la simulación.

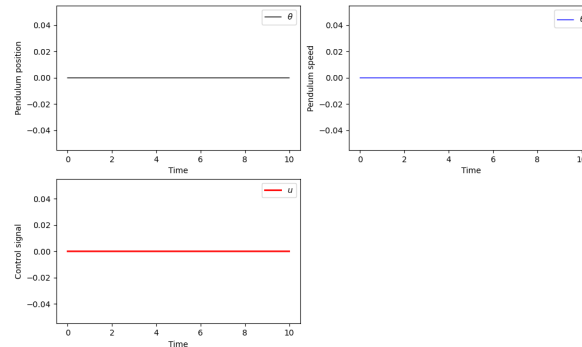


Figura 2.10: Gráficas de salida con respecto al tiempo del péndulo simple a  $\theta = 0$ .

También se puede observar en la Figura 2.10 las gráficas de posición, velocidad del péndulo y, por último, la señal de control del actuador que proporciona movimiento al motor. En todas las gráficas de posición y velocidad al nivel de 0, con esto se reafirma que el péndulo se encuentra en su condición de equilibrio estático. La señal de control en nivel 0 es debido a que no se le aplica ninguna entrada.

$$\theta = \frac{\pi}{2}$$

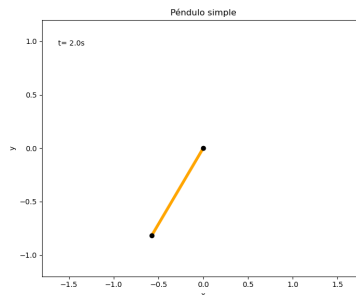


Figura 2.11: Posición del péndulo en el plano XY.

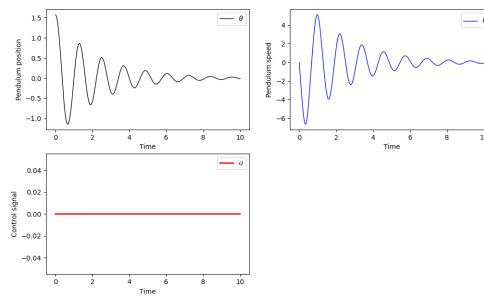


Figura 2.12: Gráficas de salida con respecto al tiempo del péndulo simple a  $\theta = \frac{\pi}{2}$ .

El péndulo liberado desde una posición diferente al equilibrio comienza con una oscilación, en la Figura 2.11 se observa al péndulo oscilando libremente.

En la Figura 2.12 se puede ver la respuesta en el tiempo del péndulo. En la gráfica de posición se puede ver la oscilación libre del péndulo, y por efecto de la fricción el péndulo se comienza a detener hasta llegar a su posición de equilibrio, de igual manera la velocidad del péndulo oscila hasta reducir a 0. La señal de control de  $u$  se mantiene en 0, pues en este momento aún no se le adapta un actuador.

### Péndulo invertido

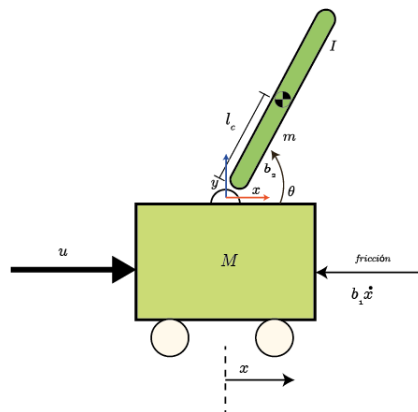


Figura 2.13: Péndulo invertido.





La posición de la barra se expresa de la siguiente manera:

$$p = \begin{bmatrix} l_c \cos \theta + x \\ l_c \sin \theta \end{bmatrix}$$

Derivando con respecto del tiempo, la velocidad es:

$$v = \frac{d}{dt} p = \dot{p} = \begin{bmatrix} -l_c \dot{\theta} \sin \theta + \dot{x} \\ l_c \dot{\theta} \cos \theta \end{bmatrix} = -l_c \dot{\theta} \sin \theta + l_c \dot{\theta} \cos \theta + \dot{x}$$

La energía cinética:

$$K = \frac{1}{2} (M + m) \dot{x}^2 + \frac{1}{2} (I + ml_c^2) \dot{\theta}^2 - ml_c \dot{x} \dot{\theta} \sin \theta$$

La energía potencial:

$$U = mgl \sin \theta$$

El lagrangiano que describe al sistema:

$$L = \frac{1}{2} (M + m) \dot{x}^2 + \frac{1}{2} (I + ml^2) \dot{\theta}^2 + ml \dot{x} \dot{\theta} \cos \theta + mgl \cos \theta$$

$$D = \frac{1}{2} b_1 \dot{x}^2 + \frac{1}{2} b_2 \dot{\theta}^2$$

La ecuación de movimiento de Euler-Lagrange:

$$\ddot{x} = \frac{u_1 - b_1 \dot{x} + ml_c \ddot{\theta} \sin \theta + ml_c \dot{\theta}^2 \cos \theta}{M + m}$$

$$\ddot{\theta} = \frac{u_2 - ml_c \ddot{x} \sin \theta - mgl_c \cos \theta - b_2 \dot{\theta}}{I + ml_c^2}$$

La representación en espacios de estados se puede expresar como:

Variables de estado:  $x, \dot{x}, \theta, \dot{\theta}$

Vector de estado:



$$z = \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}$$

Ecuación de estado

$$\dot{z} = f(u, z, t)$$

$$= \begin{bmatrix} x \\ \theta \\ \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} x \\ \theta \\ M^{-1}(\mathbf{q}) [\mathbf{u} - C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} - g(\mathbf{q})] \end{bmatrix} = \begin{bmatrix} x \\ \theta \\ \begin{bmatrix} M+m & ml_c \cos \theta \\ ml_c \cos \theta & I + ml_c^2 \end{bmatrix}^{-1} \left( \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} - \begin{bmatrix} b_1 \\ 0 \end{bmatrix} - \begin{bmatrix} -ml_c \dot{\theta} \sin \theta \\ b_2 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} - \begin{bmatrix} 0 \\ mgl_c \sin \theta \end{bmatrix} \right) \end{bmatrix}$$

La descripción paso a paso del desarrollo matemático se presenta en la sección de Apéndice A.

La verificación se realizó en Python, a condiciones estáticas y condiciones fuera del equilibrio estático, esto con el fin de probar el comportamiento y verificación del modelo. A continuación, se presenta la operación a diferentes puntos de operación.

$$\theta = x = 0$$

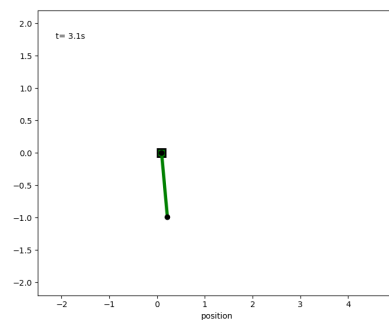


Figura 2.14: Posición del péndulo invertido en el plano XY.

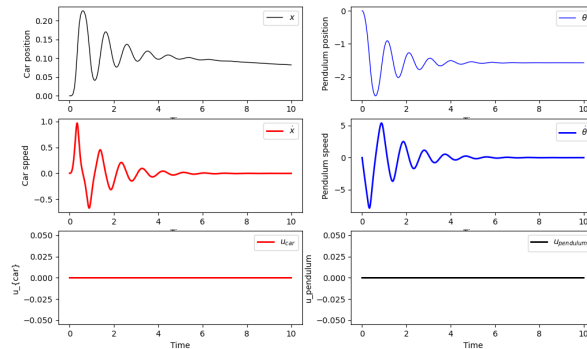


Figura 2.15: Gráficas de salida respecto al tiempo péndulo invertido a  $\theta = x = 0$ .

El péndulo invertido de la Figura 2.14 se encuentra oscilando libremente, pues al no estar alimentado, al iniciar en un ángulo de 0, la posición del péndulo es sobre el eje positivo  $x$ , al soltarlo en una posición diferente la equilibrio en  $\theta = 0$ , el péndulo oscila, por efecto de la fricción el péndulo se detiene sobre la parte negativa del eje  $y$ , pues en esta posición se logra obtener el equilibrio estático, sin embargo, el carro mantiene oscilaciones muy pequeñas que tienden a desaparecer.

Lo descrito anteriormente se puede observar en la Figura 2.15. Se puede notar que el carro no alcanza a detenerse, esto debido al tiempo de simulación, a diferencia del péndulo que se establece en la posición  $\theta = \frac{-\pi}{2}$ ,

$$\theta = \frac{\pi}{2}, x = 0$$

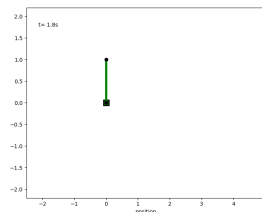


Figura 2.16: Posición en el plano XY del péndulo invertido.

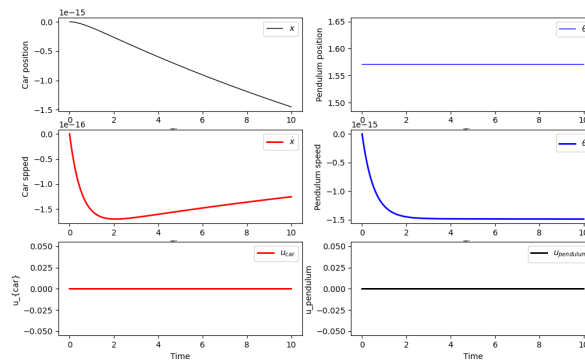


Figura 2.17: Gráficas de salida con respecto al tiempo del péndulo invertido a  $\theta = \frac{\pi}{2}$ ,  $x = 0$ .

El comportamiento del péndulo invertido a condiciones iniciales  $\theta = \frac{\pi}{2}$ ,  $x = 0$  se muestra en la Figura 2.16. El péndulo en este punto se encuentra en equilibrio y no presenta movimiento, en cambio el carro presenta oscilaciones pequeñas.

El péndulo se mantiene estático sin alimentación, y el carrito muestra movimientos pequeños, en la gráfica de posiciones se pueden ver en la Figura 2.17, al igual que las señales de control, que se mantiene en 0, y al igual se puede ver en esta figura, las gráficas de velocidad con valores muy pequeños.

### Péndulo doble

Es necesario comenzar a describir el movimiento obteniendo el Lagrangiano. Para la coordenada del centro de masa del brazo 1:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} l_{c1} \sin \theta_1 \\ -l_{c1} \cos \theta_1 \end{bmatrix}$$

Para la coordenada del centro de masa del brazo 2:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} l_1 \sin \theta_1 + l_{c2} \sin (\theta_1 + \theta_2) \\ -l_1 \cos \theta_1 - l_{c2} \cos (\theta_1 + \theta_2) \end{bmatrix}$$



La velocidad en ambas coordenadas es:

$$v_1 = \begin{bmatrix} l_{c1} \cos \theta_1 \\ l_{c1} \sin \theta_1 \end{bmatrix} \dot{\theta}_1$$

$$v_2 = \begin{bmatrix} l_1 \cos \theta_1 + l_{c2} \cos (\theta_1 + \theta_2) & l_{c2} \cos (\theta_1 + \theta_2) \\ l_1 \sin \theta_1 + l_{c2} \sin (\theta_1 + \theta_2) & l_{c2} \sin (\theta_1 + \theta_2) \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

La energía cinética total del sistema:

$$K = \frac{1}{2} \left( \dot{\theta}_1^2 (m_1 l_{c1}^2 + I_1 + I_2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos (\theta_2))) + \dot{\theta}_2^2 (m_2 l_{c2}^2 + I_2) + 2\dot{\theta}_1 \dot{\theta}_2 (m_2 l_1 l_{c2} \cos (\theta_2) + m_2 l_{c2}^2 + I_2) \right)$$

La energía potencial:

$$U = g (m_1 y_1 + m_2 y_2) = -g (m_1 l_{c1} \cos \theta_1 + m_2 (l_1 \cos \theta_1 + l_{c2} \cos (\theta_1 + \theta_2)))$$

A continuación, se presenta el Lagrangiano:

$$L = K - U = \frac{1}{2} \left( \dot{\theta}_1^2 (m_1 l_{c1}^2 + I_1 + I_2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos (\theta_2))) + \dot{\theta}_2^2 (m_2 l_{c2}^2 + I_2) + 2\dot{\theta}_1 \dot{\theta}_2 (m_2 l_1 l_{c2} \cos (\theta_2) + m_2 l_{c2}^2 + I_2) + g (m_1 l_{c1} \cos \theta_1 + m_2 (l_1 \cos \theta_1 + l_{c2} \cos (\theta_1 + \theta_2))) \right)$$

$$D = \frac{1}{2} \left( m_1 b_1 \dot{\theta}_1^2 + m_2 b_2 \dot{\theta}_2^2 \right)$$

La ecuación de movimiento de Euler-Lagrange:

$$-2m_2 \dot{\theta}_1 \dot{\theta}_2 l_1 l_{c2} \sin (\theta_2) - m_2 \dot{\theta}_2^2 l_1 l_{c2} \sin (\theta_2) + m_1 b_1 \dot{\theta}_1 + g (m_1 l_{c1} \sin \theta_1 + m_2 (l_1 \sin \theta_1 + l_{c2} \sin (\theta_1 + \theta_2))) = \tau_1$$



$$\ddot{\theta}_2 (m_2 l_{c2}^2 + I_2) + \ddot{\theta}_1 (m_2 l_1 l_{c2} \cos(\theta_2) + m_2 l_{c2}^2 + I_2) + m_2 \dot{\theta}_1^2 l_1 l_{c2} \sin(\theta_2) + m_2 b_2 \dot{\theta}_2 + g m_2 l_{c2} \sin(\theta_1 + \theta_2) = \tau_2$$

Agrupando los términos, se puede expresar la dinámica del péndulo doble en las siguientes matrices:

$$M = \begin{bmatrix} m_1 l_{c1}^2 + I_1 + I_2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(\theta_2)) & m_2 l_1 l_{c2} \cos(\theta_2) + m_2 l_{c2}^2 + I_2 \\ m_2 l_1 l_{c2} \cos(\theta_2) + m_2 l_{c2}^2 + I_2 & m_2 l_{c2}^2 + I_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix}$$

$$C = \begin{bmatrix} -2m_2 \dot{\theta}_2 l_1 l_{c2} \sin(\theta_2) + m_1 b_1 & -\dot{\theta}_2 m_2 l_1 l_{c2} \sin(\theta_2) \\ m_2 \dot{\theta}_1 l_1 l_{c2} \sin(\theta_2) & m_2 b_2 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

$$g = \begin{bmatrix} m_1 l_{c1} \sin \theta_1 + m_2 (l_1 \sin \theta_1 + l_{c2} \sin(\theta_1 + \theta_2)) \\ m_2 l_{c2} \sin(\theta_1 + \theta_2) \end{bmatrix}$$

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

Modelo en espacio de estados:

Variables de estado:  $(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$

Vector de estados:

$$x = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

La ecuación en espacios de estados:

$$\dot{x} = f(x, t, u)$$



$$\dot{x} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ M^{-1} (\mathbf{u} - (C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + g(\mathbf{q}))) \end{bmatrix}$$

Se puede consultar el proceso detallado en el sección de Apéndice A.

El modelo del péndulo doble se simuló usando Python, se simuló en los siguientes puntos de condiciones iniciales, no está alimentado, es por eso que las señales de control para ambas gráficas es 0. El código fuente de la simulación se puede consultare en la sección de Apéndice B.

$$\theta_1 = \theta_2 = 0$$

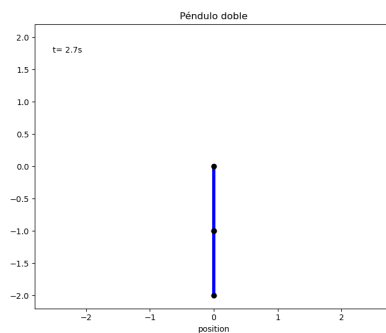


Figura 2.18: Posición en el plano XY (Péndulo doble).

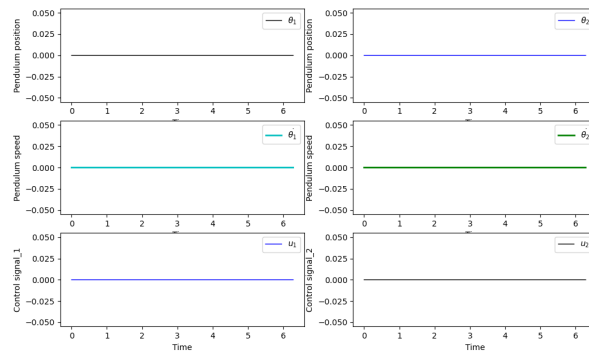


Figura 2.19: Gráficas de salida con respecto al tiempo del Péndulo doble a condiciones iniciales  $\theta_1 = \theta_2 = 0$ .

En la Figura 2.18 se muestra el comportamiento del péndulo en condiciones iniciales 0, al estar en un punto de equilibrio estático, el péndulo no muestra movimiento. Las gráficas de salida que se muestran en la Figura 2.19, todas se muestran en 0, pues al estar en equilibrio, el péndulo no se mueve, ambos brazos se mantienen en 0, al igual que las velocidades de los brazos se mantienen en 0.

$$\theta_1 = \frac{\pi}{2}, \theta_2 = 1$$

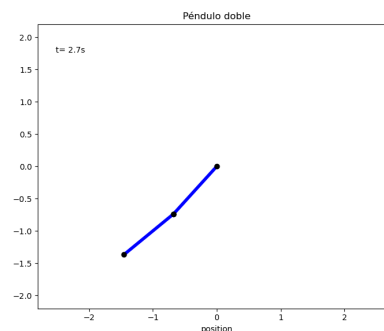


Figura 2.20: Posición en el plano XY (Péndulo doble).



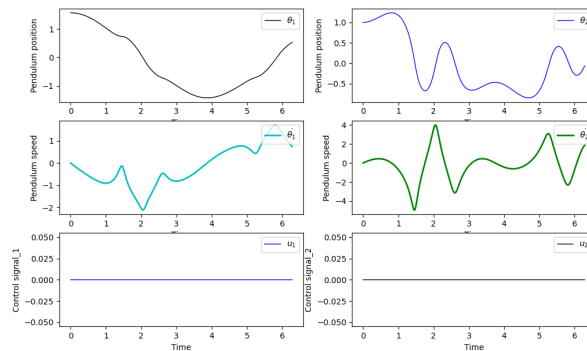


Figura 2.21: Gráficas de salida con respecto al tiempo del péndulo doble a condiciones iniciales  $\theta_1 = \frac{\pi}{2}$ ,  $\theta_2 = 1$ .

El comportamiento mostrado en la Figura 2.21 es el péndulo oscilando libremente, ambos brazos oscilan libremente, este sistema es caótico, por lo que no llega a establecerse y tiende al desorden.

La animación del movimiento de los brazos del péndulo se puede ver en la Figura 2.20.

### Selección de los indicadores de desempeño

Como se ha mencionado anteriormente, es importante seleccionar los indicadores de desempeño adecuados para el problema multiobjetivo, por lo que se hará uso de la metodología AHP, comúnmente utilizada para la toma de decisiones.

### Indicadores basados en el error y la señal de control

En la Tabla 2.12 se presentan tres indicadores de desempeño que son usados ampliamente en la literatura. Se compararon diferentes criterios que nos permitirán elegir la mejor opción.



Tabla 2.12: Variables y criterios.

	Supresión de errores	Implementación	Evaluación analítica
ISE	Buena	Fácil	Fácil
ITAE	Buena	Fácil	Difícil
IAE	Regular	Muy Fácil	Difícil

Tenemos como resultado la tabla 2.13

Tabla 2.13: Matriz de comparación de criterios.

Criterios	Supresión de errores	Facilidad de implementación	Facilidad de Evaluación analítica	Matriz normalizada			Ponderación
Supresión de errores	1	7	5	0.7447	0.8537	0.4545	0.6843
Facilidad de implementación	0.14285714	1	5	0.1064	0.1220	0.4545	0.2276
Facilidad de Evaluación analítica	0.2	0.2	1	0.1489	0.0244	0.0909	0.0881
Total	1.34285714	8.2	11				

Posteriormente, como lo indica la metodología, en las Tablas 2.14, 2.15 y 2.16 se comparan entre sí las opciones a elegir con cada uno de los criterios como eje central.

Tabla 2.14: Criterio: Supresión de errores.

Alternativas	ISE	ITAE	IAE	Matriz normalizada			Vector promedio
ISE	1	3	7	0.6774	0.6923	0.6364	0.6687
ITAE	0.33333333	1	3	0.2258	0.2308	0.2727	0.2431
IAE	0.14285714	0.33333333	1	0.0968	0.0769	0.0909	0.0882
Total	1.47619048	4.33333333	11				

Tabla 2.15: Criterio: Implementación digital.

Alternativas	ISE	ITAE	IAE	Matriz normalizada			Vector promedio
ISE	1	1	0.2	0.1429	0.4545	0.0323	0.2099
ITAE	1	1	5	0.1429	0.4545	0.8065	0.4680
IAE	5	0.2	1	0.7143	0.0909	0.1613	0.3222
Total	7	2.2	6.2				



Tabla 2.16: Criterio: Evaluación analítica.

Alternativas	ISE	ITAE	IAE	Matriz normalizada			Vector pro- medio
ISE	1	6	5	0.7317	0.7500	0.7143	0.7320
ITAE	0.16666667	1	1	0.1220	0.1250	0.1429	0.1299
IAE	0.2	1	1	0.1463	0.1250	0.1429	0.1381
Total	1.36666667	8	7				

Por último, en la Tabla 2.17 se colocan los eigenvectores de cada criterio correspondientes a cada alternativa, mostrando que la opción con un peso de priorización mayor, es el ISE.

Tabla 2.17: Selección de índice de desempeño.

Criterio/Alternativa	Supresión de errores	Facilidad de implementación	Facilidad de Evaluación analítica	Priorización
ISE	0.66869689	0.20988689	0.73199768	0.56983499
ITAE	0.24310099	0.4679514	0.12993612	0.28431552
IAE	0.08820212	0.32216171	0.1380662	0.14584949
Ponderación	0.68429495	0.22762655	0.0880785	

Por otra parte, en varios artículos [13, 21], el índice comúnmente usado para medir la suavidad de la señal de control, es el IADU, por lo que representa una buena elección como segunda función objetivo.

### Selección de la estructura del controlador

#### Péndulo simple e invertido

Como propuesta solución, para estos péndulos se seleccionó la estructura PID, debido a su amplio uso en la literatura consultada en los antecedentes. Es importante mencionar que esta estructura se aplicará a cada actuador presente en estos sistemas.



### Péndulo doble

En [30] se menciona que para el control de trayectoria uno de los controladores que presentan mejor desempeño en la resolución de este problema es la estructura PD+. Es por este motivo que se seleccionó el controlador PD+ como estructura de control para el péndulo doble.

### Definición del problema multiobjetivo

Se propone usar como funciones objetivo los índices de rendimiento ISE e IADU, con el objetivo de encontrar un conjunto de compromisos entre la disminución del error y la suavidad de la señal.

### Péndulo simple

De acuerdo con la ecuación 1.4, podemos definir el problema multiobjetivo como se muestra a continuación:

$$\mathbf{min} \mathbf{J}(\mathbf{x}) = [J_{ISE}(\mathbf{x}), J_{IADU}(\mathbf{x})]$$

Donde

$$\mathbf{x} = [K_p, K_d, K_i]$$

Sujeto a

$$0.001 \leq x_i \leq 10 \quad i = [1, 2, 3]$$

Restricciones

$$J_{ISE} \leq 10$$

$$J_{IADU} \leq 0.8$$

$$u(t) = \begin{cases} u(t) = 2.94 \text{ Nm} & \text{si } 2.94 \text{ Nm} \leq u(t) \\ u(t) = u(t) & \text{si } -2.94 \text{ Nm} \leq u(t) \leq 2.94 \text{ Nm} \\ u(t) = -2.94 \text{ mNm} & \text{si } u(t) \leq -2.94 \text{ Nm} \end{cases}$$



Se define  $J_{ISE}$  como  $\int_0^T e^2(t)dt$  donde  $e(t)$  es la diferencia entre la posición deseada y la posición de salida del sistema, es decir,  $e(t) = q_d - q$ .

$J_{IADU}$  se puede definir de la forma  $\int_0^T |u(t) - u(t-1)|dt$  donde  $u$  es la señal de control. En otras palabras, este indicador calcula la deferencia de la señal de control actual con respecto a la señal de control de un instante de tiempo anterior, con esto, lo que se medirá serán las variaciones de la señal de control y evitar variaciones abruptas para todo instante del tiempo.

Los límites de las variables de diseño  $K_i, K_d, K_i$  se establecieron con base en experimentos con el modelo ya establecido en el lenguaje Python, dando como resultado un buen comportamiento con ganancias menores a 10 y los valores de estas variables deben de estar definidas positivamente, es decir, deben de ser mayores a 0.

Los límites que se establecieron en  $J_{ISE}$  y  $J_{IADU}$  están en función de los resultados obtenidos en los artículos de referencia de los antecedentes, y con base en los requerimientos, donde se establece que se debe de minimizar el error y maximizar la suavidad de la señal.

Por último, el rango de valores de la señal de control  $u$ , se estableció tomando en cuenta el torque que proporcionan los diferentes actuadores de DC comerciales, como lo son el motor DC, motor a pasos y el servomotor.

### **Péndulo invertido:**

De forma similar, se establece el problema multiobjetivo para el péndulo invertido.

$$\mathbf{min} \mathbf{J}(\mathbf{x}) = [J_{ISE}(\mathbf{x}), J_{IADU}(\mathbf{x})]$$

Donde

$$\mathbf{x} = [K_{p_j}, K_{d_j}, K_{i_j}] \quad j = 1, 2$$

Sujeto a

$$0.001 \leq x_i \leq 20 \quad i = [1, 2, 3, 4, 5, 6]$$



Restricciones

$$J_{ISE} \leq 20$$

$$J_{IADU} \leq 0.8$$

**Péndulo doble:**

Por último, el problema multiobjetivo para el péndulo doble se establece a continuación.

$$\mathbf{min} \mathbf{J}(\mathbf{x}) = [J_{ISE}(\mathbf{x}), J_{IADU}(\mathbf{x})]$$

Donde

$$\mathbf{x} = [K_{p_j}, K_{d_j}] \quad j = 1, 2$$

Sujeto a

$$0.001 \leq x_i \leq 20 \quad i = [1, 2, 3, 4]$$

**Selección de técnicas estadísticas para evaluar el desempeño de los algoritmos metaheurísticos.**

La evaluación de los algoritmos metaheurísticos se realiza con indicadores de rendimiento, debido a su uso en la literatura el indicador seleccionado es el Hiper-volumen.

Los algoritmos al componerse de variables aleatorias son conocidos también como algoritmos estocásticos, debido a esto, la distribución sus resultados no siguen una distribución normal. Afortunadamente existen la probabilidad no paramétrica, estas pruebas pueden tratar los datos que no siguen una distribución normal.

Una prueba que se ajusta a las necesidades de este proyecto es la prueba de Wilcoxon, pues se usa para datos que muestran una relación entre sí e indican que tan diferentes son las muestras. Pero para saber realmente que algoritmo presenta mejor desempeño es necesario una prueba de signo.



La solución propuesta para evaluar y comparar el desempeño de los algoritmos es la prueba de rangos con signo de Wilcoxon, pues es ampliamente usada en la literatura consultada.

## Validación

### Problema de optimización

Para la validación del problema de optimización, se usará la librería *pymoo* de Python, esta librería permite resolver problemas de optimización con algoritmos evolutivos, tanto para problemas de un solo objetivo como para multiobjetivo. Cuenta con varios algoritmos para la optimización multiobjetivo, como lo es NSGA-II, con este algoritmo se trabajarán las pruebas de validación.

Solo se probarán el planteamiento del péndulo simple con un ajuste en la restricción de la señal de control  $u$ , pues el torque máximo que el motor ofrece es superior al establecido, además se tomará en cuenta el signo de la señal, pues indica la dirección de giro del motor, por lo que esta restricción se plantea de la siguiente manera:

$$u(t) = \begin{cases} u(t) = 2.94 \text{ Nm} & \text{si } 2.94 \text{ Nm} \leq u(t) \\ u(t) = u(t) \text{ Nm} & \text{si } -2.94 \text{ Nm} \leq u(t) < 2.94 \text{ Nm} \\ u(t) = -2.94 \text{ mNm} & \text{si } u(t) < -2.94 \text{ Nm} \end{cases}$$

A continuación, se procede a la validación de las restricciones de los valores mínimos de ISE e IADU, de la misma forma, se procede a validar cada límite con el uso de la librería *pymoo*, para eso, en el diagrama de frente de Pareto se tiene como salida ISE vs ISE e IADU vs IADU, sin tomar en cuenta los valores establecidos en el planteamiento del problema, esto con el fin de verificar el valor mínimo de ambos objetivos.

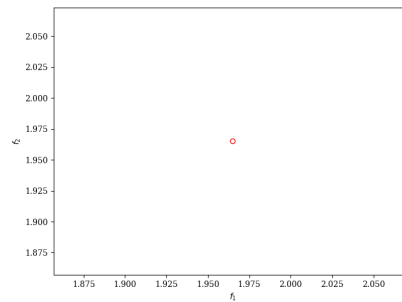


Figura 2.22: ISE vs ISE.

En la Figura 2.22 se muestra el valor mínimo de ISE que se puede obtener usando algoritmos evolutivos, este valor es de 1.96 aproximadamente, y para no forzar al algoritmo el nuevo límite establecido será de 20.

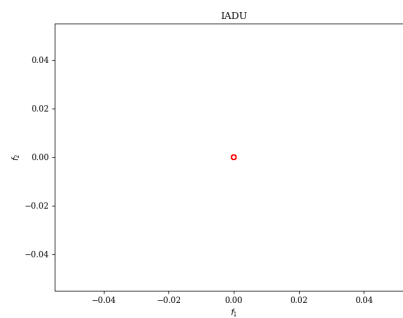


Figura 2.23: IADU vs IADU.

En la Figura 2.23 se muestra el valor mínimo de IADU que se puede obtener usando algoritmos evolutivos, este valor es muy próximo a 0, por lo que el nuevo límite se mantiene en 0.8.

Reescribiendo el problema de optimización:

### Péndulo simple

$$\min \mathbf{J}(\mathbf{x}) = [J_{ISE}(\mathbf{x}), J_{IADU}(\mathbf{x})]$$





Donde

$$\mathbf{x} = [K_p, K_d, K_i]$$

Sujeto a

$$0.001 \leq x_i \leq 10 \quad i = [1, 2, 3]$$

Restricciones

$$J_{ISE} \leq 20$$

$$J_{IADU} \leq 0.8$$

$$u(t) = \begin{cases} u(t) = 2.94 \text{ Nm} & \text{si } 2.94 \text{ Nm} \leq u(t) \\ u(t) = u(t) \text{ Nm} & \text{si } 0 \text{ Nm} \leq u(t) < 2.94 \text{ Nm} \\ u(t) = u(t) \text{ Nm} & \text{si } -2.94 \leq u(t) < 0 \text{ mNm} \\ u(t) = -2.94 \text{ mNm} & \text{si } u(t) < -2.94 \text{ Nm} \end{cases}$$

### Péndulo invertido

$$\min \mathbf{J}(\mathbf{x}) = [J_{ISE}(\mathbf{x}), J_{IADU}(\mathbf{x})]$$

Donde

$$\mathbf{x} = [K_{p_j}, K_{d_j}, K_{i_j}] \quad j = 1, 2$$

Sujeto a

$$0.001 \leq x_i \leq 20 \quad i = [1, 2, 3, 4, 5, 6]$$

Restricciones

$$J_{ISE} \leq 20$$

$$J_{IADU} \leq 0.8$$



### Péndulo doble

$$\mathbf{min} \mathbf{J}(\mathbf{x}) = [J_{ISE}(\mathbf{x}), J_{IADU}(\mathbf{x})]$$

Donde

$$\mathbf{x} = [K_{p_j}, K_{d_j}] \quad j = 1, 2$$

Sujeto a

$$0.001 \leq x_i \leq 20 \quad i = [1, 2, 3, 4]$$

## 2.2. Diseño detallado

### Modelado del sistema.

El modelado del sistema se realizó con las ecuaciones de Euler-Lagrange para obtener el modelo dinámico del sistema, para posteriormente expresarlo en matrices de Inercia, Coriolis, y gravedad, por último, se realizó un cambio de variable para obtener las ecuaciones de estado del sistema, el cual permite expresar el modelo dinámico obtenido en un sistema de primer orden para su fácil resolución.

Las simulaciones del modelo dinámico de cada uno de los péndulos se realizaron en Python, donde se presenta una simulación 2D del movimiento de cada uno de los péndulos, así como las gráficas de velocidad, posición y señal de control de cada péndulo. Todo lo mencionado anteriormente se presentó en la sección de apéndice A.

### Sintonización con métodos metaheurísticos para optimización multi-objetivo.

La sintonización de los sistemas propuestos se realizará con DE, PSO Y GA, como se plateó en el apéndice A, el problema de optimización y su validación son correctas y muestran un buen resultado.



## Interfaz Gráfica (GUI).

La GUI (Graphical User Interface) es una interfaz entre la persona y la máquina. El objetivo de una interfaz gráfica es representar el código programado de un sistema de una forma más clara y entendible para el usuario, con el propósito de simplificarle las tareas a realizar. Utilizan elementos gráficos como iconos, menús e imágenes para facilitar el manejo del usuario humano. Tanto los sistemas operativos como las aplicaciones utilizan una interfaz gráfica de usuario.

Para este proyecto, se propone la elaboración de una GUI en el ambiente de Python, mostrando en una pantalla de inicio los tres sistemas propuestos inicialmente (péndulo simple, péndulo invertido y péndulo doble). Posteriormente, dependiendo del sistema seleccionado, se solicita el ingreso de datos o variables del mismo, si los datos son correctos, se solicita seleccionar el método metaheurístico a ejecutar. Una vez realizado esto, se muestra el frente de Pareto resultante de la ejecución del programa, así como los valores de cada punto en la gráfica. Se pretende que se pueda seleccionar un conjunto de datos para realizar la simulación dinámica del mecanismo y visualizar las gráficas obtenidas. El proceso descrito se muestra en un diagrama de flujo mostrado en la Figura 3.1.



miro

Figura 2.24: Diagrama de flujo de la GUI.

### Búsqueda de opciones de bibliotecas.

- **Tkinter**

El paquete tkinter («interfaz Tk») es la interfaz estándar de Python para el kit de herramientas GUI de Tcl / Tk. Tanto Tk como tkinter están disponibles en la mayoría de las plataformas Unix, incluido macOS, así como en sistemas Windows.



### ▪ PyQt

Es uno de los módulos más utilizados en la creación de aplicaciones con interfaces gráficas en Python. Es desarrollado por Riverbank Computing Limited. PyQt proporciona enlaces para Qt4 y Qt5.

PyQt está disponible en dos ediciones, ambas ediciones se pueden construir para Python 2 y 3. PyQt contiene más de 620 clases que cubren interfaces gráficas de usuario, manejo de XML, comunicación de red, bases de datos SQL, navegación web y otras tecnologías disponibles en Qt.

### ▪ PySimpleGUI

PySimpleGUI es un módulo para el diseño y creación de interfaces gráficas en Python. Es un conector de la API de Python con el módulo Tkinter. Permite programar los mismos elementos de la interfaz de usuario desarrollada con Tkinter, pero con una interfaz más intuitiva. Permite desarrollar de manera más sencilla interfaces gráficas, además de que agrega la capacidad de personalizar diseños.

## **Análisis de ventajas y desventajas.**

### ▪ Tkinter

Ventajas.

- Disponible para sistemas operativos macOS, Windows, y plataformas Unix.
- Preinstalado en el sistema operativo Windows.
- Relativamente fácil de aprender.

Desventajas.

- Lento.



## ▪ PyQt

Ventajas.

- PyQt4 se ejecuta en Windows, Linux, Mac OS X y varias plataformas UNIX. PyQt5 también se ejecuta en Android e iOS.
- Elementos gráficos completos, pues cuenta con más de 620 clases para interfaces gráficas.
- Manejo de XML, comunicación de red, bases de datos SQL, navegación web y otras tecnologías disponibles.

Desventajas.

- Más complejo de aprender.
- En ocasiones emerge la implementación en C++ subyacente, teniendo que hacer casts entre tipos de datos, etc.

## ▪ PySimpleGUI

Ventajas.

- Utiliza los mismos controles que los de Tkinter, Qt, WxPython y Remi.
- El código se reduce entre un 50 % y un 90 %.
- Tiene una interfaz amigable.
- Mejor interfaz.

Desventajas.

- Trata las ventanas como filas de elementos de la GUI.

## Selección de biblioteca

Por la facilidad y la optimización de código que se mencionó en la sección anterior, la biblioteca que presenta mayores beneficios es PySimpleGUI, pues se pueden



implementar los elementos de otras bibliotecas, y visualmente el resultado es más estético.

### Adaptación del péndulo simple.

En la Figura 2.25 se muestra una matriz con los materiales propuestos para ser empleados en la construcción del mecanismo establecido en la Figura 2.6.

Elemento	Opciones		
	1	2	3
<b>A) PÉNDULO</b>	 Madera	 Aluminio	 Nylamid
<b>B) EJE</b>	 Acero inoxidable	 Aluminio	 Acero al carbono
<b>C) BLOQUE</b>	 ABS	 Nylamid	 PLA
<b>D) ADQUISICIÓN DE DATOS</b>	 Teensy 4.1	 Arduino UNO	 Raspberry Pi

Figura 2.25: Selección de materiales.

### Búsqueda de opciones en la selección de materiales

- Opción 1: A1-B2-C1-D2
- Opción 2: A2-B1-C2-D1
- Opción 3: A3-B3-C3-D3



## Análisis de ventajas y desventajas

Con el propósito de comparar las opciones entre sí y seleccionar la más adecuada, se realizó una tabla de ventajas y desventajas, misma que se observa en la Tabla 2.18.

Opción	Ventajas	Desventajas
1	<ul style="list-style-type: none"> <li>-Esta opción se caracteriza por su livianez, ya que los materiales a considerar son menos pesados a comparación de las otras opciones.</li> <li>-Arduino Uno es una placa de microcontrolador de uso fácil, además, es de bajo costo y es altamente flexible. Otra de las ventajas de usar esta placa es que es multiplataforma, ya que el software de Arduino funciona en los sistemas operativos Windows, Macintosh OSX y Linux.</li> </ul>	<ul style="list-style-type: none"> <li>-El material más usado para ejes es el acero.</li> <li>-El precio a pagar por el uso de las librerías es un retraso en la ejecución de las instrucciones, algunos microsegundos que en el caso de dispositivos de uso cotidiano son irrelevantes, pero significativos a la hora de hacer adquisición de datos.</li> </ul>
2	<ul style="list-style-type: none"> <li>-El aluminio es un material con una buena relación peso/resistencia.</li> <li>-El nylamid tiene una mayor durabilidad que los otros materiales a considerar.</li> <li>- La Teensy 4.1 posee un microcontrolador de 32 bits corriendo a 600MHz, al igual que algunas placas de Arduino, pero con un rendimiento muy superior.</li> <li>-Gracias al plug-in para el IDE de Arduino "Teensyduino", es posible utilizar no solo el IDE de Arduino, sino también la mayoría de librerías del entorno del mismo.</li> </ul>	<ul style="list-style-type: none"> <li>- El nylamid presenta menor resistencia y rigidez que el PLA y ABS.</li> <li>-El precio de la placa Teensy es más elevado que el de una placa Arduino Uno.</li> </ul>
3	<ul style="list-style-type: none"> <li>-El nylamid es un material ligero a comparación de los otros materiales propuestos.</li> <li>-El acero es el material más usado en ejes.</li> <li>-El PLA es un termoplástico fácil de usar con mayor resistencia y rigidez que el ABS y el nylon.</li> </ul>	<ul style="list-style-type: none"> <li>- La relación entre las especificaciones de la Raspberry Pi y los requerimientos del sistema es mayor de lo que se necesita.</li> </ul>

Tabla 2.18: Análisis de ventajas y desventajas en la selección de materiales.

## Selección de opción de materiales

A pesar de que la opción 1 es la más económica, fue descartada ya que la precisión en la adquisición de datos que conlleva usar la placa de Arduino UNO, es baja





comparada con las otras opciones. De manera similar y como se menciona en la Tabla 2.18, en el caso de la opción 3, la relación entre las especificaciones de la Raspberry Pi y los requerimientos del sistema es mayor de lo que se necesita, además de que la adquisición de la Raspberry Pi representa un costo elevado. El punto de equilibrio entre las opciones ya mencionadas, está dado por la opción 2, la cual representa la mejor opción, ya que la Teensy 4.1 posee las especificaciones necesarias para el proyecto. Por otra parte, los materiales del eje, el péndulo y el bloque son similares a las otras opciones. En conclusión, se elige la **opción 2** debido a la precisión en la tarjeta con microcontrolador y a la resistencia y economía que conllevan los demás elementos. Considerando lo anterior, en la figura 2.26 se presenta una descripción del sistema integrado.

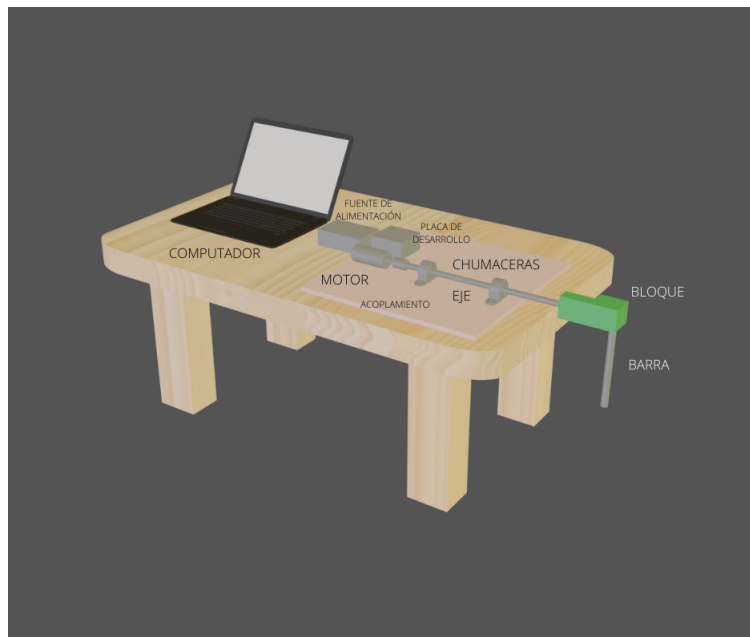


Figura 2.26: Sistema péndulo simple.

Los elementos que conforman el sistema se muestran a continuación:

**Motor:** Como se ha mencionado con anterioridad, el motor es pieza fundamental y será el encargado de suministrar el torque. Una aproximación a lo que sería el motor



se muestra en la Figura 2.27.



Figura 2.27: Motor.

Fuente: Modificado de [57].

**Eje:** El elemento mostrado en la Figura 2.28 será el encargado de transmitir el par del motor hacia el péndulo. Se busca que sea ligero y firme al mismo tiempo, por lo que una buena opción es que sea de aluminio 6061 con un diámetro de 8mm.



Figura 2.28: Eje.

**Chumacera:** Con el propósito de darle estabilidad al eje, se ocuparán dos chumaceras de piso comerciales con rodamientos de 8 mm para eje lineal, como la que se muestra en la Figura 2.29. Estos elementos cuentan con tornillos dentro del balero para sujetar la barra para tolerar vibraciones o movimiento sin dañar el eje, manteniendo la precisión del mismo.



Figura 2.29: Chumacera.

Fuente: Modificado de [58].

**Acomplamientos:** Para poder acoplar el motor al eje, se propone usar un cople flexible de aluminio de 8 mm x 8 mm, como el que se muestra en la Figura 2.30



Figura 2.30: Cople.

Fuente: Modificado de [59].

**Elemento de sujeción:** Se utilizará una pieza impresa en 3D, que servirá como conexión entre el eje y el brazo que fungirá como péndulo. Se muestra en la Figura 2.31.

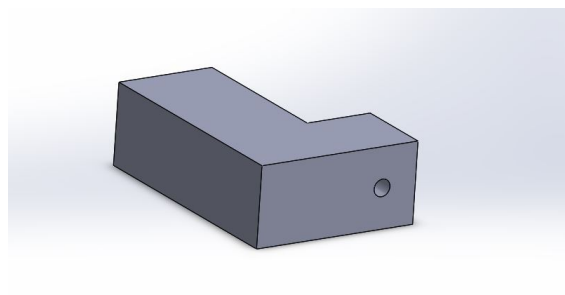


Figura 2.31: Elemento de sujeción.



**Brazo:** Elemento que nos servirá de referencia para visualizar la posición que se haya programado con el control PID. Lo podemos ver en la Figura 2.32.

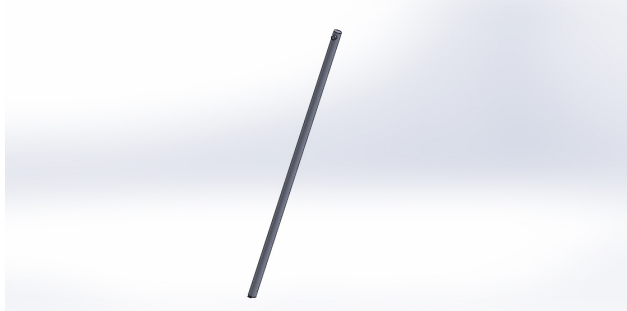


Figura 2.32: Eje de aluminio.

### Movimiento y sensor de posición

Se busca un motor de corriente directa que cuente con un sensor de movimiento. Los motores comerciales que cuenta con un sensor de movimiento acoplado son los motorreductores, y motores a pasos. Se requiere de un motor con sensor de movimiento, un driver que permita poder manipular el motor con el microcontrolador. Las propuestas a considerar son las que se presentan a continuación:

- Motorreductor con Enconder 12V JGA25-370 con puente H.

Alimentación: 6-12 V.

Torque: 0.98 Nm.

Velocidad: 12 RPM.

Resolución del encoder: 5500 PPR.

Driver: Puente H L298.

Costo total: 795 MXN.

- Azssmuk Motor de engranaje de metal.

Alimentación: 12 V.

Velocidad: 300 RPM.



Torque: 0.78 Nm.

Resolución del encoder: 240 PPR.

Driver: Puente H L298.

Costo total: 593 MXN.

- Rtelligent-Motor paso a paso Nema 23

Alimentación: 24-50 V.

Torque: 1 Nm.

Velocidad: 285 RPM.

Resolución del encoder: 1000.

Driver: Rtelligent T60.

Costo total: 1769 MXN.

- Motorreductor con engranes planetarios y encoder.

Alimentación: 6-12 V.

Torque: 2.94 Nm.

Velocidad: 285 RPM.

Resolución del encoder: 250 PPR.

Driver: Puente H L298.

Costo total: 1053 MXN.

Los criterios para considerar son:

Resolución.

Torque.

Precio.

Consumo energético.



Tabla 2.19: Matriz de comparación de criterios.

Criterios	Resolución	Torque	Precio	Consumo energético	Matriz normalizada				Ponderación
Resolución	1	0.33333333	7	5	0.230263158	0.21	0.35	0.375	0.291315789
Torque	3	1	9	7	0.690789474	0.63	0.45	0.525	0.573947368
Precio	0.14285714	0.11111111	1	0.33333333	0.032894737	0.07	0.05	0.025	0.044473684
Consumo energético	0.2	0.14285714	3	1	0.046052632	0.09	0.15	0.075	0.090263158
Total	4.34285714	1.58730159	20	13.33333333					

Las opciones se identificarán de la siguiente manera:

- A. Motorreductor con Enconder 12V JGA25-370 con puente H.
- B. Azssmuk Motor de engranaje de metal.
- C. Rteligent-Motor paso a paso Nema 23.
- D. Motorreductor con engranes planetarios y encoder.

En las Tablas 2.20, 2.21, 2.22 y 2.23 se comparan entre sí las opciones a elegir con cada uno de los criterios como eje central.

Las ponderaciones de los criterios se realizaron en la Tabla 2.19, donde se establece el nivel de importancia que tiene cada criterio en la selección del motor y encoder.

Tabla 2.20: Criterio: Resolución.

Opciones	A	B	C	D	Matriz normalizada				Ponderación
A	1	5	3	5	0.577	0.5	0.643	0.5	0.554945055
B	0.2	1	0.333	1	0.115	0.1	0.071	0.1	0.096703297
C	0.333	3	1	3	0.192	0.3	0.214	0.3	0.251648352
D	0.2	1	0.333	1	0.115	0.1	0.071	0.1	0.096703297
Total	1.733	10	4.667	10					



Tabla 2.21: Criterio: Torque.

Opciones	A	B	C	D	Matriz normalizada				Ponderación
A	1	3	0.333	0.143	0.088	0.167	0.051	0.098	0.101043911
B	0.333	1	0.2	0.111	0.029	0.056	0.031	0.076	0.047999695
C	3	5	1	0.2	0.265	0.278	0.153	0.138	0.208274867
D	7	9	5	1	0.618	0.5	0.765	0.688	0.642681527
Total	11.33	18	6.533	1.454					

Tabla 2.22: Criterio: Precio.

Opciones	A	B	C	D	Matriz normalizada				Ponderación
A	1	0.333	7	3	0.223	0.203	0.35	0.321	0.274383882
B	3	1	9	5	0.67	0.608	0.45	0.536	0.56600879
C	0.143	0.111	1	0.333	0.032	0.068	0.05	0.036	0.046299187
D	0.333	0.2	3	1	0.074	0.122	0.15	0.107	0.113308141
Total	4.476	1.644	20	9.333					

Tabla 2.23: Criterio: Consumo energético.

Opciones	A	B	C	D	Matriz normalizada				Ponderación
A	1	3	7	1	0.404	0.417	0.35	0.404	0.393589744
B	0.333	1	5	0.333	0.135	0.139	0.25	0.135	0.164529915
C	0.143	0.2	1	0.143	0.058	0.028	0.05	0.058	0.048290598
D	1	3	7	1	0.404	0.417	0.35	0.404	0.393589744
Total	2.476	7.2	20	2.476					

Tabla 2.24: Selección de motor con encoder.

Criterios/ Opciones	Resolución	Torque	Precio	Consumo energético	Priorización
A	0.55494505	0.10104391	0.27438388	0.39358974	0.267387659
B	0.0967033	0.04799969	0.56600879	0.16452991	0.095743982
C	0.25164835	0.20827487	0.04629919	0.0482906	0.199265908
D	0.0967033	0.64268153	0.11330814	0.39358974	0.437602452
Ponderación	0.29131579	0.57394737	0.04447368	0.09026316	

En la Tabla 2.24 se muestra el resultado final, donde se puede que la opción mejor evaluada es la opción D, pues ofrece una buena resolución en el encoder y además ofrece un torque de a 2.94 Nm.



Figura 2.33: Motor con encoder.

Fuente: Modificado de [57].

El motor mostrado en la Figura 2.33 es el motor seleccionado, es un motorreductor con relación de velocidad de 14:1, con una velocidad de 285 RPM, el encoder, es un encoder de efecto óptico, tiene una resolución de 1000 CPR, por último, puede operar con niveles de tensión de 6-12 V.

### Microcontrolador

La tarjeta seleccionada para este proyecto es la tarjeta Teensy 4.1, pues tiene una buena capacidad de procesamiento, además de su facilidad de programar, en la Figura 2.34, se puede observar esta tarjeta.



Figura 2.34: Teensy 4.1.

Fuente: Modificado de [60].

- Características
  - Cortex-M7 a 600 MHz.





- Unidad matemática de punto flotante, 64 y 32 bits.
  - 7936K Flash, 1024K RAM (512K estrechamente acoplado), 4K EEPROM (emulado).
  - Dispositivo USB 480 Mbit / seg y host USB 480 Mbit / seg.
  - 55 pines de entrada / salida digital, 35 pines de salida PWM.
  - 18 pines de entrada analógica.
  - 8 puertos seriales, 3 SPI, 3 I2C.
  - Gestión de encendido / apagado.
- Software
    - Arduino IDE + Teensyduino.
  - Unidad de punto flotante (FPU por sus siglas en inglés): La FPU realiza operaciones matemáticas flotantes de 32 bits y de doble precisión de 64 bits en hardware.
  - Configuración de pines

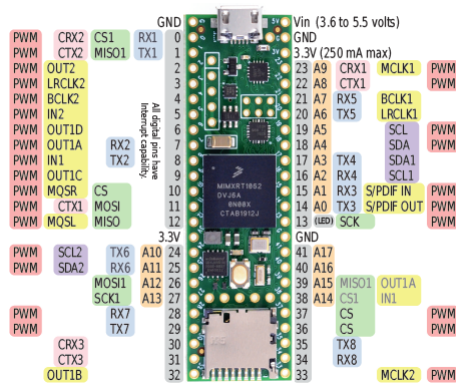


Figura 2.35: Configuración de pines.

Fuente: Modificado de [60].



La lógica es de 3.3 V, puede recibir señales de 0 a 3.3 V en los pines de entradas digitales, de igual manera su salida en alto es de 3.3 V.

Análogamente los pines analógicos solo leen de 0 a 3.3 V y cuenta con una resolución de 10 bits (rangos de entrada de 0 a 1023). No tolera tensiones mayores a 3.3 V.

### Control de péndulo simple

#### Circuito de control

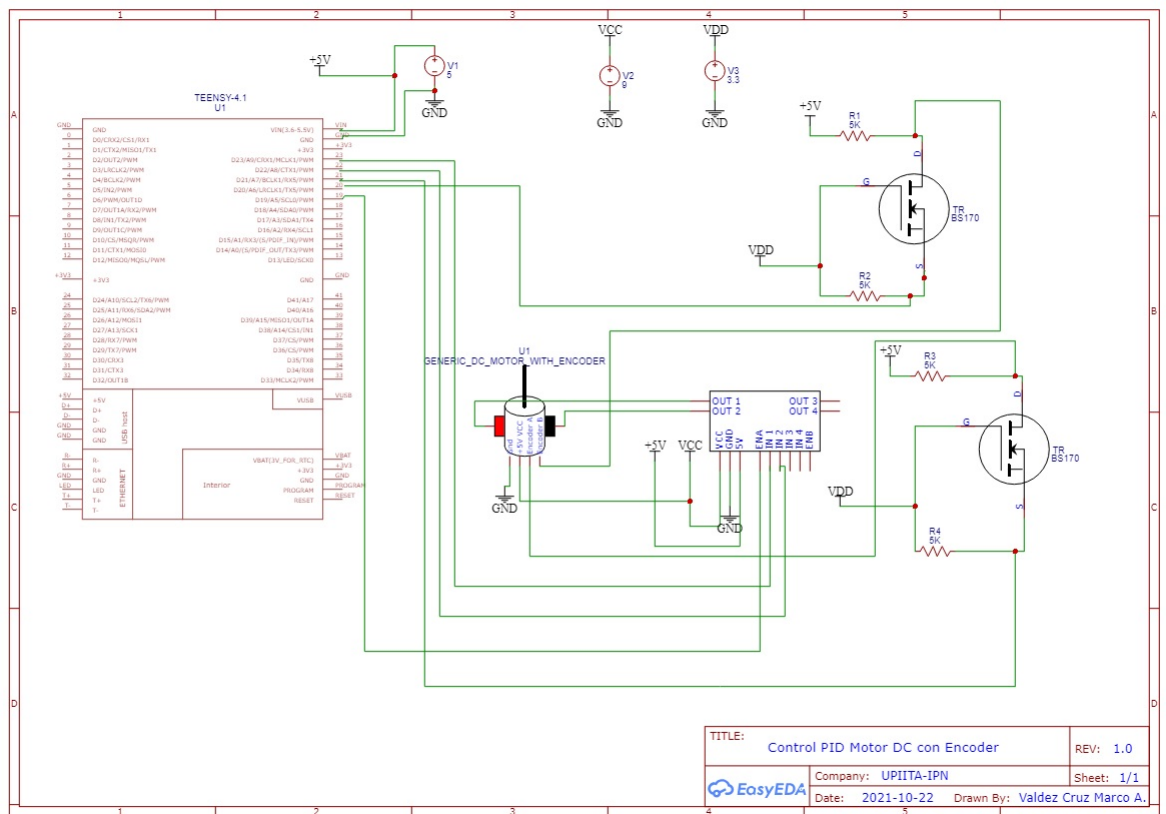


Figura 2.36: Circuito de control.

En la Figura 2.36 se muestra el circuito de control completo, contempla:

#### Teensy 4.1



Es una tarjeta de desarrollo con una memoria Flash de 8 Mbyte, cuenta con 55 pines Input/Output, de los cuales 35 pueden ser usados como salida de PWM, un procesador ARM Cortex-M7 a 600 MHz, y una lógica de activación de 3.3 V, por lo que no soporta tensiones mayores a esta. Requiere un convertidor de nivel lógico que permita adaptar de 5 V a 3.3 V.

### Módulo L298N

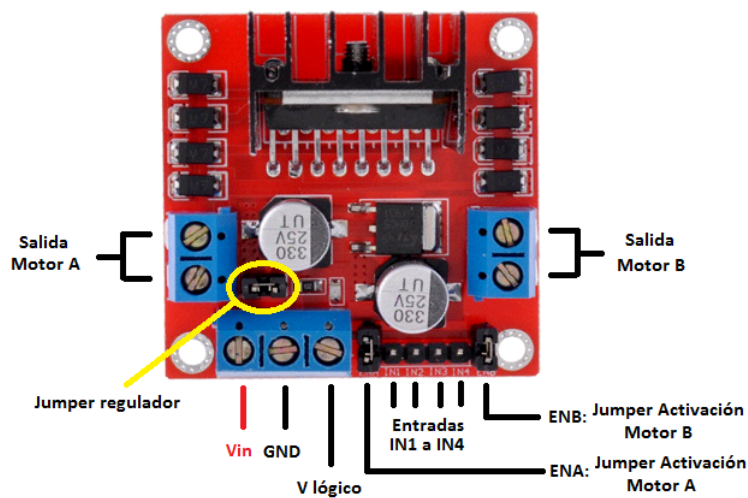


Figura 2.37: Módulo L298N.

Fuente: Modificada de [36].

En la Figura 2.37 se muestra el módulo L298N, este circuito es un driver que permite controlar dos motores DC o en su defecto un motor a pasos. Este módulo será usado para regular la posición del péndulo, con las terminales EN1 y EN2, se puede controlar el sentido de giro del motor, la terminal ENA permite regular la velocidad del motor mediante una señal de PWM, permite controlar motores de hasta 35 V. Además, es compatible con tecnologías de 5V y de 3.3 V.

### Convertidor de nivel lógico



Para esta tarea se usará un transistor MOSFET que tiene un tiempo de encendido y apagado de 4 ns, está polarizado de tal manera que al tener como entrada un alto lógico con nivel de tensión de 5V, a la salida se tenga 3.3 V como salida en alto.

## 2.3. Validación e integración de los sistemas

### Validación del algoritmo de sintonización (MOP)

Se realizó la verificación del algoritmo de optimización con el algoritmo de evolución diferencial, tomando en cuenta el nuevo planteamiento del problema presentado en la sección de validación del problema de optimización, es decir, se codificó el problema de optimización acorde a las actualizaciones presentadas en la sección antes mencionada.

Se simuló el algoritmo con una población inicial de 100 individuos y 1000 generaciones, el resultado se presenta en la Figura 2.38, donde se muestra la aproximación al frente de Pareto obtenida con DE.

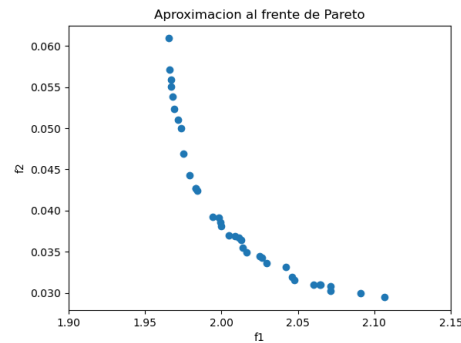


Figura 2.38: Aproximación al frente de Pareto.

### Simulación del circuito de control

El circuito de control contempla una tarjeta Teensy 4.1 que es compatible con el IDE de Arduino, es por ello por lo que se seleccionó el Arduino para simular el



funcionamiento final del subsistema de control, pues la tarjeta Tenmsy no se puede simular. Como se puede ver en la Figura 2.39, se generan 3 señales de control, 2 para controlar el sentido de giro, y una señal de PWM para regular la velocidad de giro del motor, y se recibe de retroalimentación los pulsos del encoder.

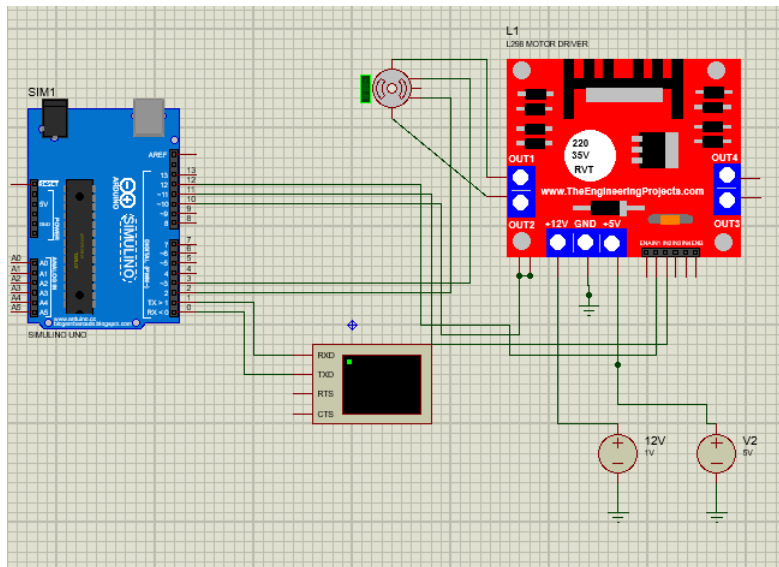


Figura 2.39: Circuito de control del péndulo.

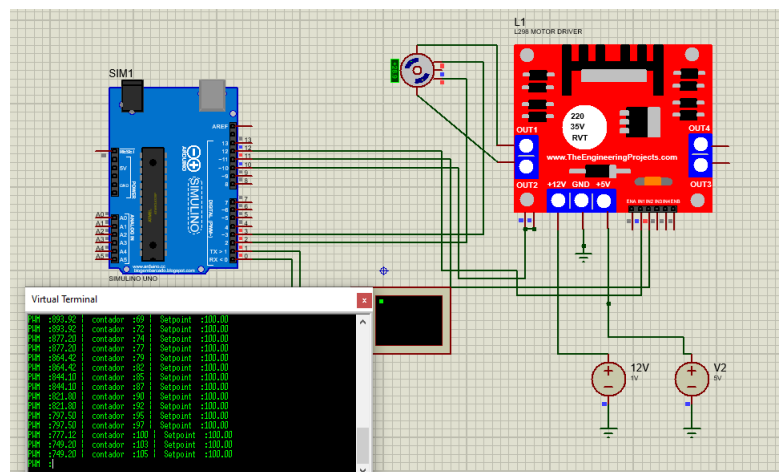


Figura 2.40: Ejecución de simulación.



La Figura 2.40 muestra una simulación de un control PID con PWM en Arduino, también se muestra la comunicación del Arduino con el puerto serie para observar la posición actual del péndulo.

### Simulación del convertidor de nivel lógico.

Por último es necesario un convertidor de nivel lógico, que permita convertir un alto lógico de 5 V a un alto lógico de 3.3 V, pues la Teensy 4.1 no soporta tensiones mayores a 3.3 V, es por eso que se usa un circuito con MOSFET, el cual se muestra en la Figura 2.41. Este circuito recibe como entrada una señal digital en alto de 5 V, y a la salida se obtiene una señal de en alto de 3.3 V, cuando se recibe una señal de 0 V, la salida es 0 V.

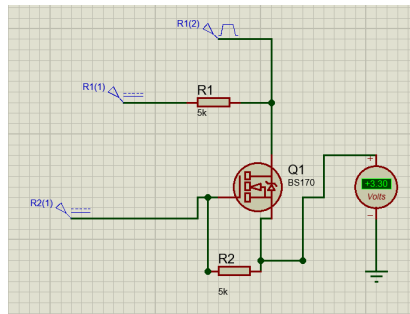


Figura 2.41: Convertidor de nivel lógico de 5 V a 3.3 V.

### Simulación de la acción de control implementada (péndulo simple).

En la presente sección se realizó la prueba de sintonización con el algoritmo de evolución diferencial para el sistema de péndulo simple. Del FP obtenido, se seleccionó el punto donde  $J_{ISE} = 1.9663022702552264$  y  $J_{IADU} = 0.06091058$ , este punto da como resultado los siguientes valores de ganancias.

$$k_p = 9.00809903857079, k_d = 0.74331509706173, k_i = 0.$$

Al hacer la prueba con el algoritmo presentado en el apéndice B, los resultados son



una aproximación del comportamiento que se espera en el experimento físico.

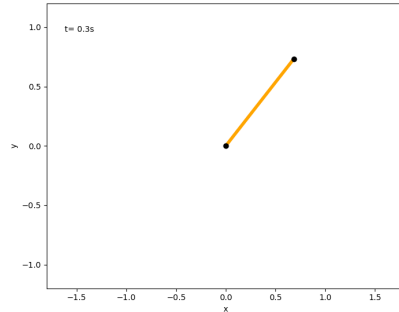


Figura 2.42: Péndulo simple (Control).

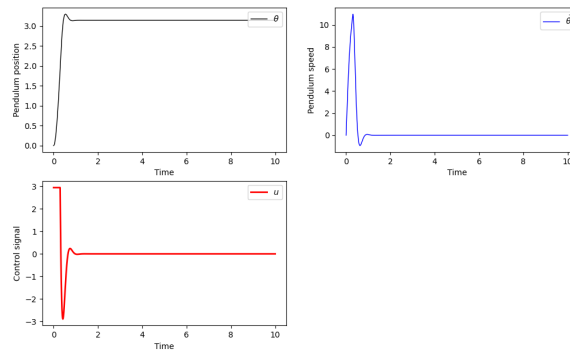


Figura 2.43: Péndulo simple (Control).

En la Figura 2.42 se puede ver la simulación del péndulo simple con la acción de control como entrada al sistema. Se puede ver en la animación como el péndulo parte de posición inicial hasta la posición deseada  $\theta = \frac{\pi}{2}$ .

Las gráficas de posición del comportamiento del péndulo y su velocidad se pueden ver en la Figura 2.43, la gráfica de la señal de control también se muestra en esta figura, y se ve que por un pequeño instante del tiempo, el actuador se satura y cae hasta un pico negativo, posteriormente, el actuador permanece en 0, pues el punto seleccionado es un punto de equilibrio estático.

---

## Implementación

---

Esta sección se divide en dos partes, cubriendo al mismo tiempo la descripción de las áreas funcionales. La primera muestra el resultado de la programación de la interfaz de usuario en Python, usando PySimpleGUI, matplotlib, numpy y math como bibliotecas principales. Esta interfaz tiene como propósito ofrecer al usuario o diseñador la posibilidad de seleccionar la planta a controlar (tipo de péndulo), para posteriormente ingresar los variables y parámetros correspondientes a la planta y seleccionar el algoritmo multiobjetivo a ejecutar (MODE, MOGA o MOPSO). Una vez completado este proceso, se podrá visualizar la aproximación al frente de Pareto con las soluciones óptimas para la sintonización fuera de línea a implementar en el controlador PID. Adicionalmente, se podrá seleccionar una de estas soluciones para realizar una simulación del control dentro de la misma interfaz. Es importante mencionar que el tiempo de ejecución dependerá de factores como las características del servidor (computador), tamaño de población y número de generaciones a iterar. En la segunda parte, se muestra la caracterización del motor y los cálculos necesarios para implementar el control en el péndulo simple físico con el conjunto de ganancias encontradas a través de la interfaz vista en la primer sección, mismas que fueron seleccionadas de acuerdo con la decisión multicriterio del diseñador.





## Modelado de la planta

Las ecuaciones en espacio de estados obtenidas y presentes en la sección *Propuesta solución*, fueron transcritas a un lenguaje de programación interpretado, con el propósito de obtener las simulaciones de la respuesta de control ante una sintonización heurística. Los parámetros usados en el péndulo simple, invertido y doble son los correspondientes a las Tablas 1, 2 y 3, respectivamente. Estos valores fueron seleccionados de manera arbitraria, sin embargo, pueden ser modificados para coincidir con el modelo del péndulo a sintonizar. Los resultados de dicha sintonización se muestran a lo largo de esta sección.

## Interfaz

Tal y como se mencionó en la sección 2.2: *Diseño detallado*, la interfaz de usuario fue desarrollada en Python mediante PySimpleGUI. Las ejecuciones mostradas a continuación fueron llevadas a cabo en dos computadores diferentes, el primero de ellos cuenta con Intel(R) Core™ i5-7200U CPU @ 2.70 GHz, 8GB RAM y el segundo cuenta con AMD Ryzen 3 3300U con Radeon Vega Mobile Gfx @ 2.10 GHz, 12GB RAM, y para programar y ejecutar los algoritmos se usó Python 3.9 como lenguaje de programación. Las IDE's usadas fueron Visual Studio Code 1.53.0, PyCharm Community Edition 2021.1.2 y Spyder 5.3.0. A continuación, se muestra el diagrama de flujo de la interfaz mostrado en la figura 3.1, así como la descripción de cada una de las partes que la conforman.

- *Inicialización de la GUI*: Como primer acción, se ejecuta el programa *.py* en la IDE de preferencia, se recomienda el uso de Visual Studio Code. Es importante mencionar que se requieren algunas bibliotecas que no vienen incluidas en la instalación estándar de Python, por lo que es necesario instalarlas si no se cuenta con ellas. Estas bibliotecas son las siguientes:

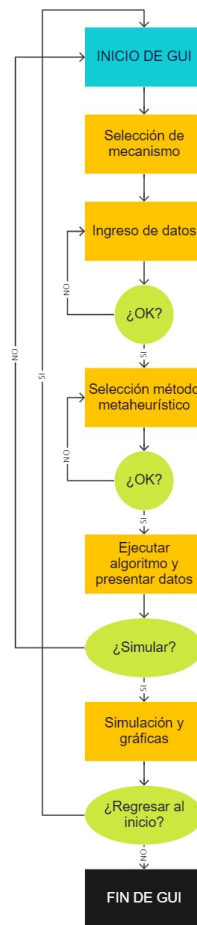


Figura 3.1: Diagrama de flujo de la GUI.

- PySimpleGUI
  - drawnow
  - os
  - scipy
  - mplotcursors
- *Menu principal*: En la primer pantalla o *layout*, se muestra una imagen del péndulo simple, péndulo invertido y péndulo doble, con el objetivo de que



el usuario seleccione el sistema a sintonizar dando *click* sobre la imagen del sistema deseado, como se puede observar en la figura 3.2. Además, se cuenta con un botón de "exit" para terminar con la ejecución del programa.

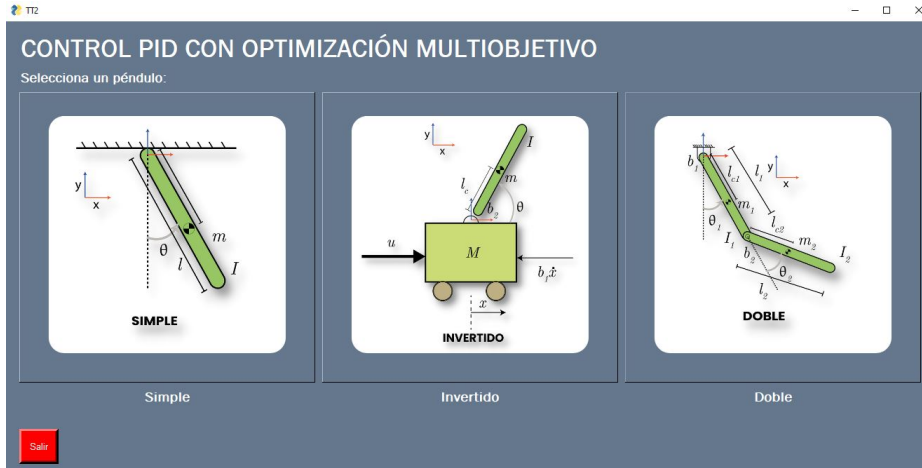


Figura 3.2: Selección de mecanismo.

- *Ingreso de datos*: Una vez que se ha seleccionado a planta a sintonizar, se ingresan los parámetros correspondientes a la dinámica del sistema y el punto de referencia al cual se desea que el péndulo llegue, en el caso del péndulo simple e invertido. Para el péndulo doble, se ha asignado inicialmente una trayectoria circular a seguir. Los parámetros mostrados en cada uno de los campos mostrados en las figuras 3.3, 3.4 y 3.5, son los correspondientes a las Tablas 1, 2 y 3, respectivamente. Estos valores fueron seleccionados de manera arbitraria, sin embargo, pueden ser modificados para coincidir con el modelo del péndulo a sintonizar. Cada uno de los péndulos puede ser sintonizado mediante alguno de los 3 diferentes algoritmos metaheurísticos descritos a lo largo de este documento, los cuales poseen los parámetros mostrados a continuación:

Nota: Estos parámetros están basados en los utilizados en diversas publicaciones de naturaleza similar y que pueden ser consultados en la sección 0.4 Antecedentes.

Optimización multiobjetivo mediante EA

### Péndulo Simple:

Masa (m): 0.5 kg  
 Longitud (l): 1 m  
 Longitud al centro masa (lc): 0.3 m  
 Fricción (b): 0.05 Ns/m  
 Momento de inercia (I): 0.006 kgm<sup>2</sup>  
 Set point (rad): 3.1416 rad

Seleccione el algoritmo metaheurístico:

DE GA PSO

Figura 3.3: Sintonización péndulo simple.

En la figura 3.3 se observa el ingreso de las variables dinámicas correspondientes al péndulo simple. Es importante recordar que las variables con mayor peso son la masa y longitud de la barra, pues afectan considerablemente la respuesta del sistema.

Sintonización del péndulo invertido: Durante el ingreso de los parámetros del péndulo invertido, el número de variables aumenta, es por ello que en la figura 3.4, se muestra la existencia de dos 'set points' o referencias.

Optimización multiobjetivo mediante EA

### Péndulo Invertido:

Masa del péndulo (m): 0.5 kg  
 Masa del carrito (M): 0.7 kg  
 Longitud (l): 1.0 m  
 Longitud al centro masa (lc): 0.3 m  
 Fricción del péndulo (b1): 0.05 Ns/m  
 Fricción del carrito (b2): 0.06 Ns/m  
 Momento de inercia (I): 0.006 kgm<sup>2</sup>  
 Set point del péndulo (rad): 1.57 rad  
 Set point del carrito (m): 0

Seleccione el algoritmo metaheurístico:

DE GA PSO

Figura 3.4: Sintonización péndulo invertido.



Sintonización del péndulo doble: Por último, el control en el péndulo doble es de seguimiento de trayectoria. Como se ha mencionado, se busca el seguimiento de una trayectoria circular, que puede ser modificada de acuerdo con los requerimientos del diseñador. Este sistema recibe las mismas variables que las vistas en la figura 3.3, solo que en esta ocasión, para cada una de las barras.

Figura 3.5: Sintonización péndulo doble.

- *Selección de método metaheurístico:* Si el ingreso de datos es correcto, se procede a seleccionar el método metaheurístico (algoritmo evolutivo o EA, por sus siglas en inglés) a utilizar para encontrar las ganancias óptimas para el controlador (PID/PD+). Así mismo, el usuario tiene la opción de modificar el tamaño de la población (soluciones iniciales), el número de generaciones (iteraciones del algoritmo) y el tamaño del archivo (soluciones factibles).

En la figura 3.6, se ha ingresado un tamaño de población de 200 individuos para MODE, con el objetivo de incrementarlas posibilidades de alcanzar el tamaño máximo permitido de soluciones factibles, que como se observa, es de 30.



Figura 3.6: Parámetros para Evolución Diferencial.

Figura 3.7: Parámetros para Algoritmo Genético.

En el caso de MOGA (figura 3.7), se solicita además de los parámetros recién mencionados, un parámetro adicional llamado  $\eta$  que tiene como funcionalidad controlar la variabilidad de la mutación o perturbación. Esta es una propiedad única de este algoritmo.

Figura 3.8: Parámetros para Optimización por Enjambre de Partículas.



Para MOPSO, se permite la modificación de 4 parámetros más: velocidad mínima, velocidad máxima,  $\alpha$  y  $\omega$ , tal como se muestra en la figura 3.8. La funcionalidad de estos parámetros está descrita en la sección *Marco referencial*.

- *Ejecución del algoritmo y visualización de datos:* Una vez se compruebe que los datos ingresados son válidos, se procede a la ejecución del algoritmo seleccionado. Como se ha mencionado con anterioridad, el tiempo de ejecución dependerá de las características del computador, así como de la cantidad de datos a procesar.



Figura 3.9: Confirmación del algoritmo para el péndulo simple.

En la figura 3.9 se observa que la interfaz arroja un mensaje que solicita la confirmación para ejecutar el algoritmo. Al mismo tiempo, se muestra una tabla del lado izquierdo con una leyenda que se mantendrá durante la ejecución del programa hasta que este termine el proceso. Será entonces cuando el mensaje será reemplazado por los valores de las ganancias y de las funciones objetivo.

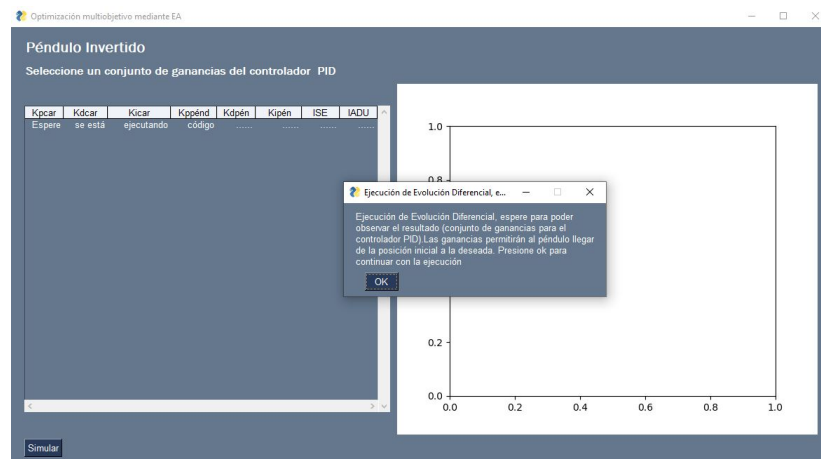


Figura 3.10: Confirmación del algoritmo para el péndulo invertido.

Para el péndulo invertido. En la figura 3.10, se muestra igualmente un mensaje con un botón que dice *OK* para que sea presionado y se lleve a cabo la ejecución del algoritmo. Nótese que además de las columnas que mostrarán el valor de las funciones objetivo, aparecen las columnas correspondientes a todas las ganancias obtenidas durante el proceso de sintonización.

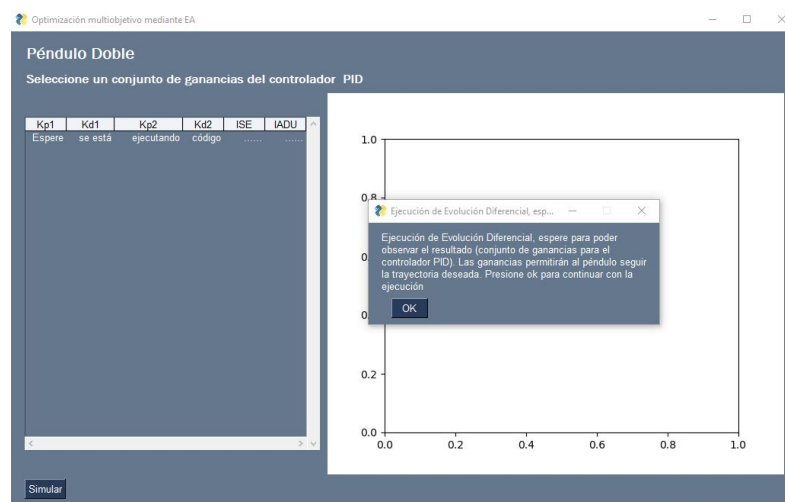


Figura 3.11: Confirmación del algoritmo para el péndulo doble.





En la figura 3.11 también se observa el mensaje de confirmación y la tabla que mostrará los valores de ganancias y funciones objetivo.

Al finalizar la ejecución, se observará en la pantalla la aproximación al frente de Pareto obtenida y una tabla con los datos correspondientes a la misma. En dicha tabla se mostrarán los valores de las funciones objetivo (ISE e IADU) y el valor de las ganancias  $K_p$ ,  $K_i$  y  $K_d$  correspondientes a cada uno de estos puntos, para péndulo simple e invertido, y las ganancias  $K_p$  y  $K_d$  para el péndulo doble. En la figura 3.12 se muestra lo mencionado anteriormente, en este caso, la sintonización fue para el péndulo simple mediante MODE.

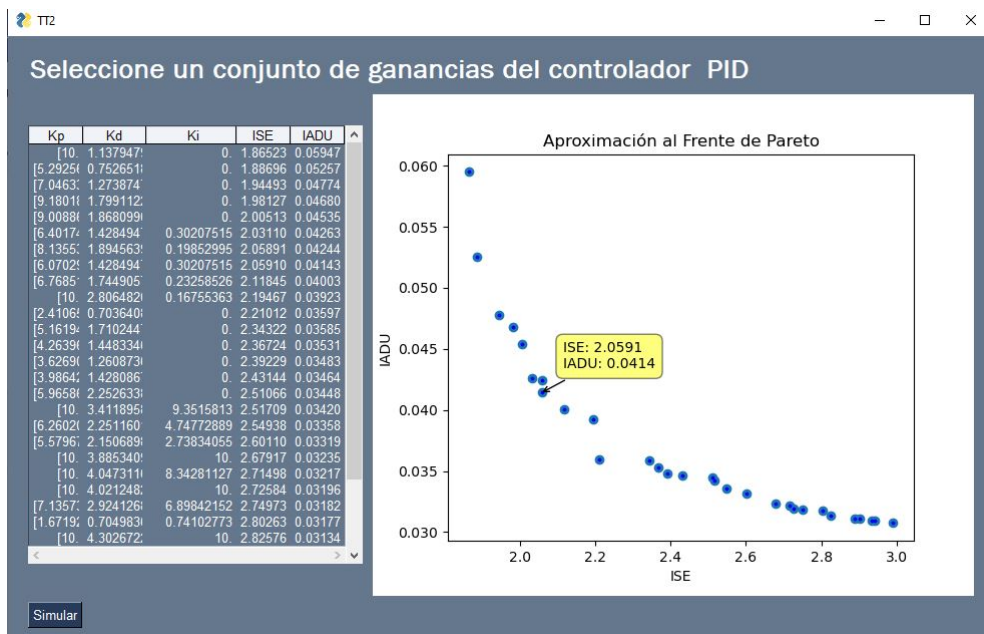


Figura 3.12: Resultados de la optimización multiobjetivo y aproximación al Frente de Pareto.

Se alcanzó el tamaño máximo de soluciones factibles y se puede apreciar que para el péndulo simple, MODE arroja un frente de Pareto diverso y distribuido. A pesar de que existe una solución que sobresale de la curva, todos los demás puntos se comportan como se esperaba. La tabla muestra el valor de ganancias



correspondientes a cada punto de la gráfica. Una pantalla similar se muestra al final de los demás sistemas y algoritmos. Los resultados de cada una de las sintonizaciones restantes, se mostrarán más adelante.

- *Simulación y graficación*: Finalmente, el diseñador podrá seleccionar una solución, con base en los criterios que el mismo decida. Esto corresponde a la parte de *Decisión multicriterio* descrita en la metodología para resolver problemas multiobjetivo. Esta parte es muy importante ya que, cada una de las soluciones representa un grado de compromiso entre los objetivos a alcanzar, por lo que cada una de estas soluciones es óptima, y por lo tanto, corresponde al diseñador seleccionar la que se ajuste a sus necesidades o preferencias. Una vez seleccionada una solución, se podrá visualizar una simulación y graficación del comportamiento de la planta ante las ganancias seleccionadas, todo dentro de la misma interfaz. Para ilustrar esto, se tienen las figuras 3.13 y 3.14

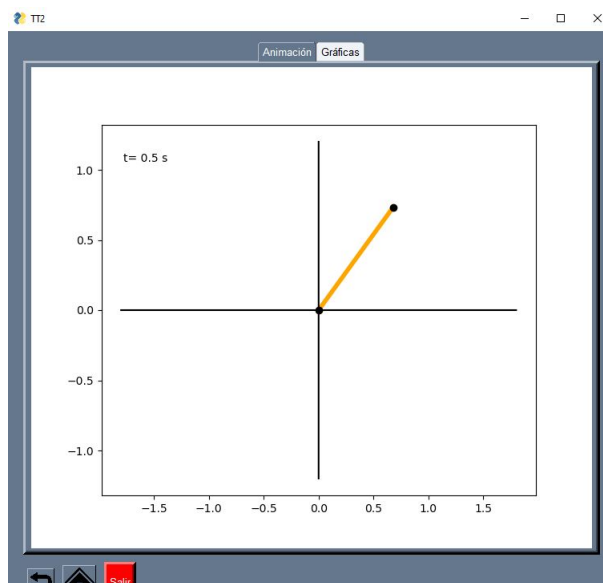


Figura 3.13: Simulación péndulo simple.

En la figura 3.13 se muestra una pestaña donde la barra de color naranja se



encontraba en su posición inicial  $setpoint = 0 \text{ rad}$ , esto es, apuntando verticalmente hacia abajo. Después, mientras avanza el tiempo, se mueve hacia la posición indicada por el usuario, que en este caso, era de  $setpoint = \pi$

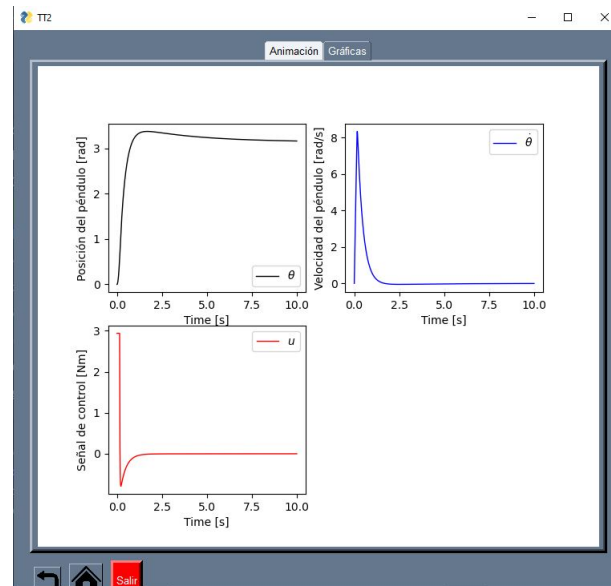


Figura 3.14: Respuesta de control péndulo simple.

Por otra parte, en la figura 3.14 se muestra otra pestaña dentro de la interfaz, la cual muestra las gráficas de posición y velocidad del sistema, así como de la señal de control. Se observa como existe un ligero sobreimpulso en la respuesta, y en poco tiempo se acerca a la estabilidad.

Finalmente, en la parte inferior de la pantalla se muestran un botón de regreso para probar otras soluciones, un botón para regresar al menú principal y un botón para cerrar el programa.

### Adaptación del péndulo simple

Comprender la ecuación del torque o par y la relación entre la velocidad y este es una parte importante en la selección y operación de un motor DC. Cuando la carga en el motor es constante, la velocidad es proporcional al voltaje suministrado. Y cuando



la tensión de alimentación es constante, la velocidad es inversamente proporcional a la carga del motor. Esta segunda relación, entre velocidad y carga (o torque), se muestra en la curva de torque-velocidad del motor.

Usando la definición del torque constante, tenemos:

$$\omega = \frac{1}{k_t} V - \frac{R}{k_t^2} \tau$$

Despejando la ecuación para el torque, se tiene que

$$\tau = \frac{V - (\omega - k_t)}{R} k_t$$

### Caracterización del motor

Se obtuvieron los parámetros del motor experimentalmente, realizando varias mediciones para determinar el torque, corriente y revoluciones por minuto (velocidad angular [RPM]) a diferentes valores de tensión. El equipo utilizado fue el siguiente:

- 1 base para mesa
- 8 nueces
- 4 varillas de enchufe
- 1 pinza para motor
- 1 fuente de alimentación
- 1 tacómetro digital láser
- 2 multímetros
- 1 regla
- 2 dinamómetros



Para la medición de la velocidad angular, se colocó un tacómetro digital apuntando el láser perpendicularmente al eje del motor para obtener la velocidad del motor en *RPM* de este a distintos valores de tensión, como puede verse en la figura 3.15 . En este caso, los incrementos fueron de 0.5 volts, desde el valor más bajo de tensión necesario para romper la inercia del motor hasta el valor de tensión nominal de acuerdo con el fabricante.



Figura 3.15: Diagrama para la estimación de parámetros ( $\omega$ ).@



Figura 3.16: Mediciones de corriente y tensión para el cálculo de la resistencia.

Por otra parte, para calcular la resistencia del motor, se midió corriente a distintos niveles de tensión, como se muestra en la figura 3.16, mientras se aplicaba una carga



al eje del motor.

Los datos obtenidos están presentes en la tabla 3.1. Se realizaron 13 mediciones para obtener la corriente mediante ley de Ohm.

$$R = \frac{V}{I}$$

Tabla 3.1: Medición de la resistencia.

Tensión [V]	Corriente [mA]	Resistencia [ $\Omega$ ]
3.19	0.712	4.48033708
3.13	0.745	4.20134228
3.13	0.746	4.19571046
3.14	0.749	4.19225634
3.13	0.75	4.17333333
2.91	0.913	3.18729463
3.15	0.729	4.32098765
2.89	0.911	3.17233809
3.13	0.751	4.1677763
3.14	0.75	4.18666667
2.91	0.907	3.20837927
4.7	1.12	4.19642857
4.66	1.102	4.22867514

Posteriormente se obtuvo un promedio, dando como resultado una resistencia de  $R = 4.192\Omega$

En la tabla 3.2 se muestran los resultados de 19 mediciones para el cálculo de  $k_t$ . Como se mencionó anteriormente, se realizaron incrementos de 0.5 V en cada medición.



Tabla 3.2: Medición de la velocidad.  
Tensión [V]    Corriente    Velocidad  
                                 [mA]                    [RPM]

Tensión [V]	Corriente [mA]	Velocidad [RPM]
3.11	0.25	48.23
3.47	0.85	54.54
4.06	0.87	64.83
4.53	0.89	72.83
5.05	0.92	81.92
5.5	0.93	90
6	0.95	98.74
6.48	0.97	106.92
7.09	1.01	117.88
7.54	1.02	125.83
8.02	1.05	134.16
8.46	1.06	141.86
9.06	1.09	152.68
9.55	1.09	160.87
9.99	1.1	169.1
10.5	1.11	178.45
11	1.13	187.95
11.49	1.14	196.5
11.98	1.15	204.99

En secciones anteriores se había mencionado la siguiente expresión:

$$\omega = \frac{1}{k_t} V - \frac{R}{k_t^2} \tau$$

Esta ecuación, de pendiente  $\frac{R}{k_t^2}$  es conocida como la curva *velocidad-torque* para un cierto voltaje V. Esta curva traza todas las posibles condiciones de operación de corriente constante con voltaje V en el motor. Suponiendo que el par de fricción es



cero, la línea intercepta el  $\tau = 0$  en

$$\omega_0 = \frac{V}{k_t} \quad (3.1)$$

De esta ecuación, podemos despejar la ecuación para obtener  $k_t$

Realizando un promedio de todos los cálculos para  $k_t$ , obtenemos un valor de  $k_t = 0.05992211714$

A su vez, la gráfica de tensión vs velocidad se muestra en la figura 3.17

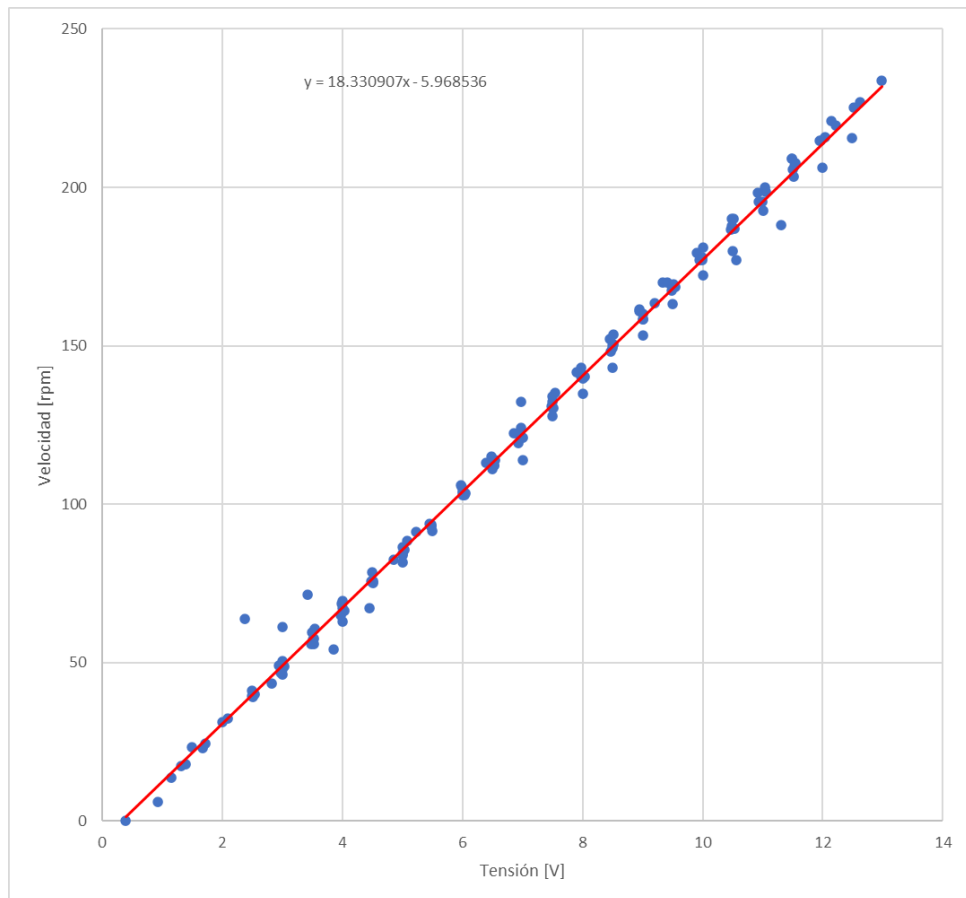


Figura 3.17: Tensión vs velocidad.





Por otra parte, la gráfica torque-velocidad se muestra en la figura 3.18

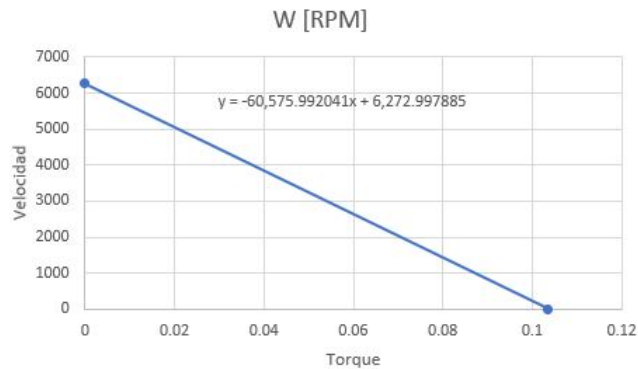


Figura 3.18: Torque vs Velocidad.

En la figura 3.17, se muestra la relación que presenta la tensión de alimentación y la velocidad en el eje de salida, misma que se usó para el cálculo de la constante  $k_t$ , adicionalmente, se trazó la curva Torque vs Velocidad mostrada en la figura 3.18, con la cual se puede saber un comportamiento idealizado del motor real.

## Montaje

Tal y como se ha descrito en la sección *Adaptación del péndulo simple* y como se observa en la figura 2.26, se consiguieron los materiales y componentes necesarios para montar el sistema del péndulo simple sobre una tabla de madera. En esta figura se aprecia la distribución de los elementos que conforman el sistema y representa la aproximación deseada.

El montaje final es el que se aprecia en la figura 3.19. Aquí se observa la colocación de tornillos en las chumaceras y de abrazaderas omega para darle estabilidad al motor. Además, dentro del gabinete negro se encuentra la tarjeta Teensy, el *driver* L298M y el regulador de tensión de 12 V. Se recomienda que el computador se mantenga cerca del área de control para facilitar la conexión entre ambos.



Figura 3.19: Sistema péndulo simple.

A continuación, se describe cada una de las partes que componen al montaje del sistema.

### Motor

En la figura 3.20, se muestra el motor seleccionado, el cual se encuentra fijo a una tabla de madera usando dos abrazaderas para tubos, y en el eje se colocó una ciruela o acoplamiento flexible de medidas de diametro 8 mm en ambos lados, y una longitud de 25 mm, esto con el fin de acoplar el eje del motor al eje principal.

Además, el motor cuenta con un encoder de resolución de 27,300 CPR (obtenida experimentalmente), el cual puede ser alimentado con una tensión de 3.3 V o 5 V, con dos canales de salida ( A y B), los cuales son conectados a los pines 2 y 3 de la tarjeta Teensy 4.1 respectivamente.



Figura 3.20: Motor.



### Prueba de resolución del encoder.

Para realizar esta prueba se hizo uso de la biblioteca de encoder para Arduino, la cual mide los CPR recibidos de un encoder. Primero se cargó el programa básico con el que cuenta esta biblioteca, se colocó el péndulo en la posición inicial de  $0^\circ$  o 0 conteos, tal como se puede mostrar en la figura 3.21.

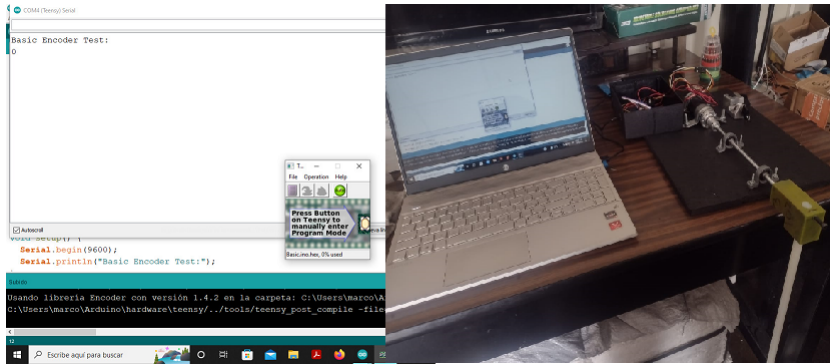


Figura 3.21: Posición 0.

De manera manual, se hizo girar el péndulo para leer los conteos en el puerto serie, encontrando que la resolución del encoder es de 27300 CPR, tal como se puede ver en la figura 3.22, donde se puede ver el péndulo en la posición  $2\pi$ .

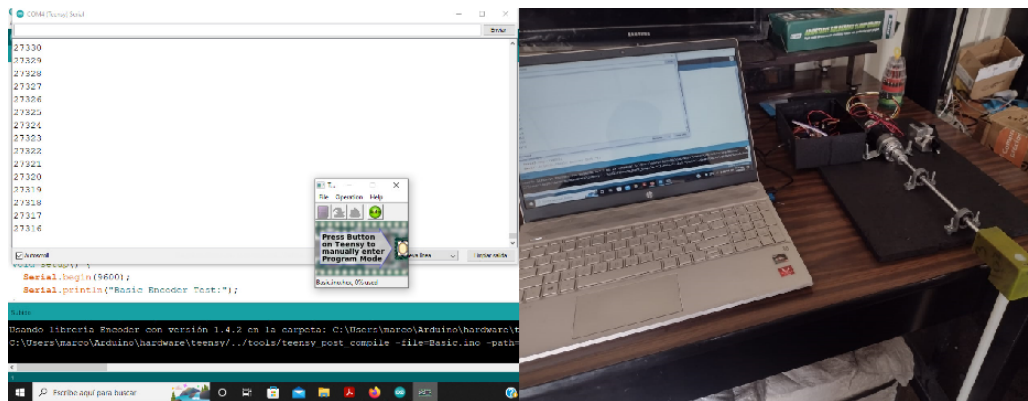


Figura 3.22: Posición  $2\pi$ .



### Chumaceras

Las chumaceras son soportes fijos que brindan estabilidad al sistema, ambas son de un diámetro de 8 mm, ajustadas de tal manera que el eje pueda moverse. Las chumaceras se pueden observar en la figura 3.23.



Figura 3.23: Chumaceras KP08.

### Elemento de sujeción.

El elemento mostrado en la figura 3.24 es una pieza impresa en 3D, sujeta a presión al eje del sistema, y cuenta con un barreno de 1/2" para poder colocar la barra, además cuenta con barrenos donde se colocan tornillos para poder ajustar la barra al elemento de sujeción.



Figura 3.24: Elemento de sujeción.



## Control del péndulo simple

Los canales A y B del encoder del motor, están desfasados 90 grados, con el propósito de percibir el sentido de giro del motor. Esto se comprobó con el apoyo de un osciloscopio.

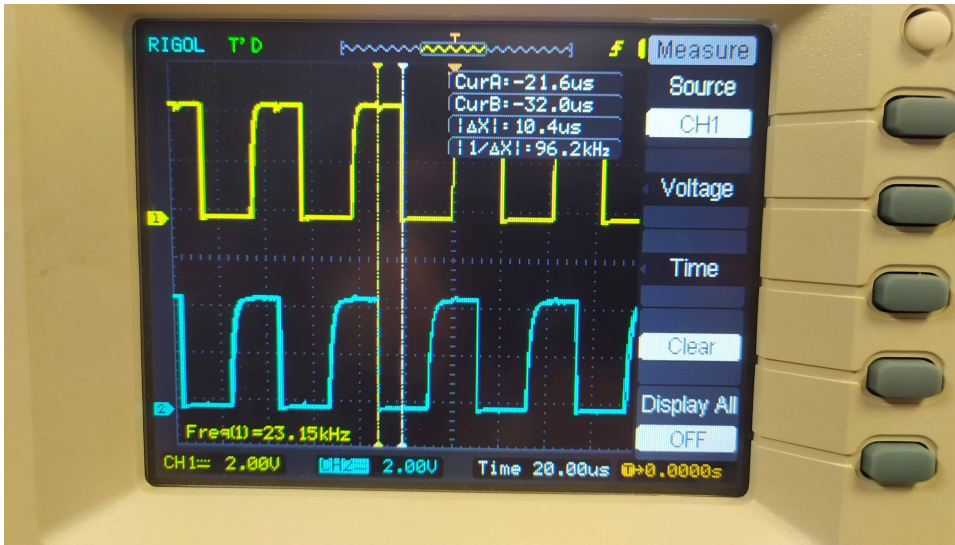


Figura 3.25: Canales del encoder.

El ángulo se obtiene mediante la siguiente ecuación

$$\phi = \omega \Delta t$$

Donde  $\omega = 2\pi F$

Con los datos mostrados en la figura 3.25, tenemos que

$$\phi = 2\pi(23.15kHz)(10.4\mu s) = 1.5127rad = 86.67grados$$

Lo cual es un valor muy cercano a los 90 grados, y por lo tanto es un valor aceptable.

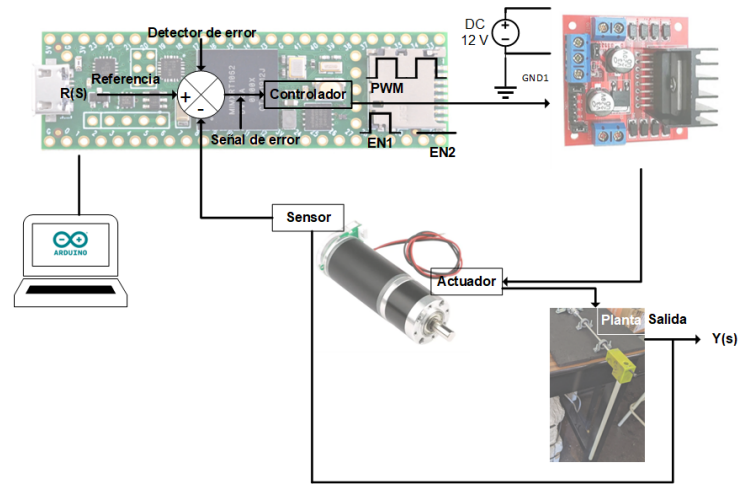


Figura 3.26: Implementación del control PID.

En la figura 3.26, se muestran todos los componentes usados en la implementación del control del péndulo simple, todos los elementos se encuentran identificados por la labor que desempeñan en el esquema de control presentado en la figura 1.1.

La tarjeta Teensy es programada gracias a la extensión del IDE de Arduino conocida como Teensyduino, el programa cargado a la tarjeta Teensy 4.1 realiza la labor de recibir la retroalimentación del encoder, convertir las señales de los canales en información de la posición del péndulo, posteriormente compara la referencia y obtiene la señal de error, realiza los cálculos de la señal de control usando las ganancias que se obtienen de la sintonización del método metaheurístico con el mejor desempeño, es decir de la señal PWM que sale de la tarjeta, así como dos salidas que definen el sentido de giro del motor, estas señales son enviadas al driver L298N, el cual está alimentado a una fuente de 12 V a 12.5 A. Este driver desempeña el papel de la etapa de potencia, teniendo como salida dos señales que son enviadas al motor. El motor es el actuador que tiene como salida el torque que permite que la planta se mueva y reciba las señales eléctricas a su entrada.

La salida de la planta es la posición actual, la cual es retroalimentada hacia la tarjeta Teensy.



---

## Resultados

---

En este capítulo se presentan los resultados obtenidos en las simulaciones, es decir, las aproximaciones al frente de Pareto correspondientes a cada proceso de sintonización y el valor de las variables objetivo y de diseño. Además, se describe el proceso realizado para encontrar el algoritmo con mejor desempeño para la sintonización *offline* de un control PID en un péndulo simple. Finalmente, se muestra el montaje final de dicho sistema y los resultados de realizar una sintonización de controladores PID mediante algoritmos evolutivos para la optimización multiobjetivo.

### Simulaciones

Se realizó la sintonización PID de cada uno de los péndulos, a través de los 3 algoritmos evolutivos, dando como resultado 9 sintonizaciones cuyos objetivos fueron optimizar los índices ISE e IADU.

Los parámetros utilizados en cada una de las ejecuciones de los algoritmos fueron los mostrados en la tabla 4.1. Donde CR es la tasa de recombinación *Crossover Rate*, F el factor de mutación,  $\eta$  el índice de distribución y por último  $c_1$  y  $c_2$  son factores empleados para ponderar el conocimiento local y global del enjambre. Los valores y métodos mostrados en la tabla 4.1 fueron establecidos con base en los diversos artículos y textos mostrados en los Antecedentes. Cada prueba se realizó





con una población de 100 individuos como tamaño de población, se ejecutaron 100 generaciones y se propuso un tamaño máximo de archivo de 30 individuos.

Tabla 4.1: Parámetros de optimización

Método de optimización	Parámetro de optimización	Valor/Método
Differential Evolution	CR	0.7
	F	0.5
Genetic Algorithm	Cruza	SBX
	Mutación	Polinomial
	$\eta$	1
Particle Swarm Optimization	$c_1$	1
	$c_2$	1

### Péndulo simple

Como primer planta a sintonizar, se tiene al péndulo simple. Se ejecutó la interfaz para este sistema mediante cada uno de los algoritmos evolutivos mencionados anteriormente.

Nota: En esta sección, con motivos de simplificación, se usan los términos DE, GA y PSO para referirse a MODE, MOGA y MOPSO, respectivamente. La aproximación del frente de Pareto para el péndulo simple mediante DE se muestra en la figura 4.1. Se encontraron las 30 soluciones esperadas y se observa una distribución de las soluciones aceptablemente uniforme con forma de rodilla invertida. Para los índices ISE e IADU, lo que se busca es el valor mínimo en ambos casos, tal que, las soluciones encontradas sean soluciones inmejorables, por lo que, cuanto más se aproximen al 0, se espera encontrar una buena aproximación al frente de Pareto, lo cual será verificado más adelante con las pruebas de hipervolumen.

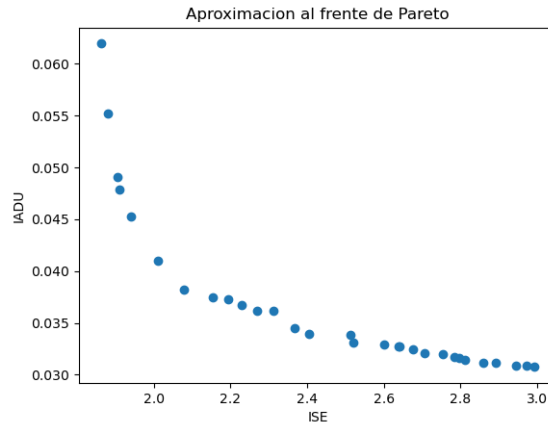


Figura 4.1: Sintonización PID del péndulo simple con DE.

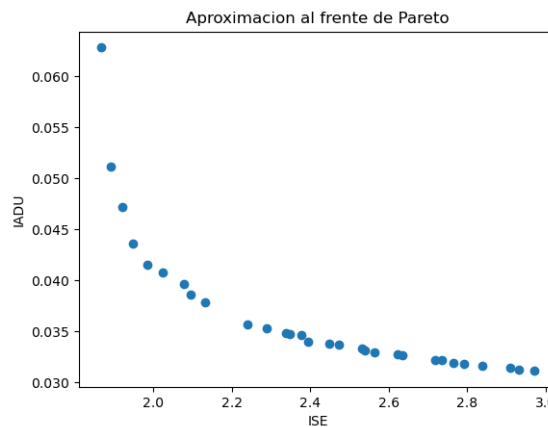


Figura 4.2: Sintonización PID del péndulo simple con GA.

De manera similar, la aproximación al frente de Pareto mediante GA mostrada en la figura 4.2 ofrece las 30 soluciones esperadas en el archivo e incluso una distribución (visualmente) más uniforme que con DE. Sin embargo, no será hasta obtener el hipervolumen cuando se verá si la calidad de este frente es mejor que la del obtenido con DE, pues ambos tienen forma de rodilla invertida y las soluciones tienden a ser bajas, es decir, lo más cercano a 0 posible para ambos índices.

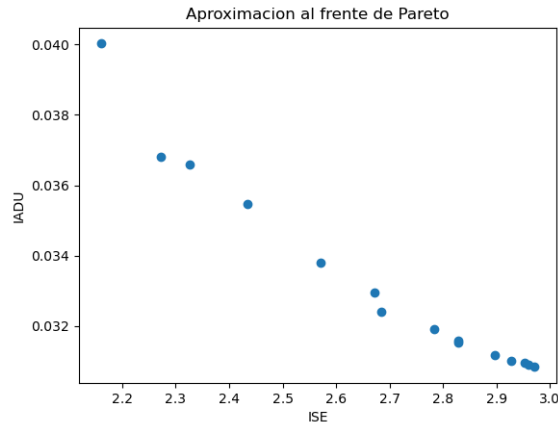


Figura 4.3: Sintonización PID del péndulo simple con PSO.

Por último, la aproximación obtenida mediante PSO mostrada en la figura 4.3 encontró 15 soluciones y la distribución es notablemente diferente a las mostradas anteriormente. No se puede reconocer la forma de rodilla invertida y tiende más a parecer una línea diagonal, por lo que, visualmente, se puede afirmar que es la peor aproximación al frente de Pareto, lo cual puede ser verificado con el valor de hipervolumen.

En la Tabla 4.2 se muestran 3 valores destacables de cada aproximación al frente de Pareto: Los extremos (primer y tercer valor de cada renglón) y el punto central (segundo valor), así como el valor de las variables de diseño correspondientes a cada uno de estos puntos. Es posible ver que, para este caso, GA presenta un frente de Pareto más distribuido y uniforme. Por otra parte, el valor de hipervolumen de este es muy cercano al de DE, sin embargo, este último presenta el mejor valor de este indicador de calidad.



Tabla 4.2: Sintonización PID del péndulo simple.

EA	$ISE$	$IADU$	$K_p$	$K_d$	$K_i$	$HV$
	2.79795155	0.03160511	4.40354265	1.87575189	3.67058561	
DE	2.94386074	0.03085924	10	4.66244081	8.49653619	17.5745083
	2.99372319	0.03073905	8.29989667	3.96275114	7.50806333	
	1.86541265	0.06282907	10	1	0.26616481	
GA	2.39375131	0.03400521	2.01583865	0.68238635	0.26375869	17.5643317
	2.97231949	0.03112209	2.01054083	0.94555266	1	
	2.15969003	0.0400267	9.23385897	2.43733758	1.05845202	
PSO	2.78254336	0.0319041	5.59463321	2.27984265	6.93528893	17.2870227
	2.97142589	0.03083343	6.2814545	2.9523348	5.88466347	

**Péndulo invertido**

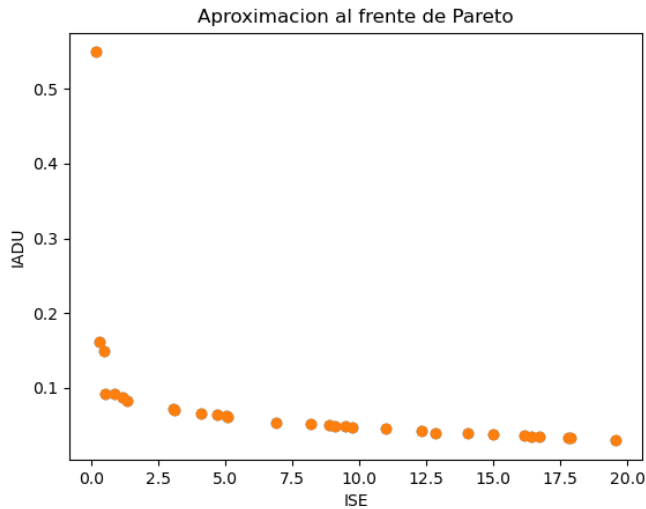


Figura 4.4: Sintonización PID del péndulo invertido con DE.

La aproximación al frente de Pareto mostrada en la figura 4.4 para el péndulo invertido mediante DE muestra 27 soluciones uniformemente distribuidas. Se observa que las soluciones están más cercanas al origen (0,0), es decir, se espera que el valor del hipervolumen sea alto. Por otra parte, conserva la forma de rodilla invertida, tal



como se obtuvo en el péndulo simple para este algoritmo.

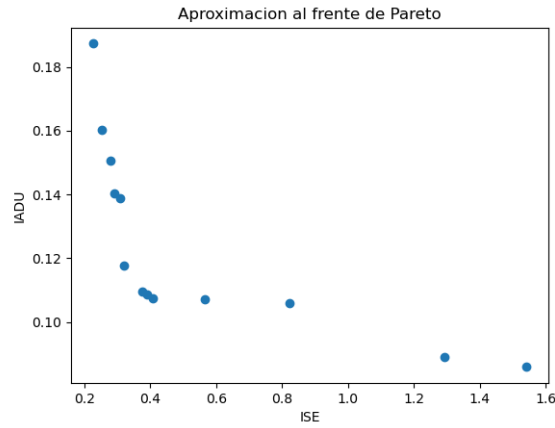


Figura 4.5: Sintonización PID del péndulo invertido con GA.

La figura 4.5 muestra la aproximación mediante GA para el péndulo invertido. La distribución no es tan uniforme como con DE y encontró 15 soluciones factibles. La forma de rodilla invertida es poco menos evidente, pero aún se conserva, como lo que se observó en el frente de Pareto obtenido con este algoritmo para el péndulo simple.

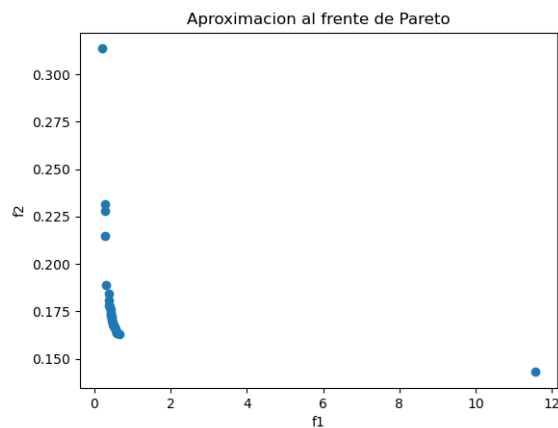


Figura 4.6: Sintonización PID del péndulo invertido con PSO.



La figura 4.6 muestra la aproximación obtenida mediante PSO. Se aprecia como la mayoría de las soluciones se concentran en una sola región, mientras que 6 de ellas están alejadas de esta región. El resultado es más parecido al de DE que al de GA. Además, como en el péndulo simple para este algoritmo, no se observa la forma de rodilla invertida, las soluciones se muestran muy aglomeradas en una misma región, pero si son soluciones que se aproximan al 0, por lo que, por inspección visual no se puede decidir que frente tiene la mejor calidad.

En la tabla 4.3 está contenida la información de las soluciones encontradas, así como del valor de hipervolumen para cada aproximación al frente de Pareto. Las variables con subíndice [1] corresponden al móvil y las variables con subíndice [2] son del péndulo.

Tabla 4.3: Sintonización PID del péndulo invertido.

EA	ISE	IADU	$Kp_1$	$Kd_1$	$Ki_1$	$Kp_2$	$Kd_1$	$Ki_1$	HV
	0.14517701	0.54932586	10	6.94656638	1.78074561	18.85084981	0.63957134	0.77787679	
DE	8.88217712	0.04966641	5.5135481	4.76179801	0	4.24823245	0.64776916	0	18.7444787
	19.58632049	0.02942955	8.60920948	10	0.29891151	2.91189265	1.12619857	0.10269126	
	2.26E-01	1.87E-01	3.50E-01	1.00E+01	5.40E-01	6.08E-03	1.88E+00	1	
GA	0.37433371	0.10953617	1	0.53498121	0.34582573	6.52287712	0.43570658	0.03369912	18.0490645
	1.54194071	0.08584723	1	1	0.88850912	5.09719581	0.99998751	0.01958774	
	0.19028324	0.31366166	3.54946614	6.3250299	3.09218412	14.8448656	1.02706693	9.09978205	
PSO	0.45307318	0.17123145	5.78491483	9.54208709	2.966472	7.47951881	0.91751088	4.98567465	16.9504352
	11.57931198	0.14295489	3.07989156	1.36111042	1.99595474	3.6884509	13.21565665	4.88965427	

De manera similar que en el péndulo simple, DE presenta notoriamente un mayor valor de hipervolumen, además, también la aproximación al frente de Pareto obtenida con este algoritmo presenta una mayor cantidad de soluciones no dominadas, a su vez, se observa una mejor distribución de estas soluciones en el espacio de objetivos. Se identifica también que con GA, los valores de las ganancias y de las funciones objetivo son muy pequeños. Adicionalmente, el valor de hipervolumen del frente arrojado con PSO, es considerablemente diferente a los encontrados con DE y GA.



## Péndulo doble

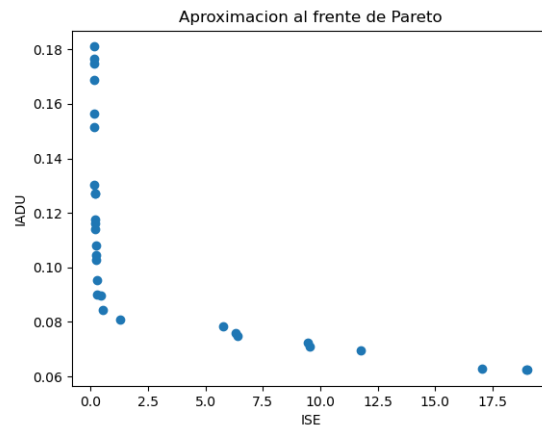


Figura 4.7: Sintonización PD del péndulo doble con DE.

La aproximación al frente de Pareto con GA para el péndulo doble se muestra en la figura 4.8. Se encontró una cantidad menor de soluciones, pero la distribución es uniforme y diversa.

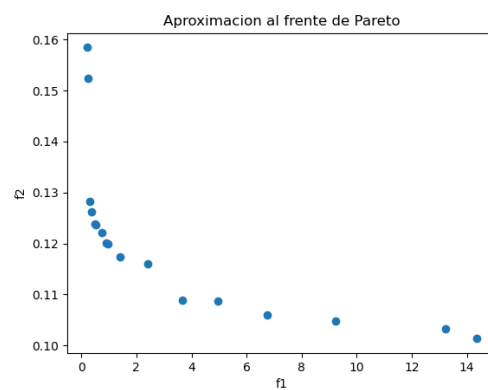


Figura 4.8: Sintonización PD del péndulo doble con GA.

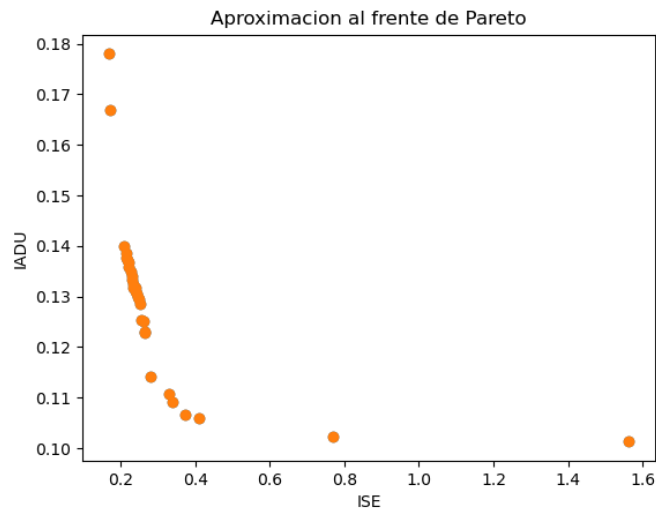


Figura 4.9: Sintonización PD del péndulo doble con PSO.

Por último, la aproximación al frente de Pareto con PSO para el péndulo doble se muestra en la figura 4.9. Algunas de las 30 soluciones se concentran en una región, sin embargo, están cerca del origen (0,0).

La tabla 4.4 contiene el valor de las funciones objetivo y el de las variables de diseño. Las variables con subíndice [1] corresponden al péndulo 1 y las variables con subíndice [2] son del péndulo 2.

GA ofrece un frente de Pareto mas uniforme, sin embargo, como se puede ver en la tabla 4.4, arroja el valor más bajo de hipervolumen, mientras que el más alto lo tiene el DE. También se observa que el valor de hipervolumen de PSO se acerca al de GA. Para los tres casos, el frente de Pareto tiene forma de rodilla invertida, variando un poco en la diversidad de soluciones.

En todos los casos, el resultado de PSO resultó ser el peor en comparación con ED Y GA, siendo estos los que ofrecían un frente de Pareto con mejores características. No obstante, este no es un motivo suficiente para descartar PSO y escoger DE





o GA como los mejores algoritmos en la sintonización de controladores para optimización multiobjetivo, para ello, se tiene que hacer uso de una prueba estadística no paramétrica.

Tabla 4.4: Sintonización PD del péndulo doble.

EA	ISE	IADU	$Kp_1$	$Kd_1$	$Kp_2$	$Kd_2$	HV
	0.1342389	0.18099516	8	1.44014819	5	0.43725703	
DE	0.24289536	0.10265973	7.81243616	2.11718778	1.17308692	0.26369117	18.4048147
	19.0325164	0.06237051	0	2.04239868	0	0.95288188	
	0.19810246	0.15845554	4.53437349	1.19659824	2.06136316	0.28164648	
GA	0.97694364	0.11991636	1	0.46560978	0.11419582	0.09429843	17.6834146
	14.3682614	0.10134755	2.6818489	0.79316987	0.05784788	1	
	0.16691614	0.17798613	5.60445285	2.0629417	4.66669833	0.52898939	
PSO	0.24193539	0.13049257	5.02071908	1.93125259	2.88713151	0.60108592	17.8164381
	1.56361449	0.10129295	2.18196891	0.70802638	1.13807083	1.95251837	

## Prueba de Wilcoxon

Como se ha mencionado en capítulos anteriores, los indicadores de calidad son métricas ampliamente usadas en optimización multiobjetivo, ya que nos sirven para analizar la calidad de conjuntos de soluciones. Una de las métricas más populares es hipervolumen, diseñada para medir los aspectos de convergencia y diversidad en un frente dado. Esta métrica calcula el volumen (en el espacio de objetivos) cubierto por miembros de un conjunto de soluciones no dominadas. Por esta razón, cuanto mayor sea el valor del hipervolumen, mayor será la calidad del frente, pues significará que se tiene una mayor diversidad y convergencia. Con el objetivo de evaluar la calidad del frente de Pareto para determinar que EA tiene un mejor desempeño en este tipo de sistemas, se requiere alguna prueba estadística no paramétrica que pueda comparar el hipervolumen obtenido de cada frente de Pareto. La prueba de Wilcoxon resulta ser una prueba adecuada para este caso. El enfoque elegido para esta prueba fue la comparación por pares, debido a su simplicidad y efectividad. El procedimiento a realizar es el siguiente



1. Organizar los datos por carpetas
2. Ejecutar el algoritmo en lenguaje R que realiza la prueba de Wilcoxon
3. Recopilar los datos obtenidos para su análisis
4. Realizar el conteo de victorias en cada algoritmo para cada uno de los sistemas

Cada uno de los EA (DE, GA y PSO) correspondientes a un sistema mecatrónico (péndulo simple, invertido y doble), fue colocado dentro de un ciclo *for* para que cada algoritmo fuera ejecutado 30 veces, ya que para que la prueba de Wilcoxon sea lo suficientemente aceptada, se recomienda que el tamaño de la muestra sea mayor o igual a 30. Dicho de otra manera, se realizaron 9 programas en los que se ejecuta cada algoritmo 30 veces. De cada ejecución, se extrajo el valor de hipervolumen correspondiente al frente de Pareto obtenido en dicha ejecución y se fue almacenando en un archivo *.csv*. Este proceso fue realizado en los laboratorios de la UPIITA (figura 4.10) y en un computador externo con características de rendimiento similares, ya que el tiempo de ejecución por cada programa es de aproximadamente 3 días en promedio.

Una vez completado el proceso anterior, se recopilaron los documentos y se almacenaron en un computador para su posterior análisis.

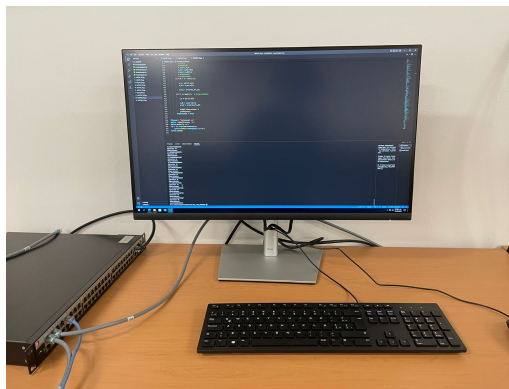


Figura 4.10: Ejecución de los programas en laboratorios de la UPIITA.



Tabla 4.5: Resultados de hipervolumen.

Péndulo simple			Péndulo invertido			Péndulo doble		
DE	GA	PSO	DE	GA	PSO	DE	GA	PSO
17.64264932	17.65222644	17.54652335	18.70199121	18.51704556	16.87320803	18.23282136	17.89481992	18.01741033
17.71747999	17.56919742	17.37068679	18.70303532	18.55283895	17.8714645	18.30973602	17.80654051	17.98464812
17.63562087	17.61465616	17.52988378	18.42486752	18.50037931	16.90082944	18.25480817	17.89963449	17.99447141
17.57486101	17.62485924	17.22009603	18.51021852	0	17.48298249	18.22649844	17.82301281	18.07341256
17.66921073	17.66834919	17.55757582	18.68579446	0	18.08789295	18.36621758	17.91506788	18.07659479
17.62150411	17.56989191	17.28262748	18.56232015	0	16.94425709	18.31120129	17.83795752	18.09652839
17.57419328	17.57441558	17.55819284	18.66566348	18.61141341	15.86976947	18.21941803	17.82634663	18.06914769
17.58152567	17.69358157	17.42366702	18.59637301	18.65091829	17.30196361	18.20373415	17.86025887	18.08201444
17.57616764	17.56726398	17.46746223	18.58525859	0	16.83046266	18.1994802	17.89319641	17.99819732
17.5755701	17.61748474	17.46532798	18.68044345	18.58938462	16.89017035	18.20742322	17.86581508	18.01402929
17.58495258	17.62625375	17.52017871	18.73371747	18.60023255	17.56350631	18.35934139	17.82738892	17.92131608
17.57575765	17.56728691	17.51293505	18.75932745	18.60566738	16.51198344	18.26511012	17.8853194	18.11363695
17.57552725	17.66060036	17.44520179	18.66609504	18.63675225	16.57220417	18.28198913	17.87928394	17.67945709
17.57530876	17.68551069	17.3698651	18.5334966	0	17.58429657	18.34123941	17.85727429	18.2585818
17.6768002	17.6351	17.38360347	18.72178726	18.59270164	17.70111536	18.28477085	17.8218547	18.06108431
17.64997649	17.5696858	17.49247593	18.69153764	0	18.0503266	18.22035369	17.85035084	18.11558162
17.57240499	17.56719459	17.45978436	18.66567583	18.59552221	17.3789039	18.1776922	17.9014418	18.17129481
17.66251419	17.6128817	17.43749032	18.65102587	18.51973712	17.27920302	18.26730239	17.87043227	17.94688936
17.5764138	17.56512234	17.33544962	18.59695436	0	16.85847457	18.18683572	17.87796036	17.89572372
17.57450914	17.58885924	17.34347215	18.65964573	18.60318357	17.66308216	18.22542229	17.797157	18.17731871
17.57539851	17.6378457	17.47926663	18.59003188	18.6274126	17.16430302	18.29197489	17.86064254	17.94080114
17.57448409	17.65723528	17.46469569	18.52981562	0	17.07907308	18.38874287	17.83563053	18.10475867
17.57535263	17.65531889	17.44555608	18.60741364	18.56241656	18.13027615	18.3289322	17.79798352	18.17605587
17.61433039	17.66447793	17.43921693	18.44384841	18.5849995	17.59416982	18.44608243	17.87751793	18.32314388
17.58815982	17.67888923	17.35833578	0	18.46793564	16.72487495	18.21695966	17.85331159	18.04590654
17.64198373	17.61135208	17.3137391	18.54632531	0	17.22351365	18.21597149	17.92620229	18.15833958
17.61189658	17.69721143	17.34370689	18.61221538	17.81906074	17.19766474	18.29825261	17.87661581	17.98310205
17.6842208	17.57885997	17.42946998	18.63979752	18.53284535	17.21454299	18.25251804	17.81367629	18.21743888
17.70258128	17.61753508	17.48636321	18.35540815	0	17.42143162	18.31496118	17.83725001	17.99391207
17.57527399	17.64213037	17.26163765	18.56327933	0	16.92715629	18.36118782	17.91461809	18.28106639



En la tabla 4.5, se muestra el valor de hipervolumen obtenido en cada ejecución para cada uno de los péndulos. Es posible observar que, para el péndulo simple, todos los resultados tienden a converger a un valor mayor que 17 y menor que 18. Para el péndulo invertido, se observan valores mayores de 16 y menores de 19, incluso, se puede apreciar que con GA, hubo ejecuciones en las que no se encontró un frente de Pareto y por lo tanto, el hipervolumen fue 0, posiblemente debido a una indeterminación. Por último, el rango de valores para el péndulo doble va desde los 17 hasta los 19, sin ninguna indeterminación. El valor promedio de hipervolumen para cada uno de los EA aplicados a los péndulos, se muestra en las tablas 4.6. 4.7 y 4.8

Tabla 4.6: Valor promedio de HV para el péndulo simple.

EA	Promedio
DE	17.61122099
GA	17.62237592
PSO	17.42481626

Tabla 4.7: Valor promedio de HV para el péndulo invertido.

EA	Promedio
DE	17.98944547
GA	11.73901491
PSO	17.2297701

Tabla 4.8: Valor promedio de HV para el péndulo doble.

EA	Promedio
DE	18.27523263
GA	17.85948541
PSO	18.0657288

Como se mencionó antes, un valor mayor de hipervolumen indicaría una calidad mayor del frente de Pareto, por lo que sería sencillo decir que, por ejemplo, para el péndulo simple, GA es el mejor algoritmo, y para el invertido y doble, DE. Sin



embargo, sería contraproducente tomar como válidas estas suposiciones, ya que la distribución de los datos no es uniforme, debido a la aleatoriedad propia de los métodos metaheurísticos. La prueba de Wilcoxon atiende este problema, por lo que los resultados de la tabla 4.5 fueron sometidos a esta prueba. El nivel de significancia  $\alpha$  se estableció de 5%, este valor está expresado en las tablas como *p-value* y nos indica la probabilidad de que se acepte la hipótesis nula  $H_0$  (no hay una diferencia significativa o relevante entre un algoritmo y otro) y en consecuencia se descarte la alternativa  $H_1$  (por ambos lados, es decir A esta en posición diferente de B), de modo que si *p-value* es grande ( $>5\%$ ) se acepta la nula, y si *p-value* es pequeño ( $\leq 5\%$ ) se acepta la alternativa.

Tabla 4.9: Prueba de Wilcoxon para el péndulo simple.

Test (A vs B)	$R_+$	$R_-$	<i>p-value</i>
DE vs GA	184	281	0.328469857573509
<b>DE</b> vs PSO	465	0	1.86264514923096e-09
<b>GA</b> vs PSO	465	0	1.86264514923096e-09

La tabla 4.9,  $R_+$  indica que DE fue mejor que GA 184 veces, mientras que  $R_-$  indica que GA fue mejor que DE 281 veces, sin embargo, *p-value* es mayor que el 5%, siendo este del 32.84% causando la validez de la hipótesis nula. Por otra parte, DE resultó 465 veces ganador al compararlo con PSO, mientras que este último tuvo nulas victorias frente a DE. Finalmente, GA resultó victorioso en su competencia con PSO, por 465 comparaciones. En estos dos últimos casos, el *p-value* fue menor al establecido, por lo que se valida la hipótesis alternativa y se selecciona al algoritmo con más victorias.

En el caso del péndulo invertido, en la tabla 4.10 se observa que esta vez, DE tuvo más victorias frente a GA, además de que el valor de significancia fue menor al 5%, por lo que la hipótesis alternativa es válida y se selecciona a DE como el ganador. De manera similar, en la prueba del mismo algoritmo contra PSO, de nuevo se valida



Tabla 4.10: Prueba de Wilcoxon para el péndulo invertido.

Test (A vs B)	$R_+$	$R_-$	$p - value$
<b>DE</b> vs GA	416	49	4.96841967105866e-05
<b>DE</b> vs PSO	435	30	3.790482878685e-06
GA vs <b>PSO</b>	190	275	0.393050180748105

$H_1$  y DE resulta el vencedor. Finalmente, al comparar GA vs PSO, se tiene que este último obtuvo más victorias, sin embargo el nivel de  $\alpha$  es mayor al 5 %

Tabla 4.11: Prueba de Wilcoxon para el péndulo doble.

Test (A vs B)	$R_+$	$R_-$	$p - value$
<b>DE</b> vs GA	465	0	1.86264514923096e-09
<b>DE</b> vs PSO	465	0	1.86264514923096e-09
GA vs <b>PSO</b>	14	451	2.04890966415405e-07

Por último, la tabla 4.11 contiene los resultados de la última prueba, la del péndulo doble. Se muestra la victoria de DE frente a GA y PSO, con un total de 465 victorias, y dado que  $\alpha$  es menor al 5 %, se valida  $H_1$ , seleccionando DE como el ganador. En cuanto GA vs PSO, este último obtuvo más victorias, y también se cumple  $H_1$ , por lo que se escoge PSO como el ganador.

Tabla 4.12: Conteo de victorias para cada EA.

EA	Péndulo simple	Péndulo Invertido	Péndulo doble	Totales
DE	1	2	2	5
GA	1	0	0	1
PSO	0	0	1	1

En la tabla 4.12, se tiene el conteo de victorias de cada algoritmo evolutivo en cada uno de los sistemas mecatrónicos. Es posible ver que para el péndulo simple, DE y GA empatan, para el péndulo invertido, destaca DE como el mejor algoritmo y



para el péndulo doble, DE y PSO son los mejores. Estos resultados coinciden con los obtenidos en los frentes de Pareto obtenidos con la interfaz. De manera general, DE consiguió 5 victorias, mientras que GA y PSO obtuvieron 1. Por lo que se concluye que para sistemas mecatrónicos, el mejor algoritmo evolutivo para la optimización multiobjetivo en la sintonización de controladores PID, es Evolución Diferencial.

## Control de un péndulo simple

Ahora que se puede saber con seguridad que el algoritmo con el mejor desempeño es DE, se procede a sintonizar el control PID del péndulo simple con este algoritmo.

Los parámetros dinámicos del péndulo simple que se usaron son los siguientes:

<b>Masa (m):</b>	<input type="text" value="0.0655"/>	kg
<b>Longitud (l):</b>	<input type="text" value="0.443"/>	m
<b>Longitud al centro masa (lc):</b>	<input type="text" value="0.2215"/>	m
<b>Fricción (b):</b>	<input type="text" value="0.05"/>	Ns/m
<b>Momento de inercia (I):</b>	<input type="text" value="0.006"/>	kgm <sup>2</sup>
<b>Set point (rad):</b>	<input type="text" value="3.1416"/>	rad

**Seleccione el algoritmo metaheurístico:**

Figura 4.11: Parámetros dinámicos del péndulo simple.

Como se puede notar en la figura 4.11 se están usando los parámetros reales de la planta, ahora los parámetros que se usaron para DE, se muestra en la figura 4.12.

Swapping the contents of a window

### Evolución diferencial

Ingresa los siguientes parámetros:

Tamaño de la población:

Número de generaciones:

Tamaño del archivo:

Figura 4.12: Parámetros DE.

Al sintonizar con estos parámetros los resultados son los que se muestran en la 4.13, y en el frente de Pareto que se muestra en la figura 4.13, este muestra una distribución uniforme y diversa en cuanto a los valores de las funciones objetivos, además de presentar una forma clara de rodilla invertida, lo cual es un buen indicador de la calidad del frente obtenido.

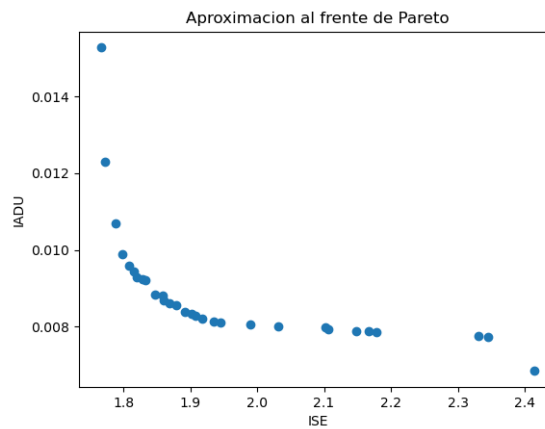


Figura 4.13: Aproximación al Frente de Pareto con DE.

Una vez sintonizado el péndulo simple y obtenidas las ganancias  $K_p$ ,  $K_i$  y  $K_d$  para el control PID mediante Evolución Diferencial, se procedió a seleccionar e implementar estas ganancias en el sistema mostrado en la figura 4.14





Tabla 4.13: Conjunto de soluciones factibles.

kp	kd	ki	f1	f2
7.03975963	1.06845197	0	1.83348906	0.00922052
5.18111254	0.86633943	0.14216748	1.86035679	0.00869079
4.49169683	0.61414812	0.0011566	1.81526779	0.00944281
3.42122877	0.46195965	0	1.81969997	0.00927438
8.4140798	1.4018695	0.39475791	1.85935655	0.00881363
4.08241627	0.57970728	0.20995218	1.82915721	0.00922395
4.22516627	0.51281756	0	1.79880693	0.00989111
9.77701477	1.8929288	0.27842587	1.90763642	0.00828887
8.85132283	1.65841274	0.6576294	1.90245461	0.00832357
4.0133615	0.51762136	0	1.80792101	0.009598
4.56162464	0.77836497	0	1.86951326	0.00860092
5.74043183	0.47997032	0	1.77303733	0.01231246
4.09658688	0.64221575	0	1.84694002	0.00884204
10	1.77877975	0	1.87852104	0.00856334
6.94148803	0.78925553	0	1.78776114	0.01068328
3.50495083	0.62259381	0	1.8915316	0.0083696
4.99625092	0.96308265	0.33282194	1.9172286	0.00821444
2.57804229	0.48439907	0	1.93524822	0.00813878
3.37162497	0.93210658	0	2.1667654	0.00788411
2.84812432	0.71389072	0.28306621	2.10700384	0.00792032
3.56551749	0.72015382	0.01723488	1.94509894	0.00810065
3.01021575	0.80533071	0	2.14836147	0.00788411
4.05077006	1.05402058	0	2.10193071	0.00798067
1.78486107	0.49318294	0.73764877	2.34562355	0.00773474
3.40177225	0.74164527	0	1.99013018	0.00804187
3.66323593	0.85949767	0	2.03161635	0.00801158
3.01021575	0.8325953	0	2.17820995	0.00785061
2.47445974	0.75459793	0.50847368	2.33104158	0.00776222
10	0.6553455	0	1.76619908	0.01528029
0.69336383	0	0.05031626	2.4149397	0.00683821



Figura 4.14: Péndulo simple

Se seleccionó el conjunto de ganancias que minimizan el objetivo  $J_1$  el cual es el indicador ISE, es decir, se seleccionó el conjunto solución que minimizara el error. Se estableció un *setpoint* de 180 grados y se realizaron las pruebas correspondientes. El resultado se muestra en la figura 4.15. Se aprecia que se llegó a la posición deseada.



Figura 4.15: Péndulo simple con control de posición.



### Propuesta alternativa

Con el objetivo de explorar los resultados obtenidos, se realizaron algunas variaciones al modelo propuesto del péndulo simple; se propuso intercambiar el motor DC por un motorreductor, utilizar un encoder incremental externo anclado al eje de acero inoxidable y colocar el sistema sobre una estructura metálica, con el fin de darle mayor estabilidad, tal y como se muestra en la figura 4.16

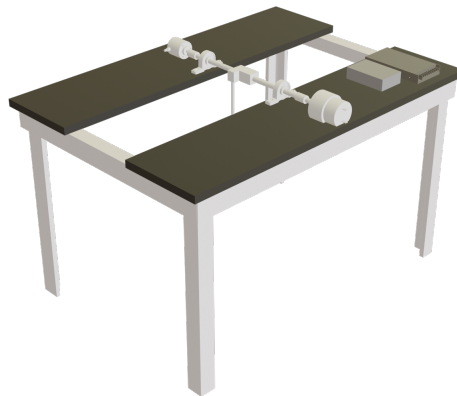


Figura 4.16: Propuesta alternativa para el péndulo simple.

Los elementos usados para construir este modelo, fueron los siguientes:

- Motorreductor Pololu 37Dx73L:

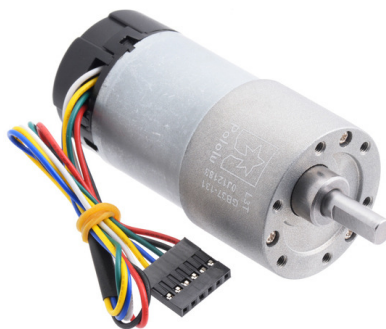


Figura 4.17: Motorreductor Pololu 37Dx73L



Este motorreductor es un motor DC con escobillas de 12 V con una caja de engranajes de metal con relación 131:1 y un codificador de cuadratura integrado que proporciona una resolución de 64 CPR del eje del motor, lo que corresponde a 8,384 CPR del eje de salida de la caja de engranajes. La caja de cambios se compone principalmente de engranajes rectos, sin embargo presenta engranajes helicoidales para la primera etapa con el objetivo de reducir el ruido y mejorar la eficiencia.

### Curvas características del motor

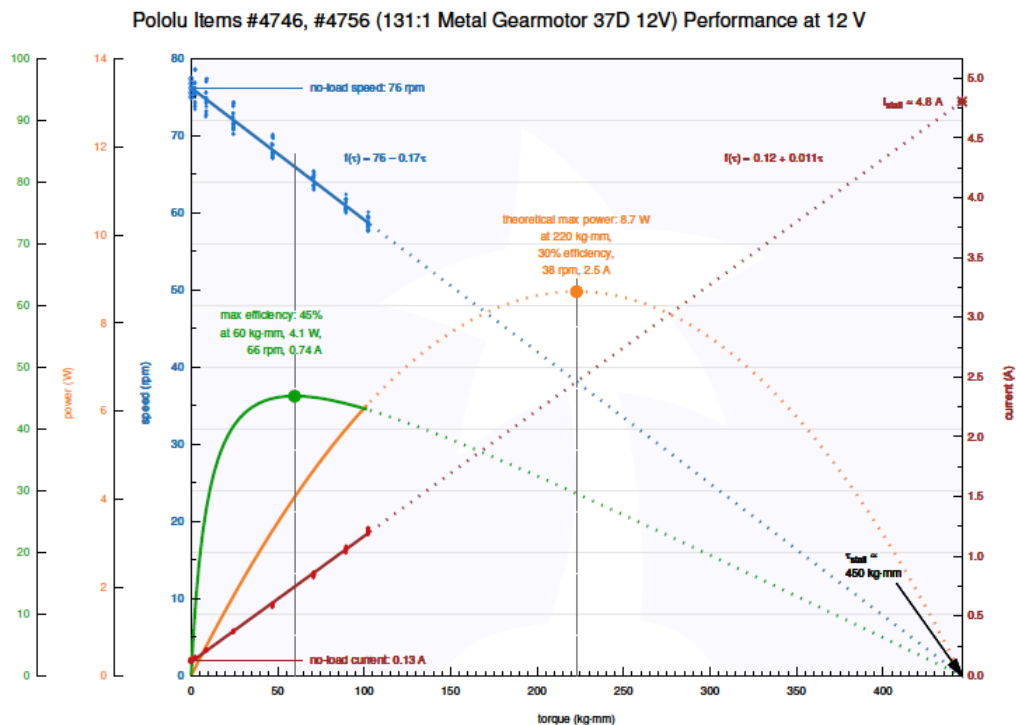


Figura 4.18: Curvas características del motor Pololu. Consultado en [1].

De acuerdo con las graficas mostradas en la figura 4.18 se identifican puntos importantes, la velocidad nominal sin carga y el torque nominal a corriente nominal. De acuerdo con ecuación 4.1, cuando el torque  $\tau=0$ , la ecuación se puede reducir a la ecuación que se muestra en la ecuación 4.2.



$$\omega = \frac{1}{k_t}V - \frac{R}{k_t^2}\tau \tag{4.1}$$

$$\omega = \frac{1}{k_t}V \tag{4.2}$$

El motor Pololu, es un motor de imán permanente, por lo que la velocidad y el torque nominal coinciden con el torque máximo. Por lo que, considerando los datos mostrados en la figura 4.19, se calcula el valor de kt.

Despejando la ecuación 4.2, para obtener kt se obtiene:

$$k_t = \frac{1}{\omega}V \tag{4.3}$$

Usando la ecuación 4:

$$k_t = \frac{12 \text{ V}}{98.07 \text{ rad/s}} = 0.1223$$

Rated Voltage	Pololu Item #	Gear Ratio	No Load		At Maximum Efficiency				Max Power W	Stall Extrapolation <sup>(2)</sup>		Graph Pages		
			Speed	Current	Speed	Torque	Current	Output		Torque	Current			
			:1 RPM	A	RPM	kg-mm	A	W		kg-mm	A			
12 V	4750 <sup>(1)</sup>	1	10,000							5				
	4747, 4757	6.25	1600		1300	4.9	1.2	6.4	12	30		5, 6		
	4748, 4758	10	1000		850	6.6	0.91	5.7	12	49		7, 8		
	4741, 4751	18.75	530		470	10	0.76	5.0	12	85		9, 10		
	4742, 4752	30	330	0.2	280	18	0.78	5.1	12	140	5.5	11, 12		
	4743, 4753	50	200		180	22	0.66	4.0	10	210		13, 14		
	4744, 4754	70	150		130	32	0.68	4.2	10 <sup>(3)</sup>	270		15, 16		
	4745, 4755	102.08	100		87	42	0.72	3.8	8 <sup>(3)</sup>	340		17, 18		
	4746, 4756	131.25	76		66	60	0.74	4.1	6 <sup>(3)</sup>	450		19, 20		
	2828, 2829	150	67		58	65	0.72	3.8	6 <sup>(3)</sup>	490		21, 22		
	4690 <sup>(1)</sup>	1	10,000									5.5		
	4688, 4698	6.25	1600			1300	5.5	0.58	7.4	14		35		23, 24
	4689, 4699	10	1000			850	7.5	0.49	6.6	14		55		25, 26
4681, 4691	18.75	530			450	13	0.49	6.1	13	95			27, 28	
4682, 4692	30	330			280	19	0.46	5.5	13	150			29, 30	
4683, 4693	50	200			170	27	0.41	4.9	12	230			31, 32	
4684, 4694	70	140			120	39	0.42	5.0	10 <sup>(3)</sup>	310			33, 34	
4685, 4695	102.08	100		86	51	0.42	4.5	8 <sup>(3)</sup>	390		35, 36			
4686, 4696	131.25	79		68	63	0.40	4.4	6 <sup>(3)</sup>	470		37, 38			
4687, 4697	150	68		59	73	0.41	4.4	6 <sup>(3)</sup>	560		39, 40			
24 V	4690 <sup>(1)</sup>	1	10,000							5.5				
	4688, 4698	6.25	1600		1300	5.5	0.58	7.4	14	35		23, 24		
	4689, 4699	10	1000		850	7.5	0.49	6.6	14	55		25, 26		
	4681, 4691	18.75	530		450	13	0.49	6.1	13	95		27, 28		
	4682, 4692	30	330	0.1	280	19	0.46	5.5	13	150	3.0	29, 30		
	4683, 4693	50	200		170	27	0.41	4.9	12	230		31, 32		
	4684, 4694	70	140		120	39	0.42	5.0	10 <sup>(3)</sup>	310		33, 34		
	4685, 4695	102.08	100		86	51	0.42	4.5	8 <sup>(3)</sup>	390		35, 36		
	4686, 4696	131.25	79		68	63	0.40	4.4	6 <sup>(3)</sup>	470		37, 38		
	4687, 4697	150	68		59	73	0.41	4.4	6 <sup>(3)</sup>	560		39, 40		

Figura 4.19: Datos nominales del Motor pololu 37D. Consultado en [1]

Ahora, tomando en cuenta la reducción, la corriente de salida máxima es de 0.03 Nm, por lo que, cuando en la ecuación 4.1 la velocidad angular  $\omega = 0$ , la ecuación queda de la siguiente manera:



$$R = \frac{k_t V}{\tau_{stall}} \quad (4.4)$$

Sustituyendo en la ecuación:

$$R = \frac{0.1223 * 12}{0.03} = 48.92 \ \Omega \quad (4.5)$$

- Encoder rotativo A38S6-400-2-2-24



Figura 4.20: Encoder rotativo A38S6-400-2-2-24.

Este encoder opera con una tensión de 5-24 V y tiene una resolución de 400 PPR, es ampliamente usado en CNC, elevadores, equipos de impresión, equipos de corte de papel, etc.

- Arduino UNO

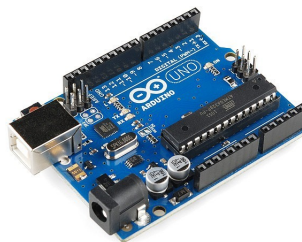


Figura 4.21: Arduino UNO.



Arduino UNO es una placa de microcontrolador de uso fácil, además, es de bajo costo y es altamente flexible. Otra de las ventajas de usar esta placa es que es multiplataforma, ya que el software de Arduino funciona en los sistemas operativos Windows, Macintosh OSX y Linux.

Los demás elementos se mantuvieron iguales (chumaceras, eje, controlador L298M y barra de nylamid), el elemento de sujeción queda abierto a ser cambiado por una nuez como la que se muestra en la figura 4.22



Figura 4.22: Nuez universal.

El cableado fue colocado en un gabinete de plástico y se ocupó una fuente de 12-24 V, junto con un regulador de tensión con salida a 12 V. El mecanismo final se muestra en la figura 4.23.



Figura 4.23: Péndulo simple alternativo.

Se realizó la sintonización de este péndulo y las ganancias seleccionadas son:  
 $k_p = 7.039759626$ ,  $k_i = 0$ ,  $k_d = 1.068451969$ .

Los resultados a diferentes ángulos son los siguientes:

$\pi$

Figura 4.24: Péndulo simple alternativo a referencia  $\pi$ .





En la figura 4.24, se puede observar el péndulo en su posición de equilibrio inestable, se puede notar como alcanza la posición de referencia, que para este caso es de pi rad o 180°.

$$\frac{\pi}{2}$$

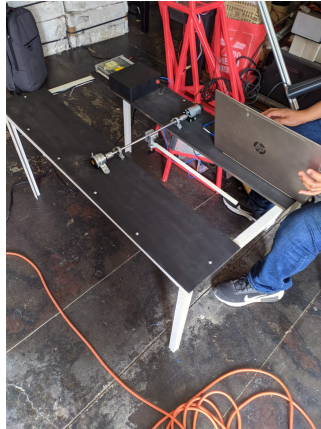


Figura 4.25: Péndulo simple alternativo a referencia  $\frac{\pi}{2}$ .

En la figura 4.25, el péndulo logro llegar a la referencia de 90° o  $\frac{\pi}{2}$ , a pesar de tener las mismas ganancias colocadas para la referencia anterior, esto nos indica que el control es robusto.

$$\frac{\pi}{4}$$



Figura 4.26: Péndulo simple alternativo a referencia  $\frac{\pi}{4}$ .



En la figura 4.26, el péndulo logro llegar a la referencia de  $45^\circ$  o  $\frac{\pi}{4}$ , esto nos indica la calidad del control aplicado al sistema, puede funcionar varios ángulos.

Por último, se opto por la contrucción de esta alternativa debido a la imprecisión del control con los componentes seleccionados, pues, los errores de lectura del encoder con el que cuenta el motor 36GP,es muy rápido y no se logra a tomar lectura de todos los conteos de una revolución provocando impresiones en la posición final del péndulo, pues si bien el control indica que se lleo a la posición deseada, visualmente se puede observar lo contrario, y aunque este problema ocurre solo ocurrio el 20 % de las ejecuciones, es importante que esto no ocurra, pues de acuerdo con el nivel de compromiso de la solución seleccionada, se debería minizar el error. Con esta alternativa, se pudo sonseguir una correcta lectura del encoder externo al motor pololu, con el uso de la biblioteca de encoder.



---

## Conclusiones

---

Los problemas de sintonización también pueden ser tratados como problemas de optimización y a su vez, los métodos metaheurísticos son procedimientos ampliamente usados en problemas de la misma naturaleza como técnica que brinda buenos resultados.

En el presente trabajo se presenta una propuesta de un problema de diseño abordado como un problema de optimización multiobjetivo para la sintonización de controladores PID, el cual permita obtener un conjunto óptimo del cual se pueda extraer una solución que cumpla los compromisos planteados en el problema de acuerdo con las necesidades de la aplicación.

Para poder cumplir la propuesta, se modelaron los sistemas de péndulo simple, péndulo invertido, y péndulo doble. Se seleccionó la ecuación de Euler-Lagrange y se ordenaron los términos en las matrices que describen la dinámica no lineal de los robots, representando los sistemas en matrices de Inercia, Fuerza centrípeta y de Coriolis, y matriz de gravedad, la representación en espacio de estados mediante estas matrices facilita la solución del planteamiento del problema porque no es necesario resolver todo el sistema para obtener el modelo final.

El método de integración logra resolver la ecuación diferencial planteada, pues la ecuación en espacios de estados permite representar a un sistema de orden mayor



en un sistema de orden uno, que, al integrarlo completo, se obtiene la solución del mismo. Debido a su facilidad de interpretación e implementación, y su buena aproximación, el método de integración seleccionado es el método de Euler, el cual mostró buenos resultados en las pruebas y simulaciones realizadas.

La estructura del controlador implementada fue la estructura PID, con la intención de no descartar ninguna acción en los sistemas de péndulo simple y péndulo invertido. En el caso del péndulo doble, el problema se enfoca en el seguimiento de trayectoria, para este caso, la literatura recomienda usar una estructura de controlador PD+.

Una vez seleccionada la estructura del controlador del sistema, lo ideal es seleccionar las variables de diseño, es decir, las variables que se buscarán a través del algoritmo de optimización, pues así lo recomienda el proceso de planteamiento de un problema de diseño como un problema de optimización.

Las variables de diseño seleccionadas fueron las ganancias del controlador  $(k_p, k_d, k_i)$ . Esto es crucial para la sintonización, pues el buscar las ganancias que permitan a la señal de control cumplir con los compromisos deseados, es en esencia el proceso de sintonización.

La verificación de las ganancias obtenidas mediante las primeras pruebas del algoritmo de optimización indica que estas ganancias ofrecen una buena respuesta en la simulación dinámica.

Una vez recopilada esta información, se plantea el problema de sintonización como uno de optimización multiobjetivo.

La implementación de Evolución Diferencial tuvo un resultado exitoso cuando este se probó con el problema de optimización planteado para la sintonización del controlador del péndulo simple, obteniendo un conjunto de soluciones en la aproximación al frente de Pareto y seleccionando una solución para la simulación dinámica del sistema antes mencionado.

Cabe mencionar que el costo computacional (número de evaluaciones del algoritmo de optimización) fue bajo en relación con el buen resultado obtenido, pues se pudo notar que, a mayor número de evaluaciones, la aproximación al frente de Pareto era

más precisa y con una mayor diversidad en sus soluciones. El tiempo de ejecución consumido por los programas podría ser menor si los algoritmos evolutivos fueran escritos en C, debido a que es un lenguaje compilado. Sin embargo, Python es una gran alternativa debido a su simplicidad y rápida lectura, ya que es un lenguaje interpretado. Además, cuenta con una gran comunidad y variedad de bibliotecas que son de gran ayuda al momento de programar. Como ejemplo de esto, está PySimpleGUI, biblioteca utilizada en este documento para la creación de la interfaz gráfica, ya que de entre la variedad de bibliotecas en Python orientadas al desarrollo de GUI's, PySimpleGUI resulta ser la más rápida y sencilla de aprender e implementar.

Un indicador para medir la calidad del frente de Pareto es el Hipervolumen, ya que los algoritmos evolutivos basados en este han demostrado un buen rendimiento en la resolución de problemas de optimización multiobjetivo, tal como se puede consultar en la literatura.

Debido a lo antes mencionado, este indicador resulta adecuado para evaluar la calidad de los frentes de Pareto obtenidos en la solución de los problemas planteados. Dado a que la naturaleza de los algoritmos evolutivos es no determinista, es decir, son algoritmos estocásticos, es necesario interpretar los datos que arroja el indicador de Hipervolumen con alguna técnica estadística no paramétrica. Dentro de la variedad de técnicas existentes, la que más se ajusta con las necesidades de este trabajo, es la prueba de signo de Wilcoxon.

El algoritmo de Evolución Diferencial fue el que mostró mejor calidad en sus aproximaciones al frente de Pareto, de acuerdo con la prueba de Wilcoxon, ya que en cada comparación con GA o con PSO, resultó ganador en la prueba de signo en la que el nivel de significancia para la validez de la hipótesis alternativa era de 5 %, con excepción de una vez, en la que se comparó con GA en el experimento del péndulo simple y el *p-value* fue del 32.84 %. Estos resultados coinciden con diversas publicaciones en las que menciona que el uso de DE ha ido en aumento debido a su excelente desempeño para optimización mono objetivo y multiobjetivo.



Es importante resaltar que esto no quiere decir que GA y PSO son deficientes, de hecho, el tiempo de ejecución consumido por GA fue el más bajo, por lo que en aplicaciones en las que el tiempo es un factor con gran peso, es conveniente implementar un Algoritmo Genético. Por otra parte, también se pudo notar que las aproximaciones al frente de Pareto obtenidas mediante PSO, mostraban soluciones con mayor proximidad entre ellas, presentando una buena distribución.

Una correcta caracterización del sistema conduce a buenos resultados, por lo que es importante identificar los parámetros del sistema, ya que los resultados en simulaciones pueden variar a los obtenidos en una planta real. Buscando reducir errores de medición que se encontraron en las primeras pruebas, se acordó realizar una variación al diseño original del péndulo simple, resultando conveniente probar un motor, encoder y tarjeta diferente al propuesto inicialmente.

Derivado del cambio de motor con una relación de velocidad mayor a la del motor seleccionado, se logró incrementar el torque brindado por el mismo, de manera que, la velocidad en el eje de salida se redujo. Al reducir la velocidad de la señal proveniente del encoder, y al mismo tiempo su resolución, se pudo cambiar el microcontrolador por uno con una tasa de muestreo menor. Lo antes mencionado permitió eliminar el problema de la inercia de la barra cuando se encontraba en sentido contrario a la gravedad, pues el motor seleccionado en muchas ocasiones fue vencido por esta fuerza. En consecuencia, se pudo mejorar la precisión en la medición de posición del péndulo.

En conclusión, fue necesaria la experimentación con los componentes reales para lograr una solución adecuada al problema planteado, así como la caracterización de cada componente para lograr un correcto funcionamiento del control implementado.

---

## Referencias

---

- [1] *Metal Gearmotor 37D*, Pololu, 05 2022. [Online]. Available: <https://www.pololu.com/category/116/37d-metal-gearmotors>
- [2] M. G. Villarreal-Cervantes and J. Alvarez-Gallegos, “Off-line pid control tuning for a planar parallel robot using de variants,” *Exupert Syst. Appl*, vol. 64, pp. 444–454, 2016.
- [3] A. Eiben and J. E. Smith, “Introduction to evolutionary computing,” 2003.
- [4] P. Fleming and R. Purshouse, “Evolutionary algorithms in control systems engineering: A survey,” *Control Eng. Pract*, vol. 10, pp. 1223–1241, 2020.
- [5] G. Reynoso-Meza, X. Blasco, J. Sanchis, and M. Martínez, “Controller tuning using evolutionary multi-objective optimisation: Current trends and applications,” *Control Eng. Pract*, vol. 28, pp. 58–73, 2014.
- [6] A. E. Ruano, S. S. Ge, T. M. Guerra, F. L. Lewis, J. C. Principe, and M. Colnarie, “Computational intelligence in control,” *Annu. Rev. Control*, vol. 38, pp. 233–242, 2014.





- [7] “Aislación sísmica edificio de gobernantes (análisis comparativo de comportamiento y costos con un edificio tradicional),” *Universidad central del Ecuador, Facultad de ingeniería, ciencias físicas y matemática*, 2014.
- [8] R. Yáñez and O. Morales, “El péndulo doble como ejemplo de sistema caótico.”
- [9] M. Cruz and D. Oscar, “Simulación de un sismo mediante el movimiento de un péndulo doble,” *Instituto Tecnológico de Matamoros*.
- [10] D. J. Block, K. J. Åström, and M. W. Spong, “The reaction wheel pendulum,” vol. 1, pp. 112–114, 2007.
- [11] U. I. de Valencia, 2017. [Online]. Available: <https://www.universidadviu.com/los-lenguajes-programacion-alto-nivel/>
- [12] L. Kumar, P. Kumar, and D. Narang, “Tuning of fractional order  $pi^{\lambda}d^{\mu}$  controllers using evolutionary optimization for pid tuned synchronous generator excitation system,” *IFAC-PapersOnLine*, vol. 51, no. 4, pp. 859–864, 2018.
- [13] G. Reynoso-Meza, H. Sánchez, and V. Ribeiro, “Control of refrigeration systems based on vapour compression using multi-objective optimization techniques,” vol. 51, no. 4, pp. 722–724, 2018.
- [14] Z. Chen, X. Yuan, B. Ji *et al.*, “Design of a fractional order pid controller for hydraulic turbine regulating system using chaotic nondominated sorting genetic algorithm ii,” vol. 84, pp. 390–404, 2014.
- [15] Y. Tian, Q. Wang, Y. Wang *et al.*, “A novel design method of multiobjective robust pid controller for industrial process,” in *2014 9th IEEE Conference on Industrial Electronics and Applications*. Hangzhou, China: IEEE, June 2014, pp. 242–246.
- [16] N. Yegireddy and S. Panda, “Design and performance analysis of pid controller for an avr system using multi-objective non-dominated shorting genetic



- algorithm-ii,” in *2014 International Conference on Smart Electric Grid (ISEG)*. Guntur, India: IEEE, September 2014, pp. 1–47.
- [17] S. Panda, “Multi-objective pid controller tuning for a facts-based damping stabilizer using non-dominated sorting genetic algorithm-ii,” vol. 33, no. 7, pp. 1296–1308, 2011.
- [18] Z. Chen, X. Yuan, B. Ji *et al.*, “Application of multiobjective controller to optimal tuning of pid gains for a hydraulic turbine regulating system using adaptive grid particle swam optimization,” vol. 56, pp. 173–187, 2015.
- [19] M. Villarreal-Cervantes, A. Rodríguez-Molina, Peñaloza-Mejía *et al.*, “Multi-objective on-line optimization approach for the dc motor controller tuning using differential evolution,” pp. 20 393—20 407, 2017.
- [20] M. Hung, L. Shu, S. Ho *et al.*, “A novel intelligent multiobjective simulated annealing algorithm for designing robust pid controllers,” vol. 38, no. 2, pp. 319—330, 2008.
- [21] A. Rodríguez-Molina, M. Villarreal-Cervantes, , Mezura-Montes *et al.*, “Adaptive controller tuning method based on online multiobjective optimization: A case study of the four-bar mechanism,” pp. 1—14, 2019.
- [22] D. G. Fernández, “Construcción de un regulador de watt para el control de un motor de corriente continua,” Ph.D. dissertation, Universidad Politécnica de Cartagena, Cartagena, España, 4 2019.
- [23] C. Lagos. Introducción histórica del control automático. [Online]. Available: <http://www.emb.cl/electroindustria/articulo.mvc?xid=474&ni=introduccion-historica-del-control-automatico&fbclid=IwAR0s7kkYXtjVCypIMXaEhgHFY9KSr5XWckMQL907F1L4mFrsKThSpa7AAuo>
- [24] C. de Prada. Reguladores pid. [Online]. Available: <https://www.eii.uva.es/~prada/Pid.pdf>



- [25] M. H. C. Botero, “Optimización multiobjetivo aplicada a problemas de planificación de tareas en ambientes de computo heterogéneo,” Ph.D. dissertation, Universidad de los Andes, Bogotá, 2010.
- [26] K. Ogata, *Ingeniería de control moderna*, 5th ed. Madrid: PEARSON EDUCACIÓN, 2010.
- [27] R. Carmona, “Análisis de estabilidad en controladores pid y neuronales pid sobre robots,” Ph.D. dissertation, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional Unidad Zacatenco, 2013.
- [28] G. Forsythe, M. Malcom, and C. Moler, *Computer Methods for Mathematical Computations*. Prentice-Hall, 1977.
- [29] R. Kent, E. Saff, and A. Snider, *Ecuaciones diferenciales y problemas con valores en la frontera*. Pearson Educación en México, 2005.
- [30] F. Reyes, *Robótica. Control de robots manipuladores*, 1st ed. México: Alfaomega, 2011.
- [31] G. R.-M. and Javier Sanchi and X. B. M. M. mez, “Algoritmos evolutivos y su empleo en el ajuste de controladores del tipo pid: Estado actual y perspectivas,” *Revista Iberoamericana de Automática e Informática industrial*, pp. 251–268, 20123.
- [32] C. Mattson and A. Messac, “Pareto frontier based concept selection under uncertainty, with visualization,” *Optimization and Engineering*, no. 6, pp. 85–115, 2005.
- [33] K. Saridakis and A. Dentsoras, “Soft computing in engineering design – a review,” *Advanced Engineering Informatics*, vol. 22, no. 2, pp. 202–221, 2008.
- [34] Herrera. Introducción a los algoritmos metaheurísticos. [Online]. Available: <https://sci2s.ugr.es/sites/default/>



- files/files/Teaching/OtherPostGraduateCourses/Metaheurísticas/  
Int-Metaheurísticas-CAEPIA-2009.pdf
- [35] A. Rodríguez-Molina, E. Mezura-Montes, M. G. Villarreal-Cervantes, and M. Aldape-Pérez, “Multi-objective meta-heuristic optimization in intelligent control: A survey on the controller tuning problem,” *Applied Soft Computing Journal*, 2020.
- [36] PROMETEC. El módulo controlador de motores l298n. [Online]. Available: <https://www.prometec.net/l298n/>
- [37] R. Storn and K. Price, “Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, pp. 341–359, 1997.
- [38] E. Mezura, M. Reyes, and C. Coello, “Multi-objective optimization using differential evolution: A survey of the state-of-the-art,” *Advances in Differential Evolution (SCI 143)*, pp. 173–196, 2008.
- [39] Das, S. Suganthan, and P. N., “Differential evolution: A survey of the state-of-the-art. evolutionary computation,” *IEEE Transactions on PP (99)*, p. 1 –28, 2010.
- [40] J. Kennedy and R. Eberhart, “Particle swarm optimization. in: Neural networks. proceedings,” *IEEE International Conference*, vol. 4, pp. 1942–1948, 1995.
- [41] C. Coello, “An introduction to multi-objective particle swarm optimizers,” *Soft Computing in Industrial Applications*, vol. 96, pp. 3–12, 2011.
- [42] C. Efrén, “Esquema adaptativo para el manejo de restricciones de límite en problemas de optimización numérica restringida,” Ph.D. dissertation, Universidad Veracruzana, Veracruz, 2019.



- [43] K. Deb, “An efficient constraint handling method for genetic algorithms,” *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2, pp. 311–338, 2000.
- [44] G. Reynozo-Meza, *Controller Tuning with Evolutionary Multiobjective Optimization A Holistic Multiobjective Optimization Design Procedure*. Madrid: Universidad Politécnica de Valencia, 2014.
- [45] V. del Cisne Orozco Orozco, “Desarrollo de una herramienta computacional para la sintonización de parámetros de controladores pid y smc para el seguimiento de trayectoria de un cuadricóptero basado en algoritmos genéticos,” Master’s thesis, Escuela Politécnica Nacional, Quito, Ecuador, 6 2018.
- [46] S. A. C. Goraldo. Índices de desempeño. [Online]. Available: <https://controlautomaticoeducacion.com/control-realimentado/indices-de-desempeno/>
- [47] H. S. S. Corrales, “Multi-objective optimization and multicriteria design of pi/pid controllers,” Ph.D. dissertation, Universidad Autónoma de Barcelona, Barcelona, España, 6 2016.
- [48] S. Jiang, Y. Oong, J. Zhang, and L. Feng, “Consistencies and contradictions of performance metrics in multiobjective optimization,” *IEEE*, vol. 44, no. 12, pp. 2391–2404, 2014.
- [49] E. Alba *et al.* Metaheurísticas multiobjetivo para optimizar el proceso de difusión en manets metropolitanas. [Online]. Available: <https://neo.lcc.uma.es/staff/paco/pdfs/jaem2007.pdf>
- [50] M. L. Bautista, E. V. Rodríguez, L. B. Vargas, and C. C. Hernández, “Pruebas estadísticas paramétricas y no paramétricas: su clasificación, objetivos y características,” *Educación y Salud Boletín Científico Instituto de Ciencias de la Salud*, vol. 9, 2020.



- [51] L. G. Aragón, *Estadística en el área de las Ciencias Sociales y Administrativas*. México: Alfaomega, 2006.
- [52] L. Saaty, *The Analytic Hierarchy Process*. McGraw-Hill, 1980.
- [53] T. Hurtado and G. Bruno, “El proceso de análisis jerárquico (ahp) como herramienta para la toma de decisiones en la selección de proveedores,” Ph.D. dissertation, Universidad Nacional Mayor de San Marcos, Lima, 2005.
- [54] J. Gausemeier and S. Moehring, “Vdi 2206- a new guideline for the design of mechatronic systems,” vol. 35, no. 2, pp. 785–790, 2002.
- [55] Wondershare. Qué es idf - definición, métodos, y beneficios. [Online]. Available: <https://www.edrawsoft.com/es/what-is-idef.html>
- [56] D. J. George, “Concept generation using morphological and option matrices,” Ph.D. dissertation, Clemson University, Carolina del Sur, 2012.
- [57] Seeed. Motor de engranes planetarios (1:14) con encoder. [Online]. Available: <https://www.seeedstudio.com/GP36-Planetary-Geared-Motor-with-Encoder-DC-Motor-1-14-1000CPR-p-4224.html>
- [58] SandoRobotics. Chumacera balero para eje lineal 8mm – kp08 (perpendicular). [Online]. Available: <https://sadorobotics.com/producto/hr0736/>
- [59] O. México. Acoplamiento flexible 8mm x 8mm. [Online]. Available: <https://openbuilds.com.mx/productos/transmision/acme-tornillo-trapezoidal/acoplamiento-flexible-8mm-x-8mm/>
- [60] PJR. Teensy 4.1. [Online]. Available: <https://www.pjrc.com/store/teensy41.html>
- [61] K. Chong and Y. Li, “Pid control system analysis, design, and technology,” *IEEE Trans. Control System Technology*, vol. 13, pp. 559–576, 2005.



- [62] K. Aström and T. Haggglund, *PID Controllers: Theory, Design and Tuning*, 2nd ed. USA: ISA, 1994.
- [63] F. S. Hillier and G. J. Lieberman, *Introducción a la investigación de operaciones*, 9th ed. McGrawHill, 2010.
- [64] J. Holland, “Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence,” *U. Michigan Press*, 1975.
- [65] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 13th ed., USA, 1988.
- [66] U. Rout, R. Sahu, and S. Panda, “Design and analysis of differential evolution algorithm based automatic generation control for interconnected power system,” *Ahin Shams Engineering Journal*, pp. 409–421, 2013.
- [67] J. A. Rodrigo, “Optimización con enjambre de partículas (particle swarm optimization),” 2019. [Online]. Available: [https://www.cienciadedatos.net/documentos/py02\\_optimizacion\\_pso](https://www.cienciadedatos.net/documentos/py02_optimizacion_pso)
- [68] L. Saaty, *The Analytic Hierarchy Process*, 1st ed. New York: McGraw-Hill, 1980.
- [69] B. M. Escárcega, “Control de movimiento de una grúa lineal,” Ph.D. dissertation, Universidad de Sonora, Sonora, 2016.
- [70] I. Ouachani, A. Rabhi *et al.*, “Renewable energy management algorithm for a water pumping system,” *Sustainability in Energy and Buildings*, 2016.
- [71] E. Raffo and E. Ruiz, “Optimización por computación evolucionaria,” *Revista de la Facultad de Ingeniería Industrial*, 2005.



- [72] Bioinformatica. Algoritmos evolutivos para problemas multiobjetivo. [Online]. Available: <https://sci2s.ugr.es/sites/default/files/files/Teaching/GraduatesCourses/Bioinformatica/Tema%2012%20-%20MOEAS.pdf>
- [73] C. Coello and M. Lechuga, "Mopso: a proposal for multiple objective particle swarm optimization," *IEEE*, vol. 2, 2002.





# Apéndice



## .1. Péndulo simple

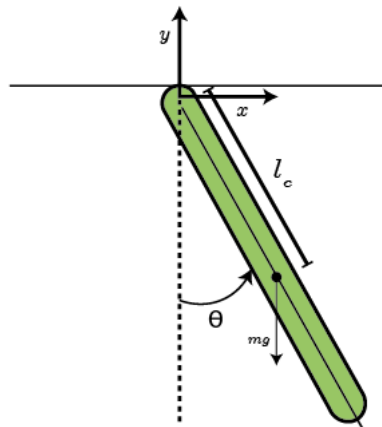


Figura 1: Péndulo simple.

De acuerdo con la Figura 1, los parámetros identificados en la misma se describen en la Tabla 1



Tabla 1: Notación péndulo simple.

Significado	Notación
Longitud al centro de masa	$l_c$
Masa de la barra	$m$
Inercia de la barra	$I$
Posición angular de la barra	$\theta$
Aceleración debida a la gravedad	$g$

La posición de la barra es:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_c \sin \theta \\ -l_c \cos \theta \end{bmatrix}$$

Obteniendo la derivada con respecto del tiempo, la velocidad esta dada de la siguiente manera:

$$v = \frac{d}{dt} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_c \cos \theta \\ l_c \sin \theta \end{bmatrix} \dot{\theta}$$

La velocidad al cuadrado es:

$$v^T v = l_c^2 \dot{\theta}^2 (\cos^2 \theta + \sin^2 \theta) = l_c^2 \dot{\theta}^2$$

Calculando la energía cinética traslacional:

$$K_T = \frac{1}{2} m v^T v = \frac{1}{2} m l_c^2 \dot{\theta}^2$$

La energía cinética de rotación

$$K_R = \frac{1}{2} I \dot{\theta}^2$$

La energía cinética total:

$$K = K_T + K_R = \frac{1}{2} m l_c^2 \dot{\theta}^2 + \frac{1}{2} I \dot{\theta}^2$$



La energía potencial del sistema:

$$U = mgh = -mgl_c \cos \theta$$

Una vez calculadas las energías del sistema, se calcula el Lagrangiano de la siguiente manera:

$$L = K - U$$

$$L = \frac{1}{2}ml_c^2\dot{\theta}^2 + \frac{1}{2}I\dot{\theta}^2 - (-mgl_c \cos \theta) = \frac{1}{2}ml_c^2\dot{\theta}^2 + \frac{1}{2}I\dot{\theta}^2 + (mgl_c \cos \theta)$$

$$D = \frac{1}{2}b\dot{\theta}^2$$

La ecuación del movimiento de Euler-Lagrange

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} + \frac{\partial D}{\partial \dot{\theta}} = \tau$$

$$\frac{\partial L}{\partial \dot{\theta}} = \frac{\partial(\frac{1}{2}ml_c^2\dot{\theta}^2 + \frac{1}{2}I\dot{\theta}^2 + (mgl_c \cos \theta))}{\partial \dot{\theta}} = ml_c^2\dot{\theta} + I\dot{\theta}$$

$$\frac{\partial L}{\partial \theta} = \frac{\partial(\frac{1}{2}ml_c^2\dot{\theta}^2 + \frac{1}{2}I\dot{\theta}^2 + (mgl_c \cos \theta))}{\partial \theta} = -mgl_c \sin \theta$$

$$\frac{\partial D}{\partial \dot{\theta}} = \frac{\partial(\frac{1}{2}b\dot{\theta}^2)}{\partial \dot{\theta}} = b\dot{\theta}$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) = \frac{d}{dt} (ml_c^2\dot{\theta} + I\dot{\theta}) = ml_c^2\ddot{\theta} + I\ddot{\theta}$$

$$ml_c^2\ddot{\theta} + I\ddot{\theta} - (-mgl_c \sin \theta) + b\dot{\theta} = \tau$$

$$ml_c^2\ddot{\theta} + I\ddot{\theta} + (mgl_c \sin \theta) + b\dot{\theta} = \tau$$



$$\ddot{\theta} = \frac{\tau - ((mgl_c \sin \theta) + b\dot{\theta})}{ml_c^2 + I}$$

Representación del modelo en ecuaciones de estado:

Las variables de estado son:  $\theta, \dot{\theta}$

Vector de estado  $\mathbf{x} = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}$

Ecuación de estado:

$$\dot{\mathbf{x}} = f(u, x, t)$$

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\theta} \\ \frac{u - ((mgl_c \sin \theta) + b\dot{\theta})}{ml_c^2 + I} \end{bmatrix}$$

## .2. Péndulo invertido

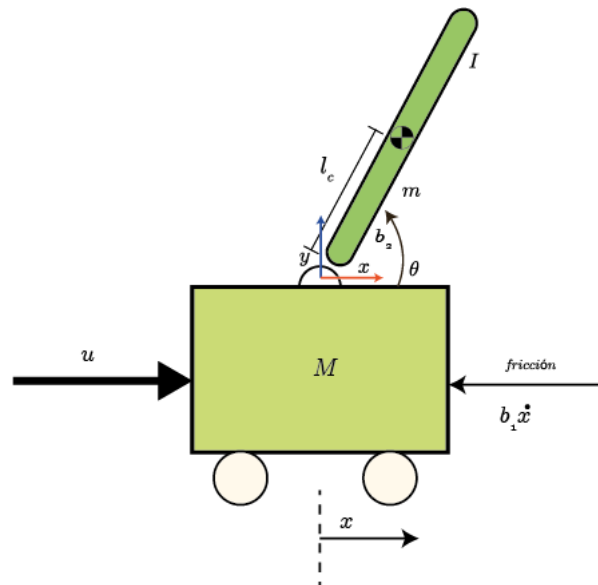


Figura 2: Péndulo invertido.



De acuerdo con la Figura 2, los parámetros identificados en la mismo se describen en el la Tabla 2

Tabla 2: Notación péndulo invertido.

Significado	Notación
Longitud al centro de masa	$l_c$
Masa de la barra	$m$
Masa del carro	$M$
Inercia de la barra	$I$
Posición angular de la barra	$\theta$
Coefficiente de fricción	$b_{1,2}$
Entrada	$u$

Es necesario comenzar a describir el movimiento obteniendo el Lagrangiano. Para la coordenada del centro de masa del brazo se tiene:

$$p = \begin{bmatrix} l_c \cos \theta + x \\ l_c \sin \theta \end{bmatrix}$$

Por lo que la velocidad queda expresada de la siguiente manera:

$$v = \frac{d}{dt} p = \dot{p} = \begin{bmatrix} -l_c \dot{\theta} \sin \theta + \dot{x} \\ l_c \dot{\theta} \cos \theta \end{bmatrix} = -l_c \dot{\theta} \sin \theta + l_c \dot{\theta} \cos \theta + \dot{x}$$

Se calcula la energía cinética del sistema:

$$K = \frac{1}{2} m v^2 + \frac{1}{2} I \dot{\theta}^2 + \frac{1}{2} M \dot{x}^2$$

Donde:

$$v = \begin{bmatrix} -l_c \dot{\theta} \sin \theta + \dot{x} \\ l_c \dot{\theta} \cos \theta \end{bmatrix} = -l_c \dot{\theta} \sin \theta + l_c \dot{\theta} \cos \theta + \dot{x}$$





$$v^2 = (l_c \dot{\theta} \cos \theta)^2 + (-l_c \dot{\theta} \sin \theta + \dot{x})^2 = l_c^2 \dot{\theta}^2 \sin^2 \theta + l_c^2 \dot{\theta}^2 \cos^2 \theta - 2l_c \dot{x} \dot{\theta} \sin \theta + \dot{x}^2 = l_c^2 \dot{\theta}^2 - 2l_c \dot{x} \dot{\theta} \sin \theta + \dot{x}^2$$

$$K = \frac{1}{2} m (l_c^2 \dot{\theta}^2 - 2l_c \dot{x} \dot{\theta} \sin \theta + \dot{x}^2) + \frac{1}{2} I \dot{\theta}^2 + \frac{1}{2} M \dot{x}^2$$

Simplificando

$$K = \frac{1}{2} (M + m) \dot{x}^2 + \frac{1}{2} (I + ml_c^2) \dot{\theta}^2 - ml_c \dot{x} \dot{\theta} \sin \theta$$

Ahora encontramos la energía potencial:

$$U = mgl \sin \theta$$

$$D = \frac{1}{2} b_1 \dot{x}^2 + \frac{1}{2} b_2 \dot{\theta}^2$$

$$L = K - U$$

$$L = \frac{1}{2} (M + m) \dot{x}^2 + \frac{1}{2} (I + ml_c^2) \dot{\theta}^2 + ml_c \dot{x} \dot{\theta} \cos \theta + mgl \cos \theta$$

$$q_1 = x$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{x}} \right) - \frac{\partial L}{\partial x} + \frac{\partial D}{\partial \dot{x}} = F$$

$$(M + m) \ddot{x} + b_1 \dot{x} - ml_c \ddot{\theta} \sin \theta - ml_c \dot{\theta}^2 \cos \theta = u_1$$



$$q_2 = \theta$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} + \frac{\partial D}{\partial \dot{\theta}} = 0$$

$$-ml_c \ddot{x} \sin \theta + (I + ml_c^2) \ddot{\theta} + mgl_c \cos \theta + b_2 \dot{\theta} = u_2$$

Por lo tanto, se llega a las siguientes ecuaciones:

$$(M + m) \ddot{x} + b_1 \dot{x} - ml_c \ddot{\theta} \sin \theta - ml_c \dot{\theta}^2 \cos \theta = u_1$$

$$-ml_c \ddot{x} \sin \theta + (I + ml_c^2) \ddot{\theta} + mgl_c \cos \theta + b_2 \dot{\theta} = u_2$$

$$\ddot{x} = \frac{u_1 - b_1 \dot{x} + ml_c \ddot{\theta} \sin \theta + ml_c \dot{\theta}^2 \cos \theta}{M + m}$$

$$\ddot{\theta} = \frac{u_2 - ml_c \ddot{x} \sin \theta - mgl_c \cos \theta - b_2 \dot{\theta}}{I + ml_c^2}$$

Un sistema mecánico se puede escribir en su forma de espacios de estados, como se muestra a continuación:

$$M(\mathbf{q}) \ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + g(\mathbf{q}) = \mathbf{u}$$

Donde

$M(\mathbf{q}) \ddot{\mathbf{q}}$  es la matriz de inercia

$C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}$  es la matriz de Coriolis y fuerza centrífuga

$g(\mathbf{q})$  es el vector de gravedad

$\mathbf{u}$  es el vector de fuerzas y pares generalizados



$$\begin{bmatrix} M + m & -ml_c \sin \theta \\ -ml_c \sin \theta & I + ml_c^2 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} b_1 & -ml_c \dot{\theta} \cos \theta \\ 0 & b_2 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ mgl_c \cos \theta \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} M + m & -ml_c \sin \theta \\ -ml_c \sin \theta & I + ml_c^2 \end{bmatrix}^{-1} \left( \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} - \begin{bmatrix} b_1 & -ml_c \dot{\theta} \cos \theta \\ 0 & b_2 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} - \begin{bmatrix} 0 \\ mgl_c \cos \theta \end{bmatrix} \right)$$

$$\ddot{\mathbf{q}} = M^{-1}(\mathbf{q}) [\mathbf{u} - C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + g(\mathbf{q})]$$

Variables de estado:  $x, \dot{x}, \theta, \dot{\theta}$

Vector de estado:

$$z = \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}$$

Ecuación de estado

$$\dot{z} = f(u, z, t)$$

$$= \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \theta \\ M^{-1}(\mathbf{q}) [\mathbf{u} - C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} - g(\mathbf{q})] \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \theta \\ \begin{bmatrix} M + m & ml_c \cos \theta \\ ml_c \cos \theta & I + ml_c^2 \end{bmatrix}^{-1} \left( \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} - \begin{bmatrix} b_1 & -ml_c \dot{\theta} \sin \theta \\ 0 & b_2 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} - \begin{bmatrix} 0 \\ mgl_c \sin \theta \end{bmatrix} \right) \end{bmatrix}$$



### .3. Péndulo doble

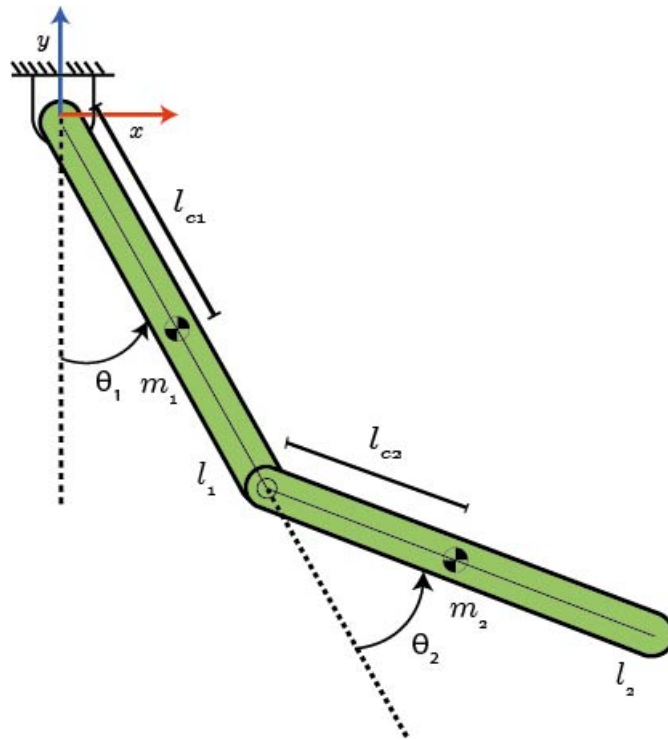


Figura 3: Péndulo doble.

La Figura 3 muestra el péndulo doble, los parámetros mostrados en ella se explican en la Tabla 3

Una vez identificados todos los parámetros, se puede comenzar a describir el modelo que describe el movimiento del sistema, el modelado se realizará con la ecuación de movimiento. Es necesario comenzar a describir el movimiento obteniendo el Lagrangiano. Para la coordenada del centro de masa del brazo 1:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} l_{c1} \sin \theta_1 \\ -l_{c1} \cos \theta_1 \end{bmatrix}$$

Para la coordenada del centro de masa del brazo 2:



Tabla 3: Notación péndulo doble.

Brazo	Descripción	Notación
1	Masa del brazo 1	$m_1$
	Longitud al centro de masa del brazo 1	$l_{c1}$
	Longitud del brazo 1	$l_1$
	Posición angular del brazo 1	$\theta_1$
	Inercia del brazo 1	$I_1$
2	Masa del brazo 2	$m_2$
	Longitud al centro de masa del brazo 2	$l_{c2}$
	Longitud del brazo 2	$l_1$
	Posición angular del brazo 2	$\theta_2$
	Inercia del brazo 2	$I_2$
	Aceleración de la gravedad	$g$

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} l_1 \sin \theta_1 + l_{c2} \sin (\theta_1 + \theta_2) \\ -l_1 \cos \theta_1 - l_{c2} \cos (\theta_1 + \theta_2) \end{bmatrix}$$

La velocidad en ambas coordenadas es:

$$v_1 = \frac{d}{dt} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \frac{d}{dt} \begin{bmatrix} l_{c1} \sin \theta_1 \\ -l_{c1} \cos \theta_1 \end{bmatrix} = \begin{bmatrix} l_{c1} \cos \theta_1 \\ l_{c1} \sin \theta_1 \end{bmatrix} \dot{\theta}_1$$

$$v_2 = \frac{d}{dt} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \frac{d}{dt} \begin{bmatrix} l_1 \sin \theta_1 + l_{c2} \sin (\theta_1 + \theta_2) \\ -l_1 \cos \theta_1 - l_{c2} \cos (\theta_1 + \theta_2) \end{bmatrix} = \begin{bmatrix} l_1 \cos \theta_1 + l_{c2} \cos (\theta_1 + \theta_2) & l_{c2} \cos (\theta_1 + \theta_2) \\ l_1 \sin \theta_1 + l_{c2} \sin (\theta_1 + \theta_2) & l_{c2} \sin (\theta_1 + \theta_2) \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

Calculando la energía cinética del sistema, es necesario calcular el cuadrado de cada una de las velocidades, lo cual se presenta a continuación:

$$v_1^T v_1 = \left( l_{c1} \dot{\theta}_1 \cos \theta_1 \right)^2 + \left( l_{c1} \dot{\theta}_1 \sin \theta_1 \right)^2 = l_{c1}^2 \dot{\theta}_1^2 (\cos^2 \theta_1 + \sin^2 \theta_1) = l_{c1}^2 \dot{\theta}_1^2$$



$$\begin{aligned}
v_2^T v_2 &= (\dot{\theta}_1(l_1 \cos \theta_1 + l_{c2} \cos(\theta_1 + \theta_2)) + \dot{\theta}_2(l_{c2} \cos(\theta_1 + \theta_2)))^2 \\
&\quad + \left( \dot{\theta}_1(l_1 \sin \theta_1 + l_{c2} \sin(\theta_1 + \theta_2)) + \dot{\theta}_2(l_{c2} \sin(\theta_1 + \theta_2)) \right)^2 \\
&= \dot{\theta}_1^2 (l_1 \cos \theta_1 + l_{c2} \cos(\theta_1 + \theta_2))^2 + \dot{\theta}_2^2 ((l_{c2} \cos(\theta_1 + \theta_2))^2 + 2\dot{\theta}_1 \dot{\theta}_2 (l_1 \cos \theta_1 \\
&\quad + l_{c2} \cos(\theta_1 + \theta_2))(l_{c2} \cos(\theta_1 + \theta_2)) + \dot{\theta}_1^2 (l_1 \sin \theta_1 + l_{c2} \sin(\theta_1 + \theta_2))^2 \\
&\quad + \dot{\theta}_2^2 ((l_{c2} \sin(\theta_1 + \theta_2))^2 + 2\dot{\theta}_1 \dot{\theta}_2 (l_1 \sin \theta_1 + l_{c2} \sin(\theta_1 + \theta_2))(l_{c2} \sin(\theta_1 + \theta_2))) \\
&= \dot{\theta}_1^2 l_1 (\cos^2 \theta_1 + \sin^2 \theta_1) + \dot{\theta}_1^2 l_{c2}^2 (\cos^2(\theta_1 + \theta_2) + \sin^2(\theta_1 + \theta_2)) \\
&\quad + 2\dot{\theta}_1^2 l_1 l_{c2} ((\cos \theta_1) (\cos(\theta_1 + \theta_2)) + (\sin \theta_1) (\sin(\theta_1 + \theta_2))) \\
&\quad + 2\dot{\theta}_1 \dot{\theta}_2 (l_1 l_{c2} (\cos \theta_1 \cos(\theta_1 + \theta_2) + \sin \theta_1 \sin(\theta_1 + \theta_2))) \\
&\quad + l_{c2}^2 ((\cos^2(\theta_1 + \theta_2) + \sin^2(\theta_1 + \theta_2))) + \dot{\theta}_2^2 l_{c2}^2 (\cos^2 \theta_1 + \sin^2 \theta_1)
\end{aligned}$$

$$v_2^T v_2 = \dot{\theta}_1^2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(\theta_2)) + 2\dot{\theta}_1 \dot{\theta}_2 (l_1 l_{c2} \cos(\theta_2) + l_{c2}^2) + \dot{\theta}_2^2 l_{c2}^2$$

La energía cinética total del sistema es:

$$\begin{aligned}
K &= \frac{1}{2} \left( m_1 v_1^T v_1 + I_1 (\dot{\theta}_1^2) + m_2 (v_2^T v_2) + I_2 (\dot{\theta}_1 + \theta_2)^2 \right) \\
&= \frac{1}{2} \left( m_1 l_{c1}^2 \dot{\theta}_1^2 + I_1 (\dot{\theta}_1^2) + m_2 (\dot{\theta}_1^2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(\theta_2)) + 2\dot{\theta}_1 \dot{\theta}_2 (l_1 l_{c2} \cos(\theta_2) + l_{c2}^2) + \dot{\theta}_2^2 l_{c2}^2 + I_2 (\dot{\theta}_1 + \theta_2)^2) \right)
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2} \left( \dot{\theta}_1^2 (m_1 l_{c1}^2 + I_1 + I_2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(\theta_2))) + \right. \\
&\quad \left. \dot{\theta}_2^2 (m_2 l_{c2}^2 + I_2) + 2\dot{\theta}_1 \dot{\theta}_2 (m_2 l_1 l_{c2} \cos(\theta_2) + m_2 l_{c2}^2 + I_2) \right)
\end{aligned}$$



$$U = g(m_1 y_1 + m_2 y_2) = -g(m_1 l_{c1} \cos \theta_1 + m_2 (l_1 \cos \theta_1 + l_{c2} \cos (\theta_1 + \theta_2)))$$

A continuación, se presenta el Lagrangiano:

$$L = K - U = \frac{1}{2} \left( \dot{\theta}_1^2 (m_1 l_{c1}^2 + I_1 + I_2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos (\theta_2))) + \right. \\ \left. \dot{\theta}_2^2 (m_2 l_{c2}^2 + I_2) + 2\dot{\theta}_1 \dot{\theta}_2 (m_2 l_1 l_{c2} \cos (\theta_2) + m_2 l_{c2}^2 + I_2 + g(m_1 l_{c1} \cos \theta_1 + m_2 (l_1 \cos \theta_1 + l_{c2} \cos (\theta_1 + \theta_2))) \right)$$

$$D = \frac{1}{2} (m_1 b_1 \dot{\theta}_1^2 + m_2 b_2 \dot{\theta}_2^2)$$

Las ecuaciones de movimiento para la coordenada generalizada  $\theta_1$ :

$$\frac{\partial L}{\partial \dot{\theta}_1} = \dot{\theta}_1 (m_1 l_{c1}^2 + I_1 + I_2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos (\theta_2))) + \dot{\theta}_2 (m_2 l_1 l_{c2} \cos (\theta_2) + m_2 l_{c2}^2 + I_2) \quad (1)$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_1} \right) = \left( \ddot{\theta}_1 (m_1 l_{c1}^2 + I_1 + I_2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos (\theta_2))) + \ddot{\theta}_2 (m_2 l_1 l_{c2} \cos (\theta_2) + m_2 l_{c2}^2 + I_2) \right.$$

$$\left. - 2m_2 \dot{\theta}_1 \dot{\theta}_2 l_1 l_{c2} \sin (\theta_2) - m_2 \dot{\theta}_2^2 l_1 l_{c2} \sin (\theta_2) \right)$$

$$\frac{\partial L}{\partial \theta_1} = -g(m_1 l_{c1} \sin \theta_1 + m_2 (l_1 \sin \theta_1 + l_{c2} \sin (\theta_1 + \theta_2)))$$



$$\begin{aligned} \frac{\partial D}{\partial \dot{\theta}_1} &= \frac{\partial}{\partial \dot{\theta}_1} \left( \frac{1}{2} \left( m_1 b_1 \dot{\theta}_1^2 + m_2 b_2 \dot{\theta}_2^2 \right) \right) = m_1 b_1 \dot{\theta}_1 \\ \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_1} \right) - \frac{\partial L}{\partial \theta_1} + \frac{\partial D}{\partial \dot{\theta}_1} &= \\ \left( \ddot{\theta}_1 (m_1 l_{c1}^2 + I_1 + I_2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(\theta_2))) + \ddot{\theta}_2 (m_2 l_1 l_{c2} \cos(\theta_2) + m_2 l_{c2}^2 + I_2) \right) &+ \\ -2m_2 \dot{\theta}_1 \dot{\theta}_2 l_1 l_{c2} \sin(\theta_2) - m_2 \dot{\theta}_2^2 l_1 l_{c2} \sin(\theta_2) + & \\ m_1 b_1 \dot{\theta}_1 + g (m_1 l_{c1} \sin \theta_1 + m_2 (l_1 \sin \theta_1 + l_{c2} \sin(\theta_1 + \theta_2))) &= \tau_1 \end{aligned}$$

Para la coordenada generalizada  $\theta_2$  :

$$\begin{aligned} \frac{\partial L}{\partial \dot{\theta}_2} &= \dot{\theta}_2 (m_2 l_{c2}^2 + I_2) + \dot{\theta}_1 (m_2 l_1 l_{c2} \cos(\theta_2) + m_2 l_{c2}^2 + I_2) \\ \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_2} \right) &= \ddot{\theta}_2 (m_2 l_{c2}^2 + I_2) + \ddot{\theta}_1 (m_2 l_1 l_{c2} \cos(\theta_2) + m_2 l_{c2}^2 + I_2) - \dot{\theta}_1 \dot{\theta}_2 (m_2 l_1 l_{c2} \sin(\theta_2)) \\ \frac{\partial L}{\partial \theta_2} &= -m_2 \dot{\theta}_1^2 l_1 l_{c2} \sin(\theta_2) - \dot{\theta}_1 \dot{\theta}_2 (m_2 l_1 l_{c2} \sin(\theta_2)) - g m_2 l_{c2} \sin(\theta_1 + \theta_2) \\ \frac{\partial D}{\partial \dot{\theta}_2} &= \frac{\partial}{\partial \dot{\theta}_2} \left( \frac{1}{2} \left( m_1 b_1 \dot{\theta}_1^2 + m_2 b_2 \dot{\theta}_2^2 \right) \right) = m_2 b_2 \dot{\theta}_2 \\ \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_2} \right) - \frac{\partial L}{\partial \theta_2} + \frac{\partial D}{\partial \dot{\theta}_2} &= \\ \ddot{\theta}_2 (m_2 l_{c2}^2 + I_2) + \ddot{\theta}_1 (m_2 l_1 l_{c2} \cos(\theta_2) + m_2 l_{c2}^2 + I_2) + & \\ m_2 \dot{\theta}_1^2 l_1 l_{c2} \sin(\theta_2) + m_2 b_2 \dot{\theta}_2 + g m_2 l_{c2} \sin(\theta_1 + \theta_2) &= \tau_2 \end{aligned}$$





$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = u$$

Donde:

$M(q)\ddot{q}$  es la matriz de inercia

$C(q, \dot{q})\dot{q}$  es la matriz de Coriolis y fuerza centrífuga

$g(q)$  es el vector de gravedad

$u$  es el vector de fuerzas y pares generalizados

Agrupando los términos:

$$M = \begin{bmatrix} m_1 l_{c1}^2 + I_1 + I_2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(\theta_2)) & m_2 l_1 l_{c2} \cos(\theta_2) + m_2 l_{c2}^2 + I_2 \\ m_2 l_1 l_{c2} \cos(\theta_2) + m_2 l_{c2}^2 + I_2 & m_2 l_{c2}^2 + I_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix}$$

$$C = \begin{bmatrix} -2m_2 \dot{\theta}_2 l_1 l_{c2} \sin(\theta_2) + m_1 b_1 & -\dot{\theta}_2 m_2 l_1 l_{c2} \sin(\theta_2) \\ m_2 \dot{\theta}_1 l_1 l_{c2} \sin(\theta_2) & m_2 b_2 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

$$g = \begin{bmatrix} m_1 l_{c1} \sin \theta_1 + m_2 (l_1 \sin \theta_1 + l_{c2} \sin(\theta_1 + \theta_2)) \\ m_2 l_{c2} \sin(\theta_1 + \theta_2) \end{bmatrix}$$

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

Modelo en espacio de estados:

Variables de estado:  $(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2)$

Vector de estados:

$$x = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$



La ecuación en espacios de estados:

$$\dot{x} = f(x, t, u)$$

$$\dot{x} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ M^{-1}(\mathbf{u} - (C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + g(\mathbf{q}))) \end{bmatrix}$$



---

## Apéndice B: Códigos y planos

---

El código fuente de la interfaz en Python, así como el de los Algoritmos Evolutivos (DE, GA y PSO en su versión multiobjetivo), se encuentra en un repositorio público de Github y es posible acceder a él a través del siguiente enlace: <https://github.com/MarcoValdezC/TT2> o si se prefiere, a través del siguiente QR



Ahí se encontrará un archivo README con la información suficiente para ejecutar el programa principal de la interfaz gráfica.

Así mismo, los planos mencionados en la sección *Adaptación del péndulo simple* se encuentran alojados en el mismo repositorio.

